

Міністерство освіти і науки України
Чернівецький національний університет імені Юрія Федьковича

Інститут фізико-технічних та комп'ютерних наук
Кафедра програмного забезпечення комп'ютерних систем

ЗВІТ

про виконання лабораторної роботи №8
з курсу “Безпека програм та даних”

Тема: Підсистема реєстрації

Виконали: Неголюк О.О., Ратушняк М.А.

Перевірив: Остапов С.Е.

ЗМІСТ

| | |
|---|----|
| ПРОТОКОЛ РОБОТИ | 3 |
| 1.1. Представлення записів аудиту у базі даних | 3 |
| 1.2. Реалізація аудиту доступу до об'єктів системи | 3 |
| 1.3. Приклад записів аудиту з реальної системи | 7 |
| 1.4. Відображення записів аудиту на клієнтській частині | 9 |
| ВІДПОВІДІ НА КОНТРОЛЬНІ ЗАПИТАННЯ | 11 |
| ВИСНОВКИ | 15 |

ПРОТОКОЛ РОБОТИ

1.1. Представлення записів аудиту у базі даних

Для зберігання записів аудиту створено таблицю `action_logs` з наступними полями:

- `id` — унікальний ідентифікатор запису (автоінкремент);
- `timestamp` — мітка часу виконання дії (встановлюється автоматично при вставці);
- `action` — назва дії (до 60 символів), що відповідає імені функції-обробника запиту;
- `is_success` — логічне поле, що вказує на успішність виконання дії;
- `reason` — текстове поле з деталями виключення (якщо дія не була успішною);
- `user_id` — ідентифікатор користувача, який виконав дію;
- `ip_address` — IP-адреса клієнта, з якої було ініційовано запит (`VARCHAR(45)`, підтримує IPv4 та IPv6).

Для забезпечення цілісності журналу аудиту створено тригери, що блокують операції `UPDATE` та `DELETE` на таблиці `action_logs`. Таким чином, журнал є `append-only` структурою — можливий лише запис нових логів та їх читання.

1.2. Реалізація аудиту доступу до об'єктів системи

Було реалізовано підсистему реєстрації (аудиту), що у вигляді декоратору `@audit`, використовує ім'я функції у якості `action` поля запису аудиту, а також

опрацьовує можливі виключення, наприклад помилки авторизації чи невдала автентифікація, встановлюючи поле `is_success`. Додаткова інформація про виключення (поле `detail` у екземпляра класу `HTTPException`) зберігається у полі `reason` запису аудиту.

Для веб-ресурсів, що потребують автентифікованого суб'єкта, декоратор використовує ідентифікатор користувача (за наявності) і записує у поле `user_id`. Також декоратор автоматично витягує IP-адресу клієнта з об'єкта `Request FastAPI` і зберігає її у полі `ip_address` для відстеження місця походження запитів.

Приклад використання:

```
@router.get("/users/{id}")
@audit()
@authorize(AccessLevel.CONFIDENTIAL)
async def read_user(
    id: Annotated[int, Path()], db: PostgresRunnerDep, subject:
    CurrentSubjectDep
) -> UserResponse:
    return auth_service.get_user_by_id(id, db=db)
```

Для отримання записів аудиту з серверної частини реалізовано GET-ендпоінт `/audit/`, доступний лише для користувачів з рівнем доступу `CONFIDENTIAL`. Ендпоінт приймає два обов'язкових параметри запиту: `start` та `end` — ISO-формат дати, що визначають діапазон часу для вибірки записів.

Endpoint повертає список об'єктів `ActionLog`, кожен з яких містить: часову мітку, назву дії, статус виконання, причину помилки (якщо така була), ім'я користувача та IP-адресу. На клієнтській частині дані доповнюються інформацією

про геолокацію: для кожної унікальної IP-адреси через сервіс ipinfo визначається місто та країна, які відображаються у колонці “Location”.

На рисунку зображено процес обробки запиту користувача у вигляді data flow діаграми:

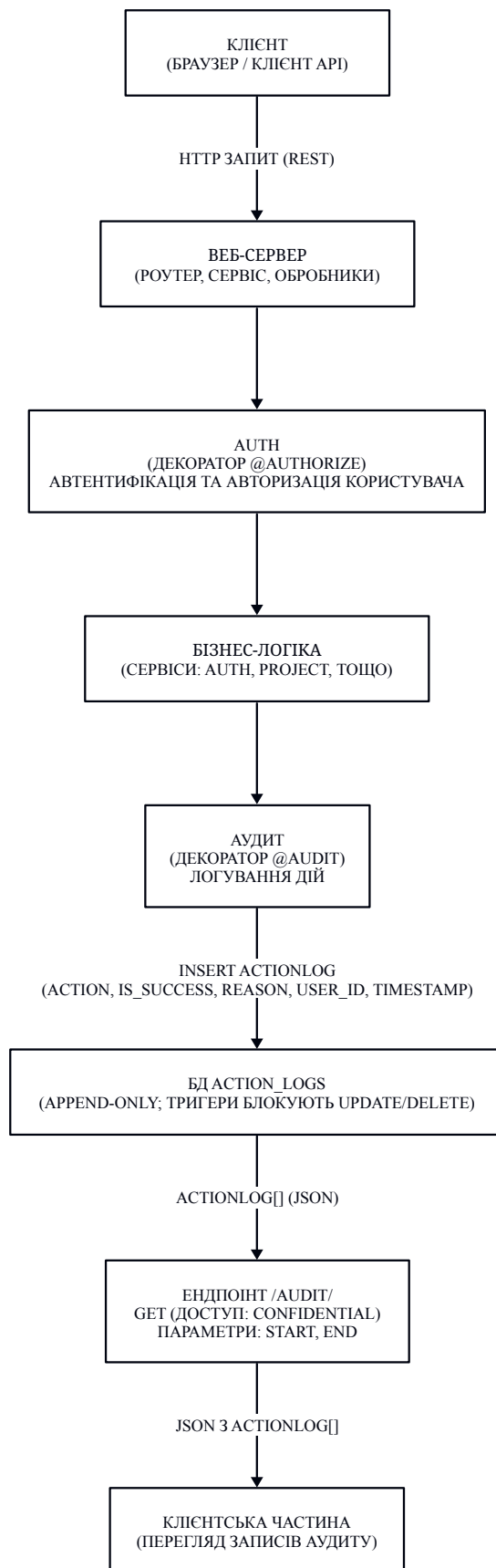


Рисунок 1.1 – Data flow діаграма підсистеми реєстрації

1.3. Приклад записів аудиту з реальної системи

Нижче наведено таблицю з прикладами записів аудиту, що демонструє різні сценарії використання системи:

| Час | Дія | Успіх | Причина | Користувач | IP |
|----------|-------------------------|-------|---|----------------------|---------------|
| 10:41:24 | login_user | ✓ | - | - | 2a02:2378:... |
| 10:41:26 | read_users | ✓ | - | John Week | 2a02:2378:... |
| 10:42:32 | create_user | ✓ | - | John Week | 2a02:2378:... |
| 10:42:33 | read_users | ✓ | - | John Week | 2a02:2378:... |
| 10:43:24 | login_user | ✓ | - | - | 194.44.128.3 |
| 10:43:33 | read_project | ✓ | - | Mykola Ratushniak | - |
| 10:43:52 | create_submission | ✗ | Student is not assigned to this project | Mykola Ratushniak | 194.44.128.3 |
| 10:43:55 | read_submissions | ✗ | Read access forbidden: CONTROLLED < RESTRICTED | Mykola Ratushniak | 194.44.128.3 |
| 10:44:23 | login_user | ✓ | - | - | 2a02:2378:... |
| 10:44:29 | read_project | ✓ | - | Stephen Curry | - |
| 10:44:43 | update_project_students | ✓ | - | Stephen Curry | - |
| 10:44:48 | read_projects | ✓ | - | Stephen Curry | - |
| 10:45:06 | login_user | ✓ | - | - | 2a02:2378:... |
| 10:45:20 | read_users | ✗ | Read access forbidden: CONTROLLED < CONFIDENTIAL | Mykola Ratushniak | 2a02:2378:... |
| 10:45:45 | login_user | ✓ | - | - | 2a02:2378:... |
| 10:46:08 | create_submission | ✓ | - | Mykola Ratushniak | 2a02:2378:... |
| 10:46:10 | read_submissions | ✗ | Read access forbidden: CONTROLLED < RESTRICTED | Mykola Ratushniak | 2a02:2378:... |
| 10:46:36 | login_user | ✓ | - | - | 194.44.128.3 |
| 10:46:48 | read_submission_content | ✓ | - | Sergii Ostapov | 194.44.128.3 |
| 10:46:50 | read_submissions | ✓ | - | Sergii Ostapov | 194.44.128.3 |
| 10:47:14 | get_login_challenge | ✓ | - | - | 194.44.128.3 |
| 10:47:14 | login_user | ✓ | - | - | 194.44.128.3 |

Таблиця 1.2 – Фрагмент журналу аудиту системи NearMuPaper

З таблиці видно типові сценарії:

- **Успішна автентифікація та робота:** Користувачі John Week та Sergii Ostapov входять у систему (login_user) та виконують дозволені операції (read_users, read_submissions, read_submission_content);
- **Адміністративні операції:** Користувач Stephen Curry виконує управління проєктами (read_projects, update_project_students, read_project);
- **Спроби несанкціонованого доступу:** Користувач Mykola Ratushniak з рівнем доступу CONTROLLED намагається прочитати списки користувачів (read_users, рівень CONFIDENTIAL) о 10:45:20 та подання (read_submissions, рівень RESTRICTED) о 10:46:10 та 10:43:55, що призводить до відмови в доступі;
- **Помилки бізнес-логіки:** Користувач Mykola Ratushniak намагається створити подання (create_submission) о 10:43:52, але отримує відмову через відсутність призначення до проєкту;
- **Відсутність користувача при автентифікації:** При операції login_user та get_login_challenge поле User порожнє, оскільки користувач ще не автентифікований;
- **Різні IP-адреси:** Записи містять як IPv4 (194.44.128.3), так і IPv6 (2a02:2378:...) адреси, а деякі записи не містять IP (позначені як "-").

1.4. Відображення записів аудиту на клієнтській частині

Функція завантаження та відображення логів

На клієнтській частині реалізовано екран `audit_catalog_screen`, який відображає логи аудиту у вигляді таблиці з можливістю навігації по датах та експорту даних.

Особливості:

- Завантажує логи за конкретну добу (за замовчуванням — поточну дату);
- Використовує GET-запит до серверного ендпоінту `/audit` з параметрами `start` та `end`, що визначають межі доби;
- Відображає дані у таблиці з колонками: `Timestamp`, `Action`, `Success`, `Reason`, `User`, `IP Address`;
- Реалізовано навігацію між датами через кнопки “Prev Day” та “Next Day”;
- Додано функціонал експорту логів у CSV-файл через кнопку “Export”.

Використання:

- Таблиця показує всі доступні записи аудиту за обрану добу;
- Символи “✓” та “✗” позначають успішні та невдалі дії відповідно;
- Порожні поля (наприклад, відсутній користувач або IP-адреса) відображаються як “-”;
- Довгі повідомлення у полі `Reason` автоматично переносяться;

ВІДПОВІДІ НА КОНТРОЛЬНІ ЗАПИТАННЯ

1. Спробуйте ідентифікувати ознаки тих чи інших порушень безпеки, спираючись на отриманий вами журнал реєстрації. Охарактеризуйте труднощі, що виникають при рішенні цієї задачі.

Аналізуючи журнал реєстрації системи NearMyPaper (Таблиця 1.2), можна ідентифікувати наступні ознаки порушень безпеки:

Спроби несанкціонованого доступу до ресурсів: У наведеному журналі видно, що користувач Mykola Ratushniak (рівень доступу CONTROLLED) тричі намагався отримати доступ до захищених ресурсів: о 10:45:20 до read_users (рівень CONFIDENTIAL) та о 10:46:10 і 10:43:55 до read_submissions (рівень RESTRICTED). Система зафіксувала відмови в доступі з детальним поясненням причини. Такі повторювані спроби протягом 6 хвилин можуть свідчити про навмисну спробу обходу системи контролю доступу.

Множинні невдалі спроби автентифікації: Хоча в наведеному прикладі всі спроби входу успішні, серія записів action=login_user з is_success=false з однієї IP-адреси може свідчити про брутфорс-атаку на облікові записи. Особливо підозрілими є спроби з різними іменами користувачів за короткий проміжок часу.

Доступ до конфіденційних ресурсів з різних IP-адрес: У журналі видно запити від Mykola Ratushniak з різних IP-адрес: спочатку з 194.44.128.3 (Chernivtsi, UA) о 10:43:24, потім з 2a02:2378:... (Netishyn, UA) о 10:44:23, і знову з 194.44.128.3 о 10:46:36. Така зміна IP-адрес за короткий період (6 хвилин) може вказувати на використання VPN або компрометацію облікового запису.

Помилки бізнес-логіки як спроби експлуатації: Користувач Mykola Ratushniak о 10:43:52 намагається створити подання (create_submission) до проєкту,

до якого він не призначений. Це може бути помилкою користувача або спробою експлуатувати вразливість у перевірці прав доступу до проєктів.

Підозрілі адміністративні операції: Користувач Stephen Curry виконує операції з управління студентами проєкту (`update_project_students`) о 10:44:43. Хоча це може бути легітимною адміністративною діяльністю, систематичне внесення змін без бізнес-потреби може свідчити про зловживання привілеями.

Труднощі при ідентифікації порушень:

- **Відсутність контексту поведінки:** Складно відрізнити легітимні помилки користувачів (помилка при введенні даних, забутий пароль) від зловмисних дій без базового профілю нормальної поведінки кожного користувача.
- **Великий обсяг даних:** При активному використанні системи журнал швидко наповнюється записами. У наведеному прикладі лише за 6 хвилин згенеровано 22 записи, що ускладнює ручний аналіз та виявлення аномалій.
- **Відсутність кореляції подій:** Складно встановити зв'язки між різними записами для виявлення складних багатоетапних атак. Наприклад, спроби доступу до `read_submissions` та `read_users` могли бути частиною більшого сценарію атаки.
- **Анонімні запити:** Записи операції `login_user` не містять інформації про користувача (`User = "-"`), оскільки автентифікація ще не завершена. Це ускладнює відстеження зловмисників до успішного входу в систему.
- **Обмежена інформація про контекст:** Журнал не містить інформації про параметри запитів, що унеможливорює детальний аналіз намірів користувача.

2. Проаналізуйте взаємодію підсистеми реєстрації з підсистемами автентифікації та управління доступом в розробленій системі.

Підсистема реєстрації (аудиту) тісно інтегрована з підсистемами автентифікації та управління доступом у системі NearMyPaper через архітектуру декораторів:

Взаємодія з підсистемою автентифікації:

Декоратор `@audit()` застосовується до всіх ендпоінтів, включаючи ті, що вимагають автентифікації. При обробці запиту декоратор отримує об'єкт `Subject` (якщо користувач автентифікований) через залежність `FastAPI CurrentSubjectDep`. Ідентифікатор користувача (`subject.id`) автоматично зберігається у полі `user_id` запису аудиту. Якщо автентифікація не пройшла успішно, декоратор фіксує виключення `HTTPException`, встановлює `is_success=False` та зберігає деталі помилки у полі `reason`, що дозволяє відстежувати невдалі спроби входу.

Взаємодія з підсистемою управління доступом:

Декоратор `@audit()` зазвичай розташовується перед декоратором `@authorize()`, що реалізує контроль доступу на основі рівнів конфіденційності (`PUBLIC`, `INTERNAL`, `CONFIDENTIAL`). Така послідовність забезпечує реєстрацію всіх спроб доступу, включаючи заборонені. Коли `@authorize()` викидає виключення через недостатній рівень доступу, декоратор `@audit()` перехоплює його у блоці `except HTTPException`, фіксує як невдалу операцію і зберігає причину відмови. Це створює повний журнал спроб доступу до захищених ресурсів.

Архітектурні особливості:

Використання декоратора-обгортки дозволяє підсистемі аудиту працювати прозоро для бізнес-логіки, не вимагаючи явних викликів функцій логування у кожному ендпоінті. Незалежність запису логів через окрему транзакцію гарантує

збереження запису навіть при відкаті основної транзакції запиту, що критично для цілісності журналу аудиту при невдалих операціях.

3. Оцініть рівень реєстрації в розробленій системі згідно НД ТЗІ

2.5-004- 99. Що можна зробити, щоб його підвищити?

Відповідно до стандарту НД ТЗІ 2.5-004-99, система NearMyPaper задовольняє наступні критерії рівня **НР-2 (Захищений журнал)**:

- Політика реєстрації, що реалізується КЗЗ, повинна визначати перелік подій, що реєструються;
- КЗЗ повинен бути здатним здійснювати реєстрацію подій, що мають безпосереднє або непряме відношення до безпеки (критерій рівнів НР-4, НР-5)
- Журнал реєстрації повинен містити інформацію про дату, час, місце, тип і успішність чи неуспішність кожної зареєстрованої події. Журнал реєстрації повинен містити інформацію, достатню для встановлення користувача, процесу і/або об'єкта, що мали відношення до кожної зареєстрованої події
- КЗЗ повинен забезпечувати захист журналу реєстрації від несанкціонованого доступу, модифікації або руйнування. Адміністратори і користувачі, яким надані відповідні повноваження, повинні мати в своєму розпорядженні засоби перегляду і аналізу журналу реєстрації

ВИСНОВКИ

У ході виконання лабораторної роботи було реалізовано та вдосконалено підсистему аудиту для системи HearMyPaper. Створено міграцію бази даних для таблиці `action_logs` з тригерами, що забезпечують незмінність записів (`append-only`). Додано поле `ip_address` для відстеження географічного походження запитів. Реалізовано модуль аудиту з репозиторієм, сервісом та роутером для керування записами. Розроблено декоратор `@audit()`, який автоматично логує всі запити до серверу, фіксуючи їх статус, деталі помилок та IP-адресу клієнта. Декоратор застосовано до всіх ендпоінтів системи (`auth`, `project`, `submission`, `pdf_to_audio`, `audit`). Створено GET-ендпоінт для отримання записів аудиту в заданому діапазоні часу з рівнем доступу `CONFIDENTIAL`.

На клієнтській частині реалізовано екран каталогу логів з навігацією по датах та функціоналом експорту даних у CSV-формат. Додано можливість перегляду логів за конкретну добу з відображенням IP-адрес клієнтів. Реалізовано форму експорту з діалогом вибору файлу для збереження результатів аналізу.