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.

Чернівці – 2025

ЗМІСТ

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

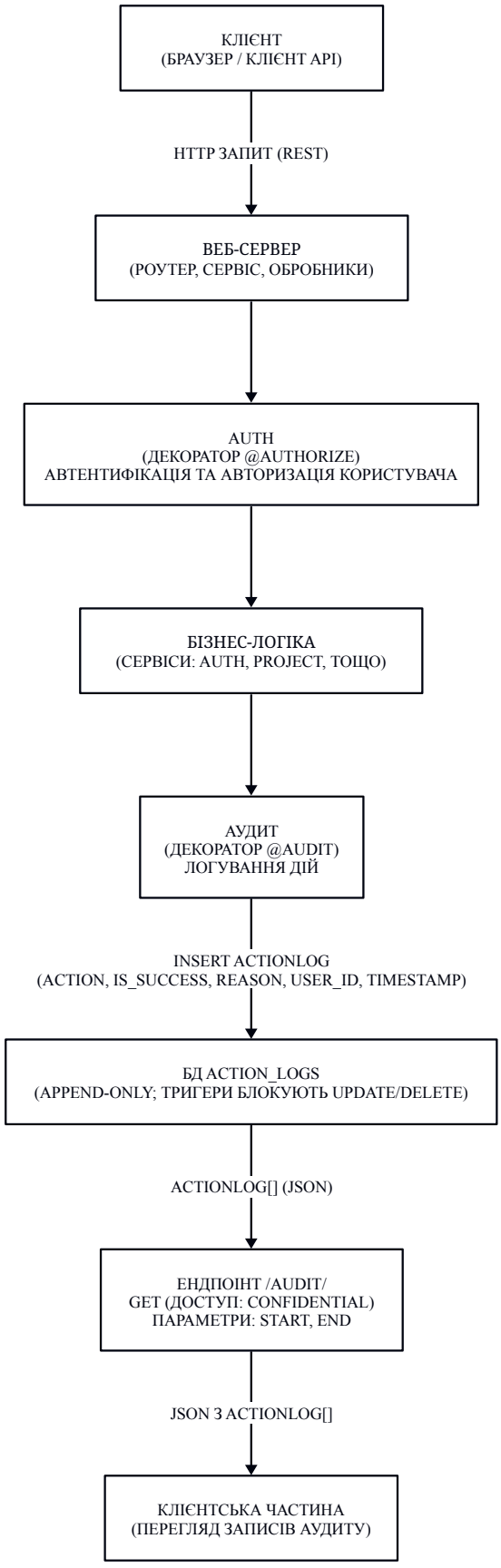

Рисунок 1.1 – Data flow діаграма підсистеми реєстрації

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
08:36:24	login_user	✓	-	-	172.18.0.5
08:36:25	read_audit_logs	✓	-	John Week	172.18.0.5
08:14:52	login_user	✓	-	-	172.18.0.5
08:14:54	read_submissions	✖	Read access forbidden: CONTROLLED < RESTRICTED	Oleksandr Noholiuk	172.18.0.5
08:15:15	login_user	✓	-	-	172.18.0.5
08:15:25	read_audit_logs	✓	-	John Week	172.18.0.5

Рисунок 1.2 – Фрагмент журналу аудиту системи HearMyPaper

З таблиці видно типові сценарії:

- **Успішна автентифікація та робота:** Користувач John Week входить у систему (login_user) та переглядає журнал аудиту (read_audit_logs);
- **Спроба несанкціонованого доступу:** Користувач Oleksandr Noholiuk з рівнем доступу CONTROLLED намагається прочитати подання (read_submissions) з рівнем RESTRICTED, що призводить до відмови в доступі;
- **Відсутність користувача при автентифікації:** При операції login_user поле User порожнє, оскільки користувач ще не автентифікований.

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 HearMyPaper (Рисунок 1.2), можна ідентифікувати наступні ознаки порушень безпеки:

Спроби несанкціонованого доступу до ресурсів: У наведеному журналі видно, що користувач Oleksandr Noholiuk (рівень доступу CONTROLLED) о 08:14:54 намагався отримати доступ до ресурсу read_submissions з рівнем конфіденційності RESTRICTED. Система зафіксувала відмову в доступі з детальним поясненням причини. Якщо такі спроби повторюються систематично, це може 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 незвичайних IP-адрес: У журналі всі запити надходять з IP 172.18.0.5. Якщо раптом з’являться запити від того ж користувача з іншої IP-адреси (особливо географічно віддаленої), це може вказувати на компрометацію облікового запису.

Аномальна активність перегляду журналів: У таблиці видно серію запитів read_audit_logs з інтервалом у 1 секунду (08:36:53-08:37:01). Хоча це може бути навігацією по датах, систематичне інтенсивне читання журналів

може свідчити про спробу приховати сліди зловмисної активності або розвідку системи.

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 23 хвилини згенеровано 26 записів, що ускладнює ручний аналіз та виявлення аномалій.
- **Відсутність кореляції подій:** Складно встановити зв’язки між різними записами для виявлення складних багатоетапних атак. Наприклад, спроба доступу до read_submissions могла 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 унеможливлює 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 HearMyPaper 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 HearMyPaper 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.