

Після того, як кінцеву ціль було виконано, можна також протестувати системи безпеки БД, які виявляють зловмисну присутність у системі шляхом реалізації пентестером заходів приховування присутності у системі.

Останньою фазою є документування та очищення наслідків тестування. Як правило, надані результати включають звіт виконавчого рівня та звіт про технічні висновки. Звіт виконавчого рівня пишеться для потреб керівництва та включає поверхневий огляд оціночної діяльності, її обсягу, найбільш критичних виявлених проблем, загальну оцінку ризиків, сильні сторони безпеки системи та графічна інформація у вигляді знімків екрану. З іншого боку, звіт про технічні висновки повинен включати всі вразливості з деталями щодо того, як відтворити проблему, пов'язані з проблемою ризику активами, рекомендовані дії щодо усунення та корисні довідкові посилання.

Критичним моментом цієї фази є процес відновлення системи до попереднього стану. Усю інформацію, яка була створена та/або зберігалася у системі, слід видалити або приховати, якщо їх усунення через якусь із причин неможливе.

## ВИСНОВКИ

За результатами роботи можна сформулювати наступні висновки:

1. На основі розгляду REST API як найбільш розповсюдженого представника веб API та його аналізу виділено їх основні загрози та вразливості, а також основні механізми забезпечення безпеки API.
2. Запропоновано набір інструментів як ручного, так і автоматизованого тестування заходів безпеки API, наведено практичні приклади їх використання.
3. Сформовано методичні рекомендації по проведенню тестування безпеки API на основі існуючих методологій тестування безпеки інформаційних систем, таких, як OWASP Web Security Testing Guide та ISSAF.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Masse, M. *REST API Design Rulebook*. 2012. 112 p.
2. Webber, J. *REST In Practice*. 2010. 448 p.
3. Madden N. *API Security in Action*. 2020. 576 p.
4. OWASP API Security Project. 2019. [Електронний ресурс] Режим доступу: <https://owasp.org/www-project-api-security/> (Дата звернення: 22.04.2022)
5. ISSAF. *Information Systems Security Assessment Framework*. 2006. 1251 p.
6. OWASP. *OWASP Web Security Testing Guide 4.0*. 2014. 224 p.

**УДК 004.9**

*Крохмалюк В. В., студент IV курсу спеціальності 122 «Комп'ютерні науки»*

*Потапова Н. А., к.е.н, доцент, доцент кафедри інформаційних технологій*

## **ПОБУДОВА КОНЦЕПТУАЛЬНОЇ МОДЕЛІ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ФІНАНСОВИМИ РЕСУРСАМИ ДЛЯ ЗАКЛАДІВ ОСВІТИ**

*Донецький національний університет імені В. Стуса, м. Вінниця*

Інформаційна система - це інтегрований набір компонентів для збору, зберігання та обробки даних, а також для надання інформації, знань та цифрових продуктів. Інформаційні системи дозволяють користувачам збирати, зберігати, упорядковувати та поширювати дані, функції, які можуть служити різноманітним цілям для компаній. Програмне забезпечення будується безпосередньо на апаратному забезпеченні інформаційної системи. Насправді програмне забезпечення — це набір інструкцій, які вказують апаратному забезпеченню, що робити. На відміну від апаратного забезпечення, програмне забезпечення не є відчутним. Ви не можете доторкнутися до нього. Коли програмне забезпечення програмується, насправді даються інструкції, які вказують апаратному забезпеченню, що робити. Тобто інформаційною системою може бути як і база даних, програмний продукт, так і веб додаток, на ньому ми і зупинимся.

Веб-додаток - це прикладна програма, яка зберігається на віддаленому сервері та доставляється через інтернет через інтерфейс браузера. Веб-сервіси за визначенням є веб-програмами, і багато веб-сайтів, хоча й не всі, містять веб-програми. Тобто можна висловитись, що будь-який компонент веб-сайту, який виконує певну функцію для користувача, кваліфікується як веб-програма.

Вибір зупинився саме на веб-додатку по ряду декількох причин.

Перша причина це міжплатформна сумісність. Більшість веб-додатків набагато більш сумісні між різними платформами, ніж традиційне встановлене програмне забезпечення. Зазвичай мінімальною вимогою є веб-браузер, яких існує багато. (Internet Explorer, Firefox, Chrome, Safari – це лише деякі з них). Тож, якщо ви використовуєте Windows, Linux чи Mac OS чи навіть Android, ви все одно можете запускати веб-додаток. Іншими словами, ви можете отримати доступ до веб-програм будь-де за допомогою веб-браузера, а також знімають з розробника відповідальність за створення клієнта, сумісного з певним типом комп'ютера або конкретною операційною системою. Поки браузер сумісний, веб-додатки можуть працювати на кількох платформах незалежно від операційної системи чи пристрою.

Другою причиною є те, що такі системи є більш керовані. Системи розробки веб-додатків необхідно встановлювати лише на сервері, що пред'являє мінімальні вимоги до робочої станції кінцевого користувача. Обслуговувати та оновлювати систему набагато простіше, будь-які оновлення клієнта можна легко розгорнути через веб-сервер, тож веб-програми завжди оновлюються, оскільки оновлення застосовуються централізовано. Усі користувачі мають доступ до однієї версії, тому це усуває будь-які проблеми з сумісністю.

Третя причина це висока можливість розгортання веб-додатку. Завдяки керованості та міжплатформній підтримці розгортання веб-додатків для будь-

якої платформи в будь-якому типі робочого середовища стає легшим. Він ідеально підходить там, де пропускна здатність обмежена, а система та дані віддалені від користувача. Найбільш зручне в розгортанні є те, що вам просто потрібно надіслати користувачеві адресу веб-сайту для входу та надати йому доступ до Інтернету. Вам не потрібно встановлювати його на жорсткий диск, тому це не спричиняє обмеження простору.

Четверта причина - захист даних в реальному часі. Більші і складні системи несуть більше даних і мають окремі системи та джерела даних. У веб-системах ці системи та процеси часто можна консолідувати, зменшуючи потребу в окремих системах. Веб-додатки забезпечують додатковий рівень безпеки, позбавляючи доступу до даних і внутрішніх серверів.

П'ята причина – це зменшені витрати. За допомогою веб-додатків можна значно знизити витрати завдяки зменшенню підтримки та обслуговування, меншим вимогам до системи кінцевого користувача та спрощеній архітектурі.

Також окрім вибору системи, в якій буде оброблятися інформація, потрібно побудувати модель, за якою буде працювати інформаційна система. Виглядати вона буде наступним чином (рис. 1).

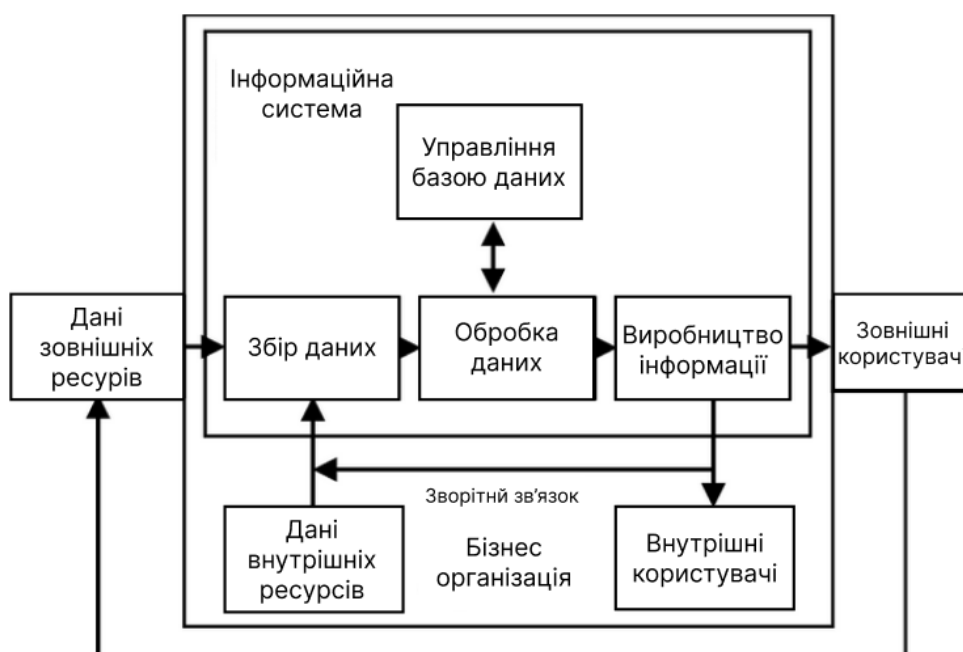


Рисунок 1 – Модель інформаційної системи

Тобто в інформаційну систему будуть потрапляти дані зовнішніх ресурсів, що являють собою числа, показники або ж інформацію, яка буде оброблятися розроблюємим додатком. Далі отримана інформація буде потрапляти в базу даних та сортуватись по відповідним таблицям або ж коміркам. Після чого, над отриманими даними, відповідно до поставленої задачі, будуть проводитися різноманітні операції: сортування в базі даних, видалення оновлення, перезапис, формалізація, фільтрація та транспортування. В результаті чого, буде отримана інформація, яка буде передана як зовнішнім, людям яким необхідні результати роботи інформаційної системи, так і внутрішнім користувачам, які являють собою користувачів для яких і проводиться розробка даного програмного

продукту. Останнім вона надається для її додаткової обробки з метою зробити більш якіснішим процес збору інформації, яка буде оброблятися.

#### **Список літературних джерел**

1. Підгорний А.З., Корольова Т.С., Павлова Т.В., Лапіна І.С. *Проблеми управління фінансами вищих навчальних закладів*, 2017 - 152 с.
2. *Розвиток інформаційних систем управління освітою як інструмент реалізації державної освітньої політики*, 2020 - 259 с.
3. George Reynolds, *Principles of Information Systems (MindTap Course List)*, 14th Edition, 2021 - 610

**УДК 339.9:004.9**

*Лухверчик С. А., студент  
Оврамець І. В., студент  
Зелінська О.В., к.т.н., доцент,  
доцент  
кафедри інформаційних технологій*

### **РОЛЬ УНІВЕРСИТЕТУ В ПІДГОТОВЦІ ІТ-СПЕЦІАЛІСТА**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Нам часто доводиться читати в різних інтернет постах, а також чути з вуст студентів та випускників, що їхні очікування від навчання в університеті не справдилися. Щоб розібратись із цією заявою, потрібно її розглянути з різних боків. Почнемо ми з взаємодії університетів та студентів між собою.

Як на наш погляд головний дисонанс, який за часту описують випускники технічних вузів, пов'язаний із нерозумінням студентів того, що таке навчання у ВУЗ-і в принципі. Перед початком написання роботи, ми підготувалися та розбили свою роботу на кілька етапів, першим з яких було невеличке анонімне опитування студентів другого курсу, Донецького національного університету імені Василя Стуса, що навчаються за спеціальністю «Комп'ютерні науки».

Студентам було задано три простих питання з обмеженими варіантами відповіді.

**1. Яка була Ваша основна мотивація, щодо вступу в університет на спеціальність ІТ?** Це опитування було запущене з метою отримання розуміння, для чого сучасні студенти, взагалі обирають навчання в університеті за даною спеціальністю.