**УДК 004.09**

*Колесов О. О., студент*

# РЕКОМЕНДАЦІЇ ІЗ ТЕСТУВАННЯ БЕЗПЕКИ API СЕРВІСІВ

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

## ВСТУП

Усі клієнт-серверні взаємодії, за допомогою яких здійснюється обмін інформацією в мережі Інтернет, реалізуються за допомогою деякої абстракції – API.

Щоразу, коли користувач відвідує будь-яку сторінку в мережі, він взаємодіє з API віддаленого сервера. В даному випадку, API — це складова частина сервера, яка отримує запити та надсилає відповіді, тому можна сказати, що без такого ключового елементу, як API, існування сучасної всесвітньої веб мережі в сучасному її представленні неможливе. Відповідно, справедливо сказати, що рівень безпеки API серверної частини визначає рівень безпеки усього веб-ресурсу, тому його різностороннє тестування є **актуальним**.

**Метою** роботи є формування структурованого та універсального ряду заходів по тестуванню заходів безпеки API сервісів у вигляді методичних рекомендацій.

**Завданням** роботи є:

- Дослідження поняття API, веб API та REST API як найбільш розповсюдженого представника веб API, аналізу основних загроз та вразливостей, а також механізмів забезпечення безпеки API сервісів.
- Аналіз існуючого набору інструментів та сфери їх застосування у тестуванні заходів безпеки API, наведення практичних прикладів його використання.
- Формування методичних рекомендацій по проведенню тестування безпеки API на основі існуючих методологій тестування безпеки інформаційних систем.

**Об’єктом** роботи є тестування інформаційної безпеки API сервісів.

**Предметом** дослідження є методичні рекомендації та інструментарій з тестування API сервісів.

## 1. ПОНЯТТЯ REST, ОСНОВНИХ ЗАГРОЗ API ТА МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ API

Термін REST (англ. Representational State Transfer – передача репрезентативного стану) вперше з’явився у 2000 році і був описаний вченим Роєм Томасом Філдінгом як стиль веб-архітектури. [1]

Незважаючи на те, що REST — це лише стиль, він значно вплинув на те, як саме програмісти розробляють та реалізують свої API як для постачальників послуг, так і для споживачів.

Веб-сервіси — це веб-сервери, які підтримують потреби сайту чи будь-якої іншої веб програми. Архітектурний стиль REST зазвичай застосовується до проектування API для сучасних веб-сервісів. Веб API, що відповідає архітектурному стилю REST, є REST API. [2]

Згідно проекту OWASP (англ. Open Web Application Security Project – відкритий проект забезпечення безпеки веб-додатків) найбільш серйозними загрозами безпеки API є: [3]

1. Зламана авторизація на рівні об’єкту
2. Зламана автентифікація користувача
3. Надмірне розкриття даних
4. Нестача ресурсів та обмеження швидкості (англ. rate-limiting)
5. Зламана авторизація на функціональному рівні
6. Масове призначення
7. Неправильна конфігурація налаштувань безпеки
8. Ін’єкції
9. Неналежне управління активами
10. Недостатнє ведення журналу та моніторинг

Відповідно, для їх протидії у API сервісах реалізовані наступні заходи забезпечення безпеки:

- Шифрування
- Ідентифікація та автентифікація
- Контроль доступу
- Аудит та логування
- Обмеження швидкості

На рисунку 1.5 [4] зображено п’ять зазначених процесів у вигляді ряду фільтрів, через які проходить запит перед тим як його обробить основна логіка API сервісу.

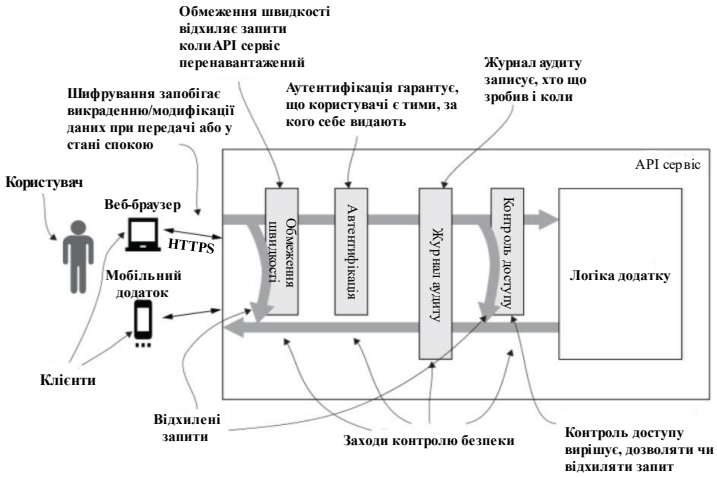


Рис. 1.5. Багатошарова система забезпечення безпеки API сервісу та даних, що ним керуються [4]

## 2. ОГЛЯД ІНСТРУМЕНТАРІЮ ТЕСТУВАННЯ БЕЗПЕКИ API

Для забезпечення безпеки REST API сервісів використовуються інструменти автоматизованого тестування, що дозволяють виявити існуючі вразливості у механізмах, за допомогою яких реалізовується безпека веб API.

Виявляти вразливості можна двома способами: пошук вручну або ж із використанням автоматизованих інструментів. Зазвичай, перший спосіб набагато надійніший і дозволяє більш ефективно досліджувати вразливості системи. Але з іншої сторони, такий процес потребує величезну кількість фінансових та часових ресурсів. Так, ціна ручного тестування комерційного веб API коливається від 1 тис. доларів до 100 тис. і здійснюється напротязі декількох місяців, в залежності від його глибини. Тому значно швидше та дешевше використовувати автоматизовані інструменти тестування веб API.

Для ручного пошуку вразливостей вручну застосовуються такі інструменти: пакет інструментів тестування Burp Suite – Burp Intruder, Burp Proxy та Burp Repeater; JWT-cracker, OAuth.Tools, SQLMap та JMeter.

За допомогою інструментів Burp Suite можна здійснювати перехоплення запитів та відповідей, що надходять до API та аналізувати їх (Burp Proxy), видозмінювати їх таким чином, щоб викрити такі вразливості, як слабкість до бурт-форс атак, атак обходу систем автентифікації та ескалації доступу (Burp Intruder) та надсилати до API безліч разів не розриваючи сесії (Burp Repeater).

В свою чергу, дуже часто для автентифікації та авторизації до API використовуються токени доступу, зазвичай JWT-токени (англ. JSON Web Token), які можуть використовувати слабкий підпис. Цим можуть скористатись зловмисники, зламавши такий підпис і згенерувавши власний токен з будь-якими властивостями, наприклад, з правами доступу адміністратора. Щоб протестувати стійкість шифрування підпису, можна використати інструмент для Kali Linux JWT-cracker.

OAuth.Tools – це інструмент, що дозволяє проаналізувати рівень безпеки та потенційні вразливості використання протоколу OAuth 2.0 для авторизації у REST API сервісах. За допомогою нього можна дослідити безпеку усіх етапів роботи реалізованого на API сервері протоколу.

Дуже частою атакою на API сервіси є ін’єкції різних типів. Так, це можуть бути SQL, NoSQL, XML, LDAP та ін. Але, найрозповсюдженішою є саме SQL ін’єкції, так як більшість веб-сервісів використовують реляційні БД для більш ефективної та простої реалізації бізнес логіки. Для тестування вразливостей веб API до SQL ін’єкції, тестувальник може застосувати інструмент SQLMap.

Для автоматизованого тестування зручно використовувати такі інструменти, як Postman, Swagger, Burp Scanner та Katalon Studio. Вони дозволяють здійснювати автоматичне сканування на вразливості різних механізмів захисту API та дозволяють написання автоматизованих тестових сценаріїв на різних мовах програмування.

## 3. ФОРМУВАННЯ УНІВЕРСАЛЬНИХ МЕТОДИЧНИХ РЕКОМЕНДАЦІЙ З ТЕСТУВАННЯ БЕЗПЕКИ API СЕРВІСІВ

Тестування безпеки API сервісів – це непроста, комплексна задача, у якої немає єдиного рішення. Тестувальники повинні звертати увагу на те, які саме механізми захисту задіяні, як саме здійснюється обмін даними між клієнтом та сервером і як реалізовані API з точки зору бізнес логіки та ін.

Метою виконання тестування безпеки API сервісу є зробити API веб-додатків максимально захищеними, щоб запобігти розголошенню конфіденційної інформації та компрометації веб серверу.

Ціллю тестування є реалізація ряду відомих загроз API сервісів у тестовому середовищі, щоб виявити існуючі вразливостей і у результаті їх аналізу розробити відповідні заходи захисту з ціллю зменшення ризику компрометації API.

У якості основи для створення методології було використано методології тестування, запропоновані у таких фреймворках, як фреймворк оцінювання безпеки інформаційних систем ISSAF (англ. Information System Security Assessment Framework) та посібник з тестування веб-безпеки OWASP WSTG (англ. OWASP Web Security Testing Guide).

ISSAF – це набір методологій, що включає в себе оцінку різних компонент інформаційних систем, починаючи від безпеки хостових систем і закінчуючи оцінкою безпеки веб-додатків. [5]

Посібник з тестування веб-безпеки OWASP є вичерпним посібником з тестування безпеки веб-програм і веб-сервісів. [6]

Процес тестування безпеки будь-якого веб API можна розділити на три послідовних фази (Рис. 3.1.)

- Фаза збору інформації
- Фаза тестування безпеки API
- Аналіз та документування результатів, очищення наслідків

Під час фази збору інформації важливо знайти та проаналізувати усю інформацію про цільове API за допомогою як технічних (прослуховування портів, сканери на вразливості, аналіз структури пакетів запитів та відповідей), так і нетехнічних методів, яку тільки можливо.

По закінченню фази збору інформації, пентестер повинен виконати ряд наступних завдань:

- Зрозуміти механізм автентифікації API сервісу та дослідити потік даних між клієнтом та API сервером;
- Дослідити структуру запитів до API;
- Проаналізувати кінцеві точки API та просканувати їх на вразливості.

Мета другої фази тестування – використання інформації, отриманої на протязі першої фази тестування для всебічного тестування безпеки API сервісу та виявлення вразливостей його кінцевих точок.

Головними цілями другої фази є виявлення та експлуатація вразливостей кінцевих точок API за допомогою інструментів як ручного, так і автоматизованого тестування.

Другу фазу тестування в свою чергу, можна поділити на наступні послідовні етапи:

- Тестування механізмів автентифікації та контролю доступу
- Тестування механізмів перевірки вводу
- Тестування механізмів обробки HTTP запитів
- Тестування механізмів обмеження швидкості

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 ризики активам, 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 ВИКОРИСТАННИХ 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*