Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: *Дослідження захищеності веб сайту ТНТУ*

Виконав: студент (ка)
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Новосад В.Д.
підпис (прізвище та ініціали)

Керівник _____ *Лечаченко Т.А.*
підпис (прізвище та ініціали)

Нормоконтроль _____ *Лобур Т.Б.*
підпис (прізвище та ініціали)

Завідувач кафедри _____ *Загородна Н.В.*
підпис (прізвище та ініціали)

Рецензент _____
підпис (прізвище та ініціали)

м. Тернопіль – 2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Новосаду Володимиру Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження захищеності веб сайту ТНТУ

Керівник роботи Лечаченко Тарас Анатолійович, доктор філософії, асисент кафедри кібербезпеки
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «03» 04 2023 року № 4/7-349

2. Термін подання студентом завершеної роботи 15.06.2023

3. Вихідні дані до роботи Вимоги до операційної системи Kali

4. Зміст роботи (перелік питань, які потрібно розробити)
ОБГРУНТУВАННЯ АКТУАЛЬНОСТІ ДОСЛІДЖЕННЯ. Обгрунтування захисту освітніх веб-сайтів. Загрози кібербезпеці освітніх сайтів. Інструменти аналізу захищеності веб-сайтів. Приклади аналізу освітніх платформ. ОГЛЯД ДОСТУПНИХ ІНСТРУМЕНТІВ. Аналіз захищеності освітніх сайтів. Аналіз інструментів, методів та технік дослідження захищеності сайтів. Наведення алгоритму роботи. ТЕСТУВАННЯ ТА АНАЛІЗ ВЕБ-САЙТУ. Підготовка програмного забезпечення до тестування. Демонстрація знайдених вразливостей. Виправлення вразливостей. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ. Долікарська допомога при отруєннях. Природне середовище і його забруднення. ВИСНОВОК

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Тема, мета, задачі. Загрози освітніх сайтів. Інструменти для аналізу захищеності веб сайтів. Огляд алгоритму тестування веб сайтів. Переваги kali linux.
Підготовка програмного забезпечення а) OWASP ZAP. Демонстрація знайдених вразливостей через OWASP ZAP а) перша б) друга в) третя г) четверта ґ) п`ята д) шоста .
Демонстрація знайдених вразливостей через nmap а) перша б) друга в) третя г) четверта.
Демонстрація знайдених вразливостей через nikto а) перша.
Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Пилипець М.І., проф. кафедри МТ		

7. Дата видачі завдання 20.01.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	20.01 – 19.02	*Виконано*
2.	Підбір джерел про тестування захищеності веб сайтів	20.02 – 03.02	*Виконано*
3.	Опрацювання джерел	04.02 – 17.04	*Виконано*
4.	Оформлення розділу «Обгрунтування актуальності дослідження»	18.04 – 23.04	*Виконано*
5.	Оформлення розділу «Огляд доступних інструментів»	24.04 – 28.04	*Виконано*
6.	Оформлення розділу «Тестування та аналіз веб-сайту»	29.04 – 16.05	*Виконано*
7.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	17.05 – 21.05	*Виконано*
8.	Оформлення кваліфікаційної роботи	22.05 – 05.06	*Виконано*
9.	Нормоконтроль	06.06 – 12.06	*Виконано*
10.	Перевірка на плагіат	10.06 – 15.06	*Виконано*
11.	Попередній захист кваліфікаційної роботи	16.06 – 19.06	*Виконано*
12.	Захист кваліфікаційної роботи	22.06.2022	

Студент _____ *Новосад В.Д.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Лечаченко Т.А.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Дослідження захищеності веб сайту ТНТУ// Кваліфікаційна робота ОР «Бакалавр» //Новосад Володимир Дмитрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-41 // Тернопіль, 2022 // С. 72 , рис. – 14, табл. – 0 , кресл. – 0 , додат. – 0.

Ключові слова:, АУДИТ БЕЗПЕКИ, СКАНУВАННЯ ПОРТІВ, АНАЛІЗ ВРАЗЛИВОСТЕЙ, SQL-ІН'ЄКЦІЇ, ПЕРЕХОПЛЕННЯ СЕСІЙ, МІЖСАЙТОВІ СКРИПТОВІ АТАКИ

Ця кваліфікаційна робота присвячений дослідженню захищеності веб-сайту Тернопільського національного технічного університету імені Івана Пулюя (ТНТУ). Метою дослідження є виявлення потенційних вразливостей та визначення рівня захищеності сайту від зловмисних атак. У роботі будуть використані методи аналізу безпеки веб-додатків, включаючи тестування на проникнення, аудит коду, сканування портів та аналіз вразливостей. Крім того, будуть досліджені популярні види атак, такі як SQL-ін'єкції, перехоплення сесій, міжсайтові скриптові атаки та інші. Результати дослідження допоможуть виявити слабкі місця в захисті веб-сайту ТНТУ та розробити рекомендації щодо його подальшого покращення. Очікується, що дана кваліфікаційна робота сприятиме забезпеченню безпеки веб-додатків та захисту конфіденційності користувачів.

ABSTRACT

Research on the security of the TNTU website // Qualification work, Bachelor's degree // Novosad Volodymyr Dmytrovych // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, Group SBs-41 // Ternopil, 2022 // P. 72, Fig. – 14, Table – 0, Drawings – 0, Appendix – 0.

Keywords: SECURITY AUDIT, PORT SCANNING, VULNERABILITY ANALYSIS, SQL INJECTIONS, SESSION HIJACKING, CROSS-SITE SCRIPTING ATTACKS.

This diploma project is dedicated to the investigation of the security of the Ternopil National Technical University (TNTU) website. The aim of the research is to identify potential vulnerabilities and determine the level of protection against malicious attacks on the website. The work will employ methods of web application security analysis, including penetration testing, code auditing, port scanning, and vulnerability analysis. Additionally, popular types of attacks such as SQL injections, session hijacking, cross-site scripting attacks, and others will be explored. The research findings will help uncover weaknesses in the defense of the TNTU website and develop recommendations for its further improvement. It is expected that this work will contribute to ensuring the security of web applications and protecting user confidentiality.

ЗМІСТ

ВСТУП

У сучасному інформаційному суспільстві веб-сайти стали невід'ємною складовою частиною діяльності багатьох організацій і установ. Зростаюча залежність від технологій та збільшення об'єму електронної комунікації зумовили необхідність забезпечення надійного функціонування та захисту веб-сайтів від різноманітних загроз.

Однак, незважаючи на активне застосування веб-технологій в університетському середовищі, питання безпеки веб-сайтів Тернопільського національного технічного університету (ТНТУ) залишається актуальним і потребує дослідження. Веб-сайт ТНТУ виконує важливі функції, такі як інформаційне забезпечення студентів та викладачів та інших важливих матеріалів, що супроводжують і висвітлють роботу університету. Отже, забезпечення високого рівня захищеності цього веб-сайту є необхідною умовою для забезпечення ефективності та довіри до університетської спільноти.

Метою даної кваліфікаційного роботи є дослідження захищеності веб-сайту ТНТУ та виявлення потенційних уразливостей, що можуть бути використані зловмисниками для незаконного доступу до системи, руйнування чи крадіжки конфіденційної інформації. В рамках дослідження будуть розглянуті основні аспекти безпеки веб-сайтів, використані методи аналізу вразливостей та рекомендації щодо підвищення рівня безпеки веб-сайту ТНТУ.

Для досягнення поставленої мети, у роботі будуть використані наукові методи дослідження, такі як аналіз літературних джерел, вивчення нормативно-правової бази, проведення експериментів та аналіз результатів. Результати дослідження дозволять отримати оцінку поточного рівня захищеності веб-сайту ТНТУ та розробити пропозиції щодо підвищення його безпеки.

Отримані в ході дослідження результати та рекомендації стануть основою для подальшого вдосконалення системи безпеки веб-сайту ТНТУ та сприятимуть підвищенню довіри користувачів до цього ресурсу.

У світлі швидкого розвитку кіберзлочинності та загроз безпеці в Інтернеті, дане дослідження має важливе значення для забезпечення безпеки веб-сайту ТНТУ та захисту інформації, що розміщується на ньому. Результати роботи можуть бути корисними для адміністрації ТНТУ, веб-розробників та спеціалістів з інформаційної безпеки, які зацікавлені в покращенні безпеки своїх веб-сайтів та запобіганні потенційним атакам.

Загалом, дане дослідження відіграє важливу роль у поліпшенні безпеки веб-сайту ТНТУ та сприяє розвитку технологічної інфраструктури університету.

1 ОБГРУНТУВАННЯ АКТУАЛЬНОСТІ ДОСЛІДЖЕННЯ

1.1 Обгрунтування захисту освітніх веб-сайтів

Актуальність аналізу захищеності освітніх сайтів полягає в тому, що ці веб-ресурси містять значну кількість важливої інформації, яка потребує належного захисту. Нижче наведені деякі аспекти, які слід враховувати при обгрунтуванні важливості проведення аналізу захищеності освітніх сайтів, зокрема сайту ТНТУ.

Конфіденційність даних є одним з ключових аспектів захищеності освітніх сайтів. Освітні ресурси зберігають різноманітну конфіденційну інформацію, таку як особисті дані студентів і співробітників, академічні досягнення, фінансові дані, результати досліджень та інше.

Якщо освітні сайти не захищені належним чином, ці дані можуть бути скомпрометовані або викрадені. Наслідки такого порушення конфіденційності можуть бути серйозними і впливати на різні сторони:

- ідентифікаційна крадіжка: якщо зловмисники отримують доступ до персональних даних студентів, викладачів або співробітників, вони можуть використовувати ці дані для злочинних цілей, таких як ідентифікаційна крадіжка. Це може включати відкриття фальшивих банківських рахунків, отримання кредитів або здійснення шахрайських дій в ім'я постраждалих осіб;

- порушення приватності: конфіденційна інформація, така як медичні дані студентів, може бути чутливою та особистою. Якщо такі дані стають доступними для незаконних осіб, це може порушити приватність осіб, що може мати негативний вплив на їхню довіру до освітньої установи;

- пошкодження репутації: витік конфіденційної інформації з освітнього сайту може призвести до серйозних наслідків для репутації установи. Публікація негативних новин про витік даних може зменшити довіру до установи серед

студентів, батьків, викладачів та інших зацікавлених сторін, що може вплинути на реєстрацію студентів, пошук фінансування та загальний престиж установи.

Отже, проведення аналізу захищеності освітніх сайтів, зокрема сайту ТНТУ, має велике значення для забезпечення конфіденційності особистих даних, запобігання можливим наслідкам порушення цілісності даних та збереження довіри до освітньої установи. Це можна досягти шляхом впровадження ефективних заходів безпеки, таких як шифрування даних, контроль доступу, аудит безпеки та постійного оновлення програмного забезпечення для запобігання використанню вразливостей.

Цілісність даних

Захищеність освітніх сайтів також важлива для забезпечення цілісності даних. Несанкціонований доступ до сайту може призвести до незаконних змін або втрати даних, що може викликати суттєві проблеми для користувачів. Наприклад, зміна академічних записів студентів або викладачів без їхньої згоди може спричинити серйозні наслідки для осіб, які покладаються на точність цих даних.

Цілісність даних є ще одним важливим аспектом захищеності освітніх сайтів. Вона відноситься до забезпечення того, що дані залишаються недоторканими, незмінними та несфальшованими протягом всього їхнього життєвого циклу.

Коли мова йде про освітні ресурси, цілісність даних є критично важливою, оскільки велика частина інформації, що зберігається на сайті, є важливою та незамінною. Наприклад, результати екзаменів, оцінки студентів, академічні досягнення та інші дані повинні бути безпечними від будь-яких незаконних змін або порушень.

Невідповідність цілісності даних може мати серйозні наслідки. Наприклад, якщо результати екзаменів студентів були змінені без дозволу або якщо академічні досягнення були підроблені, це може призвести до спотворення оцінювання та невірної оцінки успішності студентів. Крім того, порушення

цілісності даних та може підривати довіру до освітньої установи, яка може вплинути на репутацію та відношення до неї.

Для забезпечення цілісності даних освітніх сайтів необхідно використовувати різні заходи захисту. Це включає контроль доступу до системи, забезпечення правильної аутентифікації та авторизації користувачів, застосування механізмів контролю цілісності даних, наприклад хешування або цифрових підписів, а також моніторинг та виявлення будь-яких змін у дані, що можуть відбуватися без дозволу.

Окрім того, важливо мати систему резервного копіювання та відновлення даних, щоб забезпечити можливість відновлення інформації в разі її втрати або пошкодження. Регулярне оновлення програмного забезпечення та виправлення виявлених вразливостей також є важливою складовою забезпечення цілісності даних.

Загалом, забезпечення цілісності даних на освітньому сайті має критичне значення для збереження точності, недоторканості та незмінності інформації. Це допоможе зберегти надійність та довіру до освітньої установи, а також запобігти можливим негативним наслідкам, пов'язаним з порушенням цілісності даних.

Доступність

Вона є ще одним важливим аспектом захищеності освітніх сайтів. Вона відноситься до забезпечення безперебійного доступу користувачів до сайту та його ресурсів.

Доступність має велике значення для освітніх сайтів, оскільки ці ресурси використовуються студентами, викладачами, співробітниками та іншими зацікавленими особами для отримання інформації, виконання завдань, комунікації та інших цілей. Якщо сайт не є доступним, це може призвести до перешкод у навчанні та роботі, втрати продуктивності та незадоволення користувачів.

Загрози, які можуть вплинути на доступність освітнього сайту, можуть бути різноманітними. Наприклад, DDoS-атаки (розподілені атаки з відмовою в

обслуговуванні) можуть спрямовувати велику кількість штучного трафіку на сайт, затруднюючи або блокуючи доступ користувачів. Технічні проблеми з мережею або серверами також можуть призвести до перерв у доступі до сайту.

Для забезпечення доступності освітнього сайту необхідно вживати заходів для запобігання та реагування на подібні загрози. Одним з таких заходів є використання механізмів балансування навантаження, які дозволяють розподіляти трафік між декількома серверами, зменшуючи ризик перевантаження та забезпечуючи безперебійну роботу. Також важливо мати план відновлення після непередбачуваних перебоїв, щоб якомога швидше відновити доступність сайту.

Освітні сайти також повинні мати дизайн, який сприяє легкому навігації та використанню. Інтуїтивний інтерфейс та ефективне розташування інформації допоможуть користувачам знайти необхідну інформацію швидко та без зайвих перешкод. Оптимізація швидкості завантаження сторінок також є важливим аспектом доступності, оскільки довгі часи очікування можуть знеохочувати користувачів та впливати на їхнє задоволення від використання сайту.

Загалом, забезпечення доступності освітнього сайту дозволяє забезпечити безперебійний доступ користувачів до необхідної інформації та ресурсів, сприяє продуктивності та задоволенню користувачів. Для досягнення цієї мети потрібно вживати заходів для запобігання та реагування на загрози, а також розробляти зручний та ефективний інтерфейс для користувачів.

1.2 Загрози кібербезпеці освітніх сайтів

Загрози для освітніх сайтів можуть бути різноманітні і мають потенційно серйозний вплив на їхню безпеку та надійність. Ось деякі типові загрози, з якими стикаються освітні сайти.

Атаки на вразливості веб-додатків.

Атаки на вразливості веб-додатків є одними з найпоширеніших загроз для веб-сайтів, включаючи освітні ресурси. Ці атаки використовуються зловмисниками для отримання несанкціонованого доступу до системи, викрадення конфіденційної інформації, пошкодження або зламу веб-сайту. Деякі типові атаки на вразливості веб-додатків включають:

- XSS (Cross-Site Scripting): XSS-атаки використовуються для впровадження зловідомого скрипта в веб-сторінку, який виконується на браузері користувача. Це може призвести до викрадення сесій, виконання зловмисних дій в контексті автентифікованого користувача або перенаправлення на фальшиві сторінки для збору конфіденційної інформації.

- SQL Injection: це атака, при якій зловмисник вводить шкідливі SQL-запити в веб-додаток, зламуючи безпеку бази даних. Якщо додаток не належним чином перевіряє та екранує дані, введені користувачем, зловмисник може отримати доступ до конфіденційної інформації, змінити або видалити дані з бази даних.

- CSRF (Cross-Site Request Forgery): ця атака використовується для виконання небажаних дій в контексті автентифікованого користувача, коли він відвідує спеціально створену веб-сторінку або клікає на посилання. Зловмисник використовує довір'я користувача до веб-сайту, щоб змусити його виконати певну дію, наприклад, змінити пароль або зробити фінансову транзакцію.

- RFI (Remote File Inclusion) та LFI (Local File Inclusion): ці атаки використовуються для включення вразливих файлів на веб-сервері. RFI включає зовнішні файли, що містять шкідливий код, тоді як LFI використовує локальні

файли на сервері. Це може призвести до виконання шкідливого коду на сервері, отримання доступу до конфіденційних даних або навіть зламу сервера.

- File Upload Vulnerabilities: вразливості, пов'язані з завантаженням файлів, можуть дозволяти зловмисникам виконувати шкідливі дії. Наприклад, недостатньо перевірена функція завантаження може дозволити завантаження шкідливого файлу, який потім може бути виконаний на сервері або завантажений іншими користувачами.

- DDoS-атаки: DDoS (Distributed Denial of Service) атаки є одними з найпоширеніших та найруйнівніших загроз для веб-сайтів, включаючи освітні ресурси. Під час DDoS-атаки зловмисники намагаються перевантажити ресурси веб-сайту, спричиняючи перебої в його роботі та призводячи до недоступності для легітимних користувачів.

Основна ідея DDoS-атак полягає у тому, щоб надіслати велику кількість запитів до веб-сайту одночасно з різних джерел. Це може бути досягнуто шляхом використання ботнетів - мережі комп'ютерів, які підконтрольовані зловмисниками і використовуються для злочинних цілей. Зловмисники можуть використовувати комп'ютери, сервери або навіть підключені пристрої Інтернету речей для запуску DDoS-атак.

DDoS-атаки можуть мати серйозні наслідки для освітніх сайтів:

- недоступність: DDoS-атаки можуть спричинити перебої в роботі веб-сайту, призводячи до тимчасової або повної недоступності для користувачів. Це може мешкати студентам, викладачам і іншим співробітникам використовувати ресурси, які необхідні для навчання, проведення занять або обміну інформацією;

- втрата репутації: Якщо освітній сайт регулярно стикається з проблемами недоступності через DDoS-атаки, це може вплинути на його репутацію. Студенти, викладачі та інші користувачі можуть втратити довіру до сайту і шукати інші ресурси;

- втрата даних: Деякі DDoS-атаки можуть бути використані як прикривний механізм для інших злочинних дій, таких як крадіжка або втрата

даних. Під час атаки зловмисники можуть використовувати вразливості веб-сайту для злому і отримання доступу до конфіденційної інформації.

Одним із способів захисту від DDoS-атак є використання спеціалізованих рішень, таких як фаєрволи, облачні служби захисту або сервіси миттєвої масштабованості. Ці рішення можуть аналізувати трафік, виявляти аномалії та фільтрувати шкідливі запити, забезпечуючи безпеку і доступність веб-сайту.

Усвідомлення загроз DDoS-атак та вжиття заходів для їх запобігання може допомогти освітнім сайтам забезпечити безпеку та надійність своїх ресурсів для користувачів.

Фішингові атаки: Фішингові атаки є формою кіберзлочинності, в якій зловмисники намагаються отримати конфіденційну інформацію, таку як паролі, номери кредитних карток або особисті дані, шляхом вманювання легітимних користувачів виконати певні дії або розкрити свої особисті дані. Фішингові атаки спираються на соціальний інжиніринг та маніпулювання психологічними аспектами, щоб отримати доступ до цінної інформації.

Освітні сайти, включаючи сайт ТНТУ, можуть бути ціллю фішингових атак з різних причин:

- крадіжка облікових даних: Зловмисники можуть створювати підроблені сторінки входу, що нагадують офіційний вигляд освітнього сайту, і надсилати шахрайські посилання користувачам через електронну пошту або соціальні мережі. Користувачі, не підозрюючи небезпеки, можуть вводити свої облікові дані на цих підроблених сторінках, які потім можуть бути використані зловмисниками для незаконного доступу до їх акаунтів;

- розповсюдження шкідливого програмного забезпечення: Фішингові атаки можуть також включати посилання на шкідливе програмне забезпечення, які встановлюються на комп'ютері користувача після переходу по підробленому посиланню. Це може призвести до компрометації безпеки і витоку конфіденційних даних.

Соціальний інжиніринг [4]: Зловмисники можуть використовувати соціальний інжиніринг, щоб надихнути довіру та обманути користувачів освітнього сайту.

Витік інформації [1]: це серйозна загроза для освітніх сайтів і може мати значні наслідки. Витік інформації означає незаконне чи несанкціоноване розголошення конфіденційної, особистої або внутрішньої інформації, яка повинна залишатися в секреті.

Освітні сайти містять різноманітну конфіденційну інформацію, таку як особисті дані студентів, викладачів та співробітників, академічні рекорди, фінансові дані, результати досліджень та інші важливі дані. Витік цих даних може мати наступні наслідки:

- порушення конфіденційності: Витік конфіденційних даних з освітнього сайту може призвести до порушення приватності осіб, чиї дані стали доступними для сторонніх осіб. Це може включати особисту ідентифікаційну інформацію, таку як імена, адреси, номери соціального страхування, а також фінансові дані, які можуть бути використані для шахрайства або крадіжки особистості;
- пошкодження репутації: Витік конфіденційних даних може призвести до пошкодження репутації освітньої установи. Коли користувачі дізнаються про витік, вони можуть втратити довіру до сайту та вважати його ненадійним. Це може мати негативний вплив на рейтинг та статус установи;
- фінансові втрати: Витік фінансових даних, таких як номери кредитних карток чи банківські реквізити, може призвести до фінансових втрат для осіб, чиї дані були скомпрометовані. Це може призвести до крадіжки грошей з їх банківських рахунків або несанкціонованого використання їх фінансових реквізитів;
- втрати інтелектуальної власності: Для освітніх сайтів, що містять результати досліджень або інтелектуальну власність, витік такої інформації може призвести до втрати конкурентного переваги або незаконного використання цих даних сторонніми організаціями;

Усі ці наслідки витоку інформації можуть мати серйозні впливи на освітні установи, постраждалих користувачів та весь освітній сектор в цілому.

Перехоплення сесій.

Перехоплення сесій є одним з типів атак, спрямованих на отримання несанкціонованого доступу до облікових записів користувачів на веб-сайтах. Під час процесу аутентифікації на сайті, користувач отримує унікальну сесійну ідентифікацію або токен, який використовується для ідентифікації і авторизації користувача під час його візиту на сайт.

Процес перехоплення сесій включає отримання цього токену з метою його використання для підробки аутентичного сеансу користувача. Існують різні методи, за допомогою яких зловмисники можуть здійснити перехоплення сесій:

- перехоплення сесій через використання публічних мереж. Якщо користувач виконує аутентифікацію на веб-сайті через незахищену мережу, наприклад, через відкритий Wi-Fi підключення в кафе або громадському місці, злоумисники можуть перехопити передачу даних і отримати доступ до сесійного токена.

- використання атаки Man-in-the-Middle (MITM). У такому випадку злоумисники розміщуються між користувачем і веб-сайтом, підроблюючи комунікацію між ними. За допомогою MITM-атаки, злоумисники можуть перехопити токен сесії та всі обмінені дані, не залишаючись помітними для користувача;

- використання крадіжки сесійних файлів або кукі. Злоумисники можуть отримати доступ до сесійних файлів або кукі, які зберігаються на комп'ютері користувача після аутентифікації. Це може статись, якщо комп'ютер користувача заражений шкідливим ПЗ або якщо використовуються недостатньо безпечні механізми збереження сесійних даних на стороні клієнта.

Наслідки перехоплення сесій можуть бути серйозними. Зловмисники, які успішно перехопили сесійний токен, можуть притворюватися користувачем, отримувати доступ до його особистої інформації, виконувати дії в його ім'я,

робити зміни у його профілі, замовляти товари або послуги, а також виконувати інші дії, які можуть завдати шкоди як користувачеві, так і веб-сайту.

Для захисту від перехоплення сесій рекомендується використовувати безпечні методи аутентифікації та авторизації, використовувати шифрування для передачі сесійних токенів, використовувати безпечне підключення (наприклад, HTTPS), використовувати механізми захисту від MITM-атак та постійно оновлювати програмне забезпечення веб-сайту, щоб усунути вразливості, які можуть бути використані злоумисниками для перехоплення сесій.

1.3 Інструменти аналізу захищеності веб-сайтів

Інструменти аналізу захищеності веб-сайтів допомагають ідентифікувати потенційні вразливості та ризики, пов'язані з безпекою. Вони можуть бути класифіковані на автоматизовані сканери вразливостей та ручні методи, такі як кодова рецензія та пенетраційне тестування. Деякі з найпоширеніших інструментів аналізу захищеності веб-сайтів включають:

OWASP ZAP (Open Web Application Security Project Zed Attack Proxy) - це безкоштовний і відкритий інструмент для сканування вразливостей веб-додатків [4, 1]. Він розроблений спеціально для дослідження та тестування безпеки веб-додатків з точки зору різних атак, зокрема XSS (Cross-Site Scripting), SQL Injection, CSRF (Cross-Site Request Forgery) та багатьох інших.

Основні функції і можливості OWASP ZAP включають:

- активне сканування вразливостей: ZAP дозволяє автоматично сканувати веб-додатки з метою виявлення різних вразливостей. Він проводить аналіз вхідних та вихідних параметрів, перевіряє потенційні проблеми безпеки та надає звіт про виявлені уразливості;

- проксі-сервер безпеки: ZAP дозволяє перехоплювати та змінювати HTTP-запити та відповіді, що проходять через нього. Це дає можливість аналізувати та модифікувати дані, що передаються між клієнтом і сервером, для виявлення потенційних проблем безпеки;

- Сканування вразливостей API: ZAP підтримує сканування вразливостей API, включаючи REST та SOAP. Він дозволяє ідентифікувати проблеми безпеки, пов'язані з некоректними налаштуваннями, слабкими аутентифікаційними методами, недостатньою авторизацією та іншими.

- Автоматизоване тестування: ZAP може бути використаний для автоматизованого тестування безпеки веб-додатків. Він надає можливість створювати скрипти тестування, які виконуються автоматично, допомагаючи виявити потенційні проблеми безпеки шляхом автоматичного сканування та перевірки.

ZAP генерує детальні звіти про виявлені вразливості та рекомендації щодо виправлення проблем безпеки. Звіти можуть бути збережені у різних форматах, що спрощує подальший аналіз результатів та спільну роботу з командою розробників.

OWASP ZAP є популярним інструментом серед професіоналів з безпеки веб-додатків та дослідників вразливостей. Він надає потужні можливості для виявлення та виправлення проблем безпеки, допомагаючи забезпечити захист веб-додатків від потенційних атак.

Burp Suite - це набір інструментів для тестування безпеки веб-додатків [11, 3]. Він розроблений компанією PortSwigger і широко використовується спеціалістами з безпеки для виявлення та виправлення вразливостей веб-додатків. Burp Suite має різні компоненти, кожен з яких виконує конкретну функцію в процесі тестування безпеки.

Основні компоненти Burp Suite включають:

- Proxy-сервер: Це основний компонент Burp Suite, який дозволяє перехоплювати та змінювати HTTP-запити та відповіді, що проходять через

нього. Проксі-сервер дозволяє аналізувати та модифікувати дані перед їх відправкою на сервер або перед їх отриманням клієнтом. Це дозволяє досліджувати вразливості, пов'язані зі зміною даних, додаванням шкідливого коду та іншими атаками.

- Сканер вразливостей: Burp Suite має вбудований сканер вразливостей, який автоматично сканує веб-додатки на предмет різних вразливостей. Він виявляє такі типи вразливостей, як XSS (Cross-Site Scripting), SQL Injection, RCE (Remote Code Execution) та інші. Сканер вразливостей Burp Suite допомагає виявити потенційні проблеми безпеки та генерує звіт з виявленими вразливостями [12, 5].

- Repeater: Цей інструмент дозволяє повторювати HTTP-запити з мінімальними змінами. Він корисний для перевірки вразливостей, які потребують багаторазового виконання однакових запитів зі зміненими параметрами або заголовками.

- Sequencer: Sequencer використовується для аналізу стійкості токенів аутентифікації, що використовуються в веб-додатках. Він допомагає визначити, наскільки прогнозовані токени та генерувати статистику для оцінки їх стійкості.

- Decoder: Цей інструмент дозволяє декодувати та закодовувати різні типи даних, такі як URL-кодування, Base64, шестнадцяткове представлення та інші. Він допомагає аналізувати та модифікувати дані, що передаються через HTTP – запити та відповіді.

- Intruder: Цей інструмент дозволяє автоматизувати атаки на веб-додатки шляхом зміни параметрів та значень в запитах. Він корисний для перебору, вибіркового змінювання параметрів та перевірки наявності вразливостей.

Burp Suite є потужним інструментом, який забезпечує розширені можливості для аналізу та виявлення вразливостей веб-додатків. Він широко використовується професіоналами з безпеки для проведення тестування безпеки та покращення захисту веб-додатків.

Nmap (Network Mapper) - це відкрите програмне забезпечення для сканування мереж і виявлення активних хостів, портів, служб та інших характеристик мережевих систем. Він є одним з найпотужніших і популярних інструментів для сканування мереж і використовується як професіоналами з безпеки, так і адміністраторами систем для дослідження мереж та виявлення вразливостей.

Основні функції та можливості Nmap:

- сканування портів: Nmap дозволяє сканувати мережу для виявлення відкритих портів на хостах. Він може використовувати різні методи, такі як TCP, UDP, SYN, ACK, FIN сканування, для визначення стану портів (відкриті, закриті або фільтровані) [5, 10]. Це дозволяє аналізувати мережу та визначати, які служби або сервіси доступні на конкретних хостах;

- виявлення активних хостів: Nmap може виявляти активні хости в мережі шляхом відправки ICMP пакетів або TCP/UDP запитів. Він дозволяє виявити, які хости доступні в мережі та збудувати карту мережі;

- отримання інформації про систему: Nmap може отримувати детальну інформацію про операційну систему, версії сервісів та характеристики мережевих систем, які він виявляє. Це допомагає аналізувати цільові системи та виявляти можливі вразливості;

- сканування вразливостей: Nmap може виконувати сканування вразливостей, використовуючи вбудовані скрипти (Nmap Scripting Engine). Це дозволяє автоматично перевіряти системи на наявність відомих вразливостей та проводити оцінку їх стану безпеки;

- сканування великих мереж: Nmap може працювати з великими мережами, включаючи декілька тисяч хостів. Він може бути налаштований для ефективного сканування великих мереж, використовуючи різні методи і оптимізації.

Nmap є потужним інструментом для аналізу мережі та виявлення вразливостей. Він забезпечує широкі можливості сканування, аналізу та оцінки

стану безпеки мережевих систем. Професіонали з безпеки та адміністратори систем використовують Nmap для забезпечення безпеки мереж та виявлення потенційних загроз.

Nikto - це веб-сканер вразливостей, призначений для виявлення потенційних проблем безпеки на веб-серверах і додатках. Це популярний інструмент, який використовується професіоналами з безпеки для аудиту безпеки веб-додатків і пошуку вразливостей [3].

Основні функції та можливості Nikto:

- сканування веб-серверів: Nikto виконує активне сканування веб-серверів з метою виявлення вразливостей і проблем безпеки. Він перевіряє різні аспекти сервера, включаючи налаштування, наявність старих версій програмного забезпечення, налаштування безпеки та інші фактори, які можуть створювати ризик;

- виявлення вразливостей: Nikto використовує базу даних з підписами вразливостей і патчів, щоб знайти відомі проблеми безпеки на веб-серверах. Він перевіряє наявність вразливих версій програмного забезпечення, слабких налаштувань, потенційно небезпечних дій та інших проблем, які можуть бути використані зловмисниками;

- пошук скритих файлів і каталогів: Nikto аналізує веб-сервер для виявлення скритих файлів і каталогів, які можуть бути доступні, але не відображаються у звичайному інтерфейсі. Це може допомогти виявити потенційно чутливу інформацію, яка може бути доступна для несанкціонованого доступу;

- перевірка налаштувань безпеки: Nikto перевіряє налаштування безпеки веб-сервера, такі як відкриті порти, наявність необхідних захисних механізмів (наприклад, SSL/TLS), права доступу до файлів і каталогів, контроль доступу і інші аспекти безпеки. Він може виявляти слабкі налаштування, які можуть призвести до компрометації безпеки;

- звіти та результати: Nikto надає можливість створювати звіти про виявлені вразливості та проблеми безпеки. Звіти містять докладну інформацію про кожну виявлену проблему, включаючи опис, рекомендації щодо виправлення і посилання на додаткові ресурси.

Nikto є потужним інструментом для виявлення проблем безпеки на веб-серверах і додатках. Він допомагає професіоналам з безпеки виявляти потенційні вразливості і ризики, що дозволяє приймати відповідні заходи для поліпшення безпеки веб-сайтів та додатків.

1.4 Приклади аналізу освітніх платформ

Освітні платформи є широко поширеними, і багато з них було піддано аналізу захищеності. Ось кілька прикладів досліджень, що стосуються аналізу освітніх платформ:

- "Security Analysis of Learning Management Systems" [6]: У цьому дослідженні було проведено аналіз безпеки систем управління навчанням (Learning Management Systems, LMS). Дослідження охоплювало популярні LMS, такі як Moodle та Blackboard, і виявило різні вразливості, такі як недостатні заходи автентифікації та авторизації.

- "Security Analysis of Online Learning Platforms" [13]: У цьому дослідженні було проведено аналіз безпеки онлайн-платформ для навчання. Дослідження виявило різні вразливості, такі як недостатні заходи захисту від атак XSS та SQL-ін'єкцій.

- "Security Analysis of Massive Open Online Courses" [14]: Це дослідження зосереджувалося на безпеці масових відкритих онлайн-курсів [15]. Аналіз проводився на популярних платформах MOOC, таких як Coursera та edX, і

виявив різні загрози, зокрема проблеми з конфіденційністю даних та атаки на сторонні додатки.

Ці приклади ілюструють, як дослідники вивчають безпеку освітніх платформ і виявляють потенційні вразливості та загрози.

2 ОГЛЯД ДОСТУПНИХ ІНСТРУМЕНТІВ

2.1 Аналіз захищеності освітніх сайтів

Аналіз захищеності освітніх сайтів є критично важливим процесом для забезпечення безпеки та захисту веб-ресурсів освітніх установ. Освітні сайти містять значну кількість конфіденційної інформації, такої як особисті дані студентів, академічні записи, фінансова інформація, результати тестів тощо. Втрати або незаконне використання цих даних можуть мати серйозні наслідки, включаючи порушення конфіденційності, шахрайство, ідентифікаційну крадіжку та негативний вплив на репутацію освітнього закладу.

Заходи аналізу захищеності освітніх сайтів включають:

- виявлення вразливостей. Аналіз захищеності дозволяє ідентифікувати потенційні вразливості веб-додатків і інфраструктури освітнього сайту. Це можуть бути вразливості в програмному забезпеченні, недостатні контролі доступу, незахищені мережеві налаштування тощо. Виявлення цих вразливостей дозволяє вжити заходів для їх ліквідації та запобігання можливим атакам;

- захист конфіденційності. Аналіз захищеності допомагає виявляти потенційні ризики, які можуть призвести до витоку конфіденційної інформації. Це дозволяє приймати відповідні заходи безпеки, щоб захистити особисті дані студентів, викладачів та інших користувачів освітнього сайту;

- забезпечення цілісності даних. Аналіз захищеності допомагає виявляти потенційні загрози цілісності даних. Це означає перевірку та запобігання несанкціонованим змінам, втраті або порушенню цілісності даних, що зберігаються на освітньому сайті. Забезпечення цілісності даних є важливим аспектом для забезпечення довіри до освітнього закладу та його онлайн-ресурсів;

- забезпечення доступності. Аналіз захищеності допомагає ідентифікувати потенційні загрози доступності освітнього сайту. Це можуть

бути атаки DDoS, відмови в обслуговуванні або технічні проблеми, що обмежують доступ користувачів до веб-ресурсу. Забезпечення доступності освітнього сайту важливо для забезпечення безперебійного доступу до навчальних матеріалів та ресурсів для студентів, викладачів та інших зацікавлених сторін;

- збереження репутації. Аналіз захищеності допомагає уникнути інцидентів безпеки, які можуть негативно вплинути на репутацію освітнього закладу. Захищений освітній сайт забезпечує довіру користувачів та сприяє позитивному враженню професійності та дбайливості освітнього закладу щодо захисту конфіденційності та безпеки користувачів.

Усі ці аспекти аналізу захищеності сприяють створенню безпечного та надійного середовища для освітнього закладу, його співробітників та користувачів.

2.2 Аналіз інструментів, методів та технік дослідження захищеності сайтів

Аналіз захищеності веб-сайтів можна проводити за допомогою різних інструментів, методів та технік. Ось огляд деяких доступних інструментів:

Автоматизовані сканери вразливостей

Автоматизовані сканери вразливостей є корисними інструментами для виявлення потенційних проблем безпеки на веб-сайтах. Вони допомагають автоматизувати процес сканування та аналізу веб-додатків з метою виявлення вразливостей. Ось деякі ключові аспекти автоматизованих сканерів вразливостей:

- сканування вразливостей. Сканери вразливостей проводять систематичний перехід по веб-сайту та аналізують його структуру, компоненти та параметри, шукаючи можливі вразливості. Вони виконують автоматичні

запити, надсилають спеціально сформовані дані та аналізують повернені результати, щоб виявити потенційні проблеми;

- виявлення різних типів вразливостей. Сканери вразливостей можуть виявляти широкий спектр вразливостей, включаючи XSS (Cross-Site Scripting), SQL-ін'єкцію, вразливості аутентифікації та авторизації, недостатню обробку введення користувача, вразливості в конфігурації сервера та багато інших. Вони базуються на знаннях про типові вразливості та їхніх сигнатурах;

- репортинг результатів. Після завершення сканування сканери вразливостей генерують звіт, в якому перелічені виявлені проблеми безпеки. Цей звіт може включати опис вразливостей, їх критичність, рекомендації щодо виправлення та іншу корисну інформацію. Це допомагає розробникам та адміністраторам веб-сайту виправити виявлені проблеми та покращити загальну безпеку;

- інтеграція з іншими інструментами. Деякі сканери вразливостей можуть інтегруватись з іншими інструментами, такими як системи управління безпекою, інструменти для тестування продуктивності, системи контролю версій тощо. Це дозволяє спрощувати процес виявлення та виправлення вразливостей, а також координувати безпекові заходи;

- персоналізація та налаштування. Багато сканерів вразливостей надають можливість налаштувати параметри сканування відповідно до потреб користувача. Це дозволяє керувати обсягом сканування, глибиною аналізу та іншими факторами, що впливають на точність та продуктивність сканера.

- Приклади популярних автоматизованих сканерів вразливостей включають: OWASP ZAP, Burp Suite, Acunetix, Nessus, OpenVAS та Nikto. Характеристики цих інструментів було розглянуто вище (п. 1.3).

Ручні методи аналізу захищеності сайтів використовують експертний підхід, де людина вручну перевіряє веб-додаток на наявність вразливостей та потенційних проблем безпеки. Ось кілька ключових аспектів ручного аналізу захищеності:

- пентестинг (тестування на проникнення). Пентестинг - це процес активного тестування безпеки, під час якого спеціаліст забезпечує симуляцію атаки на веб-додаток, шукаючи вразливості та проблеми безпеки. Це може включати введення спеціально сформованих даних, спроби використання недостатньо захищених функцій, аналіз поведінки системи та інші активності з метою виявлення проблем;

- аналіз вихідного коду. Ручний аналіз вихідного коду дозволяє експертам виявити потенційні вразливості та слабкі місця безпеки в програмному коді веб-додатку. Це може включати пошук небезпечних функцій, некоректної обробки введення, можливих проблем з контролем доступу та інші потенційні проблеми, які можуть бути виявлені шляхом аналізу коду;

- аналіз конфігурації сервера та мережі. Ручний аналіз конфігурації сервера та мережі допомагає ідентифікувати можливі слабкі місця в налаштуваннях, які можуть призвести до вразливостей або недостатнього рівня захисту. Це включає перевірку правильності налаштування сервера, контроль доступу, застосування безпечних протоколів, управління сертифікатами SSL/TLS та інші аспекти інфраструктури;

- аналіз введення користувача. Ручний аналіз введення користувача полягає в тестуванні веб-додатка на вразливість до некоректного або шкідливого введення. Це включає спроби вводу небезпечних символів, SQL – ін'єкції, XSS – атаки, введення команд та інші техніки, спрямовані на злам системи через введення користувача;

- соціальний інжиніринг. Ручний аналіз захищеності може включати оцінку рівня вразливості системи до соціального інжинірингу. Це включає оцінку заходів безпеки, пов'язаних зі способами аутентифікації, керування доступом та навіть навичками користувачів уникати фішингових атак.

Ці методи ручного аналізу вимагають великого досвіду та експертності, оскільки вони потребують розуміння різних типів вразливостей, вміння аналізувати код та конфігурацію, а також вміння здійснювати комплексну оцінку

системи з точки зору безпеки. Вони часто використовуються спільно з автоматизованими інструментами для забезпечення більш широкого покриття тестування та виявлення вразливостей.

Моніторинг безпеки.

Моніторинг безпеки є важливою складовою процесу забезпечення захищеності веб-додатків і сайтів. Це систематичний процес спостереження, збору та аналізу інформації про події, активності та стан безпеки системи з метою виявлення потенційних загроз та вразливостей. Ось деякі ключові аспекти моніторингу безпеки:

- спостереження за журналами подій (Event Logging). Моніторинг безпеки включає запис та аналіз журналів подій, що відбуваються на веб-сайті. Це може включати відстеження спроб аутентифікації, недійсних запитів, помилок виконання, змін конфігурації та інші активності, що можуть свідчити про атаку або вразливість;

- виявлення вторгнень (Intrusion Detection). Моніторинг безпеки дозволяє виявляти незвичайну або підозрілу активність, що може бути пов'язана зі зламом або вторгненням в систему. Це може включати аналіз мережевого трафіку, виявлення аномалій у поведінці користувачів, перевірку цілісності файлів та інші методи виявлення загроз;

- моніторинг вразливостей (Vulnerability Monitoring). Постійний моніторинг вразливостей веб-додатків дозволяє виявляти нові вразливості, які можуть вплинути на безпеку системи. Це може включати сканування портів, тестування на проникнення, перевірку наявності оновлень та патчів, а також використання баз даних відомих вразливостей;

- системи виявлення і запобігання вторгнень (Intrusion Detection and Prevention Systems, IDS/IPS). Ці системи використовуються для автоматичного виявлення та блокування потенційно шкідливої активності на веб-сайті. Вони можуть аналізувати мережевий трафік, контролювати запити, виявляти атаки та надавати захист від них;

- спостереження за виробленням політик безпеки (Policy Enforcement Monitoring). Моніторинг безпеки також включає контроль дотримання політик безпеки в організації. Це може включати перевірку наявності інформаційних заходів безпеки, обмежень доступу, налаштування шифрування та інші аспекти, що відповідають встановленим стандартам безпеки.

Ці інструменти та методи моніторингу безпеки допомагають організаціям підтримувати захищеність своїх веб-додатків та сайтів, виявляти та реагувати на потенційні загрози та вразливості, а також забезпечувати відповідність стандартам безпеки та політикам організації.

Аналіз логів є важливою складовою процесу моніторингу та забезпечення безпеки веб-сайтів і додатків. Логи представляють собою записи подій, які відбуваються в системі, і містять важливу інформацію про дії користувачів, взаємодію з базою даних, стан системи, помилки та інші події.

Основна мета аналізу логів полягає в виявленні аномалій, зловживань та вразливостей, а також в забезпеченні відповідності до політик безпеки. Деякі важливі аспекти аналізу логів включають:

- виявлення атак та зловживань. Аналіз логів дозволяє виявляти незвичайні активності, що можуть свідчити про спроби несанкціонованого доступу, атаки з використанням вразливостей, спроби витоку даних тощо. Наприклад, можна виявити спроби SQL-ін'єкції, перехоплення сесій, незвичайні запити або дії, які не відповідають звичайному користуванню додатком;

- виявлення помилок та проблем. Аналіз логів допомагає виявляти помилки та проблеми в роботі системи, такі як невдалий запит до бази даних, виключення, помилки авторизації, недоступність сервісів тощо. Це дозволяє оперативно реагувати на проблеми та вживати заходів для їх виправлення;

- моніторинг відповідності. Аналіз логів дозволяє перевіряти виконання політик безпеки та внутрішніх правил організації. Можна перевіряти, чи дотримуються користувачі правил паролів, чи виконуються обмеження доступу, чи відбувається контроль доступу до конфіденційних даних тощо;

- аналіз трендів та патернів. Аналізуючи логи протягом тривалого періоду, можна виявити тренди, патерни та незвичайні зміни в активності. Наприклад, зростання незвичайних запитів або спроб авторизації може свідчити про спробу злому або спам-атаку;

- дослідження інцидентів. Логи є цінним джерелом інформації при розслідуванні інцидентів безпеки. Аналізуючи логи, можна встановити причину та масштаб інциденту, ідентифікувати зловмисників та прийняти заходи для запобігання подібним інцидентам у майбутньому.

Загалом, аналіз логів дозволяє виявляти потенційні загрози та вразливості, моніторити безпеку системи та приймати відповідні заходи для забезпечення захищеності веб-додатків та сайтів.

2.3 Наведення алгоритму роботи

Загальний алгоритм роботи аналізу захищеності веб-сайту складається з наступних кроків.

Збір інформації є першим кроком у процесі аналізу захищеності веб-сайту. Цей етап включає збір різноманітної інформації про цільовий веб-сайт, його інфраструктуру та потенційні вразливості. Основна мета збору інформації полягає у визначенні атакуваних поверхонь та наступних кроків для аналізу.

Деякі загальні методи збору інформації включають:

- пасивний збір інформації. Цей метод полягає у зборі інформації про веб-сайт без прямого взаємодії з ним. Це може включати перегляд публічно доступних даних, таких як інформація WHOIS про домен, дані DNS, інформація про власника або оператора сайту та інші відкриті джерела інформації;

- активний збір інформації. Цей метод вимагає взаємодії з веб-сайтом або його компонентами для отримання додаткової інформації. Це може включати сканування портів, виявлення активних хостів, визначення сервісів та їх версій,

збір інформації про структуру сайту, перевірку наявності вразливостей та інші активні дії;

- соціальний інжиніринг. Цей метод включає використання соціальних методів для отримання інформації про веб-сайт. Наприклад, це може бути спроба отримати інформацію від співробітників або користувачів сайту шляхом електронних листів, телефонних дзвінків або використання імітації інших осіб;

- аналіз вихідного коду. Вивчення вихідного коду веб-сайту або його компонентів може надати важливу інформацію про можливі вразливості. Це може включати перегляд HTML, CSS, JavaScript або інших мов програмування, що використовуються для розробки сайту.

Аналіз вразливостей – це процес виявлення і оцінки потенційних вразливостей веб-сайту або програмного забезпечення. Цей процес має на меті ідентифікувати слабкі місця в системі, які можуть бути використані зловмисниками для несанкціонованого доступу, пошкодження даних або виконання шкідливих дій.

Аналіз вразливостей зазвичай включає такі кроки:

- сканування. Цей крок включає використання спеціальних інструментів, таких як веб-сканери або сканери портів, для виявлення вразливих місць в системі. Сканування може включати перевірку відкритих портів, виявлення недостатньо захищених служб, перевірку конфігураційних помилок та інші види перевірок;

- аналіз вразливостей. Після виявлення потенційних вразливостей, проводиться детальний аналіз кожної з них. Це може включати перевірку вразливостей в програмному забезпеченні, слабкостей в мережевій інфраструктурі, недоліків в безпеці даних, недостатніх механізмах автентифікації та авторизації та інші види вразливостей;

- класифікація та оцінка. Вразливості класифікуються залежно від їх серйозності та потенційного впливу на систему. Оцінка ризику допомагає

визначити, як швидко та в якій мірі вразливість може бути використана зловмисниками і які заходи необхідно прийняти для усунення проблеми;

- вирішення та рекомендації. Після виявлення вразливостей та їх оцінки, важливо прийняти заходи для усунення проблем. Це може включати встановлення патчів, оновлення програмного забезпечення, зміну конфігурацій або впровадження інших заходів безпеки. Крім того, видаються рекомендації з покращення загальної безпеки системи.

Аналіз вразливостей може проводитися як автоматизовано (за допомогою спеціальних інструментів), так і вручну. Комбінація цих підходів може забезпечити більш повне та точне виявлення вразливостей. Важливо використовувати надійні та актуальні інструменти для аналізу вразливостей, а також мати достатні знання та досвід для ефективного проведення аналізу.

Аналіз конфіденційності, цілісності та доступності є важливими складовими процесу оцінки безпеки і захищеності системи. Кожен з цих аспектів має власне значення і вимагає спеціальних підходів для свого аналізу:

- конфіденційність. Конфіденційність відноситься до захисту інформації від несанкціонованого доступу або розголошення. Під час аналізу конфіденційності перевіряються механізми контролю доступу, шифрування даних, політики паролів, захисту від перехоплення інформації та інші аспекти, що забезпечують конфіденційність даних;

- цілісність. Цілісність стосується збереження цілості даних та захисту від несанкціонованої модифікації. Під час аналізу цілісності перевіряються механізми контролю доступу до даних, виявлення та запобігання вторгнень, контроль версій даних, захист від зміни даних в транзиті і в спокої, а також механізми виявлення несанкціонованої модифікації;

- доступність. Доступність відноситься до забезпечення доступу до системи та її ресурсів для авторизованих користувачів. Аналіз доступності включає перевірку механізмів балансування навантаження, резервного копіювання та відновлення, виявлення та запобігання DDoS-атак, управління

пропускною здатністю, забезпечення високої доступності системи та інші аспекти, що впливають на доступність.

У процесі аналізу цих аспектів використовуються різні методи та інструменти, включаючи перевірку конфігурацій, аудит безпеки, сканування вразливостей, тестування на проникнення, моніторинг системи та інші техніки.

Аналіз логів є важливим етапом в процесі аналізу захищеності системи. Логи є записами подій, які відбуваються в системі, і можуть містити корисну інформацію про події, помилки, активність користувачів та потенційні загрози. Аналіз логів допомагає виявити аномальну активність, вразливості, атаки та виявити ознаки компрометації системи.

Основні кроки аналізу логів включають:

- збір лог-файлів. Цей крок включає збір лог-файлів з різних джерел, таких як сервери, мережеві пристрої, додатки тощо. Лог-файли можуть містити інформацію про події, помилки, автентифікацію, доступ до ресурсів і багато іншого;
- фільтрація та обробка логів. Після збору лог-файлів вони піддаються фільтрації та обробці. Зазвичай відбираються релевантні лог-події, які вказують на потенційні проблеми або загрози. Цей крок допомагає зменшити обсяг і складність аналізуваних даних;
- аналіз подій. Після фільтрації лог-файлів виконується аналіз окремих подій. Це включає виявлення несподіваних або незвичайних активностей, пошук патернів, які можуть вказувати на атаки або вразливості, та ідентифікацію потенційних проблем;
- кореляція подій. У цьому кроці аналізуються зв'язки між різними лог-подіями. Це дозволяє виявити складні шаблони або послідовності подій, які можуть вказувати на специфічні типи атак або компрометацію системи;
- виявлення загроз та реагування. Один з основних цілей аналізу логів - виявлення потенційних загроз та вчасна реакція на них. Якщо в процесі аналізу виявляються підозрілі або вразливі активності, необхідно прийняти відповідні

заходи безпеки, такі як блокування вразливостей, відновлення даних або зміна налаштувань системи.

Аналіз логів допомагає виявляти потенційні загрози та вразливості, а також покращувати загальну безпеку системи шляхом вчасного реагування на події та виконання необхідних заходів безпеки. Він є важливою складовою частиною загального процесу аналізу захищеності та забезпечується за допомогою спеціалізованих інструментів та технологій.

Звіт та рекомендації є важливим етапом аналізу захищеності, оскільки надають детальну інформацію про знайдені проблеми, вразливості та потенційні загрози, а також рекомендації щодо подальших кроків для поліпшення безпеки системи. Основні етапи створення звіту та рекомендацій включають:

- аналіз результатів. Після завершення аналізу захищеності системи, проведення тестування та інших відповідних процедур, необхідно аналізувати отримані результати. Це включає оцінку знайдених проблем, вразливостей та потенційних загроз;

- структурування звіту. Звіт повинен бути структурованим та логічно організованим. Він може включати вступний розділ, огляд методології аналізу захищеності, опис використаних інструментів та технік, результати аналізу, включаючи знайдені проблеми та вразливості, а також рекомендації щодо їх виправлення;

- опис проблем та вразливостей. Звіт має містити детальний опис кожної виявленої проблеми та вразливості. Це включає опис їхнього характеру, потенційних наслідків та інших важливих аспектів. Крім того, можуть бути надані додаткові деталі, такі як відповідні кодові фрагменти, скріншоти або журнали подій;

- рекомендації. Найважливішою частиною звіту є рекомендації щодо виправлення виявлених проблем та вразливостей. Рекомендації повинні бути конкретними, зрозумілими та дієвими. Вони можуть включати в себе

виправлення програмного забезпечення, оновлення системи, зміну налаштувань, надання рекомендацій щодо безпеки адміністраторам та користувачам системи;

- заключення. Звіт має завершуватися заключенням, яке підсумовує проведений аналіз, основні висновки та надає загальну оцінку стану захищеності системи. Також можуть бути надані додаткові рекомендації щодо подальшого забезпечення безпеки.

Важливо враховувати, що звіт та рекомендації повинні бути підготовлені з урахуванням аудиторії, якій вони будуть представлені. Таким чином, зрозуміле і просте викладення інформації та чіткі рекомендації допоможуть забезпечити правильне розуміння та впровадження виправлень безпеки.

3 ТЕСТУВАННЯ ТА АНАЛІЗ ВЕБ-САЙТУ

3.1 Підготовка програмного забезпечення до тестування

Отже, для початку процесу тестування веб-сайтів вам потрібно вибрати операційну систему, яка буде використовуватися для цього завдання. Вибір операційної системи є важливим аспектом, оскільки вона має вплив на доступність і ефективність інструментів тестування.

Одним з найпопулярніших варіантів для тестування веб-сайтів є Kali Linux.

Kali Linux є відомою операційною системою, спеціально розробленою для етичного хакінгу, тестування захищеності та аудиту безпеки. Вона заснована на дистрибутиві Debian Linux і надає доступ до великої кількості інструментів та ресурсів для проведення різноманітних видів аналізу та тестування, включаючи тестування веб-сайтів. Однією з основних переваг використання Kali Linux є:

Переваги використання Kali Linux для тестування веб-сайтів:

- широкий спектр інструментів: Kali Linux поставляється з багатьма популярними інструментами, призначеними для сканування вразливостей, тестування на проникнення, перевірки безпеки мереж та інших завдань, пов'язаних з тестуванням веб-сайтів. Це включає такі інструменти, як Nmap, Burp Suite, OWASP ZAP, Nikto, Metasploit Framework та багато інших. Широкий спектр інструментів дозволяє тестувачам виконувати різноманітні види аналізу та тестування, виявляти та експлуатувати вразливості веб-сайтів;

- гнучкість та настроювання: Kali Linux дозволяє тестувачам налаштовувати середовище згідно зі своїми потребами. Вона надає можливість налаштування різних параметрів, включаючи мережеві налаштування, безпеку, доступ до ресурсів та інше. Це дозволяє проводити тестування веб-сайтів у

різних сценаріях та умовах, що сприяє отриманню більш точних та цілеспрямованих результатів;

- спільнота та підтримка: Kali Linux має активну спільноту користувачів та розробників, що забезпечує постійну підтримку, оновлення та вдосконалення операційної системи та її інструментів. Це означає, що ви можете розраховувати на актуальність інструментів та доступ до нових функцій, а також на наявність широкого спектру документації, форумів та ресурсів, які допоможуть вам розширити свої знання та вирішити будь-які проблеми.

Тестування веб-сайтів з використанням віртуальної машини Kali Linux має кілька переваг:

- Ізольоване середовище: Віртуальна машина Kali Linux надає ізольоване середовище для тестування. Це означає, що ви можете проводити тестування веб-сайтів безпечно і без впливу на вашу основну операційну систему чи інші додатки.

- Інструменти безпеки: Kali Linux містить набір потужних інструментів для тестування безпеки, які вже встановлені та готові до використання. Це дозволяє вам ефективно виконувати сканування вразливостей, аналізувати безпеку веб-сайтів та здійснювати інші види тестування.

- Налагодження та настройка: Віртуальна машина Kali Linux дає вам повний контроль над налаштуваннями та настройками, що дозволяє вам точно настроїти тестове середовище відповідно до вашої потреби. Ви можете змінювати налаштування мережі, доступу до ресурсів та інші параметри для досягнення оптимальних результатів тестування.

- Навчання та практика: Використання віртуальної машини Kali Linux для тестування веб-сайтів також дозволяє вам отримати цінний досвід та практику в області тестування безпеки. Ви можете вивчити різноманітні інструменти, методи та техніки, що допоможуть вам стати більш кваліфікованим тестувальником безпеки.

Враховуючи ці переваги, використання віртуальної машини Kali Linux є рекомендованим підходом для тестування веб-сайтів з точки зору безпеки та ефективності. Вона забезпечує зручне та безпечне

Покрокове встановлення Kali Linux на віртуальну машину [7]:

- завантажуємо віртуальний програмний засіб, наприклад, Oracle VM VirtualBox або VMware Workstation, та встановіть його на вашу основну операційну систему;
- завантажуємо образ Kali Linux з офіційного веб-сайту (https://www.kali.org/downloads/) відповідно до вашої архітектури та версії;
- створюємо нову віртуальну машину в обраному програмному засобі віртуалізації;
- налаштовуємо параметри віртуальної машини, включаючи присвоєння ресурсів (процесор, оперативна пам'ять, дисковий простір) та налаштування мережі;
- додаємо завантажений образ Kali Linux до віртуальної машини;
- запускаємо віртуальну машину та слідуйте інструкціям для встановлення Kali Linux, які з'являться на екрані;
- налаштовуємо мережеві налаштування, користувачів та інші параметри, які ви вважаєте необхідними;
- після завершення встановлення віртуальної машини, перезапустіть її та ви можете почати використовувати Kali Linux для тестування веб-сайтів.

Kali Linux є потужним інструментом, який може бути використаний для тестування веб-сайтів та аналізу їхньої безпеки. Проте, важливо пам'ятати, що використання Kali Linux повинно відбуватися в рамках законів та етичних принципів.

Коли ми встановили Kali Linux як нашу операційну систему для тестування веб-сайтів, першим кроком буде оновлення всіх репозиторіїв. Це важливий крок, оскільки веб-сайти постійно оновлюються і вдосконалюють свої

алгоритми захисту. Щоб мати актуальні інструменти для тестування, ми повинні мати оновлені репозиторії.

Виконується наступна команда для оновлення:

```
sudo apt-get update
```

В Kali linux встановлено багато програм по замовчування, але програма яку ми будемо використовувати не встановлена, а саме OWASP ZAP (Zed Attack Proxy), тому її спочатку треба встановити. Отже, давайте розглянемо процес встановлення OWASP ZAP на Kali Linux.

Встановіть OWASP ZAP, виконавши команду:

```
sudo apt-get install zaproxy
```

Ця команда завантажить та встановить OWASP ZAP разом з усіма залежностями (рис. 3.1).

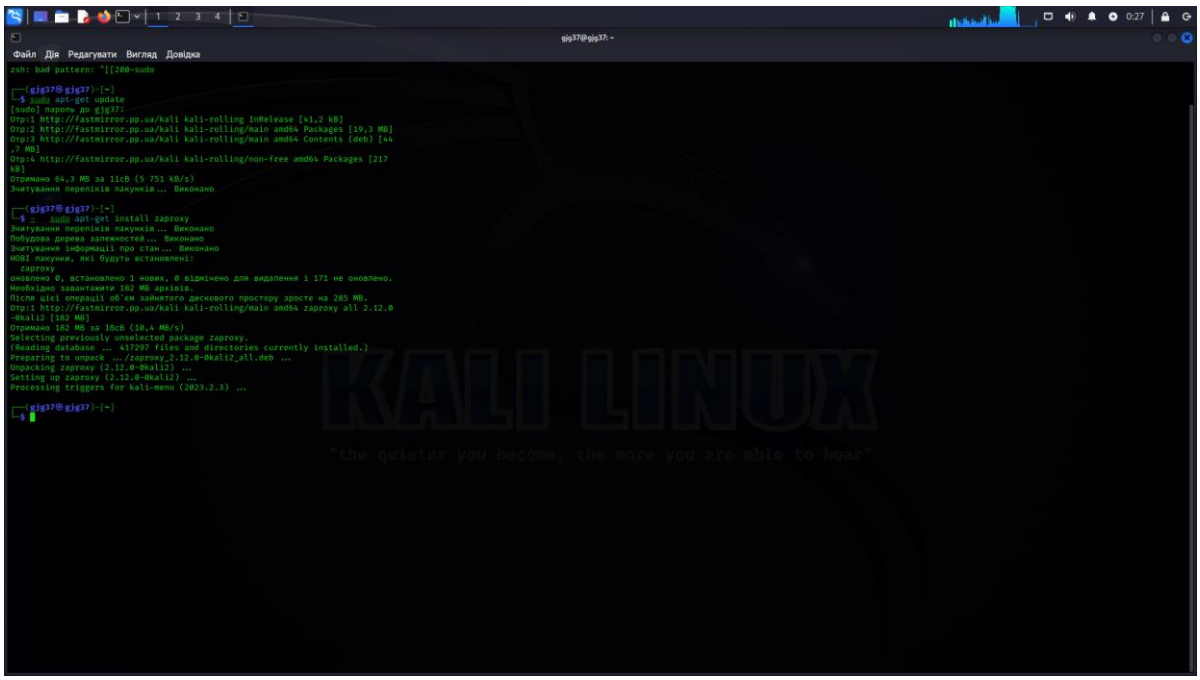

Рисунок 3.1 – Встановлення OWASP ZAP

3.2 Демонстрація знайдених вразливостей

У рамках пошуку вразливостей веб-сайту ТНТУ будуть використаєні наступні інструменти: OWASP ZAP, Nmap та Nikto.

Аналіз вразливостей за допомогою OWASP ZAP.

Для початку роботи сканера вкажемо url сайту. Після початку сканування всі виявлені помилки OWASP ZAP будуть впорядковані за рівнем серйозності вразливостей і знаходитимуться на вкладці "Сповіщення" або "Alerts" (рис. 3.2):

- серйозна вразливість (червоний прапорець)
- вразливість меншої серйозності (оранжевий прапорець)
- менш важливі вразливості (жовтий прапорець)

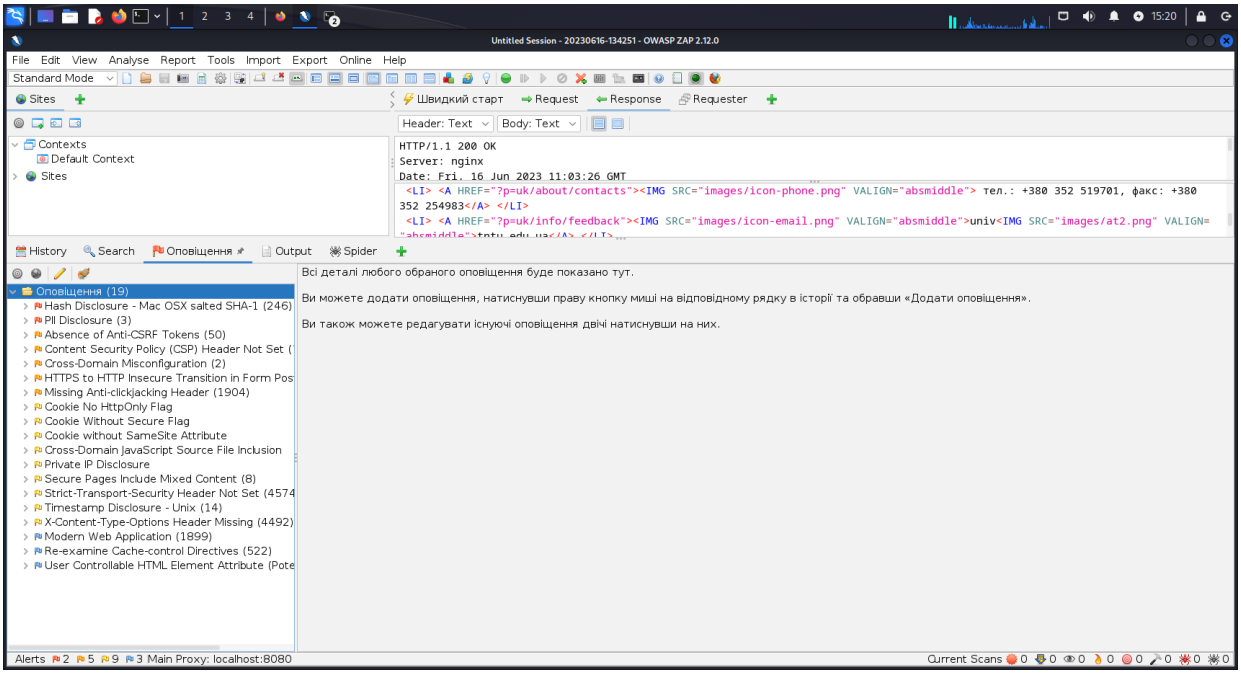

Рисунок 3.2 – Результат роботи OWASP ZAP

Під час сканування було знайдено дві серйозні вразливості та п'ять вразливостей меншої серйозності.Розглянемо кожну з них.

Критичні вразливості.

Hash Disclosure - Mac OSX salted SHA-1 стосується використання солі та алгоритму хешування SHA-1 в операційній системі Mac OSX (рис 3.3).

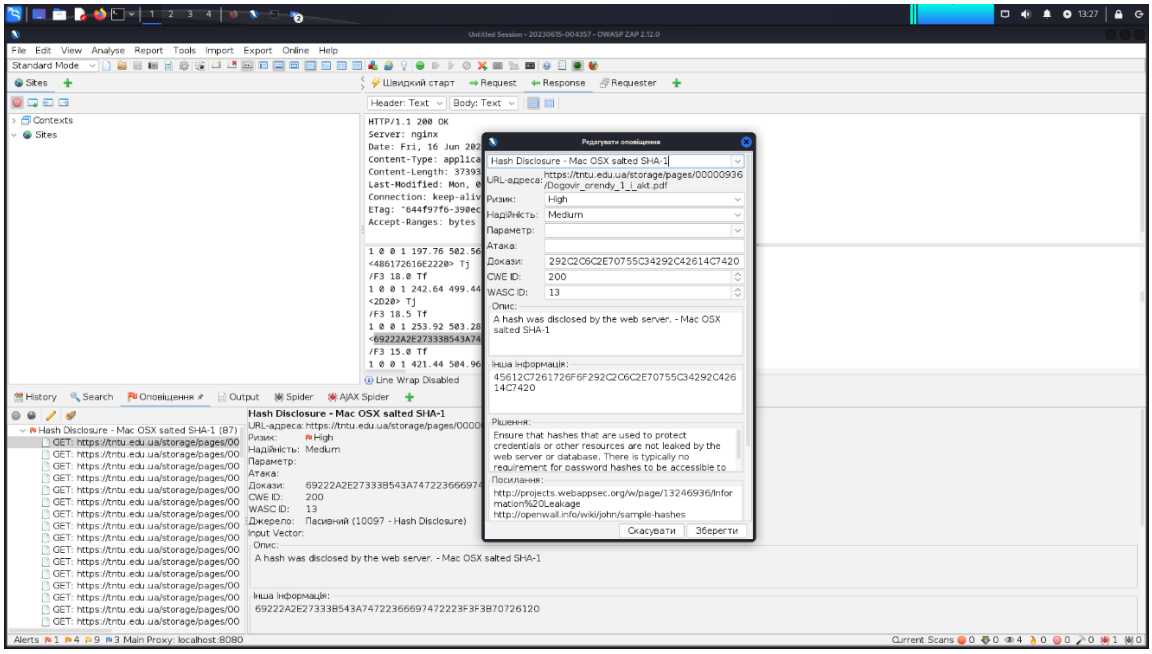

Рисунок 3.3 – Вразливість Hash Disclosure - Mac OSX salted SHA-1

Хеш-функції, такі як SHA-1, використовуються для перетворення вхідних даних в унікальний хеш-значення фіксованої довжини. Хеш-значення, як правило, використовується для зберігання паролів або інших конфіденційних даних в захешованому вигляді.

Проте, якщо використовується слабий алгоритм хешування, такий як SHA-1, і якщо в цьому процесі використовується сіль (salt), вихідні дані можуть бути піддані атакам на розкриття хеш-значень. На практиці це означає, що зловмисники можуть намагатися відновити вихідні дані або паролі, знаючи хеш-значення та сіль.

Сіль (salt) – це додатковий випадковий рядок даних, який додається до вихідних даних перед хешуванням. Солі використовуються для збільшення безпеки хеш-значень, особливо в контексті зберігання паролів або інших конфіденційних даних.

SHA-1 є застарілим алгоритмом хешування, і він вважається вразливим до атак на зіткнення. Це означає, що можливе створення двох різних вхідних повідомлень, які мають одне і те ж хеш-значення. Такі атаки додають ризик до використання SHA-1 для зберігання конфіденційних даних.

PII Disclosure розкриття особисто ідентифікованої інформації (рис 3.4).

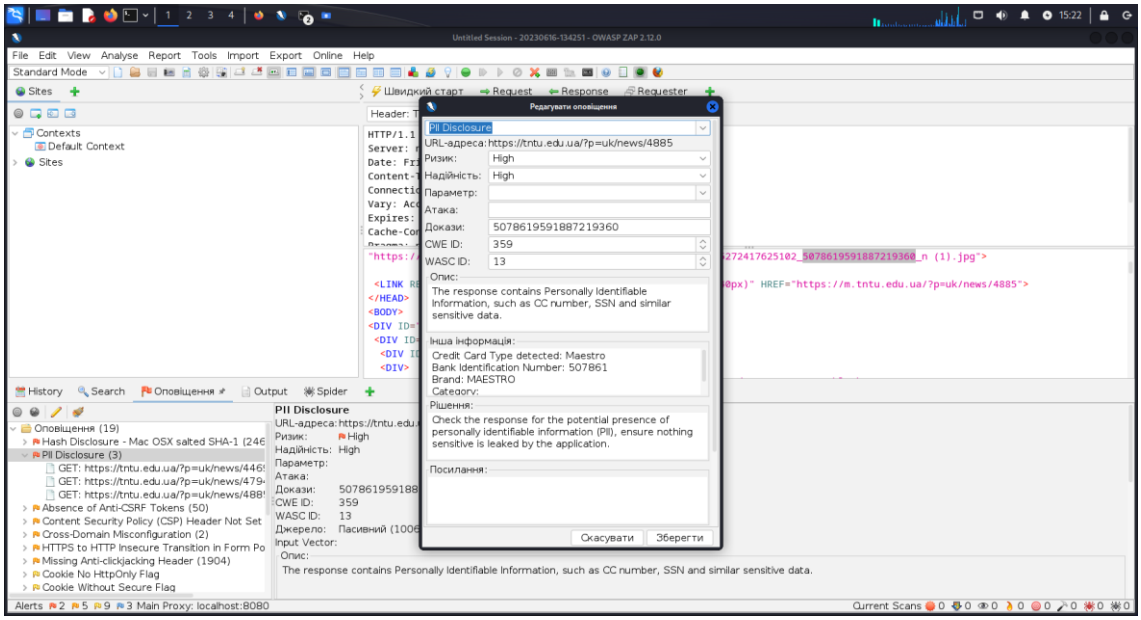

Рисунок 3.4 – Вразливість PII Disclosure

Вразливість PII Disclosure (розкриття особисто ідентифікованої інформації) стосується неконтрольованого або неправильного розкриття конфіденційних особистих даних. PII включає таку інформацію, як імена, адреси, номери соціального страхування, електронні адреси, номери телефонів та інші дані, які можуть ідентифікувати окрему особу.

Ця вразливість може мати серйозні наслідки для приватності та безпеки особи. Якщо PII розкриваються без належного контролю або стають доступними для несанкціонованих осіб, це може призвести до таких проблем:

- Ідентифікаційне шахрайство: Зловмисники можуть використовувати отриману особисту інформацію для здійснення шахрайських дій, таких як

відкриття фальшивих рахунків, використання кредитних карт або отримання фінансової вигоди на ім'я постраждалої особи.

- Порушення приватності: Розкриття особистої інформації може порушити приватність особи, особливо коли ці дані потрапляють в ненадійні руки або використовуються без згоди власника.

- Соціальний інжиніринг: Зловмисники можуть використовувати PII для здійснення соціального інжинірингу, коли вони намагаються отримати доступ до додаткової конфіденційної інформації або використовувати її для маніпуляцій або обману інших осіб.

- Репутаційна шкода: Розкриття PII може призвести до пошкодження репутації особи або організації, особливо якщо інформація стосується недоречних або нелегальних дій.

Вразливості меншої серйозності:

Absence of Anti-CSRF Tokens відсутність анти-CSRF токенів (рис 3.5).

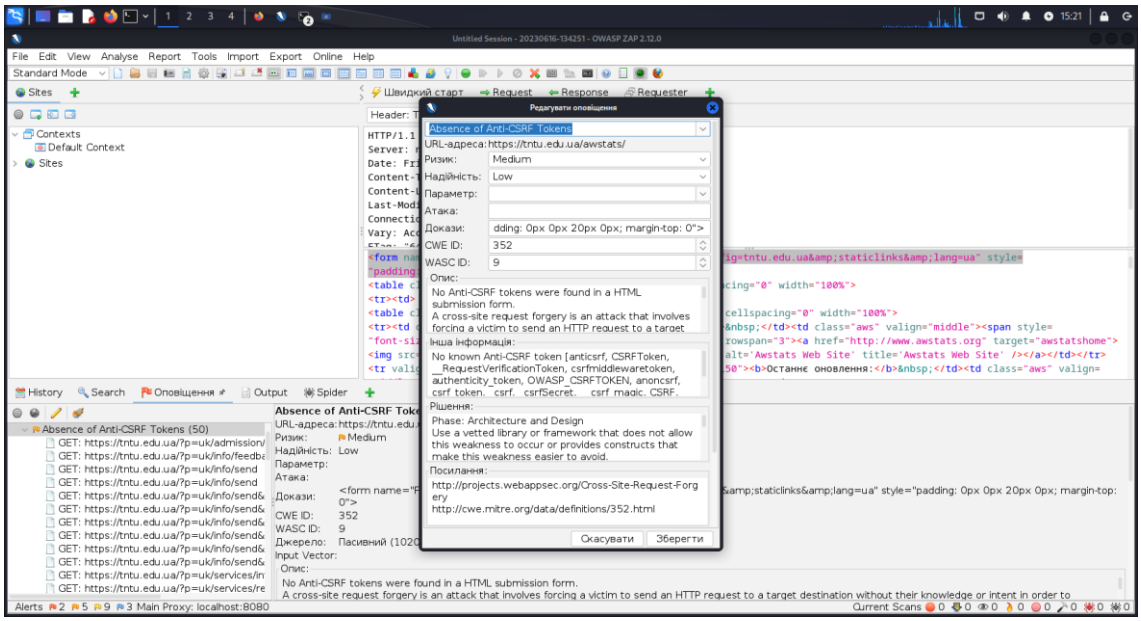

Рисунок 3.5 – Вразливість Absence of Anti-CSRF Tokens

Вразливість "Absence of Anti-CSRF Tokens" (відсутність анти-CSRF токенів) стосується веб-додатків, які не використовують захист від міжсайтової

подібності запитів (CSRF). CSRF - це атака, при якій зловмисник змушує авторизованого користувача виконати небажані дії на вразливому веб-додатку, несвідомо використовуючи його авторизаційні дані.

У випадку відсутності анти-CSRF токенів, додаток не вставляє унікальний токен у форми або запити, що дозволяє зловмиснику сконструювати спеціально сформований запит і виконати дії від імені авторизованого користувача.

Content Security Policy (CSP) Header Not Set відсутність заголовка CSP (рис 3.6).

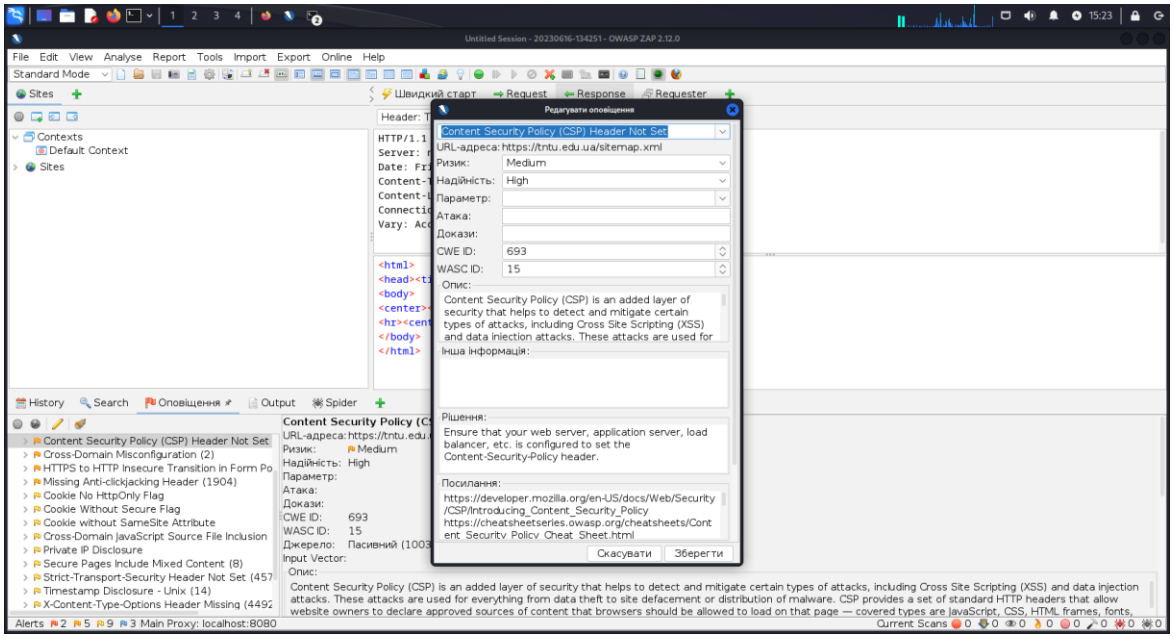

Рисунок 3.6 – Вразливість Content Security Policy (CSP) Header Not Set

Вразливість "Content Security Policy (CSP) Header Not Set" (відсутність заголовка CSP) стосується веб-додатків, які не встановлюють заголовок CSP у своїх відповідях HTTP. CSP є механізмом безпеки, який дозволяє встановити політику контенту для веб-сторінки і забороняє виконання небезпечних скриптів, стилів або інших ресурсів з ненадійних джерел.

Якщо веб-додаток не встановлює заголовок CSP, це означає, що браузери не мають вказівок щодо того, як обробляти зовнішні ресурси. Це може

створювати ризик вразливості, такі як виконання вредоносних скриптів, XSS (міжсайтовий скриптінг) або внедрення інших небажаних ресурсів.

Cross-Domain Misconfiguration неправильна конфігурація кросс-доменних запитів (рис 3.7).

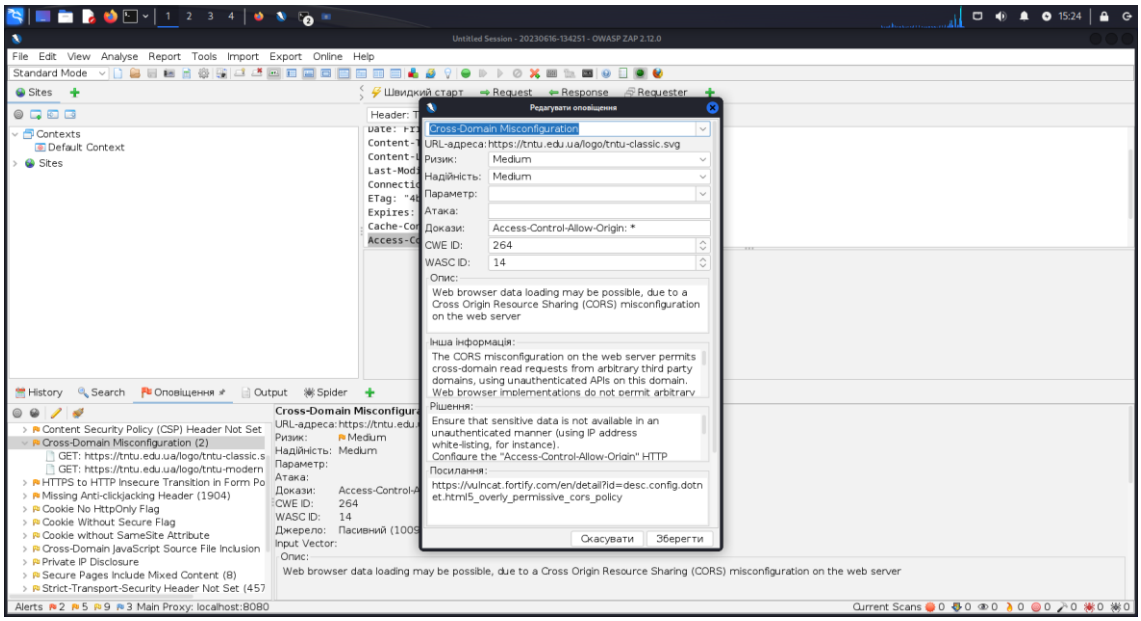

Рисунок 3.7 – Вразливість Cross-Domain Misconfiguration

Вразливість "Cross-Domain Misconfiguration" (неправильна конфігурація кросс-доменних запитів) стосується ситуації, коли веб-додаток неправильно налаштований і допускає кросс-доменні запити з ненадійних джерел. Кросс-доменні запити відбуваються, коли браузер виконує запит до ресурсу на іншому домені, порту або протоколі, ніж початковий домен сторінки.

Якщо кросс-доменні запити не належним чином обмежені, це може викликати ризик вразливості, такі як XSS (міжсайтовий скриптінг), зловживання авторизацією або отримання конфіденційних даних з інших джерел.

HTTPS to HTTP Insecure Transition in Form Post" (перехід з HTTPS на незахищене з'єднання HTTP під час відправки форми (рис 3.8).

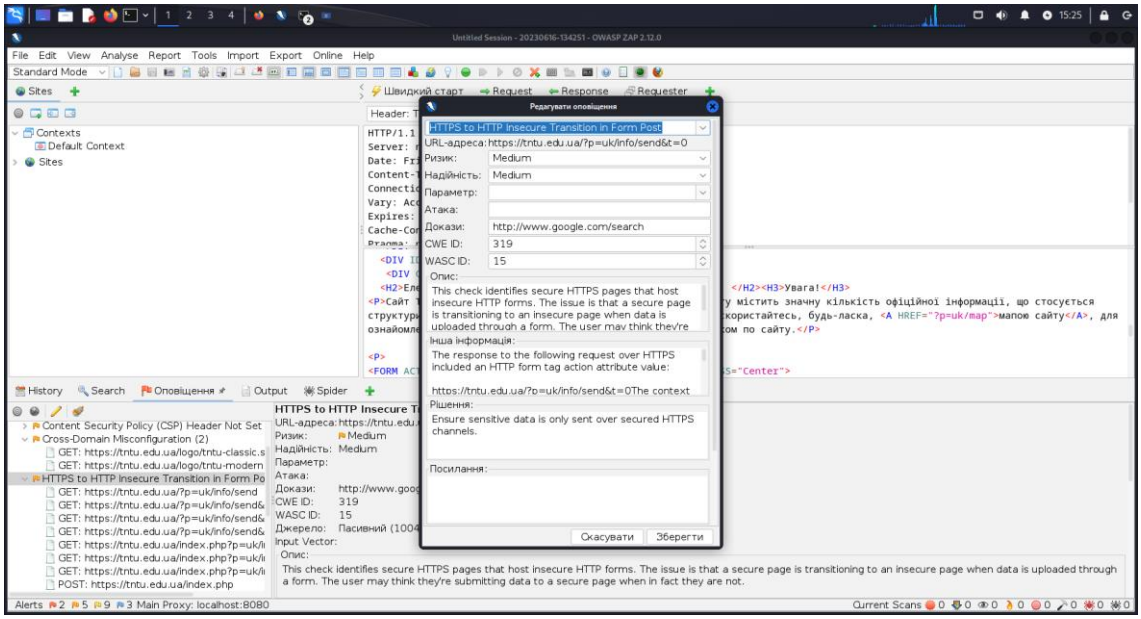

Рисунок 3.8 – Вразливість HTTPS to HTTP Insecure Transition in Form Post

Вразливість "HTTPS to HTTP Insecure Transition in Form Post" (перехід з HTTPS на незахищене з'єднання HTTP під час відправки форми) стосується ситуації, коли веб-додаток дозволяє перехід з безпечного HTTPS з'єднання на незахищене HTTP з'єднання під час надсилання форми.

Це може стати проблемою безпеки, оскільки дані, введені користувачами у форми, можуть бути викрадені або змінені зловмисниками на незахищеному з'єднанні HTTP. Крім того, така ситуація порушує принцип конфіденційності і цілісності даних.

Missing Anti-clickjacking Header" (відсутність заголовка проти clickjacking (рис 3.9).

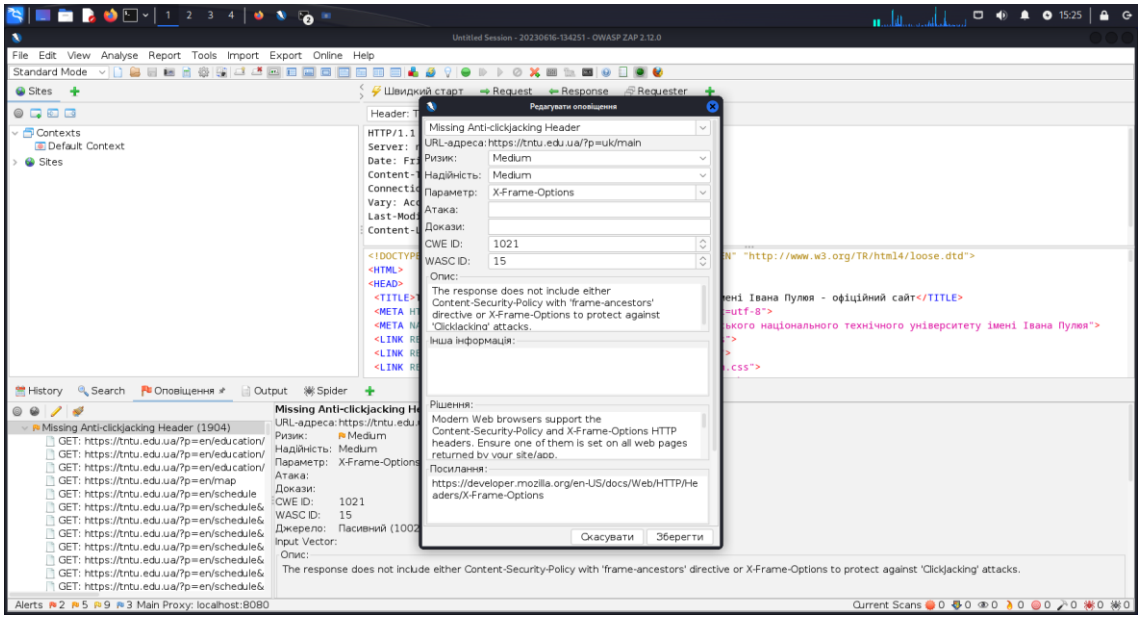

Рисунок 3.9 – Вразливість Missing Anti-clickjacking Header

Вразливість "Missing Anti-clickjacking Header" (відсутність заголовка проти clickjacking) стосується ситуації, коли веб-додаток не встановлює або неправильно конфігурує заголовок X-Frame-Options для захисту від clickjacking атак.

Clickjacking – це атака, при якій зловмисник намагається обманути користувача, відображаючи вміст зловмисної сторінки на видимій веб-сторінці, що призводить до небажаних дій або розкриття конфіденційної інформації.

Також було знайдено ряд незначних вразливостей ось короткі узагальнені пояснення про кожну з вразливостей:

- Cookie No HttpOnly Flag: Відсутність прапорця HttpOnly у cookie дозволяє JavaScript-коду отримувати доступ до цих cookie, що може призвести до крадіжки сесійних ідентифікаторів або інших конфіденційних даних.

- Cookie Without Secure Flag: Відсутність прапорця Secure у cookie дозволяє їм передаватись по незахищеному (незашифрованому) з'єднанню HTTP, що створює ризик витоку конфіденційних даних.

- Cookie without SameSite Attribute: Відсутність атрибуту SameSite у cookie може дозволити зловмиснику використовувати ці cookie для атаки, такої як CSRF (Cross-Site Request Forgery) або зловживання сесіями.

- Cross-Domain JavaScript Source File Inclusion: Ця вразливість виникає, коли веб-додаток дозволяє включати JavaScript-файли з зовнішніх джерел, що може призвести до виконання зловмисного коду або злочинних дій на сторінці.

- Private IP Disclosure: Це витік конфіденційних даних, коли веб-додаток відображає приватні IP-адреси (наприклад, IP-адреси внутрішньої мережі) на сторінці, що може надати зловмисникам цінну інформацію для атаки.

- Secure Pages Include Mixed Content: Ця вразливість виникає, коли захищена сторінка (HTTPS) включає зміст (наприклад, зображення або скрипти) через незахищене з'єднання (HTTP), що піддає ризику безпеку з'єднання та може створити ризик атаки.

- Strict-Transport-Security Header Not Set: Відсутність заголовка Strict-Transport-Security (HSTS) дозволяє зловмисникам використовувати незахищене з'єднання (HTTP) замість безпечного (HTTPS), що може створити ризик атаки або перехоплення даних.

- Timestamp Disclosure – Unix: Ця вразливість виникає, коли веб-додаток відображає інформацію про час сервера або інші системні деталі, що можуть допомогти зловмисникам при плануванні атак або використанні вразливостей.

- X-Content-Type-Options Header Missing: Відсутність заголовка X-Content-Type-Options дозволяє зловмисникам змінювати тип контенту, що може призвести до виконання зловмисного коду або злочинних дій.

Ці вразливості можуть бути вирішені шляхом правильної конфігурації сервера та веб-додатку, встановлення відповідних заголовків і атрибутів cookie, використання безпечних протоколів (наприклад, HTTPS), обмеження доступу до ресурсів і змісту з інших джерел, та застосування кращих практик забезпечення безпеки.

Аналіз вразливостей за допомогою Nmap.

Тепер проведемо пошук вразливостей за допомогою програми Nmap. Перевіримо які порти відкриті на сайті. Запустимо nmap через root та почнемо сканувати. Будемо проводитит сканування портів, версій сервісів та сканування за допомогою скриптів Nmap Scripting Engine.

Сканування портів:

nmap -p 1-1000 91.192.104.90.

Після сканування бачимо такий результат (рис 3.10).

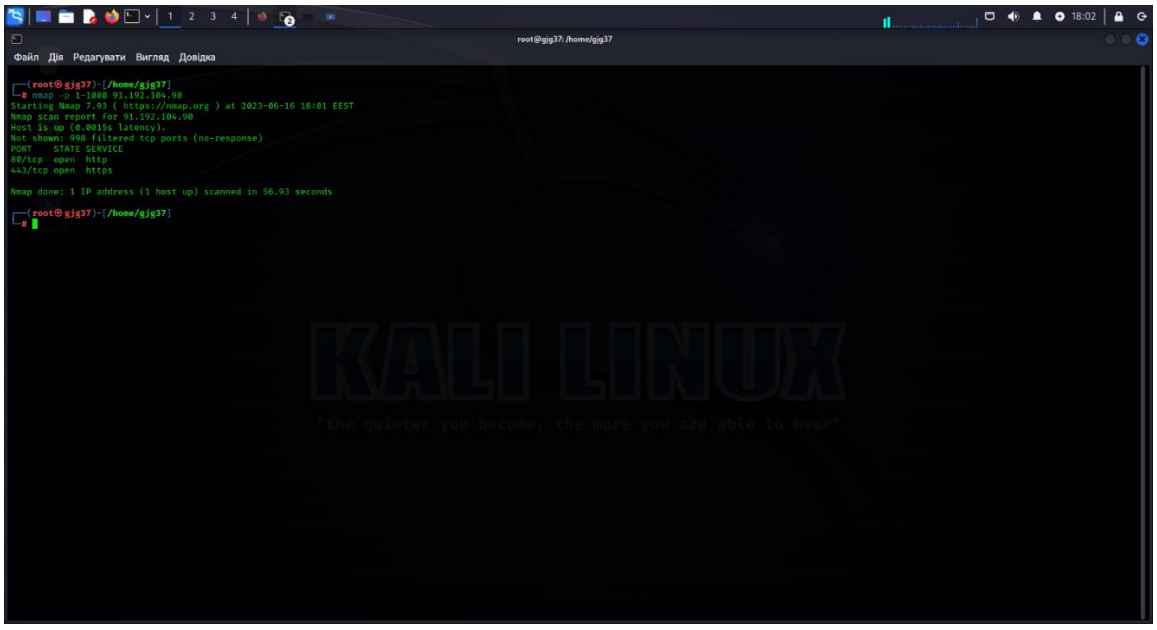

Рисунок 3.10 – Результат сканування портів

Порт "80/tcp open http" і "443/tcp open https" вказують на відкриті порти для протоколів HTTP (порт 80) і HTTPS (порт 443) відповідно.

Відкриті порти HTTP і HTTPS самі по собі не означають конкретну вразливість. Вони вказують лише на те, що веб-сервери, які працюють на цих портах, доступні для з'єднання.

Однак, веб-сервери можуть мати різні вразливості, пов'язані з конфігурацією, програмним забезпеченням або застарілими версіями. Щоб визначити конкретні вразливості на веб-сервері, необхідно провести додатковий аналіз, такий як сканування вразливостей або перевірка безпеки додатків.

Зокрема, в разі HTTP і HTTPS можуть виявлятися такі вразливості, як:

- Уразливості зв'язані зі сторонніми компонентами, які використовуються на веб-сервері (наприклад, уразливості веб-сервера або фреймворка).

- Незахищені налаштування сервера, такі як недостатня фільтрація введених даних, слабкі паролі адміністратора, відсутність захисту від атак типу Cross-Site Scripting (XSS) або SQL Injection.

- Проблеми з сертифікатами SSL/TLS, такі як застарілі або недійсні сертифікати, слабкі алгоритми шифрування або неправильна конфігурація протоколів безпеки.

Сканування версій сервісів:

```
nmap -sV 91.192.104.90
```

Результат сканування версій сервісів (рис 3.11).

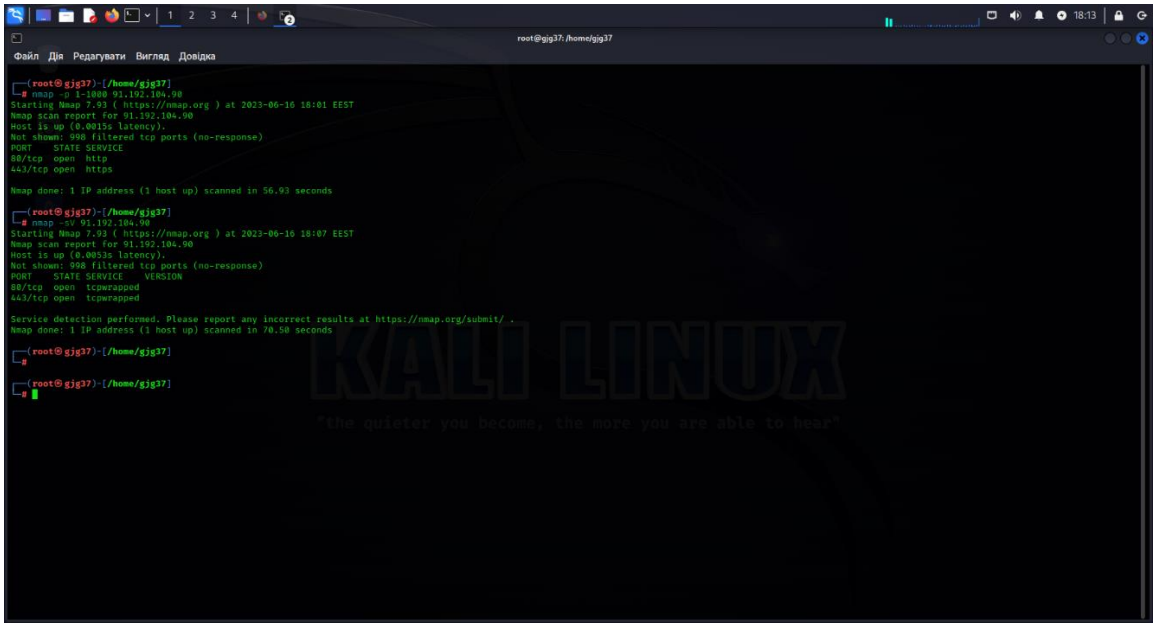

Рисунок 3.11 – Результат сканування версій сервісів

Результати сканування показують, що на відкритих портах 80/tcp і 443/tcp працюють сервіси, які вказані як "tcpwrapped".

Термін "tcpwrapped" означає, що Nmap не зміг точно визначити, який саме сервіс працює на цих портах. На сервері встановлена фільтрація, яка призводить до таких результатів сканування. Фільтрація може бути налаштована для блокування або обмеження доступу до певних портів або сервісів. Це може бути зроблено з метою забезпечення безпеки, обмеження доступу або запобігання несанкціонованому використанню ресурсів сервера.

Сканування за допомогою скриптів:

nmap -sC 91.192.104.90

Результат сканування за допомогою скриптів (рис 3.12).

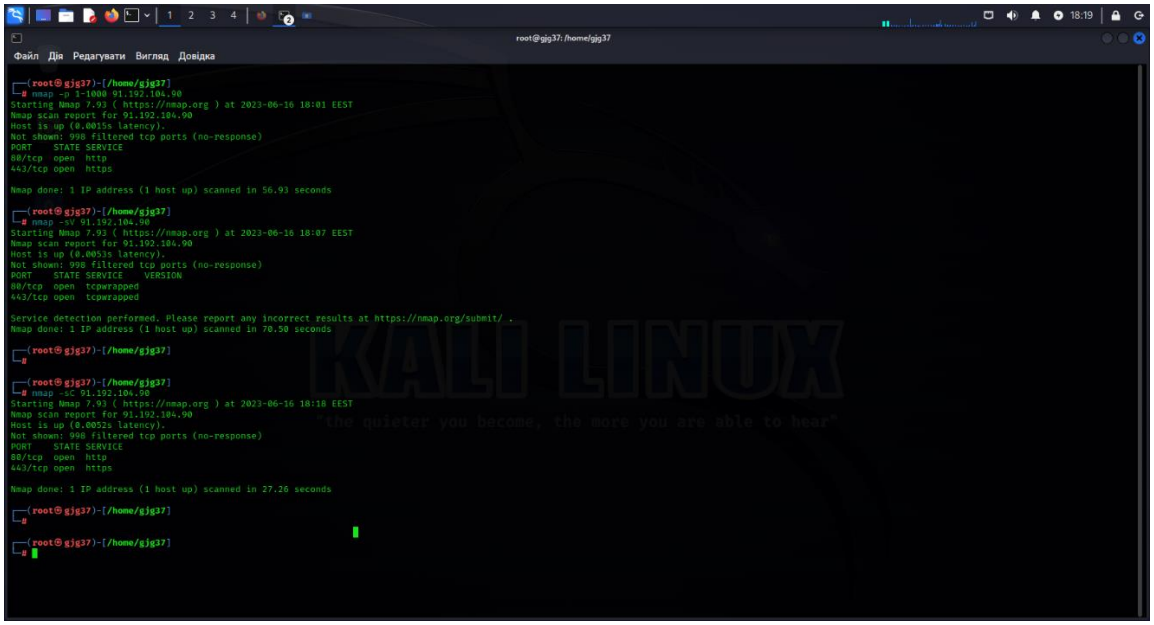

Рисунок 3.12 – Результат сканування за допомогою скриптів

Порти 80 (HTTP) та 443 (HTTPS) на вказаній IP-адресі (91.192.104.90) відкриті і відповідають на запити. Відкритість цих портів свідчить про наявність веб-сервера, який слухає HTTP- та HTTPS-запити на цьому хості.

Сканування відкритих портів за допомогою скриптів:

nmap -sC -p 80,443 91.192.104.90

Результат сканування (рис 3.13).

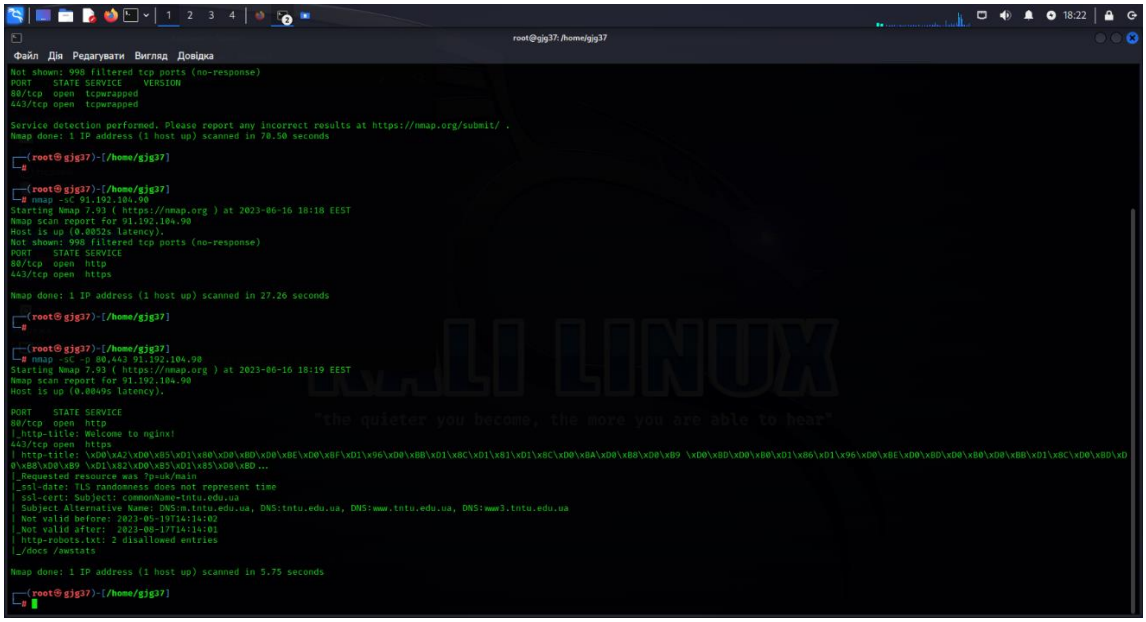

Рисунок 3.13 – Результат сканування відкритих портів за допомогою скриптів

З результатів випливає що виявлено:

- веб-сервер використовує SSL-сертифікат, що вказує на наявність шифрування HTTPS. Виявлення SSL-сертифіката на веб-сервері та наявність шифрування HTTPS означає, що з'єднання між клієнтом (браузером) та сервером захищене шляхом шифрування даних, що передаються між ними. Це забезпечує приватність та безпеку при передачі конфіденційної інформації, такої як паролі, особисті дані та фінансова інформація;

- час дії SSL-сертифіката: дійсний до 2023-08-17;

- веб-сервер має файл robots.txt з двома забороненими записами для шляхів /docs та /awstats. Файл robots.txt є стандартним файлом, який використовується веб-серверами для комунікації з пошуковими системами та іншими роботами, які індексують веб-сайти. У файлі robots.txt можна вказати правила, які обмежують доступ роботів до певних частин веб-сайту.

У цьому випадку, веб-сервер має файл robots.txt з двома забороненими записами, а саме заборона на доступ по шляхах /docs, /awstats. Це означає, що

пошукові системи та роботи не повинні індексувати або отримувати доступ до вмісту, розташованого по вище вказаних шляхах на веб-сайті.

Це стандартний підхід до контролю доступу роботів до певних частин веб-сайту та може використовуватися для захисту конфіденційної інформації або контенту, який не потребує індексації пошуковими системами.

Аналіз вразливостей за допомогою Nikto.

Тепер проведемо сканування за допомогою програми Nikto. Запустимо сканування за допомогою команди:

nikto -h https://tntu.edu.ua/?p=uk/main

Почавши сканування ми зразу можемо звернути увагу що у сайту немає захисту CloudFlare, бо якби він був, то сканер зразу би завершив сканування не видавши жодного результату

За результатами сканування виявлено кілька потенційних проблем безпеки на сайті їх використовуються для розширеного аналізу веб-сайтів і виявлення потенційних вразливостей (рис. 3.14).

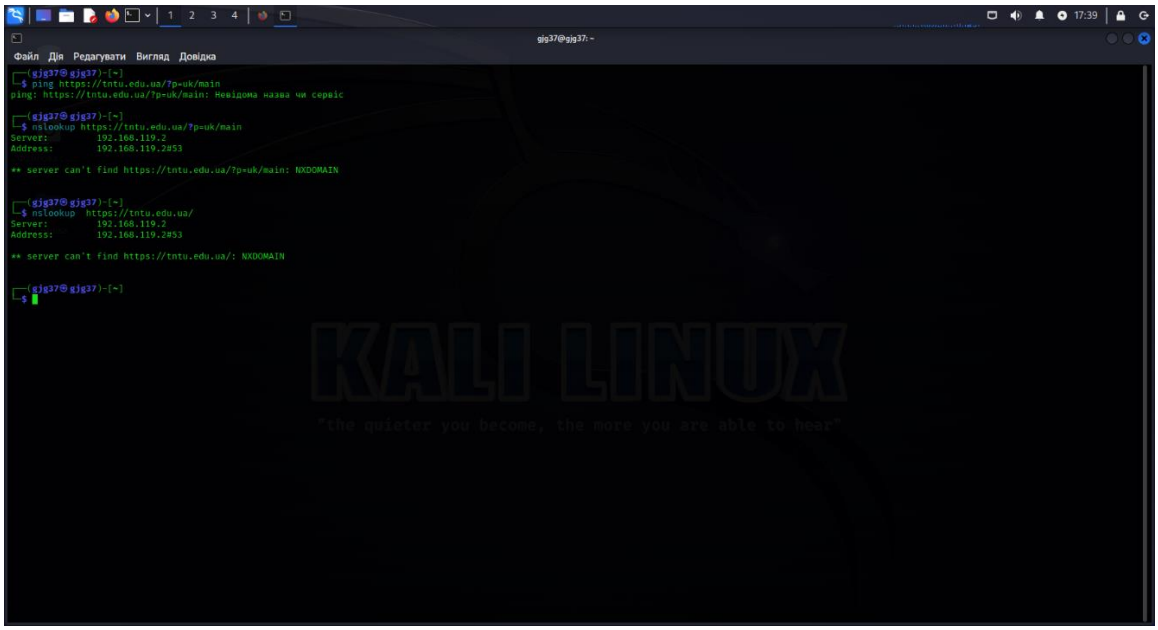

Рисунок 3.14 – Результат сканування за допомогою програми Nikto

З результатів сканування випливає наступне:

- Відсутній заголовок X-Frame-Options: Відсутність цього заголовка може створювати ризик атаки clickjacking, коли зловмисник вміщує веб-сторінку в iframe на іншому сайті. Це може призвести до зловживання і викрадення даних користувачів або виконання шкідливого коду на їхніх пристроях.

- Відсутній заголовок Strict-Transport-Security: Використання TLS для забезпечення безпеки з'єднання є важливим аспектом. Відсутність заголовка Strict-Transport-Security може покласти під загрозу безпеку з'єднання і призвести до атаки MITM (Man-in-the-Middle), коли зловмисник перехоплює комунікацію між сервером і клієнтом.

- Відсутній заголовок X-Content-Type-Options: Цей заголовок використовується для контролю типу вмісту сторінки браузером. Відсутність цього заголовка може призвести до некоректного відображення вмісту сторінки браузером, що може бути використано зловмисником для виконання атак, таких як XSS (Cross-Site Scripting).

- Редірект на головну сторінку /: Коренева сторінка сайту перенаправляється на ?p=uk/main. Це може призвести до погіршення користувацького досвіду і порушення рекомендацій SEO. Рекомендується перевірити налаштування редіректу і забезпечити правильне перенаправлення на головну сторінку.

- Файл robots.txt: Файл містить 2 записи, які варто додатково перевірити. Файл robots.txt використовується для керування поведінкою пошукових роботів на сайті. Неправильна конфігурація цього файлу може призвести до небажаних наслідків, таких як некоректне індексування сторінок або виток конфіденційної інформації.

- Потенційно цікаві ресурси: Виявлено деякі ресурси, які можуть містити важливу інформацію або потребують додаткового дослідження, такі як /config/, /admin/, /pub/, /public/, /setup/, /admin/index.php, /wp-admin/, /phpmyadmin/, /wordpress/#wp-config.php#. Ці ресурси можуть бути ціллю зловмисників, тому

рекомендується перевірити їх з точки зору безпеки і впевнитися, що вони належним чином захищені.

3.3 Виправлення вразливостей

Основні проблеми в захисті сайту полягають в недостатній конфігурації і налаштуванні веб-додатку та сервера. Нижче наведено загальний огляд проблем та пропущених моментів у захисті, OWASP ZAP:

- Hash Disclosure: Замініть слабі алгоритми хешування, такі як MD5 або SHA-1, на більш сильні, такі як SHA-256. Додайте сіль до хешу, щоб зробити його більш унікальним і менш піддається підбору.

- PII Disclosure: Перевірте, чи не відбувається ненавмисне розкриття персонально визначеної інформації (ПІІ). Приховуйте або шифруйте ПІІ відповідно до найкращих практик безпеки.

- X-Content-Type-Options Header Missing: Встановіть заголовок X-Content-Type-Options зі значенням "nosniff", щоб запобігти виконанню небезпечного контенту.

- Missing Anti-clickjacking Header: Встановіть заголовок "X-Frame-Options" зі значенням "DENY" або "SAMEORIGIN", щоб запобігти атакам clickjacking.

- HTTPS to HTTP Insecure Transition in Form Post: Забезпечте, щоб всі форми на веб-сайті передавалися по захищеному з'єднанню HTTPS, уникайте переходу з HTTPS до незахищеного з'єднання HTTP.

- Cross-Domain Misconfiguration: Налагодьте належні політики міжсайтової безпеки (CORS) і обмежте доступ до ресурсів з інших доменів.

- Content Security Policy (CSP) Header Not Set: Встановіть заголовок Content-Security-Policy, щоб обмежити виконання небезпечного контенту зовнішніми ресурсами.

- Absence of Anti-CSRF Tokens: Використовуйте захист від CSRF, додавши токени анти-CSRF до всіх форм і запитів.

- Cookie No HttpOnly Flag, Cookie Without Secure Flag, Cookie without SameSite Attribute: Встановіть правильні атрибути cookie, включаючи HttpOnly, Secure і SameSite, для забезпечення відповідної безпеки передачі та доступу до cookie.

- Private IP Disclosure: Переконайтеся, що конфігурація сервера не розкриває приватні IP-адреси.

- Secure Pages Include Mixed Content: Переконайтеся, що всі ресурси на захищених сторінках завантажуються через захищене з'єднання HTTPS.

- Strict-Transport-Security Header Not Set: Встановіть заголовок Strict-Transport-Security, щоб вимагати використання тільки захищеного з'єднання HTTPS.

- Timestamp Disclosure – Unix: Забезпечте, щоб конфігурація сервера не розкривала зайвої інформації, такої як час сервера.

- X-Content-Type-Options Header Missing: Встановіть заголовок X-Content-Type-Options зі значенням "nosniff", щоб запобігти виконанню небезпечного контенту.

Nikto

- Відсутній заголовок X-Frame-Options: Рекомендується додати заголовок X-Frame-Options до відповіді сервера зі значенням "SAMEORIGIN" або "DENY". Це допоможе запобігти атакам clickjacking.

- Відсутній заголовок Strict-Transport-Security: Рекомендується додати заголовок Strict-Transport-Security до відповіді сервера зі значенням "max-age=<кількість секунд>", щоб встановити політику строгого захисту з'єднання. Це забезпечить використання лише безпечного з'єднання TLS.

- Відсутній заголовок X-Content-Type-Options: Рекомендується додати заголовок X-Content-Type-Options до відповіді сервера зі значенням "nosniff". Це дозволить браузеру відображати вміст сторінки відповідно до MIME-типу і запобігатиме можливим атакам XSS.

- Редірект на головну сторінку /: Рекомендується перевірити правильність налаштування редіректу з кореневої сторінки на ?p=uk/main. Упевніться, що редірект виконується з кодом статусу 301 або 302 і правильно перенаправляє користувача на головну сторінку.

- Файл robots.txt: Рекомендується перевірити зміст файлу robots.txt і переконатися, що налаштування відповідають потребам сайту. Впевніться, що конфігурація не розкриває небажану інформацію або не обмежує доступ до важливих ресурсів.

- Потенційно цікаві ресурси: Рекомендується перевірити безпекові налаштування ресурсів, таких як /config/, /admin/, /pub/, /public/, /setup/, /admin/index.php, /wp-admin/, /phpmyadmin/, /wordpress/#wp-config.php#. Забезпечте, щоб ці ресурси були належним чином захищені, мали обмежений доступ і не містили конфіденційну інформацію.

Nmap

Для вирішення проблем, пов'язаних з відкритими портами HTTP (порт 80) і HTTPS (порт 443), рекомендую вжити такі кроки:

- Перевірте налаштування файрволу: Переконайтеся, що на вашому сервері або мережевому пристрої правильно налаштований файрвол, щоб дозволити з'єднання через порти 80 і 443. Переконайтеся, що правила файрволу дозволяють вхідний трафік на ці порти.

- Оновлення веб-сервера і програмного забезпечення: Переконайтеся, що ви використовуєте останню стабільну версію веб-сервера, такого як Apache, Nginx або IIS, а також оновлені версії всього програмного забезпечення, пов'язаного з веб-сайтом (фреймворки, CMS і т.д.). Регулярні оновлення

допоможуть виправити вразливості, які можуть бути відомі і доступні для експлуатації.

- Налаштування безпеки веб-сервера: Переконайтеся, що ваш веб-сервер налаштований з належними заходами безпеки, такими як:
    a) Використання безпечних протоколів: Встановіть SSL-сертифікат і налаштуйте HTTPS для шифрування комунікації між клієнтами і сервером.
    b) Налаштування коректних заголовків безпеки: Додайте належні заголовки безпеки, такі як X-Frame-Options, X-XSS-Protection, X-Content-Type-Options і Strict-Transport-Security, щоб захистити веб-сайт від різних атак.
    c) Контроль доступу і автентифікація: Налаштуйте правильні правила доступу до файлів і каталогів, а також використовуйте надійну систему автентифікації, наприклад, за допомогою сильних паролів і двофакторної аутентифікації.
- Виправлення вразливостей: Виконайте сканування вразливостей веб-сайту за допомогою спеціалізованих інструментів, таких як OWASP ZAP або Burp Suite. Це допоможе виявити можливі вразливості в програмному забезпеченні, налаштуваннях сервера або додатках, які використовуються на веб-сайті. Після виявлення вразливостей вжити відповідні заходи для їх виправлення.
- Перевірка конфігурації сервера: Переконайтеся, що конфігурація веб-сервера і додатків правильна і безпечна. Перегляньте налаштування файлів конфігурації, таких як httpd.conf або nginx.conf, і впевніться, що всі параметри встановлені на безпечні значення і відповідають вашим потребам.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при отруєннях

Долікарська допомога при отруєннях є надзвичайно важливою для надання першої допомоги людям, які стали жертвами отруєння різного походження. Отруєння можуть бути спричинені різними речовинами, включаючи харчові отрути, хімічні речовини, отруйні гази та інші отруйні речовини.

Кроки долікарської допомоги при отруєннях:

- Забезпечте безпеку. Перш за все, переконайтеся, що середовище безпечне для вас та потерпілого. Якщо отруєння сталося через хімічні речовини або отруйні гази, переконайтеся, що немає небезпеки подальшого отруєння.

- Викличте медичну допомогу. Якщо отруєння важке або потенційно небезпечне для життя, негайно викличте медичну допомогу. Зателефонуйте до служби екстреної медичної допомоги та повідомте про ситуацію.

- Відокреміть постраждалого від отруйного джерела. Якщо можливо, переконайтеся, що постраждалий перебуває в безпечному середовищі та не продовжує отримувати отруйну речовину.

- Надайте першу допомогу: Залежно від типу отруєння, надайте першу допомогу, дотримуючись загальних принципів першої допомоги. Наприклад, якщо отруєння сталося через употребу харчової отрути, залежно від симптомів, можуть знадобитися заходи, такі як промивання шлунка, надання речовини для розведення отрути тощо.

- Зберігайте дані: Якщо це можливо, зберігайте будь-які рештки отруйної речовини, яку постраждалий вживав, та надайте цю інформацію медичним працівникам для подальшого аналізу.

Долікарська допомога при отруєннях може варіюватися в залежності від типу отруєння та його важкості. Основна мета долікарської допомоги полягає в забезпеченні безпеки потерпілого, наданні першої допомоги для полегшення симптомів та звільнення від отруйної речовини. Ось декілька додаткових кроків, які можуть бути виконані при долікарській допомозі при отруєннях:

- Промивання шлунка: У деяких випадках, коли отруйна речовина ще не всосалася в кров, промивання шлунка може бути корисним. Це може бути зроблено шляхом вживання великої кількості води або спеціальних розчинів для промивання шлунка. Промивання шлунка слід проводити лише за медичними рекомендаціями і під наглядом фахівців.

- Надання активованого вугілля: Активований вугілля є ефективним засобом поглинання отруйних речовин у шлунку та їх виведення з організму. Воно доступне у формі порошку або таблеток і може бути прийняте за рекомендацією медичних фахівців.

- Підтримка дихання та кровообігу: У випадках, коли отруйна речовина впливає на дихання або кровообіг, можуть бути необхідні заходи підтримки життєво важливих функцій. Це може включати надання штучної вентиляції, проведення серцево-легеневої реанімації (СЛР) або введення ліків для стабілізації дихання та серцево-судинної системи.

- Забезпечення психологічної підтримки: Отруєння може бути фізично та емоційно навантажливим для постраждалої особи. Надання психологічної підтримки, співчуття та впевненості може бути важливим аспектом долікарської допомоги.

Важливо зазначити, що це лише загальні рекомендації, а в конкретному випадку необхідно керуватися порадами та інструкціями медичних фахівців. Негайно звертайтеся за медичною допомогою, якщо виникла ситуація отруєння, та дотримуйтесь їх рекомендацій для забезпечення безпеки та швидкого відновлення здоров'я.

Крім перерахованих вище методів, важливим аспектом долікарської допомоги при отруєннях є також:

- Збереження доказів: При підозрі на отруєння, важливо зберегти будь-які матеріали або продукти, які можуть бути пов'язані з отруєнням, для подальшого аналізу. Це можуть бути залишки їжі, напої, речовини, яка була вжита, або контейнери зі шкідливими речовинами. Збереження доказів може бути корисним для встановлення причини отруєння та при необхідності правових або медичних дій.

- Перевірка наявності протидійних засобів: В залежності від типу отруйної речовини, існують спеціальні протидійні засоби, які можуть бути застосовані для нейтралізації дії отруєння або зменшення його наслідків. Наприклад, у випадку отруєння отруйними газами, можуть бути доступні маски або респіратори для захисту органів дихання.

- Консультація зі спеціалістами: В деяких випадках, особливо при важкому отруєнні або коли немає чітких вказівок щодо долікарської допомоги, може бути необхідна консультація зі спеціалістами в галузі токсикології або отруєнь. Вони можуть надати додаткові рекомендації та вказівки щодо лікування та подальшого управління ситуацією.

- Забезпечення безпеки і запобігання отруєнням є важливою складовою долікарської допомоги. Ось декілька додаткових заходів, які можна вжити:

- Організація освіти та інформування: Розповсюдження інформації про потенційні ризики отруєння і заходи безпеки може значно знизити ймовірність виникнення отруєнь. Освіта має бути спрямована як на дорослих, так і на дітей, і повинна включати пояснення про небезпеку певних речовин, правила зберігання та вживання продуктів, а також процедури поводження у випадку отруєння.

- Постійний моніторинг і оновлення: Світ токсикології постійно змінюється, і нові речовини, їх способи використання та наслідки можуть виникати з часом. Тому важливо забезпечувати постійний моніторинг і

оновлення інформації про потенційні отруйні речовини та методи їх запобігання. Це можна зробити шляхом підписки на спеціалізовані журнали, участі в семінарах або отримання актуальної інформації від відповідних медичних організацій.

- Впровадження відповідних нормативно-правових актів: Уряди та міжнародні організації мають розробляти та впроваджувати нормативно-правові акти, що регулюють використання та обіг отруйних речовин. Це може включати заборону або обмеження використання деяких речовин, встановлення стандартів безпеки для їх виробництва та зберігання, а також встановлення відповідальності за недбале використання отруйних речовин.

- Постійний моніторинг якості продуктів: Забезпечення якості продуктів, які широко використовуються в домашньому господарстві, є важливим аспектом запобігання отруєнням. Виробники та владні органи повинні здійснювати постійний моніторинг якості продуктів та вживати необхідних заходів для виявлення та вилучення потенційно небезпечних продуктів з ринку.

4.2 Природне середовище і його забруднення

Природне середовище - це комплекс природних ресурсів, який включає атмосферу, води, ґрунти, ліси, рослини, тварин та інші живі організми, що існують на Землі. Воно є надзвичайно важливим для підтримання життя на планеті і забезпечення стабільного функціонування екосистем.

Однак, природне середовище постійно піддається впливу різних факторів, які можуть його забруднювати. Забруднення природного середовища - це введення в нього шкідливих речовин, які негативно впливають на життя рослин, тварин і людей.

Існує багато джерел забруднення природного середовища, серед яких:

- промислова діяльність. Викиди в атмосферу і скиди вод від промислових підприємств містять різні токсичні речовини, такі як сірководень, важкі метали, хлоровані сполуки тощо. Ці речовини можуть потрапляти в ґрунти, водні ресурси та накопичуватися в живих організмах, що призводить до порушення екосистем та загрози здоров'ю людей;

- відходи та сміття. Неконтрольоване звалищування відходів і неправильна утилізація сміття призводять до забруднення ґрунтів, поверхневих і підземних вод. Також у процесі розкладу органічних відходів утворюються шкідливі гази, такі як метан, які впливають на глобальне потепління та зміну клімату;

- викиди транспорту. Транспортні засоби, особливо автомобілі, викидають в атмосферу забруднюючі речовини, такі як вуглеводні, оксиди азоту та сірки. Це спричиняє забруднення повітря і сприяє формуванню смогу та інших проблем здоров'я.

- сільське господарство. Використання пестицидів, фунгіцидів та інших хімічних речовин у сільському господарстві може призводити до забруднення ґрунтів та водних ресурсів. Відходи тваринництва також можуть впливати на якість водних систем через високу концентрацію нітратів і фосфатів.

- експлуатація природних ресурсів. Добування вуглеводнів, руд, деревини та інших природних ресурсів супроводжується викидами токсичних речовин, руйнуванням екосистем та втратою біорізноманіття.

Приведу до прикладу екологічну катастрофу яку спричинила русня на Каховській ГЕС. Уночі 6 червня, війська РФ здійснили вибух на Каховській ГЕС, що призвело до зруйнування конструкцій станції. Машинна зала станції не може бути відновлена, і це призводить до загрози підтоплення для близько 80 населених пунктів. Знищення Каховської ГЕС та його наслідки, які ви описали, дійсно мають серйозні наслідки для сільськогосподарського сектору, водопостачання, рибного господарства та екології.

Затоплення орієнтовно 10 тисяч гектарів сільськогосподарських земель на правобережжі Херсонщини і значно більшої площі на лівому березі, який перебуває під окупацією, призведе до серйозних втрат для сільського господарства. Таке затоплення може призвести до значного зменшення врожайності та втрати доходів для сільських господарств.

Зупинка водопостачання 31 системи зрошення полів Дніпропетровської, Херсонської та Запорізької областей також має серйозні наслідки для сільського господарства. Втрата можливості зрошення на такій великій площі може призвести до зменшення врожайності та втрати доходів для сільськогосподарських підприємств, а також загрози продовольчій безпеці регіону.

Знищення Каховської ГЕС також має серйозні наслідки для рибного господарства. Загибель риб, включаючи молодь і дорослі особини, і висихання ікри на змілілих ділянках можуть призвести до значного зменшення рибних ресурсів та втрати доходів для риболовлі. Крім того, переміщення фауни водосховища у заплави, які потім опиняться на суходолі, також може спричинити загибель цих біоресурсів.

Потрапляння прісної води у солоні води Чорного моря спричинить загибель прісноводної риби і інших біоресурсів також становитиме серйозну екологічну проблему. Це може мати вплив на біорізноманіття та екологічну рівновагу в регіоні.

Усі ці наслідки будуть відчуватися на протязі кількох років і потребуватимуть часу для відновлення. Відновлення рибних популяцій, донних ценозів та екосистеми загалом є складним процесом, який потребує уваги та заходів з відновлення екологічного балансу.

Ці наслідки підкреслюють важливість охорони та стабільності водних ресурсів, а також необхідність узгоджених заходів для зменшення ризиків подібних техногенних катастроф у майбутньому.

Потрапляння 150 тонн мастила до річки Дніпро має серйозні наслідки для водного екосистеми та природного середовища. Основні загрози, які можуть виникнути в результаті цього забруднення, включають наступне:

- Загроза для водних організмів: Мастило, утворюючи плівку на поверхні води, може перешкоджати процесам дихання та функціонуванню водних організмів, таких як риба, водні комахи, водорості та інші водні організми. Це може спричинити масову загибель риби та інших водних організмів.

- Порушення екологічного балансу: Забруднення мастилом може порушити природний баланс у водній екосистемі. Воно може впливати на харчові ланцюги та знижувати різноманіття видів, що може мати далекосяжні наслідки для екологічної стійкості регіону.

- Загроза питної води: Дніпро є джерелом питної води для багатьох населених пунктів. Забруднення мастилом може погрожувати якісною та безпечною водопостачанням, що ставить під загрозу здоров'я людей та санітарну ситуацію.

- Вплив на берегові зони: Мастило може діставатись до берегових зон, завдаючи шкоди прибережній рослинності та тваринам, які залежать від цього середовища. Це може мати негативний вплив на екосистему річкових берегів.

Утилізація та усунення мастила є складними процесами, які вимагають спеціалізованого обладнання та екологічно безпечних методів. Час потрібен для вивчення масштабів забруднення та визначення найефективніших заходів для його ліквідації.

Враховуючи, що вплив мастила також залежатиме від лінії фронту, що проходить вздовж Дніпра, важливо координувати зусилля з усунення забруднення між військовими, екологічними та урядовими органами для мінімізації наслідків цієї катастрофи.

Знищення Каховської ГЕС та підрив дамби мають серйозні наслідки для подачі води в окупований Крим. Північнокримський канал, який є основним

джерелом води для півострова, починається неподалік від зруйнованої дамби. Завдяки цьому каналу Крим отримував питну воду та воду для сільськогосподарського зрошення.

Окупаційна влада Криму намагається запевнити про наявність достатньої кількості питної води, але знищення Каховської ГЕС унеможливлює подачу води через північнокримський канал. Оскільки водосховище Каховської ГЕС зменшується з кожною хвилиною, це призводить до зниження обсягу води, що надходить до Криму. Така ситуація загрожує погіршенням якості водопостачання та може викликати проблеми з питною водою на півострові.

Порушення системи водопостачання може мати серйозні наслідки для населення, аграрного сектору та екосистеми Криму. Брак води може призвести до обмеження доступу до питної води, засухи, зниження врожайності сільськогосподарських культур та загрози екологічного стану природних водних ресурсів.

Ця ситуація підкреслює необхідність знаходження рішення для вирішення проблеми водопостачання Криму, зокрема шляхом відновлення і реконструкції водних інфраструктур, а також сприяння врегулюванню конфлікту і поверненню контролю над водними ресурсами регіону законним владам України.

ВИСНОВКИ

Дослідження та тестування захищеності веб-сайту ТНТУ було проведено з метою виявлення потенційних вразливостей, оцінки ризиків та розробки рекомендацій щодо поліпшення безпеки системи. В результаті аналізу було встановлено, що веб-сайт має деякі недоліки, які можуть призвести до порушення конфіденційності, цілісності та доступності даних користувачів.

Серед виявлених вразливостей можна виділити використання слабких хеш-алгоритмів для збереження паролів, які можуть бути легко розшифровані зловмисниками. Також була виявлена недостатня конфігурація безпеки сервера, що збільшує ризик злому та несанкціонованого доступу до системи. Крім того, було виявлено вразливості, пов'язані з розголошенням персонально визначеної інформації (ПІІ), що може порушити приватність користувачів та стати причиною крадіжки особистих даних.

Для забезпечення належного рівня безпеки веб-сайту ТНТУ рекомендується вжити невідкладних заходів. Перш за все, необхідно оновити алгоритми хешування паролів, використовуючи більш міцні та надійні методи шифрування. Також необхідно провести аудит конфігурації безпеки сервера, забезпечити використання рекомендованих налаштувань та патчів безпеки для усунення вразливостей. Важливо регулярно оновлювати програмне забезпечення, включаючи всі компоненти веб-сайту, а також встановити механізми моніторингу та виявлення вторгнень для своєчасного виявлення потенційних загроз.

При проведенні тестування захищеності веб-сайту важливо мати на увазі, що це неперервний процес, оскільки нові загрози і вразливості постійно з'являються. Регулярне тестування і оновлення безпеки допомагають забезпечити надійний рівень захисту веб-сайту і його користувачів.

В цілому, забезпечення високого рівня захищеності веб-сайту ТНТУ є невід'ємною складовою частиною ефективного функціонування системи та забезпечення довіри користувачів. Правильне впровадження рекомендацій щодо безпеки допоможе зменшити ризик кібератак, витоку конфіденційної інформації та порушення безпеки даних, сприяючи безпечному та надійному середовищу для користувачів сайту ТНТ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code". Ligh, M., Adair, S., Hartstein, B., & Richard, M. (2010).
2. "Nmap Network Scanning: The Official Nmap Project Guide". Gordon Lyon. (2009).
3. "OWASP ZAP: Guided Penetration Testing and Ethical Hacking". Mehul Revankar.
4. "Social Engineering: The Science of Human Hacking". Chris Hadnagy. (2009).
5. "Web Application Security Testing with Burp Suite". Akash Mahajan.
6. Becher, T., Bortolameotti, R., Franchino, G. "Security Analysis of Learning Management Systems" (2019).
7. Nikto installer URL: https://cirt.net/Nikto2
8. Nikto documentation URL: https://cirt.net/Nikto2/documentation.html
9. Kali Linux download and intall on Virtual Machine URL: https://help.offsec.com/hc/en-us/articles/360049796792-Kali-Linux-Virtual-Machine
10. Nmap Reference Guide URL: https://nmap.org/book/man.html
11. Official Burp Suite site url:https://portswigger.net/burp
12. Lopes, R. H. C., Tanaka, K., Kim, K. G. "Security Analysis of Online Learning Platforms". (2017).
13. Maiorca, D., Mariconti, E., Onali, T. - "Security Analysis of Massive Open Online Courses". (2018).
14. Massive Open Online Courses, MOOC URL: https://www.mooc.org/
15. Testing Guide OWASP ZAP URL: https://owasp.org/www-project-web-security-testing-guide/