Міністерство освіти і науки України
Харківський національний університет імені В.Н. Каразіна
Факультет комп'ютерних наук
Спеціальність 125 «Кібербезпека»
Освітня програма «Безпека інформаційних та комунікаційних систем»

«Допущено до захисту»
В.о. зав. кафедрою БІСТ
Ольга МЕЛКОЗЬОРОВА

_____

« » 2023 р.

Пояснювальна записка

до кваліфікаційної роботи магістра

на тему: «Тестування на безпеку веб-сайтів, ідентифікація ризиків та застосування ефективних методів захисту»

оцінка « »
Голова ЕК
Олександр ЛЕМЕШКО _____

Керівник :
доц. кафедри БІСТ Нарєжній О.П.
Рецензент:
PhD, старш. викл. каф. штучного інтелекту та програмного забезпечення
Лисицький К.Є.
Виконавець : студентка групи КБ-61
Тимошенко Яна Сергіївна

Харків – 2023

РЕФЕРАТ

Пояснювальна записка містить 74 сторінки, 28 рисунків, 0 таблиць, 1 додаток, 28 джерел.

Метою дипломної роботи є дослідження та проведення активного сканування на безпеку для веб-сайту та демонстрації найпоширеніших вразливостей веб-сайту. Підходи до ідентифікації ризиків та визначення оптимальних методів захисту будуть проаналізовані та можуть бути взяті за основу для створення ефективного захисту.

Об’єктом дослідження дипломної роботи є веб-сайти, які забезпечують доступ до інформаційних ресурсів, послуг чи продуктів через Інтернет.

Предметом розробки є безпека веб-сайтів. Це означає, що в роботі буде здійснено звертання до аспектів, пов'язаних з захистом веб-сайтів від різних загроз і атак, таких як крос-сайтовий скриптінг, перехоплення сесій, атаки на внутрішні мережі і багато інших. Дане дослідження включає в себе такі аспекти, як аналіз вразливостей веб-сайтів, методи їхньої експлуатації та обрання відповідних методів захисту.

Методи дослідження: пенетраційні тести (Penetration Testing) веб-сайту bWAPP, сканування вразливостей (Vulnerability Scanning) веб-сайту bWAPP.

Результатами проведеної роботи є аналіз сучасних загроз і ризиків веб-сайтів, демонстрація вразливостей, методи запобігання, та активне сканування веб-сайту.

Ключові слова: КІБЕРБЕЗПЕКА, ВЕБ-САЙТИ, ТЕСТУВАННЯ НА БЕЗПЕКУ, РИЗИКИ, ВРАЗЛИВОСТІ, ЗАХИСТ, МЕТОДИ АНАЛІЗУ, КІБЕРЗАГРОЗИ, ІДЕНТИФІКАЦІЯ ЗАГРОЗ, ЕФЕКТИВНІСТЬ ЗАХОДІВ.

ABSTRACT

The explanatory note contains 74 pages, 28 figures, 0 tables, 1 appendices, 28 sources.

The aim of the thesis is to research and conduct an active security scan for a website and demonstrate the most common website vulnerabilities. Approaches to identifying risks and determining optimal protection methods will be analysed and can be used as a basis for creating effective protection.

The object of research of the thesis is websites that provide access to information resources, services or products via the Internet.

The subject of development is website security. This means that the work will address aspects related to protecting websites from various threats and attacks, such as cross-site scripting, session hijacking, attacks on internal networks, and many others. This study includes such aspects as analysing website vulnerabilities, methods of exploitation, and choosing appropriate protection methods.

Research methods: Penetration Testing of the bWAPP website, Vulnerability Scanning of the bWAPP website.

The results of this work include an analysis of current threats and risks to websites, demonstration of vulnerabilities, prevention methods, and active website scanning.

Keywords: CYBERSECURITY, WEBSITES, SECURITY TESTING, RISKS, VULNERABILITIES, PROTECTION, ANALYSIS METHODS, CYBER THREATS, THREAT IDENTIFICATION, EFFECTIVENESS OF MEASURES.

ЗМІСТ

ВСТУП

Кібербезпека стала однією з найбільш актуальних і важливих проблем сучасного інформаційного суспільства, оскільки щодня збільшується кількість кіберзагроз, інцидентів та атак на веб-сайти та інформаційні системи. Ця несподівана, але залякуюча реальність служить джерелом натхнення для цієї роботи і визначає важливу мету цих досліджень у галузі кібербезпеки.

Всесвітня пандемія COVID-19 змусила суспільство ще більше розраховувати на віртуальну присутність та використовувати веб-сервіси для найрізноманітніших цілей. В цьому контексті зростає значимість захисту даних і інформаційних ресурсів, особливо на веб-платформах.

Ця тема є дуже актуальною тому, що оперуючи в інтернеті, як користувачі, ми довіряємо веб-сайтам нашу особисту інформацію, фінансові дані та конфіденційну інформацію. Тому надійний захист цих веб-сайтів стає ключовою задачею. Водночас, атаки на веб-сайти стають більш вдосконаленими та складними. Інциденти, пов'язані з витоками даних, кібервикраденням та іншими формами кіберзлочинності, набувають частоти і масштабів, які раніше були неможливими. У цьому контексті об'єктом досліджень є комплексний підхід до тестування на безпеку веб-сайтів.

Мета дослідження полягає в тому, щоб розробити комплексний підхід до тестування на безпеку веб-сайтів, який дозволить ідентифікувати потенційні ризики та слабкі місця, а також виявити недоліки в застосовуваних методах захисту. Крім того, метою є розробка пропозицій щодо ефективних методів та стратегій захисту, які забезпечать надійну безпеку веб-сайтів.

Для досягнення цієї мети розглядаються наступні основні завдання проекту:

1) Активне сканування на безпеку веб-сайту: Проведене активне сканування на безпеку веб-сайту дозволило виявити ряд вразливостей, серед яких важливими є недостатня аутентифікація та слабкі місця в системах контролю доступу. Виявлені вразливості включають також недоліки в програмному забезпеченні, що потребують негайного виправлення;

2) Демонстрація найпоширеніших вразливостей веб-сайту: Найпоширеніші вразливості, виявлені під час сканування, були детально продемонстровані. Серед них варто відзначити SQL-ін’єкції, Cross-Site Scripting (XSS), а також вразливості, пов'язані з некоректною обробкою сесій та витоком конфіденційної інформації. Це дозволяє отримати повністю обґрунтований огляд та розуміння потенційних небезпек для безпеки веб-сайту;

3) Аналіз підходів до ідентифікації ризиків: Проведений аналіз підходів до ідентифікації ризиків дозволяє визначити основні причини вразливостей та потенційні точки атак. Це важливий крок для створення системи захисту, спрямованої на подолання конкретних загроз;

4) Оптимальні методи захисту: Розглянуті та проаналізовані різні методи захисту, такі як двофакторна аутентифікація, регулярні апдейти програмного забезпечення, використання багатофакторного контролю доступу. Визначено оптимальні методи захисту, які забезпечують високий рівень безпеки та дозволяють ефективно уникати атак;

5) Перспективи розвитку та вдосконалення системи безпеки: Обговорено перспективи розвитку та вдосконалення системи безпеки веб-сайту, включаючи впровадження нових заходів безпеки, вдосконалення процесів моніторингу та реагування на потенційні загрози;

6) Обговорення результатів: Здобуті результати підкреслюють важливість систематичного тестування та ідентифікації ризиків для забезпечення ефективного захисту веб-сайтів. Розглянуті методи захисту є необхідним елементом в сучасному інформаційному середовищі. Отримані відомості можуть слугувати основою для подальшого розвитку стратегій безпеки та підвищення стійкості веб-платформ перед різноманітними загрозами.

1 РОЗВИТОК КІБЕРБЕЗПЕКИ У СФЕРІ ВЕБ-ТЕХНОЛОГІЙ

Кібербезпека – це ключовий аспект у нашому сучасному цифровому світі, який став невід'ємною частиною нашого повсякденного життя. Все більше і більше загроз нависає над нами в онлайн-середовищі з кожним роком, і вони стають все більш витонченими та хитрими. У зв'язку з цим забезпечення безпеки інформації стає критично важливим завданням, яке стосується не лише організацій, а й усіх нас зокрема.

У сучасному світі люди все більше залежать від цифрових технологій. Вони зберігають особисті дані, проводять фінансові операції, обмінюються повідомленнями та здійснюють онлайн-покупки, що робить їх уразливими перед різноманітними видами кіберзагроз. Кіберзлочинці та хакери постійно вдосконалюють свої методи, тому захист цифрових активів та особистої інформації стає безперервним та важливим процесом.

Ефективна кібербезпека необхідна як для державних організацій, так і для приватних осіб. Важливо усвідомлювати ризики та вживати заходів щодо захисту від них, будь то використання сильних паролів, встановлення антивірусного програмного забезпечення, оновлення програм та операційних систем, а також навчання у сфері кібербезпеки. У результаті забезпечення безпеки в цифровому світі стає одним із головних завдань нашого часу, і його необхідно приймати всерйоз.

Організація кібератаки протягом перших двох десятиліть після розробки першого у світі цифрового комп'ютера в 1943 році була складною задачею. У той час величезні електронні машини працювали ізольовано, не були інтегровані в мережі, і доступ до них був обмежений лише невеликою групою фахівців, які мають знання про те, як ними керувати. У зв'язку з цим кібербезпека практично не викликала занепокоєння, оскільки можливість масових кібератак здавалася практично неможливою [22].

Цікавить той факт, що теорія, що лежить в основі створення комп'ютерних вірусів, була опублікована в 1949 році. Цей крок було зроблено піонером у галузі комп'ютерних технологій, Джоном фон Нейманом, який припустив, що комп'ютерні

програми мають потенціал для самовідтворення. Це відкриття стало початком нової епохи у кібербезпеці та підготувало ґрунт для розвитку кібератак та створення комп'ютерних вірусів, які стали серйозними загрозами у наступні десятиліття.

Вперше випадок зловмисного злому було зареєстровано у студентській газеті Массачусетського технологічного інституту. У середині 1960-х років більшість комп'ютерів були величезними мейнфреймами, розміщеними в приміщеннях з ретельно контрольованим мікрокліматом. Ці машини були дуже дорогими, і доступ до них залишався обмеженим, навіть для програмістів.

Тим не менш, ті, кому вдалося отримати доступ (найчастіше це були студенти), почали проводити свої перші експерименти зі зломом. У той час атаки не мали ні комерційної, ні геополітичної вигоди. Здебільшого хакери були просто цікавими порушниками або тими, хто прагнув удосконалити існуючі системи, збільшуючи їх швидкість та ефективність.

1967 року компанія IBM запросила школярів випробувати свій новий комп'ютер. Після вивчення доступної частини системи, учні почали глибше проникати в її внутрішню структуру, вивчаючи її програмний код та одержуючи доступ до інших компонентів системи. Цей інцидент був цінним уроком для компанії IBM, яка висловила подяку "кільком старшокласникам за їхнє невгамовне бажання перевірити систему на міцність." Ця подія стала стимулом для розробки додаткових заходів безпеки і, можливо, заклала основи для більш акцентованої уваги до аспектів кібербезпеки, яка стала невід'ємною частиною роботи розробників у наступні десятиліття. Етичний хакінг чи злом, відтоді продовжував практикуватися та розвиватися [22, 23].

З розвитком технологій комп'ютери стали зменшуватися в розмірах і ставати доступнішими для широкої публіки. Багато великих компаній почали інвестувати в технології зберігання даних та систем управління ними. Разом з цим зросла потреба у забезпеченні безпеки даних, оскільки дедалі більше людей потребував постійного доступу до інформації. У цей момент почали активно використовувати паролі та інші методи автентифікації для захисту цінних даних та систем.

Поняття "кібербезпеки" почало формуватися в період з 1960-х по 1970-і роки, паралельно з розвитком комп'ютерних технологій та створенням перших

комп'ютерних мереж. У цей час комп'ютери стали все більш поширеними та пов'язаними між собою, що сприяло появі нових загроз безпеці. Перші комп'ютерні віруси почали з'являтися, і це змусило фахівців почати розробляти засоби та методи захисту від загроз.

Один з перших комп'ютерних вірусів, відомий як Creeper, був створений в 1971 році і поширювався на комп'ютерах, підключених до мережі ARPANET (мережа агентства перспективних дослідницьких проектів - попередника інтернету) (рис 2.1).

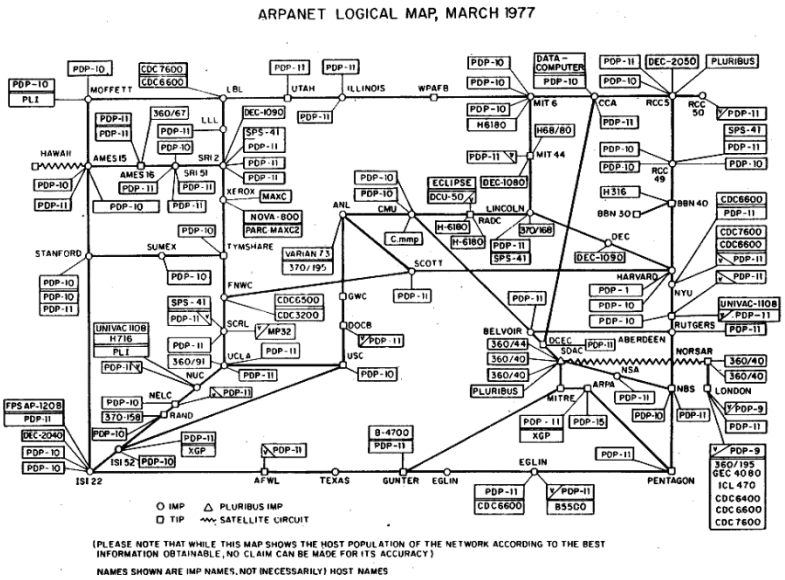

Рисунок 2.1 – Перша загальнодоступна комп'ютерна мережа ARPANET

Цей вірус, незважаючи на свою руйнівну здатність, був відносно невинним у порівнянні із сучасними малварями. Його основною функцією було виводити на екран повідомлення "I'm the creeper, catch me if you can!" («Я привид, зрозумій мене, якщо зможеш!») (рис 2.2).

```
BBN-TENEX 1.25, BBN EXEC 1.30
@FULL
@LOGIN RT
JOB 3 ON TTY12 08-APR-72
YOU HAVE A MESSAGE
@SYSTAT
UP 85:33:19   3 JOBS
LOAD AV   3.87   2.95   2.14
JOB TTY  USER      SUBSYS
1   DET  SYSTEM    NETSER
2   DET  SYSTEM    TIPSER
3   12   RT        EXEC
@
I'M THE CREEPER : CATCH ME IF YOU CAN
```

Рисунок 2.2 - Приклад послання програми Creeper

Однак поява Creeper змусила фахівців почати розробляти методи протидії таким загрозам, і в результаті було створено перший антивірус із іронічним ім'ям Reaper. Цей етап історії комп'ютерної безпеки наголошує на важливості постійного розвитку засобів захисту та боротьби з кіберзагрозами, які стали невід'ємною частиною сучасного світу інформаційних технологій.

Із зростанням популярності сучасних технологій, усунення вразливостей стало наполегливим завданням, тому що все більше організацій вдалося до використання мобільних пристроїв та телефонів для створення віддалених мереж та зв'язку. Кожна нова одиниця обладнання, підключена до мережі, набувала статусу потенційної "точки входу" для потенційних атак і, отже, вимагала комплексних заходів безпеки [23, 25].

У час збільшення залежності від комп'ютерів та мереж стало очевидно, що забезпечення безпеки стало ключовою необхідністю. Несанкціонований доступ до даних та систем міг спричинити катастрофічні наслідки. У період з 1972 по 1974 роки почалася наростаюча дискусія щодо комп'ютерної безпеки, яка зосереджувалася головним чином в урядових колах. Усвідомлення важливості захисту даних та систем призвело до появи нових методів та технологій у галузі інформаційної безпеки.

Державні агентства ESD та ARPA за участю ВПС США та інших організацій зробили значний внесок у колективну розробку системи комп'ютерної безпеки для Honeywell Multics (рівень HIS 68). Шляхом спільних зусиль Каліфорнійський університет у Лос-Анджелесі та Стенфордський дослідницький інститут також брали активну участь в аналогічних проектах.

В рамках проекту ARPA, орієнтованого на аналіз захисту, проводилися дослідження в галузі безпеки операційних систем та пошук методів автоматичного виявлення вразливостей у програмному забезпеченні.

Концепція кібербезпеки до середини 1970-х років досягла стадії зрілості, що позначилося в публікації під назвою "Operating System Structures to Support Security and Reliable Software" (Структури операційних систем для забезпечення безпеки та надійності програмного забезпечення), опублікованій у 1976 році. У ній

наголошувалося, що забезпечення безпеки стало важливим та складним аспектом розробки комп'ютерних систем.

У період з 1980-х по 1990-і роки у XX столітті питання кібербезпеки стали все більш суттєвими та актуальними. Це десятиліття відзначилося сплеском інтересу до комп'ютерної безпеки, оскільки з'явилося безліч хакерів, націлених на пошук вразливостей та порушення цілісності інформаційних систем.

На початку 1980-х років спостерігалося помітне збільшення гучних кібератак, включаючи атаки на такі організації, як National CSS, AT&T та Los Alamos National Laboratory. Особливу увагу привернув фільм 1983 під назвою "Військові ігри", де комп'ютерна програма, яка видається за гру, загрожувала захопленням ядерних ракетних систем. У цей же рік були вжиті терміни "троянський кінь" і "комп'ютерний вірус", позначивши початок нової епохи в області комп'ютерних атак і шкідливих програм [24, 25].

Цей період також пов'язаний із наростаючими загрозами кібершпигунства в умовах холодної війни. У 1985 році Міністерство оборони США опублікувало "Критерії визначення безпеки комп'ютерних систем", які часто називають "Помаранчевою книгою", яка надала рекомендації з наступних питань:

- оцінка довіри до програмного забезпечення, що обробляє секретну або конфіденційну інформацію;
- заходи безпеки, які виробникам необхідно було впровадити у свої комерційні продукти задля забезпечення безпеки інформації.

Тим не менш, у 1986 році німецький хакер на ім'я Маркус Хес продемонстрував уразливості в системі, використовуючи інтернет-шлюз у каліфорнійському місті Берклі, щоб отримати доступ до ARPANET. Його атака була спрямована на цінні військові комп'ютери, включно з мейнфреймами в Пентагоні, і його метою було продати скомпрометовану інформацію КДБ.

Цей інцидент став точкою перегляду безпеки у світі комп'ютерів та інтернету. Досвідчені користувачі почали звертати серйознішу увагу на деталі, такі як розмір файлу command.com, і помітили, що його збільшення могло бути першою ознакою потенційного зараження вірусами або шкідливим ПЗ. Цей спосіб думки згодом

сформував базові принципи кібербезпеки, включаючи моніторинг вільної оперативної пам'яті як один з індикаторів атак, і ці заходи актуальні і в наші дні.

Одним із перших документованих випадків успішного видалення вірусу "в реальних умовах" була дія Германа Бернда Фікса, який зміг знешкодити шкідливе програмне забезпечення з ім'ям Vienna. Цей вірус Vienna став одним із перших зразків шкідливого програмного забезпечення, яке поширювалося та пошкоджувало файли [22].

Примітно, що тоді вперше з'явився шифрований вірус під назвою Cascade, який інфікував файли з розширенням .COM. Через рік, Cascade став причиною серйозного інциденту в офісі компанії IBM в Бельгії, що послужило стимулом для розробки антивірусних продуктів цієї компанії. До цього моменту антивірусні рішення, створені IBM, призначалися виключно для внутрішнього використання і не мали поширення. Таким чином, поява Cascade та його вплив призвели до перегляду стратегії в галузі антивірусної безпеки всередині IBM та стимулювали розвиток антивірусних продуктів для боротьби з подібними загрозами.

У 1988 році світ вперше зіткнувся із серйозною кібератакою, яка залишила величезне враження і стала точкою відліку для подальших розробок у галузі кібербезпеки. Ця історична атака, відома як атака черв'яка Морріса (рис 2.3), була важливою подією в історії інформаційної безпеки. Розроблений Робертом Моррісом, цей черв'як став першим комп'ютерним черв'яком масштабу Інтернету, який успішно поширився, вразивши тисячі комп'ютерів [23].

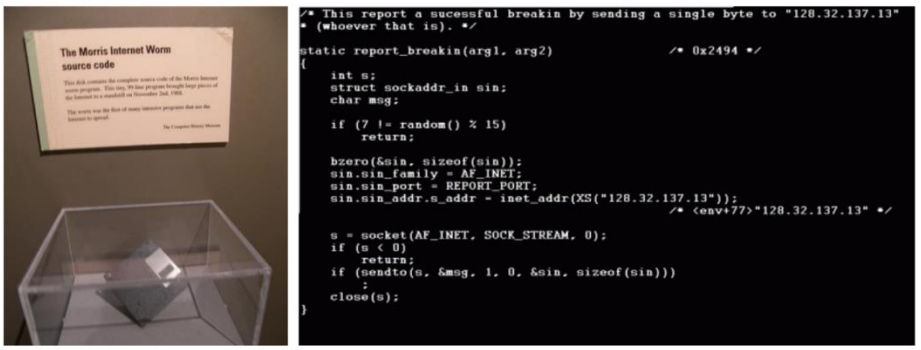

Рисунок 2.3 - Комп'ютер заражений вірусом Morris

Атака черв'яка Морріса призвела до широкого усвідомлення невідкладної необхідності зміцнення безпеки у сфері інформаційних технологій. Комп'ютерні системи того часу були недостатньо захищені від таких загроз, і атака наголосила на

реальності вразливості мереж. Ця подія змусила фахівців та організації розробляти та впроваджувати ефективні заходи безпеки, щоб запобігти подібним інцидентам у майбутньому. Таким чином, атака хробака Морріса вплинула на розвиток кібербезпеки і стала відправною точкою для довгої дороги в напрямку забезпечення безпеки в цифровому світі.

Цього ж року в Празі було засновано компанію Avast Едуардом Кучерою та Павлом Баудішем, і з того часу вона пережила вражаючий розвиток. На даний момент до команди Avast входить понад 1700 видатних фахівців, що працюють у різних куточках світу. Її продукти з успіхом справляються з блокуванням близько 1,5 мільярда атак щомісяця, що є значною цифрою і свідчить про їх важливу роль у забезпеченні кібербезпеки[23].

На зорі свого існування антивірусне програмне забезпечення являло собою прості сканери, які проводили контекстний пошук для виявлення унікальних послідовностей вірусних кодів. Багато з таких сканерів також включали "імунізатори", які впроваджували зміни в програми, щоб змусити віруси вважати комп'ютер вже зараженим і, таким чином, відмовлятися від атаки.

Однак із збільшенням числа вірусів до сотень ці імунізатори швидко втратили свою ефективність. Антивірусним компаніям також доводилося усвідомлювати, що їхня здатність реагувати обмежена існуючими атаками, і відсутність універсальної глобальної мережі, такої як Інтернет, робила ускладненим розгортання оновлень та забезпечення безпеки в онлайн-просторі.

Згодом еволюція кіберзагроз приносить із собою більш складні та витончені загрози. Хакери постійно вдосконалюють свої методи атак, включаючи такі методики, як фішинг, використання шкідливих програм та соціальна інженерія з метою дістатися до цінних даних, включаючи особисту інформацію, фінансові ресурси та корпоративні секрети[22].

У відповідь на цю загрозу, що постійно зростає, організації та уряди активно розробляють і впроваджують стратегії та заходи для забезпечення безпеки інформації та запобігання кібератакам. У сучасному світі кібербезпека стала самостійною та важливою дисципліною, в рамках якої фахівці спеціалізуються на виявленні, запобіганні та реагуванні на кіберзагрози. Ці експерти розробляють та

впроваджують захисні заходи, моніторять мережі та ресурси на предмет підозрілої активності, а також беруть активну участь у розробці стратегій для ефективного реагування на інциденти в галузі кібербезпеки. Кібербезпека стала незамінною частиною сучасного світу, і її важливість лише збільшується у міру розвитку цифрових технологій та зростання числа кіберзагроз.

В 1990 році відбулося кілька ключових подій, які суттєво вплинули на сферу комп'ютерної безпеки:

- На той час з'явилися перші поліморфні віруси - програмний код, який змінювався, зберігаючи при цьому початковий алгоритм, щоб уникнути виявлення;
- Британський комп'ютерний журнал PC Today випустив безкоштовний диск, на якому "випадково" опинився вірус DiskKiller, який інфікував десятки тисяч комп'ютерів;
- В тому ж році було засновано Європейський інститут комп'ютерних антивірусних досліджень (EICAR).

Антивірусні програми перших років свого існування базувались виключно на процесі порівняння двійкових файлів у операційній системі з вірусними сигнатурами, які були збережені в базі даних програми. Цей підхід призвів до значної кількості помилкових спрацювань, оскільки антивіруси не завжди правильно розпізнавали потенційні загрози. Крім того, ці програми вимагали великих обчислювальних ресурсів, що внаслідок цього знижувало продуктивність комп'ютерів і призводило до розчарування серед користувачів.

Коли кіберзлочинці врахували зростання кількості антивірусних сканерів на ринку, у 1992 році виникла перша програма, спрямована на боротьбу саме з цими антивірусами. Їх стратегія була спрямована на адаптацію до змін у кількості захисних систем, і це призвело до створення вищезгаданої програми, що спеціально призначалася для контрдії антивірусним заходам [23, 25].

К кінцю 1996 року багато вірусів вже використовували нові техніки та інноваційні методи, такі як прихована робота, поліморфізм і "макровіруси". Це становило новий виклик для розробників антивірусних програм, які повинні були знаходити нові можливості для виявлення та видалення загроз. Однією із завдань

стало розроблення ефективних стратегій у боротьбі із зростаючою складністю та різноманіттям вірусних атак.

У 1990-х роках спостерігався різкий зліт числа нових вірусів та шкідливих програм: від кількох десятків на початку десятиліття до п'яти мільйонів щороку до 2007 року. До середини 90-х стало зрозуміло, що кібербезпека має стати пріоритетом на масовому рівні. У відповідь на цей виклик один дослідник з НАСА розробив перший брандмауер, спираючись на принципи захисту, аналогічні тим, що використовуються у фізичних конструкціях для запобігання розповсюдженню вогню від однієї будівлі до іншої у разі реальних пожеж.

У боротьбі з багатьма варіаціями вірусів активно впроваджується евристичне виявлення як новий метод. Антивірусні сканери тепер використовують спільні сигнатури, які включають несуміжні фрагменти коду та символи підстановки. Це дозволяє виявляти віруси навіть у випадках, коли загроза прихована всередині безглуздого коду. Таким чином, евристичне виявлення надає ефективний інструмент боротьби з різноманітними формами шкідливих програм.

На початку кінця 1990-х років електронна пошта активно поширювалася, передбачаючи перспективу радикальних змін у сфері комунікацій. Разом із обіцянкою революції в обміні повідомленнями вона також несподівано розкрила новий рівень вразливості, який став вхідною точкою для різних вірусів.

В 1999 році з'явився вірус Melissa. Він заражав комп'ютер користувача через документ Word, а потім відправляв свої копії електронною поштою на перші 50 адрес, взяті з Microsoft Outlook (рис 2.4).

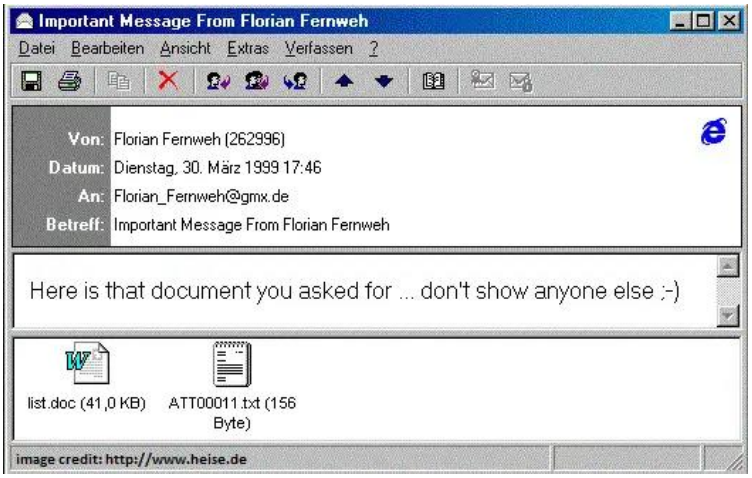

Рисунок 2.4 – Вірус Melissa

Мелісса залишається одним з найшвидше поширюваних вірусів, а виправлення завданого збитку обійшлося приблизно в 80 мільйонів доларів. Цей вірус отримав свою назву від автора вірусу, який використовував псевдонім "Kwyjibo" і вбудував в текст вірусу фрагменти з мультфільму "The Simpsons". Вірус поширювався через електронну пошту, а його розповсюдження привело до значного обмеження роботи комп'ютерних систем. Melissa була вироблена як макрос для Microsoft Word і вкладена в інфікованих документах, які мали вигляд листів або документації, що викликало цікавість у потенційних жертв. При відкритті документа вірус автоматично виконував свої деструктивні дії, розсилаючи копії самого себе по електронній пошті з адрес жертв. Вірус Melissa став визначним подією в історії комп'ютерних вірусів і має кілька характеристик, які заслуговують на увагу [23]:

- швидке поширення: Melissa вражав комп'ютери, використовуючи масове розсилання інфікованих документів через електронну пошту. Його ефективність полягала в тому, що вірус швидко поширювався серед контактів жертв, що прискорило його розповсюдження;
- деструктивний вплив: Після інфікування Melissa автоматично вставляв свій макрос у всі документи Microsoft Word, які були доступні на зараженому комп'ютері. Це може призвести до серйозного ураження робочих файлів та втрати даних;
- спосіб імплементації: Вірус використовував макроси в документах Word для виконання своїх злоякісних функцій. Це було типовим для макровірусів, які використовували слабкі місця в програмах для виконання своїх дій;
- судові наслідки: Автор вірусу, Девід Л. Сміт, був затриманий і засуджений в 1999 році. Він визнав свою провину і був засуджений до 20 місяців ув'язнення та штрафу в $5,000. Справа Melissa викликала значний інтерес з боку ЗМІ та галузі інформаційної безпеки;
- навчальний ефект: Виявлення вірусу Melissa призвело до виникнення нових підходів до боротьби зі злоякісним програмним забезпеченням. Антивірусні компанії вивчали його код і розробляли оновлення для своїх продуктів, щоб уникнути подібних загроз у майбутньому.

Дослідження Melissa підкреслило важливість освіти та навчання в галузі інформаційної безпеки. Користувачам комп'ютерів та організаціям стали наочною необхідність усвідомлення ризиків, пов'язаних із зловживанням електронною поштою та іншими цифровими засобами. Багато компаній та організацій вдосконалили свої стратегії кібербезпеки. Антивірусні програми стали більш ефективними у виявленні та нейтралізації подібних загроз. Також зростала свідомість користувачів про важливість остеріганняться підозрілого контенту в електронних повідомленнях. Можна сказати, що вірус Melissa відзначився не лише як серйозна загроза для тодішніх комп'ютерних систем, але й як каталізатор для подальших зусиль у сфері кібербезпеки. Його виникнення прискорило розвиток технологій захисту від вірусів та інших форм зловживання в цифровому середовищі, сприяючи створенню більш безпечного інтернет-простору для користувачів у подальшому.

З початком 2000-х років доступ до Інтернету розширюється в усе більше домів та офісів по всьому світу, відчиняючи перед кіберзлочинцями небачені раніше можливості використання вразливостей пристроїв та програмного забезпечення. Із зростанням кількості цифрово зберіганих даних збільшується потенційна сфера для їхнього видобутку. Іншими словами, збільшується обсяг потенційного набутку кіберзлочинців, оскільки вони можуть використовувати цей розвиток для ефективнішого здійснення атак та вторгнень.

В 2001 році з'явилася нова методика інфікування. Тепер користувачам не потрібно було завантажувати файли, достатньо було просто відвідати заражений веб-сайт. Зловмисники заміняли чисті сторінки зараженими або приховували шкідливе програмне забезпечення на порядних веб-сторінках. Служби миттєвого обміну повідомленнями також почали ставати об'єктом атак. З'явилися черви, призначені для поширення через канал IRC (Internet Chat Relay). Ці нові методи стали підступнішими, адже вони не вимагали активного завантаження від користувачів, і виявилися більш ефективними в розповсюдженні шкідливого програмного забезпечення.

Розвиток атак нульового дня, які експлуатують уразливості у новому програмному забезпеченні та додатках, призвело до зниження ефективності

антивірусів. Пошук шкідливого коду з використанням існуючих сигнатур став скрутним, оскільки відсутність цього вірусу в базі даних робить його невидимим для традиційних методів виявлення. У журналі c't зазначено, що відсоток виявлення погроз нульового дня скоротився з 40–50% у 2006 році до всього лише 20–30% у 2007 році. Ця тенденція свідчить про зростаючу складність боротьби з передовими формами шкідливого програмного забезпечення [25].

З появою активного фінансування професійних кібератак з боку злочинних організацій представники "білих" хакерів активізували свої зусилля у боротьбі з погрозами. Знаковим моментом у цьому контексті став 2000 рік, коли було представлено перший антивірусний двигун з відкритим вихідним кодом у рамках проекту OpenAntivirus. Слідом за цим у 2001 році з'явився ClamAV, що став першим у світі комерціалізованим антивірусним двигуном з відкритим вихідним кодом. Також у цей період компанія Avast зробила значний внесок, представивши безкоштовне антивірусне рішення, забезпечивши повнофункціональний захист широкого кола користувачів. Ініціатива Avast призвела до значного зростання кількості користувачів, що перевищило позначку 20 мільйонів за п'ять років.

Однією з ключових проблем антивірусів було те, що вони часто знижували продуктивність комп'ютера. Для вирішення цього питання виникла ідея переміщення програмного забезпечення з комп'ютера на хмару. У 2007 році компанія Panda Security вперше в історії поєднала хмарні технології з аналізом загроз у своєму антивірусному продукті. За цим прикладом слідувала компанія McAfee Labs у 2008 році, додаючи функцію хмарної захисту від вредних програм у VirusScan. Наступного року було створено Anti-Malware Testing Standards Organization (AMTSO) — організацію, яка встановлює стандарти тестування захисту від вредного програмного забезпечення.

Протягом останнього десятиліття в галузі інформаційної безпеки стали спостерігати важливий розвиток - впровадження кіберзахисту в операційні системи, що надає додатковий рівень безпеки для збереження даних. Цей прогрес охоплює регулярне оновлення операційних систем, встановлення патчів, удосконалення антивірусних механізмів та програмного забезпечення, а також впровадження

брандмауерів та систем захисту облікових записів з функцією управління користувачами.

З підвищенням популярності смартфонів було створено та впроваджено антивірусні програми, спеціально призначені для операційних систем Android та Windows Mobile. Це значно зміцнило загальний рівень кіберзахисту пристроїв, що працюють під управлінням зазначених операційних систем.

Протягом десятиліття, що минуло з 2010 по 2019 рік, світ став свідком ряду вибухових інцидентів, пов'язаних із витоками даних та кібератаками, які вразливо вдарили по національній безпеці численних країн і призвели до серйозних фінансових втрат для безлічі компаній. Наприклад, у 2012 році хакер із Саудівської Аравії, відомий під псевдонімом 0XOMAR, виклав в інтернет інформацію про понад 400 000 кредитних карток.

У 2013 році колишній працівник ЦРУ на ім'я Едвард Сноуден, що працював на урядових посадах у Сполучених Штатах, скопіював та розкрив секретні дані Агентства національної безпеки (АНБ). З 2013 по 2014 рік кіберзлочинці вдарили по електронній поштовій службі Yahoo, отримавши доступ до облікових записів та особистої інформації трьох мільярдів користувачів. Внаслідок цього Yahoo була оштрафована на суму 35 мільйонів доларів за утаєння цих фактів [22].

У 2017 році програма-вимагач WannaCry заразила 230 000 комп'ютерів всього за один день. У 2019 році численні атаки типу DDoS призвели до тимчасової призупинення фондового ринку Нової Зеландії.

Зі збільшенням доступності до Інтернету та постійним процесом цифровізації різних сфер життя кіберзлочинці отримали нові можливості для здійснення атак. Щоб відповісти на ці виклики, розробка кіберзахисту, спеціально призначеного для бізнесу, набула ще більшого значення. Вже у 2011 році компанія Avast випустила свій перший продукт, спрямований на задоволення потреб у сфері бізнесу. З метою протистояти все більш різноманітним формам кібератак, зловмисники відповіли власними інноваціями, такими як багатовекторні атаки та використання соціальної інженерії. Кожного дня злочинці стають все винахідливішими, примушуючи антивірусні програми переходити від традиційних методів виявлення, що базуються на сигнатурах, до стратегій "нового покоління".

Кіберзахист нового покоління охоплює широкий спектр стратегій, спрямованих на вдосконалення виявлення як нових, так і раніше невідомих загроз, а також на зменшення кількості помилкових спрацьовувань. Нижче перераховано кілька типових компонентів цього вдосконаленого кіберзахисту:

- Багатофакторна автентифікація (MFA): Одним із ключових елементів є використання багатофакторної аутентифікації, що підвищує рівень безпеки шляхом підтвердження особи через декілька джерел;
- Мережевий поведінковий аналіз (NBA): Цей компонент ґрунтується на виявленні шкідливих файлів шляхом аналізу відхилень або аномалій у їх поведінці у мережі;
- Аналіз загроз та автоматизація оновлень: Акцент робиться на систематичному аналізі потенційних загроз, з наступною автоматизованою реакцією та оновленням системи;
- Захист у реальному часі: Цей аспект, також відомий як сканування при включенні, фоновий, резидентний або автоматичний захист забезпечує миттєве реагування на можливі загрози;
- Використання пісочниці: Використання ізольованих тестових середовищ, де безпечно можна запускати підозрілі файли або URL-посилання, сприяє додатковому рівню безпеки;
- Технічна експертиза: Повторне відтворення атак та аналіз слабких місць допомагають створити більш стійку систему та запобігати витокам у майбутньому;
- Резервне копіювання та використання дзеркал: Регулярне резервне копіювання даних та використання дзеркал сприяють швидкому відновленню після атаки;
- Брандмауери веб-додатків (WAF): Надання захисту від міжсайтової підробки запитів, міжсайтового скриптингу (XSS), включення файлів та впровадження SQL дозволяє запобігати атакам на веб-додатки.

Сфера кібербезпеки постійно розвивається та адаптується до постійно змінюючихся загроз та викликів. Останнім часом спостерігається значний приріст

складних кібератак, спрямованих на різні рівні - від держав до корпорацій та приватних осіб.

Разом із впровадженням новітніх технологій, таких як штучний інтелект та блокчейн, ситуація в галузі кібербезпеки стає ще більш складною. Ці інновації можуть представляти нові можливості для захисту та виявлення загроз, але одночасно створюють нові вразливості та завдання для забезпечення безпеки.

Необхідність у співпраці та обміні інформацією стає ключовою в розвитку кібербезпеки. Тільки через спільні зусилля та обмін досвідом можна ефективно протистояти сучасним кіберзагрозам. Встановлення тісного партнерства між організаціями та державами є критичним елементом в забезпеченні колективного захисту від кібернападів.

2 АНАЛІЗ СУЧАСНИХ ЗАГРОЗ І РИЗИКІВ ВЕБ-САЙТІВ

Веб-сайти є невід'ємною частиною інтернету, і вони стають об'єктом постійних кіберзагроз і ризиків через свою доступність та важливість для суспільства. Розглянемо докладніше деякі типові загрози та ризики, які становлять серйозну загрозу веб-сайтам.

2.1 Типові загрози для веб-сайтів

1) Denial of Service, DoS та Distributed Denial of Service. DDoS атаки, Denial of Service (DoS) і Distributed Denial of Service (DDoS) атаки є способами нападу на комп'ютерну систему або мережу з метою перекриття або обмеження доступу користувачів до ресурсів цієї системи. Ці атаки спрямовані на заборону доступу або зниження доступності ресурсів, які нормально доступні користувачам, і можуть призвести до недоступності важливих веб-сайтів, служб, серверів тощо.

Denial of Service (DoS) атака: Сутність DoS атаки полягає в перевантаженні цільової системи або ресурсу, зазвичай за допомогою великої кількості запитів чи команд. Це може призвести до відмови у обслуговуванні легітимних користувачів, оскільки ресурси системи будуть витрачені на обробку шкідливих запитів [2, 3, 27].

Зазвичай, атаки DoS можуть бути такими:

- Синтаксичні атаки: Надсилання некоректних запитів або даних, що призводить до падіння системи;
- Атаки на ресурси: Захоплення або використання всіх ресурсів системи, наприклад, велика кількість одночасних підключень до сервера;
- Флуд атаки: Відправлення великої кількості запитів або даних до цільової системи, забираючи її ресурси.

2) Distributed Denial of Service (DDoS) атака: Сутність: DDoS атаки є розширеною версією атаки DoS, де зловмисники використовують багато комп'ютерів, які називаються "ботами" або "зомбі", для скоординованого нападу на цільову систему. Зазвичай, боти контролюються здалеку і вони великої кількості, що робить атаку набагато потужнішою та складнішою для виявлення та захисту. Деякі з типових DDoS атак включають:

- UDP флуд: Завитяження цільової системи запитами на протокол User Datagram Protocol (UDP), який легко підробити та піддається атакам;
- TCP флуд: Використання великої кількості підроблених TCP-підключень для перевантаження сервера;
- HTTP/HTTPS флуд: Атака на веб-сервери, надсилаючи багато запитів на конкретну веб-сторінку чи службу;
- DNS ампліфікація: Зловмисники використовують DNS-сервери для відправлення запитів на велику кількість інших серверів, змушуючи їх відправляти відповіді на ці запити до цільової системи.

3) Кросс-сайтовий скриптинг (Cross-Site Scripting, XSS) - це тип вразливості веб-додатків, який дозволяє зловмисникам впроваджувати злоякісний JavaScript-код у веб-сторінки, які потім виконуються на браузерах інших користувачів. Ця вразливість може виникнути, коли вхідні дані від користувачів не валідуються належним чином і відображаються на сторінці без достатнього фільтрування та екранування. XSS атаки можуть бути розділені на три основні категорії:

- Stored XSS (постійний XSS): У цьому типі атаки злоякісний код зберігається на сервері і відображається при кожному відвідуванні вразливої сторінки. Наприклад, зловмисник може внести злоякісний скрипт у коментарі на блогу, і кожен, хто переглядає цей коментар, може бути атакований;
- Reflected XSS (відображений XSS): У цьому випадку злоякісний код внесений в вхідні дані і відображається лише для користувачів, які виконують певний дію або переходять за певним посиланням. Наприклад, злоякісне посилання, що містить скрипт, може бути розіслане по електронній пошті або розміщене на форумах. Якщо користувачі перейдуть за цим посиланням, скрипт виконається у їхньому браузері;
- DOM-based XSS (XSS, що ґрунтується на DOM): У цьому випадку злоякісний скрипт впливає на Document Object Model (DOM) сторінки після завантаження. Ця вразливість використовується для атак на клієнтський JavaScript,

і зазвичай вона стає можливою через недоліки в коді JavaScript, який використовується на сторінці.

Шкідливі наслідки XSS атак включають в себе можливість викрадення сесійних cookie, які можуть бути використані для підробки ідентифікації користувача, перенаправлення користувачів на фішингові сторінки, внесення змін у вміст сторінки, збирання конфіденційної інформації з браузера користувача тощо.

Для запобігання XSS атакам важливо валідувати та екранувати вхідні дані на стороні сервера, використовувати HTTP заголовки Content Security Policy (CSP) для обмеження виконання скриптів, і ніколи не довіряти вхідним даним без обов'язкової перевірки. Також, регулярно виявляйте та виправляйте потенційні вразливості в програмному забезпеченні.

4) SQL-ін'єкції (SQL Injection) - це тип вразливості веб-додатків, при якій зловмисники можуть впроваджувати злоякісні SQL-запити в вхідні дані додатку. Ця вразливість може виникнути, коли вхідні дані від користувачів не валідуються належним чином або не екрануються перед використанням в SQL-запитах. Якщо злоякісні SQL-запити вдалося впровадити в базу даних, це може призвести до витоку, модифікації або видалення даних в базі даних. Основні принципи SQL-ін'єкції:

- Некоректна обробка вхідних даних: Вразливість виникає, коли вхідні дані, такі як рядки, які вводяться користувачами через веб-форми або URL-параметри, безпосередньо включаються в SQL-запити без достатньої перевірки та екранування;
- Зміна логіки запиту: Злоякісні користувачі можуть вставляти в SQL-запити спеціальні символи, які дозволяють їм контролювати запит та виконувати небажані дії. Наприклад, вони можуть модифікувати умови запиту, вилучити дані, виконати команди тощо;
- Потенційно серйозні наслідки: SQL-ін'єкція може призвести до витоку конфіденційної інформації, такої як паролі, дані користувачів, кредитні картки тощо. Також, злоякісні користувачі можуть завдати серйозної шкоди базі даних, видаливши чи модифікувавши важливі дані.

5) Вразливості в автентифікації. Вразливості в автентифікації становлять серйозну загрозу для безпеки інформаційних систем. Автентифікація - це процес перевірки ідентифікаційних даних користувача (таких як ім'я користувача та пароль) з метою підтвердження його особи та надання доступу до системи або ресурсів. Вразливості в автентифікації можуть призвести до незаконного доступу до системи, витоку конфіденційної інформації та інших серйозних проблем з безпекою. Основні види вразливостей в автентифікації:

- Паролі низької складності: Ця вразливість виникає, коли система дозволяє користувачам встановлювати або використовувати слабкі паролі, такі як "12345" або "password". Злоякісні користувачі можуть легко вгадати або зламати такі паролі;
- Брутфорс і атаки на перебір: Це спроби злому автентифікації, коли атакуючий намагається вгадати пароль, спробуючи різні комбінації паролів шляхом повторних спроб. Важливо встановлювати обмеження на кількість невдалих спроб введення паролю та вживати заходів проти брутфорсу;
- Витік сесій: Ця вразливість виникає, коли сесійні ідентифікатори або токени не захищені належним чином і можуть бути вкрадені. Злоякісний користувач може використовувати вкрадений ідентифікатор для отримання доступу до аккаунта іншого користувача.
- Недостатні перевірки ідентифікації: Вразливість виникає, коли система не належним чином перевіряє ідентифікаційні дані користувача. Наприклад, якщо система не вимагає підтвердження електронною поштою при реєстрації акаунта, злоякісний користувач може створити обліковий запис зі сторонньою електронною адресою.
- Залишкові сесії і некоректне видалення доступу: Якщо система не належним чином закриває сесії після виходу користувача або не видаляє доступ, коли це потрібно (наприклад, при зміні паролю), то можуть виникати вразливості, через які атакуючі можуть залишати доступ до системи.

6) Перехоплення даних (Data Breaches) - це ситуація, коли конфіденційна чи особиста інформація стає доступною для незаконних осіб через несанкціонований доступ до інформаційних систем, баз даних, архівів або інших

місць зберігання даних. Ця загроза представляє серйозну проблему для безпеки і приватності індивідів, підприємств та організацій. Внаслідок перехоплення даних може статися витік конфіденційної інформації, такої як особисті дані, фінансова інформація, медичні записи, корпоративні секрети, інтелектуальна власність та багато іншого. Основні характеристики та види перехоплення даних:

- Типи даних: Перехоплення даних може включати в себе різноманітні типи інформації, включаючи особисті дані (ім'я, адресу, соціальний номер тощо), фінансову інформацію (кредитні картки, банківські реквізити), медичні дані, корпоративну інформацію та інше;
- Заходи з обмеження доступу: Вразливості в системах аутентифікації та авторизації можуть призвести до незаконного доступу до даних. Якщо атакуючі отримають доступ до системи як легітимні користувачі, вони можуть мати можливість переглядати, копіювати або навіть видаляти дані;
- Вразливості в програмному забезпеченні: Помилки в програмному забезпеченні, недоліки безпеки або застарілі програмні компоненти можуть служити витоком для атак. Це може включати в себе SQL-ін'єкції, уразливості веб-додатків, недоліки в програмах для обробки даних і т.д;
- Витік інформації через зовнішні джерела: Перехоплення даних може виникнути через зовнішні джерела, такі як злоякісні програми, фішингові атаки, витоки з внутрішніх джерел, включаючи необдумані дії або зловживання правами доступу співробітників;
- Інциденти з втратою даних: Інциденти з втратою даних, такі як крадіжки комп'ютерів або пристроїв зберігання даних, можуть призвести до фізичного доступу до даних.

7) Broken Access Control" (порушення контролю доступу) в інформаційній безпеці відноситься до вразливостей, пов'язаних з недостатньою або некоректною реалізацією механізмів контролю доступу до ресурсів у веб-додатках або інших інформаційних системах. Ця загроза може призвести до незаконного доступу до конфіденційної інформації, зміни налаштувань, видалення даних або виконання інших небажаних дій. Основні причини виникнення "Broken Access Control" включають:

- Недостатній контроль доступу: Деякі ресурси і функціональності веб-додатка можуть бути доступні без належного перевірки ідентифікації та автентифікації користувачів;
- Некоректна авторизація: Якщо користувач отримав доступ до системи, але йому дозволено виконувати дії, які йому не повинно бути дозволено на основі його ролі або привілеїв, це також є видом "Broken Access Control";
- Некоректна перевірка доступу: В деяких випадках, навіть після ідентифікації та автентифікації користувача, система може не вірно перевіряти права доступу до певних ресурсів або функціональностей;
- Небезпечні стандартні налаштування: Веб-додатки часто включають стандартні налаштування для спрощення розгортання та використання. Проте, ці стандартні налаштування можуть призвести до порушення контролю доступу, якщо не налаштовані належним чином.

Наслідки "Broken Access Control" можуть бути серйозними, включаючи втрату конфіденційної інформації, руйнування даних, порушення конфіденційності та інтегритету даних, а також негативний вплив на репутацію організації. Для захисту від цієї загрози, рекомендується внедрювати належні механізми контролю доступу, відповідно налаштовувати права користувачів, та ретельно перевіряти і тестувати систему на наявність порушень контролю доступу під час розробки та експлуатації веб-додатків та інших інформаційних систем.

8) Криптографічні невдачі (Cryptographic Failures) - це ситуації, коли криптографічні методи, які застосовуються для захисту інформації та забезпечення безпеки, стають недостатньо ефективними або піддаються атакам через якісь недоліки, помилки в реалізації або зміни у загрозах. Ці невдачі можуть мати серйозні наслідки для безпеки даних, інформаційних систем та комунікацій. Типи криптографічних невдач:

- Атаки на алгоритми шифрування: Деякі криптографічні алгоритми можуть стати непридатними через розвиток обчислювальної потужності, атаки на математичні основи алгоритмів або знайдення слабких точок в їх конструкції;

- Порушення ключового управління: Недостатня або неправильна управління ключами може призвести до витоку секретної інформації або недостатньої захищеності даних;
- Недостатнє випадкове число: Важливою складовою криптографії є випадкові числа, і якщо генератори випадкових чисел не є досить випадковими, це може створити слабкість в криптографічних системах;
- Вразливості в реалізації: Помилки в реалізації криптографічних алгоритмів, такі як баги в програмному забезпеченні або неналежне конфігурування, можуть призвести до небажаних витоків інформації;
- Соціальний інженерінг і атаки на криптографічні ключі: Навіть найсильніші криптографічні системи можуть бути обмануті через атаки на людські фактори або витоки ключів;
- Регуляторні обмеження і надзвичайні ситуації: Законодавство або державні дії можуть призвести до обмежень або змін у криптографічних системах, що зроблять їх менш ефективними для забезпечення безпеки.

Криптографічні невдачі є постійною загрозою для безпеки інформації і технологічних систем. Тому важливо постійно оновлювати та вдосконалювати криптографічні методи та системи, а також дотримуватися кращих практик в області криптографії для забезпечення найвищого рівня безпеки.

9) Загроза "Insecure Design" або "небезпечний дизайн" в інформаційній безпеці вказує на ситуацію, коли інформаційна система, програмне забезпечення або апаратне забезпечення мають недостатньо ефективний дизайн, який може створити потенційні ризики для безпеки та конфіденційності даних. Ця загроза може включати в себе такі аспекти:

- Недостатній контроль доступу: Неякісний дизайн може призвести до недостатнього контролю над тим, яка інформація доступна для користувачів та як вона обробляється. Це може стати причиною витоків конфіденційних даних;
- Недостатні заходи безпеки: Небезпечний дизайн може призвести до відсутності або недостатнього рівня захисту від зловмисників. Наприклад,

відсутність аутентифікації, шифрування або слабкі паролі можуть використовувати зловмисники для отримання доступу до системи;

- Неякісна обробка даних: Недостатньо обдуманий дизайн може призвести до некоректної обробки даних, що може призвести до вразливостей і атак. Наприклад, недостатні перевірки введених даних можуть призвести до атак на введення даних;
- Слабка архітектура: Неякісна архітектура системи може робити її вразливою до різних видів атак, включаючи атаки на висновки, переповнення буфера, а також атаки на архітектурні вразливості;
- Відсутність планування на випадок інцидентів: В дизайні системи може бути недостатньо уваги до планування заходів безпеки та реагування на інциденти. В такому випадку, інциденти можуть бути неефективно управлятися, що призводить до потенційних втрат і збоїв у роботі системи.

Для зменшення ризиків, пов'язаних зі "загрозою Insecure Design", важливо включити аналіз безпеки та вдосконалення в дизайн і розробку програмного забезпечення або інформаційних систем. Також важливо використовувати найкращі практики в галузі інформаційної безпеки і регулярно аудитувати систему на наявність потенційних вразливостей.

10) Security Misconfiguration (Неправильна конфігурація безпеки) є однією з найпоширеніших загроз безпеці в інформаційних технологіях, і вона полягає в неправильній конфігурації програмного забезпечення або систем. Ця загроза може виникати внаслідок помилок в налаштуванні серверів, баз даних, мережевих пристроїв, апаратного забезпечення та інших компонентів інфраструктури, що забезпечують функціонування системи. Основні аспекти цієї загрози включають:

- За замовчуванням налаштувані параметри: Багато систем та програм мають значення за замовчуванням, які можуть бути небезпечними, якщо не змінити їх на безпечні. Зловмисники можуть використовувати ці значення для отримання несанкціонованого доступу або виконання атак;
- Неналежна авторизація та ідентифікація: Неправильна конфігурація прав доступу може призвести до того, що зловмисники зможуть звертатися до ресурсів чи функцій, до яких не мають доступу;

- Виток інформації: Недбалість у конфігурації може призвести до ненавмисного витоку конфіденційної інформації, такої як паролі, ключі, чутливі дані, інша конфіденційна інформація;
- Неправильна обробка помилок: Неправильно налаштована обробка помилок може розкрити внутрішні деталі системи, що допоможе зловмисникам здійснити атаку;
- Відкритий доступ до адміністративних інтерфейсів: Якщо адміністративні інтерфейси системи залишаються відкритими для загального доступу або не налаштовуються правильно, це може призвести до компрометації системи.

Щоб захистити систему від загрози security misconfiguration, важливо регулярно перевіряти та аудитувати конфігурацію всіх компонентів системи, виправляти виявлені помилки та встановлювати правила безпеки. Використання автоматизованих інструментів для аналізу конфігурації та моніторингу є також корисним підходом для попередження security misconfiguration.

11) Тип загрози "Vulnerable and Outdated Components" (вразливі та застарілі компоненти) відноситься до сфери кібербезпеки та інформаційної безпеки. Ця загроза стосується програмного забезпечення, яке використовується в різних системах, веб-додатках, серверах, апаратних пристроях та інших інформаційних ресурсах. Вона походить від використання вразливих та застарілих компонентів в програмному забезпеченні. Основні аспекти загрози "Vulnerable and Outdated Components" включають:

- Вразливості програмних компонентів: Застарілі компоненти, які не були оновлені або відділені від давно використовуваного програмного забезпечення, можуть мати вразливості, які можуть бути використані зловмисниками для здійснення атак. Ці вразливості можуть бути використані для отримання несанкціонованого доступу, крадіжки даних, відмови в обслуговуванні та інших видів атак;
- Брак підтримки та оновлень: Постачальники програмного забезпечення можуть припинити підтримку старих версій та компонентів, що робить їх схильними

до вразливостей і атак. Оновлення і патчі важливі для виправлення виявлених вразливостей та забезпечення безпеки;

- Інформаційний ризик: З використанням застарілих компонентів збільшується ризик витоку конфіденційної інформації, порушення нормативних вимог, та інших проблем з безпекою, які можуть призвести до фінансових втрат та пошкодження репутації організації.

Для запобігання цій загрозі, організації повинні вживати наступні заходи:

- Систематичне оновлення: Регулярно оновлювати програмне забезпечення та всі його компоненти, включаючи бібліотеки та залежності, для запобігання вразливостям;
- Моніторинг: Вести моніторинг вразливостей та виправляти їх якомога швидше. Використовуйте інструменти для автоматизації цього процесу;
- Завантаження лише відповідальних джерел: Завантажуйте компоненти програмного забезпечення лише з офіційних джерел та джерел, яким ви довіряєте;
- Моніторинг встановлених компонентів: Постійно відслідковуйте встановлені компоненти та їх версії, щоб вчасно виявляти застарілі або вразливі компоненти.

Забезпечення безпеки від загрози "Vulnerable and Outdated Components" важливо для захисту цифрових ресурсів та збереження конфіденційності, цілісності та доступності інформації.

12) Проблеми з реєстрацією подій та моніторингом безпеки є серйозною загрозою для інформаційної безпеки організації. Цей тип загрози виникає, коли системи або процеси, призначені для збору, аналізу та реагування на події безпеки, не працюють належним чином. Вони можуть бути спричинені різними факторами, такими як технічні проблеми, людські помилки, атаки на самі системи моніторингу, або навіть недостатній обсяг ресурсів для відслідковування всіх подій. Основні аспекти цієї загрози включають:

- Несправні або відключені механізми реєстрації: Якщо механізми реєстрації подій не працюють належним чином, то організація може пропустити важливі події безпеки, такі як незаконні спроби вторгнення або аномальні активності користувачів;

- Недостатня аналітика: Іноді системи моніторингу можуть збирати велику кількість данних, але не мати належних засобів аналізу. Це може призвести до того, що підозрілі активності залишаються непоміченими або не вивчаються належним чином;
- Збільшення обсягу даних: Системи моніторингу можуть швидко стати надмірною навантаженням через великий обсяг подій та даних, що потребують обробки. Це може призвести до втрати даних або до затримок у виявленні і реагуванні на загрози;
- Неправильні налаштування: Неправильне налаштування систем моніторингу та журналювання може призвести до надмірної кількості ложних позитивів або недооцінки реальних загроз.

Для захисту від цього типу загрози важливо мати належну стратегію моніторингу та реєстрації подій, а також регулярно тестувати її на працездатність. Розробка резервних планів та процедур реагування на випадок виникнення проблем з моніторингом і реєстрацією також може допомогти зменшити наслідки цієї загрози для безпеки.

2.2 Огляд методів тестування на безпеку та їх ефективності

Важливо враховувати, що безпека веб-сайту не обмежується лише застосуванням заходів захисту. Надзвичайно важливо проводити регулярне тестування на безпеку, щоб виявити можливі вразливості та ризики. Деякі методи тестування на безпеку включають [4,6]:

1) Пенетраційні тести (Penetration Testing), також відомі як пен-тести або етичне вторгнення, є методом тестування на безпеку, спрямованим на виявлення і оцінку вразливостей і потенційних загроз в інформаційних системах, додатках, мережах та інших компонентах інфраструктури. Метою пенетраційних тестів є встановлення, чи можуть злоякісні користувачі або атакуючі використовувати вразливості для незаконного доступу, витоку інформації або інших зловмисних дій. В цьому процесі використовуються техніки, інструменти та методи, схожі на ті, що використовуються злоякісними хакерами, але з метою тестування і підвищення рівня безпеки. Основні етапи та аспекти пенетраційних тестів:

- Планування: Починаючи з цього етапу, визначається ціль тестування, об'єкти, які підлягають аналізу, і методи, які будуть використовуватися під час тестування. Також узгоджується графік та обсяг робіт;
- Збір інформації (Reconnaissance): Тестери збирають інформацію про цільову систему чи мережу. Це включає в себе сканування портів, визначення IP-адрес, ідентифікацію служб і веб-сайтів, а також збір основної інформації про ціль;
- Аналіз вразливостей: Тестери використовують автоматизовані і ручні методи для виявлення вразливостей в системі. Це може включати в себе аналіз коду, перевірку конфігурації серверів, тестування на проникнення та інші методи;
- Експлуатація вразливостей: Якщо вразливості виявлені, тестери можуть спробувати їх експлуатувати для отримання доступу до системи чи виконання зловмисних дій. Проте це робиться згідно з етичними принципами, і буде зроблено все можливе, щоб не завдати реальних шкідливих наслідків;
- Звітність і рекомендації: Після завершення тестування створюється детальний звіт, який включає в себе виявлені вразливості, їх серйозність та рекомендації щодо їх виправлення. Цей звіт надається власникам системи або додатку для подальших дій;
- Виправлення і тестування на забезпеченість: Власники системи виправляють виявлені вразливості і проводять додаткові тести для перевірки, чи були вони успішно усунені.

2) Сканування вразливостей (Vulnerability Scanning) - це метод тестування на безпеку, який спрямований на виявлення вразливостей і потенційних загроз в інформаційних системах, мережах та додатках. У відміну від пенетраційних тестів, які спрямовані на активне тестування, сканування вразливостей використовується для автоматичного виявлення вразливостей шляхом аналізу системи або мережі. Цей процес надає загальний огляд безпеки та допомагає ідентифікувати потенційні проблеми, які потребують подальшого аналізу та виправлення. Основні характеристики та етапи сканування вразливостей [4,7]:

- Сканування портів: Спершу виконується сканування портів на цільовій мережі чи системі, що допомагає визначити, які служби та сервіси активні та

доступні для аналізу. Це важливо для ідентифікації потенційних точок входу для злоякісних атак;

- Зіставлення з базою даних вразливостей: Після сканування портів інструмент сканування вразливостей зіставляє результати з базою даних відомих вразливостей. Ця база даних містить інформацію про відомі вразливості, включаючи опис, серйозність та можливі наслідки;
- Аналіз результатів: Знайдені вразливості аналізуються, оцінюються за ступенем серйозності та потенційним впливом на безпеку системи. Це допомагає визначити, які вразливості потребують найбільшої уваги і негайних заходів для виправлення;
- Генерація звіту: Після завершення сканування вразливостей генерується докладний звіт, який включає в себе виявлені вразливості, їх опис, рекомендації щодо виправлення та іншу інформацію. Цей звіт надається власникам системи чи мережі для подальших дій;
- Виправлення вразливостей: На основі результатів сканування вразливостей власники системи можуть вживати заходів для усунення знайдених проблем. Це може включати в себе оновлення програмного забезпечення, зміну конфігурації, впровадження патчів безпеки та інші заходи;
- Періодичність сканування: Сканування вразливостей може виконуватися як регулярно, так і за потреби. Регулярні сканування допомагають відстежувати стан безпеки системи та реагувати на нові вразливості або зміни в інфраструктурі.

3) Тестування перевантаження (Load Testing) - це метод тестування на безпеку, який спрямований на визначення, як система чи додаток працює при різних рівнях навантаження та обсязів трафіку. Основною метою цього виду тестування є перевірка, чи здатна система витримати очікуване навантаження без важливих відмов та забезпечити високу доступність та швидкість в реальних умовах експлуатації. Основні характеристики і етапи тестування перевантаження [5,6]:

- Визначення цілей тестування: Спочатку визначаються цілі та очікувані результати тестування. Це може включати в себе визначення максимальної кількості одночасних користувачів, часу відгуку системи та інших метрик продуктивності;

- Створення тестових сценаріїв: Розробляються тестові сценарії, які відображають типову поведінку користувачів системи. Ці сценарії можуть включати в себе дії, такі як пошук, додавання товарів у кошик, авторизація користувача і т.д;
- Налаштування навантаження: За допомогою інструментів для тестування перевантаження створюється навантаження, яке відображає очікувану активність користувачів. Це може бути велика кількість запитів, одночасних з'єднань, обсяг даних тощо;
- Запуск тестування: Тестування запускається з встановленими параметрами навантаження. Під час тестування записуються результати, такі як час відгуку сервера, кількість запитів на секунду та інші метрики продуктивності;
- Аналіз результатів: Після завершення тестування аналізуються результати, щоб визначити, чи витримує система навантаження та які вчасності можуть виникнути при перевантаженні;
- Оптимізація і виправлення помилок: На основі результатів тестування можуть бути внесені зміни в інфраструктуру, програмне забезпечення або конфігурацію для покращення продуктивності і надійності системи;
- Повторне тестування: Після внесення змін чи оптимізацій може бути проведене повторне тестування для перевірки ефективності внесених змін.

2.3 Вивчення методів захисту та їх призначення

Захист веб-сайтів включає в себе використання різноманітних методів і технологій. Деякі з основних методів захисту веб-сайтів включають [10, 12]:

1) Використання HTTPS: HTTPS (Hypertext Transfer Protocol Secure) - це захищений протокол передачі гіпертекстових даних через інтернет. Цей метод захисту використовується для шифрування даних, які передаються між веб-браузером користувача та веб-сервером, що робить їх недоступними для зловмисників, які можуть намагатися перехопити чи модифікувати ці дані. Використання HTTPS важливо для забезпечення конфіденційності, цілісності та автентичності даних, які обмінюються між користувачем і веб-сервісом. Призначення та переваги HTTPS:

- Захищена передача даних: Основна мета HTTPS - це забезпечити захищену передачу даних через інтернет. Всі дані, які передаються між веб-браузером і веб-сервером, шифруються, що робить їх недоступними для незаконних осіб, які можуть намагатися перехопити чи використовувати ці дані;
- Захист від прослуховування: HTTPS запобігає прослуховуванню (sniffing) на мережевому рівні. Це означає, що навіть якщо злоякісна стороння сторона перехопить трафік між користувачем і сервером, вона не зможе розшифрувати дані, оскільки вони зашифровані;
- Автентичність сервера: HTTPS також допомагає перевірити автентичність сервера. Користувач може бути впевнений, що він з'єднується із справжнім сервером, а не із фальшивим. Це досягається за допомогою сертифікатів SSL/TLS, які виділяються авторитетними організаціями сертифікації;
- Покращує рейтинг в пошукових системах: Багато пошукових систем, такі як Google, враховують використання HTTPS при ранжуванні веб-сайтів у результатах пошуку. Використання HTTPS може покращити видимість вашого веб-сайту у пошукових системах;
- Захищає від атак середнього чоловіка: HTTPS також захищає від атак середнього чоловіка, коли зловмисник видається посередником між користувачем і сервером, а також від атак Man-in-the-Middle (MitM);
- Довірчий зовнішній вигляд: Використання HTTPS може позитивно вплинути на довірчий зовнішній вигляд вашого веб-сайту. Передача конфіденційної інформації через захищене з'єднання демонструє вашу забезпечену ставку на безпеку користувачів;
- Вимоги до даних про користувача: Якщо ваш веб-сайт збирає особисті дані користувачів, використання HTTPS може бути обов'язковим в багатьох юрисдикціях для дотримання законів про захист даних і конфіденційності.

2) Аутентифікація та авторизація - це два важливих методи захисту в інформаційній безпеці, які використовуються для контролю доступу до ресурсів та даних у комп'ютерних системах та мережах. Ці методи використовуються для

перевірки ідентичності користувачів та надання їм прав доступу відповідно до їхніх ролей і прав.

Аутентифікація:

- Призначення: Аутентифікація визначає, чи користувач або сутність (наприклад, пристрій) є тим, за кого вони видають себе. Цей процес допомагає підтвердити ідентичність особи або сутності, яка намагається отримати доступ до системи чи ресурсу;
- Методи аутентифікації: Методи аутентифікації можуть включати в себе паролі, біометричну ідентифікацію (відбитки пальців, розпізнавання обличчя), токени, двофакторну аутентифікацію і т.д;
- Захист від несанкціонованого доступу: Аутентифікація запобігає несанкціонованому доступу до системи, оскільки лише особа або сутність, яка може успішно пройти аутентифікацію, отримає доступ до ресурсів.

Авторизація:

- Призначення: Авторизація визначає, які дії, операції чи ресурси користувач має право виконувати після успішної аутентифікації. Це визначає, що користувач має право читати, записувати, видаляти або виконувати інші дії у системі;
- Керування правами доступу: Авторизація допомагає керувати правами доступу користувачів до конкретних ресурсів або функцій системи. Наприклад, адміністратор має розширені права доступу, в той час як звичайний користувач має обмежений доступ;
- Захист від несанкціонованої діяльності: Авторизація допомагає запобігти несанкціонованій діяльності, оскільки навіть аутентифікований користувач не має права на всі можливі дії в системі.

3) Валідація введених даних: Валідація введених даних - важливий метод захисту в інформаційній безпеці, призначеним для перевірки та забезпечення коректності даних, які вводяться користувачами у програмах, веб-сайтах, додатках та інших системах. Цей метод допомагає попередити введення некоректних або шкідливих даних, що може призвести до різних видів атак та недоліків в безпеці.

Основні призначення валідації введених даних:

- Запобігання вразливостям вводу даних: Валідація даних допомагає запобігти атакам, таким як SQL-ін'єкції, введення скриптів (XSS), введення шкідливих файлів і т. д. Відсіюючи некоректні або потенційно небезпечні дані, ви зменшуєте ризик успішних атак;
- Забезпечення коректності та цілісності даних: Валідація даних гарантує, що введені дані відповідають очікуваному формату та правилам, що може попередити некоректні дії користувачів або помилки в системі;
- Забезпечення безпеки та приватності: Валідація даних допомагає захистити конфіденційні дані, не допускаючи їх некоректного введення або витоку через шкідливі вводи;
- Покращення користувацького досвіду: Правильна валідація даних може попередити введення некоректних даних користувачами та надати зрозумлі повідомлення про помилки, що полегшує їм взаємодію з системою.

Методи валідації даних можуть включати в себе перевірку формату, обмеження на довжину, перевірку на числовість, регулярні вирази, контроль на наявність спеціальних символів, перевірку коректності електронної пошти, чисел і т. д. У веб-середовищі валідація також може використовувати капчі та інші методи, щоб визначити, чи діюча сутність є людиною або ботом.

4) Моніторинг та журналювання (Monitoring and Logging) - це метод захисту та безпеки, який включає в себе процес спостереження за діяльністю в інформаційних системах і мережах, а також запис цієї діяльності у журналах для подальшого аналізу та виявлення вразливостей, атак і аномалій. Цей метод допомагає підтримувати високий рівень безпеки, виявляти та реагувати на потенційні загрози та витоки даних. Основні призначення моніторингу та журналювання [11]:

- Виявлення атак і вразливостей: Моніторинг дозволяє виявляти незвичайну або підозрілу активність в системі, що може свідчити про спроби атак або вразливості. Журнали дозволяють аналізувати цю активність та вживати відповідних заходів;
- Реагування на інциденти безпеки: Моніторинг допомагає оперативно виявляти безпечні інциденти, такі як спроби несанкціонованого доступу або зміни в

системі. Це дозволяє операторам безпеки реагувати швидко та ефективно, щоб запобігти можливим уразам;

- Виявлення витоків даних: Моніторинг може виявляти незвичайну передачу даних з системи або мережі, що може бути ознакою витоку даних. Це допомагає захищати конфіденційні дані користувачів і бізнес-інформацію;
- Підтримка виконання стандартів безпеки: Багато стандартів та регуляцій (наприклад, GDPR, HIPAA) вимагають моніторингу та журналювання діяльності для забезпечення відповідності вимогам безпеки та вивчення інцидентів;
- Аналіз профілю поведінки: Моніторинг та журналювання дозволяють будувати профіль поведінки користувачів і системи, що допомагає виявити аномалії та несправності;
- Покращення безпеки в реальному часі: Моніторинг може забезпечувати інформацію в реальному часі про стан безпеки системи або мережі, що дозволяє операторам негайно реагувати на загрози;
- Збільшення ефективності реагування на інциденти: Аналіз журналів дозволяє розуміти природу і масштаб інцидентів і допомагає вдосконалювати методи реагування на подібні інциденти в майбутньому.

2.4 Розгляд існуючих робочих гіпотез та ідей в галузі кібербезпеки веб-сайтів.

У галузі кібербезпеки веб-сайтів існують численні робочі гіпотези та ідеї, спрямовані на покращення захисту. Деякі з них включають [14]:

1) Мультифакторна аутентифікація (MFA) - це робоча гіпотеза та метод захисту в галузі кібербезпеки веб-сайтів, яка полягає у використанні кількох методів аутентифікації для підтвердження ідентичності користувача перед наданням доступу до ресурсу. MFA є ефективним інструментом для підвищення рівня безпеки та захисту облікових записів користувачів інтернет-ресурсів. Основна ідея MFA полягає в тому, щоб вимагати не тільки щось, що користувач знає (наприклад, пароль), але також щось, що він має або щось, що він є (біометричні дані, такі як відбитки пальців). Зазвичай MFA використовує два або більше з наступних факторів:

- Фактор знань (знання): Це те, що користувач знає, таке як пароль, ПІН-код, відповідь на секретний питання, тощо;
- Фактор володіння (володіння): Це щось, що користувач має, наприклад, мобільний телефон, смарт-карту, USB-ключ або інші фізичні пристрої, які можуть бути використані для генерації одноразових кодів;
- Фактор біометрії (біометрика): Це щось, що є частиною тіла користувача, таке як відбитки пальців, розпізнавання обличчя, сканування радужки тощо;
- Фактор місцезнаходження (місцеперебування): Деякі системи можуть враховувати місцезнаходження користувача, наприклад, на основі його IP-адреси або GPS-координат, як додатковий фактор аутентифікації.

2) Регулярні оновлення та патчі - це одна з основних робочих гіпотез в галузі кібербезпеки веб-сайтів. Цей метод захисту передбачає регулярне оновлення програмного забезпечення та веб-сайту, включаючи операційну систему, веб-сервер, додатки, бази даних та інші компоненти. Патчі - це спеціальні корекції або оновлення, які виробники програмного забезпечення випускають для усунення вразливостей, виявлених в системі. Основні призначення регулярних оновлень і патчів [10]:

- Усунення вразливостей: Веб-сайти та їхні компоненти часто містять вразливості, які можуть бути використані зловмисниками для атак. Патчі виправляють ці вразливості, запобігаючи можливим атакам;
- Підвищення безпеки: Оновлення програмного забезпечення дозволяють вдосконалювати захист системи та міцність проти атак. Зазвичай, виробники також враховують нові методи атак у своїх оновленнях;
- Захист від відомих загроз: Патчі можуть включати у себе заходи проти відомих загроз, таких як віруси, черви, троянські програми та інші види шкідливого програмного забезпечення;
- Забезпечення сумісності: Оновлення також можуть покращувати сумісність програм та системи, що допомагає уникнути конфліктів і збоїв;

- Підтримка та обслуговування: Регулярні оновлення є частиною процесу підтримки і обслуговування веб-сайту. Вони допомагають забезпечити правильне функціонування і безпеку системи;
- Дотримання нормативних вимог: Деякі законодавчі акти та регулятори вимагають від організацій здійснювати регулярне оновлення та патчінг для забезпечення безпеки та конфіденційності даних.

3) Використання штучного інтелекту та машинного навчання. Ці технології можуть бути використані для виявлення підозрілої активності та автоматичного реагування на загрози. Використання штучного інтелекту (ШІ) та машинного навчання (МН) в галузі кібербезпеки веб-сайтів - це сучасна та дуже обіцяна робоча гіпотеза. Вона полягає у використанні автоматизованих систем, що базуються на ШІ і МН, для виявлення, захисту та реагування на загрози в мережі, спрямовані на веб-сайти та інтернет-ресурси. Цей підхід дозволяє покращити безпеку веб-сайтів шляхом виявлення атак та аномалій в реальному часі та вдосконалення методів захисту. Основні способи використання ШІ та МН в кібербезпеці веб-сайтів включають наступне [9]:

- Виявлення атак та аномалій: Системи на основі ШІ та МН можуть аналізувати трафік і діяльність на веб-сайті, виявляючи аномальну або підозрілу активність. Вони можуть виявити спроби фішингу, SQL-ін'єкцій, крос-сайтового скриптингу (XSS) та інші атаки;
- Прогнозування загроз: МН може аналізувати дані щодо нових загроз та атак, створюючи моделі для виявлення подібних атак у майбутньому;
- Системи виявлення вторгнень (IDS) та системи запобігання вторгнень (IPS): ШІ може використовуватися для створення ефективних систем виявлення вторгнень та захисту від них. Вони можуть автоматично реагувати на вторгнення та блокувати атаки;
- Автоматизована обробка вразливостей: ШІ та МН можуть допомагати виявляти та аналізувати вразливості на веб-сайті та вирішувати їх шляхом застосування патчів або інших заходів захисту;

- Аналіз журналів та інцидентів: МН може аналізувати великі обсяги журналів подій та інцидентів для виявлення несправностей, вразливостей та атак;
- Боротьба з ботами: ШІ може виявляти та відсіювати ботів та автоматизовану активність, що допомагає захистити веб-сайт від спаму, скрапінгу та інших видів несанкціонованої активності.

Цей теоретичний огляд надає загальну картину сучасних загроз, методів тестування та захисту веб-сайтів, а також існуючих гіпотез і ідей у галузі кібербезпеки. В подальших розділах дипломної роботи будуть розглядатися ці аспекти більш детально та розроблено комплексний підхід до тестування на безпеку та захисту веб-сайтів.

3 МЕТОДИ ТА ІНСТРУМЕНТИ ДЛЯ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ

В сучасному цифровому світі веб-сайти стали невід’ємною частиною комунікації та інформаційного обміну. Однак разом із зростанням важливості веб-сайтів збільшується і загроза їх безпеці. В умовах постійно зростаючої кількості кіберзлочинів та технічних ускладнень, вирішення завдань, пов'язаних із захистом веб-сайтів та ідентифікацією ризиків, стає вельми актуальним завданням. Для досягнення цієї мети, дослідники та фахівці з кібербезпеки вдосконалюють та застосовують різноманітні методи, засоби та підходи, а саме [16, 17]:

1) Пенетраційні тести (Penetration Testing): Цей метод включає в себе активне тестування веб-сайту, де спеціалісти (етичні хакери) намагаються знайти вразливості та проникнути в систему, щоб оцінити її безпеку. Пенетраційні тести допомагають виявити слабкі місця та потенційні загрози;

2) Сканування вразливостей (Vulnerability Scanning): Цей підхід використовується для автоматизованого сканування веб-сайту з метою виявлення вразливостей, таких як застарілі програмні компоненти, неправильні конфігурації та інші проблеми;

3) Тестування на перевантаження (Load Testing): Цей вид тестування визначає, як веб-сайт витримує навантаження та атаки, включаючи DDoS-атаки. Він допомагає оцінити реакцію веб-сайту на велику кількість запитів і ідентифікувати можливі точки вразливості;

4) Аналіз журналів (Log Analysis): Моніторинг та аналіз журналів діяльності веб-сайту дозволяє виявляти підозрілі або незвичайні події, що можуть бути пов'язані з інцидентами безпеки. Цей метод допомагає вчасно реагувати на потенційні загрози;

5) Аналіз коду (Code Analysis): Вивчення вихідного коду веб-сайту для виявлення потенційних вразливостей, таких як XSS або SQL Injection. Використовуються спеціалізовані інструменти для автоматизованого аналізу коду;

6) Сертифікація безпеки (Security Certification): Можна використовувати стандарти та сертифікації безпеки, такі як ISO 27001, щоб перевірити та підтвердити безпеку веб-сайту;

7) Віртуальні приватні мережі (Virtual Private Networks, VPNs): Використання VPN може допомогти у забезпеченні безпеки комунікацій між користувачами і серверами, особливо при обробці конфіденційної інформації;

8) Постійний моніторинг безпеки (Continuous Security Monitoring): Постійний моніторинг та аналіз безпеки дозволяють виявляти загрози та вразливості в реальному часі і приймати відповідні заходи.

У цьому розділі розглядаються різні методи та засоби, які використовуються для захисту веб-сайтів від існуючих загроз і забезпечення їх безпеки. Веб-сайти, в яких міститься важлива і конфіденційна інформація, повинні бути надійно захищені від потенційних атак і порушень безпеки. Використання правильних методів та засобів грає важливу роль у запобіганні і виявленні можливих загроз.

Описуються ключові методів та засоби захисту веб-сайтів та пояснюється, яким чином досліджується та аналізується їх ефективність. Використання таких методів, як HTTPS, аутентифікація, валідація введених даних та моніторинг та журналювання, є важливими складовими в підвищенні безпеки веб-сайтів [19].

Детальний аналіз цих методів дозволяє оцінити їх ефективність та виявити можливість вдосконалення захисту. Дослідження включає аналіз та моніторинг мережевого трафіку, вивчення історій входу користувачів, валідацію введених даних та визначення результатів подій, пов'язаних із захистом. Ці дії спрямовані на підвищення рівня безпеки веб-сайтів та надання користувачам і власникам впевненості у надійності їхнього ресурсу [23, 26].

Методи та засоби, які використовуються для захисту веб-сайтів від існуючих загроз:

1) HTTPS (Hypertext Transfer Protocol Secure) є криптографічним протоколом, який забезпечує безпеку під час обміну даними між веб-сервером та клієнтом. Використання HTTPS дозволяє шифрувати дані, що передаються між браузером користувача і веб-сайтом, зменшуючи ризик перехоплення інформації. Дослідження та аналіз ефективності: Ефективність HTTPS може бути оцінена

шляхом моніторингу та аналізу мережевого трафіку. За допомогою спеціальних інструментів, таких як проксі-сервери та аналізатори трафіку, можна вивчити, які дані пересилаються в незашифрованому вигляді і яким чином вони зашифровані за допомогою HTTPS. Також можна аналізувати доступні статистичні дані щодо кількості атак на веб-сайт, які були відвернуті завдяки використанню HTTPS [18];

2) Аутентифікація полягає у перевірці ідентифікації користувача або системи перед наданням доступу до ресурсів веб-сайту. Це може включати в себе використання паролів, біометричних даних, одноразових кодів тощо. Дослідження та аналіз ефективності: Ефективність аутентифікації може бути оцінена через аналіз історій входу користувачів, виявлення спроб несанкціонованого доступу і визначення того, які методи аутентифікації забезпечують найвищий рівень безпеки;

3) Валідація даних передбачає перевірку введених користувачем даних на предмет відповідності встановленим правилам і обмеженням. Це допомагає уникнути вразливостей, таких як SQL ін'єкції або внесення шкідливого коду. Дослідження та аналіз ефективності: Для аналізу ефективності валідації даних можна вивчати журнали системи на предмет спроб введення шкідливих даних та виявлення успішної блокування таких спроб. Також важливо аналізувати статистику вразливостей, пов'язаних з некоректною валідацією;

4) Моніторинг і журналювання дозволяють виявляти незвичайну або підозрілу активність на веб-сайті. Це може включати в себе аналіз журналів подій, моніторинг мережевого трафіку та використання систем сповіщення про інциденти. Дослідження та аналіз ефективності: Для оцінки ефективності моніторингу та журналювання можна досліджувати кількість виявлених загроз, час реакції на них та результати розслідування інцидентів. Дослідження та аналіз ефективності цих методів захисту може включати в себе внутрішні аудити, зовнішні тести та аналіз статистичних даних. Результати цього аналізу допомагають визначити, наскільки ефективно веб-сайт забезпечує безпеку і які можливі поліпшення слід внести для підвищення рівня безпеки.

Дослідження та аналіз ефективності цих методів захисту може включати в себе внутрішні аудити, зовнішні тести та аналіз статистичних даних. Результати

цього аналізу допомагають визначити, наскільки ефективно веб-сайт забезпечує безпеку і які можливі поліпшення слід внести для підвищення рівня безпеки [16].

Інструменти для тестування на безпеку допомагають ідентифікувати потенційні вразливості та загрози, які можуть бути використані зловмисниками для атак на веб-сайти. Основною метою тестування на безпеку є виявлення слабких місць у веб-додатках та серверному програмному забезпеченні, щоб їх можна було вчасно виправити і захистити веб-сайт від потенційних атак. Приклади інструментів які використовуються для тестування на безпеку веб-сайту [18]:

1) Сканери вразливостей:

- Nessus: Nessus є одним з найвідоміших комерційних сканерів вразливостей. Він автоматично сканує веб-сайти на наявність різних типів вразливостей, таких як вразливості на рівні ОС, мережеві проблеми та вразливості додатків;
- OpenVAS: Це безкоштовне аналогічне програмне забезпечення, яке дозволяє сканувати веб-сайти для виявлення вразливостей. OpenVAS має велику базу даних CVE (Common Vulnerabilities and Exposures), що робить його ефективним інструментом для виявлення вразливостей;
- OWASP ZAP (Open Web Application Security Project Zed Attack Proxy) - це відкрите програмне забезпечення для тестування безпеки веб-додатків. Розроблено в рамках проекту OWASP, цей інструмент дозволяє розробникам та тестувальникам виявляти та виправляти потенційні безпекові вразливості в їхніх веб-додатках [24].

2) Пейлоад-генератори та інтрудери:

- Burp Suite: Burp Suite - це інструмент для тестування на безпеку веб-додатків, який включає в себе пейлоад-генератор та інтрудер. Він дозволяє тестувати на вразливості, такі як ін'єкції SQL, Cross-Site Scripting (XSS) і багато інших. Burp Suite також має інші корисні функції, такі як Proxy і Repeater для ручного аналізу запитів і відповідей.

3) Аналізатори безпеки коду:

- Veracode: Цей інструмент спеціалізується на аналізі вихідного коду веб-додатків для виявлення вразливостей. Він надає детальні звіти про знайдені проблеми та навіть пропонує рекомендації щодо їх виправлення;
- Fortify: Інший інструмент для аналізу безпеки вихідного коду, який допомагає виявити вразливості на рівні програмного коду і надає інформацію про їх виправлення.

4) Сканери слабкостей аутентифікації та авторизації:

- OWASP Amass: Цей інструмент спеціалізується на тестуванні на слабкості аутентифікації та авторизації в веб-додатках. Він допомагає виявити проблеми, такі як слабкі паролі та некоректні налаштування авторизації;
- OAuthTester: OAuthTester призначений для тестування процесу OAuth-авторизації і виявлення можливих слабкостей в ньому.

5) Засоби для тестування на вразливості веб-серверів та конфігурації:

- Nmap: Nmap є популярним інструментом для сканування портів та виявлення служб на веб-серверах. Він допомагає визначити можливі шляхи атаки на сервер;
- SSL Labs: Цей веб-сервіс дозволяє оцінити безпеку SSL/TLS конфігурації сервера, виявити слабкості та рекомендації щодо поліпшення безпеки.

6) Фішинг-тести:

- Gophish: Gophish - це інструмент для створення та запуску атак фішингу. Він дозволяє вам тестувати стійкість співробітників до соціально-інженерних атак, відслідковувати їх відгуки та навчати персонал реагувати на такі загрози.

7) Web Application Firewalls (WAF) та IDS/IPS:

- ModSecurity: Це безкоштовний WAF, який застосовується на рівні веб-сервера та допомагає виявити та блокувати атаки на веб-додатки, такі як SQL-ін'єкції та Cross-Site Scripting;
- Snort: Snort - це інтрузіонна система виявлення і запобігання атакам (IDS/IPS), яка може бути використана для виявлення атак на веб-сайт, включаючи атаки на рівні додатків.

4 ТЕСТУВАННЯ НА БЕЗПЕКУ ВЕБ-САЙТУ

Тестування на безпеку сайту це важлива частина розробки та експлуатації будь-якого веб-проекту, яке вимагає постійної уваги та професійного підходу. Існує безліч причин, чому це важливо робити, і я хотіла би розглянути основні з них. безпека сайту є критично важливою для захисту конфіденційної інформації користувачів та даних клієнтів. У світі, де кіберзлочинність набуває обертів, непрофесійне ведення веб-проекту може призвести до витоку особистої інформації, фінансових даних або інших конфіденційних даних. Тестування на безпеку допомагає виявити потенційні слабкі місця в системі та зміцнити їх, запобігаючи можливим атакам та витокам даних.

Забезпечення безпеки сайту сприяє підвищенню довіри користувачів. Якщо вони переконані, що їхні дані та особистість належним чином захищені, вони будуть більш схильні взаємодіяти з вашим сайтом, робити покупки та надавати особисту інформацію. Навпаки, вразливості сайту можуть призвести до втрати довіри і втрати клієнтів. Безпека сайту є ключовою складовою витрат на попередження ризиків. Порушення безпеки може призвести до серйозних фінансових втрат, втрати репутації та юридичних проблем. Тестування на безпеку дозволяє вчасно виявити потенційні загрози та усунути їх, що може заощадити час, гроші та нерви в майбутньому.

Узагальнюючи, тестування на безпеку сайту є обов'язковим кроком в процесі розробки та експлуатації будь-якого веб-проекту. Воно сприяє захисту даних, збільшенню довіри користувачів, зменшенню ризиків та дотриманню законодавства. Безпека повинна бути на першому плані для всіх, хто займається створенням та обслуговуванням веб-сайтів.

Тестування на безпеку веб-сайту є важливою складовою усього процесу розробки та експлуатації веб-продуктів. В сучасному цифровому світі, де веб-сайти є не тільки інформаційними ресурсами, але і платформами для електронної торгівлі, обміну особистою інформацією та важливих операцій, забезпечення їхньої безпеки стає найвищим пріоритетом.

У даному дослідженні буде використано такі методи та інструменти, такі як:

- Пенетраційні тести (Penetration Testing) веб-сайту bWAPP;
- Сканування вразливостей (Vulnerability Scanning) веб-сайту bWAPP;
- OWASP ZAP (Open Web Application Security Project Zed Attack Proxy).

4.1 Демонстрація вразливостей та методи запобігання

Після відкриття XAMPP Control Panel та виконання першого етапу - запуску модулів Apache та MySQL (Рис.4.1), автоматично відкриється локальний хост bWAPP. Наступним кроком є проведення процедури авторизації, під час якої користувач повинен ввести логін і пароль, відповідно: логін "bee", пароль "bug" (Рис.4.2).

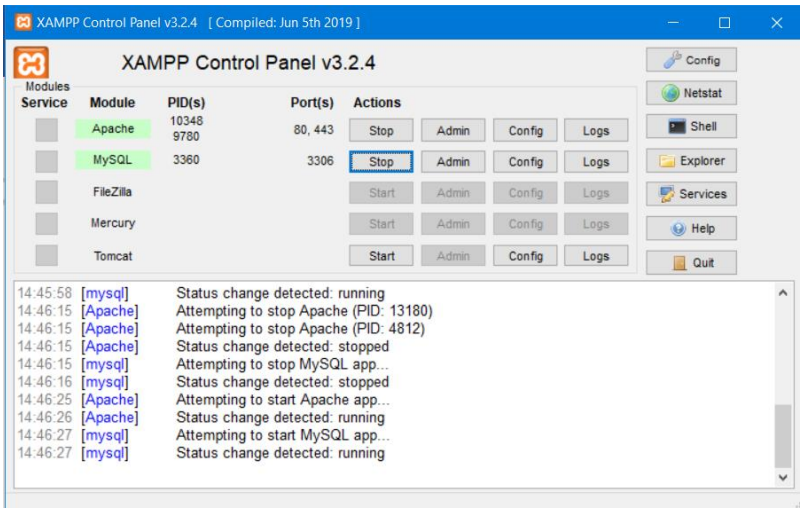

Рисунок 4.1 – Процес запуску додатку XAMPP Control Panel

Рисунок 4.2 – Процес авторизації на сайті bWAPP

1) Broken Access Control - Restrict Folder Access

Вразливість Broken Access Control, конкретно в контексті "Restrict Folder Access", демонструє проблеми з неправильним контролем доступу до папок або ресурсів веб-додатків. За допомогою bWAPP для демонстрації цієї вразливості, виконуються наступні кроки:

- Обираємо вразливість Broken Access Control - Restrict Folder Access (Рис. 4.3)

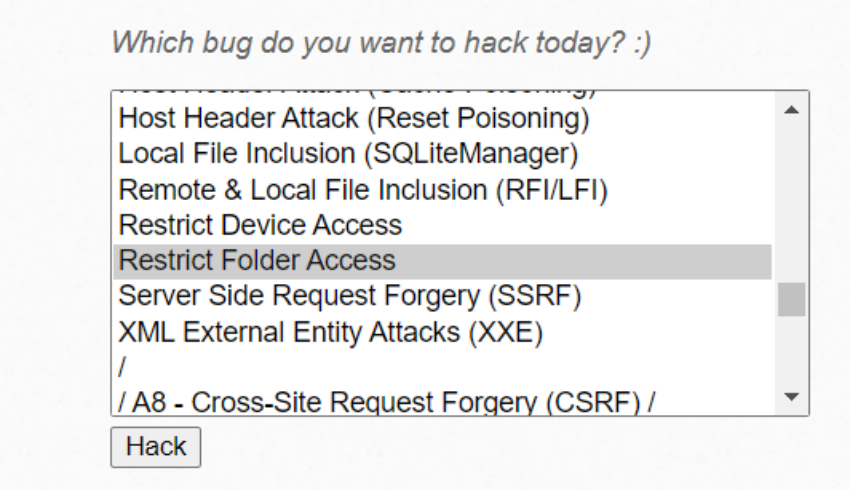

Рисунок 4.3 - Restrict Folder Access

Тільки авторизовані користувачі мають доступ до папки “documents”.

На Рис.4.4 зображено список файлів, що доступні для прочитання авторизованим користувачам.

Рисунок 4.4 – Список доступних файлів

- Здійснюємо вихід з облікового запису, попередньо скопіювавши посилання на директорію /documents;

- Після цього вставляємо посилання у рядок пошуку та переглядаємо отриманий результат, представлений на Рис.4.5. Зазначений результат включає в себе список усіх файлів, які мають обмежений доступ і призначені лише для авторизованих користувачів. Важливо звернути увагу, що, оскільки процедура авторизації не була виконана, то було здобуто доступ до цієї директорії без належних прав. Ця ситуація є ілюстрацією вразливості, відомої як "Broken Access Control - Restrict Folder Access".

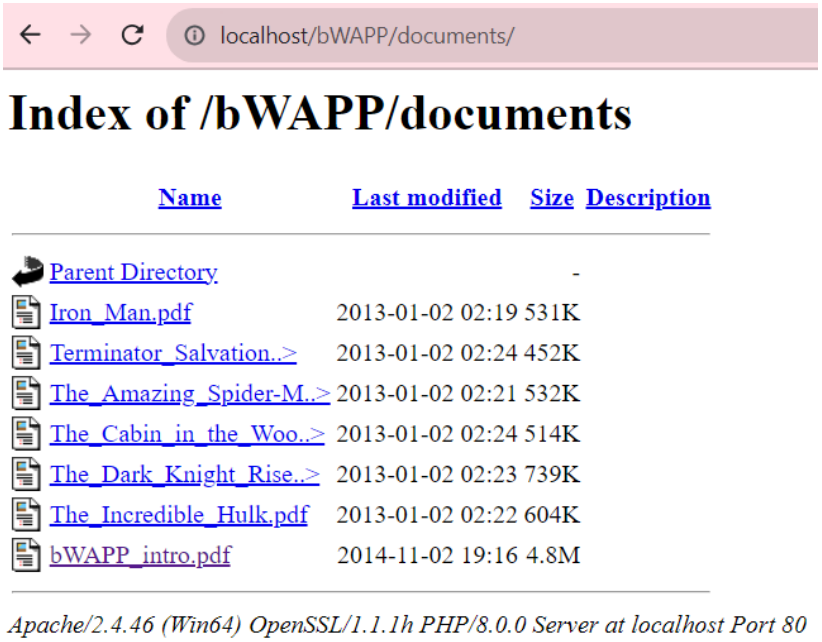

Рисунок 4.5 - Список усіх файлів, які мають обмежений доступ

Ця вразливість виникає, коли механізми обмеження доступу не належним чином налаштовані, дозволяючи несанкціонованим користувачам отримувати доступ до конфіденційної інформації або ресурсів, які мають бути захищені. У даному випадку, недостатня або неправильна авторизація дозволяє отримати доступ до файлів, які повинні бути доступні лише за відповідних умов.

2) Broken Authentication and Session Management - Insecure Login Forms

- Обираємо вразливість Broken Auth – Insecure Login Forms;
- Вводимо в строки логіну та паролю: 0'or'0'='0 (Рис. 4.6);

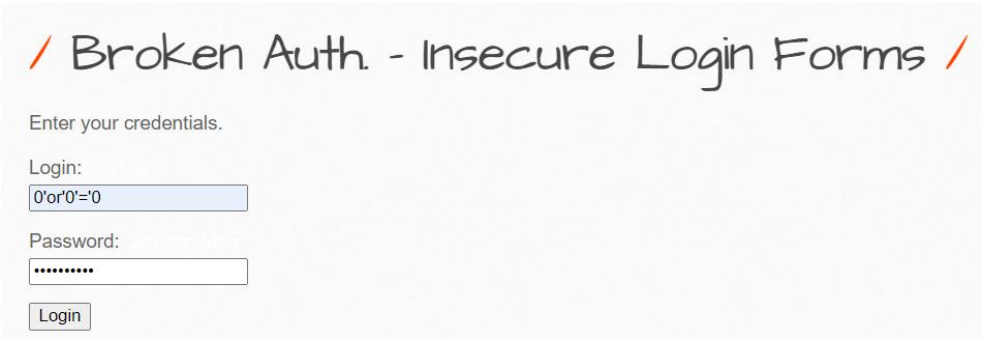

Рисунок 4.6 – Введення 0'or'0'='0

- Після введення інформації 0'or'0'='0, сайт показує, що дані не вірні (Рис.4.7);

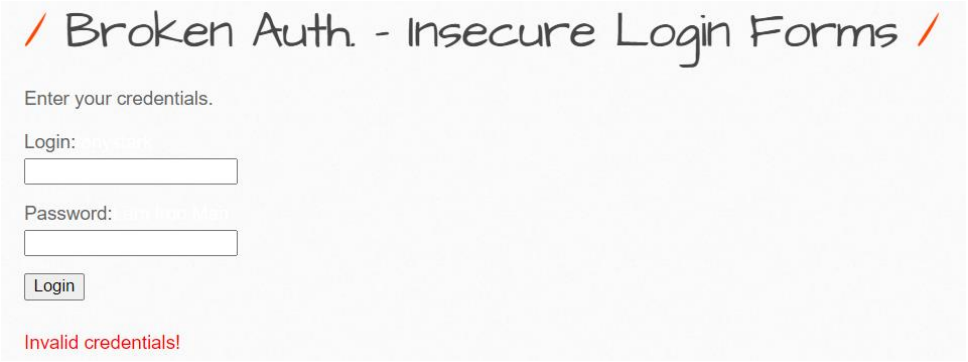

Рисунок 4.7 – Дані не вірні

- Наступним етапом є аналіз коду сторінки для отримання додаткової інформації. На Рис. 4.8 представлено фрагмент коду, при ретельному розгляді коду виявлено, що авторизаційні дані відкрито відображаються у вихідному коді сторінки. Зокрема, логін "tonystark" та пароль "I am Iron Man" можна безпосередньо визначити, навіть без необхідності робити запитання до бази даних або інших шляхів;

```
<div id="main">

    <h1>Broken Auth. - Insecure Login Forms</h1>

    <p>Enter your credentials.</p>

    <form action="/bWAPP/ba_insecure_login_1.php" method="POST">

        <p><label for="login">Login:</label><font color="white">tonystark</font><br />
        <input type="text" id="login" name="login" size="20" /></p>

        <p><label for="password">Password:</label><font color="white">I am Iron Man</font><br />
        <input type="password" id="password" name="password" size="20" /></p>

        <button type="submit" name="form" value="submit">Login</button>

    </form>

    </br >
    <font color="red">Invalid credentials!</font>
</div>
```

Рисунок 4.8 – Фрагмент коду авторизаційних даних

- Під час введення відомих логіна та пароля у форму авторизації, система підтверджує правильність цих даних (Рис. 4.9). Це свідчить про те, що веб-сайт виявляється вразливим до атак на автентифікацію та управління сесією.

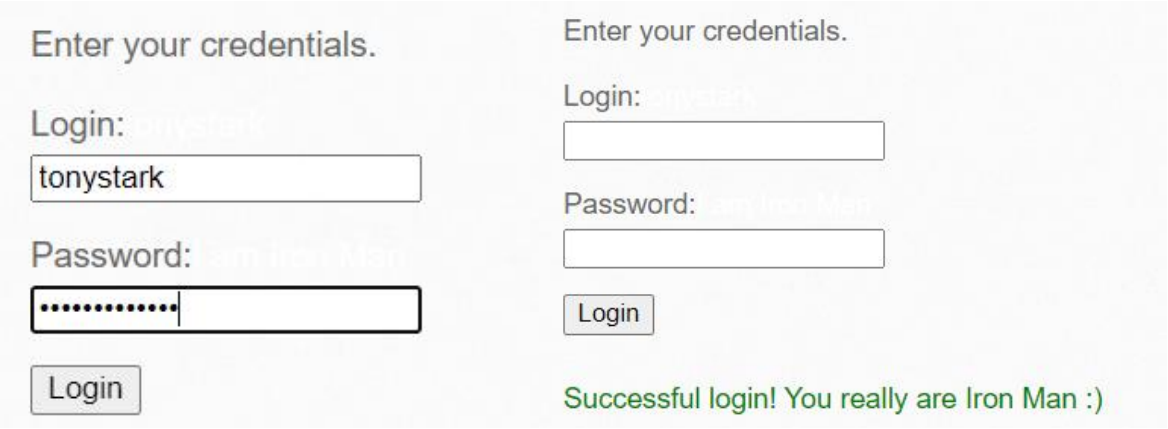

Рисунок 4.9 – Успішна авторизація на сайті

Отже, враховуючи виявлену слабкість, можна прийти до висновку, що механізми Broken Authentication and Session Management, зокрема "Insecure Login Forms", на данний момент не відповідають найвищим стандартам безпеки. Цей недолік може призвести до серйозних наслідків, таких як несанкціонований доступ до облікових записів інших користувачів і порушення конфіденційності інформації. Надзвичайно важливо акцентувати увагу на вирішенні цієї вразливості для забезпечення надійної системи автентифікації та управління сесією.

3) Logout Management. Вразливість "Logout Management" стосується недоліків у процесі виходу з системи, коли користувач виходить з облікового запису, але його сесійні дані не оброблюються належним чином. Для демонстрації цієї вразливості за допомогою bWAPP виконуються наступні кроки:

- Виходимо з облікового запису (Рис.4.10). Коли користувач виходить з облікового запису за допомогою функції логаут на веб-сайті, очікується, що його сесійні дані будуть належним чином знищені або неактивні;

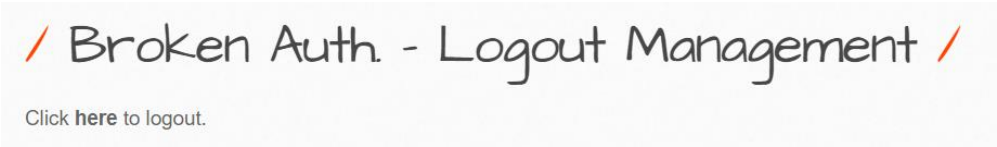

Рисунок 4.10 – Вихід з облікового запису

Вразливість виникає, коли система не видаляє або не анулює сесійні дані користувача під час виходу. Це може призвести до того, що деякі інформаційні фрагменти сесії залишаться активними.

Якщо сесійні дані залишаються активними після виходу, це відкриває можливість для атак, таких як сесійне підмінювання або перехоплення сесій. Злоумисники можуть використовувати ці залишкові сесійні дані для несанкціонованого доступу до облікового запису користувача.

Вразливість "Logout Management" може вести до доступу до конфіденційної інформації користувача, такої як особисті дані, фінансові відомості або інші чутливі дані, навіть після фактичного виходу з системи.

Для запобігання цій вразливості необхідно правильно налаштовувати процес виходу, включаючи належне знищення сесійних даних. Важливо враховувати безпеку сесій та використовувати заходи, такі як генерація нового сесійного ідентифікатора після виходу.

4) Administrative Portals. Вразливість "Administrative Portals" вказує на проблеми в управлінні адміністративними порталами веб-додатків. Для демонстрації цієї вразливості за допомогою bWAPP, виконуємо наступні кроки:

- Обираємо вразливість "Administrative Portals" на платформі bWAPP. Після переходу на адміністративний портал відображається повідомлення про блокування сторінки, що свідчить про відсутність доступу для звичайного користувача (Рис. 4.11);

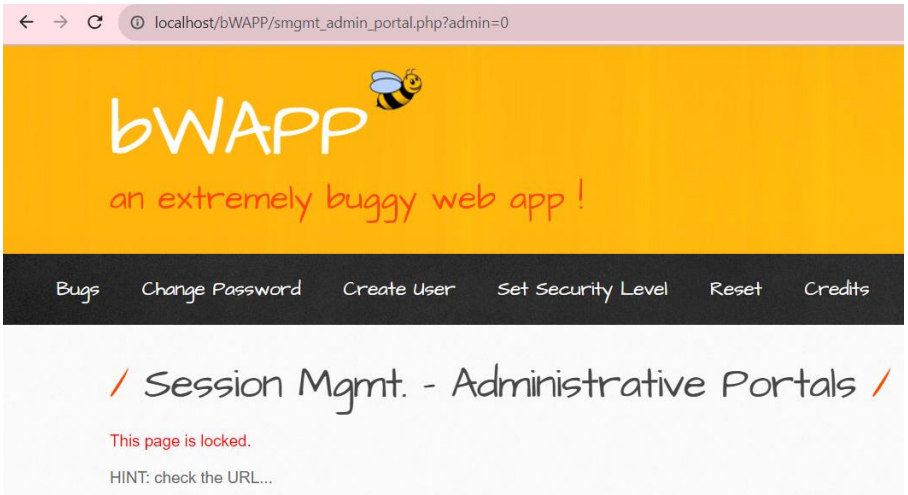

Рисунок 4.11 - Перехід на адміністративний портал та блокування сторінки

- Проводячи подальші дослідження, виявляється, що за допомогою деяких маніпуляцій у пошуковій строці можна отримати доступ до заблокованого адміністративного порталу. Зокрема, замінивши значення 0 на 1, відбувається розблокування цієї сторінки (Рис. 4.12);

Рисунок 4.12 - Заміна значення 0 на 1 та розблокування сторінки

- Цей процес вказує на вразливість у механізмі управління адміністративними порталами, де використання неправильної аутентифікації чи несанкціонованого доступу дозволяє обійти заходи безпеки.

Демонстрація вразливості "Administrative Portals" на прикладі bWAPP розкриває потенційну небезпеку, пов'язану з несанкціонованим доступом до ключових адміністративних ресурсів. У разі успішної атаки, зловмисник може отримати необмежений доступ до чутливої інформації, яка знаходиться під управлінням адміністратора, тим самим поставляючи під загрозу конфіденційність та цілісність цих даних. Ця вразливість може викликати серйозні наслідки, такі як втрата конфіденційної інформації, можливість здійснення несанкціонованих змін або втручання в роботу системи. Зокрема, зловмисники можуть отримати доступ до адміністративних функцій, які можуть включати у себе управління користувачами, налаштування безпеки та інші важливі аспекти системи.

Для вирішення цієї вразливості критично важливо вжити ефективних заходів безпеки, таких як поліпшення механізмів аутентифікації, встановлення строгих прав доступу та вдосконалення адміністративних порталів. Запровадження кращих практик управління адміністративним доступом може суттєво зменшити ризик несанкціонованого доступу та підвищити загальний рівень безпеки веб-додатка.

5) HTML Injection - Reflected (GET)

- Вводимо в текстове поле значення, що містить HTML-код: <h1>you are hacked</h1> (Рис. 4.13);

Рисунок 4.13 - Введення в текстове поле HTML коду

- Спостерігаємо за результатом: оскільки введений HTML-код відображається на сторінці так, як його було введено, то вразливість HTML Injection - Reflected (GET) є присутньою (Рис. 4.14).

Рисунок 4.14 – Результат наявності вразливості

6) HTML Injection - Reflected (POST). HTML Injection (Reflected) вказує на можливість внедрення HTML-коду у вихідний код сторінки через параметри, що передаються у HTTP-запиті. Для демонстрації цієї вразливості за допомогою bWAPP виконуємо наступні кроки:

- Введення в текстове поле значення, що включає HTML-код, дозволяє внести зміни у вихідний код сторінки.

<h2>you can inject anything here</h2>

lol та отримуємо результат (Рис 4.15);

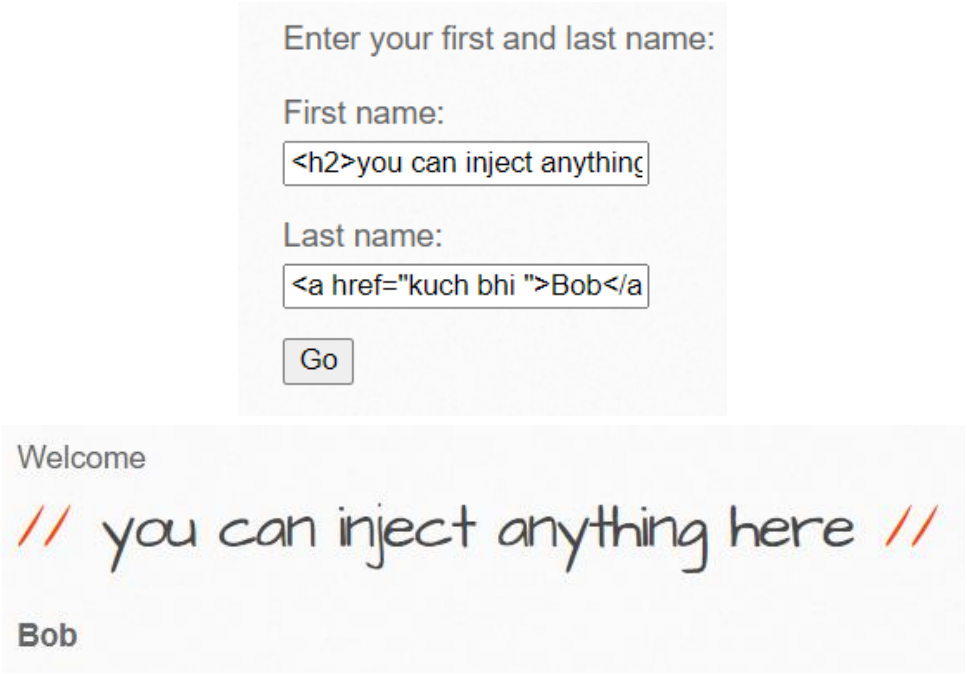

Рисунок 4.15 - Введення в текстове поле HTML коду та результат

- При наведенні на слово "Bob" у виведеному тексті, користувач може помітити появу посилання, а саме: http://localhost/bWAPP/kuch%20bhi. Однак важливо відзначити, що це посилання веде на сторінку "Page Not Found" (сторінка, що не знайдена) (Рис.4.16).

Рисунок 4.16 - Сторінка "Page Not Found"

Ця ситуація є наслідком вразливості HTML Injection - Reflected (POST), де введений HTML-код впливає на генерацію вихідної сторінки. В даному випадку, несанкціоноване внесення посилання спричинює невірний перехід на недоступну сторінку, що може бути використано для створення плутанини або для впровадження соціально-інженерних атак на користувачів.

7) XSS - Reflected (GET)

- Шляхом маніпуляції URL-параметрами, вводимо імена firstname та lastname як "Bob Marley". Як результат, веб-сайт вітає користувача словами "Welcome Bob Marley" (Рис. 4.17);

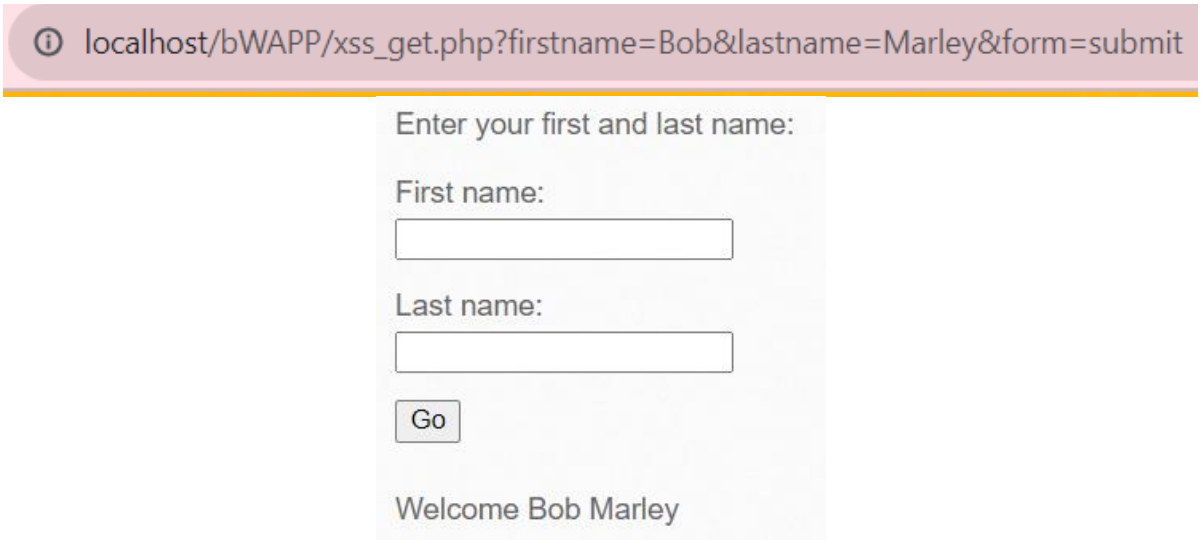

Рисунок 4.17 - Маніпуляції URL-параметрами та "Welcome Bob Marley"

- Проте, експлуатуючи вразливість XSS - Reflected (GET), вставляємо HTML-код у текстові поля, а саме:

<script>alert('This website is XSS vulnerable')

</script> (Рис.4.18);

Рисунок 4.18 – Вставка HTML-коду у текстові поля

- Після введення цього коду, веб-сайт відображає вікно підтвердження, яке свідчить про наявність вразливості (Рис.4.19). Наступним кроком є використання іншого HTML-коду для витягування інформації про куки користувача, а саме: <script>alert(document.cookie)</script>;

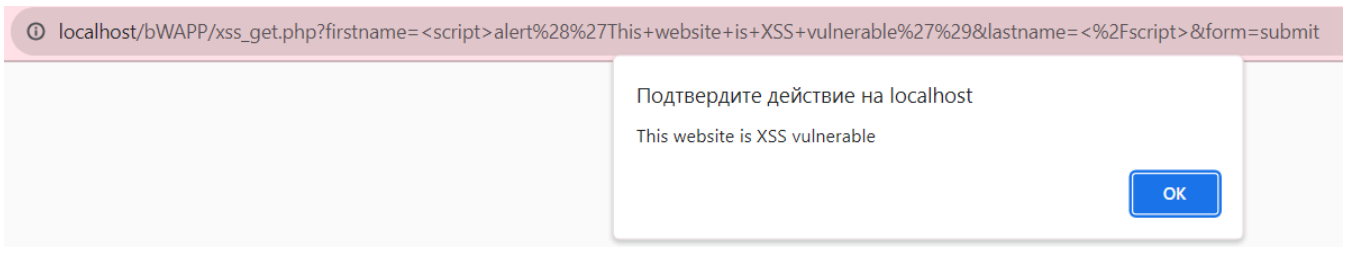

Рисунок 4.19 - Вікно підтвердження, де наявна вразливість

- Зловмисник використовує введений HTML-код для виведення повідомлення alert(document.cookie) на сторінці. Цей скрипт викликає вікно інформаційного сповіщення, яке містить дані про кукі користувача (Рис.4.20).

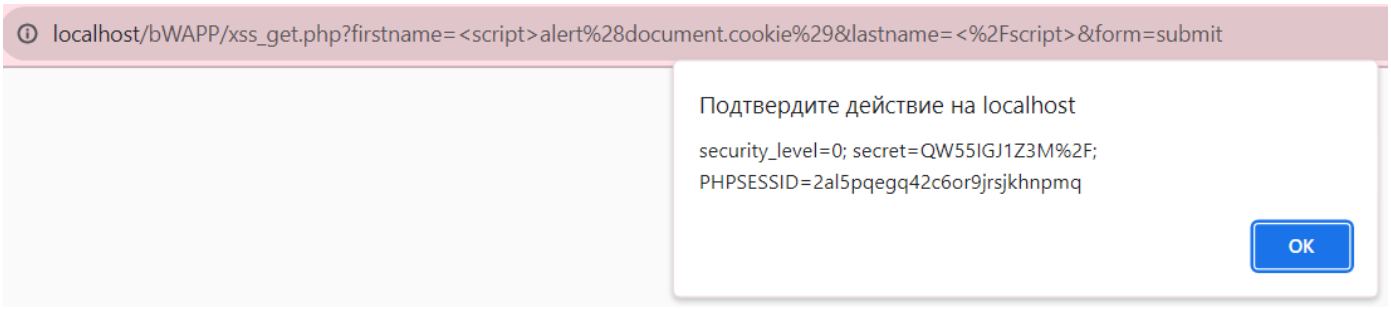

Рисунок 4.20 - Вікно підтвердження, що містить дані про кукі користувача

- У фразі alert(document.cookie), document.cookie в JavaScript використовується для доступу до інформації про куки, яка зберігається на браузері користувача. У виведеному повідомленні ми бачимо конкретні дані про куки, такі як "security_level=0;secret=QW55IGJ1Z3M%2F;PHPSESSID=2al5pqegq42c6or9jrsjkhnpmq".

Отримання конфіденційних даних, таких як дані про куки, через використання XSS, наголошує на наявність серйозної вразливості в безпеці додатка. Зазначені дані містять інформацію, яка може бути використана для несанкціонованого доступу та атак на безпеку.

Для усунення вразливості XSS - Reflected (GET) важливо вжити заходів безпеки, таких як коректна фільтрація та екранування введених даних перед їх виведенням на сторінці. Впровадження сучасних методів захисту, таких як Content Security Policy (CSP), може значно зменшити ризик подібних атак і підвищити рівень безпеки додатка.

4.2 Активне сканування веб-сайту за допомогою OWASP ZAP

У сучасному цифровому світі безпека веб-додатків є ключовим аспектом для забезпечення захисту від різноманітних кіберзагроз. З метою виявлення та усунення потенційних вразливостей веб-сайтів широко використовуються різноманітні

інструменти для активного сканування. Один із таких ефективних інструментів - OWASP Zed Attack Proxy (ZAP). Це відкрите програмне забезпечення, яке надає можливості автоматичного сканування веб-додатків для виявлення потенційних вразливостей, таких як кросс-сайтовий скриптінг (XSS), SQL-ін’єкції та інші атаки.

У цьому пункті буде розглянуто процес активного сканування веб-сайту з використанням OWASP ZAP та етапи автоматичного сканування. Буде проаналізовано звіт, створений ZAP, для ідентифікації можливих вразливостей в веб-додатку. Крім того, буде надано загальні рекомендації та вказівки щодо усунення виявлених вразливостей, забезпечуючи кращий рівень безпеки для веб-додатку.

Розглядаючи цей процес, буде отримана інформація, яка дозволить забезпечити високий рівень безпеки веб-додатку та допоможе зберегти дані користувачів від потенційних атак (Рис. 4.21, Рис. 4.22).

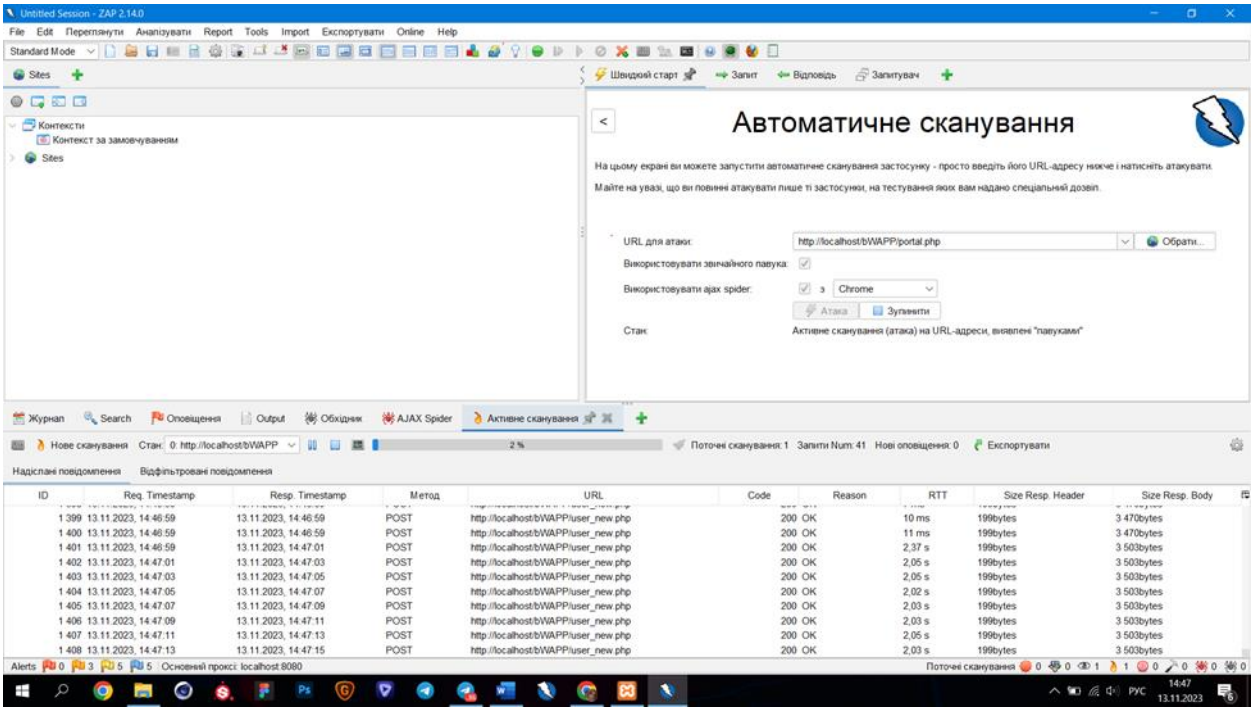

Рисунок 4.21 – Процес автоматичного сканування веб-додатка

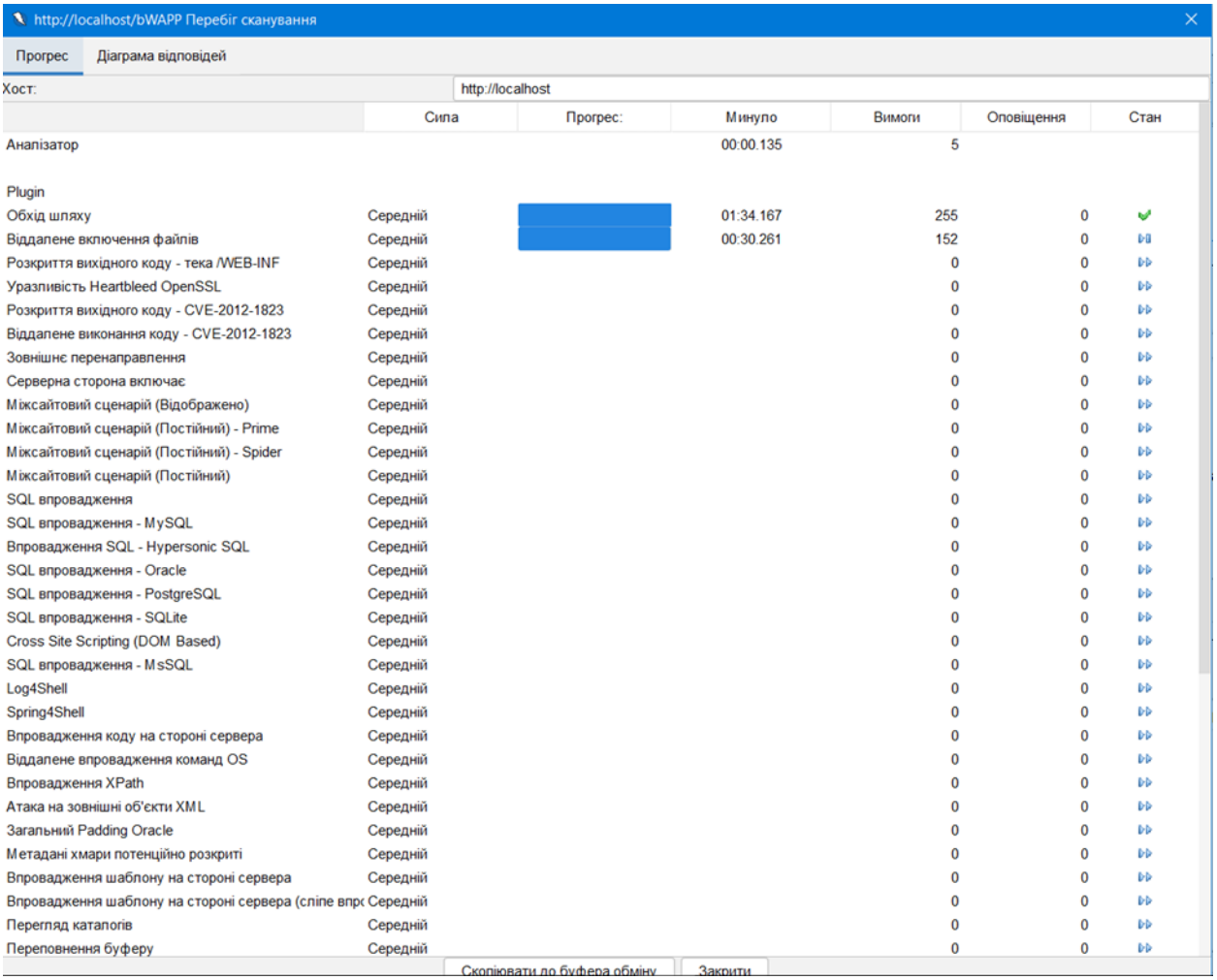

Рисунок 4.22 – Перевірка наявності вразливостей веб-додатка

Даний фрагмент звіту ZAP (OWASP Zed Attack Proxy) (Рис. 4.23) надає інформацію про конфігурацію та параметри відсканування веб-сайтів за допомогою цього інструменту. Розглянемо його складові елементи:

1) Report Parameters (Параметри звіту):

- Вказує на те, що жоден конкретний контекст не був обраний, тому всі контексти були автоматично включені за замовчуванням;
- Також, жоден конкретний веб-сайт не був обраний, тому всі сайти були включені за замовчуванням.

2) Risk Levels (Рівні ризику):

- Включені рівні ризику: Високий, Середній, Низький, Відомості;
- Жоден рівень ризику не був виключений.

3) Confidence Levels (Рівні впевненості):

- Включені рівні впевненості: Користувача підтверджено, Високий, Середній, Низький;

- Виключені рівні впевненості: Користувача підтверджено, Високий, Середній, Низький, Помилкове спрацьовування.

Ці параметри важливі для того, щоб зрозуміти, які типи ризиків та рівні впевненості включені або виключені під час використання ZAP для сканування веб-сайтів. Наприклад, зазначення відсутності виключених рівнів ризику може свідчити про те, що всі рівні ризику були включені для аналізу.

Рисунок 4.23 – Фрагмент звіту ZAP

Аналіз за рівнями ризику та впевненості (Рис. 4.24):

- Дана таблиця показує кількість зауважень для кожного рівня ризику та впевненості, які включені в звіт;
- На приклад, для рівня ризику "Середній" та впевненості "Середній" є 2 зауваження, що становлять 11,1% від загальної кількості включених зауважень;
- Загальна кількість зауважень включає 18 елементів;

- Найбільше зауважень припадає на рівень ризику "Середній" (44,4%), після чого йдуть "Низький" (27,8%) та "Відомості" (27,8%);
- Ризик "Високий" не має жодного зауваження в даному контексті;
- Показано, скільки зауважень припадає на кожен сайт для різних рівнів ризику;
- Наприклад, для сайту "http://localhost" немає зауважень на рівні ризику "Високий", 6 зауважень на рівні "Середній", 5 на рівні "Низький" та 7 на рівні "Відомості";
- Зазначено, що зауваження з рівнем впевненості "False Positive" виключено з обчислень.

		впевненість				
		Користувача підтверджено	Високий	Середній	Низький	Всього
Ризик	Високий	0 (0,0%)	0 (0,0%)	0 (0,0%)	0 (0,0%)	0 (0,0%)
	Середній	0 (0,0%)	2 (11,1%)	2 (11,1%)	2 (11,1%)	6 (33,3%)
	Низький	0 (0,0%)	1 (5,6%)	4 (22,2%)	0 (0,0%)	5 (27,8%)
	Відомості	0 (0,0%)	2 (11,1%)	2 (11,1%)	3 (16,7%)	7 (38,9%)
	Всього	0 (0,0%)	5 (27,8%)	8 (44,4%)	5 (27,8%)	18 (100%)

		Ризик			
		Високий (= Високий)	Середній (>= Середній)	Низький (>= Низький)	Відомості (>= Відомості)
Сайт	http://localhost	0 (0)	6 (6)	5 (11)	7 (18)

Рисунок 4.24 - Кількість зауважень для кожного рівня ризику та впевненості

Загальна статистика:

- Загальна кількість зауважень: 18;

- Розподіл за рівнями ризику: Середній (38,9%), Низький (27,8%), Відомості (33,3%).

Загальна картина безпеки веб-сайту "http://localhost", яка була проаналізована за допомогою інструменту тестування на безпеку ZAP, вказує на наявність різноманітних потенційних вразливостей та ризиків. Загалом, знайдено 18 зауважень, які розділені за типами атак та рівнями ризику:

1) Середній ризик:

- Виявлено різні аспекти безпеки, такі як відсутність заголовків для захисту від клікджекінгу, відсутність токенів Anti-CSRF, відсутність політики безпеки вмісту (CSP), перегляд каталогів та інші.

2) Низький ризик:

- Знайдено проблеми, такі як відсутність заголовка X-Content-Type-Options, відсутність атрибута HttpOnly у файлах cookie, а також інформаційний витік через HTTP заголовки.

3) Відомості:

- Інформаційні зауваження, такі як результати фузз-тестування агента-користувача, запити на автентифікацію, виявлення відповідей до керування сесією та інші дії, що не є критичними, але можуть мати інтерес для аналізу.

З цього звіту випливає, що важливо вжити ефективних заходів для захисту веб-сайту від потенційних загроз та вразливостей. Рекомендації для виправлення повинні бути спрямовані на встановлення необхідних заголовків безпеки, запобігання атакам типу Cross-Site Scripting (XSS), а також підвищення рівня захисту автентифікаційних механізмів та сесій. Такий комплексний підхід дозволить зменшити ризики та забезпечити стійкість веб-сайту до потенційних кібератак.

Заходи для виправлення та підвищення безпеки веб-сайту "http://localhost":

1) Середній ризик:

- Встановлення необхідних заголовків для захисту від клікджекінгу для усунення можливості атак даного типу;

- Запровадження токенів Anti-CSRF для захисту від атак, пов'язаних з фальсифікацією міжсайтових запитів;

- Встановлення політики безпеки вмісту (CSP) для обмеження можливостей використання XSS-атак та інших атак на сторінках веб-сайту;
- Вирішення проблеми перегляду каталогів для уникнення витоку чутливої інформації.

2) Низький ризик:

- Додавання заголовка X-Content-Type-Options для запобігання атакам на основі визначення типу контенту;
- Встановлення атрибута HttpOnly у файлах cookie для ускладнення доступу до них через скрипти на клієнтській стороні;
- Активна моніторинг і виправлення можливих інформаційних витоків через HTTP заголовки.

3) Відомості:

- Оцінка результатів фузз-тестування агента-користувача та вжиття заходів для зменшення можливості несанкціонованого доступу чи витоку інформації;
- Посилення захисту автентифікаційних механізмів та сесій для уникнення ризиків, пов'язаних з авторизацією та керуванням сесіями.

Загальний комплексний підхід до захисту включатиме не лише технічні заходи, але і організаційні, такі як проведення навчання персоналу з питань безпеки, визначення політик безпеки, та регулярні аудити безпеки для виявлення та усунення нових потенційних вразливостей.

ВИСНОВКИ

У ході виконання дипломної магістерської роботи було здійснено глибокий аналіз тестування на безпеку веб-сайтів, ідентифікації ризиків та застосування ефективних методів захисту. Результати цього дослідження не лише підтверджують актуальність проблем безпеки в інтернет-просторі, але і надають підґрунтя для створення комплексних та ефективних заходів захисту веб-сайтів. Перш за все, активне сканування на безпеку веб-сайту розкрило ряд серйозних вразливостей, серед яких особливо важливою є недостатня аутентифікація та слабкі місця в системах контролю доступу. На вартість безпеки веб-платформ впливають також недоліки в програмному забезпеченні, які вимагають негайного усунення. Виявлення цих вразливостей стало основою для подальшого вдосконалення стратегій та заходів безпеки.

Було детально розглянуто найпоширеніші вразливості, виявлені під час сканування, такі як, Cross-Site Scripting (XSS), а також вразливості, пов'язані з некоректною обробкою сесій та витоком конфіденційної інформації. Це надає повний обзор та розуміння потенційних небезпек для безпеки веб-сайту, виокремлюючи ключові аспекти, які вимагають уваги та негайного втручання.

Аналіз підходів до ідентифікації ризиків дозволяє визначити основні причини вразливостей та потенційні точки атак, що стає критичним етапом у створенні системи захисту. Досліджені методи захисту, такі як двофакторна аутентифікація, регулярні апдейти програмного забезпечення та багатофакторний контроль доступу, визначено як оптимальні та ефективні.

У завершальному аспекті дослідження були розглянуті перспективи розвитку систем безпеки веб-сайтів, включаючи впровадження нових заходів безпеки, вдосконалення процесів моніторингу та реагування на потенційні загрози. Отримані результати підкреслюють важливість систематичного тестування та ідентифікації ризиків для ефективного захисту веб-сайтів, а розглянуті методи захисту є необхідним елементом в сучасному інформаційному середовищі. Ці

відомості становлять основу для подальшого розвитку стратегій безпеки та підвищення стійкості веб-платформ перед різноманітними загрозами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрухів А. І. Порівняння методів оцінки захищеності корпоративних інформаційних систем / А. І. Андрухів, Д. О. Тарасов // ВісникНаціонального університету «Львівська політехніка». – 2006. – № 573 : Комп'ютерні системи та мережі. – С. 3–9.
2. Брэгг Р., Родс-Оусли М., Страссберг К. Безопасность сетей. Полное руководство. – М.: Эком, 2006. - 912 с.
3. Жуков І. А., Гуменюк В. О., Альтман І. Є. Комп'ютерні мережі та технології (навчальний посібник). Видавництво Національного авіаційного університету "НАУ-друк", 2004. – 276 с.
4. Сычев Ю. Н. Основы информационной безопасности. Учебно- практическое пособие. – М.: Изд. Центр ЕАОИ, 2007. – 300 с.
5. Юдін О. К., Богуш В. М. Інформаційна безпека держави: Навчальний посібник. – Харків: Консул, 2005. – 576 с.
6. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу. Затверджено нак. Департаменту спеціальних телекомунікаційних систем та захисту інформації СБ України від “ 28 ” квітня 1999 р. № 22 із змінами згідно наказу Адміністрації Держспецзв'язку від 28.12.2012 № 806. - 61 с.
7. НД ТЗІ 2.5-005 -99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу. Затверджено наказом Департаменту спеціальних телекомунікаційних систем та захисту інформації СБ України від 28 квітня 1999 р. № 22. Зі Зміною №1, затвердженою наказом Адміністрації Держспецзв’язку від 15.10.2008, № 172. – 20 с.
8. В. Л. Бурячок, Р. В. Киричок, П. М. Складанний. Основи інформаційної та кібернетичної безпеки, Київ, 25.04.2019, 320 с.

9. Кавун С. В. Інформаційна безпека. Навчальний посібник. Ч. 2 / С. В. Кавун, В. В. Носов, О. В. Манжай. – Харків: ХНЕУ, 2008. – 196 с
10. Бурячок В. Л. Алгоритм оцінювання ступеня захищеності спеціальних інформаційно-телекомунікаційних систем // Захист інформації, Північна Америка, 13, вер. 2011. URL: <http://jrnl.nau.edu.ua/index.php/ZI/article/view/2028/2019>. (дата звернення: 26.11.2023).
11. Кібербезпека : сучасні технології захисту. Навчальний посібник для студентів вищих навчальних закладів. / С. Е. Остапов, С. П. Євсеєв, О.Г. Король. – Львів: «Новий Світ2000», 2020 . – 678 с.
12. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. URL: https://tzi.com.ua/downloads/1.1-002-99.pdf (дата звернення: 26.11.23).
13. 2019 Cyberthreets Defense Report, Group CyberEdge, 2019 URL: https://go.illusivenetworks.com/2019-cyberthreat-defense-report (дата звернення: 19.11.23).
14. ISO/IEC 29147:2018 Information technology — Security techniques — Vulnerability disclosure / 2018
15. Bell D. E., La Padula L. J. Security Computer System: Mathematical Foumdation // Mitre Techical Report 2547. — 1973. — 2. — 35 p.
16. Framework for Improving Critical Infrastructure Cybersecurity. National Institute of Standards and Technology, 2018. – 48 p.
17. ISO/IEC FDIS 27006-1. Information security, cybersecurity and privacy protection URL: https://www.iso.org/standard/82908.html (дата звернення: 23.11.23).
18. Johnson A., Dempsey K., Ross R., Gupta S., Bailey D. Guide for Security- Focused Configuration Management of Information Systems. - NIST Special Publication 800-128, 2011. – 99 p.
19. Kissel R., Stine K., Scholl M., Rossman H., Fahlsing J., Gulick J. Security Considerations in the System Development Life Cycle. - NIST Special Publication 800-64 Revision 2, 2008. – 68 p.

20. Shiva S., Roy S., Dasgupta D. “Game theory for cyber security”, CSIIRW '10: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, 2010, Article No.: 34, Pages 1–4.
21. WASC Threet Classification v. 2.0, 2010. URL: http://projects.webappsec.org/f/WASC-TC-v2_0.pdf (дата звернення: 15.11.23).
22. Історія кібербезпеки URL: https://www.education.ua/blog/48055/ (дата звернення: 15.11.23).
23. Захист цифрового кордону URL: https://itanddigital.ru/bloghrconsulting/tpost/1v5a0o55n1-istoriya-kiberbezopasnosti-zaschita-tsif (дата звернення: 15.11.23).
24. Як тестувати веб-сайт: основні етапи і поради BrainlabURL: https://brainlab.com.ua/uk/blog-uk/yak-testuvati-veb-sayt-osnovn-etapi-poradi/ (дата звернення: 24.10.2023).
25. Історія кіберзахисту: від 40-х років до наших днів URL: https://blog.avast.com/ru/history-of-cybersecurity-avast#the-1940s (дата звернення: 15.11.23).
26. OWASP Cheat Sheet Series URL: https://github.com/OWASP/CheatSheetSeries (дата звернення: 15.11.23).
27. OWASP TOP TEN [Електронний ресурс] / Andrew van der Stock, Brian Glas, Neil Smithline, Torsten Gigler - 2021— URL: https://owasp.org/www-project-top-ten/ (дата звернення: 15.11.23).
28. Penetration Testing Execution Standard (PTES) [Електронний ресурс] / [Chris Nickerson, Dave Kennedy, та ін.] - 2014 — URL: http://www.pentest-standard.org/index.php/Main_Page (дата звернення: 15.11.23).

ДОДАТОК А

Тип сповіщення	Ризик	Рахувати
Відсутній заголовок для захисту від клацання	Середній	7 (38,9%)
Відсутність токенів Anti-CSRF	Середній	5 (27,8%)
Заголовок політики безпеки вмісту (CSP) не встановлено	Середній	9 (50,0%)
Знайдено прихований файл	Середній	2 (11,1%)
Перегляд каталогів	Середній	4 (22,2%)
Підробка параметру	Середній	7 (38,9%)
Сервер витік інформації через поля заголовка HTTP-відповіді "X-Powered-By"	Низький	7 (38,9%)
Відсутній заголовок X-Content-Type-Options	Низький	27 (150,0%)
Прапорець HttpOnly відсутній	Низький	1 (5,6%)
Сервер поширює інформацію про версію через поле заголовка HTTP-відповіді «Сервер».	Низький	29 (161,1%)

Файл cookie без атрибута SameSite	Низький	1 (5,6%)
Fuzzer агента-користувача	Відомості	72 (400,0%)
GET для POST	Відомості	1 (5,6%)
Визначено відповідь до керування сесією	Відомості	32 (177,8%)
Виявлено запит на автентифікацію	Відомості	5 (27,8%)
Керований користувачем атрибут HTML-елемента (потенційний XSS)	Відомості	3 (16,7%)
Розголошення інформації - переконалі коментарі	Відомості	1 (5,6%)
Файли Cookie з вільними межами	Відомості	26 (144,4%)
Всього		18

Оповіщення

Risk = Середній , Confidence = Високий (2)

http://localhost (2)

Заголовок політики безпеки вмісту (CSP) не встановлено (1)

▶ ОТРИМАЙТЕ http://localhost/sitemap.xml

Знайдено прихований файл (1)

▶ ОТРИМАЙТЕ http://localhost/server-status

Risk = Середній , Confidence = Середній (2)

http://localhost (2)

Відсутній заголовок для захисту від клацання (1)

▶ ОТРИМАТИ http://localhost/bWAPP/portal.php

Перегляд каталогів (1)

▶ ОТРИМАТИ http://localhost/bWAPP/fonts/

Risk = Середній , Confidence = Низький (2)

http://localhost (2)

Відсутність токенів Anti-CSRF (1)

▶ ОТРИМАТИ http://localhost/bWAPP/portal.php

Підробка параметру (1)

▶ ПУБЛІКУВАТИ http://localhost/bWAPP/login.php

Risk = Низький , Confidence = Високий (1)

http://localhost (1)

Сервер поширює інформацію про версію через поле заголовка HTTP-відповіді «Сервер». (1)

▶ ОТРИМАТИ http://localhost/robots.txt

Risk = Низький , Confidence = Середній (4)

http://localhost (4)

Витік інформації з сервера через поля заголовка відповіді HTTP "X-Powered-By" (1)

▶ ОТРИМАТИ http://localhost/bWAPP/portal.php

Відсутній заголовок X-Content-Type-Options (1)

▶ ОТРИМАТИ http://localhost/bWAPP/stylesheets/stylesheet.css

Прапорець HttpOnly відсутній (1)

▶ ОТРИМАТИ http://localhost/bWAPP/portal.php

Файл cookie без атрибута SameSite (1)

▶ ОТРИМАТИ http://localhost/bWAPP/portal.php

Risk = Відомості , Confidence = Високий (2)

http://localhost (2)

GET для POST (1)

▶ ОТРИМАТИ http://localhost/bWAPP/user_new.php

Виявлено запит на автентифікацію (1)

▶ ПУБЛІКУВАТИ http://localhost/bWAPP/login.php

Risk = Відомості , Confidence = Середній (2)

http://localhost (2)

Fuzzer агента-користувача (1)

▶ ОТРИМАТИ http://localhost/bWAPP

Визначено відповідь до керування сесією (1)

▶ ОТРИМАТИ http://localhost/bWAPP/portal.php