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

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: *"Дослідження захищеності веб-сайту кафедри кібербезпеки ТНТУ"*

Виконав: студент (ка)
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Мороз О.П.
підпис (прізвище та ініціали)

Керівник *Золотий Р.З.*
підпис (прізвище та ініціали)

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

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

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2022

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

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

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Морозу Олександру Петровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження захищеності веб-сайту кафедри кібербезпеки ТНТУ

Керівник роботи Золотий Р.З., к.т.н., доцент каф. КТ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 23 » 03 2022 року № 4/7-178

2. Термін подання студентом завершеної роботи 17.06.2022

3. Вихідні дані до роботи Вимоги до програмного забезпечення

4. Зміст роботи (перелік питань, які потрібно розробити)
ВРАЗЛИВОСТІ ВЕБ-ДОДАТКІВ НА ОСНОВІ МЕТОДОЛОГІЇ OWASP
РОЗРОБКА ВЕБ-САЙТУ КАФЕДРИ КІБЕРБЕЗПЕКИ
ДОСЛІДЖЕННЯ ЗАХИЩЕНОСТІ ВЕБ-САЙТУ КАФЕДРИ
БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Пулька Ч.В., проф. кафедри МТ		

7. Дата видачі завдання 16.02.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	16.02 – 19.02	*Виконано*
2.	Підбір джерел про існуючі вразливості веб-сайтів	20.02 – 27.02	*Виконано*
3.	Опрацювання джерел в галузі дослідження	28.02 – 16.03	*Виконано*
4.	Розроблення сайту	17.03 – 20.03	*Виконано*
5.	Дослідження захищеності сайту	20.03-05.04	*Виконано*
6.	Оформлення розділу «Вразливості веб-додатків на основі методології OWASP»	06.03 – 17.04	*Виконано*
7.	Оформлення розділу «Розробка веб-сайту кафедри кібербезпеки»	18.04 – 29.04	*Виконано*
8.	Оформлення розділу «Дослідження захищеності веб-сайту кібербезпеки»	30.04 – 13.05	*Виконано*
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи хорони праці»	14.05 – 21.05	*Виконано*
10.	Оформлення кваліфікаційної роботи	22.05 – 05.06	*Виконано*
11.	Нормоконтроль	06.06 – 12.06	*Виконано*
12.	Перевірка на плагіат	10.06 – 15.06	*Виконано*
13.	Попередній захист кваліфікаційної роботи	16.06 – 19.06	*Виконано*
14.	Захист кваліфікаційної роботи	24.06.2022	

Студент _____ *Мороз О.П.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Золотий Р.З.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Дослідження захищеності веб-сайту кафедри кібербезпеки ТНТУ// Кваліфікаційна робота ОР «Бакалавр» // Мороз Олександр Петрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-42 // Тернопіль, 2022 // С. __ , рис. – __, табл. – __ , кресл. – , додат. – ___.

Ключові слова: ВЕБ-САЙТ, ВРАЗЛИВОСТІ, БЕЗПЕКА, SQL-ІН'ЄКЦІЇ, WORD-PRESS, APACHE

Кваліфікаційна робота по питанню дослідження безпеки вебсайту кафедри кібербезпеки ТНТУ. В роботі проаналізовано перелік найбільш небезпечних вразливостей веб-додатків загалом, та системи керування вмістом WordPress зокрема; обґрунтовано вибір WordPress як середовища розробки веб-сайту. Досліджено безпеку веб-сайту кафедри кібербезпеки та запропоновано дієві шляхи підвищення ступеня захисту. Продемонстровано налаштування параметрів безпеки для веб-сайту кафедри кібербезпеки.

Метою даної роботи є розробка, дослідження безпеки веб-сайту кафедри кібербезпеки, пошук шляхів підвищення стійкості сайту до окремих видів атак шляхом налаштування параметрів безпеки.

В першому розділі описано різновиди найпопулярніших вразливостей веб-додатків, наведено приклади їх використання та запропоновано шляхи усунення. В другому розділі обґрунтовано вибір середовища розробки веб-сайту, проаналізовано основні вразливості Word-Press та наведено окремі важливі адміністрування веб-сайту. В третьому розділі наведено результати запропонованих заходів підвищення безпеки сайту. В четвертому розділі висвітлено окремі питання охорони праці та безпеки життєдіяльності

ANNOTATION

Study of security of the TNTU Cybersecurity Department website // Thesis of educational level "Bachelor" // Moroz Oleksandr Petrovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, СБс-42 group // Ternopil, 2022 // P. ___ , fig. -____, table. - __ , chair. - ___ , added. -___.

Keywords: WEB-SITE, VULNERABILITIES, SECURITY, SQL-INJECTION, WORD-PRESS, APACHE.

Qualification paper is devoted to research of the website of the Department of Cybersecurity of TNTU securit. The paper analyzes the list of the most dangerous vulnerabilities of web applications in general, and WordPress content management systems in particular; the choice of WordPress as a website development environment is justified. The security of the website of the Department of CyberSecurity has been studied and effective ways to increase the level of protection have been suggested. Configuration of security settings for the website of the Department of Cybersecurity is demonstrated.

The purpose of this work is to develop, study the security of the website of the Department of Cybersecurity, find ways to increase the resilience of the site to certain types of attacks by adjusting security settings.

The first section describes the types of most popular web application vulnerabilities, provides examples of how to use them, and suggests ways to resolve them. The second section substantiates the choice of website development environment, analyzes the main vulnerabilities of Word-Press and presents some important website administrations. The third section presents the results of the proposed measures to improve site security. The fourth section covers some issues of labor protection and life safety.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CSRF	Cross-Site Request Forgery
CMS	Content Management System
IDOR	Insecure Direct Object References
https	Hypertext Transfer Protocol Secure
LAMP	Linux, Apache, MySQL, PHP
XSS	Zero Knowledge Password Proof
SQL	Structured Query Language
IT	Інформаційні технології

ВСТУП

Пандемія коронавірусу 2020 докорінно змінила співвідношення віртуального та реального життя. Тотальні карантинні обмеження змусили людей шукати онлайн-аналоги для вирішення тих чи інших задач. Купівля та пошук необхідних товарів в онлайн-магазинах стала популярною з колосальною швидкістю, пошук необхідної інформації, онлайн-навчання зробило браузер та веб-додатки невід'ємною частиною життя людства. Але поруч зі зручностями зросли і загрози. Нехтування елементарних правил інформаційної гігієни в кіберпросторі призвело до небувалого зростання злочинності в інтернеті, веб-додатки стали мішенню багатьох хакерів та кіберзлочинців.

При цьому всі компанії, що мають сайт в інтернеті, діляться на три типи:

- ті, чий сайт вже зламали;
- ті, чий сайт ще не ламали;
- ті, хто знайомий з основними векторами атак і захистив додатки.

Представникам третьої категорії подальший виклад навряд чи буде дуже цікаво. Іншим двом категоріям слід як мінімум знати про основні вектори атак на веб-додатки та можливості їх практичного усунення - адже, попереджений, - майже захищений.

Акцент на можливості практичного застосування зроблений не випадково. Ми вважаємо, що недостатньо знати теорію для розуміння істинного масштабу загрози - адже вразливості, які виглядають не дуже страшно в теорії, найчастіше можуть нести катастрофічні для бізнесу наслідки. Більшість атак відбуваються через кілька помилок в перевірці даних, вразливостях програмного забезпечення, як написаного власноруч, так і на стороні відомих вендорів, або ж, банально, через налаштування параметрів за замовчуванням не надто професійними діями системних адміністраторів.

Сайт кафедри кібербезпеки був створений з допомогою CMS Wordpress, яка має свої вразливості. Існують багато вразливостей і на стороні сервера. Тому актуальною є задача вивчення та дослідження безпеки сайту та налаштування параметрів безпеки.

Метою даної роботи є дослідження безпеки веб-сайту кафедри кібербезпеки, пошук шляхів підвищення стійкості сайту до окремих видів атак шляхом налаштування параметрів безпеки.

Досягнення наведеної вище мети вимагає розв'язання наступних задач:

- Огляд літературних джерел за темою кваліфікаційної роботи.
- Аналіз основних вразливостей веб-додатків та CMS WordPress, зокрема.
- Аналіз основних підходів для усунення вразливостей.
- Обґрунтування вибору середовища розробки.
- Розробка веб-сайту кафедри кібербезпеки.
- Налаштування параметрів для безпеки сайту.

1. ВРАЗЛИВОСТІ ВЕБ-ДОДАТКІВ НА ОСНОВІ МЕТОДОЛОГІЇ OWASP

OWASP (Open Web Application Security Project). Це міжнародна некомерційна організація, основними завданнями якої є пошук основних вразливостей та забезпечення безпеки веб-сайтів

Згідно OWASP найпопулярнішими вразливостями є:

- Ін’єкція
- Порушена аутентифікація
- Відкриття конфіденційних даних
- Зовнішні об’єкти XML (XXE)
- Порушений контроль доступу
- Неправильні налаштування безпеки
- Міжсайтові сценарії (XSS)
- Небезпечна десеріалізація
- Використання компонентів з відомими вразливими місцями
- Недостатній облік та моніторинг

В цьому списку зібрані найнебезпечніші уразливості, які які можуть призвести до великих матеріальних втрат, бізнес-репутації і т.д. Розглянемо детальніше деякі з них.

1.1 Загальне поняття про ін'єкції

SQL (Structured Query Language - структурована мова запитів) – це мова програмування спеціального призначення, призначена для керування даними в реляційній базі даних, і використовується величезною кількістю програм і організацій. Відомо, що база даних є невід’ємною частиною кожного веб-додатку. SQL-запити використовують, щоб отримати доступ до даних, які, власне, зберігаються в цих базах даних. Якщо приділити недостатню увагу безпеці веб-додатку, то зловмисник може впровадити в форму Web-інтерфейсу

додатку спеціальний код (ін’єкцію), що містить шматок SQL-запиту, проте є шкідлим за своєю суттю.

Ін’єкція – це шкідливий код, впроваджений в мережу, який передає всю інформацію з бази даних до зловмисника. Цей тип атаки вважається основною загрозою веб-безпеки і входить до списку ризиків безпеки веб-додатків як номер один у рейтингу OWASP Top 10. Під час ін’єкційної атаки зловмисник може впровадити шматок зловмисного коду в запит, що дозволить йому змінити роботу програми, змусивши її виконувати певні команди, для прикладу, читати/змінювати/видаляти інформацію, яка була розміщена в базі даних. Ін’єкційна атака може розкрити дані або пошкодити їх, а також призвести до відмови в обслуговуванні або повного компромісу веб-сервера [2].

Ця вразливість виникає внаслідок того, що дані, які надходять від користувача не проходять достатньої перевірки. Це дозволяє зловмисникові «підсунути», спеціально підготовлені шкідливі запити, які «обдурять» додаток і дозволять виконати деякі нелегітимні дії.

В загальному, цей SQL-ін'єкції належать до більш загальної групи «Помилки валідації», проте це неєдиний вид атак, що належить до цієї групи.

1.1.1 Практичний погляд на небезпеку ін'єкцій

Всі системи (складні і не дуже) зберігають велику кількість даних і різних об'єктів. Наприклад об'єктів бізнес-логіки, таких як облікові записи користувачів, рахунки в системі, транзакції, в журналах операцій, і т.д.

Для того, щоб до цих об'єктів можна було звертатися, кожен з них, в тому чи іншому вигляді, має свій унікальний ідентифікатор. Крім того, об'єкти можуть містити довільні набори полів. Всі поля також мають свої унікальні імена або ідентифікатори.

Всі ці об'єкти, і не тільки ці, зберігаються в таблицях, де кожен рядок - 1 об'єкт.

Наприклад, об'єкт «Клієнт» може мати наступний набір полів: id, ім'я, прізвище, e-mail, мобільний телефон, і зберігатися в таблиці виду:

Таблиця .1 - Таблиця клієнта

id (идентификатор)	cl_name (имя)	cl_sur_name (фамилия)	e-mail	cell (мобильный)	other (и т.п.)
1	Ivan	Ivanov	ivan@mail	+70000000000	42

Суть взаємодії користувача з Web-додатком полягає в обміні запитами між браузером і сервером, де браузер відправляє серверу різні параметри і отримує від сервера відповідь.

Щоб отримати або змінити об'єкти (наприклад, отримати виписку по рахунку або оновити свою анкету) необхідно запросити ту чи іншу сторінку на сервері (далі по тексту - скрипт). Web-додатки з цією метою використовують форми або виклики безпосередньо URL скрипта.

Для обміну даними між браузером і сервером був розроблений протокол HTTP. Дані передаються у вигляді так званих HTTP-запитів, кожен з яких складається з заголовка і тіла запиту.

Для передачі даних з браузера користувача на сервер по HTTP-протоколу в основному використовуються два методи - GET і POST (існують ще методи PUT і DELETE, але вони використовуються переважно в API).

При передачі даних методом GET, всі параметри запиту передаються в URL сторінки в заголовку HTTP-запиту, наприклад: «http://simplethreats.ru/get_order?order=150"».

У разі використання методу POST, сервер буде отримувати дані вже в тілі HTTP-запиту, а в URL параметри перераховані не будуть. Таким чином, наприклад, відправляється більшість форм, і завжди критичні аутентифікаційні дані (передача даних в URL небезпечна, адже записи про URL і заголовки HTTP-запитів залишаються в безлічі журналів).

Звернення через GET або POST можуть проводитися до будь-яких об'єктів, будь то записи в базі даних або ж читання файлу (скажімо, картинки або скрипта з диска). А так як об'єктів бізнес-логіки в системі може бути дуже багато

і вони можуть мати самі різні типи, то і звернень по ідентифікаторів в рамках одного, навіть простого Web-додатки може відбуватися безліч.

З тим, яким чином дані потрапляють в Web-додаток ми визначилися, а що ж відбувається всередині?

1.1.2 Зміни параметрів запиту

Web-додатки, як правило, будують SQL запити, що поєднують код написаний розробником програми, з параметрами переданими користувачем. Розглянемо приклад: «SELECT title,text FROM news WHERE id=$id"».

У цьому прикладі, $ id - параметр передається користувачем (змінна частина запиту), в той час як інша частина запиту статична і була закладена розробником програми. Наявність змінних призначених для користувача даних в статичному SQL запиті, робить весь запит динамічним. У чому суть ін'єкції?

У випадку SQL- ін'єкції, її суть полягає в тому, щоб модифікувати параметри HTTP-запиту таким чином, щоб спотворити SQL запит до бази даних і «підсунути» його серверу під виглядом нормального. Це дозволить зловмиснику отримати несанкціонований доступ до даних.

Припустимо, в базі даних є таблиця «transactions» з даними про транзакції користувачів. Вона містить такі поля:

«User_id» - Унікальний ідентифікатор користувача

«Date» - Дата

«Amount» - Сума

«Description» - Призначення

Таблиця 1.2 - Таблиця транзакцій

user_id	date	amount	description
10	2015-05-26	1000	...
11	2015-05-26	1500	...
12	2015-05-26	1300	...
n	2015-05-26	x	...

Припустимо, що наш аккаунт, під яким ми перебуваємо в додатку, має ідентифікатор 10 (user_id = 10) і ці дані зберігаються в сесії додатки (були записані туди при авторизації користувача).

У загальному вигляді, якщо ми не привілейований користувач, ми можемо звернутися до бази даних для отримання інформації за своїми транзакціях, тобто за транзакціями user_id = 10.

Для того, щоб нам отримати дані про свої транзакції за 26 травня 2015р. в кабінеті користувача може використовуватися сторінка з наступним адресою:

«http://mybank.simpletreats.ru/transactions.jsp?date=2015-05-26».

При виклику якого дата «2015-05-26» буде отримана з GET-параметра в URL, а значення user_id буде отримано з сесії додатки. На основі цих даних, для отримання інформації про транзакції користувача, додаток сформує SQL-запит:

```
«SELECT * FROM transactions WHERE date = "2015-05-26" AND user_id = 10».
```

Для тих, хто не знайомий з синтаксисом SQL для кращого розуміння, розберемо цей запит.

Запит складається з оператора SELECT (дослівно з англ. - «ВИБРАТИ»), який використовується для вибору даних з таблиці. Існують ще оператори UPDATE, INSERT, DELETE, які, як неважко здогадатися з їх назв, виконують операції поновлення, вставки і видалення рядків відповідно.

Символ «*» означає, що ми вибираємо всі стовпці таблиці. Далі слід ключове слово FROM (дослівно - «З») із зазначенням імені таблиці, з якої виробляється вибірка.

Далі йде ключове слово WHERE (дослівно з англ. «ДЕ»), за яким слідує частина, яка визначає безпосередньо умови вибірки з таблиці.

Після цього йдуть умови виду «названіе_поля = значення» розділені ключовими словами AND або OR, які дають зрозуміти «І» та «АБО» відповідно.

Таким чином, переводячи запит з SQL українською ми отримаємо запит, *«ВИБРАТИ всі ПОЛЯ З таблиці transactions, де поле дата=»2015-05-26 І поле user_id=10»*:

Можливість впровадження SQL-ін’єкцій – це найнебезпечніша вразливість, дозволить зловмисникові читати / змінювати / видаляти інформацію, яка для нього не призначена.

1.1.3 Експлуатація ін'єкцій

Як це працює? У розглянутому прикладі, значення дати "2015-05-26" потрапляє в SQL-запит з URL скрипта, і в цілях даного прикладу, ніяк не фільтрується додатком.

Якщо передати в параметр date разом з датою 2015-05-26 ще трохи символів: «"2015-05-26" AND user_id = 11 --».

То замість коректного:

```
«SELECT * FROM transactions WHERE date = "2015-05-26" AND user_id = 10».
```

Вказаний запит сформує наступний SQL:

```
«SELECT * FROM transactions WHERE date = "2015-05-26" AND user_id = 11 -- "AND user_id = 10"».
```

Два мінуса «-» в синтаксисі SQL означають початок коментаря, тому всі символи після «-» інтерпретатором сприйматися не будуть, і частина запиту, в якій здійснювалася перевірка ідентифікатора користувача буде відсічена сервером бази даних. Фактично буде виконаний запит:

```
«SELECT * FROM transactions WHERE date = "2015-05-26" AND user_id = 11».
```

В результаті якого, ми отримаємо інформацію про транзакції іншого користувача. А перебираючи id - будь-якого іншого користувача.

Які бувають техніки атак при SQL-ін'єкції?

Співтовариством OWASP були описані п'ять основних методів (технік) атак при SQL-ін'єкції.

Оператор Union: підхід може бути використаний при наявності уразливості в запиті SELECT, що дозволяє об'єднати два запити в один результат або набір результатів.

Логічний метод: передбачає використання логічного умови, або умов, що дозволяють достовірно визначити істинність або хибність якогось припущення.

На підставі помилок: цей метод передбачає навмисну передачу різних некоректних запитів, з метою змусити web-додаток видати інформацію про помилку, на підставі якої, зловмисник може скласти і передати коректний інжектовані запит.

Метод з альтернативним каналом передачі даних: метод передбачає використання альтернативного каналу передачі витягнутих даних (наприклад через що виходить HTTP з'єднання з web-сервером)

Time delay: метод використовує команди бази даних, наприклад sleep для того щоб визначити затримаю за умовними запитам. Метод ефективний, коли немає можливості отримати відповідь від web-додатки (результат, помилка)

Крім того, може використовуватися комбінований підхід, що включає в себе поєднання двох або більше перерахованих технік.

За способом отримання даних виділяють три типи атак:

Пов'язаний: дані в результаті инжектированного SQL запиту витягуються тим же шляхом, яким був переданий сам інжектовані запит. Це самий прямолінійний вид атаки, в результаті якого, запитані модифікованим запитом дані, відображаються безпосередньо на сторінці web-додатки;

Чи не пов'язаний: дані в результаті инжектированного SQL запиту витягуються шляхом, відмінним від способу передачі модифікованого запиту. (Наприклад дані передаються зловмисникові в повідомленні на електронну пошту);

Дедуктивний або сліпий: В результаті SQL ін'єкції фактичного вилучення даних не відбувається, але зловмисник може отримати інформацію,

спостерігаючи за поведінкою web-сервера, в результаті відправки серії специфічних інжектованих SQL запитів.

1.1.4 Захист від ін’єкцій

Рекомендація вкрай проста - фільтрувати вхідні дані. При цьому до фільтрації краще не доводити дані, явно не відповідають формату. Іншими словами, дані на вході потрібно перевіряти на відповідність формату і видати користувачеві помилку.

Наприклад, ми повинні отримати ID новини для відображення. У нашій системі це завжди ціле число більше нуля, і ми про це знаємо. У разі якщо передано не число, або це число менше або дорівнює нулю, потрібно видати помилку і припинити виконання скрипта.

Як перейти до імплементації даної рекомендації, якщо у вас вже є велике web-додаток, що має багато різних точок входу?

Ми наведемо деякий системний підхід до реалізації захисту від ін'єкцій баз даних. Щоб звести до мінімуму потенційну загрозу від даного типу атак, потрібно послідовно виконати наступні кроки:

Визначити всі крапки отримання даних web-додатком ззовні, виписати всі точки входу (URL), назви переданих параметрів, протокол (HTTP / HTTPS) і метод їх передачі - GET, POST, PUT або DELETE.

Визначити тип даних для кожного параметра, відповівши на питання - значення якого типу ми повинні отримати?

Реалізувати перевірку кожного параметра на відповідність типу.

Реалізувати екранування лапок та інших спецсимволів за допомогою символу зворотного слеша "\". При цьому якщо параметр в запиті укладений в одинарні лапки, то саме одинарні лапки повинні бути екрановані і замінені на «\ '», аналогічно з подвійними лапками. При цьому екранувати одинарні лапки всередині подвійних не треба і навпаки. Це в теорії, на практиці ж, майже у всіх популярних фреймворків і скриптових мовах є вбудовані функції для екранування спецсимволов, як наприклад mysqli_real_escape_string в php.

У загальному вигляді, можна також рекомендувати пропускати всі запити до бази даних через якусь єдину точку (наприклад, фреймворк), при проходженні через яку вони будуть проходити спеціальну підготовку - фільтрацію і екранування.

Слід також пам'ятати, що для успішної атаки за допомогою SQL ін'єкції зловмисникові потрібно передати синтаксично правильний SQL запит. Однак, якщо web-додаток повертає інформацію про помилку в результаті неправильного инжектированного запиту, це може допомогти зловмисникові відновити логіку вихідного запиту і дати йому можливість зрозуміти, як правильно виконати ін'єкцію.

Тому, потрібно уважно стежити за тим, яка інформація про помилки віддається додатком. Проте, якщо web-додаток приховує відомості про помилки, у розробника в будь-якому випадку повинна бути можливість відновити логіку вихідного запиту (в найпростішому випадку - вести лог помилок, але не виводити їх на екран).

1.1.5 Інші типи ін'єкцій

Крім ін'єкцій баз даних, атаці типу «ін'єкція» може бути піддана будь-яка інша середовище, яке отримує необроблені дані ззовні. Ще один поширений випадок - це ін'єкція командного інтерпретатора операційної системи, так звані «OS injections».

Розглянемо такий приклад з опису «Command injections» OWASP.

Є якась соціальна мережа, що надає користувачам функціонал завантаження фотографій і їх подальшого видалення. У ній є якийсь скрипт, написаний на мові PHP зображений на рисунку 2.1, і він відповідає за видалення фотографій:

Лістинг 1.1 – Скрипт видалення фотографій

```
<?php
          $file = $_GET['filename'];
 system ("rm /var/www/user_photos/".$file);
?>
```

Типовий його виклик виглядає так: «http://mysocnet.simplethreats.ru/user_file_delete.php?filename=1246.jpg». І тягне за собою виконання команди: «rm /var/www/user_photos/123456.jpg». Дуже легко очистити всю директорію з додатком, передавши такий запит: «user_file_delete.php?filename=.../ -rf». (Насправді прогалини і якісь символи Закодуйте при передачі в URL в щось виду user_file_delete.php? Filename = ..% 2F + -rf - але для наочності, ми будемо писати з пробілами та іншими символами, що не загортаючи їх) що викличе виконання: «rm /var/www/user_photos/.. -rf» і буде рівносильно «rm /var/www/ -rf» тобто видалить (команда rf) весь вміст директорії www, де зберігаються файли програми, без підтвердження (параметр -f) і рекурсивно (параметр -r), тобто з усіма вкладеними директоріями і їх файлами. Участь - ворогу не побажаєш.

А можна не видаляти файлову систему, і передавши запит виду:

```
«user_file_delete.php?filename=123456.jpg && adduser ghost && echo ghostpass | passwd sghost -stdin»
```

Зловмисник виконає команду: «rm /var/www/user_photos/123456.jpg && adduser ghost && echo ghostpass | passwd ghost -st din»). І створить для себе обліковий запис для доступу на сервер.

Потрібно відзначити, що на практиці створити користувача таким чином майже неможливо, так як Web-сервер в 99% систем запущений від користувача без розширених прав доступу, який не може створювати інші облікові записи.

Виконання зазначеної ланцюжка команд можливо завдяки двом особливостям UNIX-систем і їх командних інтерпретаторів.

Перша особливість - це конвеєри (pipelines, pipes, «пайпи»). Суть конвеєрів полягає в можливості передачі виведення однієї команди на введення інший, якщо вони розділені оператором «|».

Друга особливість - можливість запускати комбінацію команд через логічні оператори «&&» і «||». Оператор «&&» виконає наступну зазначену

команду, в разі якщо попередня виконана успішно і є таким собі аналогом логічного «І» - виконати команду один І команду два. Оператор «||» є аналогом логічного «АБО» і виконає другу команду тільки в разі якщо не була виконана перша - виконати команду один АБО команду два.

Зрозуміло, була розглянута лише мала дещиця того, що можна віднести до розряду ін'єкцій. Величезний пласт займають так звані DOM- або HTML-ін'єкції, завдяки яким стає можливим проведення такого виду атак як XSS.

Проблема XSS в усі часи стояла досить гостро, а зараз, в епоху динамічного вмісту, коли JavaScript надає дуже потужні можливості, атаки такого роду несуть підвищену небезпеку (аж до того, що, заражене додаток може сфотографувати вас і відправити Ваше фото зловмисникові).

1.2 Ризики, пов’язані з аутентифікацією і зберіганням сесій

Автентифікація — це процес перевірки того, що фізична особа, організація або веб-сайт є тим, за кого вони себе видають. Аутентифікація в контексті веб-програм зазвичай виконується шляхом подання імені користувача або ідентифікатора та одного або кількох елементів приватної інформації, які повинен знати лише певний користувач.

Управління сеансом – це процес, за допомогою якого сервер підтримує стан об’єкта, який з ним взаємодіє. Це потрібно для того, щоб сервер запам’ятав, як реагувати на наступні запити протягом транзакції. Сеанси підтримуються на сервері за допомогою ідентифікатора сеансу, який може передаватися назад і вперед між клієнтом і сервером під час передачі та отримання запитів.

Cookie-файли є, фактично, є основним методом аутентифікації для більшості сайтів. Для типової моделі входу користувач повинен ввести своє ім’я користувача та пароль, щоб увійти на сайт. Облікові дані надсилаються та перевіряються на сервері веб-сайту, а у відповідь надсилається файл cookie. Цей файл cookie потім використовується сайтом для перевірки користувача під час наступних відвідувань. Файл cookie аутентифікації може бути або сесійним файлом cookie, термін дії якого закінчується після закриття браузера, або

постійним файлом cookie, який розпізнається сервером навіть після закінчення сеансу браузера до певної дати закінчення терміну дії.

Файли cookie для аутентифікації дозволяють здійснювати зручну онлайн-автентифікацію користувачів, але поруч з цим і зростають потенційні проблеми з безпекою, пов’язані з тим, що зловмисник зміг отримати файл cookie аутентифікації даного користувача. Якщо в системі не передбачено заходів безпеки, таких як перевірка IP-адреси, сесії, або наявності більш одного з'єднання в одній сесії, зловмисник зможе отримати доступ до системи з правами вашого облікового запису. Якщо веб-додаток пов’язаний з даними банківських карток, то наслідки такого доступу стають абсолютно зрозумілими.

1.3 Міжсайтові сценарії - XSS

Міжсайтові сценарії (також відомі як XSS) — це вразливість веб-безпеки, яка дозволяє зловмиснику скомпрометувати взаємодію користувачів із вразливою програмою. Атаки такого роду часто також називають HTML-ін'єкціями, оскільки за механізмом виконання вони схожі з SQL-ін'єкціями, але на відміну від останніх, впроваджуваний код виконується на стороні браузера. Вразливості міжсайтових сценаріїв зазвичай дозволяють зловмиснику маскуватися під користувача-жертву, виконувати будь-які дії, які може виконати користувач, і отримати доступ до будь-яких даних користувача. Якщо користувач-жертва має привілейований доступ до програми, то зловмисник може отримати повний контроль над усіма функціями та даними програми.

Міжсайтові сценарії працюють шляхом маніпулювання вразливим веб-сайтом, щоб він повертав користувачам шкідливий JavaScript. Коли шкідливий код виконується у браузері жертви, зловмисник може повністю порушити їхню взаємодію з програмою. Це загрожує тим, що зловмисник може отримати доступ до сесійного cookie-файлу, наслідки цієї події описані в п.1.2. В результаті виконання цієї атаки можуть бути вкрадені конфіденційні дані. Крім того, через JavaScript можна змінювати дані, розміщені на сторінці.

1.4 Крадіжка конфіденційної інформації

Крадіжка cookies (document.cookie), а також інформації про систему користувача, браузер, поточний час, IP-адресу, інформацію про відвідані сайти може дати зловмиснику багато важливої інформації про користувача, а також бути серйозною загрозою безпеки користувача.

Нижче(див. лістинг 1.1) приведено зразок скрипта IP-адресу відвідувача у змінну IP, а також ім'я комп'ютера в змінну host (перевірено в Mozilla, Opera):

Лістинг 1.1 – Скрипт, що повертає IP-адресу та ім’я комп’ютера

```
myAddress = java.net.InetAddress.getLocalHost ();
myAddress2 = java.net.InetAddress.getLocalHost ();
host = myAddress.getHostName ();
IP = myAddress2.getHostAddress ();
```

Щоб викрасти дані, зловмисник може створити у себе на сервері прозоре gif-зображення і php скрипт, який буде приймати ідентифікатори сесії і зберігати їх в визначеному місці. Після цього зловмиснику потрібно розмістити простенький код на сайті-жертві.

Якщо жертва відкриє сторінку з впровадженим зловмисним кодом, то відбудеться наступне:

1. Завантаження gif-зображення, якого користувач не помітить, бо зображення прозоре.

2. Спрацювання події onload, яке виконує JavaScript внаслідок завантаження прозорого зображення

3. Передача ідентифікатор сесії зловмиснику в результаті виконання простого php-скрипта.

Після того, як ідентифікатори сесії стали відомими зловмиснику, в нього є небагато часу щоб ними скористатися, зазвичай від 30 до 60 хвилин в залежності від налаштувань сервера. Якщо зловмисник встигне скористатись

отриманим кодом в своєму браузері, то отримає доступ до сайту під чужим логіном, який в найгіршому випадку може бути логіном адміністратора.

1.5 Небезпечні прямі посилання на об'єкти

Небезпечні прямі посилання на об'єкти (Insecure direct object references - IDOR) це тип уразливості контролю доступу, яка виникає, коли програма використовує ввід, наданий користувачем, для безпосереднього доступу до об’єктів. Термін IDOR був популяризований завдяки його появі в десятці найкращих OWASP 2007. Однак це лише один приклад багатьох помилок реалізації контролю доступу, які можуть призвести до обходу контролю доступу. Уразливості IDOR найчастіше пов’язані з горизонтальною ескалацією привілеїв, але вони також можуть виникати у зв’язку з вертикальною ескалацією привілеїв.

Суть цієї вразливості полягає в тому, що при виведенні будь-яких конфіденційних даних, наприклад даних карток викладачів, розкладів групи чи викладачів, для доступу до об'єкта використовується ідентифікатор, який передається у відкритому вигляді в адресному рядку браузера. Якщо при цьому відсутня, прав доступу до об'єктів, то перебираючи інформацію користувач може отримати доступ до даних інших людей. Наприклад, є сторінка, яка відображає розклад викладача і вона має адресу виду: «https://tntu.edu.ua /?p=uk/schedule&t=Золотий+Роман+Захарійович».

Змінюючи прізвище, ім’я та по-батькові викладача " schedule&t " можна отримати доступ до чужого розкладу. Експлуатація даної вразливості дуже проста і не вимагає взагалі ніяких спеціальних навичок. У випадку з розкладом, це не є смертельним, проте на окремих сайтах таким чином можна читати чужі секретні повідомлення, або ж переглядати чужі транзакції в банківських системах.

1.6 Вразливості конфігурації вебсайтів

Безпека Web-додатків вимагає безпечної конфігурації всіх складових інфраструктури: починаючи від фреймфорків - компонентів програми, і, завершуючи налаштуваннями веб-сервера, сервера баз даних і самої платформи.

Неправильна конфігурація безпеки виникає, коли проблеми в конфігураціях програми дозволяють несанкціонований доступ, використання, модифікацію та вилучення даних. Наприклад, програми, які неправильно фільтрують вхідні пакети, можуть дозволити несанкціоноване використання ідентифікатора користувача або пароля за замовчуванням

Іншим прикладом може бути крадіжка сесійного cookie через JavaScript шляхом XSS-атаки, яка можлива через налаштування сервера за замовчуванням: cookie_http only.

Правильне налаштування сервера при включеній опції cookie_httponly, робить неможливим отримати сесійний cookie через JavaScript. Часто ця просте і важливе налаштування відсутнє в таких критично важливих додатках, як особисті кабінети платіжних систем.

Загрози також несуть налаштування за замовчуванням в серверах баз даних, таких як Redis, Memcached і інших. Можуть використовувалися паролі, встановлені виробником за замовчуванням або ж службова інформація може бути розміщена на публічній IP-адресі сервера. Це дає зловмисникові можливості для читання та модифікації даних, в тому числі і сесійних cookies.

Важливим є оновлення ПЗ: уразливості знаходять кожен день в самих різних програмних компонентах - операційній системі, web-серверах, серверах баз даних, поштових серверах і т.д.

Крім того існують рекомендації використовувати мінімальний інсталяційний набір без непотрібних компонент, функцій чи документації. Бажано налаштовувати сегментовану архітектуру програми та автоматизовувати процеси.

1.7 Незахищеність критичних даних

Важливо пам’ятати, що такі критичні дані, як банківські карти, облікові дані для аутентифікації мають бути особливо захищені. Адже, в протилежному випадку, зловмисники можуть скористатись вразливостями та вкрасти або модифікувати такі слабо захищені дані.

Найпростіший приклад - передача даних по протоколу HTTP, а не HTTPS. Протокол HTTP не шифрує дані проходженні від комп'ютера користувача до Web-сервера. Тож дані у відкритому вигляді потрапляють на досить багато різних вузлів: маршрутизатор офісу або домашній роутер, маршрутизатор провайдера, маршрутизатор в дата- центрі хостинг-провайдера сервера і так далі. На кожному з цих вузлів може бути використана програма - сниффер, яка зчитує весь трафік і передає зловмисникові.

Критичні дані повинні передаватися виключно за протоколом HTTPS, про що свідчить відповідний запис в адресному рядку браузера, зображений на рисунку 1.1:

Рисунок 1.1 - Захищене зєднання

HTTPS використовує SSL-сертифікат – це сворідний спеціальний ключ, за допомогою якого здійснюється перевірка справжності та шифрування, щоб підтвердити, що він виданий саме для даного сайту. У разі, якщо сертифікат протермінований або підроблений, Ви побачите результат на рисунку 2.2:

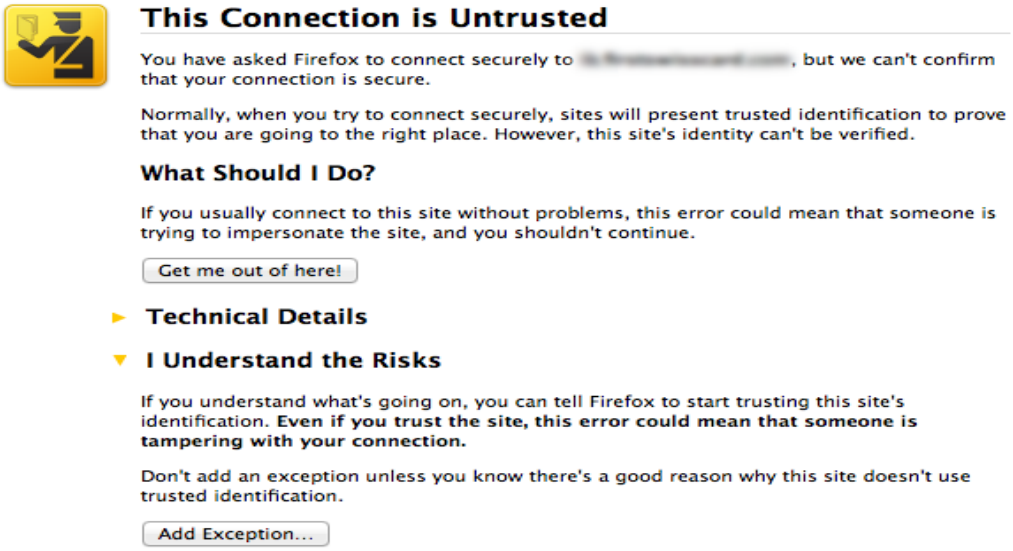

Рисунок 1.2 - Протермінований сертифікат

Ще одним прикладом може бути відсутність шифрування критичних даних, таких як паролі або номери кредитних карт. Якщо дані зашифровані, то навіть у разі отримання несанкціонованого доступу на сервер, зловмисник не зможе їх викрасти та скористатись ними. Паролі прийнято зберігати не у відкритому вигляді, а з допомогою односторонньої хеш-функції (перевірка пароля відбувається шляхом формування хеша введеного пароля і порівняння її з наявним хешом в базі).

1.8 Порушення контролю доступу

В ідеалі веб-додатки повинні робити кожну частину інформації доступною лише для певних користувачів відповідно до їхніх привілеїв. Порушений контроль доступу може призвести до критичних ризиків безпеки, дозволяючи користувачам отримати доступ до інформації, на яку вони насправді не мають права доступу.

Запобігти порушенню доступу можна такими простими кроками:

– налаштувати безпечний життєвий цикл розробки, який включає розробку й оцінку засобів безпеки та конфіденційності;

– налаштувати бібліотеку безпечних шаблонів проектування;

– використовувати моделювання загроз для критичного контролю доступу, аутентифікації, ключових потоків і бізнес-логіки;

– інтегрувати засоби контролю безпеки в історії користувачів;

– виконувати перевірку правдоподібності для кожного рівня програми, від front-end до back-end;

– Залежно від необхідного впливу та захисту, розділити системні та мережеві рівні.

1.9 Міжсайтова підробка запиту

Підробка міжсайтових запитів (CSRF) — це атака, яка змушує кінцевого користувача виконувати небажані дії у веб-додатку, в якому він наразі автентифікований (або ж зловмисник виконує дії від імені жертви). За допомогою соціальної інженерії (наприклад, надсилання посилання електронною поштою чи чатом) зловмисник може обдурити користувачів веб-програми, щоб вони виконували дії за вибором зловмисника. Якщо жертвою є звичайний користувач, успішна атака CSRF може змусити користувача виконувати запити на зміну стану, наприклад переказ коштів, зміну адреси електронної пошти тощо. Якщо жертвою є обліковий запис адміністратора, CSRF може скомпрометувати всю веб-програму.

Розглянемо деяку сторінку платіжної веб-системи для переказу коштів на інший рахунок:

```
demobank.com/transfer_money.jsp?transfer_amount=1000&transfer
_account=123456789
```

де transfer_amount - сума для переказу і transfer_account - номер аккаунта, куди повинні бути переведені кошти.

Якщо жертва відвідує сайт, створений зловмисником, від її особи таємно відправляється запит на вищевказану сторінку платіжної системи, в результаті чого гроші підуть на рахунок зловмисника, частіше всього отримати їх назад вже не вийде.

Підробку міжсайтових запитів (CSRF) було вилучено зі списку 10 top OWASP за 2017 рік, оскільки це менш значуща загроза в нинішньому технологічному середовищі. Це не означає, що нам не потрібен захист CSRF – це лише означає, що більшість організацій і фреймворків вже захищають себе, тому OWASP вважає це менш серйозним ризиком.

1.10 Використання компонентів з відомими вразливостями

При написанні web-додатків часто використовують спеціальні бібліотеки або «фреймворки» (англ - framework), які поставляються сторонніми компаніями. У більшості випадків ці компоненти є open-source (з відкритим кодом), а це дає не лише переваги, але й несе небезпеки, пов'язані з тим, що зловмисники можуть знайти zero-day вразливості і використати їх для атаки.

Вразливості також шукають і знаходять) в більш низько-рівневих компонентах системи, таких як сервер бази даних, web-сервер, і нарешті, компоненти операційної системи аж до її ядра, тому важливо дбати і про безпеку операційних систем, і не забувати про останні версії оновлень компонентів системи, а також відстежувати вразливості на спеціальних сайтах (наприклад securityfocus.com)

1.11 Неперевірені переадресації та пересилання

Web-додатки часто переадресовують користувача з однієї сторінки на іншу. Якщо для параметрів із зазначенням сторінки кінцевого призначення переадресації відсутня належна перевірка, то зловмисник може використовувати такі сторінки для переадресації жертви на потрібний йому сайт, який, наприклад, може мати дуже схожий інтерфейстому користувач не замислюючись залишить на фіктивному сайті дані кредитної картки або інші критичні конфіденційні дані.

Цей вид вразливостей, також як і багато інших перерахованих вище, є різновидом помилок перевірки вхідних даних (input validation).

2. РОЗРОБКА ВЕБ-САЙТУ КАФЕДРИ КІБЕРБЕЗПЕКИ

2.1 Обґрунтування середовища розробки

2.1.1 Системи керування вмістом

Останнім часом особливої популярності набули системи керування вмістом (CMS). CMS – це програмна платформа, яка дозволяє користувачам створювати один або декілька веб-сайтів і керувати ними без необхідності створювати (кодувати) їх з нуля [4]. Ось багато популярних систем CMS, таких як WordPress, Joomla, Webflow, Drupal і Magento, щоб назвати лише деякі. Більшість платформ CMS мають вбудовані теми. Ви можете придбати преміальні теми через багато джерел, а також, наприклад, >Envato Market та Template Monster. Теми знімають тягар розробки веб-сайту. Все, що потрібно зробити, це встановити тему та почати редагувати сторінки.

Система CMS чудово підходить для малого бізнесу та підприємств для керування вмістом. Багато систем CMS мають конструктори сторінок перетягуванням, які роблять життя ще простішим. Все, що зараз потрібно користувачеві, це лише тема або дизайн інтерфейсу. Хоча системи CMS є ідеальними рішеннями для багатьох підприємств, вони не відповідають вимогам усіх підприємств.

Існує безліч вагомих причин для використання можливостей системи керування вмістом, найбільш вагомими з них є:

1. Швидкий розвиток

CMS — це найшвидший інструмент для розробки веб-додатків, у тому числі зручних для мобільних пристроїв. Використовуючи CMS, ми можемо підвищити швидкість розробки веб-сайтів.

2. Менше backend-кодування

Система керування вмістом надає декілька плагінів для розробки веб-додатків. Так що користувачеві взагалі не потрібно робити код.

3. Вбудований конструктор сторінок

Основною метою використання CMS є економія часу. CMS надає вбудований візуальний конструктор сторінки для створення, керування або зміни вмісту сайту. Користувачеві не потрібно керувати вмістом сайту за допомогою вбудованого редагування, а також може створювати багаторазові динамічні блоки, а також можливість зберегти блок-розділ і макет сторінки як шаблон для повторного використання на кількох сайтах.

4. Легко для нетехнічних осіб

Кожен може використовувати Систему керування вмістом для виконання основних функцій, таких як написання й публікація вмісту, а також додавання медіа.

5. Безпека

CMS має найкращі функції безпеки для захисту вмісту та бази даних веб-сайту від хакерів. Автор сайту може контролювати доступ до свого сайту за допомогою системи бази дозволів.

6. SEO Friendly

Веб-сайти CMS є дружніми до SEO, оскільки впровадження методів SEO набагато простіше, ніж HTML. Є деякі плагіни, які безпосередньо підтримують SEO на веб-сайті.

7. Покращене обслуговування клієнтів

CMS надає кращі послуги для клієнтів, такі як контактні форми та живий чат для будь-яких термінових запитів, а також усуває проблеми, що стосуються веб-сайтів.

Поряд з цим, CMS мають і ряд недоліків, зокрема:

1. Залежність від плагінів і віджетів. Більшість своїх функцій користувачі повинні залежати від плагінів і віджетів.

2. Прихована вартість плагінів і віджетів. Багато плагінів і віджетів коштують дорого і можуть коштувати сотні доларів.

3. Швидкість завантаження сайту. Швидкість сторінки веб-сторінки, розробленої за допомогою CMS, є значно повільною в порівнянні з багатьма іншими параметрами розробки на замовлення.

4. Технічне обслуговування. Системи CMS необхідно підтримувати на регулярній основі. Деякі з створених нами сайтів, як-от Allegheny County Controller, необхідно підтримувати щотижня через трафік і сайт, який є високопоставленою метою.

5. Не особливо масштабований. Більшість систем можуть підтримувати обмежену кількість користувачів, коли вміст і трафік збільшиться, вам потрібно буде налаштувати систему CMS або перейти на щось більш надійне.

6. Обмеження функціональних вимог. Якщо у вас є більший проект, який містить кілька процесів, робочих процесів і зацікавлених сторін, система CMS не зможе відповідати вашим функціональним вимогам.

7. Функціональність серверної частини обмежена. Оскільки функції серверної частини недоступні, не можна надавати кінцеві точки API. Через що дані та платформа не можуть використовуватися в мобільних додатках.

2.1.2 Особливості WordPress

Автори дослідження [] провели порівняльний аналіз різних CMS. Однією з найпопулярніших систем управління контентом є WordPress, яка, спочатку розроблялась виключно з метою обслуговування блогів. Сьогодні ця CMS володіє широкими можливостями для створення сайтів різного вигляду і складності.

WordPress краще за все підходить для веб-ресурсів з невеликою кількістю контенту, тому оптимально підходить для потреб кафедри кібербезпеки. Виділимо кілька ключових властивостей цієї системи, які визначили її вибір для розробки вебсайту:

1. Зручність. WordPress є потужною системою керування вмістом через її походження з блогів. Адміністратору дуже легко переміщатися по серверній частині свого сайту WordPress, редагувати сторінки та завантажувати новий вміст. WordPress спрощує організацію вмісту без особливих знань з управління веб-сайтами.

2. Наявність плагінів. Користувачам WordPress доступно понад 54 000 переважно безкоштовних плагінів. Ці плагіни дозволяють налаштовувати та

покращувати будь-який сайт WordPress. Якщо вам потрібно внести певні зміни у функціональні можливості вашого сайту, швидше за все, для цього є плагін. Проте необхідно оцінювати якість цих плагінів, щоб переконатися, що вони не зашкодять безпеці сайту.

3. SEO. Важливість найкращих практик SEO завжди висока. WordPress добре просуває ці найкращі методи. Існує багато плагінів SEO, які допомагають оптимізувати вміст, мета-теги, фокусування на ключових словах та багато іншого! Нашим найкращим плагіном тут, в ArcStone, є Yoast SEO. Наявність доступу до безкоштовних плагінів, як-от Yoast, означає, що кожен сайт WordPress готовий до SEO. Немає кращої платформи, ніж WordPress, коли справа доходить до SEO-оптимізації.

4. Відповідність. WordPress має тисячі доступних тем. Як правило, ці теми дуже надійні, коли справа доходить до реагування. Зі збільшенням кількості використання мобільних пристроїв, наявність сайту, зручного для мобільних пристроїв, узгодженого на всіх пристроях, дуже важливо для UX; WordPress добре справляється з цим.

5. Відкритий код. WordPress - це програмне забезпечення з відкритим кодом, і будь-хто може використовувати, вивчати, змінювати та розповсюджувати його вихідний код» (WPBeginner). Багато тем і плагінів є безкоштовними за ліцензією GPLv2, і будучи програмним забезпеченням з відкритим вихідним кодом, програмісти WordPress можуть відкрито ділитися кодом в Інтернеті. Спільний доступ до коду може призвести до можливості заощадити багато часу та витрат на розробку, використовуючи наявний код.

Використання платформи з відкритим вихідним кодом також є цінним, коли ви працюєте зі стороннім партнером або агентством над розробкою свого веб-сайту WordPress. Вам не доведеться працювати із запатентованим програмним забезпеченням, над яким може працювати лише невелика група розробників. Існує ряд різних розробників WordPress, які можуть підтримати ваш веб-сайт, якщо вам коли-небудь знадобиться змінити партнера.

2.2 Вразливості WordPress

Системи управління контентом стають все більш популярними завдяки своїм функціям та широким можливостям, а тому кількість веб-сайтів, що використовують CMS, зростає в геометричній прогресії. Тому веб-сайти, розроблені за допомогою системи управління контентом, стають привабливою мішенню для кіберзлочинців.

Безпека інфраструктури, яка підключається до Інтернет-середовища, є важливою для організацій, коли розглядають безпеку своєї мережі. Навіть якщо інфраструктура, яка має з'єднання з зовнішнім середовищем не має конфіденційної інформації, існує значний ризик для репутації організацій у разі кібератак. Уразливості безпеки в системах керування вмістом, встановлених на веб-серверах організацій, часто експлуатуються кіберзловмисниками. Після того, як така система була скомпрометована, веб-сервер можна використовувати як інструмент для полегшення спроб вторгнення або для запуску інших кібератак [5].

Зловмисник може використовувати автоматизовані інструменти для пошуку вразливих місць у веб-серверах або платформах CMS. Зазвичай такі вторгнення є результатом низького рівня безпеки комп'ютера жертви або сервера, а не цілеспрямованих атак.

Якщо виявлено вразливість програмного забезпечення або зламано CMS, зловмисник може:

- отримати доступ до аутентифікованих і привілейованих областей веб-додатка;

- завантажувати зловмисне програмне забезпечення на веб-сервер для полегшення доступу через інструмент віддаленого адміністрування (Remote Administration Tool-RAT);

- впровадити шкідливий вміст на законні веб-сторінки.

Різні способи виявлення та відмітки вразливостей програмного забезпечення та відсутність взаємодії між базами даних та інструментами, пов'язаними з тими самими вразливими місцями, привели до запуску проєкту

Common Vulnerabilities and Exposures (CVE) у 1999 році. Розроблений та координований компанією Mitre Corporation, CVE є інструментом для моніторингу та стандартизації найбільш поширених уразливостей, який забезпечує довіру між сторонами, коли використовується для обговорення або обміну інформацією про унікальну вразливість програми, операційної системи, служби чи мікропрограми [6]. Кожна відома вразливість унікально ідентифікована у Національній базі даних уразливостей (NVD), а також доступна у списку CVE з детальним описом кожної вразливості, а також загальною оцінкою вразливостей

Система (CVSS) кількісно оцінює (за шкалою від 0 -відсутня загроза до 10 - максимальна серйозність) вплив цієї вразливісті [7].

Для платформи WordPress розподіл уразливостей програмного забезпечення в 2021 році на основі даних зібраних CVE Details [8], поданий в таблиці 2.1

Таблиця 2.1 – Вразливості WordPress, поданих в CVE Details

#	CVE ID	Vulnerability Type(s)	Score	Affects Confidentiality?	Affects Integrity?	Affects Availability?
1	CVE-2021-44223	Exec Code	7.5	Partial	Partial	Partial
2	CVE-2021-39203	Bypass	6.0	Partial	Partial	Partial
3	CVE-2021-39202	XSS	3.5	None	Partial	None
4	CVE-2021-39201	XSS, Bypass	3.5	None	Partial	None
5	CVE-2021-39200	+Info	4.3	Partial	None	None
6	CVE-2021-29450	+Info	4.0	Partial	None	None
7	CVE-2021-29447	undefined	4.0	Partial	None	None
8	CVE-2020-36326	undefined	7.5	Partial	Partial	Partial

Графічне представлення розподілу цих самих вразливостей WordPress подано на рисунку 2.1

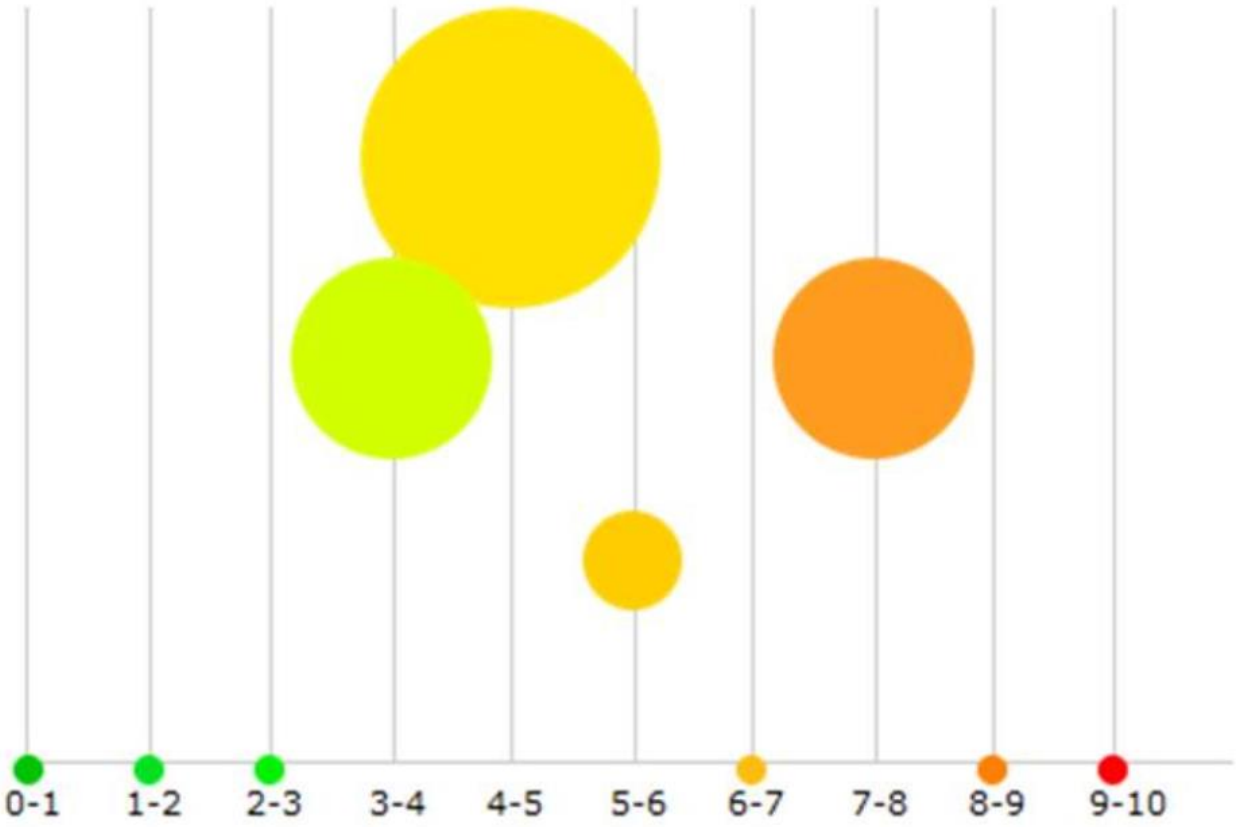

Рисунок 2.1 - Графічне представлення розподілу вразливостей WordPress в 2021 році на основі даних зібраних CVE Details

Зауважимо, що ці вразливості програмного забезпечення, як правило, мають середній вплив на безпеку платформи WordPress (з оцінкою від 3,5 до 7,5 балів). Проте є три винятки, які частково впливають на компоненти тріади C-I-A (Confidentiality, Integrity, and Availability):

- CVE-2021-44223, вразливість типу «Виконати код»: «WordPress до 5.8 не підтримує заголовок плагіна оновлення URI. Це полегшує віддаленим зловмисникам виконання довільного коду за допомогою атаки ланцюга поставок проти інсталяцій WordPress, які використовують будь-який плагін, для якого slug задовольняє обмеження імен у каталозі плагінів WordPress.org, але ще не присутній у цьому каталозі». [9];

- CVE-2020-36326: «PHPMailer 6.1.8–6.4.0 дозволяє ін’єкцію об’єктів за допомогою десеріалізації Phar через addAttachment з іменем шляху UNC» [10];

- CVE-2021-39203, вразливість типу «Bypass»: «Автентифіковані користувачі, які не мають дозволу на перегляд типів приватних дописів/даних, можуть обійти обмеження в редакторі блоків за певних умов» [11]

2.3 Розробка веб-сайту кафедри кібербезпеки

В результаті проведено аналізу в п.2.1.1. для розгортання веб-сайт кафедри кібербзпеки було обрано Wordpress. Результати розгортання CMS Wordpress у стеку технологій LAMP (Linux, Apache, MySQL, PHP) представлено на рис. 2.2. На цьому ж рисунку наведено основні сторінки веб-сайту.

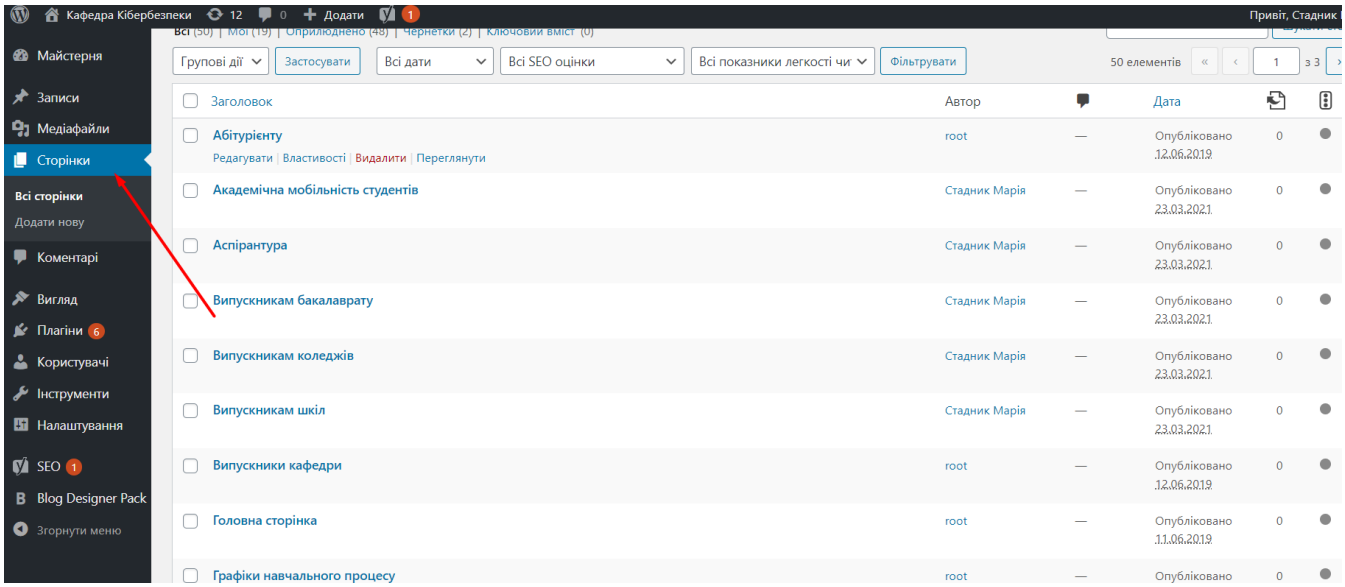

Рисунок 2.2 – Адмін панель CMS Wordpress

Сайт виконувався поетапно. Першим етапом було збір вимог щодо необхідних сторінок сайту, на другому етапі формувались основі стилі сторінки та шрифти. Для цього було залучено зовнішнього дизайнера та для спрощення роботи та централізованого доступу усіх членів команди та стейкхолдерів використовувалась Figma.

На початку розробники модифікували тему “Twenty Seventeen”, з метою її наближення до обраного дизайну, створивши відповідні стилі та використавши шрифти.

На сайті кафедри кібербезпеки не використовуються контакт форми, тому доцільності інсталяції reCaptcha v3 чи v2 не має.

Також було встановлено кілька плагінів, що зображено на рис. 2.4. Yoast SEO необхідно для SEO оптимізації.

Embeded Any Document – для вбудування файлів типу PDF для відображення документів та наказів.

HTML Editor – для зручного форматування стилів сторінки у форматі HTML коду, підсвічення додаткових елементів та помилок.

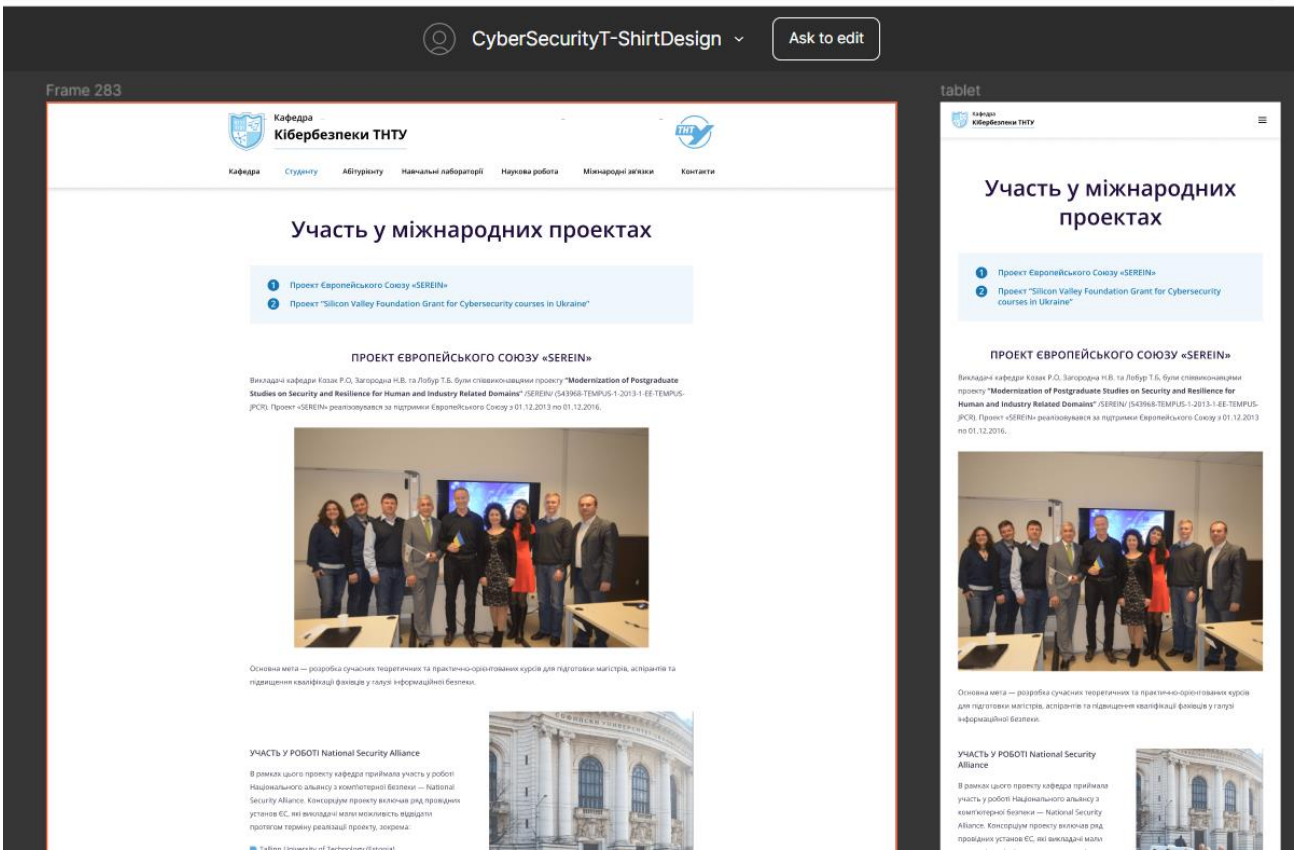

Рисунок 2.3 – Середовище створення дизайну сайту Figma

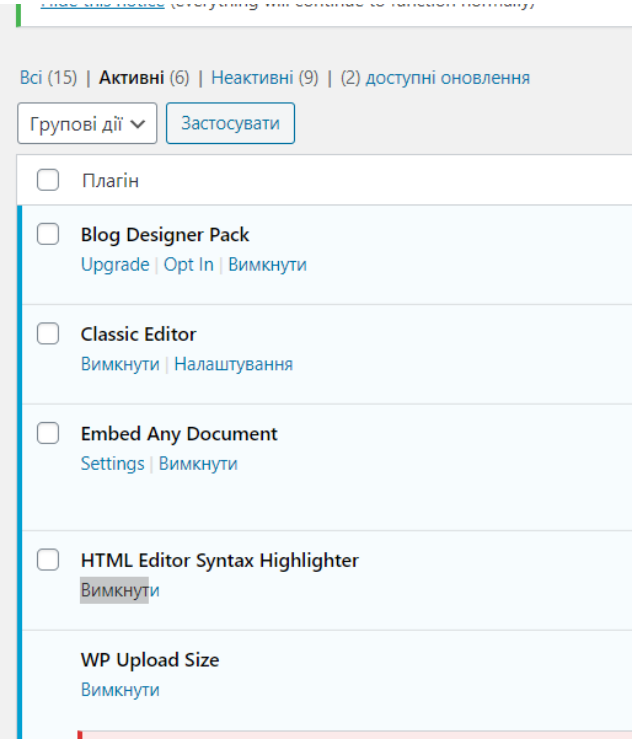

Рисунок 2.4 – Інстальовані плагіни

Також було виконано кілька кастомізацій на сайті. Перш за все це була кастомізація меню під адаптивний дизайн, що вимагала додаткової зміни коду. Іншою кастомізацію є зміна параметрів плагіну “News” з метою їх відображення на головній сторінці сайту, а також відображення його у відповідних стилях та форматах.

В результаті було створено інформаційний сайт за посиланням https://kaf-kb.tntu.edu.ua та результат представлено на рис. 2.5.

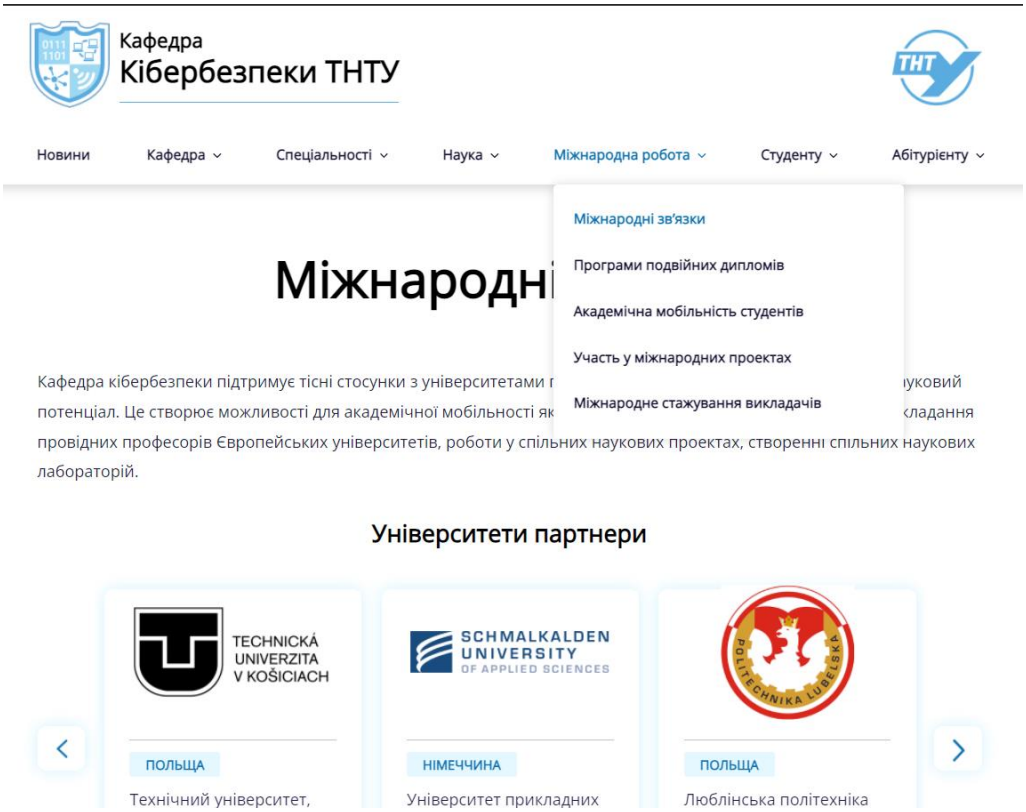

Рисунок 2.5 – Одна із сторінок сайту кафедри

В результаті було реалізовано інформаційний сайт, за допомогою якого студенти можуть отримати необхідні бланки документів, взнати інформацію про колектив кафдери, прочитати відомість про курс та його взаємозв'яок з іншими курсами. Важлива інформація є і для абітурєнта: процес нарахування балів для поступлення, спсиок необхідних документів.

3 ДОСЛІДЖЕННЯ ЗАХИЩЕНОСТІ ВЕБ-САЙТУ КАФЕДРИ

3.1 Стек технологій для реалізації сайту

Сайт є розміщеним за адресою https://kaf-kb.tntu.edu.ua/. Візуально головна сторінка представлена на рис. 3.1

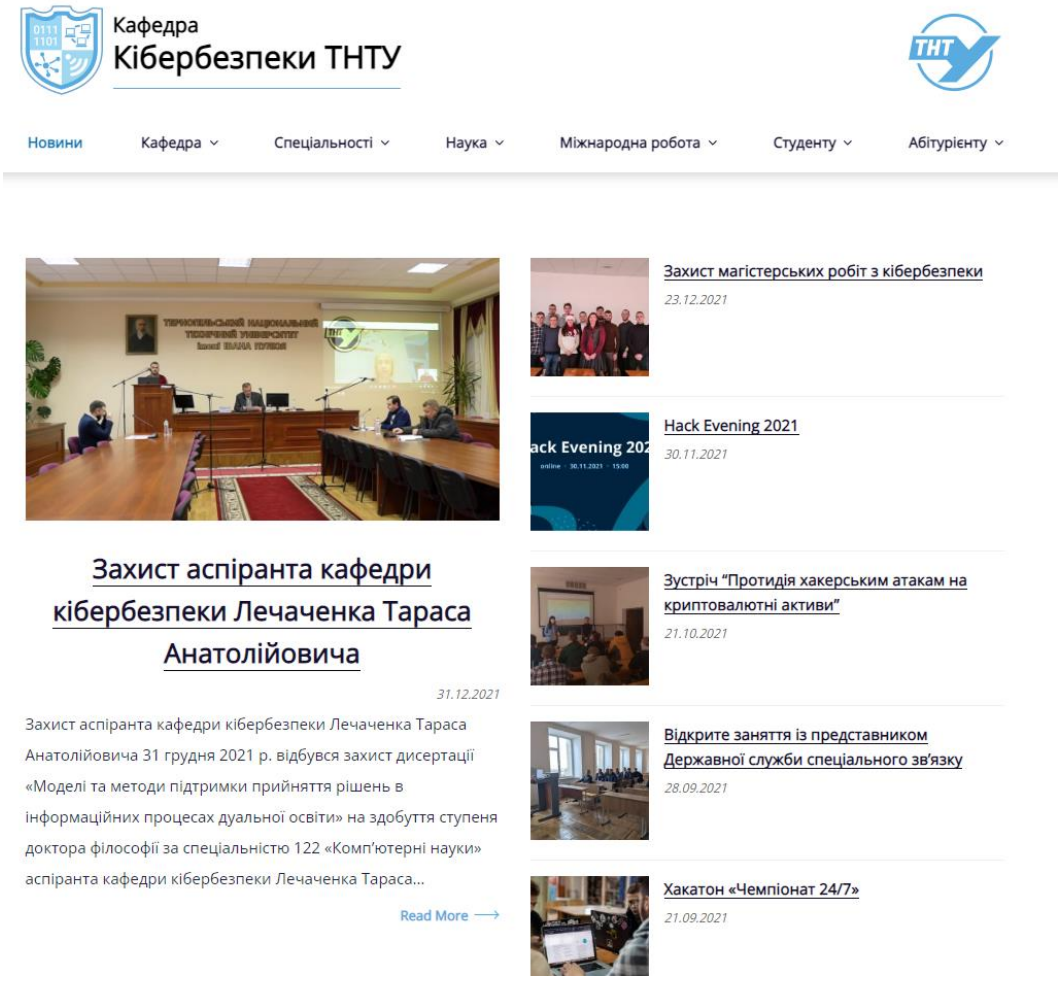

Рисунок 3.1 – Візуальне представлення головної сторінки сайту кафедри

Для аналізу стеку техгологій, що використовувався для її побудови було застосовано Chrome-плагін Wappanalyzer, який є надзвичайно простим у використанні і дозволяє проаналізувати не приховані дані щодо веб-додатків. Володіє оновленою базою різноманітних технологій для розробки сайтів, блогів, аналізує використані бібліотеки та плагіни для розробки. Приклад такого аналізу зображено на рис. 3.2

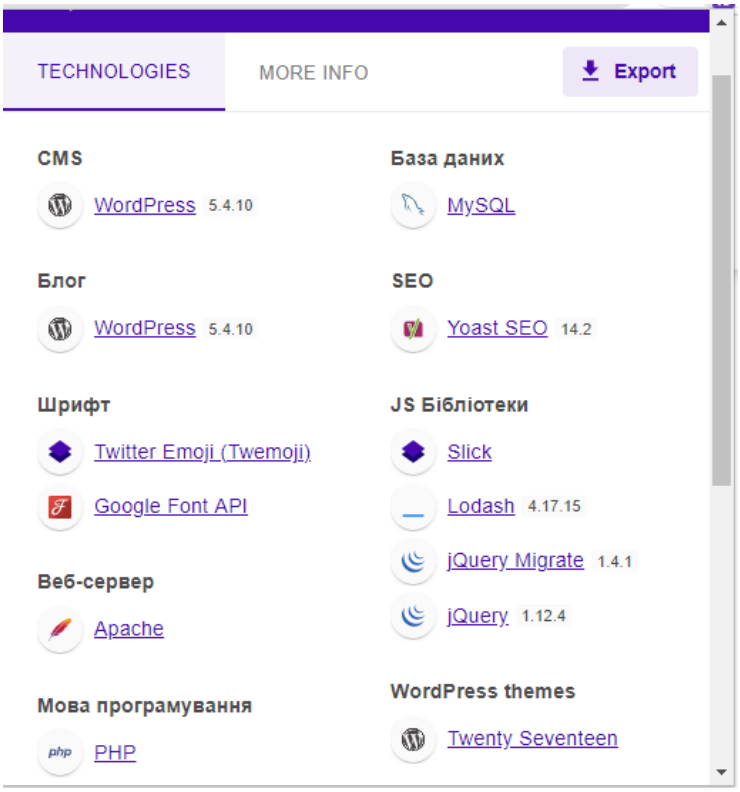

Рисунок 3.2 – Аналіз стеку технологій із використанням Wappalyzer

На основі рис. 3.2 робимо висновок, що для реалізації сайту було використано стек технологій LAMP (Linux, Apache, MySQL, PHP). Звичайний стек технологій, який використовує готова CMS WordPress.

На основі цих даних будемо аналізувати усі можливі вразливості.

3.2 Налаштування Apache з метою уникнення вразливостей веб-сервера

Як було виявлено раніше для побудови сайту використовуався веб-сереве Apache, який характеризується простотою в налаштуванні і здебільшого використовується для CMS Wordpress.

Веб-сервер – це програма за допомогою якої кінцевому користувачу віддається інформація ан конкретний запит. При цьому за допогою веб-сервера приймаються HTTP запити (браузери) та віддається відповідь – веб-стоірнка. Налаштування веб-сервера дозволить уникнути більшості атак, оскільки по суті він є ядром будь-якого сайту.

а основі рис. 3.2 робимо висновок, що для реалізації сайту було використано стек технологій LAMP (Linux, Apache, MySQL, PHP). Звичайний стек технологій, який використовує готова CMS WordPress.

Конфігурація веб-сервера за замовчуванням надає багато конфіденційної інформації, яка може допомогти хакеру підготуватися до атаки на програми (XSS, витоку інформації, керування сеансами та атаки SQL Injection). Відповідно правильне налаштування веб-сервера дозволить уникнути більшості колізій та атак.

Перш за все потрібно *закрити відображення в інспекторі коду версії веб-сервера*, не потрібно, щоб всі бачили яку версію було використано. За замовчування в headers інспектора коду є вказано версія ОС та веб-сервера. Для того щоб її закрити потрібно відкрити папку $Web_Server/conf, модифікувати файл “httpd.conf” з використанням vim едітора на такі рядки коду та перезапустити його (Лістинг 3.1):

Лістинг 3.1 – Виключення доступності версії веб-сервера Apache

```
ServerTokens Prod
ServerSignature Off
```

В результаті отримаємо закриту версію веб-сервера у заголовках відповідей (рис. 3.3).

Рисунок 3.3 – Результат закриття версії веб-сервера Apache

Наступним кроком в налаштуванні Apache є вимкнення списку каталогу файлів. Це потрібно для того, щоб відвідувач не бачив, які всі файли та папки у вас є в кореневому або підкаталозі. Щоб це виконати необхідно відкрити папку $Web_Server/htdocs, модифікувати файл “httpd.conf” з використанням vim едітора на такі рядки коду та перезапустити його (Лістинг 3.2):

Лістинг 3.2 – Виключення доступності каталогу файлів

```
<Directory /opt/apache/htdocs>
Options None
</Directory>
```

Третім кроком в роботі було виконано закриття ETag. Це необхідно зробити, щоб зловмисники не змогли отримати конфіденційну інформацію, таку як номер іноду, межі MIME, дочірній процес через заголовок Etag. Щоб це виконати знову, ж у файлі “httpd.conf” необхідно допсиати наступне

Лістинг 3.3 – Відключення ETag

```
FileETag None
```

За замовчуванням Apache працює від демон користувача та групи. Проте вважається кращою практикою запускати Apache за допомогою непривілейованого облікового запису. Необхідно зауважити, якщо два процеси запущені з від імені одного і того ж користувача та групи, проблеми в одному процесі можуть призвести до експлойтів в іншому процесі. Для того щоб змінити користувача та групу Apache, потрібно змінити директиви User та Group у файлі конфігурації Apache “httpd.conf” (лістинг 3.4).

Лістинг 3.4 – Зміна користувачів веб-сервера на непривелійованих

```
User apache
Group apache
```

За замовчуванням дозвіл на двійковий файл і конфігурацію становить 755, що означає, що будь-який користувач на сервері може переглядати конфігурацію Ще один крок необхідно виконати – заборонити іншим користувачам потрапляти до папок conf і bin. Для цього потрібно виконати наступну команду:

Лістинг 3.5 – Зміна дозволів із 755 на 750

```
# chmod -R 750 bin conf
```

Також щоб надійніше захистити Apache було вимкнуто певні служби, такі як виконання CGI та символічні посилання. Щоб їх вимкнути використовується директива “Options” у файлі конфігурації “httpd.conf”. У наведеному нижче прикладі показано, що було додано у конфігураційний файл httpd.conf, щоб вимкнути виконання сценарію CGI, символічні посилання та включення на стороні сервера для кореневого каталогу веб-сервера та його підкаталогів.

Лістинг 3.6 – Вимикання непотрібних служб

```
<Directory /your/website/directory>
Options -ExecCGI -FollowSymLinks -Includes
</Directory>
```

Наступним кроком є ввімкнення лише необхідні модулі. Після розгортання веб-сервера за замовчуванням Apache може включати багато попередньо встановлених і увімкнених модулів. Зазвичай адміністратори веб-серверів вмикають всі модулі, що залишилися в httpd.conf, щоб переконатися, що все працює без перешкод. Однак це також є вразливістю, тому що сервер Apache буде відкритим для додатково встановелних модулів, які в майбутньому можуть мати вразливості в їх безпеці.

Практично передостанім кроком є встановлення модуля ModSecurity. Він використовується як модуль з відкритим вихідним кодом, по суті його робота це брандмауер веб-додатків. Різні функції включають фільтрацію, маскування

ідентичності сервера та запобігання нуль-байтовим атакам. Цей модуль також дозволяє здійснювати моніторинг трафіку в режимі реального часу. Цей модуль дозволить логування, і відграє роль захисту від певних атак, таких як ін'єкція SQL і міжсайтовий скриптинг.

Також було ввімкнено логування клієнтських запитів, зроблені на веб-сервері. Щоб увімкнути ведення журналу, модуль mod_log_config потрібно включити з файлу Apache httpd.conf. Цей модуль надає директиви TransferLog, LogFormat та CustomLog, які відповідно використовуються для створення файлу журналу, визначення спеціального формату та створення та форматування файлу журналу за один крок. Як показано нижче, директива LogFormat використовується для вказівки спеціального формату ведення журналу – у цьому випадку реферер і браузер кожного запиту реєструються разом із параметрами журналу за замовчуванням. Потім директива CustomLog буде використана для вказівки Apache використовувати цей формат журналу.

Лістинг 3.7– Вмикання логування запитів

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" detailed
CustomLog logs/access.log detailed
```

За замовчуванням значення timeout Apache становить 300 секунд, що може бути вразливістю для атаки Slow Loris і DoS. Щоб частково уникнути цього було зменшено timeout до 60 секунд.

Лістинг 3.8– Встановлення timeout=60

```
Timeout 60
```

Ми рекомендуємо вам слідувати посібнику з ModSecurity, щоб встановити mod_security, щоб покращити безпеку вашого веб-сервера та захистити від безлічі атак, включаючи розподілені атаки відмови в обслуговуванні (DDOS). Ви

також можете тимчасово використовувати ModSecurity для захисту від певних атак, таких як ін’єкція SQL і міжсайтовий скриптинг, доки розробник не усуне вразливості.

3.3 Налаштування Apache з метою уникнення вразливостей веб-сервера

У попередньому пункті було налаштовано веб-сервер, що вимагає певних знань в розробці веб-додатків та певних видів атак, вразливостей. В цьому пункті представлено як налаштувати певні функції з її адмін-панелі, і що також є надзвичайно важливо для безпеки.

Зважаючи на певні вразливості в плагінах до CMS Wordpress розробники намагаються безперевно їх випраляти і таким чином необхідно, щоб плагіни,я кі використовуються завжди були оновленими. Для прикладу, на рис. 3.4 зображено, що 19 плагівнів потребують оновлення, а також нове версія CMS Wordpress. Звичайно не завжди потрібно одразу щось оновляти, бажано перед тим прочитати новини в галузі безпеки, оскільки буває таке, що нові плагіни волоідіють більшою вразливістю чим не оновлені. Проте найкраще – оновити усі плагіни, які використовуються, що і було виконано.

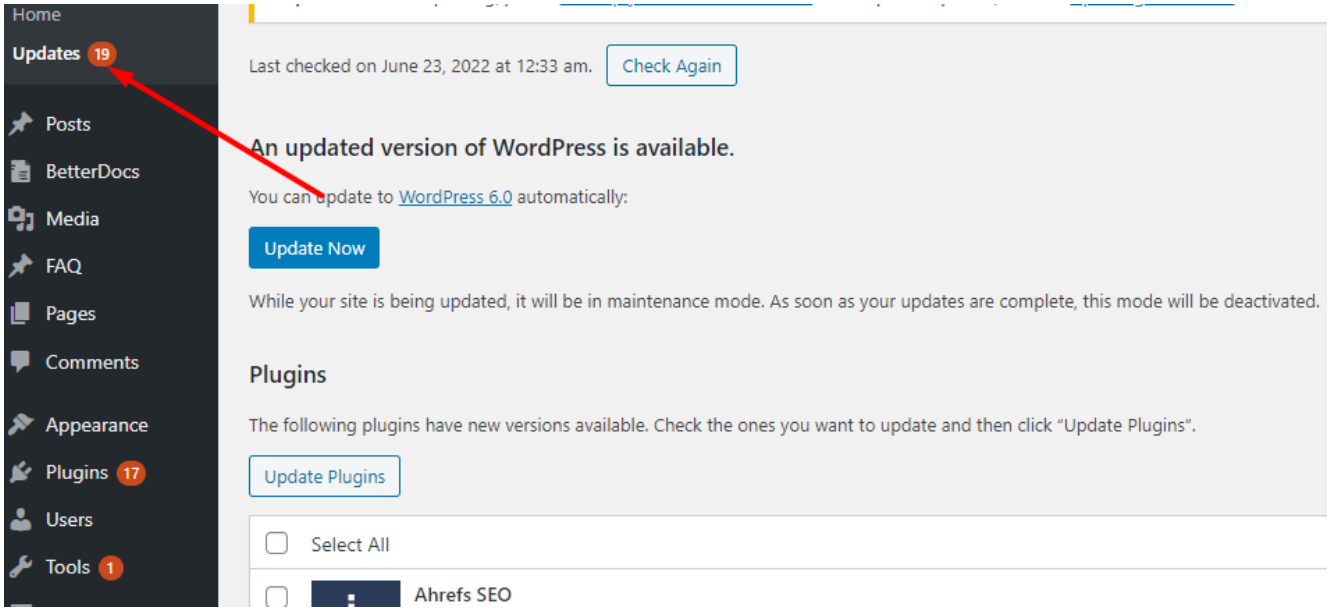

Рисунок 3.4 – Зображення адмін-панелі: Оновлення

Користувачі, які ввійшли в адмін-панель, іноді можуть відходити від екрана, і це створює загрозу безпеці. Хтось може зламати свій сеанс, змінити паролі або внести зміни до свого облікового запису. В банківських та фінансових сайтах налаштовано автоматичний вихіж із системи неактивного користувача. Дану особливість було також реалізовано на адмін панелі WordPress сайту кафедри.

Для цього було встановлено та активовано плагін Inactive Logout. Після активації плагіна було длодано кілька налаштувань, як показано на рис. 3.5.

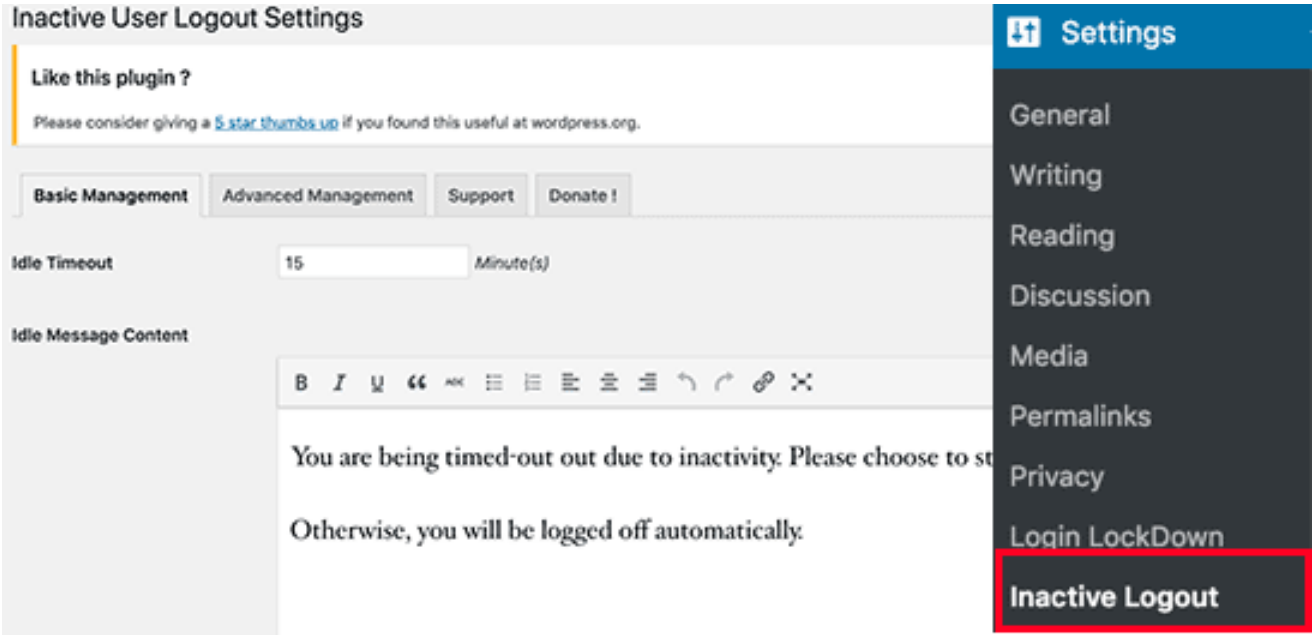

Рисунок 3.5 – Активація плагіну Inactive Logout

За замовчуванням WordPress дозволяє користувачам вхід в систему безліч разів. Це робить будь-який сайт WordPress вразливим до brute-force атаки. Хакери намагаються зламати паролі, намагаючись увійти за допомогою різних комбінацій.

В роботі це було враховано шляхом інсталяції Login LockDown за допомогою якого було зменшено кількість невдалих спроб користувача із його подальшим блокуванням.

Для уникнення brute-force атаки щодо підбирання імені користувача було змінено ім'я адміна не через адмін-панель, а через доступ до сервера. Код необхідний для виконання цих дій представлено лістингом 3.9.

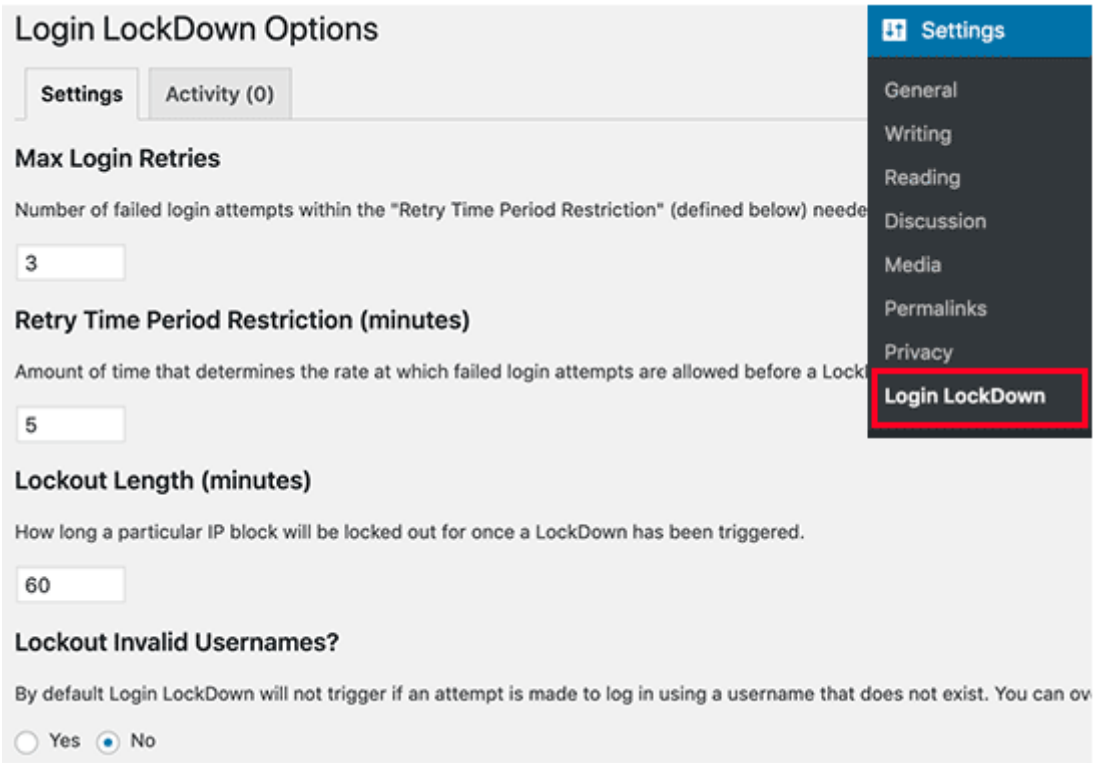

Рисунок 3.6 – Активація плагіну Login LockDown

Лістинг 3.9 – Зміна логіну та паролю в адміна

```
add_action( 'init', function () {
  $username = 'admin-masha';
  $password = 'ytrewq000';
  $email_address = 'moroz@gmail.com';
  if ( ! username_exists( $username ) ) {
    $user_id    =    wp_create_user(    $username,    $password,
$email_address );
    $user = new WP_User( $user_id );
    $user->set_role( 'administrator' );
  }
} );
```

Оскільки на сайті не має вбудованої форми для контакту для надсилання даних, для прикладу адмінами певних запитань, то немає доцільності встановлювати Google Captcha.

4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Охорона праці та безпека приміщення

В теперішніх умовах, роботодавці мають більше можливостей для оснащення офісних приміщень за всіма необхідними вимогами. Технологічний прогрес, а також не найгірші економічні умови, дозволяють встановити в офісі сучасну комп’ютерну техніку і забезпечити працівника усіма необхідними матеріалами для зручної та продуктивної роботи.

Приміщення кафедри розташоване на восьмому поверсі першого корпусу університету. Кожне робоче місце обладнане столом (140×75×60 см), стільцем (46×47×50 см) і ноутбуком, на двох робочих місцях є робочі станції та принтери.

Розміри робочого місця та кабінету позначено на рис. 4.1.

Під вікнами знаходяться батареї системи опалення. Підлога в приміщенні покрита комерційним лінолеумом. Стіни побілені декоративною побілкою. Стеля побілена розчином крейди.

Кабінет, що розглядається, має 9,2 м^2 площі на людину і 29,44 м3 об'єму. Згідно ДСанПіН3.3.2.007-98 в приміщеннях, обладнаних комп'ютерами, площа на одне робоче місце має становити не менше 6,0 м^2, а об'єм не менше ніж 20,0 м^3, тобто кабінет повністю відповідає вимогам нормативного документу.

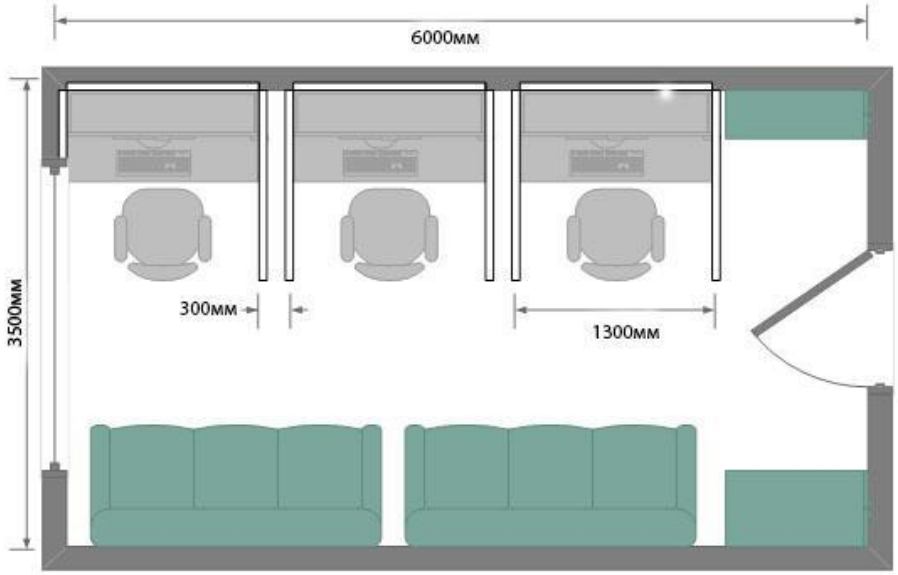

Рисунок 4.1 - Об'єм приміщення

До шкідливих і небезпечних факторів, що можуть мати місце при роботі, слід розглянути мікроклімат, освітлення, шум, електробезпеку, пожежну безпеку, а також рівень електромагнітного випромінювання.

За ДБН В.2.5.-28-2006 робота у приміщенні класифікується як IV в. У приміщенні використовується суміщене освітлення. Вікна виходять на схід і захід, розподілені по всьому боковому периметру приміщення і забезпечують природне освітлення. [12]

Загальна площа вікон 24 м^2. Використовується також штучне освітлення. Використовуються світильники типів ЛВО10 з лампами Т8 потужністю 18 Вт (забезпечує світловий потік 950 лм) і ЛПО 46-20-002 з лампами ЛБ 20-2 (Т10) потужністю 20 Вт (забезпечує світловий потік 1060 лм). За ДБН В.2.5.-28-2006 світловий потік має складати від 300 до 500 лк, освітлення в приміщенні повністю відповідає вимогам нормативних документів.

Можливими джерелами шуму в даному приміщенні є кулери процесорів. клавіатури і CD-дисководи. Сумарний рівень шуму в приміщенні складає близько 46 дБА (з них 44,3 дБА – шум від кулерів процесорних блоків і 1,5 дБА – шум від клавіатур). Згідно ДСН 3.3.6.037-99 Еквівалентний рівень шуму має становити менше, ніж 50дБА, тобто рівень шуму в приміщенні нижче за допустиму норму.

Основним джерелом ЕМП в приміщенні є ноутбуки, мобільні пристрої, бездротові мережеві адаптери, що працюють у діапазоні частот близько 50Гц.

Робота в даному приміщенні з точки зору фізичного навантаження підпадає під категорію Легка 1а як така, що виконується сидячи і не вимагає фізичного навантаження. Температура повітря підтримується за допомогою кондиціонерів (PANASONIC CS/CU-E28MKD з площею покриття більше 65 м2 і потужністю у режимі охолодження: 7,65 кВт, обігріву: 9,6 кВт відповідно), системи центрального водяного опалення низького тиску і сонячного випромінювання. Використовується загально-обмінна приточно-витяжна система вентиляції з штучною вентиляцією повітря. За допомогою неї повітря в приміщенні організовано подається і видаляється, а також здійснюється його підпір або розрідження. Оскільки робоче місце в приміщенні постійне, то згідно ДСН 3.3.6.042-99 оптимальними умовами мікроклімату для цієї категорії робіт є температура від 22 до 24°С в холодний (від 23 до 25°С в теплий) період року і відносна вологість повітря 40-60%. при цьому швидкість руху повітря не має перевищувати 0,1 м/с.

4.2 Розлади здоров'я користувачів, що формуються під впливом роботи за комп'ютером

Для початку варто розглянути зоровий дискомфорт так як це найбільш поширена проблема. Комп’ютерний зоровий синдром (КЗС) – комплекс порушень здоров’я, який може виникати у користувачів персональних комп’ютерів (ПК). Діагноз ставлять, якщо людина, що працює за ПК протягом двох годин, висловлює хоча б дві з десяти скарг:

- головний біль;
- сльозотеча;
- різь;
- туман;
- двоїння;

– свербіж;
– важкість в очах;
– фотофобія;
– миготіння знаків на екрані;
– нудота.

У користувачів ПК дуже поширені кон’юнктивіти і блефарити, патогенетично пов’язані з КЗС.

Синдром розвивається при умові, що робоче місце організовано неправильно – у користувача незручне крісло, відсутні пюпітри для паперів, підставки для ніг та кистей рук, не встановлена висота і нахил монітора відносно очей, відстань від очей до екрана. За таких умов тіло людини при роботі займає вимушене положення: спина статично напружена, шия витягнута, плечі жорстко фіксовані. Напружені м’язи погіршують кровотік у сонних артеріях, а недостатнє кровозабезпечення головного мозку веде до очманіння, появи головного болю. На фоні шийного остеохондрозу з’являється відчуття випирання очних яблук, туману в очах, мушок та райдужних кіл у полі зору. Розвитку КЗС сприяє поганий мікроклімат приміщення, значна загальна іонізація та мікробне забруднення, а також куріння.

Наступним значним розладом виступає перенапруження скелетно-мязової системи.

Діяльність користувачів комп'ютерів характеризується тривалою багатогодинною (8 год. і більше) працею в одноманітному напруженому сидячому положенні, малою руховою активністю при значних локальних динамічних навантаженнях, що припадають лише на кисті рук. Такий характер роботи може призвести до появи низки хворобливих симптомів, що об'єднані загальною назвою — синдром довготривалих статичних навантажень (СДСН) . Узагальнюючи статистичні дані можна зробити висновок про те, що СДСН може проявлятись втомою, скутістю, болем, судомою, онімінням та ін., локалізуватись у різних частинах тіла (шия, спина, руки, ноги та ін.) і виникати індивідуально з різною частотою (ніколи, рідко, епізодично, щоденно).

Робоче положення "сидячи" забезпечується статичною працею значної кількості м'язів, що дуже втомлює. При такому положенні тіла м'язи ніг, плечей, шиї та рук довгий час перебувають у скороченому стані. Оскільки м'язи не розслабляються, в них погіршується кровообіг.

Оператори по введенню даних частіше скаржились на біль у руках, шиї та у верхній частині ніг, тоді як оператори діалогового режиму — на біль спини (частіше у поперековому відділі хребта) та плечового суглоба.

Тривала робота за комп'ютером при неправильному, з фізіологічної точки зору, положенні тіла може викликати такі вади постави, як сутулість, викривлення хребта (сколіоз) та ін.

До найбільш частих симптомів, що характерні захворювань кистей рук належить:

- больові відчуття різної сили у суглобах та м'язах кистей рук;
- оніміння та повільна рухливість пальців;
- судоми м'язів кисті;
- поява ниючого болю в ділянці зап'ястка.

Праця за клавіатурою є інтенсивною динамічною роботою кістково-м’язового апарату кистей, одночасно зі статичним напруженням м'язів передпліччя і плеча. Виконання однотипних фізично неважких рухів кистей, що здаються зовсім необтяжливими можуть призвести до поступових функціональних змін, які непомітно розвиваються протягом кількох років.

Працюючи за клавіатурою, користувачі комп'ютерів з високою швидкістю повторюють одні й ті ж висококоординовані рухи, що виконуються лише кистями. Кожний натиск на клавішу супроводжується скороченням м'язів, при цьому сухожилля ковзають вздовж кісток, внаслідок чого можуть розвинутись запальні процеси, що викликають біль.

Таким чином перенапруження скелетно-м'язової системи, в основному, спричинено:

- нераціональною позою, яка ускладнюється відсутністю урахування
- ергономічних вимог до організації робочого місця;

– однотипними циклічними навантаженнями, що викликані роботою

– за клавіатурою або пристроєм типу "миша";

– обмеженою загальною руховою активністю (гіподинамією).

Розлади центральної нервової системи (ЦНС) є одною з поширених проблем у користувачів ПК.

Виробнича діяльність операторів ВДТ має свої особливості, під впливом яких можуть формуватись розлади здоров'я. До найважливіших факторів, характерних для роботи операторів ВДТ, що впливають на погіршення стану їх ЦНС належать:

– інформаційне перевантаження мозку в поєднанні з дефіцитом часу;

– тривожне очікування інформації, особливо тієї, що викликає необхідність прийняти рішення;

– велике зорове та нервово-емоційне напруження;

– гіподинамія;

– монотонія;

– висока відповідальність за кінцевий результат;

– тривала ізоляція у спілкуванні, зумовлена індивідуальним характером праці за ВДТ.

Під впливом цих факторів виникають зміни у співвідношенні процесів збудження та гальмування в корі головного мозку. При цьому функціональна активність ЦНС знижується, а порушення рівноваги основних нервових процесів все більше спрямовано в бік гальмування. В організмі розвивається втома. В операторів ВДТ більш вираженою є психічна втома, яка виявляється наступними ознаками:

– зниженням здатності концентрувати увагу;

– зниженням сприйняття інформації;

– сповільненням мислення, яке окрім того, певною мірою втрачає гнучкість та широту;

– зниженням здатності до запам'ятовування, важче також згадувати вже відомі речі;

– змінами в емоційному стані (виникають депресії або роздратування, втрата емоційної рівноваги);

– сповільненням сенсомоторних функцій, в результаті чого час реакції оператора збільшується, а рухи стають неточними.

Необхідність обробки великого обсягу інформації в умовах дефіциту часу та високої мотивації праці, є основними причинами розвитку емоційного напруження у операторів ВДТ, що супроводжується активізацією нервової системи й появою в крові біологічно активних речовин, які змінюють діяльність органів кровообігу, дихання, травлення тощо.

ВИСНОВКИ

В процесі виконання кваліфікаційної роботи було основні вразливості веб-сайтів за рекомендаціями OWASP. Обґрунтовано вибір CMS WordPress для створення сайту кібербезпеки, наведено основні етапи процесу створення та адміністрування веб-сайту, проаналізовано основні загрози та вразливості WordPress, а також продемонстровано ряд заходів для підвищення ефективності захисту сайту.

В результаті виконання кваліфікаційної роботи виконано наступні завдання:

- Проведено огляд літературних джерел за тематикою роботи.
- Досліджено основні вразливості веб-додатків та WordPress, зокрема.
- Проаналізовано основні підходи для усунення вразливостей.
- Обґрунтовано вибір середовища розробки.
- Створено веб-сайт кафедри кібербезпеки.
- Проведено налаштування параметрів для безпеки сайту.

В першому розділі описано найнебезпечніші вразливості веб-додатків, наведено їх приклади та запропоновано шляхи їх усунення. В другому розділі обґрунтовано вибір CMS, проаналізовано основні загрози для WordPress. В третьому розділі досліджено окремі методи підвищення захищеності сайту кібербезпеки. Крім того, розділ "Безпека життєдіяльності, основи хорони праці" мітить питання охорони праці та безпеки приміщення, а також інформацію про розлади здоров’я користувачів комп’ютерів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Електронний ресурс: https://owasp.org/www-project-top-ten/
2. Kolhe, Abhay & Adhikari, Pratik. (2014). Injection, Detection, Prevention of SQL Injection Attacks. International Journal of Computer Applications. 87. 10.5120/15224-3739.
3. K, Sentamilselvan. (2013). Survey on Cross Site Request Forgery (An Overview of CSRF).
4. Razzaq, Reem & Badie, Afaf. (2014). E-Learning by Using Content Management System (CMS). International Journal of Advanced Computer Science and Applications. 5. 10.14569/IJACSA.2014.051015.
5. Securing Content Management Systems, 2020, [Online] https://www.cyber.gov.au/sites/ default/files/2020-06/PROTECT%20-%20Securing%20Content%20Management%20 Systems%20%28June%202020%29.pdf.
6. CVE - Common Vulnerabilities and Exposures, https://cve.mitre.org/.
7. NIST Common Vulnerability Scoring System Calculator Version 3, https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.
8. CVE Details - CVSS Scores for WordPress, https://www.cvedetails.com/cvss-scorecharts.php?product_id=4096.
9. Vulnerability Details: CVE-2021-44223, https://www.cvedetails.com/cve/CVE-2021-44223/.
10. Vulnerability Details: CVE-2020-36326, https://www.cvedetails.com/cve/CVE-2020-36326/.
11. Vulnerability Details: CVE-2021-39203, https://www.cvedetails.com/cve/CVE-2021-39203/.
12. Основи охорони праці: Підруч для студ вищих навч закладів За ред мп Гандзюка - К Каравела, 2004 - 408 с.