Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню магістр

студента *Дзюбелюка Владислава Олександровича*

академічної групи *125м-21-1*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Підвищення захищеності медіа-сайтів, створених із застосуванням мови Python, від несанкціонованого доступу*

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	к.т.н., доц. Ковальова Ю.В.			
розділів:				
спеціальний	к.т.н., доц. Ковальова Ю.В.			
економічний	к.е.н., доц. Пілова Д.П.			
Рецензент				
Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2022

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістр

студенту *Дзюбелюку Владиславу Олександровичу* **академічної групи** *125м-21-1*
(прізвище ім'я по-батькові) (шифр)

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

на тему ***Підвищення захищеності медіа-сайтів, створених із застосуванням мови Python, від несанкціонованого доступу***

затверджену наказом ректора НТУ «Дніпровська політехніка» від 31.10.22р. № 1200-с

Розділ	Зміст	Термін виконання
Розділ 1	Підвищення захищеності медіа-сайтів, створених із застосуванням мови програмування Python, на основі застосування запропонованих рекомендацій.	20.10.2022
Розділ 2	Проаналізувати проблеми захисту, основні вразливості медіа-сайтів, створених із застосуванням мови програмування Python, дослідити методи захисту від атак та розробити рекомендації із захисту таких сайтів від несанкціонованого доступу.	16.11.2022
Розділ 3	Визначити витрати на впровадження засобів захисту та налаштування автоматизованої системи згідно рекомендаціям із захисту сайтів.	05.12.2022

Завдання видано _____ _____
(підпис керівника) (прізвище, ініціали)

Дата видачі: 05.09.2022 р.

Дата подання до екзаменаційної комісії: 12.12.2022 р.

Прийнято до виконання _____ _____
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 111 с., 12 рис., 13 табл., 4 додатка, 28 джерел.

Об'єкт дослідження: процес проектування та вдосконалення медіа-сайтів, створених із застосуванням мови програмування Python.

Мета роботи: підвищення захищеності медіа-сайтів, створених із застосуванням мови програмування Python, на основі застосування запропонованих рекомендацій.

Методи дослідження: системний підхід, методи порівняння, аналізу, індукції, дедукції, аналогії.

У роботі проаналізовано проблеми захисту, основні вразливості медіа-сайтів, створених із застосуванням мови програмування Python, досліджено методи захисту від атак та розроблено рекомендації із захисту таких сайтів від несанкціонованого доступу.

В економічному розділі визначено витрати на впровадження засобів захисту та налаштування автоматизованої системи згідно рекомендаціям із захисту сайтів, що були дані в спеціальній частині.

Практичне значення роботи полягає у тому, що при використанні розробниками рекомендацій із захисту, даних в роботі, збільшиться захищеність медіа-сайтів, створених із застосуванням мови програмування Python, від несанкціонованого доступу, що зменшить матеріальні збитки власників сайтів.

Новизна дослідження полягає у тому, що вперше визначено тип медіа-сайтів, ідентифіковані вразливості та інші загрози для безпеки медіа-сайтів, створених із застосуванням мови програмування Python та розроблені рекомендації із захисту таких сайтів від несанкціонованого доступу.

МЕДІА-САЙТ, ВРАЗЛИВІСТЬ, АТАКА, БЕЗПЕКА, WEB-РОЗРОБКА, PYTHON.

ABSTRACT

Explanatory note: 111 p., 12 pic., 13 tabl., 4 app., 28 sources.

Object of study: the process of designing and improving media sites created using the Python programming language.

Purpose: to increase the security of media sites created using the Python programming language, based on the application of the proposed recommendations.

Research methods: systematic approach, methods of comparison, analysis, induction, deduction, analogy.

The paper analyzes the problems of protection, the main vulnerabilities of media sites created using the Python programming language, investigated methods of protection against attacks and developed recommendations for protecting such sites from unauthorized access.

In the economic section, the costs of implementing security measures and setting up an automated system according to the recommendations for protecting sites that were given in the special part are determined.

The practical significance of the work lies in the fact that when developers use the recommendations for protection given in the work, the security of media sites created using the Python programming language will increase from unauthorized access, which will reduce the material losses of site owners.

The novelty of the study is that for the first time the type of media sites is determined, vulnerabilities and other threats to the security of media sites created using the Python programming language are identified and recommendations for protecting such sites from unauthorized access are developed.

MEDIA SITE, VULNERABILITY, ATTACK, SECURITY, WEB DEVELOPMENT, PYTHON.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АС – автоматизована система;

АСУ – автоматизована система керування;

АСУ ТП – автоматизована система керування технологічним процесом;

ІЗоД – інформація з обмеженим доступом;

ІТ – інформаційні технології;

КЗЗ – комплекс засобів захисту;

ККД – коефіцієнт корисної дії;

ПЕОМ – персональна електронно-обчислювальна машина;

СКБД – система керування базами даних;

ТЗІ – технічний захист інформації;

IP – Internet Protocol;

CMS – системи управління контентом.

ЗМІСТ

с.

ВСТУП

Актуальність. В усьому світі зараз спостерігається тенденція падіння тиражів щоденних друкованих засобів інформації та зростання кількості звернень за новинами до Інтернет-видань та соціальних мереж.

Python – універсальна мова програмування, що дуже швидко набирає популярність та підтримує широкі можливості для створення та підтримки медіа-сайтів.

В останні роки кількість атак, націлених на вебсайти, значно зросла, тому що з'являються все нові варіанти і засоби для атак. Ця тенденція буде продовжуватися, тому розробникам треба більше уваги приділяти захисту вебдодатків, що вони створюють. А для цього треба розуміти, від чого треба захищати свої розробки.

Таким чином, актуальним науковим завданням, що має теоретичне і практичне значення, є ідентифікація вразливостей медіа-сайтів, створених із застосуванням мови програмування Python, та розробка рекомендацій із захисту таких сайтів від атак.

Метою роботи є підвищення захищеності медіа-сайтів, створених із застосуванням мови програмування Python на основі застосування запропонованих рекомендацій.

Для досягнення зазначеної мети роботи поставлені окремі завдання:

– провести аналіз загроз та вразливостей медіа-сайтів, створених із застосуванням мови програмування Python;

– ідентифікувати можливі атаки та інші загрози безпеці інформації з обмеженим доступом, що циркулює через медіа-сайти;

– проаналізувати можливості захисту медіа-сайтів, створених із застосуванням мови програмування Python, від ідентифікованих атак;

– дослідити реалізацію стандартного функціонального профілю захищеності (технологія Т2) для забезпечення захисту інформації від загроз;

– розробити рекомендації із підвищення захищеності захисту медіа-сайтів, створених із застосуванням мови програмування Python, від несанкціонованого доступу.

Об’єкт дослідження – процес проектування та вдосконалення медіа-сайтів, створених із застосуванням мови програмування Python.

Предмет дослідження – вразливості та загрози для безпеки медіа-сайтів, створених із застосуванням мови програмування Python.

При вирішенні поставлених завдань у роботі використані: системний підхід, методи порівняння, аналізу, індукції, дедукції, аналогії.

Новизна одержаних результатів: вперше визначено тип медіа-сайтів; ідентифіковані вразливості та інші загрози для безпеки медіа-сайтів, створених із застосуванням мови програмування Python; розроблені рекомендації із захисту медіа-сайтів, створених із застосуванням мови програмування Python, від несанкціонованого доступу.

Практична цінність роботи полягає в тому, що при використанні розробниками рекомендацій із захисту, даних в роботі, збільшиться захищеність медіа-сайтів, створених із застосуванням мови програмування Python, від несанкціонованого доступу, а це, в свою чергу, призведе до того, що зменшаться матеріальні збитки власників сайтів від порушення цілісності, конфіденційності чи доступності інформації, якою вони володіють.

РОЗДІЛ 1. СТАН ПИТАННЯ. ПОСТАНОВКА ЗАВДАННЯ

1.1 Види сайтів і принцип роботи

Web-сайт (або сайт) – це сукупність web-сторінок, пов'язаних між собою, які, як правило, визначені загальним доменним ім'ям і розташовуються принаймні на одному web-сервері. Web-сторінки можуть включати в себе мультимедійний зміст. Web-сайт може бути доступним через публічну мережу за протоколом Internet Protocol (IP), таку як Інтернет, або приватну локальну мережу (LAN), по посиланню на адресу ресурсу (URL), який ідентифікує web-сайт. Все публічно доступні web-сайти колективно складають Всесвітню павутину, в той час як приватні web-сайти, як правило – частина Інтранету (внутрішньокорпоративної мережі, що використовує стандарти, технології і програмне забезпечення Інтернету).

Всю величезну кількість існуючих сайтів можна розбити на 2 основні групи: статичні сайти і динамічні сайти.

Статичний сайт – сайт, що складається з незмінних, тобто статичних, HTML-сторінок.

Статичні HTML-сторінки створюються вручну, після чого при кожному зверненні до сайту представляються користувачеві в незмінному вигляді. Для оновлення інформації на подібних сторінках, необхідно вручну внести зміни безпосередньо в програмний код сторінки, [2].

Для створення динамічних сайтів використовуються переважно інтерпретовані мови програмування. А саме, їх фреймворки та спеціальні системи управління контентом (CMS).

На відміну від статичних, динамічні сайти набагато гнучкіші в управлінні. Динамічні сайти представляють собою сукупність тексту і графіки, мови розмітки – точно так само, як і статичні сайти. Проте на додаток до цього динамічні сайти використовують також різні технології, що дозволяють «збирати» web-сторінки під час роботи.

Динамічні сайти можна розробляти «з нуля», вручну створюючи всі необхідні програмні коди, скрипти і т.д. Однак набагато частіше для створення динамічних сайтів використовуються CMS, [3].

CMS – це комп'ютерна програма або система, яку використовують для забезпечення і організації сумісного процесу створення, редагування і керування вмістом сайту (текстовими, графічними чи мультимедійними елементами). Зазвичай, в CMS вміст розглядається як сукупність неструктурованих даних предметного завдання на противагу до структурованих даних сайтів, які знаходяться під керуванням СКБД. CMS має панель управління, яка є лише частиною системи, але достатньою для керування сайтом. Існують різноманітні системи управління сайтом, що створені за різними технологіями, серед яких є платні і безкоштовні, [4].

CMS дозволяють використовувати вже готові програмні модулі і компоненти, без необхідності кожного разу створювати їх «з нуля». На основі однієї CMS можна створити будь-яку кількість динамічних сайтів.

Динамічні сайти в браузері формуються з декількох частин або ж браузер заповнює інформацією вже готові шаблони сторінок. У динамічних сайтах реалізовано поділ змісту та оформлення web-сторінок – це дозволяє оперативно змінювати інформацію на сайтах без необхідності змінювати програмні коди сторінок.

Приклад CMS одного з фреймворків мови програмування Python – Django зображений на рисунку 1.1.

Подібний підхід до формування web-сторінок – одна з найголовніших переваг динамічних сайтів. Поділ контента і дизайну сайту дає можливість керувати сайтом будь-якому користувачеві, навіть без знання web-програмування.

Динамічні сайти можуть підлаштовуватися під своїх відвідувачів, реагуючи на їх дії. Для цього використовуються технології серверних,

клієнтських скриптів, із застосуванням яких і створюються сценарії поведінки сайту при певних діях користувачів.

Крім перерахованих переваг, динамічні сайти мають і ряд недоліків. У порівнянні зі статичними сайтами, динамічні більш «важкі», дають велике навантаження на сервер – відтак вони більш вимогливі до хостингу, ресурсів сервера.

Щоб динамічні сайти працювали, потрібно додаткове програмне забезпечення, тоді як для відображення статичних сайтів досить одного лише браузера. Це робить розробку і підтримку динамічних сайтів дорожчою в порівнянні зі статичними сайтами.

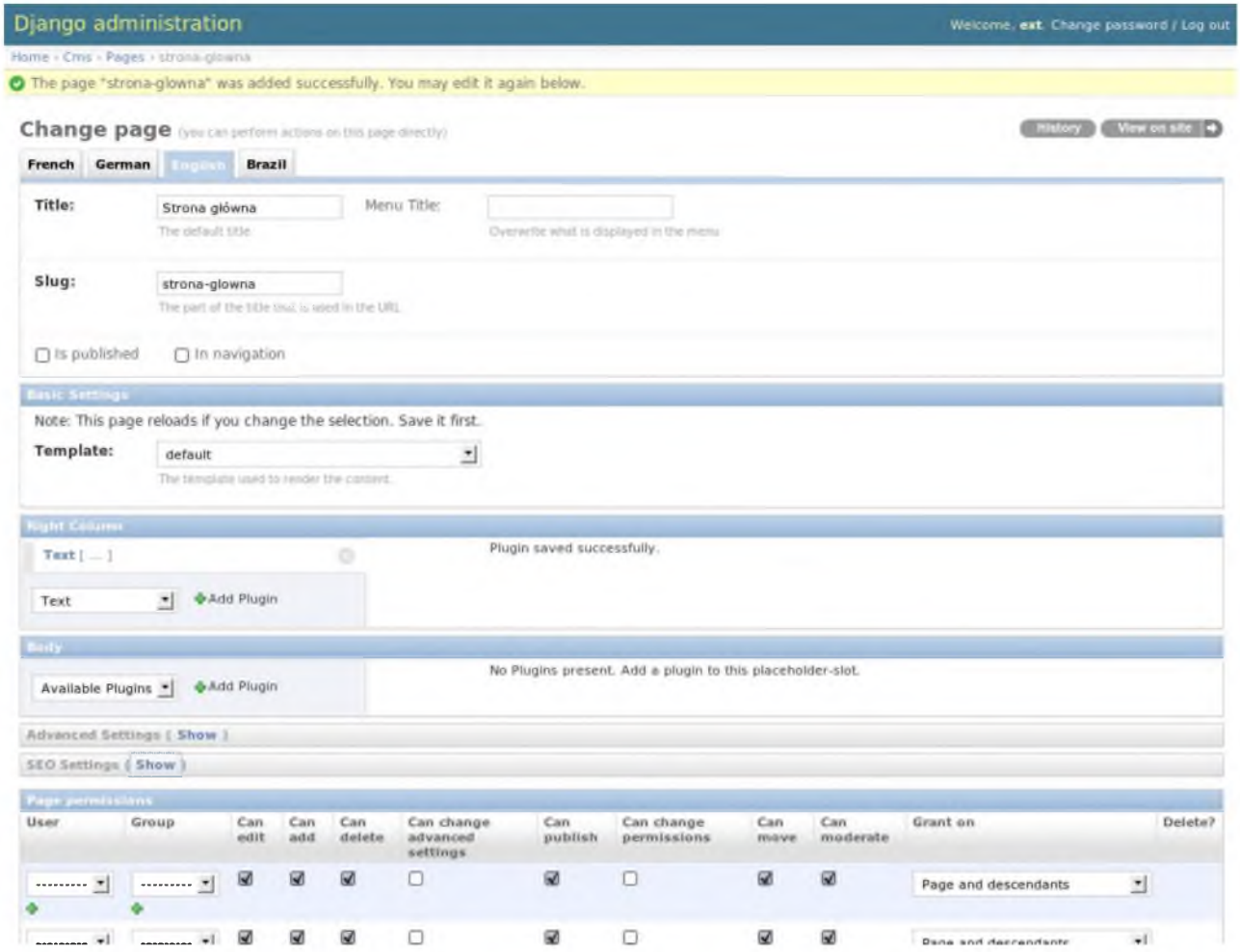

Рисунок 1.1 – Візуальна частина системи управління контентом фреймворку Django

Функціонування WEB-сторінки забезпечується автоматизованою системою (АС), із застосуванням якої здійснюється актуалізація розміщених на WEB-сторінці інформаційних ресурсів та керування доступом до них.

До складу АС, яка забезпечує функціонування WEB-сторінки, входять: ОС, фізичне середовище, в якому вона знаходиться і функціонує, середовище користувачів, оброблювана інформація, у тому числі й технологія її оброблення, [5].

Розглянемо принцип роботи динамічного сайту.

В браузері формується запит (наприклад, http://site.com/page).

http:// – протокол взаємодії браузера з сервером. У нашому випадку це HTTP.

site.com – це адреса сайту або доменне ім'я, що служить для ідентифікації конкретного сайту із застосуванням служби DNS.

/page – підзапит до сайту. Це може бути шлях до певного файлу або каталогу на сервері. Тут же можуть бути задані різні параметри HTTP-запиту.

Після того, як URL сайту введений в адресний рядок, і натиснута кнопка Enter, браузер формує пакет даних, який посилає по мережі. Цей пакет містить URL запитуваної сайту, а також інші дані запиту, оформлені згідно з протоколом HTTP. Переданий URL дозволяє проміжним вузлам в мережі Інтернет доставити пакет з HTTP-запитом за адресою до потрібного сервера.

На фізичному web-сервері повинна бути запущена відповідна програма, яка називається web-сервер та необхідна для обробки вхідних HTTP-запитів. Найпопулярнішим web-сервером на даний момент є програма Apache.

Після отримання пакету з HTTP-запитом web-сервер визначає, які дії необхідні для його обробки. Якщо HTTP-запит здійснюється до звичайної HTML-сторінці, то web-сервер просто передає її вміст браузеру. Якщо ж HTTP-запит здійснюється до якого-небудь скрипту (наприклад, PHP-скрипту), web-сервер передає запит на обробку відповідною програмою (інтерпретатор), що

відповідає за обробку цього типу скриптів. Оброблювач скрипта в свою чергу може викликати інші програми в ході своєї роботи, наприклад, СУБД MySQL.

Результатом роботи обробника скриптів є HTML-код, який web-сервер посилає назад на комп'ютер користувача. Згенерований HTML-код web-сервер упаковує в HTTP-пакет, який і передається по мережі назад клієнту. Отримана HTTP-відповідь потрапляє в браузер клієнта, який копіює з неї HTML-код і генерує на його основі графічне представлення запитаної сторінки, [6].

Схему роботи сайту представлено на рисунку 1.2.

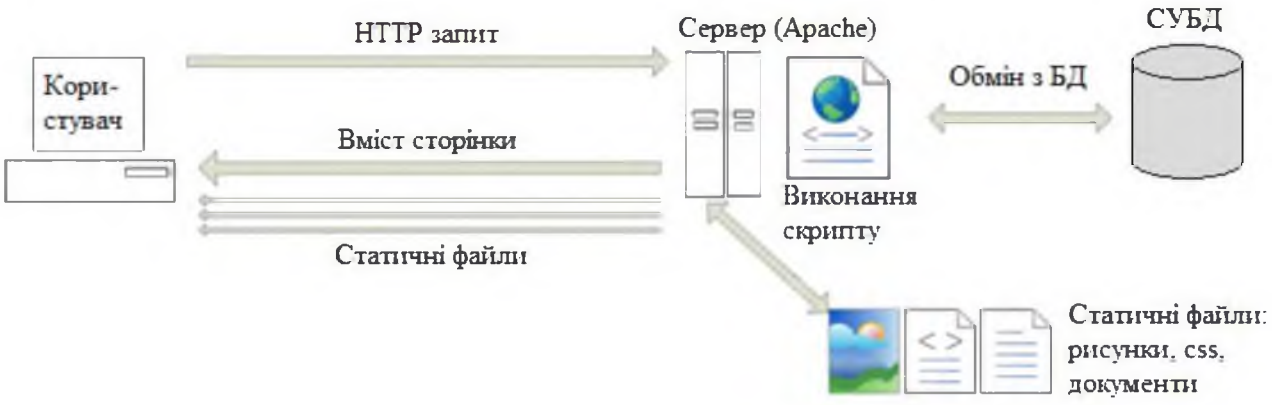

Рисунок 1.2 – Схема роботи динамічного web-сайта

1.2 Визначення медіа-сайту

Потребу розуміти, що є медіа, зумовлено їхньою значною роллю в сучасному світі, світі інтенсивних інформаційних відносин, що неможливі в таких величезних масштабах без ужитку медіа.

У комунікації, медіа (англ. media – засоби, способи) – це канали та інструменти; їх використовують, щоб зберігати, передавати й подавати інформацію або дані. Медіа часто згадувано як синонім до масс-медіа або новинних медіа, але в ширшому користуванні вони означають єдине середовище; його використовують, щоб передавати будь-які дані в яких-будь цілях, [7].

Отже, медіа можна розглядати як комунікацію - процес передачі або обміну інформацією. Комунікація може бути односторонньою та

двосторонньою. При односторонній комунікації поведінка одного з його учасників обмежена лише передачею інформації, а іншого - тільки її прийомом, [8]. При двосторонній – усі учасники можуть як передавати, так і приймати інформацію.

У контексті мережі Інтернет одностороння комунікація сприймається як Інтернет-ЗМІ, а двостороння – як соціальні медіа.

Інтернет-ЗМІ (Інтернет-видання, Інтернет-газета) – регулярно оновлюваний інформаційний сайт, який ставить своїм завданням виконувати функцію засобу масової інформації (ЗМІ) і користується певною популярністю і авторитетом (має свою постійну аудиторію), [9, с.432].

В усьому світі зараз спостерігається тенденція падіння тиражів щоденних друкованих засобів інформації та зростання кількості відвідувань Інтернет-ЗМІ. Це пов'язано з тим, що Інтернет стає все більш доступним і швидким, а також свідчить про те, що люди все більше і більше звертаються за новинами до Інтернет-ЗМІ, які оперативно подають новини в режимі реального часу, на відміну від щоденних газет, які поступово занепадають.

Популярність Інтернет-ЗМІ вираховується із застосуванням рейтингових систем спеціалізованих сайтів, які збирають та обробляють статистику відвідувань інших сайтів.

На сторінках Інтернет-ЗМІ, зареєстрованого в рейтинговій системі, може бути встановлений лічильник, який рахує кількість відвідувань. Виходячи з цих даних вираховується рейтинг Інтернет-ЗМІ. Чим популярніше сторінка, чим більше відвідувачів на неї заходили, тим вище у рейтингу вона розташована.

Рейтинг характеризує привабливість матеріалів Інтернет-ЗМІ для масової аудиторії, але не свідчить про її змістовність і корисність для читачів та впливовість на цільові аудиторії. Також рейтинг Інтернет-ЗМІ важливий для рекламодавців.

На сьогодні Інтернет-ЗМІ є важливим та впливовим джерелом інформації.

Соціальні медіа – вид мас-медіа, ряд онлайнових технологій, завдяки яким споживачі контенту через свої дописи стають його співавторами і можуть взаємодіяти, співпрацювати, спілкуватися, ділитися інформацією або брати участь у будь-якій інший соціальній активності із теоретично усіма іншими користувачами певного сервісу, [10].

Завдяки появі соціальних медіа різко збільшилась наявна у світі кількість інформації та пришвидшилось поширення новин до більшої кількості користувачів.

Рисунок 1.3 – медіа-сайт «Mediavillage» (тип Інтернет-ЗМІ)

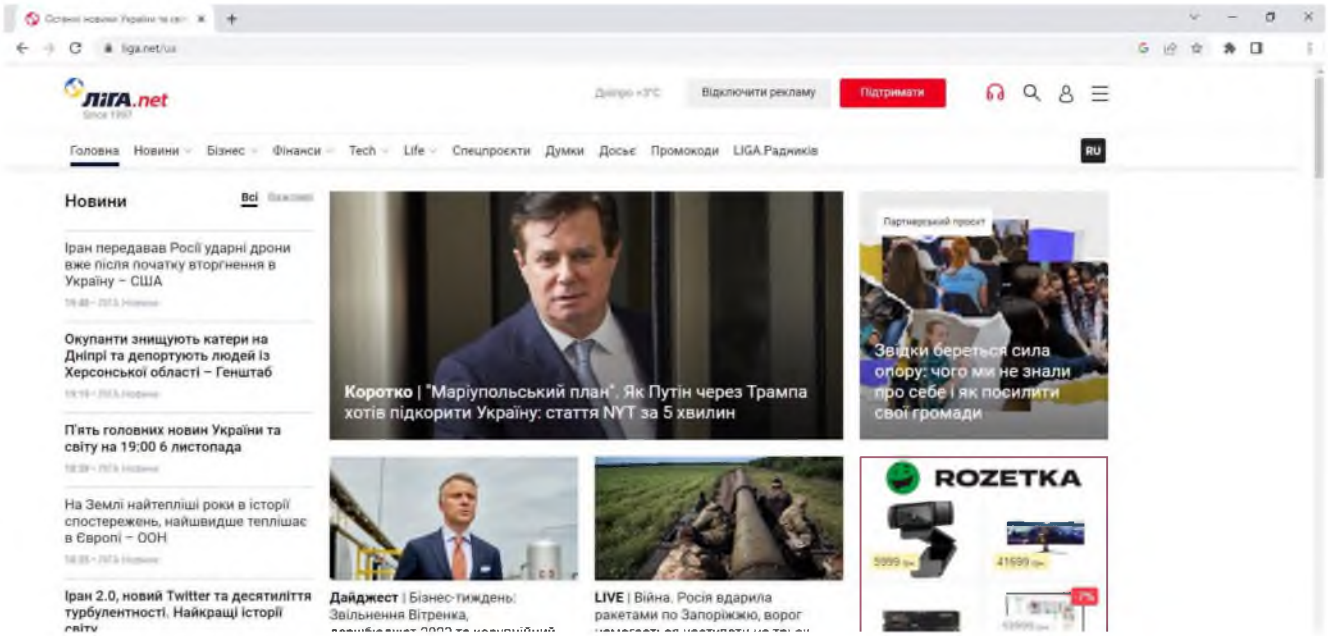

Рисунок 1.4 – медіа-сайт «Ліга.Новости» (тип Інтернет-ЗМІ)

Відмінні особливості соціальних медіа:

– можливість коригування опублікованої інформації;
– інтерактивність;
– доступне відстеження популярності публікацій;
– швидка доступність старих матеріалів;
– мультимедійність;
– відсутність процедури узгодження матеріалів;
– необмеженість за обсягом;
– посилання на інші матеріали;
– неповний контроль над вмістом сторінки.

Яскравими прикладами соціальних медіа є соціальні мережі ВКонтакте, Facebook, Одноклассники, Instagram, тощо.

Обидва типи сайтів відносяться до сайтів передачі медіа-контенту – сайтів, які містять медіа-контент (текст, цифрові зображення, відео кліпи тощо) та дають можливість передавати його іншим користувачам.

1.3 Інформаційне наповнення сайту

Інформація WEB-сторінки та сайту в цілому поділяється на дві категорії:

– загальнодоступна інформація;

– технологічна інформація.

До загальнодоступної інформації відноситься публічно оголошувана інформація, користуватися якою можуть будь-які фізичні або юридичні особи (користувачі інформаційних ресурсів), що мають доступ до мережі Інтернет.

До технологічної інформації WEB-сторінки відноситься технологічна інформація системи захисту та технологічна інформація щодо адміністрування та управління обчислювальною системою АС і засобами обробки інформації – дані про мережеві адреси, імена, персональні ідентифікатори та паролі користувачів, їхні повноваження та права доступу до об'єктів, інформація журналів реєстрації дій користувачів, інша інформація баз даних захисту, встановлені робочі параметри окремих механізмів або засобів захисту, інформація про профілі обладнання та режими його функціонування, робочі параметри функціонального ПЗ тощо.

Технологічна інформація призначена для використання тільки уповноваженими користувачами з числа співробітників СЗІ та персоналу, що забезпечує функціонування АС.

Способи і методи обробки інформації WEB-сторінки (зберігання, супроводження, передачі, введення, актуалізації та використання інформації) визначають технології оброблення інформації, [5].

Весь контент охороняється законом про авторське право, [11], оскільки він є продуктом інтелектуальної праці і має своїх авторів і власників. Окрім якості контенту одним з важливих критеріїв є його доступність. Особливу важливість для користувача має актуальність контенту, його значущість на даний час і достовірність наданих даних, а також відповідність контенту до поставлених цілей.

Унікальний контент (ексклюзивний контент) - це інформація, яка не має аналогів на ресурсах схожої тематики або розміщена на web-сайті з дозволу правовласника, така, що є результатом інтелектуальної праці та охороняється законом про авторське право. Зазвичай контент на сайт додається із застосуванням системи управління контентом (CMS), що згадувалась у попередньому підрозділі.

Унікальні статті, що написані для конкретного ресурсу, розміщуються на ньому і є першоджерелом, будь-який передрук допустимий лише з дозволу законного власника і за його умовами. Грамотне, якісно виконане і цікаве наповнення сайту здатне значно допомогти компанії і її сайту – підвищити відвідуваність, популярність, прибуток, [12].

1.4 Поняття та види мов програмування

Мова програмування – формальна знакова система, призначена для запису комп'ютерних програм. Мова програмування визначає набір правил, які задають зовнішній вигляд програми і дії, які виконує комп'ютер під її управлінням.

Мови програмування призначені для написання комп'ютерних програм, які застосовуються для передачі комп'ютеру інструкцій з виконання того чи іншого обчислювального процесу та організації управління окремими пристроями.

Мова програмування відрізняється від природних мов тим, що вона призначена того, щоб передавати команди та дані від людини до комп'ютера, в той час як природні мови використовуються, як правило, для спілкування людей між собою.

Можна узагальнити визначення «мов програмування» – це спосіб передачі команд, наказів, чіткого керівництва до дії; тоді як людські мови служать також для обміну інформацією, [13].

З часу створення перших програмованих машин було придумано більш ніж вісім тисяч мов програмування (включаючи нестандартні, візуальні та езотеричні мови). Щороку їх кількість збільшується.

За способом реалізації мови програмування поділяються на компільовані й інтерпретовані.

Програма, що написана із застосуванням компільованої мови, використовує компілятор (особливу програму) який перетворює (компілює) її в машинний код (набір інструкцій) для даного типу процесора і далі вона збирається в виконуємий модуль, який може бути запущений на виконання як окрема програма. Іншими словами, компілятор переводить вихідний текст програми з мови програмування високого рівня в двійкові коди інструкцій процесора, [14].

Якщо програма написана із застосуванням інтерпретованої мови, то інтерпретатор безпосередньо виконує (інтерпретує) вихідний текст без попереднього перекладу. При цьому програма залишається мовою оригіналу і не може бути запущена без інтерпретатора. Процесор комп'ютера, в зв'язку з цим, можна назвати інтерпретатором для машинного коду, [15].

Приклад компільованих мов: C++, Objective-C, Delphi.

Приклад інтерпретованих мов: PHP, Javascript, Ruby, Pascal, Python.

Деякі мови перебувають між компільованими і інтерпретованими. Вони використовують віртуальну машину.

Приклади таких мов: Java і C#.

Для систем, розрахованих на постійну роботу (наприклад, для серверів, на яких розташовуються web-сайти), безпека грає дуже значну роль.

В останні роки кількість атак, націлених на web-сайти, значно зросла, так як з'являються все нові варіанти і засоби для атак. І ця тенденція схильна до продовження, [16].

Багато популярних мов програмування та тих, що набирають популярність, використовують інтерпретацію. Отже, сайти, написані із застосуванням інтерпретованих мов програмування, цікаві як розробникам, так і зловмисникам. Такі сайти вимагатимуть все більшої уваги в плані захисту інформації.

Інтерпретована мова програмування – мова програмування, в якій вихідний код програми не перетворюється в машинний код для безпосереднього виконання центральним процесором (як в компільованих мовах), а виконується із застосуванням спеціальної програми-інтерпретатора.

До інтерпретованих мов можна віднести всі скриптові мови.

Інтерпретовані (скриптові) мови швидко стають мовами загальної реалізації для багатьох областей. Переважно, вони використовуються там, де час розробника більш важливий, ніж час виконання (і навіть там, де важливий час виконання; наприклад, завдяки вбудованим операціями високого рівня швидкість виконання програм, написаних на Python, така ж, або навіть швидше, ніж програм, написаних на Java). Зараз дуже часто трапляється використання позначення "динамічної мови" замість "скриптової мови", посилаючись на відсутність контролю типів, що виконується в процесі компіляції.

Інтерпретовані мови програмування дозволяють розробникам з'єднувати разом різні пакети програм, а також погоджувати системи, що отримуються в результаті.

Все частіше інтерпретовані мови самі по собі використовуються в якості повноцінних базових інструментальних платформ.

Природно, що інтерпретовані мови використовуються для автоматизації завдань системного адміністрування.

Існує вірогідність, що інтерпретовані або оперативно компільовані мови все більше і більше будуть заступати на зміну попередньо компільованих мов. Компіляція з часом буде розглядатися просто як інструмент оптимізації (чим вона власне і є), використання якого в усіх випадках навряд чи розумно. Вона все ще буде корисна при відправці автономного коду за межі керуємого середовища, однак, компіляція все частіше буде розглядатися просто як спосіб упаковки. З іншого боку, межа між компіляцією і інтерпретацією, яка завжди була трохи довільна, буде розмита ще більше. У мови Perl вже є фаза оперативної компіляції перед інтерпретацією. Майбутнє за сумісністю платформ, так що компілятори

все частіше будуть націлені на абстрактні "віртуальні машини" (як JVM у Sun або CLR у Microsoft), які нашаровуються на апаратні засоби.

Динамічні мови в якості домінуючих мов реалізації в багатьох областях можуть з часом перегнати Java і C++.

Привабливість інтерпретованих мов програмування полягає в тому, що вони мають більш складний інструментарій і підтримують більш прогресивні техніки програмування. Наприклад, можливості сортування даних в Perl вбудовані напряму в мову. Те, що в мову "вмонтовані" всі основні інструменти програмування, позбавляє від необхідності створювати їх самостійно і означає, що для вирішення конкретної проблеми потрібно писати менше коду, що збільшує продуктивність розробника.

Інтерпретовані мови дозволяють швидко виконувати доопрацювання коду без значної втрати часу на очікування закінчення компіляції.

Кількість людей, що не володіють підготовкою, яку мають традиційні комп'ютерні фахівці, але можуть зайнятися написанням сценаріїв, стало на порядок більше. Інакше кажучи, програмуванню на скриптових мовах простіше навчитися. Щоб стати "середнім" програмістом на C++, необхідний більший досвід роботи, ніж для того, щоб стати "середнім" програмістом на PHP.

Щодо недоліків інтерпретованих мов програмування, то головною проблемою, як і раніше, є час виконання. Є багато галузей розробки, де швидкість занадто важлива, щоб можна було програмувати безпосередньо на скриптовій мові. Ця проблема зазвичай вирішується тим, що код ретельно обраної частини програми (приблизно 10-30%) пишеться на мові низького рівня (такій, як C або C++); наприклад, в Python є розвинені механізми для того, щоб вставити такий код (як і в більшості інших динамічних мов).

Загальною проблемою всіх скриптових мов є відсутність якісного інтегрованого середовища розробки (IDE). Звичайно, деякі інтегровані середовища розробки існують, проте в них бракує потужності, як у Visual Studio.

Ключовим нетехнічним, однак важливим недоліком є відсутність маркетингового бюджету. Багато динамічних мов ідеально підходять для багатьох проєктів, проте їм важко конкурувати з такими гігантами маркетингу, як Sun (Java) і Microsoft (C#), які продовжують просувати свої технології як єдині можливі, [17]. В історії є приклади того, як технічна перевага нівелюється чудовим маркетингом, [18].

1.5 Аналіз найпоширеніших інтерпретованих мов програмування

Далі буде приведена коротка характеристика найпоширеніших інтерпретованих мов програмування – PHP, Python, Ruby, JavaScript.

1) PHP – інтерпретована скриптова мова програмування, створена для генерації HTML-сторінок на web-сервері і роботи з базами даних.

Ця мова виявилася досить гнучкою і потужною, тому набула великої популярності і використовується в проєктах будь-якого масштабу: від простого блогу до найбільших web-додатків в мережі Інтернет.

В області web-програмування PHP є на сьогоднішній день однією з найбільш поширених завдяки простоті, швидкості виконання і багатій функціональності. Також, PHP поширюється вільно. Синтаксис мови нагадує синтаксис C++. PHP підтримується переважною більшістю надавачів мережевого хостингу.

З моменту створення PHP значно змінилася. Однією з сильних сторін PHP є можливість розширення ядра. Інтерфейс написання розширень привернув до PHP безліч сторонніх розробників, що працюють над своїми модулями, що дало PHP можливість працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API (інтерфейс прикладного програмування). PHP підтримує об’єктно-орієнтоване програмування (деструктори, відкриті, закриті і захищені члени і методи, final-члени і методи, інтерфейси і клонування об'єктів) та XML, [19].

Скрипти, написані на мові PHP, зазвичай зберігаються в файлах з розширенням .php, які містять в собі суміш звичайних HTML-тегів зі спеціальною розміткою: відкриваючим тегом <? Php і закриваючим ?>.

Переваги PHP:

– є вільним програмним забезпеченням, поширюваним під особливою ліцензією (PHP license);

– легка в освоєнні на всіх етапах;

– підтримується великою спільнотою користувачів і розробників;

– має розвинену підтримку баз даних;

– є величезна кількість бібліотек і розширень мови;

– може використовуватися в ізольованому середовищі;

– пропонує "рідні" засоби організації web-сесій, програмний інтерфейс додатків;

– є досить повною заміною середовища ASP (Active Server Pages) від Microsoft;

– може бути розгорнута майже на будь-якому сервері;

– портовано під велику кількість апаратних платформ і операційних систем.

Недоліки PHP:

– не підходить для створення додатків для стаціонарних комп'ютерів або системних компонентів;

– дуже обмежені можливості обробки стандартних помилок та виключень, [20];

– глобальні параметри конфігурації впливають на базовий синтаксис мови, що ускладнює налаштування сервера і розгортання додатків;

– web-додатки, написані на PHP, часто мають проблеми з безпекою, [21].

Проєкти, що використовують PHP:

Zend, Yahoo, Facebook, Google, NASA, W3C.

2) Python – інтерпретована, об'єктно-орієнтована мова програмування високого рівня. Вона підтримує класи, модулі (які можуть бути об'єднані в пакети), обробку винятків, а також багатониткову обробку. Python відноситься до класу мов з динамічною типізацією, надає програмісту автоматичне видалення об'єктів, які вже не будуть використані додатками і зручні високорівневі структури даних, такі як словники, списки, кортежі та ін. Python поєднує широкий функціонал з простим і зрозумілим синтаксисом, продуманою модульністю і масштабованістю. Python – одна з найпростіших об'єктно-орієнтованих мов для навчання і застосування.

Python адаптована для роботи у різних середовищах та працює майже на всіх відомих платформах. Існують адаптації під Windows, всі варіанти UNIX (включаючи Linux), Mac OS, і т.д. При цьому, на відміну від багатьох систем, що можуть працювати у різних середовищах, на кожній платформі Python підтримує всі характерні для даної платформи технології (наприклад, Microsoft COM/DCOM).

Інтерпретатор мови Python поширюється вільно на підставі ліцензії Python Software Foundation (PSF) Licence.

У стандартний комплект поставки Python входить інтегроване середовище розробки IDLE, в якій редагувати програми буде набагато зручніше, ніж в простому текстовому редакторі. IDLE написаний на Python з використанням платформонезалежної бібліотеки Tcl, тому легко запускається в будь-якій операційній системі, для якої існує реалізація Python. IDLE також має вбудовану систему налагодження.

Стандартна бібліотека мови Python різноманітна і надає програмісту безліч можливостей. Однак, якщо не достатньо можливостей стандартної бібліотеки, то існує безліч бібліотек, які надають інтерфейс до всіх можливих системних викликів на різних платформах.

Python і переважна більшість бібліотек до нього безкоштовні і поставляються у вихідних кодах. Більш того, на відміну від багатьох відкритих

систем, ліцензійна політика на Python ніяк не обмежує його використання в комерційних системах і не накладає ніяких зобов'язань, крім вказівки авторських прав.

Інтерпретатор Python можна використовувати як для запуску скриптів різного призначення, так і в режимі інтерактивної оболонки.

Переваги Python:

– відкрита розробка;

– досить проста у вивченні, отже, багато хакерів-початківців можуть використовувати саме цю мову;

– особливості синтаксису стимулюють програміста писати код, що добре розуміється;

– надає засоби швидкого прототипування і динамічної семантики;

– безліч корисних бібліотек і розширень мови можна легко використовувати в своїх проєктах завдяки гранично уніфікованого механізму імпорту і програмним інтерфейсами;

– механізми модульності добре продумані і можуть бути легко використані;

– абсолютно все в Python є об'єктами об’єктно-орієнтованого програмування (ООП), але при цьому об'єктний підхід не нав'язується програмісту.

Недоліки Python:

– не дуже вдала підтримка багатопоточності, [22];

– на Python створено досить небагато якісних програмних проєктів у порівнянні з іншими універсальними мовами програмування, наприклад, з Java;

– початкова обмеженість засобів для роботи з базами даних.

Проєкти з використанням Python:

Yahoo Maps, Zope Corporation, Linux Weekly News, Shopzilla, Ultraseek.

3) Ruby – інтерпретована скриптова мова високого рівня для швидкого і зручного об'єктно-орієнтованого програмування. Була створена під впливом таких мов, як Perl, Eiffel і Smalltalk.

Ruby має велику кількість засобів для обробки текстів, для вирішення системних завдань. Ruby є повністю вільною мовою програмування з можливістю копіювання, модифікації і поширення. Ruby перенесена на безліч платформ. Вона розроблялася на Linux, але працює на багатьох версіях Unix, DOS, Windows, Mac OS, і т.д. Метою створення Ruby було створення справжньої об'єктно-орієнтованої інтерпретованої мови програмування.

Ruby має простий і зрозумілий синтаксис, дозволяє обробляти виключення в стилі Java і Python, дозволяє легко перевизначати оператори, які насправді є методами. Ruby – повністю об'єктно-орієнтована мова програмування. Всі дані в Ruby є об'єктами в розумінні SmallTalk. Наприклад, число «1» – це екземпляр класу Fixnum. Також підтримується додавання методів в клас і навіть в конкретний екземпляр під час виконання програми. Ruby свідомо не підтримує множинне успадкування, замість якого існує концепція модулів. Ruby містить автоматичний збирач сміття. Він працює для всіх об'єктів Ruby, так що не треба піклуватися про підрахунок посилань навіть у зовнішніх бібліотеках. Ruby не вимагає оголошення змінних. Мова використовує прості угоди для позначення області видимості, [23].

Ruby має незалежну від ОС підтримку багатопоточності. Вона характеризується динамічною типізацією і автоматичним управлінням пам'яттю. Мова Ruby використовується в web-розробці в складі відкритого web-фреймворку Rails, частіше званого Ruby on Rails (RoR).

Переваги Ruby:

– відкрита розробка;

– може впроваджуватися в HTML-розмітку;

– належить до мов програмування надвисокого рівня (VHLL), тобто має високий рівень абстракції і предметним підходом в реалізації алгоритмів;

– реалізує концептуально чисту об'єктно-орієнтовану парадигму;

– надає просунуті методи маніпуляції рядками і текстом;

– легко інтегрує в свої програми високопродуктивні сервери баз даних (DB2, MySQL, Oracle і Sybase);

– завдяки VHLL програми на Ruby добре масштабуються і легко супроводжуються;

– є простий програмний інтерфейс для створення багатопоточних додатків;

– можливості мови можна розширити із застосуванням бібліотек, написаних на C або Ruby;

– зарезервовані слова можуть бути ідентифікаторами, якщо це не створює неоднозначності для сценарію або програми, які використовуються для збору інформації з сайтів для подальшого розміщення на власних ресурсах;

– додаткові можливості для забезпечення безпеки (фреймворк Ruby on Rails за замовчуванням сильно заточений під безпеку проєкту. При використанні інструментів RoR виключені SQL ін'єкції і XSS атаки. Всі вхідні параметри екранується за замовчуванням);

– вбудований відладник.

Недоліки Ruby:

– Ruby менш продуктивний в порівнянні з багатьма іншими мовами, застосовуваними в web-розробці;

– необережність у роботі з багатопоточністю може привести до споживання більшої кількості обчислювальних ресурсів та додаткових затрат інфраструктури;

– створювалась для Unix систем, що викликає труднощі у роботі на Windows;

– невелика кількість документації;

– Ruby відносно повільно розробляється і розвивається.

Проєкти, що використовують Ruby:

Google SketchUp, 37signals, GitHub, Shopify, Indiegogo, Basecamp.

4) JavaScript – це мова програмування від компанії Netscape, яка є реалізацією стандарту ECMAScript.

У більшості випадків при згадці JavaScript мається на увазі так званий клієнтський JavaScript, інтерпретатор якого вбудований в Web-браузери. Однак JavaScript спочатку була розроблена як універсальна мова програмування для вбудовування в будь-який додаток і забезпечення можливості написання в ньому сценаріїв.

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності web-сторінок.

Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожим на Java, але при цьому легким для використання непрограмістів. Мовою JavaScript не володіє будь-яка компанія або організація, що відрізняє його від ряду мов програмування, використовуваних в web-розробці.

Всупереч поширеній помилці, крім деякої синтаксичної схожості, мови Java і JavaScript нічого не пов'язує. Схожість імен – не більше, ніж хід маркетологів (первинна назва мови – LiveScript – було змінено на JavaScript в останню хвилину), [24].

Переваги JavaScript:

– надає велику кількість можливостей для вирішення найрізноманітніших завдань. Гнучкість мови дозволяє використовувати безліч шаблонів програмування стосовно до конкретних умов;

– має чималу кількість готових бібліотек, які дозволяють значно спростити написання коду і нівелювати недосконалості синтаксису;

– має широкі можливості для розробки різноманітних додатків, що дозволяє застосування мови у багатьох областях.

Недоліки JavaScript:

– необхідність забезпечувати відображення і роботу у всіх популярних браузерах;

– система успадкування в мові викликає труднощі в розумінні того, що відбувається. В JavaScript реалізовано успадкування, засноване на прототипах, на відміну від інших об'єктно-орієнтованих мов програмування, де клас нащадок успадковує батьківський клас. Але в JavaScript цю функцію виконують безпосередньо об'єкти, що є незвичним для багатьох користувачів;

– відсутня стандартна бібліотека. JavaScript не надає ніяких можливостей для роботи з файлами, потоками введення-виведення та багато інших корисних функцій;

– синтаксис в цілому ускладнює розуміння.

Порівняння інтерпретованих мов програмування.

+ – Зазначена можливість присутня.

- – Зазначена можливість відсутня.

+/- – Можливість підтримується не в повному обсязі.

Таблиця 1.1 - Порівняльна характеристика можливостей інтерпретованих мов програмування

Можливість	PHP	Python	Ruby	JavaScript
Парадигми				
Імперативна	+	+	+	+
Об'єктно-орієнтована	+	+	+	+
Функціональна	+/-	+	+	+/-
Рефлексивна	+	+	+	+
Узагальнене програмування	+	+	+	+
Логічна	-	-	-	-
Декларативна	+	+	+	+/-

Розподілена	-	+/-	+/-	-
Інтерпретатор				
Інтерпретатор з відкритим кодом	+	+	+	+
Можливість компіляції	+	+	+	+
Bootstrapping	+/-	+	+	+
Інтерпретатор командного рядка	+	+	+	+
Умовна компіляція	+	-	-	+/-
Типи і структури даних				
Кортежі	+/-	+	+	-
Багатовимірні масиви	+/-	+/-	+/-	+/-
Динамічні масиви	+/-	+/-	+/-	+/-
Асоціативні масиви	+	+	+	+
Контроль кордонів масивів	-	+	-	-
Спискові включення	-	+	-	-
Цілі числа довільної довжини	+/-	+	+	-
Цілі числа з контролем кордонів	-	-	-	-
Інше				
Підтримка Unicode в ідентифікаторах	+/-	+	+	+
Автоматичне звільнення пам'яті	+	+	+	+
Наявність бібліотек для роботи з графікою і мультимедіа	+/-	+	-	+

Парадигми:

Імперативна – повинна описувати не стільки саму задачу (опис, «ЩО» потрібно отримати), скільки її рішення («ЯК» отримати).

Об'єктно-орієнтована – заснована на представленні у вигляді об'єктів, які є екземплярами того чи іншого класу і втілює застосування концепції абстрагування. Об'єкт при цьому поєднує в собі як дані, так і методи, їх обробляють. Як правило, підтримуються характерні можливості: успадкування, інкапсуляція і поліморфізм.

Рефлексивна – передбачається написання програм, які можуть змінювати свою власну поведінку. Характеризується наявністю в мові потужних механізмів

інтроспекції (можливість визначити тип і структуру об'єкта під час виконання програми). Програма може оперувати власним кодом як даними.

Функціональна – дозволяє записувати програму як композицію функцій. У чистому функціональному мові немає змінних. Так як функції не мають побічних ефектів, вони можуть виконуватися в будь-якому порядку.

Узагальнене програмування – узагальнене програмування дозволяє записувати алгоритми, які використовують дані будь-якого типу.

Логічна – програма представляє собою опис фактів і правил виведення в деякому логічному обчисленні. Бажаний результат, який часто записується як питання, виходить системою в результаті спроби застосування описаних правил – шляхом логічного висновку. Цікавими особливостями є відсутність детермінованості в загальному випадку, внутрішня схильність до розпаралелювання.

Декларативна – описує не стільки рішення задачі, скільки саму задачу («ЩО» потрібно отримати), а яким чином отримати рішення, вже повинен визначати комп'ютер.

Розподілена – мова, що містить спеціальні конструкції для підтримки розпаралелювання програми на кілька комп'ютерів.

Інтерпретатор:

Інтерпретатор з відкритим кодом – наявність повноцінного інтерпретатора з відкритим кодом.

Можливість компіляції – можливість компіляції в нативний код або в byte-код з можливістю компіляції «на льоту» (це технологія збільшення продуктивності програмних систем, що використовують байт-код, шляхом трансляції байт-коду в машинний код безпосередньо під час роботи програми).

Bootstrapping – наявність повноцінного bootstrapping-компілятора (тобто компілятора, написаного тією ж мовою, що він компілює, і успішно компілює самого себе).

Інтерпретатор командного рядка – можливість вводити інструкції мови рядок за рядком з їх негайним виконанням. Може використовуватися в якості калькулятора.

Умовна компіляція – можливість вмикати / вимикати частини коду в залежності від значення символів умовної компіляції.

Типи і структури даних:

Багатовимірні масиви – наявність вбудованих в мову багатовимірних масивів.

Динамічні масиви – наявність вбудованих в мову динамічних масивів (здатних змінювати свій розмір під час виконання програми).

Асоціативні масиви – наявність вбудованих в мову асоціативних масивів або хеш-таблиць.

Спискові включення – наявність спискових включень (або їх аналога).

Кортежі – можливість повернути з функції/методу кортеж (tuple) – неіменованого тип даних, що містить кілька безіменних полів довільного типу.

Цілі числа довільної довжини – підтримка цілих чисел необмеженої розрядності. Повинна бути можливість записати як завгодно велике ціле число із застосуванням літерала.

Цілі числа з контролем кордонів – можливість визначити тип, значеннями якого можуть бути цілі числа тільки певного інтервалу, наприклад [-5..27], при цьому привласнення змінної такого типу значення, що виходить за зазначені рамки, має викликати помилку.

Інше:

Автоматичне звільнення пам’яті – можливість використовувати автоматичний процес звільнення пам'яті, зайнятої невикористовуваними об'єктами.

Підтримка Unicode в ідентифікаторах – можливість включення Unicode-символів (наприклад, букв національних алфавітів) в ідентифікатори.

1.6 Дослідження механізмів забезпечення безпеки web-сайту

Безпека сайту складається з трьох речей:

– безпеки програмної частини (CMS, скриптів);

– безпеки сервера (хостингу);

– обізнаності та акуратності адміністратора сайту або тих, хто працює з сайтом як адміністратор.

Якщо всі три складових організовані належним чином, то сайт буде неприступним для хакерів і вірусів.

Розглянемо безпеку програмної частини.

Програмна частина – це система управління сайтом або скрипти, на яких працює сайт. Надійність програмної частини має на увазі відсутність вразливостей, що дозволяють зловмисникові отримати доступ до бази даних, файлової системи або панелі адміністратора сайту.

Поява в складі програмного забезпечення сайту будь-якого CMS обов'язково робить цей сайт менш безпечним. Справа в тому, що сформований з статичних HTML-сторінок сайт обслуговується виключно HTTP-сервером (наприклад, найвідомішим сервером Apache). Поширені HTTP-сервери дуже ретельно перевірені на наявність вразливостей, але навіть в них продовжують знаходити нові "діри". Будь-яка ж CMS встановлюється на додаток до HTTP-сервера, і, звичайно, уразливості CMS додаються до вже наявних.

При цьому, внесок CMS в кількість вразливостей, по відношенню до статичного сайту, дуже великий. Це обумовлено двома причинами: по-перше, за безпекою основ утримання серверів – зокрема, за надійністю HTTP-сервера – ретельно стежать фахівці хостинг-провайдера (а стежити за безпекою "особистої" CMS доведеться власнику сайта); по-друге, навряд чи якась з існуючих CMS пройшла настільки ж серйозні випробування на стійкість, як серверні додатки нижчого рівня.

Втім, те, що CMS неминуче призводить до зниження безпеки сайту, не повинно стає перешкодою на шляху впровадження CMS.

Практично в кожній CMS або в скрипті існують вразливості. Частина з них опублікована у відкритому доступі (публічні вразливості), інша не доступна широкій аудиторії і використовується зловмисниками для цільових атак на сайти. Для того щоб програмна частина сайту була надійна і неприступна, потрібно приділяти увагу проблемі безпеки.

Далі буде розглянута безпека сервера (хостингу). Хостинг, на якому розміщується сайт, є другим важливим моментом, що впливає на безпеку сайту в цілому. Рівень захищеності сайту багато в чому залежить від рівня рівня безпеки сервера, на якому він розміщується.

Web-сервер формується кількома шарами ПЗ, кожен з яких схильний до різноманітних способів атаки, як показано на діаграмі нижче. Слід пам'ятати, що метою атаки може стати будь-який з блоків (таблиця 1.2).

Хостинг може бути загальний або спеціалізований («виділений»).

Таблиця 1.2 – Складові частини web-серверу

<table>
<tr><td colspan="4">Операційна система</td></tr>
<tr><td colspan="3">HTTP сервер</td><td rowspan="2">MySQL сервер</td><td rowspan="4">Інші служби (FTP, SMTP)</td></tr>
<tr><td colspan="3">Серверні розширення, серверні модулі</td></tr>
<tr><td>Web-сайт 1</td><td colspan="2">Web-сайт 2</td><td colspan="2" rowspan="2">Web-сайт n</td></tr>
<tr><td>Блог</td><td>Галерея зображень</td><td>CMS</td></tr>
</table>

Якщо якісь області web-сайту повинні бути доступні тільки деяким клієнтам або зареєстрованим користувачам, для подібного розмежування доступу потрібно метод перевірки автентичності користувачів.

Існує кілька способів автентифікації користувачів: базова автентифікація, дайджест-автентифікація і HTTPS.

При використанні базової автентифікації ім'я користувача і пароль включаються до складу web-запиту. Навіть якщо контент з обмеженим доступом

не дуже важливий, цей метод краще не використовувати, тому що користувач може застосовувати один і той же пароль на декількох web-сайтах. Опитування Sophos показав, що 41% користувачів застосовують для всієї своєї діяльності в мережі Інтернет всього один пароль, будь то сайт банку або районний форум. Використання більш безпечних методів автентифікації може захистити користувачів від подібних помилок.

Дайджест-автентифікація, що підтримується всіма популярними серверами і браузерами, дозволяє надійно шифрувати ім'я користувача і пароль в запиті. Вона допомагає забезпечити безпеку імен і паролів та знижує ймовірність успішної атаки на сервер.

Протокол HTTPS дозволяє шифрувати всі дані, що передаються між браузером і сервером, а не тільки імена користувачів і паролі. Протокол HTTPS (заснований на системі безпеки SSL) слід використовувати в разі, якщо користувачі повинні вводити важливі особисті дані – адреса, номер кредитної картки або банківські відомості.

При виборі системи автентифікації рекомендується використовувати найбезпечніший варіант з наявних.

До захисту інформації можна підходити по-різному. Можна спробувати створити абсолютно надійний і недоступний іншим канал зв'язку. Але досягти цього вкрай складно, принаймні на сучасному рівні розвитку науки і техніки, оскільки існуючі методи і засоби передачі інформації одночасно дають можливість несанкціонованого доступу до неї. Можна використовувати загальнодоступний канал зв'язку, але передавати дані в перетвореному вигляді, щоб відновити їх міг лише адресат. Розробкою методів перетворення інформації, що забезпечує її шифрування, і займається криптографія.

1.7 Аналіз можливостей шифрування

Шифруванням називається процес перетворення даних в формат, в якому вони можуть бути прочитані (у всякому разі, теоретично) тільки передбачуваним одержувачем повідомлення. Одержувач розшифровує дані із застосуванням

ключа або секретного пароля. Шифрування даних в мережі Інтернет має сенс тільки в тому випадку, якщо сценарії, в яких використовуються засоби шифрування, працюють на захищеному сервері. Оскільки сценарні мови працюють на стороні сервера, перед шифруванням дані повинні бути відправлені на сервер в простому текстовому форматі. Якщо дані передаються через незахищене з'єднання, існує чимало способів перехоплення цієї інформації в процесі її пересилання від користувача на сервер.

Інтерпретовані мови програмування підтримують деякі алгоритмів шифрування.

Далі будуть описані інструменти, із застосуванням яких можливо здійснити шифрування/дешифрування даних сайту.

Python. У даній мові за шифрування/дешифрування даних відповідає бібліотека «PyCrypto». Вона призначена для забезпечення надійної і стабільної бази для написання програм на Python, які вимагають криптографічні функції.

У PHP для шифрування/дешифрування використовуються розширення Mcrypt або OpenSSL.

Mcrypt – це інтерфейс до бібліотеки з однойменною назвою «Mcrypt», яка підтримує велику кількість блокових алгоритмів.

Модуль OpenSSL використовує функції бібліотеки «OpenSSL» для генерації і перевірки електронних підписів, а також для упаковки (шифрування) і розпакування (розшифрування) даних. Бібліотека OpenSSL пропонує багато можливостей, проте не всі вони підтримуються модулем. Можливо в майбутньому функціонал модуля буде розширений.

У мові програмування Ruby за шифрування/дешифрування також відповідає бібліотека «OpenSSL».

Далі представлене порівняння криптографічних можливостей мов програмування PHP, Python, Ruby, [28].

Якщо в мові немає вбудованих засобів криптографічного захисту, то існує дві можливості.

По-перше, можна спробувати зробити все "з нуля". Та найчастіше це не надається можливим, так як пов'язано з реалізацією порівняно складних алгоритмів.

Другий спосіб – це використання об'єктних модулів, створених з використанням трансляторів з інтерпретованих мов програмування.

Таблиця 1.3 - Порівняльна характеристика криптографічних можливостей інтерпретованих мов програмування PHP, Python, Ruby

PHP	Python	Ruby
Стандарти: – DES; – 3DES; – ГОСТ 28147-89. Алгоритми: – Blowfish; – 3-WAY; – SAFER-SK64; – SAFER-SK128; – Twofish; – RC2. Додатково, підтримуються RC6 і IDEA.	Стандарти: – DES; – DES3; – AES; – RSA; – MD5; – SHA; – HMAC. Алгоритми: – CM2; – ARC4; – Blowfish; – CAST; – XOR; – MD2; – MD4; – RIPEMD; – RIPEMD160; – SHA256; – DSA. Допоміжні криптографічні утиліти: генератор випадкових чисел, перетворення числа в рядок і інші.	Стандарти: – AES; – DES. Алгоритми: – RC2; – RC2-40; – RC2-64; – CAST5; – BLOWFISH; – RC4; – RC4-40.

Є ряд мов, на яких вже реалізовані необхідні алгоритми. Трансляторами цих мов забезпечується отримання об'єктних кодів, які можна використовувати при розробці програми на іншій мові. Приклад роботи представлений на рисунку 1.5.

Рисунок 1.5 – Приклад схеми об'єднання модулів, створених із застосуванням різних мов програмування

1.8 Порівняння поширеності мов програмування

Для порівняння поширеності мов програмування було вибрано два рейтинги: «TIOBE programming community index» від сервісу TIOBE та «PYPL: PopularitY of Programming Language index» від Github.

Вони щомісячно публікують актуальний рейтинг найпопулярніших мов програмування.

«TIOBE programming community index» оцінює популярність мов програмування, на основі підрахунку результатів пошукових запитів, що містять назву мови (запит виду + "<language> programming").

Рейтинг «PYPL: PopularitY of Programming Language index» створюється шляхом аналізу частоти пошукових запитів за мовою програмування та підручниками з неї. Індекс має слоган, який говорить, що більше підручників з мови програмування шукали, тим популярнішою мова є. Це провідний показник. Неопрацьовані дані надходять з Google Trends.

Для формування індексу використовується пошук в декількох найбільш відвідуваних порталах: Google, Blogger, Wikipedia, YouTube, Baidu, Yahoo !, Bing, Amazon.

Основні властивості-відмінності паралельного рейтингу PYPL перед TIOBE:

– PYPL базується на статистиці частот пошукових запитів в заданий діапазон часу (використовуючи Google Trends), а не на частоті згадок-цитування, як у TIOBE;

– PYPL враховує більш просунуті варіанти синтаксису пошукових запитів. Наприклад, враховуються навіть національні запити-аналоги для всіх популярних мов (а не тільки англійську, як в TIOBE), [29].

Таблиця 1.4 - Рейтинг популярності мов програмування «TIOBE Programming Community Index»

Позиція	Мова програмування	Доля	Зміна (за рік, %)
1	Java	20.846%	+4.8 %
2	C	13.905%	-1.84 %
3	C++	5.918%	-1.04 %
4	C#	3.796 %	-1.15 %
5	Python	3.33 %	+0.64 %
6	PHP	2.994 %	-0.02 %
7	Javascript	2.566 %	-0.73 %
8	Perl	2.524 %	+1.18 %
9	Ruby	2.345 %	+1.28 %
10	.NET	2.273 %	+0.15 %

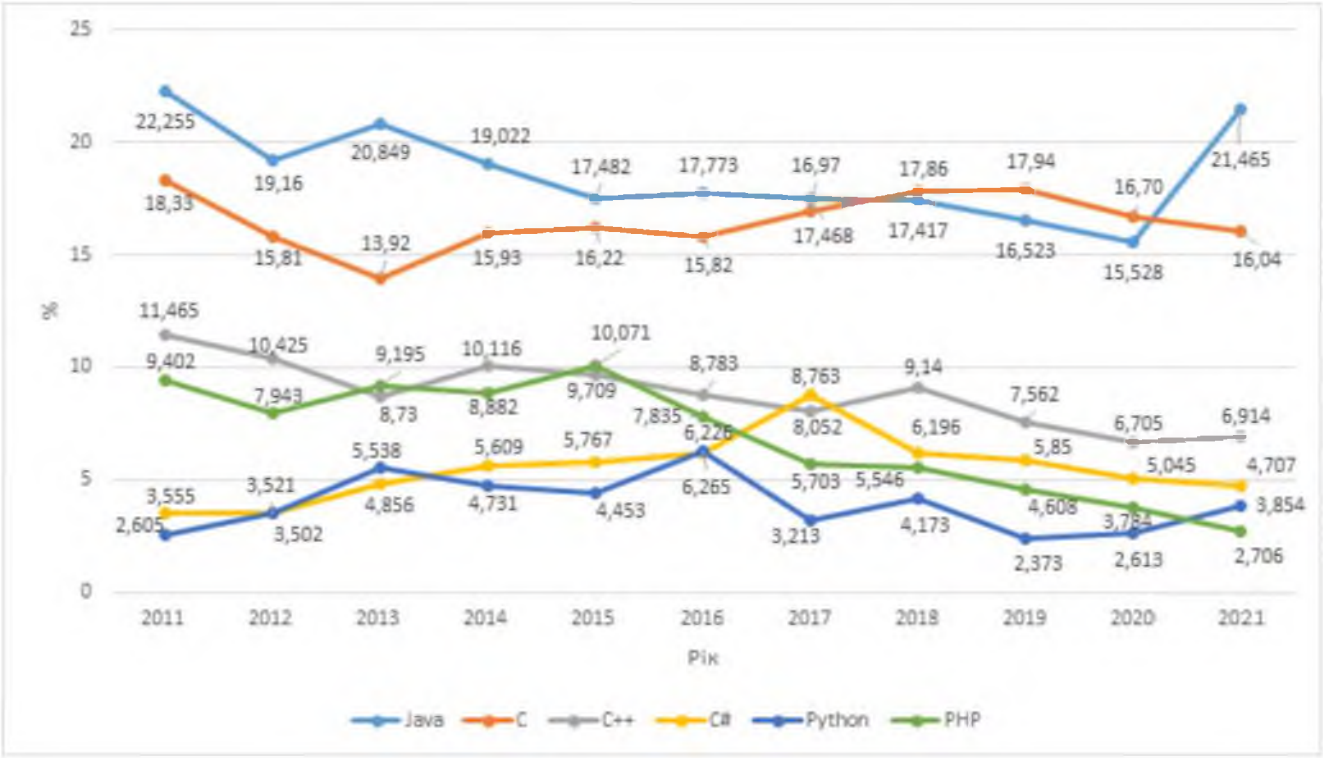

Рисунок 1.6 – Графік тенденцій розвитку найпопулярніших мов програмування на основі рейтингу «TIOBE Programming Community Index»

Таблиця 1.5 - Рейтинг популярності мов програмування «PYPL PopularitY of Programming Language»

Позиція	Мова Програмування	Доля	Зміна (за рік, %)
1	Java	24.0 %	-0.2 %
2	Python	12.4 %	+1.8 %
3	PHP	10.6 %	-0.8 %
4	C#	8.9 %	-0.4 %
5	Javascript	7.5 %	+0.5 %
6	C++	7.4 %	-0.3 %
7	C	7.2 %	+0.1 %
8	Objective-C	4.7 %	-0.7 %
9	R	3.1 %	+0.5 %
10	Swift	3 %	+0.4 %

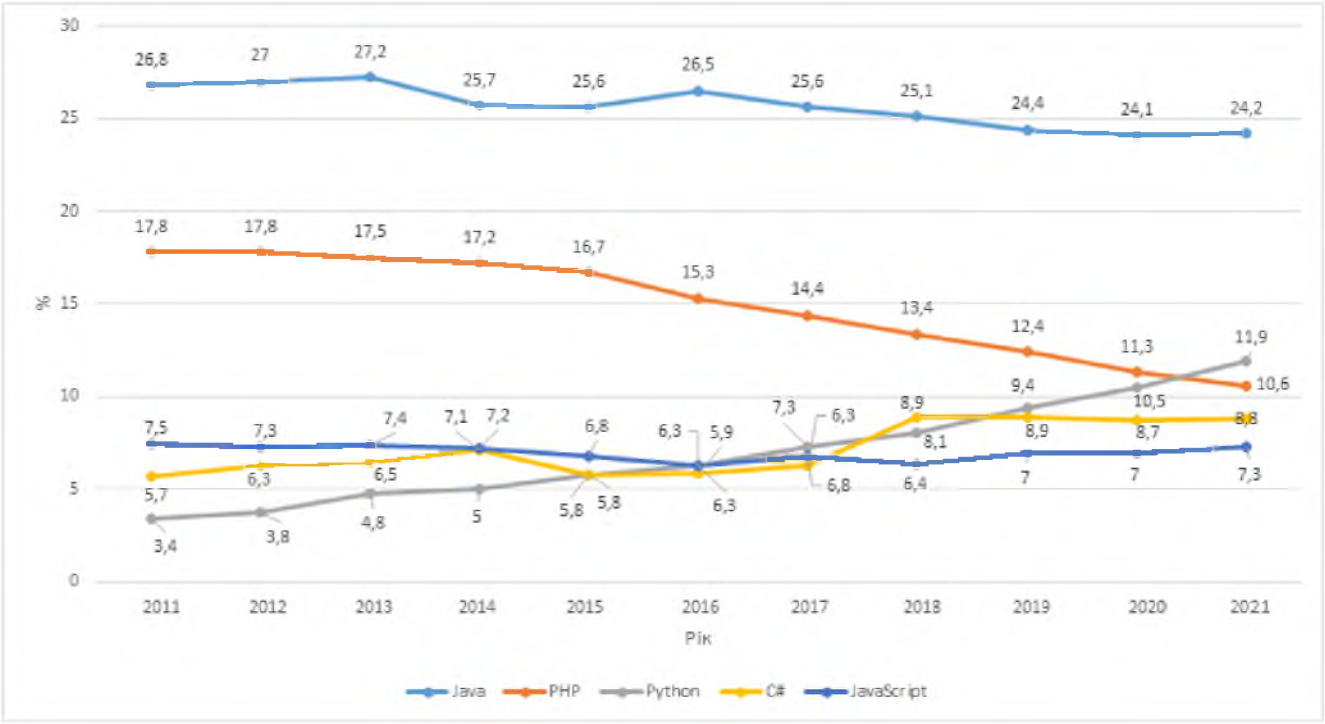

Рисунок 1.7 – Графік тенденцій розвитку найпопулярніших мов програмування на основі рейтингу «PYPL PopularitY of Programming Language»

Рейтинги показують, що мови програмування, які використовують інтерпретатор (Java, Python, PHP, JavaScript), посідають на високих місцях.

Java є незмінним лідером у цьому рейтингу. Python дуже швидко зростає. А PHP повільно втрачає популярність. Та все ж вона замикає тройку лідерів у рейтингу «PYPL: PopularitY of Programming Language index».

1.9 Вибір та обґрунтування мови програмування для дослідження безпеки медіа-сайтів

Інтерпретовані (сценарні) мови програмування дуже швидко розвиваються і набирають популярність. В якості основних мов в деяких областях реалізації вони можуть дуже швидко обігнати компільовані мови.

Їх основна перевага в тому, що вони мають більш складний інструментарій та підтримують більш прогресивні техніки програмування.

А головними недоліками інтерпретованих мов є час виконання завдань, і маркетингова складова.

Для створення web-додатків найчастіше використовуються інтерпретовані (сценарні) мови програмування з метою максимально розширити інтерактивні можливості web-ресурсу.

Вибір проводився між найбільш поширеними інтерпретованими мовами програмування (PHP, Python, Ruby, JavaScript). Для дослідження безпеки медіа-сайтів було вибрано мову програмування Python.

Python – універсальна мова програмування, із застосуванням якої можна робити будь-які додатки в діапазоні від Інтернет-сайтів та додатків до роботів для персональних комп'ютерів і системних сервісів. Мова дуже швидко набирає популярність та підтримує широкі можливості для створення та підтримки медіа-сайтів завдяки різноманітним фреймворкам та їх системам управління контентом.

1.10 Фреймворки web-додатків

На основі багатьох інтерпретованих мов створені фреймворки web-додатків (web-фреймворки), які значно полегшують створення додатків.

Фреймворк web-додатків (Web application framework) - це програмна платформа, призначена для створення динамічних web-сайтів, мережевих додатків, сервісів або ресурсів. Він спрощує розробку і позбавляє від необхідності написання рутинного коду. Велика кількість фреймворків спрощує доступ до баз даних, розробку інтерфейсу, і також зменшують дублювання коду.

Більшість сайтів мають очевидний набір базових функцій: обробка сесій і авторизація, валідація запитів і т.д. Фреймворк звільняє від необхідності переписувати всі ці функції заново при створенні сайту. Залишається тільки спроєктувати і реалізувати лише функції взаємодії сайту з користувачем.

Розвинені фреймворки, що включають в себе функції для роботи з одними даними (статтями, темами, постами, фотографіями і т.д.), називаються фреймворками управління контентом (Content Management Framework – CMF).

При додаванні до цього елементів інтерфейсу для кінцевого користувача сайту, виходить так звана система управління контентом або CMS (Content Management System). CMS дозволяє отримати сайт з прототипом інтерфейсу і мінімальною функціональністю відразу після установки або після певного налаштування, тобто взагалі без програмування.

При цьому більшість CMS надають програмісту як програмний інтерфейс CMF, так і інтерфейс для розширення своєї функціональності.

1.11 Висновки до першого розділу

У першому розділі магістерської роботи були досліджені види, особливості та структура сайтів, проаналізовано поняття медіа-сайту, його типовий склад та особливості функціонування.

Також був проведений аналіз і дослідження основних типів мов програмування (компільовані та інтерпретовані) та визначено, що для створення web-додатків на сьогоднішній день широкого використання набувають інтерпретовані (скриптові) мови програмування.

Серед медіа-сайтів, що створюються із застосуванням найпопулярніших інтерпретованих мов програмування (PHP, Python, Ruby, JavaScript), для подальшого аналізу вразливостей було вибрано медіа-сайти, створені із застосуванням мови програмування Python. Python – універсальна мова програмування, із застосуванням якої можна робити будь-які додатки в діапазоні від Інтернет-сайтів та додатків до роботів для персональних комп'ютерів і системних сервісів. Мова дуже швидко набирає популярність та підтримує широкі можливості для створення та підтримки медіа-сайтів завдяки різноманітним фреймворкам та їх системам управління контентом.

У якості прикладу медіа-сайту для обстеження було вирішено вибрати web-сайт www.mediavillage.com, який створений із застосуванням мови програмування Python та містить різноманітний медіа контент.

Ґрунтуючись на проведених дослідженнях, в спеціальній частині роботи необхідно:

– провести аналіз загроз та вразливостей медіа-сайтів, створених із застосуванням мови програмування Python;

– ідентифікувати можливі атаки та інші загрози безпеці інформації з обмеженим доступом, що циркулює через медіа-сайти;

– проаналізувати можливості захисту медіа-сайтів, створених із застосуванням мови програмування Python від ідентифікованих атак;

– дослідити реалізацію стандартного функціонального профілю (технологія Т2) для забезпечення захисту інформації від загроз;

– розробити рекомендації із захисту медіа-сайтів.

РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА

2.1 Аналіз загроз об’єкту дослідження

Модель загроз – абстрактний формалізований чи неформалізований опис методів і засобів здійснення загроз.

Мета розробки – виявлення та первинний аналіз пріоритетних напрямків побудови системи захисту інформації, відокремлення незначущих та встановлення взаємозв'язків можливих каналів витоку ІзОД.

2.1.1 Теоретичні відомості щодо аналізу загроз

У даній частині розглядаються основні поняття та принципи створення моделі загроз. Встановлюється найвищий гриф секретності інформації яка обробляється в АС.

Моделювання та класифікацію джерел загроз та їх проявів, доцільно проводити на основі аналізу взаємодії логічного ланцюга:

Рисунок 2.1 – Схема логічного ланцюга

Джерело загрози - це потенційні антропогенні, техногенні або стихійні носії загрози безпеці.

Фактор - це властиві об'єкту інформатизації причини, що призводять до порушення безпеки інформації на конкретному об'єкті і зумовлені недоліками процесу функціонування об'єкта інформатизації, властивостями архітектури АС, протоколами обміну і інтерфейсами, застосовуваними програмним забезпеченням і апаратною платформою, умовами експлуатації.

Загроза – це можлива небезпека (потенційна або реально існуюча) здійснення будь-якого діяння (дії або бездіяльності), спрямованого проти об'єкта захисту, що завдає шкоди власнику чи користувачу, який проявляється в небезпеці спотворення і втрати інформації.

Наслідки – це можливі наслідки реалізації загрози (можливі дії) при взаємодії джерела загрози через наявні фактори (уразливості).

Класифікація загроз інформаційній безпеки, приймається, що загрозами безпеки інформації є:

– розкрадання (копіювання) інформації;
– знищення інформації;
– модифікація (спотворення) інформації;
– порушення доступності (блокування) інформації;
– заперечення дійсності інформації;
– нав'язування хибної інформації.

Класифікація джерел загроз.

Носіями загроз безпеки інформації є джерела загроз. В якості джерел загроз можуть виступати, як суб'єкти (особистість) так і об'єктивні прояви. Причому, джерела загроз можуть знаходитися як всередині організації що захищаєте - внутрішні джерела, так і поза нею - зовнішні джерела.

Поділ на внутрішні і зовнішні джерела виправдано тому, що для однієї і тієї ж загрози методи парирування для зовнішніх і внутрішніх джерел можу бути різними.

Всі джерела загроз безпеці інформації можна розділити на три основні групи:

I Антропогенні джерела загроз - зумовлені діями суб'єкта.

II Техногенні джерела загроз - зумовлені якістю та особливостями роботи технічних засобів.

III Стихійні джерела загроз - зумовлені діями стихій.

Антропогенні джерела загроз.

В якості антропогенного джерела загроз можна розглядати суб'єкт, що має доступ (санкціонований або несанкціонований) до роботи зі штатними засобами безпеки. Суб'єкти (джерела), дії яких можуть призвести до порушення безпеки інформації можуть бути як зовнішні, так і внутрішні.

Зовнішні джерела можуть бути випадковими або навмисними і мати різний рівень кваліфікації. До них відносяться:

– кримінальні структури;
– потенційні злочинці і хакери;
– несумлінні партнери;
– технічний персонал, постачальники телекомунікаційних послуг;
– представники наглядових організацій та аварійних служб.

Внутрішні суб'єкти (джерела), як правило, являють собою висококваліфікованих фахівців в області розробки і експлуатації програмного забезпечення і технічних засобів, знайомі зі специфікою розв’язування завдань, структурою та основними функціями та принципами роботи програмно-апаратних засобів захисту інформації, мають можливість використання штатного обладнання та технічних засобів мережі. До них відносяться:

– основний персонал (користувачі, адміністратори, розробники);
– представники служби захисту інформації;
– допоміжний персонал (прибиральники, охорона);
– технічний персонал (життєзабезпечення, експлуатація).

Кваліфікація антропогенних джерел інформації відіграють важливу роль в оцінці їх впливу і враховується при ранжуванні джерел загроз.

Техногенні джерела загроз.

Технічні засоби, що є джерелами потенційних загроз безпеки інформації так само можуть бути зовнішніми:

– засоби зв'язку;
– мережі інженерних комунікації (водопостачання, каналізації);
– транспорт.

і внутрішніми:

– неякісні технічні засоби обробки інформації;
– неякісні програмні засоби обробки інформації;
– допоміжні засоби (охорони, сигналізації, телефонії);
– інші технічні засоби, що застосовуються в установі.

Стихійні джерела загроз.

Стихійні джерела потенційних загроз інформаційній безпеці, як правило є зовнішніми по відношенню до об'єкту, що захищається і під ними розуміються насамперед природні катаклізми:

– пожежі;
– землетрусу;
– повені;
– урагани;
– різні непередбачені обставини;
– нез'ясовні явища;
– інші форс-мажорні обставини.

2.1.2 Ранжування джерел загроз

Всі джерела загроз мають різну ступінь небезпеки $(К_{оп})_i$, яку можна кількісно оцінити, провівши їх ранжування. При цьому, оцінка ступеня небезпеки проводиться за непрямими показниками. В якості критеріїв порівняння (показників) можна, наприклад, вибрати:

– можливість виникнення джерела $(К_1)_i$ - визначає ступінь доступності до об'єкту, що захищається (для антропогенних джерел), віддаленість від об'єкта, що захищається (для техногенних джерел) або особливості обстановки (для випадкових джерел);

– готовність джерела $(К_2)_i$ - визначає ступінь кваліфікації $(К_{2.1})_i$ і привабливість $(К_{2.2})_i$ вчинення діянь з боку джерела загрози (для антропогенних джерел), або наявність необхідних умов (для техногенних та стихійних джерел);

– фатальність $(К_3)_i$ - визначає ступінь непереборності наслідків реалізації загрози.

Кожен показник оцінюється експертно-аналітичним методом за п'ятибальною системою. Причому, 1 відповідає самому мінімальному ступеню впливу оцінюваного показника на небезпеку використання джерела, а 5 - максимальному.

$(К_{оп})_i$ для окремого джерела можна визначити як відношення добутку вищенаведених показників до максимального значення – 125, за формулою (2.1):

$$(К_{оп})_i = \frac{(К_1 * К_2 * К_3)}{125}. \quad (2.1)$$

Ступінь доступності до об'єкту, що захищається може бути класифікована за наступною шкалою:

1 Відсутність доступності - антропогенний джерело загроз не має доступу до технічних засобів і програмам, що обробляють захищається інформацію.

2 Низький ступінь доступності - антропогенний джерело загроз має дуже обмежену можливість доступу до технічних засобів і програмам, що обробляють інформацію, що захищається (характерно для зовнішніх антропогенних джерел).

3 Першій середній ступінь доступності - антропогенний джерело загроз має обмежену можливість доступу до програмних засобів в силу введених обмежень у використанні технічних засобів, функціональних обов'язків або за родом своєї діяльності (характерно для внутрішніх антропогенних джерел зі звичайними правами доступу, наприклад, користувачі, або зовнішніх антропогенних джерел , що мають право доступу до засобів обробки та передачі інформації, що захищається, наприклад, хакери, технічний персонал, постачальників послуг).

4 Другій середній ступінь доступності - антропогенний джерело загроз має можливість опосередкованого, не визначеного функціональними обов'язками, (за рахунок побічних каналів витоку інформації, використання можливості доступу до привілейованих робочих місць) доступу до технічних і програмних засобів обробки інформації, що захищається (характерно для внутрішніх антропогенних джерел).

5 Висока ступінь доступності - антропогенне джерело загроз має повний доступ до технічних і програмних засобів обробки інформації, що захищається (характерно для внутрішніх антропогенних джерел, наділених максимальними правами доступу, наприклад, представники служб безпеки інформації, адміністратори).

Ступінь віддаленості від об'єкта, що захищається можна характеризувати наступними параметрами:

1 Сильно віддалені об'єкти - об'єкт захисту розташовується на значній відстані від джерел техногенних загроз, повністю виключає будь-які дії на об'єкт, що захищається, в тому числі і по вторинним проявам.

2 Віддалено розташовані об'єкти - об'єкт захисту розташований на віддаленні від джерела техногенних загроз, що виключає можливість його прямого впливу.

3 Середньо віддалені об'єкти - об'єкти захисту розташований на видаленні від джерел техногенних загроз, на якому прояв впливу цих загроз може надати не суттєвий вплив на об'єкт захисту.

4 Близько розташовані об'єкти - об'єкти захисту розташовані в безпосередній близькості від джерел техногенних загроз і будь-який прояв таких загроз може зробити істотний вплив на об'єкт, що захищається.

5 Співпадаючі об'єкти - об'єкти захисту самі містять джерела техногенних загроз і їх територіальний поділ неможливо.

Особливості обстановки характеризуються розташуванням об'єктів захисту в різних природних, кліматичних, сейсмологічних, гідрологічних та

інших умов. Особливості обстановки можна оцінити за наступною шкалою:

1 Безпечні умови - об'єкт захисту знаходиться поза межами зони дії природних катаклізмів і на об'єкті відсутні передумови виникнення стихійних джерел загроз.

2 Слабо небезпечні умови - об'єкт захисту знаходиться поза межами зони дії природних катаклізмів, проте на об'єкті є передумови виникнення стихійних джерел загроз.

3 Помірно небезпечні умови - об'єкт захисту розташований в зоні в якій по проведеним спостереженням протягом довгого періоду відсутні прояви природних катаклізмів, але є передумови виникнення стихійних джерел загроз на самому об'єкті.

4 Небезпечні умови - об'єкт захисту розташований в зоні, в якій багаторічні спостереження показують можливість прояву природних катаклізмів.

5 Дуже небезпечні умови - об'єкт захисту розташований в зоні дії природних катаклізмів.

Кваліфікація антропогенних джерел відіграє важливу роль у визначенні їх можливостей по здійсненню протиправних діянь. Прийнята наступна класифікація рівня кваліфікації по можливості (рівню) взаємодії з мережею що захищається:

1 Нульовий рівень - визначається відсутністю можливості будь-якого використання програм.

2 Перший рівень - обмежується можливістю запуску задач / програм з фіксованого набору, призначеного для обробки інформації, що захищається (рівень некваліфікованого користувача).

3 Другий рівень - враховує можливість створення і запуску користувачем власних програм з новими функціями з обробки інформації (рівень кваліфікованого користувача, програміста).

4 Третій рівень - визначається можливістю управління функціонуванням мережею, тобто впливом на базове програмне забезпечення, її склад і конфігурацію (рівень системного адміністратора).

5 Четвертий рівень - визначається всім обсягом можливостей суб'єктів, що здійснюють проєктування та ремонт технічних засобів, аж до включення до складу мережі власних технічних засобів з новими функціями з обробки інформації (рівень розробника і адміністратора).

Нульовий рівень є найнижчим рівнем можливостей з ведення діалогу джерела загроз з мережею що захищається. При оцінці можливостей антропогенних джерел передбачається, що суб'єкт, що здійснює протиправні дії, або має, або може скористатися правами відповідного рівня.

Привабливість вчинення діяння з боку джерела загроз встановлюється наступним чином:

1 Не привабливий рівень - інформація не представляє інтерес для джерела загрози.

2 Слабо привабливий рівень - захищаються інформаційні ресурси містять інформацію, яка при її накопиченні та узагальненні протягом певного періоду може завдати шкоди організації, що здійснює захист.

3 Помірно привабливий рівень - захищаються інформаційні ресурси, містять інформацію, розголошення якої може завдати шкоди окремим особам.

4 Привабливий рівень - захищаються інформаційні ресурси містять інформацію, яка може бути використана для отримання вигоди на користь джерела загрози або третіх осіб.

5 Особливо привабливий рівень - захищаються інформаційні ресурси містять інформацію, яка може завдати непоправної шкоди і призвести до краху організації, що здійснює захист.

Необхідні умови готовності джерела визначаються виходячи з можливості реалізації тієї чи іншої загрози в конкретних умовах розташування об'єкта. При цьому передбачається:

1 Загроза не можна реалізувати - тобто відсутні передумови для реалізації передбачуваного події.

2 Загроза слабо реалізовується - тобто існують об'єктивні причини на самому об'єкті або в його оточенні, що перешкоджають реалізації загрози.

3 Загроза помірно реалізовується - тобто умови сприятливі для реалізації загрози, проте довгострокові спостереження не припускають можливості її активізації в період існування та активної діяльності об'єкта захисту.

4 Загроза реалізовується - тобто умови сприятливі або можуть бути сприятливі для реалізації загрози (наприклад, активізація сейсмічної активності).

Ступінь непереборності наслідків прояву загрози (фатальність) визначається за наступною шкалою:

1 Відсутність наслідків - результати прояви загрози не можуть вплинути на діяльність об'єкта захисту.

2 Переборні наслідки - результати прояви загрози можуть призвести до часткового руйнування (знищення, втрати) об'єкта захисту, що не вимагають великих витрат на його відновлення і, практично не впливають на обмеження часу доступу до захищається інформаційних ресурсів.

3 Частково переборні наслідки - результати прояви загрози можуть призвести до часткового руйнування об'єкта захисту і, як наслідок, до значних витрат на відновлення, обмеження часу доступу до ресурсів, що захищаються.

4 Практично непереборні наслідки - результати прояви загрози можуть призвести до руйнування (знищення, втрати) об'єкта і до значних витрат (матеріальним, тимчасовим та інше) на відновлення наслідків, порівнянних з витратами на створення нового об'єкту та суттєвого обмеження часу доступу до ресурсів, що захищаються.

5 Непереборні наслідки - результати прояви загрози можуть призвести до повного руйнування (знищення, втрати) об'єкта захисту, як наслідок до непоправних втрат і виключенню можливості доступу до захищається інформаційних ресурсів.

Результати проведеного ранжирування стосовно конкретного об'єкта захисту зводяться в таблицю, що дозволяє визначити найбільш небезпечні для даного об'єкта джерела загроз безпеці інформації.

При виборі припустимого рівня джерела загроз. передбачається, що джерела загроз, що мають коефіцієнт $(К_{оп})_i$ менше за 0,1 ... 0,2 можуть надалі не враховуватися, як малоймовірні.

Визначення актуальних (найбільш небезпечних) загроз здійснюється на основі аналізу розташування об'єктів захисту і структури побудови інформаційної системи, а також інформаційних ресурсів, що підлягають захисту.

Виходячи з того що модель загроз вимагає особливого оформлення у записці кваліфікаційної роботи коректніше називати опис методів і засобів здійснення загроз як аналіз загроз, якій приведено у таблицях 2.1-2.3.

2.1.3 Антропогенні джерела загроз

Коефіцієнти розраховані за формулою 2.1, де можливість виникнення джерела $(К_1)_i$ – визначає ступінь доступності до об'єкту, що захищається, готовність джерела $(К_2)_i$ – визначає ступінь кваліфікації і привабливість вчинення діянь з боку джерела загрози, фатальність $(К_3)_i$ – визначає ступінь непереборності наслідків реалізації загрози.

Таблиця 2.1 – Джерела антропогенних загроз

№	Джерело загрози	$К_1$	$К_{2.1}$	$К_{2.2}$	$К_2$	$К_3$	Коефіцієнт
1	2	3	4	5		6	7
1	Кримінальні структури	2	1	3	2	2	0,064
2	Потенційні злочинці і хакери	3	3	3	3	3	0,216
3	Несумлінні партнери	1	1	2	1,5	2	0,024

4	Технічний персонал, постачальників телекомунікаційних послуг	2	2	2	2	0,064
			2			
5	Представники наглядових організацій та аварійних служб	3	1	2	2	0,072
			1,5			
Внутрішні джерела загроз						
6	Користувачі, яким надано право доступу тільки до загальнодоступної інформації (далі – «Користувач»)	5	4	4	4	0,64
			4			
7	Користувачі, яким надано повноваження забезпечувати керування АС (далі – «Інженер»)	5	5	3	4	0,64
			4			
8	Допоміжний персонал (прибиральники, охорона)	2	1	1	2	0,032
			1			
9	Технічний персонал (життєзабезпечення, експлуатація)	2	1	1	2	0,032

Розрахунок $К_2$ було проведено за формулою 2.2, де А – кваліфікація антропогенного джерела, а В привабливість вчинення дії.

$$(К_2)_i = \frac{A_i + B_i}{2}. \quad (2.2)$$

2.1.4 Техногенні джерела загроз

Коефіцієнти розраховані за формулою 2.2, де можливість виникнення джерела $(К_1)_i$ – визначає ступінь віддаленості від об'єкта, що захищається, готовність джерела $(К_2)_i$ – визначає ступінь наявність необхідних умов, фатальність $(К_3)_i$ – визначає ступінь непереборності наслідків реалізації загрози.

Таблиця 2.2 – Джерела техногенних загроз

№	Джерело загрози	$К_1$	$К_2$	$К_3$	Коефіцієнт
1	2	3	4	5	6
1	Засоби зв'язку	3	2	2	0,096

2	Мережі інженерних комунікації (водопостачання, каналізації)	2	2	2	0,064
3	Транспорт	1	1	2	0,016
Внутрішні джерела загроз					
4	Неякісні технічні засоби обробки інформації	4	3	3	0,288
5	Неякісні програмні засоби обробки інформації	4	3	2	0,192
6	Допоміжні засоби (охорони, сигналізації, телефонії)	3	2	2	0,096
7	Інші технічні засоби, що застосовуються на підприємстві	3	2	2	0,096

2.1.5 Стихійні джерела загроз

Коефіцієнти розраховані за формулою 2.1, де можливість виникнення джерела $(К_1)_i$ – визначає особливості обстановки, готовність джерела $(К_2)_i$ – визначає ступінь наявності необхідних умов, фатальність $(К_3)_i$ - визначає ступінь непереборності наслідків реалізації загрози.

Таблиця 2.3 – Джерела стихійних загроз

№	Джерело загрози	$К_1$	$К_2$	$К_3$	Коефіцієнт
1	Пожежі	3	3	3	0,216
2	Землетрусу	1	1	2	0,016
3	Повені	3	3	3	0,216
4	Урагани	2	2	2	0,064
5	Різні непередбачені обставини	2	1	2	0,032
6	Нез'ясовні явища	1	1	2	0,016
7	Інші форс-мажорні обставини	1	1	2	0,016

У ході аналізу джерел загроз на основі формул було виявлено, що найнебезпечнішим джерелом антропогенних загроз є основний персонал, техногенних загроз - неякісні технічні засоби обробки інформації, стихійних загроз – пожежі та повені.

Вирішено враховувати джерела загроз, коефіцієнт яких вище за 0,15, а інші відкинути як малоймовірні. [36].

2.1.6 Ранжування вразливостей

Для ранжування вразливостей було вирішено вибрати показник доступності, що визначає зручність (можливість) використання вразливості джерелом загроз (масогабаритні розміри, складність, вартість необхідних засобів, можливість використання неспеціалізованої апаратури).

Для техногенних джерел загроз. Ступінь відкритості вразливості через особливість побудови приміщення чи використання програмного і технічного забезпечення:

1 використати вразливість для атаки практично неможливо;

2 захищена вразливість;

3 вразливість майже незахищена;

4 слабо захищена вразливість;

5 вразливість відкрита.

Для антропогенних джерел загроз. Доступність до вразливості, що визначається діями працівників:

1 вразливість практично недоступна: внаслідок дії працівників використати вразливість неможливо;

2 частково недоступна вразливість: внаслідок дії працівників використати вразливість можливо, але із значними труднощами;

3 майже недоступна вразливість: внаслідок дії працівників використати вразливість можливо, але з труднощами;

4 доступна вразливість: дії працівників призводять до можливості використання вразливості;

5 високо доступна вразливість: внаслідок дії працівників відкрито використовується вразливість.

Для стихійних джерел загроз. Можливість використати вразливість через форс-мажорні обставини:

1 неможливо використати вразливість: внаслідок стихійного лиха джерело загроз не може використати вразливість;

2 практично неможливо використати вразливість: внаслідок стихійного лиха доступ до вразливості послаблюється;

3 можливо використати вразливість: внаслідок стихійного лиха вразливість стає більш доступною для джерела загроз, але реалізація атаки потребує значних витрат;

4 практично доступна вразливість: внаслідок стихійного лиха джерело загроз може використати вразливість;

5 доступна вразливість: внаслідок стихійного лиха втрачається захист вразливості, [37].

Таблиця 2.4 – Ранжування вразливостей

№	Вразливості	Показник доступності
Антропогенні джерела загрози		
1	Помилка адміністрування	4
2	Бездіяльність в аварійній ситуації	3
3	Випадкове чи навмисне завдання шкоди обладнанню	2
4	Випадкове чи навмисне завдання шкоди ПЗ	3
5	Навмисне фізичне руйнування	2
6	Порушення правил використання обладнання	5
7	Копіювання інформації	4
8	Випадкове чи навмисне відключення засобів захисту	5
9	Випадкове чи навмисне видалення інформації	5
10	Порушення правил використання антивірусного ПЗ	4
11	Втрата портативного носія інформації	4
12	Помилка при обробці інформації	3
13	Випадкове чи навмисне помилкове змінення даних	3
Техногенні джерела загрози		

14	Некоректна робота ПЗ	3
15	Некоректна робота засобів захисту	5
16	Некоректна робота операційної системи	2
17	Некоректна робота технічного компоненту АС	3
18	Вихід з ладу технічного компоненту АС	3
19	Вихід з ладу портативного носія для резервування	2
Стихійні джерела загрози		
20	Займання електричної проводки	3
21	Стихійні пожежі	4
22	Займання серверу / ПК	3
23	Підпал будівлі	4
24	Стихійні повені	4

2.1.7 Створення моделі загроз

Як було вказано вище, враховуються лише джерела загроз, коефіцієнт яких вище за 0,15.

Вплив на інформаційний ресурс може буди незначний, середній та критичний.

Незначний – інформація, що обробляється, втрачає деякі властивості, але може бути відновлена в прийнятні терміни і з мінімальними втратами.

Середній – інформація, що обробляється, втрачає свої деякі властивості, може бути відновлена, втрати підприємства в цьому випадку менше тих, які були б в результаті повної втрати і неможливості відновити інформацію.

Критичний – інформація, що обробляється, може бути знищена, змінена без можливості відновлення, втрачає свої властивості. Втрати підприємства в цьому випадку дуже великі.

Небезпечність загрози визначається добутком коефіцієнта джерела загрози та доступності вразливості.

Таблиця 2.5 – Модель загроз

№	Джерело загрози	Вразливості	Загроза	Н	К	Д	Ц	В
Антропогенні джерела загрози								
1	Інженер	Помилка адміністрування	Знищення інформації	2.56	-	+	+	1
2		Бездіяльність в аварійній ситуації	Знищення та / або спотворення інформації	1.92	-	+	+	2
3		Випадкове чи навмисне завдання шкоди обладнанню	Знищення та / або спотворення інформації	1.28	-	+	+	2
4		Випадкове чи навмисне завдання шкоди ПЗ	Знищення та / або спотворення інформації	1.92	-	+	+	2
5		Навмисне фізичне руйнування	Знищення обладнання чи / та інформації	1.28	-	+	+	2
6		Порушення правил використання обладнання	Знищення інформації	3.2	-	+	-	1
7		Копіювання інформації	Поширення інформації	2.56	+	-	-	3
8		Випадкове чи навмисне відключення засобів захисту	Поширення інформації	3.2	+	-	-	3
9		Випадкове чи навмисне видалення інформації	Знищення інформації	3.2	-	+	-	1
10		Порушення правил використання антивірусного ПЗ	Знищення та / або спотворення інформації	2.56	-	+	+	2
11	Користувач	Бездіяльність в аварійній ситуації	Знищення та / або спотворення інформації	1.92	-	+	+	1
12		Втрата портативного носія інформації	Поширення інформації	2.56	+	+	-	3
13		Помилка при обробці інформації	Спотворення інформації	1.92	-	-	+	2
14		Випадкове чи навмисне завдання шкоди обладнанню	Знищення та / або спотворення інформації	1.28	-	+	+	2
15		Випадкове чи навмисне завдання шкоди ПЗ	Знищення та / або спотворення інформації	1.92	-	+	+	

16		Навмисне фізичне руйнування	Знищення обладнання чи / та інформації	1.28	-	+	+	1
17		Порушення правил використання обладнання	Знищення інформації	3.2	-	+	-	1
18		Копіювання інформації	Поширення інформації	2.56	+	-	-	3
19		Випадкове чи навмисне видалення інформації	Знищення інформації	3.2	-	+	-	1
20		Порушення правил використання антивірусного ПЗ	Знищення та / або спотворення інформації	2.56	-	+	+	2
21		Випадкове чи навмисне помилкове змінення даних	Спотворення інформації	1.92	-	-	+	2
22	Потенційні злочинці і хакери	Завдання шкоди обладнанню	Знищення та / або спотворення інформації	0.432	-	+	+	2
23		Завдання шкоди ПЗ	Знищення та / або спотворення інформації	0.648	-	+	+	2
24		Копіювання інформації	Поширення інформації	0.864	+	-	-	3
25		Видалення інформації	Знищення інформації	1.08	-	+	-	1
26		Змінення даних	Спотворення інформації	0.648	-	-	+	1
Техногенні джерела загрози								
27	Неякісні програмні засоби обробки інформації	Некоректна робота ПЗ	Блокування інформації	0.864	-	+	-	3
28		Некоректна робота засобів захисту	Поширення інформації	1.44	+	-	-	2
29		Некоректна робота операційної системи	Блокування інформації	0.576	-	+	-	3
30	Неякісні технічні засоби обробки інформації	Некоректна робота технічного компоненту АС	Блокування інформації	0.576	-	+	-	2
31		Вихід з ладу технічного компоненту АС	Знищення інформації	0.576	-	+	-	1
32		Вихід з ладу портативного носія для резервування	Знищення інформації	0.384	-	+	-	1
Стихійні джерела загрози								
33	Пожежі	Займання електричної проводки	Знищення інформації	0.648	-	+	-	1

34		Стихійні пожежі	Знищення інформації	0.864	-	+	-	1
35		Займання серверу / ПК	Знищення інформації	0.648	-	+	-	1
36		Підпал будівлі	Знищення інформації	0.864	-	+	-	2
37	Повені	Стихійні повені	Знищення інформації	0.864	-	+	-	2

У таблиці використовуються скорочення: К – конфіденційність, Д – доступність, Ц – цілісність, де знак «+» відображає що дана властивість інформації втрачена, а «-» відображає що дана властивість інформації збережена, Н – небезпечність, В – вплив.

2.2 Обґрунтування відповідності захищеності медіа-сайту профілю захищеності (технологія Т2)

Стандартний функціональний профіль захищеності являє собою перелік мінімально необхідних рівнів послуг, які повинен реалізовувати КЗЗ обчислювальної системи АС, щоб задовольняти певні вимоги щодо захищеності інформації, яка обробляється в даній АС.

Згідно з документом НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу», в процесі оцінки спроможності комп'ютерної системи забезпечувати захист оброблюваної інформації від несанкціонованого доступу розглядаються вимоги двох видів:

– вимоги до функцій захисту (послуг безпеки);
– вимоги до гарантій.

В контексті критеріїв комп'ютерна система розглядається як набір функціональних послуг. Кожна послуга являє собою набір функцій, що дозволяють протистояти певній множині загроз. Кожна послуга може включати декілька рівнів. Чим вище рівень послуги, тим більш повно забезпечується захист від певного виду загроз. Рівні послуг мають ієрархію за повнотою захисту, проте не обов'язково являють собою точну підмножину один одного. Рівні починаються з першого (1) і зростають до значення n, де n — унікальне для

кожного виду послуг.

Функціональні критерії розбиті на чотири групи, кожна з яких описує вимоги до послуг, що забезпечують захист від загроз одного із чотирьох основних типів.

Конфіденційність. Загрози, що відносяться до несанкціонованого ознайомлення з інформацією, становлять загрози конфіденційності. Якщо існують вимоги щодо обмеження можливості ознайомлення з інформацією, то відповідні послуги треба шукати в розділі "Критерії конфіденційності". В цьому розділі описані такі послуги (в дужках наведені умовні позначення для кожної послуги): довірча конфіденційність, адміністративна конфіденційність, повторне використання об'єктів, аналіз прихованих каналів, конфіденційність при обміні (експорті/імпорті).

Цілісність. Загрози, що відносяться до несанкціонованої модифікації інформації, становлять загрози цілісності. Якщо існують вимоги щодо обмеження можливості модифікації інформації, то відповідні послуги треба шукати в розділі "Критерії цілісності". В цьому розділі описані такі послуги: довірча цілісність, адміністративна цілісність, відкат і цілісність при обміні.

Доступність. Загрози, що відносяться до порушення можливості використання комп'ютерних систем або оброблюваної інформації, становлять загрози доступності. Якщо існують вимоги щодо захисту від відмови в доступі або захисту від збоїв, то відповідні послуги треба шукати в розділі "Критерії доступності". В цьому розділі описані такі послуги: використання ресурсів, стійкість до відмов, горяча заміна, відновлення після збоїв.

Спостереженість. Ідентифікація і контроль за діями користувачів, керованість комп'ютерною системою становлять предмет послуг спостереженості і керованості. Якщо існують вимоги щодо контролю за діями користувачів або легальністю доступу і за спроможністю комплексу засобів захисту виконувати свої функції, то відповідні послуги треба шукати у розділі "Критерії спостереженості". В цьому розділі описані такі послуги: реєстрація,

ідентифікація і автентифікація, достовірний канал, розподіл обов'язків, цілісність комплексу засобів захисту, самотестування, автентифікація при обміні, автентифікація відправника (невідмова від авторства), автентифікація одержувача (невідмова від одержання).

Крім функціональних критеріїв, що дозволяють оцінити наявність послуг безпеки в комп'ютерній системі, цей документ містить критерії гарантій, що дозволяють оцінити коректність реалізації послуг. Критерії гарантій включають вимоги до архітектури комплексу засобів захисту, середовища розробки, послідовності розробки, випробування комплексу засобів захисту, середовища функціонування і експлуатаційної документації. В цих Критеріях вводиться сім рівнів гарантій (Г-1, ..., Г-7), які є ієрархічними.

Як зазначено у документі НД ТЗІ 2.5-010-03 «Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу», до складу АС, яка забезпечує функціонування WEB-сторінки, входять: ОС, фізичне середовище, в якому вона знаходиться і функціонує, середовище користувачів, оброблювана інформація, у тому числі й технологія її оброблення. Під час забезпечення захисту інформації мають бути враховані всі характеристики зазначених складових частин, які впливають на реалізацію політики безпеки WEB-сторінки, [5].

З урахуванням особливостей надання доступу до інформації WEB-сторінки, типових характеристик середовищ функціонування та особливостей технологічних процесів оброблення інформації, зазначених у розділі 6 документа НД ТЗІ 2.5-010-03, визначаються наступні мінімально необхідні рівні послуг безпеки для забезпечення захисту інформації від загроз:

– за умови, коли WEB-сервер і робочі станції розміщуються на території установи-власника WEB-сторінки або на території оператора (технологія Т1), мінімально необхідний функціональний профіль визначається:

КА-2, ЦА-1, ЦО-1, ДВ-1, ДР-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1;

– за умови, коли WEB-сервер розміщується у оператора, а робочі станції

– на території власника WEB-сторінки, взаємодія яких з WEB-сервером здійснюється з використанням мереж передачі даних (технологія Т2), мінімально необхідний функціональний профіль визначається:

КА-2, КВ-1, ЦА-1, ЦО-1, ЦВ-1, ДВ-1, ДР-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1, НВ-1.

Технологія Т1 різниться від технології Т2 способом передачі інформації від робочої станції до WEB-сервера, а саме: наявністю у другому випадку незахищеного середовища, яке не контролюється, і додатковими вимогами щодо ідентифікації та автентифікації між КЗЗ робочої станції й КЗЗ WEB-сервера під час спроби розпочати обмін інформацією та забезпечення цілісності інформації при обміні.

При функціонуванні медіа-сайтів, як правило, використовується технологія Т2, тому для неї буде проведений аналіз реалізації.

2.2.1 Вимоги до реалізації функціональних послуг безпеки інформації

Базова адміністративна конфіденційність

КЗЗ повинен реалізувати рівень КА-2.

Ця послуга дозволяє адміністратору безпеки керувати потоками інформації від захищених об'єктів до користувачів.

Політика адміністративної конфіденційності стосується: користувачів усіх категорій, крім визначених згідно з 6.3.1 "а"; об'єктів, що містять технологічну інформацію системи захисту та технологічну інформацію щодо управління АС; системного та функціонального програмного забезпечення, що використовується для актуалізації, захисту загальнодоступної інформації та супроводження WEB-сторінки; доступу користувачів до окремих видів периферійних пристроїв (принтерів, накопичувачів інформації тощо), використання яких передбачено технологією обробки інформації.

КЗЗ повинен здійснювати розмежування доступу на підставі атрибутів доступу користувача і захищеного об’єкта. Доступ до загальнодоступної

інформації встановлюється для користувачів усіх категорій. Призначення атрибутів доступу користувачам і процесам до захищених об'єктів здійснюється адміністратором безпеки на основі аналізу функціональних та службових обов'язків окремих користувачів.

КЗЗ повинен надавати тільки адміністратору безпеки права доступу до технологічної інформації системи захисту та процесів, що забезпечують її актуалізацію, супроводження та аналіз. Доступ до процесів, що забезпечують ведення системних процесів з адміністрування й забезпечення функціонування АС в цілому, окремих її компонентів та сервісів, а також до технологічної інформації щодо управління АС повинен надаватись тільки користувачам, які мають відповідні повноваження.

Запити на зміну прав доступу повинні оброблятися КЗЗ тільки в тому випадку, якщо вони надходять від адміністратора безпеки.

Права доступу до кожного захищеного об'єкта, визначеного політикою безпеки послуги, повинні встановлюватися в момент його створення або ініціалізації.

Конфіденційність при обміні

КЗЗ повинен реалізувати рівень КВ-1.

Ця послуга дозволяє забезпечити захист об’єктів від несанкціонованого ознайомлення з інформацією, що міститься в них, під час їх експорту/імпорту через незахищене середовище.

Політика мінімальної конфіденційності при обміні стосується: користувачів, яким надано право супроводження системи захисту та управління АС; об’єктів, які містять технологічну інформацію системи захисту та технологічну інформацію щодо управління АС під час її передавання між віддаленими компонентами АС.

КЗЗ повинен забезпечувати захист від безпосереднього ознайомлення з інформацією, що міститься в об’єкті, який передається.

КЗЗ повинен забезпечувати можливість реєстрації подій, які призвели або

можуть призвести до порушення конфіденційності інформації, що міститься в об’єктах, які передаються.

Мінімальна адміністративна цілісність

КЗЗ повинен реалізувати рівень ЦА-1.

Ця послуга дозволяє керувати потоками інформації від користувачів до захищених об’єктів WEB-сторінки.

Політика мінімальної адміністративної цілісності стосується: користувачів усіх категорій; загальнодоступної інформації WEB-сторінки; файлової системи та функціонального ПЗ, що використовується для актуалізації, захисту загальнодоступної інформації та супроводження WEB-сторінки; створеної в процесі супроводження WEB-сторінки технологічної інформації системи захисту та технологічної інформації щодо управління АС.

КЗЗ повинен здійснювати розмежування доступу на підставі атрибутів доступу користувачів і захищених об’єктів. Розмежування доступу здійснюється на рівні надання (встановлення заборони) користувачеві прав модифікувати об’єкт.

Право визначати множину об'єктів АС, цілісність яких забезпечується КЗЗ, надається адміністратору безпеки.

КЗЗ повинен надавати можливість адміністратору безпеки для кожного захищеного об’єкта визначити домен, якому повинні належати ті користувачі і/або групи користувачів, що мають право модифікувати об’єкт. Тільки йому надається право включати і вилучати користувачів та об'єкти до/з конкретних доменів.

Призначення атрибутів доступу користувачам і процесам до захищених об'єктів та запити на зміну цих прав повинні оброблятися КЗЗ тільки в тому випадку, якщо вони надходять від адміністратора безпеки.

Користувачам, які мають доступ тільки до загальнодоступної інформації WEB-сторінки, забороняється модифікувати будь-які захищені об’єкти.

Адміністратору безпеки надається право модифікувати функціональне ПЗ,

що використовується для захисту загальнодоступної інформації, та технологічну інформацію системи захисту. Користувачам, що мають повноваження щодо управління АС, надається відповідно до функціональних обов'язків право модифікувати технологічну інформацію та функціональне ПЗ, що використовується для актуалізації загальнодоступної інформації та супроводження WEB-сторінки.

Права доступу до захищених об'єктів WEB-сторінки повинні встановлюватися в момент їх створення або ініціалізації.

Цілісність при обміні

КЗЗ повинен реалізувати рівень ЦВ-1.

Ця послуга дозволяє забезпечити захист WEB-сторінки від несанкціонованої модифікації інформації, яка передається між WEB-сервером та робочими станціями у разі використання технології Т2, під час експорту/імпорту інформації через незахищене середовище. Політика послуги стосується всіх об'єктів, що передаються.

КЗЗ повинен забезпечувати контроль за цілісністю інформації в повідомленнях, які передаються, а також бути здатним виявляти факти їх несанкціонованого видалення або дублювання.

КЗЗ повинен забезпечувати можливість реєстрації подій, які призвели до порушення цілісності повідомлень, їх несанкціонованого видалення або дублювання.

Відкат

КЗЗ повинен реалізувати рівень ЦО-1.

Ця послуга забезпечує можливість відмінити окрему операцію або послідовність операцій і повернути захищений об'єкт після внесення до нього змін до попереднього наперед визначеного стану.

Політика обмеженого відкату стосується користувачів, яким надано право супроводження системи захисту та управління АС; об'єктів, які містять публічну інформацію; функціонального програмного забезпечення, що використовується

для актуалізації, захисту публічної інформації та супроводження WEB-сторінки; створеної в процесі супроводження WEB-сторінки технологічної інформації системи захисту та технологічної інформації щодо управління АС. Якщо стосовно якогось з об'єктів зазначених категорій в процесі обробки не передбачається можливості його модифікації, політика послуги на нього не розповсюджується.

До складу АС повинні входити автоматизовані засоби, які дозволяють адміністратору безпеки, співробітнику СЗІ, користувачу, який має повноваження щодо управління АС, відкатити або відмінити певний набір (множину) операцій, виконаних над захищеним об'єктом WEB-сторінки за певний проміжок часу.

Факт використання послуги має реєструватись в системному журналі. Відміна операції не повинна призводити до видалення з журналу запису про операцію, яка пізніше була відмінена, якщо остання підлягала реєстрації відповідно до вимог послуги безпеки НР-2.

Використання ресурсів

КЗЗ повинен реалізувати рівень ДР-1.

Ця послуга дозволяє керувати використанням користувачами послуг та ресурсів.

Політика використання ресурсів, що реалізується КЗЗ, стосується: користувачів загальнодоступної інформації; адміністратора безпеки та користувачів, яким надано повноваження щодо управління АС; файлової системи; системного та функціонального програмного забезпечення; технологічної інформації щодо управління АС; окремих периферійних пристроїв (принтерів, накопичувачів інформації і т.ін.); обчислювальних ресурсів АС і передбачає можливість встановлення обмежень на їх використання.

Обмеження щодо використання окремим користувачем та/або процесом обсягів обчислювальних ресурсів АС або кількості об'єктів встановлюються адміністратором безпеки або користувачами, яким надано повноваження щодо управління АС. Запити на зміну встановлених обмежень повинні оброблятися

КЗЗ тільки в тому випадку, якщо вони надходять від зазначених користувачів.

Спроби користувачів перевищити встановлені обмеження на використання ресурсів повинні реєструватися в системному журналі.

Відновлення після збоїв

КЗЗ повинен реалізувати рівень ДВ-1.

Політика відновлення після збоїв, що реалізується КЗЗ, стосується: системного та функціонального програмного забезпечення; засобів захисту інформації та засобів управління системи захисту; засобів адміністрування та управління обчислювальною системою АС – і гарантує повернення АС у відомий захищений стан після відмов або переривання обслуговування, спричинених помилковими діями користувачів, неврахованою функціональною недостатністю програмного та апаратного забезпечення (наприклад, можливою наявністю не виявлених під час проєктування незадекларованих функцій), іншими непередбачуваними ситуаціями.

Політика відновлення, яка реалізується КЗЗ, повинна визначати множину типів відмов WEB-сторінки і переривань обслуговування, після яких можливе повернення у відомий захищений стан без порушення політики безпеки. Для кожної з відмов повинні бути чітко визначені і задокументовані рівні відмов, у разі перевищення яких необхідна повторна інсталяція WEB-сторінки.

Після відмови WEB-сторінки або переривання обслуговування, КЗЗ повинен перевести WEB-сторінку до стану, з якого повернути її в режим нормального функціонування може тільки адміністратор безпеки і користувачі, які мають повноваження щодо управління АС. Для кожного з них повинна бути визначена множина допустимих виконуваних ними операцій з метою повернення АС у відомий захищений стан.

Повернення АС з режиму, що визначається погіршеними характеристиками обслуговування, в режим нормального функціонування повинно здійснюватися із застосуванням ручних (не автоматизованих) процедур.

Реєстрація

КЗЗ повинен реалізувати рівень HP-2.

Послуга дозволяє контролювати небезпечні відповідно до політики безпеки WEB-сторінки дії користувачів всіх категорій із захищеними об'єктами.

Політика реєстрації стосується: користувачів усіх категорій; публічної інформації WEB-сторінки; системного та функціонального програмного забезпечення, що використовується для актуалізації, захисту публічної інформації та супроводження WEB-сторінки; створеної в процесі супроводження WEB-сторінки технологічної інформації системи захисту та технологічної інформації щодо управління АС.

КЗЗ повинен забезпечувати реєстрацію всіх подій, які мають безпосереднє відношення до безпеки. До них відносяться наступні класи подій:

– вхід/вихід або намагання входу/виходу в/із системи користувачами будь-яких категорій;

– реєстрація та видалення або намагання реєстрації та видалення користувачів будь-якої категорії в системі;

– зміна атрибутів доступу користувачем будь-якої категорії та дії, що призвели до цього;

– отримання або намагання отримання доступу користувачем будь-якої категорії до будь-яких захищених процесів і об'єктів АС;

– створення користувачем будь-якої категорії твердих копій та виведення їх на друкуючі пристрої;

– модифікація або спроби модифікації захищених процесів і об'єктів АС, у тому числі факти та спроби порушення цілісності КЗЗ;

– спроби використання обчислювальних ресурсів АС з перевищенням встановлених квот;

– інші події, обов'язковість реєстрації яких передбачена політикою реалізації окремих послуг безпеки інформації.

КЗЗ повинен надавати можливість визначення переліку реєстраційних подій виключно адміністратору безпеки.

Реєстрація всіх подій, що мають безпосереднє відношення до безпеки, здійснюється в журналі реєстрації, який повинен містити інформацію стосовно дати, часу, місця, типу і наслідків зареєстрованої події (успішність/неуспішність), ім'я (IP-адресу) та/або ідентифікатор причетного до цієї події користувача. Реєстраційна інформація повинна бути достатньою для однозначної ідентифікації користувача, процесу і/або об'єкта, що мали відношення до кожної зареєстрованої події.

КЗЗ повинен мати механізми захисту для гарантування безпечної передачі інформації журналу реєстрації на віддалену робочу станцію адміністратора безпеки WEB-сторінки (для технології Т2).

Адміністратор безпеки і користувачі, яким надано повноваження щодо управління АС, повинні мати засоби перегляду та аналізу журналу реєстрації, а КЗЗ повинен забезпечувати захист журналу реєстрації від несанкціонованого доступу, модифікації або руйнування.

Ідентифікація і автентифікація

КЗЗ повинен реалізувати рівень НИ-2.

Ідентифікація і автентифікація дозволяють КЗЗ визначити і перевірити особу суб'єкта, що намагається одержати доступ до захищених об'єктів WEB-сторінки.

Політика ідентифікації і автентифікації стосується: всіх користувачів WEB-сторінки, які намагаються одержати доступ до системного та функціонального програмного забезпечення, що використовується для актуалізації, захисту публічної інформації та супроводження WEB-сторінки; створеної в процесі супроводження WEB-сторінки технологічної інформації системи захисту та технологічної інформації щодо управління АС; задіяного для цього периферійного обладнання.

КЗЗ повинен однозначно ідентифікувати категорії користувачів WEB-сторінки і за атрибутами кожної з цих категорій визначати послуги, що їм доступні. Ідентифікація здійснюється на підставі особистого імені та/або IP-

адреси користувача.

КЗЗ повинен автентифікувати адміністратора WEB-сторінки, співробітників СЗІ та користувачів, які мають повноваження щодо управління АС, з використанням захищеного механізму на підставі особистого пароля. Автентифікація користувачів, що мають виключне право доступу тільки до публічної інформації, не здійснюється.

Дозвіл на виконання будь-яких дій з інформацією та обладнанням WEB-сторінки, що контролюються КЗЗ, надається користувачу тільки після успішного завершення процедур ідентифікації та/або автентифікації його КЗЗ відповідно до категорії користувача.

КЗЗ повинен забезпечувати захист даних автентифікації від несанкціонованого доступу, модифікації або руйнування.

Ідентифікація і автентифікація при обміні

КЗЗ повинен реалізувати рівень НВ-1.

Ця послуга дозволяє у разі використання технології Т2 компонентам КЗЗ WEB-сервера і віддаленої робочої станції здійснити взаємну ідентифікацію, перш ніж розпочати взаємодію.

Послуга ідентифікації і автентифікації при обміні стосується адміністратора безпеки та користувачів, яким надані повноваження щодо супроводження WEB-сторінки, технологічної інформації системи захисту.

КЗЗ повинен надавати доступ до процесів, що забезпечують ініціалізацію обміну даними, тільки адміністратору безпеки і користувачам, яким надано повноваження щодо супроводження WEB-сторінки.

Обмін інформацією між компонентами КЗЗ повинен здійснюватися тільки після ідентифікації і автентифікації КЗЗ-відправником КЗЗ-отримувача інформації. Результати процедури ідентифікації і автентифікації є дійсними протягом всього сеансу обміну (незалежно від кількості об’єктів, що експортуються) і втрачають свою силу після його закінчення.

Процедура ідентифікації і автентифікація компонентів КЗЗ повинна

здійснюватися на підставі їхніх імен, IP-адрес і паролів.

Підтвердження ідентичності має виконуватися на підставі затвердженого в АС протоколу автентифікації.

Достовірний канал

КЗЗ повинен реалізувати рівень НК-1.

Ця послуга повинна гарантувати користувачу будь-якої категорії можливість безпосередньої взаємодії з КЗЗ, а також те, що ніяка взаємодія користувача з АС не може бути модифікованою іншим користувачем або процесом. Послуга визначає вимоги до механізму встановлення достовірного зв'язку між користувачем і КЗЗ.

Політика достовірного каналу стосується користувачів усіх категорій та компонентів системного та функціонального програмного забезпечення, які задіяні для реалізації механізмів КЗЗ.

Достовірний канал повинен використовуватися для початкової ідентифікації і автентифікації. Зв'язок з використанням даного каналу повинен ініціюватися виключно користувачем.

Розподіл обов'язків

КЗЗ повинен реалізувати рівень НО-1.

Ця послуга дозволяє розмежувати повноваження користувачів, визначивши категорії користувачів з певними і притаманними для кожної з категорій функціями (ролі). Послуга призначена для зменшення потенційних збитків від навмисних або помилкових дій користувачів і обмеження авторитарності керування АС.

Політика розподілу обов'язків, що реалізується КЗЗ, стосується користувачів усіх категорій і повинна визначати щонайменше такі ролі:

– адміністратора безпеки;

– користувачів, яким надано право доступу до певних видів інформації (публічної, технологічної, системного та функціонального ПЗ).

Кількість користувачів, які мають доступ до технологічної інформації та

системного і функціонального ПЗ повинна бути мінімізована, щоб обмежити їх коло тільки тими, кому необхідний такий доступ для виконання функціональних обов'язків, що передбачаються експлуатаційною та розпорядчою документацією на WEB-сторінку.

Адміністратору безпеки дозволяється доступ до всієї інформації WEB-сторінки. У разі необхідності його роль може дублюватися уповноваженим співробітником СЗІ. Повноваження всіх інших користувачів щодо доступу до інформації надаються їм адміністратором безпеки.

КЗЗ повинен присвоїти користувачу атрибути, якими однозначно характеризується надана йому роль. Користувач може виступати в певній ролі тільки після того, як він виконає дії, що підтверджують прийняття ним цієї ролі.

Цілісність комплексу засобів захисту

КЗЗ повинен реалізувати рівень НЦ-1.

Ця послуга визначає міру здатності КЗЗ WEB-сторінки захищати себе і гарантувати свою спроможність керувати захищеними об’єктами.

Політика цілісності КЗЗ повинна визначати склад КЗЗ, механізми контролю цілісності його компонентів та порядок їх використання.

Політика цілісності КЗЗ стосується: адміністратора безпеки; окремих компонентів системного та функціонального програмного забезпечення, які задіяні для реалізації механізмів КЗЗ; засобів захисту інформації, а також технологічної інформації системи захисту - і забезпечує взаємодію зазначених об’єктів.

Політика реалізації послуги повинна гарантувати, що всі послуги безпеки доступні тільки через інтерфейс КЗЗ і всі запити на доступ до захищених об'єктів контролюються КЗЗ. Якщо існують обмеження, недотримання яких може призвести до надання послуг в обхід інтерфейсу КЗЗ і порушення цілісності КЗЗ, то такі обмеження повинні бути описані і задокументовані. До користувачів має бути доведено порядок їх роботи з дотриманням цих обмежень, а КЗЗ повинен надавати адміністратору можливість здійснення контролю за цим порядком.

КЗЗ повинен повідомляти адміністратора безпеки про порушення цілісності будь-якого компонента КЗЗ. WEB-сторінка під час цього має бути переведена до стану, в якому доступ до неї користувачів, крім адміністратора безпеки, заборонено. Повернення до нормального режиму функціонування може бути здійснено тільки адміністратором після відновлення відповідності цього компонента еталону.

Самотестування

КЗЗ повинен реалізувати рівень НТ-1.

Самотестування дозволяє КЗЗ перевірити і на підставі цього гарантувати правильність функціонування і цілісність певної множини функцій захисту WEB-сторінки.

Політика самотестування поширюється на адміністратора безпеки, компоненти системного та функціонального програмного забезпечення, які задіяні для реалізації механізмів КЗЗ, засоби захисту інформації.

До складу КЗЗ повинна входити множина тестових процедур, яка враховує особливості функціонування компонентів конкретної WEB-сторінки і достатня для оцінки правильності виконання всіх критичних для безпеки публічної та технологічної інформації системи захисту функцій, а сам КЗЗ повинен бути здатним контролювати їх виконання.

КЗЗ повинен забезпечувати виконання тестів за запитом адміністратора безпеки.

У разі некоректного виконання якогось із тестів КЗЗ повинен перевести АС до стану, коли забороняється надання користувачам доступу до WEB-сторінки, або до стану, коли забороняється надання доступу до інформації з використанням функцій, для яких тест не було виконано. Повернути АС до нормального функціонування може тільки адміністратор безпеки після відновлення працездатності КЗЗ і повторного виконання повного набору тестів.

КЗЗ повинен забезпечувати відповідність набору тестів (неможливість будь-якої модифікації) версії КЗЗ. Зміна тестів можлива лише у процесі

інсталяції нової версії КЗЗ.

2.2.2 Вимоги до реалізації критеріїв гарантій

Критерії гарантій включають вимоги до архітектури КЗЗ, середовища розробки, послідовності розробки, середовища функціонування, документації, випробувань КЗЗ.

Гарантії реалізації послуг безпеки повинні відповідати рівню Г2 у відповідності до НД ТЗІ 2.5-004, [40].

Таблиця 2.6 - Реалізація профілю захищеності

Критерій	Реалізація
КА-2 Базова адміністративна конфіденційність	Не реалізується.
КВ-1 Конфіденційність при обміні	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Захист обміну даними”, що забезпечує захист вмісту об’єктів від несанкціонованого ознайомлення та модифікації під час експорту/імпорту через незахищене середовище.
ЦА-1 Мінімальна адміністративна цілісність	Не реалізується.
ЦВ-1 Цілісність при обміні	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Захист обміну даними”, що забезпечує захист вмісту об’єктів від несанкціонованого ознайомлення та модифікації під час експорту/імпорту через незахищене середовище.
ЦО-1 Відкат	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Відновлення системи після збоїв та відкат”, що включає в себе компонент відкату, тобто відміни певної операції або послідовності операцій зі збереженням цілісності даних користувача.
ДР-1 Використання ресурсів	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Використання ресурсів”, що включає логічно відокремлені функціональні компоненти щодо використання дискових квот та пріоритетів виконання процесів.

Продовження таблиці 2.6

1	2
ДВ-1 Відновлення після збоїв	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Відновлення системи після збоїв та відкат”, що включає в себе компонент відновлення системи після збоїв і повернення ОЕ до відомого захищеного стану після відмови або переривання обслуговування.
НР-2 Реєстрація	Windows Server Microsoft Windows Server 2022 Реалізується сервісом Сервіс ”Реєстрація", що включає логічно відокремлені функціональні компоненти щодо реєстрації подій, перегляду журналів аудиту та захисту журналу аудиту від переповнення.
НИ-2 Ідентифікація і автентифікація	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Ідентифікація та автентифікація”, що включає наступні логічно відокремлені функціональні компоненти: – база даних атрибутів користувачів; – варіанти входу в систему; – достовірний канал; – процес входу в систему і вибір протоколу автентифікації; – імперсонація.
НВ-1 Ідентифікація і автентифікація при обміні	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Захист обміну даними”, що забезпечує захист вмісту об’єктів від несанкціонованого ознайомлення та модифікації під час експорту/імпорту через незахищене середовище.
НК-1 Достовірний канал	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Ідентифікація та автентифікація”, що включає наступні логічно відокремлені функціональні компоненти: – база даних атрибутів користувачів; – варіанти входу в систему; – достовірний канал; – процес входу в систему і вибір протоколу автентифікації; – імперсонація.

Продовження таблиці 2.6

1	2
НО-1 Розподіл обов'язків	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Керування безпекою", що включає наступні логічно відокремлені функціональні компоненти: – підтримка в системі ролей користувачів; – механізм „Групова політика"; – керування функціями безпеки ОЕ.
НЦ-1 Цілісність комплексу засобів захисту	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Захист КЗЗ", що включає наступні логічно відокремлені функціональні компоненти: – захист при старті; – розподіл доменів; – контролю доступу до об'єктів; – захист файлів Windows; – перевірка цифрових підписів системних файлів; – обмеження ПЗ.
НТ-1 Самотестування	Windows Server Microsoft Windows Server 2022 Реалізується сервісом „Захист КЗЗ", що включає наступні логічно відокремлені функціональні компоненти: – захист при старті; – розподіл доменів; – контролю доступу до об'єктів; – захист файлів Windows; – перевірка цифрових підписів системних файлів; обмеження ПЗ. [41].

2.3 Проблеми захисту інформації у web-додатках, створених із застосуванням мови програмування Python

Інформаційна безпека реалізується впровадженням системи безпеки. Проблема захисту інформації є багатоплановою і комплексною та охоплює ряд важливих завдань.

Основні проблеми захисту інформації при роботі в комп'ютерних системах можна умовно розділити на три типи:

– перехоплення інформації (порушення конфіденційності інформації);

– модифікація інформації (спотворення вихідного повідомлення або заміна іншою інформацією);

– підміна авторства (крадіжка інформації і порушення авторського права), [42].

Захист web-додатків стає критично важливою для безпеки Internet ресурсів.

Web-додатки мають такі важливі переваги, як простота і звичність інтерфейсу, можливість віддаленої роботи через мережу Інтернет, швидкість розробки програми. Разом з цим web-додатки створюють велике число проблем, пов'язаних з забезпечення інформаційної безпеки, адже їх розробка дуже часто виконується в стислі терміни, а додаток стає доступним через мережу Інтернет і для користувачів, і для зловмисників. Вразливості дозволяють зловмисникам викрадати конфіденційну інформацію, проводити несанкціоновані зміни даних, порушувати доступність додатка. В даний час проблема забезпечення безпеки web-додатків досить актуальна, так, більше 60% від усіх виявлених вразливостей відносяться до web-додатків, [43].

В даний час для розробки web-додатків застосовується велика кількість різноманітних інструментальних засобів, і мов програмування. Найбільш популярними серед них є PHP, Perl, Python, Ruby, Java/JSP, .Net/ASP, CGI. Всі технології, крім CGI, побудовані на основі інтерпретованих мовах програмування. Щодо CGI, то це – інтерфейс зв'язку web-сервера з програмою, що запускається цим сервером і створює динамічний документ. CGI-програма може бути написана на будь-якій мові програмування, головне, щоб вона могла бути виконана сервером. Найчастіше використовуються інтерпретована мова програмування (PHP, Perl, Python) зважаючи на багато причин. Це простота

написання та зрозумілість коду, відсутність необхідності в компіляції, але в той же час надзвичайна гнучкість і достатня для CGI потужність.

При створенні і підтримці web-додатку, не має значення, якою мовою він був реалізований, неможливо не зіткнутися з проблемою захисту інформації.

Замовники проєктів не зацікавлені в тому, щоб їх інформація була вкрадена, спотворена, або була порушена її конфіденційність. Тому розробники мов програмування працюють над тим, щоб web-додатки, які розробляються із застосуванням їх продуктів, були максимально захищені. Для цього розробники з кожною новою версією вдосконалюють мови програмування, додаючи все нові можливості для захисту інформації тим, хто розробляє web-додатки із застосуванням цих мов програмування. Кожен виробник реалізував алгоритми, які, на його погляд, є найбільшим захистом для безпеки web-додатків.

Але, ніякі алгоритми не є досконалими. Абсолютно кожна мова програмування має свої недоліки, в тому числі і в питанні захисту інформації.

Вразливості в web-додатках часто виникають через недосконаліть коду, написаного на скриптових мовах.

Основні вразливості мови програмування Python та web-додатків, написаних з їх використанням:

- підключення зовнішніх файлів;
- використання глобальних змінних;
- впровадження команд (SQL-ін'єкція, SSI ін'єкція та інші);
- вразливості генераторів псевдовипадкових чисел (ГПВЧ);
- міжсайтовий скриптінг;
- міжсайтова підробка запиту;
- CGI-вразливості;
- відсутність безпечного стирання пам’яті.

Програма експлуататор (комп'ютерна програма, фрагмент програмного коду або послідовність команд) використовує вразливості в програмному забезпеченні для проведення атаки на обчислювальну систему, [44].

Метою атаки може бути як захоплення контролю над системою (підвищення привілеїв), так і порушення її функціонування (DoS-атака).

Інформація, отримана в результаті виявлення вразливості, може бути використана як для написання нової програми експлуататора, так і для усунення вразливості. Тому в ній однаково зацікавлені обидві сторони — і зловмисник, і виробник програмного забезпечення, що зламують. Характер поширення цієї інформації визначає час, який потрібен розробнику до випуску оновлення.

Після нейтралізації вразливості виробником шанс успішного застосування програми експлуататора починає стрімко зменшуватися. Тому особливою популярністю серед хакерів користуються так звані «0day програми експлуататори», що використовують вразливості, що недавно з'явилися та ще не стали відомі.

Інтерпретована мова програмування може генерувати вихідний код під час виконання і відразу передавати його до інтерпретатора.

Це означає, що web-додатки, написані з їх допомогою особливо схильні до атак, заснованим на впровадженні коду.

2.4 Аналіз можливих атак на медіа-сайти, створені із застосуванням мови програмування Python

Далі буде приведено характеристику найпоширеніших атак, що можна провести на медіа-сайти, створені із застосуванням мови програмування Python, та методи захисту від подібних атак.

2.4.1 Міжсайтовий скриптінг (XSS атака)

XSS вразливості дозволяють користувачеві вставити власні JS скрипти в браузери інших користувачів. Це досягається із застосуванням збереження шкідливих скриптів в базі даних, які потім запитуються і відображаються

браузерами інших користувачів, або примушення користувача натиснути на посилання, яке дозволить шкідливому скрипту виконатися в браузері користувача. Однак, XSS атаки можуть відбуватися з будь-якого недовірених джерела даних, такого як cookies або web-сервіси, в разі, коли дані були недостатньо очищені перед їх розміщенням на сторінці.

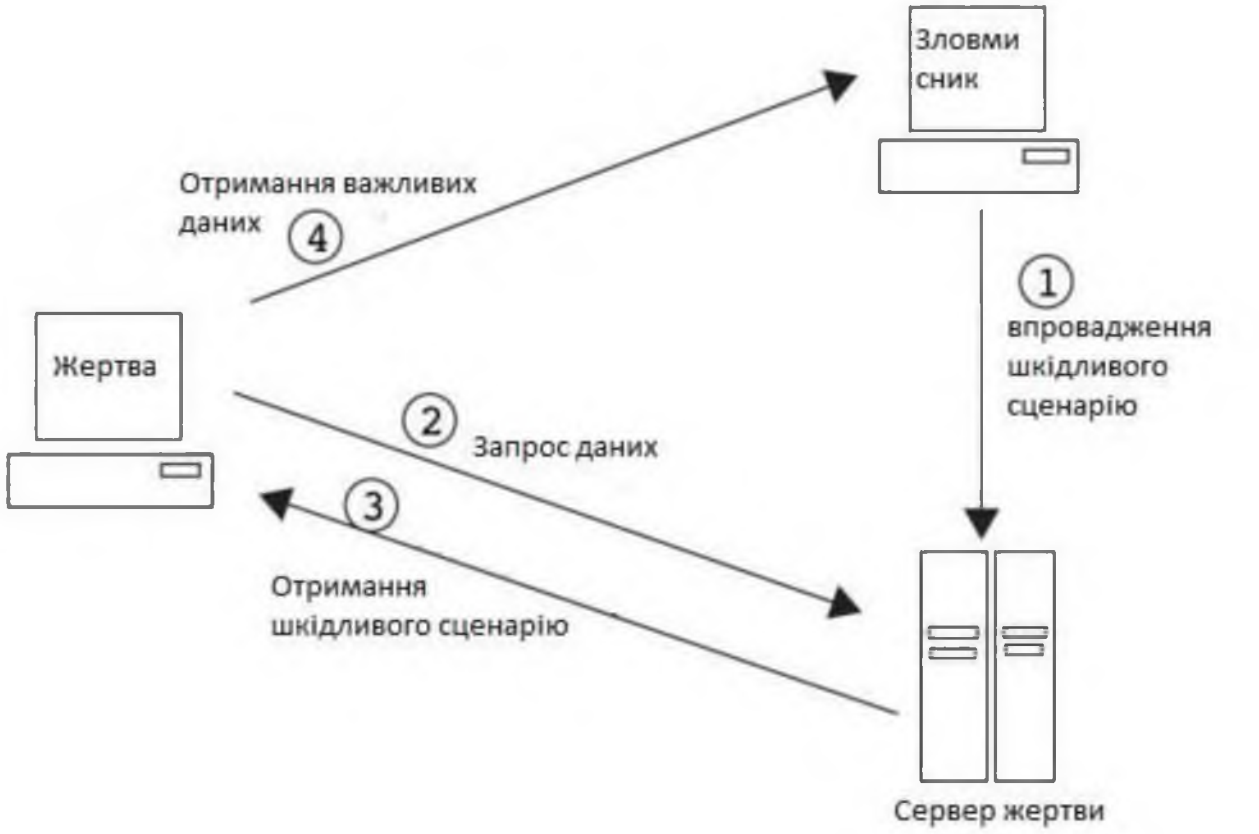

Рисунок 2.2 – схема XSS атаки

Захист: Використання шаблонів фреймворків мови програмування Python забезпечує захист від більшості XSS атак. Із застосуванням цих шаблонів екрануються спеціальні символи, які могли б якось вплинути на отриманий HTML. Але незважаючи на те, що екранування може захистити від багатьох видів шкідливого введення, воно не дає стовідсоткового захисту.

Треба фільтрувати дані, які вводяться та виводяться, а також всі поля, які можуть змінюватися користувачами. Сюди відносяться дані, одержувані із запитів GET і POST, а також запити, що повертаються з бази даних.

Також потрібно бути дуже обережним при збереженні HTML в базі даних, особливо якщо цей HTML буде відображений згодом.

Ще одним засобом захисту буде використання заголовка ContentSecurityPolicy, що дозволяє задавати список, до якого заносяться бажані джерела, з яких можна довантажувати різні дані, наприклад, JS, CSS, зображення та інше.

2.4.2 Міжсайтова підробка запиту (CSRF атака)

CSRF вразливість використовує недоліки протоколу HTTP. Якщо жертва заходить на сайт, створений зловмисником, від її особи таємно відправляється запит на інший сервер (наприклад, на сервер платіжної системи), який здійснює якусь шкідливу операцію (наприклад, переказ грошей на рахунок зловмисника). Для здійснення даної атаки жертва повинна бути автентифікована на тому сервері, на який відправляється запит, і цей запит не повинен вимагати будь-якого підтвердження з боку користувача, який не може бути проігнорований або підроблений атакуючим скриптом, [45].

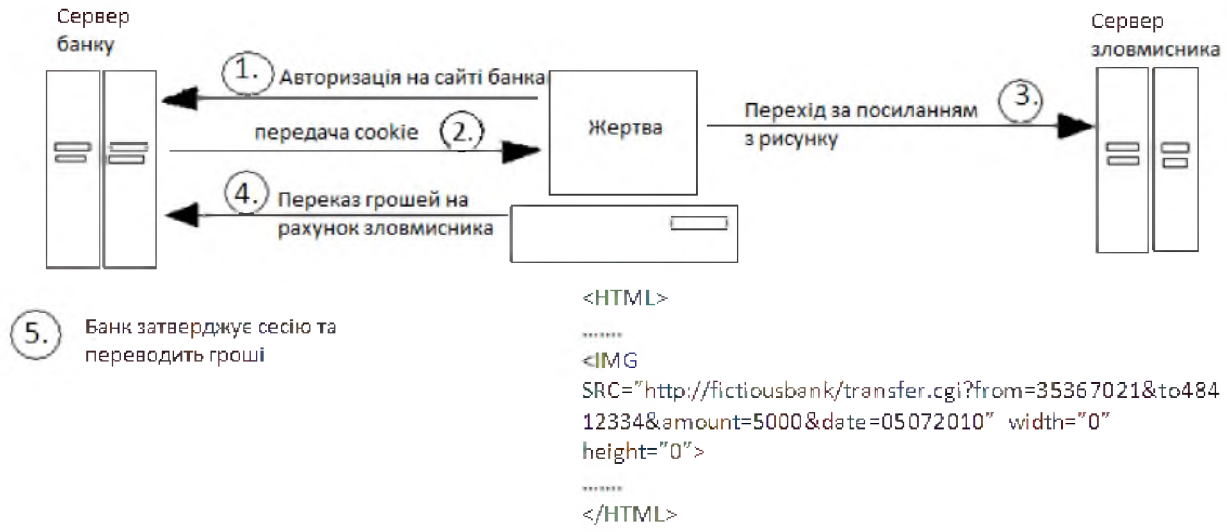

Рисунок 2.3 – схема CSRF атаки

Захист: По-перше, треба переконатися, що ніякі GET-запити не виробляють побічні ефекти.

Щодо метода POST, потрібно включити в кожну форму <form>, що відправляється цим методом, приховане секретне поле, значення якого генерується в кожному сеансі заново. При обробці форми на сервері слід перевірити це поле і вивести виняток, якщо перевірка не пройшла. Так працює система захисту від CSRF-атак в фреймворках мови програмування Python.

2.4.3 SQL-ін'єкція

SQL-ін'єкція – це тип атаки, коли недобросовісний користувач має можливість виконати в базі даних певний SQL запит. Результатом виконання такого запиту може бути видалення або навіть витік даних.

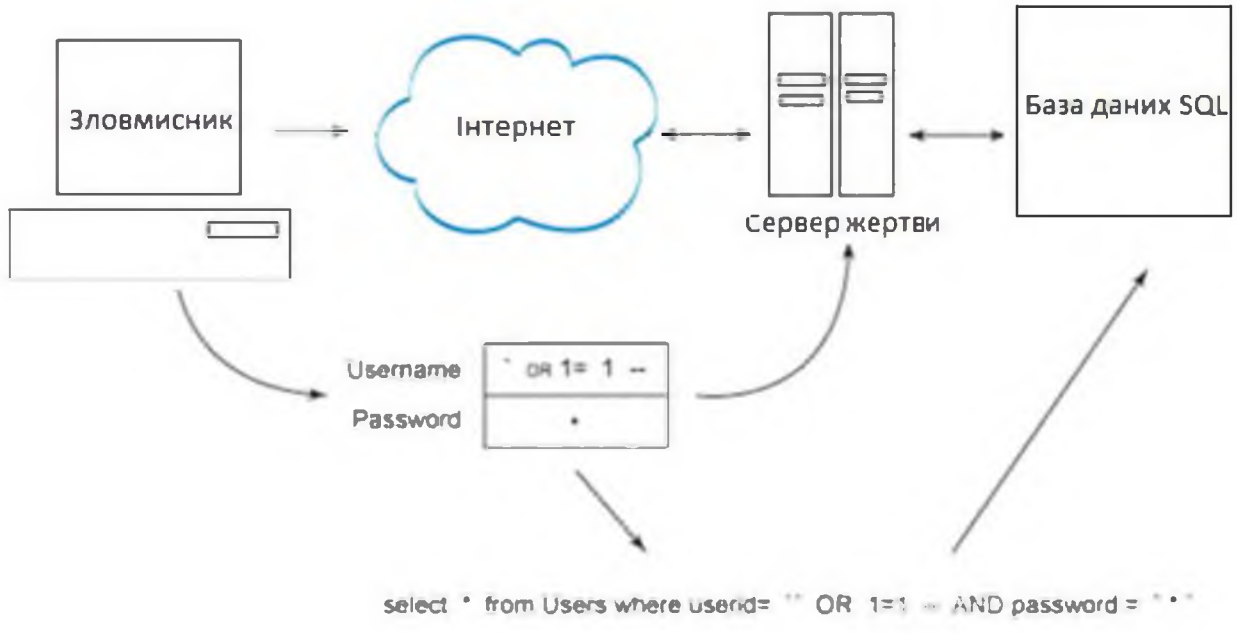

Рисунок 2.4 – SQL-ін'єкція

Захист: Потрібно екранувати все, що формує SQL запит. Використання фреймворків мови програмування Python забезпечить правильне екранування сформованого SQL запиту із застосуванням відповідного драйвера бази даних.

2.4.4 Впровадження серверних розширень (SSI-ін'єкція)

SSI-ін'єкція — техніка атаки на web-додаток, при якій зловмисник посилає конструкції коду, які в наслідку будуть виконані на сервері.

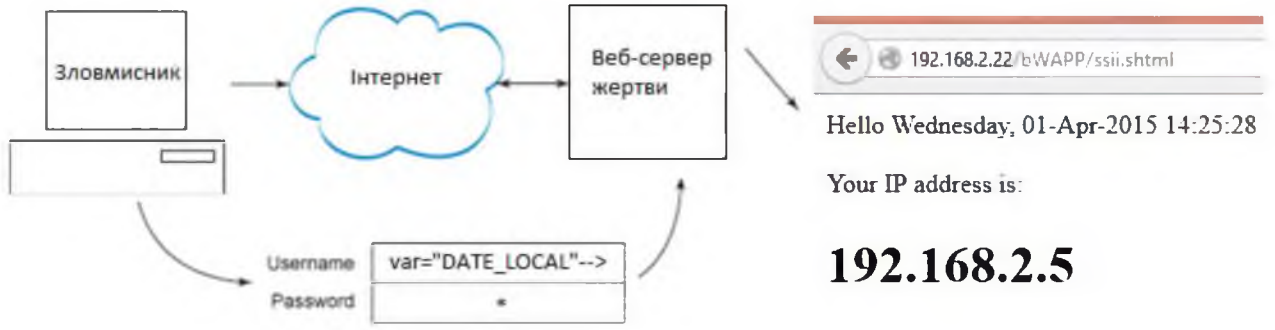

Рисунок 2.5 – SSI-ін'єкція

Вразливості, що призводять до можливості здійснення даних атак, зазвичай полягають у відсутності перевірки даних, наданих користувачем, перед збереженням їх у інтерпретованому сервером файлі.

Перед генерацією HTML сторінки сервер може виконувати сценарії, наприклад Server-site Includes (SSI). У деяких ситуаціях вихідний код сторінок генерується на основі даних, наданих користувачем.

Якщо зловмисник передає серверу оператори SSI, він може отримати можливість виконання команд операційної системи або включити в неї заборонений вміст при наступному відображенні.

Захист: протидія така ж як і при SQL ін'єкції – необхідно фільтрувати всі дані, що надходять зі сторони користувача.

2.5 Інші проблеми захищеності та вразливості медіа-сайтів

2.5.1 Вразливості псевдовипадкових чисел

В Python існує дві основні функції для генерації псевдовипадкових чисел: random() і SystemRandom().

RANDOM() використовує алгоритм Mersenne Twister (MT) (модуль _random), однак перш за все намагається ініціювати його із застосуванням SEED, взятого з urandom, що перетворює ГПВЧ в ГВЧ (генератор випадкових чисел). Якщо викликати urandom не вдається (наприклад, відсутня/dev/urandom або не вдається викликати потрібну функцію з бібліотеки advapi32.dll), то в якості SEED використовується int (time.time() * 256) (що забезпечує недостатню ентропію).

SYSTEMRANDOM() викликає модуль urandom, який використовує зовнішні джерела для генерації випадкових даних.

Зміни в реалізації алгоритму MT полягає в тому, що замість одного числа, заснованого на одному з 624 чисел з поточного стану ГПВЧ (state), використовуються два числа.

Так само, на відміну від PHP, форматувати генератор можна не тільки із застосуванням long-змінної, але і із застосуванням будь-якій послідовності байт (відбувається виклик init_by_array()), що і відбувається при імпорті модуля random із застосуванням зовнішнього джерела ентропії (береться 32 байта з urandom), а в разі, коли це не вдається, використовується time().

Захист:

– використовувати або модуль urandom, або функцію random.SystemRandom();

– проводити ініціалізацію із застосуванням random.seed () перед кожним викликом random.random () с достатньою ентропією SEED (якщо модуль urandom недоступний, то використовуйте, наприклад, значення функції md5 (time.time () * (int) salt1 + str (salt2)), де salt1 і salt2 инициализируются в процесі установки web-додатки);

– обмежити висновок випадкових чисел в web-додатку (досить використовувати хеш-функції типу md5), [46].

2.5.2 Відсутність безпечного стирання пам'яті

Стирання пам'яті використовується, щоб захистити секретні дані або ключовий матеріал від нападників з доступом до неініціалізованої пам'яті. Це

може використовуватися також якщо у нападника є певний місцевий призначений для користувача доступ або через те, як інше програмне забезпечення використовує неініціалізовану пам'ять.

Python не надає API для розробників, щоб здійснити функціональність надійно, відповідно, майже все програмне забезпечення в Python потенційно вразливе для цього нападу. Але рекомендації The CERT з безпеки для розробників оцінюють цю проблему як "Серйозність: середня, вірогідність: мала, вартість виправлення: дорога, щоб відновити". Отже, дана вразливість не є високим ризиком для більшості користувачів, але все ж вона є.

2.5.3 CGI-вразливості

Назвемо основні типи CGI-вразливостей:

– Система авторизації на сайті може бути влаштована таким чином, що після перевірки введених логіна та пароля CGI-програма встановлює cookie з ідентифікаційним номером користувача (id) і в подальшому виробляє автентифікацію тільки на підставі цієї інформації. Таким чином, для доступу під чужим логіном зловмисникові потрібно лише змінити ім'я користувача в cookie.

– Якщо форма авторизації на сайті використовує метод GET, це означає, що логін і пароль передаються як частина URL і їх видно в рядку браузера. URL швидше за все записується в журнал відвідувань браузера і надалі може бути переглянутий іншим користувачем. Або цей URL може навіть бути запропонований системою автозаповнення URL-адреси — зловмиснику залишається лише натиснути Enter! Зберігання відкритого пароля в cookie або hidden-полях форми теж далеко не безпечно, але потребує від зловмисника трохи більших навичок.

– У ситуації, коли CGI-програма Інтернет-магазину використовує для остаточного складання і відправки замовлення ціну товару або загальну суму замовлення, яку одержують із hidden-поля форми підтвердження замовлення, зловмисникові потрібно лише сфабрикувати запит з вигідною для нього ціною.

– До деякої інформації на сайті доступ, як правило, здійснюється за унікальним номером (id) матеріалу в даному розділі. Якщо з якоїсь причини той чи інший матеріал не повинен відображатися на сайті (або саме даному користувачеві), то він позначається в базі даних як прихований і не показується в загальному списку матеріалів даного розділу. Але якщо при запиті такого матеріалу не проводиться перевірка на «прихованість», то зловмисник може отримати доступ до недозволеному матеріалу змінивши унікальний номер запитуваної матеріалу.

– Відсутність перевірки даних, що посилаються скрипту, на метасимволи (такі, наприклад, як &; ` '\" | *? ~ <> ^ () [] {} $), що призводить до виводу скриптом вмісту файлу або самого файлу, який адміністратор сервера повинен приховувати, [47].

2.6 Рекомендації із захисту сайтів

2.6.1 Безпека програмної частини

Програмна частина – це система управління сайтом або скрипти, на яких працює сайт. Надійність програмної частини має на увазі відсутність вразливостей, що дозволяють зловмисникові отримати доступ до бази даних, файлової системи або панелі адміністратора сайту.

Щоб в програмній частині не було вразливостей, розробники повинні розробляти скрипти з оглядкою на безпеку, що виконується не завжди. Правда життя така, що практично в кожній CMS або в скрипті існують вразливості. Частина з них опублікована у відкритому доступі (публічні вразливості), інша не доступна широкій аудиторії і використовується зловмисниками для цільових атак на сайти. Для того щоб програмна частина сайту була надійна і неприступна, потрібно приділяти увагу проблемі безпеки.

Якщо сайту працює на одній з популярних систем управління сайтом, потрібно стежити за оновленнями і латками (patch), і своєчасно оновлювати CMS до самої останньої доступної версії.

Якщо сайт працює на скриптах власної розробки, потрібно виконати сканування сайту доступними засобами пошуку вразливостей (XSpider'ом, Acunetix Web Vulnerability Scanner'ом, утилітами для пошуку SQL ін'єкцій, XSS, RFI та іншими), перевірити вихідний код сайту засобами статичного аналізу вихідного коду (RIPS) і, якщо будуть виявлені вразливості, виправити їх. Крім регулярних оновлень скриптів і CMS є ще один важливий момент, який посилює безпеку і надійність скриптів - це правильна конфігурація сайту.

Для найкращого захисту пропонуються наступні засоби:

– грамотне розмежування прав доступу до файлів і тек;

– закриття доступу до технологічної інформації сайту (каталогам з резервними копіями, конфігураційним файлів тощо), встановлення додаткового захисту на вхід в панель адміністратора;

– код сайту не повинен знаходитися на корені web-сервера. В іншому випадку не виключена ймовірність зловмисного або випадкового виконання коду або відображення у текстовому вигляді;

– дуже важливо тримати як сам сайт, так і його окремі модулі (плагіни, теми оформлення) в актуальному стані - регулярно оновлювати, відмовлятися від використання застарілих версій скриптів які на постійній основі не підтримуються своїми розробниками. Python стає все більш популярною мовою програмування. Оновлення виходять достатньо часто, що дає можливість тримати сайт та його модулі актуальними. Виправлення помилок в ПЗ web-серверів, оновлення CMS-плагінів і доповнень дозволяє підвищити рівень безпеки сайту;

– необхідно перевіряти усі файли, що завантажуються на сайт.

Дані заходи дозволяють значно знизити ймовірність злому сайт, навіть при наявності вразливостей в програмній частині.

2.6.2 Безпека сервера (хостингу)

Другим важливим моментом, що впливає на безпеку сайту в цілому, є хостинг, на якому розміщується сайт. Рівень захищеності сайту багато в чому залежить від рівня рівня безпеки сервера, на якому він розміщується.

Хостинг може бути загальний або спеціалізований («виділений»).

При спільному використанні хостингів відповідальність за безпечну налаштування сервера лежить на адміністратора хостинг-компанії. Для виділеного сервера ця відповідальність лежить на власнику сервера.

Як у випадку загального хостингу, так і в разі, виділеного сервера конфігурація повинна забезпечувати мінімальну свободу дій, що не порушують працездатність сайту. Тобто на сервері повинні бути дозволені тільки найнеобхідніші функції, а все інше - заборонено. Наприклад, якщо сайт не виконує зовнішніх підключень до інших серверів, повинні бути відключені опції зовнішніх з'єднань. Якщо сайт не використовує системні виклики, ці функції необхідно відключити. Крім того, повинна бути обмежена область видимості файлової системи з скриптів. Про все це повинен подбати системний адміністратор сервера.

Як відомо, на одному сервері загального хостингу розміщуються сотні сайтів, і кожному сайту потрібні свої функції. Тому хостинг-компанії максимально лояльно підходять до питань налаштувань сервера, дозволяючи практично всі. Природно, це позначається на загальному рівні безпеки всіх сайтів, розміщених на їх серверах. Тому власнику сайтів потрібно ретельно підходити до питання вибору хостингу: вибирати потрібно той, який дозволяє робити налаштування web-сервера, а не використовувати установки за замовчуванням.

Налаштування виділеного -сервера повинен проводити досвідчений системний адміністратор, який ізолює сайт від іншої частини системи, максимально обмежить свободу скриптів і область їх видимості, а також

організовує механізми контролю цілісності файлової системи, систему резервного копіювання і збирання інформації.

У питаннях захисту оптимальним є багаторівневий підхід. Перший рівень – міжмережевий екран і операційна система, далі – антивірус, який може заповнити будь-які проломи, що виникають.

Необхідності для забезпечення захисту web-сервера:

– не встановлювати непотрібні компоненти. Будь-який компонент несе з собою окрему загрозу. Чим їх більше, тим вище сумарний ризик;

– обмежити кількість спроб автентифікації користувачів із застосуванням та налаштувань web-сервера;

– комп'ютер, з якого виконується робота з сайтом, повинен бути захищений комерційним антивірусним програмним забезпеченням. Для реалізації послуг КА-2 та ЦА-1 рекомендується використовувати програмне забезпечення антивірусного захисту ESET File Security для Microsoft Windows Server версії 4.3.X або інше сертифіковане програмне забезпечення антивірусного захисту. Необхідно регулярно оновлювати антивірусну базу та виконувати перевірку на наявність вірусів . Якщо з сайтом працює кілька людей, дана вимога застосовується до кожного;

– у тому випадку, якщо передбачено завантаження файлів на сайт, бажано обмежити розмір файлів, що завантажуються на сайт, щоб уникнути DOS атаки. Це робиться в налаштуваннях web сервера.

– виконувати резервне копіювання файлів і баз даних. Резервні копії варто робити регулярно і зберігати мінімум 3 останні копії в різних місцях і на різних носіях. Заздалегідь дізнатися, як відновлювати дані із застосуванням резервних копій.

– додати ресурс в сервіси і інструменти для web-майстрів Google. Це дає можливість одразу дізнатись про появу на сайті шкідливого коду. Регулярно перевіряти наявність повідомлень про проблеми і віруси в інструментах web-майстрів.

2.6.3 Обізнаності та акуратності адміністратора сайту

Власники сайту, зазвичай, приділяють мало уваги питанням безпеки, припускаючи, що програмна частина бездоганна, налаштування сервера надійні і безпечні. Хоча така необережність найбільш часто є причиною злому сайтів і зараження вірусами.

Основи, яких повинен постійно дотримуватись адміністратор (власник) сайту:

– паролі від FTP/SSH/панелі адміністратора потрібно міняти регулярно, хоча б раз на місяць;

– не зберігати паролі в програмах (FTP-клієнтах, браузері, електронній пошті);

– не використовувати однакові паролі для різних сервісів;

– пароль повинен бути не коротше 6 символів, оптимально - 10-14. Ідеальний пароль, з точки зору безпеки, повинен бути безглуздим містити цифри, букви, як прописні, так і малі, а якщо є можливість - і спеціальні символи.

– працювати з безпечного протоколу SFTP або SCP.

2.6.4 Автентифікація

Якщо якісь області web-сайту повинні бути доступні тільки деяким клієнтам або зареєстрованим користувачам, для подібного розмежування доступу потрібно метод перевірки автентичності користувачів.

Існує кілька способів автентифікації користувачів: базова автентифікація, дайджест-автентифікація і HTTPS.

– При використанні базової автентифікації ім'я користувача і пароль включаються до складу web-запиту. Навіть якщо контент з обмеженим доступом не дуже важливий, цей метод краще не використовувати, тому що користувач може застосовувати один і той же пароль на декількох web-сайтах. Опитування

Sophos показало, що 41% користувачів застосовують для всієї своєї діяльності в Інтернеті всього один пароль, будь то сайт банку або районний форум. Треба намагатися захищати користувачів від подібних помилок, використовуючи більш безпечні методи автентифікації.

– Дайджест-автентифікація, підтримувана всіма популярними серверами і браузерами, дозволяє надійно шифрувати ім'я користувача і пароль в запиті. Вона допомагає забезпечити безпеку імен і паролів, що виробляє відповідне враження на користувачів і знижує ймовірність успішної атаки на сервер.

– Протокол HTTPS дозволяє шифрувати всі дані, що передаються між браузером і сервером, а не тільки імена користувачів і паролі. Протокол HTTPS (заснований на системі безпеки SSL) слід використовувати в разі, якщо користувачі повинні вводити важливі особисті дані - адреса, номер кредитної картки або банківські відомості.

При виборі системи автентифікації рекомендується використовувати найбезпечніший варіант з наявних. Інші варіанти відлякають клієнтів, що піклуються про захист своїх даних, і можуть привести до виникнення зайвого ризику для користувачів.

2.7 Висновки до другого розділу

У спеціальній частині магістерської роботи було створено модель загроз. У ході аналізу джерел загроз на основі формул було виявлено, що найнебезпечнішим джерелом антропогенних загроз є основний персонал, техногенних загроз - неякісні технічні засоби обробки інформації, стихійних загроз – пожежі та повені.

Було досліджено реалізацію стандартного функціонального профілю захищеності (технологія Т2) для забезпечення захисту інформації від загроз.

Також були проаналізовані проблеми захисту, основні вразливості web-додатків, створених із застосуванням мови програмування Python, та розглянуті методи захисту від атак, націлених на web-сайти.

Основні вразливості мови програмування Python та web-додатків, написаних з їх використанням:

- підключення зовнішніх файлів;
- використання глобальних змінних;
- впровадження команд (SQL-ін'єкція, SSI ін'єкція та інші);
- вразливості генераторів псевдовипадкових чисел (ГПВЧ);
- міжсайтовий скриптінг;
- міжсайтова підробка запиту;
- CGI-вразливості;
- відсутність безпечного стирання пам’яті.

Наприкінці розділу були дані рекомендації із підвищення захищеності медіа-сайтів, створених із застосуванням мови програмування Python, від несанкціонованого доступу.

РОЗДІЛ 3. ЕКОНОМІЧНА ЧАСТИНА

3.1 Вступ

Безпека сайту складається з трьох речей:

– безпеки програмної частини (CMS, скриптів);

– безпеки сервера (хостингу);

– обізнаності та акуратності адміністратора сайту або тих, хто працює з сайтом як адміністратор.

Якщо всі три складових організовані належним чином, то сайт буде неприступним для хакерів і вірусів.

В спеціальній частині даної роботи були дані рекомендації із захисту сайтів, що включають в себе налаштування автоматизованої системи та впровадження засобів захисту.

В економічній частині буде визначено витрати на впровадження засобів захисту та налаштування системи згідно рекомендаціям, що були дані в спеціальній частині.

Вартість впровадження засобів захисту сайтів буде визначено за формулою 3.1.

$$С_{впр} = t_{впр} \cdot (С_{мч} + С_{зп}), \text{грн}, \quad (3.1)$$

де $t_{впр}$ – трудомісткість налаштування системи, годин;

$С_{мч}$ – вартість 1 години машинного часу, грн/год;

$С_{зп}$ – заробітна плата працівників, грн/год.

3.2 Собівартість компонентів системи

У якості сервера було вирішено вибрати Patriot Tower. Він обладнаний 32 гігабайтами оперативної пам’яті, двома жорсткими дисками, кожен з яких має ємкість два терабайти, підтримує технології RAID 0,1,5,10. Також на вибраному сервері вже встановлена операційна система Mircosoft Windows Server 2022. Потужність сервера дорівнює 0,8 кВт.

Серед антивірусів був вибраний ESET File Security for Microsoft Windows Server - рішення для бізнесу, яке покликане забезпечити захист файлового сервера на ОС Microsoft Windows від шкідливого ПЗ.

Щодо архіватора – вибір був зроблений на користь WinRar. Це надійний та якісний архіватор, що дуже давно на ринку та користується загальною повагою та популярністю.

Інше обладнання було вибране серед того, що є доступним на ринку.

Вартість компонентів системи вказана у таблиці 3.1.

Таблиця 3.1 – Собівартість системи

№	Компонент	Ціна, грн.
Обладнання		
1	Сервер Patriot Tower	27000
Ліцензійне ПЗ		
2	Операційна система Mircosoft Windows Server 2022	34111
3	Антивірус ESET File Security for Microsoft Windows Server	3615
4	Архіватор WinRar	950
	Усього	65676

Вартість сервера становить 27000 грн.

Вартість ліцензійного ПЗ становить 38676 грн.

3.3 Розрахунок трудомісткості налаштування системи

Трудомісткість налаштування системи визначається тривалістю кожної робочої операції.

Таблиця 3.2 – Розрахунок трудомісткості налаштування системи

№	Робоча операція	Час (год/рік)
Безпека програмної частини		
1	Розмежування прав доступу до файлів і тек (20000 користувачів)	1
2	Встановлення додаткового захисту на вхід в панель адміністратора	0.25
3	Оновлення компонентів	1,67
4	Перевірка файлів, що завантажуються на сайт	2
Безпека серверу		
5	Перевірка встановлюємих компонентів	1,67
6	Переналаштування сервера для обмеження кількості спроб автентифікації користувачів	0,17
7	Переналаштування сервера для обмеження розміру файлів, що завантажуються на сайт	0,17
8	Резервне копіювання файлів і баз даних	31
9	Перевірка повідомлень про проблеми і віруси в інструментах web-майстрів.	8
Обізнаність та акуратність адміністратора сайту		
10	Зміна паролів від FTP / SSH / панелі адміністратора	1
Усього		46,93

Отже, трудомісткість налаштування системи дорівнює 46,93 годин.

3.4 Розрахунок вартості 1 години машинного часу

Вартість 1 години машинного часу розраховується за формулою 3.2.

$$С_{мч} = P \cdot С_е + \frac{Ф_{перв} \cdot Н_а}{F_p} + \frac{К_{лпз} \cdot Н_{апз}}{F_p}, \text{грн/год}, \quad (3.2)$$

де Р – встановлена потужність сервера, кВт;

$С_е$ – тариф на електричну енергію, грн/кВт·година;

$Ф_{перв}$ – первісна вартість серверу на початок року, грн.;

$Н_а$ – річна норма амортизації на сервер, частки одиниці;

$Н_{апз}$ – річна норма амортизації на ліцензійне програмне забезпечення, частки одиниці;

$К_{лпз}$ – вартість ліцензійного програмного забезпечення, грн.;

$F_р$ – річний фонд робочого часу (за 40-годинного робочого тижня Fр = 1920 год).

Тариф на електроенергію для населення, яке розраховується з енергопостачальною організацією за загальним розрахункових засобів обліку та об'єднане шляхом створення юридичної особи, житлово-експлуатаційним організаціям, крім гуртожитків, становить 1,44 грн (за 1 кВт·год, з ПДВ).

Річну норму амортизації буде визначено за формулою 3.3.

$$Н = \frac{1}{T} \cdot 100\%, \%, \quad (3.3)$$

де Т – період використання компонента.

Планується використовувати сервер та обладнання протягом 3 років. А програмне забезпечення - протягом 5 років.

Річна норма амортизації на сервер – 33%.

Річна норма амортизації на ліцензійне ПЗ – 20%.

Отже, можна порахувати вартість 1 години машинного часу за формулою 3.2:

$$С_{мч} = 0{,}8 \cdot 1{,}44 + \frac{27000 \cdot 0{,}33}{1920} + \frac{38676 \cdot 0{,}2}{1920} = 9{,}82 \text{ грн/год}$$

3.5 Заробітна плата працівників

Як правило, на підприємствах, що створюють web-додатки, з АС працюють системний адміністратор та backend розробник.

До обов’язків системного адміністратора входить встановлення операційної системи та необхідного для роботи програмного забезпечення, здійснення конфігурації ПЗ, підтримка сервера і ПЗ в працездатному стані, встановлення прав доступу, забезпечення резервного копіювання та безпеки інформації.

Середня заробітна плата системного адміністратора – 28000 грн (180 грн/год). Заробітна плата тут і далі вказана без вирахування Єдиного соціального внеску (22%).

Back end розробник створює та запускає на web-сервері скрипти, що написані за допомогою мови програмування, підтримує безпеку програмної частини (CMS, скриптів).

Середня заробітна плата старшого backend розробника – 33600 грн (210 грн/год).

Сумарна заробітна плата за годину дорівнює 360 грн.

3.6 Розрахунок вартості впровадження засобів захисту сайтів

Згідно з формулою 3.1, вартість впровадження засобів захисту сайтів буде складати

$$C_{впр} = 46{,}93 \cdot (9{,}82 + 360) = 17355{,}63 \text{ грн.}$$

3.7 Економічна ефективність

Згідно зі статтею 188-39 Кодексу України про адміністративні правопорушення та Закону України «Про внесення змін до деяких законодавчих актів України щодо посилення відповідальності за порушення законодавства про захист персональних даних» позитивний економічний ефект полягає у тому, що якщо недодержуватися встановленого законодавством про захист персональних даних порядку захисту персональних даних у базі ПД, що призвело до незаконного доступу до них – це тягне за собою накладання штрафу від трьохсот до тисячі неоподаткованих мінімумів доходів громадян. Повторне правопорушення протягом року тягне за собою накладання штрафу у звичайному розмірі (якщо ці дії не тягнуть кримінальної відповідальності). Згідно податковому кодексу України, один неоподаткований мінімум доходів громадян складає 17 гривень.

Отже, якщо не впровадити засоби захисту згідно рекомендаціям, с підприємства, власністю якого є web-додаток, можуть стягнуті штраф у розмірі від 1700 гривень до 34000 гривень.

Для розрахунків економічної ефективності було взято максимальний штраф (34000 гривень) для демонстрації імовірної найгіршої ситуації.

Впровадження засобів захисту та налаштування автоматизованої системи, що склало 17355,63 гривень, окупиться менш ніж за 1 рік.

3.8 Висновки до третього розділу

В економічній частині роботи було визначено витрати на впровадження засобів захисту та налаштування автоматизованої системи згідно рекомендаціям із захисту сайтів, що були дані в спеціальній частині.

Враховуючи систему штрафів за порушення законодавства про захист персональних даних, один максимальний штраф становить 34000 гривень. Це означає, що вартість впровадження засобів захисту та налаштування автоматизованої системи згідно рекомендаціям, що були дані у спеціальній частині, склала 17355,53 грн. та окупиться менш ніж за 1 рік.

ВИСНОВКИ

У магістерській роботі розв'язано актуальне наукове завдання щодо ідентифікації вразливостей медіа-сайтів, створених із застосуванням мови програмування Python. В ході розв'язання поставлених задач були отримані наступні наукові та практичні результати:

Досліджено види, особливості та структура сайтів, проаналізовано поняття медіа-сайту, його типовий склад та особливості функціонування. Виділено два типи сайтів передачі медіа-контенту – Інтернет-ЗМІ та соціальні медіа.

Проведено аналіз і дослідження основних типів мов програмування (компільовані та інтерпретовані). Визначено, що для створення web-сайтів на сьогоднішній день широкого використання набувають інтерпретовані (скриптові) мови програмування.

Обґрунтовано вибір мови програмування, що використовується для створення медіа-сайтів, для подальшого аналізу. Python – універсальна мова програмування, із застосуванням якої можна робити будь-які додатки в діапазоні від Інтернет-сайтів та додатків до роботів для персональних комп'ютерів і системних сервісів. Мова дуже швидко набирає популярність та підтримує широкі можливості для створення та підтримки медіа-сайтів завдяки різноманітним фреймворкам та їх системам управління контентом. У якості прикладу медіа-сайту для обстеження було вибрано web-сайт www.mediavillage.com, який створений із застосуванням мови програмування Python та містить різноманітний медіа контент.

Створено модель загроз. У ході аналізу джерел загроз на основі формул виявлено, що найнебезпечнішим джерелом антропогенних загроз є основний персонал, техногенних загроз - неякісні технічні засоби обробки інформації, стихійних загроз – пожежі та повені.

Досліджено відповідність захищеності медіа-сайту стандартному функціональному профілю захищеності (технологія Т2) для забезпечення

захисту інформації від загроз. Виявлено, що необхідний профіль захищеності можна реалізувати із застосуванням сервісів операційної системи для серверу (Microsoft Windows Server 2022) та програмного забезпечення антивірусного захисту (ESET File Security для Microsoft Windows Server).

Проаналізовано проблеми захисту, основні вразливості медіа-сайтів, створених із застосуванням мови програмування Python, та досліджено методи захисту від атак, націлених на такі web-сайти.

Основні вразливості мови програмування Python та web-сайтів, написаних з їх використанням:

– підключення зовнішніх файлів;
– використання глобальних змінних;
– впровадження команд (SQL-ін'єкція, SSI ін'єкція та інші);
– вразливості генераторів псевдовипадкових чисел (ГПВЧ);
– міжсайтовий скриптінг;
– міжсайтова підробка запиту;
– CGI-вразливості;
– відсутність безпечного стирання пам’яті.

Дані рекомендації із підвищення захищеності медіа-сайтів (безпеки програмної частини, безпеки серверу та обізнаності і акуратності адміністратора сайту), створених із застосуванням мови програмування Python, від несанкціонованого доступу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Словник великий. Статичний сайт [Електронний ресурс]. – Режим доступу: http://wikitwiki.in.ua/index.php?newsid=249111.

2. Статичні чи динамічні сайти: що обрати? [Електронний ресурс]. – Режим доступу: http://webstudio2u.net/ua/design-web/391-static-or-dynamic.html.

3. НД ТЗІ 2.5-010-03 «Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу».

4. Как работает динамический веб-сайт [Електронний ресурс]. – Режим доступу: http://promo-creative.com/kak_rabotaet_veb_sajt.html.

5. Modern Language Association (MLA): «media.» / Online Etymology Dictionary, 2008 [Електронний ресурс]. – Режим доступу: http://dictionary.reference.com/browse/media.

6. Практика медиа измерений. Аудит. Отчетность. Оценка эффективности PR / Игорь Райхман — М.: Альпина Паблишер, 2013. — 432 с.

7. Вікіпедія. Соціальні медіа [Електронний ресурс]. Режим доступу: https://uk.wikipedia.org/wiki/Соціальні_медіа.

8. Закон України «Про авторське право і суміжні права».

9. Лекція 4. Кросплатформність. Види і типи сучасних мов програмування [Електронний ресурс]. – Режим доступу: http://lib.mdpu.org.ua/e-book/vstup/L4.htm#L41.

10. Словник великий. Мова програмування [Електронний ресурс]. – Режим доступу: http://wikitwiki.in.ua/index.php?newsid=1278&news_page=3.

11. WebStudip. PHP [Електронний ресурс]. – Режим доступу: http://www.webostudio.com/ua/stats/php.

12. System Development. Python asyncio [Електронний ресурс]. – Режим доступу: http://sysdev.me/python-asyncio/.

13. Використання Java скриптів [Електронний ресурс]. – Режим доступу: http://uk.shram.kiev.ua/site/java_use/.

14. Вікіпедія. Сравнение языков программирования [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Сравнение_языков_программирования.

15. Безпека сайту [Електронний ресурс]. – Режим доступу: http://gks.com.ua/uk/secutiry-site.html.

16. Википедия. Стандарты криптографии [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/стандарты_криптографии.

17. DocForge. Web application framework [Електронний ресурс]. – Режим доступу: http://web.archive.org/web/20150723163302/http://docforge.com/wiki/Web_application_framework.

18. НД ТЗІ 1.1-003-99 “Термінологія в галузі захисту інформації в комп’ютерних системах від несанкціонованого доступу”.

19. Студопедія. Основні проблеми безпеки інформаційних технологій [Електронний ресурс]. – Режим доступу: http://studopedia.org/12-73883.html.

20. Теорія інформаційну безпеку і методологія захисту / Д.Д. Кирьянов [Електронний ресурс]. – Режим доступу: http://reflist.su/besplatno/referat_zvwfgv/.

21. Метод ранжування загроз [Електронний ресурс]. – Режим доступу: http://wiki.tneu.edu.ua/index.php?title=Метод_ранжування_загроз.

22. НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу».

23. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.

24. НД ТЗІ 2.5-004-99 Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу.

25. Основні аспекти інформаційної безпеки [Електронний ресурс]. – Режим доступу: http://ukrbukva.net/96040-Osnovnye-aspekty-informacionnoiy-bezopasnosti.html.

26. НД ТЗІ 1.1-002-99 Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу.

27. Закон України Про інформацію.

28. НД ТЗІ 2.5-005-99 «Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу».

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітка
1	А4	Реферат	2	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	3	
4	А4	Вступ	2	
5	А4	1 Розділ	36	
6	А4	2 Розділ	51	
7	А4	3 Розділ	6	
8	А4	Висновки	1	
9	А4	Перелік посилань	3	
10	А4	Додаток А	1	
11	А4	Додаток Б	1	
12	А4	Додаток В	1	
13	А4	Додаток Г	1	

ДОДАТОК Б. Перелік документів на оптичному носії

1 Титульна сторінка.doc

2 Завдання.doc

3 Реферат.doc

4 Список умовних скорочень.doc

5 Зміст.doc

6 Вступ.doc

7 Розділ 1.doc

8 Розділ 2.doc

9 Розділ 3.doc

10 Висновки.doc

11 Перелік посилань.doc

12 Додаток А.doc

13 Додаток Б.doc

14 Додаток В.doc

15 Додаток Г.doc

Презентація.pptx

ДОДАТОК В. Відгуки керівників розділів

Відгук керівника економічного розділу:

Керівник розділу _____ _____

(підпис) (ініціали, прізвище)

ДОДАТОК Г. ВІДГУК
на кваліфікаційну роботу магістра на тему:
Підвищення захищеності медіа-сайтів, створених із застосуванням мови Python, від несанкціонованого доступу
Дзюбелюка Владислава Олександровича

Пояснювальна записка складається з титульного аркуша, завдання, реферату, списку умовних скорочень, змісту, вступу, трьох розділів, висновків, переліку посилань та додатків, розташованих на 112 сторінках та містить 12 рисунків, 13 таблиць, 28 джерел та 4 додатка.

Об'єкт дослідження: процес проектування та вдосконалення медіа-сайтів, створених із застосуванням мови програмування Python.

Мета роботи: підвищення захищеності медіа-сайтів, створених із застосуванням мови програмування Python, на основі застосування запропонованих рекомендацій.

Методи дослідження: системний підхід, методи порівняння, аналізу, індукції, дедукції, аналогії.

У роботі проаналізовано проблеми захисту, основні вразливості медіа-сайтів, створених із застосуванням мови програмування Python, досліджено методи захисту від атак та розроблено рекомендації із захисту таких сайтів від несанкціонованого доступу.

Практичне значення роботи полягає у тому, що при використанні розробниками рекомендацій із захисту, даних в роботі, збільшиться захищеність медіа-сайтів, створених із застосуванням мови програмування Python, від несанкціонованого доступу, що зменшить матеріальні збитки власників сайтів.

Студент показав достатній рівень володіння теоретичними положеннями з обраної теми, показав здатність формувати власну точку зору (теоретичну позицію).

Робота оформлена та написана грамотною мовою. Містить необхідний ілюстрований матеріал. Автор добре знає проблему, уміє формулювати наукові та практичні завдання і знаходить адекватні засоби для їх вирішення.

В цілому робота задовольняє усім вимогам і може бути допущена до захисту, а його автор заслуговує на оцінку «_____».

Керівник