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 бакалавра

студента *Бочіна Ігоря Ігоровича*

академічної групи *125-18-3*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *125 Кібербезпека*

на тему *Метод протидії атаці типу "розбиття відповіді HTTP" на WEB сервер*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	к.т.н., доц. Герасіна О.В.			
розділів:				
спеціальний	ас. Мілінчук Ю.А.			
економічний	к.е.н, доц. Романюк Н.М.			
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 бакалавра

студенту *Бочіну Ігорю Ігоровичу* **академічної групи** *125-18-3*
(прізвище ім'я по-батькові) (шифр)

напряму підготовки *125 Кібербезпека*
(код і назва спеціальності)

на тему *Метод протидії атаці типу "розбиття відповіді HTTP" на WEB сервер*

затверджену наказом ректора НТУ «Дніпровська політехніка» від 18.05.2022 № 268-с.

Розділ	Зміст	Термін виконання
Розділ 1	*Дослідження видів атак на веб-сервери*	20.03.2022
Розділ 2	*Тестування захищеності web – серверів від мережевих атак*	30.05.2022
Розділ 3	*Техніко-економічне обґрунтування доцільності запровадження запропонованих в роботі рішень.*	10.06.2022

Завдання видано _____ Мілінчук Ю.А
(підпис керівника) (прізвище, ініціали)

Дата видачі: 14.01.2022р.

Дата подання до екзаменаційної комісії: 10.06.2022р.

Прийнято до виконання _____ Бочін І.І.
(підпис студента) (прізвище, ініціал

РЕФЕРАТ

Пояснювальна записка: 114 с., 10 рис., 7 табл., 4 додатка, 26 джерел.

Об'єкт розробки: вразливості WEB серверів при мережевих атаках.

Предмет розробки: методи протидії мережевим атакам на WEB сервери.

Мета кваліфікаційної роботи: розробити метод протидії атаці «розбиття відповіді HTTP» на WEB сервер та практично довести ефективність його використання.

У першому розділі були визначені джерела вразливості веб-серверів, досліджено види атак на веб-сервери, досліджено методологію атак на веб-сервери.

У другому розділі були визначені інструментарії, що використовуються при атаках на веб сервери, проведено тестування захищеності веб серверу при атаці через протокол HTTP.

В економічному розділі виконаний розрахунок економічної ефективності створення обґрунтованих рекомендацій захисту інформації.

Практичне значення роботи полягає в тому, що запропонована методика може бути використана як набір рекомендацій для виявлення вразливостей типу HTTP Parameter Pollution, а також у подальшому може бути реалізована як додатковий модуль для вже існуючого сканера вразливостей веб-додатків.

ВЕБ СЕРВЕР, ДЖЕРЕЛА ВРАЗЛИВОСТІ, МЕТОДОЛОГІЯ АТАКИ, ТЕСТУВАННЯ ЗАХИЩЕНОСТІ, ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ, ВРАЗЛИВІСТЬ HTTP.

THE ABSTRACT

Explanatory note: 114 p., 10 fig., table 7, 4 applications, 26 of the source.

Object of development: vulnerabilities of WEB servers during network attacks.

Subject of development: methods of counteracting network attacks on WEB servers.

The purpose of the thesis: to develop a method of counteracting the attack "HTTP response" on the WEB server and practically prove the effectiveness of its use.

The first section identified the sources of vulnerabilities in web servers, investigated the types of attacks on web servers, investigated the methodology of attacks on web servers,

The second section identified the tools used in attacks on web servers, tested the security of the web server when attacked via HTTP.

The economic section calculates the economic efficiency of creating sound recommendations for information protection.

The practical value is that the recommended methodology can be used as a set of recommendations for detecting vulnerabilities such as HTTP Parameter Pollution, and can then be implemented as an additional module for an existing web application vulnerability scanner.

WEB SERVER, SOURCES OF VARIABILITY, ATTACHMENT METHODOLOGY, PROTECTION TESTING, VIRTUAL ENVIRONMENT, HTTP IMPROVEMENT.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АС – автоматизована система;

БД – база даних;

КЗЗ – комплекс засобів захисту;

НД – нормативный документ;

НДР – науково-дослідна робота;

НСД – несанкціонований доступ;

ОС – операційна система;

ПЗ – програмне забезпечення;

СЗІ – система захисту інформації;

СКБД – система керування базами даних;

ТЗІ – технічний захист інформації;

DoS – відмова в обслуговуванні (denial of service);

HTTP – гіпертекстовий протокол передачі (hypertext transfer protocol);

IP – інтернет протокол (internet protocol);

ID – ідентифікаційний документ (identity document);

LDAP – полегшений протокол доступ до директорій (lightweight directory access protocol);

SSH – мережевий протокол рівня застосунків, що дозволяє проводити віддалене управління комп'ютером і тунелювання TCP-з'єднань

SQL – мова структурних запитів (structured query language);

SSI – серверні додатки (*server side includes);*

TCP – протокол управління передачею (transmission control protocol);

URL – уніфікована адреса ресурсу (uniform resource locator);

WEB – всесвітня мережа (world wide web);

XSS – міжсайтовий скриптинг (cross site scripting).

ЗМІСТ

ВСТУП

Веб сервери та розташовані на них Web додатки є ключовими елементами поширення інформації в сучасному форматі. Актуальність їх захисту веб-додатків зростає з тенденцією до перенесення стандартних клієнт – серверних додатків у середовище Інтернет, розвиток технології веб 2.0, а також розвиток різних ланок Інтернет бізнесу. Саме вони дали змогу багатьом компаніям не тільки закріпитися на ринках, але й привабити нових клієнтів. Але окрім потенційних користувачів-клієнтів, веб сервери приваблюють різноманітних зловмисників, які мають свою метою отримання конфіденційної інформації.

Світова статистика з інформаційної безпеки свідчить про те, що майже 85% атак на веб сервери компаній припадає на веб-додатки, які вони використовують. Більше половини з цих атак експлуатують вразливості у настройках веб серверів та в недоліках коду в розташовані на них Web додатків.

Поширені вразливості веб серверів були розглянуті на веб Application Security Consortium, де була прийнята загальна класифікація загроз, яка складалася з шести класів атак, в кожному з яких були розглянуті найбільш ймовірні типи:

- клас атак, що використовує механізми автентифікації;
- клас атак, що використовує механізми авторизації;
- клас атак, що використовує механізми атаки на клієнтів;
- клас атак, що використовує механізми виконання коду на сервері;
- клас атак, направлених на розголошення інформації;
- клас атак, що використовує механізми логіки роботи веб-додатка.

Атаки, що базуються на вразливості протоколу HTTP основною перевагою якого є можливість обходу міжмережевих екранів, хоча й тільки недавно були зафіксовані, але постійно вдосконалюються. Незважаючи на,

перший погляд, просту реалізацію, експлуатація вразливості, що базується на такому розповсюдженому протоколу, як HTTP, може бути основою для використання інших видів не менш загрозливих атак: міжсайтового скриптингу та ін..

Останні дослідження у сфері інформаційної безпеки свідчать про те, що близько 50% веб сервери та розташовані на них Web додатки є потенційно вразливими для HTTP атаки типу «Забруднення Параметрів». Більша половина з вразливих додатків відноситься до фінансових та державних установ. Тому дуже важливим є не тільки локалізація даної вразливості для вже існуючих рішень, але тестування, знаходження та попередження цієї вразливості на етапі розробки та введення в експлуатацію.

Об'єктом розробки є вразливості WEB серверів при мережевих атаках.

Предмет розробки: методи протидії мережевим атакам на WEB сервери.

Метою кваліфікаційної роботи є розробка методу протидії атаці "розбиття відповіді HTTP» на WEB сервер та проведення розрахунків , щодо ефективність його використання.

РОЗДІЛ 1. АТАКИ НА ВЕБ СЕРВЕРИ

Веб-сервер - це мережевий додаток, що обслуговує HTTP-запити від клієнтів, зазвичай веб-браузерів. Веб-сервер приймає запити і повертає відповіді, зазвичай разом з HTML-сторінкою, зображенням, файлом, медіа-потоком або іншими даними. веб-сервери - основа Всесвітньої павутини.

Простий обмін між клієнтом та веб – сервером виглядає таким чином [3]:

Браузер користувача аналізує URL – адресу та виділяє окремі частини – шлях, версію протоколу та інше.

Сервер доменних імен (DNS) перетворює доменне ім'я веб-сайту, на який зайшов користувач, на IP – адресу.

Браузер користувача визначає, який протокол потрібно використовувати – HTTP або FTP та якої версії.

Браузер посилає запит до веб-серверу,в якому визначається які дані потрібно передати користувачу.

веб-сервер відповідає на запити браузера. Він підтверджує, що дана адреса існує, знаходить потрібні файли, запускає відповідні сценарії та повертає результати назад до браузеру. У разі, коли файл не може бути знайдений, або команду запит неможливо виконати, веб-сервер відправляє повідомлення про помилку користувачу.

Браузер переводить дані, що були отримані від веб-серверу у HTML та відображає користувачеві.

Створенням програмного забезпечення веб-серверів займаються багато розробників, але найбільшу популярність в WWW отримали такі програмні продукти, як Apache (Apache Software Foundation), IIS (Microsoft), QZHTTP (він же qq.com), Google Веб Server (GWS, Google Inc .) тощо.

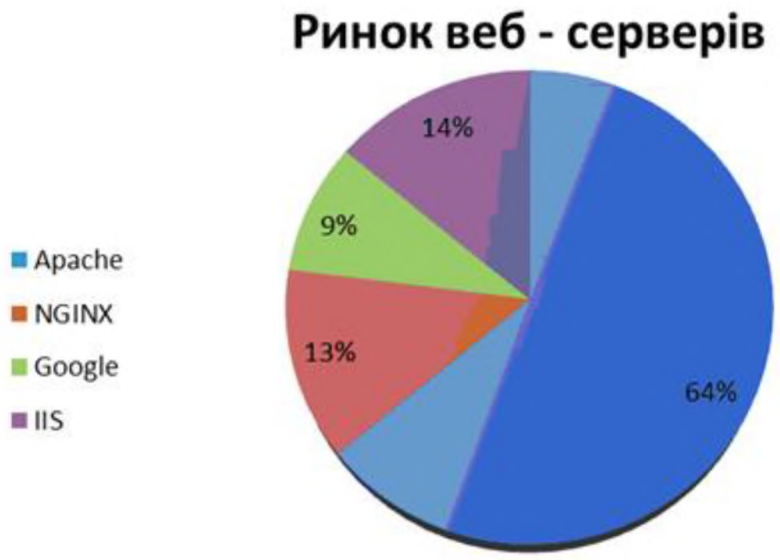

Рисунок 1.1 – Розподіл ринку веб- серверів у 2021 році

Apache - безкоштовний веб - сервер з відкритим вихідним кодом, розповсюджується під сумісною з GPL ліцензією. Apache вже багато років є лідером по поширеності у Всесвітній павутині в силу своєї надійності, гнучкості, масштабованості і безпеки.

IIS (Internet Information Services) - пропріетарний набір серверів для декількох служб Інтернету, розроблений Microsoft і поширюваний з ОС сімейства Windows NT. Основним компонентом IIS є веб-сервер, також підтримуються протоколи FTP, POP3, SMTP, NNTP.

QZHTTP - модифікований Apache, який використовується на китайському порталі qq.com. На ньому розміщені сервіси онлайн-щоденників і миттєвого обміну повідомленнями.

Google Веб Server (GWS) - розробка компанії Google на основі веб-сервера Apache. GWS оптимізований для виконання додатків сервісу Google Applications.

Nginx – це HTTP-сервер, сполучений з поштовим проксі-сервером. Розроблено І.Сисоевим для компанії Рамблер. Восени 2004 року вийшов перший публічно доступний реліз, зараз nginx використовується низкою великих сайтів.

lighttpd - веб-сервер, що розробляється з розрахунком на швидкість і захищеність при використанні на сильно навантажених сайтах, а також відповідність стандартам. lighttpd - безкоштовне програмне забезпечення, яке розповсюджується за ліцензією BSD [4].

1.1 Джерела вразливості веб серверів

Згідно з [1] джерелами уразливості Веб серверів можуть бути:

1. Неправильні права доступу до файлів і каталогів.
2. Встановлення сервера за замовчуванням.
3. Увімкнуто небажані послуги, включаючи управління контентом та віддалене адміністрування.
4. Безпека конфліктує з юзабіліті веб додатків, що виконуються на веб сервері.
5. Відсутність належної політики безпеки, процедур та технічного обслуговування.
6. Неправильна автентифікація з зовнішніми системами.
7. Відсутність резервних копії або зразків файлів.
8. Неправильні конфігурації в веб-сервері, операційних системах та мережах.
9. Помилки серверного програмного забезпечення, веб ОС та веб- додатків.
10. Неправильно налаштовані сертифікати SSL та параметри шифрування.
11. Адміністрування або налагодження, які включені або доступні на веб-серверах.
12. Використання самопідписаних сертифікатів та стандартних сертифікатів.

1.2. Архітектура та логіка виконання на сервері веб-додатків

Веб-додаток - це клієнт – серверний додаток, в якому сервером виступає веб-сервер, а клієнтом виступає веб-браузер, що знаходиться на стороні користувача. Логіка роботи веб-додатку є розподіленою між сервером та клієнтом. Зберігання даних здійснюється на сервері, обмін інформацією відбуваються по мережі. До голових переваг такої архітектури можна віднести міжплатформенність веб-додатків – клієнти не залежать від конкретної операційної системи.

Клієнтська частина веб-додатка реалізує інтерфейс користувача, формує запити до сервера і обробляє відповіді від нього.

Серверна частина отримує запит від клієнта, виконує обчислення, після цього формує веб-сторінку і відправляє її клієнту по мережі з використанням протоколу HTTP.

Архітектура веб-додатка містить:

- клієнтську частину;
- серверну частину;
- канал передачі даних.

Клієнтською частиною є браузер. За допомогою браузеру користувач має можливість звертатися до форм веб-додатків, а також відправляти та приймати дані.

Канал передачі даних представляє транспортний засіб по Internet за допомогою стеку протоколів TCP/IP [1].

Серверна частина веб-додатка може бути представлена:

- веб-серверами;
- серверами додатків;
- серверами баз даних;
- файл-серверами;
- проксі-серверами;
- firewall'ами;
- поштовими серверами.

На даний момент існує та успішно застосовується різні види технологій побудови веб-додатків. Усі такі додатки мають загальну ціль – реалізацію бізнес – логіки на стороні серверу та генерацію коду для клієнту. На рисунку 1.1 зображена схема роботи веб-додатка.

1. Користувач ініціює запит до веб- сервери використовуючи для цього свій веб-браузер.
2. веб-сервер перенаправляє отриманий запит до доступного серверу веб-додатків.
3. Сервер веб-додатку виконує задачу, отриману у запиті.
4. Сервер веб-додтаку отримує доступ до серверу баз даних.
5. Сервер веб- додатку формує відповідь та повертає її до веб-серверу.
6. веб-сервер посилає відповідь користувачеві, яка містить дані про успішність операції та дані, що запитувалися.
7. Інформація з'являється на моніторі користувача.

До основних недоліків подібної схеми роботи можна віднести роботу тільки в режимі запит – відповідь, тобто не має даних про попередні кроки користувача або постійної інформації.

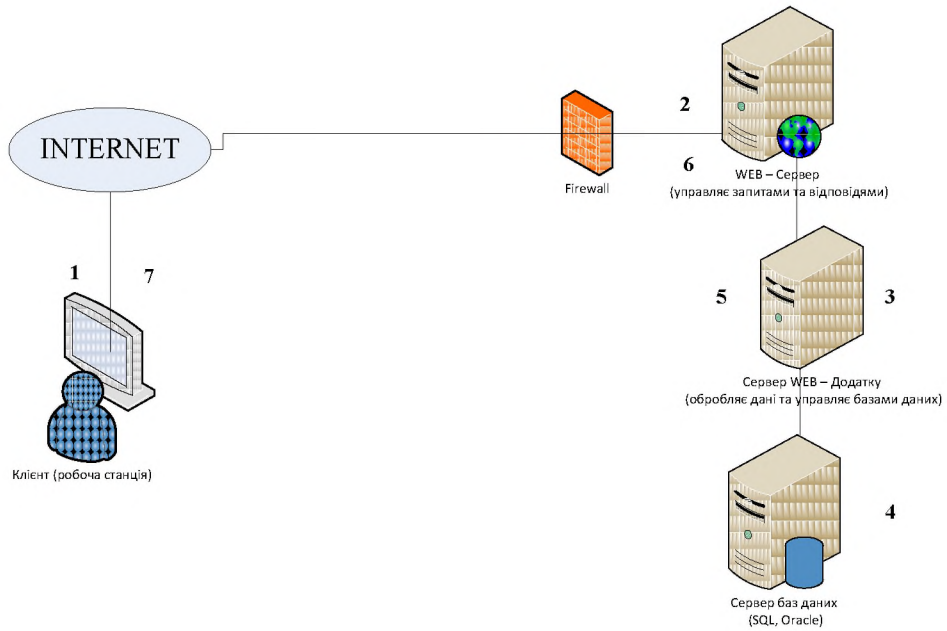

Рисунок 1.2 - Архітектура та логіка роботи веб – додатка на веб сервері

Веб-додаток розробляється на одній із серверних мов програмування. Скомпільовані варіанти розміщуються на окремому сервері додатків, доступ до якого має веб-сервер, який обробляє клієнтські запити. Сервер додатків в свою чергу має доступ до серверів баз даних.

Логіка роботи веб-додатку ґрунтується на формуванні запитів на клієнтській стороні, передачі їх по каналу даних (у формі HTTP-запитів) та подальшій обробці на веб-сервері, сервері додатків, який в свою чергу має можливість взаємодіяти с серверами баз даних для отримання потрібної інформації. .

Багаторівнева система обробки запитів дозволяє працювати одночасно з різними частинами серверної частини і отримувати доступ до окремих сегментів управління веб-сервером, сервером додатків та сервером БД.

Особливістю такої моделі, з точки зору інформаційної безпеки, є те, що некоректний запит з боку клієнта може вплинути на працездатність всієї серверної частини програми або у випадку неправильної авторизації, надати певні права доступу на роботу з серверами БД.

1.3 Механізм атак на веб-додатки які розташовані на веб сервері

Згідно з Center for Internet Security (CIS) простежується стійка тенденція до зростання атак на веб сервері.

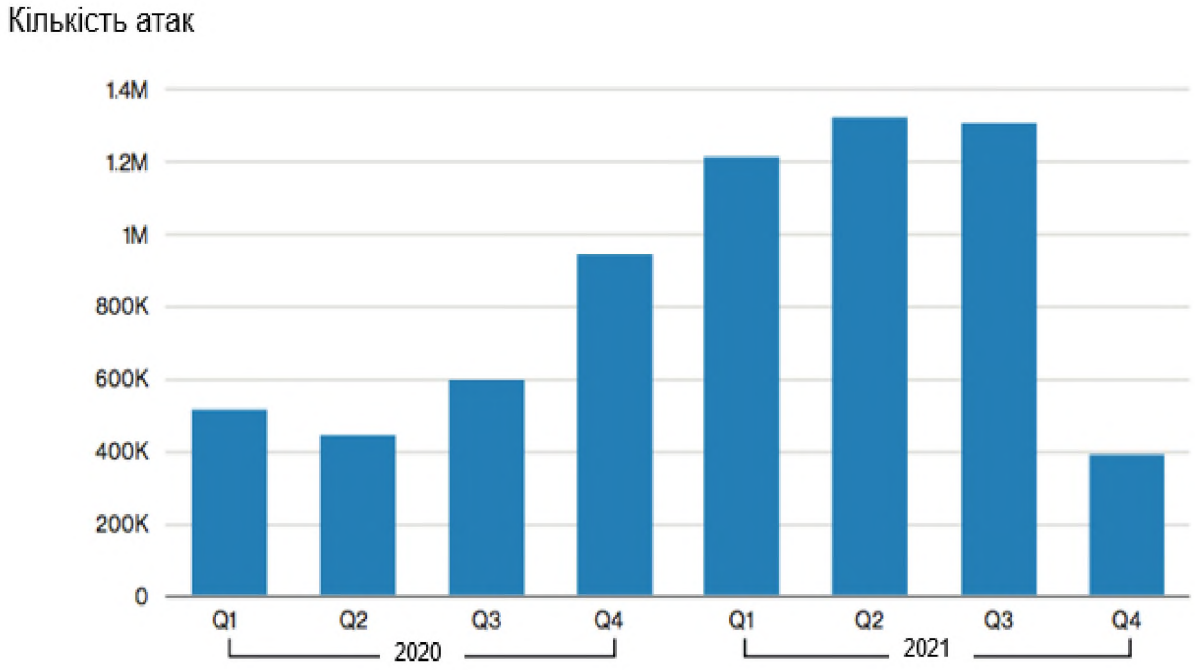

Рисунок 1.2 Статистика атак на веб сервери

Класифікація атак на веб-додатки має ієрархічну структуру та розділяється на шість основних класів. Класифікація атак наведені на рисунку 1.3

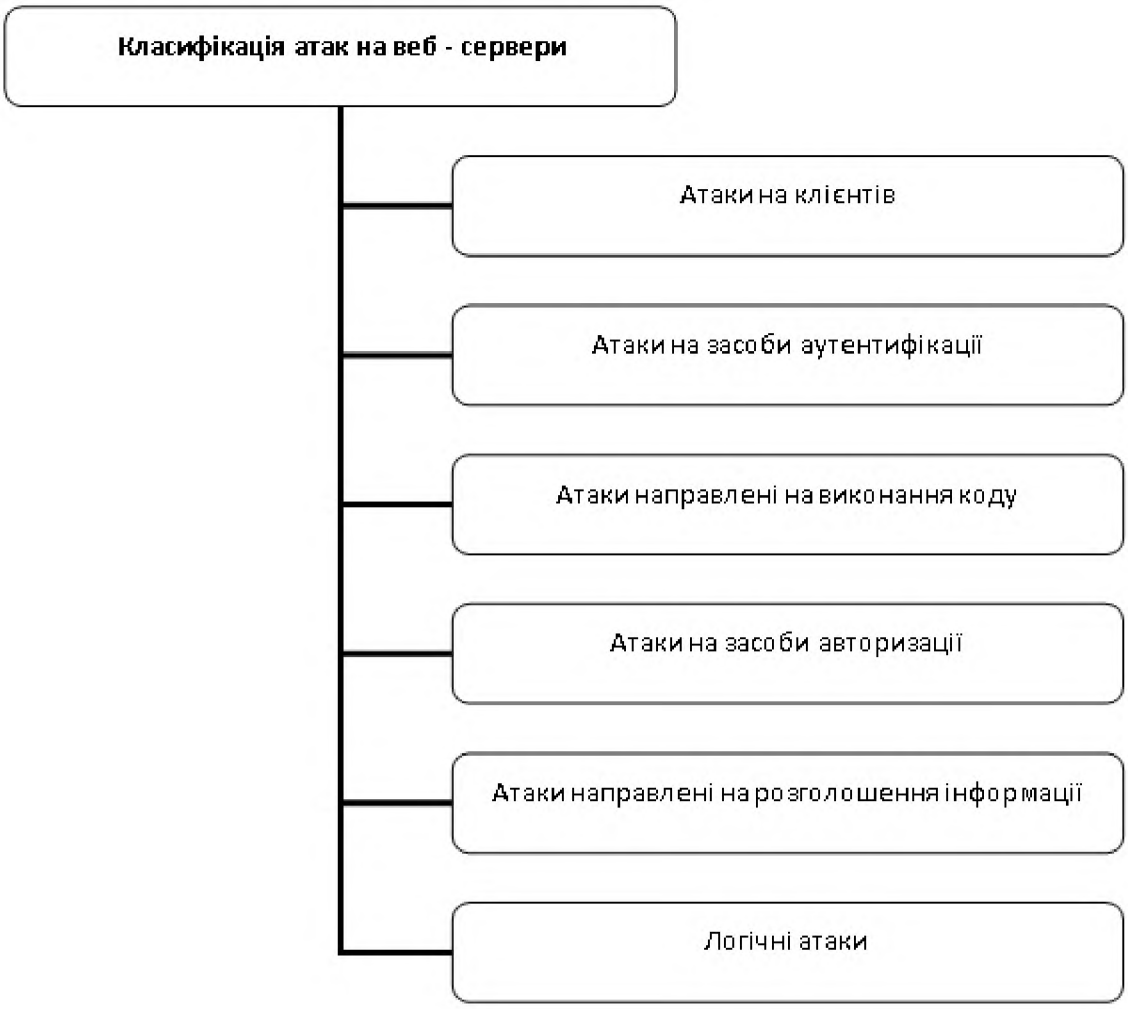

Рисунок 1.4 – Класифікація атак на веб сервери

1.3.1 Атаки на клієнтів

Експлуатуючи довіру, що виникає між користувачем сайту та сервером під час експлуатації додатку, зловмисник має можливість використовувати різні методи для проведення атак на клієнтів сервера. Наявність даної вразливості дозволяє зловмиснику передавати серверу виконуваний код, який у подальшому буде пере направлений до веб-браузеру користувача. Зловмисний код може бути написаний за допомогою HTML/JavaScript, VBScript, ActiveX, Java, Flash та ін. Зловмисні дії коду полягають у

можливості читання, модифікації або передачі даних, доступних для веб-браузеру.

Використовуючи вразливість цього типу зловмисник може скомпрометувати аккаунт користувача, пере направити веб-браузер на інший сервер або виконати підміну вмісту веб-сайту. При міжсайтовому виконанні сценаріїв передача коду здійснюється через URL, у заголовках HTTP-запиту, значеннях полів форм. Розрізняють два типи атак: постійні (бережені) та непостійні (відбиті). Головна відмінність між цими атаками полягає у тому, що у відбитих атаках передача коду серверу здійснюється у рамках одного HTTP-запиту, а в збереженому – в різних.

Збережені вразливості виникають у випадку передачі коду серверу та зберіганні на ньому деякий проміжок часу. Найчастіше дані вразливості використовуються на форумах, чатах та поштах.

Відбиті вразливості виникають у випадку, коли користувач самостійно переходить по посилання, яке було сформовано зловмисником. В процесі завантаження веб-сайту зловмисний код буде передано веб-браузеру користувача і виконано [6].

При використанні уразливості «Підміна HTTP-запиту» зловмисник посилає серверу спеціальним чином сформований запит, відповідь на який інтерпретується метою атаки як два різні відповіді. Друга відповідь повністю контролюється зловмисником, щодає йому можливість підробити відповідь сервера.

В результаті реалізації атаки зловмисник може виконати наступні дії:

- міжсайтове виконання сценаріїв;
- модифікація даних кеша сервера-посередника.

Сервери – посередники, що використовуються для доступу на даний веб-сайт, можуть зберігати підроблену зловмисником відповідь на жорсткому диску. При цьому на всі наступні запити користувачів сервер повертає

модифіковані зловмисником дані – а в результаті – виникає підміна сторінок сервера на стороні клієнта.

Суть міжкористувальницької атаки полягає у тому, що сервер – посередник розділяє одне TCP-з'єднання з сервером між декількома користувачами. В результаті другий користувач у відповідь на запит може отримати сторінку, сформовану зловмисником.

При перехопленні сторінок із даними, що призначені для користувача зловмисник отримує відповідь сервера замість самого користувача. Таким чином, він може дістати доступ до конфіденційної інформації.

При модифікації HTTP-запиту до веб-серверу зловмисник може модифікувати HTTP-запит або сторінку, з якою взаємодіє сам користувач з метою отримання доступу до конфіденційних даних або виконання дій, що становлять загрозу. Прикладом атак такого типу може бути атака HTTP Parameter Pollution, основною перевагою якого є можливість обходу міжмережевих екранів.

1.3.2 Атаки на засоби автентифікації

Атак даного класу направлені на експлуатацію вразливостей в механізмах аутентифікації веб-серверів.

Підбір – автоматизований процес спроб на похибок, основною метою якого є вгадування пароля та імені користувача, номера кредитної картки тощо.

Основною проблемою для забезпечення механізму аутентифікації є той факт, що користувачі у більшості випадків обирають легковгадувані паролі.

Приклади простих паролів:

password1, deer2000, john1234,qwerty, 12345, asdfgh, fred, stopstop, treetree, passpass.

В таких випадках існує потенційна загроза підбору пароля зі сторони зловмисника: використовуючи електронний словник, він може підібрати дані користувача, необхідні для аутентифікації.

Найчастіше підбір використовується для отримання ключів шифрування. У тому випадку, коли сервер використовує ключові комбінації недостатньо великої довжини, зловмиснику стає можливим отримати потрібний ключ, просто перебравши усі можливі комбінації.

Існує два види підбору – прямий та зворотній.

1. Прямий підбір – один варіант імені користувача для різних варіантів паролів.
2. Зворотній підбір – один варіант пароля для різних імен користувачів

До недоліків даних атак можна віднести те, що в залежності від криптостійкості пароля час на його підбір зростає до декількох днів або років. Тому підбір використовується зазвичай у випадку коли блокування у разі неправильного вводу даних на веб-сайті відсутнє [7].

Існують випадки, коли веб-сервер дозволяє зловмиснику мати доступ до важливої інформації без належної автентифікації.

Щоб не використовувати автентифікацію, деякі ресурси по дефолту використовують певну адресу, яка не вказана на основних сторінках сервера або інших загальнодоступних ресурсах. Необхідний URL може бути знайдений шляхом перебору типових файлів і директорій (таких, як /admin/) з використанням повідомлень про помилки журналів перехресних посилань або шляхом простого читання документації. Подібні ресурси мають бути захищені адекватно важливості їх вмісту і функціональних можливостей [7].

Небезпечне відновлення паролів. Реалізація даної вразливості дозволяє зловмиснику несанкціоноване отримувати, відновлювати або модифікувати паролі інших користувачів.

Прикладом реалізації функції відновлення паролю є використання «секретного питання», відповідь на який вказується в процесі реєстрації. Питання або вибирається із списку, або вводиться самим користувачем. Ще один механізм дозволяє користувачеві вказати «підказку», яка допоможе йому згадати пароль. Інші способи вимагають від користувача вказати частину персональних даних - таких, як номер паспорта, домашня адреса, поштовий індекс і так далі, - які потім використовуватимуться для встановлення особи. Після того як користувач доведе свою ідентичність, система відобразить новий пароль або перешле його поштою.

Вразливості, що ґрунтуються на недостатній перевірці при відновленні пароля, виникають у тому випадку, коли зловмисник отримує дані, що використовуються механізмом відновлення.

Це можливо, коли інформація для відновлення пароля є легкої для вгадування або сам процес відновлення має недоліки – і в результаті його можливо обійти.

1.3.3 Атаки направлені на виконання коду

Логіка роботи веб-додатку у більшості випадків ґрунтується на даних, що були передані зі сторони клієнта. Крім даних необхідних для авторизації, дані передані користувачем можуть використовуватися для формування запитів та генерації динамічного вмісту веб-сторінок.

У випадку, коли на етапі розробки веб-додатку вимоги безпеки не враховуються, зловмисник має можливість модифікувати виконувані команди [7].

Вразливість переповнення буфера – найпоширеніша на даний момент в область безпеки ПЗ. Переповнення виникає у випадку, коли об'єм даних перевищує розмір виділеного під них буфера. Коли буфер переповнюється, дані переписують інші області пам'яті, що призводить до виникнення

помилки. Переповнення буфера може викликати відмови в обслуговуванні, призводячи до ушкодження пам'яті і викликаючи помилки в програмах.

Переповнення буфера може викликати відмови в обслуговуванні, змінити шлях виконання програми і виконати в її контексті різні дії, перезаписувати службові області пам'яті, також, при переповненні можуть бути переписані значення змінних у програмі [7].

Атака на функції форматування рядків. При використанні цих атак шлях виконання програми модифікується методом перезапису областей пам'яті за допомогою функцій форматування символьних змінних. Уразливість виникає, коли призначені для користувача дані застосовуються як аргументи функцій форматування рядків - таких, як fprintf, printf, sprintf, setproctitle, syslog і так далі.

Впровадження операторів LDAP. Атаки цього типу спрямовані на веб-сервери, які створюють запити до служби LDAP на основі даних, що вводяться користувачем. Протокол LDAP працює поверх транспортних протоколів Internet (TCP/UDP).

веб-сервер використовує дані, надані користувачем, для генерації динамічних сторінок шляхом створення запитів по протоколу LDAP. У тому випадку, коли інформація, отримана від клієнта не перевіряється належним чином, то зловмисник може отримати можливість модифікувати LDAP-запит.

Виконання команд ОС. Атаки цього класу спрямовані на виконання команд операційної системи на веб-сервері шляхом маніпуляції вхідними даними. Якщо інформація, отримана від клієнта, належним чином не верифікуються, атакуючий отримує можливість виконати команди ОС. Вони будуть виконуватися з тим же рівнем привілеїв, з яким працює компонент додатку, що виконує запит (сервер СУБД, веб-сервер і т.д).

Більшість мов сценаріїв дозволяють запускати команди ОС під час виконання, використовуючи варіанти функції exec. Якщо дані, отримані від

користувача передаються цій функції без перевірки, зловмисник може виконати команди ОС на відстані.

Впровадження операторів SQL. Атаки даного типу використовують веб-сервери, що створюють SQL-запити до серверів СКБД, що формуються на основі даних користувача.

Мова запитів SQL є спеціалізованою мовою програмування, що дозволяє створювати запити до серверів СКБД. Більшість серверів підтримують цю мову у варіантах, стандартизованих ISO і ANSI. У більшості сучасних СКБД присутні розширення діалекту SQL специфічні для цієї реалізації (T-SQL в Microsoft SQL Server, -PL SQL в Oracle і т.д. Якщо інформація, отримана від клієнта, належним чином не верифікуються, атакуючий отримує можливість модифікувати запит до SQL-серверу, що відправляється додатком. Запит буде виконуватися з тим же рівнем привілеїв, з яким працює компонент додатку, що виконує запит (сервер СКБД, веб-сервер і т.д). У результаті зловмисник може отримати повний контроль над сервером СУБД і навіть його операційною системою.

Розрізняють два основних методи експлуатації операторів SQL: звичайна атака та атака всліпу.

- Звичайна атака – виконується підбір параметрів запиту на основі інформації про помилки, що були згенеровані веб-додатком.
- Атака всліпу – додавання до запитів виразів, що завжди повертають істинне або помилкове значення.

Атаки даного класу дозволяють зловмиснику передати виконуваний код, який надалі буде виконаний на веб-сервері. Уразливості, що приводять до можливості здійснення даних атак, зазвичай полягають у відсутності перевірки даних, наданих користувачем, перед збереженням їх у скриптової сервером файлі.

Якщо атакуючий передає серверу оператори SSI, він може отримати можливість виконання команд операційної системи або включити до неї заборонене вміст при наступному відображенні.

Впровадження операторів XPath. Ці атаки спрямовані на веб-сервера, які створюють запити на мові XPath на основі даних, що вводяться користувачем. Мова XPath 1.0 розроблена для надання можливості звернення до частин документу на мові XML. Він може бути використаний безпосередньо або як складова частина XSLT-перетворення XML-документів, або як виконання запитів XQuery. Синтаксис XPath близький до мови SQL-запитів.

1.3.4 Атаки на засоби авторизації

Процес авторизації полягає у визначенні чи має користувач дозвіл на здійснення тієї чи іншої дії. Більшість веб-ресурсів мають декілька типів користувачів, кожен із яких має свій спектр дозволених дій. Доступ до дій та даних, що не входять до списку дозволених повинен бути обмежений. Головним завданням зловмисника у цьому випадку є підвищення власних привілеїв для отримання доступу до захищених даних.

Недостатня авторизація

Недостатня авторизація полягає у тому, що веб-сервер дозволяє зловмиснику мати доступ до інформації або функції, які повинні бути обмежені для цього користувача. Процедура авторизації направлена на розмежування доступу до веб-ресурсу. Правила доступу повинні бути чітко вказані – повинна виконуватися політика безпеки. Доступ до важливих даних сайту повинен бути дозволений тільки адміністраторам.

Деякі сервери після аутентифікації зберігають в cookie або прихованих полях ідентифікатор «ролі» користувача ПЗ. Якщо розмежування доступу ґрунтується на перевірці цього параметра без верифікації приналежності до ролі при кожному запиті, зловмисник може підвищити свої привілеї, модифікувавши значення cookie.

Відсутність тайм-ауту сесії. У разі якщо для ідентифікатора сесії або облікових даних не передбачений таймаут або його значення дуже велике, зловмисник може скористатися старими даними для авторизації. Це підвищує уразливість сервера для атак, пов'язаних з крадіжкою ідентифікаційних даних.

Фіксація сесії. Використовуючи цю атаку, зловмисник вказує ідентифікатору сесії користувача задане значення. Залежно від функціональних можливостей сервера, існує декілька способів зафіксувати значення ідентифікатора сесії. Для цього можуть використовуватися атаки типу «міжсайтове виконання сценаріїв», підготовка сайту за допомогою попереднього HTTP-запиту, або інші види атак. Після фіксації значення ідентифікатора сесії, зловмисник чекає моменту, коли користувач увійде до системи [7].

Після входу користувача зловмисник використовує ідентифікатор сесії для отримання доступу до системи від імені користувача.

1.3.5 Атаки направлені на розголошення інформації

Головною ціллю атак даного виду є отримання додаткової інформації про ПЗ веб-додатку. Завдяки цим вразливостям, зловмисник може визначити ПЗ, що використовується, версії клієнта та серверів, шлях розташування тимчасових файлів, резервних копій.

Індексування директорій. Наданням списку файлів в директорії є нормальною поведінкою веб-сервера, якщо сторінка, що відображується за умовчанням (index.html/home.html/default.htm), відсутня. Коли користувач запрошує основну сторінку веб-сайту, він зазвичай вказує доменне ім'я сервера без імені конкретного файлу. Сервер переглядає основну директорію, знаходить в ній файл, використовуваний за умовчанням, і на його основі генерує відповідь. Якщо такий файл відсутній, як відповідь може повернутися список файлів в директорії сервера. Ця ситуація аналогічна виконанню

команди «ls» (Unix) або «dir» (Windows) на сервері і форматуванню результатів у вигляді HTML. В цьому випадку зловмисник може дістати доступ до даних, не призначених для вільного доступу.

Використовуючи індексування директорій, можна дістати доступ до наступних даних:

- резервні копії (.bak, .old, .orig);
- тимчасові файли. Такі файли повинні видалятися сервером автоматично, але іноді залишаються доступними;
- приховані файли, назва яких починається з символу «.»;
- угода про імена. Ця інформація може допомогти передбачити імена файлів або директорій (admin або Admin, back - up або backup);
- перелік користувачів сервера. Дуже часто для кожного з користувачів створюється директорія з ім'ям, заснованим на назві облікового запису;
- імена файлів конфігурації (.conf, .cfg, .config);
- вміст серверних сценаріїв або виконуваних файлів у разі невірно вказаних розширень або дозволів [7].

Існує три основні сценарії отримання списку файлів веб-сервера:

1. Помилки конфігурації. Подібні проблеми виникають, коли адміністратор помилково вказує в конфігурації сервера цю опцію. Подібні ситуації часто виникають при налаштуванні складних конфігурацій, де деякі директорії мають бути доступні для перегляду;

2. Деякі компоненти веб-сервера дозволяють отримувати список файлів, навіть якщо це не дозволено в конфігураційних файлах. Зазвичай це виникає в результаті помилок реалізації, коли сервер генерує список файлів при отриманні певного запиту;

3. Бази цих пошукових машин (Google, Wayback machine) можуть містити кеш старих варіантів сервера, включаючи списки файлів.

Ідентифікація програмного забезпечення. Визначення версій програмного забезпечення використовується зловмисником для отримання інформації про використовуваних сервером і клієнтом операційних системах, веб-серверах та інтернет-браузерах. Також ця атака може бути спрямована на інші компоненти програмного веб-забезпечення, наприклад службу каталогу або сервер баз даних або використовувані технології програмування. Зазвичай подібні атаки здійснюються шляхом аналізу різної інформації, що надається веб-сервером.

Для визначення версій клієнтського програмного забезпечення зазвичай використовується аналіз HTTP-запитів (порядок дотримання заголовків, значення User-agent і так далі). Проте для цих цілей може застосовуватися і інша техніка. Так, наприклад, аналіз заголовків поштових повідомлень, створених за допомогою клієнта Microsoft Outlook, дозволяє визначити версію встановленого на комп'ютері Інтернет-браузеру Internet Explorer.

Наявність детальної і точної інформації про використовувані програмного забезпечення дуже важлива для зловмисника, оскільки реалізація багатьох атак (наприклад переповнювання буфера) специфічно для кожного варіанту операційної системи або програмного забезпечення. Крім того, детальна інформація про інфраструктуру дозволяє понизити кількість помилок.

Просочування інформації. Ці вразливості виникають в ситуаціях, коли сервер публікує важливу інформацію, наприклад, коментарі розробників або повідомлення про помилки, яка може бути використана для компрометації системи. Цінні з точки зору зловмисника дані можуть міститися в коментаріях HTML, повідомленнях про помилки або просто бути присутнім у відкритому вигляді. Існує величезна кількість ситуацій, в яких може статися просочування інформації. Вона не обов'язково призводить до виникнення вразливості, але часто дає зловмиснику інформацію до подальшої побудови атаки. З просочуванням важливої інформації можуть виникати ризики різної міри,

тому необхідно мінімізувати кількість службової інформації, доступної на клієнтській стороні.

Аналіз доступної інформації дозволяє зловмисникові провадити розвідку і отримати уявлення про структуру директорій сервера, використовуваних SQL- запитах, назвах ключових процесів і програм сервера. Часто розробники залишають коментарі в HTML-сторінках і коді сценаріїв для полегшення пошуку помилок і підтримки програмного забезпечення. Ця інформація може варіюватися від простих описів деталей функціонування програми до (у гірших випадках) імен користувачів і паролів, використовуваних при відладці. Просочування інформації може відноситися і до конфіденційних даних, оброблюваним сервером. Це можуть бути ідентифікатори користувача (ІНН, номери водійських посвідчень, паспортів і т.д.), а також поточна інформація (баланс особового рахунку або історія платежів). Багато атак цієї категорії виходять за рамки захисту програмного веб-забезпечення і переходять в область фізичної безпеки. Просочування інформації в цьому випадку часто виникає коли, в Інтернет-браузері відображується інформація, яка не повинна виводитися у відкритому виді навіть користувачеві [7].

Зворотний шлях в директоріях. Ця техніка атак спрямована на отримання доступу до файлів, директорій і команд, що знаходяться поза основною директорією веб-сервера. Зловмисник може маніпулювати параметрами URL з метою отримати доступ до файлів або виконати команди, що розташовуються у файловій системі веб-сервера. Для подібних атак потенційно уразливий будь-який пристрій, що має веб - інтерфейс. Багато веб-серверів обмежують доступ користувача певною частиною файлової системи, зазвичай званої веб document root або CGI root. Ці директорії містять файли, призначені для користувача, і програми, необхідні для отримання доступу до функцій веб-забезпечення. Більшість базових атак, що експлуатують зворотний шлях, засновані на впровадженні в URL символів «./» для того щоб

змінити розташування ресурсу, який оброблятиметься сервером. Оскільки більшість веб-серверів фільтрують цю послідовність, зловмисник може скористатися альтернативними кодуваннями для представлення символів переходу по директоріях. Популярні прийоми включають використання альтернативних кодувань, наприклад Unicode («.%u2216» або «.%c0%af»), використання зворотного слешу («.\») в Windows-серверах, символів URLEncode («%2e% 2e% 2f») або подвійного кодування URLEncode («.%255c»). Навіть якщо веб-сервер обмежує доступ до файлів певним каталогом, ця уразливість може виникати в сценаріях або

CGI-програмах. Можливість використання зворотного шляху в каталогах досить часто виникає в додатках, що використовують механізми шаблонів чи завантажують текст їх сторінок з файлів на сервері. У цьому варіанті атаки зловмисник модифікує ім'я файлу, що передається як параметр CGI - програми або серверного сценарію. В результаті зловмисник може отримати початковий код сценаріїв. Досить часто до імені файлу, що запитується, додаються спеціальні символи - такі, як «%00» - з метою обходу фільтрів.

Передбачуване розташування ресурсів. Передбачуване розташування ресурсів дозволяє зловмисникові дістати доступ до прихованих даних або функціональних можливостей. Шляхом підбору зловмисник може дістати доступ до вмісту, не призначеного для публічного перегляду. Тимчасові файли, файли резервних копій, файли конфігурації або стандартні приклади часто є метою подібних атак. В більшості випадків перебір може бути оптимізований шляхом використання стандартної угоди про імена файлів і директорій сервера. Отримувані зловмисником файли можуть містити інформацію про дизайн додатка, інформацію з баз даних, імена машин або паролі, шляхи до директорій. Приховані файли також можуть містити вразливості, відсутні в основному застосуванні.

1.3.6 Логічні атаки

Атаки цього класу спрямовані на експлуатацію функцій ПЗ або логіки його функціонування. Логіка ПЗ є очікуваним процесом функціонування програми при виконанні певних дій. Як приклади можна привести відновлення паролів, реєстрацію облікових записів, аукціонні торги, транзакції в системах електронної комерції. ПЗ може вимагати від користувача коректного виконання декількох послідовних дій для отримання певного результату. Зловмисник може обійти ці механізми або використовувати їх у своїх цілях.

Зловживання функціональними можливостями. Ця атака спрямована на використання функцій програмного веб-забезпечення з метою обходу механізмів розмежування доступу. Деякі механізми програмного веб-забезпечення включаючи функції забезпечення безпеки можуть бути використані для цих цілей. Наявність вразливості в одному з другорядних компонентів ПЗ може привести до компрометації усього ПЗ. Рівень ризику і потенційні можливості зловмисника у разі проведення атаки дуже сильно залежать від конкретного ПЗ. Зловживання функціональними можливостями дуже часто використовується спільно з іншими атаками - такими, як зворотний шлях в директоріях і так далі. Приклади зловживання функціональними можливостями включають:

- використання функцій пошуку для отримання доступу до файлів за межами кореневої директорії веб-сервера;
- використання функції завантаження файлів на сервер для перезапису файлів конфігурації або впровадження серверних сценаріїв;
- реалізацію відмови в обслуговуванні шляхом використання функції блокування облікового запису при багаторазовому введенні неправильного пароля.

Відмова в обслуговуванні. Цей клас атак спрямований на порушення доступності веб-сервера. Атаки, спрямовані на відмову в обслуговуванні, реалізуються на мережевому рівні, проте вони можуть бути спрямовані і на прикладний рівень. Використовуючи функції програмного веб забезпечення зловмисник може вичерпати критичні ресурси системи або скористатися вразливістю, що призводить до припинення функціонування системи. Зазвичай DoS-атаки спрямовані на вичерпання критичних системних ресурсів - таких, як обчислювальні потужності, оперативна пам'ять, дисковий простір або пропускна спроможність каналів зв'язку. Якщо якийсь з ресурсів досягне максимального завантаження, ПЗ цілком буде недоступне. Атаки можуть бути спрямовані на будь-який з компонентів веб-забезпечення, наприклад, такі як сервер СКБД, сервер аутентифікації і т.д. [7].

Недостатня протидія автоматизації. Недостатня протидія автоматизації виникає, коли сервер дозволяє автоматично виконувати операції, які повинні проводитися вручну. Для деяких функцій програмного забезпечення необхідно реалізовувати захист від автоматичних атак. Автоматизовані програми можуть варіюватися від нешкідливих роботів пошукових систем до систем автоматизованого пошуку вразливостей і реєстрації облікових записів. Подібні роботи генерують тисячі запитів в хвилину, що може привести до падіння продуктивності усього веб-серверу. Протидія автоматизації полягає в обмеженні можливостей подібних програмних засобів.

Недостатня перевірка процесу. Вразливості цього класу виникають, коли сервер недостатньо перевіряє послідовність виконання операцій ПЗ. Якщо стан сесії користувача і ПЗ належним чином не контролюється, ПЗ може бути вразливий для шахрайських дій. В процесі доступу до деяких функцій ПЗ очікується, що користувач виконає ряд дій в певному порядку. Якщо деякі дії виконуються невірно або в неправильному порядку, виникає помилка, що призводить до порушення цілісності. Прикладами подібних функцій виступають переведення, відновлення паролів, підтвердження купівлі,

створення облікового запису і т.д. В більшості випадків ці процеси складаються з ряду послідовних дій, здійснюваних в чіткому порядку. Для забезпечення коректної роботи подібних функцій веб-забезпечення повинно чітко відстежувати стан сесії користувача і її відповідність поточним операціям. В більшості випадків це здійснюється шляхом збереження стану сесії в cookie або прихованому полі форми HTML. Але оскільки ці значення можуть бути модифіковані користувачем, обов'язково повинна проводитися перевірка цих значень на сервері. Якщо цього не відбувається, зловмисник дістає можливість обійти послідовність дій, тобто, логіку ПЗ.

1.4 Вразливості протоколу HTTP

HTTP (*HyperText Transfer Protocol* - протокол передачі гіпертексту) - символьно-орієнтований клієнт-серверний протокол прикладного рівня без збереження стану, який використовується сервісом World Wide Веб. Основним об'єктом маніпуляції в HTTP є ресурс, на який вказує URI *(Uniform Resource Identifier*- унікальний ідентифікатор ресурсу) в запиті клієнта. Основними ресурсами що зберігаються на сервері є файли, але ними можуть бути й інші логічні (напр. каталог на сервері) або абстрактні об'єкти (напр. ISBN). Протокол HTTP дозволяє вказати спосіб подання (кодування) одного і того ж ресурсу за різними параметрами: mime-типу, мови і т. д. Завдяки цій можливості клієнт і веб-сервер можуть обмінюватися двійковими даними, хоча даний протокол є текстовим [11]. Структура протоколу визначає, що кожне HTTP-повідомлення складається з трьох частин (рис. 1.7), які передаються в наступному порядку:

1 Стартовий рядок (англ. Starting line) – визначає тип повідомлення;
2 Заголовки (англ. Header) – характеризують тіло повідомлення, параметри передачі та інші відомості;

3 Тіло повідомлення (англ. Body) – безпосередньо дані повідомлення. Обов’язково повинно відділятися від заголовків порожнім рядком.

1.4.1 Стартовий рядок HTTP

Стартовий рядок є обов'язковим елементом, тому що вказує на тип запиту / відповіді; заголовки і тіло повідомлення можуть бути відсутні.

Стартові рядки розрізняються для запиту і відповіді. Рядок запиту виглядає так:

Метод URL HTTP/ Версія протоколу

Приклад запиту:

GET /веб-programming/index.html HTTP 1.1

Стартовий рядок відповіді сервера має наступний формат:

HTTP /Версія Код Стану (Пояснення)

Наприклад, на попередній наш запит клієнтом даної сторінки сервер відповів рядком:

HTTP/2.1 200 Ok

1.4.2 Методи протоколу

Метод HTTP (англ. HTTP Method) - послідовність з будь-яких символів, крім керівних і роздільників, яка вказує на основну операцію над ресурсом. Зазвичай метод являє собою короткий англійське слово, записане заголовними буквами (Табл. 1.1). Назви методу чутливі до регістру [11].

Таблиця 1.1 Методи протоколу HTTP

Метод	Коротке описання
OPTIONS	Використовується для визначення можливостей веб-сервера або параметрів з'єднання для конкретного ресурсу. Передбачається, що запит клієнта може містити тіло

	повідомлення для вказівки відомостей що його цікавлять. Формат тіла і порядок роботи з ним у даний момент не визначений. Сервер поки повинен його ігнорувати. Аналогічна ситуація і з тілом у відповіді сервера. Для того щоб дізнатися можливості всього сервера, клієнт повинен вказати в URI зірочку - «*». Запити «OPTIONS * HTTP/2.0» можуть також застосовуватися для перевірки працездатності сервера (аналогічно «пінгування») і тестування на предмет підтримки сервером протоколу HTTP версії 2.0.Результат виконання цього методу не кешується.

Продовження таблиці 1.1

Метод	Коротке описання
GET	Використовується для запиту вмісту зазначеного ресурсу. За допомогою методу GET можна також розпочати будь-який процес. В цьому випадку в тіло відповідного повідомлення слід включити інформацію про хід виконання процесу. Клієнт може передавати параметри виконання запиту в URI цільового ресурсу після символу «?»: GET / path / resource? Param1 = value1 & m2 = value2 HTTP/2.0 Відповідно до стандарту HTTP, запити типу GET вважаються ідемпотентними - багаторазове повторення одного і того ж запиту GET повинне приводити до однакових результатів (за умови, що сам ресурс не змінився за час між запитами). Це дозволяє кешувати відповіді на запити GET. Крім звичайного методу GET, розрізняють ще умовний GET і частковий GET. Умовні запити GET містять заголовки If-Modified-Since, If-Match, If-Range і подібні. Часткові GET містять в запиті Range. Порядок виконання подібних запитів визначено стандартами окремо.
HEAD	Аналогічний методу GET, за винятком того, що у відповіді сервера відсутнє тіло. Запит HEAD звичайно застосовується для вилучення метаданих, перевірки наявності ресурсу (валідація URL) і щоб дізнатися, чи не змінився він з моменту останнього звернення. Заголовки відповіді можуть кешуватися. При розбіжності метаданих ресурсу з відповідною інформацією в кеші копія ресурсу позначається як застаріла.

POST	Застосовується для передачі даних користувача заданому ресурсу. Наприклад, в блогах відвідувачі зазвичай можуть вводити свої коментарі до записів в HTML-форму, після чого вони передаються серверу методом POST і він поміщає їх на сторінку. При цьому передані дані (у прикладі з блогами - текст коментаря) включаються в тіло запиту. Аналогічно за допомогою методу POST звичайно завантажуються файли.

Продовження таблиці 1.1

Метод	Коротке описання
POST	На відміну від методу GET, метод POST не вважається ідемпотентним, тобто багаторазове повторення одних і тих же запитів POST може повертати різні результати (наприклад, після кожної відправки коментаря з'являтиметься одна копія цього коментаря). При результатах виконання 200 (Ok) і 204 (No Content) в тіло відповіді слід включити повідомлення про підсумок виконання запиту. Якщо був створений ресурс, то серверу слід повернути відповідь 201 (Created) із зазначенням URI нового ресурсу в заголовку Location. Повідомлення відповіді сервера на виконання методу POST не кешується.
PUT	Застосовується для завантаження вмісту запиту на вказаний у запиті URI. Якщо по заданому URI не існувало ресурсу, то сервер створює його і повертає статус 201 (Created). Якщо ж було змінено ресурс, то сервер повертає 200 (Ok) або 204 (No Content). Сервер не повинен ігнорувати некоректні заголовки Content-* що передаються клієнтом разом з повідомленням. Якщо якийсь з цих заголовків не може бути розпізнаний або не допустимий при поточних умовах, то необхідно повернути код помилки 501 (Not Implemented). Фундаментальна відмінність методів POST і PUT полягає в розумінні призначень URI ресурсів. Метод POST припускає, що за вказаною URI буде проводитися обробка переданого клієнтом вмісту. Використовуючи PUT, клієнт

	припускає, що вміст що завантажується відповідає ресурсу, що знаходиться за даним URI. Повідомлення відповідей сервера на метод PUT не кешуються.
PATCH	Аналогічно PUT, але застосовується тільки до фрагмента ресурсу.
DELETE	Видаляє вказаний ресурс.
TRACE	Повертає отриманий запит так, що клієнт може побачити, що проміжні сервера додають або змінюють в запиті.

Кожен сервер зобов'язаний підтримувати як мінімум методи GET і HEAD. Якщо сервер не розпізнав зазначений клієнтом метод, то він повинен повернути статус 501 (Not Implemented). Якщо серверу метод відомий, але він не застосовний до конкретного ресурсу, то повертається повідомлення з кодом 405 (Method Not Allowed). В обох випадках серверу слід включити в повідомлення відповіді заголовок Allow зі списком підтримуваних методів.

Найбільш затребуваними є методи GET і POST - на людино-орієнтованих ресурсах, POST - роботами пошукових машин і оффлайн-браузерами.

1.4.3 Коди стану

Код стану інформує клієнта про результати виконання запиту і визначає його подальшу поведінку. Набір кодів стану є стандартом, і всі вони описані у відповідних документах RFC.

Кожен код представляється цілим тризначним числом. Перша цифра вказує на клас стану, наступні - порядковий номер стану (рисунок 1.8). За кодом відповіді зазвичай слід короткий опис англійською мовою.

Рисунок 1.5 - Структура коду стану HTTP

Введення нових кодів повинне проводитися тільки після погодження з IETF.

Застосовувані в даний час класи кодів стану та деякі приклади відповідей сервера наведено у табл. 1.2:

Таблиця 1.2 - Коди стану протоколу HTTP

Клас кодів	Коротке описання
1xx Informational(Інформаційний)	У цей клас виділені коди, що інформують про процес передачі. В HTTP/2.0 повідомлення з такими кодами повинні ігноруватися. В HTTP/2.0 клієнт повинен бути готовий прийняти цей клас повідомлень як звичайну відповідь, але нічого відправляти серверу не потрібно. Самі повідомлення від сервера містять тільки стартовий рядок відповіді і, якщо потрібно, декілька специфічних для відповіді полів заголовка. Проксі-сервера подібні повідомлення повинні відправляти далі від сервера до клієнта. Приклади відповідей сервера: 100 Continue (Продовжувати)

	101 Switching Protocols (Перемикання протоколів) 102 Processing (Йде обробка)
2xx Success(Успішно)	Повідомлення даного класу інформують про випадки успішного приймання та обробки запиту клієнта. В залежності від статусу сервер може ще передати заголовки і тіло повідомлення. Приклади відповідей сервера: 200 OK (Успішно) 201Created (Створено) 202 Accepted (Прийнято) 204 No Content (Немає вмісту) 206 Partial Content (Частковий зміст)
3xx Redirection (Перенаправлення)	Коди статусу класу 3xx повідомляють клієнтові, що для успішного виконання операції потрібно провести наступний запит до іншого URL. У більшості випадків нова адреса вказується у полі Location заголовка. Клієнт в цьому випадку повинен, як правило, зробити автоматичний перехід. Звернімо увагу, що при зверненні до наступного ресурсу можна отримати відповідь з цього ж класу кодів. Може вийти навіть довгий ланцюжок з перенаправлень, які, якщо будуть проводитися автоматично, створять надмірне навантаження на устаткування. Тому

	розробники протоколу HTTP рекомендують після другої поспіль подібної відповіді обов'язково запитувати підтвердження на перенаправлення у користувача (раніше рекомендувалося після 5-го). За цим стежити зобов'язаний клієнт, так як поточний сервер може перенаправити клієнта на ресурс іншого сервера. Приклади відповідей сервера: 300 Multiple Choices (Множинний вибір) 301 Moved Permanently (Переміщено назавжди) 304 Not Modified (Не змінювалося)
4xx Client Error(Помилка клієнта)	Коди даного класу призначені для випадків, в яких клієнт робить невірні запити або коли заданий файл не знайдений за заданою адресою. Приклади відповідей сервера: 400 Bad Request (Неправильний запит) 401 Unauthorized (Несанкційонований доступ) 404 Not Found (Не знайдено)
5xx Server Error(Помилка сервера)	Коди 5xx виділені під випадки невдалого виконання операції з вини сервера. Для всіх ситуацій, крім використання методу HEAD, сервер повинен включати в тіло повідомлення пояснення, яке клієнт відобразить користувачеві.

	Приклади відповідей сервера: 500 Internal Server Error (Внутрішня помилка сервера) 502 Bad Gateway (Непрацюючий шлюз) 503 Service Unavailable (Сервіс недоступний) 504 Gateway Timeout Шлюз не відповідає)

1.4.4 Заголовки HTTP

Заголовок HTTP *(HTTP Header)* - це рядок в HTTP-повідомленні, що містить розділену двокрапкою пару виду «параметр-значення». Формат заголовка відповідає загальному формату заголовків текстових мережевих повідомлень ARPA (RFC 822). Як правило, браузер і веб-сервер додають у повідомлення більш ніж по одному заголовку. Заголовки повинні відправлятися раніше тіла повідомлення і відокремлюватися від нього хоча б одним порожнім рядком (CRLF).

Назва параметра має складатися мінімум з одного друкованого символу (ASCII-коди від 33 до 126). Після назви відразу повинен слідувати символ двокрапки. Значення може містити будь-які символи ASCII, крім перекладу рядка (CR, код 10) і повернення каретки (LF, код 13).

Пробільні символи на початку і наприкінці значення обрізаються. Послідовність декількох пробільних символів всередині значення може сприйматися як один пропуск. Регістр символів в назві і значення не має значення (якщо інше не передбачено форматом поля) [11].

Приклад заголовків відповіді сервера:

Server Apache/2.2.3 (CentOS)

Last-Modified: Wed, 09 May 2015 17:13:15 GMT

Content-Type: text/html; charset=UTF-8

Accept-Ranges: bytes

Date: Thu, 23 May 2015 4:04:36 GMT

Content-Length: 2945

Connection: keep-alive

200 OK

Всі HTTP-заголовки поділяються на чотири основні групи:

- General Headers (Основні заголовки) – повинні додаватися до будь-якого повідомлення клієнта та сервера.
- Request Headers (Заголовки запиту) – застосовуються тільки в запитах клієнта;
- Response Headers (Заголовки відповіді) – присутні тільки у відповідях сервера;
- Entity Headers (Заголовки сутності) – супроводжують кожну сутність повідомлення;

Сутності *(entity,* в перекладах також зустрічається назва "об'єкт") - це корисна інформація, передана в запиті або відповіді. Сутність складається з метаінформації (заголовки) і безпосередньо вмісту (тіло повідомлення).

В окремий клас заголовки суті виділені, щоб не плутати їх з заголовками запиту або заголовками відповіді при передачі множинного вмісту (multipart / *). Заголовки запиту і відповіді, як і основні заголовки, описують все повідомлення в цілому і розміщуються тільки в початковому блоці заголовків, у той час як заголовки суті характеризують вміст кожної частини окремо, розташовуючись безпосередньо перед її тілом.

У таблиці 1.3 наведено короткий опис деяких HTTP-заголовків.

Таблиця 1.3 - Заголовки HTTP

Заголовок	Група	Коротке описання
Allow	Entity	Список методів, застосовних до запитуваного ресурсу.
Content-Encoding	Entity	Застосовується при необхідності перекодування вмісту (наприклад, gzip / deflated).
Content-Language	Entity	Локалізація вмісту (мова(и))
Content-Length	Entity	Розмір тіла повідомлення (в октетах)
Content-Range	Entity	Діапазон (використовується для підтримки багато поточного завантаження чи дозавантаження)
Content-Type	Entity	Вказує тип вмісту (mime-type, наприклад text / html). Часто включає вказівку на таблицю символів локалі (charset)
Expires	Entity	Дата / час, після якої ресурс вважається застарілим. Використовується проксі-серверами
Last-Modified	Entity	Дата / час останньої модифікації сутності
Cache-Control	General	Визначає директиви управління механізмами кешування. Для проксі-серверів.
Connection	General	Задає параметри, необхідні для конкретного з'єднання.
Date	General	Дата і час формування повідомлення
Pragma	General	Використовується для спеціальних

		вказівок, які можуть (опціонально) застосовуватися до будь-якого одержувачу по всьому ланцюжку запитів / відповідей (наприклад, pragma: no-cache).
Transfer-Encoding	General	Задає тип перетворення, що застосовується до тіла повідомлення.

Продовження таблиці 1.3

Заголовок	Група	Коротке описання
Via	General	Використовується шлюзами і проксі для відображення проміжних протоколів і вузлів між клієнтом і веб-сервером.
Warning	General	Додаткова інформація про поточний статус, яка не може бути представлена в повідомленні.
Accept	Request	Визначає застосовні типи даних, очікуваних у відповіді.
Accept-Charset	Request	Визначає кодування символів (charset) для даних, очікуваних у відповіді.
Accept-Encoding	Request	Визначає застосовні формати кодування / декодування вмісту (напр, gzip)
Accept-Language	Request	Відповідні мови. Використовується для узгодження передачі.
Authorization	Request	Облікові дані клієнта, що запитує ресурс.
From	Request	Електронна адреса відправника
Host	Request	Ім'я / мережеву адресу [і порт] сервера. Якщо порт не вказаний, використовується 80.
If-Modified-Since	Request	Використовується для виконання умовних методів Якщо запитуваний ресурс змінився, то він передається з сервера, інакше - з кешу.
Max-Forwards	Request	Представляє механізми обмеження

		кількості шлюзів і проксі при використанні методів TRACE і OPTIONS.
Proxy-Authorization	Request	Використовується при запитах, що проходять через проксі, що вимагають авторизації
Referer	Request	Адреса, з якого виконується запит. Цей заголовок відсутній, якщо перехід виконується з адресного рядка або, наприклад, по посиланню з js-скрипта.

Продовження таблиці 1.3

Заголовок	Група	Коротке описання
User-Agent	Request	Інформація про користувача агента (клієнта)
Location	Response	Адреса перенаправлення
Proxy-Authenticate	Response	Повідомлення про статус з кодом 407.
Server	Response	Інформація про програмне забезпечення сервера, що відповідає на запит (це може бути як веб-так і проксі-сервер)

Тіло HTTP повідомлення (*message-body*), якщо воно присутнє, використовується для передачі сутності, зв'язаної із запитом або відповіддю.

Тіло повідомлення (*message-body*) відрізняється від тіла сутності (entity-body) тільки у тому випадку, коли при передачі застосовується кодування, вказане у заголовку Transfer-Encoding. У всіх інших випадках тіло повідомлення ідентичне тілу сутності [11].

Заголовок Transfer-Encoding повинен відправлятися для вказівки будь-якого кодування передачі, застосованого додатком у цілях гарантування безпечної та правильної передачі повідомлення. Transfer-Encoding – це властивість повідомлення, а не сутності, та вона може бути додана або видалена будь-яким додатком у ланцюжку запитів/відповідей [12].

Присутність тіла повідомлення у запиті відмічається додаванням до заголовків запитів поля заголовка Content-Length або Transfer-Encoding. Тіло повідомлення (message-body) може бути додане у запит тільки коли метод запиту допускає тіло сутності (entity-body).

Усі відповіді містять тіло повідомлення, можливо нульової довжини, крім відповідей на запит методом HEAD та відповідей с кодами статусу 1xx (інформаційні), 204 (немає вмісту), та 304 (не модифікований) [11].

1.5 Висновки до першого розділу

Тема захисту веб серверів на сьогоднішній час є актуальною – вона постійно розвивається, з'являються нові вразливості. Тому важливим є виділення тих типів вразливостей, які на сьогоднішній день є найбільш розповсюдженими, найменш досліджуваними та тим не менш становлять потенційну загрозу для інформаційної безпеки додатку.

Вразливість типу HTTP Parameter Pollution знайдена дуже недавно, а також експлуатує недоліки протоколу HTTP та HTTPS, які широко використовуються для взаємодії між клієнтом та веб-додатком. Саме тому задача розробки методики для тестування та виявлення вразливостей цього типу є головним шляхом для вирішення проблеми безпеки веб-додатків.

Поставлена задача потребує:

- Визначити джерела уразливості Web серверів .
- Проаналізувати види атак на Web сервери.
- Проаналізувати методи атак на Web сервери.
- Визначити інструментарій що використовується при атаках на Web сервери.
- Провести тестування захищеності Web серверу при атаці типу «розбиття відповіді HTTP».
- Розробити метод протидії.

РОЗДІЛ 2. ТЕСТУВАННЯ ЗАХИЩЕНОСТІ WEB – СЕРВЕРІВ

- Для досягнення поставленої мети необхідно:
- Визначити джерела уразливості Web серверів.
- Дослідити види атак на Web сервери.
- Дослідити методологію атак на Web сервери.
- Визначити інструментарій що використовується при атаках на Web сервери.
- Провести тестування захищеності Web серверу при атаці однією із запропонованих методик

Результати повинні відповідати вимогам Закону України «Про інформацію», Закону України «Про захист персональних даних», Закону України «Про захист інформації в інформаційно-телекомунікаційних системах», «Положення про технічний захист інформації в Україні», що затверджено указом Президента України від 27 вересня 1999 р. №1229/99, НД ТЗІ 1.1-002-99 «Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу», НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу», НД ТЗІ 1.1-003-99 «Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу», «Про вищу освіту», Закону України «Про освіту», «Положення про організацію навчального процесу у вищих навчальних закладах», що затверджено наказом Міністерства освіти України від 2 червня 1993 р. №161, нормативних документів з технічного захисту інформації, державних стандартів України в галузі інформаційної безпеки та інших законів України, що стосуються забезпечення безпеки інформації.

Результати розробки мають бути подано у вигляді, що дозволяє безпосереднє використання для створення засобів захисту яка розташована на серверах.

Економічний ефект повинен бути позитивним завдяки зменшенню витрат на придбання та використання засобів для тестування захищеності веб-серверів від атак зі сторони клієнта

Соціальний ефект від реалізації результатів роботи очікується позитивним завдяки створенню умов для реалізації можливостей працівникам підприємства підвищити продуктивність праці та її комфортність.

2.1 Аналіз циркулюючої інформації на веб-серверах

Всю інформацію, що циркулює у межах функціонування веб-серверу можна розділити на такі види: клієнтську, серверну та периферійну.

До клієнтської інформації можна віднести вміст форм, що присутні на веб-сторінці, а також специфіку побудови запиту, який формується на стороні клієнту.

До серверної інформації можна віднести інформацію про тип веб-серверу, що використовується у додатку; інформацію, що циркулює у базах даних; інформацію щодо механізмів захисту у додатку; інформацію про допоміжні модулі; інформацію про середовище, яке забезпечує зв'язок між основними ресурсами та зовнішніми системами.

Головною особливістю веб-додатків, що створюються на базі клієнт – серверної архітектури є те, що клієнтські дані обробляються н стороні сервера, що збільшує можливість доступу до даних на веб-сайті.

До периферійної інформації відносяться дані про особливості веб-додатка, які можуть бути отримані зі сторони клієнта. До цього типу можна віднести дані про середовище розробки додатку та особливості його функціонування, мову скриптів, що застосовуються у додатку, ПЗ веб-сервера, ОС та інші елементи.

Ключовою особливістю периферійної інформації є її доступність та відкритість. З позиції зловмисника, дана інформація є цінною для пошуку

вразливостей у веб-додатку, оскільки містить дані про елементи архітектури серверної частини додатку [16].

До такої інформації відносять:

а) HTTP запити та відповіді сервера на них;

б) формат і написи на сторінках, що повідомляють про помилки;

в) вихідні коди доступних сторінок;

Вихідний код сторінок може дати вичерпну інформацію про програму. На основі даної інформації можуть бути побудовані власні запити до серверної частини веб-додатку. Метод тестування захищеності веб-додатків, ґрунтується на базових знаннях про додаток та основні дані, що супроводжують появлення помилок в роботі додаток.

Для запобігання атакам, які засновані на використанні механізмів автентифікації та авторизації, виконанні коду та інших типів атак, потрібно блокувати доступність периферійної інформації, що циркулює у веб-додатку. Саме блокування периферійної інформації має бути ключовим напрямком підвищення захисту веб-додатка.

2.2 Модель загроз

Загрози для веб-додатку можна класифікувати лише за ймовірністю їх втілення та рівнем впливу на ресурси.

Ймовірність здійснення атаки використовуючи загрози та вразливості:

А – низька;

Б – середня;

В – висока.

Ймовірність та оцінка загроз наведені у таблицях 2.2 та 2.3

Таблиця 2.1 – Модель загроз для веб-додатків

Загроза	Ймовірність втілення	Рівень впливу

Загрози для операторів		
Атаки на веб-сервер	В	середній
Навмисне виведення із ладу обладнання	Б	високий
Порушення нормальної роботи: швидкості передачі інформації	Б	високий
Використання службового становища для передачі інформації третім особам	В	високий
Навмисне або не навмисне невірне налагодження обладнання	Б	критичний
Навмисне або не навмисне введення неправдивої інформації	В	високий
Знищення технічних засобів передачі та обробки інформації	А	критичний

Таблиця 2.2 – Модель загроз для користувачів

Загрози для користувачів		
Знищення конфіденційної інформації	Б	середній
Порушення функцій користувача	Б	середній
Блокування профілю користувача	Б	низький
Доступ до інформації користувача	А	високий
Перехоплення даних через додаток	Б	низький
Крадіжка даних, що передаються	В	критичний

Рівні впливу на ресурс:

– критичний – інформація/ресурс/мережа може бути знищена, змінена без можливості відновлення. В даному випадку втрати підприємства є значними;

– високий – інформація/ресурс/мережа може втратити деякі свої властивості, але може бути відновлена. В даному випадку втрати підприємства є меншими, ніж внаслідок повної втрати і неможливості відновити інформацію;

– середній – інформація/ресурс/мережа втрачає деякі властивості, але може бути відновлена в прийнятні терміни і з мінімальними втратами;

– низький – інформація/ресурс/мережа може зазнати невеликих змін, які можливо відновити в найкоротший термін.

2.3 Модель порушника

Модель порушника являє собою формальний опис порушника, його можливих дій та результатів його дій щодо додатка.

По відношенню до АС порушники можуть бути внутрішніми або зовнішніми.

Внутрішніми порушниками можуть бути:

а) адміністратори безпеки веб-додатків;

б) адміністратори веб-серверів;

в) адміністратори серверів баз даних.

Зовнішніми порушниками щодо веб-додатків є усі користувачі, що можуть мати доступ до її клієнтської частини. Зареєстровані користувачі також є зовнішніми порушника, оскільки архітектура будь-якого веб-додатка регламентує доступність клієнтської частини до інформації, що циркулює на сервері.

Внутрішні порушники мають доступ до всієї циркулюючої інформації на серверній частині, тому результатом їх дій може бути порушення конфіденційності, цілісності та доступності інформації веб-додатка при будь-яких видах атак чи порушеннях архітектури серверної частини додатка.

Можливість проведення атак зі сторони клієнта, тобто зі сторони зовнішнього порушника є небезпечними за випадку завчасного аналізу веб-додатка, тобто наявності інформації про архітектуру та опис можливих шляхів доступу до неї.

Зовнішніми порушниками можуть бути:

а) зареєстровані користувачі додатка;

б) зловмисники, що мають мету порушення доступності, цілісності та конфіденційності інформації, що циркулює в веб-додатках;

в) аналітики веб-додатків.

Метою порушника є:

а) отримання інформації у потрібному обсязі та асортименті;

б) наявність можливості вносити зміни в інформаційні потоки у відповідності зі своїми намірами;

в) нанесення збитків шляхом знищення матеріальних та інформаційних цінностей, що є на серверній частині додатків.

Зареєстровані користувачі додатка мають можливість ведення діалогу з АС через клієнтську частину додатка. Мають можливість запуску фіксованого набору завдань за допомогою модулів додатка, що реалізують заздалегідь передбачені функції обробки інформації. Володіють інформацією про основні закономірності формування в ній запитів до сервера даних та вміють користуватися засобами, що їх формують. Використовують при своїх атаках виключно агентурні методи одержання відомостей, тобто роботу с клієнтом.

Зловмисники, що мають мету порушення доступності, цілісності та конфіденційності інформації, що циркулює в додатках можуть створювати і запускати власні програми з новими функціями обробки інформації, що мають на меті атаку на клієнтську частину додатка. Володіють високим рівнем знань та досвідом роботи з технічними засобами системи та їхнього обслуговування та використовують виключно агентурні методи одержання відомостей, тобто ПЗ, що має інструменти щодо досягнення мети порушника.

Аналітики веб-додатків мають можливість аналізу АС, тобто мати вплив на додаток на рівні сервера, що дозволяє їм розглянути програмне забезпечення системи і конфігурацію її устаткування. Володіють високим рівнем знань у галузі обчислювальної техніки та програмування, проектування та експлуатації АС. Використовують способи і засоби активного впливу на АС, що змінюють конфігурацію системи, тобто використовують заздалегідь створені моделі та методики, щоб отримати доступ до додатків.

Адміністратори безпеки мають повний обсяг можливостей щодо додатків на рівні проектування, реалізації, впровадження, супроводження АС, аж до включення до складу АС власних засобів з новими функціями обробки інформації. Володіють інформацією про функції та механізм дії засобів захисту. Використовують способи і засоби активного впливу на АС, що змінюють конфігурацію системи, за допомогою доступу до серверної частини додатка та наявності можливості її конфігурувати [14-15].

2.4 Профіль захищеності

Інформація, що циркулює у веб-додатках повинна мати рівні гарантій, які забезпечують її цілісність, доступність та конфіденційність. Рівні гарантій зіставляються із існуючими профілями захищеності. Рівні гарантій можуть бути розширеними, якщо необхідно виконати необхідні додаткові умови для нормального функціонування додатку згідно запропонованим бізнес – механізмам.

Головними особливостями веб-додатків є той факт, що забезпечення цілісності та доступності інформації, що циркулює у додатку є найбільш пріоритетним завданням. Висока пріоритетність пояснюється тим, що користувач повинен отримати доступ до потрібної йому інформації у будь – який момент за допомогою клієнтського програмного забезпечення.

Забезпечення конфіденційності інформації у веб-додатках включає в себе механізми захисту на серверній стороні із використанням програмних

засобів, а також за допомогою процедур, що створюються на етапі розробки додатка.

НД ТЗІ 2.5 – 010 – 03 визначає наступні мінімально необхідні рівні послуг безпеки для забезпечення захисту інформації від загроз:

а) за умови, коли веб-сервер і робочі станції розміщуються на території установи-власника веб-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;

б) за умови, коли веб-сервер розміщується у оператора, а робочі станції – на території власника веб-сторінки, взаємодія яких з веб-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 Т2 є прикладом веб-додатку, оскільки спосіб передачі інформації від робочої станції, а саме клієнта веб-додатка до веб-сервера включає наявність незахищеного середовища тобто мережі Internet, яке не контролюється, і наявністю додаткових вимог щодо ідентифікації та автентифікації між робочої станції й КЗЗ веб-сервера під час спроби розпочати обмін інформацією та забезпеченням цілісності інформації при обміні.

Таблиця 2.3 – Профіль захищеності веб-серверів

Критерії	Опис критеріїв
Критерій конфіденційності	
КА-2	Базова адміністративна конфіденційність
КВ-1	Мінімальна конфіденційність при обміні
Критерій цілісності	
ЦА-1	Мінімальна адміністративна цілісність
ЦО-1	Обмежений відкат

ЦВ-1	Мінімальна цілісність при обміні
Критерії	Опис критеріїв
Критерій доступності	
ДВ-1	Ручне відновлення після збоїв
ДР-1	Використання ресурсів
Критерій спостережності	
НР-2	Реєстрація
НИ-2	Ідентифікація і автентифікація
НК-1	Однонаправлений достовірний канал
НО-1	Розподіл обов’язків
НЦ-1	Цілісність комплексу засобів захисту
НТ-1	Самотестування за запитом
НВ-1	Автентифікація при обміні

Цей профіль захищеності є базовим для використання та за власником залишається право реалізації, у разі необхідності, окремих послуг безпеки інформації зазначених профілів з більш високим рівнем, доповнення цих профілів іншими послугами, а також реалізація послуг безпеки з більш високим рівнем гарантій [13-15].

2.5 Реалізації функціональних послуг безпеки

Базова адміністративна конфіденційність

КЗЗ повинен реалізувати рівень КА-2.

Ця послуга дозволяє адміністратору безпеки керувати потоками інформації від захищених об'єктів до користувачів. Тобто для веб-додатка цей рівень повинен забезпечувати передачу конфіденційної інформації у вигляді відповіді веб-сервера до клієнта.

КЗЗ повинен здійснювати розмежування доступу на підставі атрибутів доступу користувача і захищеного об'єкта, тобто на підставі представлення певних ідентифікаторів у запиті зі сторони клієнта, що дозволяють обробляти його на стороні сервера та відповідати користувачу за його ідентифікатором.

Цей механізм можна реалізувати за допомогою механізму сесій, що існує у веб-додатках після аутентифікації користувача у базі даних.

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 чи імпорту через незахищене середовище.

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

Для забезпечення цього рівня веб-додаток повинен мати механізми, що шифрують чи захищають запити та відповіді на них від клієнта до сервера і у зворотному напрямку. Такими механізмами є процедури використання протоколів HTTPs, що є модифікацією базового протоку передачі даних від веб-додатка до клієнта, що використовує шифровані транспортні механізми TSL и SSL.

Мінімальна адміністративна цілісність

КЗЗ повинен реалізувати рівень ЦА-1.

Ця послуга дозволяє керувати потоками інформації від користувачів до захищених об'єктів веб-сторінки.

КЗЗ повинен здійснювати розмежування доступу на підставі атрибутів доступу користувачів і захищених об'єктів. Розмежування доступу здійснюється на рівні надання користувачеві прав модифікувати об'єкт. Прикладом є той самий механізм сесій з включенням до неї певних додаткових атрибутів.

Права доступу до захищених об’єктів веб-додатка повинні встановлюватися в момент їх створення або ініціалізації. На етапі розробки потрібно чітко регламентувати до якої інформації клієнтські додатки повинні мати доступ.

Цілісність при обміні

КЗЗ повинен реалізувати рівень ЦВ-1.

Ця послуга дозволяє забезпечити захист веб-сторінки від несанкціонованої модифікації інформації, яка передається між веб-сервером та робочими станціями під час експорту чи 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.

Цілісність при обміні для веб-додатку забезпечується за допомогою шифрування даних, які циркулюють по каналу зв’язку.

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.

До складу АС, що є системою що забезпечує функціонування веб-додатка, повинні входити автоматизовані засоби, які дозволяють адміністратору безпеки, користувачу, який має повноваження щодо управління АС, відкатити або відмінити певний набір операцій, виконаних над захищеним об’єктом веб-сторінки за певний проміжок часу. Тобто для забезпечення цього рівня системи, що забезпечують роботу додатка, повинні мати backup-сервери баз даних, що повинні містити дані для оперативного відновлення нормального функціонування додатка. Цей рівень забезпечує безперервну роботу веб-додатка, що впливає на його доступність та цілісність.

Використання ресурсів

КЗЗ повинен реалізувати рівень ДР-1.

Ця послуга дозволяє керувати використанням користувачами послуг та ресурсів.

Розмежування доступу користувача до інформації, що циркулює на стороні сервера здійснюється на етапі розробки клієнта, що повинен не мати доступу до серверної частини, окрім запитів з існуючих форм.

Некоректні запити та використання механізму автентифікації у якості забезпечення доступу до цієї інформації повинні реєструватися та блокуватися, тобто веб-додаток повинен мати механізм перевірки запитів на їх коректність. Забезпечення цього рівня безпеки дозволяє зменшити вразливість від атак класу авторизація.

Цей критерій для веб-додатка забезпечує його доступність. Для виконання цього критерії можливо використання механізмів конфігурування веб-серверів. Для веб-додатків цей критерій забезпечується за конфігурації перезапису mod_rewrite, за допомогою якого сервер модифікує URL при їх завантаженні.

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 КСЗІ; 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 проектування незадекларованих функцій), іншими непередбачуваними ситуаціями.

Після відмови веб-сторінки або переривання обслуговування, КЗЗ повинен перевести веб-сторінку до стану, з якого повернути її в режим нормального функціонування може тільки адміністратор безпеки і користувачі, які мають повноваження щодо управління АС.

Цей рівень для додатків можливо забезпечити за допомогою включення до АС додаткових серверів, що є копіями веб-серверів та серверів баз даних веб-додатку, за допомогою яких можна відновити нормальну роботу серверної частини і обробляти запити зі сторони клієнта.

Реєстрація

КЗЗ повинен реалізувати рівень НР-2.

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

Критерії реєстрації є базовою потребою для веб-додатка, оскільки забезпечує конфіденційність роботи клієнта з інформації, яка знаходиться на сервері.

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, ім’я та ідентифікатор причетного до цієї події користувача. Реєстраційна інформація повинна бути достатньою для однозначної ідентифікації користувача, процесу і об'єкта, що мали відношення до кожної зареєстрованої події.

Реєстрація повинна бути реалізована як механізм на серверній частині веб-додатку у вигляді додаткового модуля. Цим модулем може бути як програмний firewall чи модуль конфігурації сервера, що ідентифікує користувача, а також перевіряє коректність його запитів до сервера.

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 веб-

сторінки. Для веб-додатку цей критерій забезпечує конфіденційність, оскільки встановлення зв'язку на базі ідентифікатора та зіставлення його з певним клієнтом є механізмом створення безпечного конфіденційного сеансу.

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

Дозвіл на виконання будь-яких дій з інформацією та обладнанням веб-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.

Цей механізм можна реалізувати за допомогою механізму встановлення сесій, а також додаткових механізмів ідентифікації, коли клієнт представляє веб-додатку певний електронний ключ чи інший елемент, що може встановити особу користувача.

Для сучасних веб-додатків можливий додатковий модуль для підтвердження свого ідентифікатора з використанням мобільних телефонів, за допомогою якого проходить процедура автентифікації.

Ідентифікація і автентифікація при обміні

КЗЗ повинен реалізувати рівень НВ-1.

Ця послуга дозволяє компонентам КЗЗ веб-сервера і віддаленої робочої станції здійснити взаємну ідентифікацію, перш ніж розпочати взаємодію.

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 для веб-додатка та його компонентів КЗЗ повинна здійснюватися на підставі їхніх імен, паролів чи ідентифікаторів встановлених сесій.

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.

Достовірний канал повинен використовуватися для початкової ідентифікації і автентифікації, тобто створювати цілісний та доступний канал для встановлення сеансу між клієнтом та сервером. Зв'язок з використанням даного каналу повинен ініціюватися виключно користувачем.

Для забезпечення цього рівня веб-додаток повинен мати механізми, що шифрують чи захищають запити та відповіді на них від клієнта до сервера і у зворотному напрямку. Такими механізмами є процедури використання протоколів HTTPS, що є модифікацією базового протоку передачі даних від веб-додатка до клієнта, що використовує шифровані транспортні механізми TSL и SSL.

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.

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. Цей механізм додатку допомагає клієнту мати певний рівень доступності

до нього та створюється на етапі реєстрації користувача додатка, коли до бази даних користувач потрапляє з певними повноваженнями чи обмеженнями користуванням веб-додатком, інформацією, що циркулює у ньому чи інших компонентів серверної частини додатка.

Цілісність комплексу засобів захисту

КЗЗ повинен реалізувати рівень НЦ-1.

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

КЗЗ повинен повідомляти адміністратора безпеки про порушення цілісності будь-якого компонента КЗЗ. веб-сторінка під час цього має бути переведена до стану, в якому доступ до неї користувачів, крім адміністратора безпеки, заборонено.

Компоненти КЗЗ веб-додатку повинні у разі виключної ситуації звертатися до еталонних компонент, що дозволяють нормально функціонувати серверній частині. Це реалізується за допомогою використання серверів, де знаходяться копії додатка, які є еталонними.

Самотестування

КЗЗ повинен реалізувати рівень НТ-1.

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

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

Наявність тестів у вигляді програмного забезпечення дозволяє створювати ефективний КЗЗ, оскільки тести дозволяють знайти вразливості додатка та за допомогою власних механізмів закрити напрями атак на серверну частину з боку клієнта.

Розробка методики тестування захищеності веб-додатка і є найбільш значимою. Кожний з веб-серверів та додатків використовує лише тестування системи на предмет інтеграції до них вірусів чи інших елементів, що можуть зупинити роботу АС.

2.6 Критерії гарантій сторінок сайту

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.

Архітектура

Програмне забезпечення, призначене для реалізації КЗЗ, повинне будуватися за модульним принципом.

Склад послуг безпеки, а також механізмів захисту, що реалізують кожну з послуг, визначається політикою безпеки інформації в АС і повинен відповідати її вимогам. Якщо не всі вимоги політики безпеки реалізуються КЗЗ, то вони повинні підтримуватися організаційними та іншими заходами захисту КСЗІ. У складі КЗЗ не повинні міститися послуги та використовуватися засоби, які мають не передбачені політикою безпеки функції. Використання таких засобів можливе за умови вилучення цих функцій або гарантування неможливості їх активізації.

Мають бути описані особливості архітектури компонентів КСЗІ та їх призначення. Стиль опису – неформалізований, вимоги щодо детального опису не висуваються.

Середовище розробки

Мають бути визначені всі стадії та етапи життєвого циклу АС, а для кожної стадії та етапу – перелік і обсяги необхідних робіт та порядок їх

виконання. Всі стадії та етапи робіт повинні бути задокументовані. Види та зміст документів встановлено державними стандартами.

На всіх стадіях життєвого циклу повинні існувати процедури керування конфігурацією АС. Ці процедури повинні визначати технологію відслідковування та внесення змін в апаратне та програмне забезпечення КСЗІ, тестове покриття і документацію та гарантувати, що без дотримання цієї технології ніякі зміни не можуть бути внесені. Технологія слідкування та внесення змін повинна гарантувати постійну відповідність між документацією і реалізацією поточної версії КЗЗ.

Послідовність розробки

Для всіх стадій життєвого циклу АС повинні бути розроблені функціональні специфікації КСЗІ.

На підготовчому етапі створення КСЗІ має бути виконане обстеження середовищ функціонування АС, в результаті якого визначаються об'єкти захисту, здійснюється класифікація інформації та розробляється модель загроз для інформації і концепція політики безпеки інформації в АС. На підставі цих даних мають бути сформульовані функціональні специфікації вимог із захисту інформації в АС. Ці специфікації мають бути викладені в окремому розділі в технічному завданні на створення АС або окремому технічному завданні на створення КСЗІ.

Функціональні специфікації політики безпеки і моделі політики безпеки повинні містити перелік і опис послуг безпеки, що надаються КЗЗ, а також правила розмежування доступу до захищених об'єктів АС.

Функціональні специфікації проекту архітектури КСЗІ повинні містити модель захисту, де враховані всі суттєві загрози і для кожної з них визначено можливі варіанти їх блокування (попередження) за допомогою КЗЗ або організаційними чи іншими заходами захисту. Якщо існує неоднозначність, повинні надаватися додаткові аргументи на користь вибору того чи іншого варіанту.

Функціональні специфікації детального проекту КСЗІ повинні містити принципи побудови, функціональні можливості, опис функціонування кожного механізму захисту та взаємодії механізмів між собою у складі КЗЗ. Повинні бути розроблені документи, що регламентують використання засобів КЗЗ, а також організаційних та інших заходів захисту, які входять до КСЗІ. Як реалізація детального проекту може розглядатися технічний або робочий проекти.

Окремі етапи робіт повинні бути задокументовані відповідно до вимог НД ТЗІ 1.4-001-2000 у вигляді окремих розділів плану захисту інформації в АС або вимог інших нормативно-правових актів і нормативних документів з ТЗІ.

Середовище функціонування

Повинні існувати засоби інсталяції, генерації і запуску КЗЗ, які гарантують, що експлуатація АС починається з безпечного стану, а також існувати документи (інструкції), які регламентують порядок керування цими процедурами. Якщо можливі різні варіанти конфігурації КЗЗ, то всі вони повинні бути описані в інструкціях.

Документація

Документація на КЗЗ у вигляді окремих документів або розділів інших документів повинна містити опис послуг безпеки, що реалізуються КЗЗ, а також настанови для різних категорій користувачів (адміністратора безпеки, адміністратора баз даних, адміністратора сервісів, звичайного користувача тощо) стосовно використання послуг безпеки.

Випробування

Випробування КЗЗ можуть проводитись як самостійно, так і у складі КСЗІ. Для проведення випробувань розробник КЗЗ повинен підготувати програму і методику випробувань, розробити процедури (тести) випробувань усіх механізмів, що реалізують послуги безпеки.

Програма і методика випробувань КЗЗ, тестове покриття, результати випробувань КЗЗ входять до складу обов'язкового комплекту документації, яка надається організатору експертизи під час проведення державної експертизи КСЗІ в АС [14 – 15].

2.7 Види атак на веб сервери

DoS/DDoS атака

Зловмисники можуть надсилати на веб-сервер численні фальшиві запити, які призводять до збою веб-сервера або стають недоступними для законних користувачів.

Зловмисники можуть націлюватися на високопрофесійні веб-сервери, такі як банки, шлюзи платіж за кредитними картками, державні послуги тощо.

Взлом DNS-сервера

Нападник компрометує DNS-сервер і змінює налаштування DNS, щоб весь запит, що йшов до цільового веб-сервера, повинен бути перенаправлений на його власний шкідливий сервер.

DNS ампліфікація

Оскільки в протоколі UDP не провадиться перевірка IP-адрес джерела, зловмисник генерує запити від імені сервера-жертви, вказуючи його IP-адресу в поле вихідного адреси. Основною метою зловмисника є заповнення каналу сервера-жерт ви об'ємними відповідями від публічних DNS-серверів.

Атака на обхід каталогу

У випадках атак на перехоплення каталогу хакери використовують послідовність ../ (dot-dot-slash) для доступу до обмежених каталогів за межами корневого каталогу веб сервера.

Man-in-the-Middle/Sniffing атака

Нападки "людина в середині" (MUM) дозволяють зловмиснику отримати доступ до конфіденційної інформації, перехоплюючи та змінюючи зв'язок між кінцевим користувачем та веб-серверами. Зловмисник виступає як проксі таким чином, що все спілкування між користувачем і сервером проходить через нього

Фішинг атака

Нападник вказує користувачеві можливість відправляти дані про вхід для веб-сайту, який виглядає законним, але переспрямовує на шкідливий веб-сайт, розміщений на веб-сервері нападника. Нападник вловлює введені облікові дані та використовує його для висування себе за допомогою веб-сайту, розміщеного на законному цільовому сервері. Нападник може виконувати неавторизовану або зловмисну операцію із цільовим сервером веб-сайту

Атаки веб-додатків

Вразливості веб-додатків, що працюють на веб-сервері, забезпечують широкий шлях атаки для компромісу веб-серверів.

Порушення сайтів

Виправлення в Інтернеті відбувається, коли вторгнення зловмисне змінює вигляд веб-сторінки вставляючи або замінюючи прогностичні та часто суперечливі дані. Дефектні сторінки призводять до відвідування певної пропаганди або вводять в оману інформацію, доки не буде виявлена. Зловмисники використовують різноманітні методи, наприклад, ін'єкцію MYSQL для доступу та зміни сайту.

Неправильна конфігурація веб-сервера

Неправильна конфігурація сервера стосується недоліків конфігурації веб-інфраструктури, які можуть використовуватися для запуску різних атак на веб-серверах, таких як переміщення каталогів, вторгнення серверів та крадіжку даних.

Атака поділу відповіді HTTP

Атака розщеплення HTTP-відповіді полягає у додаванні даних відповідей заголовка у поле вводу, щоб сервер розділив відповідь на дві відповіді. Зловмисник може контролювати другу відповідь, спрямовуючи користувача на шкідливий веб-сайт, тоді як інші відповіді будуть відкинуті веб-переглядачем.

Отрута кешу веб сервера (Web Cache Poisoning Attack)

Зловмисник змушує кеш-пам'ять веб-сервера скинути фактичний вміст кешу і надсилає спеціально створений запит, який буде зберігатися в кеш-пам'яті.

Атака SSH Bruteforce

SSH-протоколи використовуються для створення зашифрованого тунелю SSH між двома хостами для передачі незашифрованих даних через незахищену мережу. Зловмисники можуть отримати перебором облікові дані SSH для отримання несанкціонованого доступу до тунелю SSH. SSH тунелі можуть бути використані для передачі шкідливих програм та інших експлуатацій жертвам без виявлення їх існування.

Взлом паролю Web сервера

Зловмисник намагається використовувати слабкі місця для зриву добре вибраних паролів. Найпоширеніші знайдені паролі - пароль, root, адміністратор, адміністратор, демонстрація, тест, гість, qwerty, імена тварин тощо. Зловмисники використовують різні методи, такі як соціальна інженерія, підроблення, фішинг, використання троянських коней або вірусів, прослуховування під телефон, запис натискання клавіш тощо. Багато спроб злому починаються з руйнування паролів і доводять веб-серверу, що вони є дійсними користувачами.

2.8 Методологія атак на веб-сервери.

2.8.1 Збір інформації за допомогою веб-сервісів

- Збір інформації передбачає збирання інформації про цільову компанію
- Зловмисники шукають інформацію про компанію в Інтернеті, групах новин, дошках оголошень тощо
- Зловмисники використовують інструменти, щоб отримати деталі, такі як доменне ім'я, IP-адреса тощо.

2.8.2 Збір інформації з файлу Robots.txt

- Файл robots.txt містить список каталогів і файлів веб-сервера, які власник веб-сайту хоче сховати від веб-сканерів.
- Нападник може просто замовити файл Robots.txt з URL-адреси та отримати конфіденційну інформацію, таку як структура кореневого каталогу, інформацію про систему керування вмістом тощо, про цільовий веб-сайт.

2.8.3 Footprinting веб сервера

Збирання цінних даних на рівні системи, такі як дані про облікові записи операційну систему, версії програмного забезпечення, імена серверів та деталі схеми бази даних.

2.8.4 Віддзеркалення веб-сайту

- Дзеркало веб-сайту, щоб створити повний профіль структуру каталогу сайту, структури файлів, зовнішніх посилань тощо.
- Пошук коментарів та інших елементів у вихідному коді HTML

Сканування уразливості

- Виконуються сканування уразливості для виявлення слабких місць у мережі та визначення того, чи можна використовувати систему
- Знімається мережевий трафік, щоб дізнатись про наявність активних систем, мережевих служб, програм та вразливостей

- Перевіряється інфраструктура веб-сервера для будь-яких неправильних конфігурацій, застарілого вмісту та вразливостей

2.8.5 Взлом сесії

- Змінюються дійсні ідентифікатори сеансів, щоб отримати несанкціонований доступ до веб-сервера і переглянути дані.
- Використовуються методи вилучення сеансів, такі як фіксація сеансів, послідовне з'єднання, міжсторінкові сценарії та ін., Щоб отримати правильні файли cookie-файлів та ідентифікатори сеансу.

2.8.6 Взлом паролю веб-сайту

- Використовується методи взлому паролем, такі як атака з перебором слів і знаків , атака за допомогою словників тощо з метою підвищити повноваження на сервері .

2.9 Інструментарій що використовується при атаках.

Відповідно до завдання роботи був визначений інструментарій що використовується при атаках на Web сервери. Результати були представлені в таблиці 2.2

Таблиці 2.2 Інструментарій що використовується при атаках на веб-сервери

Методологія атаки	Назва інструменту	Тип інструменту	Мета атаки
1	2	3	4
Збір інформації	Whois, Traceroute, Active Whois, сканери	Веб сервіси Інтернету	Отримання деталей, таких як доменне ім'я, IP-адреса, прізвища співробітників тощо

	соціальних мереж.		
Footprinting Веб-сервера	• ID Serve • Httpre • Netcraft • **Zenmap**	Сканер безпеки	• дані про облікові записи • операційна система, • версії програмного забезпечення, • імена серверів та тип та деталі схеми бази даних.
Віддзеркалення веб-сайту	• HTTrack, WebCopier Pro, • BlackWidow	Mirror Tools	• Структуру каталогу сайту, • структури файлів та зовнішні посиланя • Пошук коментарів та інших елементів у вихідному коді HTML
Взлом сесії	• **Burp Suite,** • Firesheep, • JHijack	Веб сервіси Інтернету	Змінити дійсні ідентифікатори сеансів щоб отримати несанкціонований доступ до веб-сервера і переглянути дані. Фіксація сеансів щоб отримати правильні файли cookie-файлів та ідентифікатори сеансу.

Взлом паролю веб-сайту	• Brutus • Internet Password Recovery Toolbox • THC-Hydra.	Password Cracker	Дані про пароль адміністратора системи, адміністратора сайта, паролі облікових записів користувачів з метою підвищити повноваження на серверів.
Сканування уразливості	• HP WebInspect, • Web Acunetix	Сканер безпеки	• Слабкі місця у мережі, • наявність активних систем, мережевих служб, програм та вразливостей • інфраструктура веб-сервера

2.10 Аналіз вразливості HTTP

Вразливість HTTP Parameter Pollution (HPP) дозволяє зловмиснику вставляти параметри всередину URL-ів, що генеруються веб-додатком. Результат цієї атаки залежить насамперед від логіки роботи додатка та може змінюватися від простих незручностей до повної зміни поведінки веб-додатку.

Наприклад, розглянемо типові HTTP GET та POST – запити.

Приклад HTTP GET – запиту:

GET /foo?par1=val1&par2=val2 HTTP 1.1

User-Agent: Mozilla 39.0

Host: Host

Accept: */*

Приклад HTTP POST – запиту:

POST /foo HTTP 2.0

User-Agent: Mozilla 39.0

Host: Host

Accept: */*

Content-Length: 19

par1=val1&par2=val2c

В обох випадках вразливими є параметри запитів - par1=val1&par2=val2

Незважаючи на те, що додавання нового параметру може інколи бути достатнім для здійснення атаки на веб-додаток, зловмисник, як правило, більш зацікавлений у зміненні значення вже існуючого параметру. Це може бути досягнене із використанням так званого «маскування» існуючого параметру – додаванням нового параметру з таким же самим ім'ям [19].

В даному випадку наведений вище HTTP GET – запит буде включати в себе такі параметри:

par1=val1& par1=val3&par2=val2

Яким чином веб – сервер буде обробляти подібний запит залежить насамперед від типу веб – серверу.

2.6.2 Пріоритет параметрів HTTP-GET та HTTP-POST запитів

Для більш повного розуміння вразливості HPP потрібно розглянути поняття пріоритету параметру у HTTP GET та POST – запитах.

Протягом взаємодії із веб-додатком користувачу часто потрібно ввести деякі дані до програми, що потім згенерує веб-сторінку, що була потрібна. Протокол HTTP дозволяє браузеру користувача передавати інформацію всередині самого URL (GET параметри), у заголовках HTTP – запитів (поле Cookie) та всередині тіла запиту (POST параметри) . Використання того, чи

іншого способу передачі інформації залежить від веб-додатку, а також від типу та кількості даних, що передаються [20].

У типовій реалізації, веб-сторінки, всі пункти чек боксу мають однакове ім'я, і тому в подальшому веб-браузер буде відправляти окремий параметр для кожного пункту, який був обраний користувачем. Для підтримки цієї функціональності, більшість мов програмування, які використовуються для розробки сторінок, використовують методи, що отримують весь список значень для конкретного параметру [20].

Наприклад JSP має метод:

String[] parameterValues = request.getParameterValues();

Цей метод збирає всі значення параметру та повертає список строк.

Але проблема постає, коли розробник очікує прийняти одне значення для одного параметру.

Наприклад:

String variableName = requset.getParameter("txtUserName");

Даний метод повертає лише одне значення. Припустимо, що даному методу в запиті передається декілька параметрів з однаковими іменами та різними значеннями. В такому випадку метод getParameter може повернути перше, останнє або комбінацію значень параметрів, що передаються у запиті.

На даний момент результат подібних запитів залежить від мови програмування, яка застосовується для розробки веб-додатку та веб-сервера, який обробляє дані запити [21].

Важливо відзначити той факт, що веб-додаток повертає тільки одне значення не є вразливістю. Але якщо веб – розробник не зверне увагу на дану проблему, в майбутньому присутність дубльованих параметрів у запитах може спричинити аномальну поведінку веб-додатку, а також може бути потенційно використане зловмисником у комбінації з іншими видами атак [17].

Існує декілька видів вразливості HTTP Parameter Pollution, але основним видом вважається вразливість на стороні клієнта.

2.11 Вразливість сервера при HTTP запиті на стороні клієнта

Суть цієї атаки полягає у переконанні жертви перейти по зловмисному URL, що використовує вразливість HPP. Для прикладу, розглянемо веб-додаток, що дозволяє користувачеві віддати свій голос у різних типах голосувань. веб-додаток, написаний із застосування технології Java Server Pages (JSP), отримує єдиний параметр, під назвою poll_id, який унікально ідентифікує голосування, в якому користувач бере участь в даний момент. На основі значення цього параметру, веб-додаток генерує сторінку, що включає в себе список посилань, кожна з яких дозволяє голосувати за одного кандидата.

Наприклад, наступний рисунок (Рисунок 2.1) ілюструє сторінку для голосування, що включає в себе двох кандидатів, на якій користувач може віддати свій голос за того, чи іншого кандидата, натисканням на бажане посилання:

```
URL: http://host/election.jsp?poll_id=4568
Link 1: <a href="vote.jsp?poll_id=4568&candidate=white"> Vote for mr.White</a>
Link 1: <a href="vote.jsp?poll_id=4568&candidate=green"> Vote for ms.Green</a>
```

Рисунок 2.1 – Код посилань на тестовій сторінці для голосування

Припустимо, що є зловмисник, який зацікавлений у перемозі мс. Грін. Аналізуючи веб-сторінку, він розуміє, що , веб-додаток належним чином не перевіряє параметр poll_id. В такому випадку, зловмисник може використати вразливість HPP для додавання іншого параметру до голосування.

Для того, щоб реалізувати атаку, зловмисник генерує модифіковане посилання. Модифіковане посилання наведено на рисунку 2.3:

Рисунок 2.2 – Модифікований зловмисником запит

Після цього модифіковане посилання відправляється потенційній жертві. Користувач, що переходить по даному зміненому Url буде пере- направлений на оригінальну веб-сторінку, де зможе віддати свій голос у голосуванні.

Проте, параметр poll_id використовується веб-додатком для генерації посилання на сторінці. Тому після переходу по модифікованому Url, зловмисне значення candidate буде вставлене в усі посилання на сторінці (рисунок 2.3):

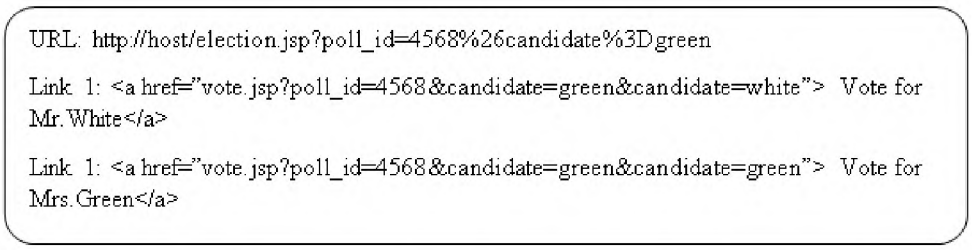

Рисунок 2.3 – Сторінка голосування із модифікованими посиланнями

В цьому разі неважливо, на яке посилання натисне користувач - веб-додаток (в нашому випадку JSP-скрипт vote.jsp) буде отримувати два параметри із назвою candidate. Крім того, перший параметр завжди буде мати значення green.

Можна зробити припущення, що розробник даного веб-додатку очікував отримати від користувача тільки одне ім'я кандидата. До того ж, специфіка роботи даного веб-додатку, написаного із застосуванням технології Java Server Pages, полягає в тому, що коли отримані у запиті параметри обробляються на стороні сервера, тільки перший параметр з іменем candidate буде прийнятий. Другий параметр з таким же іменем буде відкинутий.

В даному випадку після модифікації посилання зловмисником, користувач, незалежно від свого вибору, буде голосувати за мс. Грін.

Роблячи висновок, потрібно зазначити, що якщо веб-додаток є вразливим до HPP, зловмиснику стає можливим модифікувати Url таким чином, що після одного переходу по даному Url, зміст веб-сторінки буде змінений – усі посилання будуть вести до голосування за користувачку. Брук.

2.12 Тестування захищеності веб-серверу. Планування.

Згідно з завданнями дипломної роботи було проведено тестування веб сервера. Для цього був розроблений план тестування.

Джерело уразливості:

помилки серверного програмного забезпечення, ОС та веб- додатків

- Вид атаки :
  - атака поділу відповіді HTTP
- Методологія атаки :
  - взлом сесії
- Жертва атаки :
  - віртуальна машина під управлінням Windows 10
- Атакований web сервер:
  - WAMP Server (Windows, Apache, MySQL, PHP)
- Об'єкт атаки:
  - учбовий веб-сайт «Your Voice»
- Інструментарій атаки:
  - сканер уразливості Zenmap, інспектор мережевого трафіку та проксі-сервер WebScarab

2.13

Дана методика представляє собою алгоритм тестування та рекомендації, щодо проведення перевірки захищеності веб-додатків. Методика базується на

детальному вивченні даних про веб-додаток, про http-запити та http-відповіді від сервера, що супроводжують роботу додатку. Запропонована методика передбачає застосування спеціалізованого програмного забезпечення для модифікації http-запитів та дослідження системи.

Загальний алгоритм тестування згідно запропонованої методики зображений на рисунку 2.5.

Спеціалізоване програмне забезпечення, що використовується у додатку включає в себе:

- Zenmap – сканування додатку;
- Burpsuite – перехоплення та модифікація запиту;
- WebScarab – проксі-сервер для запису модифікованого запиту.

Можливо використовувати інші проксі-сервера (наприклад Parus Proxy) та засоби сканування додатку.

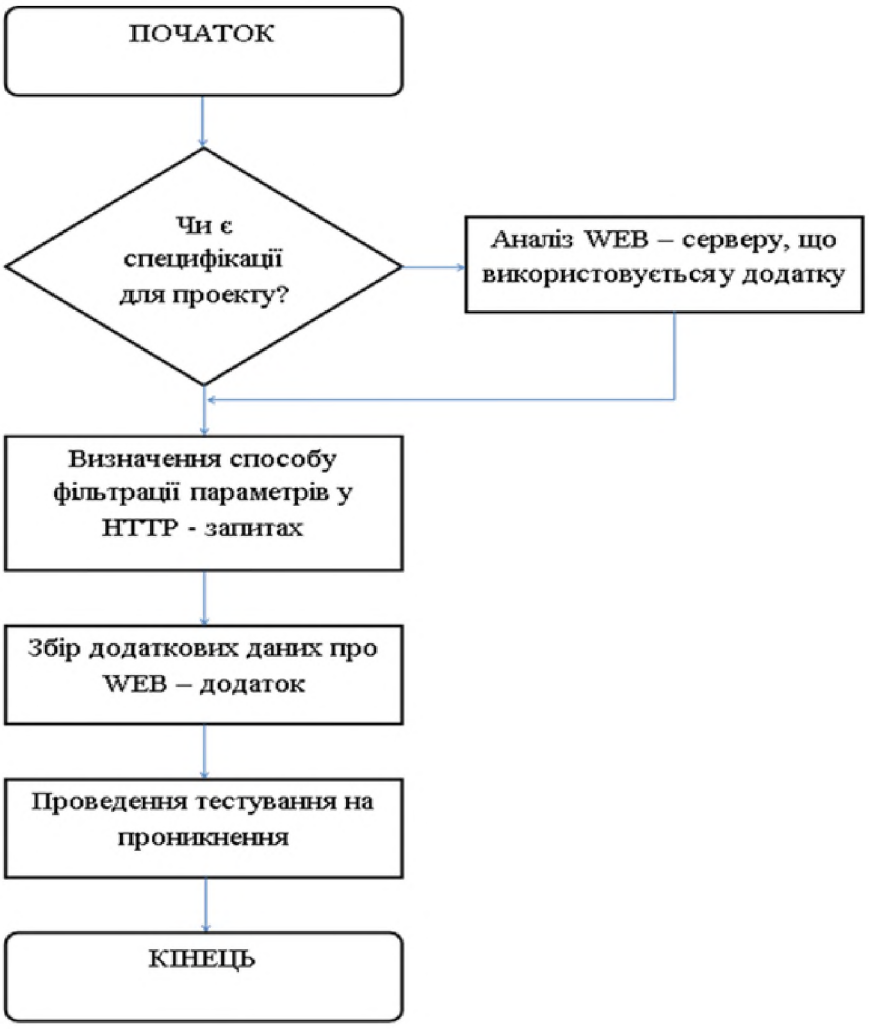

Рисунок 2.5. Загальний алгоритм тестування захищеності на вразливість HTTP

2.14 Протидія атаці. Аналіз веб-серверу, що використовується у додатку

Для проведення тестування захищеності веб – додатку насамперед потрібно визначити (або отримати інформацію) щодо веб – серверу, який обробляє HTTP – запити від користувачів.

Веб-сервер, що розглядається, типу WAMP:

- **Windows** – операційна система,
- **Apache** - сервер,
- **MySql** – система управління базами даних,
- **PHP** – мова програмування, що застосовується для створення веб-додатків.

WAMP Сервер — це заздалегідь налаштований сервер, який має в собі мінімальний набір програмного забезпечення, що пришвидшує розробку проектів для розробників та роботу дизайнерів, а також має гнучке налаштування.

В загальному випадку при проведенні тестування можливі два випадки:

1. Вся потрібна інформація щодо веб – додатку надана замовниками тестування. В цьому випадку потрібно проаналізувати отримані дані та виділити ключові факти, які будуть потрібними при подальшому проведенні тестування.

Насамперед це:

- Тип та версія веб – серверу, що використовується в даному додатку;
- Технології, що застосовувалися при розробці веб – додатку;

2. Всієї потрібної для тестування інформації замовниками не надано.

В цьому випадку потрібно застосовувати допоміжне програмне забезпечення – програми Nmap(ZenMap) та BurpSuite.

Розглянемо отримання необхідної інформації на прикладі програми Nmap(ZenMap).

Zenmap 5.61 – утиліта для дослідження мережі та перевірки безпеки. Методика тестування за допомогою Zenmap заснована на використанні IP-пакетів з різними властивостями, які можуть допомогти для визначення доступних хостів у мережі, служб, операційних систем, які обслуговують ці хости. Також дані пакети дозволяють визначити типи пакетних фільтрів або брандмауерів, що допомагає досліджувати механізми захисту додатків.

Вихідними даними Zenmap є список просканованих цілей з додатковою інформацією по кожній в залежності від заданих опцій. Ключовою інформацією є таблиця, де міститься інформація про доступність портів і хостів. Ця таблиця містить номер порту, протокол, ім'я служби і стан. Стан може мати значення open, filtered, closed або unfiltered. Відкрито означає, що додаток на цільовій машині готовий для з'єднання або прийняття пакетів на цей порт. Фільтрується означає, що брандмауер, мережевий фільтр або якась інша перешкода в мережі блокує порт, і утиліта не може встановити відкритий цей порт або закритий. Закриті порти не пов'язані ні з додатками, так що вони можуть бути відкриті в будь-який момент.

Порти розцінюються як не фільтровані, коли вони відповідають на запити, але утиліта не може визначити, відкриті вони або закриті. Zenmap видає комбінації, коли порт відкритий і фільтрується або закритий, але також фільтрується, коли не може визначити, яке з цих двох станів описує порт. Ця таблиця також може надавати деталі про версію програмного забезпечення, якщо це було запитано.

На додаток до таблиці важливих портів утиліта може надавати подальшу інформацію про цілі: перетворені DNS імена, припущення про використовуваної операційній системі, типи пристроїв і MAC адреси.

Zenmap використовує безліч різних методів сканування, таких як UDP, TCP (connect), TCP SYN (напіввідкрите), FTP proxy (прорив через ftp), Reverse-ident, ICMP (ping), FIN, ACK, Xmas tree, SYN та NULL сканування.

2.15 Протидія атаці. Збір додаткових даних про веб – додаток

Перед тим, як перейти до тестування на вразливість HTTP, необхідно отримати додаткову інформацію щодо роботи веб – додатку.

До цієї інформації належить:

- Дані про «нормальну» роботу веб – додатку. Під «нормальною» роботою розуміється всі HTTP – запит та відповіді від сервера, які виникають у процесі користування додатком.
- Дані щодо основних параметрів HTTP – запитів, що користуються у процесі роботи додатку.

Для дослідження «нормальної» роботи веб – додатку слід використовувати додаткове програмне забезпечення.

На рисунку 2.6 зображено принцип дослідження «нормальної» роботи веб – додатку.

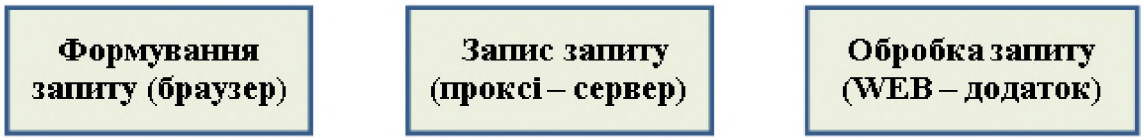

Рисунок 2.6 – Схема дослідження «нормальної» роботи веб – додатку

- В якості браузеру використовується один із найпоширеніших на сьогоднішній час браузерів: Mozilla Firefox, Google Chrome, Opera, Internet Explorer, Safari.
- В якості проксі – серверу може використовуватися як звичайні проксі-сервера (наприклад Paros Proxy), так і спеціалізоване програмне забезпечення для інспектування мережевого трафіку (WebScarab, BurpSuite), які мають функції проксі сервера.

Принцип дослідження згідно схеми, зображеної на рисунку 2.7, полягає у тому, що проводиться тестування з позиції користувача. Виконується перехід по посиланням, інші дії з позиції користувача, а усі HTTP – запити та HTTP – відповіді відображаються на проксі – сервері.

Після проведення дослідження є доступ до переліку HTTP – запитів та відповідей, які вважаються коректними для даного веб – додатку та не призводять до помилок у його роботі.

Після отримання переліку HTTP – запитів потрібно детально дослідити параметри, що в них передаються.

Особливу увагу слід приділити одиничним параметрам, що передаються у процесі роботи веб – додатку.

Наприклад:

http://127.0.0.1/mutillidae/index.php?page=userpoll.php&choice=nmap&initials=alexander&user-poll-php-submit-button=Submit+Vote.

У даному запиті передаються такі параметри:

choice=nmap – означає вибір користувача;

initials=alexander – дані про користувача;

Дані параметри слід дослідити, виконуючи тестування на проникнення.

2.16 Протидія атаці. Тестування захищеності веб – додатку «Ваша думка» на вразливість HTTP

За допомогою даного веб – додатку користувач може обрати будь – який мережевий сканер, який він використовує найчастіше (рисунок 2.11).

1. Згідно наданих специфікацій даний веб – додаток написаний із використанням технології PHP та застосовує веб – сервер Apache 2.4.2 або використовується сканування за допомогою Zenmap.
2. На основі даних, наведених у таблиці 1 робиться висновок, що для даної технології та типу серверу у HTTP – запитах фільтрується тільки останній параметр, а перший параметр відкидається.

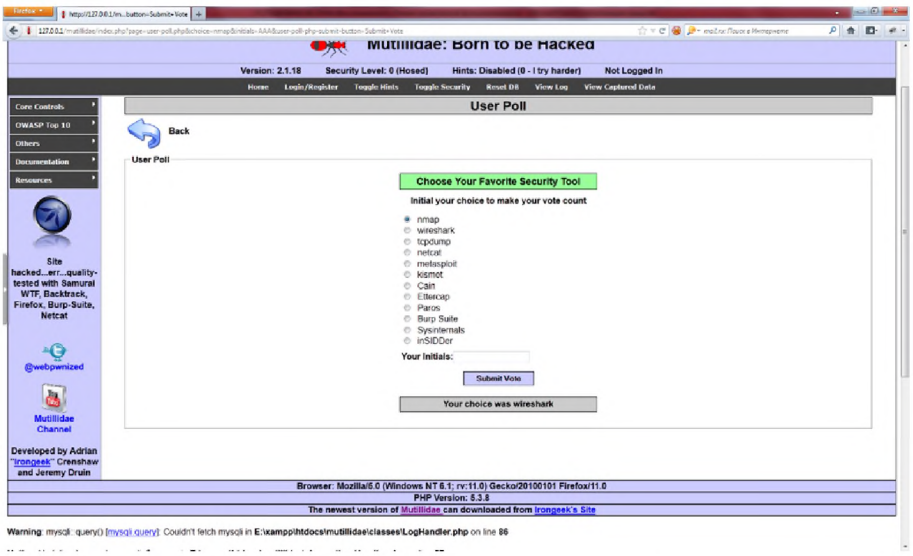

Рисунок 2.7 – Загальний вигляд додатку для тестування

3. Досліджується «нормальна» робота веб – додатку. Для цього використається проксі - сервер Paros Proxy. (або WebScarab). На рисунку 2.8 наведений перелік діючих запитів у додатку, отриманий за допомогою програмного продукту WebScarab:

Method	Host	Path	Parameters	Status
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=inSIDDer&initials=&user-poll-php-submit-button=Submit+Vote	200 OK
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=Sysinternals&initials=&user-poll-php-submit-button=Submit+Vote	200 OK
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=Burp+Suite&initials=&user-poll-php-submit-button=Submit+Vote	200 OK
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=Paros&initials=&user-poll-php-submit-button=Submit+Vote	200 OK
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=kismet&initials=&user-poll-php-submit-button=Submit+Vote	200 OK
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=metasploit&initials=&user-poll-php-submit-button=Submit+Vote	200 OK
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=netcat&initials=&user-poll-php-submit-button=Submit+Vote	200 OK
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=tcpdump&initials=&user-poll-php-submit-button=Submit+Vote	200 OK
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=nmap&initials=&user-poll-php-submit-button=Submit+Vote	200 OK
GET	http://127.0.0.1:80	/mutillidae/index.php	?page=user-poll.php&choice=wireshark&initials=&user-poll-php-submit-button=Submit+Vote	200 OK

Рисунок 2.8 – Перелік діючих запитів у веб – додатку

Схема роботи додатку:

Користувач обирає програмний продукт для тестування захищеності систем, який він найбільш часто використовує, із запропонованого списку, а також вносить дані про себе у поле «Your Initials». Після цього користувач підтверджує свій вибір натисканням кнопки «Submit Vote». В результаті коректної роботи додатку на сторінці відображається повідомлення із текстом «Your choice was …», де замість пропуску висвітлюється назва обраного

програмного продукту із списку. В залежності від обраного продукту, результуюче повідомлення буде змінюватися.

Наприклад:

Обирається варіант – nmap та заносяться дані про учасника голосування – Alex.

Формується запит:

http://127.0.0.1/mutillidae/index.php?page=user-poll.php&choice=nmap&initials=Alex&user-poll-php-submit-button=Submit+Vote

Результат виконання запиту відображається на рисунку 2.13:

Рисунок 2.9 Результат опросу у експериментальному додатку

Результуюче повідомлення означає, що був обраний варіант – nmap

1 Виконання тестування на проникнення.

1.1 Формується запит «2» із іншим варіантом для голосування – wireshark.

В результаті отримується запит 2:

http://127.0.0.1/mutillidae/index.php?page=user-poll.php&choice=wireshark&initials=Alex2&user-poll-php-submit-button=Submit+Vote

Згідно логіки роботи додатку користувач повинен бути пере направлений на сторінку, яка буде містити не тільки список для голосування, але й повідомлення, в якому буде записаний обраний варіант.

1.2 За допомогою програмного продукту BurpSuite виконується перехоплення запиту 2, що сформований браузером

1.3 Проводиться модифікація перехопленого запиту. Згідно даних, отриманих на попередніх кроках даної методики, додаток використовує сервер Apache та технологію PHP. Згідно таблиці пріоритетів параметрів визначається, що в даному експериментальному випадку буде фільтруватися останній параметр, а перший буде відкидатися.

Саме тому модифікація запиту буде включати в себе вставлення у строку запиту параметру choice із значенням - nmap.

Модифікований запит:

http://127.0.0.1/mutillidae/index.php?page=user-poll.php&choice=wireshark&choice=nmap&initials=Alex2&user-poll-php-submit-button=Submit+Vote

Результат виконання модифікованого запиту зображений на рисунку 2.10:

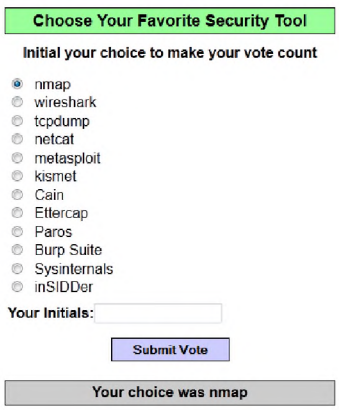

Рисунок 2.10 Результат виконання модифікованого запиту.

Результат виконання модифікованого запиту свідчить про наявність вразливості HTTP для параметру із іменем «choice», а також для веб – додатку в цілому.

2.17 Протидія атаці. Вимоги до протоколу тестування

Після проведення всіх етапів тестування захищеності кожен знайдений вразливий параметр зазначається у звіт про проведення тестування. Незалежно від форми подання, звіт повинен містити основну інформацію про дефект безпеки:

- Параметри системи, на якій проводилося тестування.
- Назва та версія веб-браузеру на якому виконувалося тестування.
- Версія веб-додатку, яка піддавалася тестуванню.
- Кожний параметр, який є вразливим для атаки HTTP.
- Для кожного вразливого параметру вказується повний шлях для репродукції даного дефекту.
- Дата тестування.

2.18 Висновок до другого розділу

Проблема інформаційної безпеки веб-додатку повинна вирішуватися на всіх етапах життєвого циклу додатка.

Тестування безпеки повинно проводитися як на етапі створення окремої функціональної одиниці додатку, так і на етапі системного тестування. Всі знайдені вразливості повинні розглядатися розробниками як дефекти у ПЗ – тому повинні бути усунуті якнайшвидше. Критичність дефектів безпеки повинна бути розподілена так само, як і критичність звичайних функціональних дефектів.

Комплексний підхід до тестування захищеності веб-додатків дозволить значно зменшити ймовірність порушення цілісності, доступності та

конфіденційності інформації, що циркулює у додатку, а також забезпечити логіку його роботи.

Запропонована методика може бути використана як набір рекомендацій для виявлення вразливостей типу HTTP Parameter Pollution, а також у подальшому може бути реалізована як додатковий модуль для вже існуючого сканера вразливостей веб-додатків.

РОЗДІЛ 3. ЕКОНОМІЧНА ЧАСТИНА

Метою виконання економічного розділу є визначення того, чи буде використання запропонованих засобів та заходів інформаційної безпеки вигідним для підприємства. Щоб з'ясувати це, необхідно визначити розмір капітальних та експлуатаційних витрат на заходи і засоби інформаційної безпеки, визначити величину відвернених втрат та, на основі цього, розрахувати коефіцієнт повернення інвестицій та термін окупності капітальних інвестицій. На основі розрахованих показників можна буде визначити, наскільки прибутковим або збитковим є запропонований проект.

3.1 Розрахунок капітальних витрат на придбання і налагодження складових системи інформаційної безпеки.

До фіксованих (капітальних) варто відносити наступні витрати на підприємстві:

- витрати на залучення зовнішніх консультантів (спеціаліста з розробки політики безпеки інформації);
- витрати на первісні закупівлі апаратного забезпечення (датчики розбиття вікон та датчики відкриття вікон, електронні замки та смарт-картки, сейфи для збереження носіїв інформації);
- витрати на інтеграцію системи інформативної безпеки у вже існуючу корпоративну систему (встановлення обладнання та налагодження системи інформаційної безпеки);

Для підрахунку заробітної платні залученого працівника, який створює або дороблює політику безпеки, необхідно розрахувати трудомісткість розробки політики безпеки інформації. Вона визначається тривалістю кожної робочої операції цього працівника:

$$t = t_o + t_a + t_в + t_д \text{ , годин,} \qquad (3.1)$$

де:

$t = 75o$ – тривалість проведення обстеження ІТС підприємства

$t_а = 14$ – тривалість процесу аналізу можливих загроз та ризиків;

$t_в = 20$ –тривалість визначення вимог до заходів, методів та засобів захисту, вибору основних рішень з забезпечення безпеки інформації;

$t_д = 12$ – тривалість документального оформлення політики безпеки.

t = 75+14+20+12=121 година.

У даному випадку, витрати на розробку політики безпеки інформації включають в себе лише заробітну плату робітника, який залучається для створення політики безпеки. В його оплату вже включено всі витрати, яких він зазнає (витрати на електроенергію, тощо). Виконавець не використовує у своїй роботі ноутбуки чи комп'ютери компанії. Заробітна плата виконавця враховує основну і додаткову заробітну плату, а також відрахування на соціальні потреби (пенсійне страхування, страхування на випадок безробіття, соціальне страхування тощо) та визначається за формулою:

$$К_{пр} = t \cdot З_{іб}, \text{ грн,} \qquad (3.2)$$

де:

t =121 – загальна тривалість розробки політики безпеки, годин;

$З_{іб} = 75$ – средньогодинна заробітна плата спеціаліста з інформаційної безпеки з нарахуваннями, грн/годину.

$$К_{пр} = 121 \cdot 75 = 9075 \text{ грн.}$$

Таким чином, капітальні (фіксовані) витрати на проектування та впровадження проектного варіанта системи інформаційної безпеки складають:

$$К = K_{пр} + K_{др} + K_{дв} + K_{ез} + K_{ск} + K_{с} + K_{вз}, \text{ грн}, \qquad (3.3)$$

де:

$K_{пр}$ – вартість розробки політики безпеки інформації (9075 грн);

$K_{др}$ – вартість закупівлі датчиків розбиття скла ($443 \text{ грн} \cdot 6 \text{ шт} = 2658 \text{ грн}$);

$K_{дв}$ – вартість закупівлі датчиків відкриття вікон ($56 \text{ грн} \cdot 13 \text{ шт} = 728 \text{ грн}$);

$K_{ез}$ – вартість закупівлі електронних замків ($675 \text{ грн} \cdot 2 \text{ шт} = 1375 \text{ грн}$);

$K_{ск}$ – вартість закупівлі смарт-карток ($10 \text{ грн} \cdot 30 \text{ шт} = 300 \text{ грн}$);

$K_{с}$ – вартість закупівлі сейфів ($3420 \text{ грн} \cdot 3 \text{ шт} = 10260 \text{ грн}$);

$K_{д}$ – витрати на встановлення датчиків(1000 грн);

$K_{вз}$–витрати на встановлення електронних замків(1500 грн);

$$К = 9075 + 2658 + 728 + 1375 + 300 + 10260 + 1000 + 1500 =$$

$$= 26896, \text{ грн.}$$

3.2 Розрахунок річних експлуатаційних витрат на утримання і обслуговування об'єкта проектування

Експлуатаційні витрати – це поточні витрати на експлуатацію та обслуговування об'єкта проектування за визначений період (рік), що виражені у грошовій формі.

Для компанії актуальними можуть бути експлуатаційні витрати на:

- заробітну плату системному адміністратору (оскільки коло його обов’язків було розширено в нових розділах політики безпеки);

- електроенергію, що споживається новим обладнанням (новими датчиками сигналізації та електронними замками); журнали обліку знімних носіїв.

Додаткова заробітна плата ($С_з$), що сплачується робітнику за виконання нових обов'язків складає 8-10% від основної заробітної плати.

Системному адміністратору необхідно доплачувати за такі види робіт:

- контроль виконання користувачами розділів політик безпеки;
- налаштування групових політик (створення правил щодо використання знімних носіїв);
- резервне копіювання;

Тому, пропонується доплачувати системному адміністратору 10% від основної заробітної плати.

Основна заробітна плата системного адміністратора складає 15000 грн з перерахуваннями. Отже, за виконання нових обов'язків, адміністратор отримуватиме додатково 1500 грн. Таким чином, окрім основної заробітної платні, адміністратор отримуватиме на рік:

$$С_з = 1500 \cdot 12 = 18000 \text{ , грн.}$$

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року ($С_е$), визначається за формулою:

$$С_е = (Р_з + Р_д) \cdot F_р \cdot Ц_е \text{ , грн,} \qquad (3.4)$$

де:

$Р_з$ = 0,0002– встановлена потужність електронних замків (апаратури інформаційної безпеки), кВт;

$Р_з$ = 0,004 – встановлена потужність датчиків (апаратури інформаційної безпеки), кВт;

$F_р$ =8760 год - річний фонд робочого часу системи інформаційної безпеки

(за умови безперервного режиму роботи);

$Ц_е$ = 2,01 - тариф на електроенергію, грн/кВт·годин.

$$С_е = (0{,}0002 + 0{,}004) \cdot 8760 \cdot 2{,}01 = 73{,}95 \text{ , грн.}$$

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки (Стос) у відсотках від вартості капітальних витрат (13%). До таких витрат може бути віднесено обслуговування нових датчиків та електронного замка, отже, необхідно врахувати 1% від їх вартості.

$$С_{тос} = 0{,}01 \cdot (2658 + 728 + 1375) = 47{,}61 \text{ , грн.}$$

Крім того, організації періодично необхідно купляти журнали обліку знімних носіїв. Такий журнал необхідний трьом відділам приблизно раз на 2 роки:

$С_{жо}$ – вартість закупівлі журналів обліку знімних носіїв $= 35 \text{ грн} \cdot 3 \cdot 0{,}5 = 52{,}5 \text{ грн}$;

Отже, річні поточні (експлуатаційні) витрати на функціонування системи інформаційної безпеки складають:

$$С = С_з + С_е + С_{тос} + С_{жо} = 18000 + 74 + 48 + 54 = 18176 \text{ , грн.}$$

(3.5)

4.3 Визначення річного економічного ефекту від впровадження об'єкта проектування

Необхідно визначити величину відвернених збитків, що розраховується, виходячи з імовірності виникнення інциденту інформаційної безпеки й можливих економічних втрат від нього.

Далі буде визначено можливі збитки від таких загроз:

- проникнення у приміщення злочинців або конкурентів у неробочий час;
- проникнення у приміщення злочинців або конкурентів у робочий час через відсутність контролю за переміщенням відвідувачів у робочий час;
- фішинг та інші загрози, пов'язані з використанням електронної пошти та передачею внутрішніх документів через незахищене середовище. Можливість перехоплення інформації через відсутність регламенту щодо каналів передачі інформації між робітниками;
- можливість крадіжки паперових та електронних (знімних) носіїв інформації, несанкціонованого знищення, або несанкціонованого ознайомлення з відповідною інформацією;
- збій серверу та втрата інформації, що знаходиться на ньому, у результаті збою та відсутності належної системи резервного копіювання.

Внаслідок проникнення в офіс в неробочий час, може бути викрадено ноутбук або знімний носій. Внаслідок цього може бути зупинена діяльність системного адміністратора, порушено діяльність бухгалтерії або відділу маркетингу і продажів. Може бути втрачено проекти внаслідок порушення умов договору з клієнтом, через це ж компанія може зазнати репутаційних збитків.

Таким чином, упущена вигода від простою атакованого вузла (системного адміністратора) становить:

$$U_{\text{а}} = \Pi_{\text{п}} + \Pi_{\text{в}} + V \quad \text{, грн,} \tag{3.6}$$

де:

Пп – оплачувані втрати робочого часу та простої співробітників атакованого вузла, грн;

Пв – вартість відновлення працездатності вузла (переустановлення системи, зміна конфігурації та ін.), грн;

V – втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності системного адміністратора являють собою втрату його заробітної плати (оплата непродуктивної праці) за час простою:

$$П_п = \frac{\Sigma З_с}{F} \cdot t_п \quad , \text{ грн,} \qquad (3.7)$$

де:

F =176 год – місячний фонд робочого часу (40-годинний робочий тиждень);

Зс = 16000 грн - розмір заробітної платні працівника;

tп = 24 год - час простою вузла внаслідок атаки. Простій займає приблизно

3 дні по 8 годин робочого часу.

$$П_п = \frac{16000}{176} \cdot 24 = 2181, \text{ грн.}$$

Витрати на відновлення працездатності вузла включають кілька складових:

$$П_в = П_{ви} + П_{пв} + П_{зч}, \text{ грн,} \qquad (3.8)$$

де:

$П_{ви}$ – витрати на повторне уведення інформації, грн;

$П_{пв}$ – витрати на відновлення вузла або сегмента корпоративної мережі, грн;

$П_{зч}$ =13049 грн– вартість заміни устаткування або запасних частин, грн.

Витрати на повторне введення інформації $П_{ви}$ розраховуються виходячи з розміру заробітної плати співробітників атакованого вузла або сегмента корпоративної мережі $З_с$, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ви}$:

$$П_{ви} = \frac{\Sigma З_с}{F} \cdot t_{ви} = \frac{16000}{176} \cdot 8 = 727 \text{ , грн.} \quad (3.9)$$

Витрати на відновлення вузла або сегмента корпоративної мережі можна не враховувати, оскільки існують шаблони для всіх необхідних налаштувань і, якщо відновлення й необхідне, його можливо виконати за дуже короткий проміжок часу. $П_{пв} = 0$.

Втрати від зниження очікуваного обсягу продажів за час простою атакованого вузла також можна не враховувати, оскільки виконання проектів не залежить від системного адміністратора. V=0.

$$U_a = 2181 + 727 + 13049 = 15957, \text{ грн.}$$

Приблизно таку саму суму збитків зазнає компанія, якщо буде викрадено ноутбук бухгалтера або ноутбук працівника відділу маркетингу та продажів.

Крім того, у тому випадку, якщо буде вкрадено знімний носій з ескізами або готовими роботами, буде порушено договір з клієнтом – компанія втратить проект з ним і може зазнати репутаційного збитку.

Середня вартість розробки брендбуку -23400 грн.

Середня вартість розробки web дизайну сайту або розробки інтерфейсу додатка -15000 грн.

Середня тривалість одного проекту – 3 тижні, паралельне виконання 3 проектів. Компанія виконує приблизно 40 замовлень на рік.

В разі втрати замовлення компанія втрачає до 38400 грн. Якщо страждає репутація компанії то передбачається, що вона може втратити до 10% проектів, тобто, приблизно 4 проекти. Збиток тоді становитиме до 153600 грн. на наступний рік.

Передбачається, що одна така загроза може реалізуватися приблизно раз на рік.

Таким чином, загальний збиток від цієї загрози організації складе:

$$B = 15957 + 38400 + 153600 = 207957\text{грн.}$$

Внаслідок проникнення в офіс в робочий час, може бути втрачено проекти внаслідок порушення умов договору з клієнтом, через це ж компанія може зазнати репутаційних збитків. Отже, В=38400+153600=192000 грн. Те ж саме стосується загроз перехоплення інформації через відсутність регламенту щодо каналів передачі інформації між робітниками, крадіжки паперових та електронних (знімних) носіїв інформації. Збій серверу та втрата інформації, що знаходиться на ньому може спричинити затримку в роботі працівників та зменшення кількості проектів, що може виконуватись одночасно. Наприклад, в разі втрати ескізів, до проекту може бути залучено додаткових працівників для пришвидшення відновлення матеріалів. Тоді зазначені працівники не зможуть брати участь в нових проектах компанії. В = 38400 грн.

У таблиці 4.1 наведено актуальні загрози для підприємства та величини можливих збитків від реалізації цих загроз.

Таблиця 3.1 – Оцінка величини збитків

Загроза	Збиток B, грн	Ймовірність R	B · R, грн
Проникнення у приміщення злочинців або конкурентів у неробочий час	207957	0,17	35352
Проникнення у приміщення злочинців або конкурентів у робочий час через відсутність контролю за переміщенням відвідувачів у робочий час	192000	0,32	61440
Фішинг та інші загрози, пов'язані з використанням електронної пошти та передачею внутрішніх документів через незахищене середовище. Можливість перехоплення інформації через відсутність регламенту щодо каналів передачі інформації між робітниками	192000	0,14	26880
Можливість крадіжки паперових та електронних (знімних) носіїв інформації, несанкціонованого знищення, або несанкціонованого ознайомлення з відповідною інформацією	192000	0,38	72960

Збій серверу та втрата інформації, що знаходиться на ньому, у результаті збою та відсутності належної системи резервного копіювання	38400	0,12	4608
Всього:	201240 грн.		

3.4 Визначення та аналіз показників економічної ефективності запропонованого у кваліфікаційній роботі проектного рішення

Загальний ефект від впровадження системи інформаційної безпеки становить:

$$E = B \cdot R - C = 201240 - 18176 = 183064\text{грн.} \qquad (3.10)$$

де:

B – загальний збиток від атаки на вузол корпоративної мережі, грн;

R – очікувана імовірність атаки на вузол або сегмент корпоративної мережі, частки одиниці;

C – щорічні витрати на експлуатацію системи інформаційної безпеки, грн.

Оцінка економічної ефективності системи захисту інформації здійснюється на основі визначення та аналізу наступних показників:

- сукупна вартість володіння (TCO);
- коефіцієнт повернення інвестицій ROSI (Return on Investment for
- Security);
- термін окупності капітальних інвестицій T_o.

TCO у даному випадку не використовується, оскільки не порівнюються декілька варіантів проектів.

ROSI показує, скільки гривень додаткового прибутку приносить одна гривня капітальних інвестицій на впровадження системи інформаційної безпеки.

ROSI = E/K = 183064/26896+= 6,8 (3.11)

де:

Е – загальний ефект від впровадження системи інформаційної безпеки, грн;

К – капітальні інвестиції за варіантами, що забезпечили цей ефект, грн.

Для остаточної оцінки варіантів і вибору найбільш ефективного з них необхідно порівняти значення ROSI з бажаним значенням показника ефективності $Е_н$.

Організація здійснює фінансування капітальних інвестицій за рахунок реінвестування власних коштів, тому в якості $Е_н$ приймається бажана норма прибутковості альтернативних варіантів вкладення коштів К (на депозитний рахунок у банку).

Проект визнається економічно доцільним, якщо розрахунковий коефіцієнт ефективності перевищує річний рівень прибутковості альтернативного варіанта:

$$ROSI > \frac{(N_{деп} - N_{інф})}{100}$$

де:

$N_{деп}$ =14,5 – річна депозитна ставка або прибутковість альтернативного варіанту вкладення коштів, %;

$N_{інф}$ = 9,2 – річний рівень інфляції, %.

6,8> 0,05, отже проект є економічно доцільним.

Термін окупності капітальних інвестицій T_o показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки:

$$T_0 = \frac{K}{E} = \frac{1}{ROSI} = 0{,}147 \text{ року.}$$

3.5 Висновок про економічну доцільність проектного рішення

В цьому розділі було визначено розмір капітальних (26896 грн) та експлуатаційних (18176 грн) витрат на заходи і засоби інформаційної безпеки, величину відвернених втрат (201240 грн) та, на основі цього, розраховано коефіцієнт повернення інвестицій (6,8) та термін окупності капітальних інвестицій (0,147 року). На основі розрахованих показників можна зробити висновок, що запропоновані заходи та засоби є вигідними для компанії, оскільки термін окупності капітальних інвестицій є досить малим (менше двох місяців) та розрахунковий коефіцієнт ефективності перевищує річний рівень прибутковості альтернативного варіанта (річної депозитної ставки з врахуванням інфляції).

ВИСНОВКИ

Актуальність захисту веб серверів від атак, що базуються на недосконалості протоколів передачі даних стає з кожним роком все більшою. Все більше галузей бізнесу напряму залежать від якості та захищеності своїх веб серверів та розташованих на них додатків.

Найбільш проблематичною сферою захисту інформаційних ресурсів від несанкціонованого доступу є відсутність стандартизованих підходів щодо безпеки веб серверів.

Аналіз реалізації вразливості протоколу передачі даних HTTP на стороні клієнта, а також розробка методики підтвердження наявності у додатків що розташовані на сервері вразливостей цього типу, дасть змогу своєчасно виправляти всі «слабкі» місця у системі безпеки додатку. Чим раніше буде виявлена та виправлена вразливість – тим менші збитки зазнає власник веб серверу додатку при реалізації атак.

Розроблена методика тестування повинна бути інтегрована у комплексний процес тестування. В цьому випадку стане можливим більш точна та повна перевірка захищеності окремого додатка. Тестування захищеності в цілому повинно проводитися не тільки на етапі системного тестування додатку, але й вже на етапі тестування окремих його частин.

Запропонована методика тестування розглядає веб сервер с позиції користувача. Саме тому головними векторами для перевірки та аналізу у даній методиці стали HTTP-запити від клієнта до сервера. Даний підхід дозволяє найбільш ефективним способом оцінити можливі ризики компрометації даних та порушення цілісності, доступності та конфіденційності інформації, що циркулює.

Вихідними даними тестування стали вразливі параметри, що передаються у HTTP-запитах. Локалізація та визначення найбільш критичних параметрів дозволить на етапі проектування системи унеможливити

використання таких параметрів або переглянути логіку роботи додатку для забезпечення більш надійної передачі параметрів від клієнта до серверної частини додатку.

Економічна ефективність та доцільність впровадження даної методики доводить: методика може бути впроваджена, так як збитки від реалізації атаки набагато перевищують вартість розробки методики.

ПЕРЕЛІК ПОСИЛАНЬ

1 Ex.Mi, v31_v37, Иван aLLy Комиссаров, Марк Бруцкий-Стемпковский - “Атаки на веб и WordPress”, 2020 — 244 с.

2 Коллектив авторов журнала "Хакер" - “Взлом. Приемы, трюки и секреты хакеров”, 2019 — 192с.

3 Web Server – WikiTweet (Електронний ресурс) / Спосіб доступу: URL: http://www.wikitweet.net/ShowArticle.aspx/Web%20Server-12. – Загол. з екрана.

4 Модели работы веб серверов (Електронний ресурс) / Спосіб доступу: URL: http://algolist.manual.ru/web/servers.php. – Загол. з екрана.

5 OWASP: Cross – site Scripting (XSS) (Електронний ресурс) / Спосіб доступу: URL: https://www.owasp.org/index.php/Cross-site_Scripting_(XSS). – Загол. з екрана.

6 Виды информационных угроз и способы борьбы с ними (Електронний ресурс) / Спосіб доступу: URL: http://www.dokwork.ru/2012/01/blog-post.html. – Загол. з екрана.

7 OWASP: LDAP Injection (Електронний ресурс) / Спосіб доступу: URL: https://www.owasp.org/index.php/LDAP_injection. – Загол. з екрана.

8 OWASP: SQL Injection (Електронний ресурс) / Спосіб доступу: URL: https://www.owasp.org/index.php/SQL_injection. – Загол. з екрана.

9 OWASP: XPath Injection (Електронний ресурс) / Спосіб доступу: URL: https://www.owasp.org/index.php/XPATH_injection. – Загол. з екрана.

10 Гипертекстный протокол HTTP (Електронний ресурс) / Спосіб доступу: URL: http://www.intuit.ru/department/network/pami/7/. – Загол. з екрана.

11 HTTP протокол (Електронний ресурс) / Спосіб доступу: URL: http://www.inattack.ru/article/http-protokol/78.html#.T7X7Deg9XK0. – Загол. з екрана

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

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

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

15 OWASP AppSec Europe 2009. HTTP Parameter Pollution, May 2009. (Електронний ресурс) / Спосіб доступу: URL: https://owasp.org/www-pdf-archive/AppsecEU09_CarettoniDiPaola_v0.8.pdf. – Загол. з екрана.

16 Automated Discovery of Parameter Pollution Vulnerabilities in Web Applications. (Електронний ресурс) / Спосіб доступу: URL: http://www.iseclab. org/people/embyte/ papers/hpp.pdf .– Загол. з екрана.

17 HTTP Parameter Pollution (Електронний ресурс) / Спосіб доступу: URL: http://raz0r.name/articles/http-parameter-pollution/. – Загол. з екрана

18 How to Detect HTTP Paramter Pollution Attacks (Електронний ресурс) / Спосіб доступу: URL: http://www.acunetix.com/blog/whitepaper-http-parameter-pollution/. – Загол. з екрана

19 HTTP Parameter Contamination (Електронний ресурс) / Спосіб доступу: URL: http://www.securitylab.ru/analytics/406673.php – Загол. з екрана

20 HTTP Parameter Pollution (Електронний ресурс) / Спосіб доступу: URL: http://tacticalwebappsec.blogspot.com/2009/05/http-parameter-pollution.html– Загол. з екрана

21 Зарплати українських розробників — зима 2022 (Електронний ресурс) / Спосіб доступу: URL: https://dou.ua/lenta/articles/salary-report-devs-winter-2022/

22 Закон України «Про державну підтримку малого підприємництва» // 2008. – С.1-2.

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

24 НД ТЗІ 1.4-001-2000. Типове положення про службу захисту інформації в автоматизованій системі

25 Методичні рекомендації до підготовки та захисту дипломної роботи (проекту) для студентів галузі знань 1701 «Інформаційна безпека» та спеціальності 125 «Кібербезпека» / Т.В. Бабенко, М.В. Корнєєв, О.В. Кручінін, Д.С. Тимофєєв ; Нац. гірн. ун-т. – Д. : НГУ, 2016. – 44 с.

26 Методичні вказівки до виконання економічної частини дипломного проекту (для студентів напряму підготовки 1701 Інформаційна безпека)/ Упорядн.: О.Г. Вагонова, І.В. Шереметьєва, Ю.О. Волотковська, Н.М. Романюк. – Дніпропетровськ: ДВНЗ "Національний гірничий університет", 2013. – 17 с

ДОДАТОК А. ВІДОМІСТЬ МАТЕРІАЛІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ

№	Формат	Найменування	Кількість аркушів	Примітки
Документація				
1	А4	Реферат	2	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	2	
4	А4	Вступ	1	
5	А4	Розділ 1. Стан питання. Постановка задачі	40	
6	А4	Розділ 2. Спеціальна частина	42	
7	А4	Розділ 3. Економічна частина	16	
8	А4	Висновок	1	
9	А4	Перелік посилань	1	
10	А4	Додаток А. Відомість матеріалів дипломної роботи	1	
1	А4	Додаток Б.	1	

1		Перелік документів на оптичному носії		
12	А4	Додаток В. Відгук керівника економічного розділу	1	
13	А4	Додаток Г. Відгук керівника дипломної роботи	1	

ДОДАТОК Б. ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

– Бочін І.І. 125-18-3.docx
– Бочін І.І. 125-18-3.pptx

ДОДАТОК В. ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

Керівник економічного розділу

ДОДАТОК Г. ВІДГУК КЕРІВНИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ

ВІДГУК на кваліфікаційну роботу
студента групи 125-18-3 Бочіна Ігоря Ігоровича на тему «Метод протидії атаці типу "розбиття відповіді HTTP" на WEB сервер»

Пояснювальна записка складається зі вступу, трьох розділів і висновків, викладених на 114 сторінках.

Метою кваліфікаційної роботи є розробка методу протидії атаці "розбиття відповіді HTTP» на WEB сервер та проведення розрахунків, щодо ефективність його використання.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра спеціальності 125 «Кібербезпека». Для досягнення поставленої мети в кваліфікаційній роботі вирішуються наступні задачі: визначеня джерел уразливостей веб-серверів, аналіз видів та методів атак на веб сервери, визначення інструментарію, що використовується при атаках та проведення тестування захищеності веб-серверу.

Практичне значення роботи полягає в тому, що запропонована методика може бути використана як набір рекомендацій для виявлення вразливостей типу HTTP Parameter Pollution.

До недоліків роботи слід віднести недостатню конкретизацію об'єкта розробки та незначні невідповідності вимогам оформлення.

За час дипломування Бочін Ігор Ігорович проявив себе фахівцем, здатним самостійно вирішувати поставлені задачі та заслуговує присвоєння кваліфікації бакалавра за спеціальністю 125 Кібербезпека, освітньо-професійна програма «Кібербезпека».

Кваліфікаційна робота заслуговує оцінки « 78 / добре ».

Рівень запозичень у кваліфікаційній роботі не перевищує вимог "Положення про систему виявлення та запобігання плагіату".

Керівник кваліфікаційної роботи к.т.н., доц. Герасіна О.В.

Керівник спеціальної частини ас. Мілінчук Ю.А.