НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«СИСТЕМА ЗАХИСТУ ТА РЕЗЕРВНОГО КОПІЮВАННЯ ДАНИХ ВЕБСЕРВЕРУ»

Виконав: здобувач 4 курсу
Рівень бакалавр
Галузь знань 12 «Інформаційні технології»,
спеціальність 125 «Кібербезпека»
Середа Максим Вікторович

Керівник: к.т.н., доцент
Бойко Віктор Дмитрович

Рецензент: проф.д.т.н, доцент
Соколов Артем Вікторович

Одеса – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
Факультет **кібербезпеки та інформаційних технологій**
Кафедра **кібербезпеки**
Галузь знань: **12 Інформаційні технології**
Спеціальність: **125 Кібербезпека**
Назва освітньої програми: **Кібербезпека**

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки
професор С.А. Горбаченко
_____ «____ »_____ 20__ року

З А В Д А Н Н Я
на кваліфікаційну (бакалаврську) роботу
здобувачу Середі Максиму Вікторовичу

1. Тема роботи: «Система захисту та резервного копіювання даних вебсерверу» Керівник роботи: к.т.н, доцент Бойко Віктор Дмитровиязатверджені наказом НУ ОЮА від «18» березня 2025 року № 548-6
2. Строк подання здобувачем роботи «19» травня 2025 року
3. Дата видачі завдання «16» вересня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області та пошук науково-технічної інформації за темою роботи	Вересень-жовтень 2024	
2	Узгодження теми з науковим керівником, формулювання мети завдань дослідження	Листопад 2024	
3	Робота над першим розділом	Листопад-грудень 2024	
4	Робота над другим розділом	Січень-лютий 2025	
5	Робота над третім розділом	Лютий-березень 2025	
6	Уточнення подальшого обсягу роботи та його коригування	Березень-травень 2025	
7	Оформлення звітної частини	Травень 2025	
8	Захист роботи	10.06.2025	

Здобувач _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Середа М.В. Система захисту та резервного копіювання даних вебсерверу - Кваліфікаційна робота бакалавра за спеціальністю 125 «Кібербезпека », освітньою програмою «Кібербезпека та захист даних». Кваліфікаційна робота викладена на 40 сторінках основного тексту і 45 сторінках загального тексту, містить 3 розділи, 28рисунків, 13 використаних джерел.

Останніми роками спостерігається значне зростання кількості кіберінцидентів, зокрема - DDoS-атак, SQL-ін’єкцій, атак типу "людина посередині" (MITM) та інших загроз, спрямованих на компрометацію вебресурсів. У зв’язку з цим розробка надійної системи захисту вебсерверів і впровадження ефективних механізмів резервного копіювання є критично важливими завданнями в галузі кібербезпеки.

Метою даної кваліфікаційної роботи є дослідження та впровадження практичних рішень для забезпечення безпеки та збереження даних вебсерверу, зокрема шляхом налаштування фаєрволу, систем запобігання вторгненням та організації автоматичного резервного копіювання. У роботі проаналізовано сучасні загрози вебінфраструктурі, розглянуто популярні платформи та програмні засоби, що використовуються для забезпечення стабільності та захисту серверних систем.

Наукова новизна та практична значущість дослідження полягає в комплексному підході до розробки безпечної серверної архітектури, що поєднує теоретичні засади інформаційної безпеки з прикладними інструментами на базі відкритого програмного забезпечення.

КІБЕРБЕЗПЕКА, ВЕБСЕРВЕР, ЗАХИСТ ДАНИХ, РЕЗЕРВНЕ КОПІЮВАННЯ, ФАЄРВОЛ, FAIL2BAN, RSYNC.

ABSTRACT

Sereda M.V. Web server data protection and backup system - Bachelor’s thesis in specialty 125 “Cybersecurity”, educational program “Cybersecurity and Data Protection”. The qualification work is set out on 40 pages of the main text and 46 pages of the general text, contains 3 sections, 28 figures, 13 references.

In recent years, there has been a significant increase in the number of cyber incidents, including DDoS attacks, SQL injections, man-in-the-middle (MITM) attacks and other threats aimed at compromising web resources. In this regard, the development of a reliable web server security system and the implementation of effective backup mechanisms are critical tasks in the field of cybersecurity.

The purpose of this qualification work is to research and implement practical solutions to ensure the security and safety of web server data, in particular by setting up firewalls, intrusion prevention systems, and organizing automatic backups. The paper analyzes modern threats to web infrastructure, examines popular platforms and software tools used to ensure the stability and protection of server systems.

The scientific novelty and practical significance of the study lies in an integrated approach to the development of a secure server architecture that combines the theoretical foundations of information security with applied tools based on open source software.

CYBERSECURITY, WEB SERVER, DATA PROTECTION, BACKUP, FIREWALL, FAIL2BAN, RSYNC.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

DDoS - Distributed Denial of Service

UFW - Uncomplicated Firewall

MySQL - My Structured Query Language

Rsync - Remote Synchronize

HTTP - Hypertext Transfer Protocol

HTTPS - Hypertext Transfer Protocol Secure

IP - Internet Protocol

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

VPN - Virtual Private Network

LAN - Local Area Network

WAN - Wide Area Network

SQL - Structured Query Language

ОС - Операційна Система

ВСТУП

У сучасному цифровому світі, де інформаційні технології проникають у всі сфери людської діяльності - від бізнесу до державного управління - питання забезпечення безпеки та збереження даних набуває надзвичайної актуальності. Особливої уваги заслуговують вебсервіси, які є основою функціонування безлічі онлайн-платформ, систем електронної комерції, банківських установ, освітніх порталів тощо. Будь-який збій або атака на вебсервер може призвести не лише до фінансових втрат, а й до втрати репутації та витоку конфіденційної інформації.

Мета дослідження: Розробити комплексну систему захисту та резервного копіювання даних вебсерверу, спрямовану на запобігання кіберзагрозам, мінімізацію ризиків втрати даних та забезпечення стабільної роботи вебсерверу.

Основні завдання: Провести аналіз вразливостей та загроз для вебсерверів.

Дослідити технології резервного копіювання даних.

Розробити практичні механізми захисту вебсерверу, включаючи налаштування фаєрволу, використання інструментів типу Fail2ban для запобігання атакам.

Впровадити систему автоматичного резервного копіювання з використанням Rsync та інших інструментів.

Протестувати розроблену систему на практичних прикладах.

Результати:

У роботі запропоновано систему захисту вебсерверу, яка включає:

Налаштування фаєрволу (ufw) для фільтрації мережевого трафіку.

Використання Fail2ban для блокування підозрілих IP-адрес.

Реалізацію автоматичного резервного копіювання даних за допомогою Rsync та скриптів для створення дампів баз даних.

Розроблена система дозволяє ефективно захищати вебсервер від кіберзагроз, забезпечує надійне зберігання даних та швидке відновлення після можливих інцидентів.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз статистки

У сучасному цифровому світі, коли практично всі аспекти нашого життя та бізнесу пов’язані з цифровими технологіями, втрата даних може мати катастрофічні наслідки. Навіть короткочасна втрата доступу до важливих файлів може викликати значні перебої в роботі бізнесу, а для окремих користувачів – призвести до втрати незамінних спогадів чи особистих документів.

Зараз Amazon щодня відстежує на сотні мільйонів більше потенційних кіберзагроз. Всього за шість-сім місяців кількість загроз зросла зі 100 мільйонів до 750 мільйонів на день. (The Wall Street Journal)

Більше атак з вимогами. Великі атаки з використанням програм-вимагачів стали набагато частішими. У 2011 році відбувалося п’ять великих атак на рік. У 2024 році щодня відбувається від 20 до 25 великих атак з використанням програм-вимагачів. (The New York Times)

Втручання у вибори. Кіберзагрози зараз становлять більший ризик для національної безпеки. Іноземний вплив на вибори є більшим, ніж будь-коли раніше. Кібератаки мають на меті зірвати голосування і вибори. (Wired)

Фінансовий вплив. Фінансові втрати від кіберінцидентів зросли в чотири рази з 2017 року. Хоча більшість втрат від кібератак невеликі (близько $0,5 млн), великі інциденти можуть призвести до величезних збитків. Раз на 10 років компанія може втратити до 2,5 мільярдів доларів від серйозної кібератаки. (Міжнародний валютний фонд)

У 2023 і 2024 роках було прийнято понад 170 законів про захист даних, спрямованих на боротьбу з витоками даних. Компанії ставляться до безпеки даних серйозніше, ніж будь-коли. (MIT Sloan) (див. рисунок 1.1)

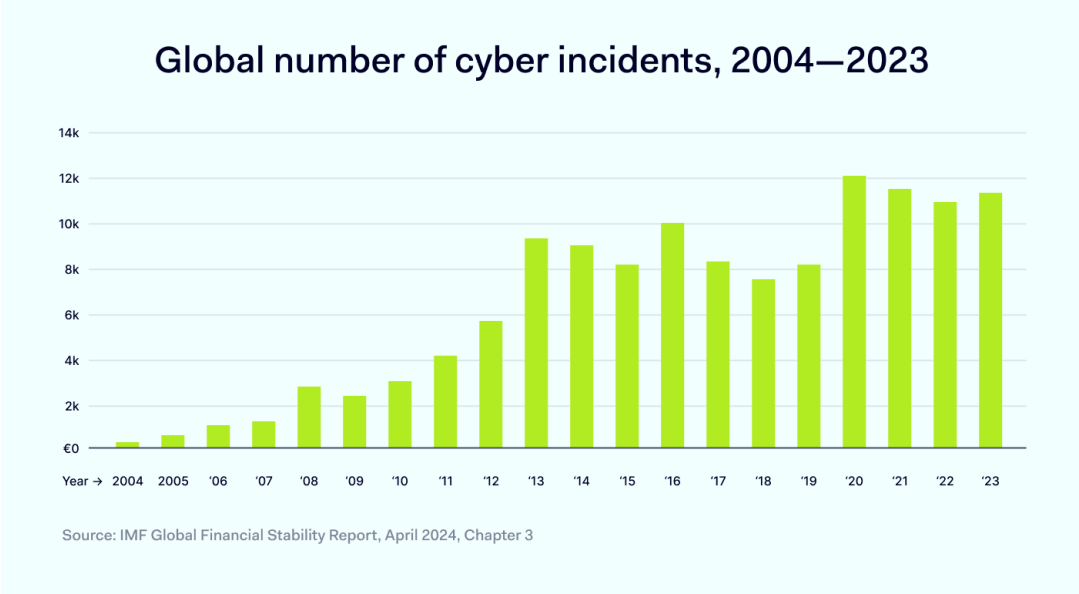

Рисунок 1.1 – Глобальна кількість кіберінцидентів, 2004-2023

Найуразливіші галузі у 2024 році

Деякі галузі стикаються з більшою кількістю кібератак через цінні дані, якими вони володіють. У 2024 році охорона здоров'я та телекомунікації стали основними цілями кіберзагроз.

Організації охорони здоров'я є першочерговими цілями через чутливі дані, такі як медичні картки пацієнтів. За словами Ребекки Райт, професора кібербезпеки в Барнардському коледжі, лікарні особливо вразливі до програм-вимагачів, оскільки їх важко захистити і вони покладаються на різні системи та сторонніх постачальників.

Найбільше занепокоєння викликає те, що зловмисники часто не намагаються викрасти дані. Замість цього вони намагаються порушити роботу сервісів, щоб змусити лікарні платити викуп.

Ось чому охорона здоров'я була виділена як ціль у 2024 році:

Вплив на послуги. Кібератака на Change Healthcare у лютому 2024 року порушила мільйони рецептів. Це призвело до затримок у постачанні ліків та медичної допомоги. Кібератаки призвели до скасування операцій та затримки лікування. Багато медичних практик зіткнулися з проблемою закриття через втрату доходів, що поставило під загрозу надання допомоги пацієнтам (The Lancet).

Лікарні стикалися з атаками програм-вимагачів, які виводили з ладу системи. Наприклад, лікарні Вознесіння зазнали затримок у наданні допомоги пацієнтам (The New York Times).

59% опитаних IT-фахівців у сфері охорони здоров'я заявили, що за останні 2 роки зазнали щонайменше одну атаку з використанням програм-вимагачів, а в середньому за останні 2 роки - 4 атаки з використанням програм-вимагачів (Звіт Ponemon Institute).

92% фахівців з IT-безпеки в організаціях охорони здоров'я США повідомили про щонайменше одну кібератаку минулого року, порівняно з 88% у 2023 році (The HIPAA Journal).

З січня по вересень 2024 року середньосвітовий показник кількості атак на одну медичну організацію становив 2 018 на тиждень. Це на 32% більше, ніж у 2023 році (IndustrialCyber).

У секторі охорони здоров'я Північної Америки кількість щотижневих атак зросла на 20% у порівнянні з минулим роком. Він залишається головною ціллю через цінні дані пацієнтів та потужну цифрову інфраструктуру (IndustrialCyber)

Витоки конфіденційних даних, які використовуються для шпигунства. У серпні 2024 року іранські кіберактивісти отримали доступ до мереж охорони здоров'я США з метою шпигунства і допомогли зловмисникам заробити на програмному забезпеченні з вимогою викупу (Американська асоціація лікарень)

1.2 Аналіз вразливостей вебсерверів

Безпека є центральною проблемою для кожного вебдодатку, оскільки всілякі користувачі отримують доступ до вебсерверів і намагаються завдати шкоди вебсервісам. Для захисту вебсерверів та вебдодатків від атак та вразливостей застосовуються різні методи захисту. Для боротьби з вразливостями безпеки запропоновано функціональний шлях програмування для комунікації з програмними атаками та вразливостями для розпізнавання та комунікації з програмними атаками та вразливостями. Сканери веббезпеки використовуються

для виявлення вразливостей у вебсервісах. «Різні сканери мають різну ефективність при виявленні вразливостей у різних вебдодатках".

Вразливості вебдодатків

Уразливості в системі можуть призвести до компрометації безпеки всього вебдодатку. Зловмисники зазвичай намагаються знайти слабкі місця вебдодатків через знання вразливостей додатків, щоб отримати з цього вигоду. Використання таких вразливостей сприяє вчиненню кіберзлочинів.

За даними Gartner Security, прикладний рівень наразі містить 90% всіх вразливостей». Open Web Application Security Project (OWASP) надає найновішу інформацію та рекомендації щодо найбільш важливих вразливостей вебдодатків, яких повинна дотримуватися будь-яка організація, щоб обмежити недоліки вебдодатків. Під час розробки, придбання та створення нових API для будь-якого вебдодатку необхідно враховувати вразливості OWASP.

Основних ризики безпеки OWASP такі:

1. Ін’єкції

Згідно з версією OWASP, ін’єкція відбувається, коли зловмисник хоче вставити деякі коди для доступу або отримання даних. Існують різні типи ін’єкцій, найпоширенішими з яких є SQL, LDAP. Їх легко виявити при дослідженні коду. Сканери можуть допомогти зловмисникам знайти дефекти ін’єкцій. Результат згаданої ін’єкції може бути катастрофічним для організації. Іноді ін’єкція може призвести до повного поглинання комп’ютера. Бізнес-враження залежить від неадекватності мети та даних.

2. Порушена автентифікація

Вразливості зламаної автентифікації та управління сеансами є загальними і мають критичний вплив після використання. Вони дозволяють зловмиснику викрасти привілейовані облікові записи користувачів, а також приховати свої дії. Ця уразливість виникає через те, що розробники використовують кастомні схеми автентифікації, які схеми містять вразливості в управлінні паролями, виходом з системи, оновленням облікового запису та таймаутами. Виявлення порушеної

автентифікації може бути ускладнене тим, що кожна клієнтська реалізація автентифікації та управління сеансами є унікальною.

3. Вплив на конфіденційні дані

За останні кілька років це була найпопулярніша атака. Поширеним недоліком є те, що конфіденційні дані легко не зашифрувати. Коли застосовується криптографія, популярними є слабкі системи генерації та адміністрування ключів, а також використання слабких алгоритмів, протоколів та шифрів, особливо для слабких методів зберігання хешування паролів. Вразливості на стороні сервера в основному легко виявити для даних, що передаються, але важко для даних, що знаходяться в стані спокою.

4. Зовнішні об’єкти XML (XXE)

«Декілька старих процесорів XML допускають термін зовнішньої сутності, URI, який оцінюється під час обробки XML». Інструменти SAST можуть виявити цю проблему, досліджуючи залежності та конфігурацію. Інструменти DAST вимагають додаткових стандартних кроків для виявлення та використання цієї проблеми. Ручні тестувальники повинні бути підготовлені до перевірки XXE. Ці вразливості можуть бути застосовані для вилучення даних, виконання віддаленого запиту з сервера, сканування внутрішніх систем, проведення атаки на відмову в обслуговуванні.

5. Порушення контролю доступу

У вебдодатках існують різні рівні контролю доступу, які здебільшого кожен розробник намагається повністю заповнити. З іншого боку, існують тестери безпеки вебдодатків, які можуть виконувати автоматичне виявлення вразливостей додатків. Крім того, існує ручне тестування, яке може виявити слабкі місця в контролі доступу. Другий спосіб тестування контролю доступу у вебдодатку для виявлення слабких місць - це ручне тестування і різні методи, такі як (GET vs PUT), прямі посилання на об’єкти.

6. Неправильна конфігурація безпеки

Неправильна конфігурація може бути на будь-якому рівні, що може скомпрометувати вебдодатки. Це може бути мережеві служби, конфігурація

вебсервера, конфігурація сервера додатків, база даних та інтеграція з базою даних, фреймворки, що використовуються для розробки, кастомний код без урахування стандартів, а також у попередньо встановлених віртуальних машинах, контейнерах або різних типах сховищ. Автоматичні сканери корисні для виявлення неправильних конфігурацій, використання облікових записів за замовчуванням або конфігурацій за замовчуванням, додаткової допомоги, застарілих переваг.

7. Міжсайтовий скриптинг (XSS)

Атаки на міжсайтовий скриптинг (XSS) має дуже загальний характер і може створювати помірний вплив, наприклад, виконувати скрипти в браузері жертви, щоб перехоплювати сеанси користувачів, руйнувати вебсайти, впроваджувати ворожий вміст, і перенаправляти користувачів на шкідливі сайти. XSS уразливості – це ін’єкції в HTML-дані, які генерує вебдодаток , що дозволяє зловмиснику вставити скрипт на вебсторінки, за якими спостерігають інші користувачі.

8. Небезпечна десеріалізація

Можливо, що зловмисники постачають децентралізовано ворожі або підроблені об’єкти, що призводить до вразливості додатків та API. Наслідок двох основних типів атак:

Атаки, пов’язані з об’єктами та структурою даних: У цьому типі атаки зловмисник модифікує логіку додатку або виконує довільний віддалений код, якщо додаток містить класи, які можуть змінити поведінку під час або після десеріалізації.

Типові атаки на фальсифікацію даних: Прикладом можуть бути атаки, пов’язані з контролем доступу, де використовуються ті ж самі структури даних , але вміст може бути змінений.

Серіалізація може використовуватися в додатках як:

– Віддалений та міжпроцесний зв’язок (RPC/IPC)

– Дротові протоколи, вебсервіси, брокери повідомлень

– Кешування/персистентність

– Бази даних, кеш-сервери, файлові системи

– HTTP-кукі, параметри HTML-форм, токени автентифікації API.

9. Використання компонентів з відомими вразливостями

Для швидшої розробки додатків розробники використовують різні вбудовані та попередньо розроблені компоненти. Якщо компонент застарілий або має вразливість, це призводить до витоку даних .

10.Недостатній моніторинг логів

Недостатній моніторинг логів дозволяє зловмисникам атакувати системи та скомпрометувати або знищити всі дані . Дослідження порушень показують, що час виявлення порушення становить понад 200 днів. Моніторинг логів зазвичай виявляється зовнішніми сторонами, а не внутрішніми .

1.3 Основні загрози для вебсерверів

Як уже згадувалося, DDoS-атаки та SQL-ін’єкції належать до ключових загроз для вебсерверів. Далі буде проведено більш детальний їх аналіз [5] .

Розподілена атака на відмову в обслуговуванні (DDoS-атака) – це зловмисна спроба порушити нормальну роботу сервера, служби або мережі, на яку спрямована атака, шляхом переповнення цільового сервера або навколишньої інфраструктури величезним потоком інтернет-трафіку.

Ефективність DDoS-атак досягається завдяки використанню декількох скомпрометованих комп’ютерних систем як джерел атакуючого трафіку. Експлуатовані машини можуть включати комп’ютери та інші мережеві ресурси, такі як пристрої Інтернету речей.

З високого рівня, DDoS-атака схожа на несподіваний затор, який забиває шосе, не даючи регулярному трафіку прибути до місця призначення.

DDoS-атаки здійснюються за допомогою мереж підключених до Інтернету комп’ютерів.

Ці мережі складаються з комп’ютерів та інших пристроїв (наприклад, пристроїв Інтернету речей), які були заражені шкідливим програмним забезпеченням, що дозволяє зловмиснику дистанційно керувати ними. Ці окремі пристрої називаються ботами (або зомбі), а група ботів – ботнетом.

Після створення ботнету зловмисник може керувати атакою, надсилаючи віддалені інструкції кожному боту.

Коли ботнет атакує сервер або мережу жертви, кожен бот надсилає запити на IP-адресу жертви, що може призвести до перевантаження сервера або мережі, що спричинить відмову в обслуговуванні звичайного трафіку.

Оскільки кожен бот є легітимним інтернет-пристроєм, відокремити атакуючий трафік від звичайного може бути складно.

Слід також згадати різні типи DDoS-атак

Атаки націлені на різні частини мережі, і вони класифікуються відповідно до рівнів мережевого з'єднання, на які вони спрямовані. З'єднання в Інтернеті складається з семи різних «рівнів», як визначено в моделі взаємодії відкритих систем (OSI), створеній Міжнародною організацією зі стандартизації. Модель дозволяє різним комп'ютерним системам «розмовляти» одна з одною.

Атаки на основі об'єму або об'ємні атаки

Цей тип атак спрямований на контроль всієї доступної пропускної здатності між жертвою і глобальною мережею Інтернет. Ампліфікація системи доменних імен (DNS) є прикладом атаки на основі обсягу. У цьому сценарії зловмисник підробляє адресу жертви, а потім надсилає запит на пошук DNS-імені на відкритий DNS-сервер із підробленою адресою.

Коли DNS-сервер надсилає відповідь із записом DNS, вона надсилається замість нього на адресу цілі, в результаті чого ціль отримує посилення початкового невеликого запиту зловмисника.

Протокольні атаки

Протокольні атаки використовують всю доступну потужність вебсерверів або інших ресурсів, таких як брандмауери. Вони використовують слабкі місця на 3 і 4 рівнях стеку протоколів OSI, щоб зробити ціль недоступною.

SYN-флуд є прикладом протокольної атаки, в якій зловмисник надсилає на ціль переважну кількість запитів рукостискання протоколу управління передачею (TCP) з підробленими вихідними IP-адресами. Сервери, на які спрямована атака,

намагаються відповісти на кожен запит на з’єднання, але остаточне рукостискання ніколи не відбувається, перевантажуючи ціль в процесі роботи.

Атаки на рівні додатків

Ці атаки також мають на меті виснажити або перевантажити ресурси цілі, але їх важко назвати зловмисними. Атаки на рівні додатків, які часто називають DDoS-атаками 7-го рівня (посилаючись на 7-й рівень моделі OSI), націлені на рівень, на якому генеруються вебсторінки у відповідь на запити протоколу передачі гіпертексту (HTTP).

Сервер виконує запити до бази даних для створення вебсторінки. У цій формі атаки зловмисник змушує сервер жертви обробляти більше, ніж зазвичай. HTTP-флуд є різновидом атаки на рівні додатків і схожий на постійне оновлення веббраузера на різних комп’ютерах одночасно. Таким чином, надмірна кількість HTTP-запитів перевантажує сервер, що призводить до DDoS-атаки.

Атаки на рівні додатків

Ці атаки також мають на меті виснажити або перевантажити ресурси жертви, але їх важко назвати зловмисними. Атаки на рівні додатків, які часто називають DDoS-атаками 7-го рівня (посилаючись на 7-й рівень моделі OSI), націлені на рівень, на якому генеруються вебсторінки у відповідь на запити протоколу передачі гіпертексту (HTTP).

Сервер виконує запити до бази даних для створення вебсторінки. У цій формі атаки зловмисник змушує сервер жертви обробляти більше, ніж зазвичай. HTTP-флуд є різновидом атаки на рівні додатків і схожий на постійне оновлення веббраузера на різних комп’ютерах одночасно. Таким чином, надмірна кількість HTTP-запитів перевантажує сервер, що призводить до DDoS-атаки.

SQL-ін’єкція (SQLi) – це тип атаки, за допомогою якого кіберзлочинці намагаються використати вразливості в коді програми, вставляючи SQL-запит у звичайні поля введення або форми, такі як ім’я користувача або пароль. Потім SQL-запит передається до баз даних SQL, що лежать в основі програми.

Атаки з використанням SQL-ін’єкцій є успішними, якщо вебформа для введення даних дозволяє користувацьким операторам SQL запитувати базу даних

напряму. Ці атаки також поширилися з використанням спільних кодових баз, таких як плагіни WordPress, які містять вразливість в базовому шаблоні коду. Ця вразливість переноситься на весь додаток і може вплинути на сотні тисяч вебсерверів, які використовують цей спільний код.

Шкода може бути величезною. Зловмисник, який добре знає SQL, вводить запити у вебдодатку без жодних параметрів перевірки вхідних даних, а потім легко отримує доступ до клієнтських файлів або конфіденційної фінансової інформації компанії.

Подібно до DDoS-атак , SQL-ін’єкції також мають різні типи.

Внутрішньосмуговий SQLi є поширеним типом атак і відомий своєю простотою та ефективністю. Цей метод має дві варіації: на основі помилок та на основі об’єднань.

1. SQLi на основі помилок

Зловмисники вставляють SQL-запити, сподіваючись, що база даних поверне повідомлення про помилки, які можуть дати зловмисникам інформацію про базу даних та її структуру.

2. SQLi на основі UNION

У цьому сценарії зловмисники використовують оператор UNION SQL для того, щоб база даних повернула єдину відповідь по протоколу передачі гіпертексту (HTTP). Потім зловмисники можуть оцінити відповідь на предмет підказок про вміст бази даних.

Вивідний (сліпий) SQLi

В інференційних або сліпих SQLi-атаках шахраї запитують базу даних і спостерігають за поведінкою сервера, щоб зібрати інформацію про структуру бази даних. Ці типи атак є повільнішими, але вони можуть бути настільки ж шкідливими, як і інші типи SQLi. Існує два типи: булеві та засновані на часі.

1. Булевий

Зловмисник робить запит до бази даних, і, вивчаючи, чи була HTTP-відповідь змінена або залишилася незмінною, він може визначити, чи був результат істинним або хибним.

2. Залежно від часу

Як випливає з назви, зловмисник вивчає час відповіді в секундах на результат запиту. Як і у випадку з логічним типом, зловмисник уважно дивиться на HTTP-відповідь, щоб визначити, чи був запит істинним чи хибним.

Позасмуговий SQLi

Це особливий випадок SQLi, який може працювати лише за умови ввімкнення певних функцій сервера баз даних, що використовується додатком. Позасмуговий SQLi вважається альтернативою внутрішньосмуговому та інференційному методам. Позасмуговий SQLi не покладається на зловмисника, який запитує базу даних для вивчення повідомлень про помилки або HTTP-відповідей. Замість цього він очікує, що сервер згенерує систему доменних імен (DNS) або HTTP-запити, щоб зловмисник міг отримати такі дані, як імена користувачів і паролі.

Атака «людина посередині» (MITM) - це загальний термін для позначення ситуації, коли зловмисник позиціонує себе в розмові між користувачем і додатком – або для підслуховування, або для того, щоб видати себе за одну зі сторін, створюючи враження, що відбувається звичайний обмін інформацією.

Метою атаки є викрадення особистої інформації, такої як облікові дані для входу в систему, реквізити рахунків і номери кредитних карток. Цілями зазвичай стають користувачі фінансових додатків, SaaS-бізнесу, сайтів електронної комерції та інших вебсерверів, де потрібен вхід в систему.

Інформація, отримана під час атаки, може бути використана для багатьох цілей, включаючи крадіжку особистих даних, несанкціоновані перекази коштів або незаконну зміну пароля.

Крім того, вона може бути використана для того, щоб закріпитися всередині захищеного периметра на етапі проникнення під час атаки з використанням сучасних постійних загроз (APT).

У широкому сенсі, MITM-атака - це еквівалент того, що листоноша відкриває вашу банківську виписку, записує дані вашого рахунку, а потім запечатує конверт і доставляє його до ваших дверей (див. рисунок 1.2).

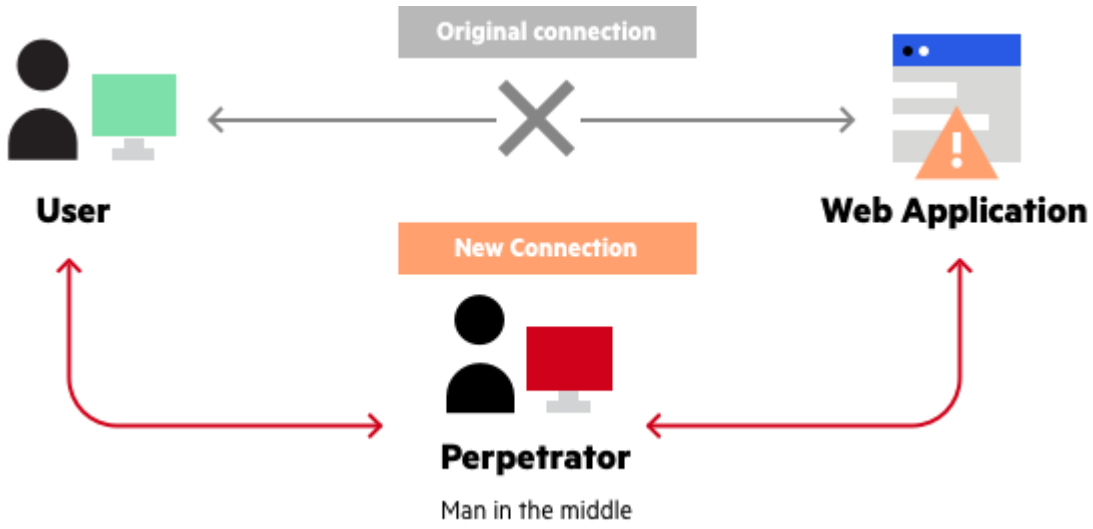

Рисунок 1.2 – Приклад MITM-атаки

Успішне виконання MITM складається з двох окремих етапів: перехоплення та розшифрування.

Перехоплення

На першому етапі перехоплюється трафік користувача через мережу зловмисника до того, як він досягне місця призначення.

Найпоширеніший (і найпростіший) спосіб зробити це - пасивна атака, в якій зловмисник робить безкоштовну, шкідливу точку доступу до WiFi загальнодоступною. Зазвичай вони мають назву, яка відповідає їхньому місцезнаходженню, і не захищені паролем. Як тільки жертва підключається до такої точки доступу, зловмисник отримує повний контроль над будь-яким обміном даними в Інтернеті.

Зловмисники, які бажають застосувати більш активний підхід до перехоплення, можуть запустити одну з наступних атак:

Підміна IP-адреси – зловмисник маскується під програму, змінюючи заголовки пакетів в IP-адресі. В результаті користувачі, які намагаються отримати доступ до URL-адреси, пов’язаної з додатком, перенаправляються на вебсайт зловмисника.

Підміна ARP - це процес зв’язування MAC-адреси зловмисника з IP-адресою законного користувача в локальній мережі за допомогою підроблених ARP-

повідомлень. В результаті, дані, надіслані користувачем на IP-адресу хоста, замість цього передаються зловмиснику.

Підміна DNS, також відома як отруєння кешу DNS, передбачає проникнення на DNS-сервер і зміну адресного запису вебсайту. В результаті користувачі, які намагаються отримати доступ до сайту, перенаправляються за зміненим записом DNS на сайт зловмисника.

Розшифрування

Після перехоплення будь-який двосторонній SSL-трафік потрібно розшифрувати без попередження користувача або програми. Для цього існує кілька методів:

Підміна HTTPS надсилає фальшивий сертифікат до браузера жертви після першого запиту на з'єднання з захищеним сайтом. Він містить цифровий відбиток пальця, пов'язаний зі скомпрометованим додатком, який браузер перевіряє за наявним списком довірених сайтів. Після цього зловмисник може отримати доступ до будь-яких даних, введених жертвою до того, як вони будуть передані додатку.

SSL BEAST (браузерний експлойт проти SSL/TLS) націлений на уразливість TLS версії 1.0 в SSL. Тут комп'ютер жертви заражається шкідливим JavaScript, який перехоплює зашифровані файли cookie, що надсилаються вебдодатком. Потім компрометується ланцюжок блоків шифрування (CBC) додатку, щоб розшифрувати його файли cookie та токени автентифікації.

Перехоплення SSL відбувається, коли зловмисник передає підроблені ключі автентифікації як користувачеві, так і додатку під час рукостискання TCP. Це створює видимість безпечного з'єднання, хоча насправді людина посередині контролює весь сеанс.

Зняття SSL знижує рівень HTTPS-з'єднання до HTTP, перехоплюючи TLS-автентифікацію, що надсилається від програми користувачеві. Зловмисник надсилає користувачеві незашифровану версію сайту додатку, зберігаючи при цьому захищений сеанс з додатком. При цьому зловмисник бачить весь сеанс користувача.

2 СУТНІСТЬ ВЕБСЕРВЕРУ

2.1 Операційна система

Сервери функціонують на конкретних операційних системах. Щоб зробити вибір, потрібно проаналізувати деякі з найпопулярніших варіантів та дослідити їх ключові особливості.

Windows Server - це, можливо, найбільш зручний вибір для організації зберігання файлів. Але не слід порівнювати Windows Server з більшістю ОС, що застосовуються на звичайних комп'ютерах. Ця система відрізняється універсальністю і добре підходить для серверів додатків, файлових та поштових. Проте, під час експлуатації Windows Server можуть виникати питання з безпекою. Річ у тім, що велика частина вірусного програмного забезпечення створена саме для продуктів Microsoft, що може потенційно знижувати рівень захисту в процесі адміністрування. Однак варто зазначити, що нові версії системи демонструють кращу опірність до шкідливого програмного забезпечення. Додатково допомогти з цим можуть надійні антивірусні рішення.

FreeBSD – операційна система, що має солідний вік, проте й досі лишається дієвим рішенням для файлових серверів. Її прем'єрний випуск відбувся 1993 року. Це стабільне програмне забезпечення, здатне протистояти вірусам і хакерським зазіханням. Крім цього, FreeBSD забезпечує ефективне використання ресурсів у ситуаціях, коли одночасно запущено багато додатків. Вона також підтримує широкий спектр апаратних засобів. Значним плюсом FreeBSD є її гнучкість у налаштуваннях.

Недоліками можна вважати певну складність в інсталяції деяких пристроїв, нестачу детальної документації та необхідність володіти навичками роботи з командним рядком. FreeBSD не обладнана окремим графічним інтерфейсом. Налаштування реалізуються шляхом редагування відповідних конфігураційних файлів. Відтак, для користувача, який не має спеціалізованих знань, це може виявитись непростою задачею.

Дистрибутив Debian для Linux вражає своєю багатогранністю, ідеально підходячи як для серверів, так і для персональних комп’ютерів. Debian гарантує надійну та безперебійну роботу, через що багато хто визначає його як оптимальну операційну систему для серверного застосування. Хоча, певним недоліком можна вважати порівняно рідкі випуски оновлень.

Щодо вибору ОС для сервера, варто згадати про Red Hat Enterprise Linux. Ця система – чудовий вибір для корпоративних додатків, що зумовлює її широке використання. Головна перевага – її високий рівень надійності. Проте, Red Hat Enterprise Linux є платною ОС. Нові версії випускаються кожні три роки.

Ubuntu – безкоштовний дистрибутив з інтуїтивним керуванням, що добре підійде для серверів, які не зазнають інтенсивних навантажень на постійній основі. Варто відзначити, якщо ви зупинили вибір саме на серверній Ubuntu, то ви також обираєте відмінний рівень захисту даних. На сьогодні це одна з найпопулярніших систем для серверів, здатна забезпечити роботу навіть досить великих проектів [2].

Windows та Linux відрізняються одна від одної, що відіграє основну роль у виборі відповідного рішення.

Ядра найважливіший компонент для будь-якої ОС, тому його можна назвати вирішальним у питанні про те, на який сервер краще встановити.

Ядро Linux це:

Моноліт, реалізований одним файлом. Навіть якщо є потреба покращити функції, для цього потрібно використовувати модулі.

Взаємодія з програмами через системні виклики.

Таким чином, програмне забезпечення може працювати без змін на різних платформах Linux.

Має вбудовані драйвери для підвищення безпеки.

Ядро Windows має значні відмінності. Він характеризується наступними параметрами:

Велика кількість фрагментів бібліотек dll, кожна з яких має свій функціонал.

Відсутність використання системних викликів.

Застосування спеціальних бібліотек, які викликають функції з ntdll.dll.

Керування драйверами здійснюється через бібліотеку hal.dll. У цьому випадку кожен драйвер підключається до ядра окремо.

Система може адаптуватися до різного програмного забезпечення.

Ядро Windows дуже адаптивне. Також нові версії WinServ ідеально адаптовані для роботи в багатопоточному режимі.

Говорячи про Linux, відзначимо, що ця ОС пропонує наступні можливості:

Початок роботи з основного каталогу з подальшим підключенням дисків, розташованих у підкаталогах [11].

Сортування файлів за каталогами. При цьому враховується їх тип.

Розташування запам’ятовуючих пристроїв в алфавітному порядку.

Особливості розміщення файлів у Windows:

Linux-подібна класифікація дисків і розділів.

Кожна програма знаходиться в окремій папці з налаштуваннями, ресурсами та файлами.

Якщо розглядати, яку ОС вибрати для VPS виходячи з особливостей файлових систем і дисків, то відзначимо, що Windows використовується набагато рідше, так як ціна таких серверів висока через необхідність купувати ліцензію на ОС. Тому найкращим варіантом за співвідношенням ціни та якості буде Ubuntu Server.

У зберіганні налаштувань важливий не тільки комфорт користувача, але й безпека. Параметри Linux зберігаються в стандартних файлах і застосовуються до всіх користувачів. Якщо говорити про налаштування програмного забезпечення, то вони знаходяться в прихованих підкаталогах.

Тепер давайте подивимося, як саме зберігаються налаштування Windows:

Усі вони зберігаються у відповідному реєстрі.

Використовується розбиття по гілках і ключах.

Доступна опція для віддаленої зміни налаштувань програмного забезпечення.

Обидва варіанти досить безпечні, але у Windows є явний недолік - її неможливо перенести.

Специфіка управління програмним забезпеченням та оновленнями

Використання Linux забезпечує високий рівень безпеки та безперебійне оновлення. Що стосується Windows, то ця ОС має такі характеристики:

відсутність сховищ;

необхідність самостійного завантаження та встановлення програмного забезпечення;

програмне забезпечення, що самооновлюється.

Підводячи підсумок: певного лідера немає. Одних користувачів приваблює інтуїтивно зрозумілий інтерфейс Windows, а інших - Linux, який характеризується високими стандартами безпеки.

2.2 Вебсервер

Apache та Nginx є двома найпоширенішими вебсерверами з відкритим програмним кодом, які займають провідні позиції у глобальній мережі. Разом вони забезпечують функціонування більше ніж половини всього інтернет-трафіку. Обидва сервери відзначаються здатністю ефективно обробляти широкий спектр робочих навантажень, а також інтеграцією з іншими додатками для створення комплексного вебстеку [3, 4].

Попри значну кількість схожих характеристик, Apache та Nginx не є повністю взаємозамінними рішеннями. Кожен із них має свої унікальні переваги, які обумовлюють доцільність використання того чи іншого сервера залежно від конкретних умов. У цьому матеріалі представлено аналіз особливостей функціонування цих вебсерверів у різних сценаріях.

Вебсервер Apache був створений у 1995 році й уже через рік здобув широку популярність. Ідея його розробника, Роберта Маккула, полягала в тому, щоб створити потужне та гнучке програмне забезпечення, яке легко інтегрується з іншими системами. Цей задум успішно реалізували, і сьогодні Apache користується великою популярністю серед системних адміністраторів завдяки можливості розширення його функціоналу за допомогою модулів. Що стосується вебсервера Nginx, його створили у 2002 році. Ігор Сисоєв, розробник цього продукту, поставив

перед собою завдання створити легку, але надзвичайно продуктивну платформу, здатну справлятися з великими навантаженнями та масштабуватися без втрат ефективності. Проте Nginx поки важко конкурувати з Apache через значну базу користувачів останнього, попри його складніший характер.

Функціонування вебсервера Apache базується на створенні окремого процесу або потоку для кожного користувацького запиту. Цей підхід є досить простим у впровадженні, але не найкращим для проєктів із високою кількістю завдань. Кожен процес споживає оперативну пам’ять і системні ресурси, що обмежує його ефективність. Таким чином, Apache краще підходить для сайтів із низьким рівнем навантаження. У той час головним принципом роботи Nginx є використання дочірніх процесів, які займаються обробкою запитів. Завдяки цій архітектурі Nginx чудово підходить для високонавантажених ресурсів, здатних обслуговувати тисячі з’єднань одночасно.

Однією з ключових функцій Apache є можливість налаштовувати обробку запитів на рівні каталогів за допомогою спеціального прихованого файлу htaccess. Цей файл дозволяє впроваджувати такі налаштування, як авторизація та аутентифікація, управління кешуванням і контроль доступу до ресурсів. Крім того, зміни в конфігурації можна вносити безпосередньо під час роботи вебсервера, що не вимагає його перезавантаження або будь-яких додаткових налаштувань.

На відміну від Apache, вебсервер Nginx не підтримує аналогічної гнучкості. Він використовує лише один централізований конфігураційний файл, який обробляється основним процесом. Для застосування нових налаштувань потрібно надіслати сигнал цьому процесу і виконати перезавантаження сервера.

2.3 Бази даних

Реляційні системи управління базами даних (RDBMS), зокрема PostgreSQL та MySQL, відіграють вирішальну роль у процесах зберігання, організації та доступу до даних, які є основою для розробки додатків і виконання аналітичних завдань [7]. Ці дві системи є одними з найпопулярніших серед відкритих баз даних,

демонструючи значний розвиток протягом тривалого періоду їх існування та пропонуючи широкий спектр функціональних можливостей. Поряд із багатьма спільними рисами, PostgreSQL і MySQL істотно відрізняються як своїми технічними архітектурними підходами, так і загальною філософією дизайну, що визначає напрямок їхньої еволюції та сферу застосування.

MySQL є безкоштовною системою управління реляційними базами даних, розробленою компанією «TcX» із метою підвищення ефективності обробки великих обсягів даних. Спочатку вона мала значну схожість із системою mSQL, однак у процесі еволюції суттєво розширила свої можливості. Наразі MySQL належить до найпоширеніших систем управління базами даних, активно застосовується для створення динамічних вебресурсів завдяки високому рівню сумісності з різними мовами програмування.

PostgreSQL є потужною об’єктно-реляційною системою управління базами даних з відкритим кодом, яка розробляється вже більше 35 років. Вона здобула широку популярність завдяки своїй високій надійності, функціональності та продуктивності. PostgreSQL особливо добре підходить для корпоративних додатків з інтенсивними операціями запису й виконання складних запитів.

До головних переваг MySQL належать її висока продуктивність у роботі з транзакціями, широка підтримка інструментів і середовищ розробки, а також стабільна робота при невеликих обсягах даних. Крім того, система пропонує ефективну реплікацію та забезпечує швидке виконання простих запитів. Завдяки своїй популярності MySQL інтегрується з багатьма вебсервісами, а також має розвинену документацію та підтримку спільноти.

Втім, MySQL має низку обмежень. Основними недоліками є її недостатня функціональність для обробки складних аналітичних запитів, відсутність підтримки розширених конструкцій SQL у старіших версіях, а також нижча ефективність при роботі з великими масивами даних порівняно з іншими системами. Архітектура MySQL оптимізована насамперед для OLTP-навантажень, що знижує її ефективність у контексті аналітичних задач.

PostgreSQL, у свою чергу, вирізняється широкими можливостями для аналітики: підтримкою складних запитів, загальних табличних виразів, віконних функцій, часткових індексів та роботи з JSON-даними. Ця система може успішно справлятися як із транзакційними, так і з аналітичними навантаженнями, завдяки чому є універсальним рішенням для різноманітних потреб.

Проте до недоліків PostgreSQL слід віднести значне споживання ресурсів при роботі з великими обсягами даних та необхідність детального налаштування для досягнення оптимальної продуктивності за умов високого навантаження. Ще однією її особливістю є більша тривалість процесу оптимізації запитів у складних сценаріях використання порівняно з іншими системами управління базами даних [12].

2.4 Захист вебсерверу

Фаєрвол, також відомий як вогняна стіна, являє собою міжмережевий екран, що функціонує на базі апаратних чи програмних засобів. Його основне завдання - контролювати мережевий трафік, що проходить через нього. По суті, фаєрвол перевіряє кожен пакет даних, який приходить або відправляється з вашого пристрою, і блокує ті, які можуть бути потенційно небезпечними або викликають підозру.

Фаєрвол виконує широкий спектр завдань, серед яких ключовими є:

фільтрація мережевого трафіку;

оперативне реагування та блокування спроб вторгнення;

аналіз взаємозв’язків між подіями на мережевих пристроях для виявлення потенційно підозрілої активності.

Брандмауер відіграє фундаментальну роль у забезпеченні безпеки мережі, виступаючи першою лінією оборони. Його головна функція - фільтрувати трафік і контролювати доступ до ресурсів системи. За допомогою файрволу можна значно посилити конфіденційність даних, обмежуючи доступ лише для авторизованих користувачів і мінімізуючи ризик несанкціонованих проникнень.

Файрвол застосовує різноманітні методи перевірки даних, включаючи оцінку правил доступу, інспекцію пакетів і аналіз стану з’єднань. Завдяки цьому він здатний блокувати підозрілі IP-адреси, порти чи протоколи, одночасно запобігаючи спробам атак або злому системи.

Існує декілька поколінь мережевих екранів, кожне з яких має унікальні підходи до фільтрації мережевого трафіку.

Брандмауери першого покоління функціонували на основі зіставлення базових параметрів мережевих пакетів, таких як джерело, призначення, порт і протокол. Вони зазвичай реалізовувалися у вигляді фільтрів пакетів.

Удосконалення були запроваджені у другому поколінні, яке використовує аналіз стану з’єднання для моніторингу ініціалізації та поточного стану кожного сеансу взаємодії.

У свою чергу, брандмауери третього покоління, до яких належать і сучасні моделі, здатні фільтрувати дані на всіх рівнях моделі OSI. Вони володіють розвиненими можливостями ідентифікації, зокрема дозволяють виявляти специфічні програми та популярні протоколи, такі як FTP і HTTP. Окрім цього, сучасні мережеві екрани часто інтегрують додаткові механізми безпеки, включно з підтримкою VPN, системами запобігання та виявлення вторгнень (IPS/IDS), управлінням ідентифікацією користувачів і вебфільтрацією. Ці розширені функції значно підвищують рівень захисту мереж і конфіденційності даних.

Метод грубої сили являє собою техніку, яка базується на послідовному переборі можливих комбінацій паролів або ключів шифрування до моменту виявлення правильного значення, часто із залученням автоматизованих інструментів. Ця стратегія функціонує за принципом спроб і помилок та здебільшого використовується для здобуття несанкціонованого доступу до інформаційних систем, мереж або облікових записів [13].

Атаки типу грубої сили належать до числа найстаріших і найпримітивніших підходів, що їх застосовують зловмисники у сфері кібербезпеки. Незважаючи на свою простоту, цей метод залишається дієвим завдяки легкості реалізації та

перспективі значної вигоди для злочинців. Ці атаки можуть бути спрямовані як на персональні облікові записи, так і на великомасштабні корпоративні бази даних.

Метод грубої сили передбачає послідовний перебір численних комбінацій символів, цифр та знаків з метою підбору пароля або ключа шифрування. Кіберзлочинці зазвичай автоматизують цей процес за допомогою спеціалізованих програмних інструментів, що дозволяє їм швидко перевіряти величезну кількість можливих варіантів. Ефективність такого методу напряму залежить від складності та довжини пароля чи ключа. Короткі та прості паролі набагато легше зламати, тоді як довші та складніші потребують значно більше часу і ресурсів для їх підбору. Хоч цей метод і здається базовим, він може виявитися дуже результативним у випадках із слабкими паролями або недостатньо потужними системами безпеки.

3 ПРАКТИЧНА ЧАСТИНА

3.1 Розгортання серверу та вебсерверу

Почнемо з встановлення Ubuntu LTS:

Обираємо мову сервера(бажано завжди обирати англійську, див. рисунок 3.1):

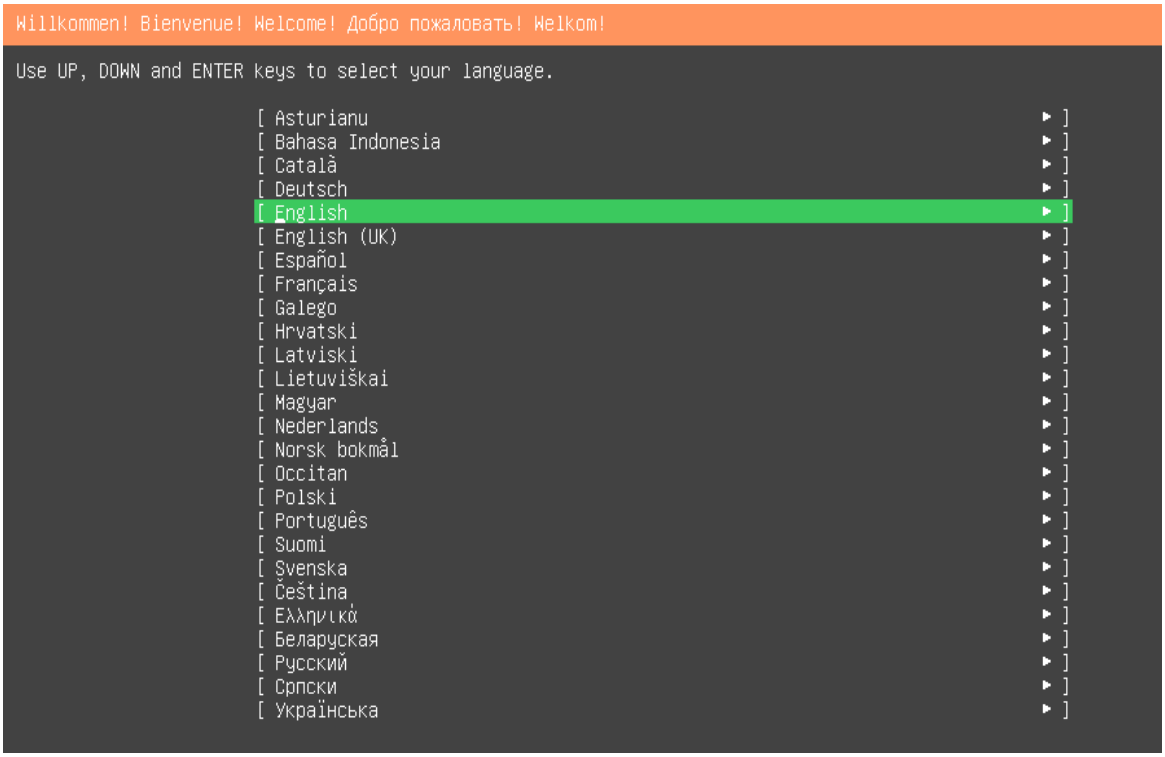

Рисунок 3.1 – Мова серверу

Вибираємо розкладку клавіатури (Див. рисунок 3.2):

Рисунок 3.2 – Мова розкладки клавіатури

Обираємо тип встановлення Ubuntu Server (Див. рисунок 3.3):

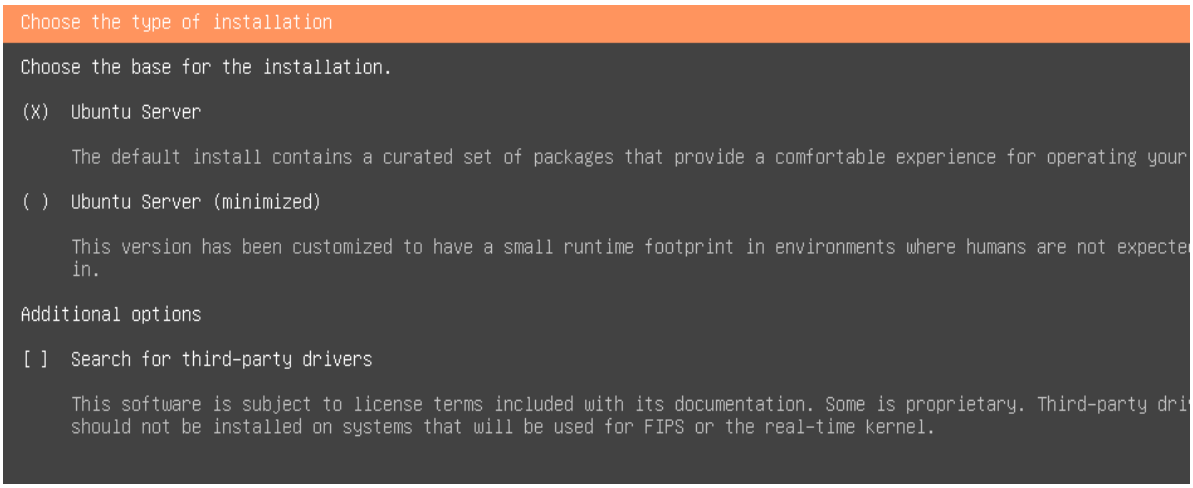

Рисунок 3.3 – Тип встановлення Ubuntu Server

Налаштуйте мережу. За замовчуванням, отримання IP-адреси налаштовано за DHCP (Див. рисунок 3.4):

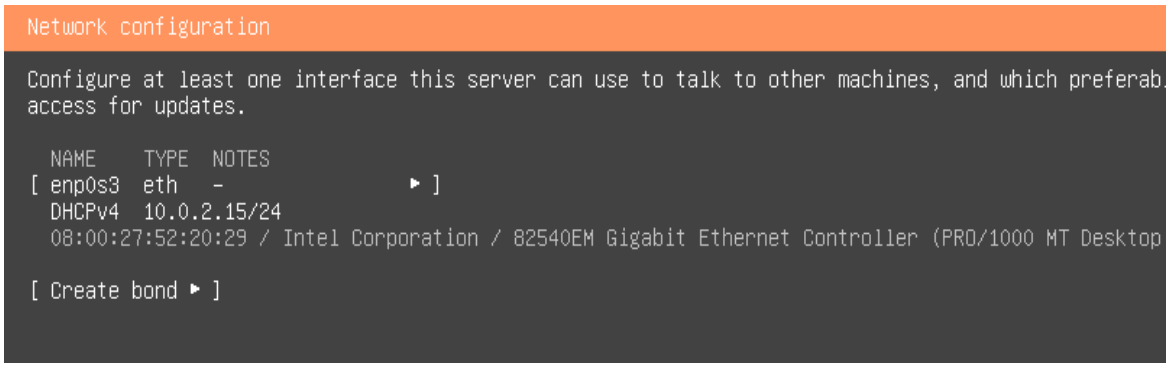

Рисунок 3.4 – Налаштування мережі

Якщо є proxy-сервер, то його можна вказати (Див. рисунок 3.5):

Рисунок 3.5 – Налаштування proxy-сервера

Далі інсталятор запропонує вам найближче дзеркало (Mirror), виходячи з вашого регіонального розташування. Залишаємо за замовчування (Див. рисунок 3.6):

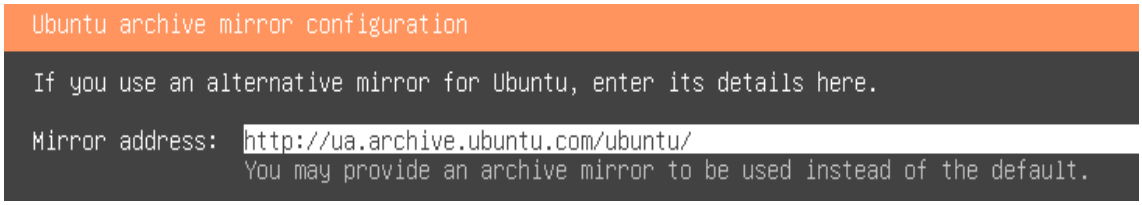

Рисунок 3.6 – Посилання на дрезкало

На цьому етапі буде запропоновано розмітити дисковий простір. Вибравши «Use an entire disk» установник сам розмітить диски в автоматичному режимі. Залежно від завдань, ви можете виконати розбивку розділів на власний розсуд, вибравши «Custom storage layout» (Див. рисунок 3.7):

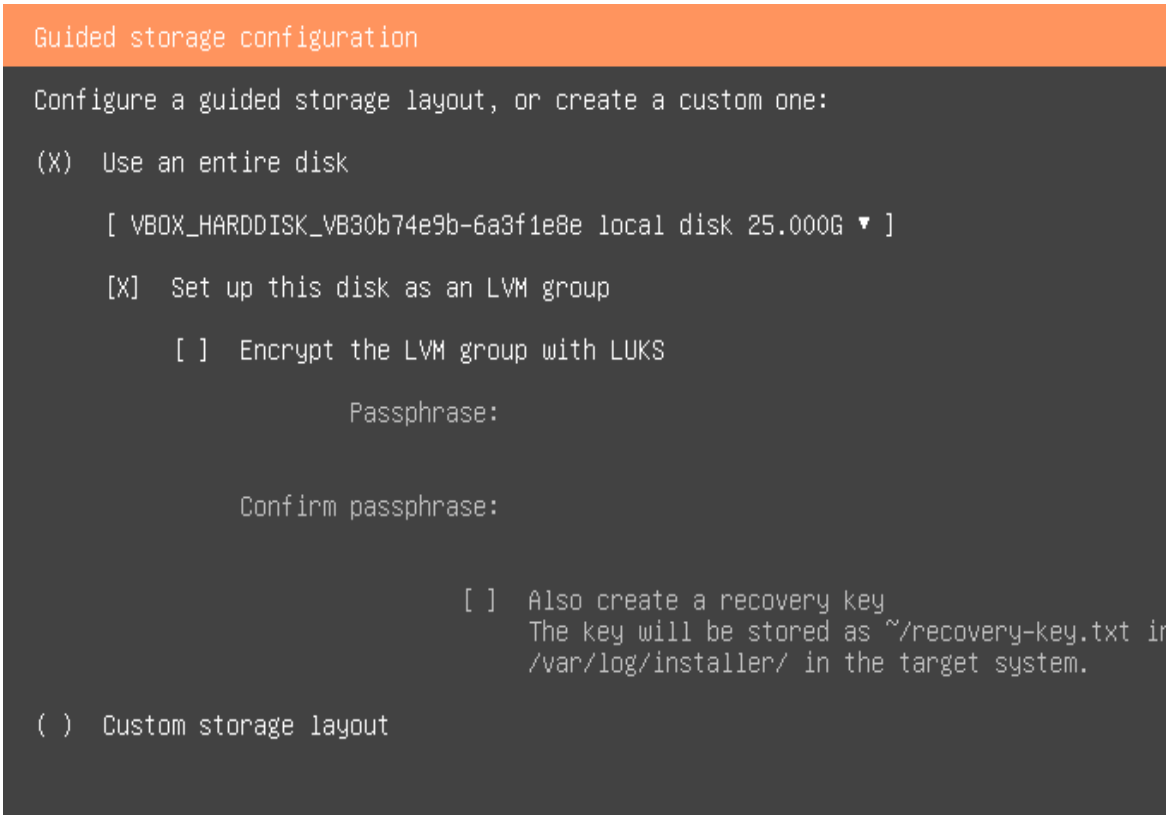

Рисунок 3.7 – Розмітка диску

Виберіть «Continue» для підтвердження налаштувань (Див. рисунок 3.8):

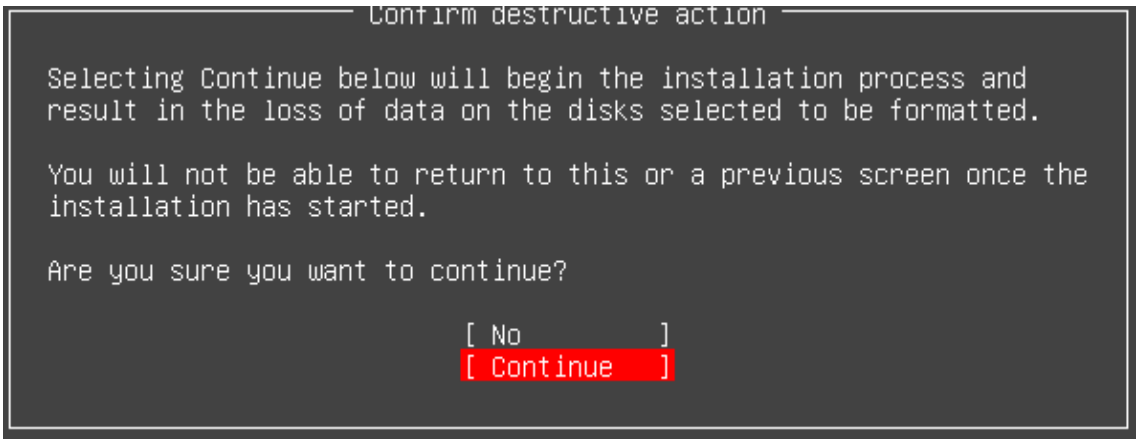

Рисунок 3.8 – Підтвердження налаштувань

Зазначте ім’я сервера та надайте дані користувача, необхідні для підключення до нього. Зверніть увагу, що вибір користувача має виключати "root" і "admin", оскільки ці облікові записи зарезервовані системою для адміністративних потреб (Див. рисунок 3.9):

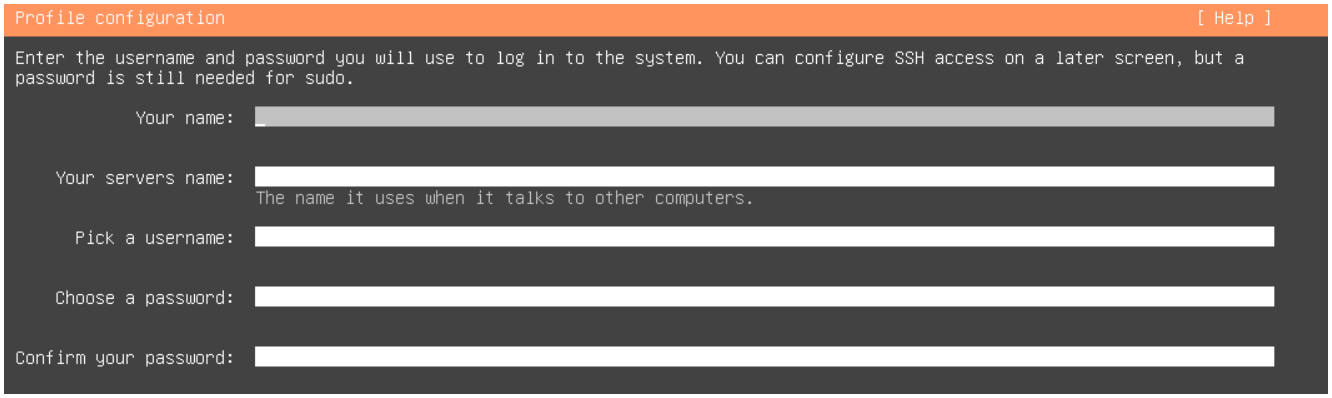

Рисунок 3.9 – Налаштування імені сервера та даних користувача

Виберіть опцію встановлення OpenSSH Server, щоб забезпечити можливість віддаленого підключення до серверу, після чого натисніть «Готово» (Див. рисунок 3.10).

Рисунок 3.10 – Опція встановлення OpenSSH

На даному етапі перед вами відкриється широкий перелік доступних компонентів для встановлення. Особливу увагу привертає інтеграція з Kubernetes версії 1.18, сервісом etcd, а також можливості інтеграції з платформою Google Cloud, базою даних PostgreSQL версії 10, системою моніторингу Prometheus та багатьма іншими інструментами. Процес вибору компонентів слід здійснювати зважено та обґрунтовано, враховуючи реальну необхідність їх використання для розв'язання ваших завдань (Див. рисунок 3.11).

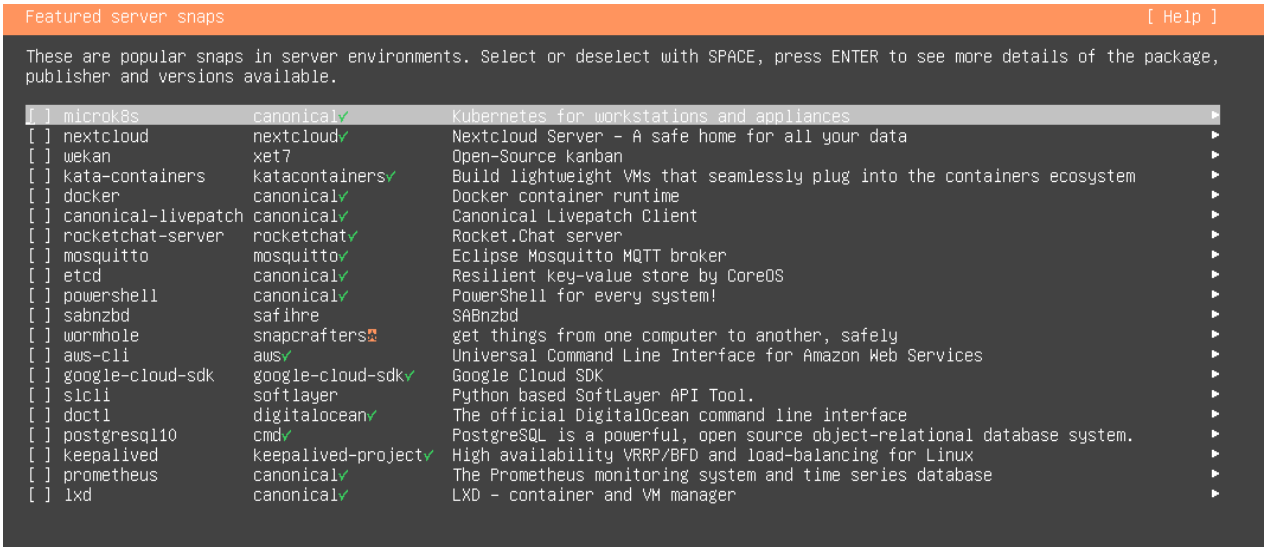

Рисунок 3.11 – Компоненти для встановлення

Після підтвердження, чекаємо встановлення та перезапускаємо систему. Для вебсерверу було обрано Apach2 (Див. рисунок 3.12):

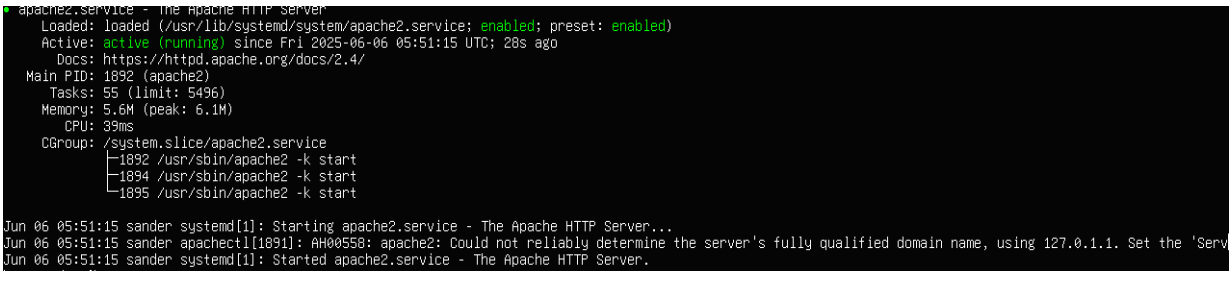

Рисунок 3.12 – Перевірка статусу Apache2

3.2 Розгортання системи захисту

Перши із методів захисту було обрано фаервол ufw та встановлено за допомогою команди [9, 10].

sudo apt install ufw

Перевірка роботи ufw. Спочатку ufw неактивний (Див. рисунок 3.13).

```
root@sander:~# ufw status
Status: inactive
```

Рисунок 3.13 – Перевірка роботи ufw

Заборонимо всі вхідні з’єднання та дозволимо всі вихідні. Таким чином, служби зможуть безперешкодно з’єднуватися "назовні", але всі зовнішні запити відхилятимуться (Див. рисунок 3.14).

```
root@sander:~# ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
root@sander:~# ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Рисунок 3.14 – Налаштування ufw

Відкриємо вхідні з’єднання для служби ssh. Зробимо це відразу за назвою служби, номером порту тощо (Див. рисунок 3.15).

```
root@sander:~# ufw allow OpenSSH
Rules updated
Rules updated (v6)
root@sander:~# ufw allow ssh
Rules updated
Rules updated (v6)
root@sander:~# ufw allow 22
Rules updated
Rules updated (v6)
```

Рисунок 3.15 – Налаштування ufw

Після активуємо файрвол та перевіряємо статус (Див. рисунок 3.16):

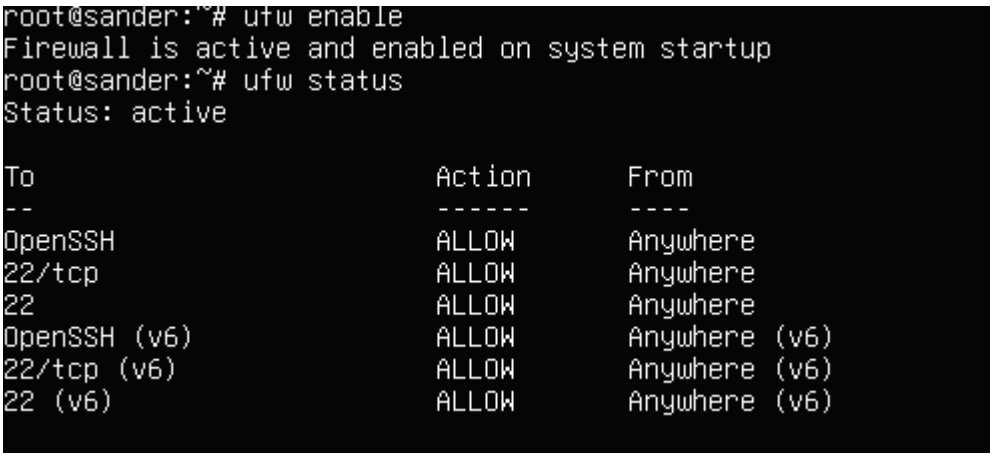

Рисунок 3.16 – Активація та перевірка статусу ufw

Таким чином firewall налаштований на максимальну безпеку та можливість доступу до ssh.

Далі була налаштовано утиліту Fail2ban від взламу грубої шкоди

sudo apt install fail2ban

Після встановлення одразу активуємо служби (Див. рисунок 3.17):

sudo systemctl enable --now

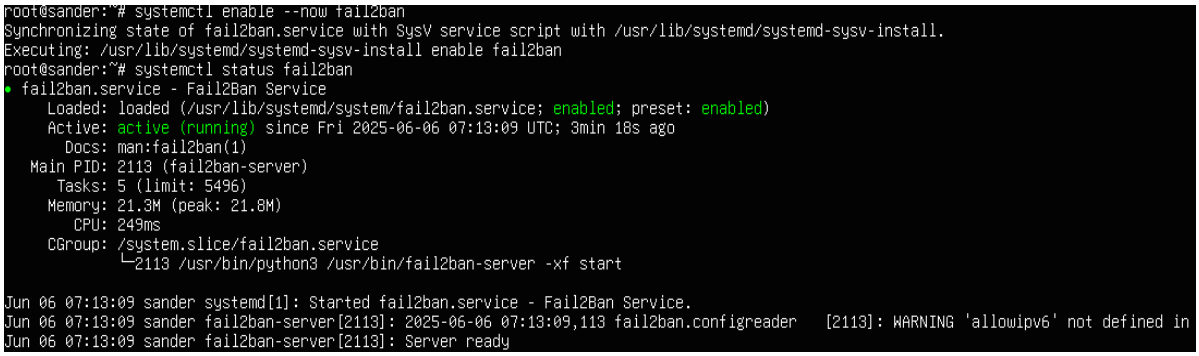

Рисунок 3.17 – Перевірка статусу fail2ban

Основний конфігураційний файл знаходиться за адресою /etc/fail2ban/jail.conf. Однак, спершу бажано не редагувати його безпосередньо. Замість цього скопіюйте його під назвою jail.local. Всі корективи будемо вносити саме у цьому файлі, оскільки такий підхід рекомендують розробники. Будь-які зміни, які ви здійсните у jail.local, автоматично замінять відповідні налаштування в jail.conf (Див. рисунок 3.18):

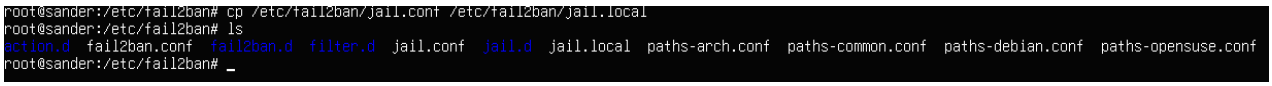

Рисунок 3.18 –– Файли у директорії

Потрібно знайти рядок з позначкою #ignoreip у конфігураційному файлі та видалити коментар, додавши через пробіл вашу зовнішню статичну IP-адресу. У підсумку рядок матиме вигляд:

ignoreip = 127.0.0.1/8 ::1 178.20.***.**

де "178.20.***.**" слід замінити на вашу реальну IP-адресу.

Також у цій же секції [DEFAULT] можна налаштувати параметри, такі як findtime і maxretry, відповідно до ваших потреб. Наприклад:

bantime = 600

findtime = 3600

maxretry = 6

Параметри findtime та maxretry визначають умови, за яких буде ініційовано автоматичне блокування небажаних користувачів. Maxretry відповідає за кількість невдалих спроб входу, а findtime вказує часовий проміжок, протягом якого користувач повинен пройти успішну аутентифікацію. У разі перевищення встановлених значень будь-якого з цих параметрів клієнт опиняється в списку заблокованих. Bantime задає тривалість блокування і вимірюється в секундах. Ці параметри будуть зустрічатися в тексті неодноразово. Якщо для будь-якого сервісу ці параметри не задані вручну, система автоматично підхоплює значення з секції [DEFAULT].

Після внесення змін до конфігураційного файлу Fail2ban необхідно перезапускати службу.

Захист сервісу sshd за допомогою Fail2ban

Насамперед очистимо наш дефолтний конфігураційний файл:

$ cat /dev/null > /etc/fail2ban/jail.d/defaults-debian.conf

У каталозі jail.d/*.*, що відповідає за налаштування сервісів, створимо файл sshd.conf і додамо в нього такі рядки (Див. рисунок 3.19):

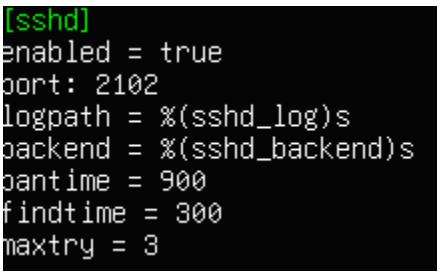

Рисунок 3.19 – Налаштування sshd.conf

Щоб застосувати зміни, перезапустимо сервіс і внесемо його в автозавантаження (Див. рисунок 3.20):

Рисунок 3.20 – Перезапуск сервісу і внесеня його в автозавантаження

Давайте детально розглянемо налаштування наших дій, що будуть використовуватися повторно для конфігурації інших захищених сервісів:

[sshd] – визначення сервісу, який налаштовується, у цьому випадку це sshd.

enabled = true – параметр, що активує моніторинг даного сервісу.

port: 2102 – якщо ви використовуєте стандартний порт (22) для служби SSH, цей рядок можна пропустити. Проте, якщо використовується нестандартний порт, як-от 2102 у нашому прикладі, його необхідно вказати.

bantime = 900 – тривалість блокування порушника (у секундах).

findtime = 300 – часовий період для виявлення порушення.

maxretry = 3 – максимально допустима кількість невдалих спроб аутентифікації. На четвертій спробі IP-адреса порушника буде заблокована.

Як це працює: Наприклад, порушник із IP-адресою 123.123.12.13 намагається підібрати пароль до облікового запису вашого сервера. Параметр "findtime" задає тривалість моніторингу IP-адреси з моменту першої появи її у логах (/var/log/auth.log). Якщо протягом 300 секунд (5 хвилин) порушник зробить понад три помилки аутентифікації, його IP буде заблоковано на 900 секунд (15 хвилин).

Далі треба налаштувати захист Apache2

Припустімо, що на ваших вебресурсах існують секції, доступ до яких потребує аутентифікації. Для забезпечення базового захисту цих розділів від атак методом перебору паролів доцільним є використання інструменту Fail2ban. За умовчанням, для середовища Apache2 засоби захисту вже інтегровані, проте вони потребують активації для їхнього початку функціонування. Зокрема, необхідно здійснити коригування конфігураційного файлу, який розташований за адресою /etc/fail2ban/jail.local. У цьому файлі слід знайти блок [apache-auth], в якому першочергово зазначений лише параметр logpath. Для активації захисту до цього блоку потрібно додати відповідні значення (Див. рисунок 3.21).

```
[apache-auth]
enabled = true
filter = nginx-http-auth
port     = http,https
bantime = 600
maxretry = 6

logpath  = %(apache_error_log)s
```

Рисунок 3.21 – Налаштування блоку apache-auth

Тепер переходимо до файлу apache-auth.conf (Див. рисунок 3.22)

nano /etc/fail2ban/filter.d/apache-auth.conf

```
failregex = ^ \[error\] \d+#\d+: \*\d+ user "\S+":? (password mismatch|was not found in ".*"),client:>
            ^ \[error\] \d+#\d+: \*\d+ no user/password was provided for basic authentication,client:>
            ^client (?:denied by server configuration|used wrong authentication scheme)\b
            ^user (?!`)<F-USER>(?:\S*|.*?)</F-USER> (?:auth(?:oriz|entic)ation failure|not found|deni>
            ^Authorization of user <F-USER>(?:\S*|.*?)</F-USER> to access .*? failed\b
            ^%(auth_type)suser <F-USER>(?:\S*|.*?)</F-USER>: password mismatch\b
            ^%(auth_type)suser `<F-USER>(?:[^']*|.*?)</F-USER>' in realm `.+' (auth(?:oriz|entic)atio>
            ^%(auth_type)sinvalid nonce .* received - length is not\b
            ^%(auth_type)srealm mismatch - got `(?:[^']*|.*?)' but expected\b
            ^%(auth_type)sunknown algorithm `(?:[^']*|.*?)' received\b
            ^invalid qop `(?:[^']*|.*?)' received\b
            ^%(auth_type)sinvalid nonce .*? received - user attempted time travel\b
            ^(?:No h|H)ostname \S+ provided via SNI(?:, but no hostname provided| and hostname \S+ pr>

ignoreregex =
```

Рисунок 3.22 – Завершення налаштування Fail2ban

На цьому система захисту налаштована

3.3 Розгортання системи резервного копіювання

Rsync входить до базового програмного забезпечення більшості дистрибутивів Linux-подібних операційних систем і, тому, не потребує встановлення. Але, у разі чого, її завжди можна встановити із відкритого репозиторію стандартними засобами

Нами буде створено два каталоги, один з яких буде джерелом даних, а інший буде місцем призначення, тобто, його вміст буде синхронізуватися із «джерелом» за допомогою rsync.

Хоча rsync, ймовірно, уже встановлено. Щоб переконатися, що rsync встановлено та оновлено, виконайте на обох комп’ютерах наступне:

sudo apt install rsync

На цільовому комп’ютері для отримання з джерела замість надсилання з джерела до цільового. Для цього вам потрібно налаштувати пару ключів SSH. Після створення пари ключів SSH перевірте доступ без пароля від цільового комп’ютера до вихідного (Див. рисунок 3.23, рисунок 3.24).

```
root@sander:/etc/apache2# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:o3utUYGoslz9uQaQzHCYMj4VAFOpJ4/OgaoQr2zkbU8 root@sander
The key's randomart image is:
+--[ED25519 256]--+
|oo.*o.           |
|+ X.=  . .       |
|.B.% o. . .      |
|o+o Bo     .     |
|*.+ o..  S .     |
|o= * E.o +       |
|. + o ..+.       |
|     . .oo.      |
|      .oo.       |
+----[SHA256]-----+
```

Рисунок 3.23 – Генерація публічного ключа

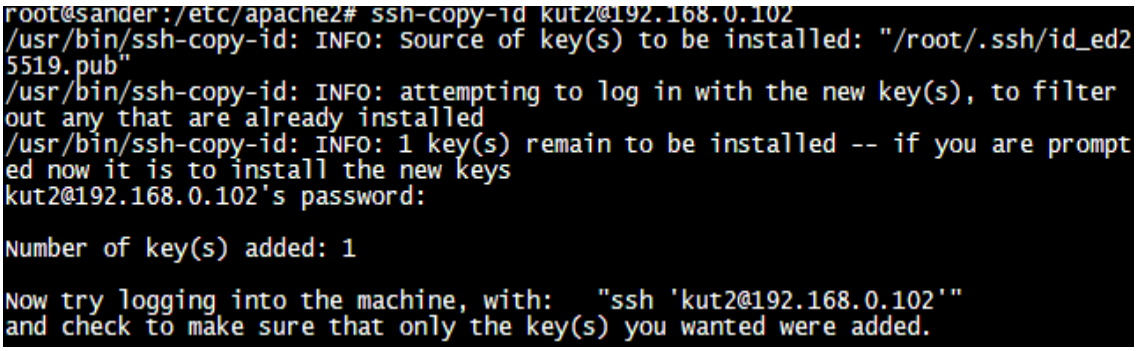

Рисунок 3.24 – Передача публічного ключа іншому серверу

Далі встановіть сценарій на цільовому комп'ютері, створивши для нього файл. Щоб створити файл, використовуйте цю команду [6]:

```
nano /usr/local/sbin/rsync_dirs
```

Додаємо вміст:

```
#!/usr/bin/env bash
sudo sync -a /etc/apache2 --delete kut2@192.168.0.102:/home/kut2/copy/apache2-copy/
sudo sync -a /var/www/html --delete kut2@192.168.0.102:/home/kut2/copy/var-copy/
```

Після створення скрипта треба зробити файл виконавчим:

```
chmod +x /usr/local/sbin/rsync_dirs
```

Використовуюси crontab, щоб скрипт виконувався автоматично за розкладом. Щоб запускати цей сценарій щовечора об 23:00 (Див. рисунок 3.25) [1]:

```
crontab -e
```

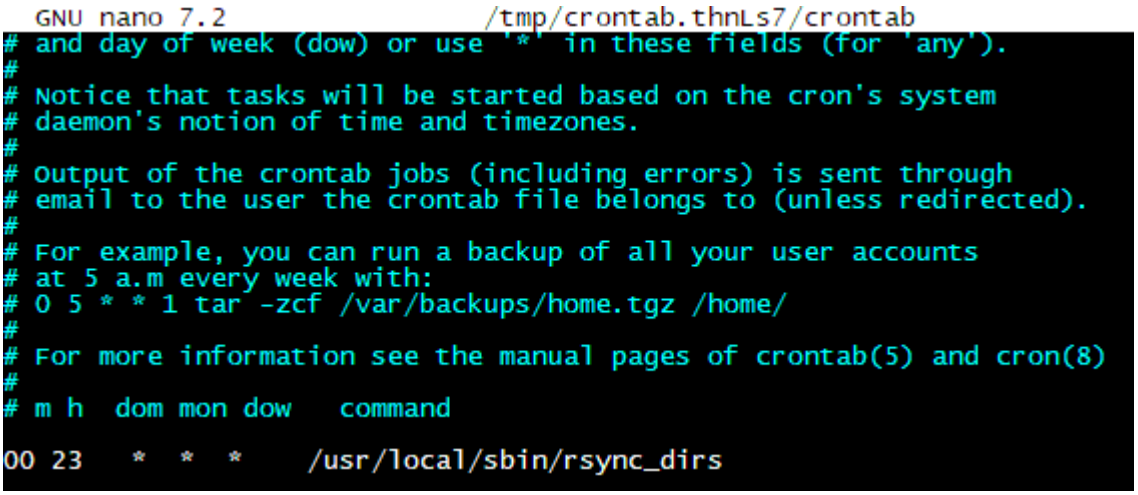

Рисунок 3.25 – Налаштування crontab -e

Для резервного копіювання БД створимо окремого користувача:

```
sudo mysql -u root
```

Тепер треба створити користувача та видати йому доступи (Див. рисунок 3.26):

```
mysql> CREATE USER 'backup'@'localhost' IDENTIFIED BY 'Timetoshine00_';
Query OK, 0 rows affected (0.05 sec)

mysql> GRANT SELECT, LOCK TABLES ON sander.* TO 'backup'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Рисунок 3.26 – Створити користувача та видати йому доступи

Далі пишемо скрипт для створення дампу та відправки його на інший сервер (Див. рисунок 3.27)

```
#!/bin/bash

DB_USER="backup"
DB_PASS="Timetoshine00_"
DB_NAME="назва_бази"
BACKUP_DIR="/home/kut/mysql_backups"
DATE=$(date +"%Y-%m-%d_%H-%M-%S")
BACKUP_FILE="${DB_NAME}_backup_${DATE}.sql"
ARCHIVE_FILE="${BACKUP_FILE}.gz"

REMOTE_USER="kut2"
REMOTE_HOST="192.168.0.102"
REMOTE_DIR="/home/kut2/copy/sql-copy"

mkdir -p "$BACKUP_DIR"

mysqldump -u "$DB_USER" -p"$DB_PASS" "$DB_NAME" > "$BACKUP_DIR/$BACKUP_FILE"

gzip "$BACKUP_DIR/$BACKUP_FILE"

scp "$BACKUP_DIR/$ARCHIVE_FILE" "$REMOTE_USER@$REMOTE_HOST:$REMOTE_DIR/"

rm "$BACKUP_DIR/$ARCHIVE_FILE"
```

Рисунок 3.26 – Скрипт для створення дампу

На цьому налаштування завершено.

ВИСНОВОК

У межах кваліфікаційної роботи виконано успішну розробку та дослідження комплексу заходів, спрямованих на забезпечення кібербезпеки й безперебійної роботи сучасних вебсерверів. У реаліях, де цифрові активи стали критично важливими для функціонування будь-якої організації, а загрози постійно еволюціонують, актуальність цієї теми є незаперечною.

Досягнуті результати дослідження охоплюють кілька важливих аспектів:

Аналіз загроз. Проведено всебічне вивчення актуальних кіберзагроз, пов’язаних із вебсерверами, включаючи виявлення вразливостей і методів DDoS-атак. Це дало змогу розробити комплексний підхід до їх ефективної нейтралізації.

Систематизація інструментів. Проаналізовано й оцінено ефективність сучасних технологій, таких як міжмережеві екрани UFW, вебсервер Nginx, системи управління базами даних MySQL і утиліта Rsync для синхронізації даних. Ці інструменти стали основою інтегрованої системи захисту.

Практична реалізація. Здійснена імплементація ключових модулів системи продемонструвала не тільки її теоретичну доцільність, а й практичну ефективність у реальних умовах експлуатації.

Запропонована система є ефективним інструментом для підвищення захисту даних і забезпечення безперервної роботи вебресурсів. Її результати можуть успішно адаптуватися та впроваджуватися для забезпечення безпеки різноманітних вебдодатків та інформаційних інфраструктур.

ПЕРЕЛІК ПОСИЛАНЬ

1. Crontab - maintains crontab files for individual users. *index - Debian Manpages*. URL: https://manpages.debian.org/unstable/cronie/crontab.1.en.html.
2. Enterprise open source and linux | ubuntu. *Ubuntu*. URL: https://ubuntu.com/.
3. Manpages of apache2 in debian jessie - debian manpages. *index - Debian Manpages*. URL: https://manpages.debian.org/jessie/apache2/index.html.
4. Nginx documentation. *nginx*. URL: https://nginx.org/en/docs/.
5. Omer Yoachimik, Jorge Pacheco. Record-breaking 5.6 Tbps DDoS attack and global DDoS trends for 2024 Q4. *blog.cloudflare.com*. URL: https://blog.cloudflare.com/ddos-threat-report-for-2024-q4/.
6. Rsync - a fast, versatile, remote (and local) file-copying tool. URL: https://download.samba.org/pub/rsync/rsync.1.
7. The MySQL Command-Line Client. *MySQL*. URL: https://dev.mysql.com/doc/refman/8.4/en/mysql.html.
8. Types of backup: full, incremental and differential backup - adivi 2025. *Adivi Corporation*. URL: https://adivi.com/blog/types-of-backup/.
9. Ubuntu Manpage: ufw - program for managing a netfilter firewall. *Ubuntu Manpage:Welcome*. URL: https://manpages.ubuntu.com/manpages/trusty/man8/ufw.8.html.
10. Website security - Learn web development | MDN. *MDN Web Docs*. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/First_steps/Website_security.
11. Boiko V., Vasilenko N., Slatvinska V. Distributed Systems Log Protection from Cyberattacks by Verkle Trees // Information Technology for Education, Science, and Technics. Springer Nature Switzerland, 2024. P. 221-234.
12. Бойко В., Василенко М., Слатвінська В. Моделювання живучості та відновлення інформаційно-комунікаційних мереж в умовах дії кіберзагроз // Інформаційні технології та суспільство. 2024. № 1 (12). P. 13-19.

13. Бойко В., Горбаченко С. Тестування на проникнення як ефективний інструмент менеджменту кібербезпеки // Інформаційні технології та суспільство. 2023. № 3 (9). Р. 23-29.