ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Технологія захисту сервера на основі методології STRIDE»

зі спеціальності *125 Кібербезпека та захист інформації*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Михайло ЧУМАК
(підпис)

Виконав: здобувач вищої освіти групи БСДМ-61
ЧУМАК Михайло
(прізвище, ім'я)

Керівник д-р філос., доц., МАРЧЕНКО Віталій
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

ASLR – Address Space Layout Randomization

CIS – Center for Internet Security

CTEM – Continuous Threat Exposure Management

CVE – Common Vulnerabilities and Exposures

CWE – Common Weakness Enumeration

DDoS – Distributed Denial of Service

DEP – Data Execution Prevention

EDR – Endpoint Detection and Response

ENISA – European Union Agency for Cybersecurity

IDS – Intrusion Detection System

IDPS – Intrusion Detection and Prevention Systems

IPS – Intrusion Prevention System

ISMS – Information Security Management System

LAN – Local Area Network

MFA – Multi-Factor Authentication

NFV – Network Function Virtualization

NIST – National Institute of Standards and Technology

NVD – National Vulnerability Database

OWASP – Open Worldwide Application Security Project

SMTP – Simple Mail Transfer Protocol

SQL – Structured Query Language

SSD – Solid-State Drive

SSH – Secure Shell

SSL – Secure Sockets Layer

ВСТУП

Актуальність дослідження. Електронні комунікації один із головних пріоритетів України - функціонування цифрової держави. Саме тому Міністерство цифрової трансформації України сформували стратегію для ефективного розвитку сфери до 2030 року, яка зосереджена на забезпеченні розвитку сфери електронних комунікацій в сучасних умовах з урахуванням світових тенденцій та особливостей розвитку України, а також потреб у змінах для реалізації державної політики та міжнародних зобов'язань України.

Захист інформації залишається однією з найактуальніших задач внаслідок надзвичайно широкого розповсюдження як автоматизованих систем обробки інформації, так і розширення локальних та глобальних комп'ютерних мереж, якими передаються величезні об'єми інформації державного, військового, комерційного, приватного характеру. Розвиток кіберзахисту - невід'ємна складова цифровізації України.

Сучасні наукові та практичні підходи до кіберзахисту критичної інфраструктури охоплюють широкий спектр методологій, міжнародних стандартів, національних галузевих вимог, архітектурних принципів та технічних рішень, які постійно еволюціонують у відповідь на нові та дедалі складніші загрози кіберпростору.

Серверні системи складають ядро сучасної цифрової інфраструктури організацій усіх рівнів - від малих підприємств до великих корпорацій, державних установ та критичної інфраструктури. Центральна роль серверів у сучасних організаціях - це пріоритетна мішень для різних категорій зловмисників. Проблема захисту серверів розвивалася в кілька етапів, кожний з яких визначався технологічним контекстом та типами загроз. Сучасний період - поширення мікросервісів, контейнеризації, хмарних, віртуалізованих і SDN/NFV середовищ додали нових

аспектів і проблеми захисту набули надзвичайної актуальності. Атаки стали більш вишуканішими і цільовими, кіберзлочинці все частіше використовують штучний інтелект для підвищення продуктивності та оптимізації своїх зловмисних дій. Захист серверів є однією з найбільш критичних складових сучасної інформаційної безпеки, вимагаючи нових підходів до захисту. Розв'язання проблеми захисту серверів потребує комплексного підходу, який поєднує теоретично обґрунтовані моделі, моделювання загроз з практичними механізмами безпеки та постійним урахуванням еволюції дестабілізуючої дії спектру вразливостей.

Моделювання загроз є одним із найбільш визнаних наукових підходів до забезпечення безпеки серверних систем з самого початку їх проектування. Практичний інструмент моделювання загроз - методологія STRIDE представляє один із найбільш значущих інженерно-технічних підходів до категоризації та опису загроз у контексті розробки та експлуатації програмного забезпечення, допомагає виявляти потенційні вектори атак та розробляти ефективні механізми захисту, включаючи серверні додатки.

Проаналізовано найкритичніші аспекти забезпечення безпеки серверів: конфіденційності, цілісності та доступності, характер сучасних загроз і типових слабких місць реальних систем, виділено групи проблем, які системно повторюються у більшості інцидентів і прийнято як «найбільш критичні» для будь-якої серверної інфраструктури. Проведено оцінювання сучасних методів і засобів захисту серверів та сформовано напрями удосконалення.

Побудова моделі загроз серверної системи за методологією STRIDE представлена у третьому розділі і базується на результатах теоретичного аналізу першого розділу та комплексного аналітичного дослідження проведеного в другому розділі. Запропоновано технологію забезпечення захисту серверів за методологією STRIDE та сучасні підходи щодо виявлення і усунення загроз, що передбачає перехід від абстрактної класифікації загроз до конкретного комплексу заходів, інтегрованих в архітектуру та процеси експлуатації серверної системи. Сформовано комплекс

рекомендацій щодо усунення виявлених загроз, підвищення захищеності серверної інфраструктури та можливості практичного застосування технології забезпечення захисту серверів.

Тема кваліфікаційної роботи з вирішення комплексу проблем захисту та своєчасного усунення кіберзагроз серверної інфраструктури є сучасною актуальною складовою забезпечення безпеки держави та суспільства.

Об'єкт дослідження - серверна інфраструктура.

Предмет дослідження - методологія STRIDE

Мета роботи - розробити технологію забезпечення захисту серверів за методологією STRIDE та рекомендації щодо її реалізації.

Наукові завдання

- дослідити сутність проблеми захисту серверних систем та практичні підходи до їх захисту;

- провести аналіз: методів та засобів забезпечення захисту серверів; найбільш критичних аспектів забезпечення безпеки серверів та визначити напрями удосконалення;

- дослідити побудову моделі загроз серверної системи за методологією STRIDE;

- розробити технологію забезпечення захисту серверів на основі методології STRIDE.

Практичне значення одержаних результатів: проведено оцінку ефективності запропонованої технології забезпечення захисту серверів основі методології STRIDE та рекомендації фахівцям з кібербезпеки її практичного застосування.

Методи дослідження – опрацювання науково-технічної літератури за темою роботи, аналіз міжнародних стандартів та їх порівняння, національних галузевих вимог, Закони України.

Результати кваліфікаційної роботи апробовані на Всеукраїнській науко-практичній конференції «Актуальні проблеми кібербезпеки», яка відбулася 29 жовтня в Державному університеті інформаційно-комунікаційних технологій, м. Київ.

1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ЗАХИСТУ СЕРВЕРІВ

1.1. Теоретичні основи та сутність проблеми захисту серверних систем

Серверні системи складають ядро сучасної цифрової інфраструктури організацій усіх рівнів - від малих підприємств до великих корпорацій, державних установ та критичної інфраструктури. На серверах зосереджено обчислювальні ресурси, зберігаються базові дані, реалізуються ключові бізнес-процеси, здійснюється обробка інформації, забезпечується доступ користувачів до сервісів.

Як зазначає Ross Anderson у фундаментальній роботі «Security Engineering», переміщення даних із локальних серверів окремих організацій до великих хмарних дата-центрів призвело до того, що масштаб можливих інцидентів безпеки значно зріс: витоки даних, які раніше обмежувалися кількома сотнями записів в одній організації, тепер можуть охопити десятки мільйонів обліковців, кредитних карток або особистої інформації [13].

Теоретичні основи захисту серверних систем складаються з кількох взаємопов'язаних аспектів: інженерії безпеки (security engineering), моделювання загроз (threat modeling), математичної криптографії, теорії контролю доступу і управління ризиками. Розуміння цих основ критично важливе для розробки ефективних стратегій захисту, адже, як зазначає Anderson, багато систем функціонують неправильно або взагалі відмовляють саме тому, що їх розробники не врахували питання захисту з позиції потенційного зловмисника. Інженерія безпеки - це база побудови систем, здатних залишатися надійними в умовах зловмисних дій, помилок та випадковостей.

У найзагальнішому розумінні *сервер* - це обчислювальна система, яка надає сервіси (послуги) іншим комп'ютерам - клієнтам, через мережевий зв'язок. Історично

серверами були великі мейнфреймові комп'ютери, розташовані в захищених кімнатах із контрольованим фізичним доступом. Проте еволюція інформаційних технологій привела до різноманітних форм реалізації серверної функціональності:

• *фізичні виділені сервери* - традиційні комп'ютери або спеціалізовані пристрої, розташовані у локальних дата-центрах організації;

• *віртуальні сервери* - екземпляри операційних систем, виконані на гіпервізорах, які дозволяють розміщувати кілька віртуальних машин на одному фізичному обладнанні;

• *контейнеризовані сервіси* - мікросервіси, запаковані в контейнери (наприклад, Docker), які відокремлюють прикладне ПО від інфраструктури;

• *хмарні екземпляри* - серверні вузли, розміщені у хмарних дата-центрах провайдерів (AWS, Microsoft Azure, Google Cloud);

• *безсерверні функції* (Serverless) - функціональність, виконувана як послуга (AWS Lambda, Azure Functions), де розробник оперує лише кодом, а вся інфраструктура керується провайдером.

Незалежно від форми реалізації, всі типи серверів виконують три ключові функціональні напрями:

• *зберігання даних* - серверні системи утримують великі обсяги структурованої і неструктурованої інформації у реляційних та нереляційних базах даних, хмарних сховищах об'єктів, файлових системах, системах резервного копіювання. Ці дані включають персональну інформацію клієнтів, фінансові записи, медичні картки, комерційні секрети, інтелектуальну власність;

• *обробка даних* - сервери виконують обчислювальні операції різної складності: валідацію та санітизацію вхідних даних, бізнес-логіку застосунків, трансформацію даних, аналітичні розрахунки, машинне навчання, генерацію звітів. Порушення цілісності або конфіденційності на етапі обробки може мати катастрофічні наслідки;

• *надання інтерфейсів доступу* - сервери експортують свою функціональність через стандартизовані протоколи та інтерфейси: HTTP/HTTPS для веб-сервісів, REST/GraphQL для API, SQL для баз даних, SMTP/IMAP/POP3 для поштових сервісів, SSH для адміністрування, gRPC для мікросервісної взаємодії. Кожен із цих інтерфейсів є потенційною точкою входу для зловмисника.

Фундаментом теорії інформаційної безпеки, включаючи захист серверів, є класична **CIA-тріада** - триєдність основних цілей безпеки: конфіденційність, цілісність, доступність (рисунок 1.1).

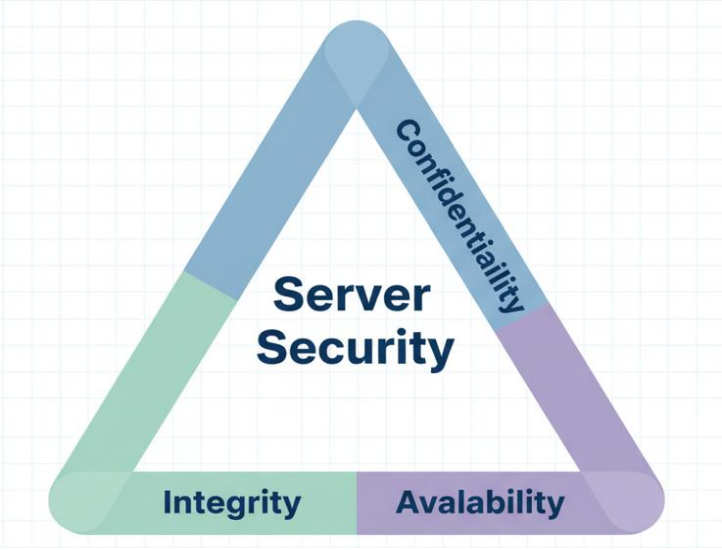

Рис. 1.1. CIA-тріада для безпеки серверів: конфіденційність, цілісність, доступність

Конфіденційність означає, що дані доступні лише авторизованим суб'єктам. Забезпечується шифруванням даних у спокої (full-disk encryption), при передачі (TLS 1.3, IPSec, SSH, VPN) та при обробці в пам'яті (ізоляція процесів, Intel SGX, AMD SEV).

Цілісність гарантує, що дані та код не були несанкціоновано змінені. Охоплює цілісність даних (БД, конфігурації, журнали аудиту), коду (відсутність backdoor,

malware) та конфігурації серверів. Забезпечується криптографічними хешами (SHA-256, SHA-3), цифровими підписами, file integrity monitoring та code signing.

Доступність забезпечує доступ легітимних користувачів до сервісів у потрібний час. Загрози: фізичні (відключення живлення, збої обладнання), мережеві (DDoS-атаки, які виснажують ресурси), логічні (crash, memory exhaustion, deadlock) та людський фактор (помилкова конфігурація). Забезпечується резервним копіюванням, георедундантністю, disaster recovery, балансуванням навантаження та auto-scaling.

На доповнення до класичної CIA-тріади, сучасна теорія безпеки виділяє ще три критичні аспекти: *автентичність* (Authenticity) - можливість достовірно перевірити, що особа чи компонент, який звертається до сервера, справді є тим, за кого себе видає. Це досягається механізмами аутентифікації: паролі, цифрові сертифікати X.509, токени (JWT, OAuth), багатофакторна аутентифікація (MFA), біометрія; *неспростовність* (Non-repudiation) - неможливість для суб'єкта заперечити, що він виконав певну дію, задокументовану на сервері. Для цього необхідно журнали аудиту з мітками часу, цифрових підписів транзакцій, захищеного зберігання логів; *авторизація* (Authorization) - контроль, які дії можуть виконувати аутентифіковані користувачі й процеси. Це реалізується через системи контролю доступу (RBAC - Role-Based Access Control, ABAC - Attribute-Based Access Control, PBAC - Policy-Based Access Control).

Центральна роль серверів у сучасних організаціях - це пріоритетна мішень для різних категорій зловмисників. Anderson визначає основні категорії загроз серверам[13]:

Спецслужби і *держави*. NSA, GCHQ, APT28, APT29, Sandworm проводять APT-кампанії з 0-day exploits на критичну інфраструктуру; Snowden revelations продемонстрували масштаб державного кібершпигунства [21].

Організована кіберзлочинність. Ботнети, ransomware-as-a-service, Initial Access Brokers; массові атаки на фінансові установи та е-комерцію[21].

Конкуренти і бізнес-розвідка. Економічний шпіонаж; спроби отримати комерційні секрети та плани розробки.

Внутрішні зловмисники (Insiders). Персонал з легітимним доступом; 0.8% від всіх зафіксованих векторів атак

.

Хактивісти. Ідеологічно мотивовані групи (79.4% інцидентів в ЄС); DDoS-кампанії; NoName057(16) 60% DDoS-заяв.

Дослідники та етичні хакери. Responsible disclosure; виявлення вразливостей для покращення безпеки.

• *ransomware* залишається найбільш руйнівною загрозою - попри зниження на 11% порівняно з 2024 роком - це зловмисні ПЗ основний інструмент кіберзлочинності. Штами Akira та SafePay були серед найбільш поширених, з окремими інцидентами, що призводили до порушення роботи сервісів. Initial Access Brokers продовжують торгувати низькоефективним, високооб'ємним доступом через VPN та RDP;

• *фішинг* домінує як початковий вектор атаки - за даними ENISA, фішинг (включаючи malspam, vishing, malvertising) становить близько 60% усіх ідентифікованих початкових векторів зараження. Експлуатація вразливостей становить 21.3% початкових векторів доступу, причому 68% з них призводять до розгортання шкідливого ПЗ;

• *державні кібероопераці проти ЄС* продовжуються - кібершпигунські активності державних акторів особливо спрямовані на сектор публічної адміністрації. Foreign Information Manipulation and Interference (FIMI) все частіше націлені на аудиторії ЄС;

• *хактивізм* домінує за обсягом інцидентів - переважно через низькоімпактні DDoS-кампанії проти веб-сайтів організацій країн-членів ЄС. 79.4% інцидентів є ідеологічно мотивованими;

• *топ-5 цільових секторів в ЄС* включають: публічну адміністрацію - 38.2% (домінують низькоімпактні DDoS на 94.8%, ransomware особливо впливає на муніципалітети); транспорт - 7.5% (авіація, логістика, морський сектор); фінанси - 4.8%; енергетику - 4.5%; цифрову інфраструктуру та сервіси - 2.2% (залишаються високоцінними цілями як стартові майданчики для подальших атак);

AI використовується для оптимізації атак - державні актори та кіберзлочинці все частіше використовують штучний інтелект для підвищення продуктивності та оптимізації своїх зловмисних дій.

Отже, розв'язання актуальної проблеми захисту серверів потребує комплексного підходу, який поєднує теоретично обґрунтовані моделі, такі як моделювання загроз з практичними механізмами безпеки та постійним урахуванням еволюції дестабілізуючій дії спектру, описаного, зокрема, в OWASP Top 10 та ENISA Threat Landscape [20].

1.2. Огляд наукових і практичних підходів до захисту серверів

Захист серверів є однією з найбільш критичних складових сучасної інформаційної безпеки, оскільки сервери часто зберігають та обробляють конфіденційну інформацію організацій, забезпечуючи доступ для внутрішніх та зовнішніх користувачів. Сучасні наукові та практичні підходи до захисту серверів охоплюють широкий спектр методологій, міжнародних стандартів, архітектурних принципів та технічних рішень, які постійно еволюціонують у відповідь на нові та дедалі складніші загрози кіберпростору. Це неперервне еволюціонування обумовлено тим фактом, що зловмисники постійно розробляють нові технології та методи для обходу існуючих механізмів захисту.

Моделювання загроз є одним із найбільш значущих наукових підходів до забезпечення безпеки серверних систем з самого початку їх проектування. За

визначенням Uzunov та Fernandez (2014), *моделювання загроз* - це процес, який використовується для аналізу потенційних атак або загроз і може підтримуватися бібліотеками загроз або таксономіями атак [9]. Цей підхід дозволяє системним адміністраторам, розробникам та архітекторам ідентифікувати слабкі місця серверної інфраструктури ще на етапах проектування та розробки, коли внесення змін є найменш витратним. Систематичний огляд літератури, проведений Xiong та Lagerström (2019), продемонстрував, що більшість методів моделювання загроз (27 з 29 досліджених статей) залишаються ручними процесами, що є часозатратними та схильними до помилок людини [9]. Проте спостерігається тенденція до автоматизації цього процесу через використання спеціалізованих інструментів, таких як Microsoft SDL Threat Modeling Tool, які дозволяють проводити детальний аналіз загроз системам через діаграми потоків даних (Data Flow Diagrams, DFD).

Практичний інструмент моделювання загроз - методологія STRIDE, розроблена корпорацією Microsoft в 1999 році і призначалася для обліку атак розробки Windows [19]. Незважаючи на солідний вік, актуальність вона не втратила,стала галузевим стандартом для категоризації та аналізу загроз безпеки. Модель STRIDE класифікує всі можливі загрози за шести основними категоріями, які охоплюють різні аспекти безпеки (Рис.1.2).

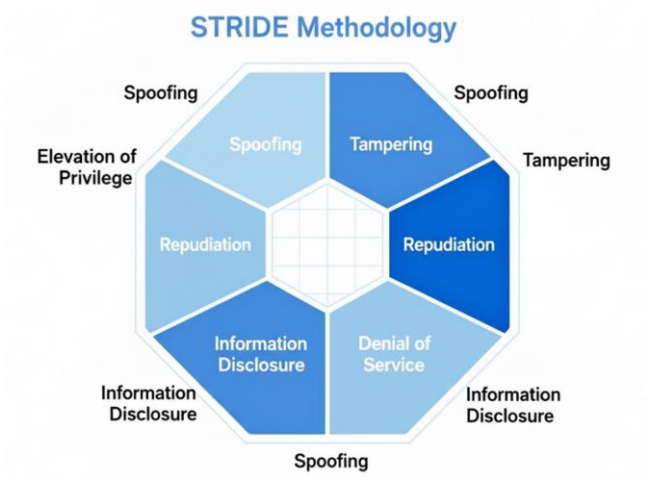

Рис. 1.2. STRIDE модель: шість категорій загроз безпеки систем

• Spoofing - видавання себе за іншого користувача.

• Tampering - несанкціонована модифікація даних.

• Repudiation - заперечення виконаних дій.

• Information Disclosure - витік конфіденційної інформації.

• Denial of Service - порушення доступності сервісів.

• Elevation of Privilege - отримання вищих прав доступу [18].

Для практичного застосування методології STRIDE проводиться аналіз безпеки серверів та створюється спрощена модель архітектури системи (рисунок 1.3).

.

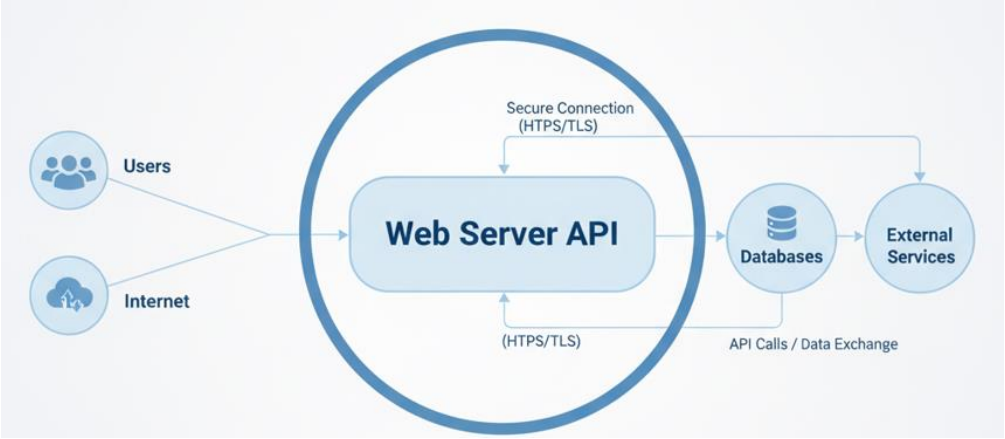

Рис. 1.3. Спрощена модель сервера для моделювання загроз з методологією STRIDE

Ця модель показує взаємодію користувачів та зовнішніх систем з серверною інфраструктурою, включаючи веб-сервер, API, базу даних та зовнішні сервіси. Пунктирна лінія «Trust boundary» розділяє надійну внутрішню частину системи від потенційно небезпечного зовнішнього світу. Така архітектура дозволяє систематично застосовувати кожну категорію STRIDE для кожного потоку даних, з'єднань та компонентів, виявляючи потенційні вразливості. Якісна модель загроз допомагає

зробити продукт безпечнішим та забезпечує відповідність міжнародним стандартам. Міжнародні стандарти безпеки встановлюють фундаментальні рамки для захисту серверної інфраструктури та управління ризиками інформаційної безпеки

NIST SP 800-123 "Guide to General Server Security" є базовим документом федерального рівня США, що визначає комплексні принципи та практичні рекомендації для захисту серверів, призначених для забезпечення послуг державного і приватного секторів [5]. Згідно з цим стандартом, процес забезпечення безпеки сервера включає послідовність критичних етапів: планування безпеки, під час якого визначаються цілі безпеки, ролі персоналу та вимоги до системи в контексті організаційної політики безпеки; захист операційної системи через патчинг усіх відомих вразливостей, зміцнення конфігурації за рахунок вилучення непотрібних сервісів та налаштування механізмів автентифікації; захист серверного програмного забезпечення через безпечну інсталяцію, конфігурацію та управління дозволами доступу; постійна підтримка безпеки через систематичний моніторинг, регулярне резервне копіювання та періодичне проведення тестування безпеки.

Фундаментальні принципи безпеки серверів, встановлені NIST, спрямовані на створення стійких до атак систем [12]:

• *простоти* (Simplicity) наголошує на тому, що складність архітектур та конфігурацій часто є джерелом спектру (множинних) проблем безпеки, які важко виявити та виправити;

• *безпечного збою* (Fail-Safe) вимагає, щоб система при виникненні будь-якого збою зберігала захищений стан, охороняючи критично важливі дані та сервіси від несанкціонованого доступу;

• *повного посередництва* (Complete Mediation) передбачає, що всі запити до ресурсів мають проходити через механізми контролю доступу, без пошуків альтернативних шляхів обходу цих контролів;

• *відкритого дизайну* (Open Design) констатує, що безпека не повинна залежати від конфіденційності реалізації та внутрішніх деталей систем безпеки, оскільки такий підхід (security through obscurity) неминуче зазнає невдачі при наявності мотивованого зловмисника;

• *розділення привілеїв* (Separation of Privilege) вимагає розділення функцій та ролей, щоб жодна особа або процес не мав надмірних можливостей. Найвпливовіший із всіх принципів - *найменші привілеї* (Least Privilege) - гарантує, що кожен процес, користувач або система отримує лише мінімально необхідні права для виконання своїх функцій, що значно обмежує потенційну шкоду від скомпрометованого облікового запису або програми.

Набір перевірених практик кібербезпеки **CIS Controls** розроблений Center for Internet Security та представляють 18 критичних елементів управління безпекою, які формують дорожну карту для захисту організацій від найпоширеніших сучасних та еволюціонуючих кіберзагроз [8]. Серед найбільш релевантних для захисту серверів є *Control* 4 "Secure Configuration of Assets", який вимагає розробки, впровадження та постійного оновлення зміцнених конфігурацій для операційних систем, додатків, мережевих пристроїв та інших IT-активів. *Control* 5 фокусується на управлінні дозволами доступу, обмежуючи права користувачів і програм на основі принципу найменших привілеїв. *Control* 6 виконує керування ресурсами та видалення непотрібних функцій, а *Control* 3 забезпечує керування припиненням облікових записів та видалення прав доступу відключених користувачів. Організації, що впроваджують набір перевірених практик кібербезпеки CIS Controls, отримують стандартизований та науково обґрунтований підхід до захисту, який також помітно допомагає відповідати вимогам нормативних регуляторів та міжнародних стандартів [8].

Стандарт **ISO/IEC 27001:2022** (оновлена версія) набуває ключового значення завдяки актуальності з питань управління ризиками, встановлює комплексну систему

управління інформаційною безпекою (СУІБД) з набором 93 контролів безпеки, організованих у чотири основні категорії: організаційні, персональні, фізичні та технологічні. *Організаційні* контролі встановлюють стратегію та політики безпеки на рівні управління організацією, включаючи розробку документованої політики інформаційної безпеки, визначення ролей та відповідальностей, управління ризиками та планування дій при реагуванні на інциденти. *Персональні* контролі зосереджуються на залученні, навчанні та контролі персоналу, забезпечуючи, що всі співробітники розуміють свої обов'язки щодо безпеки. *Фізичні* контролі захищають інформаційні активи від фізичного доступу, крадіжки та пошкодження через замки, системи контролю доступу та спостереження. *Технологічні* контролі включають механізми шифрування, контроль доступу на основі ідентичності, логування та моніторинг активності, а також засоби для визначення та виправлення вразливостей.

Концепція **Defense in Depth** (Глибокий захист) є фундаментальною архітектурною парадигмою сучасної кібербезпеки, що передбачає впровадження багаторівневого, перекриваючого захисту для більш ефективного стримування, виявлення та затримки кіберзагроз на різних рівнях мережевої та системної архітектури (рисунок1.4) [13].

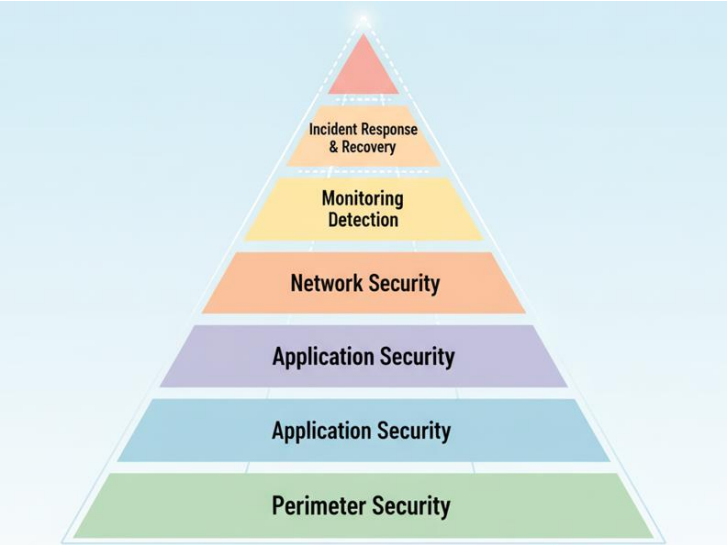

Рис. 1.4. Defense in Depth: багаторівнева архітектура захисту серверів

На відміну від традиційного підходу "castle and moat", який покладається на одну сильну лінію оборони на периметрі, глибокий захист передбачає розміщення механізмів безпеки на кожному рівні системи, так що компрометація однієї лінії захисту не призводить до повної втрати безпеки.

Рівень 1 - *периметровий захист* формує першу лінію захисту організації, розміщується на межі між внутрішньою корпоративною мережею та зовнішнім інтернетом. Традиційні брандмауери (firewalls) здійснюють контроль трафіку, фільтрують вхідний та вихідний трафік на основі попередньо визначених правил безпеки та дозволяють проходити лише легітимному трафіку, що відповідає встановленим критеріям. Системи запобігання вторгненням (IPS - Intrusion Prevention Systems) виходять далі за межі пасивного контролю, активно аналізують та блокують шкідливий трафік у реальному часі, виявляють сигнатури відомих атак та аномальну поведінку мережі, що потенційно вказує на нові загрози.

Рівень 2 - *Захист кінцевих точок (Endpoint Protection)* зосереджується на захисті окремих комп'ютерів, серверів та мобільних пристроїв. Цей рівень включає сучасне антивірусне програмне забезпечення та засоби, орієнтовані на виявлення та реагування на інциденти (Endpoint Detection and Response, EDR), які застосовують машинне навчання для виявлення аномальної поведінки програм та процесів, навіть якщо це нові, раніше невідомі загрози. Управління патчами на цьому рівні полягає в систематичному та своєчасному застосуванні оновлень для операційних систем та всіх встановлених програм, закриваючи відомі вразливості, які зловмисники могли б експлуатувати.

Рівень 3 - *захист додатків (Application Protection)* орієнтований на захист веб-додатків та сервісів від специфічних атак на рівні додатків. Web Application Firewalls (WAF) аналізують HTTP-запити та запобігають атакам типу SQL-ін'єкції, яка дозволяє зловмисникам маніпулювати запитами до баз даних, Cross-Site Scripting (XSS), який

вставляє шкідливий код у веб-сторінки, та розподілені атаки типу "отримання послуг" (DDoS), які намагаються перевантажити веб-сервер запитами.

Рівень 4 - *Захист даних (Data Protection)* гарантує конфіденційність та цілісність інформації незалежно від їх стану. Шифрування даних у стані спокою на дисках та в сховищах даних, а також шифрування при передачі через мережу лишають дані непридатними для використання у разі їх перехоплення. Системи запобігання витоку даних (DLP - Data Loss Prevention) моніторять передачу конфіденційної інформації та запобігають несанкціонованому виведенню критичних даних за межі організації.

Рівень 5 - *Мережева безпека (Network Security)* включає сегментацію мережі та практику мікросегментації, яка розділяє мережу на невеликі, ізольовані зони, що виконують певні функції. При такому підході, якщо один сегмент мережі скомпрометований, зловмисник мав би знову долати механізми безпеки для проникнення в інші сегменти, обмежуючи його можливості для латерального руху через мережу організації.

Рівень 6 - *Моніторинг та аудит (Monitoring and Auditing)* забезпечує постійну видимість активності в системі та мережі. Системи управління інформацією про безпеку та подіями (SIEM - Security Information and Event Management) забезпечують централізований збір логів з усіх систем, надають кореляцію подій для виявлення складних атак та дозволяють проводити як негайне реагування, так і подальше розслідування інцидентів безпеки.

Рівень 7 - *Інцидент-реагування та відновлення (Incident Response and Recovery)* складається з заздалегідь розроблених планів та процедур, які дозволяють організації швидко виявити, обмежити та виправити наслідки успішних атак, мінімізуючи час зупинки систем та втрату даних.

Дослідження демонструють, що організації, які впроваджують багаторівневий захист, зменшують успішність масштабних атак на 87%, що доводить ефективність цього підходу [15].

Методологія **OWASP Top 10** є визнана в усьому світі, довідник найбільш критичних ризиків безпеки веб-додатків і веб-сервісів, оновлюючись кожні кілька років на основі даних про дійсні атаки та вразливості, виявлені у виробництві. На рисунку 1.5 представлено OWASP Top 10 2021 - найбільш критичні ризики безпеки веб-додатків та серверів [20].

Рис. 1.5. OWASP Top 10 2021: найбільш критичні ризики безпеки веб-додатків та серверів

Версія 2021 року виявила, що *Broken Access Control* (порушення контролю доступу) посідає перше місце в рейтингу, оскільки це питання було виявлене у 94% протестованих додатків. Ця категорія включає такі проблеми як несанкціонований доступ до облікових записів інших користувачів, доступ до адміністративних панелей без належних прав, можливість перегляду або модифікації конфіденційних даних інших користувачів та навіть доступ до функцій, недоступних для звичайних користувачів.

Cryptographic Failures (криптографічні збої), яка займає друге місце, охоплює проблеми з неправильним або повністю відсутнім шифруванням, що призводить до

розкриття чутливої інформації, такої як медичні дані, фінансові записи чи персональні дані під час передачі або зберігання.

Injection (ін'єкція) залишається однією з найнебезпечніших вразливостей, включаючи SQL-ін'єкції, які дозволяють зловмисникам маніпулювати запитами до баз даних, NoSQL-ін'єкції для баз даних документів, та ін'єкції операційної системи для виконання довільних команд на сервері.

Insecure Design (небезпечне проектування) є відносно новою категорією, що фокусується на необхідності розгляду безпеки з самого початку життєвого циклу розробки, а не введення механізмів безпеки після події.

Security Misconfiguration (невдала конфігурація безпеки) відображає поширеність неправильно налаштованих серверів з надмірними дозволами, застарілими налаштуваннями за замовчуванням, встановленими непотрібними сервісами та не використовуваними функціями. Це дозволяє зловмисникам експлуатувати самі видимі та легко доступні вразливості, часто не потребуючи розвинених технічних навичок.

Практичні підходи до значного *посилення безпеки серверів* (Server Hardening) передбачають застосування строгих конфігурацій безпеки для систематичного зменшення поверхні атаки та кількості потенційних вразливостей, які можуть бути експлуатовані зловмисниками. Ці дії здійснюються як на етапі первісної конфігурації, так і на постійній основі протягом всього життєвого циклу сервера.

Управління обліковими записами: вимкнення гостьових облікових записів, перейменування/вимкнення стандартних адміністративних облікових записів (ім'я: admin, root), політика надійних паролів (мін. 15 символів).

Конфігурація служб: видалення непотрібних сервісів та протоколів (Telnet, SMBv1), мінімальна конфігурація ОС.

Мережна конфігурація: активація вбудованого брандмауера ОС, явний дозвіл необхідних портів/протоколів, ізоляція VLAN.

Управління патчами: регулярні оновлення ОС та ПО, пріоритизація за CVSS, тестування на тестовій копії перед розгортанням.

Безпечна розробка програмного забезпечення (рисунок 1.6) встановлює фундаментальні принципи, які повинні застосовуватися при розробці як операційних систем, так і серверних додатків.

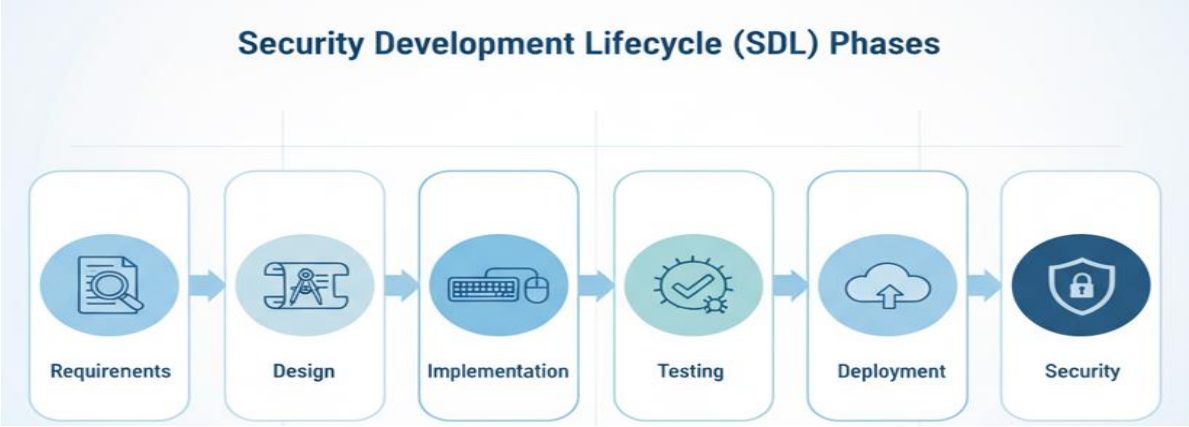

Рис.1.6. Security Development Lifecycle: етапи безпечної розробки серверного ПО

Методологія *SDL (Security Development Lifecycle)* розроблена Microsoft та визнана як стандарт в індустрії, дозволяє знизити ймовірність виникнення вразливостей, максимально ускладнити їхню експлуатацію та прискорити виправлення. Все це дозволяє зробити безпечнішим на кожному етапі розробки продукту, від планування до розгортання та управління життєциклом [17]. Ключовими фазами SDL є визначення вимог безпеки на початку проекту, проведення аналізу загроз через моделювання загроз, здійснення безпечної розробки коду з строгим дотриманням принципів безпечного кодування, проведення комплексного тестування безпеки на кількох етапах розробки, забезпечення безпечного розгортання та управління після розгортання.

Модель *SD3* (Secure by Design, by Default, and in Deployment) охоплює весь процес розробки, надаючи пріоритет безпеці від початкового етапу проектування до розгортання та обслуговування, встановлює три ключові аспекти розробки безпечного програмного забезпечення:

• *Secure by Design* означає, що безпека має враховуватися з самого початку проектування системи, від визначення архітектури до вибору алгоритмів і протоколів.

• *Secure by Default* передбачає, що за замовчуванням активовані лише найнеобхідніші функції, всі потенційно небезпечні функції вимкнені або потребують явної активації адміністратором.

• *Secure in Deployment* гарантує, що організації та адміністратори мають всі необхідні інструменти та рекомендації для безпечного розгортання та підтримки системи у виробничому середовищі.

Особливу увагу дослідники та розробники приділяють захисту від *переповнення буфера* (Buffer Overflow), яке залишається однією з найпоширеніших та найнебезпечніших вразливостей у системному коді [12]. Переповнення буфера дозволяє зловмисникам писати дані поза межами виділеної пам'яті, що може привести до виконання довільного коду, отримання контролю над всією системою, крадіжки конфіденційної інформації або відмови в обслуговуванні. Використання безпечних функцій обробки рядків (наприклад, strsafe.h у Windows), які перевіряють межі буфера перед кожною операцією запису, та увімкнення опцій компілятора/GS у Visual C++, яка вставляє перевірки цілісності стека під час виконання, помітно зменшує ризик успішної експлуатації таких вразливостей.

Системи виявлення та запобігання вторгненням - критичні компоненти моніторингу сучасних серверних архітектур.

IDS (Intrusion Detection Systems) - це пасивна система моніторингу, яка аналізує мережевий трафік та журнали системи в пошуках сигнатур відомих атак та аномальної поведінки, генеруючи сповіщення при виявленні потенційних загроз. IDS порівнює

мережевий трафік та пакети з базою даних відомих кіберзагроз та позначає підозрілі пакети для розслідування безпеки, але не розриває з'єднання та не зупиняє трафік [6].

IPS (Intrusion Prevention Systems) є активним механізмом безпеки, що розміщується безпосередньо в потоці мережевого трафіку між внутрішньою корпоративною мережею та зовнішнім Інтернетом, діючи як фільтр у реальному часі. На відміну від пасивного IDS, IPS може активно блокувати шкідливий трафік, розривати з'єднання з підозрілими хостами, переконфігуровувати правила брандмауера в процесі та навіть ізолювати уражені сегменти мережі без втручання людини. Сучасні рішення часто об'єднують обидві технології в гібридні системи IDPS (Intrusion Detection and Prevention Systems), що забезпечують як виявлення, так і автоматичне реагування на загрози, надаючи організаціям комплексну видимість та контроль над мережевою активністю.

Сучасні підходи до захисту серверів базуються на комплексному, багаторівневому поєднанні теоретичних методологій, визнаних в усьому світі галузевих стандартів, архітектурних принципів та практичних технічних засобів.

1.3. Аналіз термінології та формування понятійно-категоріального апарату у сфері безпеки серверів

Формування коректного та узгодженого понятійно-категоріального апарату є критичною передумовою для наукового дослідження проблеми безпеки серверів, оскільки дозволяє уникнути неоднозначностей, двозначних інтерпретацій та непорозумінь при описі загроз, контролів, процесів захисту та управління ризиками. У цій семантичній системі перетинаються два основні рівні:

• інженерно-технічна площина моделювання загроз та аналізу атак, де домінують терміни, що безпосередньо описують способи нападу та вектори компрометації серверів;

• управлінсько-нормативна площина систем управління інформаційною безпекою, де ключову роль відіграє міжнародний стандарт ISO/IEC 27001:2022 та пов'язані з ним нормативні документи, які задають методологічні та процедурні рамки для побудови комплексної системи захисту.

Методологія STRIDE, розроблена корпорацією Microsoft як невід'ємна складова Security Development Lifecycle (SDL), представляє один із найбільш значущих інженерно-технічних підходів до категоризації та опису загроз у контексті розробки та експлуатації програмного забезпечення, включаючи серверні додатки. STRIDE виступає не лише технічною класифікацією типів атак і вразливостей, але й фактично задає базовий семантичний каркас для опису та обговорення властивостей, які повинні бути захищені у серверних системах з позиції розробника та архітектора. Це властивості: автентичності суб'єкта, цілісності даних, неспростовності дій, конфіденційності інформації, доступності сервісів та контрольованості привілеїв доступу. Кожна з цих властивостей має прямий аналог в управлінських термінах ISO/IEC 27001:2022, що дозволяє встановити «міст» між інженерною та управлінською площинами (рисунок 1.7).

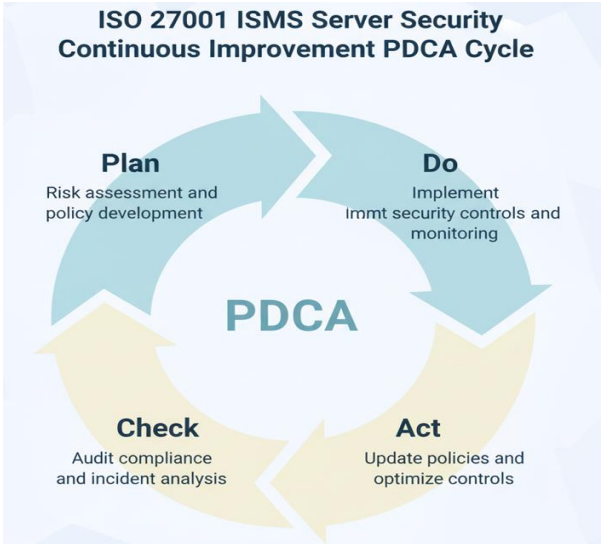

Рис. 1.7. PDCA-цикл ISO/IEC 27001 для управління безпекою серверів

ISO/IEC 27001:2022 як міжнародний стандарт систем управління інформаційною безпекою встановлює комплексний, організаційно-орієнтований понятійний апарат, який охоплює не лише технічні механізми безпеки, але й управлінські процеси, людські фактори та організаційну культуру безпеки.

На найвищому рівні стандарт визначає три базові *властивості інформації та інформаційних активів*, які повинні бути захищені та дотримуватися: *конфіденційності*, *цілісності*, *доступності*.

Стандарт ISO/IEC 27001:2022 також формалізує ключовий набір *термінів управління ризиками*, які мають суттєве значення для аналізу та планування безпеки серверів (рисунок 1.8):

Рис.1 8. Взаємозв'язок активів, загроз, уразливостей та ризиків в управлінні безпекою

Актив (Asset) визначається як «що-небудь, що має цінність для організації», це включає як фізичні активи (серверне обладнання, комунікаційне устаткування, дата-центри), так і нефізичні активи (дані, програмне забезпечення, інтелектуальна власність, репутація, люди та їхні знання).

Загроза (Threat) в ISO/IEC 27001:2022 формально визначається як «потенційна причина небажаного інциденту безпеки», це може бути або природного походження (природні катастрофи, стихійні лиха), або результатом людської діяльності (дії кіберзлочинців, помилки персоналу, навмисні вразливості).

Уразливість (Vulnerability) означає «слабкість в активі або групі активів, які можуть бути використані однією або більше загрозами». У контексті серверів уразливості можуть існувати на різних рівнях - в апаратному забезпеченні (вічні недоліки чипів, які не можуть бути виправлені оновленнями), в операційній системі та ядрі (баги у коді ОС), в прикладному програмному забезпеченні (вразливості у веб-сервісах), в мережевій конфігурації (неправильно налаштовані брандмауери) та в організаційних процесах (відсутність навчання персоналу).

Ризик (Risk) визначається як «вплив невизначеності на цілі», у практиці управління безпекою ризик розраховується як комбінація ймовірності того, що загроза експлуатує уразливість, та наслідків (впливу) реалізації такої загрози для організації.

Контроль (Control) у термінах ISO/IEC 27001:2022 визначається як «міра, яка модифікує ризик», фактично контроль можна розуміти як будь-який захід, який зменшує, усуває, передає або приймає ризик. Для серверів контролі можуть бути організаційними (політики та процедури), технічними (криптографія, брандмауери, IDS/IPS), фізичними (замки на дверях дата-центрів, системи контролю доступу) або процедурними (плани реагування на інциденти).

Система управління інформаційною безпекою (SUIB, ISMS) в термінах ISO/IEC 27001:2022 визначається як «сукупність взаємопов'язаних елементів для встановлення політики і цілей безпеки та досягнення цих цілей». ISMS включає людей, процеси, технологію та організаційну структуру, які, працюючи в координації, забезпечують планування, впровадження, моніторинг, огляд і вдосконалення заходів інформаційної

безпеки. ISMS передбачає також постійне вдосконалення через цикл PDCA (Plan - Планування, Do - Впровадження, Check - Перевірка, Act - Дія/Вдосконалення):[3]

Проведений аналіз понятійного апарату у сфері безпеки серверів через призму STRIDE та ISO/IEC 27001:2022, визначено прямі відповідності: кожна категорія загроз STRIDE проектується на певну властивість інформації, а кожна властивість, у свою чергу, реалізується через набір контролів, визначених у ISO/IEC 27001:2022.

Сучасна теорія та практика безпеки серверів базуються на розумінні і застосуванні цього інтегрованого понятійного апарату, де інженерна модель загроз надає деталізацію та специфіку щодо типів атак, тоді як управлінський фреймворк ISO/IEC 27001:2022 забезпечує систематичність, масштабованість і орієнтацію на досягнення організаційних цілей безпеки. Таке поєднання теоретично обґрунтованого підходу з практичною реалізацією дозволяє організаціям розробляти, впроваджувати та постійно удосконалювати ефективні системи захисту серверної інфраструктури.

Методологія STRIDE є потужним інструментом для моделювання загроз, допомагає виявляти потенційні вектори атак та розробляти ефективні механізми захисту. Однією з ключових переваг є структурований підхід, що дозволяє побудувати систему аналізу загроз [12, 14].

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ СЕРВЕРІВ

2.1. Еволюція підходів до захисту серверних систем та сучасні тенденції

Еволюція методів та засобів захисту серверних систем є відображенням постійної гонки між нападниками та захисниками, між технологічними можливостями і новими векторами атак, між локальними парадигмами безпеки й глобальними викликами кіберпростору. На кожному етапі еволюції (рисунок 2.1) домінуючі підходи, архітектурні парадигми та механізми безпеки були сформовані переважаючим типом загроз, технологіями кіберзагроз, а також розвитком міжнародних стандартів і нормативно-правової бази.

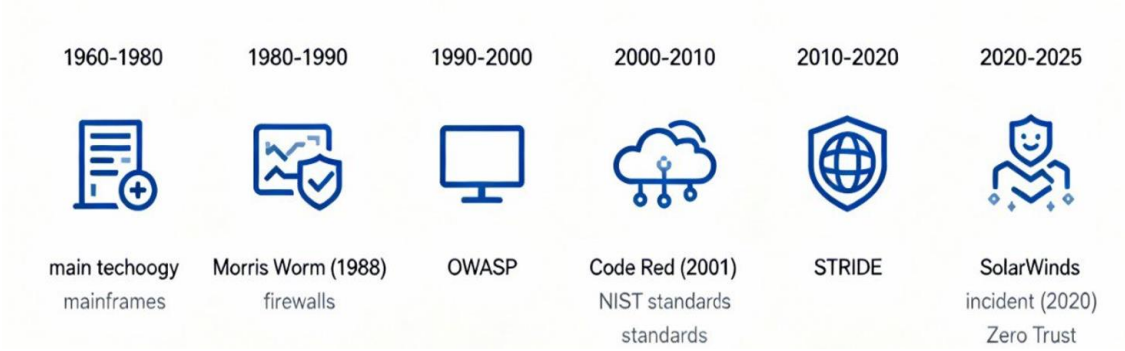

Рис.2.1. Еволюція підходів до захисту серверів (1960–2025)

Період ранніх мейнфреймів (1960–1980-ті роки) характеризувався перевагою фізичної безпеки як основного механізму захисту серверної інфраструктури. Сервери розташовувалися у спеціалізованих, «закритих» комп'ютерних залах із суворим контролем доступу, значно відрізняючись від теперішнього поняття глобально розподіленої серверної інфраструктури. Загрози розглядалися переважно як

внутрішні - від персоналу або підрядників, які мали легітимний фізичний доступ до залів. На цьому етапі розроблялись перші теоретичні моделі безпеки, зокрема модель Белла-Лападули для багаторівневого контролю доступу та фільтрації інформаційних потоків, які залишаються релевантними й сьогодні [13]. Ідентифікація користувачів здійснювалася через прості облікові записи і журнали фізичних відвідувань, криптографія була практично відсутня та використовувалася лише в критично важливих оборонних й державних системах.

З появою персональних комп'ютерів, локальних мереж та, особливо, з глобальним поширенням Інтернету в 1990-х роках, спектр загроз серверам радикально трансформувався. Сервери більше не були ізольованими острівцями, захищеними фізичними замками і охоронцями (рисунок 2.2). Вони стали вузлами глобальної мережі, потенційно вразливими для атак із будь-якої точки світу без попередження чи фізичної присутності нападника. Атака Morris Worm 1988 року стала культовою демонстрацією цієї нової реальності, доповідаючи про можливість мережевого розповсюдження шкідливого коду на швидкості, яка виходила за межі можливостей людського контролю.

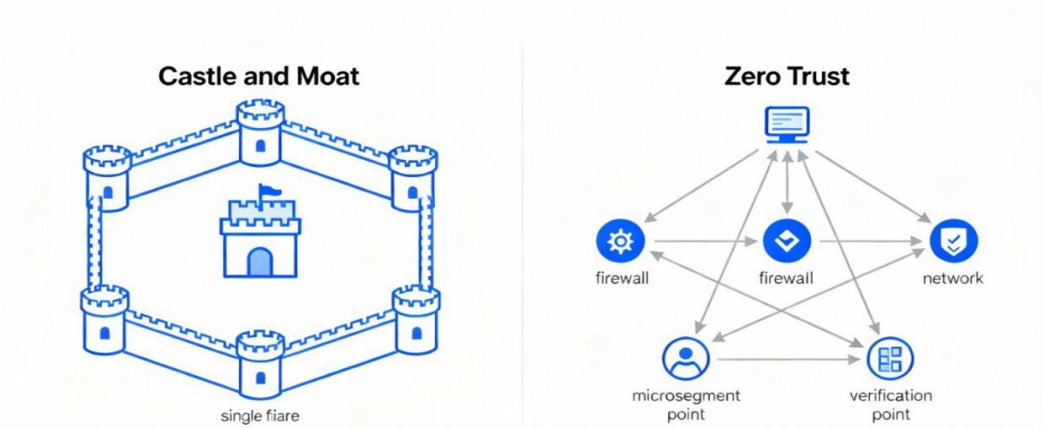

Рис. 2.2. Архітектурні парадигми безпеки: від "Castle and Moat" до Zero Trust

У відповідь на цей виклик виник новий клас технологій захисту: традиційні брандмауери фільтрували трафік на основі IP-адрес та портів, а системи виявлення вторгнень (IDS) аналізували мережевий трафік у пошуках сигнатур відомих атак. Криптографічні протоколи SSL/TLS та SSH замінили небезпечні канали передачі даних і адміністрування. Архітектурна парадигма "замку з ровом" ("Castle and Moat") домінувала в цей період, припускаючи, що міцний периметр, утворений потужним брандмауером, достатній для захисту внутрішніх серверів [5, 6, 14].

Період масового розповсюдження Інтернету і вебу- (1990–2000-ті роки) розкрив невідомі раніше вразливості, які витікали не з мережевої топології, а з логіки самих додатків. Масові мережеві черв'яки Code Red 2001 року й Nimda 2001 року продемонстрували катастрофічну швидкість експлуатації вебу-вразливостей. Code Red за кілька днів скомпрометував понад 359 000 серверів, експлуатуючи буферне переповнення в Microsoft Internet Information Server, і розпочав DDoS-атаку проти Білого дому. На цьому етапі з'явилися класичні вразливості типу SQL-ін'єкція, яка дозволяє маніпулювати запитами до баз даних, Cross-Site Scripting (XSS), що вставляє зловмисний JavaScript у веб-сторінки, та Buffer Overflow, яка експлуатує перевищення меж пам'яті. Потреба систематизувати та категоризувати ці вразливості привела до виникнення OWASP Top 10 у 2003 році, яка стала глобальним стандартом для розуміння вразливостей у веб-додатках [20]. Практики "зміцнення" серверів - видалення непотрібних служб, закриття невикористовуваних портів, застосування патчів - стали стандартною операційною процедурою. Концепція "захисту в глибину" (Defense in Depth) закріпилась як розуміння того, що жодна одна лінія оборони не достатня, і механізми безпеки повинні бути розташовані на кожному рівні архітектури, від периметра до хоста, від мережі до прикладного рівня.

Період стандартизації й ризик-орієнтованих підходів (2000 - 2010-ті роки) трансформував захист серверів із набору окремих інструментів на систему управління ризиками, побудовану на міжнародних стандартах. NIST, американська федеральна

служба стандартизації, почала видавати всеосяжні керівництва, включаючи NIST SP 800-53 "Security and Privacy Controls for Information Systems and Organizations" і NIST SP 800-123 "Guide to General Server Security", які систематизували вимоги до захисту операційних систем, управління ідентичністю та доступом, моніторингу й реагування на інциденти. На глобальному рівні міжнародна організація стандартизації розробила стандарт ISO/IEC 27001 "Information Security Management System", який описує процес управління інформаційною безпекою, у якому організація ідентифікує активи (включаючи сервери), виявляє загрози і уразливості, оцінює ризики, обирає набір контролів та впроваджує їх у циклі PDCA (Plan-Do-Check-Act). Center for Internet Security розробив практичний набір CIS Controls, який виділяє 18 критичних елементів управління, включаючи "Secure Configuration of Assets", "Access Control Management" і управління резервними копіями [3, 4, 5]. На цьому етапі сформувалась дисципліна управління уразливостями, яка включає регулярне сканування, класифікацію за критичністю з використанням CVSS (Common Vulnerability Scoring System), пріоритизацію та планування виправлень. Європейське агентство з кібербезпеки (ENISA) почало видавати річні звіти про ландшафт загроз, які документували типи й кількість інцидентів, цільові сектори і методи атак, дозволяючи організаціям планувати захист на основі даних про реальні загрози. Паралельно розпочалось масове переміщення серверної інфраструктури до хмарних платформ, починаючи з 2006 року, коли Amazon запустила AWS. Це створило нову парадигму безпеки: модель спільної відповідальності (Shared Responsibility Model), де провайдер хмари відповідає за безпеку платформи, а клієнт - за безпеку своїх віртуальних серверів; ризики міжорендаторної ізоляції, коли тисячі віртуальних серверів консолідуються на спільному обладнанні; нові механізми захисту API керування серверами.

Період моделювання загроз як системної практики (2010 - 2020-ті роки) переніс моделювання загроз з академічного дослідження в індустріальну практику. Microsoft

із виданням Security Development Lifecycle (SDL) популяризував модель STRIDE, яка категоризує загрози за шістьма типами, охоплюючи всі найважливіші властивості безпеки: Spoofing (підміна ідентичності), Tampering (фальсифікація даних), Repudiation (заперечення дій), Information Disclosure (розголошення інформації), Denial of Service (відмова в обслуговуванні) й Elevation of Privilege (підвищення привілеїв) [17, 19]. Систематичний огляд літератури, проведений Xiong та Lagerström 2019 року, показав, що більшість методів моделювання загроз залишаються ручними і графічними процесами, але спостерігається тенденція до автоматизації через інструменти типу Microsoft Threat Modeling Tool, які інтегруються в кінцеві конвеєри розробки [9]. Маючи на увазі STRIDE, на додачу розвивалися інші методи, такі як Attack Trees (діаграми дерева атак), PASTA (Process for Attack Simulation and Threat Analysis), OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation), кожен із яких мав свої переваги й недоліки, але всі вони розпочинали з логіки систематичного перебору потенційних загроз для кожного компонента системи. Компанії, такі як Microsoft, Google й Amazon, почали формалізувати процеси розробки ПЗ з урахуванням безпеки з самого початку (Secure by Design), замість додавання безпеки в кінці розробки. SDL передбачає визначення вимог безпеки на етапі планування, проведення аналізу загроз через моделювання загроз, кодування з урахуванням безпеки та тестування безпеки на кількох етапах. До кінця 2010-х років розвинулася культура DevSecOps, де механізми безпеки були інтегровані в конвеєри безперервної розробки і розгортання (CI/CD), включаючи сканування вихідного коду на вразливості, сканування образів контейнерів, автоматичне застосування політик безпеки на основі інфраструктури як коду.

Період хмарних, віртуалізованих і SDN/NFV середовищ (2015 - дотепер) характеризується докорінною трансформацією архітектури серверної інфраструктури, вимагаючи нових підходів до захисту. Software-Defined Networking (SDN) замінив традиційні фізичні мережеві пристрої програмно-керованими

контролерами (OpenFlow, OpenDaylight), що управляють маршрутизацією і політиками безпеки. Network Function Virtualization (NFV) перетворила мережеві функції, такі як брандмауери та маршрутизація, з апаратних пристроїв на програмне забезпечення.

Дослідження з монографій про безпеку SDN показали, що компрометація SDN контролера дає зловмиснику глобальний вплив на весь трафік мережі, потребуючи нових механізмів захисту контрольної площини - шифрування каналів до контролерів, верифікації конфігурацій, ізоляції функцій управління. Виявилось, що традиційна модель "замку з ровом" з периметровим брандмауером невдала в середовищі мікросервісів і хмарних систем, де сервіси часто розташовані у різних місцях. Розвинулась практика мікросегментації, де мережа розділена на невеликі ізольовані сегменти, між якими застосовуються суворі політики фільтрації ("east-west" безпека), і кожен сегмент має свої механізми захисту. Парадигма "довіряй, але перевіряй" замінилась парадигмою Zero Trust ("ніякої довіри за замовчуванням"), де жоден сервер, користувач чи пристрій не вважається довіреним автоматично, кожен запит до ресурсу потребує явної верифікації ідентичності, пристрою та контексту [18]. За даними ENISA Threat Landscape 2023 (рис.2.3).

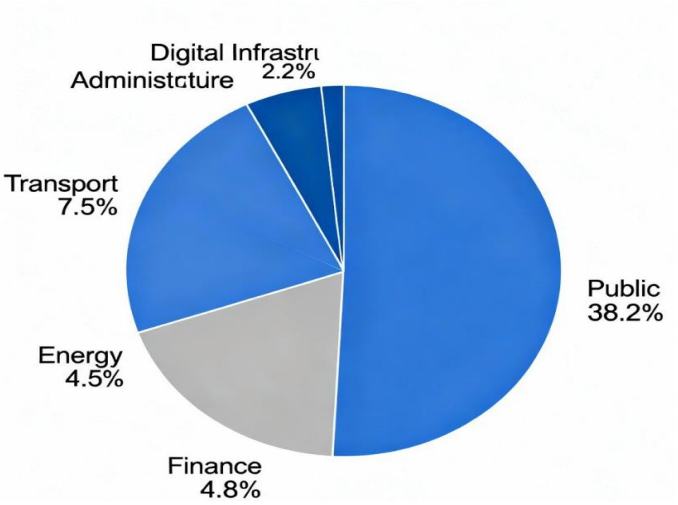

Рис.2.3. Статистика цільових секторів кібератак в ЄС (за даними ENISA 2023)

Хмарні віртуалізовані середовища стали однією з головних цілей для атак типу компрометація облікових даних адміністраторів, атаки на API управління, експлуатація вразливостей у гіпервізорах і контейнеризаційних платформах [21].

Сучасні тенденції (2020-ті роки) характеризуються новими й вишуканішими типами атак, які вимагають адаптації підходів захисту. Ransomware залишається найбільш руйнівною загрозою для серверів, як свідчить ENISA Threat Landscape 2023, характеризується експлуатацією вразливостей у точках входу, боковим рухом через мережу організації, шифруванням файлових серверів і серверів баз даних, вимаганням грошей за розшифрування. Паралельно спостерігається перехід від простого шифрування даних до data extortion - крадіжки даних з подальшою загрозою розкриття без необхідності шифрування, що робить атаки ще більш небезпечними для організацій із чутливими даними. Supply chain attacks на кшталт SolarWinds (2020) й Log4Shell (2021) продемонстрували, що нападники все частіше атакують не окремі організації, а постачальницькі ланцюги, з компрометацією одного постачальника ПЗ, які можуть вразити десятки тисяч організацій. Штучний інтелект використовується як нападниками для автоматизації фішингу, підбору паролів та пошуку вразливостей, так і захисниками для виявлення аномалій у поведінці серверів, виявлення нових типів атак й пріоритизації уразливостей. NIST Cybersecurity Framework 2.0 (лютий 2024) вводить концепцію Continuous Threat Exposure Management (CTEM) - безперервного управління експозиціями до загроз, яке передбачає постійне виявлення відкритих портів, зайвих служб, вразливих версій ПЗ, безперервну оцінку ризиків і автоматизоване видалення виявлених експозицій [8]. Framework також посилює акцент на Governance (GOVERN) на рівні вищого керівництва, управлінні ризиків від постачальників, адаптивних підходах до нових загроз, визнаючи, що сервери часто розташовані за межами контролю організації і потребують інтегрованого управління ризиками на рівні розширеної екосистеми.

Еволюція підходів до захисту серверів демонструє перехід від локальної, переважно фізичної безпеки до глобальної, розподіленої і багаторівневової системи управління ризиками, яка вимагає поєднання теоретично обґрунтованих моделей (STRIDE, моделювання загроз), практичних механізмів (криптографія, мікросегментація, Zero Trust), міжнародних стандартів (NIST, ISO/IEC 27001, CIS Controls) та постійної адаптації до еволюціонуючого спектру кіберзагроз.

2.2. Аналіз найбільш критичних аспектів забезпечення безпеки серверів

Критичні аспекти забезпечення безпеки серверів випливають із поєднання класичної моделі конфіденційності, цілісності та доступності (CIA), характеру сучасних загроз і типових слабких місць реальних систем. Узагальнення рекомендацій NIST SP 800-123 щодо загальної безпеки серверів, практик рівня CISSP та емпіричних даних ENISA і OWASP дозволяє виділити декілька груп проблем, які системно повторюються у більшості інцидентів і тому мають розглядатися як «найбільш критичні» для будь-якої серверної інфраструктури (рисунок 2.4) [5].

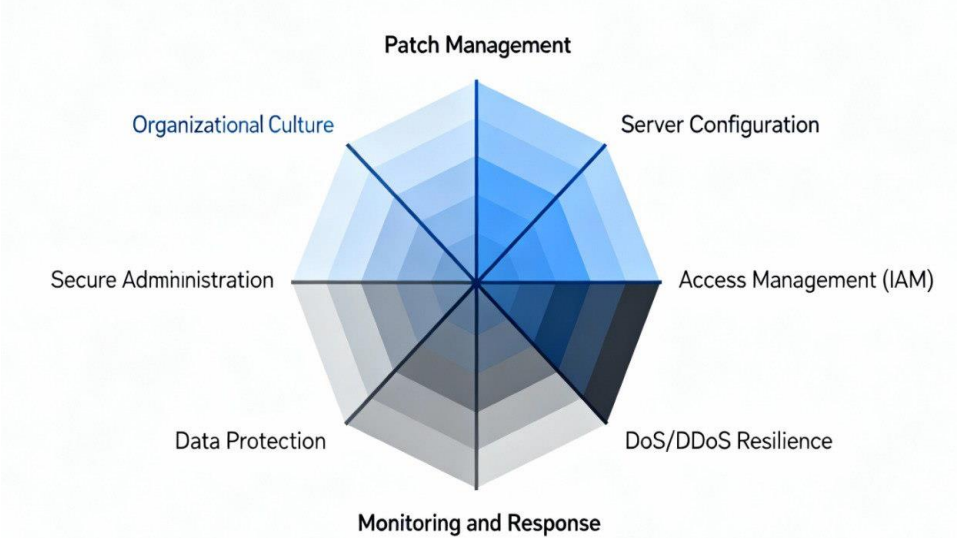

Рис.2.4. Блок критичних аспектів безпеки серверів

Перший і найбільш очевидний блок критичних аспектів пов'язаний із вразливостями програмного забезпечення сервера та операційної системи, а також з ефективністю процесів патч-менеджменту.

NIST прямо вказує, що більшість поширених атак проти серверів ґрунтується на експлуатації відомих дефектів ОС і серверних застосунків, які могли бути усунуті своєчасним застосуванням оновлень. У практиці це означає, що навіть формально «захищений» з точки зору архітектури сервер, залишений без регулярного оновлення, впродовж кількох місяців перетворюється на мішень для масового вторгнення. ENISA у своєму огляді CVE та CWE показує, що критичні вразливості дуже часто належать до класів переповнення буфера, некоректної роботи з пам'яттю, неконтрольованого завантаження файлів небезпечних типів, тобто таких дефектів, які виправляються виключно на рівні коду та оновлень. Тому здатність організації оперативно виявляти наявність відомих вразливостей (сканування, NVD, KEV-каталог CISA) і планово закривати їх патчами є одним із базових критичних чинників безпеки (рисунок 2.5).

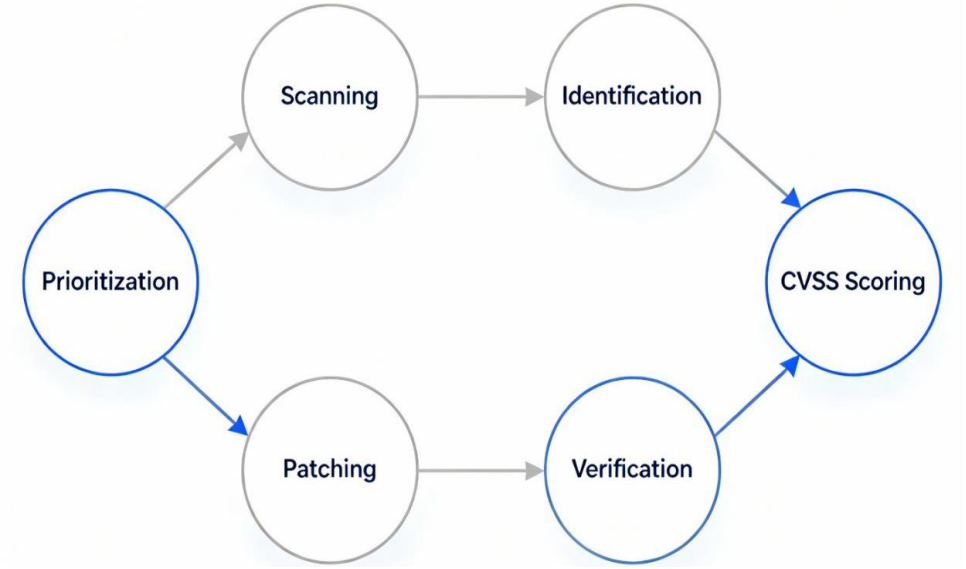

Рис. 2.5. Цикл управління вразливостями (Vulnerability Management Lifecycle)

Це особливо критично враховуючи, що за даними ENISA, більш ніж половина критичних CVE включаються до каталогу CISA Known Exploited Vulnerabilities (KEV), що означає їх активну експлуатацію нападниками, а часовий проміжок між публікацією вразливості та розробленням експлойта може становити лише кілька днів.

Другим системним аспектом, який постійно фігурує серед причин реальних інцидентів, виступають помилки конфігурації серверів і сервісів. NIST SP 800-123 підкреслює, що типовий «заводський» стан ОС і серверного ПЗ орієнтований на функціональність і простоту використання, а не на безпеку, і тому вимагає цілеспрямованого «зміцнення» (hardening). Практика OWASP показує, що систематичні помилки конфігурації (надлишкові служби, відсутність обмежень доступу, незахищені адміністративні інтерфейси, слабкі або дефолтні облікові записи) залишаються одним із найпоширеніших джерел компрометації серверів і веб-застосунків. Статистика ENISA щодо найкритичніших CWE демонструє, що такі слабкості, як некоректне обмеження шляхів до директорій (path traversal), неконтрольоване завантаження файлів небезпечних типів або відсутність аутентифікації для критичних функцій, систематично призводять до серйозних інцидентів. В кожному з цих випадків мова йде не про «дірки» у криптоалгоритмах, а про неправильні налаштування або недогляд при розгортанні системи. Практичні наслідки цього включають: не конкретний запуск надлишкових серверних модулів, залишені дефолтні облікові записи і паролі, непрефіксовані адміністративні інтерфейси (наприклад, незахищена консоль управління), директорії та файли з прав доступу «для всіх», а також недостатньо жорсткі обмеження на розмір завантажень, виконання скриптів у каталогах користувачів тощо. Отже, якість процесів стандартного базового конфігурування серверів, наявність і дотримання чек-листів, а також регулярний аудит конфігурацій є критичною умовою зниження ризику.

Третій фундаментальний аспект - управління ідентифікацією, автентифікацією та авторизацією на серверах. NIST наголошує, що правильне налаштування користувацьких облікових записів, політик паролів, груп доступу й ресурсних контролів є одним із ключових кроків початкового «зміцнення» операційної системи. CISSP-підхід доповнює це класичними принципами найменших привілеїв (least privilege) і розподілу обов'язків (separation of duties): кожен користувач та процес мають отримувати лише той мінімальний набір прав, який необхідний для виконання їхніх функцій, а критичні операції мають вимагати участі декількох ролей. ENISA, аналізуючи масив CVE, фіксує, що відсутність чи некоректність авторизації, а також неналежна автентифікація (CWE-287, CWE-863, CWE-306) входять у число вразливостей, які найчастіше лежать в основі успішних експлойтів проти серверів і веб-API. Дані ENISA показують (рисунок 2.6), що CWE-287 (Improper Authentication) та CWE-863 (Incorrect Authorization) входять до топ-15 найпоширеніших класів вразливостей (рисунок 2.6), що приводять до CRITICAL-серйозності.

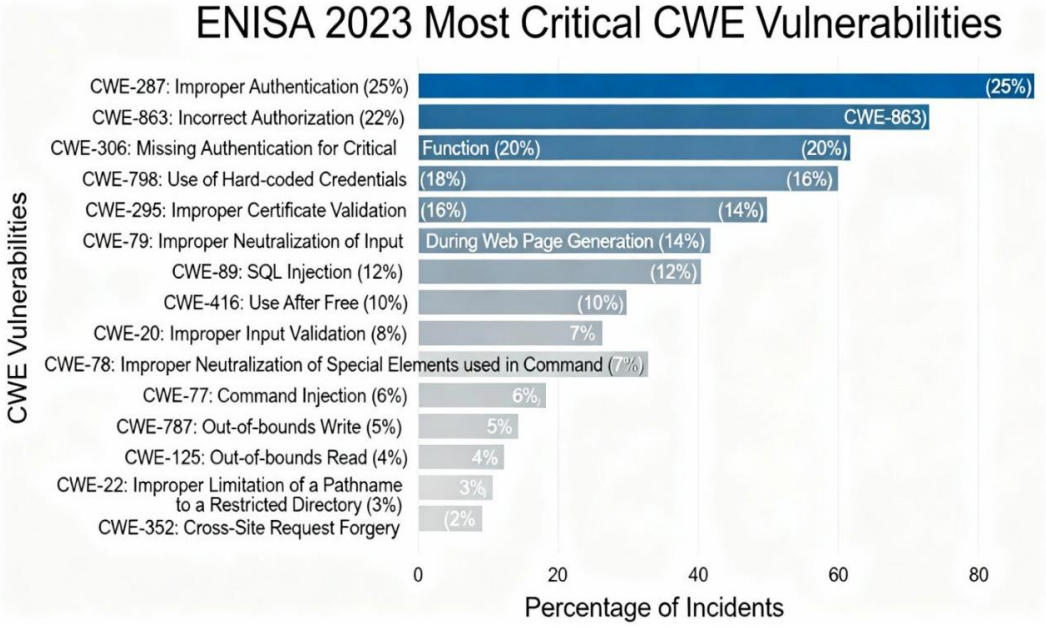

Рис. 2.6. Топ-15 CWE вразливостей за рівнем критичності (за даними ENISA)

У практичному вимірі це означає, що невірно налаштована рольова модель, наявність сервісних облікових записів із надмірними правами, використання спільних або жорстко закодованих облікових даних, а також відсутність багатофакторної автентифікації в точках адміністрування й віддаленого доступу - усе це створює критичні вектори атаки, які часто використовуються як злочинними групами, так і APT-акторами. Особливо критичним є незахищення адміністративних облікових записів, оскільки їх компрометація дає нападнику майже повний контроль над сервером.

Не менш важливим критичним виміром є захист даних, що обробляються та зберігаються на серверах. У контексті NIST це охоплює як конфіденційність, так і цілісність і доступність, причому засоби технічного контролю (криптографія, контроль доступу, резервне копіювання) мають розглядатися в комплексі з політиками класифікації й управління життєвим циклом інформації (рисунок 2.7).

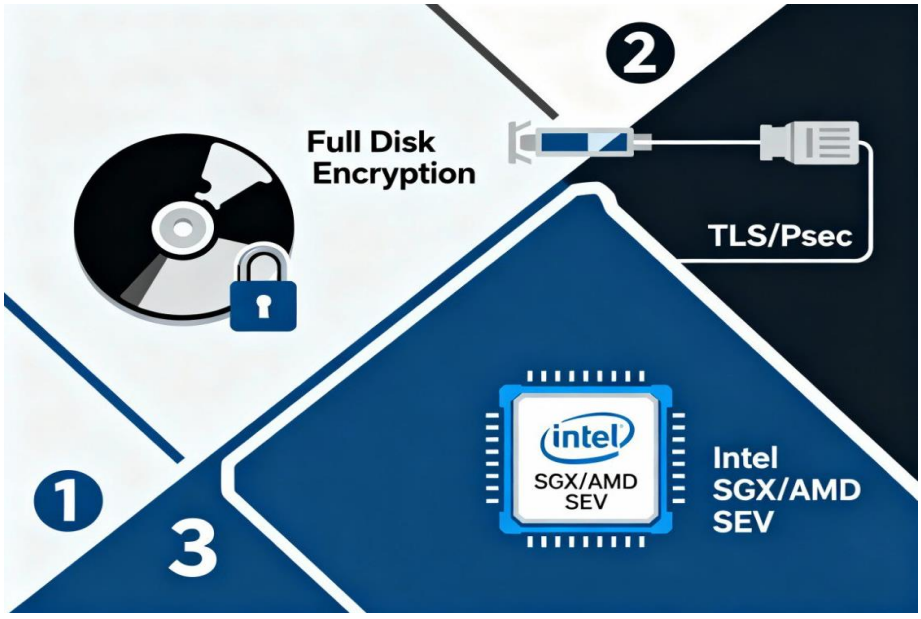

Рис. 2.7. Модель захисту даних: Data at Rest, in Transit, in Use

CISSP підкреслює, що цінність даних часто значно перевищує вартість носія, а отже використання якісних носіїв, повноцінного шифрування (наприклад, AES-256 для даних «на спокої»), правильної утилізації й санітизації носіїв (відповідно до NIST SP 800-88) є не рекомендацією «для бажаючих», а обов'язковою умовою запобігання витокам. За даними CISSP, неправильно утилізовані жорсткі диски та backup-сховища з конфіденційною інформацією залишаються одним із найпростіших способів витоку великих обсягів даних, оскільки процес стирання на рівні файлової системи часто залишає дані доступними для відновлення спеціалізованими інструментами. ENISA, аналізуючи загрози проти даних, наголошує на тому, що масові витоки часто спричинені не складними 0-day атаками, а банальними помилками: незашифровані резервні копії, публічно доступні сховища (S3-бакети, відкриті архіви), погано керовані права доступу. За даними ENISA, у першій половині 2023 року спостерігався перелік інцидентів, де організації залишали резервні копії та файли з критичними даними в публічно доступних хмарних сховищах без шифрування та обмеження доступу. Отже, відсутність системної політики класифікації інформації, чітко визначених вимог до шифрування даних «у спокої» (data at rest) і «в транзиті» (data in transit), а також до строків зберігання і процедур знищення створює критичну прогалину в загальній безпеці сервера [14, 15].

Окремий блок критичних аспектів стосується можливості організації вчасно виявляти атаки та реагувати на них. NIST SP 800-123 прямо вказує, що без правильно налаштованого ведення журналів, регулярного перегляду логів, використання автоматизованих засобів аналізу подій і періодичного тестування безпеки неможливо підтримувати належний рівень захисту серверів. CISSP систематизує це у вигляді концепцій аудиту, моніторингу й використання засобів IDS/IPS та SIEM для централізації збору й кореляції подій (рисунок 2.8) [6].

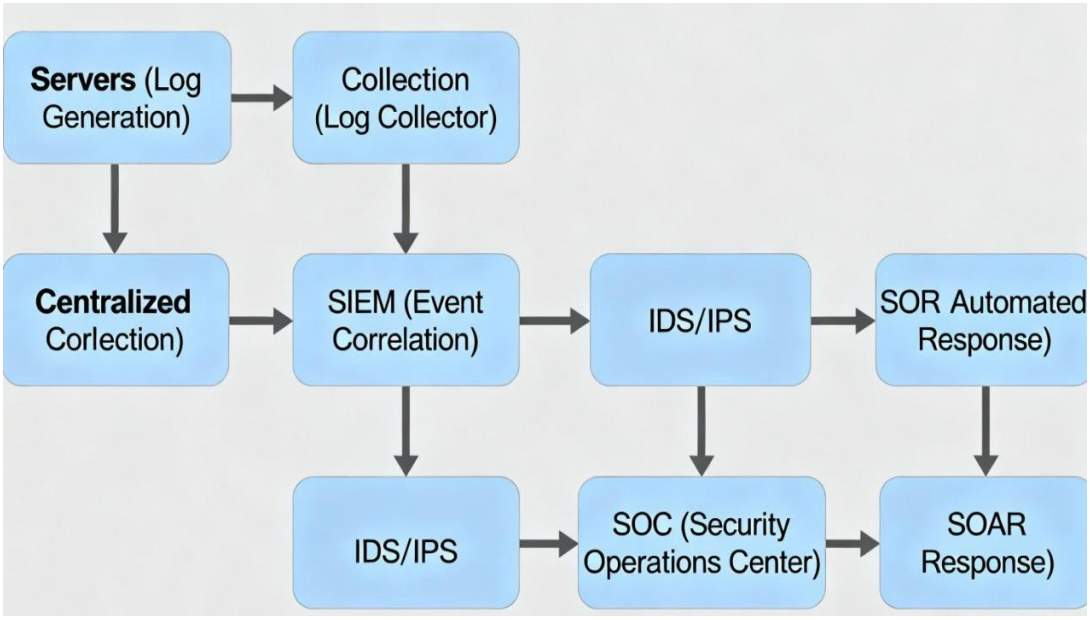

Рис. 2.8. Компоненти системи виявлення та реагування на інциденти

ENISA в низці рекомендацій для протидії malware та ransomware підкреслює важливість побудови безперервного моніторингу, де журнали з серверів, мережевих пристроїв і систем виявлення інцидентів збираються в єдину систему, а на їх основі будуються use case-и для виявлення аномалій та підозрілої активності.

Даючи детальні рекомендації щодо моніторингу, ENISA вказує на важливість налаштування use case-ів для виявлення таких феноменів, як: ненормальні логіни в нічний час, спроби підвищення привілеїв, масові невдалі спроби входу (brute force), зміни критичних конфігурацій або видалення логів, запуск невідомих процесів із привілеями адміністратора, відкриття нових портів та ініціювання підозрілих мережевих з'єднань. Практика показує, що відсутність якісного ведення журналів на серверах, надмірна кількість шумних логів без належної структури, відсутність узгодженого набору показників «нормальної поведінки» та формальних процедур аналізу подій призводять до того, що навіть серйозні компрометації систем

залишаються непоміченими місяцями, дозволяючи нападникам спокійно розповсюджуватися у мережі організації.

Окремо слід виділити аспекти, пов'язані зі стійкістю серверів до атак на доступність (DoS/DDoS, збої інфраструктури, ransomware). NIST вказує на необхідність планування резервних копій, тестування процедур відновлення, наявність тестових серверів, а також чітко прописаних процедур реагування на компрометацію, включаючи вибір між відновленням із резерву та повним перевстановленням системи. ENISA, аналізуючи ландшафт загроз 2023 року, фіксує, що ransomware та інші атаки проти доступності залишаються одними з домінуючих інцидентів, причому спостерігається зростання практики data extortion без обов'язкового шифрування - тобто акцент зміщується на недоступність сервісу та шантаж організації витоком даних. За даними ENISA, у першій половині 2023 року більше половини ransomware-інцидентів було пов'язано саме з data extortion без шифрування, що демонструє еволюцію тактик і необхідність підготовки до сценаріїв, де дані не обов'язково будуть зашифровані, а лише крадені з погрозою розкриття. CISSP, зі свого боку, розглядає відмовостійкість, географічне резервування, належну організацію бекапів (у тому числі онлайн-копій) як ключові засоби підтримання доступності критичних сервісів. У цьому контексті серйозною критичною помилкою є відсутність відокремлених резервних копій, які неможливо легко знищити з скомпрометованого сервера, не протестовані процедури відновлення, одноточкова відмова (single point of failure) у вигляді єдиного файлового чи базового сервера, а також відсутність планів безперервності бізнесу і тестованих сценаріїв Disaster Recovery.

Важливим і часто недооціненим критичним аспектом є безпечне адміністрування та організація віддаленого доступу до серверів. NIST окремим розділом описує ризики віддаленого адміністрування, наголошуючи, що воно має бути дозволене лише після усвідомленого аналізу ризиків, із застосуванням сильних

механізмів автентифікації (ключові пари, двофакторна автентифікація), обмеженням допустимих джерел підключення, використанням захищених протоколів (SSH, HTTPS, VPN) замість небезпечних (Telnet, FTP), а також суворим застосуванням принципу найменших привілеїв. Багато успішних атак - як це відзначає CISSP на прикладах APT-кампаній - починалися з компрометації одного сервера через слабкі або повторно використані облікові дані адміністратора, після чого зловмисники підвищували привілеї, змінювали налаштування брандмауера, відкривали додаткові порти (наприклад, RDP на 3389), вимикали записи журналів та закріплювалися в інфраструктурі роками. Таким чином, неформальний підхід до адміністрування, використання особистих робочих станцій адміністраторів без належного захисту, відсутність ізольованих адміністративних сегментів мережі та спеціальних адміністративних робочих станцій прямо трансформуються в критичні ризики для серверів. Практичні рекомендації NIST включають обов'язковість встановлення 2FA/MFA на адміністративні облікові записи, ведення окремого журналу адміністративних дій, обмеження географічного розташування точок адміністративного входу, використання спеціалізованих Privileged Access Management (PAM) рішень для контролю та аудиту дій адміністраторів.

Практично всі джерела наголошують на людському факторі та організаційних процесах як на окремому критичному вимірі безпеки серверів. NIST підкреслює важливість наявності чіткої організаційної політики безпеки, розподілу ролей (CIO, ISSO, адміністратори), процедур зміни конфігурацій, управління ризиками, сертифікації та акредитації систем. Ці процедури, хоча й можуть здаватися бюрократичними, насправді є необхідним каркасом, який запобігає «дикому заходу» в керуванні серверною інфраструктурою. CISSP доповнює це численними прикладами, коли втрати даних і компрометація серверів стали наслідком неналежного поводження з носіями, відсутності розуміння політик класифікації, слабких знань персоналу щодо шифрування, утилізації носіїв і базових принципів

безпечної роботи. ENISA, у свою чергу, підкреслює, що більшість успішних атак - від phishing-кампаній, які ведуть до встановлення malware на сервери, до складних соціотехнічних сценаріїв - експлуатують саме людські помилки, брак обізнаності та формальної відповідальності. За даними ENISA, більш ніж 70% успішних атак містили соціотехнічний компонент, а значна частина інцидентів починалася з фішингового листа до працівників із доступом до серверної інфраструктури. Відсутність системної програми підвищення обізнаності, нерозвинена культура «trust, but verify» щодо дій з привілейованими обліковими записами, формальне ставлення до внутрішнього аудиту і ротації обов'язків створюють передумови для того, щоб навіть технічно добре захищені сервери ставали жертвами внутрішніх зловживань або елементарної недбалості [21].

Узагальнюючи, найбільш критичні аспекти забезпечення безпеки серверів утворюють взаємопов'язану систему: вразливості ПЗ та якість патч-менеджменту, правильність конфігурації служб і сервісів, суворе управління доступом та автентифікацією, комплексний захист даних, спроможність до виявлення й реагування на інциденти, стійкість до атак на доступність, безпечне адміністрування і зріла організаційна культура безпеки [15]. Ігнорування будь-якого з цих аспектів створює «слабку ланку», якою обов'язково скористаються зловмисники, як це наочно демонструють як статистика вразливостей NIST і ENISA, так і практичні кейси, що розглядаються в професійній літературі рівня CISSP. Інтеграція усіх цих аспектів у когерентну систему управління безпекою серверів, із регулярним перевірянням і коригуванням, залишається головним завданням для організацій, які прагнуть забезпечити стійкість своєї серверної інфраструктури до постійно еволюціонуючого ландшафту кіберзагроз.

2.3. Оцінка існуючих методів і засобів захисту та визначення напрямів удосконалення

Оцінювання сучасних методів і засобів захисту серверів доцільно проводити не лише з позицій «наявні / відсутні» механізми, а й з точки зору того, наскільки вони дійсно закривають актуальні загрози, описані в профільних стандартах і оглядах загроз (рисунок 2.9).

Method	Compl-lexity	Automation	Focus (architecture/ risk/business)		Application (development/ operation)
STRIDE	✓				
Attack Trees	✓	✓		✓	
PASTA	✓	✓		✓	
OCTAVE	✓	✓		✓	

Рис. 2.9. Порівняльна таблиця методів моделювання загроз

У цьому контексті корисно посилатися на рекомендації NIST SP 800-123 щодо загальної безпеки серверів, систематизацію засобів і сервісів безпеки в класичних підручниках з криптографії та мережевої безпеки (Stallings), підходи CISSP до побудови організаційних і технічних контролів, а також на емпіричні висновки ENISA та сучасні методології моделювання загроз на кшталт PASTA [5, 14, 21].

Перший важливий напрям оцінки - це мережеві засоби захисту, історично сформовані навколо брандмауерів, DMZ-зон, VPN-каналів та сегментації мережі. Stallings підкреслює, що ці механізми формують основу для реалізації базових сервісів безпеки за OSI-архітектурою: контроль доступу, фільтрація трафіку, забезпечення

цілісності сесій та захист від DoS-атак. NIST SP 800-123 рекомендує обов'язкове використання таких засобів як частини базових кроків захисту, проте наголошує на критичній важливості правильного розміщення цих засобів у топології мережі, актуальності правил фільтрації й узгодження з організаційною політикою безпеки. На практиці організації часто стикаються з проблемами: надто «відкриті» правила брандмауера, неконтрольоване нагромадження винятків, відсутність мікросегментації та сегментації довіри, що дозволяє зловмиснику після початкового проникнення вільно переміщатися всередині мережі та аналізувати системи. VPN-рішення, побудовані на IPsec або TLS, забезпечують важливий захист трафіку для віддаленого адміністрування та географічно розподілених систем, але ENISA звертає увагу на те, що неправильно конфігуровані VPN - з відсутністю багатофакторної автентифікації, використанням застарілих протоколів, слабкими шифруванням або надто широкими мережевими повноваженнями клієнтів - перетворюються на «тунелі» для атак всередину мережевого периметра з мінімальною видимістю для стандартних засобів моніторингу. Тому традиційні мережеві засоби залишаються необхідними, але явно недостатніми для захисту від сучасних атак на прикладному рівні, компрометації облікових записів та внутрішніх загроз. Напрямом удосконалення є перехід від простої периметрової безпеки до мікросегментації і принципів Zero Trust, які передбачають мінімізацію довіри до внутрішнього трафіку та обов'язкову багатошарову автентифікацію й авторизацію для кожного запиту.

Другим критичним напрямом оцінки залишається зміцнення (hardening) серверів та засоби захисту на рівні хоста. NIST SP 800-123 розглядає цю площину як базовий захист: видалення або деактивацію зайвих сервісів і мережевих портів, налаштування автентифікації на рівні ОС, контроль ресурсів, встановлення додаткових засобів безпеки (антивірус, HIDS, host-based firewall) та регулярне тестування безпеки. CISSP доповнює цей підхід акцентом на стандартизацію конфігурацій, регулярний аудит систем щодо відповідності політикам безпеки й

впровадження засобів контролю цілісності файлів, білих списків (white-listing) та поведінкових методів захисту.

Однак у реальному світі організації часто стикаються з фрагментованістю цих практик: одні сервери жорстко зміцнені і відповідають затвердженим профілям безпеки, тоді як інші експлуатуються з дефолтними налаштуваннями; немає централізованого контролю конфігурацій, що призводить до їх дрейфу з часом; традиційні антивірусні засоби виявляють лише відомі сигнатури і не справляються із цільовими атаками, безфайловим malware та зловмисною активністю у пам'яті. ENISA у своїх рекомендаціях наголошує на необхідності переходу від класичних антивірусних рішень до сучасних EDR/XDR-платформ, здатних аналізувати поведінку процесів, взаємодію між хостами й виявляти аномалії, що відповідають реальним тактикам зловмисників за фреймворком MITRE ATT&CK. Напрямом удосконалення в цій площині є максимальна автоматизація hardening-процесів (інфраструктура як код, Ansible, Chef, PowerShell DSC), впровадження безперервної оцінки стану конфігурацій (continuous compliance) та перехід від чисто сигнатурних засобів до поведінкових, контекстних і аналітичних технологій захисту хостів.

Третій важливий напрям - оцінка криптографічних механізмів та їх дійсного застосування. Stallings систематизує криптографію як фундамент для побудови сервісів автентифікації, конфіденційності, цілісності, невідмовності і доступності, опираючись на симетричне (AES) й асиметричне шифрування (RSA, ECDH), алгоритми цілісності (хеш-функції, HMAC) та протоколи автентифікації. CISSP доповнює це практичними підходами до класифікації інформації, шифрування даних «на спокої» і «в русі», керування ключами (генерація, розподіл, зберігання, ротація, знищення) та дотриманням вимог NIST SP 800-88 щодо санітизації носіїв.

З точки зору сучасних серверних систем, криптографія як технологія здебільшого розвинена добре: використовуються стійкі алгоритми (AES-128/256, SHA-2, TLS 1.2/1.3), IPsec-тунелі, шифрування дисків. Проте фактичні проблеми

полягають у використанні застарілих версій протоколів (SSL 3.0, ранні версії TLS), недостатній перевірці сертифікатів, неправильному керуванні ключами (ключі зберігаються у відкритому вигляді на серверах, у коді), а також непослідовному і вибірковому застосуванні шифрування в критичних точках системи. За даними ENISA, вразливості CWE-798 (Hardcoded Credentials) та CWE-295 (Improper Certificate Validation) залишаються серед найпоширеніших причин інцидентів, що демонструє розрив між наявністю криптографічних механізмів та їх правильною реалізацією. Напрямом удосконалення є криптогнучкість (готовність швидко переходити на нові стандарти), побудова централізованих сервісів керування ключами (HSM, KMS), жорстка заборона застарілих протоколів, а також забезпечення шифрування на всіх критичних ділянках передачі та зберігання даних, включно з внутрішніми каналами зв'язку.

Четвертий аспект оцінки стосується моніторингу, виявлення і реагування на інциденти, що є, за словами NIST SP 800-123, обов'язковою складовою підтримання безпеки серверів. Організацією логів, їх захистом, регулярним аналізом, використанням автоматизованих засобів виявлення аномалій, періодичним тестуванням безпеки та чітко визначеними процедурами відновлення після компрометації займаються як стандартні, так і CISSP - рекомендовані практики. ENISA в огляді 2023 року підкреслює, що попри наявність SIEM-систем у багатьох організацій спостерігаються типові проблеми: неповна інвентаризація серверів, деякі системи взагалі не генерують або не відправляють логи; надлишок не налаштованих напівофіційних правил, що веде до «втоми від сповіщень» (alert fatigue); відсутність use case-ів, спеціалізованих на виявлення конкретних сучасних загроз (ransomware, supply chain attacks, APT-кампанії). Через це частина реально виявлених інцидентів залишається невисокою, а виявлені інциденти фіксуються із значною затримкою, що дає нападникам час на закріплення і розповсюдження. Напрямом удосконалення є побудова зрілої функції моніторингу з повною інвентаризацією активів, уніфікованим

форматом логів, централізацією журналів, розробкою кореляційних правил і сценаріїв, спеціалізованих на конкретні загрози, інтеграцією засобів реагування (SOAR, автоматизовані плейбуки), регулярними навчаннями, а також розширенням моніторингу на рівень додатків, баз даних, хмарних сервісів та SDN/NFV-інфраструктури.

П'ятий напрям оцінки - розвиток засобів безпеки на прикладному рівні та у нових архітектурах. Значна частина сучасних критичних атак спрямована не проти класичної мережевої інфраструктури, а проти веб-застосунків, REST/SOAP-API та сервісних архітектур (мікросервіси, хмарні середовища). OWASP та ENISA демонструють, що критичні інциденти часто пов'язані з ін'єкціями, XSS, некоректною автентифікацією й авторизацією, небезпечним завантаженням файлів, SSRF та іншими вразливостями рівня додатку, які не фільтруються традиційними брандмауерами. У сучасних SDN/NFV-середовищах, що застосовуються для гнучкого керування мережевими функціями, з'являються нові засоби захисту (контроль доступу до контролера SDN, політики мікросегментації, віртуалізовані мережеві функції безпеки), але паралельно виникають і нові вектори атак: компрометація контролера, вразливості в оркестрації, атаки на API-інтерфейси управління. Класифікація Stallings щодо розміщення механізмів безпеки на різних рівнях стеку показує важливість розповсюдження контролів не лише на мережевому рівні, а і ближче до додатків, оскільки саме там реалізуються сучасні загрози. Напрямом удосконалення є інтеграція засобів захисту веб - та API-рівня (WAF, API-шлюзи з контролем автентифікації й авторизації, захист від ін'єкцій), безпеки SDN/NFV- площини керування, а також упровадження безпеки у CI/CD-конвеєри для виявлення вразливостей додатків до їх розгортання.

Важливим напрямом розвитку є також перехід від формального дотримання чек-листів контролів до ризик-центричних методологій, які дозволяють будувати захист саме там, де це максимально знижує ризик для організації. Методологія PASTA

(Process for Attack Simulation and Threat Analysis) пропонує поетапний підхід від визначення бізнес-цілей й технічного обсягу, через декомпозицію архітектури й побудову діаграм потоків даних (DFD), аналіз загроз і вразливостей, моделювання атак до аналізу залишкового ризику та визначення контрзаходів. Такий підхід дозволяє зіставляти технічні вразливості з реальними сценаріями атак і мотивами зловмисників, замість розгляду їх у відриві від контексту; кількісно оцінювати ймовірність та вплив загроз; пріоритезувати заходи не за формальним рівнем критичності CWE/CVE, а за вкладом у зниження сукупного ризику для ключових бізнес-процесів (рисунок 2.10) [21].

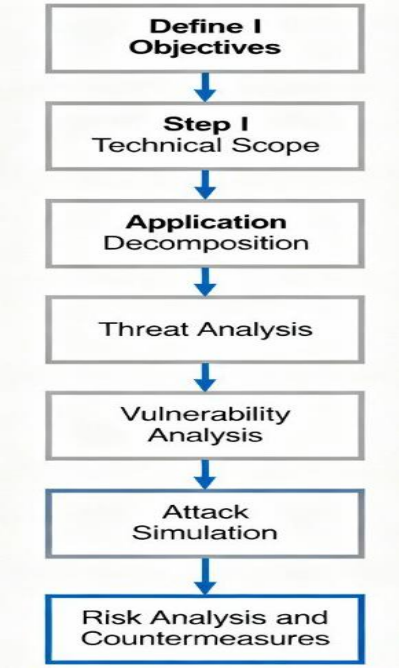

Рис. 2.10. Процес моделювання загроз за методологією PASTA

Інтеграція PASTA-подібних підходів у процеси аналізу і проєктування серверних рішень дозволяє більш усвідомлено підходити до вибору методів та засобів захисту.

Узагальнюючи, сучасний спектр засобів захисту серверів можна охарактеризувати як технологічно зрілий, але фрагментований і неповно інтегрований у єдину систему безпеки. Стандарти NIST SP 800-123, криптографічна теорія Stallings,

керівництва CISSP та аналітика ENISA задають чіткі орієнтири, проте на практиці часто спостерігаються реактивний характер безпеки, орієнтація на периметр при недостатній увазі до прикладного рівня і нових архітектур, а також слабка прив'язка заходів безпеки до реального бізнес-ризику та сценаріїв атак. Ключові напрями вдосконалення передбачають модернізацію архітектури захисту (Zero Trust, мікросегментація, криптографія на всіх рівнях, захист додатків), максимальну автоматизацію процесів (hardening, патч-менеджмент CI/CD-інтеграція), посилення моніторингу і реагування (SIEM/EDR/XDR, плейбуки, навчання), криптогнучкість, централізоване керування ключами та впровадження ризик-центричних методологій на кшталт PASTA, які пов'язують технічні засоби з конкретними бізнес-цілями й загрозами. Саме комплексне поєднання цих напрямів, а не ізольоване використання окремих інструментів, створює основу для побудови стійкої до сучасних кіберзагроз серверної інфраструктури.

3 ТЕХНОЛОГІЯ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ СЕРВЕРІВ НА ОСНОВІ МЕТОДОЛОГІЇ STRIDE

3.1. Побудова моделі загроз серверної системи за методологією STRIDE

Побудова моделі загроз серверної системи за методологією STRIDE представлена в третьому розділі і базується на результатах теоретичного аналізу першого розділу та комплексного аналітичного дослідження проведеного в другому розділі.

Методологія STRIDE вважається однією з найбільш адаптованих та практично відпрацьованих у сфері моделювання загроз для систем з архітектурою «клієнт-сервер» та активно впроваджена в промислових практиках розроблення та розгортання критичної інфраструктури.

Обґрунтування вибору STRIDE як методу моделювання. Методологія STRIDE, розроблена в Microsoft науковцями Loren Kohnfelder та Praerit Garg на межі 1990-2000-х років. Її сутність полягає в систематичному перегляді компонентів і взаємодій системи крізь призму шести основних категорій загроз, кожна з яких корелює з фундаментальною властивістю безпеки:

- Spoofing - підміна ідентичності, порушення автентичності;
- Tampering - підробка та модифікація, порушення цілісності;
- Repudiation - відмова від дій, порушення невідмовності;
- Information Disclosure - розголошення інформації, порушення конфіденційності;
- Denial of Service - відмова в обслуговуванні, порушення доступності;

- Elevation of Privilege - підвищення привілеїв, порушення авторизації.

Кожна з цих категорій інтерпретується як порушення відповідної бажаної властивості безпеки. Фундаментальні властивості та стандартизовані міжнародні норми (ISO/IEC 27001, NIST), забезпечують методологічну наступність між теоретичними положеннями, аналітикою ризиків та розробкою технології захисту [8].

Вибір STRIDE обґрунтований кількома факторами:

- методологія забезпечує систематичність та повторюваність процесу виявлення загроз, що дозволяє уникнути випадкових пропусків критичних векторів атак;

- STRIDE легко навчається та використовується командами, які не мають глибокої експертизи в безпеці, що важливо при розповсюджуванні культури безпеки у організації;

- методологія органічно інтегрується з практиками Secure Development Lifecycle (SDL), яких дотримуються провідні розробники та операційні команди;

- методологія STRIDE вже довела свою ефективність на десятках тисяч проектів глобально, від Microsoft-систем до проектів з відкритим вихідним кодом та мережевої інфраструктури.

Підготовка моделі системи: побудова DFD з виділенням довірчих меж. Базою для застосування STRIDE виступає адекватна модель системи, найчастіше у вигляді діаграми потоків даних (Data Flow Diagram, DFD) з виділенням довірчих меж (trust boundaries). Відповідно до рекомендацій OWASP та Microsoft, коректна відповідь на перше ключове запитання маніфесту моделювання загроз «що саме моделюється?» потребує такого подання, яке відображає чотири основні типи елементів:

- зовнішні сутності (external entities) як джерела та приймачі інформації поза межами системи;

- процеси (processes) як логічні одиниці обробки даних;

- сховища даних (data stores) як депозитарії інформації;

- потоки даних (data flows) як канали передачі інформації між елементами.

Зовнішні сутності включають користувачів веб-сервісу (як зовнішніх, так і внутрішніх), адміністраторів та операторів, інтеграційні сервіси третіх сторін (платіжні шлюзи, служби електронної пошти, системи СМС для двофакторної автентифікації), а також потенційного зловмисника як зовнішній актор для цілей аналізу.

Процеси охоплюють веб-сервер та сервер застосунків, що обробляють HTTP(S)-запити, сервер аутентифікації і авторизації (наприклад, LDAP, Active Directory, або спеціалізований IdP), сервер баз даних, що управляє централізованим сховищем даних, сервер ведення журналів та кореляції подій (подібних SIEM-системам), сервер резервного копіювання, а також засоби адміністрування (SSH-гейт, консолі управління).

Сховища даних складаються з основної бази даних прикладного рівня, довідкових та конфігураційних баз, файлових сховищ (у тому числі розміщення резервних копій), журналів подій безпеки та аудиту, тимчасових файлів, кешів.

Потоки даних включають HTTP(S) та REST API-запити від клієнтів до веб-сервера з передачею облікових даних та корисної інформації, SQL-запити від сервера застосунків до сервера баз даних, обмін обліковими даними та токенами сесій з сервером аутентифікації, потоки структурованих журналів до централізованої підсистеми логування та аналізу, резервне копіювання даних на окремий сервер чи в хмарне сховище, адміністраторські з'єднання через SSH, RDP, VPN або іншими захищеними каналами.

На основі цієї детальної структури проводиться критичний етап - виділення довірчих меж. Довірча межа є лінією, яка розділяє області системи, контрольовані різними суб'єктами або що мають різні рівні привілеїв та довіри. Типовими для сучасної серверної системи є: межа між зовнішнім середовищем (публічним Інтернетом) і демілітаризованою зоною (DMZ), де розташовано фронтальні веб-

сервери, за якою слідує межа між DMZ та захищеною внутрішньою мережею організації, в якій розміщуються сервери застосунків, баз даних, служб каталогів; межа між внутрішньою мережею та привілейованим адміністративним сегментом, з якого дозволено доступ до консолей керування та критичних сервісів; межі між різними віртуальними мережами (VLANs) та сегментами в випадку SDN/NFV- або хмарного середовища, якщо інфраструктура віртуалізована. Біля довірчих меж загрози «кластеризуються» найбільш щільно, оскільки зіштовхуються принципово різні суб'єкти з різними наборами привілеїв, різні прошарки керування та кооперaційні домени.

Методологія STRIDE підкреслює важливе уточнення: загрози можуть виникати і всередині однієї межі довіри - наприклад, при складному розборі (parsing) даних або при обробці вхідних повідомлень від внутрішніх компонентів, коли один компонент в одній частині мережі атакує іншу. Тому модель повинна відображати як зовнішні точки входу (entry points) у систему, так і критичні точки всередині системи, де можуть виникнути атаки.

Застосування методології STRIDE-per-element для структурованого аналізу. Після побудови DFD-моделі та виділення довірчих меж застосовується STRIDE-аналіз. Згідно з підходом STRIDE-per-element, який активно застосовується в Microsoft Threat Modeling Tool та рекомендується OWASP та більшістю організацій, що займаються безпекою, кожен тип елементів діаграми асоціюється з певним піднабором типових загроз. Це дозволяє зробити процедуру виявлення загроз структурованою і повторюваною [19].

Для зовнішніх вразливостей у контексті серверної системи актуальними залишаються перш за все загрози Spoofing та Repudiation. Підміна ідентичності користувачів (як кінцевих, так і адміністраторів) через викрадення облікових даних, перехоплення токенів сесій, зламані VPN-доступи, фішинг та інші вектори безпосередньо пов'язана з критичними проблемами управління ідентифікацією та

автентифікацією. Детальний аналіз таких сценаріїв включає розгляд способів, якими зловмисник може втручатися у процес автентифікації: перехопленням небезпечних каналів передачі даних (HTTP без шифрування, небезпечні версії TLS), словниковими атаками на паролі за відсутності механізмів обмеження кількості спроб входу або блокування облікового запису, фішинговими атаками, які насправді роблять користувача жертвою, видаючи себе за легітимний сервіс та збираючи облікові дані. STRIDE-модель дозволяє зафіксувати сценарії, коли компрометація облікового запису користувача дає можливість зловмиснику не лише доступ до даних цього користувача, але й можливість виконання операцій від його імені, що в контексті адміністраторів означає практично повний контроль над сервером [17].

Процеси (веб-сервер, сервер застосунків, сервер БД, сервер аутентифікації, лог-сервер) є основними «жертвами» майже всіх класів STRIDE-загроз. Для них релевантними є сценарії:

- Spoofing, коли зловмисник запускає фальшивий екземпляр сервісу під тим самим ім'ям або адресою, що дозволяє перенаправити трафік на компрометований сервер;
- Tampering (модифікація), коли здійснюється вплив на внутрішній стан процесу шляхом SQL-ін'єкцій, переповнення буфера, маніпуляції параметрами та вхідними даними;
- Repudiation (відмова від дій), коли спроби персоналу заперечити виконання критичної адміністративної операції залишаються без належного аудиту та ведення журналів;
- Information Disclosure (розголошення інформації), коли відбувається виведення чутливих даних у повідомленнях про помилки та «проливання» внутрішніх деталей конфігурації, версій компонентів, структури бази даних:

- Denial of Service (відмова в обслуговуванні), коли здійснюється перевантаження обчислювальних ресурсів складними запитами, породження надмірної кількості сесій та з'єднань;
- Elevation of Privilege (підвищення привілеїв), коли здійснюється експлуатація вразливостей у коді сервера для виконання довільних команд із правами системного користувача або гостя до користувача та далі до root.

У моделі загроз STRIDE кожен процес серверної системи розглядається окремо з урахуванням його специфічних функцій, спектру доступу до даних, розташування відносно довірчих меж і критичних аспектів безпеки, виявлених раніше.

Сховища даних (особливо основна база даних, конфігураційні сховища, журнальні сховища, резервні копії) розглядаються як потенційні мішені для класів загроз Tampering, Information Disclosure та Denial of Service. У рамках моделі STRIDE фіксуються детальні сценарії: зловмисник змінює критичні записи в БД через експлуатацію вразливостей у розробленому коді БД неправильної конфігурації прав доступу, що призводить до маніпуляції правами користувачів, підміни фінансових даних, модифікації облікових записів, порушення цілісності критичної інформації з потенційно катастрофічними наслідками для бізнесу, зчитує чутливу інформацію персональні дані, облікові записи, криптографічні ключі, конфігурації сервісів через обхід контролю доступу, використання вразливостей (як-то time-of-check-time-of-use), side-channel атак на виконання операцій, запозичених даних, блокує або деградує доступ до сховища через заповнення дискового простору, шифрування даних за допомогою ransomware, порушення цілісності метаданих, знищення (фізичне або логічне) критичних структур даних.

Для журнальних сховищ додатково актуальними є загрози Repudiation, коли зловмисник намагається видалити, модифікувати або затерти критичні записи журналів з метою приховування слідів своєї атаки та ускладнення судово-технічного аналізу (digital forensics).

Потоки даних у межах серверної системи, особливо ті, що перетинають довірчі межі (клієнт-веб-сервер, веб-сервер-сервер застосунків, за стосунок-сервер БД, сервери-сховище резервних копій), виступають у моделі STRIDE природними точками фокусування для загроз Spoofing, Tampering, Information Disclosure та Denial of Service. Для кожного такого потоку проводиться детальний аналіз можливих сценаріїв: підміна кінцевої точки через man-in-the-middle атаки, DNS та ARP-спуфінг, підміну API-endpoint, що зумовлює хибну ідентифікацію сервера веб-клієнтом та дає можливість зловмиснику перехопити, модифікувати або перенаправити трафік, модифікація трафіку між компонентами через вставку або зміну пакетів, replay-атаки, порушення цілісності протоколу, що може призвести до маніпуляцій із транзакціями, конфігураціями, станом, пасивне прослуховування трафіку з метою отримання облікових даних, API-ключів, криптографічних матеріалів, конфіденційних записів і подальшого їх використання для атак на інші компоненти системи, блокування або деградація каналу через перевантаження мережі, флудинг та зловживання ресурсами протоколу, що викликає відмову в обслуговуванні.

Інтеграція результатів STRIDE з фундаментальними положеннями та критичними аспектами. STRIDE-аналіз органічно поєднується з «рамкою» запитань Threat Modeling Manifesto: «що моделюється?», «що може піти не так?», «що з цим робити?», «чи достатньо добре виконано?». Така послідовність запитань забезпечує трансформацію аналізу з чисто теоретичного виявлення загроз до практичних дій [14].

Важливим аспектом є можливість застосування STRIDE в декількох варіантах. STRIDE-per-element - передбачає структурований перегляд для кожного типу елементів діаграми. Проте на практиці часто виявляється корисним застосування STRIDE-per-interaction, де аналізуються не окремі елементи, а пари та послідовності взаємодій. Такий підхід особливо цінний для складних серверних систем з великою кількістю інтегрованих сервісів та API, де багато взаємодій перетинають довірчі межі і потребують окремого детального розгляду.

Для досліджуваної серверної системи доцільно використати комбінований підхід: на верхньому рівні STRIDE-per-element для побудови фундаментальної, повної карти загроз, яка охоплює всі основні компоненти та класи вразливостей і забезпечує повноту охоплення. Для найбільш критичних взаємодій - деталізований STRIDE-per-interaction дозволяє точніше врахувати специфічні для кожної взаємодії вектори атак і визначити, які механізми захисту є критичними для кожної окремої взаємодії. Таке комбінування забезпечує як повноту, так і деталізацію аналізу.

Практична реалізація моделювання: від виявлених загроз до дій послаблення. Готова модель загроз серверної системи, побудована за методологією STRIDE із використанням DFD, довірчих меж та комбінованого аналізу елементів і взаємодій, виконує критичну подвійну роль.

По-перше, вона є формалізованим відображенням результатів другого розділу в термінах конкретної архітектури: виявлені раніше критичні аспекти безпеки (патч-менеджмент, конфігурації серверів, управління ідентичністю, ведення журналів, резервне копіювання, безпечне адміністрування, стійкість до DoS-атак) отримують явне відображення у вигляді конкретних загроз STRIDE, прив'язаних до елементів моделі та їх взаємодій [19].

По-друге, модель загроз виступає безпосереднім вихідним матеріалом для розробки конкретних рекомендацій та пропозицій: кожна група виявлених загроз (Spoofing адміністративних сесій через викрадення облікових даних, Tampering даних у БД через SQL-ін'єкції, Information Disclosure через помилки сервера та журнали, Denial of Service критичних сервісів через відсутність ресурсів, Elevation of Privilege на сервері застосунків через вразливості буфера) окремо аналізується та трансформується у вимоги до архітектури системи, специфічних конфігурацій, процедур експлуатації та технологій контролю, які мають відповідати критеріям оптимальності (найменший вплив на продуктивність і простоту використання),

цільової ефективності (реальне зменшення ризику) та практичної реалізації (можливість впровадження в існуючу інфраструктуру).

За кожною із виявлених загроз стають видимими відповідні технічні контрзаходи для загроз:

- Spoofing - сильна багатофакторна автентифікація, верифікація сертифікатів TLS, моніторинг та обмеження невдалих спроб входу;
- Tampering - регулярне патчування, криптографічні механізми цілісності (HMAC, цифрові підписи), контроль прав доступу файлів, статичний аналіз коду;
- Repudiation - централізоване ведення журналів, криптографічний захист журналів, термінові дії у критичних операціях;
- Information Disclosure - шифрування даних, контроль помилок, безпечна утилізація;
- Denial of Service - механізми rate-limiting, резервне копіювання, плани відновлення;
- Elevation of Privilege - принцип найменших привілеїв, централізована авторизація, тестування безпеки.

Використання активно впровадженого у промисловій практиці підходу і підтриманого OWASP International, Microsoft та організаціями, що розробляють критичну інфраструктуру, дозволяє гарантувати структурованість, відтворюваність, практичну орієнтованість побудованої моделі на реальні загрози сучасним серверним системам у світі глобально розподіленого Інтернету та постійно еволюціонуючого спектру кіберзагроз.

3.2 Розробка технології забезпечення захисту серверів та заходів щодо усунення виявлених загроз

Розробка технології забезпечення захисту серверів на основі запропонованої у підрозділі 3.1 моделі загроз за методологією STRIDE передбачає перехід від абстрактної класифікації загроз до конкретного комплексу заходів, інтегрованих в архітектуру, конфігурацію та процеси експлуатації серверної системи. Такий комплекс базується на фундаментальні властивості безпеки (автентичність, цілісність, невідмовність, конфіденційність, доступність та коректна авторизація), а також на структуровану модель загроз за STRIDE для типової корпоративної серверної системи. Оптимальність запропонованих заходів досягається шляхом поєднання трьох рівнів:

- архітектурного (розміщення компонентів і довірчих меж);
- технічного (налаштування протоколів, сервісів, засобів контролю доступу, ведення журналів тощо);
- організаційного (процедури керування обліковими записами, оновленнями, інцидентами, аудит).

Ефективність забезпечується орієнтацією на реальні класи атак, виявлені в моделі загроз (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege), а практична реальність - базується на стандартизованих контрзаходах NIST SP 800-53 та перевірених практиках Security Development Lifecycle (SDL) [4, 7].

Базова ідея технології захисту серверів полягає в тому, що для кожного класу загроз STRIDE формується набір контрольних заходів, прив’язаний до конкретних елементів моделі (зовнішні сутності, процеси, сховища даних, потоки даних) та підтриманий відповідними контрольними сімействами NIST (AC - Access Control, IA - Identification and Authentication, AU - Audit and Accountability, SC - System and Communications Protection, SI - System and Information Integrity тощо). Ці заходи впроваджуються у вигляді єдиної технологічної «межі безпеки» навколо серверної

системи, що дозволяє послідовно зменшувати поверхню зловмисних атак, реалізовувати принцип найменших привілеїв та багаторівневий захист.

Для загроз типу ***Spoofing*** (підміна ідентичності користувачів, адміністраторів, сервісів, серверів та каналів зв'язку), виявлених у моделі загроз серверної системи, ключовим напрямом є зміцнення автентифікації та керування обліковими записами. На рівні зовнішніх сутностей це означає відмову від анонімного доступу там, де це не є строго необхідним, впровадження багатофакторної автентифікації для всіх облікових записів з привілеями адміністратора та операторів, а також жорстке обмеження і моніторинг використання спільних та групових облікових записів, як це рекомендується у сімействі контролів AC-2 «Account Management» і AC-7/AC-10/AC-11 щодо блокування доступу, обмеження сесій та блокування пристроїв.

Для серверних процесів доцільно застосувати окремі сервісні облікові записи з мінімальними правами замість використання надпривілейованих облікових записів типу Local System або root, а також відмовитись від жорстких облікових даних у конфігураційних файлах або коді - натомість використовуючи механізми захищеного зберігання ключів та секретів (DPAPI, сховища секретів тощо).

Для потоків даних та взаємодій між серверами необхідно забезпечити автентифікацію кінцевих точок на основі сертифікатів або захищених ключів (контролі IA-2, SC-11 «Trusted Path» і SC-12/SC-13 «Cryptographic Protection»), виключити використання анонімних або слабко автентифікованих внутрішніх протоколів доступу до адміністративних інтерфейсів та баз даних.

Для усунення загроз ***Tampering*** (модифікація даних на диску, у мережі або пам'яті) необхідно реалізувати цілісний контур захисту цілісності даних та конфігураційних об'єктів серверної системи. На рівні сховищ даних це означає застосування механізмів керування доступом (AC-3 «Access Enforcement», AC-6 «Least Privilege», AC-16 «Security Attributes») для жорсткого розмежування прав на читання, запис та виконання, а також використання криптографічних механізмів

контролю цілісності (HMAC, цифрові підписи) для критичних файлів конфігурації, виконуваних модулів та резервних копій.

Для мережевих потоків, особливо тих, що перетинають довірчі межі (зовнішня мережа-DMZ, DMZ-внутрішня мережа, внутрішня мережа-адміністративний сегмент), необхідно забезпечити захист цілісності за допомогою протоколів TLS сучасних версій, VPN-тунелювання та застосування граничних засобів контролю (SC-7 «Boundary Protection», SC-8 «Transmission Confidentiality and Integrity», AC-4 «Information Flow Enforcement»). У відповідності до рекомендацій SDL реалізація внутрішніх перевірок вхідних даних у серверних процесах (валідація довжини, формату, діапазону значень) повинна відбуватися на основі «білих списків» та принципу «відхиляй все, що не відповідає очікуваному формату», а не спроб «очистити» вже шкідливі дані. Це безпосередньо зменшує ризик модифікації внутрішнього стану процесів через ін’єкційні атаки або переповнення буфера.

У протидії загрозам ***Repudiation*** (відмови від проведених дій, оспорювання авторства операцій) центральне місце займає побудова надійної, криптографічно захищеної системи ведення журналів та аудиту. Відповідно до контролів AU-2 «Event Logging», AU-6 «Audit Review, Analysis, and Reporting», AU-8/AU-10 «Time Stamps, Non-Repudiation» необхідно забезпечити централізований збір журналів з усіх критичних компонентів (веб-сервери, сервери застосунків, сервери БД, сервери аутентифікації, засоби адміністрування), зберігаючи записи в захищеному сховищі, доступ до якого обмежено. Журнали повинні містити уніфіковану інформацію про операції входу, зміни прав доступу, виконання адміністративних дій, доступ до критичних даних і спроби модифікації журналів, а часові мітки мають бути синхронізовані на всіх серверах (NTP, AU-8). Для мінімізації ризику маніпуляції журналами рекомендується використовувати криптографічні ланцюги хешів чи цифрові підписи для блоків журналів, що дозволить виявляти будь-які спроби

несанкціонованої зміни записів, а також перенаправляти журнали до окремого SIEM-сервера, фізично та логічно ізольованого від серверів, що генерують події.

Практика SDL підкреслює, що наявність якісного аудиту є передумовою для ефективного реагування на інциденти та для обґрунтованого заперечення або підтвердження дій користувачів.

Для загроз ***Information Disclosure*** (несанкціоноване розголошення інформації) технологія захисту серверів має забезпечити як криптографічний, так і некриптографічний контроль конфіденційності. На рівні передачі даних усі потоки, які перетинають довірчі межі або містять чутливу інформацію (облікові дані, персональні дані, фінансові операції, конфігураційні дані), повинні бути захищені за допомогою криптографічних протоколів (SC-8, SC-12, SC-13), причому використання застарілих алгоритмів чи протоколів (наприклад, старі версії SSL, слабкі шифри типу DES) повинно бути виключене або залишене лише як строго контрольований механізм зворотної сумісності.

У сховищах даних необхідно впровадити шифрування «в режимі спокою» (SC-28 «Protection of Information at Rest») для вмісту таблиць з високочутливими даними, резервних копій, дампів баз даних і конфігураційних файлів, а ключі шифрування зберігати у спеціалізованих сховищах ключів чи апаратних модулях безпеки, з використанням механізмів керування ключовим матеріалом, рекомендованих NIST.

Некриптографічні заходи включають заборону виведення чутливої інформації в повідомленнях про помилки, ретельну перевірку вмісту журналів та тимчасових файлів на предмет присутності персональних даних чи секретів, а також обмеження обсягу даних, доступних окремим сервісам та ролям - замість широкого доступу до всієї бази даних запроваджується принцип «мінімально необхідних даних». Важливим елементом технології є також безпечна утилізація носіїв даних (контролі MP-6/MP-7), що запобігає витоку даних через виведені з експлуатації диски, резервні носії тощо.

Усунення загроз ***Denial of Service*** (відмови в обслуговуванні) вимагає поєднання архітектурних рішень, механізмів керування ресурсами і планів відновлення. На мережевому рівні доцільно використати граничні пристрої (файрволи, балансувальники навантаження, мережеві екрани) з підтримкою обмеження швидкості (rate limiting), фільтрації аномального трафіку та протидії відомим шаблонам DoS-атак (SC-7, AC-4).

На рівні серверних процесів необхідно реалізувати механізми контролю ресурсів - обмеження максимальної кількості одночасних сесій (AC-10 «Concurrent Session Control»), тайм-аути бездіяльності (AC-12), обмеження складності і тривалості виконання окремих запитів, захист від «expensive queries» у базах даних, а також можливість відмови від обробки надмірних чи некоректних запитів на ранній стадії (input throttling).

Для сховищ даних рекомендується застосовувати порогові значення заповнення дискового простору, моніторинг обсягу журналів і тимчасових файлів з автоматичним очищенням або архівуванням, аби уникнути сценаріїв заповнення диска, що призводять до відмови в обслуговуванні. Додатково потрібен план безперервності бізнесу та відновлення після збоїв (контролі CP-2, CP-9), який визначає процедури резервного копіювання, відновлення сервісів та пріоритизації ресурсів у разі масштабної DoS-атаки.

Практики SDL підкреслюють важливість урахування аспектів доступності вже на стадії проектування: вибір масштабованої архітектури, відмовостійких кластерів, географічно розподілених ресурсів.

Загрози ***Elevation of Privilege*** (підвищення привілеїв), що характеризуються найбільш важкими наслідками (повна компрометація сервера), потребують особливо жорстких заходів у частині керування привілеями, ізоляції процесів та валідації вхідних даних.

На рівні операційної системи та серверних сервісів доцільно впровадити принцип найменших привілеїв (AC-6 «Least Privilege»), структурно розділяючи високопривілейовані та низькопривілейовані компоненти: наприклад, ядро системи чи невеликий «адміністративний» процес працює від імені облікового запису з високими привілеями і виконує лише обмежений набір функцій (керування службами, зміна конфігурації), тоді як основний код обробки запитів користувачів виконується від імені низькопривілейованих облікових записів.

Для контролю авторизації необхідна централізація перевірок прав доступу у вигляді компонентів - «reference monitor», що відповідають вимогам AC-25 (тамперостійкість, універсальність, аналізованість), і застосування ролей та атрибутів для керування правами (RBAC та ABAC - AC-7, AC-13).

На рівні програмного коду критично важливо уникати типових вразливостей, які дозволяють переписувати пам’ять процесів або обходити авторизацію. Для цього застосовуються безпечні бібліотеки, заборона небезпечних API (бібліотека «banned APIs»), механізми захисту стеку та розміщення коду (DEP, ASLR), а також статичний та динамічний аналіз коду відповідно до практик SDL. Необхідно усі шляхи виконання коду, що ведуть до підвищення привілеїв (наприклад, setuid-програми, скрипти встановлення, модулі керування), повинні бути ідентифіковані в моделі загроз і піддані поглибленому аналізу та накладанню додаткових обмежень.

Наведені групи заходів не є ізольованими. Запропоновані рекомендації на глибокому аналізі технології передбачають їх інтеграцію в єдиний життєвий цикл розроблення та експлуатації серверних систем. Практика SDL пропонує включення вимог безпеки (у тому числі вимог, похідних від STRIDE-моделі) у технічне завдання, застосування аналізу загроз під час проектування, реалізацію coding-стандартів, що забороняють небезпечні конструкції, систематичний аналіз коду і тестування з урахуванням виявлених загроз, а також підтримку процесів оновлення та реагування на інциденти. Стандарти NIST SP 800-53, у свою чергу, надають формалізовану

систему контролів, які можуть бути використані як «каркас» для формалізації впроваджуваних заходів - від керування обліковими записами та контролю доступу до ведення журналів, криптографічного захисту.

На основі запропонованої т**ехнології забезпечення захисту серверів** за методологією STRIDE сформовано комплекс рекомендацій і заходів щодо усунення виявлених загроз, які базуються на представлених теоретичних положеннях, емпіричному аналізу, структурі STRIDE-моделі, а також на сучасних стандартах безпеки та практики безпечного життєвого циклу [19]. Такий підхід забезпечує не лише зниження поточного рівня ризику для серверної системи, але й створює основу для подальшого удосконалення захисту з урахуванням еволюції загроз та вимог практики.

3.3. Оцінка ефективності запропонованої технології забезпечення захисту серверів та можливості її практичного застосування

Запропоновано рекомендації застосування т**ехнології забезпечення захисту серверів за** методологією STRIDE і сучасні підходи до кіберстійкості системи як механізму оцінювання ефективності захисту. Це дозволяє не лише описати набір заходів, але й оцінити їхню результативність та придатність до практичного застосування в реальних організаціях. Ефективність STRIDE-орієнтованих рекомендацій захисту серверів ґрунтується на поєднанні взаємопов'язаних складових:

- функціональна повнота - наскільки рекомендації покривають всі класи загроз STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) щодо серверної інфраструктури;
- відповідність нормативним вимогам і кращим практикам - наскільки реалізовані заходи узгоджуються з міжнародними стандартами NIST SP 800-53, NIST Cybersecurity Framework 2.0, NIST SP 800-61, NIST SP 800-55, ISO/IEC 27002, а також

з Законом України «Про захист інформації в інформаційно-комунікаційних системах» та національними галузевими вимогами до кіберзахисту критичної інфраструктури [8].

Наведено варіант практичного застосування т**ехнології забезпечення захисту серверів за** методологією STRIDE та рекомендації фахівцям з кібербезпеки.

Для практичного застосування методології STRIDE проводиться аналіз безпеки серверів та створюється спрощена модель архітектури системи. На рисунку 3.1 представлено систему проведення категоризації та аналізу загроз безпеки за методологією STRIDE. Така архітектура дозволяє систематично застосовувати кожну категорію STRIDE для кожного потоку даних, з'єднань та компонентів, виявляючи потенційні вразливості. Пунктирна лінія розділяє надійну внутрішню частину системи від потенційно небезпечного зовнішнього світу.

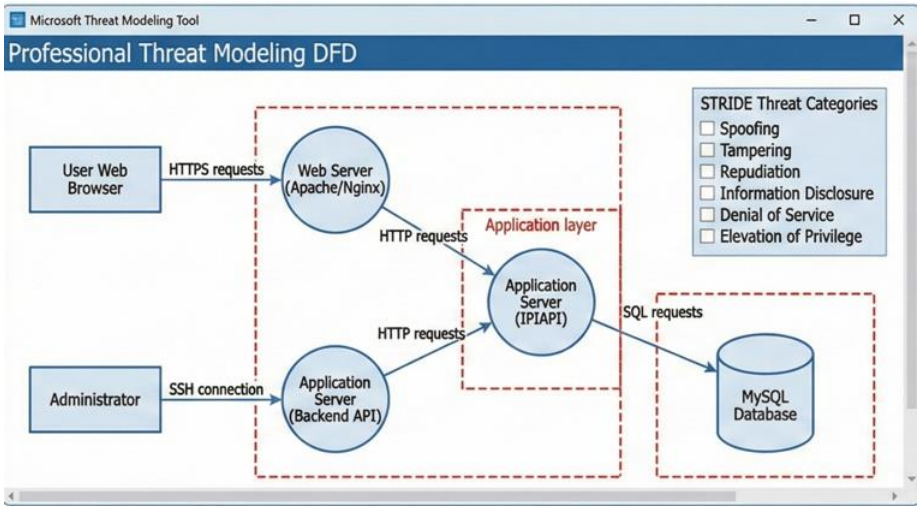

Рис. 3.1. Система проведення категоризації та аналізу загроз безпеки за методологією STRIDE

Система проведення категоризації та аналізу загроз безпеки за методологією STRIDE складається з п'яти основних частин:

- User Web Browser - користувач, який використовує систему через браузер;
- Web Server (Apache/Nginx) - веб-сервер, що приймає запити користувача;
- Application Server (Backend API) - сервер, що обробляє дані та логіку системи;
- Administrator - адміністратор, який управляє системою через SSH;
- MySQL Database - база даних, де зберігаються всі дані.

Потоки даних:

- HTTPS запити: користувач → веб-сервер (захищені);
- HTTP запити: веб-сервер → сервер додатків (внутрішні);
- SQL запити: сервер додатків → MySQL (запити до БД);
- SSH з'єднання: адміністратор → сервер додатків (управління).

Межі довіри (де криється небезпека):

1. Зовнішня межа - розділяє користувачів від серверів системи.
2. Внутрішня межа - розділяє сервер додатків від бази даних,

Дві основні лінії розділяють надійні та ненадійні зони. Саме на цих межах виникає більшість атак, тому ми зосереджуємо проведення аналізу на них.

Рекомендації фахівцям:

При проведенні аналізу систем за методологією STRIDE Microsoft Threat Modeling Tool автоматично виявляє можливі загрози для кожного компонента, потоку даних та межі довіри. Це дозволяє систематично перевіряти всю систему на вразливості (рисунок 3.2).

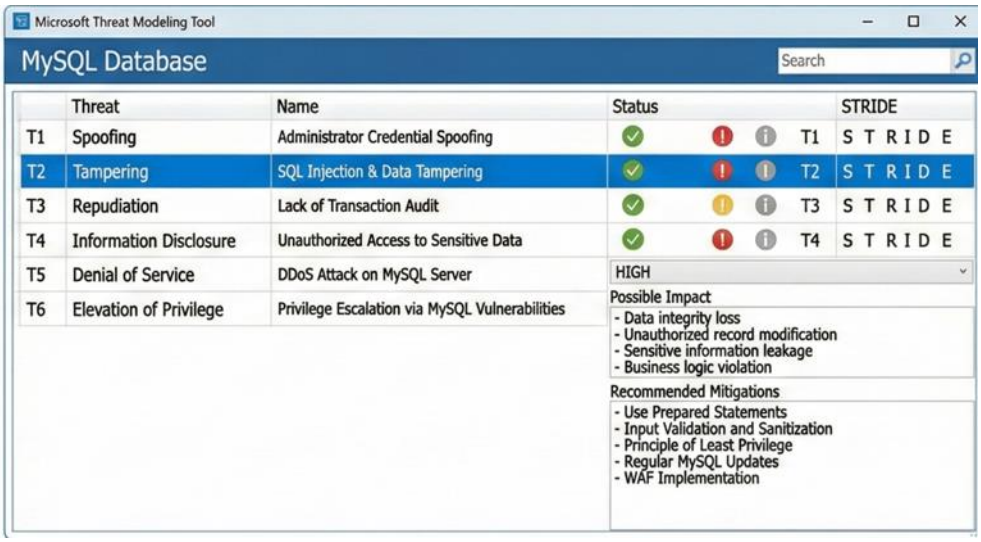

Рис. 3.2. Результат аналізу загроз компонента MySQL Database за методологією STRIDE

Після застосування методології STRIDE до бази даних MySQL - виявлено шість загроз, по одній для кожної категорії:

- T1 - Administrator Credential Spoofing (S - високий ризик): підробка облікових даних адміністратора для несанкціонованого доступу до БД;

- T2 - SQL Injection & Data Tampering (T - високий ризик): SQL-ін'єкції та несанкціонована зміна даних через вхідні поля;

- T3 - Lack of Transaction Audit (R - середній ризик): відсутність логування операцій унеможливлює виявлення атак;

- T4 - Unauthorized Access to Sensitive Data (I - високий ризик): витік персональних та конфіденційних даних через обхід контролю доступу;

- T5 - DDoS Attack on MySQL Server (D - середній ризик): DDoS атаки на БД призводять до її недоступності;

- T6 - Privilege Escalation via MySQL Vulnerabilities (E - високий ризик):експлуатація вразливостей для отримання адміністраторських прав.

Проведений аналіз демонструє, що методологія STRIDE охопила всі аспекти безпеки MySQL - від автентифікації до доступності. Жодна категорія загроз не залишилась поза увагою.

На рисунку 3.3 представлено скріншот виявлення варіанту загрози Spoofing (підміна ідентичності).

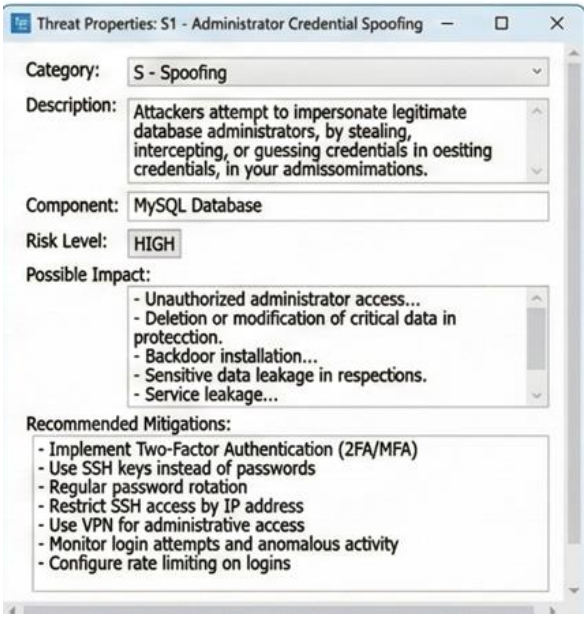

Рис. 3.3. Аналіз загрози категорії S (Spoofing)

Загроза: зловмисники намагаються видати себе за адміністраторів БД шляхом крадіжки, перехоплення або підбору облікових даних.

Можливі вразливості:

- несанкціонований адміністративний доступ до БД;
- видалення або модифікація критичних даних;
- встановлення бекдорів у системі;
- витік конфіденційної інформації.

Рекомендовані заходи протидії:

- впровадження двофакторної автентифікації (2FA/MFA);

- використання SSH ключів замість паролів;
- обмеження SSH доступу за IP-адресою;
- використання VPN для адміністративного доступу;
- регулярна зміна паролів;
- моніторинг спроб входу та аномальної активності;
- обмеження кількості спроб входу (rate limiting).

На рисунку 3.4. наведено скріншот варіанту загрози Tampering (фальсифікація даних).

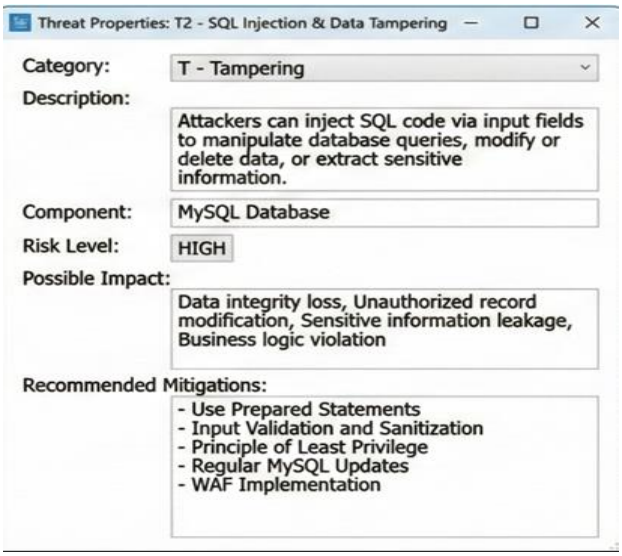

Рис. 3.4. Аналіз загрози категорії T (Tampering)

Загроза: зловмисники впроваджують SQL код через поля введення для маніпуляції запитами, модифікації або видалення даних.

Можливі вразливості:

- втрата цілісності даних через несанкціоновану модифікацію;
- витік конфіденційної інформації з БД;
- порушення бізнес-логіки додатку.

Рекомендовані заходи протидії:

- використання Prepared Statements (параметризовані запити);
- валідація та санітізація всіх вхідних даних;
- принцип найменших привілеїв для користувачів БД;
- регулярне оновлення MySQL;
- впровадження Web Application Firewall (WAF).

На рисунку 3.5. представлено скріншот варіанту загрози Denial of Service (відмова в обслуговуванні).

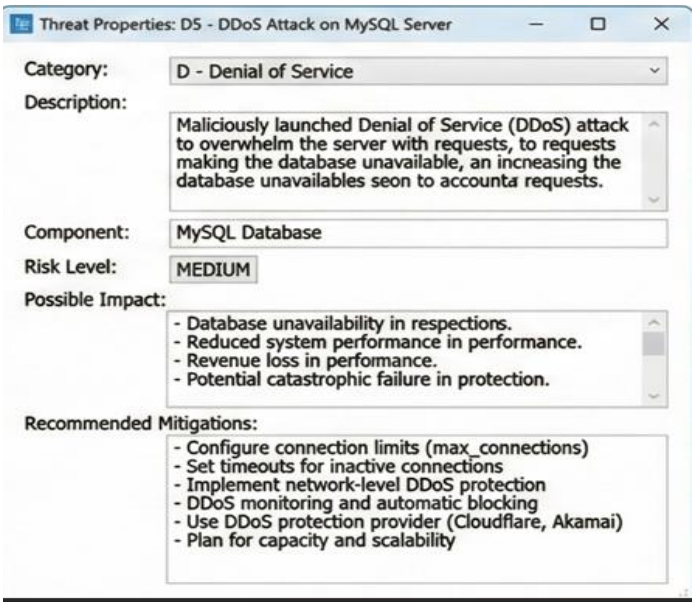

Рис.3.5. Аналіз загрози категорії D (Denial of Service)

Загроза: DDoS атака перевантажує сервер запитами, створюючи базу даних недоступною для легітимних користувачів.

Можливі вразливості:

- недоступність бази даних та всього додатку;
- зниження продуктивності системи;
- втрата доходу через простій сервісу;

- потенційна катастрофічна відмова критичних систем.

Рекомендовані заходи протидії:

- налаштування обмеження одночасних з'єднань (max_connections);
- встановлення timeout для неактивних з'єднань;
- захист на рівні мережі з фільтрацією аномального трафіку;
- моніторинг та автоматичне блокування DDoS атак;
- використання сервісів DDoS-захисту (Cloudflare, Akamai);
- планування ресурсів та можливість масштабування.

Регулярне переглядання метрик у розрізі STRIDE-загроз дозволяє оперативно виявляти нові ризики та адаптувати захист до змінних умов кіберпростору, відповідно до оновлених вимог нормативних документів, таких як NIST SP 800-53 реліз 5.2.0, що забезпечує постійну актуальність системи захисту від еволюціонуючих загроз.

За наявності передумов (керована інфраструктура моніторингу, регулярний перегляд моделей, постійний цикл моделювання-впровадження-вимірювання-коригування, повнота та точність даних про логування) сформовані рекомендації на основі методології STRIDE відповідають вимогам оптимальності (прив'язка до ризиків і уникнення надлишкових заходів), цільової ефективності (орієнтація на ключові властивості безпеки) та практичної реальності - досвід впровадження на реальні засоби та у критичні інфраструктури з високими вимогами до безперервної роботи, де вплив гіпотетичних атак може мати критичні наслідки.

STRIDE-методологія добре масштабується для великих систем, забезпечуючи системне виявлення загроз без необхідності мати «ідеальну» модель з самого початку, що дозволяє організаціям почати впровадження навіть без повної готовності інфраструктури. Її можна впроваджувати поетапно - спочатку для найбільш критичних серверів, потім поширювати на інші компоненти, згідно з пріоритизацією ризиків та наявних ресурсів.

Рекомендації за методологією STRIDE сприяють зниженню ризиків та підвищенню захищеності серверної інфраструктури, враховуючи як технічні, так і організаційні аспекти управління безпекою, але важливо враховувати обмеження та адаптувати під конкретні бізнес-завдання.

ВИСНОВКИ

Кваліфікаційна робота присвячена сучасному підходу вирішення комплексу актуальних проблем захисту та своєчасного усунення кіберзагроз серверної інфраструктури.

В роботі проведено дослідження практичного інструменту моделювання загроз методології STRIDE, визначено мету та завдання.

1. Проведено а**наліз:**

- наукових і практичних підходів до захисту серверів, зокрема, моделювання загроз як одного з найбільш значущих наукових підходів до забезпечення безпеки серверних систем;

- визнаних міжнародних стандартів безпеки, які встановлюють фундаментальні рамки для захисту серверної інфраструктури та управління ризиками сучасної кібербезпеки: NIST SP 800-123 "Guide to General Server Security", ISO/IEC 27001:2022 (оновлена версія), SDL (Security Development Lifecycle), визнана як стандарт в індустрії, набір перевірених практик кібербезпеки CIS Controls, концепція Defense in Depth (Глибокий захист)та ін.

- сучасних основних категорій загроз серверам: ransomware, фішинг, хактивізм, штучний інтелект для підвищення продуктивності та оптимізації зловмисних дій;

- понятійного апарату у сфері безпеки серверів через призму STRIDE та ISO/IEC 27001:2022, визначено прямі відповідності: кожна категорія загроз STRIDE проектується на певну властивість інформації, а кожна властивість, у свою чергу, реалізується через набір контролів, визначених у ISO/IEC 27001:2022.

2. Визначено: основні ключові функціональні напрями серверів і триєдність основних цілей безпеки CIA-тріаду - конфіденційність, цілісність, доступність доповнені сучасними критичними аспектами: автентичністю, неспростовністю,

авторизацією. Розглянуто елементи порушення їх захисту, наслідки та практичні механізми забезпечення безпеки.

3. Проведено вивчення існуючих методів і засобів захисту серверів та напрямів удосконалення. Визначено ключові заходи вдосконалення, які передбачають модернізацію архітектури захисту (Zero Trust, мікросегментація, криптографія на всіх рівнях, захист додатків), максимальну автоматизацію процесів (hardening, патч-менеджмент CI/CD-інтеграція), посилення моніторингу і реагування, криптогнучкість, централізоване керування ключами та впровадження ризик-центричних методологій на кшталт PASTA, які пов'язують технічні засоби з конкретними бізнес-цілями та загрозами. Комплексний підхід поєднання цих напрямів створює основу для побудови стійкої до сучасних кіберзагроз серверної інфраструктури.

4. Досліджено **п**рактичний інструмент моделювання загроз методологію STRIDE - галузевий стандарт для категоризації та комплексного аналізу загроз безпеки, яка класифікує всі можливі вразливості за шести основними категоріями, що охоплюють різні аспекти безпеки: Spoofing, Tamperin, Repudiation, Information Disclosure Denial of Service, Elevation of Privilege.

Моделювання загроз за методологія STRIDE допомагає виявити потенційні вектори атак та розробити ефективні механізми захисту. Однією з ключових переваг STRIDE є структурований підхід, що дозволяє побудувати систему аналізу загроз.

5. Запропоновано технологію забезпечення захисту серверів за методологією STRIDE, яка передбачає перехід від абстрактної класифікації загроз до конкретного комплексу заходів, інтегрованих в архітектуру, конфігурацію та процеси експлуатації серверної системи. Сутність технології захисту серверів полягає в тому, що для кожного класу загроз STRIDE формується набір контрольних заходів, прив’язаний до конкретних елементів моделі та підтриманий відповідними контрольними сімействами NIST. Ці заходи впроваджуються у вигляді єдиної технологічної «межі

безпеки» навколо серверної системи, що дозволяє послідовно зменшувати поверхню зловмисних атак, реалізовувати принцип найменших привілеїв та багаторівневий захист.

6. Надано рекомендації фахівцям з кібербезпеки щодо застосування запропонованої т**ехнології забезпечення захисту серверів за** методологією STRIDE і сучасні підходи до кіберстійкості системи. Представлено варіант практичного застосування т**ехнології,** проведено аналіз безпеки серверу, запропоновано спрощену модель архітектури системи проведення категоризації та аналізу загроз безпеки та результати підтверджено скріншотами виявлення варіантів вразливостей. Проведений аналіз демонструє, що методологія STRIDE охопила всі аспекти безпеки MySQL - від автентифікації до доступності, жодна категорія загроз не залишилась поза увагою.

Таким чином, впровадження технології забезпечення захисту серверів за методологією STRIDE є сучасним, актуальним вирішенням питання захисту та своєчасного усунення кіберзагроз серверної інфраструктури.

ПЕРЕЛІК ПОСИЛАНЬ

1. Закон України «Про захист інформації в інформаційно-комунікаційних системах» : від 05.07.1994 р. № 80/94-ВР (в ред. від 20.04.2025 р.). URL: https://zakon.rada.gov.ua/laws/show/80/94-вр (дата звернення: 14.12.2025).

2. Закон України «Про основні засади забезпечення кібербезпеки України» : від 05.10.2017 р. № 2163-VIII (із змін. і доп., що набирають чинності з 19.10.2025 р.). URL: https://zakon.rada.gov.ua/laws/show/2163-19 (дата звернення: 14.12.2025).

3. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. Geneva: ISO, 2022. URL: https://www.iso.org/standard/81574.html (дата звернення: 12.12.2025).

4. Scott Rose, Oliver Borchert, Stu Mitchell, Sean Connelly. Security and Privacy Controls for Information Systems and Organizations. NIST Special Publication 800-53 Revision 5. URL: https://doi.org/10.6028/NIST.SP.800-53r5 (дата звернення: 12.12.2025).

5. Karen Scarfone, Murugiah Souppaya, Paul Hoffman. Guide to General Server Security. NIST Special Publication 800-123. URL: https://doi.org/10.6028/NIST.SP.800-123 (дата звернення: 12.12.2025).

6. Karen Scarfone, Peter Mell. Guide to Intrusion Detection and Prevention Systems (IDPS). NIST Special Publication 800-94. URL: https://doi.org/10.6028/NIST.SP.800-94 (дата звернення: 12.12.2025).

7. Chew Elizabeth, Swanson Marianne, Stine Kevin, Bartol Nadya, Brown Anthony, Robinson Will. Performance Measurement Guide for Information Security. NIST Special Publication 800-55 Revision 1. URL: https://doi.org/10.6028/NIST.SP.800-55r1 (дата звернення: 12.12.2025).

8. NIST Cybersecurity Framework (CSF) 2.0. National Institute of Standards and Technology. URL: https://www.nist.gov/cyberframework (дата звернення: 12.12.2025).

9. Xiong Wenjun, Lagerström Robert. Threat modeling – A systematic literature review. Computers & Security. 2019. Vol. 84. P. 53–69. URL: https://doi.org/10.1016/j.cose.2019.03.010 (дата звернення: 12.12.2025).

10. Torr Peter. Demystifying the Threat-Modeling Process. IEEE Security & Privacy. 2005. Vol. 3, No. 5. P. 66–70. URL: https://doi.org/10.1109/MSP.2005.119 (дата звернення: 12.12.2025).

11. Shostack Adam. Threat Modeling: Designing for Security. John Wiley & Sons, Inc., 2014. 624 p.

12. Howard Michael, LeBlanc David. Writing Secure Code. 2nd ed. Microsoft Press, 2002. 704 p.

13. Anderson Ross. Security Engineering: A Guide to Building Dependable Distributed Systems. 3rd ed. John Wiley & Sons, Inc., 2020. 1232 p.

14. Stallings William. Cryptography and Network Security: Principles and Practice. 7th ed. Pearson Education, Inc., 2017. 767 p.

15. Chapple Mike, Seidl David. (ISC)2 CISSP Certified Information Systems Security Professional Official Study Guide. 9th ed. John Wiley & Sons, Inc., 2021. 1104 p.

16. UcedaVélez Tony, Morana Marco M. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. John Wiley & Sons, Inc., 2015. 325 p.

17. Howard Michael, Lipner Steve. The Security Development Lifecycle. Microsoft Press, 2006. 288 p.

18. SDN and NFV Security: Security Analysis of Software-Defined Networking and Network Function Virtualization / ed. by Rahamatullah Khondoker. Springer International Publishing, 2016. 217 p.

19. The STRIDE Threat Model. Microsoft Learn. URL: https://learn.microsoft.com/uk-ua/azure/security/develop/threat-modeling-tool-threats (дата звернення: 12.12.2025).

20. OWASP Top Ten Web Application Security Risks. OWASP Foundation. URL: https://owasp.org/Top10/ (дата звернення: 12.12.2025).

21. ENISA Threat Landscape 2023. European Union Agency for Cybersecurity, 2023. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023 (дата звернення: 12.12.2025).

22. OWASP Application Threat Modeling. OWASP Foundation. URL: https://owasp.org/www-community/Application_Threat_Modeling (дата звернення: 12.12.2025).

23. Common Weakness Enumeration (CWE) and National Vulnerability Database (NVD). MITRE Corporation, National Institute of Standards and Technology. URL: https://cwe.mitre.org/ та https://nvd.nist.gov/ (дата звернення: 12.12.2025).