ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ, ПРАВА ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Пояснювальна записка

до кваліфікаційної роботи на здобуття ступеня вищої освіти магістр

на тему: **«Модель віртуалізованої інфраструктури інформаційної системи з врахуванням атак на http сервер»**

Виконав: здобувач вищої освіти
за освітньо-професійною програмою
Інформаційні управляючі системи та технології
спеціальності 126 Інформаційні системи та технології
ступеня вищої освіти магістр
групи 126ІСТмд_21
Ростовський Н.М.
Керівник: Поночовний Ю.Л.
Рецензент: Брикун О.М.

Полтава – 2023 року

ВСТУП

Актуальність теми. Віртуалізація – це процес перенесення програм на абстрактний рівень, який відокремлює фізичне обладнання від операційної системи. Цей процес дозволяє краще використовувати та управляти ІТ-ресурсами, забезпечуючи більшу гнучкість. Віртуалізація дозволяє декільком віртуальним машинам з неоднорідними операційними системами та програмами працювати ізольовано, але використовувати ресурси однієї фізичної машини – хоста.

Використання віртуалізованих середовищ накладає нові вимоги до їх ефективності. Наприклад, продуктивність віртуалізованих сховищ вимагає нових підходів до оцінювання через збільшений попит на продуктивність від кількох віртуальних систем на одному хості.

Проведений аналіз нормативної бази, можливостей інструментів віртуалізації показав, що існуючі публікації та документи більше орієнтовані на забезпечення якості системи, а не враховують актуальні ризики, такі як кібератаки на компоненти інфраструктури, зокрема вебсервери.

Розвиток методів та моделей оцінювання надійності та готовності програмного забезпечення в останні десятиліття активно досліджували такі вчені, як В. Харченко, А. Горбенко, К. Тріведі, А. Боярчук, О. Іванченко. Однак, ці дослідження не розглядали питання моделювання функціонування віртуалізованої інфраструктури в умовах кібератак на http сервер.

Зв'язок роботи з науковими програмами, темами. Робота відповідає дослідженням в межах науково-дослідної роботи «Розвиток підприємництва: управлінські, економічні, інноваційна та правові аспекти» відповідно до договору №9 від 15.05.2023 р. між ТОВ «ПАФ Гарант» та Полтавським державним аграрним університетом (розділ «Обґрунтування показників оцінювання гарантоздатності розподілених інформаційних систем»).

Метою кваліфікаційної роботи є точності оцінювання надійності та доступності віртуалізованої інформаційної системи шляхом врахування зміни інтенсивності атак на вразливості її http сервера.

Завданнями кваліфікаційної роботи є:

– аналіз зовнішніх впливів на інформаційні системи,

– аналіз інтенсивності та критичності атак на компоненти віртуалізованої інфраструктури інформаційної системи,

– побудова та дослідження моделі віртуалізованої інфраструктури інформаційної системи з врахуванням атак на http сервер.

Об'єктом дослідження є процеси розгортання та супроводу віртуалізованої інформаційної системи, міграції віртуальних машин внаслідок апаратних та програмних відмов та функціонування їх в умовах атак на http сервер.

Предметом моделі оцінювання надійності та доступності віртуалізованої інформаційної системи шляхом врахування зміни інтенсивності атак на вразливості http сервера.

Методи дослідження – проведені в роботі дослідження базуються на методах теорії ймовірності, системного і марковського аналізу, систем масового обслуговування, які використовувалися при розробці комплексу марковських моделей віртуалізованої інформаційної системи.

Інформаційна база кваліфікаційної роботи складається з наукових статей, міжнародних аналітичних видань і звітів, матеріалів наукових конференцій інтернет-ресурсів, що місять інформацію про архітектуру сучасних вебсистем, а також даних, отриманих від провідних IT-компаній у сфері віртуалізації обчислювальних ресурсів.

Елементи наукової новизни полягають у розроблені та досліджені аналітичної моделі функціонування віртуалізованої інформаційної системи в умовах відмов внаслідок апаратних та програмних відмов та атак на http сервер.

Практична значущість роботи полягає в можливості повторного застосування та модифікації розробленого програмного коду моделі для оцінювання показників надійності та доступності інформаційних систем, а також показників економічної ефективності. Отримані результати можуть бути корисними для IT фахівців при моделюванні спеціалізованих інформаційних управляючих систем.

Апробація результатів дослідження відбувалася шляхом оприлюднення доповідей на наукових конференціях, семінарах.

Публікації. За результатами проведеного дослідження опубліковано тези: «Імітаційна модель для оцінювання безпеки системи з віртуалізованою інфраструктурою з врахуванням атак на компоненти», Матеріали XII Міжнар. наук. конференції «Інформаційні технології в енергетиці та агропромисловому комплексі», м. Львів, 04-06 жовтня 2023 р.; «Аналіз стану інформаційної безпеки: видів загроз і методів їх усунення», Матер. науково-практичної конференції за підсумками виробничої практики здобувачів вищої освіти спеціальності «Інформаційні системи та технології», 17 вересня 2023 р., м. Полтава.

Структура та обсяг кваліфікаційної роботи логічно пов'язані з задачами досліджень. Робота містить перелік умовних позначень, вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг текстової частини дипломної роботи складає 70 сторінок формату А4. Вона містить 18 рисунків і 10 таблиць. У роботі використано 43 науково-технічних джерела.

РОЗДІЛ 1
АНАЛІЗ ЗОВНІШНІХ ВПЛИВІВ НА ІНФОРМАЦІЙНІ СИСТЕМИ

1.1 Зовнішні фактори порушення безпеки інформаційної системи

Основним аспектом у забезпеченні інформаційної безпеки є визначення, аналіз та класифікація можливих загроз. Загрози можна поділити за джерелом (внутрішні та зовнішні), видом (фізичні, логічні, комунікаційні, людські) та ступенем злого наміру (випадкові чи навмисні) [1].

Зовнішні фактори, що створюють дестабілізуючі умови, викликають загрози безпеці функціонування об'єктів, вразливих у системах інформаційної безпеки, таких як [2]:

− помилки оперативного та обслуговуючого персоналу під час експлуатації ІС;

− спотворення інформації у телекомунікаційних каналах, яка надходить від зовнішніх джерел і передається споживачам, а також неприпустимі зміни параметрів потоків інформації;

− відмови апаратури;

− зміни у складі та конфігурації ІС, які виходять за рамки перевірених під час тестування або сертифікації;

Повне усунення цих загроз принципово неможливе. Основна мета полягає у визначенні факторів, які впливають на ці загрози, розробці методів та засобів для зменшення їх впливу на безпеку ІС, а також ефективному розподілі ресурсів для забезпечення захисту, який був би стійким у відношенні до будь-яких негативних впливів.

Зовнішні (ненавмисні) фактори можуть також впливати на реальні складні системи, часто маючи більш серйозні наслідки, ніж злочинні дії. Наприклад, катастрофи типу Чорнобиля, катастрофи літаків та ін. можуть бути наслідком системних та алгоритмічних помилок проєктування в поєднанні з

непередбаченими дестабілізуючими факторами [3, 4]. Ці фактори мають свої характеристики, тому потребують окремого аналізу та відповідних методів захисту. Таким чином, теорія та практика забезпечення безпеки інформаційних систем охоплює заходи при випадкових дестабілізуючих впливах без злочинного втручання в ІС.

Зовнішні загрози можна розділити на дві категорії [5]:

– випадкові або ненавмисні;

– навмисні.

Випадкові загрози – це ті, що виникають без зловмисних дій і можуть спричинити найбільші втрати інформації, наносячи до 80% шкоди ресурсам [6, 7]. Вони можуть призвести до знищення, порушення цілісності та доступності інформації, створюючи підґрунтя для зловмисного впливу. Природні лиха та аварії можуть мати руйнівні наслідки для матеріальних ресурсів, що зберігають інформацію, тому що останні піддаються фізичному знищенню, інформація втрачається або стає недоступною. Збої і відмови в роботі складних систем – неминучі. Вони можуть призвести до порушень роботи технічних засобів, знищення або спотворення даних і програм. Це також може порушити конфіденційність інформації.

Другий клас загроз – це навмисно створені. Цей клас досліджений не достатньо, постійно розширюється новими загрозами. Загрози цього типу можуть бути розділені за п’ятьма групами за їх фізичною суттю та механізмами реалізації.

У цьому класі широко використовуються традиційні методи шпигунства і диверсій, що застосовуються для отримання інформації або її знищення на об’єктах, які не мають інформаційних систем [7, 8]. Ці методи дієві і в умовах застосування інформаційних систем, часто використовуються для проникнення в систему та знищення ресурсів.

Список методів шпигунства і диверсій включає такі пункти [9]:

- підслуховування;
- візуальне спостереження;

- розкрадання документів та пристроїв для зберігання інформації;
- втрата програм та атрибутів системи захисту;
- підкуп та шантаж співробітників;
- збір та аналіз відходів з пристроїв для зберігання інформації;
- підпалення;
- вибухи.

Для вирішення проблем інформаційної безпеки потрібно поєднати законодавчі, організаційні, технологічні та стандартизаційні заходи. Теорія та практика забезпечення безпеки використання інформаційних технологій та систем часто фокусуються на захисті від злочинних втручань, спотворень та втрат програмних засобів і баз даних. Для цього розроблені спеціальні методи та засоби захисту від несанкціонованого доступу, вірусів та витоку інформації через електромагнітні канали. Це означає, що існують особи, які мають інтерес до доступу до програм та даних для несанкціонованого використання, спотворення чи знищення.

1.2 Модель порушника інформаційної безпеки

Для запобігання можливих загроз необхідно не лише забезпечити захист інформації, а й спробувати ідентифікувати категорії порушників та методи, які вони використовують. Порушник – це особа, яка спробує виконати заборонені операції з помилкових причин, з невідомих міркувань або усвідомлено злим умислом (з корисливих мотивів) або без нього (для розваг, самовираження та інших причин) [10, 11]. Вони використовують різноманітні можливості, методи та засоби для цього. Зловмисник – це порушник, який навмисно порушує з корисливих мотивів [12].

Неформальна модель порушника відображає його практичні та теоретичні можливості, передбачені знання, час і місце дій тощо. Для досягнення своїх цілей

порушник повинен прикласти певні зусилля та витратити певні ресурси. Досліджуючи причини порушень, можна або вплинути на самі ці причини (якщо це можливо), або точніше визначити вимоги до системи захисту від цього виду порушень чи злочинів.

У кожному випадку для конкретної технології обробки інформації може бути визначена модель порушника, яка має бути адекватною реальному порушнику.

При розробці моделі порушника визначаються [13]:

– припущення про категорії осіб, які можуть бути порушниками;

– припущення про мотиви дій порушника (цілі, які він переслідує);

– припущення про кваліфікацію порушника та його технічні можливості (використані методи та засоби для порушення);

– обмеження та припущення про характер можливих дій порушників.

Порушники щодо інформаційних ресурсів можуть бути внутрішніми (персонал) або зовнішніми (сторонніми особами). Внутрішніми порушниками можуть бути особи з різних категорій персоналу: користувачі, технічний персонал, розробники програм, служба безпеки тощо.

Сторонні особи, що можуть бути порушниками [14]:

– клієнти або представники організацій;

– відвідувачі за будь-якими приводами;

– представники організацій, які взаємодіють із вашою організацією;

– представники конкуруючих організацій або особи, що діють за їх завданням;

– особи, що ненавмисно або навмисно порушують правила проходу;

– будь-які особи поза контрольованою територією.

Можна виділити три основні мотиви порушень: безвідповідальність, самовираження та корисливість. Порушення безпеки інформаційних систем може бути спричинене корисливими мотивами користувача системи, який намагається отримати доступ до збереженої, переданої та оброблюваної інформації в системі.

Навіть якщо система має засоби, які ускладнюють таке проникнення, повністю захистити її від проникнення практично неможливо.

Порушників можна класифікувати за наступними категоріями [15]:

1. За рівнем знань про інформаційну систему:

– особа, яка володіє знаннями про функціональні особливості системи та механізм даних;

– користувач із високим рівнем знань і досвідом технічних засобів системи;

– людина з високими знаннями у програмуванні та обчислювальній техніці;

– особа, що розуміє принципи засобів захисту системи.

2. За рівнем можливостей:

– використання агентурних методів отримання інформації;

– використання пасивних засобів без модифікації системи;

– використання штатних засобів та недоліків для подолання захисту;

– використання активних методів впливу на систему.

3. За часом дії:

– під час функціонування системи;

– у період неактивності системи;

– в будь-який час, як під час функціонування, так і в період неактивності.

4. За місцем дії:

– зовсім без доступу на контрольовану територію;

– з контрольованої території без доступу до будівлі;

– усередині приміщень без доступу до технічних засобів системи;

– з робочих місць користувачів;

– з доступом до зони даних або управління системою.

Існують обмеження та припущення про характер дій порушників, такі як ускладнення можливості створення коаліцій порушників та наслідки, що можуть бути результатом помилок адміністраторів та користувачів. Усі ці аспекти важливі для розуміння методів захисту інформаційних систем та запобігання можливим загрозам.

1.3 Аналіз концептуальних засад теорії інформаційної безпеки

Основна мета аналізу полягає в розгляді концепцій теорії інформаційної безпеки та їх переосмисленні. Теорія захисту інформації визначається як набір основних ідей, спрямованих на захист інформації в сучасних системах, надаючи усвідомлення про сутність проблеми захисту, закономірності її розвитку та зв'язки з іншими галузями знань. Ця теорія формується та розвивається на основі практичного досвіду вирішення завдань захисту і визначає стратегії для покращення практики захисту інформації. Згідно з визначенням, основні завдання теорії захисту інформації можуть бути узагальнені [16, 17]:

- надання повної та адекватної інформації щодо походження, сутності та розвитку проблеми захисту інформації;
- чітке відображення структури та зв'язків з іншими галузями знань;
- акумуляція досвіду попередніх досліджень і практичних рішень у сфері захисту інформації;
- надання необхідних науково-методологічних та інструментальних засобів для ефективного розв'язання завдань захисту;
- формулювання науково обґрунтованих перспектив розвитку теорії та практики захисту інформації.

Тепер, щодо компонентів теорії захисту інформації на сьогоднішній день, вони наступні [18, 19]:

- повна та систематизована інформація щодо проблеми захисту інформації;
- аналіз результатів попередніх досліджень і практичного досвіду, що відображає стійкі тенденції в розвитку цієї області;
- науково обґрунтована постановка завдань захисту інформації в сучасних системах обробки;
- загальні стратегічні напрямки організації захисту інформації;
- методи для ефективного розв'язання завдань захисту;
- методологічна та інструментальна база для вирішення завдань захисту;

– організація та забезпечення робіт із захисту інформації;

– прогноз перспектив розвитку теорії та практики захисту інформації.

Розвиток теорії інформаційної безпеки на сучасному етапі пов'язаний із врахуванням нових умов, які є характерними для інформатизації суспільства. Сьогодні інформаційна безпека направлена на не лише захист інформації, а й на захист людей та електронних систем від руйнівного впливу інформації [21]. Також, із поширенням автоматизованих технологій обробки інформації, виникає потреба у забезпеченні якості інформації. Розв'язання цих завдань обумовлює ефективність діяльності об'єктів та формування концепції управління інформацією, що об'єднує різні поняття в цій сфері. Також важливо покласти акцент на вдосконалення науково-методологічної бази та інструментів для розв'язання завдань інформаційної безпеки та інформатизації суспільства.

Розвиток теорії та практики інформаційної безпеки в сучасних умовах характеризується такими проблемами [7, 22]:

– недостатність теоретичних основ для адекватного опису процесів інформаційної безпеки в умовах невизначеності та непередбачуваності;

– нестача нормативно-методичних документів, що забезпечують ефективний захист інформації;

– необхідність стандартизації підходів до створення систем захисту інформації та управління ними.

Вирішення цих проблем вимагає комплексного підходу, а також забезпечення балансу інтересів особи, суспільства і держави при захисті інформації.

1.4 Аналіз видів та наслідків мережевих атак

Мережева атака – це напад на комп'ютерну систему з використанням мережевих протоколів [23]. Загроза безпеці означає потенційну ситуацію, в якій

можуть бути пошкоджені основні сервіси: цілісність, конфіденційність і доступність інформації.

Основні складові загрози інформаційної безпеки включають джерело впливу на систему, спосіб впливу, об'єкти впливу та результат цього впливу (завдана шкода). Фактори, що впливають на інформаційну безпеку, можна класифікувати як об'єктивні або суб'єктивні, внутрішні або зовнішні, залежно від їхнього походження та впливу на об'єкти системи. Розрізнення джерел на суб'єктивні та об'єктивні зумовлене потребою у визначенні винуватості за заподіяну шкоду інформації. Також різницю між внутрішніми та зовнішніми джерелами виправдано тим, що для однакових загроз потребуються різні методи захисту. Також важливо врахувати, що як зовнішні, так і внутрішні загрози можуть бути як навмисними, так і не навмисними [24].

За статистикою, опублікованою інститутом SANS (System Administration, Networking and Security, www.sans.org), до найбільш поширених атак відносять несанкціонований доступ до паролю і конфіденційної інформації, несанкціоноване виконання команд через помилки типу «Переповнення буфера», порушення прав доступу, атаки «відмова в обслуговуванні» та завантаження ворожого змісту (троянські програми, мобільний код Java і ActiveX, віруси) [21, 26].

Несанкціонований доступ до паролю полягає в крадіжці або підборі пароля користувача. Для цієї атаки вразливі будь-які компоненти системи. Починаючи з отримання доступу до паролю користувача з обмеженими правами, зловмисник може намагатися отримати пароль адміністратора.

Несанкціоноване виконання команд часто пов'язане з помилками в операційних системах, що призводить до переповнення вхідного буфера і виконання неправильно сприйнятих даних як команди.

Порушення прав доступу – це найпоширеніше порушення політики безпеки, коли неправильно задані права доступу дозволяють зловмисникам отримати доступ до ресурсів системи.

Атаки типу «відмова в обслуговуванні» призводять до перевантаження системи, блокуючи обробку інших запитів.

Завантаження зловмисного змісту зазвичай включає в себе програми типу «троянські коні», віруси, які наносять шкоду системі, перехоплюють конфіденційні дані або роблять копії конфіденційних файлів.

Для розробки сценаріїв атак на високому рівні необхідно створити модель системи, що враховує ймовірні фактори атак та можливості об'єктів системи. На основі цієї моделі можна виділити потенційні сценарії атак та розробити заходи безпеки.

1.5 Класифікація вразливостей інформаційної безпеки

Уразливості є властивістю об'єктів інформатизації, що відповідають недолікам у функціонуванні, особливостям архітектури автоматизованих систем, протоколам обміну та інтерфейсам, застосовуваним програмним рішенням і апаратній базі, умовам експлуатації та розташуванню, та іншим чинникам [27].

Уразливості можуть присутні в як програмно-апаратних, так і в організаційно-правових аспектах інформаційної безпеки (ІБ). Більшість вразливостей організаційно-правового характеру обумовлена відсутністю нормативних документів в підприємствах, які регулюють питання інформаційної безпеки. Наприклад, відсутність установленої концепції або політики інформаційної безпеки, яка б визначала вимоги до захисту інформаційних систем та шляхи їхнього забезпечення.

Уразливості програмно-апаратного забезпечення можуть присутні в компонентах програмного забезпечення та апаратних пристроях робочих станцій користувачів ІС, серверах, а також у засобах комунікації та каналах зв'язку ІС [28].

Джерела загроз можуть використовувати ці уразливості для порушення безпеки інформації та для незаконних переваг (шкоди власникам, користувачам інформації). Також, можливі несправедливі дії джерел загроз, спрямовані на активізацію цих вразливостей, що може призвести до шкоди.

Існують різні підходи до класифікації вразливостей інформаційних систем та технологій. Визначено, що вразливості безпеки інформації можуть бути [29, 30]:

– об'єктивними;

– суб'єктивними;

– випадковими.

Об'єктивні уразливості залежать від особливостей будови та технічних характеристик використовуваного обладнання на захищеному об'єкті. Повне усунення цих вразливостей неможливе, але їх можна істотно зменшити за допомогою технічних та інженерно-технічних методів парування загроз безпеки інформації.

Суб'єктивні уразливості залежать від дій персоналу і, головним чином, можуть бути вирішені організаційними та програмно-апаратними методами. До них відносяться помилки у підготовці та використанні програмного забезпечення, керування складними системами та експлуатація технічних засобів.

Випадкові уразливості виникають через особливості навколишнього середовища, що оточує об'єкт захисту та неочікувані обставини. Ці фактори зазвичай є не передбачуваними, і усунення їх можливо лише шляхом проведення комплексу організаційних та інженерно-технічних заходів для протидії загрозам інформаційної безпеки. У табл. 1.1 наведено види і характеристики вразливостей відносно етапів життєвого циклу ІС [31].

Таблиця 1.1 Класифікація вразливостей відповідно до етапів життєвого циклу ІС

Етапи життєвого циклу ІС	**Категорія вразливості**	**Виявлення**	**Усунення**
Проєктування	Вразливості проєктування	Трудомісткий і довгий процес	Трудомісткий і довгий процес. Іноді усунення неможливе
Реалізація	Вразливості реалізації	Відносно важко і довго	Не складно, але відносно довго
Експлуатація	Вразливості конфігурації	Легко та швидко	Легко та швидко

Уразливості проєктування. Цей тип уразливостей є найбільш серйозним і складним для усунення, оскільки вони властиві основним елементам проєкту та алгоритмам реалізації базових функцій. Як правило, уразливості проєктування виникають як наслідок недооцінки вимог безпеки при постановці завдань.

Уразливості реалізації. Ця категорія уразливостей з'являється на етапі реалізації в програмному або апаратному забезпеченні коректного з погляду безпеки проєкту. Виявляються і усуваються подібного роду уразливості відносно нескладно, оскільки відповідні зміни в програмному або апаратному забезпеченні, як правило, не зачіпають принципових елементів системи.

Уразливості конфігурації. Цей вид, поряд з уразливостями реалізації, є найпоширенішою категорією вразливостей. Локалізувати і усунути такі уразливості зазвичай найпростіше. Проблема полягає лише в тому, щоб визначити сама наявність уразливості конфігурації. У кращому випадку це відбувається на етапі тестування системи, в гіршому – індикатором наявності уразливості конфігурації є успішно проведена атака. Типовим джерелом вразливостей конфігурації є широко застосовувана більшістю користувачів установка програмного забезпечення зі значеннями «за замовчуванням».

Уразливості проєктування є найбільш серйозними, оскільки вони можуть бути використані для порушення основних сервісів інформаційної безпеки: цілісності, конфіденційності та доступності інформації. Уразливості реалізації та конфігурації також є важливими, але їх усунення зазвичай є відносно простим.

1.6 Збитки від реалізації атак на інформаційні системи

Важливою характеристикою при розгляді завдання забезпечення інформаційної безпеки є величина збитку, заподіяного інформаційних ресурсів в результаті реалізації загрози (атаки). Для опису збитку представляється доцільним виділити значимість і тип цілі, на яку націлена та чи інша атаки, і визначити ступінь досягнення цієї мети [32].

Прийнято виділяти чотири рівні значимості інформації [33]:

1. Життєво важлива – незамінна інформація, наявність якої необхідно для функціонування організації.

2. Важлива – інформація, яка може бути замінена або відновлена, але процес відновлення дуже важкий і вимагає великих витрат.

3. Корисна – інформація, яка корисна і яку важко відновити, однак організація може функціонувати і без неї.

4. Неістотна – інформація, яка практично не потрібна організації.

Мету атаки можна класифікувати за порушення основних характеристик безпеки (цілісності, конфіденційності, доступності) [34], а ступінь її досягнення може являти собою, наприклад, кількісний або якісний показник, що характеризує погіршення рівня основних сервісів безпеки. Загроза, як випливає з визначення, це небезпека заподіяння шкоди. У цьому визначенні виявляється жорсткий зв'язок технічних проблем з юридичною категорією, якою є «збиток».

Прояви можливого збитку можуть бути різні [35, 36]:

– моральний і матеріальний збиток діловій репутації організації;

– моральний, фізичний чи матеріальний збиток, пов'язаний з розголошенням персональних даних окремих осіб;

– матеріальний (фінансовий) збиток від розголошення захищається (конфіденційної) інформації;

– матеріальний (фінансовий) збиток від необхідності відновлення порушених захищаються інформаційних ресурсів;

– матеріальний збиток (втрати) від неможливості виконання взятих на себе зобов'язань перед третьою стороною;

– моральний і матеріальний збиток від дезорганізації діяльності організації;

– матеріальний і моральний збиток від порушення міжнародних відносин.

Збиток інформаційних ресурсів пов'язаний з реалізацією таких основних загроз безпеки інформації [37]:

1. З точки зору забезпечення конфіденційності:

- розкрадання (копіювання) інформації і засобів її обробки;
- втрата (ненавмисна втрата, витік) інформації і засобів її обробки;

2. З точки зору забезпечення доступності:

- блокування інформації;
- знищення інформації і засобів її обробки;

3. З точки зору забезпечення цілісності:

- модифікація (спотворення) інформації;
- заперечення автентичності інформації;
- нав'язування неправдивої інформації.

Збиток від реалізації загроз інформаційної безпеки може бути значним і мати різні наслідки. Тому важливим завданням є розробка та реалізація ефективних заходів щодо запобігання та мінімізації ризиків успішних атак на інформаційну систему.

1.7 Аналіз вимог до безпеки (надійності та готовності) інформаційних систем з вебкомпонентами

Готовність – властивість об'єкта бути в змозі виконувати потрібну функцію при заданих умовах в даний момент часу або протягом заданого інтервалу часу за умови забезпечення необхідними зовнішніми ресурсами [38]. Готовність (availability) є однією з властивостей, які складають поняття «надійність», «безпека» і «гарантоздатність» (dependability).

Надійність (англ. reliability) – властивість технічних об'єктів зберігати у часі в установлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання та транспортування [39]. Під технічними

об'єктами розуміють пристрої, прилади, механізми, машини, комплекси обладнання, будівельні конструкції і споруди, технологічні операції і процеси, системи зв'язку, інформаційні системи, автоматизовані системи управління технологічними процесами тощо.

Гарантоздатність (англ. dependability) – властивість системи гарантувати виконання покладених на неї функцій [40]. Характеризує ступінь працездатності системи на будь-якому інтервалі часу функціонування за умови її справності в початковий момент. Ця властивість має на увазі високі показники таких характеристик системи як безвідмовність, відмовостійкість, готовність, безпека, обслуговуваність, спостережливість та інших важливих показників систем критичного застосування.

Забезпечення готовності здійснюється за рахунок підвищення [41, 42]:

− безвідмовності (reliability) – властивості обладнання (виробу, системи) безупинно зберігати працездатний стан в заданих умовах експлуатації протягом деякого проміжку часу або аж до виконання певного обсягу роботи, без вимушених перерв. Безвідмовність характеризує надійність виробу і визначається набором показників, що обираються з врахуванням виду виробу та умов його експлуатації;

− ремонтопридатності (maintainability) – властивості об'єкта бути пристосованим до підтримання та відновлення стану, в якому він здатний виконувати потрібні функції за допомогою технічного обслуговування та ремонту;

− забезпеченості технічного обслуговування і ремонту (maintenance support).

Вимоги до надійності та доступності ІС визначені рядом міжнародних та національних стандартів [4, 43], а також документами, що регулюють цю сферу [6, 8]. Важливо зауважити, що для оцінки продуктивності інформаційних систем, які працюють безперервно, більш прийнятною є властивість доступності, а не тільки надійності. Адже доступність системи залежить від певного сполучення характеристик, таких як безвідмовність, можливість ремонту та забезпеченість технічного обслуговування та відновлення [9]. Коефіцієнт доступності виступає як складений показник надійності і дорівнює відношенню загального часу, коли

система працює без перебоїв, до загального часу її функціонування з урахуванням відмов ($T_{ПР} + T_{НПР}$):

$$K_I(t) = T_{ПР} / (T_{ПР} + T_{НПР}). \quad (1.1)$$

У літературних джерелах можлива вказівка значень готовності в дробовому або процентному вигляді [10, 11], або як рівень готовності, або на жаргоні – кількість дев'яток після коми (табл. 1.2).

Для майданчиків, де розгорнуті вебкомпоненти інформаційних систем (хостів), використовується спеціальний термін – «аптайм», який вимірюється у відсотках [12]. Найоптимальніший випадок хостингу, коли аптайм становить 100%, це означає, що хостинг був доступний цілодобово протягом усього року; 99,9% – такий показник аптайму позначає, що сервіс був недоступний близько 45 хвилин щомісяця. Загальні стандарти для українських хостингів переважно коливаються від 95,72% до 99,63% аптайму.

Таблиця 1.2 – Вимоги до готовності інформаційної системи

HAL	Рівень готовності (%)	$T_{простоїв}$ 1/год	Характеристика системи
1	90,0%	876 годин	Необслуговувана, некерована (unmanaged)
2	99,0%	87 годин 36 хвилин	Обслуговується, керована (managed)
3	99,9%	8 годин 46 хвилин	Добре обслуговується, добре керована (well-managed)
4	99,99%	52 хвилин 33 секунд	Стійка до відмов (fault-tolerant)
5	99,999%	5 хвилин 15 секунд	З високою готовністю (high-availability)

У деяких випадках [13] для оцінки готовності інформаційних систем використовується шкала HAL (High Availability Level), що базується на показнику загальної тривалості простоїв системи протягом певного періоду експлуатації (наприклад, за рік). Визначені значення допустимої тривалості простоїв системи за шкалою HAL подані у таблиці 1.2. Особливу увагу приділяють вимогам до комерційних інформаційних систем, що вимагають ймовірність безвідмовної роботи на рівні не нижче другого рівня (HAL-2) [1].

Висновки до розділу 1

У першому розділі виконано аналіз зовнішніх факторів впливу на інформаційні системи, серед яких акцентовано увагу на зловмисних діях та атаках. Зовнішні загрози інформаційній безпеці розділено на дві категорії: випадкові та навмисні. Випадкові загрози, такі як природні лиха та аварії, є найбільш поширеними і можуть призвести до найбільших втрат інформації. Навмисні загрози, такі як шпигунство та диверсії, також можуть мати серйозні наслідки, але їх частіше можна запобігти або мінімізувати за допомогою належних заходів безпеки.

Порушення безпеки інформаційних систем можуть призвести до значних матеріальних і моральних збитків. Тому важливою є розробка і реалізація заходів щодо забезпечення інформаційної безпеки.

Потенційні порушники можуть бути як внутрішніми, так і зовнішніми. Внутрішніми порушниками можуть бути співробітники організації, які мають доступ до інформації. Зовнішніми порушниками можуть бути особи, які не мають доступу до інформації, але мають намір отримати його незаконно.

Для визначення потенційних загроз інформаційній безпеці необхідно провести аналіз вразливостей інформаційних систем. Уразливості можуть бути об'єктивними, суб'єктивними або випадковими. Розрахунок збитків від реалізації атак на інформаційні системи є складним завданням. При цьому необхідно враховувати значимість інформації, тип цілі, на яку націлена атака, і ступінь досягнення цієї мети.

До інформаційних систем, які працюють безперервно, більш прийнятною є властивість доступності (стосовно показників надійності та безпеки). Коефіцієнт доступності виступає як комплексний показник і дорівнює відношенню загального часу, коли система працює без простоїв, до загального часу її функціонування з урахуванням відмов.

Важливо зазначити, що забезпечення інформаційної безпеки є комплексним завданням, яке потребує комплексного підходу. Для цього необхідно розробити і

реалізувати ефективні заходи щодо виявлення, усунення вразливостей інформаційних систем, а також захисту від зовнішніх і внутрішніх загроз.

На основі проведеного аналізу сформульовані загальне завдання, яке полягає у розробці та дослідженні моделі доступності віртуалізованої інфраструктури інформаційної системи з врахуванням атак на http сервер. Загальне завдання розділено на три часткові задачі, дві з яких розглянуті у наступних розділах.

РОЗДІЛ 2
АНАЛІЗ ІНТЕНСИВНОСТІ ТА КРИТИЧНОСТІ АТАК НА КОМПОНЕНТИ ВІРТУАЛІЗОВАНОЇ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Загальні відомості про досліджувані компоненти інформаційної системи

Далі буде розглянуто структуру інформаційної системи, яка складається з двох базових компонентів: HTTP сервера Apache та сервера баз даних SQL, зокрема СУБД MySQL. У цьому розділі буде розглянуто загальні характеристики вразливостей вебсерверів та окремо кожного компонента.

Apache є вільним вебсервером, що функціонує на різних операційних системах: Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Його перевагами вважають надійність і можливість гнучкої конфігурації. Він дозволяє підключати зовнішні модулі для отримання даних, використання СУБД для користувачів, змінювати повідомлення про помилки тощо, а також підтримує IPv6.

Apache був розроблений наприкінці 1995 року і походить від гумористичного «a patchy» (англ. «латочка»), оскільки виправляв помилки популярного тоді сервера NCSA HTTPd 1.3. Починаючи з версії 2.x, сервер був переписаний повністю і більше не містить коду NCSA. Наразі розробка відбувається у версії 2.4, а в 1.3 і 2.0 видаються лише виправлення безпеки. На даний момент останні версії – 2.4.58 (19 жовтня 2023 року) для 2.4 гілки та 2.2.27 (26 березня 2014 року) для 2.2 гілки.

Вебсервер Apache розробляється та підтримується відкритим співтовариством розробників під егідою Apache Software Foundation та використовується в багатьох програмних продуктах, таких як Oracle та IBM WebSphere.

З квітня 1996 року до сьогодення він залишається найпопулярнішим HTTP-сервером в Інтернеті. Згідно зі статистикою Netcraft:

- у серпні 2007 року використовувався на 51% всіх вебсерверів;
- у травні 2009 року – на 46%;

- у січні 2011 року – на 59%, що складає більше 160 млн. сайтів;
- у листопаді 2023 року – на 22.74%, що складає більше 248 млн. сайтів.

Основні можливості ядра Apache включають обробку конфігураційних файлів, протокол HTTP та систему завантаження модулів. Ядро повністю розробляється Apache Software Foundation і не включає сторонніх програмістів, відмінно від модулів. Теоретично, ядро Apache може функціонувати без модулів, але це суттєво обмежує його можливості. Ядро повністю написане на мові програмування C. У Apache реалізовані різні механізми забезпечення безпеки та обмеження доступу до даних.

У деяких MPM-модулях є можливість запуску кожного процесу Apache за різними uid і gid відповідно до користувачів і груп користувачів. Також існує механізм suexec для запуску скриптів і CGI-додатків з правами користувача.

Для шифрування переданих даних між клієнтом і сервером використовується механізм SSL, що реалізований через бібліотеку OpenSSL. Для перевірки вебсервера використовуються сертифікати X.509. Існують зовнішні засоби забезпечення безпеки, наприклад, mod_security.

MySQL є вільною реляційною системою управління базами даних. Розробкою та підтримкою цієї системи займається корпорація Oracle, яка отримала права на торговельну марку після придбання компанії Sun Microsystems. Продукт розповсюджується за ліцензією GNU General Public License або за власною комерційною ліцензією. Крім того, розробники можуть створювати функціональність за замовленням ліцензійних користувачів, що призвело до появи механізму реплікації у попередніх версіях.

MySQL використовується переважно для малих і середніх додатків та входить до складу серверів WAMP, AppServ, LAMP, а також в портативні збірки серверів, таких як Денвер і XAMPP. Зазвичай його використовують як сервер, до якого звертаються локальні або віддалені клієнти, але він також постачається з бібліотекою внутрішнього сервера, що дозволяє включати MySQL в автономні програми.

Гнучкість MySQL досягається завдяки підтримці різних типів таблиць, включаючи MyISAM, які підтримують повнотекстовий пошук, та InnoDB, які забезпечують транзакції на рівні окремих записів. Також у MySQL є таблиці типу EXAMPLE, що демонструють принципи створення нових типів таблиць. Завдяки відкритій архітектурі та ліцензуванню GPL, в системі постійно з'являються нові типи таблиць.

Спільнотою розробників було створено різні відгалуження коду, такі як Drizzle, OurDelta, Percona Server і MariaDB. Вони були створені ще до придбання компанією Oracle компанії Sun.

MySQL підтримується на великому спектрі платформ, включаючи AIX, FreeBSD, Linux, Mac OS X, Windows, і існує навіть порт MySQL для OpenVMS. Офіційний сайт СУБД пропонує вихідні коди, а також готові виконувані модулі MySQL, оптимізовані для різних операційних систем.

2.2 Системи оцінювання вразливостей та їх характеристики

Наразі існують розбіжності визначення терміна «уразливість» на рівні стандартів (табл. 2.1). Деякі стандарти визначають уразливість виключно як програмний дефект. Проте це не є універсальним правилом, оскільки є випадки програмних дефектів, які ніколи не стають уразливими (або вони недоступні для зловмисників, або виявлені й усунені під час тестування, інформація про які не фіксується в репозитаріях вразливостей). Також є вразливості, які не можна віднести до програмних дефектів, такі як слабкі паролі та помилки конфігурації системи. Різниця між уразливостями та програмними дефектами може бути проілюстрована на наведеній нижче ілюстрації (рис. 2.1).

Починаючи з 1999 року зусиллями компанії MITRE Corporation (http://www.mitre.org) впроваджуються незалежні від різних виробників стандарти і засоби ідентифікації та обліку вразливостей, атак, конфігурацій і інших елементів інформаційної безпеки.

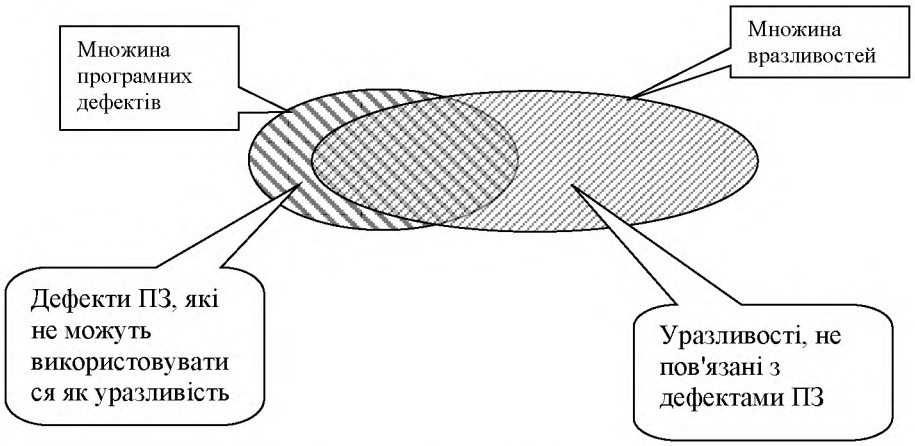

Рисунок 2.1 – Перетин множин вразливостей і дефектів ПЗ

Таблиця 2.1 – Стандартні визначення вразливості інформаційних систем

Стандарт	**Вразливість (vulnerability)**	**Слабке місце (weakness)**
X.800, X.1500, X.1521	Будь-яке слабке місце, яке може бути використане для порушення системи або інформації, яка в ній міститься.	Недолік або дефект, який, хоча і не зізнається сам по собі в якості уразливості, міг би в якийсь момент стати вразливістю або міг би сприяти привнесення інших вразливостей.
X.1520	Будь дефект програмного забезпечення, який може бути використаний для порушення цілісності системи або міститься в цій системи інформації.	
X.1524	Будь-яке слабке місце в програмному забезпеченні, яке може бути використане для порушення системи або міститься в ній інформації.	Дефект або вада в коді, проєктуванні, архітектури або розгортанні програмного забезпечення, здатний в певний момент стати вразливістю або приводити до виникнення інших вразливостей.

У розробці CVE, крім фахівців MITRE, взяли участь представники компаній ISS, Cisco, BindView, Axent, NFR, L-3, CyberSafe, CERT, а також вчені з університетів Carnegie Mellon, SANS, UC Davis Computer Security Lab, CERIAS та інші, за підтримки від Internet Security Systems, Cisco, Axent, BindView, IBM та інших компаній. У цій роботі використовується переклад стандартів та термінів відповідно до рекомендацій МСЕ (ITU-T) серії X-15xx, що були введені у 2012 році.

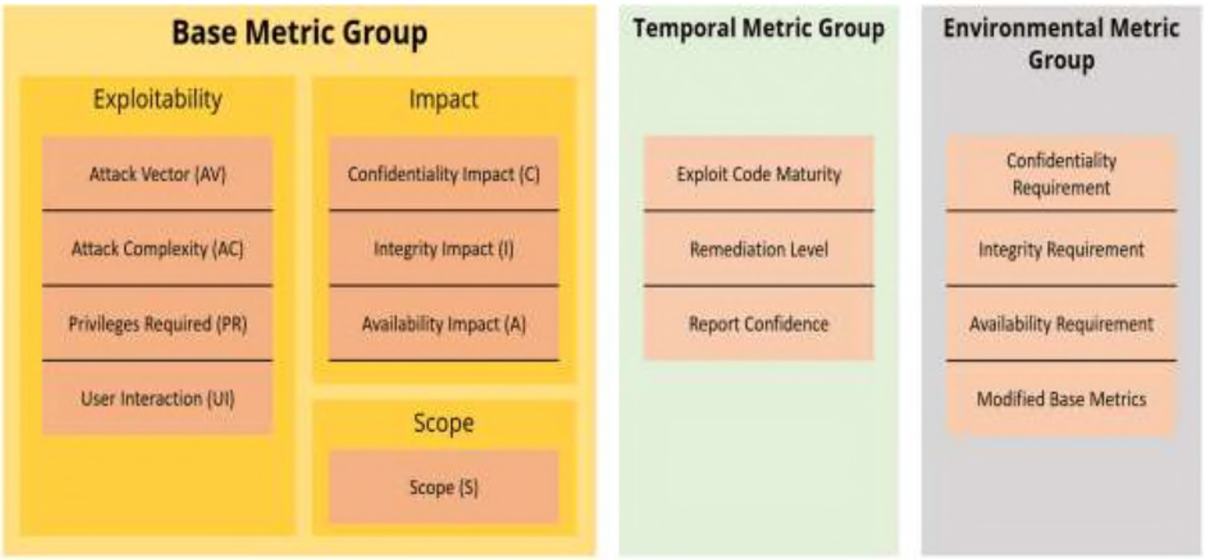

Рисунок 2.2 – Групи показників системи CVSS, які впливають на доступність прямим і непрямим чином

На сьогодні існують кілька систем класифікації вразливостей (уразливостей), які активно використовуються в освітніх і технологічних процесах. Одна з найвідоміших і повніших – CVE (Common Vulnerabilities and Exposures), керівником якої є компанія NCSD (National Cyber Security Division). Повна база CVE розміщена на сервері Національної Бази уразливостей США (NVD – nvd.nist.gov) або на офіційному вебсайті (cve.mitre.org/data/downloads). На кінець 2023 року база CVE включала 294982 записи.

Кожна уразливість в базі CVE має, як мінімум, чотири атрибути:

– унікальний ідентифікатор (наприклад, CVE-2013-0125);

– коротка загальна інформація про уразливість та продукти, що її мають (Summary);

– дата публікації (Published);

– рівень серйозності (CVSS Severity),

А також інші атрибути, які можна отримати, деталізуючи конкретну уразливість на окремій вебсторінці. Серед них «Vulnerability Type», що використовує класифікатор зі словника CWE (Common Weakness Enumeration), і

базова метрика (CVSS Base Score) – параметр, що розраховується за спеціальною формулою на основі експертних оцінок окремих метрик AccessComplexity, Authentication, AccessVector, ConfImpact, IntegImpact, AvailImpact. Крім базової, також використовуються комплексні метрики CVSS Temporal Equation (TemporalScore) і CVSS Environmental Equation (EnvironmentalScore).

Слід відзначити, що метрика, пов'язана з доступністю (AvailImpact, AvailabilityImpact), приймає значення «none» (0), «partial» (0.275) і «complete» (0.66).

Виходячи з класифікаторів вразливостей бази NVD, можна виділити підмножини вразливостей за типом програм та за впливом на доступність. Також очевидно, що не всі уразливості можуть гарантовано (з 100% вірогідністю) вивести систему з ладу (із доступу). Тут полягає одна з істотних відмінностей уразливостей від внутрішньосистемних дефектів і помилок, які забезпечують виведення системи (або її компоненту) з ладу з 100% вірогідністю.

Наразі діє кілька десятків різних репозитаріїв уразливостей, які в основному презентовані у вигляді інтерактивних вебсервісів. При виборі конкретного репозитарію були враховані такі критерії:

– повнота (обсяг та кількість вразливостей);

– доступність інформації (безкоштовна основа даних);

– зручність отримання даних (пристосовані інтерфейси);

– підтримка оцінки вразливостей за системою CVSS.

Наразі на офіційному сайті cve.mitre.org зареєстровано 41 репозитарій. Серед них найбільшим за обсягом та частотою оновлень вважається nvd.nist.gov. До його переваг можна віднести:

– вбудовані засоби пошуку та фільтрації бази;

– доступність всієї бази у форматах XML та JSON;

– безкоштовний доступ до даних.

Серед недоліків цієї бази слід зазначити лише базовий набір оцінок, обмежене охоплення CWE та обмежену функціональність розширеного пошуку.

2.3 Вибір оцінок вразливостей з системи CVSS

Оскільки розглядаються моделі доступності, основний інтерес спрямований на оцінку впливу уразливості на доступність. Ця оцінка представлена у двох основних вимірах (рис. 2.3, а):

– група «Базові показники» має показник «вплив на доступність» (Availability impact, A), який визначається як «Відсутній» (None, N), «Частковий» (Partial, P) або «Повний» (Complete, C);

– група «Показники середовища» включає показник «Вимога доступності AR», який може мати три рівні: «низький» (low), «середній» (medium) або «високий» (high).

Для оцінки та вибору наборів уразливостей важливо враховувати інші показники (рис. 2.3, б):

– група «Базові показники» має показник «вектор доступу» (Access Vector, AV), який описується як «Локальний» (Local, L), «Сусідня мережа» (Adjacent network, A) або «Мережевий» (Network, N);

– група «Часові показники» має показник «можливість експлуатації» (Exploitability, E), який визначається як «Неперевірена» (Unproven, U), «Доведено правильність концепції» (Proof-of-Concept, POC), «Функціональна» (Functional, F), «Висока» (High, H) або «Не визначено» (Not Defined, ND).

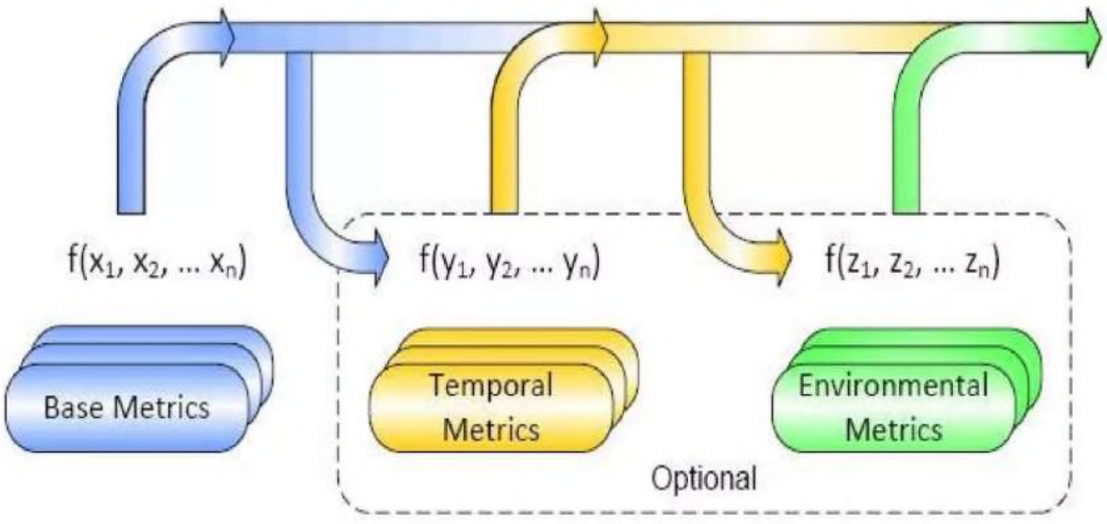

Рисунок 2.3 – Послідовність розрахунку і обов'язковість оцінок CVSS

Оцінка «Базова формула» (BaseScore) має значення в діапазоні від 0 до 10 і приблизно класифікується як «Низький» (0 ... 4), «Середній» (4 ... 7) або «Високий» (7 ... 10).

Важливо зауважити, що далеко не всі репозиторії надають інформацію по групам «Часові показники» і «Показники середовища». Тим часом, як зазначається в самому стандарті CVSS, часова оцінка не перевищує базову і не більше ніж на 33% менше її; а оцінка середовища, що знаходиться в межах від 0 до 10, не перевищує часову оцінку. Тобто при початкових розрахунках можна обмежиться тільки базовими показниками (рис. 2.3).

2.4 Критерії формування вибірки вразливостей, небезпечних для доступності вебкомпонент інформаційних систем

Оцінюючи особливості галузі досліджень, яка стосується комерційних інформаційних систем, були встановлені такі обмеження для створення вибірок вразливостей:

– показник «вектор доступу» – лише значення «Мережевий» (Network, N);

– показник «вплив на доступність» (Availability impact, A) – лише значення «Часткове» (Partial, P) та «Повне» (Complete, C).

Додатковим критерієм фільтрації вибірки є вебкомпоненти, в яких очікується виявлення уразливості, тобто сервер Apache та сервер баз даних MySQL.

Щодо інструментів та послідовності формування вибірки: для аналізу невеликої кількості вразливостей (до 50) можна скористатися розширеним пошуком на сайті бази даних. Однак цей пошук розділений на сторінки по 20 записів на кожній, що ускладнює отримання та обробку великої кількості даних і робить непрактичною їхню ручну обробку для вирішення поставлених завдань. Тому було обрано більш прийнятний варіант обробки бази вразливостей у вигляді XML-документів.

Таблиця 2.2 – Підмножина вразливостей доступності http сервера Apache в період 01.2023 – 10.2023

№з/п	name	published	base_score	CVSS_vector
1	CVE-2023-0002	2023-03-21	7,5	(AV:N/AC:L/Au:N/C:P/I:P/A:P)
2	CVE-2023-0003	2023-03-21	7,5	(AV:N/AC:L/Au:N/C:P/I:P/A:P)
3	CVE-2023-0098	2023-03-18	5,0	(AV:N/AC:L/Au:N/C:N/I:N/A:P)
4	CVE-2023-0032	2023-03-16	4,3	(AV:N/AC:M/Au:N/C:N/I:N/A:P)
5	CVE-2023-0107	2023-04-15	7,5	(AV:N/AC:L/Au:N/C:P/I:P/A:P)
6	CVE-2023-0050	2023-04-01	5,0	(AV:N/AC:L/Au:N/C:N/I:N/A:P)
7	CVE-2023-0075	2023-05-31	5,0	(AV:N/AC:L/Au:N/C:N/I:N/A:P)
8	CVE-2023-0109	2023-05-08	4,3	(AV:N/AC:M/Au:N/C:N/I:N/A:P)
9	CVE-2023-0110	2023-05-08	4,3	(AV:N/AC:M/Au:N/C:N/I:N/A:P)
10	CVE-2023-0095	2023-05-31	5,0	(AV:N/AC:L/Au:N/C:N/I:N/A:P)
11	CVE-2023-0111	2023-04-17	6,5	(AV:N/AC:L/Au:S/C:P/I:P/A:P)
12	CVE-2023-0112	2023-04-29	7,5	(AV:N/AC:L/Au:N/C:P/I:P/A:P)
13	CVE-2023-0113	2023-04-29	7,5	(AV:N/AC:L/Au:N/C:P/I:P/A:P)
14	CVE-2023-0114	2023-04-30	7,5	(AV:N/AC:L/Au:N/C:P/I:P/A:P)
15	CVE-2023-0116	2023-05-08	5,8	(AV:N/AC:M/Au:N/C:N/I:P/A:P)
16	CVE-2023-1882	2023-03-02	7,5	(AV:N/AC:L/Au:N/C:P/I:P/A:P)
17	CVE-2023-2668	2023-03-28	5,0	(AV:N/AC:L/Au:N/C:N/I:N/A:P)

Для цього потрібно завантажити XML-документи з відповідного року на вебсайті. Для дослідження були вибрані дані, які містяться в розділі «NVD / CVE XML Feed with CVSS and CPE mappings (version 1.2)».

Таблиця 2.3 – Підмножина вразливостей доступності служби SQL в період 01.2023-10.2023

№з/п	name	published	base_score	CVSS_vector
1	CVE-2023-0384	2023-04-15	4.0	(AV:N/AC:L/Au:S/C:N/I:N/A:P)
2	CVE-2023-0386	2023-01-15	4.0	(AV:N/AC:L/Au:S/C:N/I:N/A:P)
3	CVE-2023-0401	2023-01-15	4.0	(AV:N/AC:L/Au:S/C:N/I:N/A:P)
4	CVE-2023-0402	2023-01-15	4.0	(AV:N/AC:L/Au:S/C:N/I:N/A:P)
5	CVE-2023-0412	2023-01-15	4.0	(AV:N/AC:L/Au:S/C:N/I:N/A:P)
6	CVE-2023-0420	2023-01-15	2.8	(AV:N/AC:M/Au:M/C:N/I:N/A:P)
7	CVE-2023-0437	2023-01-15	3.5	(AV:N/AC:M/Au:S/C:N/I:N/A:P)
8	CVE-2023-2419	2023-04-15	4.0	(AV:N/AC:L/Au:S/C:N/I:N/A:P)
9	CVE-2023-2430	2023-04-15	3.5	(AV:N/AC:M/Au:S/C:N/I:N/A:P)
10	CVE-2023-2431	2023-04-15	2.6	(AV:N/AC:H/Au:N/C:N/I:N/A:P)
11	CVE-2023-2432	2023-04-15	2.8	(AV:N/AC:M/Au:M/C:N/I:N/A:P)
12	CVE-2023-2436	2023-04-15	6.0	(AV:N/AC:M/Au:S/C:P/I:P/A:P)
13	CVE-2023-2438	2023-04-15	3.5	(AV:N/AC:M/Au:S/C:N/I:N/A:P)
14	CVE-2023-2440	2023-04-15	5.1	(AV:N/AC:H/Au:N/C:P/I:P/A:P)

Отримані з сайту XML – документи були оброблені за допомогою табличного редактора MS Exel та редактора XML – XML Marker 2.2.

Після відкриття документа в редакторі у шпальтах було встановлено умови фільтрації:

- CVSS_vector – містить – AV: N, A: C і A: P;
- Ns1: descript – містить – Apache (приклад для вивчення атак на Apache).

Отримані підмножини вразливостей після фільтрації потребують фіксації параметрів «published» і «CVSS_base_score» (табл. 2.2 і 2.3). Важливо зауважити, що для утворення підгруп вразливостей для конкретного елемента HTTP сервера краще використовувати відповідні ідентифікатори CWE, а не шукати за ключовими словами (наприклад, DNS). Але наразі репозиторій nvd.nist.gov не містить повних ідентифікаторів CWE та не включає зв'язок між вразливостями та CWE у XML-документах. Тому цей метод утворення підгруп вразливостей залишається придатним лише після оновлення бази даних NVD.

2.5 Модель розрахунку інтенсивності атак на компоненти інформаційної системи

Отримані результати вказують на те, що відкритий репозиторій містить дані, які дозволяють створювати підмножини вразливостей, що впливають на доступність елементів вебсервера в залежності від їх критичності. Одержані підмножини також включають час публікації інформації (Published). Оскільки інформація про уразливості може потрапляти до бази як від експертів із безпеки, так і після їх використання зловмисниками, час реєстрації уразливості відображає ступінь зацікавленості конкретним елементом інформаційної системи дослідниками. Очевидно, що інтерес до застарілих версій програмних продуктів поступово згасає.

На основі аналізу припущено, що інтенсивність атак на доступність конкретної вебкомпоненти ІС можна визначити, враховуючи максимальне

значення усередненої за рік частоти фіксації інформації про вразливості, що можуть бути використані для подібних атак. При цьому критичність атаки визначається як середнє значення базової оцінки CVSS. Послідовність розрахунку інтенсивності атак на доступність наступна:

1. Створення структурної схеми ІС з урахуванням послідовно-паралельних зв'язків елементів, що впливають на доступність (аналогічно структурі надійності).

2. Формування підмножин вразливостей, що впливають на доступність компонент ІС.

3. Визначення значень усередненої частоти фіксації інформації про вразливості в утворених підмножинах за рік.

4. Визначення інтенсивності атак на доступність компонент ІС як максимального значення усередненої частоти фіксації інформації про вразливості.

5. Визначення критичності атаки як середнього значення базової оцінки CVSS для вразливостей обраної підмножини за рік.

За даними з таблиці 2.2, середня зацікавленість уразливістю доступності Apache у 2023 році склала $1{,}9 \cdot 10^{-4}$ (1/год), а середня критичність атак становила 6,04. Щодо MySQL, показники такі: середня зацікавленість уразливістю: $1{,}5 \cdot 10^{-4}$ (1/год), середня критичність атак: 3,84.

2.6 Методи відновлення функціонування інформаційної системи після атаки

Операції відновлення інформаційних систем після успішної атаки часто пов'язують з випуском та встановленням «патча» або коригуванням. Проте не всі атаки повністю заблоковують роботу системи, і не кожен сервер припиняє функціонування після вторгнення. Процеси відновлення можуть включати наступні етапи:

– повторне встановлення складових програмного забезпечення;

– поновлення середовища, в якому працює ПЗ;

– переінсталяція програми;

– перезапуск середовища роботи;

– перезапуск http серверів;

– перезапуск додатків;

– відновлення збережених даних;

– Відновлення інформації сесій.

Додатково до цього можуть включатися такі дії:

– встановлення «патча» для програмного забезпечення;

– встановлення нової версії програмного забезпечення.

У даному дослідженні не розглядається сценарій усунення причин атаки за умови задоволення вимог зловмисників, не враховуючи при цьому усунення вразливостей.

2.7 Вплив багатопотоковості на рівні архітектури вебкомпонент інформаційної системи

Завжди однією з основних викликів для вебархітекторів була багатопотоковість. З початком ери вебсервісів ця потреба зросла. Сучасні популярні сайти здатні обслуговувати сотні тисяч або навіть мільйони користувачів одночасно, і це стало нормою. Колись багатопотоковість була необхідна для роботи з повільними ADSL або dial-up-підключеннями. Тепер це вимагається для роботи з мобільними пристроями та новими архітектурами додатків, які потребують постійного та швидкого з'єднання – клієнти очікують миттєвих оновлень у Twitter, новин або соціальних мереж.

Ще одним важливим аспектом, що впливає на багатопотоковість, є зміна поведінки браузерів, які відкривають до шести одночасних з'єднань для прискорення завантаження сайту. Уявіть простий вебсервер Apache, який генерує короткі відповіді в 100 Кбайт – зазвичай це текст або зображення. На генерацію

сторінки може піти декілька секунд, але для передачі клієнту потрібно 10 секунд, якщо швидкість передачі 80 кбіт/с. Вебсервер може швидко «витягнути» 100 кілобайт контенту, але потім буде повільно передавати їх клієнту. При зростаючому навантаженні, тисячі клієнтів одночасно запитують ту саму інформацію. Якщо для кожного клієнта потрібно додатковий мегабайт пам'яті, для відправки 1000 клієнтам 100 кілобайт контенту знадобиться 1 гігабайт пам'яті.

При постійних з'єднаннях проблема обробки багатопотоковості стає ще більш складною, оскільки клієнти залишаються підключеними, щоб уникнути затримок, пов'язаних з встановленням нового HTTP-з'єднання.

Для обробки збільшеного навантаження, пов'язаного з розширенням аудиторії Інтернету та підвищенням рівня багатопотоковості, основа функціональності сайту повинна складатися з дуже ефективних компонентів. Тут важливі всі елементи – апаратні ресурси (CPU, пам'ять, диски), мережеві можливості, архітектура програм та сховищ даних. Проте саме вебсервер приймає та обробляє клієнтські запити, тому йому необхідне нелінійне масштабування зі зростанням числа одночасних з'єднань та оброблених запитів на секунду.

2.8 Проблеми архітектури http – сервера Apache

Apache залишається одним з найпопулярніших вебсерверів. Його початки сягають початку 1990-х, коли його створено під наявне програмне забезпечення та обладнання, що існувало в той час, та рівень розвитку Інтернету. Стандартно вебсайт розміщувався на окремому фізичному сервері, де працював лише один екземпляр Apache. Однак наприкінці 2000-х стало очевидно, що така модель з одним фізичним сервером неефективно справляється з потребами швидкозростаючих вебсервісів. Хоча Apache виявився корисним для розвитку,

початково він був створений для створення копій сервера для кожного нового з'єднання, що в сучасних умовах унеможливлює потрібну масштабованість.

Apache має потужну екосистему сторонніх сервісів, що надає розробникам різноманітні інструменти для створення додатків. Проте за цими перевагами ховається недолік – робота з великою кількістю інструментів для одного програмного продукту обмежує можливості масштабування.

Традиційні підходи до обробки одночасних з'єднань, засновані на роботі з потоками або процесами, виконують обробку кожного з'єднання окремим процесом чи потоком, що може призводити до блокування операцій введення/виводу та невикористання ресурсів процесора та пам'яті. Це може призвести до низької продуктивності через надмірне перемикання контексту та інші проблеми. Ці аспекти особливо помітні у вебсерверах старого типу, таких як Apache.

2.9 Огляд архітектури вебсервера nginx

З самого початку свого існування, основною метою nginx було досягнення високої продуктивності та ефективного використання серверних ресурсів, одночасно забезпечуючи можливість динамічного масштабування вебсайтів. Це привело до розробки асинхронної, модульної та орієнтованої на події архітектури nginx.

Nginx активно використовує мультиплексування та нотифікацію подій, розподіляючи конкретні завдання між окремими процесами. Підключення обробляються шляхом ефективного циклу виконання за допомогою певної кількості однопотокових процесів (воркерів). Кожен воркер nginx в змозі обробляти багато тисяч одночасних з'єднань та запитів за секунду. Високорівневу структуру архітектури nginx можна побачити на рисунку 2.4.

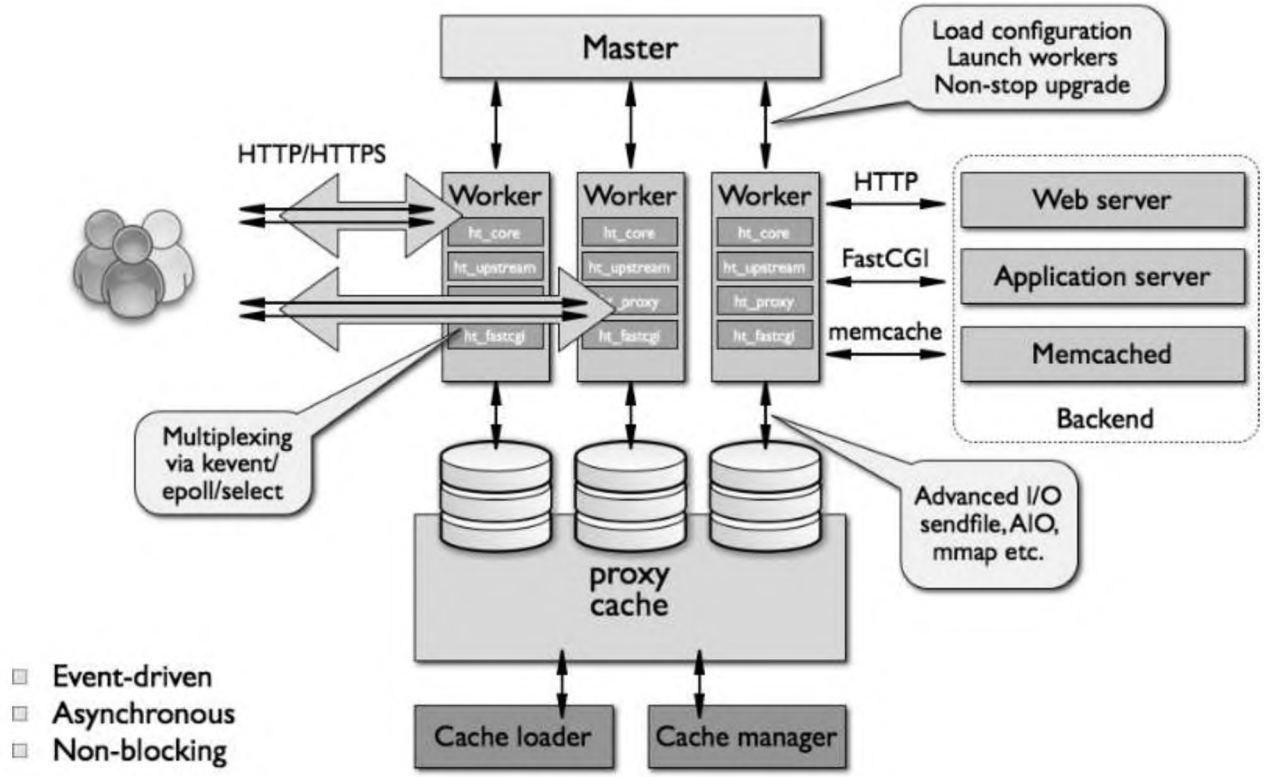

Рисунок 2.4 – Архітектура вебкомпоненти ІС на базі серверу nginx

Worker в nginx складається з ядра та функціональних модулів. Ядро відповідає за утримання циклу виконання і виконання відповідних частин коду модулів на кожному етапі обробки процесу. Модулі, зазвичай, надають основну функціональність рівню додатків. Вони також здійснюють читання та запис в мережу та сховище, трансформацію контенту, проведення вихідної фільтрації і, у разі роботи в режимі проксі, передачу запитів вищим серверам.

Модульна архітектура nginx дає змогу розробникам розширювати функціональність вебсервера без необхідності зміни основного коду. Всередині nginx існує кілька видів модулів – ядерні, подієві, обробники фаз, протоколи, фільтри, балансувальники навантаження, обробники змінних тощо. Проте nginx не підтримує динамічне завантаження модулів – вони компілюються разом з ядром під час створення збірки. Розробники мають намір додати можливість завантажувати модулі динамічно в майбутньому.

Для керування мережевими з'єднаннями, обробки вхідних та вихідних даних, nginx використовує механізми нотифікацій та ряд механізмів оптимізації обробки введення/виводу на диску в операційних системах Linux, Solaris і BSD, таких як kqueue, epoll і event ports.

2.10 Модель роботи worker-процесів

Масштабованість та ефективність ресурсів сервера – ключові аспекти, які nginx досягає завдяки своїй архітектурі. Не створюючи окремі процеси чи потоки для кожного з'єднання, nginx використовує спеціальний worker для прийому нових запитів з загального сокета. Кожен worker-процес запускає високоефективний цикл обробки, що дозволяє обслуговувати тисячі з'єднань.

Спеціальних механізмів розподілу з'єднань між різними worker-процесами немає – це завдання виконується в ядрі операційної системи. У процесі завантаження створюється набір сокетів, а потім worker постійно обробляє їх в процесі обробки HTTP-запитів і відповідей.

Одна з найважчих частин коду worker nginx – це цикл виконання. Він включає різноманітні внутрішні виклики та активно використовує концепцію асинхронної обробки завдань. Ця асинхронна обробка реалізована через модульність, попередження подій, колбек-функції і розширені таймери з метою уникнення блокувань.

Nginx ефективно використовує ресурси пам'яті та цикли процесора, оскільки він уникає постійного створення і знищення процесів і потоків. Механізм worker надійно контролює стан мережі і сховища, додаючи нові з'єднання в цикл обробки і асинхронно обробляючи їх до завершення, після чого з'єднання деактивується.

Nginx демонструє ефективне використання багатоядерних систем: кожен worker-процес створюється для кожного ядра, уникаючи блокувань та трешіння потоків. Такий підхід дозволяє більш ефективно використовувати фізичні пристрої зберігання та рівномірно розподіляти завантаження між worker-процесами.

Для оптимальної продуктивності nginx може налаштовуватися для різних сценаріїв використання процесора та диска. У разі великої обробки TCP/IP-з'єднань чи використання SSL кількість worker-процесів має відповідати кількості ядер. Якщо ж головне навантаження падає на дискову систему, кількість worker-процесів може бути більшою, щоб оптимізувати завантаження і вивантаження контенту.

У наступних версіях nginx планується розв'язати проблему блокування дискового введення/виводу, а також поширити підтримку вбудованих скриптів, для уникнення залежностей від роботи worker-процесів.

2.11 Ролі процесів nginx

Nginx запускає кілька процесів у пам'яті: один master-процес і декілька worker-процесів, також працюють кілька службових процесів, таких як менеджер і завантажувач кеша. У версіях nginx 1.x всі процеси однопоточні та використовують механізми поділу пам'яті для взаємодії один з одним.

Master-процес відповідає за кілька завдань:

- читання та валідація конфігурації;
- створення, зв'язування і закриття сокетів;
- запуск, припинення та підтримка заданої кількості worker-процесів;
- реконфігурація без переривання роботи сервісу;

– керування постійними оновленнями (запуск нових «бінарників» і відкат до попередньої версії, якщо необхідно);

– повторне відкриття лог-файлів;

– компіляція вбудованих Perl-скриптів.

Worker-процеси відповідають за обробку з'єднань, отриманих від клієнтів, а також надають функціональність reverse proxy і фільтрації, а також виконують майже всі функції, що повинен робити nginx. Зазвичай для відстеження поточного стану вебсервера системному адміністратору потрібно спостерігати за worker-процесами, оскільки саме вони найкраще відображають його стан.

Процес завантажувача кеша відповідає за перевірку та оновлення елементів кешу на диску та у пам'яті. Він готує екземпляри nginx до роботи з вже збереженими на диску файлами, оновлює метадані контенту в кеші, а потім завершує роботу.

Менеджер кешу контролює актуальність кешу. При нормальному функціонуванні вебсервера він залишається в пам'яті, а в разі збою його перезапускає master-процес.

Висновки до розділу 2

У розділі було розглянуто методи аналізу вразливостей, що впливають на доступність вебкомпонентів інформаційних систем. Було показано, що відкритий репозиторій CVE містить дані, які дозволяють формувати підмножини вразливостей, що впливають на доступність елементів вебсервера в залежності від їх критичності. Одержані підмножини також включають час публікації інформації (Published). Оскільки інформація про уразливості може потрапляти до бази як від експертів із безпеки, так і після їх використання зловмисниками, час реєстрації уразливості відображає ступінь зацікавленості конкретним елементом інформаційної системи дослідниками.

На основі аналізу було припущено, що інтенсивність атак на доступність конкретної вебкомпоненти ІС можна визначити, враховуючи максимальне значення усередненої за рік частоти фіксації інформації про вразливості, що можуть бути використані для подібних атак. При цьому критичність атаки визначається величиною базової оцінки CVSS.

Для перевірки припущення було сформовано підмножини вразливостей для Apache і MySQL за період 2023 року. Було показано, що максимальне значення усередненої за рік частоти фіксації інформації про вразливості, що можуть бути використані для атак на доступність, істотно відрізняється для різних елементів HTTP сервера. Так, для HTTP сервера Apache максимальне значення становить 1,5 вразливості на місяць, а для бази даних MySQL – 0,5 вразливостей на місяць.

Відповідно до отриманого припущення, інтенсивність атак на доступність Apache вище, ніж інтенсивність атак на доступність MySQL. Цей результат підтверджується даними про використання вразливостей для атак на ці компоненти. Так, у 2023 році було зареєстровано 15 атак на Apache, пов’язаних з використанням вразливостей, а на MySQL – 5 атак.

Отримані результати свідчать про те, що запропонований метод розрахунку інтенсивності атак на доступність вебкомпонентів ІС дозволяє оцінити ризики, пов’язані з використанням застарілих версій програмних продуктів.

РОЗДІЛ 3
РОЗРОБЛЕННЯ ТА ДОСЛІДЖЕННЯ МОДЕЛІ ВІРТУАЛІЗОВАНОЇ ІНФРАСТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ З ВРАХУВАННЯМ АТАК НА HTTP СЕРВЕР

3.1 Розробка структурної схеми надійності інформаційної системи

Складна багаторівнева віртуалізована інформаційна система може бути представлена за допомогою схем різного рівня вкладеності (рис. 3.1) [42].

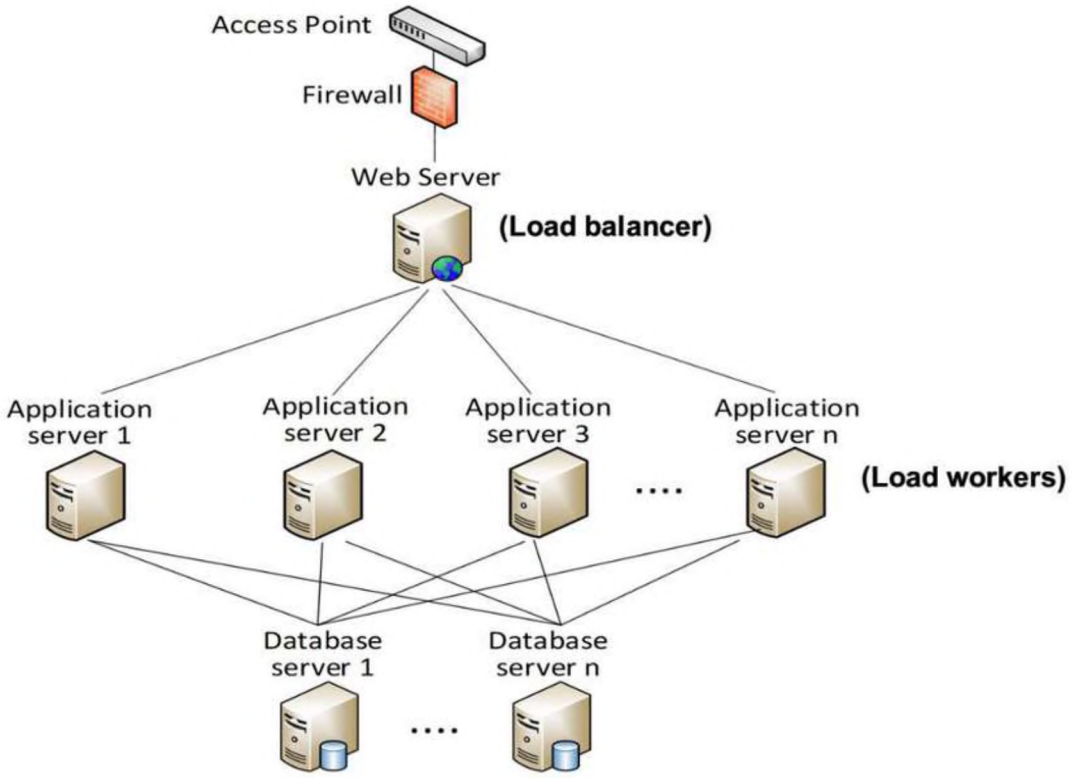

Рисунок 3.1 – Взаємодія вебкомпонент віртуалізованої інформаційної системи

У цьому дослідженні було вирішено обмежитися на початковому етапі структурною схемою готовності (надійності) інформаційних систем (ІС), що складається з двох основних елементів. Ця схема описує взаємодію двох ключових компонентів архітектури: сервісу nginx FrontEnd та сервісу Apache BackEnd. Таке рішення прийнято з урахуванням класифікації CVE, яка визначає підмножини вразливостей в цих сервісах.

Недоступність будь-якого з цих сервісів може спричинити відмову у обслуговуванні клієнтів. Отже, інформаційна система буде складатися з двох послідовних елементів, кожен з яких відображатиме стан працездатності зазначених сервісів (рис. 3.2).

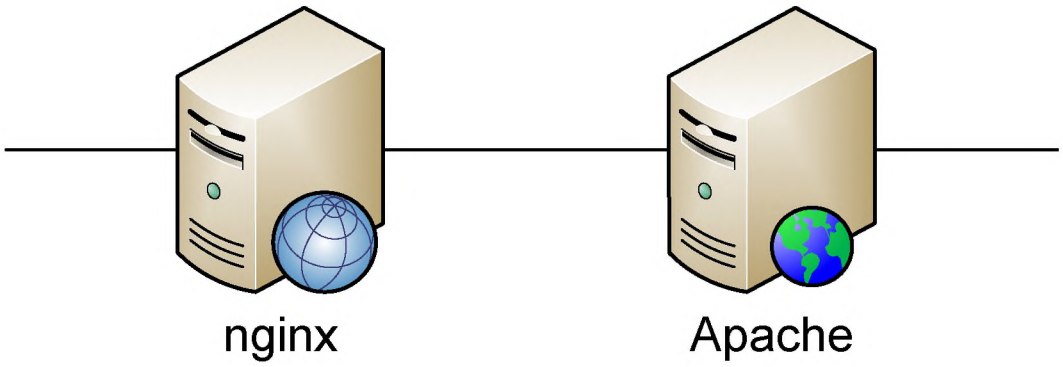

Рисунок 3.2 – Структурна схема готовності (надійності) вебкомпонент ІС

Зрозуміло, що цю інформаційну систему можна уточнити, оскільки кожен сервіс, насамперед, має структуру клієнт-сервера з розподіленою архітектурою (і, відповідно, властиві йому відмови та відновлення як для клієнтської, так і для серверної частини), а також базується на апаратно-програмних комплексах обслуговування (і, відповідно, характерні для неї відмови апаратних та програмних засобів). Проте ці аспекти виходять за рамки дослідження даної роботи.

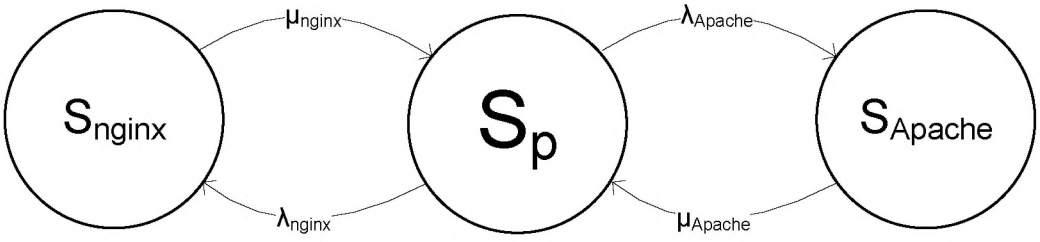

Рисунок 3.3 – Розмічений граф функціонування віртуалізованої ІС

Граф станів і переходів з урахуванням інформаційної системи включатиме один стан, що працює коректно, та три стани непрацездатності (рис. 3.3). Пізніше цей граф розглядається як частина структурної моделі з урахуванням дефектів вебкомпонент.

Система диференціальних рівнянь Колмогорова, що відповідає зазначеному на рис. 3.3 графу, має наступний вигляд:

$$\begin{cases} dP_0(t)/dt = -(\lambda_1 + \lambda_2)P_0(t) + \mu_1 P_1(t) + \mu_2 P_2(t), \\ dP_1(t)/dt = -\mu_1 P_1(t) + \lambda_1 P_0(t), \\ dP_2(t)/dt = -\mu_2 P_2(t) + \lambda_2 P_0(t), \\ P_0(t) + P_1(t) + P_2(t) = 1. \end{cases} \quad (3.1)$$

Функція готовності визначається, як сума вірогідності знаходження системи в працездатних станах P_0 :

$$A(t) = P_0(t). \quad (3.2)$$

Використання методу Маркова накладає певні обмеження на визначення причин відмов у вебкомпонентах. У загальному випадку втрата працездатності ІС пояснюється виявленням різноманітних дефектів, що мають різні причини [43]. Цей набір можна розподілити на підгрупи за різними характеристиками, наприклад, за етапом прояву відмов розрізняють конструктивні та експлуатаційні дефекти.

Доцільно розділити моделі оновлень для усунення дефектів на дві групи. Кожна з них враховує виправлення дефектів у конфігурації (без втручання у програмний код вебкомпонент) і у програмному коді мережевих служб (із зміною параметра потоку відмов ПЗ для відповідної компоненти). В якості основної вебкомпоненти, яка часто стикається з дефектами, обрано службу Apache.

Таблиця 3.1 містить значення вхідних параметрів для моделей віртуалізованої ІС. Оскільки розглянуті моделі є програмними конструкціями середовища Matlab, вона містить символьні позначення вхідних параметрів у латинській транскрипції.

Таблиця 3.1 – Значення вхідних параметрів моделей віртуалізованої ІС

№з/п	Name	Matlab-name	Часовий період	Значення	Од.вим.
1.	Інтенсивність прояву дефектів ПЗ служби nginx (початкова)	langinx	3,8 років	3e-5	1/год
2.	Інтенсивність прояву дефектів ПЗ служби Apache	laapache	10 днів	0.0042	1/год
3.	Інтенсивність відновлення служби nginx	munginx	1,5 години	0.67	1/год
4.	Інтенсивність відновлення служби Apache	muapache	0,5 години	2	1/год
5.	Зміна інтенсивності прояву дефектів ПЗ служби Apache, обумовлене усуненням дефекту програмного коду	deltalapache		4e-4	1/год
6.	Інтенсивність розробки оновлень ПЗ, в яких усуваються дефекти доступності	lapath	6 місяців	2.28e-4	1/ год
7.	Інтенсивність відновлення служби після установки патча (або оновлень ПЗ з усуненням дефектів)	mupath	2 години	0.5	1/ год
8.	Кількість дефектів	nv		3…30	

Використання марківського моделювання систем з відмовами та відновленням, а також методів багатофрагментного моделювання передбачає проведення таких кроків [41]:

1. Створення діаграми станів і переходів ІС (графа станів), що базується на структурній схемі готовності системи [42].

2. Визначення вхідних параметрів моделі готовності (МГ) ІС з урахуванням відмов і відновлень програмного забезпечення.

3. Розрахунок показників ІС (функцій готовності і коефіцієнта готовності). Для цього потрібно в граф станів системи ввести відповідні параметри МГ, скласти систему лінійних диференціальних рівнянь (СЛДР) за правилом Колмогорова.

Після розв'язання СЛДР чисельним методом [18] та знаходження значень Pi(t), значення шуканого показника готовності визначаються як сума ймовірності перебування системи у всіх працездатних станах.

3.2 Розробка однофрагментної моделі віртуалізованої інформаційної системи без врахування атак на її вебкомпоненту

Для розробки математичної моделі оцінювання віртуалізованої інформаційної системи без врахування атак та проведення оновлень програмного забезпечення необхідно зробити декілька припущень, серед основних з яких важливі наступні:

– потік подій, що переводить систему з одного функціонального стану в інший, має властивості стаціонарності, ординарності і відсутності післядій.

– кожен компонент ІС може перебувати у працездатному або непрацездатному стані в довільний момент часу.

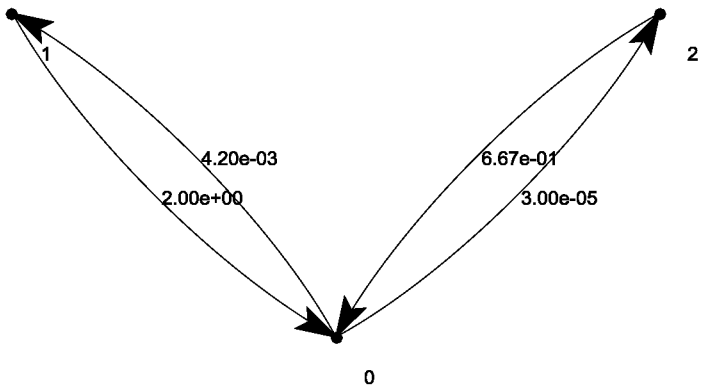

Рисунок 3.4 – Розмічений граф станів і переходів моделі ІС без урахування атак та проведення оновлень ПЗ

Система диференціальних рівнянь Колмогорова не відрізняється від (3.1). Функція готовності визначається, як сума вірогідності знаходження системи в працездатних станах (3.2).

Таблиця 3.2 – Матриця коефіцієнтів СЛДУ Колмогорова

-0,00423000000000000	2	0,666666666666667
0,00420000000000000	-2	0
3,00000000000000e-05	0	-0,666666666666667

Рішення СЛДУ (3.1) методом ode15s для вхідних параметрів дозволило отримати результати, представлені на рис. 3.5.

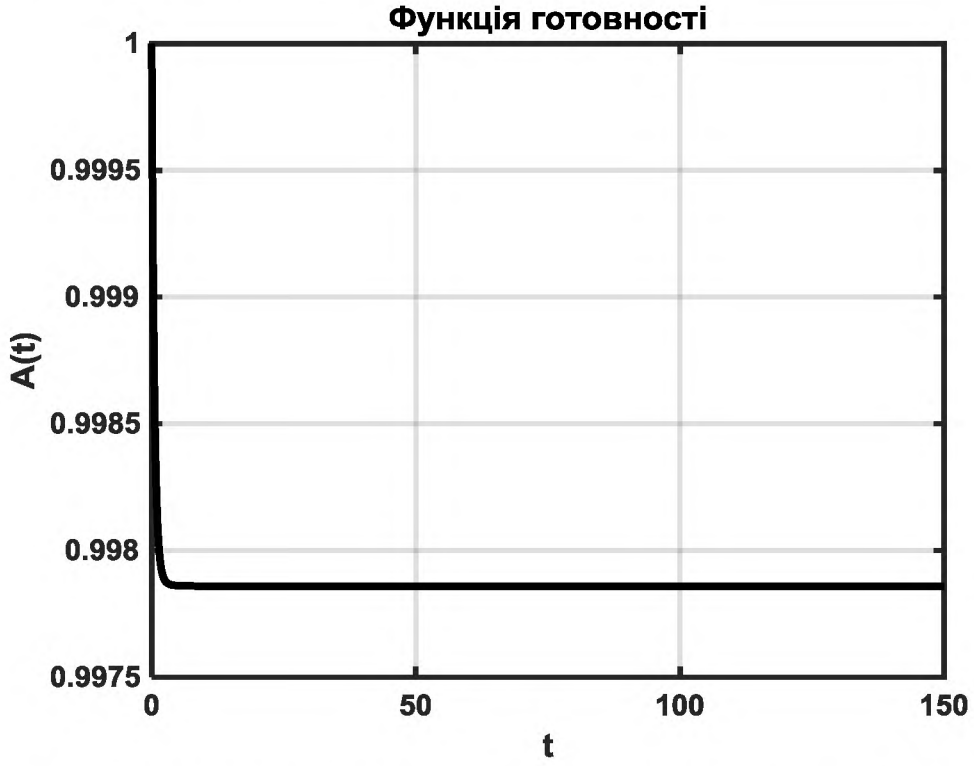

Рисунок 3.5 – Графічна залежність зміни функції готовності ІС без атак та процедур оновлення, отримані за допомогою однофрагментної моделі

Коефіцієнт готовності ІС складе $К_Г$ = 0,9978. Функція готовності, визначена за допомогою ОФМ, досягає стаціонарного стану протягом перших 10 годин роботи.

Розглянемо математичну модель для оцінювання готовності інформаційної системи без врахування проведення оновлень програмного забезпечення та ігноруючи відмови служби Apache (рис. 3.6).

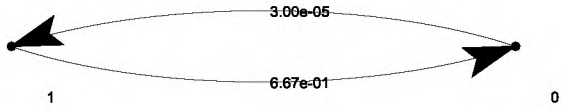

Рисунок 3.6 – Розмічений граф станів і переходів моделі ІС без урахування відмов служби Apache

Система диференціальних рівнянь Колмогорова має вигляд (3.3), функція готовності визначається, як сума вірогідності знаходження системи в працездатних станах (3.2).

$$\begin{cases} dP_0(t)/dt = -\lambda_1 P_0(t) + \mu_1 P_1(t), \\ dP_1(t)/dt = -\mu_1 P_1(t) + \lambda_1 P_0(t), \\ P_0(t) + P_1(t) = 1. \end{cases} \quad (3.3)$$

Таблиця 3.3 – Матриця коефіцієнтів СЛДУ Колмогорова

-0,00055	0,666667	1	0,333333
3,00E-05	-0,66667	0	0
1,50E-05	0	-1	0
0,0005	0	0	-0,33333

Рішення СЛДУ (3.3) методом ode15s дозволило отримати результати, представлені на рис. 3.7.

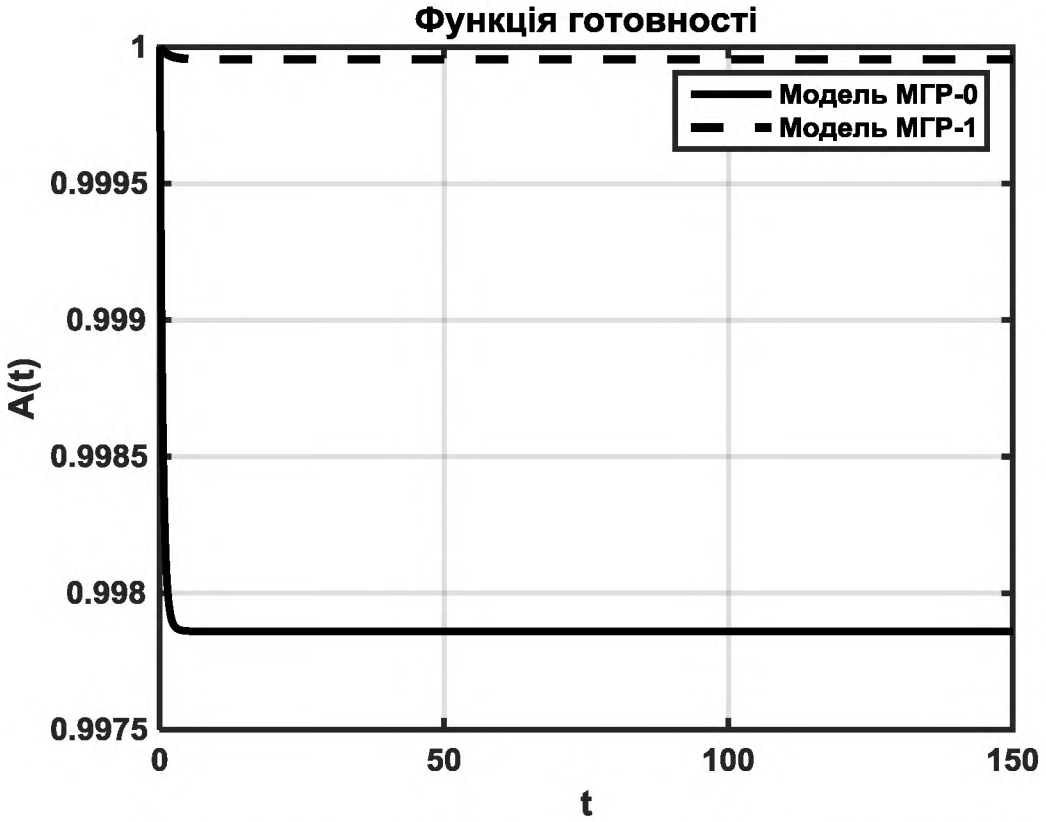

Рисунок 3.7 – Порівняння графіків зміни функції готовності для однофрагментних моделей ІС на часовому проміжку [0…150]

Стаціонарний коефіцієнт готовності ІС буде рівний $K_Г$ = 0,99995. Аналіз графіків на рис. 3.7 дозволяє зробити наступні висновки, що функції готовності, отримані за допомогою ОФМ приймають стаціонарний стан за перші 40 годин роботи; а відмови http сервера Apache знижують рівень готовності на $2*10^{-3}$.

3.3 Розробка та дослідження багатофрагментної моделі віртуалізованої інформаційної системи

Дана модель описує функціонування віртуалізованої інформаційної системи в умовах зміни параметра потоку відмов http сервера Apache при усуненні дефектів патчерізацією. На відміну від однофрагментних моделей, в багатофрагментній моделі (БФМ) розглядається один варіант переходу в стан патчерізації: після прояву дефекту з інтенсивністю lapath. Програма моделювання представлена в додатку А, а розмічений орграф для системи з трьома оновленнями, побудований з її допомогою – на рис. 3.8.

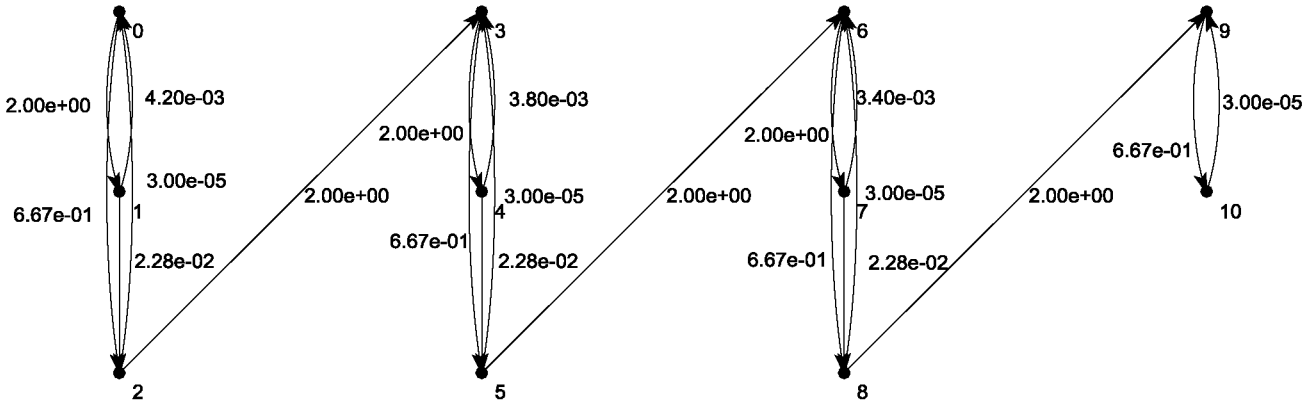

Рисунок 3.8 – Розмічений граф станів і переходів моделі функціонування ІС в умовах усунення дефектів програмного коду розробкою патчів і оновлень ПЗ і перезапуску служби Apache

Згідно графа на рис. 3.8, спочатку ІС функціонує в умовах прояву відмов і відновлення служб Apache і nginx. Після прояву дефекту служби Apache (перехід в

стан «1») система втрачає працездатність. Після прояву дефекту можливий такий варіант розвитку подій – дефект виявляється і система переходить в стан патчеризації «3» і запускається механізм розробки патча.

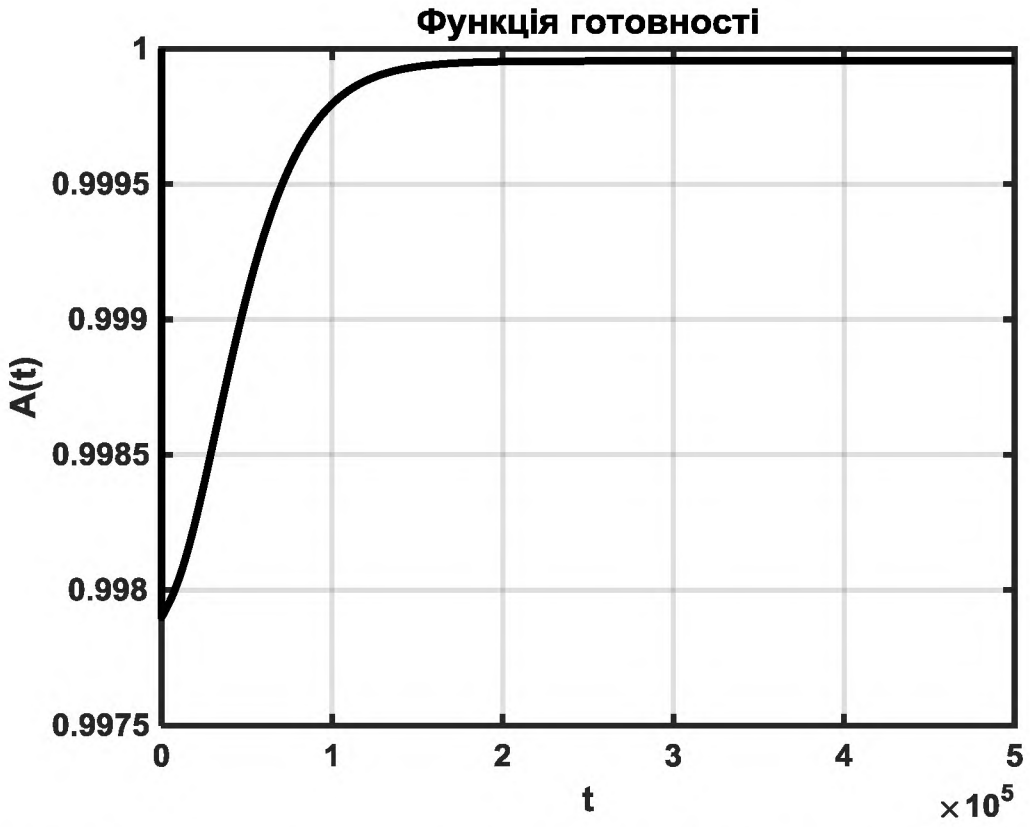

Рисунок 3.9 – Графік зміни функції готовності для БФМ в умовах усунення дефектів програмного коду розробкою патчів і оновлень ПЗ

Після установки патча дефект програмного коду усувається і система переходить у новий стан «4», в якому інтенсивність потоку відмов служби Apache зменшується на $\Delta\lambda_{Apache}$. Після прояву та усунення всіх дефектів система продовжує функціонувати в умовах прояву відмов і відновлення решти служб (стани «9 ... 10»). Рішення СЛДУ Колмогорова було виконано в системі Matlab за допомогою методу ode15s для тимчасового інтервалу [0 ... 500000] годин. На рис. 3.10 представлений графік зміни функції готовності для моделі ІС в умовах усунення дефектів програмного коду розробкою патчів і оновлень ПЗ.

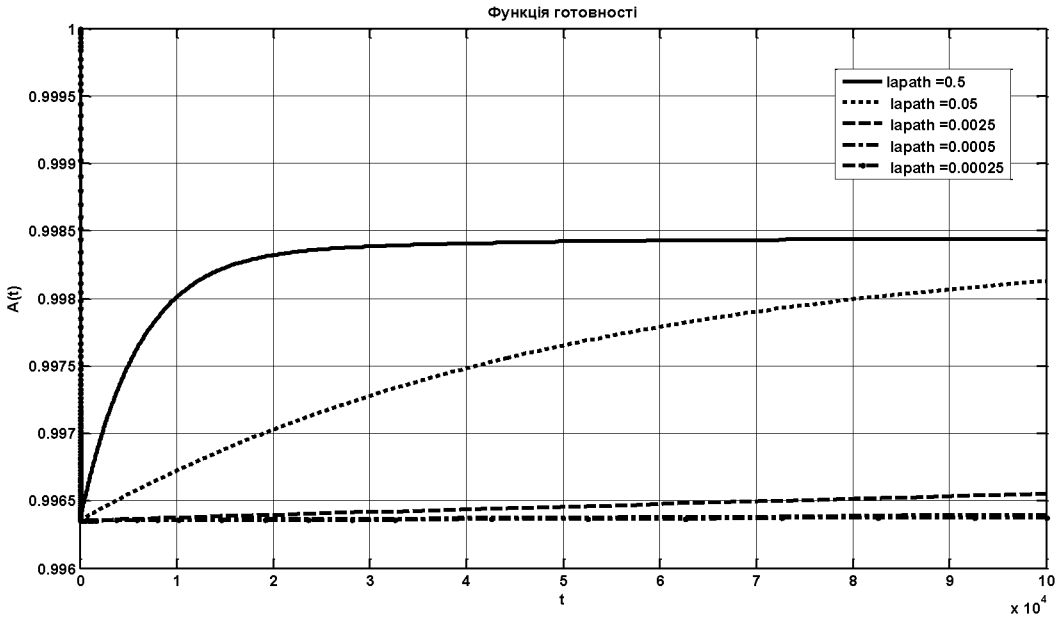

Рисунок 3.10 – Графіки зміни функції готовності моделі ІС при різній інтенсивності розробки оновлень ПЗ на часовому інтервалі [0 ... 100000]

Для моделі функціонування віртуалізованої інформаційної системи було обрано параметр lapath для вивчення його впливу на характер поведінки та значення функції готовності. Для аналізу впливу цього параметра була розроблена спеціальна циклічна програмна конструкція, яка представлена в додатку А. Результати моделювання у вигляді графічних залежностей показані на рис. 3.10.

З графіків на рис. 3.10 видно, що значення параметра lapath має вплив на темп переходу функції готовності у стаціонарний режим. Отже, враховуючи обмеження в технічному завданні на час досягнення стаціонарного коефіцієнта готовності, запропонованою моделлю можна визначити необхідне значення параметра lapath.

3.4 Аналіз технологій віртуалізації програмних засобів

Розробники віртуалізаційних засобів часто пропонують комплексні рішення для віртуалізації інфраструктури. Ці рішення дозволяють підприємствам та малим бізнесам перетворювати, керувати та оптимізувати свою IT-інфраструктуру

завдяки віртуалізації. Вони надають можливості комплексної віртуалізації, управління та оптимізації ресурсів, забезпечують високу доступність програм та пропонують автоматизацію в рамках інтегрованих пропозицій. Наприклад, «VMware Infrastructure» – один із таких інтегрованих пакетів для комплексної віртуалізації.

VMware Infrastructure включає такі компоненти (рис. 3.11) [31]:

– VMware ESX Server: додаток віртуалізації, який працює на фізичних серверах і абстрагує процесор, пам'ять, сховище та мережеві ресурси, надаючи їх декільком віртуальним машинам;

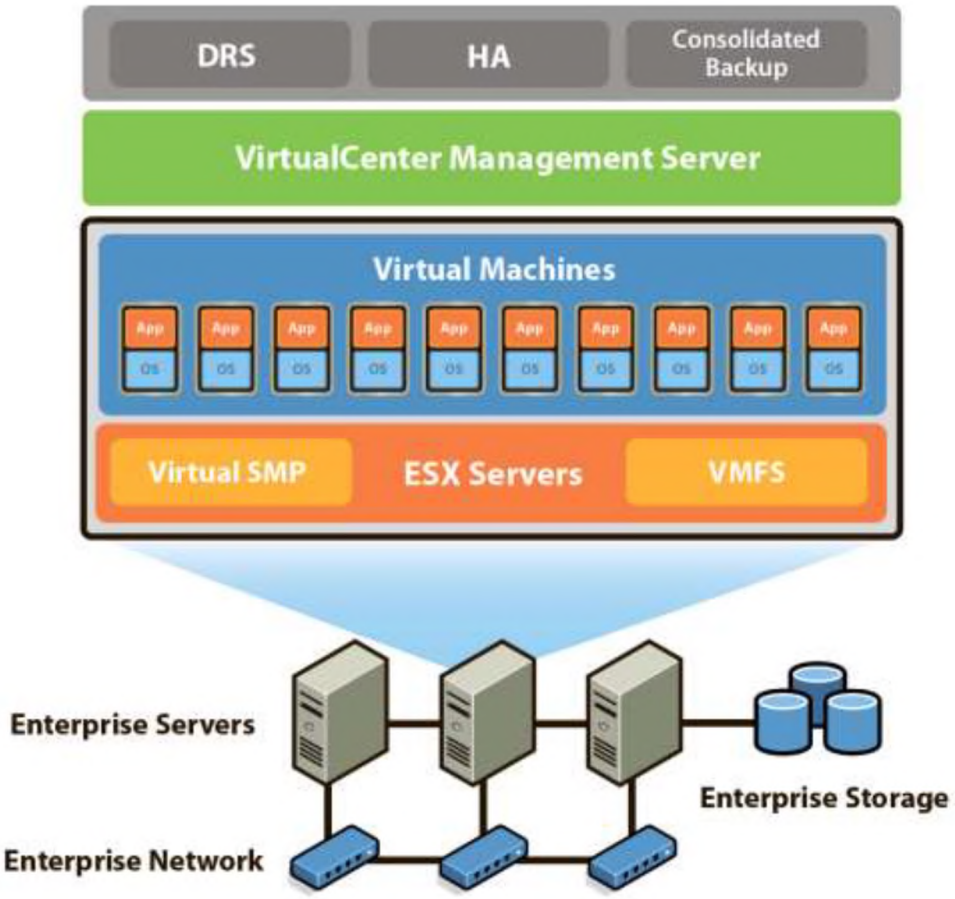

Рисунок 3.11 – Компоненти VMware Infrastructure

– файлова система віртуальних машин VMware (VMFS) – це високопродуктивна кластерна файлова система, призначена для віртуальних машин;

– VMware Virtual Symmetric MultiProcessing (SMP) дозволяє одній віртуальній машині використовувати кілька фізичних процесорів одночасно;

– сервер управління VirtualCenter є центральною точкою для налаштування, забезпечення та керування віртуалізованою IT-інфраструктурою;

– клієнт віртуалізованої інфраструктури (VI Client) – це інтерфейс, який дозволяє адміністраторам та користувачам віддалено підключатися до сервера управління VirtualCenter або окремих установок ESX сервера з будь-якого ПК з Windows.

IT-відділ підприємства за допомогою інфраструктури VMware може створити віртуальний центр обробки даних, використовуючи існуючі стандартні технології та обладнання. Немає потреби придбати спеціалізоване обладнання. Крім того, VMware Infrastructure дозволяє створювати віртуальний центр обробки даних, який централізовано управляється серверами управління і доступний для керування широким спектром інтерфейсів.

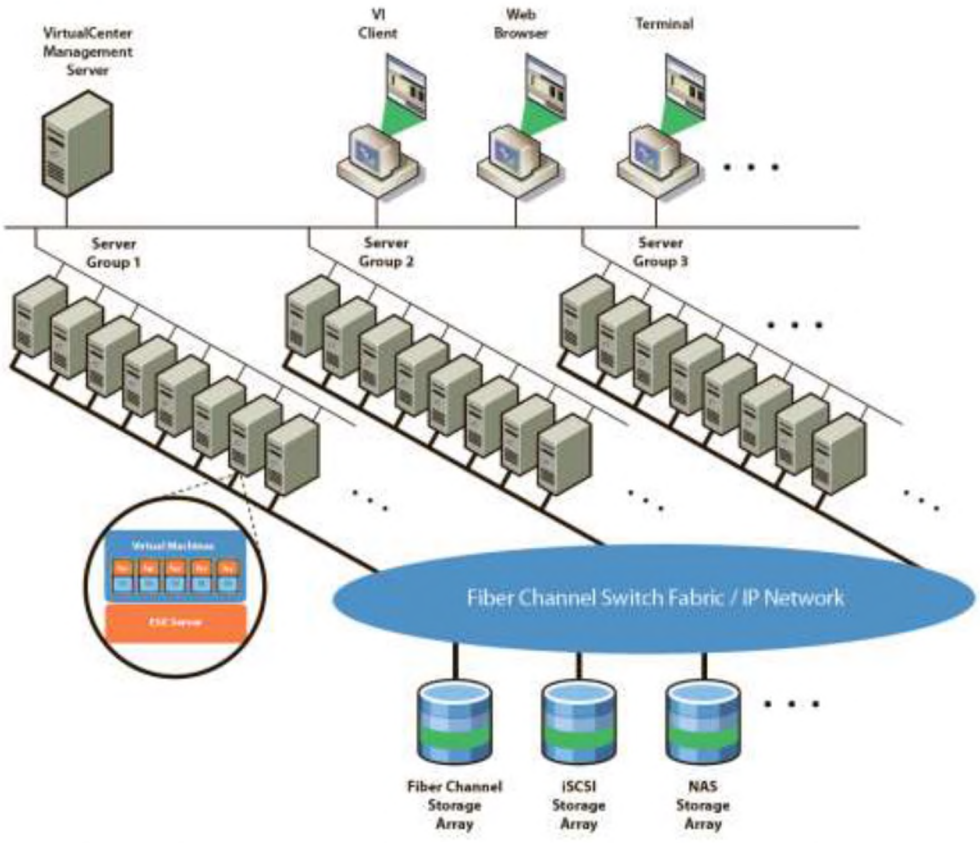

Рисунок 3.12 – Елементи фізичної топології віртуалізованого центру обробки даних VMware Infrastructure [31]

Віртуалізований центр обробки даних VMware Infrastructure складається з таких складових елементів (рис. 3.12): обчислювальні сервери, мережеві масиви зберігання даних, IP-мережі, сервер управління та клієнти для ПК.

Обчислювальні сервери – це стандартні x86 або x64 сервери, що працюють під управлінням ПЗ VMware ESX без основної ОС. Кожен обчислювальний сервер називається автономним хостом у віртуальному середовищі. Ряд таких конфігурованих серверів можна об'єднати та підключити до однієї підсистеми мережі та сховища даних для створення сукупної ресурсної групи у віртуальному середовищі, що називається кластером.

SAN-масиви Fibre Channel, масиви iSCSI SAN та масиви NAS – широко використовувані технології зберігання даних, що підтримуються VMware Infrastructure для відповіді на різні потреби сховищ в ЦОД. Спільне використання цих масивів зберігання даних між групами серверів через мережеві зони зберігання даних дозволяє об'єднувати ресурси зберігання та надавати більшу гнучкість у наданні їх віртуальним машинам.

Кожен обчислювальний сервер може мати кілька гігабітних мережевих інтерфейсних карт Ethernet (NIC), щоб забезпечити високу пропускну здатність і надійну мережу для всього центру обробки даних.

Сервер управління VirtualCenter є зручною єдиною точкою управління центром обробки даних. Він працює на базі операційної системи Windows Server та надає різні важливі сервіси для центру обробки даних, такі як контроль доступу, моніторинг продуктивності та конфігурація. Він узгоджує ресурси з окремих обчислювальних серверів для спільного використання між віртуальними машинами у всьому центрі обробки даних.

Як показано на рис. 3.13, VirtualCenter досягає цього, керуючи призначенням віртуальних машин на обчислювальних серверах. Сервер управління VirtualCenter також управляє призначенням ресурсів віртуальним машинам на даному обчислювальному сервері на основі політик, встановлених системним адміністратором.

Обчислювальні сервери продовжують працювати, навіть якщо сервер управління VirtualCenter недоступний (наприклад, через відмову мережі). Обчислювальними серверами можна керувати окремо, і вони продовжуватимуть працювати з призначеними віртуальними машинами на основі останніх призначень ресурсів. При відновленні доступу до сервера управління VirtualCenter, він зможе знову управляти центром обробки даних в цілому.

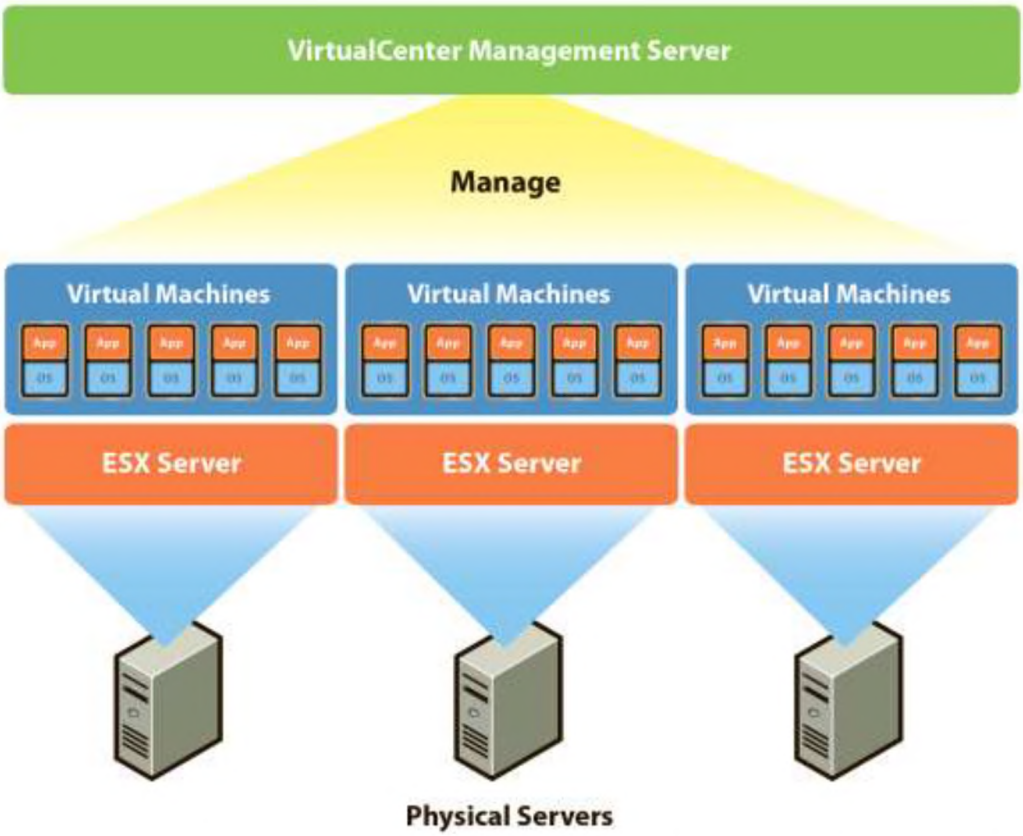

Рисунок 3.13 – Розподіл ВМ на фізичні сервери за допомогою сервера управління VirtualCenter [32]

Інфраструктура VMware пропонує різноманітні інтерфейси для управління центрами обробки даних та доступу до віртуальних машин. Користувачі можуть обирати той інтерфейс, що найбільше відповідає їхнім потребам: через додаток «Клієнт віртуальної інфраструктури» (Клієнт VI), веббраузер або служби терміналів (наприклад, служби терміналів Windows або Xterm).

VMware Infrastructure віртуалізує всю IT-інфраструктуру підприємства, включаючи сервери, сховища та мережі. Це дозволяє агрегувати різноманітні

ресурси та створювати простий та уніфікований набір елементів у віртуальному середовищі. За допомогою VMware Infrastructure можна ефективно керувати IT-ресурсами та динамічно надавати їх різним бізнес-підрозділам та проєктам, не обтяжуючись основними відмінностями та обмеженнями обладнання.

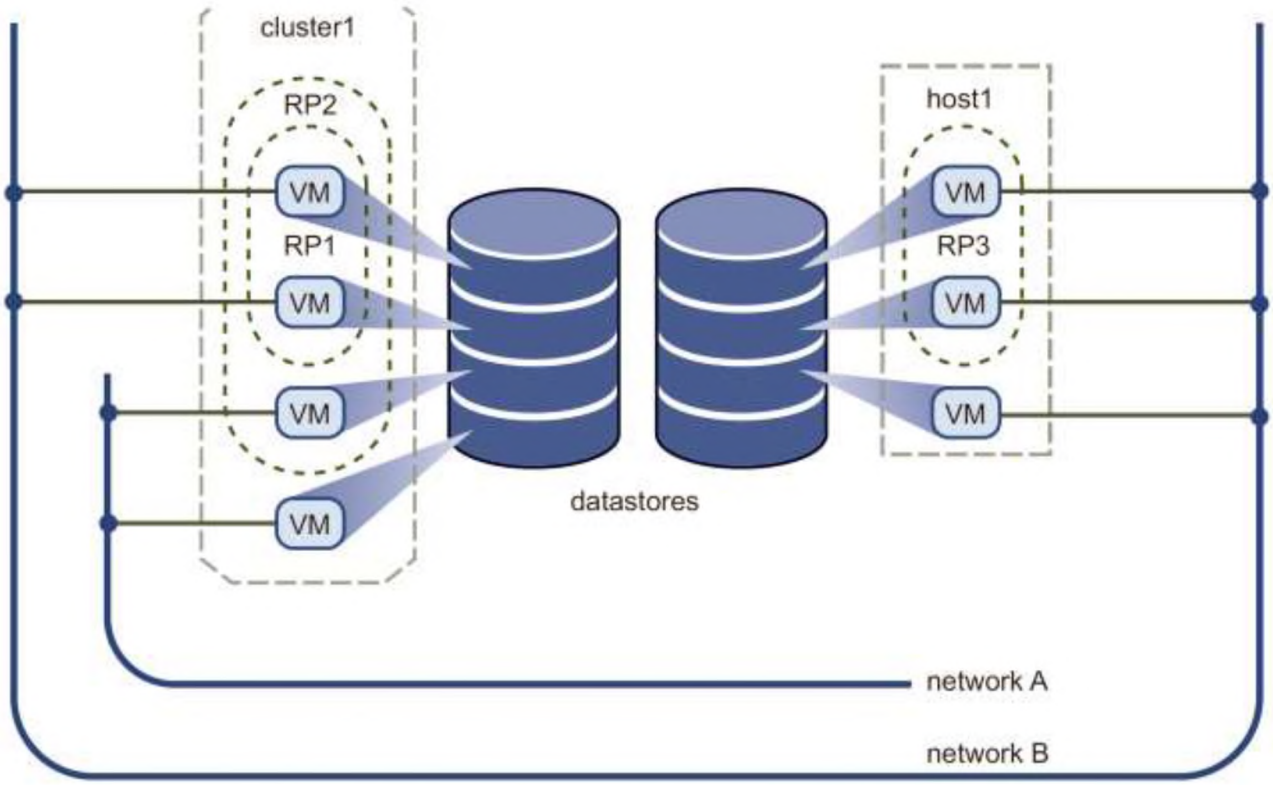

Рисунок 3.14 – Архітектура віртуального центру обробки даних [1]

Як зазначено на рис. 3.14, VMware Infrastructure надає простий набір віртуальних елементів для створення віртуального центру обробки даних:

– обчислювальні ресурси та пам'ять, відомі як хости, кластери та ресурсні пули (Hosts, Clusters, RP);

– сховища даних, які представляють комбінації фізичних ресурсів зберігання;

– мережеві ресурси, що об'єднують віртуальні машини між собою та з фізичною мережею.

Хост – це віртуальне відображення обчислювальних ресурсів та пам'яті фізичної машини, на якій працює ESX Server. Кластер об'єднує фізичні машини для управління та роботи як єдине ціле, утворюючи сукупні обчислювальні ресурси та пам'ять. Сховища даних відображають поєднання основних фізичних ресурсів

зберігання в ЦОД. Ці ресурси можуть бути локальними дисками SCSI сервера, SAN Fibre Channel, iSCSI SAN або мережевими сховищами NAS. Створені віртуальні машини споживають ресурси, подібно фізичним пристроям. Вони можуть динамічно збільшувати або зменшувати використання ресурсів, відповідно до навантаження.

Процес створення віртуальних машин набагато швидший та простіший, порівняно з фізичними. Нові віртуальні машини можна розгорнути за кілька секунд, уникнувши процесів закупівлі обладнання та очікування на інсталяцію ОС та драйверів. Ресурси, які виділяються віртуальним машинам, базуються на політиках, встановлених адміністратором системи. Ці політики можуть резервувати ресурси для конкретної віртуальної машини та встановлювати пріоритети використання ресурсів.

3.5 Розрахунок витрат від функціонування віртуалізованої інформаційної системи

Витрати на створення інформаційної системи включають в себе ряд складових, таких як: витрати на електроенергію, витрати на хостинг (розміщення у мережі інтернет), оплату праці програмістів, витрати на офісні приладдя та матеріали для комп'ютерів, оренду приміщення, амортизацію комп'ютерного обладнання та інші подібні витрати.

Таблиця 3.4 – Споживання електроенергії вебкомпонентами ІС

Найменування	кількість	кВт/год	кВт за добу (приблизно)	кВт в місяць
Комп'ютер (Сервер)	1	0,17	1,53	45,9
Освітлення	3	0,36	9,72	291,6
Спліт	1	0,7	6,3	189
Всього:		1,23		526,5

Розрахунок електроенергії для дев'ятигодинного робочого дня.

Для підприємств 1 кВт/г = 2,28 грн.

В місяць: 2,28*526,5= 1201,89 грн.

У мережі Web-сайт планується розмістити на ресурсах провайдера міста Полтава, що забезпечить зручне обслуговування.

Заробітна плата програмісту становить 14000 грн.

Таблиця 3.5 – Розрахунок щомісячних витрат на підтримку ІС

Найменування	Сума, грн.	ЄСВ, грн.
Зарплата програміста	14000	422,65
Зарплата кур'єра	2000	422,65
Транспортні витрати кур'єра	350	
Електроенергія	1263,6	
Хостинг	200	
Інтернет	100	
Інші витрати	300	
Разом:	8213,6	845,3
Всього витрат:	19058,9	

Rпост = 19058,9 грн. – постійні щомісячні витрати.

Так як приміщення та обладнання вже є в наявності в компанії розрахуємо річну суму амортизаційних відрахувань. Річна сума амортизаційних відрахувань розраховується за формулою:

$$A = \frac{Ф * N_A}{100\%}, \quad (3.4)$$

де Ф – первісна вартість основних фондів за видами, грн .;

NA – норма амортизації за видами основних фондів, у %.

Річну суму амортизаційних відрахувань відобразимо в таблиці 3.6

Таблиця 3.6 – Розрахунок річної суми амортизаційних відрахувань

Елементи основних фондів.	К-сть	Вартість, грн.	Сума грн.	Норма амортизації, %	Амортизаційні відрахування, грн.
Комп'ютер	1	9300	9300	15%	2325
Спліт система	1	6000	6000	15%	900
Приміщення	13,6м^2	2700	36720	2%	734,4
Разом:					3959,4

Таким чином, річна сума амортизаційних відрахувань становить 3959,4 гривень.

Виходячи з того, що трудомісткість створення інформаційної системи становить 10 днів, розраховуємо амортизацію обладнання за цей період за формулою:

$$А_{факт} = \frac{A_{год} * T_{факт}}{365}, \quad (3.5)$$

$Т_{факт}$ – число календарних днів на розробку вебсайту;

$A_{год}$ – річна сума амортизаційних відрахувань.

Розрахуємо суму амортизаційних відрахувань для перерахованої групи обладнання з урахуванням числа календарних днів на розробку програмного забезпечення ІС за формулою:

$$А = 3959,4 * 10/365 = 108,5 \text{ грн.}$$

Заробітна плата програміста становить 14000 грн. Відповідно, витрати на заробітну плату включаються в собівартість програми з урахуванням роботи над програмою протягом 12 днів складуть:

$$ЗП_{пр} = \frac{ЗП_{мес} * T_{факт}}{Д}, \quad (3.6)$$

де $ЗП_{пр}$ – заробітна плата в місяць програміста, грн.;

$Т_{факт}$ – число календарних днів на розробку ІС;

Д – число днів в періоді (місяць).

$ЗП_{пр} = 14000*10/22 = 6363,63$ грн.

Таблиця 3.7 – Розрахунок щомісячних матеріальних витрат

Найменування	Сума, грн/міс.
Електроенергія	1263,6
Хостинг	200
Інтернет	100
Інші витрати	300
Разом:	1863,6

$З_м$= 1863,6 гривень в місяць.

Отже, витрати на період розробки програмного продукту розрахуємо за формулою:

$$З_{пр} = З_м * Т_{факт}/Д\ , \qquad (3.7)$$

де $З_м$ – щомісячні витрати, грн.;

$Т_{факт}$ – число календарних днів на розробку вебсайту;

Д – число днів в періоді (місяць).

$$З_{пр} = 1863,6*10/22 = 847,1 \text{ грн.}$$

Розрахуємо собівартість програмного продукту за формулою:

$$С_{ст} = З_{пр} + ЗП_{пр} + ЕСВ + А, \qquad (3.8)$$

$С_{ст}$ – собівартість розробки програми

$$С_{ст} = 847,1+1818+422,65+108,5 = 3196,25 \text{ гривень.}$$

Дана собівартість є приблизною, оскільки в ній не враховані деякі деталі, які істотно не вплинуть на підсумок.

$$С_{ст} \approx 3300 \text{ гривень.}$$

Виходячи з нормального рівня рентабельності 20% ми можемо визначити ціну розробленої нами програми:

$$Ц = C_{ст} + \frac{C_{сп} * R}{100\%}\ , \qquad (3.9)$$

де $С_{ст}$ – собівартість розробки програми;

R – планований рівень рентабельності.

$$Ц = 3300 + 3300*20/100 = 4290 \text{ гривень.}$$

Так як приміщення та обладнання вже є в наявності у компанії витрати на впровадження програмного продукту складуть 4290 гривень.

$$П = (Тоб_н + Тоб_н * Тоб_{\%}) * П_{\%}, \qquad (3.10)$$

де П – передбачуваний дохід;

$Тоб_н$ – базовий варіант товарообороту;

$Тоб_{\%}$ – передбачуваний процент приросту товарообігу;

$П_{\%}$ – передбачуваний приріст прибутку.

Візьмемо передбачуваний приріст прибутку рівний 50 %

$$П_1 = (35000+35000*10/100) *50/100 = 19250 \text{ грн.}$$

Розрахунок виграшу від підвищення готовності вебкомпоненти ІС:

1. Місячний прибуток ІС = 19250 грн.

2. Тоді за 1 годину сайт приносить 19250/(30*24)=26,73 грн.

3. При готовності 0,997 сайт простоює 2,16 години в місяць, а отже втрачає 57,75 грн.

4. При готовності 0,9997 сайт простоює 0,216 години в місяць, а отже втрачає 5,77 грн

5. Виграш складає 57,75 – 5,77 = 51,98 грн в місяць.

Висновки до розділу 3

У третьому розділі було розглянуто моделювання віртуалізованої інформаційної системи з урахуванням відмов, атак на http сервер і оновлень програмного забезпечення. Були розглянуті різні технології віртуалізації програмних засобів, зокрема VMware Infrastructure та розраховані витрати від функціонування віртуалізованої інформаційної системи.

Аналіз отриманих результатів моделювання віртуалізованої інформаційної системи з усуненням дефектів програмного коду показав, що:

1) для прискорення переходу функції готовності в стаціонарний стан необхідно підвищувати значення параметру lapath, при чому значення цього параметру впливає також на мінімум функції готовності на початковому етапі експлуатації;

2) в початковий період експлуатації готовність систем з додатковим станом патчерізації нижче, ніж у систем без усунення дефектів ПЗ.

Для забезпечення високої надійності віртуалізованої інформаційної системи рекомендується використовувати такі заходи:

– використання віртуальних машин з високою надійністю, таких як VMware HA.

– розміщення віртуальних машин на серверах, що підтримують технологію їх розгортання із попередньо встановленим програмним забезпеченням.

– розробка плану відновлення після аварії, що включатиме сценарії відновлення віртуальних машин та даних.

Для підвищення рентабельності розробки та підтримки віртуалізованої інформаційної системи рекомендується використовувати стандартне програмне забезпечення, яке легко підтримувати та оновлювати; хмарні технології для зберігання даних та розміщення віртуальних машин; а також провести автоматизацію завдань управління віртуалізованою інфраструктурою.

ВИСНОВКИ

У дипломній роботі обгрунтовано і вирішено актуальну задачу розробки моделі оцінювання надійності та доступності віртуалізованої інформаційної системи шляхом врахування зміни інтенсивності атак на вразливості http сервера Apache.

1. Виконано аналіз зовнішніх факторів впливу на інформаційні системи, серед яких акцентовано увагу на зловмисних діях та атаках. Зовнішні загрози інформаційній безпеці розділено на дві категорії: випадкові та навмисні. Випадкові загрози, такі як природні лиха та аварії, є найбільш поширеними і можуть призвести до найбільших втрат інформації. Навмисні загрози, такі як шпигунство та диверсії, також можуть мати серйозні наслідки, але їх частіше можна запобігти або мінімізувати за допомогою належних заходів безпеки.

2. Потенційні порушники можуть бути як внутрішніми, так і зовнішніми. Внутрішніми порушниками можуть бути співробітники організації, які мають доступ до інформації. Зовнішніми порушниками можуть бути особи, які не мають доступу до інформації, але мають намір отримати його незаконно.

3. До інформаційних систем, які працюють безперервно, більш прийнятною є властивість доступності (стосовно показників надійності та безпеки). Коефіцієнт доступності виступає як комплексний показник і дорівнює відношенню загального часу, коли система працює без простоїв, до загального часу її функціонування з урахуванням відмов. Важливо зазначити, що забезпечення інформаційної безпеки є комплексним завданням, яке потребує комплексного підходу. Для цього необхідно розробити і реалізувати ефективні заходи щодо виявлення, усунення вразливостей інформаційних систем, а також захисту від зовнішніх і внутрішніх загроз.

4. Розглянуто методи аналізу вразливостей, що впливають на доступність вебкомпонентів інформаційних систем. Було показано, що відкритий репозиторій CVE містить дані, які дозволяють формувати підмножини вразливостей, що впливають на доступність елементів вебсервера в залежності від їх критичності.

Одержані підмножини також включають час публікації інформації (Published). Оскільки інформація про уразливості може потрапляти до бази як від експертів із безпеки, так і після їх використання зловмисниками, час реєстрації уразливості відображає ступінь зацікавленості конкретним елементом інформаційної системи дослідниками.

5. На основі аналізу було припущено, що інтенсивність атак на доступність конкретної вебкомпоненти ІС можна визначити, враховуючи максимальне значення усередненої за рік частоти фіксації інформації про вразливості, що можуть бути використані для подібних атак. При цьому критичність атаки визначається величиною базової оцінки CVSS. Для перевірки припущення було сформовано підмножини вразливостей для Apache і MySQL за період 2023 року. Було показано, що максимальне значення усередненої за рік частоти фіксації інформації про вразливості, що можуть бути використані для атак на доступність, істотно відрізняється для різних елементів HTTP сервера. Так, для HTTP сервера Apache максимальне значення становить 1,5 вразливості на місяць, а для бази даних MySQL – 0,5 вразливостей на місяць.

6. Розглянуто моделювання віртуалізованої інформаційної системи з урахуванням відмов, атак на http сервер і оновлень програмного забезпечення. Були розглянуті різні технології віртуалізації програмних засобів, зокрема VMware Infrastructure та розраховані витрати від функціонування віртуалізованої інформаційної системи.

Аналіз отриманих результатів моделювання віртуалізованої інформаційної системи з усуненням дефектів програмного коду показав, що:

– для прискорення переходу функції готовності в стаціонарний стан необхідно підвищувати значення параметру lapath, при чому значення цього параметру впливає також на мінімум функції готовності на початковому етапі експлуатації;

– в початковий період експлуатації готовність систем з додатковим станом патчерізації нижче, ніж у систем без усунення дефектів ПЗ.

7. Для забезпечення високої надійності віртуалізованої інформаційної системи рекомендується використовувати такі заходи:

– використання віртуальних машин з високою надійністю, таких як VMware HA.

– розміщення віртуальних машин на серверах, що підтримують технологію їх розгортання із попередньо встановленим програмним забезпеченням.

– розробка плану відновлення після аварії, що включатиме сценарії відновлення віртуальних машин та даних.

8. Для підвищення рентабельності розробки та підтримки віртуалізованої інформаційної системи рекомендується використовувати стандартне програмне забезпечення, яке легко підтримувати та оновлювати; хмарні технології для зберігання даних та розміщення віртуальних машин; а також провести автоматизацію завдань управління віртуалізованою інфраструктурою.

Таким чином, поставлені задачі розв’язано у повному обсязі. Напрямком подальших досліджень є розробка та дослідження імітаційних моделей віртуалізованої інформаційної системи при непуасонівських потоках впливів на систему.