Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістра
(освітній ступінь))

на тему: **Методи і засоби виявлення та оцінювання вразливостей веб-серверів у «розумних» комп'ютерних системах**

Виконав: студент (ка) 6 курсу, групи СІм-61
спеціальності 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

	(підпис)	Харитон Б.В. (прізвище та ініціали)
Керівник	(підпис)	Яцишин В.В. (прізвище та ініціали)
Нормоконтроль	(підпис)	Тиш Є.В. (прізвище та ініціали)
Завідувач кафедри	(підпис)	Осухівська Г.М. (прізвище та ініціали)
Рецензент	(підпис)	Дуда О.М. (прізвище та ініціали)

Тернопіль
2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра комп'ютерних систем та мереж

ЗАТВЕРДЖУЮ
Завідувач кафедри *Осухівська Г.М.*
«_____» 2022 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *магістр*
(назва освітнього ступеня)

за спеціальністю *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

студенту *Харитону Богдану Володимировичу*
(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) *Методи і засоби виявлення та оцінювання вразливостей веб-серверів у «розумних» комп'ютерних системах*
Керівник проекту (роботи) *Яцишин Василь Володимирович, к.т.н., доц.*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
Затверджені наказом ректора від «*06*» *грудня* 2022 року №*4/7-986*
2. Термін подання студентом завершеної роботи
3. Вихідні дані до роботи *Типи web-серверів, принципи функціонування серверів, види загроз і вразливостей, методи виявлення вразливостей програмного забезпечення*
4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Аналіз сучасних підходів до виявлення та оцінювання вразливостей комп'ютерних систем 2. Модель і метод виявлення та оцінювання вразливостей комп'ютерних систем 3. Засіб автоматизації та результати виявлення та оцінювання вразливостей веб-серверів у розумних комп'ютерних системах 4. Охорона праці та безпека в надзвичайних ситуаціях. Висновки
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Актуальність і мета дослідження. 2. Задачі дослідження, , об'єкт і предмет, наукова новизна і практична цінність дослідження. 3. Статистика вразливостей web-серверів. 4. Метод оцінювання захищеності web-серверів. 5. Ієрархічна модель безпеки web-серверів 6. Use case діаграма засобу виявлення та оцінювання вразливості web-серверів. 7. Архітектура засобу оцінювання захищеності. 8. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та	*Осухівська Г.М.*		
безпека в надзвичайних			
ситуаціях			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	*Аналіз сучасних підходів до виявлення та оцінювання*		*виконано*
	вразливостей комп'ютерних систем		
2.	*Модель і метод виявлення та оцінювання вразливостей*		*виконано*
	комп'ютерних систем		
3.	*Засіб автоматизації та результати виявлення та*		*виконано*
	оцінювання вразливостей веб-серверів у розумних комп'ютерних системах		
4.	*Охорона праці та безпека в надзвичайних ситуаціях*		*виконано*
5.	*Оформлення пояснювальної записки*		*виконано*
6.	*Оформлення графічного матеріалу*		*виконано*
7.	*Попередній захист кваліфікаційної роботи магістра*		*виконано*
8.	*Захист кваліфікаційної роботи магістра*		

Студент _____ *Харитон Б.В.*
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) _____ *Яцишин В.В.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Тема кваліфікаційної роботи: “ Методи і засоби виявлення та оцінювання вразливостей веб-серверів у «розумних» комп’ютерних системах” // Кваліфікаційна робота // Харитон Богдан Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем та програмної інженерії, група СІм-61 // Тернопіль, 2022 // с. –92, рис. – 28, табл. – 11, аркушів А1 – 8, додат. – 1, бібліогр. – 21.

Ключові слова: метод, засіб, виявлення, web-сервер, вразливість, комп’ютерна система.

Мета кваліфікаційної роботи магістра полягає у дослідження методів і засобів виявлення та оцінювання вразливостей веб-серверів у «розумних» комп’ютерних системах.

У кваліфікаційній роботі обгрунтувано модель виявлення та оцінювання вразливостей веб-серверів у «розумних» комп’ютерних системах в основі якої лежить використання методології ядер безпеки.

Розроблено метод виявлення та оцінювання вразливостей веб-серверів у «розумних» комп’ютерних системах за рахунок побудови профілю вимог вразливості веб-серверів та його формального представлення у вигляді нотацій теорії множин, що в перспективі дало змогу автоматизувати процес формування ядер безпеки.

Побудовано та представлено у вигляді шарів Фаулера архітектуру засобу автоматизації процесів виявлення та оцінювання вразливостей веб-серверів, що дало змогу спроектувати програмні компоненти і відношення між ними з подальшою імплементацією у програмному коді. Запропоновано і проведено процедури кількісного оцінювання щодо вразливості веб-серверів у комп’ютерних системах, наведено результати експериментальних досліджень щодо захищеності від загроз веб-серверів Apache та IIS.

ABSTRACT

The theme of the thesis: " Methods and means of "smart" computer systems' web server vulnerabilities detection and assessment " /Master thesis / Kharyton Bohdan Volodymyrovych / Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm -61 // Ternopil, 2022// p. - 92, fig. – 28, table. – 11, Sheets A1 – 8, Add – 1, Ref. – 21.

Keywords: method, tool, detection, web-server, vulnerability, computer system.

The purpose of the master's qualification work is to research methods and means of detecting and evaluating web server vulnerabilities in "smart" computer systems.

The qualification work substantiates the model of detection and assessment of web server vulnerabilities in "smart" computer systems, which is based on the use of security cores methodology.

A method of detecting and evaluating web server vulnerabilities in "smart" computer systems has been developed by building a profile of web server vulnerability requirements and its formal representation in the form of set theory notations, which in the future made it possible to automate the process of forming security kernels.

The architecture of the tool for automating the detection and assessment of web server vulnerabilities was built and presented in the form of Fowler layers, which made it possible to design software components and the relationship between them with further implementation in the software code.

Quantitative evaluation procedures for the vulnerability of web servers in computer systems are proposed and carried out, and the results of experimental studies on the protection against threats of Apache and IIS web servers are given.

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ

БД	База Даних
ДНФ	Диз’юнктивна Нормальна Форма
ДДНФ	Досконала Диз’юнктивна Нормальна Форма
ЖЦ	Життєвий Цикл
КС	Комп’ютерна Система
ПС	Програмна Система
ПП	Програмний Продукт
ПЗ	Програмне Забезпечення
CASE	Computer Aided Software Engineering
COTS	Commercial-Of-The-Shelf
ER	Entity Relationships
UML	Unified Modeling Language
XML	Extended Markup Language

ВСТУП

Актуальність теми. Проектування, реалізація та впровадження комп’ютерних систем на сучасному етапі розвитку інформаційних технологій набуває значної популярності та актуальності. Це, насамперед, пов’язано із розвитком технологічних можливостей та інструментальних засобів, які за короткий проміжок часу дозволяють автоматизувати процеси, притаманні конкретній предметній області. Враховуючи не високу вартість апаратного забезпечення, розвиненість засобів програмування та їх «відкритість», а також застосування підходів повторно використовуваних компонентів спостерігається тренд до впровадження розумних комп’ютерних систем у різних сферах діяльності. Прикладом ефективного використання розумних комп’ютерних систем на побутовому рівні є «розумний будинок». Для підприємств промислового та агро сектору розумні комп’ютерні системи виконують функції управління конвейєрними виробництвами, обробкою ґрунтів та інтелектуальним відеонаглядом. Однак, незважаючи на широкий спектр функціональності, які забезпечують такі системи, відкритим залишається питання щодо вразливості їх функціонування. Більшість розумних систем не працюють в автономному режимі і вимагають взаємодії із зовнішніми сервісами, що архітетурно передбачає використання певних сервісів взаємодії та відповідної комунікаційної інфраструктури. Найбільш популярними на сьогодні є комп’ютерні системи, які використовують опрацювання даних і запитів за допомогою веб-серверів. Тому актуальною науковою і практичною задачею є дослідження та оцінювання потенційної вразливості веб-серверів, як одних з центральних компонентів управління розумними комп’ютерними системами. Дослідженню безпеки апаратного і програмного забезпечення комп’ютерних систем присвячено багато наукових і практичних публікацій, що сприяло утворенню цілого напряму кібербезпеки. Однак в рамках комп’ютерної інженерії це є одним з аспектів забезпечення ефективності функціонування «розумних систем». Найбільш вагомих результатів при дослідження комп’ютерних систем досягнуто при аналізі їх

надійності (В.С. Харченко, Є.В. Бебешко S. Russo, О.М. Тарасюк, G. Mayers, R. Bloomfield, А.А. Орехова, M. Holsted). Цього вдалося досягти за рахунок впровадження методів і засобів щодо точності, повноти і адекватності трансляції потреб користувачів щодо робастності та надійності функціонування різного типу систем. На захищеність, або ступінь вразливості компонентів комп'ютерних систем впливає дуже багато факторів, що зумовлені як функціональністю самої системи, так і зовнішні фактори, пов'язані з людським фактором. Комплексного підходу щодо опису потенційних ризиків та оцінювання можливих вразливостей веб-серверів поки що не запропоновано. Тому доцільним є проведення досліджень, розробка і впровадження методів та інструментальних засобів, які б забезпечували можливість виявлення потенційно вразливих місць при функціонуванні веб-серверів, як центральних елементів управління розумними комп'ютерними системами.

Мета кваліфікаційної роботи магістра полягає у дослідження методів і засобів виявлення та оцінювання вразливостей веб-серверів у «розумних» комп'ютерних системах.

Для цього в роботі розв'язуються наступні задачі:

– аналіз наукових та прикладних досліджень у галузі безпеки та надійності апаратного і програмного забезпечення комп'ютерних систем;

– дослідження баз даних відомих вразливостей та їх вплив на функціонування комп'ютерних систем;

– обґрунтування моделі представлення загроз і вразливостей комп'ютерних систем;

– розробка методу виявлення та оцінювання вразливостей веб-серверів;

– розробка програмного засобу підтримки процесів оцінювання вразливостей веб-серверів у розумних комп'ютерних системах.

Об'єкт дослідження: процес виявлення та оцінювання загроз і вразливостей веб-серверів у розумних комп'ютерних системах.

Предмет дослідження: методи і засоби виявлення та оцінювання вразливостей веб-серверів.

Методи дослідження: При вирішенні задач кваліфікаційної роботи застосовувались такі методи і засоби: аналіз та узагальнення – при проведенні аналізу існуючих методів і засобів виявлення та оцінювання вразливсті web-серверів; теорії надійності, теорії ймовірності і математичної статистики, теорії множин – для формалізації та побудови моделі і методу виявлення та оцінювання рівня вразливості web-серверів; проектування та програмування – при побудові архітектури та бази даних програмного засобу автоматизації процесу виявлення та оцінювання вразливості web-серверів; експеримент та вимірювання – для апробації запропонованого методу і моделі.

Наукова новизна отриманих результатів. Наукова новизна результатів дослідження полягає в наступному:

– уперше розроблено метод виявлення та оцінювання вразливостей веб-серверів у «розумних» комп’ютерних системах за рахунок формування сукупності вимог до загроз веб-серверів та його формального представлення у вигляді елементів теорії множин, що в перспективі забезпечує автоматизацію процесу створення ядер безпеки;

– удосконалено і доповнено модель виявлення та оцінювання вразливості веб-серверів у «розумних» комп’ютерних системах за допомогою ядер безпеки, що дало можливість забезпечити структурованість та автоматизацію розрахунку параметрів вразливостей веб-серверів, а також наростити значення достовірності і точності експертних оцінок.

Практичне значення одержаних результатів. Впровадження методу і засобу виявлення та оціюванн вразливості веб-серверів у розумних комп’ютерних систем забезпечує автоматизацію процесів і процедур побудови ядер безпеки, централізованого керування ними, а також допомагає експертами при прийнятті рішень.

Публікації. Результати кваліфікаційної роботи апробовані на X науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року) як тези конференцій.

1. Яцишин В.В., Харитон Б.В. Архітектура системи підтримки процесів виявлення та оцінювання вразливостей веб-серверів у комп'ютерних системах. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С.96
2. Яцишин В.В., Харитон Б.В. Схема реляційної бази даних для зберігання та опрацювання вразливостей веб-серверів у розумних комп'ютерних системах. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 101

Структура роботи. Кваліфікаційна робота містить розрахунково-пояснювальну записку та графічний матеріал. До складу записки входить вступу, 4 розділи, загальні висновки, список використаних джерел і додатки. Обсяг роботи: розрахунково-пояснювальна записка – 93 арк. формату А4, графічна частина – 8 аркушів формату А1.

РОЗДІЛ 1
АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО ВИЯВЛЕННЯ ТА ОЦІНЮВАННЯ ВРАЗЛИВОСТЕЙ КОМП'ЮТЕРНИХ СИСТЕМ

1.1. Аналіз ролі і принципів функціонування веб-серверів у розумних комп'ютерних системах

При побудові «розумних» комп'ютерних систем базовою є архітектура клієнт – сервер. При цьому передбачається взаємодія клієнта у вигляді web-браузера та одержання і відправлення інформації на web-сервер. Обмін даними забезпечує комп'ютерна мережа з доступом до Internet. Особливу цінність представляє інформація, що знаходиться на сервері, тому актуальною задачею є визначення захищеності, як частини безпеки, web-серверів.

Для формування відповідей на запит користувача, web – сервер може використовувати дані, що зберігаються в базах даних [1]. Принцип роботи web – додатків представлено на функціональній схемі на рис. 1.1.

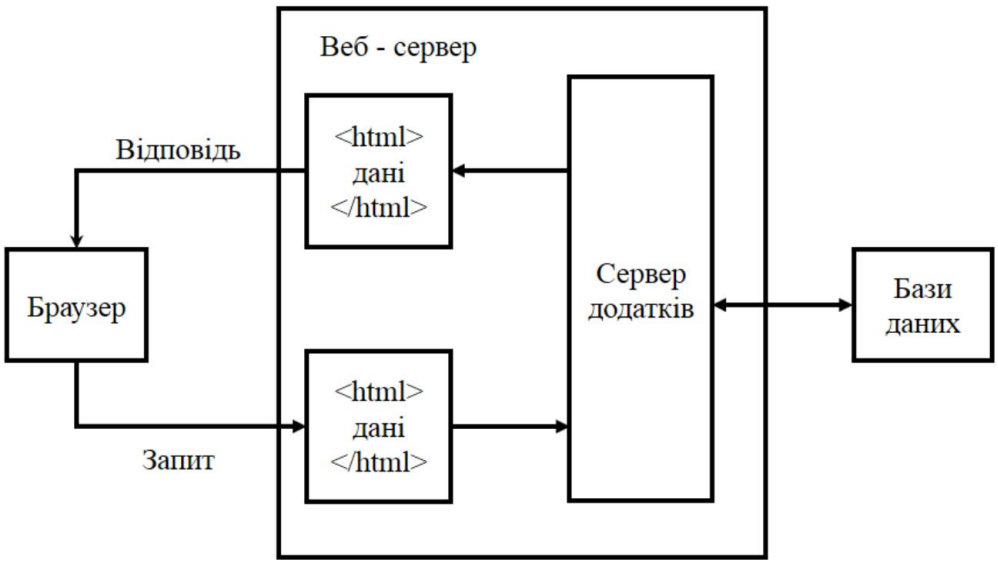

Рис. 1.1. Схема функціонування веб-додатків

Клієнтська частина відповідає за формування запитів до web – сервера та опрацювання відповідей. Серверна частина відповідає за обробку запитів від

клієнта, виконує обчислення, формує відповідь та відправляє її клієнту за допомогою протоколу HTTP («Hypertext Transfer Protocol»).

Для побудови клієнтської та серверної частин додатка використовуються різні технології та мови програмування. Як правило, код серверної частини, що опрацьовує запити клієнта, може бути реалізований за допомогою мов програмування [2]:

– Ruby on Rails;
– PHP («Hypertext Preprocessor»);
– C#;
– Java;
– Python;
– JavaScript.

На стороні клієнта використовуються [2]:

– таблиці стилів «Cascading Style Sheets»;
– мова гіпертекстової розмітки «Hypertext Markup Language»;
– JavaScript.

Найбільш популярними web – серверами для функціонування web – додатків, є [3]:

– Apache;
– Internet Information Services (IIS);
– Nginx;
– Google Web Server.

Отже, зважаючи на різноплановість та різну повноту функціональності технологій, які можна застосовувати при реалізації «розумних» комп'ютерних систем з web-серверами, актуальним є дослідження та аналіз потенційних вразливостей та загроз веб – серверів, а також можливих використання «слабких місць» потенційними зловмисниками.

Класифікацію web – додатків для керування «розумними» комп'ютерними системами за особливостями їх побудови та функціональних властивостей можна сформувати таким чином [4]:

– статичні web – додатки (веб – сторінки такого додатка передаються до клієнтської частини у такому вигляді, у якому зберігаються на сервері);

– динамічні web – додатки (відповіді сервери формуються залежно від запитів з клієнтської частини);

– web – додатки у сфері електронної комерції (веб – додатки цього типу включають елементи реєстрації користувачів, а також механізми, що дозволяють проводити платежі через Інтернет);

– web – портали (цей тип веб – додатків включає елементи, що потребують реєстрації користувача: форуми, онлайн – чати, електрона пошта та ін.);

– web – додатки з елементами анімації, де використовується технологія Flash;

– web – додатки з системами управління контентом (WordPress, Joomla, Drupal), які використовуються для постійного оновлення вмісту, призначеного для користувачів.

Робота всіх web-серверів заснована на протоколі HTTP (Hypertext Transfer Protocol), який визначає спосіб обміну інформацією web-серверів з браузерами та іншим клієнтським ПЗ. Коли встановлюється web-сервер, то за замовчуванням його конфігурація налаштовується на виконання найбільш загальних завдань, таких як відображення простих web-сторінок. Хоча функції, виконувані web-серверами, значно розширилися з часу появи самих серверів, головним їхнім завданням є передача браузеру документів HTML. HTML є мовою форматування, яку використовують браузери для відображення тексту і графіки. Всі web-сервери підтримують протокол передачі гіпертексту HTTP, який визначає, як інформація передається між браузером і web-сервером. Web-сервери і браузери повинні дотримуватися одних і тих же правил, що визначені специфікацією HTTP. Ця узгодженість дозволяє користувачеві, який працює, наприклад, з браузером Netscape або з браузером Internet Explorer, бачити одні й ті ж сторінки на будь-якому сервері. Двома найбільш популярними web-серверами є Apache, випущений Apache Software Foundation, і Internet Information Services (IIS) компанії Microsoft. Згідно з даними Netcraft майже дві третини всіх web-серверів використовують

Apache і менше однієї третини IIS. Обидва типи серверів публікують HTML-сторінки і виконують інші завдання, необхідні для забезпечення інтерактивності web-сторінок. Після встановлення web-сервера, можна змінити параметри його конфігурації, зокрема, номер порту, через який сервер прослуховує запити з web, адреси, з яких web-сервер зчитує HTML-файли (кореневий каталог (root) сервера), і налаштування, що визначають робочі характеристики комп'ютера в залежності від рівнів трафіку.

Функції web-сервера можна розширити, наприклад, щоб він приймав запити з декількох доменів, створивши таким чином віртуальні сервери.

1.2. Аналіз вразливостей та загроз web-серверів

Наведемо статистику вразливостей для трьох найбільш поширених засобів розробки веб-додатків: PHP, Java і ASP.NET. Три чверті (76%) сайтів, написаних на мові програмування PHP, містять критичні уразливості. Веб-ресурси, написані на Java і ASP.NET, виявилися менш уразливими: 70 і 55% додатків відповідно містять критичні уразливості. Усі додатки, написані на трьох зазначених мовах програмування містять вразливості як мінімум середнього ступеня ризику. На рис. 1.2 наведено статистику вразливостей web-додатків написаних на різних мовах програмування.

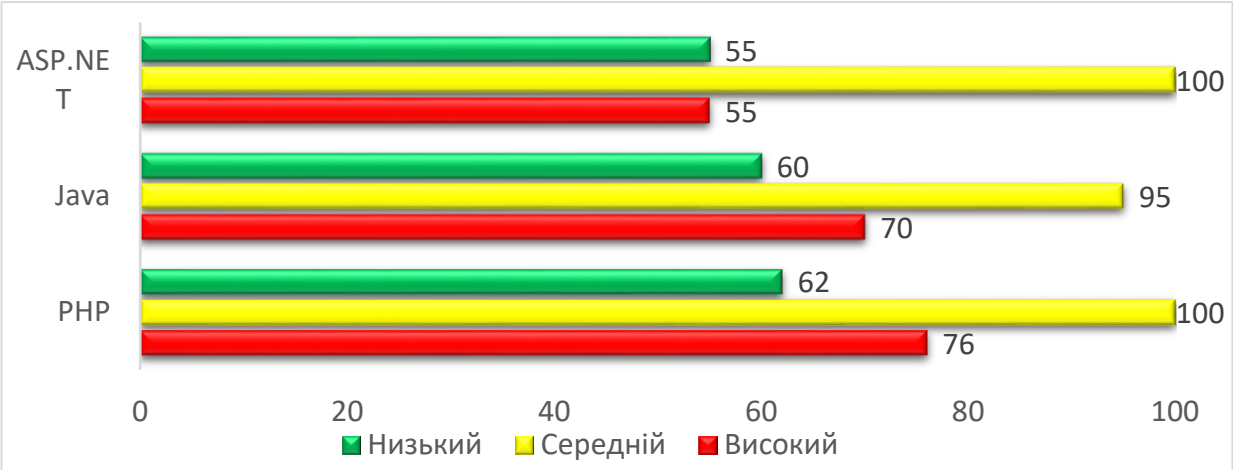

Рис. 1.2. Статистика вразливостей web-додатків за мовами програмування

Найбільша частка сайтів з критичними вразливостями (75%) функціонувала на базі web-сервера Apache Tomcat. Значно зросла кількість вразливих ресурсів під керуванням web-сервера Microsoft IIS (71%), Nginx (57%), і лише відносно сайтів під керуванням Apache помітна позитивна динаміка: 60% ресурсів замість 88% містять критичні уразливості

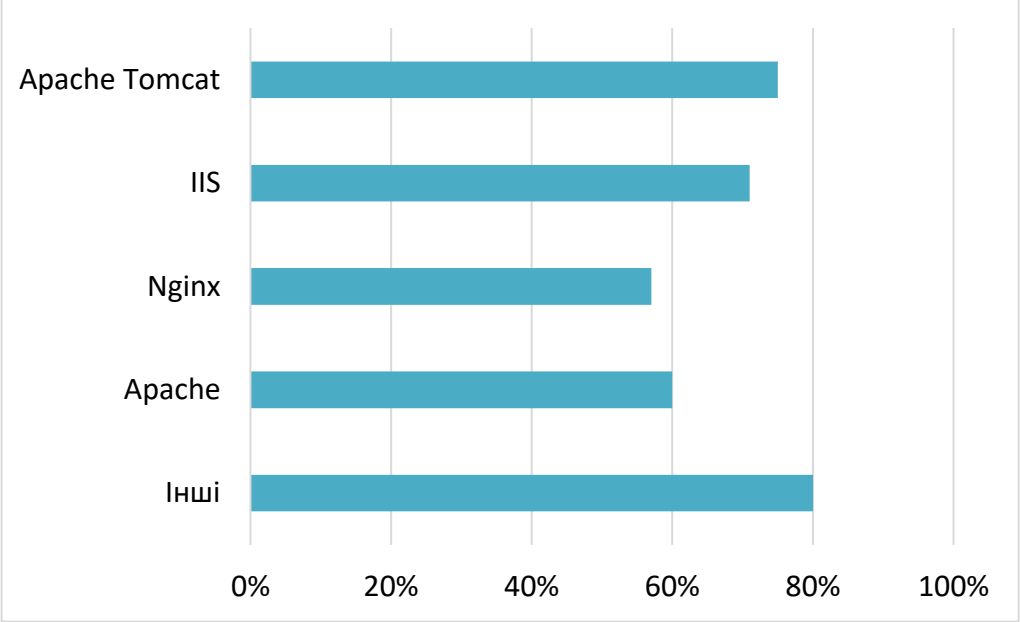

Рис. 1.3. Критичні вразливості web-додатків за типами web-серверів

Деякі вразливості додатків, визначені згідно класифікації WASC TC v.2, є наслідком некоректного адміністрування. Далі наведено статистику вразливостей для найбільш поширених web-серверів – Apache, Nginx і Microsoft IIS. Web-додатки, що функціонують на базі Apache Tomcat, не містили помилок адміністрування: всі виявлені недоліки відповідних систем були пов’язані з помилками в коді (Java). Це не означає, що помилки адміністрування не характерні для даного web-сервера, оскільки кількість розглянутих систем на цій платформі невелика (4 шт.).

Найбільш поширені помилки, а також частки уразливих сайтів під управлінням web-серверів Apache, IIS і Nginx наведені у табл. 1.1 (всі представлені уразливості характеризуються середнім ступенем ризику).

Таблиця 1.1

Найбільш поширені помилки та вразливості під керування різних web-серверів

IIS	сайти, %	Nginx	сайти, %	Apache	сайти, %
Неправильне налаштування web-додатку (Application Misconfiguration)	57	Витік інформації (Information leakage)	48	Витік інформації (Information leakage)	44
Витік інформації (Information leakage)	43	Неправильне налаштування web-сервера (Server Misconfiguration)	22	Неправильне налаштування web-сервера (Server Misconfiguration)	36
Неправильне налаштування web-сервера (Server Misconfiguration)	43	Прогнозоване місце розташування ресурсів (Predictable Resource Location)	13	Прогнозоване місце розташування ресурсів (Predictable Resource Location)	24
Неправильні налаштування прав доступу до файлової системи (Improper filesystem permissions)	14	Неправильні налаштування прав доступу до файлової системи (Improper filesystem permissions)	13	Неправильне налаштування конфігурації додатку (Application Misconfiguration)	16
Прогнозоване місце розташування ресурсів (Predictable Resource Location)	-	Неправильне налаштування конфігурації додатку (Application Misconfiguration)	9	Неправильні налаштування прав доступу до файлової системи (Improper filesystem permissions)	4

Найпоширенішою помилкою адміністрування є розголошення важливих даних (Information Leakage). Приблизно 45% всіх досліджених інформаційних ресурсів підпадають під цю уразливість. Найчастіше дана уразливість зустрічається у web-ресурсах під управлінням веб-сервера Nginx. Порівняння часткою вразливих ресурсів під керуванням різних web-серверів для кожної уразливості адміністрування наведено на діаграмі нижче (рис. 1.4).

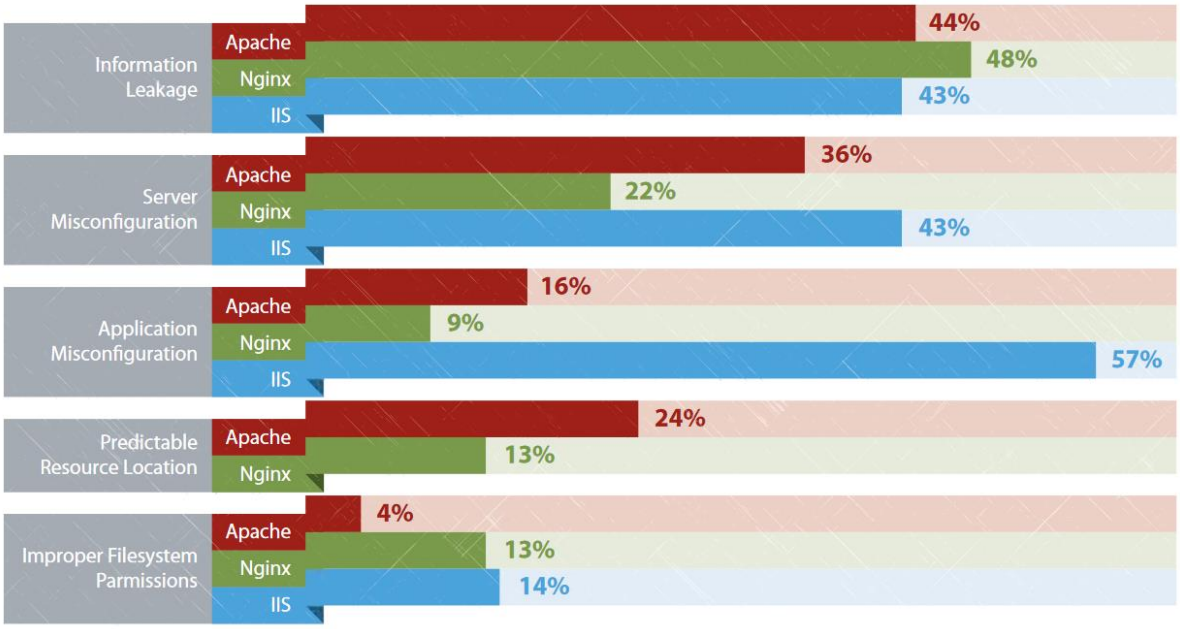

Рис. 1.4 Відношення вразливих сайтів на різних web-серверах

Таким чином, визначено основні вразливості як програмного забезпечення, що функціонує на web-серверах, так і самих web-серверів, які в подальшому необхідно врахувати при розробці методу та інструментального засобу виявлення та оцінювання вразливості web-серверів при проектуванні «розумних» комп’ютерних систем.

1.3. Класифікація існуючих вразливостей web-додатків

Поняття вразливості описане в таких стандартах, як ISO/IEC 29147:2014 «Information technology. Security techniques. Vulnerability disclosure» (в Україні діє стандарт ДСТУ ISO/IEC 29147:2016 «Інформаційні технології. Методи захисту. Розкриття вразливостей») та ISO/IEC 27000:2016. В стандарті ISO/IEC 29147 сказано, що вразливість – це слабке місце в програмному забезпеченні, апаратному забезпеченні або онлайн – сервісі. Слабке місце в системі може бути викликане недоліками проектування програмного та апаратного забезпечення, недоліками в керуванні процесом розробки та ін. [5]. У стандарті ISO/IEC 27000:2016 вразливість визначається як слабке місце активу або засобу контролю та управління, яке може бути використане однією або кількома загрозами [6]. В документі НД ТЗІ 1.1 – 003 – 99 сказано, що вразливість системи – нездатність системи протистояти реалізації певної загрози або сукупності загроз [7].

Класифікацією вразливостей web – серверів та web-додатків займається Консорціум з безпеки веб – додатків WASC (Web Application Security Consortium).

Згідно даних офіційного веб – сайту Консорціуму WASC, він представляє собою неприбуткову організацію, що складається з міжнародних експертів, фахівців з інформаційної безпеки та безпеки мережі Інтернет [8]. Консорціумом WASC розроблено спеціальний документ, в якому описана класифікація вразливостей web – додатків – WASC Threat Classification. Згідно з документом WASC Threat Classification, вразливості web – додатків поділяються за наступними етами життєвого циклу програмного забезпечення [9]:

– етап проектування – охоплює вразливості, які можуть з’явитися внаслідок помилок у проектуванні web – додатка;

– етап реалізації – охоплює вразливості, які можуть з’явитися через помилки під час реалізації компонентів веб – додатка, наприклад – написання програмного коду;

– етап розгортання – охоплює вразливості, які можуть виникнути під час налаштування web – додатка до роботи, наприклад – неправильна конфігурація web – сервера;

Класифікація вразливостей за етапами життєвого циклу web – додатка наведена у табл. 1.2 [9].

Таблиця 1.2

Фрагмент класифікації вразливостей згідно документа WASC Threat Classification за етапами життєвого циклу додатка

Вразливість	Етапи життєвого циклу додатку		
	Проектування	Реалізація	Розгортання
Зловживання функціональними можливостями (Abuse of Functionality)	+		
Неправильна конфігурація додатка (Application Misconfiguration)		+	+
Підбір (Brute Force)	+	+	
Переповнення буфера (Buffer Overflow)		+	
Підміна контенту (Content Spoofing)		+	
Передбачуване значення ідентифікатора сесії (Credential /Session Prediction)		+	
Міжсайтове виконання сценаріїв (Cross-Site Scripting)		+	
Підробка міжсайтових запитів (Cross-Site Request Forgery)	+	+	
Відмова в обслуговуванні (Denial of Service)	+	+	

Продовження табл. 1.2

Вразливість	Етапи життєвого циклу додатку		
	Проектування	Реалізація	Розгортання
Індексування директорій (Directory Indexing)	+		
Атака на функції форматування стрічок (Format String)	+		
Підміна HTTP – відповідей (HTTP Response Smuggling)	+		
Розщеплення HTTP – відповідей (HTTP Response Splitting)	+		
Підміна HTTP – запитів (HTTP Request Smuggling)	+		
Розщеплення HTTP – відповідей (HTTP Request Splitting)	+		
Переповнення цілого значення (Integer Overflows)	+		
Неправильне розділення доступу до файлової системи (Improper Filesystem Permissions)	+	+	
Неправильна обробка вхідних даних (Improper Input Handling)	+		
Неправильна обробка вихідних даних (Improper Output Handling)	+		
Витік інформації (Information Leakage)	+	+	+
Незахищене індексування (Insecure Indexing)	+	+	

Продовження табл. 1.2

Вразливість	Етапи життєвого циклу додатку		
	Проектування	Реалізація	Розгортання
Недостатня протидія автоматизації (Insufficient Anti-automation)	+	+	
Недостатня аутентифікація (Insufficient Authentication)	+	+	
Недостатня авторизація (Insufficient Authorization)	+	+	
Недостатня реалізація механізму відновлення пароля (Insufficient Password Recovery)	+	+	
Недостатня перевірка процесу (Insufficient Process Validation)	+	+	
Недостатня тривалість сеансу (Insufficient Session Expiration)	+	+	+
Недостатній захист транспортного рівня (Insufficient Transport Layer Protection)	+	+	+
Впровадження операторів LDAP (LDAP Injection)	+		
Інєкції через поштові команди (Mail Command Injection)	+		
Додавання нульових байтів до даних (Null Byte Injection)	+		
Виконання команд ОС (OS Commanding)	+		
Зворотній шлях у директоріях (Path Traversal)	+		

Продовження табл. 1.2

Вразливість	Етапи життєвого циклу додатку		
	Проектування	Реалізація	Розгортання
Передбачуване розташування ресурсів (Predictable Resource Location)	+	+	
Додавання віддалених файлів (Remote File Inclusion, RFI)	+	+	
Обхідний маршрут (Routing Detour)	+		
Неправильна конфігурація сервера (Server Misconfiguration)	+		
Фіксація сесії (Session Fixation)	+	+	
SQL ін’єкції (SQL Injection)	+		
Зловживання перенаправленням URL (URL Redirector Abuse)	+	+	
XPath ін’єкції (XPath Injection)	+		
Неефективна обробка даних XML – аналізаторами (XML Attribute Blowup)	+		
Аналіз XML – вводу (XML External Entities)	+		
Розширення XML – об’єктів (XML Entity Expansion)	+		
XML ін’єкції (XML Injection)	+		
Ін’єкція запитів XQuery (XQuery Injection)	+		

Помилки, що можуть бути допущені під час різних етапів життєвого циклу програмного забезпечення, описані в списку CWE (Common weakness enumeration),

який сформовано корпорацією MITRE. Згідно даними веб – сайту корпорації MITRE, CWE є базовим стандартом при ідентифікації слабких місць у програмному забезпеченні, та запобіганню їх появи [10]. Кожному недоліку у списку присвоєно власний ідентифікатор (ID), список містить близько тисячі CWE ID. При описі вразливості в базах даних вразливостей може бути вказано CWE ID для додаткової інформації. Кожен CWE ID у списку CWE має наступний опис [11]:

– короткий та/або розширений опис про конкретний CWE ID;

– етап, під час якого може бути допущена розробником помилка, яка згодом може привести до виникнення вразливості (етап проектування, експлуатації);

– мова програмування/платформа, які є основою функціонування програмного забезпечення (Java, C++, або будь – яка), також в якості прикладу можуть бути наведені фрагменти програмного коду, в якому допущено помилку, яка призводить до неправильного функціонування або виникнення вразливості в програмному забезпеченні;

– посилання на записи в міжнародних базах даних вразливостей, які містять дані про існуючі вразливості в компонентах програм, до яких призвела помилка, що має CWE ID;

– шляхи до виправлення помилки, або правильний варіант фрагмента програмного коду, що гарантує безпеку обробки даних;

– зв’язки з іншими CWE ID, внаслідок яких виникла помилка в конкретному місці програмного забезпечення на конкретному етапі.

Іншим варіантом класифікації вразливостей та ризиків web – додатків є рейтинг 10 найпоширеніших вразливостей та ризиків – OWASP Top – 10.

OWASP – це некомерційна організація, діяльність якої орієнтована на підвищення безпеки програмного забезпечення [12]. Остання редакція документа OWASP Top – 10 випущена організацією OWASP за 2017 рік і виглядає наступним чином [13]:

– вставка інструкцій (Injection) – відбувається вставка SQL – інструкцій, які передаються на обробку веб – додатку, який після їх отримання може почати виконувати довільні команди;

– некоректна аутентифікація (Broken Authentication) – функції аутентифікації можуть бути реалізовані таким чином, що дозволяють обходити паролі, або отримувати ідентифікатори користувачів;

– витік критичних даних (Sensitive Data Exposure) – недостатня захищеність персональних даних;

– атаки на засоби аналізу XML – вводу (XML Eternal Entities) – зловмисники можуть завантажувати XML – файли на сервер або включати шкідливий код у документ XML;

– неправильний контроль доступу (Broken Access Control) – контроль доступу реалізовано таким чином, що авторизовані користувачі можуть мати в системі такі повноваження, які не повинні мати;

– небезпечна конфігурація оточення (Security Misconfiguration) – відсутність оновлень та неправильна конфігурація окремих компонентів web – додатків може нести в собі додаткову загрозу безпеці;

– міжсайтове виконання сценаріїв (XSS) – зловмисник отримує можливість виконувати сценарії у браузері жертви, перехоплювати сценарії користувача, перенаправляти користувачів на інші веб – сайти;

– незахищений процес десеріалізації (Insecure Deserialization) – зловмисник може порушати логіку роботи веб – додатка, підробляючи об’єкти додатка, що призводить до віддаленого виконання коду зловмисника;

– використання компонентів з відомими вразливостями (Using Components with Known Vulnerabilities) – програмне забезпечення, у якого термін підтримки вже вичерпаний, або яке є неоновленим, може залучити більше зловмисників до його зламу;

– недостатній моніторинг та ведення журналів подій (Insufficient Logging and Monitoring) – погано організований механізм ведення журналів подій та моніторингу може призвести до того, що зловмисники можуть неодноразово робити атаки на web – додатки, залишаючись непоміченими.

Згідно досліджень компанії Vercode, яка займається тестуванням захищеності програмного забезпечення, при проведенні тестування на

проникнення у 74 % програмних додатків знаходиться як мінімум одна вразливість зі списку OWASP Top – 10 [14].

1.4. Аналіз інструментальних засобів виявлення вразливостей та оцінювання захищеності комп'ютерних систем

Компонентно-орієнтований підхід використовується при побудові сучасних «розумних» комп'ютерних систем, в тому числі системах критичного застосування. Критичні комп'ютерні системи використовуються в атомній енергетиці, військових і аерокосмічних комплексах та ін.. У зв'язку з цим, оцінювання надійності і безпеки компонентів (COTS – Commercial-Of-The-Shelf) є актуальним напрямком сучасних досліджень. В останні роки активно формуються групи експертів і розвиваються різноманітні ресурси, спрямовані на пошук, облік і зберігання даних про проблеми в інформаційній безпеці програмного забезпечення КС. Їх мета – проінформувати розробників і користувачів КС, допомогти виявити і розв'язати вже відомі проблеми надійності та безпеки, а також своєчасно визначати нові вразливості і усувати їх на ранніх стадіях виявлення, до того, як зловмисники встигли скористатися ними. На підставі інформації з таких ресурсів є можливість провести дослідження показників надійності і безпеки різних OTS компонентів, використовуючи кількісну міру або оцінку імовірності.

Коротко дамо характеристику основним, найбільш повним і корисним ресурсам, які можна використати для аналізу характеристик надійності та безпеки комп'ютерних систем, які використовують web-сервери.

1.4.1. CVE

Підтримується корпорацією Mitre [1]. CVE не є самостійною базою або ресурсом вразливостей, а позиціонується як словник стандартизованих ідентифікаторів вразливостей та інших елементів, які пов'язані з незахищеністю даних.

Головна мета CVE – стандартизувати імена всіх відкритих і публічно відомих вразливостей в програмних продуктах. Основне завдання – полегшити обмін даними між окремими базами вразливостей та інструментами інформаційної безпеки. Вміст словника CVE є результатом спільної роботи багатьох експертів з інформаційної безпеки з усього світу. CVE надається в режимі вільного доступу і фінансується безпосередньо урядом США. Абсолютна більшість існуючих баз і ресурсів вразливостей програмного забезпечення використовують CVE індекси і синхронізуються з інформацією, доступною в CVE.

1.4.2. NVD

Національна база вразливостей [2], один з найбільш передових ресурсів вразливостей, який інтегрує всі бази вразливостей, доступні в США. На сьогоднішній день база NVD вже містить більше 50 тисяч записів, і ця кількість постійно зростає, збільшуючись в середньому на 12-17 нових записів щодня. NVD базується здебільшого на інформації з CVE і синхронізує свої поповнення зі словником CVE. Рейтинг серйозності вразливостей в базі NVD виставляється відповідно до стандарту CVSS [3].

Інформація з бази NVD доступна безкоштовно і надається в форматі xml файлу.

1.4.3. Secunia

Датська комерційна організація [4] пропонує послуги щодо захисту інформаційних ресурсів на основі реалізованих нею програмних продуктів для пошуку комп’ютерних вірусів і вразливостей у програмному забезпеченні. Secunia має свій власний штат фахівців з безпеки, які тестують, перевіряють, контролюють і оцінюють публічні повідомлення про помилки і слабкі місця. У той же час фахівці Secunia проводять і свою власну роботу з пошуку дефектів і взлому у безпеці. Знайдені ними проблеми відправляються розробникам програмного забезпечення, які підтверджують або спростовують наявність уразливості, і якщо вразливість

дійсно є, створюють оперативне виправлення або випускають оновлену версію свого ПЗ.

Ресурс також надає безкоштовну веб-версію з доступом до бази вразливостей і хорошим інтерфейсом пошуку по цій базі. Користувачі можуть отримати повний список вразливостей, знайдених у їхньому продукті, подивитися рейтинг серйозності знайдених вразливостей і дізнатися, чи існує спосіб виправлення для кожної конкретної проблеми. Також є можливість побудувати деякі графіки та діаграми для більш наочного подання інформації.

Крім розглянутих вище ресурсів, існує багато інших інформаційних джерел: SecurityFocus, CERT, OSVDB, Security Tracker та ін. Кожен з них є цінним і в тому чи іншому випадку може бути корисний при виявленні і проведенні оцінювання безпеки інформаційних систем, а також виявлення потенційних вразливостей та загроз.

1.5. Висновки до розділу

Результати, одержані в даному розділі, полягають в наступному:

1. Проведено аналіз функціональності та організації комп’ютерних систем в основі яких лежить використання web-серверів. У результаті цього, виявлено можливі компоненти ураження та способи проникнення «зловмисників» для нанесення шкідливого впливу на безпеку та надійність функціонування серверів. Це дало змогу врахувати потенційно-слабкі місця web-серверів та визначити атрибути вразливостей, рівень їх критичності, які необхідно врахувати в процесі захищеності комп’ютерних систем.

2. Проаналізовано та класифіковано відомі загрози та вразливості web-серверів, встановлено ступінь їх автоматизованого та ручного виявлення, що в подальшому повинні бути включеними у метод оцінювання безпеки та захищеності web-серверів.

3. Проаналізовано методи оцінювання загроз та вразливостей web-серверів, у результаті якого встановлено, що більшість методів передбачає експертне оцінювання загроз, а також застосування багатьох автоматизованих засобів їх виявлення. Це підкреслює необхідність та актуальність задач оцінювання захищеності web-серверів, які б давали можливість централізовано з одного координаційного центру використовувати як знання експертів з безпеки, так і повноту можливостей інструментальних засобів.

4. Проаналізовано інструментальні засоби виявлення загроз безпеки для web-серверів і встановлено, що більшість з них представляє собою бази даних відомих загроз та вразливостей, що дало змогу в перспективі використати їх при побудові засобу автоматизації процесу оцінювання захищеності web-серверів.

РОЗДІЛ 2
МОДЕЛЬ І МЕТОД ВИЯВЛЕННЯ ТА ОЦІНЮВАННЯ ВРАЗЛИВОСТЕЙ КОМП'ЮТЕРНИХ СИСТЕМ

2.1. Особливості процесів виявлення та оцінювання вразливості веб-серверів

Особливості процесів виявлення та оцінювання вразливостей різних інформаційних систем, в тому числі і веб-серверів, а також детальний аналіз використання в процесі оцінювання формальних нотацій та методології ядер безпеки, показали необхідність подальшого вдосконалення, більш глибокої формалізації та автоматизації цих процесів. Для досягнення цієї мети запропоновано ввести додаткові етапи при оцінюванні безпеки і як наслідок здатності до вразливості комп'ютерних систем. Основними з таких стадій є наступні:

– аналіз і виявлення сукупності вимог, оцінювання на відповідність яким, може бути формалізована і повністю або частково автоматизована;

– визначення множини дій та операцій, що виконуються експертом в процесі оцінювання та виявлення вразливостей і загроз, які є відносно незалежними, майже не перетинаються з іншими процедурами, і використовуються або при деяких модифікаціях можуть адаптуватися для використання в різного роду системах;

– об'єднання окремих операцій та етапів процесу виявлення та оцінювання у логічні блоки, що відповідають за обчислення значень певних вимог і дозволяють проводити оцінювання окремих компонентів і підсистем. Виділення таких блоків дає змогу уніфікувати і в подальшому автоматизувати окремі частини оцінювання процесів різних проектів, при цьому скорочуючи об'єм ручної роботи і підвищуючи достовірність значень оцінок;

– проведення аналізу всіх можливих даних, які можна отримати про досліджувану систему, і виявлення серед них множини таких, з якими можна працювати формально. До таких даних, зазвичай, відносяться дані про результати

тестування, початковий та виконуваний код та ін. Дані природньою мовою (документація проекту, опис процесів розробки тощо) на цій стадії можуть додатково опрацьовуватись за допомогою різних утиліт та алгоритмів, що дозволяють представити документ у формальному вигляді або видобувати з нього необхідну формалізовану інформацію;

– вибір варіанта реалізації та написання програмного коду, що виконує опрацювання формалізованих даних і видає результати оцінювання певних характеристик і вимог, що дозволить автоматизувати певні етапи процесу оцінювання.

Проведення експертом зазначених попередніх етапів дозволяє:

– обрати методи і засоби для автоматизованого і формалізованого оцінювання функціональної безпеки систем різного виду, скоротити обсяг ручної праці, і як наслідок, кількість помилок експертів;

– спростити процес оцінювання для зацікавлених осіб, які не є експертами в цій галузі;

– прискорити та автоматизувати процес виявлення та оцінювання загроз системи щодо відповідності вимогам, які до неї висунуто, а також надати рекомендації щодо побудови документу ядер безпеки і, в кінцевому випадку, забезпечити максимально безпечне функціонування комп’ютерної системи.

Перед проведенням аналізу вразливостей та оцінюванням складних систем доцільно провести декомпозицію системи на підсистеми та побудувати ієрархію цих елементів. Найбільш вдалою є декомпозиція складних систем на підсистеми відповідно до природи системи і виділенням в них різних класів об’єктів, що вимагають оцінювання вразливості: апаратних і програмних компонентів, структур і процесів організації розробки, а також функцій і логіки, що керується людиною. Аналіз цих об’єктів відбувається шляхом декомпозиції на більш прості завдання. Наприклад, оцінювання програмної системи передбачає виконання окремих задач оцінювання повторно використовуваних компонентів і компонентів власної розробки. Схематична декомпозиція системи і етапи проведення оцінювання представлено на рис. 2.1.

Іншим можливим підходом до побудови ієрархії компонентів системи і подальшого аналізу є виділення основних вузлів на основі ієрархії, що прийнята в теорії надійності. Для прикладу такими вузлами можуть бути безвідмовність, живучість, здатність до обслуговування системи. Здатність до обслуговування системи повинна, в свою чергу, підтверджуватися аналізом ремонтопридатності, придатності до управління та контролю, і т.п.

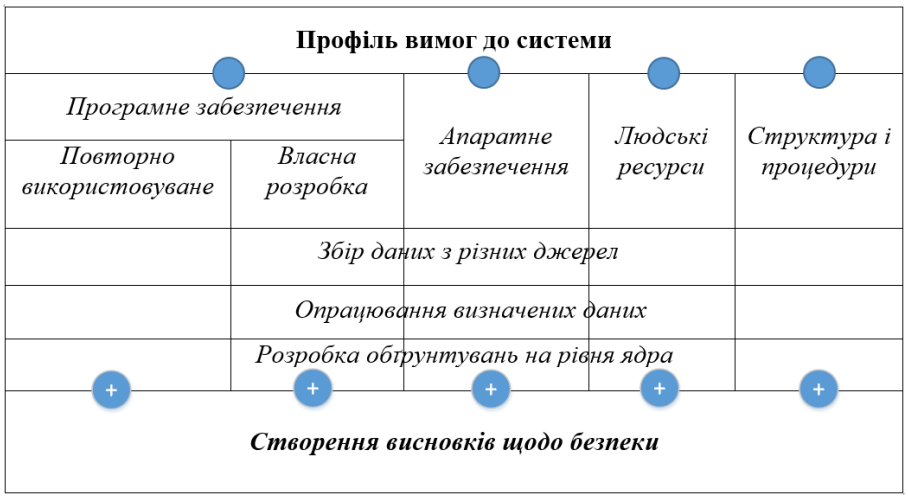

Рис. 2.1. Етапи побудови документації щодо безпеки з врахуванням декомпозиції системи

Якою б не була структура ієрархії, процес аналізу повинен представляти собою формальне розбиття всього процесу виявлення та оцінювання загрозна ряд підзадач і послідовне їх виконання. Візуалізація структури процесу оцінювання може бути представлена у вигляді орієнтованого графа, дерева або ж однією з відомих формальних нотацій.

2.2. Побудова моделі виявлення та оцінювання вразливостей web-серверів

Процес виявлення та оцінювання вразливостей web-сервера та комп’ютерної системи в цілому можна представити у вигляді множини трьох базових компонентів, які забезпечують реалізацію пріоритетних задач. Функціонування таких компонентів передбачає послідовне їх виконання, у чітко встановленому

порядку. Тоді, формально, модель виявлення та оцінювання вразливостей можна зобразити у вигляді кортежа

$$M_S = <\{ S_r \},\{ S_m \},\{ S_S \}> \quad (2.1)$$

де $\{ S_r \}$ – компонент роботи з вимогами;

$\{ S_m \}$ – компонент оцінювання відповідності вимогам із застосуванням наявних методів;

$\{ S_S \}$ – компонент аналізу, опрацювання і формування результатів документу ядра безпеки.

Для більш деталізованого представлення моделі, її можна описати за допомогою відповідних множин і сигнатури

$$M_S = < R_O, R_p, T, D_r, A_r, P, F, C_{RM}, C_{DM,} M, D_s, A, Res, A_{res}, D_O, SC > \quad (2.2)$$

Наступний крок передбачає формальне представлення детального опису кожного елемента моделі (2.1)

$$S_r = \{ R_O, R_P, T, D_r, A_r, P, F \} \quad (2.3)$$

де R_O – узагальнений профіль вимог, до складу якого входять нормативні документи і специфікації стандартів конкретного домену;

R_p – профіль власних вимог, сформований на базі профілю R_O;

T – сукупність видів вимог у R_p;

D_r – сукупність інформації про комп’ютерну систему для формування профілю вимог R_p;

A_r – алгоритми та функції виявлення критеріїв формування R_p з врахуванням інформації про комп'ютерну систему;

P – сукупність параметрів, які є важливими при побудові профілю R_p на основі R_O;

F – фільтр, який застосовується для встановлення елементів приватного профілю R_p з врахуванням R_O та P.

Профіль вимог R_O формує вхідний перелік елементів для цього компоненту. Застосовуючи до визначеної сукупності елементів: підмножини, що належить до загального профайлу вимог $R_O^{'} \subseteq R_O$; типів $T_i \subseteq T$ із застосуванням параметрів $P^{'} \subseteq P$, які одержані з інформації про комп'ютерну систему $D_r^{'} \subseteq D_r$ можна отримати наступну функціональну залежність

$$R_{pi} = F_j\{ R_O^{'}, T_i, P_r^{'} \} \qquad (2.4)$$

$$P_i = A_{ri}\{ D_r \}$$

Варто відмітити, що процес виявлення та оцінювання вразливостей, пов'язаних з вимогами можна частково або повністю автоматизувати та формалізувати, оскільки серед сукупності R існує підмножина $R_f \subseteq R$, які підлягають математичному опису та автоматизації. Таке трактування дозволить в перспективі розробити метод виявлення та оцінювання потенційних загроз для web-серверів.

На виході підсистеми S_r одержують сукупність розподілених за типами вимог, які висуваються до конкретної «розумної» комп'ютерної системи. Такий компонент оцінювання можна описати за допомогою наступного виразу

$$S_m = \{ R_p, M, D_s, A, Res \} \qquad (2.5)$$

де R_p – сукупність вхідних даних для компоненту оцінювання, які представляють вимоги до конкретного веб-сервера;

M – сукупність методів, які використовуються при оцінюванні вразливостей;

D_s – сукупність інформації про комп'ютерну систему;

A – сукупність алгоритмів, які можуть бути застосованими в процесі оцінювання безпеки чи вразливостей;

Res – сукупність результатів, які отримано при оцінюванні.

Множину методів оцінювання формують з існуючих функціонально-різних методів, які використовуються для виявлення та оцінювання вразливостей і загроз для програмного забезпечення. До таких методів належать:

– «статичний аналіз програмного коду» [11, 19];
– «функціональне і структурне тестування» [20, 21];
– «засів дефектів» [22].

Проте, серед великої кількості таких методів, важливим та ефективним є метод, що базується на формуванні ядер безпеки. Формально, даний метод можна описати наступним чином.

$$M = \{ SCC, M_O \} \tag{2.6}$$

де SCC – сукупність ядер безпеки, яка застосовується для проведення оцінювання вразливості комп'ютерної системи;

M_O – сукупність допоміжних методів виявлення та оцінювання загроз.

Будь-який метод, який входить до ядра безпеки в процесі оцінювання вразливості web-серверів представляється як

$$M = \{ R, D_m, A_m, Res_m \} \tag{2.7}$$

де R – сукупність вимог, які висуваються до «розумної комп'ютерної системи» та є вхідними даними для опрацювання визначеним методом;

D_m – інформація про комп'ютерну систему, яка потрібна для забезпечення коректності застосування методу;

A_m – сукупність алгоритмів, що формує працездатність методу;

Res_m – сукупність результатів, одержаних внаслідок застосування методу.

Важливо відмітити, що на вхід конкретного методу повинні подаватися дані у наперед визначеному і підтримуваному форматі. На практиці, для трансформації вимог та інформації про комп'ютерну систему та її складові застосовуються спеціальні операції перетворення (конвертори) C_{RM} і C_{DM}, де C_{RM} – операція-функція конвертації вимоги у підтримуваний методом формат, C_{DM} – операція-функція конвертації інформації про комп'ютерну систему для представлення вхідних параметрів.

Загалом перетворення, або по-іншому конвертер, можна описати так, як показано нижче

$$C = \{ D_{in}, F_{in}, A_c, D_{out}, F_{out} \} \tag{2.8}$$

де $\{ D_{in}, F_{in} \}$ – представлення множини вхідних даних у форматі, в якому вони зберігаються у джерелі інформації;

$\{ D_{out}, F_{out} \}$ – сукупність результуючих даних у наперед заданому і підтримуваному форматі;

A_c – алгоритм конвертації формату даних.

Сукупність інформації про комп'ютерну систему, що оцінюється на вразливість, варто також описати у вигляді формальних та неформальних даних

$$D_s = \{ D_{sf}, D_{snf} \} \tag{2.9}$$

де D_{sf} – сукупність формалізованої інформації про комп'ютерну систему;

D_{snf} – сукупність неформалізованої інформації про комп'ютерну систему.

Інформація або дані про систему у формалізованому виді володіють певною структурою (форматом)

$$Form = \{ Form_1, ..., Form_t \} \tag{2.10}$$

де $Form$ – формати даних;

t – кількість форматів.

Формат представлення даних може використовувати різні методи, що формально описується так

$$M_{form} = \{ M_{form_1}, ..., M_{form_m} \} \tag{2.11}$$

де M_{form} – методи представлення даних у визначеному форматі;

m – потужність множини M_{form}.

З іншого боку, елементи M_{form} можна представити у вигляді функцій, тобто вони дають змогу побудувати функціональну залежність. Це означає, що на базі деякої сукупності $D'_{snf} \subseteq D_{snf}$, що представляє собою неформалізовані дані, можна одержати формалізовані елементи $D''_{sf} \subseteq D'_{sf}$

$$M_{form} : D'_{snf} \to D''_{sf} \tag{2.12}$$

або забезпечити представлення даних у наперед встановленому форматі

$$M_{form_k} : D_{sf_i} \to D_{sf_j} \tag{2.13}$$

Множина алгоритмів щодо виявлення та оцінювання вразливостей web-серверів складається з декількох підмножин. До них належить:

– підмножина алгоритмів A_s – дає змогу обрати кращі методи оцінювання;

– підмножина A_f – дозволяє забезпечити формалізацію процесу оцінювання;

– підмножина A_m – включає алгоритми кількісного оцінювання результатів та використання визначених методів;

– підмножина A_n – для формування досі неіснуючих ядер.

$$A = \{ A_s, A_f, A_m, A_n \} \tag{2.14}$$

Результати експертного оцінювання і дані, одержані при застосуванні ядер безпеки необхідно виразити у кількісному вигляді. Для цього потрібно обчислити метрику (и), яка виражає рівень відповідності кожній окремій вимозі. Результат кількісних оцінок, виражений за допомогою метрик, формує деяку множину результатів Res. Ця сукупність є вихідною множиною компоненту оцінювання і водночас є вхідною множиною для підсистеми S_s – компонент аналізу результатів вразливостей та формування звітів з безпеки.

$$S_s = \{Res, A_{res}, D_O, SC \} \tag{2.15}$$

де Res – множина кількісних значень оцінок для окремо взятої вимоги;

A_{res} – сукупність алгоритмів, які застосовуються при опрацюванні вхідних результатів;

D_O – сукупність допоміжної інформації, яка врахована при формуванні результатів;

SC – звіт щодо вразливості та потенційних загроз комп’ютерної системи.

$$A_{res} : Res \to SC \tag{2.16}$$

Характерною особливістю для використовуваної множини алгоритмів при формуванні результатів оцінювання безпеки є згортка метрик згідно пріоритетів вимог. Як наслідок, для сукупності даних D_O потрібно врахувати кількісне значення коефіцієнтів пріоритету. Вони можуть бути задані перед початком процесу оцінювання або уточненими експертами..

Результатом функціонування цього компоненту є сформований звіт щодо вразливостей web-серверів та комп'ютерної системи в цілому, тобто кількісно виражений звіт з безпеки.

Схематично компоненти підходу у вигляді множин та відповідних відношень між ними показано на рис. 2.2.

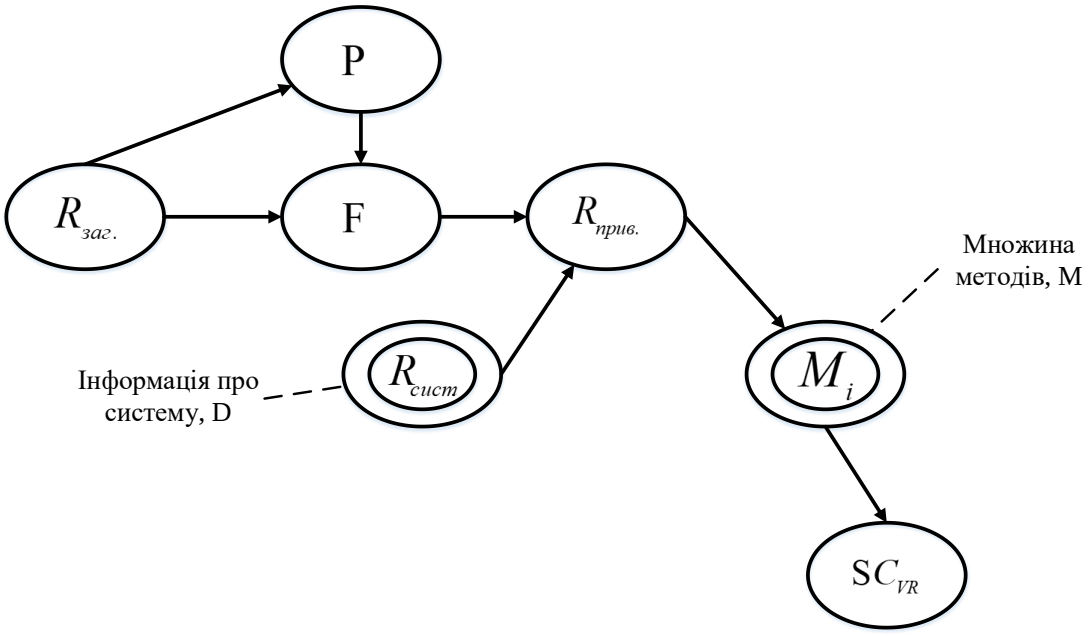

Рис. 2.2. Підхід до виявлення та оцінювання вразливостей із застосуванням теорії множин

Для більш детального представлення процесів виявлення та оцінювання вразливостей, а також формування звіту-висновку з безпеки зображено на рис. 2.3.

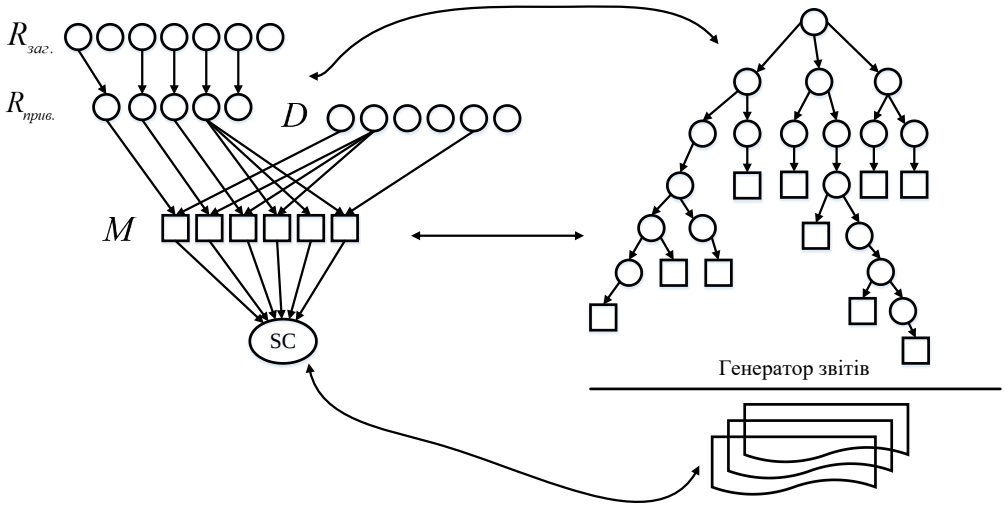

Рис. 2.3. Ієрархічна структура моделі при формуванні звітів щодо вразливостей комп’ютерної системи

Візуалізація повної схеми виявлення та оцінювання вразливостей комп’ютерних систем показана на рис. 2.4.

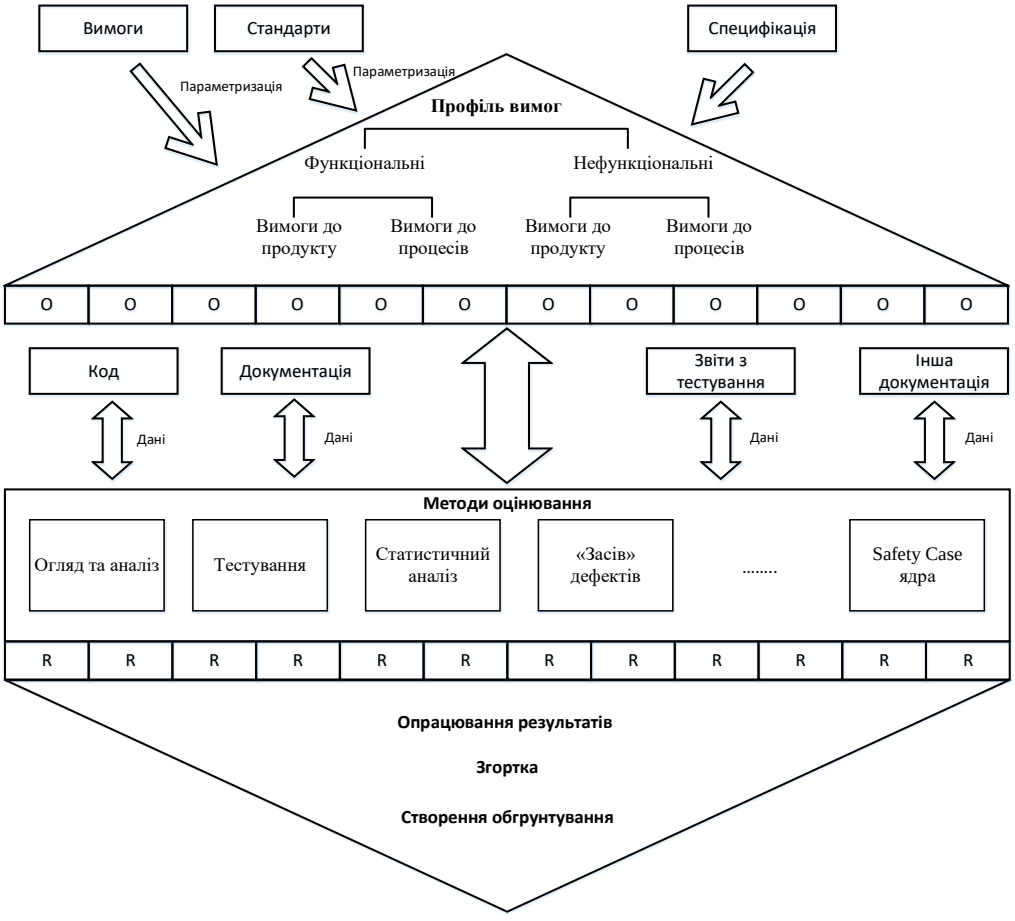

Рис. 2.5. Структура моделі виявлення та оцінювання вразливостей веб-серверів

2.3. Метод виявлення та оцінювання вразливостей «розумних» комп’ютерних систем на основі web-серверів

Вихідними даними для реалізації методу є набір документів з описом вимог до системи, що складається з нормативних документів, які визначають вимоги для конкретного класу систем, стандартів, а також специфікації, складеної для конкретної комп’ютерної системи. Задокументовані обмеження, що накладаються на підхід до реалізації конкретних проектів, також відносяться до вихідних даних. Такими обмеженнями можуть бути, наприклад, обмеження, що накладаються на операційне середовище, технології розробки, інструментальні засоби, вибір елементної бази, залучення персоналу, обмеження, накладені бізнес-логікою проекту. Також вихідними даними можна вважати набір доступних даних про оцінювану систему: дані про структуру, задіяних розробників, процес реалізації функціональності системи, компоненти, що використовуються, результати тестування, проектна і призначена для користувача документація, вихідний і виконуваний код і т.п.

Метод оцінювання вразливостей системи включає етапи, опис яких наведено нижче.

Аналіз і організація вимог до системи.

На цьому етапі на основі загального профілю вимог аналізуються і визначаються ті вимоги, яким повинна відповідати система. Будується приватний профіль вимог:

$$\begin{aligned} &R_p = \{ R_1, R_2, ..., R_n \}, \\ &A_p : R_{заг.} \to R_p \end{aligned} \qquad (2.17)$$

де $R_{заг.}$ – загальний профіль вимог;

R_p – приватний профіль вимог;

A_p – множина алгоритмів, які використовуються для побудови приватного профілю із загального.

Класифікація вимог.

На даному етапі виконується аналіз вимог, класифікація їх за категоріями функціональні та нефункціональні. У даному випадку найбільш важливими з точки зору безпеки системи є нефункціональні вимоги. У результаті виконання цієї стадії одержуємо набір окремих вимог до оцінюваної системи.

Вимоги щодо безпеки системи групуються і після цього проводиться їхнє оцінювання. Рекомендованими є наступні категорії вимог:

- вимоги до програмного забезпечення R_{SW};
- вимоги до апаратного забезпечення R_{HW};
- вимоги, що пов’язані з людино-машинною взаємодією R_H;
- вимоги до структури і процесів розробки R_{SP}.

$$R = R_{SW} \cup R_{HW} \cup R_H \cup R_{SP} \tag{2.18}$$

Альтернативним варіантом групування може бути групування за різними показниками, наприклад:

- вимоги до безвідмовності;
- вимоги до готовності системи;
- вимоги до цілісності;
- вимоги до живучості.

$$\begin{aligned} &G_1 = \{ R_1, R_3, ..., R_k \}, \\ &G_2 = \{ R_2, R_5, ..., R_p \}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ &R = G_1 \cup G_2 \cup ... \cup G_x \end{aligned} \tag{2.19}$$

де $R_1 ... R_n$ – вимоги до безпеки;

$G_1 ... G_x$ – групи вимог, сформульованих за одним з підходів.

Окремою задачею є класифікація вимог за принципом можливості формалізації та автоматизації процесу їх оцінювання. Вимоги, для яких можливе таке оцінювання, об'єднуються у підмножину $R_f \subseteq R$ для подальшого аналізу.

2.3.1. Оцінювання системи на відповідність вимогам безпеки

Це важливий етап, на якому визначається відповідність або невідповідність системи кожній із сформованих вимог приватного профілю. Для цього необхідно провести формальне подання процесу оцінювання за допомогою однієї з нотацій. Для вирішення цієї задачі пропонується використання підходу ядер безпеки, за допомогою яких з’являється можливість прискорити і автоматизувати процес побудови формального представлення і проведення оцінювання на відповідність системи вимогам, забезпечуючи при цьому повноту і достовірність такого оцінювання. На даному етапі виконуються наступні дії.

1) Визначається, які вимоги і процедури виявлення та оцінювання вразливостей відповідають функціональності вже розроблених ядер безпеки. Для цих процесів використовуються існуючі ядра з передачею на вхід необхідних параметрів, визначених, виходячи з вимог до конкретної системи.

2) Після використання ядер, експерт будує формальне представлення і проводить оцінювання системи на відповідність вимогам, що залишились і які не покриті функціональністю існуючих ядер безпеки. Для цього використовуються дані про систему, а також набір відомих методів оцінювання, таких як статистичний аналіз, «засів» дефектів, методи статистичного аналізу, імовірнісний підхід та ін.

Послідовність дій, що проводиться на етапах 1 і 2, може бути описана наступним чином: ядро безпеки або експерт (при відсутності ядра для оцінювання певної функціональності), отримує на вході набір вимог до певної частини або компоненту системи, створює формальне представлення процесу оцінювання, який

необхідно провести, будує візуальне представлення на основі однієї з нотацій, потім добувається та нормалізується інформація по кожній з вимог, а саме: для кожної конкретної вимоги інформація добувається з документації, з коду, з результатів тестування, і т. д, потім ця інформація опрацьовується і за допомогою неї виявляється відповідність системи кожній конкретній вимозі. Таким чином, можна відзначити, що функціональність ядер безпеки є аналогічною роботі експерта з оцінювання окремих вимог, але всі процеси відбуваються автоматично.

3) Наступним рекомендованим кроком є аналіз множини формалізованих вимог R_f і виділення серед них підмножини $R_f^{'}$ – вимоги, які були оцінені експертом без використання ядер. Аналіз результатів оцінювання вимог $R_f^{'}$ експертом на кроці 2, дозволить визначити, яку частину функціональності доцільно реалізувати в додаткових ядрах безпеки для подальшого використання при оцінюванні різних систем. Це дозволяє постійно збільшувати колекцію ядер і в кінцевому підсумку зменшувати кількість ручної праці, при цьому прискорюючи і удосконалюючи процес виявлення та оцінювання вразливостей в цілому.

2.3.2. Побудова обґрунтування безпеки

На підставі результатів оцінювання, отриманої на попередньому етапі, будується звіт про безпеку. Для цього експерт об’єднує результати, отримані при власному оцінюванні, з результатами, сформованими на виході ядер безпеки, які були використані при оцінюванні. На цьому етапі на підставі метрик відповідності системи кожній з вимог, а також заздалегідь заданим ваговим коефіцієнтам для цих вимог, можна зробити адитивну згортку для отримання загального результату.

Кінцевий результат записується у документ обґрунтування безпеки, побудованому для оцінюваної системи.

Ключовою особливістю даного методу є використання при оцінюванні вимог формальних нотацій, а також автоматизація процесу оцінювання за допомогою попередньо розроблених ядер безпеки. Така формалізація та автоматизація процесу

допомагають поліпшити розуміння і зменшити ймовірність помилок експерта, підвищуючи достовірність оцінок в цілому.

2.3.3. Повнота і достовірність оцінок

Повнота оцінювання при використанні представленого методу забезпечується покриттям всіх вимог формальними твердженнями з подальшими їх обґрунтуваннями, побудованими на основі формальних нотацій. Для аналізу покриття вимог будується матриця трасування, що дозволяє встановити зв'язок для кожної вимоги з відповідними їй одним або декількома формальними твердженнями. Вигляд такої матриці наведений у табл.2.1.

Таблиця 2.1

Матриця трасування вимог

Вимога/ Твердження	R_1	R_2	R_3	R_4	R_5	R_6	R_7	…	R_n
C_1	1	1	1	1	1	1	1		1
C_2	1	0	1	0	0	0	1		0
C_3	1	0	1	0	0	0	0		0
C_4	0	1	0	1	1	0	0		1
C_5	0	0	0	1	1	0	0		0
C_6	0	1	0	0	0	0	0		1
C_7	0	0	0	0	0	1	1		0
C_8	0	0	0	0	0	0	0		0
....									
C_k	0	1	0	0	0	0	0		1

Значення в кожній комірці таблиці a_{ij} проставляються виходячи з правила

$$a_{ij} = \begin{cases} 1, \textit{ якщо вимога } R_i \textit{ покривається твердженням } C_j \\ 0, \textit{ в іншому випадку} \end{cases} \quad (2.20)$$

У загальному випадку, таблицю покриття слід будувати таким чином, щоб твердження верхнього рівня покривало всі наявні вимоги до системи. Твердження

більш низьких рівнів повинні покривати повністю хоча б одну з вимог, при цьому одна вимога може бути покрита декількома різними твердженнями.

Побудована таким чином матриця трасування представляє простий і ефективний спосіб простежити зв'язки між вимогами і твердженнями, виявити вимоги, не покриті формальними твердженнями, і покрити ці вимоги з метою забезпечення 100% повноти оцінювання. Матриця також показує, які твердження необхідно оновити в разі зміни вимог до системи.

При необхідності мінімізації сформульованої кількості тверджень слід провести додатковий аналіз, включивши в матрицю трасування тільки твердження нижнього рівня, за умови, що в сформульованому наборі тверджень не існує таких тверджень, які покривають одну з вимог частково.

З метою розв'язку задачі покриття вимог множиною тверджень $MC = \{ C_q \}$ і визначення мінімальних систем тверджень SC_k, що належать MC, необхідно знайти мінімальне покриття всіх вимог тверджень. Для цього потрібно отримати досконалу диз'юнктивну нормальну форму функції $f(C_1, C_2, ..., C_q)$, побудувавши її таблицю істинності.

Вигляд таблиці істинності представлений у табл. 2.2. Значення функції, рівне 1, відповідає комбінації тверджень, яка покриває всі вимоги до системи.

Для побудови ДДНФ розглядаються ті твердження, при яких функція рівна 1. При цьому змінна записується з інверсією, якщо її значення дорівнює 0, і без інверсії, якщо значення дорівнює 0. Результуюча ДДНФ буде представлена диз'юнкцією всіх отриманих елементарних кон'юнкція

$$f(C_1, C_2, ..., C_q) = \neg C_1 \cdot C_2 \cdot ... \cdot \neg C_q \vee \neg C_1 \cdot C_2 \cdot ... \cdot C_q \vee ... \vee C_1 \cdot C_2 \cdot ... \cdot C_q \qquad (2.21)$$

Далі здійснюється перехід від отриманої ДДНФ до скороченої форми шляхом перетворень, заснованих на операціях «склеювання» і «поглинання». Результат виконання перетворень призведе до отримання скороченої ДНФ вигляду

$$f(C_1,C_2,...,C_k)=\neg C_1\cdot C_3\cdot...\cdot C_k\vee...\vee C_2\cdot...\cdot C_j \qquad (2.22)$$

Члени, одержаної скороченої форми (елементарні кон'юнкції тверджень) є простими імплікантами функцій.

Таблиця 2.2

Матриця трасування вимог

C_1	C_2	C_3	…	C_q	$f(C_1,..... C_q)$
0	0	0		0	0
0	0	0		1	0
0	0	1		0	0
0	0	1		1	0
0	1	0		0	0
0	1	0		1	0
0	1	1		0	1
0	1	1		1	1
1	0	0		0	0
				…	…
1	1	1		1	1

Прості імпліканти використовуються для побудови матриці імплікацій, яка дозволяє здійснити перехід від скороченої ДНФ до мінімальної форми

$$f(C_1,C_2,...,C_k)=C_2\cdot...\cdot C_j\vee...\vee C_i\cdot...\cdot C_k \qquad (2.23)$$

Отримані таким чином мінімальні системи тверджень дозволяють в результаті вибрати оптимальну за заданим критерієм систему тверджень SC_{opt}

Приріст достовірності досягається за рахунок зменшення кількості помилок при проведенні оцінювання. Цей приріст можна оцінити, ввівши додатковий коефіцієнт скорочення кількості помилок K. У загальному випадку цей коефіцієнт буде визначатися таким відсотковим виразом

$$K = (1 - \frac{N_{new}}{N_{initial}}) * 100\% \quad (2.24)$$

де $N_{initial}$ – кількість помилок, які виявлено без використання методу;

N_{new} – кількість помилок виявлених з використання методу.

Представлений метод передбачає досягнення високих показників достовірності за рахунок формалізації і автоматизації процесу оцінювання. Для таких цілей необхідно визначити множину вимог R_f перевірку на відповідність яким можна формалізувати і частково або повністю автоматизувати. Наступний крок буде полягати в автоматизованому покритті цих вимог відповідними інструментальними засобами. Якщо припустити, що для перевірки кожної вимоги використовується визначений метод, то кількість помилок можна розрахувати за формулою

$$N = \sum_{m=1}^{M} \frac{|R_m|}{|R_p|} Q_m \quad (2.25)$$

де Q_m – ймовірність помилки, при використанні *m*-го методу;

R_m – множина вимог, оцінених за допомогою *m*-го методу;

M – множина використаних методів оцінювання.

З метою оцінювання можливого приросту достовірності при використанні запропонованого методу в порівнянні з використанням ручної праці експерта, будемо умовно вважати ймовірність помилок інструментальних засобів однаковою і рівною Q_{auto}, ймовірність помилки експерта рівна Q_{ex}. І запишемо наступні вирази для допустимої кількості помилок

$$N_{new} = |R_{auto}| \cdot Q_{auto} + |R_{ex}| \cdot Q_{ex}, \quad (2.26)$$

$$N_{initial} = |R_p| \cdot Q_{ex}$$

де R_p – приватний профіль вимог;

R_{auto} – множина вимог приватного профілю, що покрита інструментальними засобами, $R_{auto} \subseteq R_f$;

R_{ex} – множина вимог приватного профілю, що не покрита інструментальними засобами та визначається експертом вручну, $R_{ex} = \dfrac{R_p}{R_{auto}}$.

У такому випадку формула розрахунку коефіцієнта зменшення кількості помилок (1.34) матиме вигляд

$$\begin{aligned} K &= \left(1 - \frac{|R_{auto}| \cdot Q_{auto} + |R_{ex}| \cdot Q_{ex}}{|R_p| \cdot Q_{ex}}\right) * 100\% = \\ &= \left(1 - \frac{|R_{auto}| \cdot Q_{auto} + |R_p - R_{auto}| \cdot Q_{ex}}{|R_p| \cdot Q_{ex}}\right) * 100\% = \\ &= \left(1 - \frac{|R_{auto}| \cdot Q_{auto}}{|R_p| \cdot Q_{ex}} - 1 + \frac{|R_{auto}|}{|R_p|}\right) * 100\% = \frac{|R_{auto}|}{|R_p|}\left(1 - \frac{Q_{auto}}{Q_{ex}}\right) * 100\% \end{aligned} \quad (2.27)$$

Наведена формула (2.27) показує, що коефіцієнт зменшення кількості помилок при використанні запропонованого методу залежить від двох основних складових:

– відношення потужності множини або по суті кількості вимог, покритих автоматичними засобами оцінювання, до загальної кількості вимог приватного профілю;

– значення ймовірності помилкової роботи автоматичних засобів в порівнянні із значенням показника ймовірності помилки при ручному оцінюванні.

Виграш щодо зменшення кількості помилок буде тим вище, чим більша частина вимог оцінюється автоматичними методами і чим менша ймовірність помилок автоматичних засобів в порівнянні з ручним оцінюванням експерта. Максимально безпомилкової буде оцінка при $R_{auto} \to R_p$ і $Q_{auto} \to 0$, тобто при

повному покритті всіх вимог приватного профілю автоматичними абсолютно безпомилковими засобами.

Однак, аналіз реальних проектів показав, що потенційно формалізуються і автоматично перевіряються приблизно 80-85% всіх вимог, тобто потенційно існує можливість створити інструментальні засоби для формалізації і автоматизації процесу оцінювання систем на відповідність 80-85% приватного профілю вимог. Значення достовірності розрахунків автоматичними методами є вищим по відношенню до достовірності розрахунку вручну, за рахунок виключення помилок, обумовлених людським фактором.

З метою кількісного оцінювання максимально можливого приросту достовірності при використанні запропонованої моделі та методу, достовірність розрахунків автоматичними засобами можна умовно прийняти рівною 1. Імовірність помилки експерта варіюється в залежності від складності проведених розрахунків, але на практиці стандартом вважається значення помилки, що не перевищує 5%. Тому в якості максимально допустимого показника помилкової оцінки експерта можна прийняти граничну помилку рівну 0.05, і відповідно, достовірність розрахунків експерта рівну 0.95. Всі вимоги для зручності розрахунку будемо вважати рівнозначними.

При таких припущеннях початкове значення загального показника якості експертизи при використанні виключно ручної праці для оцінювання досліджуваної системи на відповідність усім вимогам приватного профілю, складе $QE_1 = 0{,}95$.

У той же час при використанні запропонованої моделі та методів досягається можливість підвищення загальної якості експертизи до значення $QE_1 = 0{,}85 * 1 + 0{,}15 * 0{,}95 = 0{,}9925$, що відповідає приросту значення загальної якості експертизи на 4.47%.

Коефіцієнт зменшення кількості помилок при дпних припущеннях складає $K = \frac{|R_{auto}|}{|R_p|}\left(1 - \frac{Q_{auto}}{Q_{ex}}\right) * 100\% = 0{,}85 * \left(1 - \frac{0}{0{,}05}\right) * 100\% = 85\%$, що є доволі високим показником.

На практиці одержання максимального виграшу щодо підвищення достовірності та загальної якості експертизи є досить складним завданням, оскільки досить часто відбувається покриття автоматичними засобами не тільки множини вимог R_f, а тільки окремої його частини $R_f' \subset R_f$, а ймовірність помилок використовуваних інструментальних засобів завжди відмінна від нуля. В цілому, можна стверджувати, що запропонований метод є ефективним і дозволяє досягти поставленої мети щодо підвищення достовірності та автоматизації процесу оцінювання функціональної безпеки інформаційних систем.

Для підвищення ефективності процесу виявлення та оцінювання вразливостей комп'ютерних систем запропоновано використання ядра безпеки і безпечної інфраструктури, що дозволяють формалізувати, уніфікувати і частково автоматизувати процес визначення оцінок функціональної безпеки. Обґрунтовано процес створення ядер, особливості реалізації, використання даних, внутрішня структура ядер і принципи їх роботи. Побудовано теоретико-множинну модель виявлення та оцінювання функціональної вразливості комп'ютерних систем, яка є параметризованою, базується на формальній системі і дозволяє уніфікувати і формалізувати процес оцінювання відповідності інформаційних систем нормативним вимогам. Представлений метод оцінювання функціональної безпеки інформаційних систем з використанням параметризованих ядер безпеки, який дозволяє найбільш ефективно використовувати запропоновану модель оцінювання, забезпечити повноту оцінювання та частково автоматизувати процес побудови документа ядра безпеки.

2.4. Висновки до розділу

1. Досліджено особливості виявлення та оцінювання вразливостей веб-серверів у комп’ютерних системах і як результат обгрунтовано використання методології ядер безпеки, удосконалено функціональні можливості цього підходу за рахунок формалізованого представлення критеріїв захищеності, практичного застосування формальних і графічних нотацій та здатності до автоматизації процесів виявлення та оцінювання загроз. Це дало змогу забезпечити високу ефективність у порівнянні з існуючими методами виявлення та оцінювання загроз.

2. Удосконалено і доповнено модель виявлення та оцінювання вразливості веб-серверів у «розумних» комп’ютерних системах за допомогою ядер безпеки, що дало можливість забезпечити структурованість та автоматизацію розрахунку параметрів вразливостей веб-серверів, а також наростити значення достовірності і точності експертних оцінок.

3. Розроблено метод виявлення та оцінювання вразливостей веб-серверів у «розумних» комп’ютерних системах за рахунок побудови профілю вимог вразливості веб-серверів та його формального представлення у вигляді нотацій теорії множин, що в перспективі дало змогу автоматизувати процес формування ядер безпеки.

4. Запропоновані рішення щодо виявлення та оцінювання вразливостей надали можливість оцінити захищеність веб-серверів і сформували базис для розвитку методів та інструментів прогнозування і запобігання загрозам.

РОЗДІЛ 3
ЗАСІБ АВТОМАТИЗАЦІЇ ТА РЕЗУЛЬТАТИ ВИЯВЛЕННЯ ТА ОЦІНЮВАННЯ ВРАЗЛИВОСТЕЙ ВЕБ-СЕРВЕРІВ У РОЗУМНИХ КОМП'ЮТЕРНИХ СИСТЕМАХ

3.1. Аналіз домену та функціональних вимог щодо системи автоматизації процесів виявлення та оцінювання вразливостей комп'ютерних систем

Автоматизація процесу виявлення та оцінювання вразливості веб-серверів у розумних комп'ютерних системах за допомогою обґрунтованої моделі та методу передбачає побудову рольової моделі, тобто визначення та опис функціональних можливостей фахівців при виконанні визначених процесів. Головними «акторами» у таких процесах є:

- фахівець з виявлення та оцінювання загроз;
- експерт у галузі безпеки комп'ютерних систем.

Найбільш важливі функціональні можливості фахівця з виявлення та оцінювання вразливості веб-серверів показано на рис. 3.1. Опишемо більш детально функціональні можливості цього фахівця щ точки зору виявлення та оцінювання вразливостей веб-серверів.

Для того, щоб забезпечити повноту сукупності потенційних загроз і вразливостей комп'ютерних систем на основі веб-серверів потрібно надати доступ до open source баз даних ы знань, наприклад, CERT, CVE і т.п.. З метою забезпечення такого доступу, запропоновано скористатися технологією API або використати провайдери віддаленого доступу. При цьому важливо переконатися у їх доступності і працездатності, тобто виконати тестування шляху підключення до баз даних або перевірку доступності і коректності API інтерфейсів.

Не менш важливою функціональністю, якою потрібно забезпечити фахівця з виявлення та оцінювання загроз веб-серверів є автоматизація побудови ядер безпеки. Це забезпечить можливість отримати кількісні значення оцінок за визначеними вимогами безпеки програмної складової КС. Для цього необхідно

провести аналіз існуючих інструментів автоматизації щодо перевірки на вразливість програмного забезпечення та критерії за якими одержують кількісну оцінку. Після цього інженер із безпеки програмного забезпечення повинен забезпечити інтеграцію існуючих засобів із системою управління процесом оцінювання захищеності web-серверів.

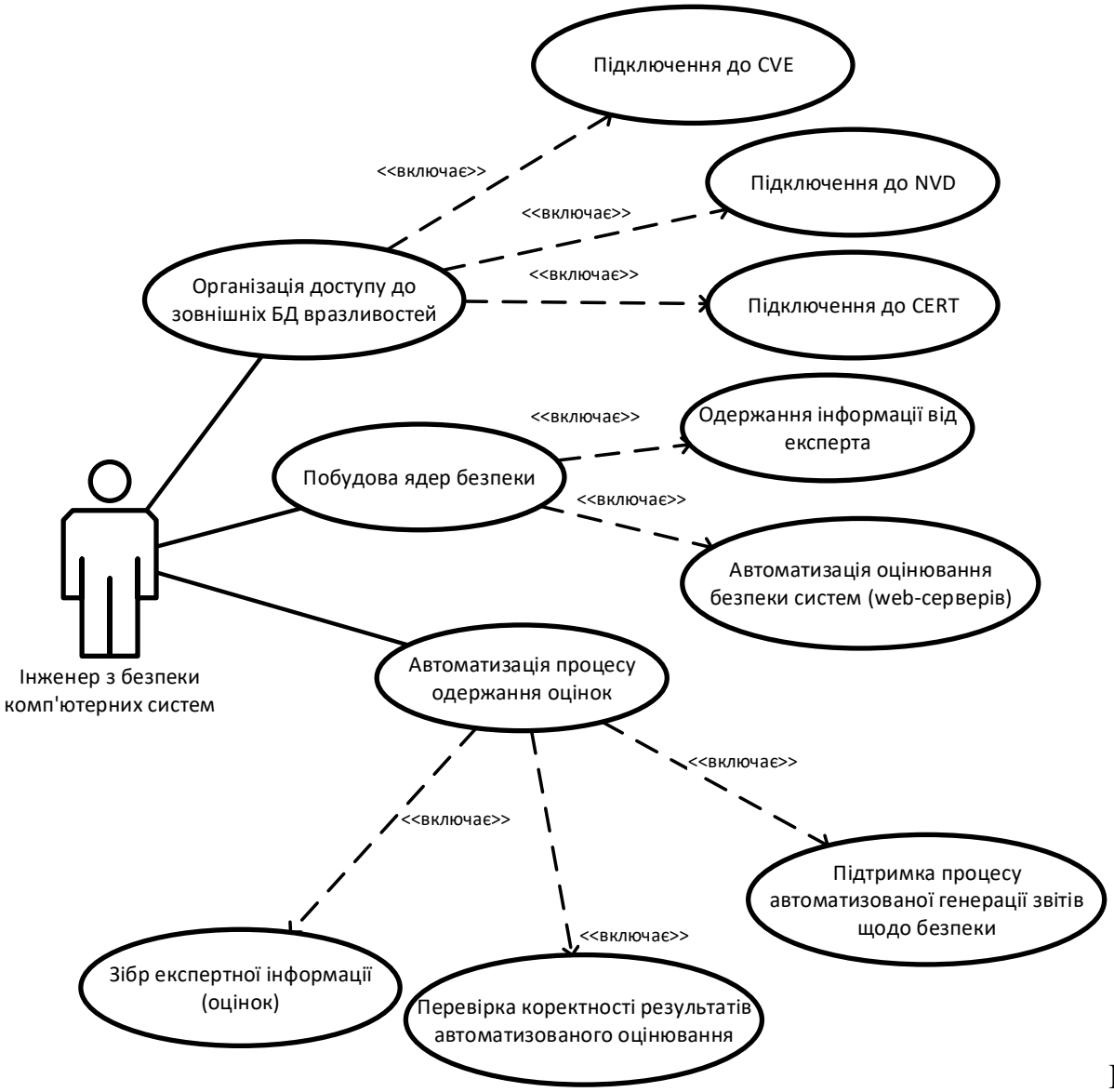

Рис. 3.1. Діаграма варіантів використання для представлення функціональних можливостей фахівця з безпеки

Ще однією важливою функцією інженера з безпеки комп’ютерних систем є автоматизація процесу одержання оцінок, яка включає в себе автоматизовану

генерацію звіту щодо безпеки програмної чи комп’ютерної системи, перевірку коректності даних, одержаних у результаті автоматизованого оцінювання безпеки web-серверів, а також збір експертної інформації, зокрема оцінок експерта, які одержано внаслідок ручної перевірки безпеки програмного забезпечення.

Функції експерта, що потребують автоматизації під час виявлення та оцінювання вразливості веб-серверів наведено на рис. 3.2 у вигляді use case діаграми.

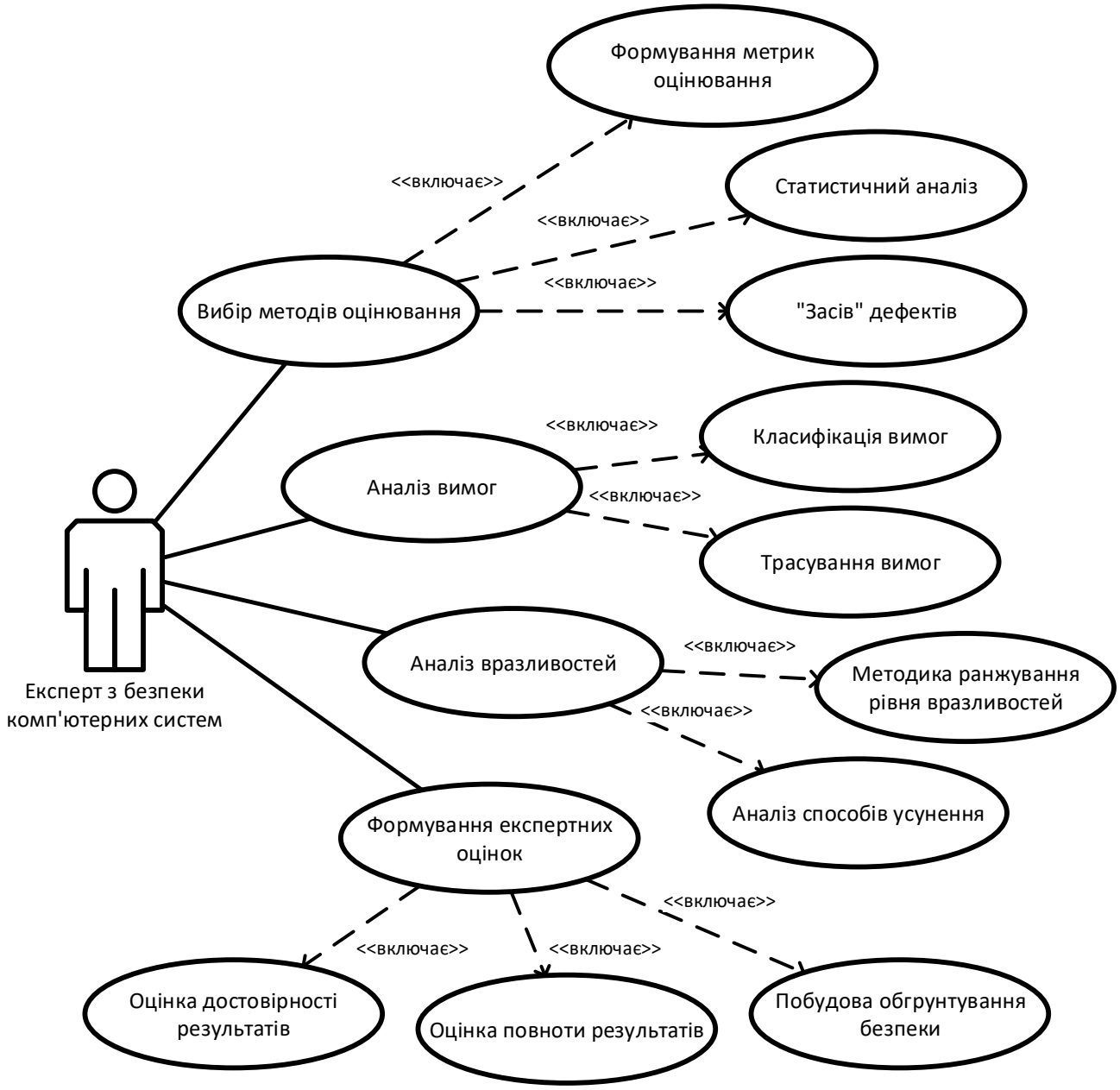

Рис. 3.2. Діаграма варіантів використання «Функції експерта з безпеки»

Основні функціональні вимоги, які необхідно реалізувати у системі підтримки оцінювання безпеки web-серверів, експертом полягають в автоматизованому виборі методів і відповідних критеріїв оцінювання, аналізі вимог та потенційних вразливостей до web-серверів, їх трасування, а також формування експертних оцінок за визначеними характеристиками безпеки і надійності.

Вибір методів оцінювання безпеки web-серверів включає в себе визначення та обґрунтування метрик безпеки, визначення вхідних даних і застосування статистичних методів оцінювання, побудова алгоритму «засіву дефектів» та ін. При цьому, в перспективі, необхідно передбачити побудову бази даних або бази знань методів залежно від обраного методу.

Аналіз вимог передбачає їх класифікацію за функціональними та нефункціональними критеріями. Безпека та захищеність відноситься до класу характеристики надійності програмного забезпечення, а відповідно надійність є комплексною нефункціональною вимогою до програмного забезпечення. Хоча надійність тісно пов’язана із функціональністю програмного забезпечення. Важливим також на етапі аналізу вимог є їх трасування на структурні компоненти програмного забезпечення. Трасування вимог дає змогу виявити коректне чи некоректне виконання функцій певними модулями оцінюваної системи, а також виявляти загрози та вразливості компонентів web-сервера.

Аналіз вразливостей дає змогу експерту встановити потенційно уражені компоненти web-серверів, а також автоматизувати процес їхнього ранжування за визначеною методикою.

Аналіз вразливостей тісно пов’язаний із способами їх усунення. Для цього у системі підтримки оцінювання безпеки web-серверів необхідно також передбачити зв’язок між базою даних вразливостей та базою даних способів їх усунення.

Формування експертних оцінок є складним процесом, що передбачає оцінювання та визначення кількісних критеріїв для встановлення повноти і достовірності результатів оцінювання безпеки web-серверів. При цьому важливим

етапом є автоматизація процесу побудови обґрунтувань щодо безпеки web-серверів.

Визначивши основні функціональні вимоги та спроектувавши базу даних для зберігання інформації про вразливості та загрози web-серверів потрібно спроектувати архітектуру інструментального засобу.

3.2. Проектування схеми бази даних щодо виявлення та оцінювання вразливості веб-серверів

Проведення кількісного оцінювання показників надійності та функціональної безпеки програмних компонентів і систем передбачає роботу з інформацією. З метою систематизації, зберігання і подальшої обробки інформації можна використовувати реляційні бази даних. Ключовим моментом є попереднє коректне проектування бази даних: вона повинна включати необхідні відношення, збережені процедури, атрибутам, у базі даних повинні бути проставлені індекси і задані зв’язки між реляційними відношеннями. Організація бази даних, структура її відношень і атрибутів повинні бути зручними для подальшої роботи з інформацією, автоматичної обробки даних та оцінювання різних характеристик.

При оцінюванні OTS компонентів доцільним є побудова бази даних з урахуванням аналізу семантичної інформації з відкритих джерел даних про уразливість. На рис 3.3 показана модель бази даних у вигляді діаграми "сутність-зв’язок", що відображає логічне представлення і взаємозв’язок основних відношень, що використовуються у процесі оцінювання характеристик готових програмних продуктів.

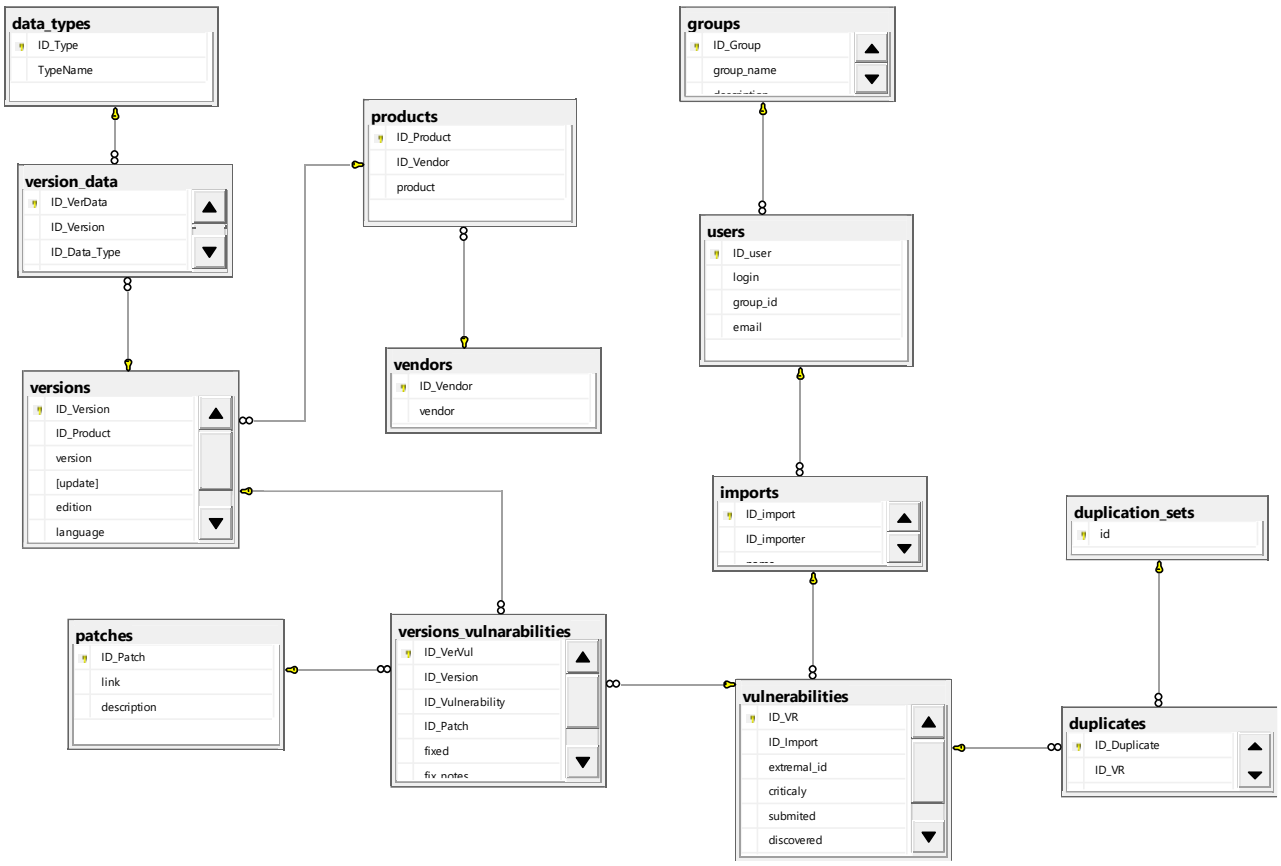

Рис. 3.3. Схема бази даних для зберігання та керування загрозами

Опишемо коротко відношення і зв’язки, представлені на діаграмі (рис. 3.3). Ключовим є відношення "вразливості" ("vulnerabilities"), яке включає в себе інформацію про уразливість, таку як ідентифікатор уразливості в розробленій базі даних, ідентифікатор уразливості в CVE або інших ресурсах (якщо такий є), опис уразливості на природній мові, рейтинг критичності , дату виявлення і дату появи інформації про уразливість у відкритому доступі, а також ідентифікатор користувача, який повідомив про знайдену уразливість, і ідентифікатор імпорту, під час якого цю уразливість внесли в базу даних.

Імпорт (відношення "imports") необхідний для того, щоб користувачі мали можливість використовувати свої власні дані про помилки при роботі з базою даних. Розроблена додатково утиліта дозволяє імпортувати дані у вигляді файлів, представлених в csv форматі. Такі файли автоматично розбираються і добута інформація записується у базу даних. Відношення «вразливость» ("vulnerabilities") пов’язане з відношенням "версії продуктів" ("productversion") і відповідно з

відношенням продукти (products) через додаткове відношення «версії-вразливості» ("versions_vulnerabilities ").

Крім ідентифікатора вразливостей і продуктів, це додаткове відношення може містити дату виправлення вразливостей, яка, на жаль, відсутня практично у всіх ресурсах вразливостей, розглянутих вище.

На нашу думку, поле дата є дуже важливим при оцінюванні надійності і безпеки програмних продуктів, так як вона дозволяє відслідковувати моменти виправлення вразливостей, а також оцінювати час, що витрачається компаніями на виправлення виявлених вразливостей, тим самим визначаючи показник часу відновлення програмного продукту, адже виявлення уразливості є по суті потенційною відмовою системи безпеки ПЗ. Для зберігання дати усунення вразливості щодо «версії-уразливості» створено атрибут "виправлено" ("fixed"). Додатково в базі даних створено відношення «типи даних» ("datatypes") і відношення «версії даних» ("version_data"), що зв’язує його з відношенням версій продуктів. Це дозволяє не обмежуватися інформацією про уразливість і додавати в базу даних інформацію різних типів, доступну для програмних продуктів.

Такою інформацією можуть бути файли вихідного коду і результати внутрішнього тестування (для відкритих компонентів), дані про кваліфікацію розробників, популярність ПЗ і т.п. Для зберігання інформації про зареєстрованих та незареєстрованих користувачів системи існує відношення "користувачі" ("users"). Відношення "групи" ("groups") містить інформацію про групи, до яких можуть належати користувачі, що дозволяє налаштовувати права доступу і безпеку в системі.

3.3. Проектування архітектури системи автоматизації

У загальному випадку архітектуру системи підтримки процесу оцінювання безпеки web-серверів можна представити за шарами Фаулера, як показано на рис. 3.5.

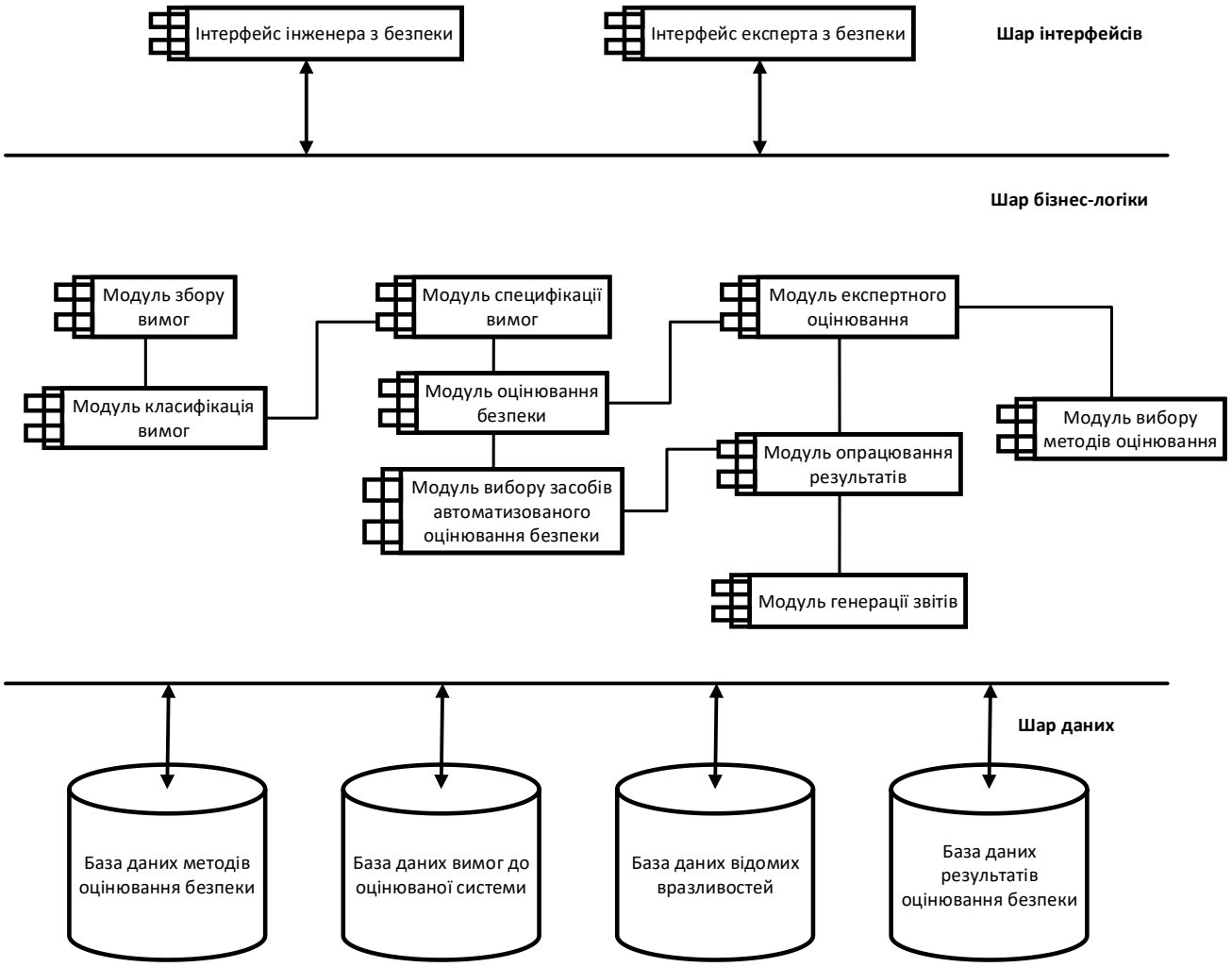

Рис. 3.5. Архітектура системи підтримки процесу оцінювання безпеки web-серверів

На рівні шару користувацьких інтерфейсів визначено два інтерфейси – інтерфейс інженера з безпеки та інтерфейс експерта.

Інтерфейс інженера з безпеки програмного забезпечення повинен реалізовувати функції, які наведено на рис. 3.2 і мати відповідні елементи керування для виконання функціональних обов’язків інженера. До основних функціональних обов’язків інженера з безпеки програмного забезпечення відноситься:

– організація доступу до зовнішніх баз даних вразливостей;
– побудова Safety Case ядер;

– автоматизація процесу одержання кількісних оцінок щодо критеріїв безпеки web-серверів.

Інтерфейс експерта з безпеки програмного забезпечення повинен забезпечувати виконання наступних функціональних можливостей:

– вибір методів оцінювання безпеки web-серверів;

– аналіз вимог;

– аналіз вразливостей;

– формування експертних оцінок.

У шарі бізнес-логіки передбачено модуль для збору вимог до програмного забезпечення, в даному випадку до web-серверів. Він передбачає логіку запису та редагування даних щодо вимог безпеки до програмного забезпечення в базу даних і передачу цієї сукупності у модуль класифікації вимог.

Модуль класифікації вимог може бути реалізований як інтелектуальний модуль з реалізацією можливості автоматичної класифікації на функціональні та нефункціональні вимоги, а після цього класифікації нефункціональних вимог за критеріями безпеки web-серверів. Однак необхідно також передбачити можливість ручної класифікації вимог експертом або інженером з безпеки програмного забезпечення. Тому між шаром представлення (користувацьких інтерфейсів) передбачено прямий зв’язок відповідних інтерфейсів з модулем оцінювання у шарі бізнес-логіки.

Модуль специфікації вимог безпеки до web-серверів призначений для структурування та фіксації інформації щодо критеріїв надійності за функціональними блоками та компонентами.

Модуль оцінювання безпеки дає змогу проводити оцінювання безпеки web-серверів на основі вимог безпеки із застосуванням як автоматизованих засобів визначення мір відповідних критеріїв, так і експертним шляхом. Модуль оцінювання є проміжним модулем, що забезпечую зв’язок між вимогами до безпеки, відомими вразливостями, експертними оцінками та методами оцінювання.

Модуль автоматизованого оцінювання передбачає наявність способів і методів доступу до інструментальних засобів автоматизованого визначення критеріїв оцінювання безпеки та одержання і запис у відповідну базу даних оцінок щодо безпеки компонентів чи вимог.

Модуль експертного оцінювання реалізує можливість вибору методів із вказанням відповідних початкових параметрів методів, забезпечує ручне оцінювання відповідності рівня безпеки певного web-сервера, здійснює структурування даних до автоматизованого генерування звітів з оцінювання.

Модуль вибору методів оцінювання безпеки web-серверів надає засоби налаштування параметрів методів і відповідних алгоритмів, а також процедури опрацювання вхідних даних.

Модуль опрацювання результатів дає змогу оцінювати повноту і достовірність одержаних оцінок як автоматичним шляхом, так і за допомогою експертного оцінювання.

Модуль генерації звітів дозволяє згенерувати документацію щодо результатів оцінювання безпеки web-серверів, які визначені автоматичним шляхом і експертним оцінюванням.

База даних методів оцінювання безпеки містить назви методів оцінювання безпеки, статичні параметри для ефективної їх роботи.

База даних вимог до оцінюваної системи зберігає вимоги до web-серверів, які стосуються їх безпеки.

База даних вразливостей містить посилання на зовнішні ресурси та бази даних щодо загроз web-серверів.

У базі даних результатів оцінювання зберігається інформація про критерії оцінювання безпеки і їх кількісні міри.

3.4. Процедура оцінювання вразливості різних типів веб-серверів

Одним з можливих варіантів оцінювання та порівняння, аналогічних по функціональності програмних продуктів, є оцінка загальної (кумулятивної) кількості знайдених в них вразливостей за певні проміжки часу на різних стадіях експлуатації.

Однією з базових характеристик надійності є загальна (кумулятивна) кількість N_i - відмов ПЗ до моменту часу $t = t_i \cdot i$

$$N_i = n_1 + n_2 + ... + n_i \tag{3.1}$$

де n_i – кількість відмов ПЗ на і-му інтервалі тестування;

t_i – тривалість часу інтервалу;

t – загальний час тестування.

Вважаючи виявлення уразливості відмовою системи безпеки web-сервера, доцільно ввести метрику "кількість виявлених вразливостей ПЗ" і проводити розрахунок значення цієї метрики при аналізі безпеки програмних продуктів.

Аналіз кумулятивної кількості виявлених вразливостей можна провести за допомогою логістичної кривої, яка представляє собою сигмоїдальну криву і є гладкою монотонно зростаючою всюди диференціюється S-подібною нелінійною функцією.

В якості аналітичного виразу для логістичної функції будемо використовувати такий вираз

$$f(t) = \frac{\alpha}{1 + e^{-\frac{t-\mu}{s}}} \tag{3.2}$$

де α – параметр, що відповідає за загальну кількість вразливостей;

s – параметр нахилу логістичної кривої, зміна якого дозволяє побудувати функції з різною крутизною;

μ – медіана, параметр, що визначає положення кривої по осі абсцис.

В якості порівнюваних web-серверів були обрані версії програмних продуктів «Apache» і «Internet Information Services (IIS)». Результат апроксимації кумулятивних профілів відмов обраних програмних продуктів за допомогою логістичної функції представлений на рис. 3.6.

Апроксимація кумулятивного профілю відмов з використанням логістичної кривої, яку також називають кривою життєвого циклу, дозволяє визначити, на якій стадії розвитку перебуває досліджуваний програмний продукт. Для цього можна умовно виділити три періоди еволюції продукту: початкова стадія, активний розвиток і період зрілості. На початковій стадії зростання загальної кількості виявлених вразливостей приблизно відповідає експоненті (степенева функція), в період розвитку зростання переходить в лінійну фазу, і потім на етапі зрілості продукту практично зупиняється.

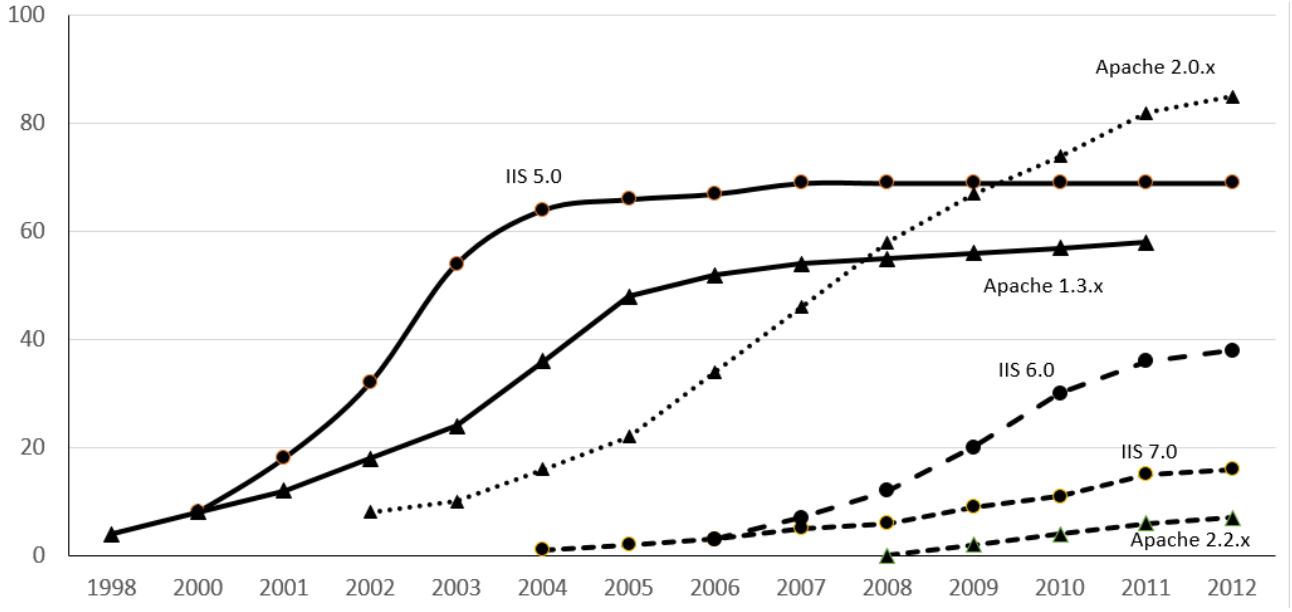

Рис. 3.6. Апроксимація кумулятивного профілю відмов з використанням логістичної кривої

Дослідження логістичної функції дозволяє визначити межі цих етапів. Аналітичний вираз першої похідної функції є функцією щільності розподілу ймовірності з урахуванням параметра α кумулятивної кількості вразливостей.

Графік кумулятивної функції розподілу (логістичної функції) і відповідної функції щільності розподілу логістичної функції наведено на рис. 3.7.

Математичне сподівання μ функції щільності розподілу ймовірності відповідає значенню середини інтервалу активного розвитку продукту на S-подібній кривій, з границями цього інтервалу в точках $[\mu-\sigma;\mu+\sigma]$, де σ – середньоквадратичне відхилення. Значення середньоквадратичного відхилення, виражається через параметр s логістичної функції $f(x)$ наступним чином

$$\sigma=\frac{\pi}{\sqrt{3}}s \tag{3.3}$$

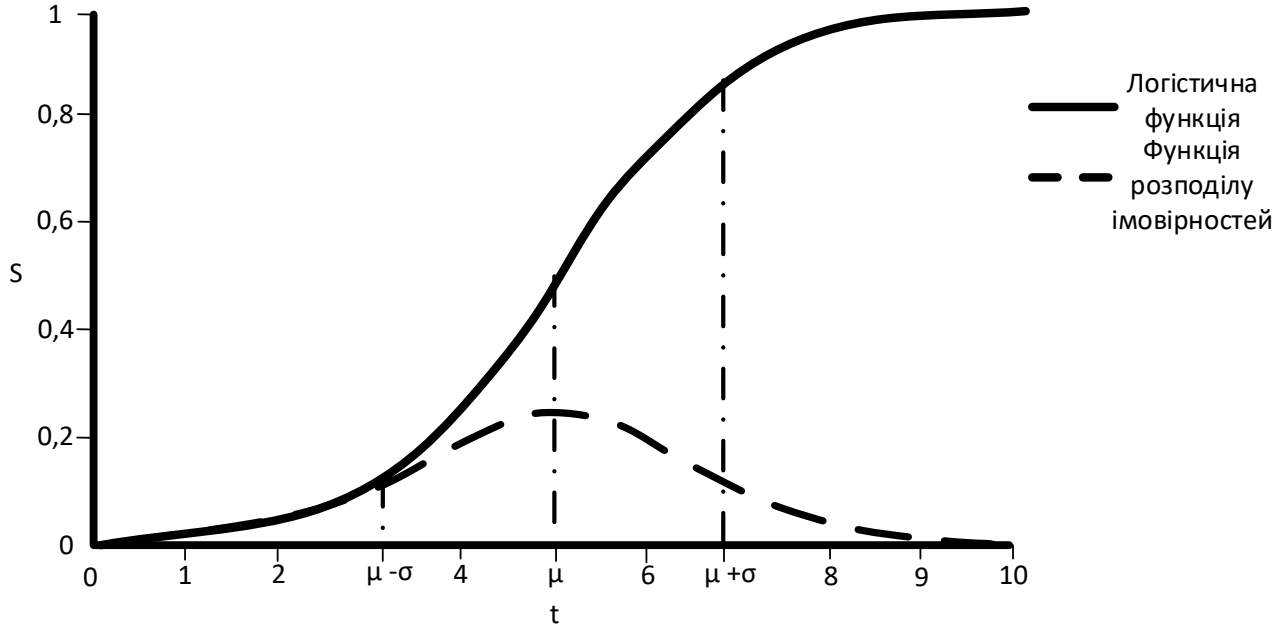

Рис. 3.7. Логістична функція та функція розподілу її щільності імовірності

При виборі web-серверів перевагу варто віддавати програмним продуктам, які пройшли етапи початкового та інтенсивного розвитку і знаходяться в на стадії

зрілості. Це дозволить мінімізувати ймовірність виявлення нових вразливостей при їх використанні.

З метою формального визначення стадії розвитку, а отже, очікуваної стабільності функціонування продукту за кумулятивним профілем відмов, апроксимованого за допомогою логістичної функції, введемо додаткову метрику τ - коефіцієнт вразливостей ПЗ. Значення показника оцінюється наступним виразом:

$$\tau = \frac{t - \mu}{\sigma} \tag{3.4}$$

де t – момент часу дослідження;

μ – значення середини інтервалу активного розвитку продукту;

σ – середньоквадратичне відхилення.

На основі значення показника τ можна зробити висновки про поточну стадію продукту і можливий подальший розвиток:

– $\tau < -1$ характеризує початкову стадію, висока ймовірність виявлення великої кількості вразливостей в майбутньому;

– $\tau \in [-1;0)$ характеризує першу половину стадії інтенсивного розвитку продукту, на якій буде знайдена основна кількість вразливостей, очікується активне зростання кількості виявлених вразливостей;

– $\tau \in [0;1]$ характеризує другу половину стадії інтенсивного розвитку продукту, ймовірність виявлення нових вразливостей зменшується;

– $\tau > 1$ характеризує стадію зрілості продукту, низька ймовірність виявлення нових вразливостей.

Чим більше значення показника τ, тим більш стабільним є програмний продукт і тим краще його використання в порівнянні з іншими аналогами. Таким чином, обчислення даної метрики дозволяє визначити стадію розвитку, на якій знаходиться досліджуваний web-сервер, прогнозувати рівень виявлення нових

вразливостей в майбутньому, а також порівняти аналогічні програмні продукти з метою вибору найкращого варіанту.

Значення показника τ, розрахованого для розглянутих Apache і IIS компонентів, представлено у таблиці 3.1.

Далі необхідно зазначити, що аналіз кількості виявлених вразливостей і порівняння кумулятивних профілів відмов є важливим, але в той же час досить поверхневим методом оцінювання, що не дозволяє зробити точних висновків про безпеку розглянутих програмних продуктів. Кожна вразливість індивідуальна, і ступінь серйозності потенційного впливу на вразливу систему дуже відрізняється для різних видів вразливостей.

Таблиця 3.1

Коефіцієнти вразливостей, обчислені для web-серверів Apache та IIS

Програмний продукт	Коефіцієнт вразливості, г
Apache 1.3	3.2
Apache 2.0	1.8
Apache 2.2	2.1
IIS 5.0	6.1
IIS 6.0	0.9
IIS 7.0	1.7

Необхідно враховувати рівень серйозності загрози для кожної виявленої вразливості при визначені оцінки. Крім того, важливим показником є швидкість виходу оновлення для нейтралізації загрози в системі безпеки. У зв’язку з цим для більш детального аналізу пропонується оцінювати такі важливі показники захищеності ПЗ як рівень критичності відмов і середній час відновлення після відмови. Під відмовою в даному випадку розуміється виявлення уразливості (відмова системи безпеки ПЗ).

Однією з проблем виявлених загроз є те, що критичність вразливостей періодично варіюється в залежності від джерел інформації, тому що різні джерела використовують різні системи оцінювання. В даному випадку пропонується дотримуватися універсальної системи оцінок CVSS, у зв'язку з чим, всі значення, виставлені в інших системах, необхідно попередньо переконвертувати в CVSS. Для визначення загальної оцінки значення критичності вразливостей в системі CVSS можна умовно розділити на три групи, як показано в табл. 3.2

Таблиця 3.2

Відповідність значень CVSS системи умовним рівнем критичності загроз

CVSS значення	Умовний рівень критичності
1-3	Низький (Low)
3-7	Середній (Medium)
7-10	Високий (High)

Наведена у табл. 3.2 відповідність дозволяє оцінити відсоткове співвідношення кількості вразливостей високого, середнього і низького рівня критичності, які виявлені у досліджуваних програмних продуктах протягом усього часу їх існування. Кругові діаграми, що показують таке співвідношення для розглянутих Apache і IIS веб-серверів, представлені відповідно на рис. 3.8 та рис. 3.9.

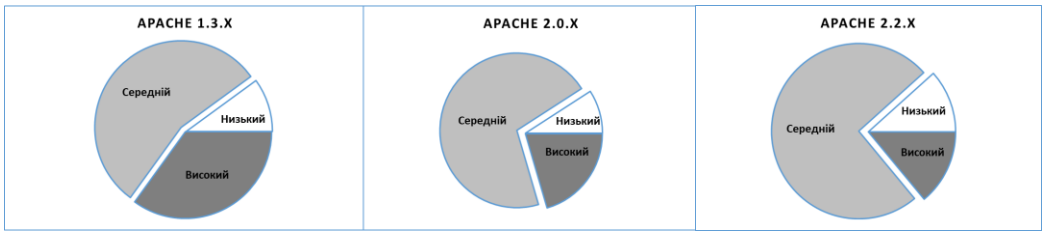

Рис. 3.8. Відношення вразливостей з різним рівнем критичності

Рис. 3.9. Відношення вразливостей з різним рівнем критичності для web-сервера IIS

Побудовані діаграми показують, що більшість вразливостей, виявлених е розглянутих продуктах, мають середній рейтинг критичності. Аналіз діаграм дозволяє простежити деякі закономірності, пов'язані з розрахованими вище показниками. Наприклад, для веб-сервера Apache найкритичніші уразливості були знайдені в продукті версії 1.3. Цей факт можна пояснити тим, що гілка 1.3 є першою з випущених гілок веб-сервера Apache, а також найдовше існуючою версією цього веб-сервера, яка має високе значення коефіцієнта τ, вже пройшла стадію інтенсивного розвитку, тому в даній версії було знайдено основну кількість наявних вразливостей, в тому числі і критичних, за загальною кількістю перевищує кількість вразливостей, знайдених в наступних, більш пізніх версіях даного веб-сервера.

Далі пропонується обчислити рейтинги серйозності вразливостей в розглянутих продуктах за окремі інтервали часу t_i. В якості довжини часового інтервалу в даному випадку зручно вибрати період в 1 місяць. Обчислення проводяться за наступною формулою:

$$S_m = \sum_{n=1}^{N} S_n \qquad (3.5)$$

де S_m – сумарний рейтинг серйозності загроз, обчислений за деякий інтервал часу t_i;

N – загальна кількість загроз, виявлена протягом цього інтервалу часу;

n – номер загрози (вразливості), що змінюється від 1 до N;

S_n – рейтинг критичності n-ої загрози (вразливості).

Далі необхідно провести аналіз часу відновлення web-серверів. Аналіз дозволить визначити середню кількість часу, який потрібен комерційної корпорації Microsoft і відкритого спільноті Apache для того, щоб виправити виявлені загрози.

Більшість відкритих джерел інформації про загрози не ведуть облік виправлень і не містять інформацію про дату усунення уразливості. Іноді така інформація існує, але у різних джерелах вона різна. Зазвичай процес виявлення уразливості такий: вразливість обговорюється в списках розсилки, і якщо вона підтверджена, публікується в якості попередження про небезпеку. Потім ця інформація поширюється по різних ресурсах вразливостей, і час появи в цих ресурсах може варіюватися складаючи різницю від декількох днів до навіть декількох років . Тому, як правило, для розрахунку показника часу відновлення необхідно провести додаткове дослідження щодо пошуку та аналізу інформації, а також доповнити базу даних, знайденою інформацією про дату виправлення.

Практичний досвід аналізу показує, що з розглянутих відкритих джерел в першу чергу інформація про вразливості та загрози безпеки найбільш часто потрапляє в бібліотеку CVE, на якій базується багато інших ресурсів.

Однак, перевагу завжди варто віддавати інформації, що розміщена на офіційних сайтах або в розсилках (якщо такі є) компаній-виробників програмних продуктів, оцінку чи порівняння яких необхідно провести. В даному випадку дата випуску виправлень за знайденими загрозами може бути використана з офіційних інформаційних видань Apache і IIS веб-серверів, а саме Apache Security Reports [15] і IIS Microsoft Security Bulletins [16]. Цю інформацію і слід помістити в базу даних для використання в подальших розрахунках.

Маючи інформацію про дати виправлення вразливостей, з’являється можливість виконати більш точну оцінку показників.

Порівняння різних версій веб-сервера Apache показує, що Apache 2.0 був завжди найменш стабільною версією з великою кількістю загроз у безпеці, в той час як найперша, витримана часом версія продукту Apache 1.3, демонструвала

кращі показники. IIS 5.0 показав себе найбільш стійким і безвідмовним веб-сервером протягом останніх років, однак, можна помітити, що він мав значну кількість відкритих загроз у процесі свого розвитку. Його наступники - веб-сервера IIS 6 і IIS 7 також демонструють дуже хороші показники в плані інформаційної безпеки.

Проведемо розрахунок показників безвідмовності розглянутих web-серверsв і порівняємо характеристики за цими показниками. Для оцінювання показників надійності будемо використовувати інтервальний метод, використовуючи припущення, що інтенсивність відмови $\lambda(t)$є постійною величиною $\lambda(t)=const$ на певному інтервалі часу $[t,t+\Delta t]$. Обчислення інтенсивності відмов на кожному i -му інтервалі часу проводиться за формулою

$$\lambda(i)=\frac{n_i}{t_i} \tag{3.6}$$

де n_i – кількість вразливостей, виявлених на i-му інтервалі часі;

t_i – довжина інтервалу часу.

Всі показники надійності також усереднюються для цього інтервалу часу.

При використанні даного підходу необхідно вибрати часовий інтервал, виходячи з якого буде проводитися розрахунок інтенсивності відмов. Найбільш просту наближену оцінку можна провести, вибравши максимальний часовий інтервал, тобто прийнявши значення інтенсивності постійним протягом всього життєвого циклу. Нижче представлена така оцінка для досліджуваних версій продуктів.

Для наближеної оцінки значення інтенсивності приймається константою протягом усього життєвого циклу і обчислюється виходячи з кількості вразливостей, знайдених за всі роки існування продукту. Середні показники інтенсивності відмов для досліджуваних програмних продуктів представлені у табл. 3.3., а середні напрацювання на відмову наведені у табл. 3.4.

Таблиця 3.3

Середня інтенсивність відмов веб-серверів Apache та IIS, обчислена за повний час спостереження

Програмний продукт	Середня інтенсивність відмов / міс
Apache 1 .3	0.4115
Apache 2.0	0.7295
Apache 2.2	0.6751
IIS 5.x	0.5535
IIS 6.0	0.1512
IIS 7.0	0.1261

Таблиця 3.4

Середні напрацювання на відмову веб-серверів Apache та IIS

Програмний продукт	Середнє напрацювання на відмову (міс.)
Apache 1 .3	2.43
Apache 2.0	1.37
Apache 2.2	1.48
IIS 5.x	1.8
IIS 6.0	6.61
IIS 7.0	7.93

При оцінюванні багатокомпонентних систем, побудованих з використанням COST компонентів, доцільно проводити оцінювання окремих компонентів на відповідність визначеним вимогам безпеки, зокрема захищеності, із застосуванням ядер безпеки. Оцінювання проводиться на основі даних про уразливість програмних компонентів, доступних в описаних вище джерелах і ресурсах. Нижче викладені етапи та принцип роботи ядра, розробленого з метою оцінювання програмних OTS компонентів.

1 На вхід ядра надходить набір вимог, на відповідність яким потрібно оцінити певну частину системи. Вимога може висуватися до всіх або окремо взятих компонентів. Приклад формулювання вимоги: «Кількісні показники надійності всіх OTS компонентів системи повинні бути: Імовірність безвідмовної роботи не менше 0.9 за 1000 годин; Критичність відкритих вразливостей не більше 4 за шкалою CVSS». Граничні значення кількісних показників по кожній з вимог, на відповідність яким необхідно провести перевірку, вибираються зі списку або вводяться експертом в стартовому діалозі ядра. Значення є опціями, при їх відсутності ядро продовжує функціонувати в режимі оцінювання "не виробляючи порівняння з кількісними показниками.

2 Ядро для оцінювання OTS компонентів будує формальний алгоритм, який потрібно використовувати для перевірки заданих вимог.

3 Якщо вимога є заданою для всіх OTS компонентів системи, програмна частина, що відповідає за роботу з даними про систему звертається до проектної документації, здійснює пошук за ключовими словами і автоматично будує список всіх OTS компонентів системи, що використовуються в системі. При бажанні експерт може доповнити або відкоригувати цей список.

4 Програмна частина, що забезпечує роботу з внутрішніми даними ядра (в даному випадку внутрішніми даними буде база даних, побудована на основі різних ресурсів і джерел вразливостей), витягує необхідні значення для кожного з OTS компонентів.

5 Програмний код, що виконує розрахунки характеристик отримує дані по компонентах і виконує розрахунок ймовірності безвідмовної роботи і інших заданих на вході вимог для кожної з них.

6 Результати розрахунків, виконаних на попередньому етапі при необхідності передаються на блок порівняння, який генерує порівняння їх із заданими на вході граничними значеннями (в прикладі це значення ймовірності 0.9 і критичності 4), після чого повертається результат порівняння.

7 Модуль візуалізації виводить результати розрахунків і показує відповідність або не відповідність їх кожній заданій вимозі. Модуль генерації документу повертає результат оцінки у формі звіту з безпеки.

Варто відмітити, що ядро допомагає експерту у виконанні операцій, які можна автоматизувати. Використання ядер безпеки допускає, але не вимагає усунення експерта з процесу оцінювання. На кожному етапі експерт може брати участь в оцінюванні вразливості веб-серверів комп’ютерних систем.

3.5. Висновки до розділу

1. У результаті застосування підходу UML визначено та побудовано рольову модель процесу виявлення та оцінювання вразливостей комп’ютерних систем, що дало змогу виявити функціональні вимоги до засобу автоматизації побудови моделі та імплементації методу виявлення та оцінювання загроз веб-серверів з врахуванням особливостей методології ядер безпеки.

2. Побудовано та представлено у вигляді шарів Фаулера архітектуру засобу автоматизації процесів виявлення та оцінювання вразливостей веб-серверів, що дало змогу спроектувати відповідні компоненти і залежності між ними з майбутньою імплементацією у програмному коді.

3. Запропоновано і проведено експерименти для кількісного оцінювання можливості ураження веб-серверів у комп’ютерних системах, наведено результати таких досліджень щодо захищеності від загроз веб-серверів Apache та IIS.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Усі дослідження методів і засобів виявлення та оцінювання вразливостей веб-серверів у розумних комп'ютерних системах здійснювалися з дотриманням правил та норм охорони праці і вимог техніки безпеки, що є невід'ємною частиною виконання всіх видів робіт при проведенні даного дослідження. Робоче приміщення та місце має відповідати вимогам щодо охорони праці при організації роботи з візуальними дисплейними терміналами електронно обчислювальних машин (ВДТ). У даному підрозділі розглядаються умови в приміщенні, де розроблялася кваліфікаційна робота.

Згідно ДБН В.2.5-28:2018 [22] приміщення у якому проводилися дослідження методів та засобів виявлення та оцінювання вразливостей веб-серверів у комп'ютерних системах відповідає санітарним нормам. Денне (природне) освітлення приміщення відбувається за системою однобічного бічного освітлення. Природне світло проникає у приміщення через три віконні отвори. Також наявні жалюзі з можливістю захисту працюючих від прямого попадання сонячних променів і регулювання рівня освітленості в приміщенні. Вікна приміщення орієнтовані на північний схід. Оскільки будинок розташований у відносній віддаленості від прилеглих будівель, то які-небудь перешкоди природному освітленню розглянутого приміщення відсутні.

Всередині приміщення стіни обклеєні світлими шпалерами, стеля побілена (переважає білий колір).

В приміщенні використовується система загального рівномірного штучного освітлення. У приміщенні маються внутрішні джерела постійного шуму:

- вентилятори блоків ЕОМ;
- принтери;
- дисководи.

Це відповідає нормам державних санітарних правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [23]. Шум, створюваний усіма перерахованими джерелами, можна кваліфікувати як постійний.

Фактичний виміряний рівень шуму в робочій зоні склав 43 дБА, що задовольняє нормативному рівню шуму (не повинний перевищувати 50 дБА), та не перевищує санітарних норм виробничого шуму, ультразвуку та інфразвуку [24].

Аналіз стану електробезпеки в робочому приміщенні де проводяться дослідження методів та засобів кваліфікаційної роботи показав що:

– усі прилади в кабінеті використовують напругу 220 В;

– електропроводка захована і ізольована від працівників спеціальним коробом;

– кожне робоче місце з ПЕОМ обладнане окремими розетками по 220 В;

– у приміщенні знаходяться наступні споживачі електроенергії: 4 ПЕОМ, 4 візуально дисплейні термінали та 8 світильників (по 4 лампи);

– відносна вологість повітря – 60%, температура повітря +22 - +24 °С, струмопровідний пил і хімічно активні речовини в повітрі відсутні;

– підлога: ізолююча – лінолеум.

Проаналізувавши наведене вище, можна сказати, що кабінет відноситься до приміщень без підвищеної електронебезпеки.

ПЕОМ, що використовуються в даному кабінеті підключаються до трифазної мережі і мають захисне занулення (за допомогою окремого захисного нульового провідника). Корпуси ВДТ та принтера виготовлені з пластику і не являються струмопровідними. Щодо корпусів самих ПЕОМ, вони виготовлені зі струмопровідного матеріалу, крім передньої панелі, що виготовлена з пластику.

До роботи не допускаються особи, які не пройшли навчання з техніки безпеки. Даний кабінет задовольняє вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями що, відображені в НПАОП 0.00-7.15-18 [25].

З огляду на можливість виникнення пожежі слід з’ясувати, які речовини і матеріали можуть горіти. У приміщенні, що розглядається, можуть горіти вироби з дерева, пластмас, тканини і паперу. Горючі рідини, пил та волокна у приміщенні не використовуються і не виділяються. Тому приміщення, що аналізується, відноситься, відповідно до нормативної документації, до зони П-ІІа і до категорії пожежної небезпеки В.

Експлуатація ліній електромережі практично повністю унеможливлює виникнення електричного джерела загоряння в наслідок короткого замикання та перевантаження проводів. Застосовуються дроти з важкогорючою і негорючою ізоляцією.

Приміщення має один вихід, оскільки в ньому працює менше 25 чоловік. Ширина проходу між робочими місцями у приміщенні перевищує 1 м. Будинок має три виходи – головний і 2 запасних. Коридор між приміщеннями має два виходи на різні сходи, одні з яких ведуть до головного виходу, а другі - до спеціального евакуаційного виходу.

Для гасіння пожежі кімната обладнана ручними вуглекислотними вогнегасниками ВВК-1,4. У загальному коридорі встановлені пінні вогнегасники ВВП. На сходах присутній спеціальний щит пожежного гідранта з відповідним рукавом. Розглянуте приміщення обладнане датчиками централізованої системи пожежної сигналізації.

У даному підрозділі було проаналізовано основні проблеми охорони праці, що можуть виникнути під час роботи працівника. Визначено основні вимоги до приміщення, де проводяться дослідження методів та засобів виявлення та оцінювання вразливостей веб-серверів при проектуванні комп’ютерних систем, мікроклімату в приміщенні, освітлення та основних ергономічних характеристик.

4.2. Захист населення у надзвичайних ситуаціях від впливу хімічних речовин

Значна кількість великих катастроф, що відбулися на території України за останній час (серед яких особливе місце займає Чорнобильська), змістила

пріоритети у призначенні Цивільної оборони від захисту населення в умовах воєнного часу на захист населення від наслідків надзвичайних ситуацій техногенного і природного характеру, від галузевого (відомчого) формування і функціонування на функціональні принципи формування і реагування на надзвичайні ситуації [27].

Прийняті Верховною Радою України закони: «Про Цивільну оборону України» (1999 р.), «Про захист людини від впливу іонізуючих випромінювань» (1998 р.), «Про захист населення і територій від надзвичайних ситуацій техногенного і природного характеру» (2000 р.) чітко визначили призначення і завдання Цивільної оборони України, відповідальність виконавчої влади всіх рівнів щодо захисту життя і здоров'я людини від наслідків надзвичайних ситуацій, державну важливість цієї проблеми [27].

Організація безпеки і захисту населення України, об'єктів економіки і національного надбання держави повинна розглядатися як невід'ємна частина державного будівництва, як найважливіша функція центральних органів виконавчої влади, місцевих державних адміністрацій і виконавчих органів влади.

Рівень національної безпеки не може бути достатнім, якщо у загальнодержавному масштабі не буде вирішене завдання захисту населення, об'єктів економіки і національного надбання від надзвичайних ситуацій техногенного і природного характеру.

Особливо важливим є інформування працівників організацій про способи і засоби захисту людей від шкідливого впливу хімічних речовин.

Широкий спектр застосування хімічних речовин у народному господарстві сприяє значному розширенню виробництва та застосуванню їх в промисловості. При цьому значно збільшується їх асортимент: одержують багато нових хімічних сполук, які становлять небезпеку для оточуючого середовища і людей [27].

Шкідливою речовиною є така речовина, яка при контакті з організмом людини, у випадку порушення вимог безпеки, може викликати виробничі травми, професійні захворювання або відхилення стану здоров'я від норми. Шкідливі хімічні речовини використовуються як сировина (хлор для виготовлення хлорного

вапна) чи допоміжний матеріал (бензол, який застосовується як розчинник). У деяких випадках вони є побічними продуктами, що створюються у технологічних процесах. За фізіологічним впливом на організм людини всі шкідливі речовини поділяються на такі групи: подразнюючі, що вражають шляхи дихання, очі, шкіру, слизові оболонки (аміак, кислоти, сірчасті сполуки тощо); задушливі, які викликають токсичний набряк легень (сірководень, вуглекислий газ, метан, інертні гази, азот і т. д.); наркотичні, що спричиняють наркотичний вплив і впливають на центральну нервову систему (ацетон, бензин, леткі вуглеводні тощо); соматичні (миш'як, ртуть, свинець й ін.); канцерогенні речовини, що впливають, як правило, на злоякісні новоутворення — пухлини (циклічні аміни, азбест, нікель, хром тощо).

За ступенем впливу на організм людини шкідливі речовини поділяються на чотири класи небезпеки: I – надзвичайно шкідливі, II – високошкідливі; III – помірно шкідливі, IV – малошкідливі. Чинними в Україні є значення гранично допустимих концентрацій шкідливих речових у повітрі робочої зони, що містяться в переліку «Предельно допустимые концентрации (ПДК) вредных веществ в воздухе рабочей зоны» № 4617-88, доповненнях № 1–7 до нього, а також ГДК та орієнтовані безпечні рівні впливу, затверджені Головним державним санітарним лікарем України після 1 січня 1997 р.

Найбільш поширеними і небезпечними речовинами, що використовуються у промисловості і побуті, є аміак і хлор. Аміак використовується у промислових побутових холодильниках на м'ясокомбінатах, молокозаводах, овочевих базах, тобто там, де є необхідність в охолодженій продукції. При малих концентраціях він діє на людину збуджуючи, при великих – може призвести до інвалідності. Найкращі методи захисту в даних випадках – це застосування ізолюючого протигазу, респіратора, захисного костюма типу Л-1, гумових чобіт, рукавичок.

Значно поширений промисловий продукт – хлор використовується для знезараження питної води, вибілювання тканин та як сировина для багатьох хімічних підприємств. У зв'язку з його використанням трапляється чимало випадків отруєння. У разі потрапляння хлору на шкіру виникають опіки. Запобігти враженню хлором можна за допомогою застосування індивідуальних засобів

захисту — протигазу, кисневого ізолюючого приладу, спеціального захисного костюма, гумових чобіт, рукавиць.

При виробництві або застосуванні хімічних речовин вони, потрапляючи у робочі приміщення чи безпосередньо на працівників, являють небезпеку для здоров'я та нормальної життєдіяльності організму.

Ступінь токсичності хімічних речовин та характер викликаних ними патологічних зрушень залежать від низки факторів: хімічної структури речовини (чим вища дисперсність, тим глибше і швидше вони проникають у дихальні шляхи); розчинності в організмі працівника (чим вища розчинність, тим більша токсичність хімічної речовини); концентрації у повітрі (чим вища концентрація хімічних речовин, тим швидше настає отруєння); тривалості дії хімічних речовин.

Умови зовнішнього середовища (наприклад, температура, вологість тощо) можуть посилювати чи послаблювати дію токсичних речовин. Так, висока вологість повітря посилює токсичні дії на організм соляної кислоти, фтористого водню.

Певний токсичний ефект хімічних речовин залежить від індивідуальних особливостей організму. Перенесені або існуючі хвороби, загальне ослаблення організму знижують його опорність дії хімічних речовин. У таких людей токсикація протікає довше й у важчій формі.

Виділяють гострі та хронічні отруєння. Гострі отруєння виникають у тих випадках, коли в організм надходить велика концентрація хімічних речовин (унаслідок аварії чи нещасного випадку). Хронічні отруєння виникають у результаті багаторазового проникнення незначних концентрацій хімічних речовин, які мають властивість накопичуватися в організмі (свинець, ртуть). У таких випадках симптоми початкових стадій отруєння виявляються найчастіше при проведенні періодичних медичних оглядів.

Дія хімічних речовин може бути місцевою та загальною. Місцева дія зумовлюється опосередковано впливом дратівних речовин на тканини організму. Наприклад, мінеральні кислоти (соляна, азотна) та луги подразнюють шкіру. Загальна дія виявляється після потрапляння хімічних речовин у кров, причому

деякі речовини діють на окремі органи, інші – викликають загальне отруєння організму.

Шляхи проникнення отруйних речовин в організм людини: через шкіру, органи дихання та шлунок . Ступінь ураження отруйними речовинами залежить від їх токсичності, вибірковості дії, тривалості, а також від їх фізико-хімічних властивостей.

За тривалістю дії шкідливі речовини ділять на три групи: летальні, що призводять до смерті (5% випадків) – термін дії до 10 діб; тимчасові, що призводять до нудоти, блювоти, набряку легенів, болю у грудях – термін дії від 2 до 5 діб; короткочасні – тривалість декілька годин (подразнення у носі, ротовій порожнині, головний біль, задуха, загальна слабкість).

Сильнодіючі отруйні речовини – це такі токсичні хімічні сполуки, котрі використовуються у народному господарстві, вилив або викид яких в довкілля може привести до зараження його з небезпечними концентраціями для здоров'я або життя людей . До об'єктів, котрі виробляють, використовують та зберігають СДОР, відносяться підприємства хімічної, нафтохімічної промисловості; підприємства, що мають холодильні установки, в яких у якості холодоагенту використовується аміак; водопровідні та очисні споруди, на яких застосовується хлор; залізничні станції та магістралі; склади і бази з запасами отрутохімікатів або інших речовин для дезінфекції та дератизації [27].

У мирний час спричинити викид СДОР в довкілля можуть виробничі аварії, стихійні лиха, пожежі. При цьому виникають зони хімічного зараження, площа яких може досягати кількох квадратних кілометрів. Адміністративно-територіальна одиниця більше 10% населення якої може опинитися в зоні можливого хімічного зараження сильнодіючими отруйними речовинами при аваріях на хімічно-небезпечних об’єктах, називається хімічно-небезпечною адміністративно-територіальною одиницею.

Якщо зону можливого хімічного зараження у результаті викиду СДОР потрапляє: в місті – квартал; у заміській зоні – селище або сільський населений

пункт, то це називають масовим ураженням [27]. За токсичним проявом СДОР в залежності від інтоксикації умовно поділяться на 6 груп:

– задушливої дії: (хлор, фосген, хлорид сірки, хлорпікрин);

– загально-отруйної дії: (ціанистий водень, оксид вуглецю);

– задушливої та загально-отруйної дії (азотна кислота, сірчаний ангідрид, фтористий водень);

– нейротропні отрути, які вражають клітини центральної нервової системи (сірководень);

– задушливої та нейротропної дії (аміак);

– метаболічні отрути, які порушують обмін речовин у клітинах (дихлоретан).

Отруйні речовини, в яких температура кипіння приблизно +20 °С, при розливі випаровуються і рухаються за напрямком вітру. Такі речовини в небезпечних концентраціях виявляються на далеких відстанях від місця аварії.

Захист працівників від несприятливого впливу хімічних речовин здійснюється за допомогою таких заходів:

– удосконалення і розробки нових технологічних процесів, які виключають використання шкідливих хімічних речовин;

– застосування безперервних технологічних процесів, автоматичного контролю за технологічним процесом;

– заміни шкідливих речовин менш шкідливими (заміни метилового спирту бутиловим, жовтого фосфору – червоним при виробництві сірників);

– установлення концентрації хімічних речовин у сумішах (кількість миш’яку в кислотах для травлення металу не повинна перевищувати 0,2 %);

– комплексної механізації та автоматизації процесів, що супроводжуються шкідливими виділеннями;

– дистанційного управління технологічними процесами;

– раціонального планування цехів і обладнання (ізоляції шкідливих речовин);

– удосконалення конструкції обладнання (герметизації тощо);

– влаштування місцевої вентиляції для відсмоктування шкідливих речовин безпосередньо від місця їх утворення;

– використання індивідуальних засобів (спецодягу, окулярів, шоломів, масок, протигазів та респіраторів, антисептичних паст і т. д.);

– контролю за станом повітряного середовища на робочих місцях;

– токсикологічної експертизи і гігієнічної стандартизації всіх хімічних речовин.

Провівши аналіз отруйних хімічних речовин та їх вплив на організм людини, можна зробити висновок про їх широке застосування у побуті та на виробництві. Тому для організації безпеки праці на підприємствах необхідно розробляти та впроваджувати інструкції щодо поводження з такими речовинами, інформувати про методи захисту від їх негативної дії та можливі наслідки.

ВИСНОВКИ

Основні наукові та практичні результати полягають в наступному.

1. Проведено аналіз функціональності та організації комп’ютерних систем, які використовують web-сервери, у результаті якого виявлено можливі компоненти ураження та способи проникнення «зловмисників» для нанесення шкідливого впливу на безпеку та надійність функціонування системи. Це дало змогу врахувати потенційно-слабкі місця веб-серверів та визначити атрибути вразливостей, рівень їх критичності, які необхідно врахувати в процесі оцінювання захищеності web-серверів.

2. Проаналізовано методи оцінювання загроз та вразливостей web-серверів, у результаті якого встановлено, що більшість методів передбачає експертне оцінювання загроз, а також застосування багатьох автоматизованих засобів їх виявлення. Це підкреслює необхідність та актуальність задач оцінювання захищеності web-серверів, які б давали можливість централізовано з одного координаційного центру використовувати як знання експертів з безпеки, так і повноту можливостей інструментальних засобів.

3. Удосконалено і доповнено модель виявлення та оцінювання вразливості веб-серверів у «розумних» комп’ютерних системах за допомогою ядер безпеки, що дало можливість забезпечити структурованість та автоматизацію розрахунку параметрів вразливостей веб-серверів, а також наростити значення достовірності і точності експертних оцінок.

4. Розроблено метод виявлення та оцінювання вразливостей веб-серверів у «розумних» комп’ютерних системах за рахунок побудови профілю вимог вразливості веб-серверів та його формального представлення у вигляді нотацій теорії множин, що в перспективі дало змогу автоматизувати процес формування ядер безпеки.

5. Запропоновані рішення щодо виявлення та оцінювання вразливостей надали можливість оцінити захищеність веб-серверів і сформували базис для розвитку методів та інструментів прогнозування і запобігання загрозам.

6. У результаті застосування підходу UML визначено та побудовано рольову модель процесу виявлення та оцінювання вразливостей комп’ютерних систем, що дало змогу виявити функціональні вимоги до засобу автоматизації побудови моделі та імплементації методу виявлення та оцінювання загроз веб-серверів з врахуванням особливостей методології ядер безпеки.

7. Побудовано та представлено у вигляді шарів Фаулера архітектуру засобу автоматизації процесів виявлення та оцінювання вразливостей веб-серверів, що дало змогу спроектувати програмні компоненти і відношення між ними з подальшою імплементацією у програмному коді.

8. Запропоновано і проведено процедури кількісного оцінювання щодо вразливості веб-серверів у комп’ютерних системах, наведено результати експериментальних досліджень щодо захищеності від загроз веб-серверів Apache та IIS.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрашов А.А. Таксономические модели профилирования требований информационно-управляющих систем критического применения. Радиоэлектронные и компьютерные системы. Киев. 2010. №7(48) С. 104-108.

2. ДСТУ 2844 -94. Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення. Чинний від 01.01.96. К. Держстандарт України. 1995. 15 с.

3. ДСТУ 2850 -94. Програмні засоби ЕОМ. Показники та методи оцінювання якості. – Чинний від 01.01.96. – К. Держстандарт України. 1994. 20 с.

4. ДСТУ 2853 -94. Програмні засоби ЕОМ. Підготовлення та проведення випробувань. – Чинний від 01.01.96. – К. Держстандарт України. 1994.17 с.

5. ДСТУ 3918-1999 (ISO/IEC 12207:1995) Інформаційні технології. Процеси життєвого циклу програмного забезпечення, К.: Держстандарт України. 2000. 49 с.

6. Лавріщева К.М. Програмна інженерія. К. 2008. 312 с.

7. Лаврищева Е.М. Методы программирования. Теория, инженерия, практика. К.: Наук. Думка. 2006. 451 с.

8. Маторин С. Анализ и моделирование бизнес-систем: системологическая объектно-ориентированная технология. Харьков: ХНУРЭ. 2002. 322 с.

9. Тарасюк О.М. Методы и инструментальные средства метрико-вероятностной оценки качества программного обеспечения информационно-управляющих систем критического назначения: дис. канд. техн. наук. Харьков. 2004. 201 с.

10. Харченко В.С. Интеллектуальная система поддержки сертификации программного обеспечения систем критического применения. Труды Международной конференции «Автоматика-2002». Донецк. 2002. С. 46-48.

11. Яцишин В.В., Харитон Б.В. Схема реляційної бази даних для зберігання та опрацювання вразливостей веб-серверів у розумних комп’ютерних системах. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С.96.

12. Яцишин В.В., Харитон Б.В. Архітектура системи підтримки процесів виявлення та оцінювання вразливостей веб-серверів у комп'ютерних системах. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С.101.

13. Functional safety of electrical, electronic and programmable electronic safety related systems/ International Electrotechnical Commission, IEC 61508. Parts 1 to 7 – Geneva, Switzerland – 1998 – 2000.

14. Safety management requirements for defence systems (part 1 and 2). Defence Standard 00-56. – Ministry of Defence. Directorate of Standartization. Glasgow, UK. 2007.

15. Capability Maturity Model / M. Paulk, B. Curtis, M. Chrissis, Ch. Weber – IEEE Software. 1993. Vol. 10. 4. P. 18–27.

16. A Detailed Guide On Web Server Security – How To Secure Web Server? URL: https://cydomedia.com/a-detailed-guide-on-web-server-security-how-to-secure-web-server/ (дата звернення 10.12.2022 р.).

17. How to set up a secure web server. URL: https://blog.avast.com/create-a-secure-web-server-avast (дата звернення 05.12.2022 р.).

18. Web Server Security. URL: https://www.techopedia.com/definition/15712/web-server-security (дата звернення 02.12.2022 р.)

19. Жидецький В. Охорона праці користувачів. Львів: Афіша, 2000. 176 с.

20. Катренко Л.А., Катренко А.В. Охорона праці в галузі комп'ютинґу. Львів: Магнолія-2006. 2012. 544 с.

21. Желібо Є. Безпека життєдіяльності. К.: 2001. 483 с.

Додаток А

Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

X НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

7–8 грудня 2022 року

ТЕРНОПІЛЬ
2022

УДК 004.4
В. Яцишин, Б. Харитон
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

АРХІТЕКТУРА СИСТЕМИ ПІДТРИМКИ ПРОЦЕСІВ ВИЯВЛЕННЯ ТА ОЦІНЮВАННЯ ВРАЗЛИВОСТЕЙ ВЕБ-СЕРВЕРІВ У КОМП'ЮТЕРНИХ СИСТЕМАХ

UDC 004.4
V. Yatsyshyn, B. Kharyton

ARCHITECTURE OF THE SUPPORT SYSTEM FOR THE DETECTION AND ASSESSMENT OF WEB SERVER VULNERABILITY PROCESSES IN COMPUTER SYSTEMS

У загальному випадку архітектуру системи підтримки процесу виявлення та оцінювання вразливості web-серверів можна представити за шарами Фаулера, як показано на рис. 1.

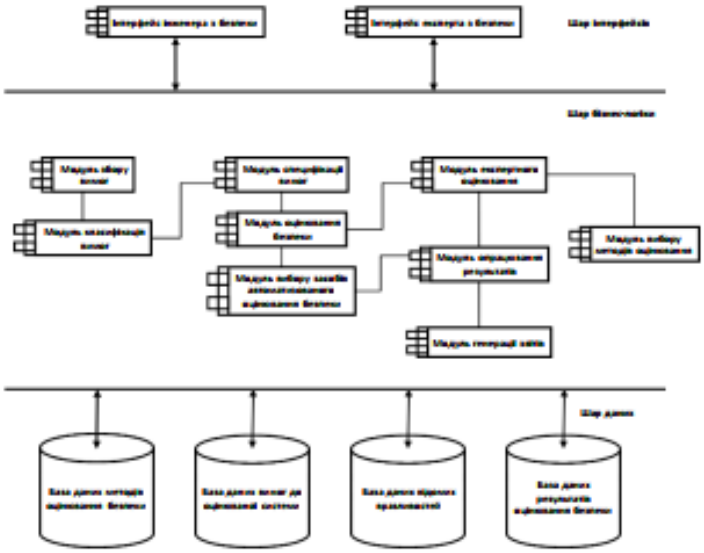

Рисунок 1. Архітектура системи підтримки процесу оцінювання безпеки web-серверів

На рівні шару користувацьких інтерфейсів визначено два інтерфейси – інтерфейс інженера з безпеки та інтерфейс експерта. У шарі бізнес-логіки передбачено модуль для збору вимог до програмного забезпечення. Модуль класифікації вимог може бути реалізований як інтелектуальний модуль з реалізацією можливості автоматичної класифікації на функціональні та нефункціональні вимоги, а після цього класифікації нефункціональних вимог за критеріями безпеки web-серверів. Модуль оцінювання безпеки дає змогу проводити оцінювання безпеки web-серверів на основі вимог безпеки із застосуванням як автоматизованих засобів визначення мір відповідних критеріїв, так і експертним шляхом. База даних методів оцінювання безпеки містить назви методів оцінювання безпеки, статичні параметри для ефективної їх роботи.

УДК 004.4
В. Яцишин, Б. Харитон
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

СХЕМА РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ ДЛЯ ЗБЕРІГАННЯ ТА ОПРАЦЮВАННЯ ВРАЗЛИВОСТЕЙ ВЕБ-СЕРВЕРІВ У РОЗУМНИХ КОМП'ЮТЕРНИХ СИСТЕМАХ

UDC 004.4
V. Yatsyshyn, B. Kharyton

A RELATIONAL DATABASE SCHEME FOR STORING AND PROCESSING WEB SERVER VULNERABILITIES IN SMART COMPUTER SYSTEMS

Проведення кількісного оцінювання показників надійності та функціональної безпеки програмних компонентів і систем передбачає роботу з інформацією. З метою систематизації, зберігання і подальшої обробки інформації можна використовувати реляційні бази даних. Ключовим моментом є попереднє коректне проектування бази даних: вона повинна включати необхідні відношення, збережені процедури, атрибутам, у базі даних повинні бути проставлені індекси і задані зв'язки між реляційними відношеннями.

Організація бази даних, структура її відношень і атрибутів повинні бути зручними для подальшої роботи з інформацією, автоматичної обробки даних та оцінювання різних характеристик. При оцінюванні OTS компонентів доцільним є побудова бази даних з урахуванням аналізу семантичної інформації з відкритих джерел даних про уразливість. На рис. 1 показана модель бази даних у вигляді діаграми «сутність-зв'язок», що відображає логічне представлення і взаємозв'язок основних відношень.

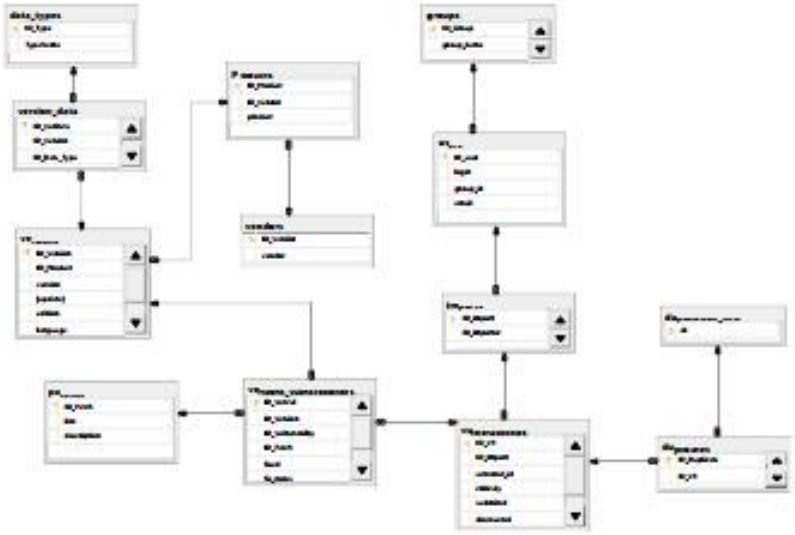

Рисунок 1. Схема бази даних для зберігання та керування загрозами

Ключовим є відношення «вразливості» («vulnerabilities»), яке включає в себе інформацію про уразливість, таку як ідентифікатор уразливості в розробленій базі даних, ідентифікатор уразливості в CVE або інших ресурсах (якщо такий є), опис уразливості на природній мові, рейтинг критичності, дату виявлення і дату появи інформації про уразливість у відкритому доступі, а також ідентифікатор користувача, який повідомив про знайдену уразливість, в також ідентифікатор імпорту, під час якого цю уразливість внесли в базу даних.