Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню магістра

студента *Мацайтіса Дмитра Ігоровича*

академічної групи *125м-19-2*

напряму підготовки *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Методи активного аудиту безпеки web-ресурсів комерційного підприємства*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	д.ф.-м.н., проф. Кагадій Т.С.			
розділів:				
спеціальний	ст. викл. Тимофєєв Д.С.			
економічний	к.е.н., доц. Пілова Д.П.			
Рецензент				
Нормоконтролер	ст. викл. Тимофєєв Д.С.			

Дніпро
2020

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістра

студенту *Мацайтісу Дмитру Ігоровичу* **академічної групи** *125м-19-2*
(прізвище ім'я по-батькові) (шифр)

напряму підготовки *125 Кібербезпека*
(код і назва спеціальності)

на тему *Методи активного аудиту безпеки web-ресурсів комерційного підприємства*

затверджену наказом ректора НТУ «Дніпровська політехніка» від _____ № ____

Розділ	Зміст	Термін виконання
Розділ 1	Розглянути статистичні дані та провести аналіз щодо актуального стану безпеки веб – ресурсів. Проаналізувати вразливості веб – ресурсів та існуючі методики тестування на проникнення. Розглянути додаткові стандарти та методології.	10.10.2020
Розділ 2	Виконати аналіз вразливостей веб – ресурсів. Дослідити обрані вразливості. Розробити алгоритм тестування веб – ресурсів на проникнення на основі аналізу вразливостей да існуючих методів тестування.	20.11.2020
Розділ 3	Провести економічне обґрунтування доцільності елементів активного аудиту веб - ресурсів та проведено розрахунок витрат на впровадження запропонованих рішень щодо поліпшення активного аудиту веб – ресурсів МКП.	05.12.2020

Завдання видано _____ Кагадій Т.С.
(підпис керівника) (прізвище, ініціали)

Дата видачі: 02.09.2020р.

Дата подання до екзаменаційної комісії: 12.12.2020р.

Прийнято до виконання _____ Мацайтіс Д.І.
(підпис студента) (прізвище, ініціали

РЕФЕРАТ

Пояснювальна записка: ___ с., __ рис., __ табл., __ додатки, __ джерел.

Об’єкт досліджень: процес аудиту безпеки веб – ресурсів.

Метою кваліфікаційної роботи є: покращення якості та ефективності аудиту безпеки веб – ресурсів комерційного підприємства (МКП).

В першому кваліфікаційної роботи проведено аналіз щодо актуального стану безпеки веб – ресурсів. Проведено аналіз вразливостей веб – ресурсів та проаналізовано існуючі методики тестування на проникнення.

В другому розділі кваліфікаційної роботи приведено загальні дані щодо інструмента аналізу тестування на проникнення. Проведено категоріювання вразливостей та їх аналіз з запропонованими методами протидії. Розроблено алгоритм обрання метода тестування при тестуванні веб – ресурсу на проникнення.

У третьому розділі кваліфікаційної роботи розраховано економічну доцільність впровадження та використання розробленого алгоритму обрання оптимальних рішень під час тестування на проникнення типового МКП.

Актуальність роботи пов’язана зі значним зростанням використання вразливостей зловмисниками в галузі моніторингу веб – ресурсів та проблемах їх аудиту із за всесвітньої пандемії (COVID – 19).

ВЕБ – РЕСУРСИ, КОМЕРЦІЙНЕ ПІДПРИЄМСТВО, АНАЛІЗ ВРАЗЛИВОСТЕЙ, АНАЛІЗ СТАТИСТИЧНИХ ДАНИХ, ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

РЕФЕРАТ

Пояснительная записка: _страниц, _рисунка., _таблицы, _приложений, _источников.

Объект исследований: процесс аудита безопасности веб – ресурсов.

Целью работы является: улучшение качества и эффективности аудита безопасности веб - ресурсов коммерческого предприятия (МКП).

В первом разделе квалификационной работы проведен анализ относительно актуального состояния безопасности веб - ресурсов. Проведен анализ уязвимостей веб - ресурсов и проанализированы существующие методики тестирования на проникновение.

Во втором разделе квалификационной работы приведены общие данные по инструменту анализа тестирования на проникновение. Проведено категорирование уязвимостей и их анализ с предложенными методами противодействия. Разработан алгоритм избрания метода тестирования при тестировании веб - ресурса на проникновение.

В третьем разделе рассчитано экономическую целесообразность внедрения и использования разработанного алгоритма избрания оптимальных решений при тестировании на проникновение типичного МКП.

Актуальность работы связана со значительным ростом кибер - криминала в области мониторинга веб - ресурсов и проблемах их аудита из-за всемирной пандемии (COVID - 19) а также постоянно эволюционирующих уязвимостей.

ВЭБ - РЕСУРСЫ, МКП, АНАЛИЗ УЯЗВИМОСТЕЙ, АНАЛИЗ СТАТИСТИЧЕСКИХ ДАННЫХ, ТЕСТИРОВАНИЕ НА ПРОНИКНОВЕНИЕ, МЕТОДЫ ТЕСТИРОВАНИЯ, МЕТОДЫ ПРОТИВОДЕЙСТВИЯ, АЛГОРИТМ МЕТОДА ТЕСТИРОВАНИЯ РЕСУРСА

ABSTRACT

Explanatory note: _pages, _images, _tables, _supplements, _sources.

Object of research: the process of security audit of web resources.

The aim of the work is: to improve the quality and efficiency of the security audit of web resources of a commercial enterprise (PCE).

The first chapter of the thesis analyzes the current state of security of web resources in the world and in Ukraine. The analysis of vulnerabilities of web resources is carried out and the existing methods of penetration testing are analyzed.

The second section of the thesis provides general data on the penetration testing analysis tool. The categorization of vulnerabilities and their analysis with the proposed methods of counteraction have been carried out. An algorithm for choosing a testing method when testing a web resource for penetration has been developed.

In the third section, the economic feasibility of introducing and using the developed algorithm for selecting optimal solutions for penetration testing of a typical MCP is calculated.

The relevance of the work is associated with a significant increase in cybercrime in the monitoring of web resources and the problems of their audit from the global pandemic (COVID - 19) and also constantly evolving vulnerabilities.

WEB - RESOURCES, SCE, VULNERABILITY ANALYSIS, STATISTICAL DATA ANALYSIS, PENETRATION TESTING, TESTING METHODS, METHODS OF COUNTERACTION, ALGORITHM OF THE RESOURCE TESTING METHOD

СПИСОК УМОВНИХ СКОРОЧЕНЬ

WAF – Web application firewall - Мережевий екран веб – ресурсу;

API - Application Programming Interface – Прикладний програмний інтерфейс;

HTML - HyperText Markup Language – Мова розмітки гіпертексту;

SOAR – Security Orchestration, Automation and Response - Система управління, автоматизації та реагування;

ОС - Operating System – Операційна система;

SQL - Structured Query Language – Мова структурних запитів;

T-SQL – Transact-SQL - Процедурне розширення мови SQL;

ASP – Active Server Pages - Активні серверні сторінки;

XML – Extensible Markup Language – Стандарт побудови мов розмітки;

SSL – Secure Sockets Layer - Криптографічний протокол безпечного з'єднання;

FTP – File Transfer Protocol - Протокол передачі файлів;

SFTP – SSH File Transfer Protocol - Протокол передачі файлів по захищеному каналу;

SSH – Secure Shell - Шифрований протокол віддаленого керування комп'ютером;

TLS – transport layer security - Криптографічний протокол передачі даних;

PFS – Perfect forward secrecy - Протокол який зберігає сесійні ключі;

PHP – Hypertext Preprocessor - Скриптована мова програмування;

SOAP – Simple Object Access Protocol - Протокол доступу до об'єктів;

CSP – Content Security Policy - Політика безпеки контенту;

MFA – Multi-factor authentication - Багатофакторна автентифікація;

МКП – Мале комерційне підприємство;

APT - Advanced Persistent Threat – Загроза підвищеної складності;

BAS - Breach and Attack Simulation – класифікація продуктів, симулюючих порушення та реалізації вразливостей;

Зміст

ВСТУП

У наш час безпека веб – ресурсів є невід'ємною частиною, та одним із найважливіших напрямків інформаційної безпеки. Експоненційне збільшення кількості веб – ресурсів водночас приводять до зростання кількості конфіденційної інформації, яка знаходиться на серверах віддаленого доступу (remote access) хостингу або провайдера (особливо веб – ресурсів, пов'язаних з хмарними технологіями).

Зі зростанням кількості веб – ресурсів також зростає попит на них зі сторони кіберзлочинців, тому використання вразливостей є найбільш актуальною проблемою аудиту безпеки веб - ресурсів. Використання вразливостей веб – ресурсів малих комерційних підприємств та загалом підприємств має не тільки інтерес з боку хакерів, а й можливий політичний мотив у гібридних війнах та ін.

Отже загальне удосконалення превентивних методів та засобів захисту веб – ресурсів від атак залишається актуальною проблематикою веб захисту загалом та частково інформаційної безпеки.

1 СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Поточний стан проблеми безпеки та вразливостей веб – ресурсів малого комерційного підприємства

Більшість організацій усвідомлює роль активного аудиту безпеки веб ресурсів та їх важливість в еру IT. Веб – ресурси та інформаційна безпека тісно пов'язані між собою і працюють в середовищі з постійно зростаючим рівнем небезпеки, що характеризується безперервним збільшенням кількості атак і постійного посилення щодо запобіжних заходів. Найчастіше організації нездатні передбачити небезпеку, тому і існують компанії, які надають послуги аудиту безпеки (в тому числі – веб ресурсів) за допомогою сертифікованих аудиторів – пентестерів, які проводять тестування на проникнення та використовують вразливості для локалізації та нейтралізації різноманітних проблем в системі безпеки. Мале комерційне підприємство (далі МКП) завжди залишатиметься мішенню для кібер – шахраїв та мисливців легкої наживи за допомогою існуючих вразливостей веб ресурсів, тому що у порівнянні з великими компаніями – не мають в своєму розпорядженні великі ресурси для постійного моніторингу та аудиту безпеки свого підприємства.

МКП це підприємство, яке не перевищує показники, визначені в “Господарському кодексу України про кількість робітників”. [1] В іншому випадку їх зараховують до великих підприємств. Класифікація здійснюється здебільшого незалежно від правової форми підприємства.

Поділ на групи відбувається за критеріями, які в залежності від показників підприємства:

- кількості працівників
- річного балансу

До МКП не відносять підприємства, в яких всі власники разом володіють менш ніж як 25% всієї вартості. Кількість працівників - $\leq$ 50 осіб. Річний дохід - $\leq$ 10 млн евро.

Українське законодавство визначає в розділі в п. 3 ст. 55 Господарського кодексу України - суб'єктами малого підприємництва є:

Фізичні особи, зареєстровані в установленому законом порядку як фізичні особи - підприємці, в яких середня кількість працівників за звітний період (календарний рік) не перевищує 50 осіб та річний дохід не перевищує суму, еквівалентну 10 мільйонам євро, визначених за середньорічнім курсом національного банку України;

Юридичні особи - суб'єкти господарювання будь-якої організаційно-правової форми та форми власності, в яких середня кількість працівників за звітний період (календарний рік) не перевищує 50 осіб та річний дохід від будь-якої діяльності не перевищує суму, еквівалентну 10 мільйонам євро, визначення за середньорічнім курсом національного банку України.

Розглянемо статистику та її ранжування серед зарубіжних експертних компаній та зрозуміємо з чим все таки мають справи експертні агентства.

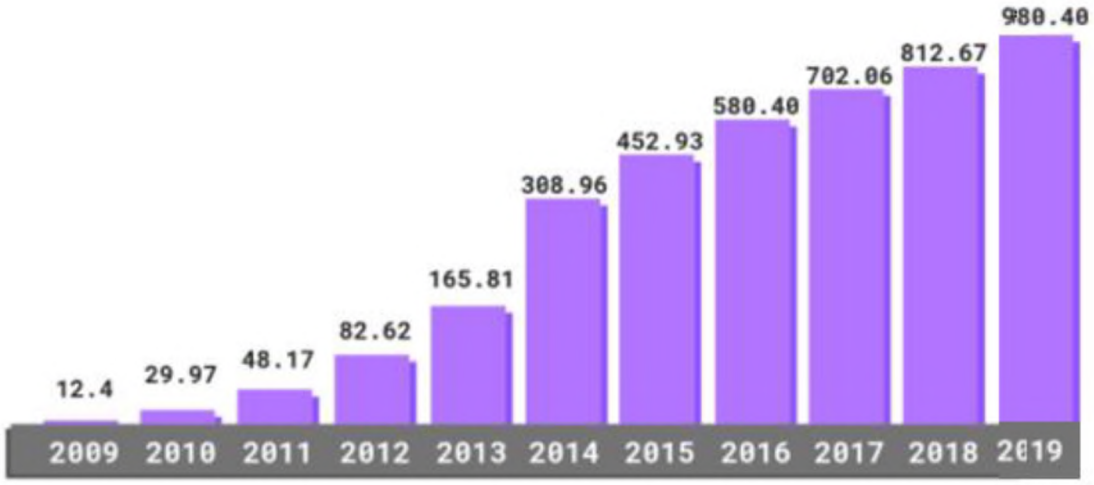

Рисунок 1.1 Статистика зростання кількості виявленого шкідливого програмного забезпечення з кожним роком (в мільйонах)

С початку всесвітньої пандемії (COVID-19) рівень кіберзлочинів виріс на 600%, а інфікування зараженим програмним забезпеченням з кожним роком постійно збільшується.

Ознайомившись з інформацією Purplesec, Owasp Top 10 та аналітики CAPEC [2], [3], [4] маємо змогу побачити такі дані:

- 92% Усіх атак на інтернет ресурси та користувачів здійснюється за допомогою електронної пошти;
- У 2020 році кількість нових варіантів шкідливих програм для мобільних пристроїв збільшилось на 54%;
- Онлайн сервіси з безкоштовного завантажування від третіх осіб у 30% випадках містять експлоіти та кей – логгери;
- Android ОС та ОС створені на її базі залишаються найбільш бажаним шматом для хакерів – більш ниж 250 тисяч юзерів були атаковані Trojan – Baner.AndroidOS.Asacub та на даний момент 98% усіх шкідливих програм випадають на долю Android ОС;
- За останній рік кількість шкідливих програм для MacOS та кількість спроб проникнення збільшилось на 165%;
- Кожного для створюється більш ніж 230 000 нових прототипів шкідливих програм які беруть за основу не закриті експлоіти та комбінують їх з методами проникнення;
- Більш ніж 18 мільйонів веб – ресурсів піддаються атакам кожного дня. 34% підприємств, потерпілих від атак потребують неділю чи більше для повного або часткового відновлення.

1.2 Аналіз типових вразливостей веб – ресурсів

Проаналізуємо статистичні данні класифікацій видів атак на веб ресурси, їх розповсюдженість та причинні наслідки.

В таблиці 1.2 маємо змогу побачити існуючі та найбільш використовувані зловмисниками типи вразливостей та кількість їх реалізації у відсотках (згідно з аналізом OWASP TOP 10). [5]

Таблиця 1.2 – Аналіз вразливостей інтернет – ресурсів

1	Ін'єкції – SQL, NoSQL, OS, etc	70%
2	Влазлива автентифікація;	56%
3	Порушення конфіденційності даних	47%
4	Ін'єкція зовнішніх сутностей XML;	29%
5	Порушення контролю доступу	26%
6	Помилки в конфігураціях безпеки;	24%
7	Міжсайтовий скриптинг XSS	16%
8	Небезпечна десеріалізація	15%
9	Використання компонентів із відомими вразливостями	10%
10	Недостатній моніторинг ресурсу	8%

1. Однією з найпопулярніших вразливостей є ін'єкції – SQL. Це перше, що аудитор повинен перевірити.

SQL Injections (ін'єкція - впровадження коду) — один з поширених способів злому сайтів та програм, що працюють з базами даних, заснований на впровадженні в запит довільного SQL-коду. Атаки SQL-ін'єкцією дозволяють зловмисникам підробляти ідентифікаційні та існуючі дані, викликати проблеми з відмовою, такі як анулювання транзакцій або зміна балансу, вирішувати повне

розкриття всіх даних в системі, знищувати дані або робити їх недоступними з інших причин за допомогою ролі адміністратора сервера баз даних.

2. Вразлива автентифікація. Багато веб-додатків не захищають конфіденційні дані, такі як кредитні карти і облікові дані для автентифікації (передача персональних даних і даних кредитних карт по протоколу HTTP замість HTTPS; відсутність шифрування критичних даних, таких як паролі або номери кредитних карт).

Для ідентифікації веб-додаток використовує так звані сесійні кукі. Після того, як користувач ввів логін та пароль та додаток його авторизував, у браузері зберігається спеціальний ідентифікатор, який у подальшому попереджає сервіс при кожному запрошенні вами веб сторінки.

У випадку, якщо ваш ідентифікатор отримав зловмисник - а в системі не були реалізовані перевірки сесії IP-адрес або перевірки наявності більш одного з'єднання в одній сесії, зловмисник може отримати доступ до системи з правами вашого облікового запису. Якщо це інтернет-банк або кабінет платіжної системи – наслідки можуть бути катастрофічними.

3. Порушення конфіденційності даних, як вразливість реалізується у 47% випадків.

Багато веб-додатків не захищають конфіденційні дані, такі як кредитні картки та дані для автентифікації. Зловмисники можуть викрасти або модифікувати такі слабо захищені дані для використання у своїх цілях.

Самий простий приклад - передача даних за протоколом HTTP. Дані, які передаються за протоколом HTTP - не шифруються, а при проходженні таких даних від комп'ютера користувача до веб-сервера - пройдуть достатньо багато різних вузлів зв’язку: маршрутизатор (домашній або робочий), маршрутизатор провайдера, маршрутизатор на каналі, маршрутизатор в даних- центр хостинг-провайдера сервера і так далі. На кожному з цих вузлів може знаходитися зловмисник та його програмне забезпечення для сканування даних - сніфер - програма, яка зчищає весь трафік і передає зловмиснику. Останній отримує дані та обшукує їх на предмет персональних даних та даних кредитних карт.

Такі дані повинні передаватися виключно за протоколом HTTPS. Ще одна задача SSL-сертифіката (а саме так називається спеціальний ключ, за допомогою якого здійснюється перевірка підключеності та шифрування в HTTPS) - підтверджується, що він видає саме для свого веб-сайту. У випадку, якщо сертифікат прострочений або підроблений – браузер попередить вас про вразливе підключення

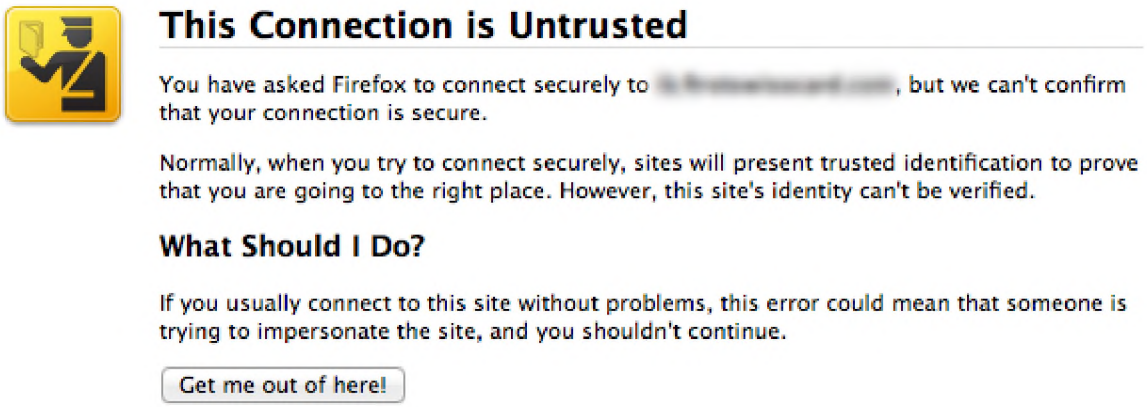

Рисунок – 1.2 Повідомлення про відсутність SSL сертифікату на сайті (повідомлення від браузера)

4. Впровадження зовнішніх сутностей XML – вид вразливості - ін'єкція, заснований на впровадженні в XML-запит до сервера атрибутів і сутностей, що дозволяють отримати неавторизований доступ до даних. Наприклад, коли в XML запит вказують зовнішній файл, який знаходиться на сервері.

5. Порушення контролю доступу – вразливість в методах авторизації, яка дозволяє порушнику отримати підвищені привілеї у додатку. Один конкретний тип проблеми контролю доступу - це адміністративні інтерфейси, які дозволяють адміністраторам сайтів керувати сайтом через інтернет. Такі функції часто використовуються, щоб дозволити адміністраторам сайтів ефективно керувати користувачами, даними та вмістом на своєму сайті (так звані адмін – панелі).

6. Помилки в конфігураціях безпеки - некоректне налаштування параметрів безпеки. Безпека веб-додатку вимагає наявності безпечної

конфігурації всіх компонентів інфраструктури: компонентів програми (таких як програмний каркас, веб-сервер, сервера баз даних і самої платформи). Програмне забезпечення повинно бути в актуальному стані: вразливості знаходять кожен день в самих різних програмних компонентах - операційній системі, веб-серверах, серверах баз даних, поштових серверах і т.д.

7. Міжсайтовий скриптинг XSS - міжсайтове виконання сценаріїв - помилка валідації призначених для користувача даних, яка дозволяє передати JavaScript (далі JS – мова програмування) код на виконання в браузер користувача. Через JS можна змінювати дані, розташовані на сторінці, наприклад, там можуть бути реквізити для банківського переказу. Атаки такого роду часто також називають HTML - ін'єкціями, адже механізм їх впровадження дуже схожий з SQL-ін'єкціями, але на відміну від останніх, впроваджуваний код виповнюється в браузері користувача.

8. Небезпечна десериалізація - це вразливість, яка виникає, коли ненадійні дані використовуються для зловживання логікою додатка, для атаки типу «відмова в обслуговуванні» (DoS - Denial of Service) або навіть для виконання довільного коду після його десеріалізації. Серіалізація відноситься до процесу перетворення об'єкта в формат, який можна зберегти на диску (наприклад, зберегти в файл або сховище даних), відправити через потоки (наприклад, stdout - стандартний вивід) за умовою "пов'язаний" з монітором - все, що ви напишете в нього, ви введете на екран.) або відправити по мережі.

Десеріалізація перетворює серіалізовані дані, що надходять з файлу, потоку або мережевого сокета, в об'єкт. Веб-додатки використовують серіалізацію і десеріалізацію на регулярній основі, і більшість мов програмування навіть надають вбудовані функції для серіалізації даних.

9. Використання компонентів з відомими уразливими - Необхідна увага на web-додатки, які написані з використанням спеціальних бібліотек або «програмних каркасів» (англ - framework) і поставляються сторонніми компаніями. Їх компоненти мають відкритий вихідний код, а це означає, що вони також використовуються мільйонами користувачів у всьому світі і

можуть бути вразливі. Звичайно вразливості шукають (і знаходять) в більш низькорівневих компонентах системи, таких як сервер бази даних, web-сервер і компоненти операційної системи аж до її ядра. Тому важливо використовувати останні версії компонентів і стежити за відомими уразливими, які з'являються на сайтах типу securityfocus.com.

10. Недостатній облік та моніторинг - відсутність журналів та моніторингу). Недостатня реєстрація та моніторинг комп'ютерних систем, додатків і мереж надають безліч шлюзів для проб і порушень, які важко або неможливо ідентифікувати і усунути без контрольного журналу. Типова лог-архітектура генерує журнали безпеки, аналізує, зберігає і контролює їх. Це важливо не тільки для боротьби з загрозами, що виникають через недостатню реєстрації та моніторингу, а й для відповідності нормативним вимогам.

При розробці часто виникає потреба перевірки валідності даних для деякого алгоритму. Формально це можна описати таким чином: нехай ми отримуємо деяку структуру даних, перевіряємо її значення на відповідність деякій області допустимих значень (ОДЗ) і передаємо її далі. Згодом ця ж структура даних може бути піддана такій же перевірці. У разі незмінності структури, повторна перевірка її валідності - зайва дія.

Хоча валідація може дійсно бути довгою, проблема тут не тільки в продуктивності. Набагато неприємніше - зайва відповідальність. У розробника немає впевненості чи потрібно перевіряти структуру на валідність ще раз. Крім зайвої перевірки, можна навпаки допустити відсутність будь-якої перевірки, невірно припускаючи, що структура була перевірена раніше. Таким чином, допускаються несправності в методах, які очікують перевірену структуру і працюють некоректно зі структурою, чиє значення виходить за деяку область допустимих.

Тому теоретично, зловмисник може отримати доступ до будь – якого веб ресурсу. Потрібно лише знайти “вірний” важіль, та достатня кількість часу.

Згідно аналізу статистики OWASP – 82% вразливостей знаходять в коді за допомогою його тестування.

Оцінка безпеки за допомогою WhiteBox тестування проводиться одночасно кількома фахівцями для максимального покриття і виявлення якомога більшої кількості вразливостей. Ця робота також включає в себе ручний і автоматизований аналіз коду. Автоматичне виявлення допомагає прискорити тестування, але вимагає ручної верифікації для виключення помилкових спрацьовувань. У той час як ручні методи займають більше часу, виявлені вразливості будуть реальними.

Рисунок 1.3 – Співвідношення кількості знайдених вразливостей за допомогою WhiteBox тестування (у коді).

1.3 Дослідження методик тестування на проникнення

Тест на проникнення, також відомий як pentesting, являє собою імітацію кібер-атаки на комп'ютерну систему типового підприємства (або імітацію атаки на існуючий ресурс компанії) для перевірки на наявність вразливостей, які можуть бути використані для експлуатації. У контексті безпеки веб-додатків тест на проникнення зазвичай використовується для доповнення брандмауера веб-додатків (WAF).

Тестування на проникнення може включати в себе спробу злому будь-якої кількості прикладних систем (наприклад, інтерфейсів прикладних протоколів (API), зовнішніх / внутрішніх серверів) для виявлення вразливостей.

Ін'єкції, надані за допомогою тесту на проникнення, можуть бути використані для тонкої настройки політик безпеки WAF.

Організації всіх типів і розмірів змушені мати справу з кібер-атаками, які з кожним роком стають все складніше і стійкіше до засобів протидії. Основними інструментами, використовуваними для перевірки захищеності систем і даних, як і раніше є сканери вразливостей і тести на проникнення (пентести). Їх результати не тільки використовуються для оцінки ризиків, а й є обов'язковою умовою для дотримання різних нормативних вимог. Сканування на наявність вразливостей і пентести корисні для оцінки захищеності організації в певний момент часу. Однак вони не дозволяють скласти цілісну картину стану безпеки, особливо коли справа стосується комплексних атак з багатьох векторах. Найефективнішим засобом тестування стресостійкості організації до наростання хвилі кібер-злочинності є моделювання цілеспрямованих атак по безлічі векторів, що отримало назву «імітація зломів і атак» (Breach & Attack Simulations, BAS). Тестування захисту є настільки складним завданням для технічних фахівців, зосереджених на повсякденному забезпеченні безпеки, що багато хто навіть не намагаються їм займатися. Засоби імітації зломів і атак допомагають перетворити оцінку стану безпеки в системний і автоматизований процес.

В таблиці 1.2 розглянемо переваги та недоліки тестування на наявність вразливостей автоматизованим методом (за допомогою різноманітних програм (як платних, так і безкоштовних) [6]

Таблиця 1.2 Переваги та недоліки автоматичного сканування на наявність вразливостей

Переваги	Недоліки
Може бути автоматизоване, просте, може виконуватися за розкладом	Відсутність знань про динаміку процесу
Знаходить відомі вразливості	Можливо побачити лише

	моментальний знімок стану безпеки
Результати можуть бути отримані всього за декілька годин	Погано виявляє невідомі або не популярні вразливості. Під час оновлення – ресурси компанії потрапляють у зону ризику
Не потребують спеціальних знань	Відображає дуже велику кількість вразливостей, як правило – 10% яких використовуються для реалізації атак, та потребуючі великих затрат
База відомих вразливостей постійно поповнюється	Високий коефіцієнт помилкових спрацювань – від 30 до 60%
Може бути більш економічним рішенням пентестінгу	Не враховуються міри протидій від самої вразливості
Можливе проведення декількох сканувань одночасно	Призначене для допоміжних, але не ключових систем, працюючих у реальному часі; Може збільшити навантаження на виробничу потужність системи та приводити до збоїв

Тести на проникнення (або пентести) виконуються вручну співробітниками компанії або зовнішніми консультантами, які намагаються оцінити захищеність інфраструктури організації шляхом її безпечного злому. Для цього можуть використовуватися уразливості операційних систем, служб або додатків, неправильні конфігурації або недостатньо обережне поводження користувачів. Іншими словами, проводиться атака на мережу, додатки, пристрої та співробітників організації з метою перевірити, чи можуть хакери здійснити такий злом. За результатами тестування також стає ясно, наскільки глибоко зміг би проникнути зловмисник і який обсяг даних він міг би вкрасти або використовувати в своїх цілях.

В таблиці 1.3 розглянемо переваги та недоліки ручного тестування (або пентестінгу)

Таблиця 1.3 Переваги та недоліки ручного тестування на проникнення

Переваги	Недоліки
Виявляються слабкі місця, які не можливо побачити під час автоматичного сканування на вразливості	Результат залежить від знань та досвіду аудитора
Можливо виявити критичні вразливості інфраструктури	Межі тестової середи не дають змогу використовувати усі можливості, які є у потенційного, реального хакера
Пентестер (він же аудитор) може використовувати будь – які технології для атаки – навіть ті, які з'явились сьогодні	Аудитор не може перевірити абсолютно всі існуючі технології атак та вразливостей
Результуючий звіт допомагає ліквідувати вразливості	Звіт про результати тестування потрібно очікувати декілька днів – тижнів
Ручне тестування на проникнення являться викликом для засобів забезпечення безпеки мережі	Не надає повної картини ситуації, адже в ручну неможливо перевірити усі аспекти системи (увесь код, де компілювання програми, тощо) Результати пентеста, проведеного в ручну, відображають статус системи в визначений відрізок часу. Висока вартість такого тестування не дозволяє проводити його часто

Імітації цілеспрямованих кібер-атак, також звані "тестуванням Red Team" або "дружнім взломом", завойовують все більшу популярність. Крім виявлення

критичних вразливостей і загальної оцінки безпеки попереджуючий підхід до тестування дозволяє отримати цінну інформацію про здатність IT-служб виявляти і блокувати атаки прямо під час їх здійснення. Атаки можуть складатися з безлічі етапів, що дозволяє імітувати різні типи супротивників і виявляти недоліки системи інформаційної безпеки.

В таблиці 1.4 розглянемо переваги та недоліки тестування методом “червоної команди”

Таблиця 1.4 - Переваги та недоліки тестування методом “червоної команди”

Переваги	Недоліки
Імітує тактику; технології та процедури справжніх хакерів	Імітації потрібно проводити регулярно
Дозволяє підготуватися до реальних атак за подібних обставин	Потребує підготовлену групу персоналу
Попереджуючий підхід	Результати тестування важко буде порівняти, тому що вони можуть бути виконані за різними правилами та в різних умовах
Більш економічне рішення в порівнянні з ручним тестуванням (пентестом)	В роботі беруть участь багато людей – як персонал компанії, так і найняті аудитори
Дає змогу знайти невідомі досі проблеми в не стандартних місцях; Дозволяє оцінювати ефективність інфраструктури безпеки	По причині недостатньої автоматизації – тестування важко повторювати по одному і тому сценарії

Розглянемо методи тестування на проникнення (пентест) за ДСТУ ISO/IEC TS 27008:2019 та стандарти OSSTMM, NIST SP800-115

- Тестування сліпим методом

Аудитор, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, тестує об'єкт перевірки без будь-яких попередніх знань та додаткових характеристик, крім загальнодоступних.

Об'єкт перевірки готується до перевірки особою, завчасно знаючою всі деталі перевірки. Сліпа перевірка в основному здійснюється на основі навичок аудитора, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою. Обсяг і глибина сліпої перевірки можуть бути настільки великими, наскільки дозволяють знання і працездатність аудитора. Таким чином, це тестування має обмежене застосування при перевірках безпеки і його слід уникати. Його зазвичай називають "етичним хакінгом".

- Тестування подвійним сліпим методом

Аудитор, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, тестує об'єкт перевірки без будь-яких попередніх знань та додаткових характеристик, крім загальнодоступних. Аудитор заздалегідь не повідомляє підприємство про область перевірки або які тести будуть використовуватися. При подвійній сліпій перевірці тестується готовність об'єкта перевірки до невідомих параметрів.

- Тестування методом сірого ящика

Аудитор, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, тестує об'єкт перевірки, маючи в своєму розпорядженні обмежені знання про його захист і активи, але повні знання про доступні тести. Об'єкт перевірки готується до перевірки особою, яка завчасно знає всі деталі перевірки. Перевірка методом сірого ящика здійснюється на основі навичок аудитора, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою. Основною властивістю цього тестування є результативність. Обсяг і глибина залежать від якості інформації, наданої аудитору, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, перед тестуванням, а також від належних знань аудитора, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою. Таким чином, це тестування має

обмежене застосування при перевірках безпеки і його слід уникати. Цей вид тестування часто називається "тестуванням вразливостей", і воно найчастіше зініціюється об'єктом як дії по самооцінці.

- Тестування методом подвійного сірого ящика

Аудитор, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, тестує об'єкт перевірки, маючи в своєму розпорядженні обмеженим знанням про його захист і активах, але повним знанням про доступні тестах. Аудитор заздалегідь повідомляє про область і терміни перевірки, але не про тести. Перевірка методом подвійного сірого ящика тестує підготовленість об'єкта до невідомим параметрами розгляду. Обсяг і глибина залежать від якості інформації, наданої аудитору, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, і об'єкту перевірки перед тестуванням, а також від застосовуваних знань аудитора, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою.

- Тестування тандемним методом

Аудитор, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, і об'єкт перевірки готуються до перевірки, для обох заздалегідь відомі всі деталі перевірки. При тандемному методі тестується захист і заходи та засоби контролю і управління об'єкта. Основною властивістю даного тестування є доскональність, оскільки аудитор, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, має повне уявлення про всі тестах і відповідні дії. Обсяг і глибина залежать від якості інформації, наданої аудитору, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, перед тестуванням, а також від належних знань аудитора, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою. Це тестування часто називають "внутрішньою перевіркою", і аудитор, який проводить перевірку заходів і засобів контролю та управління

інформаційною безпекою, часто відіграє активну роль в загальному процесі забезпечення безпеки.

- Інверсійний метод

Аудитор, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою, тестує об'єкт перевірки, маючи в своєму розпорядженні повним знанням про його процесах і операційної безпеки, однак об'єкту перевірки нічого не повідомляється про те, що, як або коли буде тестувати аудитор, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою. Основною властивістю цього тестування є перевірка підготовленості об'єкта до невідомим параметрами і напрямками розгляду. Обсяг і глибина залежать від якості інформації, наданої аудитору, який проводить перевірку заходів і засобів контролю та управління інформаційною безпекою.

Рисунок 1.4 – Методи тестування за ДСТУ ISO/IEC TS 27008:2019

Також, методики можна класифікувати за іншими стандартами, наприклад:

The Open Source Security Testing Methodology Manual - (OSSTMM) - один з найпопулярніших стандартів, розроблений Institute for Security and Open Methodologies (ISECOM). [8]

OSSTMM це універсальний стандарт, тому що може бути базовим керівництвом при проведенні тестування на проникнення. За допомогою OSSTMM пентестер може налаштувати індивідуальну оцінку рівня безпеки для кожної конкретної компанії, з огляду на її бізнес-процеси, технологічні та галузеві особливості.

OSSTMM пропонує детальний план тестування, метрики для оцінки рівня безпеки та рекомендації по складанню підсумкового звіту. Автори стандарту гарантують, що тест, який проведено згідно OSSTMM, буде докладним і всебічним, а результати - вимірними і заснованими на фактах.

NIST SP800-115 - стандарт інформаційної безпеки, розроблений National Institute of Standards and Technology. У підрозділі стандарту SP 800-115 описані технічні питання оцінки рівня інформаційної безпеки, порядок проведення тестування на проникнення, наведені рекомендації з аналізу результатів і розробці заходів щодо зниження ризиків. Остання версія документа приділяє велику увагу зниженню ризиків від кібер-атак. [9]

OWASP - Open Web Application Security Project (OWASP) - відкрите інтернет-співтовариство, яке пропонує саму вичерпну методологію для тестування додатків, сайтів і API. OWASP випустив кілька документів і керівництв. [10]

OWASP надає відкритий доступ до таких матеріалів, як:

- Керівництво для розробників (OWASP Developer Guide). Тут дані рекомендації з написання безпечного і надійного коду.
- Перевірка безпеки коду (OWASP Code Review). Керівництво можуть використовувати веб-розробники, а також менеджери продуктів. Документ пропонує ефективні методи перевірки безпеки існуючого коду. Методологія OWASP описує тестування на кожній стадії життєвого циклу розробки додатків: визначення вимог, проектування, розробка, впровадження та підтримка. При цьому тестуються не тільки самі додатки, але і технології, процеси, а також люди. Друга важлива перевага: OWASP можуть застосовувати як пентестери, так і веб-розробники.

Спільнота OWASP також випустило багатоплатформовий інструмент для проведення автоматичного тестування - OWASP ZAP, частково схожий з Burp Suite.

PTES - Penetration Testing Methodologies and Standards (PTES) пропонує рекомендації для проведення базового пентеста, а також кілька розширених варіантів тестування, для організацій з підвищеними вимогами до інформаційної безпеки. Одна з переваг PTES в тому, що він дає докладний опис цілей і очікувань від пентеста. [11]

З існуючих методологій можливо виділити такі, як:

Методологія OSSTMM пропонує п'ять основних каналів (напрямків) для тестування операційної безпеки. Поділ на канали допомагає комплексно оцінити рівень захищеності організації і полегшує процес тестування. [12]

- Безпека людини. Безпека, яка безпосередньо залежить від фізичного, якого психологічного взаємодії людей.
- Фізична безпека. Будь-який матеріальний (неелектронний) елемент безпеки, робота якого має на увазі фізичне або електромеханічне вплив.
- Безпровідний зв'язок. Безпека всіх бездротових засобів зв'язку, від Wi-Fi до інфрачервоних датчиків.
- Телекомунікації. Аналогові або цифрові засоби телефонного зв'язку. В основному це стосується телефонії, а також передачі службової інформації з телефонних ліній зв'язку.
- Мережі передачі даних. Безпека внутрішніх і зовнішніх корпоративних мереж, інтернет-підключень і мережевого устаткування.

NIST SP800-115 описує такі методи, як: [13]

- Методи обстеження: огляд документації, логів, правил, конфігурації системи, сніффінг мережі, перевірка цілісності файлів.
- Методи перевірки цільових вразливостей: злом паролів, соціальна інженерія, пентест.
- Оцінка безпеки: координація, обробка даних, аналіз і оцінка.

Дії за підсумками тестування:

- Рекомендації щодо зниження ризиків;
- Складання звіту,
- Усунення вразливостей.

ISSAF Information System Security Assessment Framework (ISSAF) розроблений Open Information Systems Security Group (OISSG) охоплює велику кількість питань, пов'язаних з інформаційною безпекою. У ISSAF наведені докладні рекомендації з тестування на проникнення. Описано утиліти, якими можна провести пентест, вказівки щодо їх використання, а також детально роз'яснено, які результати і за яких параметрах можна отримати в результаті тестування. [14]

ISSAF вважається досить складною і докладної методологією, яку можна адаптувати для перевірки інформаційної безпеки будь-якої організації. Кожен етап тестування згідно ISSAF ретельно документується. Також надано рекомендації щодо використання конкретних інструментів на кожному з етапів.

Методологія ISSAF пропонує конкретний порядок кроків при імітації злому:

- Збір інформації;
- Складання карти мережі;
- Виявлення уразливості;
- Проникнення;
- Отримання доступу і підвищення привілеїв;
- Утримання доступу (maintaining access);
- Компрометація віддалених користувачів і сайтів;
- Приховування слідів проникнення.

1.4 Постановка задачі

Згідно з проведеним аналізом щодо стану безпеки веб - ресурсів, та спираючись на аналіз статистичних даних OWASP TOP 10, Purplesec.us, CAPEC, SANS, та огляду стандартів OSSTMM та NIST SP800-115 з подальшим загальним оглядом основних етапів тестування на проникнення та

рекомендацій приходимо до висновку, що активний аудит безпеки веб – ресурсів є необхідністю. Згідно з проаналізованими даними про вразливості веб – ресурсів та кількості типів стандартизації тестування на проникнення як і їх методологій – розглянути кожну вразливість у дії та запропоновано шляхи як їх усунення, так і пом'якшення наслідків від їх реалізації. Розробити алгоритм на прикладі діяльності компанії Cymulate та однієї з методологій тестування на проникнення, згідно з ДСТУ ISO/IEC TS 27008:2019 та обґрунтовано її практичну цінність за допомогою поетапного аналізу створеного алгоритму.

Висновок

В першому розділі проаналізовано статистичні дані OWASP, Purplesec.us, CAPEC та розглянуто типові вразливості веб ресурсів, описано вектор використання вразливості зловмисником, та дані по порушенням безпеки на тих чи інших системах згідно зі статистикою OWASP TOP 10, Purplesec.us, CAPEC та SANS. Проаналізовано методики тестування за допомогою ДСТУ ISO/IEC TS 27008:2019, стандартів OSSTMM та NIST SP800-115,OWASP, методи ISSAF та PTE.

2 СПЕЦІАЛЬНА ЧАСТИНА

2.1 Загальні відомості про тестування на проникнення

Об’єкт аналізу типовий до існуючих веб ресурсів на ринку, але навмисно е вразливим. bWAPP – ресурс містить в собі не тільки вразливості, які входять то OWASP TOP 10 статистики кожного року, а і всі відомі – як популярні так і не популярні вразливості. Він допомагає ентузіастам безпеки, системним інженерам, розробникам і студентам як виявляти і запобігати атакам, так і ефективно боротися з вразливостями веб-ресурсів.[15]

Burp Suite – це платформа для виконання тестів з безпеки веб-додатків. Різні інструменти Burp ефективно працюють разом для підтримки всього процесу тестування, від складання карти сайту і аналізу поверхні атаки додатки до пошуку і експлуатації вразливостей безпеки. Burp дає повний контроль, дозволяє комбінувати просунуті ручні техніки з автоматизуванням.[16]

BurpSuite містить наступні ключові компоненти:

Перехоплює проксі, який дозволяє перевіряти і модифікувати трафік між вашим браузером і цільовим додатком;

Високий рівень сканер веб-додатків для автоматизованого виявлення ряду вразливостей;

Інструмент Intruder для виконання потужних призначених для користувача атак для пошуку і експлуатування незвичайних вразливостей;

Інструмент Repeater для маніпуляцій і повторної відправки індивідуальних запитів;

Інструмент Sequencer для тестування хаотичних сесійних токенів (маркерів). Можливість зберігати роботу і відновлювати робочий процес пізніше.

На прикладі bWAPP та за допомогою open source інформації ми проаналізуймо та проранжуємо вразливості типових веб – ресурсів більш детально, та виділимо ефективні шляхи та варіанти для пом'якшення негативного ефекту від їх реалізації, або повної чи часткової протидії.

2.2 Ранжування вразливостей та методи протидії їм

Згідно з проаналізованими дослідженнями Purplese.us, CAPEC та Owasp Top 10 ми маємо топ 10 найрозповсюджених вразливостей, а саме:

- Ін'єкції – SQL, NoSQL, OS, etc.;
- Вразлива автентифікація;
- Порушення конфіденційності даних;
- Ін'єкція зовнішніх сутностей XML;
- Порушений контроль доступу;
- Помилки в конфігураціях безпеки;
- Міжсайтовий скриптинг XSS;
- Небезпечна десеріалізація;
- Використання компонентів із відомими вразливостями;
- Недостатній моніторинг ресурсу;

Розглянемо вразливість через ін'єкцію на прикладі вразливості YITH WooCommerce Wishlist – плагіну WordPress.

Введення коду відбувається, коли зловмисник надсилає неправомірні дані до веб-програми з наміром змусити зробити щось, для чого додаток не був розроблений / запрограмований.

String query = “SELECT * FROM accounts WHERE custID = ‘” + request.getParameter(“id”) + “‘”;

Цей запит можна використати, викликавши веб-сторінку, що виконує його, із такою URL-адресою: http://example.com/app/accountView?id= 'або' 1 '=' 1, викликаючи повернення всіх рядків, збережених у таблиця бази даних.

Основною вразливістю введення коду є відсутність перевірки даних, або валідації що використовуються веб-додатком. Все, що приймає параметри як вхідні дані, може бути вразливим до ін’єкції.

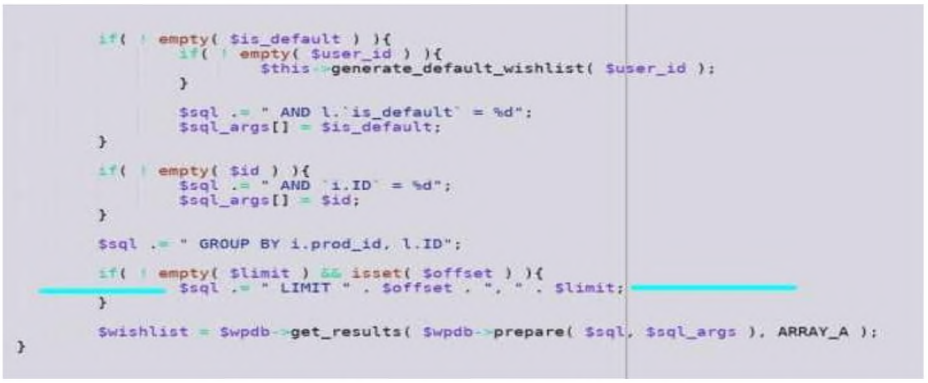

Рисунок 2.1 – Приклад SQL ін'єкції

Якщо зловмисники можуть встановити довільні значення для змінної $ limit, вони можуть змінити запит таким чином, що може призвести до повного скомпрометування даних на декількох серверах.

Як запобігти ін'єкціям SQL?

Залежить від технології, яку використовує веб – ресурс.

Якщо WordPress – звести до мінімуму використання довільних тем та кількість плагінів. Якщо у вас є спеціальний веб-додаток і спеціальна команда розробників, вам слід переконатися, що у вас є вимоги до безпеки, яких мають дотримуватися команда розробників в процесі написання програмного забезпечення. Це дозволить їм проводити аудит протягом життєвого циклу проекту та виконувати валідацію коду. Запобігання ін'єкціям SQL вимагає збереження даних окремо від команд та запитів.

- Використання безпечного API, який повністю уникає використання інтерпретатора або забезпечує параметризований інтерфейс або міграцію для використання Інструментів реляційного відображення об’єктів (ORM).
- Навіть якщо параметризовано, збережені процедури все одно можуть запровадити ін'єкцію SQL, якщо PL / SQL або T-SQL об'єднує запити та дані або виконує ворожі дані за допомогою EXECUTE IMMEDIATE або exec ().

Використовуйте позитивну перевірку або введення в “WhiteList” на стороні сервера. Це не повний захист, оскільки для багатьох програм потрібні спеціальні символи, такі як текстові області або API для мобільних додатків. Для будь-яких залишкових динамічних запитів уникайте спеціальних символів, використовуючи конкретний синтаксис.

Структуру SQL, таку як імена таблиць, назви стовпців тощо, неможливо захистити, і тому надані користувачем назви структур небезпечні. Це поширена проблема програмного забезпечення для написання звітів.

Використовуйте LIMIT та інші елементи керування SQL у запитах, щоб запобігти масовому розкриттю записів у разі скомпрометування.

З цих рекомендацій можна абстрагувати дві речі:
Відокремлення даних від логіки веб-додатків та впровадження, налаштування та / або обмеження експозиції даних у разі успішних атак ін’єкцією SQL;

Без відповідних заходів в ролі валідації – ін’єкції несуть за собою серйозний ризик для власників веб-сайтів. Також можливо використовувати додаткові утиліти для поліпшення безпеки шляхом сканування коду:

- HP Scrawlr - ця безкоштовна утиліта сканера може виявити та визначити, чи сприйнятливий ваш веб-сайт до вразливості введення SQL. Утиліта сканує веб-сайт, аналізуючи поля введення кожної веб-сторінки на наявність вразливостей введення SQL.

- Аналізатор вихідного коду Microsoft для ін'єкції SQL - цей засіб аналізує ваш статичний вихідний код ASP, написаний у VBScript (не ASP.NET), і

виявляє можливі вразливості до атак ін'єкції SQL. Потім інструмент формує звіт, в якому детально описуються виявлені вразливості.

Згідно до аналізу досліджень OWASP, наступною вразливістю розглянемо вразливу автентифікацію.

Ця вразливість дозволяє зловмиснику використовувати ручні та автоматичні методи, щоб спробувати отримати контроль над будь – яким обліковим записом, який є в системі.

На прикладі bWAPP при розгляді вразливої автентифікації ми маємо змогу побачити, що веб – ресурси з даною вразливістю дають зловмиснику побачити дані користувача при перегляді html коду сторінки.

Рисунок 2.2 Приклад витоку конфіденційної інформації при перегляді коду html сторінки

Тепер ми побачимо інший недолік відсутності валідації коду.

Обираємо ‘Session Mgmt. – Administrative Portals’ та встановлюймо рівень безпеки на “низький”. Якщо ви помітили URL-адресу '/bWAPP/smgmt_admin_portal.php?admin=0', після '?' Додається рядок зі значенням '0', що означає, що ідентифікатор сеансу був переданий у рядок запиту, де кожен міг бачити і маніпулювати цінностями.

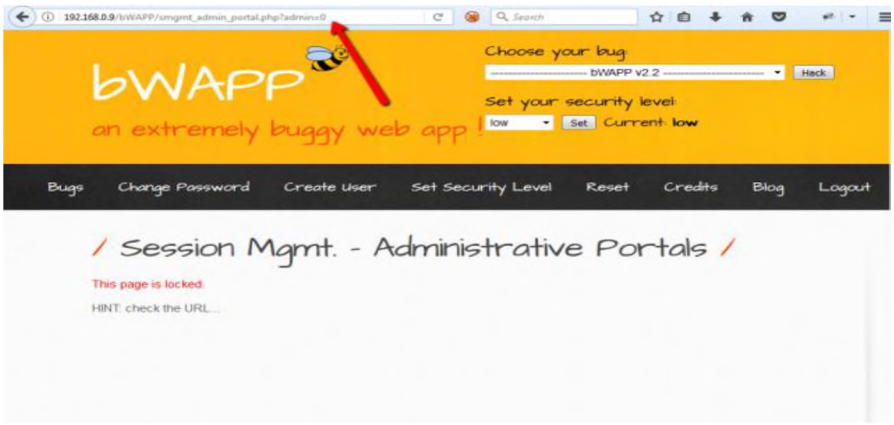

Рисунок 2.3 Приклад підбору id сторінки для отримання доступу до конфіденційних даних

Аналізуючи існуючі вразливості, зв'язані з автентифікацією, можемо виділити такі основні, як:

- Веб ресури, які використовують звичайний, не зашифрований або слабко хешований пароль користувачів та адміністрації;
- Відсутня, або неефективна багатофакторна автентифікацію;
- Використовують слабкі, або неефективні процеси відновлення паролів, такі як "секретні питання" або не мають відсрочення змоги логіну після декількох не вдалих спроб – bruteforce;

Валідація коду та тестування на проникнення можуть бути використані для діагностики проблем автентифікації та управління сеансами. Ми повинні ретельно переглянути кожен аспект механізму автентифікації, щоб забезпечити постійний захист облікових даних користувача. Незважаючи на те, що ми не розглядали функції забутого / зміненого пароля, потрібно ще раз перевірити реалізації цих механізмів.

Тезісно, можемо виділити такі рішення, як

- Там, де це можливо, впроваджуйте багатофакторну автентифікацію, щоб запобігти автоматизованим заповненням облікових даних, та брутфорсом;

- Впровадьте перевірку слабких паролів, наприклад, тестування нових або змінених паролів у списку 10 000 найгірших паролів;

- Поєднайте політику довжини, складності та ротації паролів із вимогами NIST 800-63 B у розділі 5.1.1 щодо «Запам'ятаних секретів» або іншими сучасними, доказовими політиками щодо паролів;

- Обмежте або все частіше затримуйте невдалі спроби входу. Записуйте всі помилки та попереджуйте адміністраторів, коли виявляються набивання облікових даних або брутфорс;

- Використовуйте захищений вбудований менеджер сеансів на стороні сервера, який генерує новий ідентифікатор випадкової сесії з високою ентропією після входу – наприклад CyberArk. Ідентифікатори сеансів не повинні бути в URL-адресі. Ідентифікатори також слід надійно зберігати та анулювати після виходу, простою та абсолютних таймаутів;

Розглянемо наступну вразливість.

Витік конфіденційних даних є однією з найпоширеніших вразливостей в аналітиці OWASP.

Ми повинні захистити такі конфіденційні дані, як:

- Повноваження доступу;
- Дані кредитних карт клієнтів;
- Паспортні дані або будь які дані ідентифікації користувача;
- Медична інформація та ін.;

Згідно з загальним регламентом GPDR – всі компанії, які збирають, змінюють, оброблюють та зберігають чи видаляють персональні дані громадян ЄС – як резидентів так і відвідувачів – несуть за них повну відповідальність.

Існує два типи конфіденційних даних:

Збережені дані - дані в стані спокою, які зберігаються на носіях чи серверах;

Транзуючі дані - дані, які передаються внутрішньо між серверами або веб-браузерам.

Обидва типи даних потребують захисту. Одним та найголовнішим із способів захистити їх на веб-сайті є SSL сертифікат.

Сертифікат SSL. Це стандартна технологія безпеки для встановлення зашифрованого зв'язку між веб-сервером і браузером. Сертифікати SSL допомагають захистити цілісність даних, що передаються між хостом (веб-сервером або брандмауером) та клієнтом (веб-браузер).

Не шифрування конфіденційних даних є основною причиною того, чому ці атаки все ще широко поширені. Навіть зашифровані дані можна зламати через слабкі:

Цю вразливість зазвичай дуже важко використати; проте наслідки успішного нападу можуть бути фатальними.

Перелічимо способи та рекомендації за допомогою яких ми можемо мінімізувати витік конфіденційних даних:

- Класифікувати оброблювані дані, які зберігаються чи передаються веб – ресурсом підприємства;
- Розмежувати засоби контролю доступу відповідно до класифікованої інформації;
- Не зберігати конфіденційні дані без потреби – якщо вони не використовуються в вашій бізнес – моделі – видалить їх, скористайтесь скороченням або токенізацією. Дані що не зберігаються – неможливо вкрасти;
- Шифрувати всі дані під час процесу передачі (транспортування) за допомогою протоколів захисту, таких як TLS, з шифрами прямої секретності (PFS);
- Використовувати HSTS - Strict Transport Security;
- Не використовувати хешування для відповідей, які містять конфіденційні дані;
- Зберігати паролі та дані конфіденційного класу за допомогою сильних адаптивних хеш – функцій з коефіцієнтом затримки, такі як Argon2, bcrypt, PBKDF2;
- Виконувати валідацію загального коду та всіх директив зберігання, конфігурацій та налаштувань для унеможливлення використання експлоітів.

XML ін’єкція

XML ін’єкція, або XXE – це тип атаки на додаток, який аналізує XML та може призвести до витіку конфіденційних даних навіть до віддаленого виконавця коду (RCE).

Більшість аналізаторів XML як правило вразливі до атак XXE, тому відповідальність за те, щоб додаток був повністю захищено від цієї атаки полягає повністю на розробника.

Як це працює? Продемонструємо на прикладі bWAPP та Burpsuite.

- XML External Entity Attacks і встановлюємо низький рівень безпеки;

Перш ніж натиснути на кнопку, ми просто захочемо перехопити пакет, перш ніж він надійде на сервер, і перевірити його на власні очі. Для цього вам знадобиться налаштування Burpsuite для ролі перехоплюючого проксі.

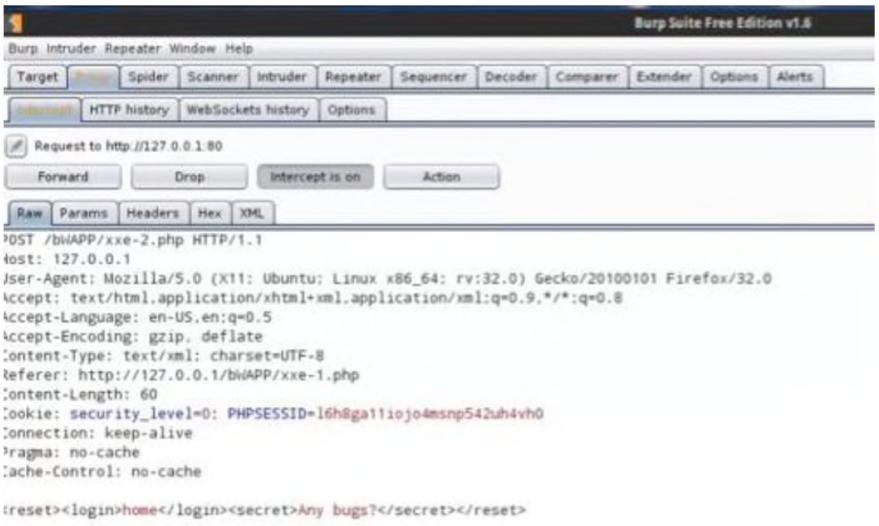

Рисунок 2.4 Інтерфейс Burpsuite та потрібного нам перехопленого пакету

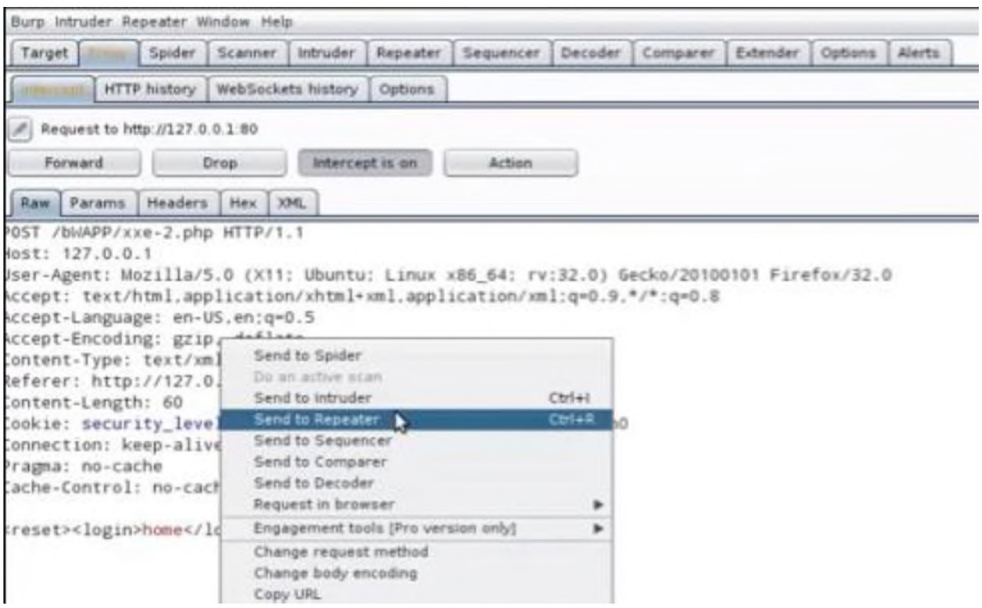

Рисунок 2.5 Інтерфейс Burpsuite та потрібного нам перехопленого пакету – відсилання до репітеру

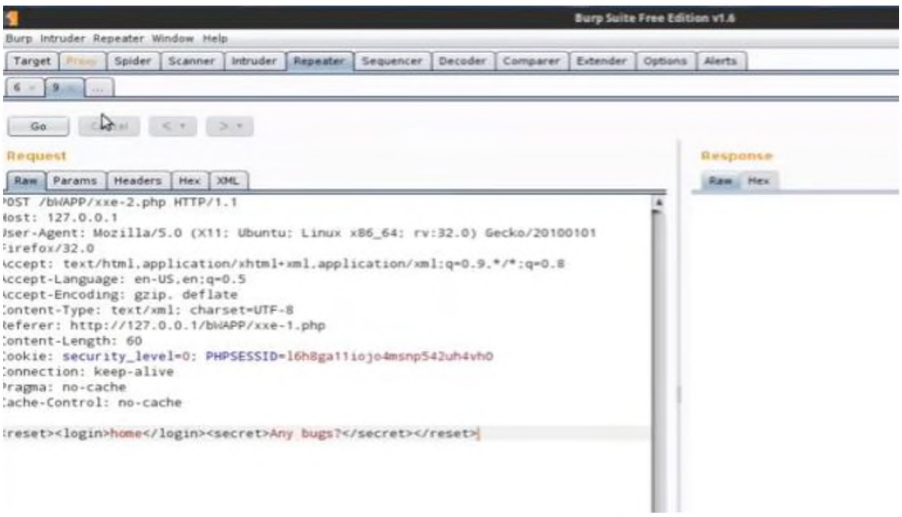

Рисунок 2.6 Іінтерфейсу Burpsuite та потрібного нам перехопленого пакету – маніпулювання

Тепер, коли у нас є наш запит у ретрансляторі, ми можемо маніпулювати пакетом і подивитися результати. Додамо власні теги XML.

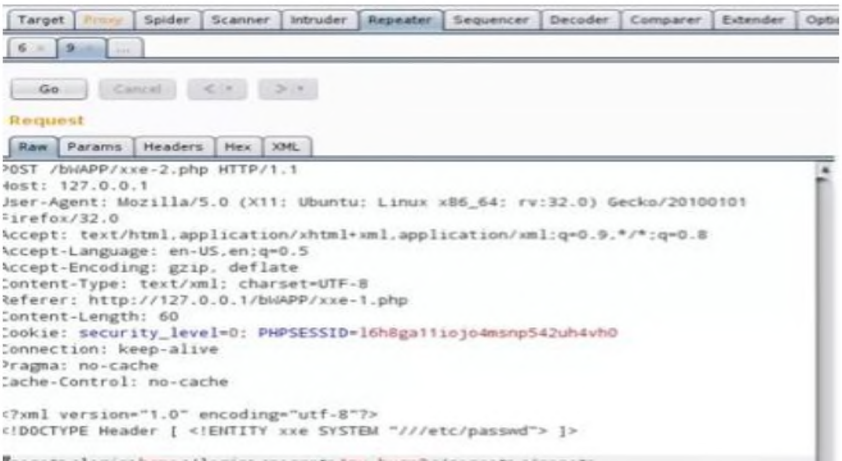

Рисунок 2.7 Інтерфейс Burpsuite та потрібного нам перехопленого пакету – маніпулювання тегами

Для нас це простий запит за допомогою якого ми зможемо увійти у файлову систему сервера та вивести вміст файлу / etc / passwd. Тож давайте надішлемо запит, та перевіримо відповідь на нашій панелі відповідей у ретрансляторі.

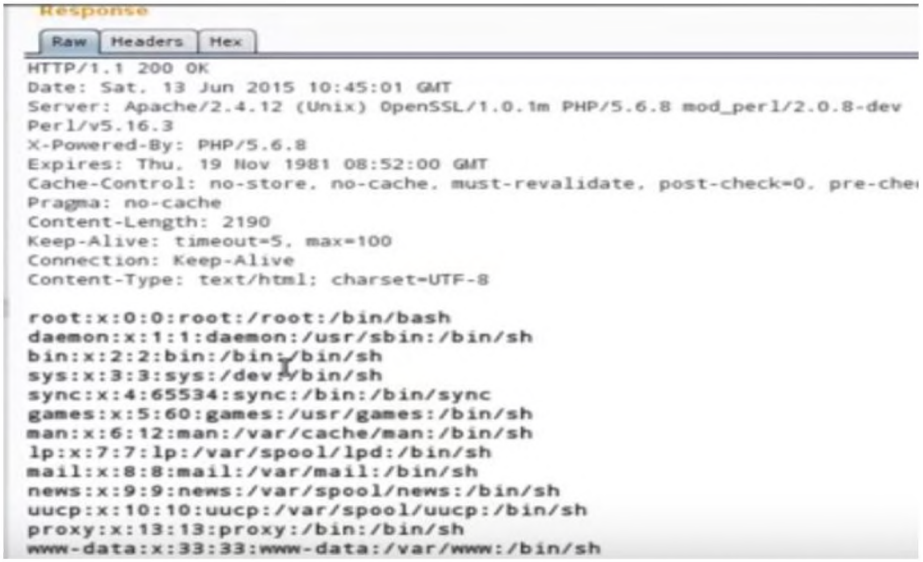

Рисунок 2.9 Відповідь bWAPP на нашу XML ін’єкцію

Отримуємо відповідь 200 OK і вміст папки, яку запитували.

Найбезпечніший спосіб запобігти XXE - це завжди повністю відключити DTD (Зовнішні об’єкти).

Залежно від синтаксичного аналізатора, метод повинен бути подібним до наступного:

factory.setFeature ("http://apache.org/xml/features/disallow-doctype-decl", true);

Вимкнення DTD також робить аналізатор захищеним від атак відмови в послугах (DOS), таких як Billion Laughs. Якщо неможливо повністю вимкнути DTD, тоді зовнішні сутності та зовнішні декларації типу документа повинні бути вимкнені способом, специфічним для кожного синтаксичного аналізатора.

Також можна виділити такі рекомендації, як:

- Використовувати менш складні формати даних, такі як JSON, і уникати серіалізації конфіденційних даних;
- Оновити всі XML-процесори та бібліотеки, що використовуються додатком або базовою операційною системою;
- Використовувати засоби перевірки залежностей останньої версії – SOAP (Simple Object Access Protocol);
- Засоби SAST можуть допомогти виявити XXE у вихідному коді, але все ж ручна валідація залишається найкращою альтернативою у великих та складних програмах з інтеграціями.

Вразливість контролю доступу

У безпеці веб-ресурсів контроль доступу означає обмеження кількості розділів або сторінок, які відвідувачі можуть побачити та отримати до них доступ.

Наприклад, якщо ви є власником магазину електронної комерції, вам, ймовірно, потрібен доступ до адміністративної панелі, щоб додати нові товари або налаштувати рекламну акцію на майбутні свята. Однак навряд чи це знадобиться комусь іншому. Дозвіл решті відвідувачів веб-сайту перейти на вашу сторінку входу лише відкриває ваш магазин електронної комерції для атак. І в цьому проблема майже всіх основних систем управління вмістом (CMS - Content Management System).

За замовчуванням вони надають доступ на сторінку входу адміністратора. Більшість з них також не змусять вас встановити двофакторний метод автентифікації (2FA).

Приклади контролю доступу в зоні ризику:

- Доступ до контрольної / адміністративної панелі хостінгу;

- Доступ до сервера через FTP / SFTP / SSH;
- Доступ до адміністративної панелі веб-сайту;
- Доступ до інших програм на вашому сервері;
- Доступ до бази даних;

Зловмисники можуть використовувати недоліки авторизації, щоб: отримати доступ до несанкціонованих функціональних можливостей та / або даних, переглядати конфіденційні файли, змінювати права доступу

Такі рекомендації можна виділити щодо запобігання вразливостям контролю доступу:

- Розмежовуйте ролі по правам доступу;
- Завжди видаляйте не актуальні ролі;
- Вводьте аудит до веб сайту та хостингу – хто та що робить, коли і чому;
- Вводьте двофакторну автентифікацію;
- Вимкніть перелік каталогів веб – сервера та переконайтесь, що метадані файлів (.git наприклад) та файли резервних копій відсутні в коренях веб;
- Впроваджуйте механізм контролю доступу один раз і повторно використовуйте, мінімізуючи CORS (Cross-origin resource sharing);
- Моделі контролю доступу повинні забезпечувати право власності на записи, а не визнавати, що користувач може створювати, читати, оновлювати або видаляти будь-які записи.

Помилки в конфігурації безпеки.

За своєю сутністю, вразливість конфігурації реалізується підбором, або Brute Force - це випробування багатьох можливих комбінацій. Існують також такі вразливі вектори, як:

- Невиправлені недоліки
- Конфігурації за замовчуванням
- Невикористані сторінки
- Незахищені файли та каталоги
- Непотрібні послуги

Однією з найпоширеніших вразливостей є збереження конфігурацій CMS за замовчуванням.

Сучасні CMS-програми (хоча і прості у використанні) можуть бути складними для кінцевих користувачів. Безумовно, найпоширеніші реалізації вразливостей повністю автоматизовані. Багато з цих атак покладаються на те, що користувачі мають лише налаштування за замовчуванням. Це означає, що велику кількість атак можна відсіяти, змінивши налаштування за замовчуванням при встановленні CMS.

Помилки в конфігурації безпеки можуть охоплювати та траплятися як вразливість на будь-якому рівні стека додатків, включаючи:

- Мережеві послуги;
- Платформа;
- Веб-сервер;
- Сервер додатків;
- База даних;
- Код сторінки;
- Попередньо встановлені віртуальні машини;
- Контейнери зберігання;

Розглянемо приклади реалізації вразливостей:

Сервер додатків має пре - інстальовані зразки програм, які не видаляються з робочого сервера. Ці програми можуть мати відомі недоліки безпеки, які зловмисники використовують для компрометації сервера. Якщо однією з цих програм є консоль адміністратора, і облікові записи за замовчуванням не були змінені, зловмисник входить із паролями за замовчуванням і бере “керування” на себе.

- Постачальник хмарних послуг має за замовчуванням видаляти дозволи на спільний доступ, які відкриті Інтернету іншими користувачами CSP. Це дозволяє зловмиснику отримати доступ до збережених конфіденційних даних в хмарному сховищі.

Рекомендації щодо найліпшого запобігання цьому типу можемо виділити такі:

- Видаліть та не використовуйте не потрібні програмні каркаси та функції;
- Завжди перевіряйте та розмежовуйте дозволи у хмарному сховищі;
- Середовища розробки, контролю якості та виробництва повинні налаштовуватися однаково, з різними даними, що використовуються в кожному середовищі. Бажано автоматизувати цей процес для мінімізації зусиль.
- Сегментована архітектура додатків, яка забезпечує ефективне та безпечне розділення між компонентами з сегментацією, контейнеризацією або групами хмарної безпеки.

Міжсайтовий скриптинг (XSS)

Широко поширена вразливість. Атаки XSS полягають у введенні шкідливих сценаріїв на стороні клієнта на веб-сайт та використовуючи веб-ресурс як метод розповсюдження. Він дозволяє зловмиснику вводити шкідливий зміст на веб-сайт і змінювати спосіб його відображення, змушуючи браузер жертви виконувати код, наданий зловмисником, під час завантаження сторінки. XSS присутній приблизно у двох третинах усіх додатків.

Як правило, вразливості XSS вимагають певного типу взаємодії з боку користувача, або через соціальну інженерію, або під час відвідування певної сторінки. Якщо вразливість XSS не виправлена, це може бути дуже небезпечним для будь-якого веб-сайту.

Типи атак XSS:

- Відображений XSS: Додаток або API включають в себе неперевірені користувацькі введення як частину виводу HTML. Успішна атака може дозволити зловмисникові виконувати довільний HTML та JavaScript у браузері жертви. Зазвичай користувачеві доведеться взаємодіяти з деяким шкідливим посиланням, яке вказує на сторінку, контрольовану

зловмисником, наприклад, зловмисні веб-сайти, що рекламують товари, тощо.

- Збережений XSS: Додаток або API зберігає дані користувача, які згодом переглядаються іншим користувачем або адміністратором. Зберігається XSS часто вважають високим або критичним ризиком.
- DOM XSS: програмні каркаси JavaScript, односторінкові програми та API, які динамічно включають керовані зловмисником дані на сторінку, є вразливими до DOM XSS. В ідеалі додаток не надсилатиме контрольовані зловмисником дані до небезпечних JavaScript API. Типові атаки XSS включають викрадення сеансів, поглинання облікового запису, обхід MFA, заміну або деформацію вузла DOM (наприклад, панелі входу троянських програм), атаки на браузер користувача, такі як завантаження шкідливого програмного забезпечення, реєстрація клавіш та інші атаки на стороні клієнта.

Рекомендації щодо зниження ризику реалізації вразливості XSS

Насамперед рекомендується використовувати зовнішні брандмауери, такий, як Suciri WebSite Firewall – він помітно унеможливить а у разі успіху – якомога сильніше послабить вектор та силу атаки.

- Профілактичні заходи для зменшення ймовірності XSS-атак повинні враховувати відокремлення ненадійних даних від активного вмісту браузера;
- Використовуйте програмні каркаси, які автоматично уникають XSS за дизайном, такі, як Ruby on Rails, React JS.
- Уникайте ненадійних даних запиту HTTP на основі контексту у вихідних даних HTML (тіло, атрибут, JavaScript, CSS або URL-адреса) дозволить усунути вразливості Reflected and Stored XSS;
- Застосовуйте контекстно-залежне кодування при зміні документа браузера на стороні клієнта діє проти DOM XSS. Якщо цього не вдається уникнути, до API браузера можна застосувати подібні контекстно-залежні техніки екранування.

- Увімкнення політики безпеки вмісту (CSP) - це поглиблений захист, що пом’якшує контроль проти XSS. Це ефективно, якщо не існує інших вразливих місць, які дозволяли б розміщувати шкідливий код за допомогою локального файлу (наприклад, перезаписування або вразливі бібліотеки з дозволених мереж доставки вмісту).

Вразливість серіалізації користувача

Цей тип вразливості був доданий опитуваннями у сфері безпеки інформації, а не на основі кількісних досліджень.

Процес серіалізації - це перетворення об’єктів у байтові рядки. Процес десеріалізації - це перетворення рядків байтів в об’єкти.

Приклади сценаріїв вразливої десериалізації:

- Додаток React викликає набір послуг Spring Boot. Будучи функціональними програмістами, команда А намагалася забезпечити незмінність її коду. Рішення, яке вони придумали, полягає у серіалізації стану користувача та передачі його туди-сюди з кожним запитом. Зловмисник помічає підпис об'єкта Java “R00” і використовує інструмент Java Serial Killer для віддаленого виконання коду на сервері додатків.
- PHP використовує серіалізацію об’єктів для збереження „супер” файлу cookie, що містить ідентифікатор користувача, роль, пароль та ін.:

a:4:{i0;i:132;i:1;s:7:"Mallory";i:2;s:4:"user";i:3;s:32:
”b6a8b3bea87fe0e05022f8f3c88bc960″;}

Зловмисник змінює серіалізований об'єкт, щоб надати собі права адміністратора:

a:4:{i:0;i:1;i:1;s:5:"Alice";i:2;s:5:"admin";i:3;s:32:
”b6a8b3bea87fe0e05022f8f3c88bc960″;}

Одним із векторів атак, представлених щодо цього ризику безпеки, був супер-cookie, що містить серіалізовану інформацію про зареєстрованого користувача. У цьому файлі cookie була вказана роль користувача. Зловмисник зможе успішно десеріалізувати та змінити об'єкт, щоб надати собі роль

адміністратора, та знову серіалізувати його. Цей набір дій може скомпрометувати весь веб-ресурс.

Найкращий спосіб захистити веб-програму від цього типу ризику - не приймати серіалізовані об’єкти з ненадійних джерел.

Також можемо виділити такі рекомендації, як:

- Впровадження перевірок цілісності, таких як цифрові підписи, на будь-яких серіалізованих об'єктах, щоб запобігти створенню ворожих об'єктів або фальсифікації даних;

- Застосування жорстких обмежень типу під час десеріалізації перед створенням об'єкта, оскільки код зазвичай очікує визначеного набору класів. Були продемонстровані обходи цієї техніки, тому покладатися виключно на це не бажано.

- Виділення та запуск коду, який десеріалізується в середовищах з низькими привілеями, коли це можливо.

- Реєстрація винятків та помилок десеріалізації, наприклад, коли вхідний тип не є очікуваним типом, або десеріалізація видає винятки.

- Обмеження або моніторинг вхідного та вихідного підключення до мережі з контейнерів або серверів, які десеріалізуються.

- Моніторинг десеріалізації, попередження, якщо користувач постійно десеріалізується.

Дев’ята та передостання вразливість – використання скомпрометованих компонентів, або компонентів з існуючими, відомими вразливостями.

У 2020 році 56% усіх програм CMS були застарілими у користувачів на момент зараження.

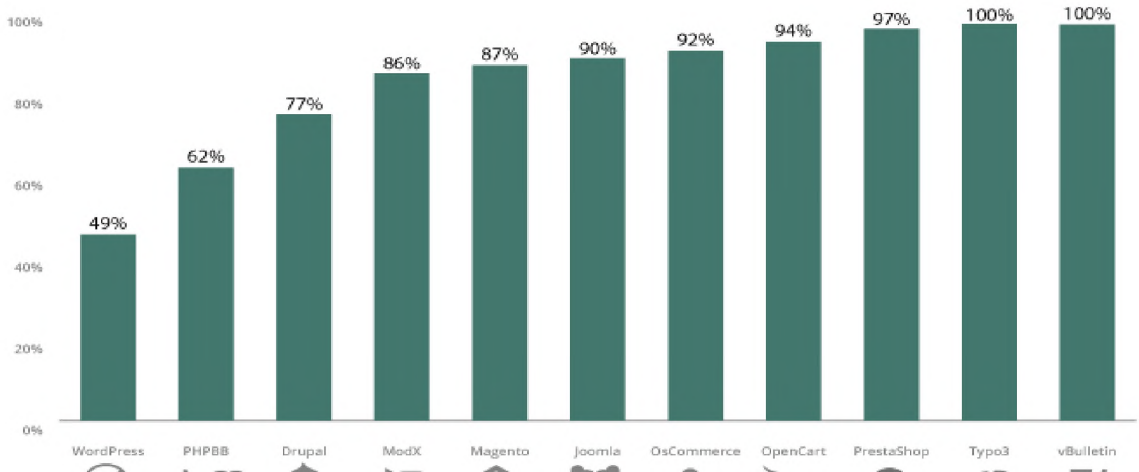

Рисунок 2.10 Графічне відображення процентної кількості атак на компоненти з вразливістю у 2020 році

Виділимо декілька способів боротьби з цією вразливістю:

- Видаліть усі непотрібні залежності;
- Складіть опис усіх своїх компонентів на стороні клієнта та сервера;
- Відстежуйте такі джерела за допомогою (CVE) та Національної база даних про вразливості (NVD) на наявність вразливостей у компонентах;
- Встановлюйте компоненти тільки з надійних джерел (офіційних). - Позбудьтесь компонентів, які не підтримуються активно;
- Використовуйте віртуальне виправлення за допомогою фаєрволу;
- Недостатній моніторинг та покриття логами;

Останнім пунктом для розгляду згідно з аналізом статистики OWASP є недостатній моніторинг веб ресурсів, їх аудит та покриття логами;

Недостатній моніторинг може привести до таких сценаріїв:

- Програмне забезпечення форуму з відкритим кодом, кероване невеликою командою, було зламане через недолік програмного забезпечення. Зловмисникам вдалося знищити внутрішнє сховище вихідного коду, що містить наступну версію та весь вміст форуму. Хоча джерело можна було відновити - відсутність моніторингу, логів призвело би до набагато гіршого порушення. Внаслідок цієї проблеми проект програмного забезпечення форуму більше не активний.
- Зловмисник сканує користувачів із загальним паролем. Вони можуть взяти на себе всі облікові записи за допомогою цього пароля. Для всіх інших користувачів це сканування залишає лише один помилковий логін. Через кілька днів це може бути зроблено повторно з іншим паролем.

Як можна запобігти цій вразливості?

Ведення журналів аудиту є важливим, щоб не відставати від будь-яких підозрілих змін на веб-сайті.

Журнал аудиту - це документ, який фіксує події на веб-сайті, щоб ви могли виявити аномалії та підтвердити з відповідальною особою, що обліковий запис не скомпрометований.

2.3 Rapid 7 як інструмент аудиту безпеки веб – ресурсів

Розглянемо програмне забезпечення Rapid7 як ще один ключ до вирішення проблем сучасної галузі IT, а саме – аудиту безпеки веб-ресурсів.

Rapid 7 має 4 основних продукти, а саме:

- InsightIDR;
- InsightVM;
- InsightAppSec;
- InsightConnect;

InsightIDR від Rapid7 – це центр безпеки для виявлення і реагування на інциденти, моніторинг автентифікації і видимість кінцевих точок.

InsightIDR виявляє несанкціонований доступ з зовнішніх і внутрішніх загроз і виділяє підозрілі дії.

InsightIDR об'єднує всю міць криміналістичної експертизи кінцевих точок, пошуку по журналам і складних панелей моніторингу в одному рішенні.

InsightIDR об'єднує дані на локальному або виділеному хост-комп'ютері, який централізує ваші дані. Rapid7 використовує аналітику цих даних для зіставлення користувачів, облікових записів, аутентифікації, попереджень і привілеїв. Аналіз дає уявлення про поведінку користувача при пошуку відомих індикаторів компрометації. Rapid7 рекомендує зберігати виділені колектори локально для збору даних про події, даних журналів і даних кінцевих точок.

Коли ви підключаєте всі потоки даних до InsightIDR, то можете скористатися всіма вбудованими функціями, а саме:

- Об'єднання даних в єдиному звіті безпеки;
- Аналіз необроблених журналів, даних кінцевих точок і мережевого трафіку;

- Отримання оповіщення про підозрілу активність;
- Пріоритет подій;
- Розслідування події.

InsightVM надає повністю доступний, масштабований і ефективний спосіб збору даних про уразливість, шукає рішення проблем і мінімізує ризики. InsightVM використовує технології аналітики і кінцевих точок для виявлення вразливостей в режимі реального часу, локалізує їх, передає на подальшу діагностику.

InsightAppSec допоможе вам частково перекрити відсутність валідації - увійдіть в систему і почніть сканування. Сканування в InsightAppSec можна налаштувати для задоволення потреб в тестуванні і забезпечення повного охоплення додатків. Охоплення сканування в сучасних додатках і API-інтерфейсів може бути проблемою для деяких інструментів DAST, але механізм сканування InsightAppSec був розроблений з урахуванням цих проблем.

Хоча InsightAppSec зв'язано з хмарним сервером - він також може сканувати внутрішні програми (наприклад, тестові білди) за допомогою механізму сканування. Всі результати зберігаються в хмарі, та тільки вам надається доступ до цієї інформації задля аудиту.

InsightConnect - це рішення для організації та реагування на автоматичну реакцію (SOAR) Rapid7 - за допомогою нього ви можете пришвидшити, впорядкувати та інтегрувати ваші трудомісткі процеси безпеки, практично не вимагаючи кодування, необхідного вашій команді безпеки. Коли ви використовуєте InsightConnect, ви можете автоматично запускати процеси з кількома рішеннями.

2.3 Розробка алгоритму прийняття рішень та методологій при тестуванні веб – ресурсу на проникнення

Алгоритм прийняття рішень - спосіб компактного представлення моделі зі складною логікою; інструмент для упорядкування етапів тестування на проникнення, які повинні бути реалізовані в продукті. Це взаємозв'язок між

безліччю умов і дій. В алгоритмі прийняття рішень представлений набір умов, одночасне виконання яких повинно привести до успішного проведення аудиту безпеки веб – ресурсів шляхом тестування на проникнення.

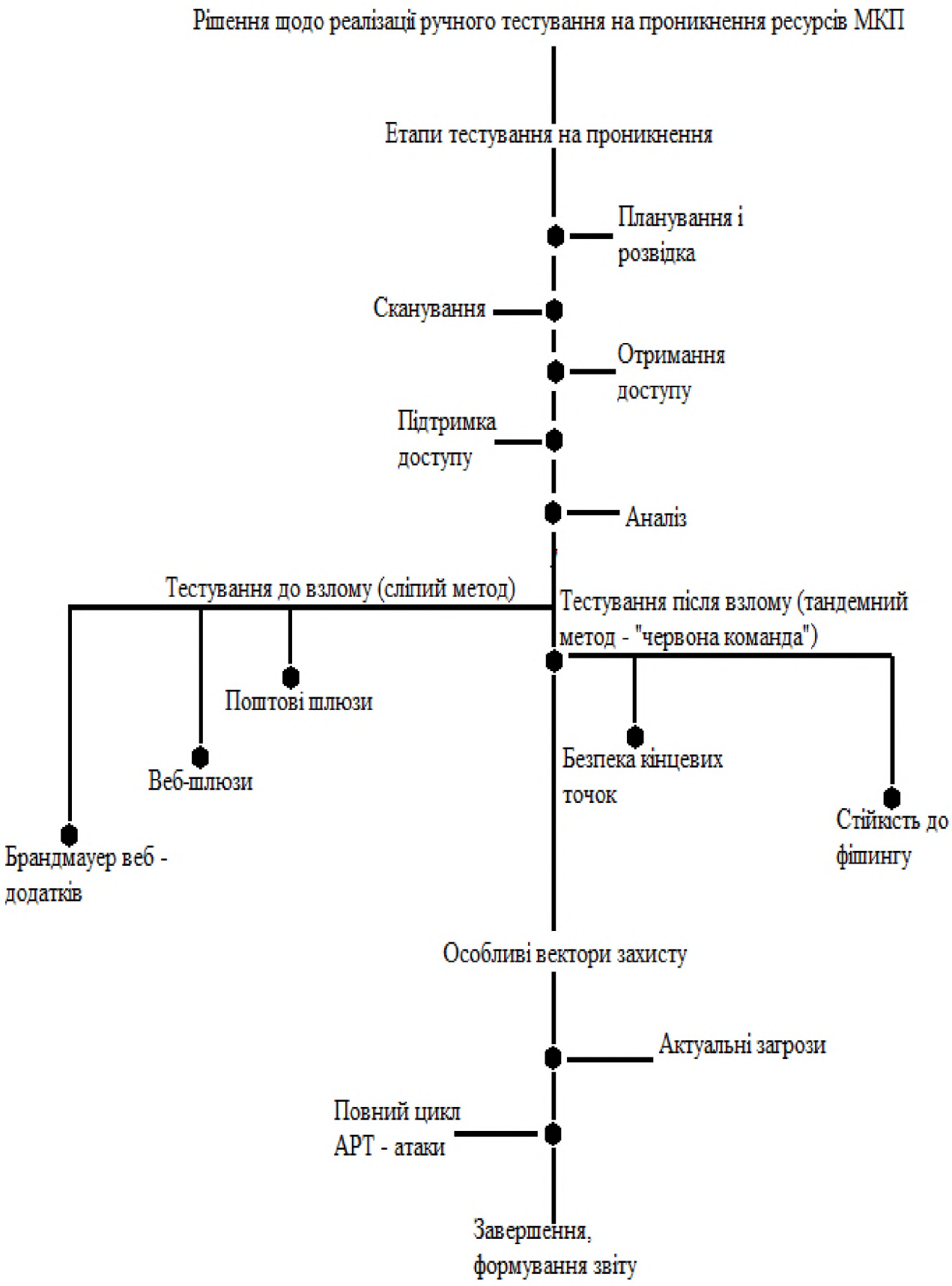

Рисунок 2.11 Алгоритм обрання рішень під час тестування на проникнення

Тестування на проникнення можливо розділити на 5 етапів [17]

1. Планування і розвідка

Визначення сфери охоплення і цілей випробування, включаючи системи, які повинні бути розглянуті, і методи випробування, які повинні бути використані.

Збір розвідувальних даних (наприклад, мережеві і доменні імена, поштовий сервер) для кращого розуміння того, як працює об'єкт і його потенційних вразливостей.

2. Сканування

Наступним кроком є розуміння того, як тестований об’єкт буде реагувати на різні спроби вторгнення. Зазвичай це робиться за допомогою:

- Статистичного аналізу - перевірка коду програми для оцінки його поведінки під час роботи. Ці інструменти можуть сканувати весь код за один прохід.
- Динамічного аналізу - перевірки коду програми в робочому стані. Це більш практичний спосіб сканування, так як він забезпечує нас інформацією про продуктивність програми в реальному часі.

3. Отримання доступу

На цьому етапі використовуються атаки веб-ресурсів типової досліджуваної організації, такі як міжсайтовий скриптинг, SQL-ін'єкція і експлоіти, для виявлення вразливостей.

Пентестери намагаються використовувати ці уразливості, як правило, шляхом підвищення привілеїв, крадіжки даних, перехоплення трафіку і т.п., щоб зрозуміти, шкоди яких масштабів вони можуть завдати.

4. Підтримка доступу

Мета цього етапу полягає в виявленні чи можна використовувати вразливість для досягнення постійної присутності в експлуатованої системі - досить довго, щоб поганий актор отримав до неї глибокий доступ. Ідея полягає в тому, щоб імітувати сучасні постійні погрози, які часто залишаються в

системі протягом декількох місяців з метою крадіжки самої конфіденційної інформації організації.

5. Аналіз.

Результати тесту на проникнення потім зводяться до звіту з детальним описом:

- Конкретні вразливості, які були використані та знайдені;
- Чи був отриманий доступ до конфіденційних даних;
- Кількість часу, протягом якого тестер зміг залишитися в системі непоміченим.

Основне завдання тесту на проникнення — ідентифікація максимально можливого числа вразливостей інформаційної системи (ІС) за обмежений час при певних умовах і стані ІС.

При приведенні алгоритму тесту на проникнення в дію, вирішуються такі завдання, як:

- оцінка поточного стану системи захисту інформації ІС;
- виявлення вразливостей інформаційної системи;
- використання виявлених вразливостей для отримання несанкціонованого доступу чи здійснення несанкціонованого впливу на інформацію для демонстрації наявності вразливостей і існування високо ймовірної загрози інформаційної системи;
- вироблення рекомендацій щодо підвищення ефективності захисту інформації в ІС.

Алгоритм включає в себе комбіновану методологію компанії Cymulate та типовий “сліпий” метод тестування, згідно з ДОДАТКОМ Д – Методологія Cymulate.

Платформа імітації зломів і атак (BAS) Cymulate розвиває ідею імітації цілеспрямованих атак і оцінює фактичну готовність організації до відбиття кібер-загроз. Cymulate виявляє критичні уразливості інфраструктури,

проводячи кібер-атаки з кількох векторах так, як це робили б реальні зловмисники. Пробні атаки здійснюються по шаблонах реальних хакерських угруповань, державних кібер-військ і навіть від імені уявних неблагонадійних співробітників. Модель SaaS дозволяє запускати імітації в будь-який час і в будь-якому місці, не впливаючи на користувачів або інфраструктуру. Завдяки платформі Cymulate організації можуть безперервно тестувати стійкість своєї інфраструктури до кібер-атаки, глобальним епідеміям вірусів і цілеспрямованим APT-атакам.

Цілі алгоритму – якомога ліпше та найшвидше перевірити основні частини життєвої сутності веб – ресурсів підприємства.

Алгоритм включає в себе не тільки етапи підготовки, та сліпе тестування брандмауера та веб + почтових ресурсів, а і залучення "червоної команди" аудитора, та мануалів з тестування для створення "червоної команди" в межах підприємства. "Червона команда" починає з перевірки безпеки кінцевих точок та стійкості ресурсів то фішингу.

Після цього – тестові групи об`єднаються, та перевіряють особливі вектори захисту за допомогою актуальних вразливостей та запускають повний цикл атаки APT – KillChain. Після цих всіх дій – формується звіт.

Висновок

У даному розділі було проаналізовано вразливості веб – ресурсів на прикладі спеціалізованого середовища для тестування – bWAPP. Було запропоновано методи протидії та дані поради для пом'якшення або унеможливлення використання кожної з вразливостей. На основі ДСТУ ISO/IEC TS 27008:2019 – методів тестування на проникнення та методик компанії Cymulate було створено алгоритм обрання рішень під час тестування ресурсів типового МКП на проникнення. Були приведені та описані етапи пентестінгу, та принцип роботи алгоритму як поетапної інструкції для тестування на проникнення типового МКП.

3 ЕКОНОМІЧНИЙ РОЗДІЛ

Мета цього розділу – обґрунтування економічної доцільності впровадження алгоритму оптимального шляху тестування на проникнення для веб – ресурсів МКП. Для вирішення цього завдання необхідно здійснити розрахунок капітальних витрат; експлуатаційних витрат, які визначають величину витрат на обслуговування впровадженої системи на рік; величину можливого збитку; величину економічного ефекту; показники економічної ефективності.

3.1 Розрахунок (фіксованих) капітальних витрат

Капітальні (фіксовані) витрати є сумою витрат щодо створення і придбання основних фондів і нематеріальних активів, що підлягають амортизації.

Капітальні (фіксовані) витрати на проектування та впровадження проектного варіанта системи інформаційної безпеки складають:

$$К = К_{пр} + К_{зпз} + К_{пз} + К_{аз} + К_{навч} + К_{н} \quad (3.1)$$

де $К_{пр}$ – вартість розробки проекту інформаційної безпеки та залучення для цього зовнішніх консультантів;

$К_{зпз}$ – вартість закупівель ліцензійного основного й додаткового програмного забезпечення (ПЗ);

$К_{пз}$ – вартість створення основного й додаткового програмного забезпечення;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів;

$К_{навч}$ – витрати на навчання технічних фахівців і обслуговуючого персоналу;

$К_н$ – витрати на встановлення обладнання та налагодження системи інформаційної безпеки.

3.1.1. Визначення витрат на підвищення рівня інформаційної безпеки підприємства шляхом виконання алгоритму тестування на проникнення при забезпеченні аудиту безпеки веб – ресурсів МКП.

3.1.1.1 Визначення трудомісткості розробки методики виявлення вразливостей при забезпеченні аудиту безпеки веб – ресурсів МКП.

Трудомісткість розробки визначається тривалістю кожної робочої операції:

$$t = t_{тз} + t_в + t_а + t_р + t_д, \text{ годин, } (3.2)$$

де $t_{тз}$ – тривалість складання технічного завдання на використання алгоритму оптимального тестування веб – ресурсів на проникнення при забезпеченні аудиту безпеки веб – ресурсів МКП, $t_{тз}=48$;

$t_в$ – тривалість аналізу існуючих інформаційних джерел організації, необхідних для роботи алгоритму, вивчення ТЗ, літературних джерел за темою тощо, $t_в=24$;

$t_а$ – тривалість аналізу існуючих вразливостей безпеки веб - ресурсів, $t_а=100$;

$t_р$ – тривалість повного завершення тестування на проникнення за допомогою розробленого алгоритму методики виявлення вразливостей при забезпеченні аудиту безпеки веб – ресурсів МКП, $t_м=180$;

$t_д$ – тривалість підготовки технічної документації, $t_д=40$.

Отже,

$$t = 48+24+100+180+40 = 392 \text{ годин. } (3.3)$$

3.1.1.2. Розрахунок витрат на підвищення рівня безпеки веб – ресурсів МКП шляхом впровадження алгоритму тестування на вразливості.

Витрати на розробку заходів безпеки Кпз складаються з витрат на заробітну плату спеціаліста з аудиту безпеки *Ззп* і вартості витрат машинного часу *Змч*:

$$К_{пз} = З_{зп} + З_{мч} = 199920 + 7480,56 = 207400,56 \text{ грн. (3.4)}$$

$$З_{зп} = t\, З_{пр} = 392*500 = 199920 \text{ грн. (3.5)}$$

де t – загальна тривалість операцій, годин;

Зпр – середньогодинна заробітна плата аудитора (тестування на проникнення) з нарахуваннями, грн/годину.

Вартість машинного часу для налагодження програми на ПК визначається за формулою:

$$З_{мч} = t \cdot С_{мч} = 392 \cdot 13,98 = 5480,16 \text{ грн. (3.6)}$$

де t – трудомісткість операцій із побудови ефективної системи доступу персоналу, годин;

$С_{мч}$ – вартість 1 години машинного часу ПК, грн./година.

Вартість 1 години машинного часу ПК визначається за формулою:

$$Смч = P \cdot t_{нал} \cdot Се + \frac{Ф_{зал} \cdot На}{F_р} + \frac{К_{лпз} \cdot Н_{апз}}{F_р}, \text{ грн; (3.7)}$$

$$Смч = 0,7 \cdot 8 \cdot 1,55 + \frac{15000 \cdot 0,5}{1920} + \frac{7000 \cdot 0,4}{1920} = 13,98 \text{ грн. (3.8)}$$

Розробка методики оцінки захищеності веб-додатків містить багато перевірок та включає в себе тестування веб-додатків на велику кількість вразливостей, а також включає в себе різні методи їх пошуку. При цьому планується придбання програмного забезпечення Rapid 7, вартістю 115000 грн.,

Використання опції “Пошук вразливостей веб-ресурсів” можливо здійснити за допомогою використання придбаного П.О; в якому буде проводитись тестування.

Для застосування методики із виявлення вразливостей при забезпеченні аудиту безпеки веб-ресурсів МКП необхідно здійснити навчання відповідних спеціалістів вартістю 30000 грн. ($К_н$=30000 грн.).

Таким чином, капітальні (фіксовані) витрати на підвищення рівня безпеки аудиту веб ресурсів МКП шляхом впровадження та введення в дію розробленого алгоритму тестування на проникнення складуть:

$$К = 207400,56 + 115000 + 30000 = 352400,56 \text{ грн.} \quad (3.9)$$

3.1.1 Розрахунок поточних витрат

Річні поточні витрати на впровадження аудиту безпеки веб - ресурсів МКП складають:

$$С = С_в + С_к + С_{ак}, \text{ грн.} \quad (3.10)$$

де $С_в$ - вартість відновлення й модернізації системи;

$С_к$ - витрати на керування аудитом в цілому;

$С_{ак}$ - витрати, викликані активністю користувачів веб - ресурсів.

При застосуванні алгоритму тестування для виявлення вразливостей при забезпеченні безпеки аудиту веб-веб ресурсів МКП, вартість модернізації системи складає 28000 грн. щорічно.

Витрати на керування аудитом безпеки веб-ресурсів підприємства ($С_к$) складають:

$$С_к = С_н + С_а + С_з + С_{ел} + С_о + С_{тос}, \text{ грн.} \quad (3.11)$$

Річний фонд амортизаційних відрахувань ($С_а$) визначається прямолінійним методом, виходячи з вартості активів та строку їх корисного використання. Вартість П.З. Rapid 7 складає 115000 грн.. Строк корисного використання – 2 роки. Таким чином,

$$С_а=115000/2=57500 \text{ грн.} \quad (3.12)$$

Відповідно до специфіки застосуванні алгоритму із виявлення вразливостей при тестуванні на проникнення для забезпечення аудиту безпеки веб – ресурсів МКП необхідно проходити періодичне навчання протягом року, сумарна вартість якого складає 30000 грн. на рік.

Річний фонд заробітної плати інженерно-технічного персоналу, що обслуговує систему аудиту безпеки веб - ресурсів ($С_з$), складає:

$$С_з = З_{осн} + З_{дод}, \text{ грн.} \quad (3.13)$$

Основна заробітна плата визначається, виходячи з місячного посадового окладу, а додаткова заробітна плата – в розмірі 8-10% від основної заробітної плати.

Основна заробітна плата одного спеціаліста з інформаційної безпеки на місяць складає 30000 грн. Додаткова заробітна плата – 8% від основної заробітної плати. Отже,

$$С_з = 30000*12 + 30000*12*0,09= 392400 \text{ грн.} \quad (3.14)$$

Ставка ЄСВ для всіх категорій платників з 01.09.2020 р. складає 22%.

$$С_{єв} = 392400*0,22 =86328 \text{ грн.} \quad (3.15)$$

Вартість електроенергії, що споживається апаратурою з П.З для підтримки в робочому стані аудиту безпеки веб – ресурсів МКП протягом року ($С_{ел}$), визначається за формулою:

$$С_{ел} = Р \cdot F_р \cdot Ц_е, \text{ грн.}, (3.16)$$

де Р – встановлена потужність апаратури аудиту веб - ресурсів інформаційної безпеки, (Р=1 кВт);

$F_р$ – річний фонд робочого часу безперервного аудиту веб - ресурсів ($F_р$ = 1920 год.);

$Ц_е$ – тариф на електроенергію, ($Ц_е$ = 1,55 грн./кВт за годину).

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року:

$$С_{ел} = 1*1920*1,55 = 2\ 976 \text{ грн. } (3.17)$$

Витрати на технічне й організаційне адміністрування системи аудиту безпеки веб - ресурсів визначаються у відсотках від вартості капітальних витрат - 2% (Стос = 352400,56*0,02=6508 грн).

Витрати на керування системою інформаційної безпеки ($С_к$) визначаються:

$$С_к = 57500+392400+86328+6508+2\ 976=$$
$$= 545\ 712 \text{ грн. } (3.18)$$

Таким чином, річні поточні витрати на функціонування системи інформаційної безпеки складають:

$$С = 28000 + 545\ 712 = 573\ 712 \text{ грн. } (3.19)$$

3.2 Оцінка можливого збитку

3.2.1 Оцінка величини збитку

Для розрахунку вартості такого збитку можна застосувати наступну спрощену модель оцінки.

Необхідні *вихідні дані* для розрахунку:

$t_п$ – час простою вузла або сегмента корпоративної мережі внаслідок атаки, 10 годин;

$t_в$ – час відновлення після атаки персоналом, що обслуговує корпоративну мережу, 8 години;

$t_{ви}$ – час повторного введення загубленої інформації співробітниками атакованого вузла або сегмента корпоративної мережі, 11 годин;

$З_о$ – заробітна плата обслуговуючого персоналу (адміністраторів та ін.), 25000 грн./міс.;

$З_с$ – заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі, 25000 грн./міс.;

$Ч_о$ – чисельність обслуговуючого персоналу (адміністраторів та ін.), 5 осіб.;

$Ч_с$ – чисельність співробітників атакованого вузла або сегмента корпоративної мережі, 25 осіб.;

О – обсяг прибутку атакованого вузла або сегмента корпоративної мережі, 3 мільйони 650 тис. грн. у рік;

$П_{зч}$ – вартість заміни встаткування або запасних частин, грн;

I – число атакованих сегментів корпоративної мережі, 2;

N – середнє число атак на рік, 40.

Упущена вигода від простою атакованого сегмента корпоративної мережі становить:

$$U = П_п + П_в + V, \quad (3.20)$$

де $П_п$ – оплачувані втрати робочого часу та простої співробітників атакованого вузла або сегмента корпоративної мережі, грн;

$П_в$ – вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн;

V – втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності співробітників атакованого вузла або сегмента корпоративної мережі являють собою втрати їхньої заробітної плати (оплата непродуктивної праці) за час простою внаслідок атаки:

$$П_П = \frac{\sum Зс}{F} \cdot t_n = \frac{25000 \cdot 40}{176} \cdot 10 = 56818 \quad \text{грн, (3.21)}$$

де F – місячний фонд робочого часу (при 40-а годинному робочому тижні становить 176 ч).

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі включають кілька складових:

$$П_в = П_{ви} + П_{пв} + П_{зч}, \quad (3.22)$$

де $П_{ви}$ – витрати на повторне уведення інформації, грн.;

$П_{пв}$ – витрати на відновлення вузла або сегмента корпоративної мережі, грн;

$П_{зч}$ – вартість заміни устаткування або запасних частин, грн.

Витрати на повторне введення інформації $П_{ви}$ розраховуються виходячи з розміру заробітної плати співробітників атакованого вузла або сегмента корпоративної мережі $З_с$, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ви}$:

$$П_{ви} = \frac{\sum Зс}{F} \cdot t_{ви} = \frac{25000 \cdot 40}{176} \cdot 11 = 62500 \text{ грн. (3.23)}$$

Витрати на відновлення сегмента корпоративної мережі $П_{пв}$ визначаються часом відновлення після атаки $t_в$ і розміром середньогодинної заробітної плати обслуговуючого персоналу (адміністраторів):

$$П_{пв} = \frac{\sum Зо}{F} \cdot t_в = \frac{25000 \cdot 8}{176} \cdot 11 = 12500 \text{ грн. (3.24)}$$

Таким чином, витрати на відновлення працездатності вузла або сегмента корпоративної мережі складають:

$$П_в = 62500+12500 = 75000 \text{ грн. (3.25)}$$

Втрати від зниження очікуваного обсягу прибутків за час простою атакованого вузла або сегмента корпоративної мережі визначаються виходячи із средньогодинного обсягу прибутку і сумарного часу простою сегмента корпоративної мережі:

$$V = \frac{O}{Fг} \cdot (t_П + t_В + t_{ВИ}) \text{ (3.26)}$$

$$V = \frac{3650000}{2080} \cdot (10+8+11) = 50866 \text{ грн. (3.27)}$$

де $F_г$ – річний фонд часу роботи філії (52 робочих тижні, 5-ти денний робочий тиждень, 8-ми годинний робочий день) становить близько 2080 год.

$$U = 56818 + 75000 + 50866 = 182684 \text{ грн. (3.28)}$$

Таким чином, загальний збиток від атаки на сегмент корпоративної мережі організації складе:

$$B = \sum_i \sum_n U = \sum_2 \sum_{40} 182684 = 14\ 614\ 720 \text{ грн.} \quad (3.29)$$

3.2.2 Загальний ефект від впровадження системи інформаційної безпеки

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної безпеки і становить:

$$E = B \cdot R - C \text{ грн.,} \quad (3.30)$$

де B – загальний збиток від атаки у разі перехоплення інформації, тис. грн.;

R – вірогідність успішної реалізації атаки на сегмент мережі, частки одиниці (R=0,2);

С – щорічні витрати на експлуатацію системи аудиту безпеки веб - ресурсів, тис. грн.

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної безпеки:

$$E = 14\ 614\ 720 * 0{,}2 - 573\ 712 = 2\ 349\ 232 \text{ грн.} \quad (3.31)$$

3.3 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки

Коефіцієнт повернення інвестицій ROSI показує, скільки гривень додаткового прибутку приносить одна гривня капітальних інвестицій на впровадження системи інформаційної безпеки:

$$ROSI = \frac{E}{K}, \quad \text{частки одиниці,} \quad (3.32)$$

де E – загальний ефект від впровадження системи інформаційної безпеки грн.;

К – капітальні інвестиції за варіантами, що забезпечили цей ефект, грн.

Коефіцієнт повернення інвестицій ROSI:

$$ROSI = \frac{2\,349\,232}{573\,712} = 4{,}0, \quad \text{частки одиниці, (3.33)}$$

Проект визнається економічно доцільним, якщо розрахункове значення коефіцієнта повернення інвестицій перевищує величину річної депозитної ставки з урахуванням інфляції:

$$ROSI > (N_{деп} - N_{інф})/100), \quad (3.34)$$

де $N_{деп}$ – річна депозитна ставка, (5,5 %);

$N_{інф}$ – річний рівень інфляції, (5%).

Розрахункове значення коефіцієнта повернення інвестицій:

$$4 > (5{,}5 - 5)/100 = 4 > 0{,}005. \quad (3.35)$$

Термін окупності капітальних інвестицій T_o показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки. Відповідно термін окупності розробки методики із виявлення вразливостей при забезпеченні безпеки веб-додатків комерційного банку:

$$T_O = \frac{K}{E} = \frac{1}{ROSI} = \frac{1}{4} = 0{,}25, \quad \text{років (3 місяці). (3.36)}$$

Висновок

Згідно з наведеними розрахунками можна дійти висновку, що використання алгоритму тестування на проникнення для аудиту безпеки веб-ресурсів МКП є економічно доцільним. Такі висновки зроблені, виходячи з коефіцієнту повернення інвестицій ROSI, який складає 4 (ROSI=4) та означає, що на 1 гривню капітальних витрат приходиться 4 грн. економічного ефекту. Період окупності при цьому складе 3 місяці. Капітальні інвестиції плануються на рівні 352400,56 грн., а експлуатаційні витрати в обсязі 573 712 грн. на рік.

ВИСНОВКИ

Аналіз статистичних даних о безпеці та аудиту веб-додатків свідчать про велику кількість вразливостей та величезну потребу в активному аудиті безпеки веб – ресурсів МКП. Існуючі рекомендації потребують додаткових коректувань для поліпшення ефективності мір протидії вразливостям. Існуючі алгоритми тестування на перший погляд вирішують всі проблеми безпеки, але коштують забагато, або охоплюють не всі оптимальні точки тестування.

Саме тому в роботі були вирішити наступні задачі:

– Проаналізовані вразливості веб-ресурсів згідно з нормативними документами та методиками тестування на проникнення;

– Проаналізовані загрози веб – ресурсів та запропоновано методи протидії;

– Запропоновано комбінований алгоритм тестування на проникнення за допомогою існуючого “сліпого методу” тестування за ДСТУ ISO/IEC 27008:2019 та комбінованої методології Cymulate;

– Розраховано економічну ефективність та доцільність впровадження алгоритму та в дію шляхом закупівлі обладнання, та залучення команди аудиторів, або аудитора (або створення та навчання робітника самої компанії), яка має таку потребу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Господарський кодекс України (ГКУ), редакція від 17.06.2018 [Електронний ресурс] - Режим доступу до ресурсу: https://urist-ua.net/.

2. Як запобігти кібер-атакам – дослідження 2020 року [Електронний ресурс] - Режим доступу до ресурсу: https://purplesec.us/prevent-cyber-attacks/.

3. Топ 10 найрозповсюджених вразливостей по версії OWASP [Електронний ресурс] - Режим доступу до ресурсу: https://owasp.org/www-project-top-ten/

4. Найрозповсюджені веб – вразливості [Електронний ресурс] - Режим доступу до ресурсу: https://capec.mitre.org/about/

5. Топ 10 найрозповсюджених вразливостей по версії OWASP – вектори дії [Електронний ресурс] - Режим доступу до ресурсу: https://www.veracode.com/security/owasp-top-10

6. Переваги та недоліки ручного та автоматизованого тестування на проникнення [Електронний ресурс] - Режим доступу до ресурсу: https://l.cymulate.com/hubfs/Whitepaper/Cymulate.

7. ДСТУ ISO/IEC 27008:2019 [Електронний ресурс] // ДСТУ. - 2019. - Режим доступу до ресурсу: http://online.budstandart.com/ru/catalog/doc-page.html?id_doc=85798.

8. Стандарт OSTMM [Електронний ресурс]- Режим доступу до ресурсу: https://www.isecom.org/research.html.

9.Стандарт N SP 800-115 [Електронний ресурс] - Режим доступу до ресурсу: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf.

10. Проекти OWASP [Електронний ресурс] - Режим доступу до ресурсу: https://owasp.org/projects/.

11. Стандарт PTES [Електронний ресурс] - Режим доступу до ресурсу: http://www.pentest-standard.org/index.php/Main_Page.

12. Методології тестування OSTMM [Електронний ресурс] - Режим доступу до ресурсу: https://www.isecom.org/OSSTMM.3.pdf.

13. Основні методики пентестінгу за версією NIST [Електронний ресурс] - Режим доступу до ресурсу: https://www.nist.gov/publications/technical-guide-information-security-testing-and-assessment.

14. Методології ISAF [Електронний ресурс] - Режим доступу до ресурсу: https://ro.ecu.edu.au/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1181&context=ism.

15. bWAPP як середовище для тестування на проникнення [Електронний ресурс] - Режим доступу до ресурсу: https://kali.tools/?p=2330.

16. BurpSuite – платформа для аудиту безпеки веб - ресурсів[Електронний ресурс] - Режим доступу до ресурсу: https://habr.com/ru/post/328382.

17. Загальні етапи тестування на проникнення [Електронний ресурс] - Режим доступу до ресурсу:https://pentaroot.com/five-phases-of-penetration-testing/.

ДОДАТОК А. ВІДОМІСТЬ МАТЕРІАЛІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ

№	Формат	Найменування	Кількість листів	Примітки
1	А4	Реферат	3	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	1	
4	А4	Вступ	1	
5	А4	Стан питання. Постановка задачі	19	
6	А4	Спеціальна частина	26	
7	А4	Економічна частина	12	
8	А4	Висновки	1	
9	А4	Перелік посилань	2	
10	А4	Додаток А	1	
11	А4	Додаток Б	1	
12	А4	Додаток В	1	
13	А4	Додаток Г	1	
14	А4	Додаток Д	1	

ДОДАТОК Б. ПЕРЕЛІК МАТЕРІАЛІВ НА ОПТИЧНОМУ НОСІЇ

MatsaitisDI125m19_2.docx

MatsaitisDI125m19_2.pptx

ДОДАТОК В. ВІДГУКИ КЕРІВНИКІВ РОЗДІЛІВ

Відгук керівника економічного розділу:

Керівник розділу _____ _____
(підпис) (ініціали, прізвище)

ДОДАТОК Д. ВІДГУК КЕРІВНИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ

Метою кваліфікаційної роботи є поліпшення методу тестування веб – ресурсів малих комерційних підприємств на проникнення

Для досягнення поставленої мети у кваліфікаційній роботі вирішуються наступні задачі: проведення аналізу релевантних загроз веб ресурам малих комерційних підприємств. Розглянуто статистику експертних агентств та типи тестування на проникнення (як автоматизованого, так і ручного), розглянуто їх переваги та недоліки. Розроблено методику проведення теста з використанням компілювання існуючих методів та методологій в алгоритм прийняття рішень тестування на проникнення.

Перевагою кваліфікаційної роботи є запропоновані методи протидії вразливостям веб – ресурсів (згідно до статистик OWASP TOP 10) та розроблений алгоритм прийняття рішень під час тестування веб – ресурсу підприємства на проникнення, а саме – вибір методики та комбінація з модифікованою методологією Cymulate.

Серед недоліків слід відмітити таке: оформлення пояснювальної записки до кваліфікаційної роботи виконано з деяким відхиленням від стандартів, недостатньо повно доведено ефективність запропонованого рішення.

Під час виконання кваліфікаційної роботи Мацайтіс Д.І. проявив себе фахівцем, здатним самостійно вирішувати поставлені задачі.

В цілому кваліфікаційна робота виконана у відповідності до вимог, що ставляться до кваліфікаційної роботи і заслуговує оцінки 70 “задовільно”, а студент Мацайтіс Дмитро Ігорович присвоєння йому кваліфікації магістра за спеціальністю 125 Кібербезпека, освітньо-професійна програма “кібербезпека”.

Рівень запозичень у кваліфікаційній роботі не перевищує вимог “Положення про систему виявлення та запобігання плагіату”.

Керівник кваліфікаційної роботи,
доктор фізико – математичних наук,
професор Т.С. Кагадій

Керівник спеціального розділу,
старший викладач Д.С. Тимофєєв

ДОДАТОК Г. ОПИС МЕТОДИКИ ТЕСТУВАННЯ CYMULATE

Методика CYMULATE має в собі 3 основні рішення для тестування на проникнення, а саме:

1. Тестування на загальні вразливості;
2. Тестування на підозрілі та вразливі дані в трафіку;
3. Тестування рівня “Чорної скриньки” за декількома векторами;

1. Тестування на загальні вразливості - оцінка вразливості програмного забезпечення до різних атак та включає в себе :

- Перевантаження системи (відмова в обслуговувані клієнтів);
- Цілеспрямоване введення помилок з ціллю проникнути в систему в ході відновлення;
- Перегляд несекретних даних з метою знайти ключ для входу в систему.

2. Тестування на підозрілі та вразливі дані в трафіку виконується за допомогою сканерів вразливостей, та включає в себе 4 етапи

- Виявляє активні IP-адреси, відкриті порти, запущену операційну систему і додатки.
- Складає звіт про безпеку (необов'язковий крок).
- Намагається визначити рівень можливого втручання в операційну систему або додатки (може спричинити збій).
- На заключному етапі сканер може скористатися вразливістю, викликавши збій операційної системи або програми.

3. Тестування рівня “Чорної скриньки” за декількома векторами дає змогу аудиторам тестувати на вразливості як інформаційний периметр підприємства, так і внутрішню мережу.