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

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня бакалавра

студента *Сустрєтова Іллі Олексійовича*

академічної групи *125-19-2*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Аналіз вразливостей корпоративних інформаційних систем в ТОВ «МОДЕРА РОЗВИТОК УКРАЇНА» з використанням автоматизованих інструментів*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	ст. викл. Святошенко В.О.	85	добре	
розділів:				
спеціальний	ст. викл. Святошенко В.О.			
економічний	к.е.н., доц. Пілова Д.П.	90		
Рецензент	к.т.н. Шедловський І.В.	85	добре	
Нормоконтролер	проф. Гусєв О.Ю.	90	відм.	

Дніпро

2023

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

«_____»_____2023 року

ЗАВДАННЯ
на кваліфікаційну роботу ступеня бакалавра

студенту *Сустрєтову Іллі Олексійовичу* **академічної групи** *125-19-2*
(прізвище ім'я по-батькові) (шифр)

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

на тему ***Аналіз вразливостей корпоративних інформаційних систем в ТОВ «МОДЕРА РОЗВИТОК УКРАЇНА» з використанням автоматизованих інструментів***

затверджену наказом ректора НТУ «Дніпровська політехніка» від 16.05.2023 № 350-с

Розділ	Зміст	Термін виконання
Розділ 1	Проаналізувати інформаційну систему та методології для тестування на проникнення	15.03.23-28.04.23
Розділ 2	Провести тестування системи на вразливість та запропонувати методи захисту	29.04.23 – 01.06.23
Розділ 3	Розрахунки капітальних витрат, витрат на експлуатацію системи безпеки та термін окупності інвестицій застосування запропонованих методів	02.06.23 – 15.06.23

Завдання видано _____ **Святошенко В.О.**
(підпис керівника) (прізвище, ініціали)

Дата видачі: _____

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____ **Сустрєтов І.О.**
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 74 с., 20 рис., 15 табл., 6 додатків, 21 джерело.

Об’єкт розробки: Комплекс заходів захисту від атак.

Предмет розробки: Удосконалений комплекс заходів захисту від атак на об’єкті інформаційної діяльності.

Мета кваліфікаційної роботи: Підвищення рівня захисту інформації в інформаційній системі ТОВ «МОДЕРА РОЗВИТОК УКРАЇНА».

Методи розробки: аналіз, обстеження, тестування, опис та розрахунки.

Робота містить 3 розділи, висновки і додатки.

У першому розділі проаналізована інформаційна система підприємства, роботу співробітників та їх обов’язки, теоретичні вразливості проекту і нормативно правових документів, був сформований висновок щодо аналізу методологій для тестування на проникнення.

У спеціальній частині роботи описано вимоги щодо тестування, проведено додатковий аналіз підприємства, розглянуто модель порушника, запропоновано комплекс заходів щодо усунення виявлених загроз та проведено висновки щодо виконаної роботи.

У економічному розділі виконані розрахунки на доцільність усунення виявлених загроз, капітальних витрат, витрат на експлуатацію системи безпеки та термін окупності інвестицій.

Практичне значення роботи полягає у можливості впровадження комплексів заходів захисту в різні інформаційні системи в підприємствах.

ІНФОРМАЦІЙНА БЕЗПЕКА, ВРАЗЛИВІСТЬ, ОЦІНКА ЗАХИЩЕНОСТІ, ПЕНТЕСТІНГ, XSS/CSRF, БЕЗПЕКА ІНФОРМАЦІЙНО І КОМУНІКАЦІЙНИХ СИСТЕМ

ABSTRACT

Explanatory note: 74 p., 20 figures, 15 tables, 6 appendices, 21 sources.

Object of development: A set of measures to protect against attacks.

Subject of development: An improved set of measures to protect against attacks on the object of information activity.

Purpose of the qualification work: Increasing the level of information security in the information system of MODERA DEVELOPMENT UKRAINE LLC.

Development methods: analysis, survey, testing, description and calculations.

The work contains 3 chapters, conclusions and appendices.

The first section analyzes the company's information system, the work of employees and their responsibilities, theoretical vulnerabilities of the project and regulatory documents, and concludes on the analysis of penetration testing methodologies.

The special part of the paper describes the requirements for testing, conducts an additional analysis of the enterprise, considers the model of the intruder, proposes a set of measures to eliminate the identified threats, and draws conclusions on the work performed.

In the economic section, we calculate the feasibility of eliminating the identified threats, capital expenditures, costs of operating the security system, and the payback period of investments.

The practical significance of the work lies in the possibility of implementing security measures in various information systems in enterprises.

INFORMATION SECURITY, VULNERABILITY, SECURITY ASSESSMENT, PENTESTING, XSS/CSRF, SECURITY OF INFORMATION AND COMMUNICATION SYSTEMS

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АС – автоматизована система;

ІС – інформаційна система;

ІТС – інформаційно-телекомунікаційна система;

НСД – несанкціонований доступ;

ОС – операційна система;

ТОВ – товариство з обмеженою відповідальністю;

API – Application Programming Interface;

CORS – Cross-Origin Resource Sharing;

CRM – управління взаємовідносинами з клієнтами;

CSRF – Cross-Site Request Forgery;

DDoS – Distributed Denial of Service;

DOM – об'єктна модель документа;

DSS – Digital Signature Standard;

HTML – HyperText Markup Language;

HTTP – HyperText Transfer Protocol;

OAuth – Open Authorization;

ORM – Object-Relational Mapping;

OSSTMM – Open-Source Security Testing Methodology Manual;

OWASP – Open Web Application Security Project;

SQL – Structured Query Language;

TCP – Transmission Control Protocol;

XSS – Cross-Site Scripting;

ZAP – Zed Attack Proxy;

ЗМІСТ

ВСТУП

Сьогодні інформаційна безпека є гарячою темою у всіх новинах та Інтернеті. Майже щодня з'являються новини про псування веб-сторінок, витоки даних мільйонів облікових записів користувачів та паролів або номерів кредитних карток з веб-сайтів, а також про крадіжки особистих даних у соціальних мережах.

Такі терміни, як кібератака, кіберзлочин, хакер і навіть кібервійна стають реальністю 21-ого століття, частиною повсякденного лексикону в засобах масової інформації. Усе це, а також реальна потреба в захисті конфіденційних даних і репутації компаній, змусило їх бути більш уважними до питань інформаційної безпеки і своєї репутації та змусили організації краще усвідомити необхідність знати, де їхні системи є вразливими. Особливо це стосується тих, які доступні для всього світу через Інтернет, як вони можуть бути атаковані, і які будуть наслідки, з точки зору втрати інформації або компрометації системи, якщо атака буде успішною, і що більш важливо, як виправити ці вразливості та мінімізувати ризик – головні питання для них. Вирішенням задач по виявленню вразливостей і дослідженню їх впливу на організацію - це те, для чого потрібне тестування на проникнення.

Тест на проникнення - це атака або атаки, котра здійснена кваліфікованим фахівцем з безпеки, який використовує ті ж методи та інструменти, що й справжні зловмисники, щоб виявити всі можливі слабкі місця в системах організації. Ці слабкі місця експлуатуються і вимірюється їхній вплив.

Коли тест завершено, тестувальник на проникнення повідомляє про всі свої висновки і розповідає, як їх можна виправити, щоб запобігти майбутнім пошкодженням.

В кваліфікаційній роботі розглянуто підприємство «МОДЕРА РОЗВИТОК УКРАЇНА», його інформаційну структуру та досліджено різні типи вразливостей, методологій, котрі можуть використати зловмисники.

1 СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність питання

Кібербезпека - одна з найбільш швидкозростаючих галузей в індустрії інформаційних технологій. Щодня фахівці з безпеки виявляють нові та нові загрози, а активи організацій стають об'єктами атак з боку зловмисників.

Через ці загрози в цифровому світі в багатьох організаціях з'являються нові професії для людей, які можуть допомогти захистити і зберегти їхні активи.

У всьому світі існує величезний попит на фахівців з кібербезпеки, оскільки багато організацій починають розуміти потребу в кваліфікованих фахівцях, які допоможуть їм захистити свої активи. Одним з найцінніших активів будь-якої організації є дані.

Зловмисники, такі як хакери, вдосконалюють свої плани, а хакерство перетворилося на бізнес. Зловмисники використовують сучасні та витончені атаки і погрози, щоб скомпрометувати системи і мережі своїх жертв, викрасти їхні дані, використовуючи різні методи для проникнення, щоб обійти систему захисту, та продають викрадені дані в різних форумах та біржах.

Раніше хакери користувались тільки ручними інструментами, але сьогодні у їхньому арсеналі багато новітніх, автоматизованих інструментів, такі як програми-вимагачі, які є криптографічним шкідливим програмним забезпеченням, призначеним для компрометації вразливих систем.

Після зараження системи вірусом-вимагачем, він зашифровує всі дані на локальних дисках, окрім операційної системи. Крім того, програми-вимагачі можуть також скомпрометувати будь-яке хмарне сховище, пов'язане з зараженої системи.

Наприклад, в системі користувача є Google Drive, Microsoft OneDrive або Dropbox, і дані постійно синхронізуються. Якщо система заражена, інфекція також може вплинути на дані в хмарному сховищі. Однак, деякі хмарні провайдери мають вбудований захист від таких типів загроз.

Програми-вимагачі шифрують дані і тримають їх у заручниках, показуючи вікно з вимогою оплати на робочому столі жертви з вимогою здійснити платіж для відновлення даних. У цей час зловмисник викрадає дані та продає їх.

Наразі найбільш популярний суб'єкт загроз - хакер. Однак, існують й інші типи суб'єктів загроз, які беруть участь у кібератаках.

Найпопулярніші суб'єкти загроз:

- Сценарій дитини - сценарій дитини є поширеним типом суб'єкта загрози, це людина, яка не розуміє технічних деталей кібербезпеки, щоб самостійно здійснити кібератаку. Однак, зазвичай слідує інструкціям або навчальним посібникам справжніх хакерів, щоб здійснити свої власні атаки на систему або мережу. На перший погляд цей суб’єкт нешкідливий, оскільки людина не має необхідних знань і навичок, але вони можуть завдати такої ж шкоди, як і справжній хакер, якщо будуть дотримуватися інструкціям зловмисників в інтернеті. Ці типи хакерів можуть використовувати інструменти не знаючи, як вони працюють, тим самим завдаючи більшої шкоди.
- Хактивіст - у багатьох країнах світу існує багато соціальних і політичних програм, а також багато людей і груп, які підтримують або не підтримують ці програми. Часто можна зустріти протестувальників, які організовують мітинги, марші або навіть здійснюють незаконні дії, такі як псування громадського майна. Існує тип суб'єкта загрози, який використовує свої хакерські навички для здійснення зловмисної діяльності на підтримку політичного чи соціального порядку денного. Таких людей зазвичай називають хактивістами. Хоча деякі хактивісти використовують свої хакерські навички з добрими намірами, але це не звільняє від притягнення до юридичної відповідальності.
- Інсайдер - багато зловмисників усвідомили, що зламати організацію через інтернет складніше, а простіше зробити це зсередини, через внутрішню мережу об'єкта. Деякі зловмисники створюють фальшиві особисті дані та автобіографію з наміром влаштуватися на роботу в організацію-мішень і стати її співробітником. Після того, як такий суб'єкт загрози стане співробітником, він

отримає доступ до внутрішньої мережі і зможе краще зрозуміти архітектуру мережі та вразливі місця в системі безпеки. Таким чином, цей тип загрози може впроваджувати мережеві імплантати в мережі та створювати бекдори для віддаленого доступу до критично важливих систем.

- Спонсоровано державою - Багато битв зараз ведуться в кіберпросторі. Це називається кібернетичною війною. Багато країн усвідомили необхідність створення засобів захисту своїх громадян і національних активів від хакерів та інших країн зі зловмисними намірами. Тому уряд країни наймає державних хакерів, які відповідають за захист своєї країни від кібератак і загроз. Деякі країни використовують цей тип суб'єкта загрози для збору розвідувальної інформації про інші країни і навіть компрометують системи, які контролюють інфраструктуру комунальних послуг або інших критично важливих ресурсів, необхідні країні.

- Організована злочинність - У всьому світі часто з’являються новини про злочини синдикатів та організованих злочинних угруповань. В кібербезпеці також існують злочинні організації, що складаються з групи людей, що переслідують однакові цілі. Кожна людина в групі, як правило, є експертом або має кілька спеціальних навичок, наприклад, одна людина може відповідати за проведення обширної розвідки об'єкта, в той час як інша особа відповідає за розробку просунутої постійної загрози. У складі такої організованої злочинної групи зазвичай є особа, яка відповідає за фінансування групи, щоб забезпечити найкращі ресурси, які можна купити для забезпечення успіху атаки. Намір цього типу зловмисників, як правило, масштабні, наприклад, викрадення даних про жертву і продати їх з метою отримання фінансової вигоди.

- Чорний капелюх - хакер у чорному капелюсі - це суб'єкт загрози, який використовує свої навички зі зловмисною метою. Ці хакери можуть бути ким завгодно, і їхня причина здійснення злому системи або мережі може бути випадковою. Іноді вони можуть зламувати, щоб знищити репутацію своєї жертви, викрасти дані або навіть як особистий виклик, щоб довести свою правоту, заради розваги.

- Білі капелюхи - хакери в білих капелюхах. Цей тип хакерів, які використовують свої навички, щоб допомогти організаціям і людям захистити свої мережі та захистити свої активи від зловмисників. Етичні хакери та тестувальники на проникнення є прикладами хакерів білого капелюха, оскільки ці люди використовують свої навички, щоб допомагати іншим у позитивний та етичний спосіб.

- Сірий капелюх - хакер сірого капелюха - це людина, яка метафорично знаходиться між білим і чорним капелюхом. Це означає, що хакер у сірому капелюсі має хакерські вміння і може бути білим капелюхом вдень як професіонал з кібербезпеки і чорним капелюхом вночі, використовуючи свої навички зі зловмисними намірами.

З постійним розвитком нових технологій допитливі уми багатьох людей завжди знайдуть спосіб глибше зрозуміти технології, що лежать в основі тієї чи іншої системи. Це часто призводить до виявлення вразливостей у системі і, врешті-решт, дозволяє людині скористатися вразливістю.

1.2 Аналіз об’єкту дослідження «МОДЕРА РОЗВИТОК УКРАЇНА»

Підприємство «МОДЕРА РОЗВИТОК УКРАЇНА» займається продажом CRM-системи для автосалонів, головна мета якої є автоматизація процесу документообігу, покупки і продажу автомобіля, та розробка лендінг сторінок для компаній. Організація почала вести свою діяльність з 2013 року. За 10 років існування було створено багато проектів, що призвело до розширення спектру послуг, наразі приблизно 50-70 проектів, котрі активно ведуть свою діяльність. Адреса головного офісу: м. Дніпро, вулиця Січових Стрільців 4б.

Працівники є головним ресурсом для кожної компанії для реалізації певних проектів, завдань. Нижче наведено таблицю про загальний обсяг працівників.

Таблиця 1.1 – Штат працівників підприємства

Посада	Кількість працівників на посаді	Рівень кваліфікації
Директор	1	Високо-кваліфіковані робітники
Менеджер з продажу	3	Кваліфіковані робітники
Керівник проекту	5	Високо-кваліфіковані робітники
Служба підтримки	15	Кваліфіковані робітники
Бухгалтер	2	Кваліфіковані робітники
Програміст	35	Високо-кваліфіковані робітники
Системний адміністратор	2	Високо-кваліфіковані робітники
Провідний програміст	4	Високо-кваліфіковані робітники
Тестувальник	3	Високо-кваліфіковані робітники

Головний на підприємстві – директор. У його повноваження входить керування всіма проектами, надання відпусток, затвердження нових працівників, розподіл фінансовий частини (зарплатня), проведення співбесід з працівниками (кожен тиждень), контроль за виконанням планів та графіку.

Спеціаліст з питань кібербезпеки повинен проводити хмарний аналіз та аудит системи, контролювати доступи працівників, аналізувати трафік співробітників.

Системний адміністратор повинен стежити за стабільною роботою серверів та обладнанням в офісі, завчасним його обслуговуванням, оновленням систем та програмного забезпечення.

Бухгалтер відповідає за обіг документації стосовно фінансів, виставлення рахунку клієнтам для оплати, створювати щомісячні звіти для підприємства, перерахунок зарплат та податків.

Менеджер з продажу повинен шукати нових клієнтів, брати участь в демонстраціях системи та проводити зустрічі з можливими майбутніми клієнтами.

В обов'язки співробітника служби підтримки входить: збір первинної інформації від клієнтів, комунікація з клієнтами, при знаходженні помилок в системі, первинний аналіз помилок, створення звітів, комунікація з відділом розробки або якості продукту, стосовно помилок, створення перекладів для проекту, тестування нових функцій.

Програміст знаходиться під керуванням провідного програміста, в його обов'язки входить, аналіз та оцінка макетів задач, виконання доручень провідного програміста, виправлення помилок, допомога відділу підтримки клієнтів.

Провідний програміст повинен будувати інформаційну систему, керування та оновлення необхідних ресурсів для розробки, співбесіди з новими працівниками, аналіз знайдених помилок, розроблення автоматизованих систем для аналізу коду.

Тестувальник – знаходження вразливостей на етапі раннього тестування, впровадження нових методів тестування, створення автоматизованих тестів, керування процесом релізу, створення тестової документації, керування процесами тестування, створення архітектури для тестування, налаштування середовища для тестування, комунікація з відділом розробки та підтримки клієнтів.

Керівник проекту повинен слідкувати за розробкою функцій, створювати вимоги для технічних задач, створення плану розробки, участь в оцінці задач та спілкування з замовником.

Більш детальна структура одного з проектів підприємства на рисунку 1.1.

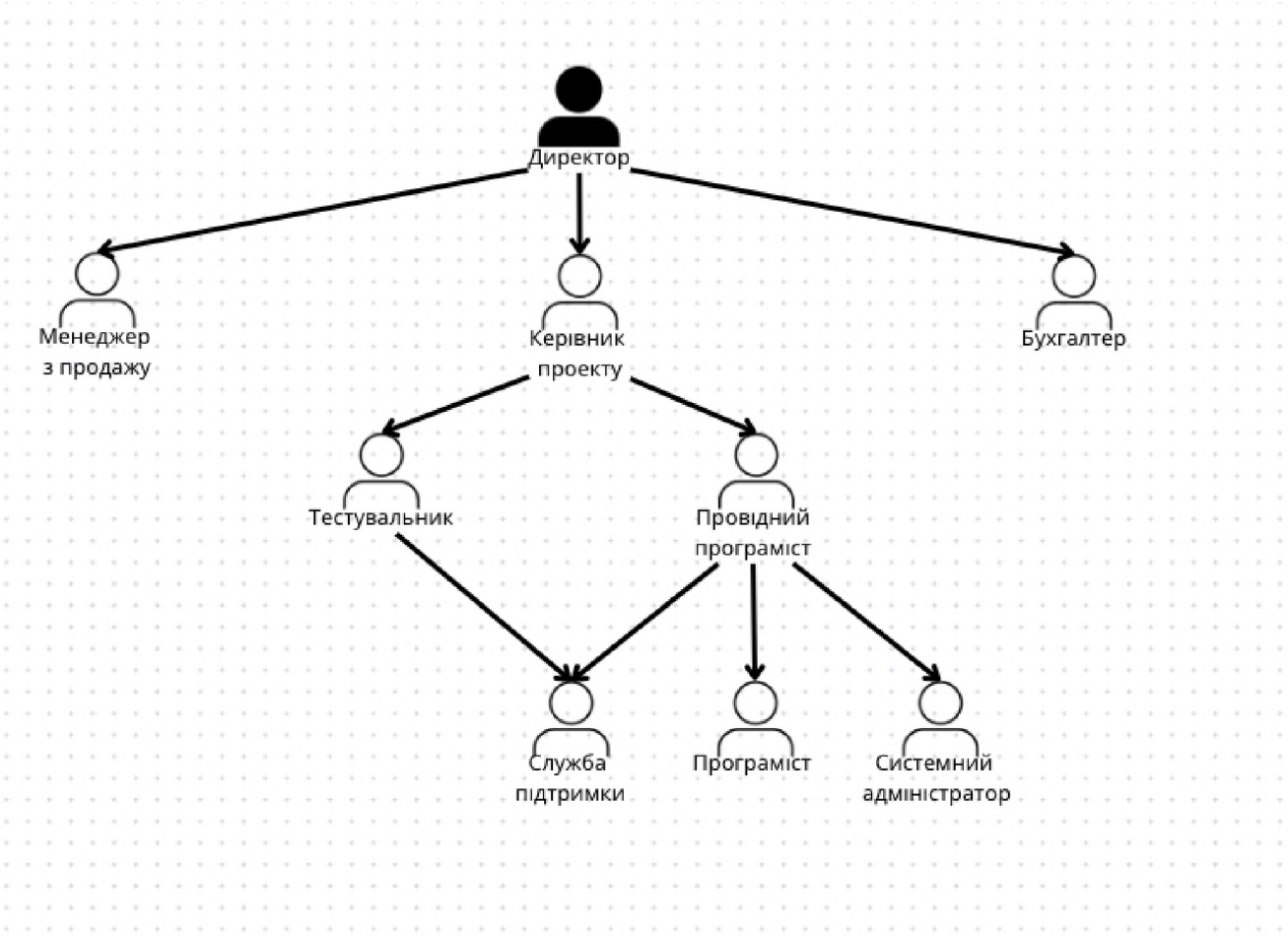


Рисунок 1.1 - Більш детальна структура на прикладні одного з проектів

Більшість проектів в підприємстві як мову програмування використовують: PHP, React, JavaScript. Для збереження даних використовують MySQL. Кожний спеціаліст з підприємства має свою корпоративну пошту, котра закінчується @modera.com.

Для комунікації команди використовується Slack, завдяки цьому кожен працівник може побачити поштову адресу колеги.

Доступом до репозиторіїв розробки слугує SSH ключ, котрий спеціаліст з безпеки заносить до білого листу. Інтегровані системи в проект: SendGrid, Ark integration, Print2Pdf, Redis, RabbitMq, Ari Register, Lursoft.

Як хмарні ресурси в підприємстві використовують AWS та Jenkins. Більш детально зображено на рисунку 1.2.

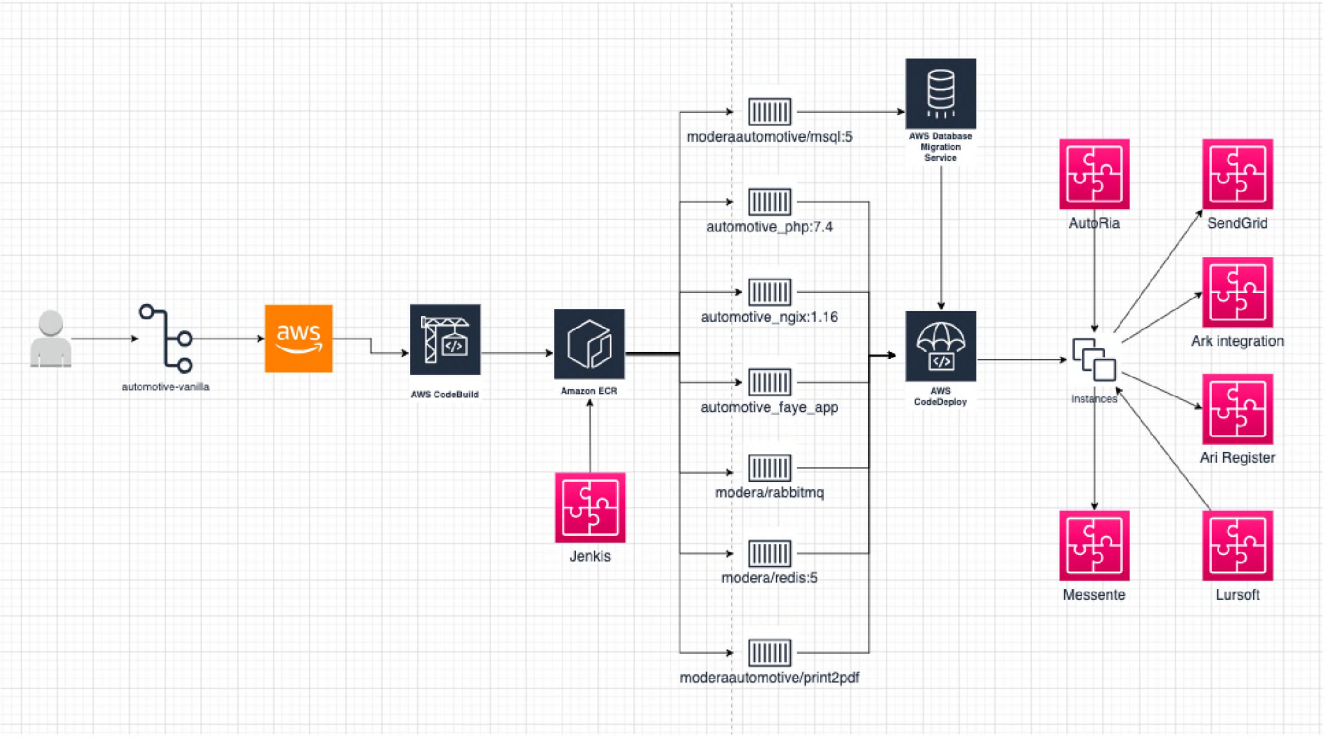


Рисунок 1.2 – Більш детальна структура сервісів

Всі технічні дані потрібні для аналізу можливих вразливостей та вибору більш ефективної методології для тестування на вразливість. Наприклад, дані, що проект має в собі MySQL дає зрозуміти, що можливо використати ін'єкції для отримання конфіденційних даних.

Інформаційна система має собою мережу "Passive Star", який налічує один комутатор, тобто ІТС можна кваліфікувати як багатомашинний багатокористувацький комплекс, який в свою чергу має доступ до мережі Інтернет, в якому циркулює інформація різних ступенів. Комплекси з такими характеристиками відносяться до класу АС 3.

Структурна схема ІТС підприємства представлена на рисунку 1.3.

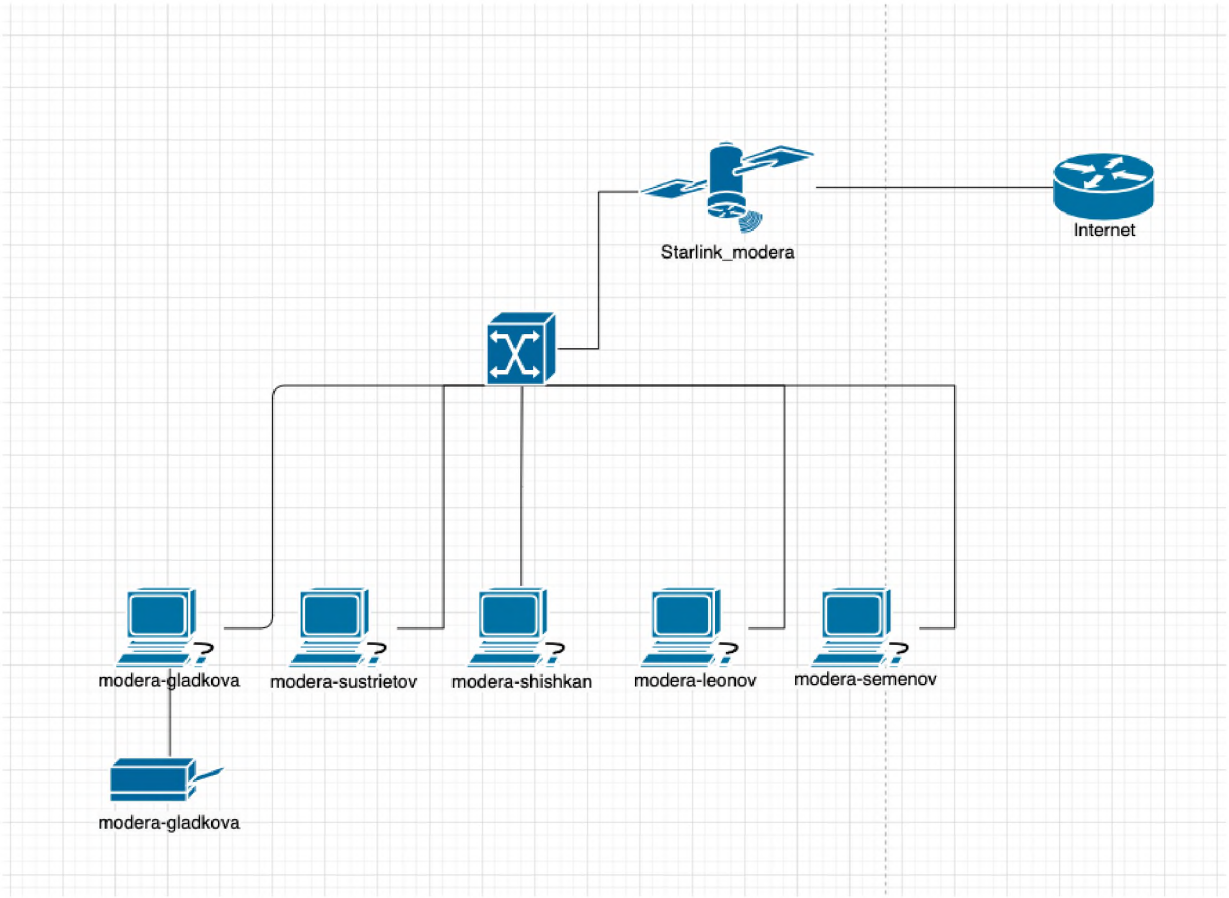


Рисунок 1.3 – Структурна схема ІТС

1.3 Аналіз нормативно правової бази підприємства

Нормативно-правова база з точки зору діяльності будь-якого підприємства передбачає наявність певного переліку нормативних документів.

Обсяг цих документів залежить від сфери діяльності підприємства, галузі розробки продукції компанії, нормативної поведінки в залежності від встановлених державних законів тощо.

Для ознайомлення зі сферою діяльності підприємства досліджено та розглянуто перелік документів, які складають нормативно-правову базу підприємства:

- Договір медичного страхування
- Документ, що свідчить про комерційну таємницю підприємства
- Журнал обліку технічної безпеки
- Журнал обліку пожежної безпеки

- Трудовий договір

1.4 Теоретичний аналіз кібератак і вразливостей CRM-системи підприємства «МОДЕРА РОЗВИТОК УКРАЇНА»

CRM-система стало важливим інструментом для бізнесу в управлінні взаємовідносинами з клієнтами та замовниками.

Однак зі збільшенням обсягу конфіденційних даних, що зберігаються в цифрових системах управління, для організацій стало вкрай важливо впроваджувати надійні заходи безпеки CRM, щоб захиститися від кіберзагроз і забезпечити конфіденційність даних.

Витік даних може мати серйозні наслідки, включаючи репутаційні збитки, юридичні та фінансові санкції, а також втрату довіри клієнтів. Тому захист програмного забезпечення CRM і забезпечення конфіденційності даних - це вже не варіант, а реальна необхідність, яку потрібно вирішувати на випередження.

На CRM-системи можуть бути здійснені різні типи кібератак, кожна з яких потенційно загрожує безпеці та конфіденційності конфіденційних даних.

Найпоширеніші типи кібератак на CRM-системи.

- Фішинг - це поширена кібератака, яка може вплинути на CRM-системи. Фішингові атаки можуть бути дуже ефективними, оскільки вони часто використовують людську поведінку, наприклад, цікавість або страх. Ці атаки передбачають використання електронних листів або повідомлень, які виглядають як повідомлення з законного джерела, наприклад, від надійного постачальника або ділового партнера, щоб обманом змусити користувачів надати конфіденційну інформацію, наприклад, облікові дані для входу або фінансові дані.
- Шкідливе програмне забезпечення, призначене для завдання шкоди комп'ютерним системам, викрадення даних і порушення нормальної роботи. Цей тип атак може бути руйнівним для CRM-систем, призводячи до втрати конфіденційних даних клієнтів, простою системи та шкоди репутації бізнесу. Шкідливе програмне забезпечення включає віруси, трояни та програми-вимагачі, які можуть бути використані для отримання несанкціонованого доступу до CRM-

систем, крадіжки даних або блокування доступу користувачів до їхніх облікових записів до моменту сплати викупу.

- SQL-ін'єкції. Ці атаки використовують уразливості в програмному забезпеченні, що використовується для створення та управління CRM-системами, дозволяючи зловмисникам отримати несанкціонований доступ до баз даних і викрасти або маніпулювати конфіденційними даними. Атака SQL-ін'єкцій відбувається, коли зловмисник надсилає шкідливий SQL-код як частину користувацького введення, наприклад, під час заповнення форми або пошукового запиту. База даних може виконати цей код, що потенційно дозволяє зловмиснику переглядати, змінювати або видаляти дані в CRM-системі.

- DDoS-атаки полягають у перевантаженні CRM-системи трафіком, що призводить до її збою або недоступності. Це може призвести до втрати сервісу та потенційної втрати або крадіжки даних. Під час DDoS-атаки багато зламаних комп'ютерів заливають цільову систему трафіком, створюючи так звану бот-мережу і зменшуючи її здатність обробляти законні запити.

- Соціальна інженерія. Атаки соціальної інженерії покладаються на людську взаємодію, щоб обманом змусити користувачів надати конфіденційну інформацію або виконати дії, які можуть поставити під загрозу безпеку CRM-системи. Наприклад, зловмисники можуть викрасти конфіденційні дані з CRM-систем, такі як імена клієнтів, адреси та платіжні дані, обманом змусивши користувачів надати цю інформацію або свої облікові дані для входу в систему. Атаки соціальної інженерії також можуть включати розповсюдження шкідливого програмного забезпечення через вкладення або посилання в електронних листах, що може поставити під загрозу безпеку CRM-системи.

В CRM-системах існує кілька поширених вразливостей, які можуть поставити під загрозу безпеку та конфіденційність даних. Нижче наведені приклади найпопулярніші з них:

- Слабка автентифікація та контроль доступу. Ці компоненти можуть становити значні вразливості в CRM-системах. Аутентифікація - це процес перевірки особи користувача, а контроль доступу - це механізми, які обмежують

доступ користувача до різних частин CRM-системи. Слабка автентифікація та контроль доступу можуть дозволити зловмисникам отримати несанкціонований доступ до CRM-системи, викрасти конфіденційні дані або змінити інформацію.

Відсутність шифрування - це вразливість, яка може зробити системи управління взаємовідносинами з клієнтами вразливими до атак. Шифрування - це процес кодування інформації таким чином, щоб її могли прочитати лише авторизовані користувачі за допомогою відповідного ключа. Без шифрування конфіденційні дані, що зберігаються в CRM-системі, можуть бути перехоплені та прочитані зловмисниками, які отримають доступ до мережі або системи. Деякі поширені приклади даних, які повинні бути зашифровані в CRM-системі, включають інформацію, що дозволяє ідентифікувати особу, фінансові дані та конфіденційну ділову інформацію. Доступ до цих даних можуть отримати як зовнішні зловмисники, так і внутрішні загрози, наприклад, співробітники, які можуть мати несанкціонований доступ до CRM-системи.

- Відсутність оновлень і виправлень. Регулярні оновлення та виправлення програмного забезпечення необхідні для усунення вразливостей безпеки та захисту від нововиявлених загроз. Відсутність оновлень може зробити систему вразливою до атак.

- Інтеграція сторонніх додатків або сервісів з CRM-системою може призвести до появи нових вразливостей і підвищити ризик витоку даних. Кожна інтеграція може діяти як додаткова точка доступу, тому потрібно переконатися, що провайдери дотримуються суворих стандартів безпеки.

- Внутрішні загрози. Внутрішні загрози можуть становити серйозний ризик для безпеки та конфіденційності даних. Співробітники, підрядники або партнери з авторизованим доступом можуть навмисно або випадково спричинити порушення.

Дуже важливо регулярно оцінювати та усувати ці вразливості в CRM-системах, щоб гарантувати, що дані залишаються безпечними та конфіденційними.

1.5 Аналіз існуючих методологій для тестування на проникнення

Для забезпечення найкращих результатів тестування, незалежно від застосовуваних тестів на проникнення, потрібно дотримуватися методології проведення тестування на проникнення.

Найбільш популярні стандартні методи проведення тестування:

- Методологія з тестування OWASP.
- Методологія з тестування на проникнення PCI.
- Стандарт виконання тестування на проникнення
- Методологія NIST 800-115.
- Керівництво з методології тестування безпеки з відкритим вихідним кодом (OSSTMM).

Методологія з тестування OWASP

OWASP - цей проект об'єднав розробників програмних засобів з відкритим вихідним кодом. Люди, що входять до цієї спільноти, створюють програми для захисту веб-додатків і веб-сервісів.

Усі додатки створюються з урахуванням досвіду боротьби з програмами, що завдають шкоди веб-сервісам і веб-додаткам.

OWASP - це відправна точка для системних архітекторів, розробників, постачальників, споживачів і спеціалістів з безпеки, тобто всіх фахівців, які беруть участь у проектуванні, розробці, розгортанні та перевірці на безпеку всіх веб-сервісів і веб-додатків.

Головною перевагою методології OWASP є те, що за представленими результатами тестів можна отримати всебічний опис усіх загроз. Методологія OWASP визначає всі небезпеки, які можуть вплинути на роботу як системи, так і додатків, і оцінює ймовірність їхньої появи.

За допомогою описаних в OWASP загроз можна визначити загальну оцінку виявлених проведеним тестуванням ризиків і виробити відповідні рекомендації щодо усунення недоліків.

Посібник з тестування OWASP насамперед зосереджує увагу на таких питаннях:

- Методи та інструменти тестування веб-додатків.
- Збір інформації.
- Перевірка автентичності.
- Тестування бізнес-логіки.
- Дані випробувань.
- Тестування атак типу "відмова в обслуговуванні".
- Перевірка управління сесіями.
- Тестування веб-сервісів.
- Тест AJAX.
- Визначення ступеня ризиків.
- Імовірність загроз.

Методологія з тестування на проникнення PCI

В даній методології зібрані методи, що відповідають вимогам PCI. Причому в керівництві, можна знайти нормативи не тільки за стандартом PCI v3.2.

Він створений Радою безпеки за стандартами PCI, у якій визначено методи тестування на проникнення в рамках програм управління вразливостями.

Стандарт PCI Data Security Standard (PCI DSS) версії 3.2 було випущено у квітні 2016. Після оновлення стандарту, з'явилися додаткові вказівки та сім нових вимог для усунення проблем, пов'язаних із порушеннями секретності особистих даних власників карток, а також для захисту від наявних експлойтів. До стандарту PCI DSS V. 3.2 було включено різні зміни, більшість з яких стосуються постачальників послуг.

До цих змін було додано нові вимоги до тестування на проникнення, згідно з якими тестування із сегментацією для постачальників послуг виконувалося принаймні кожні шість місяців або після будь-яких значних змін в елементах управління/методах сегментації. Крім того, у цьому стандарті міститься кілька вимог, які зобов'язують постачальників послуг протягом року безперервно

відслідковувати та підтримувати критично важливі елементи управління безпекою.

Стандарт виконання тестування на проникнення

Стандарт складається з семи основних розділів. Вони охоплюють усі вимоги, умови та методи проведення тестування на проникнення: від розвідки і до спроб проведення пентестів; етапи збору інформації та моделювання загроз, щоб домогтися найкращих результатів перевірки, спеціалісти працюють інкогніто; етапи дослідження вразливостей, експлуатації та пост-експлуатації, коли практичні знання спеціалістів у сфері безпеки з'єднуються з даними, отриманими під час проведення тестів на проникнення; і як заключний етап - звітність, у якій вся інформація надається у вигляді, зрозумілому клієнту.

Сьогодні діє перша версія, в якій всі стандартні елементи випробувані в реальних умовах і затверджені. Друга версія перебуває на стадії розробки. У ній усі вимоги буде деталізовано, уточнено та вдосконалено.

Оскільки план кожного тесту на проникнення розробляється індивідуально, у ньому можуть бути застосовані різні тести: від тестування веб-додатків до проведення тестів, передбачених для тестування методом "чорного ящика". За допомогою цього плану одразу можна визначити очікуваний рівень складності конкретного дослідження і застосувати його в необхідних, на думку організації, обсягах і областях.

Основні розділи розглянутого стандарту:

- Попередня угода на взаємодію.
- Збір розвідданих.
- Моделювання загроз.
- Аналіз вразливостей.
- Експлуатація.
- Пост-експлуатація.
- Складання звіту.

Методологія NIST 800-115

Спеціальне видання Національного інституту стандартів і технологій є технічною методологією з тестування та оцінювання інформаційної безпеки.

Публікація підготовлена Лабораторією інформаційних технологій у NIST. У методології оцінка безпеки трактується як процес визначення того, наскільки ефективно оцінювана організація відповідає конкретним вимогам безпеки. Документ нечасто оновлюється, але він не застарів і може послужити як довідник для побудови методології тестування.

У цьому довіднику пропонуються практичні рекомендації з розробки, впровадження та ведення технічної інформації, тестів безпеки та процесів і процедур експертизи, що охоплюють ключові елементи або технічне тестування на безпеку та експертизу.

Ці рекомендації можна використовувати для кількох практичних завдань. Наприклад, пошук вразливостей у системі або мережі та перевірка відповідності політиці або іншим вимогам.

Стандарт NIST 800-115 надає великий план для випробувань на проникнення. Він дає змогу переконатися, що програма тестування на проникнення відповідає рекомендаціям.

Методологія тестування безпеки з відкритим вихідним кодом

OSSTMM - документ, досить складний для читання і сприйняття. Але він містить велику кількість актуальної та дуже докладної інформації з безпеки.

Це також найвідоміша методологія із безпеки. Причина такої популярності в наступному: ці інструкції приблизно на десятиліття випереджають усі інші документи в індустрії безпеки.

Мета OSSTMM - у розвитку стандартів перевірки безпеки Інтернету. Цей документ призначений для формування найбільш докладного основного плану для тестування, що, своєю чергою, забезпечить доскональне і всебічне випробування на проникнення. Незалежно від інших організаційних особливостей, таких як корпоративний профіль постачальника послуг із тестування на проникнення, це випробування дасть змогу клієнту переконатися в рівні технічної оцінки.

1.6 Постановка задачі

Для досягнення мети кваліфікаційної роботи – підвищення рівня захисту інформації в інформаційній системі ТОВ «МОДЕРА РОЗВИТОК УКРАЇНА» необхідно розв'язати наступні задачі:

- Створити вимоги до тестування
- Проаналізувати модель порушника і загроз
- Протестувати систему на вразливість згідно вимог до тестування
- Запропонувати комплекс заходів захисту для виправлення вразливостей

1.7 Висновок

У першому розділі кваліфікаційної роботи проведено обстеження підприємства, що займається розробкою та підтримкою веб-додатків «МОДЕРА РОЗВИТОК УКРАЇНА». Проаналізовано роботу особового складу підприємства та його обов'язки, також проаналізовано можливі вразливості одного з проектів, а саме, CRM-системи для управлінням автосалоном. На основі аналізу підприємства та CRM-системи і огляду методологій тестування на вразливість можна зробити висновок, що для найефективнішого тестування потрібно використовувати методологію OWASP.

2 СПЕЦІАЛЬНА ЧАСТИНА

2.1 Розробка вимог до тестування на вразливість

У першому розділі розглянуто інформаційну систему компанії, тому можна виконати тестування на проникнення веб-додатку, враховуючи теоретичні відомості про вразливості CRM-системи, на наявність критичних та високорівневих вразливостей, але для ефективного тестування необхідно спочатку сформувати вимоги до процесу тестування:

- Детальний опис інструментів, які будуть використовуватися в тестуванні;
- Детальний опис процедури підготовки до тестування;
- Опис програмного забезпечення, яке може бути використано для тестування;
- Розробка методів та підходів до тестування;
- Детальний опис термінів і понять у сфері інформаційної безпеки;

Беручи до уваги вищезазначені вимоги, можна зробити висновок, що тестування на проникнення слід розділити на кілька пунктів, а саме

- Збір інформації про інформаційну систему;
- Аналіз портів;
- Тестування інструментів перевірки вхідних даних;
- Тестування засобів аутентифікації;
- Тестування засобів контролю доступу;
- Тестування компонентів, які можуть мати вразливості;
- Створення методів захисту системи;
- Результати тестування;

Також слід не забувати про створення моделі порушника для більш детального розуміння причин та можливості використання вразливостей, котрі будуть знайдені під час тестування на проникнення. Завдяки моделі порушника можна зрозуміти, хто може найбільшою загрозою в ролі інсайдеру.

2.2 Розробка моделі порушника і загроз

Згідно з НД-ТЗІ 1.1-003-99 «Термінологія в галузі захисту інформації комп'ютерних систем від несанкціонованого доступу» від " 28 " квітня 1999 р.:

Загроза (threat) — будь-які обставини або події, що можуть бути причиною порушення політики безпеки інформації і/або нанесення збитків АС.

Несанкціонований доступ до інформації; НСД до інформації (unauthorized access to information) — доступ до інформації, здійснюваний з порушенням ПРД.

Модель порушника (user violator model) — абстрактний формалізований або неформалізований опис порушника.

Ризик (risk) — функція ймовірності реалізації певної загрози, виду і величини завданих збитків.

Порушник (user violator) — користувач, який здійснює несанкціонований доступ до інформації.

Модель порушника показує опис можливих дій, які базується на аналізі типу зловмисника, повноважень, практичних та теоретичних можливостях нанесення шкоди АС. Відносно дослідженню АС, можна прийти висновку що потенційними зловмисниками можуть бути в першу чергу – технічний персонал.

Нижче проведено процедури пов'язані з виявленням негативних факторів пливу на ІТС.

Модель порушника:

Таблиця 2.1 – Категорія порушників, визначених у моделі (Внутрішні за відношенням до ІТС)

Позначення	Визначення категорії	Рівень загрози
1	2	3
ПВ1	Провідні інженери (провідний програміст, провідний інженер якості продукту)	4
ПВ2	Технічний персонал який обслуговує будівлю та приміщення (сантехніки, тощо)	2

Продовження таблиці 2.1

1	2	3
ПВ3	Користувачі	2
ПВ4	Відділ підтримки	1
ПВ5	Адміністратор безпеки	5
ПВ6	Персонал, який обслуговує технічні засоби ІТС (програмісти, системний адміністратор, тестувальники)	2
ПВ7	Керівники	5

Таблиця 2.2 – Категорія порушників, визначених у моделі (Зовнішні за відношенням до ІТС)

Позначення	Визначення категорії	Рівень загрози
ПЗ1	Відвідувачі	1
ПЗ2	Представники організацій, що взаємодіють з питань технічної підтримки функціональності	1
ПЗ3	Хакери (особа що намагається отримати несанкціонований доступ)	3
ПЗ4	Агенти	2

Таблиця 2.3 – Специфікація моделі порушника за мотивами

Позначення	Мотив порушення	Рівень загрози
М1	Безвідповідальність (недбалість)	3
М2	Корислива цілеспрямованість	5
М3	Професійний обов’язок	3

Таблиця 2.4 – Специфікація моделі порушника за рівнем кваліфікації та обізнаності щодо ІТС

Позначення	Основні кваліфікаційні ознаки порушника	Рівень загрози
К1	Не володіє знаннями про ІТС, та не має навичок користуванням штатними засобами системи	1
К2	Має навички щодо користування ПК на рівні користувача	2
К3	Має базові знання ІТС, володіє достатніми знаннями у галузі програмування та обчислюваної техніки	4
К4	Знає структуру, функції , механізми захисту інформації в ІТС і її недоліки	5

Таблиця 2.5 – Специфікація моделі порушника за показником можливостей використання засобів для реалізації загроз

Позначення	Характеристика можливостей порушника	Рівень загрози
З1	Має фізичний доступ до ІТС, але не є авторизованим користувачем	1
З2	Має можливість запуску заздалегідь підготовних функцій та скриптів	3
З3	Має можливість конфігурувати програмне забезпечення та комплекс засобів захисту ІТС	5
З4	Не має фізичного доступу до ресурсів ІТС	1

Таблиця 2.6 – Специфікація моделі порушника за часом дії

Позначення	Характеристика можливостей порушника	Рівень загрози
Ч1	Під час функціонування ІТС	4
Ч2	Під час перерви у роботі для обслуговування та ремонту	3
Ч3	Під час бездіяльності компонентів у системі (планова перерва, неробочий час)	4

Таблиця 2.7 – Специфікація моделі порушника за місцем дії

Позначення	Характеристика місця дії порушника	Рівень загрози
Д1	Усередині приміщень, але без доступу до технічних засобів ІТС	2
Д2	З робочих місць (користувачів, операторів)	2
Д3	З контрольованої території без доступу у будинки та споруди	1
Д4	З доступом у зону керування засобами забезпечення безпеки ІТС відділу інформаційних технологій	4

В таблиці 2.8, сформовані профілі можливостей порушників всіх категорій з урахуванням зазначених факторів в таблиця 2.1 – 2.7 . У графі “Ефективний рівень загроз” наведено рейтингову оцінку загроз порушника з відповідними характеристиками.

Таблиця 2.8 – Модель порушника

Посада	Категорія порушника	Мотив порушення	Кваліфікація	Можливості	Час дії	Місце дії	Сума загроз
1	2	3	4	5	6	7	8
Директор	ПВ7	М1	К4	З2	Ч1	Д4	15
	5	3		3	4		
	ПЗ4	М2	5	З3	Ч3	4	23
	2	5		5	4		
Менеджер	ПВ3	М1	К2	З1	Ч1	Д2	14
	2	3		1	4		
	ПЗ4	М3	2	З2	Ч3	2	13
	2	3		3	4		
Служба підтримки	ПВ4	М1	К2	З2	Ч1	Д2	15
	1	3		3	4		
	ПЗ4	М3	2	З2	Ч3	2	16
	2	3		3	4		

Продовження таблиці 2.8

1	2	3	4	5	6	7	8
Провідний програміст	ПВ1	М3	К4	З3	Ч2	Д4	26
	4	3		5	3		
	ПЗ4	М2	5	З3	Ч3	4	25
	2	5		5	4		
Сантехнік	ПВ2	М1	К1	З4	Ч2	Д3	10
	2	3		1	2		
	ПЗ4	М3	1	З1	Ч3	1	12
	2	3		1	4		
Адміністратор безпеки	ПВ5	М1	К4	З2	Ч1	Д4	24
	5	3		3	4		
	ПЗ4	М2	5	З3	Ч3	4	25
	2	5		5	4		

Із таблиці 2.8 видно, що найбільшу загрозу несе адміністратор безпеки, провідний програміст, директор через високу кваліфікацію та знання ПЗ. Вони можуть становити загрозу для підприємства у разі підкупу.

Таблиця 2.9 – Модель внутрішнього порушника

Посада	Категорія порушника	Мотив порушення	Кваліфікація	Можливості	Час дії	Місце дії	Сума загроз
Директор	ПВ7	М1	К4	З2	Ч1	Д4	15
Менеджер	ПВ3	М1	К2	З1	Ч1	Д2	14
Служба підтримки	ПВ4	М1	К2	З2	Ч1	Д2	15
Провідний програміст	ПВ1	М3	К4	З3	Ч2	Д4	26
Сантехнік	ПВ2	М1	К1	З4	Ч2	Д3	10
Адміністратор безпеки	ПВ5	М1	К4	З2	Ч1	Д4	24

Із таблиці 2.9 видно, що найбільшу загрозу становлять: провідний програміст та адміністратор безпеки. Всі працюють в одному офісі та

підпорядковуються директору, тому організація роботи цих осіб повинна бути найбільш контрольованою, так як вони є основними потенційними порушниками інформаційної безпеки.

Модель загроз з визначенням рівня ризиків та збитків

Таблиця 2.10 – Загрози конфіденційності інформації

№	Механізм реалізації	Рівень		Сума загроз
		ризики	збитки	
К.1	Халатність співробітників підприємства	2	2	4
К.2	Фішингові атаки	1	2	3
К.3	SQL-ін'єкції	3	3	6
К.4	Методи соціальної інженерії	2	1	3
К.5	Слабка автентифікація та контроль доступу	2	3	5

Таблиця 2.11 – Загрози цілісності інформації

№	Механізм реалізації	Рівень		Сума загроз
		ризики	збитки	
Ц.1	Відсутність вчасного резервного копіювання	1	3	4
Ц.2	Відсутність підтвердження відправника інформації що надходить на обробку	1	2	3
Ц.3	Атака на відмову в обслуговуванні	3	3	6
Ц.4	Помилки користувачів, які призвели до модифікації або видаленню інформації	1	3	4
Ц.5	Помилки програмного забезпечення, в наслідок яких стала можливою модифікація даних	2	1	3

Таблиця 2.12 – Загрози доступності інформації

№	Механізм реалізації	Рівень		Сума загроз
		ризики	збитки	
Д.1	Помилка користувача, яка призвела до знищення даних	2	2	4
Д.2	Навмисне видалення або деформація даних	1	2	3
Д.3	Прояви помилок програмного забезпечення, що призвели до втрати доступу до інформації	1	1	2
Д.4	Відсутність оновлення і виправлень системи	2	2	4
Д.5	Помилка при створенні шифрування	1	3	4

Таблиця 2.13 – Загрози спостереженості

№	Механізм реалізації	Рівень		Сума загроз
		ризики	збитки	
С.1	Прояви помилок програмного забезпечення, яке призвело до втрати спостереженості	2	1	3
С.2	Навмисні помилки персоналу, які призвели до втрати спостереженості	2	1	3
С.3	Ненавмисні помилки персоналу, які призвели до втрати спостереженості	3	3	6
С.4	Навмисне порушення спостереженості користувачами	2	1	3
С.5	Безпосередній доступ до системи будь-яким методом	2	2	4

Таблиця 2.14 – Узагальнена таблиця загроз

Види загроз	1	2	3	4	5	Сума загроз
Конфіденційності	4	3	6	3	5	21
Цілісності	4	3	6	4	3	20
Доступності	4	3	2	4	4	17
Спостереженості	3	3	6	3	4	19

На основі даних, отриманих з таблиці 2.14, найбільшими загрозами підприємству є конфіденційність і цілісність. Для зменшення загроз потрібно провести тестування на проникнення і базуючись на отриманих результатів, розробити комплекс заходів захисту.

2.3 Тестування системи на проникнення

Тестування на проникнення системи повинно починатися з розвідки. Для збору інформації буде використовуватися Nmap. Оскільки відкриті порти необхідні для будь-якого типу зв'язку через Інтернет, вони можуть становити ризик для безпеки. Цей інструмент також може дати уявлення про ефективність конфігурацій безпеки і брандмауера.

Nmap або Network Mapper - це мережевий інструмент, який використовується для сканування служб, операційних систем і хостів у комп'ютерній мережі. Деякі з найважливіших функцій Nmap включають сканування портів, виявлення хостів, керування версіями, інвентаризацію мережі, дактилоскопію стека TCP або IP та багато іншого.

Сканування портів - це процес, який дозволяє визначити, які порти в мережі відкриті, а які закриті. Відкриті порти використовуються для надсилання та отримання інформації. Скануючи всі порти, можна визначити, які з них відкриті, і, можливо, виявити конфіденційну мережеву інформацію.

Список необхідних параметрів:

- -sV – функція для визначення версії. Ця функція може бути корисною для диференціації справді відкритих портів і тих, що фільтруються.
- -O: Увімкнути виявлення ОС

Результат перевірки відкритих портів та сканування запущених служб зображено на рисунку 2.1

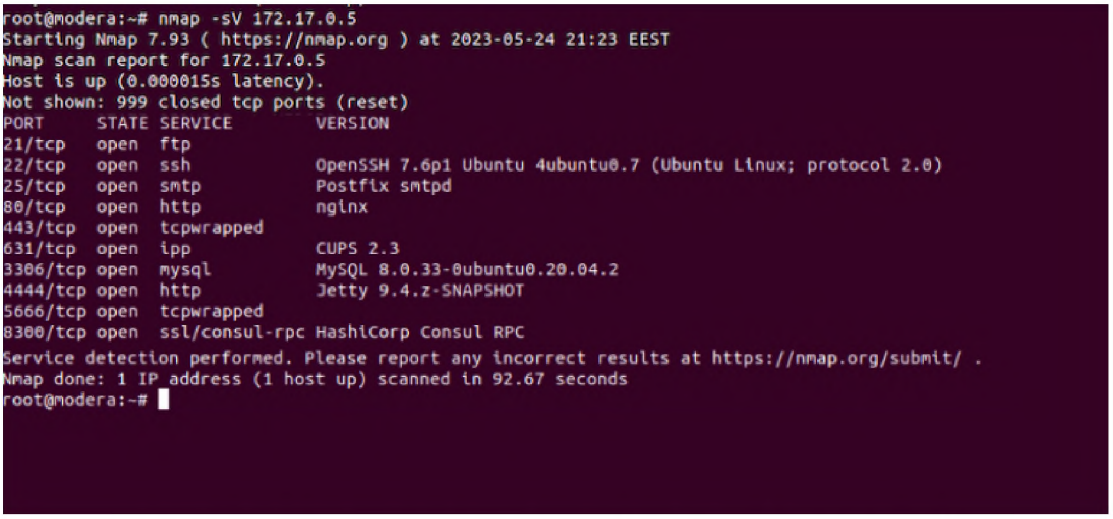


Рисунок 2.1 – Результат перевірки відкритих портів

Використовуючи команду «nmap -O 172.17.05» визначимо можливі типи операційної системи.

Рисунок 2.2 – Результат виявлення операційної системи

Отже з допомогою утиліти, відомо, що на веб сервері є такі сервіси: OpenSHH 7.6p1, MySQL 8.0.33 та Jetty 9.4. Можлива операційна система Linux з версіями 4.15 до 5.6.

Тепер потрібно використати те, що було знайдено завдяки Nmap. З відомих відкритих портів потрібно знайти експлойти.

Експлойт — це будь-яка атака, яка використовує вразливі місця в програмах, мережах, операційних системах або обладнанні. Експлойти зазвичай

приймають форму програмного забезпечення або коду, спрямованого на контроль над комп'ютерами або викрадення мережевих даних.

Searchsploit — це інструмент, доступний для Linux. За допомогою цього інструменту можна зібрати експлойти CVE з exploit-db.

- searchsploit http

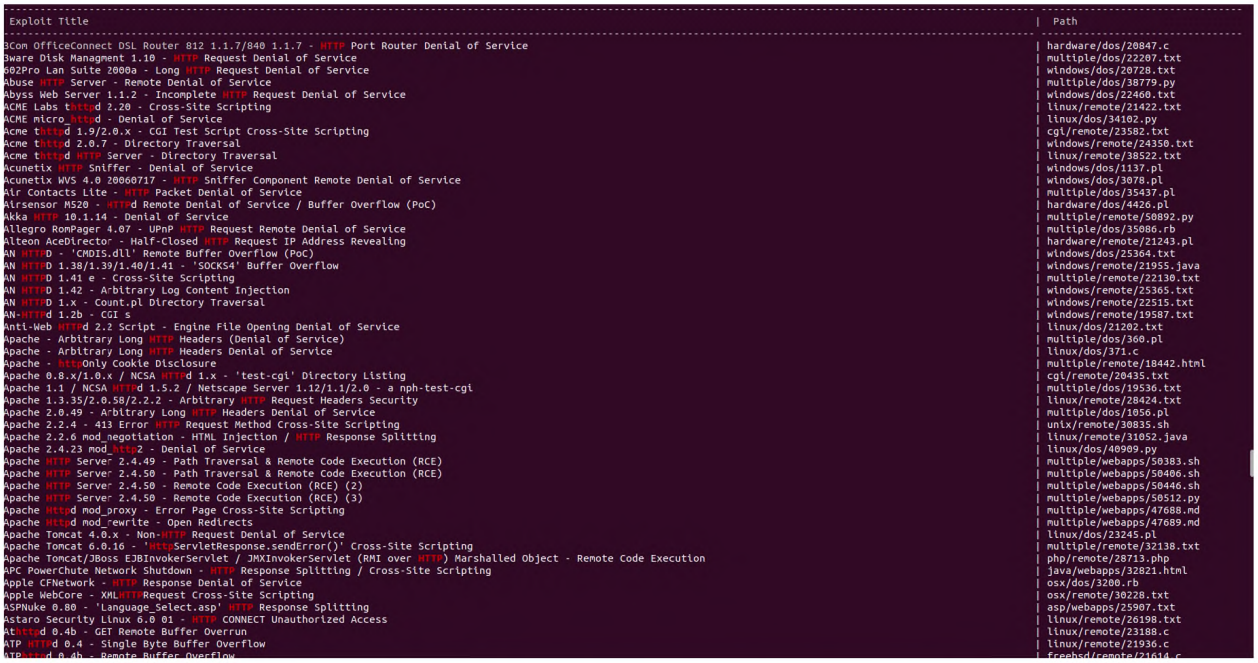


Рисунок 2.3 – Результат перевірки експлойтів http

- searchsploit ssh

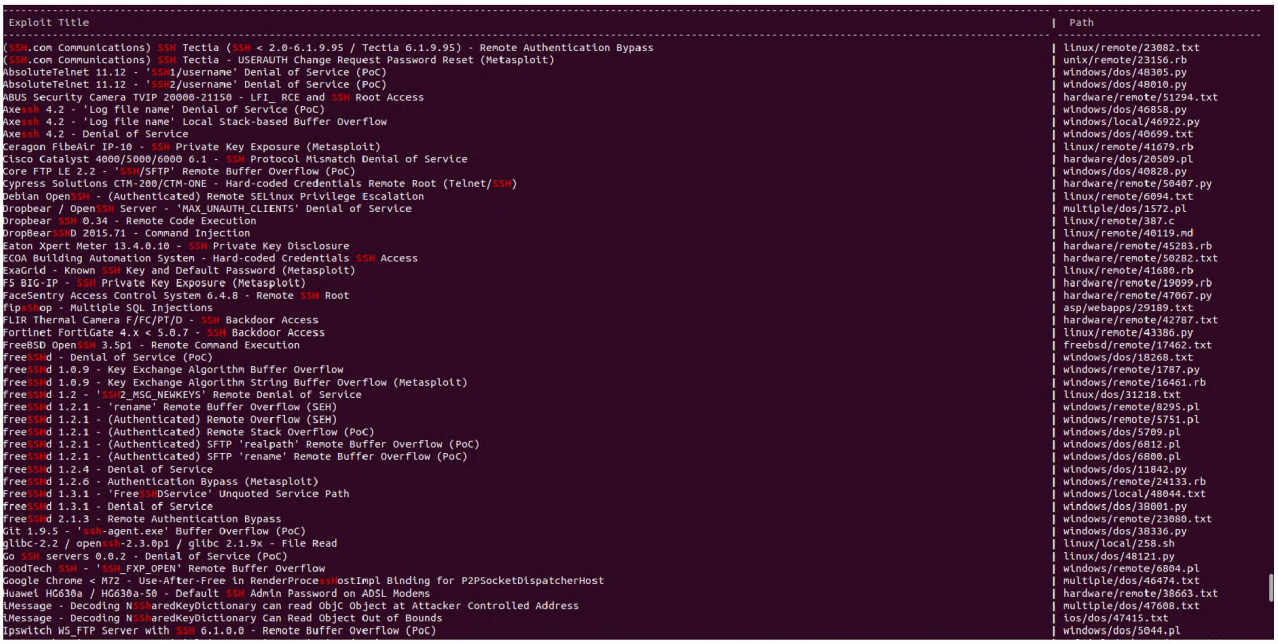
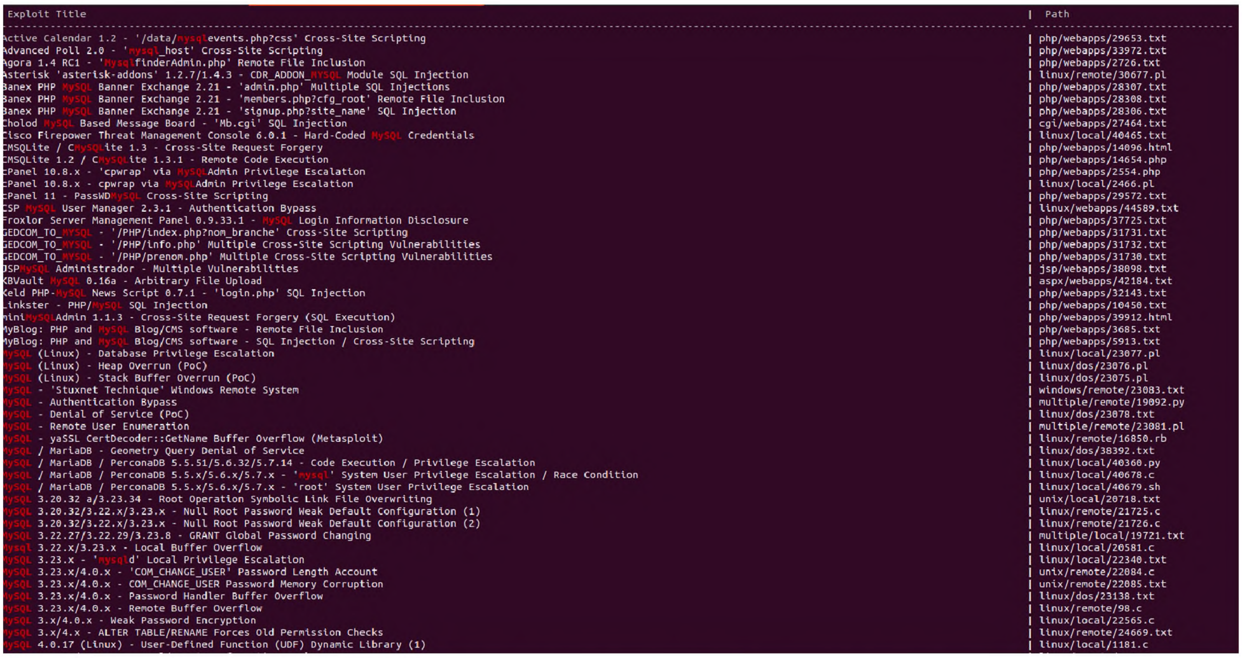


Рисунок 2.4 – Результат перевірки експлойтів ssh

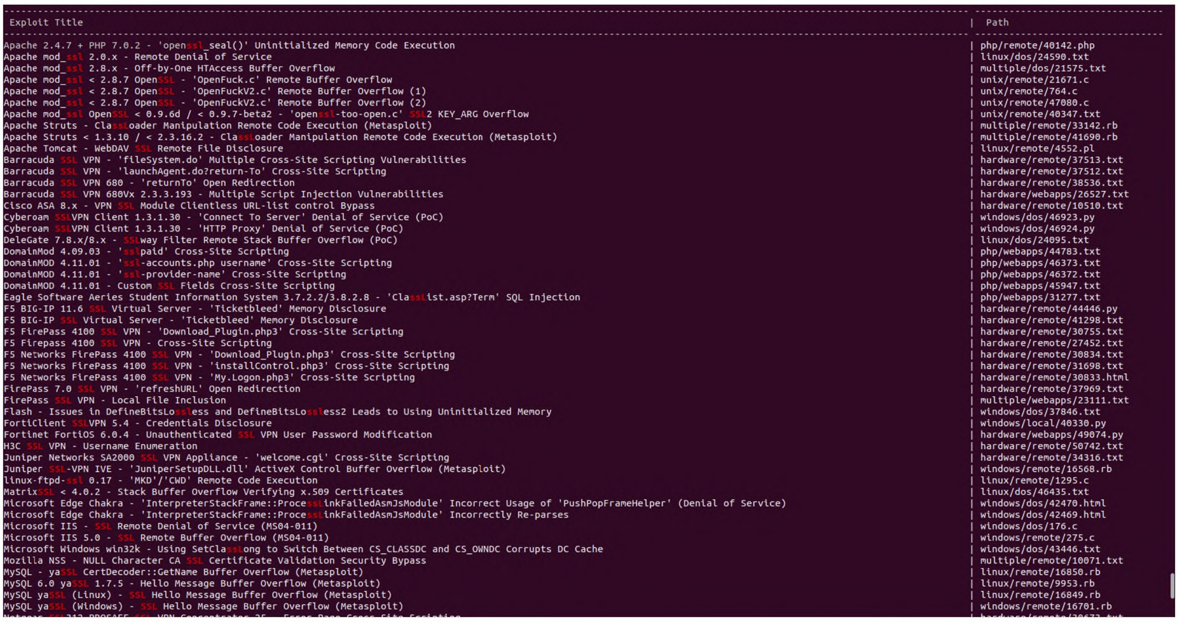
- searchsploit mysql



Exploit Title	Path
Active Calendar 1.2 - '/data/mysqlevents.php?css' Cross-Site Scripting	php/webapps/29653.txt
Advanced Poll 2.0 - 'mysql_host' Cross-Site Scripting	php/webapps/33972.txt
Agora 1.4 RC1 - 'MysqlfinderAdmin.php' Remote File Inclusion	php/webapps/2726.txt
Asterisk 'asterisk-addons' 1.2.7/1.4.3 - CDR_ADDON_MYSQL Module SQL Injection	linux/remote/30677.pl
Banex PHP MySQL Banner Exchange 2.21 - 'admin.php' Multiple SQL Injections	php/webapps/28307.txt
Banex PHP MySQL Banner Exchange 2.21 - 'members.php?cfg_root' Remote File Inclusion	php/webapps/28308.txt
Banex PHP MySQL Banner Exchange 2.21 - 'signup.php?site_name' SQL Injection	php/webapps/28306.txt
Cholod MySQL Based Message Board - 'Mb.cgi' SQL Injection	cgi/webapps/27464.txt
Cisco Firepower Threat Management Console 6.0.1 - Hard-Coded MySQL Credentials	linux/local/40465.txt
CMSQLite / CMySQLite 1.3 - Cross-Site Request Forgery	php/webapps/14096.html
CMSQLite 1.2 / CMySQLite 1.3.1 - Remote Code Execution	php/webapps/14654.php
cPanel 10.8.x - 'cpwrap' via MySQLAdmin Privilege Escalation	php/webapps/2554.php
cPanel 10.8.x - cpwrap via MySQLAdmin Privilege Escalation	linux/local/2466.pl
cPanel 11 - PassWDMySQL Cross-Site Scripting	php/webapps/29572.txt
CSP MySQL User Manager 2.3.1 - Authentication Bypass	linux/webapps/44589.txt
Froxlor Server Management Panel 0.9.33.1 - MySQL Login Information Disclosure	php/webapps/37725.txt
GEDCOM_TO_MYSQL - '/PHP/index.php?nom_branche' Cross-Site Scripting	php/webapps/31731.txt
GEDCOM_TO_MYSQL - '/PHP/info.php' Multiple Cross-Site Scripting Vulnerabilities	php/webapps/31732.txt
GEDCOM_TO_MYSQL - '/PHP/prenom.php' Multiple Cross-Site Scripting Vulnerabilities	php/webapps/31730.txt
JSPMySQL Administrador - Multiple Vulnerabilities	jsp/webapps/38098.txt
KBVault MySQL 0.16a - Arbitrary File Upload	aspx/webapps/42184.txt
Keld PHP-MySQL News Script 0.7.1 - 'login.php' SQL Injection	php/webapps/32143.txt
Linkster - PHP/MySQL SQL Injection	php/webapps/10450.txt
miniMySQLAdmin 1.1.3 - Cross-Site Request Forgery (SQL Execution)	php/webapps/39912.html
MyBlog: PHP and MySQL Blog/CMS software - Remote File Inclusion	php/webapps/3685.txt
MyBlog: PHP and MySQL Blog/CMS software - SQL Injection / Cross-Site Scripting	php/webapps/5913.txt
MySQL (Linux) - Database Privilege Escalation	linux/local/23077.pl
MySQL (Linux) - Heap Overrun (PoC)	linux/dos/23076.pl
MySQL (Linux) - Stack Buffer Overrun (PoC)	linux/dos/23075.pl
MySQL - 'Stuxnet Technique' Windows Remote System	windows/remote/23083.txt
MySQL - Authentication Bypass	multiple/remote/19092.py
MySQL - Denial of Service (PoC)	linux/dos/23078.txt
MySQL - Remote User Enumeration	multiple/remote/23081.pl
MySQL - yaSSL CertDecoder::GetName Buffer Overflow (Metasploit)	linux/remote/16850.rb
MySQL / MariaDB - Geometry Query Denial of Service	linux/dos/38392.txt
MySQL / MariaDB / PerconaDB 5.5.51/5.6.32/5.7.14 - Code Execution / Privilege Escalation	linux/local/40360.py
MySQL / MariaDB / PerconaDB 5.5.x/5.6.x/5.7.x - 'mysql' System User Privilege Escalation / Race Condition	linux/local/40678.c
MySQL / MariaDB / PerconaDB 5.5.x/5.6.x/5.7.x - 'root' System User Privilege Escalation	linux/local/40679.sh
MySQL 3.20.32 a/3.23.34 - Root Operation Symbolic Link File Overwriting	unix/local/20718.txt
MySQL 3.20.32/3.22.x/3.23.x - Null Root Password Weak Default Configuration (1)	linux/remote/21725.c
MySQL 3.20.32/3.22.x/3.23.x - Null Root Password Weak Default Configuration (2)	linux/remote/21726.c
MySQL 3.22.27/3.22.29/3.23.8 - GRANT Global Password Changing	multiple/local/19721.txt
Mysql 3.22.x/3.23.x - Local Buffer Overflow	linux/local/20581.c
MySQL 3.23.x - 'mysqld' Local Privilege Escalation	linux/local/22340.txt
MySQL 3.23.x/4.0.x - 'COM_CHANGE_USER' Password Length Account	unix/remote/22084.c
MySQL 3.23.x/4.0.x - COM_CHANGE_USER Password Memory Corruption	unix/remote/22085.txt
MySQL 3.23.x/4.0.x - Password Handler Buffer Overflow	linux/dos/23138.txt
MySQL 3.23.x/4.0.x - Remote Buffer Overflow	linux/remote/98.c
MySQL 3.x/4.0.x - Weak Password Encryption	linux/local/22565.c
MySQL 3.x/4.x - ALTER TABLE/RENAME Forces Old Permission Checks	linux/remote/24669.txt
MySQL 4.0.17 (Linux) - User-Defined Function (UDF) Dynamic Library (1)	linux/local/1181.c

Рисунок 2.5 – Результат перевірки експлойтів mysql

- searchsploit ssl



Exploit Title	Path
Apache 2.4.7 + PHP 7.0.2 - 'openssl_seal()' Uninitialized Memory Code Execution	php/remote/40142.php
Apache mod_ssl 2.0.x - Remote Denial of Service	linux/dos/24590.txt
Apache mod_ssl 2.8.x - Off-by-One HTAccess Buffer Overflow	multiple/dos/21575.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow	unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)	unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)	unix/remote/47080.c
Apache mod_ssl OpenSSL < 0.9.6d / < 0.9.7-beta2 - 'openssl-too-open.c' SSL2 KEY_ARG Overflow	unix/remote/40347.txt
Apache Struts - ClassLoader Manipulation Remote Code Execution (Metasploit)	multiple/remote/33142.rb
Apache Struts < 1.3.10 / < 2.3.16.2 - ClassLoader Manipulation Remote Code Execution (Metasploit)	multiple/remote/41690.rb
Apache Tomcat - WebDAV SSL Remote File Disclosure	linux/remote/4552.pl
Barracuda SSL VPN - 'fileSystem.do' Multiple Cross-Site Scripting Vulnerabilities	hardware/remote/37513.txt
Barracuda SSL VPN - 'launchAgent.do?return-To' Cross-Site Scripting	hardware/remote/37512.txt
Barracuda SSL VPN 680 - 'returnTo' Open Redirection	hardware/remote/38536.txt
Barracuda SSL VPN 680Vx 2.3.3.193 - Multiple Script Injection Vulnerabilities	hardware/webapps/26527.txt
Cisco ASA 8.x - VPN SSL Module Clientless URL-list control Bypass	hardware/remote/10510.txt
Cyberoam SSLVPN Client 1.3.1.30 - 'Connect To Server' Denial of Service (PoC)	windows/dos/46923.py
Cyberoam SSLVPN Client 1.3.1.30 - 'HTTP Proxy' Denial of Service (PoC)	windows/dos/46924.py
DeleGate 7.8.x/8.x - SSLway Filter Remote Stack Buffer Overflow (PoC)	linux/dos/24095.txt
DomainMod 4.09.03 - 'sslpaid' Cross-Site Scripting	php/webapps/44783.txt
DomainMOD 4.11.01 - 'ssl-accounts.php username' Cross-Site Scripting	php/webapps/46373.txt
DomainMOD 4.11.01 - 'ssl-provider-name' Cross-Site Scripting	php/webapps/46372.txt
DomainMOD 4.11.01 - Custom SSL Fields Cross-Site Scripting	php/webapps/45947.txt
Eagle Software Aeries Student Information System 3.7.2.2/3.8.2.8 - 'ClassList.asp?Term' SQL Injection	php/webapps/31277.txt
F5 BIG-IP 11.6 SSL Virtual Server - 'Ticketbleed' Memory Disclosure	hardware/remote/44446.py
F5 BIG-IP SSL Virtual Server - 'Ticketbleed' Memory Disclosure	hardware/remote/41298.txt
F5 FirePass 4100 SSL VPN - 'Download_Plugin.php3' Cross-Site Scripting	hardware/remote/30755.txt
F5 Firepass 4100 SSL VPN - Cross-Site Scripting	hardware/remote/27452.txt
F5 Networks FirePass 4100 SSL VPN - 'Download_Plugin.php3' Cross-Site Scripting	hardware/remote/30834.txt
F5 Networks FirePass 4100 SSL VPN - 'installControl.php3' Cross-Site Scripting	hardware/remote/31698.txt
F5 Networks FirePass 4100 SSL VPN - 'My.Logon.php3' Cross-Site Scripting	hardware/remote/30833.html
FirePass 7.0 SSL VPN - 'refreshURL' Open Redirection	hardware/remote/37969.txt
FirePass SSL VPN - Local File Inclusion	multiple/webapps/23111.txt
Flash - Issues in DefineBitsLossless and DefineBitsLossless2 Leads to Using Uninitialized Memory	windows/dos/37846.txt
FortiClient SSLVPN 5.4 - Credentials Disclosure	windows/local/40330.py
Fortinet FortiOS 6.0.4 - Unauthenticated SSL VPN User Password Modification	hardware/webapps/49074.py
H3C SSL VPN - Username Enumeration	hardware/remote/50742.txt
Juniper Networks SA2000 SSL VPN Appliance - 'welcome.cgi' Cross-Site Scripting	hardware/remote/34316.txt
Juniper SSL-VPN IVE - 'JuniperSetupDLL.dll' ActiveX Control Buffer Overflow (Metasploit)	windows/remote/16568.rb
linux-ftpd-ssl 0.17 - 'MKD'/'CWD' Remote Code Execution	linux/remote/1295.c
MatrixSSL < 4.0.2 - Stack Buffer Overflow Verifying x.509 Certificates	linux/dos/46435.txt
Microsoft Edge Chakra - 'InterpreterStackFrame::ProcessLinkFailedAsmJsModule' Incorrect Usage of 'PushPopFrameHelper' (Denial of Service)	windows/dos/42470.html
Microsoft Edge Chakra - 'InterpreterStackFrame::ProcessLinkFailedAsmJsModule' Incorrectly Re-parses	windows/dos/42469.html
Microsoft IIS - SSL Remote Denial of Service (MS04-011)	windows/dos/176.c
Microsoft IIS 5.0 - SSL Remote Buffer Overflow (MS04-011)	windows/remote/275.c
Microsoft Windows win32k - Using SetClassLong to Switch Between CS_CLASSDC and CS_OWNDC Corrupts DC Cache	windows/dos/43446.txt
Mozilla NSS - NULL Character CA SSL Certificate Validation Security Bypass	multiple/remote/10071.txt
MySQL - yaSSL CertDecoder::GetName Buffer Overflow (Metasploit)	linux/remote/16850.rb
MySQL 6.0 yaSSL 1.7.5 - Hello Message Buffer Overflow (Metasploit)	linux/remote/9953.rb
MySQL yaSSL (Linux) - SSL Hello Message Buffer Overflow (Metasploit)	linux/remote/16849.rb
MySQL yaSSL (Windows) - SSL Hello Message Buffer Overflow (Metasploit)	windows/remote/16701.rb

Рисунок 2.6 - Результат перевірки експлойтів ssl

- searchsploit vnc

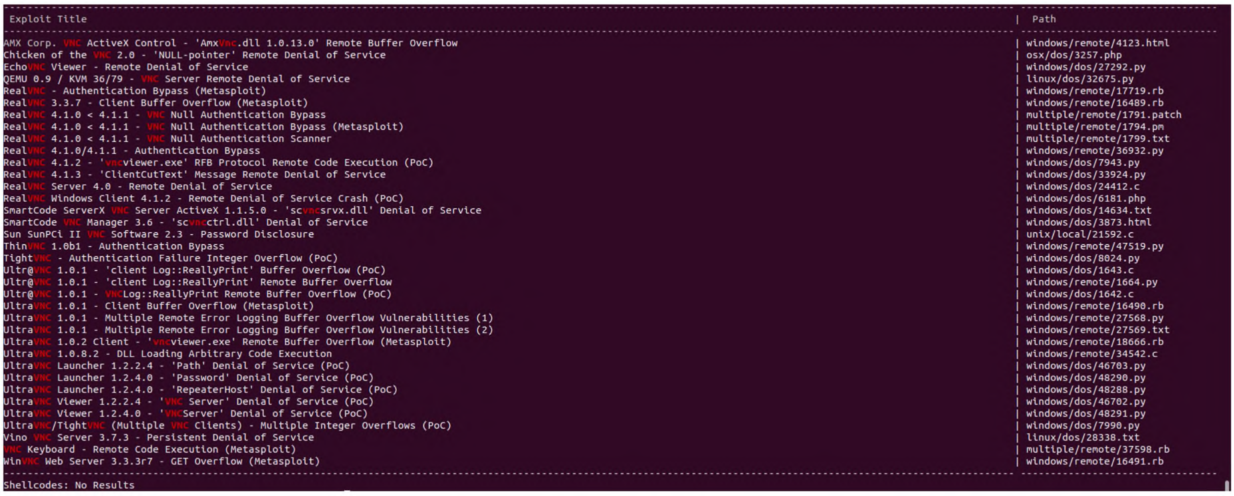


Рисунок 2.7 – Результат перевірки експлойтів vnc

З рисунків, котрі зображені вижче, можна зробити висновок, що експлойтів не було знайдено. При цій умові, потрібно продовжувати тестування через UI компоненти.

SQL-ін’єкція - це введені користувачем дані, які можуть бути використані для формування операторів SQL, які потім виконуються програмою в базі даних. Під час цієї атаки програма не може належним чином обробляти дані, введені користувачем.

Якщо це так, зловмисник може надати програмі несподівані вхідні дані, які потім використовуються для формування та виконання операторів SQL у базі даних. Наслідки такого вчинку можуть бути критичними.

Як випливає з самої назви, метою атаки SQL-ін’єкції є впровадження шкідливого коду SQL.

Кожне поле веб-сайту може бути використане, як ключ до бази даних. У формі входу користувач вводить дані для входу, у полі пошуку користувач вводить пошуковий текст, а у формі збереження даних користувач вводить дані, які потрібно зберегти. Всі вказані дані надходять до бази даних.

Якщо замість правильних даних буде введено будь-який шкідливий код, існує ймовірність серйозного пошкодження бази даних і всієї системи.

SQL-ін'єкції виконується за допомогою мови програмування SQL. SQL використовується для керування даними, що зберігаються в базі даних. Тому під час цієї атаки код мови програмування використовується як шкідлива ін'єкція.

Це одна з найпопулярніших атак, оскільки бази даних використовуються практично для всіх технологій.

Більшість програм використовують певний тип бази даних. Програма, що тестується, може мати інтерфейс користувача, який приймає введення користувача, який використовується для виконання таких завдань:

- Показати відповідні збережені дані

користувачеві, наприклад, програма перевіряє облікові дані користувача за допомогою інформації для входу, введеної користувачем, і надає користувачеві лише відповідні функції та дані.

- Зберігати дані, введені користувачем, у базу даних, наприклад , коли користувач заповнює форму та надсилає її, програма продовжує зберігати дані в базі даних; потім ці дані стають доступними для користувача в тому самому сеансі, а також у наступних сесіях.

Для тестуванні форми достатньо використати, якісь спеціальні символи SQL. Наприклад, використаємо ‘

Наприклад в нас є поле для пошуку компаній і якщо введемо «Test'», то з'явить помилка, про те, що некоректне введення SQL запиту.

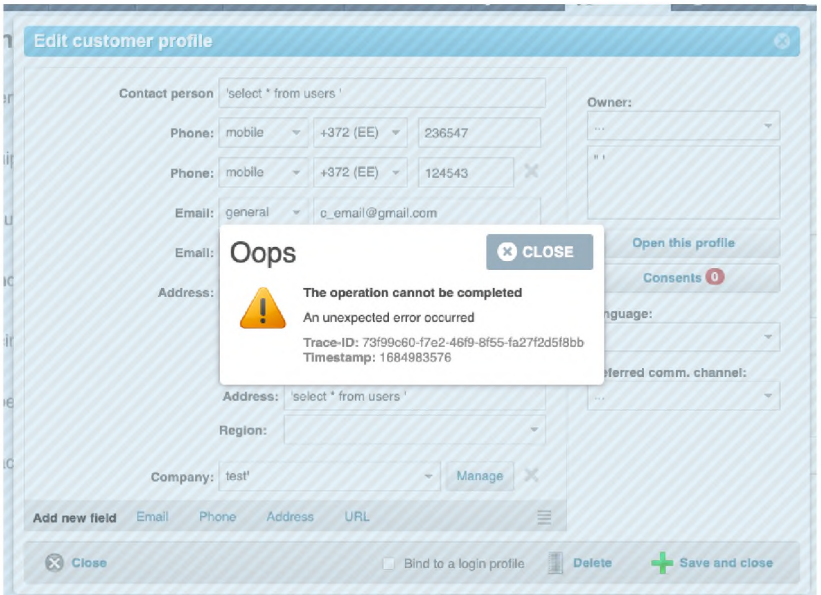


Рисунок 2.8 – Помилка через некоректний SQL запит

В логах це виглядає так: [app.CRITICAL: Uncaught PHP Exception "Doctrine\ORM\Query\QueryException": "[Syntax Error] line 0, col 128: Error: Expected Doctrine\ORM\Query\Lexer::T_CLOSE_PARENTHESIS, got "" at /var/www/vendor/doctrine/orm/lib/Doctrine/ORM/Query/QueryException.php line 32 {"exception":"[object] (Doctrine\\ORM\\Query\\QueryException(code: 0): [Syntax Error] line 0, col 128: Error: Expected Doctrine\\ORM\\Query\\Lexer::T_CLOSE_PARENTHESIS, got " at /var/www/vendor/doctrine/orm/lib/Doctrine/ORM/Query/QueryException.php:32, Doctrine\\ORM\\Query\\QueryException(code: 0): SELECT c FROM Modera\\AutomotiveCustomerDatabaseBundle\\Entity\\Company c WHERE c.name LIKE '%test"%' ORDER BY INSTR(c.name,'test"), c.name ASC at /var/www/vendor/doctrine/orm/lib/Doctrine/ORM/Query/QueryException.php:21)","trace_id":"73f99c60-f7e2-46f9-8f55-fa27f2d5f8bb"}
{"process_id":593,"memory_peak_usage":"2 MB"}

Отримали помилку, отже SQL-ін'єкція можлива. Переходимо до утиліти sqlmap та виконуємо наступну команду:

sqlmap –u "https://alpha.dev4.modera.org/backend/#tradein;v1=tradein&v2=edittradein&v2-carId=7&v3=update-customer&v3-id=1"` -- cookie="PHPSESSID=vghh66fgcrtch6r53424rghgcfcfxddjtt" -T modera_car_body, де

- -u – параметр, який слугує для вводу цілі в форматі веб-URL;
- PHPSESSID – файл cookie PHPSESSID дозволяє веб-сайтам зберігати серіалізовані дані про стан. Він використовується для встановлення сеансу користувача та передачі даних про стан через тимчасовий файл cookie, який зазвичай називають файлом cookie сеансу. (термін дії закінчується при закритті браузера);
- -T назва необхідної таблиці

Результат виконаної команди з усіма даними з таблиці "Тип автомобіля":

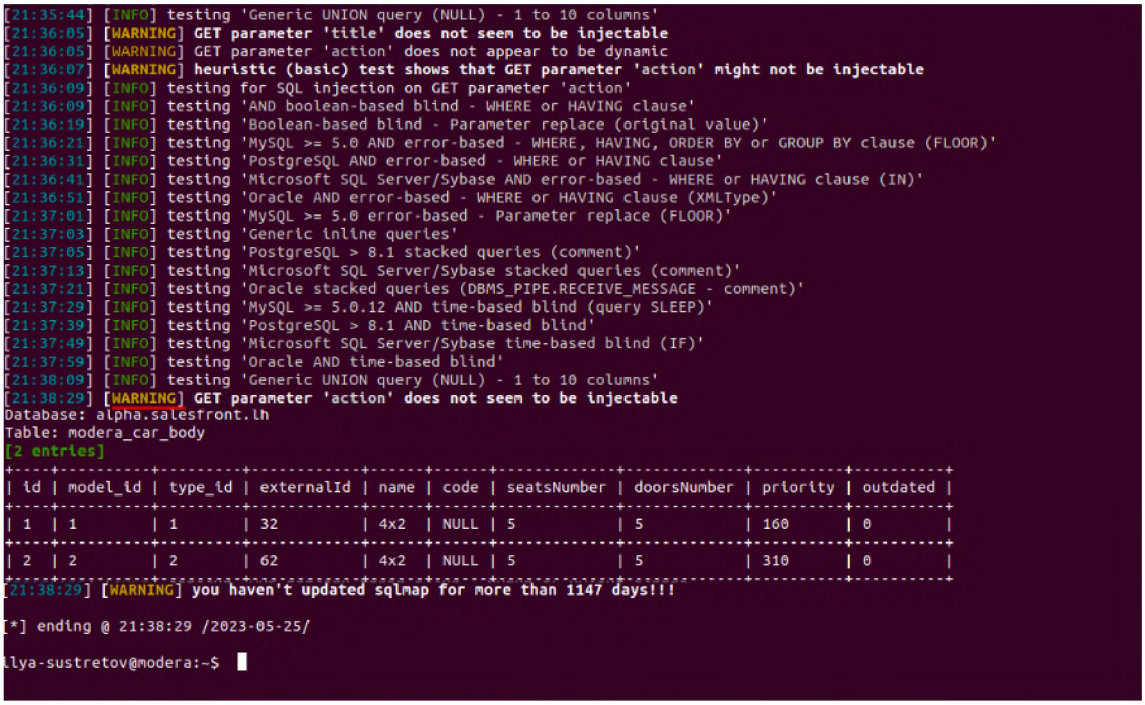


Рисунок 2.9 – Результат роботи команди

Результат роботи, котрий продемонстровано вище, доводить проблему вразливості до SQL-ін’єкцій.

Наступний крок - атака XSS - це ін'єкційна атака, яка використовується проти клієнтської системи.

Мови сценаріїв, які можуть виконуватися у веб-браузері, наприклад, Javascript, впроваджуються на веб-сторінку. Нічого не підозрюючи користувачі відвідують цю сторінку, і скрипт, наданий зловмисником, в браузері користувача. Існує два типи XSS-атак.

Перший називається збереженою або постійною XSS. Це означає, що зловмиснику вдалося зберегти скрипт всередині в базі даних, так що скрипт відображається на веб-сторінці, коли користувач заходить на сайт. Це може бути, наприклад, поле для коментарів у блозі. Оскільки скрипт обробляється браузером, користувач ніколи не бачить його і навіть не знає, що він запущений. За допомогою такої атаки можна засіяти сайт і просто чекати, поки нічого не підозрюючи користувачі відвідають його.

Другий тип XSS-атаки називається відображеною/непостійною. При непостійній атаці зловмисник створює URL-адресу з вбудованим в неї скриптом, який викликає сторінку, вразливу до XSS-атаки. При такому типі атаки

зловмисникові потрібно створити URL-адресу з параметрами, які включають скрипт, а потім відправити його користувачам. Це можна зробити через електронною поштою, використовуючи додатковий скрипт в HTML-повідомленні, щоб приховати справжню URL-адресу.

Оскільки додаткові параметри і скрипт, швидше за все, видадуть атаку. Сама атака дуже проста. Потрібно надати набір HTML-тегів, які вказують на те, що є якийсь скрипт. Коли це відображається у веб-браузері, браузер бачить теги скрипта і запускає те, що знаходиться всередині тегів. Дуже простим прикладом може бути щось на кшталт:

<script>alert('ця сторінка вразлива до XSS');</script>

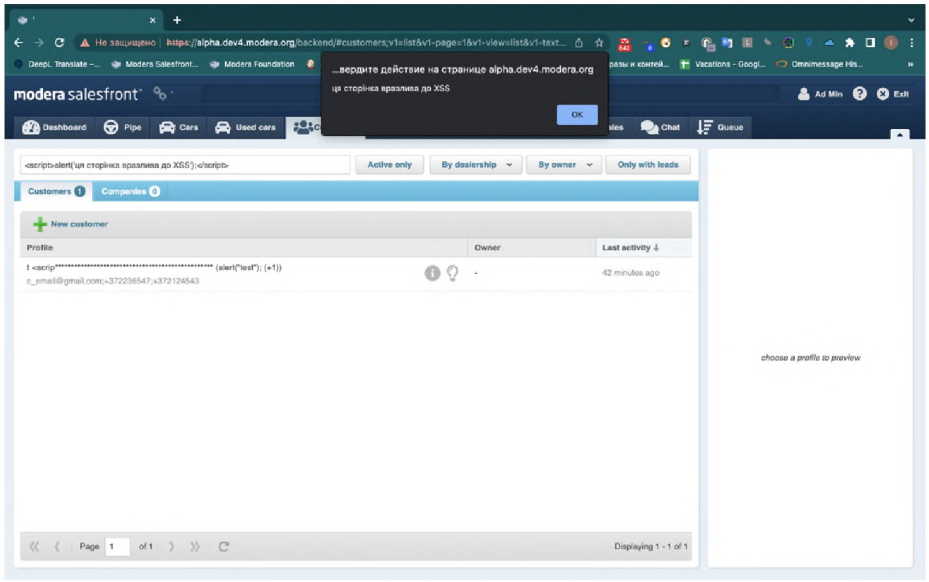


Рисунок 2.10 – Приклад застосування XSS

Можна також використати більш складний алгоритм. Наприклад, створити Javascript, який краде інформацію у користувача. Оскільки сторінка від третьої сторони не повинна мати можливості красти інформацію, яка зберігається від імені інших сайтів, можна використати XSS-атаку для збору цієї інформації, а потім передати її на інший сайт, де зловмисник може зібрати її пізніше.

Прикладом може бути щось на зразок цього:

<script>document.write('')</script></script>.

Ця невелика атака перехоплює cookie зі сторінки за допомогою document.cookie і вбудовує його в URL-адресу. URL знаходиться в тезі

зображення, а це означає, що браузер відправить запит на цю URL-адресу. URL-адреса може навіть не існувати, оскільки запит буде записано в журнал, який пізніше можна буде проаналізувати на наявність файлів cookie та IP-адресу. Посилання на goodsite.php може вказувати, з якого сайту прийшов файл cookie, якщо було атаковано кілька сайтів. Звичайно, за допомогою цього методу можна викрасти й іншу інформацію. Використання документа, за яким слідує метод або властивість, як в document.write або document.cookie, є об'єктною моделлю документа.

DOM - це спосіб перетворення веб-сторінки на колекцію об'єктів, до яких можна отримати доступ за допомогою об'єкта document.

Крім того, документ має методи, такі як write, до яких можна звертатися для виконання дій над документом. У цьому випадку збирається файл cookie з документа, в якому знаходиться цей скрипт і за допомогою методу write вставляє файл cookie в URL-адресу, що в підсумку призведе до відправкою cookie на зловмисний сервер. XSS-атаки виникають через неправильну перевірку введених даних. У більшості випадків немає жодних причин приймати HTML-теги як вхідні дані.

CSRF - це атака на користувача. Ця атака використовує несанкціоновану веб-сторінку, яка містить прихований запит. Іноді під час такої атаки визивається GET для POST-атаки. У більшості випадків, коли користувач відправляє інформацію на веб-сервер або просить його виконати якусь дію, користувач відправляє POST-запит, який зазвичай запускається натисканням кнопки.

Кнопка може бути пов'язана з веб-формою, що призведе до відправки всіх даних форми на сервер у POST-запиті. Якщо є веб-додаток, який приймає GET-запити разом з параметрами URL-адреси, то можна приховати цей запит, як показано у коді HTML нижче.

```
<img src="http://www.myexample.com/transfers.php?from=8888&to=5046475&
amount=10000" style="width:1px;height:1px">
```

Тег img відправить GET-запит до сервера, і оскільки він не отримає зображення, то немає ризику, що на сторінці щось з'явиться.

Щоб обмежити ризик появи чогось на сторінці, можна обмежити розмір зображення до 1x1 піксель, як у прикладі. Мало шансів, що користувач бачить таке маленьке зображення. Тим часом, можна зробити так, щоб сторінка виглядала абсолютно як завгодно, що може здатися привабливим для користувача.

Запит на транзакцію, особливо якщо користувач нещодавно увійшов до банку, і у нього є активний файл cookie для автентифікації, обробляється приховано, і користувач просто не знає про те, що вона відбулася.

Одним із способів захисту від такого роду атак є заборона програмного доступу через GET-запити.

POST-запит, не може бути виконаний за допомогою чогось на кшталт тегу img. Замість цього потрібно мати дію, яка може бути згенерована сторінкою.

Існують і інші способи захисту, які можна використовувати, в тому числі перевірка реферала, тобто заголовка, що вказує на сторінку, з якої прийшов запит. Якщо реферер не відповідає необхідному домену, краще не дозволяти запиту продовжуватися.

Ці засоби захисту не є надійними, але вони значно ускладнять виконання таких запитів.

Найголовніший розділ тестування – це тестування автентифікації. Автентифікація дозволяє увійти у веб-додаток, щоб мати персоналізований досвід перегляду, тоді як управління сеансами відстежує запити та відповіді, щоб можна було виконувати багатокрокові дії, такі як покупки та оплата рахунків.

Коли був винайдений протокол HTTP, ні автентифікація, ні керування сесіями не розглядалися, оскільки це протокол без статусу, тому використання цих двох функцій у міру розвитку Інтернету виявилося дуже складною ситуацією.

На жаль, автентифікація та керування сеансами пов'язані з уразливостями в багатьох веб-додатках. Інструменти та методи, що використовуються для їх, дещо відрізняються, але через тісний взаємозв'язок між автентифікацією та автентифікації та керування сеансами має сенс дослідити їх разом.

Атаки обходу шляху відбуваються, коли хакери можуть пройти через структуру каталогів веб-сервера. Це найчастіше трапляється, коли веб-додатки дозволяють завантажувати дані, і користувач (зловмисник) вводить шкідливе значення, яке обробляється веб-сервером і дозволяє отримати доступ до конфіденційних каталогів на веб-сервері.

Найпоширеніша атака на автентифікацію використовує інструмент атаки на основі проксі (наприклад, Intruder в Burp Suite), щоб перебрати облікові дані для входу в систему легітимного користувача.

У цьому типі атаки не так вже й багато методів для отримання даних, але вони дуже успішні, оскільки користувачі продовжують вибирати слабкі паролі. Для перевірки цього методу буде використовуватися Burp Intruder як інструмент, а також список найпоширеніших слабких паролів. Існує кілька аспектів автентифікації у веб-додатку, які необхідно враховувати для цих атак, а саме:

- Логін для входу в додаток
- Зміна пароля
- Секретні питання
- Передбачувані імена користувачів
- Передбачуваний початковий пароль
- Паролі, термін дії яких ніколи не закінчується

Термін "файл cookie" буде використовуватися у значенні "сеансовий файл cookie" або "ідентифікатор сеансу". Атаки на керування сеансами можливі лише у двох варіантах:

- атака на те, наскільки надійно згенеровано ідентифікатор сеансу (вимірювання ентропії)
- атака на те, як файл cookie використовується і обробляється веб-додатком.

Атакувати спосіб генерації файлів cookie дуже складно, оскільки більшість фреймворків управління сеансами, що йдуть в комплекті з веб-серверами, здатні створювати файли cookie, які дуже важко вгадати, навіть якщо зловмисник має

тонни обчислювальних потужностей, щоб згенерувати тисячі файлів cookie за короткий час.

Набагато більш застосовною атакою є дослідження того, як додаток використовує файли cookie. Цей тип атаки не вимагає розуміння того, як було створено файл cookie, а натомість зосереджується на доступі до файлу cookie та його зловмисному використанні.

Коли веб-сервер встановлений і налаштований, веб-додатку надається фрагмент файлової системи на веб-сервері, в якому додатку дозволено працювати. Ці дозволені каталоги, як правило, складаються з декількох папок в глибині файлової системи веб-сервера і включають в себе файлові системі веб-сервера і 100% того, що потрібно веб-додатку для роботи в звичайних умовах: код, зображення, база даних, таблиці стилів і все інше, що може знадобитися додатку.

Додаток ніколи не повинен намагатися отримати доступ до ресурсів, які знаходяться за межами визначених для нього каталогів, оскільки інші ресурси на веб-сервері не можуть бути застосовані до області застосування програми.

Здатність зловмисника вийти за межі цього обмеженого світу і отримати доступ до ресурсів на веб-сервері, до яких він не повинен мати доступу, є основною концепцією атаки обходу шляху.

Автентифікація насправді відбувається в багатьох інших частинах веб-додатку окрім головної сторінки входу в систему. Вона також присутня, коли користувач змінює пароль, оновлює інформацію про свій обліковий запис, використовує функцію для відновлення пароля, відповідає на секретні питання, а також коли використовує опцію "запам'ятати мене".

Якщо будь-який з цих процесів автентифікації несправний,то безпека всіх інших механізмів автентифікації може бути скомпрометована. Найстрашніше в уразливостях автентифікації те, що вони можуть відкрити двері для всіх інших облікових записів, які можуть бути скомпрометованими.

Для початку атаки, потрібно зрозуміти, які параметри надсилаються до додатку під час звичайної спроби автентифікації. Потрібно просто спробувати увійти до веб-застосунку, немає абсолютно ніякої різниці в тому, що було введено

в якості імені користувача і пароля. На рисунку 2.11 в звичайну форму для автентифікації були введені дані.

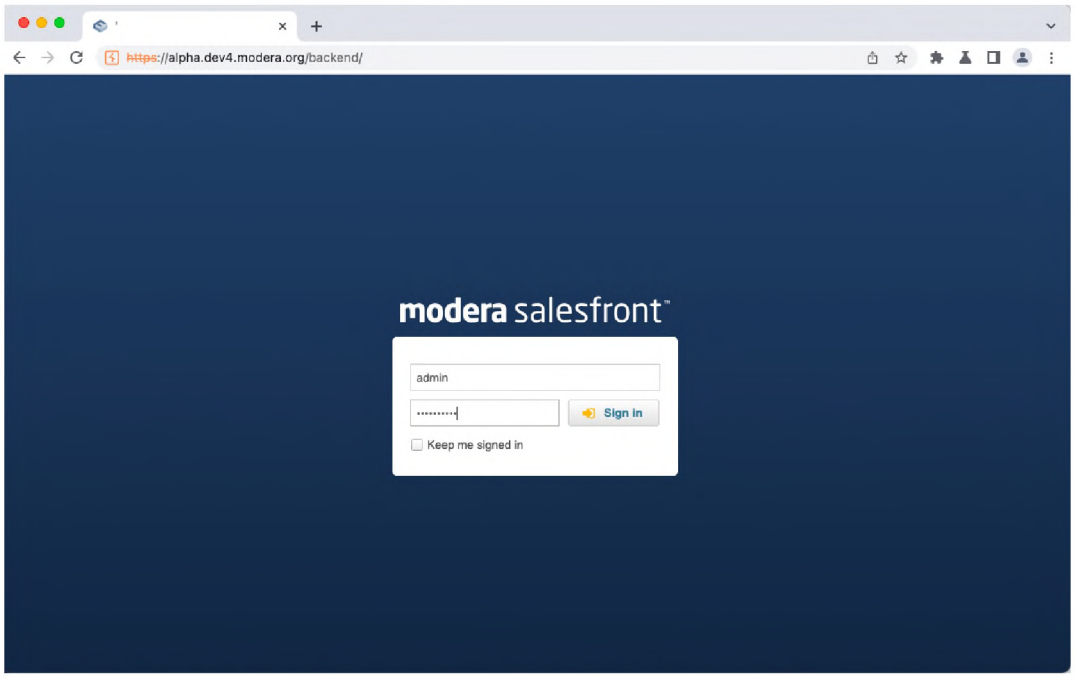


Рисунок 2.11 – Введення даних для автентифікації

Після того, як була надіслана спроба входу за допомогою кнопки “Sign in”, можна побачити параметри, які використовуються під час спроби автентифікації, як показано на рисунку 2.12.

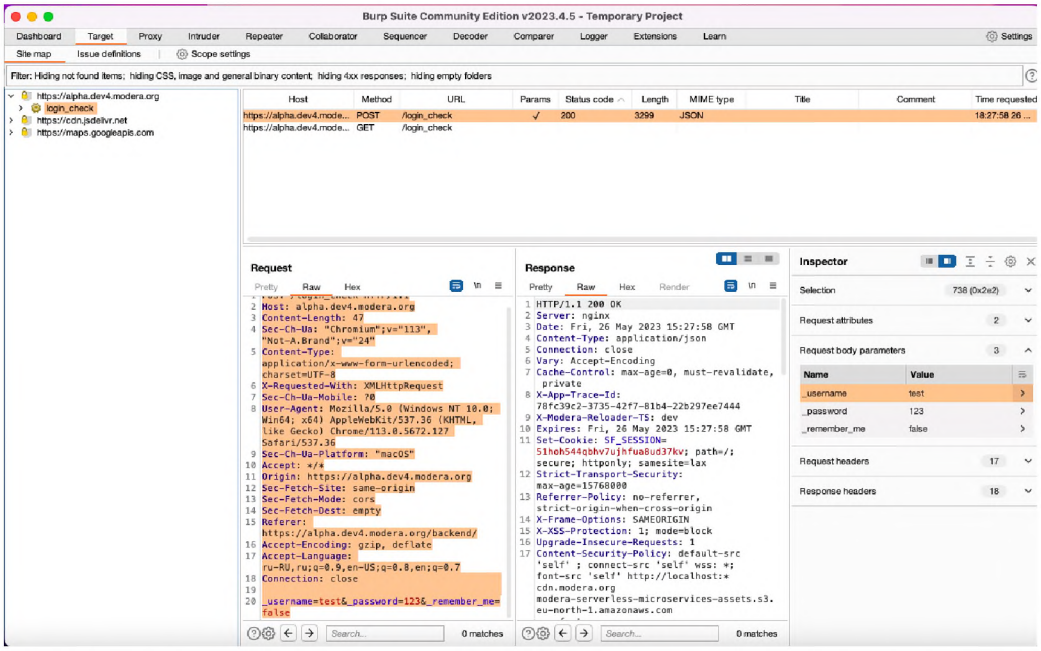


Рисунок 2.12 – Параметри при перехопленні (Burp Site)

На даний момент потрібні лише параметри імені користувача і пароля для цієї атаки; інші параметри залишаються без уваги. Наразі цілком очікувано, що ця

спроба входу буде невдалою. Єдина мета зараз - отримати дійсну спробу автентифікації в історії проксі, щоб можна було змінити значення параметрів для використання слабкого процесу автентифікації.

Можливі типи атаки, котрі може використати зловмисник:

- Снайперська - ця атака використовує один набір даних і націлена на кожний атрибут по черзі, перш ніж перейти до наступного значення. Це найбільш застосовне при нечіткому пошуку таких вразливостей, як міжсайтовий скриптинг (XSS).
- Battering Ram - ця атака також використовує один набір даних, але використовує одні і ті ж самі дані в усіх позначених параметрів одночасно. Це найчастіше застосовується, коли атака вимагає вставити один і той самий вхідний код в декілька місць наприклад, ім'я користувача одночасно в заголовок файлу cookie і в тіло повідомлення.
- Pitchfork- ця атака використовує кілька наборів даних для кожного параметра і виконує ітерацію по всіх наборах одночасно. Це найбільш застосовна атака коли атака вимагає використання пов'язаних значень в декількох параметрах в запиті, наприклад, параметр user_ID і відповідний параметр first_name. Атака буде просувати кожне паралельно таким чином, будуть виконані перші значення кожного набору даних, потім друге значення кожного з них, і так далі.
- Кластерна бомба - ця атака використовує декілька наборів даних, але різні набори для кожного параметра і повторюється для кожного набору по черзі щоб переконатися, що використовуються всі можливі комбінації. Ця атака найбільш застосовна коли атака вимагає використання різних вхідних даних в різних місцях запиту, таких як запиті, наприклад, ім'я користувача та пароль. Така атака зупиняється на першому наборі даних (наприклад, імені користувача) і перебирає всі паролі з цим першим іменем користувача. Після того, як усі значення паролів буде перебрано для першого імені користувача, ім'я користувача змінюється на друге ім'я користувача і весь список паролів буде використано з цим другим іменем користувача.

Очевидно, що для злому аутентифікації потрібно використовувати тип атаки кластерної бомби.

Набор даних - це значення, які потрібно перебрати під час грубого перебору.

Найпопулярніші сеансові атаки, які використовуються правопорушниками для використання вразливостей сеансів:

- Перехоплення сеансу: Це коли ідентифікатор сеансу користувача викрадається і використовується зловмисником, щоб привласнити особистість користувача. Викрадення ідентифікатора сеансу може бути виконана кількома різними способами, але найпоширенішим є XSS.
- Фіксація сеансу: Це коли зловмисник отримує дійсний ідентифікатор сеансу додатком, а потім передає цей сеанс несвідомому користувачеві. Зазвичай це робиться за допомогою веб-адреси, на яку користувач повинен перейти за посиланням. Після того, як користувач натискає на посилання і входить в додаток, зловмисник може використовувати той самий ідентифікатор сеансу, щоб авторизуватися, а система бачила нібито необхідного користувача. Ця атака також відбувається, коли веб-сервер приймає будь-яку сесію від користувача (або зловмисника) і не призначає новий сеанс після аутентифікації. У цьому випадку зловмисник використовує власну, заздалегідь обрану сесію, щоб відправити її жертві. Ці атаки працюють, тому що ідентифікатор сеансу можна використовувати повторно (або відтворюватися) в декількох сеансах.
- Пожертвування сеансу: Це дуже схоже на фіксацію сеансу, але замість того, щоб припускати особистість користувача, зловмисник передає ідентифікатор сеансу зловмисника користувачеві в надії, що користувач виконає дію несвідомо. Класичним прикладом є передача користувачеві дійсного ідентифікатора сеансу, який пов'язаний зі сторінкою профілю зловмисника, на якій немає жодної інформації. Коли користувач заповнює форму (з паролем, даними кредитної картки інформацією про кредитну картку та іншими реквізитами), ця інформація фактично прив'язується до акаунту зловмисника.

- Ідентифікатор сеансу в URL-адресі: Це коли ідентифікатори сеансу передаються в якості параметрів URL-адреси в якості параметрів URL під час циклу запиту та відповіді. Якщо ця функція присутня, зловмисник може передати таку URL-адресу користувачеві, щоб провести будь-яку з атак описаних вище.

Перевірити, наскільки надійно генеруються ідентифікатори сеансів, можна за допомогою Burp Sequencer, який перевіряє випадковість значень сеансів, де безпека програми залежить від непередбачуваності цих випадкових ідентифікаторів сеансу.

Burp Sequencer це дуже зручний інструмент, який виконує детальний аналіз зібраних ідентифікаторів сеансів і відображає результати у вигляді зрозумілих графіків і таблиць.

Burp Sequencer перевіряє гіпотезу ("ідентифікатор сеансу насправді згенерований випадковим чином") на основі колекції зібраних ідентифікаторів сеансів, щоб обчислити ймовірність фактичної випадковості. Він перевіряє, чи дійсно файл cookie сеансу є випадковим порівняно з безліччю інших сесійних файлів cookie. Якщо ця ймовірність падає нижче рівня значущості, ідентифікатор сеансу класифікується як невипадковий. За замовчуванням Sequencer використовує стандарт FIPS 0.0002-0.03% для значущості, але можна налаштувати цей вимір.

FIPS - це набір стандартів і рекомендацій для шифрування, аутентифікації, валідації і аудиту процесів у Сполучених Штатах та Канаді.

Кроки для проведення тесту за допомогою Burp Sequencer, тест і аналіз дуже прості у виконанні:

- Знайти запит в історії проксі-сервера, який містить ідентифікатор сеансу у відповіді. Цей ідентифікатор сесії і є тим, що потрібно протестувати і проаналізувати за допомогою Sequencer.
- Визначити ідентифікатор сеансу у Sequencer, якщо його не було визначено автоматично.
- Встановити у Sequencer потрібні параметри, наприклад, кількість потоків і швидкість запиту, щоб визначити швидкість, з якою будуть збиратися

ідентифікатори сеансів. Дуже важливо отримати ідентифікатори сеансів якомога швидше, не втрачаючи при цьому сеанси інших користувачів. Якщо можна отримати великий послідовний потік ідентифікаторів сеансів, тестування буде більш точнішим.

- Почати захоплення. Стандарт FIPS вимагає 20 000 ідентифікаторів сеансів, щоб бути надійними.
- Переглянути результати тестів на згенерованих діаграмах.

Результати тестування Sequencer можна переглянути з точки зору загального рівня значущості та на рівні бітів. Нижче наведено результати для різних рівнів значущості, де виявлено, що для рівня значущості 0,001% ентропія становить понад 130 біт ентропії для рівня значущості 0,001% (нижній стовпчик на діаграмі).

Ентропія – це міра непередбачуваності. Отже, чим вища ентропія в ідентифікаторах сеансів, тим більша впевненість, що вони згенеровані випадковим чином, як показано на рисунку 2.13.

Рисунок 2.13 – Результати ентропії для тестів

Для дотримання стандарту FIPS, результати на бітовому рівні є особливо важливими тому що користувач може перемикатися між кількома вкладками. Результати перевірки на бітовому рівні також можна дізнатися. Приклад на рисунку 2.14.

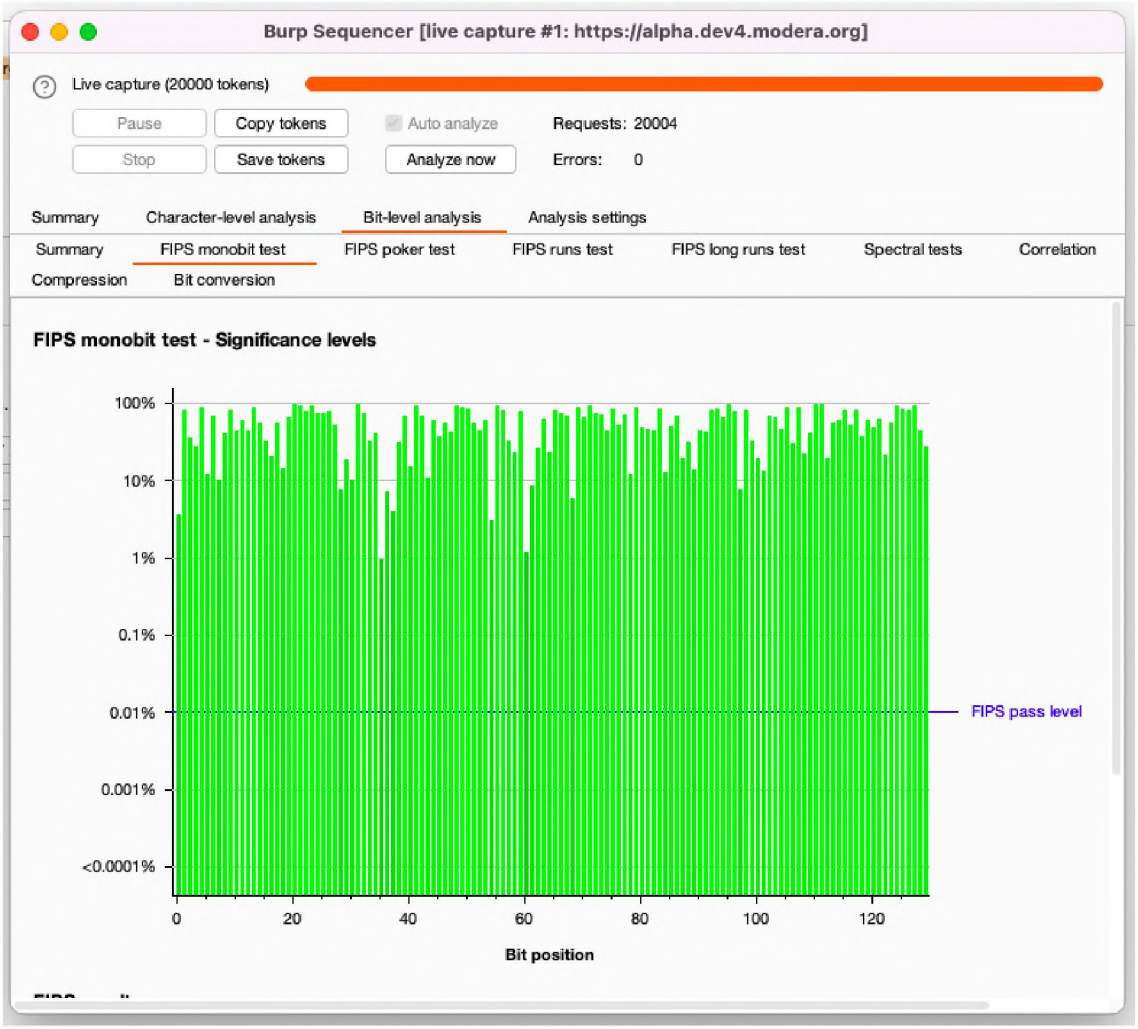


Рисунок 2.14 – Результат перевірки бітового рівня

Ефективні атаки на ідентифікатори сеансів базуються на концепції повторного використання файлів cookie.

Не має значення, кому було видано файл cookie, як порушник викрав цей файл, або як він планує його використовувати. Важливо лише те, що додаток може працювати зі старими файлами cookie, які використовуються більш ніж один раз.

Для перевірки потрібно виконати серію тестів додатка, отримавши дійсний ідентифікатор сеансу, щоб перевірити, чи не вразливий він до повторного використання файлів cookie. Для перевірки потрібно:

- Вийти із програми та оновити сторінку, щоб перевірити, чи користувач все ще можете отримати доступ до сторінки у веб-додатку, яка повинна вимагає активного сеансу.
- Скопіювати та вставити дійсний ідентифікатор сеансу і використати його знов після виходу з системи.
- Вийти з браузера або припинити користуватися ним на кілька годин, щоб перевірити ліміт часу очікування програми після того, як користувач отримали дійсний ідентифікатор сеансу.
- Скопіювати і вставити цей ідентифікатор сеансу в текстовий файл і потім увійти в систему.
- Порівняти ідентифікатор сеансу, який був виданий користувачу, коли він вперше відвідав сайт, та ідентифікатор сесії, який користувач отримав після успішної автентифікації. Вони повинні відрізнятися. Якщо це не так, то це велика уразливість, пов'язана з пожертвуванням сеансу.
- Увійти в один і той самий додаток з двох різних браузерів, щоб перевірити, чи підтримує додаток подвійний вхід. Та перевірити ідентифікатори сесії.

Атаки обходу шляху відбуваються, коли порушник намагається обійти будь-які засоби захисту та перевірки авторизації, встановлені адміністратором веб-сервера та командою веб-програмістів, котрі налаштували так веб-додаток, щоб усі користувачі веб-додатку перебували лише у вказаних каталогах.

Ці атаки часто виконуються аутентифікованими користувачами додатку; таким чином вони можуть повністю перевірити, до чого має доступ звичайний аутентифікований користувач, щоб краще сформувати зловмисний запит на посилання.

Спроба визначити, які параметри задіяні під час звичайного використання програми з гостьового облікового запису буде дуже складно. Цей метод атаки не доступний для дослідженого веб-додатку, так як додаток має для аутентифікований користувачів тільки можливість вводу даних для авторизації.

Ручне тестування може займати багато часу та ресурсу, особливо якщо веб-додаток, який тестується, має великий розмір.

В процесі тестування потрібно отримати повний список всіх сторінок сайту. Цей процес часто називають павутинним переглядом. Звичайно, можна самостійно переглянути всі сторінки сайту і занотувати їх, але набагато простіше зробити це за допомогою автоматизованих інструментів. Автоматизовані інструменти значно пришвидшують цей процес.

Існує низка програм або пакетів, які можуть виконати частину початкового аналізу. Серед них є як комерційні, так і інструменти з відкритим вихідним кодом. Хоча ці програми виконують по суті однакові завдання, вони виконують їх по-різному. Наприклад: nikto, OWASP ZAP, Burp Suite, Paros, W3AF. Далі робота буде поєднувати в собі два інструмента: OWASP ZAP та Burp Suite (для перевірки знайдених проблем)

Проект Open Web Application Security Project (OWASP) має інструмент тестування веб-додатків, який називається Zed Attack Proxy (ZAP) . ZAP працює як проксі-сервер, тобто перехоплює запити від веб-браузерів, які налаштовані на використання ZAP як веб-проксі-сервера, щоб можна було маніпулювати його запитами. Усі веб-запити з браузера будуть надсилатися на проксі, щоб бути перенаправлення на сервер, до якого має надійти запит.

ZAP працює не тільки з запитами, які були надіслані з веб-браузера. Він також може бути використаний для самостійного ініціювання запитів.

ZAP можна використовувати для перевірки відомих вразливостей, в тому числі і згаданих вище. Для цього він ініціює запити до веб-сервера, а потім аналізує відповіді.

На Рисунку 2.15 можна побачити деякі результати сканування в нижній частині вікна. Запити, в яких виявлено проблеми, позначені прапорцями.

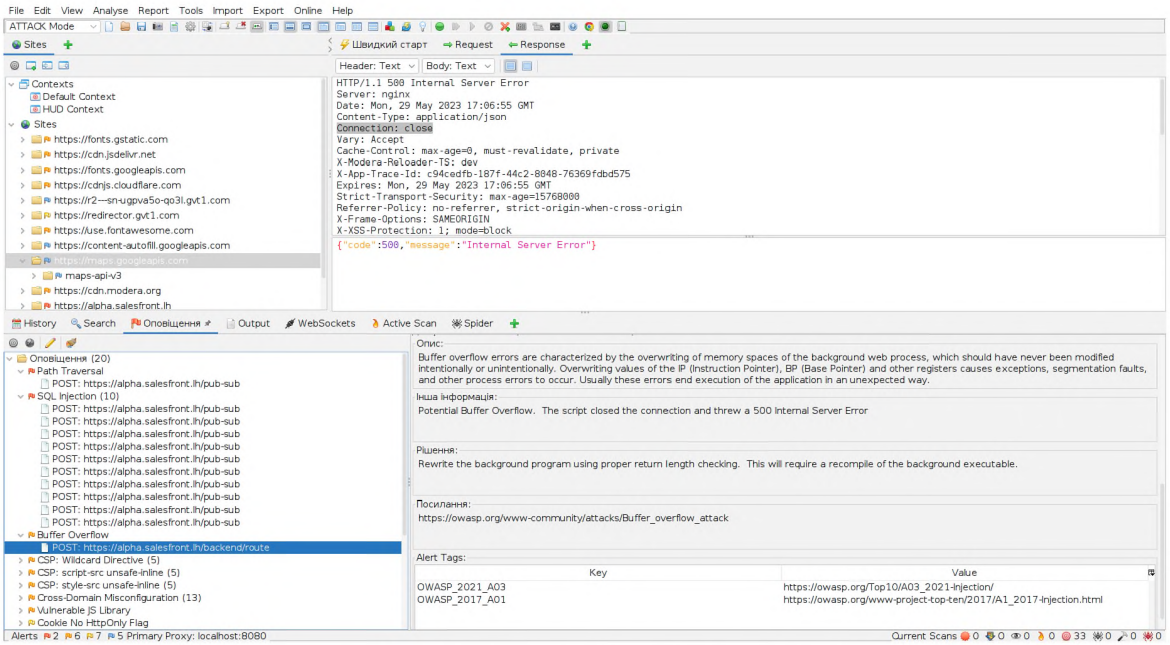


Рисунок 2.15 – Результат сканування ZAP

Щоб виявити вразливість, ZAP покладається на відповіді від веб-сервера, включаючи коди помилок і пошук певних слів у відповідях.

Деякі з показаних запитів позначені як проблеми критичного, середнього рівня ризику, а інші - як проблеми низького рівня ризику. Як і у випадку з автоматизованими сканерами вразливостей, такими як Nessus і Nexpose, потрібно буде вручну перевірити, чи те, що виявив ZAP, є дійсною вразливістю чи це псевдо помилка.

Для перевірки можна використати інструмент Burp Sequencer, котрий раніше вже використовувався. Наприклад, є помилка з приводу переповненого буфера обміну, котрий має середній рівень загрози. Спочатку потрібно скопіювати запит, який має вразливість з інструменту ZAP, та вставити його в Burp Sequencer. Як це зроблено на рисунку 2.16

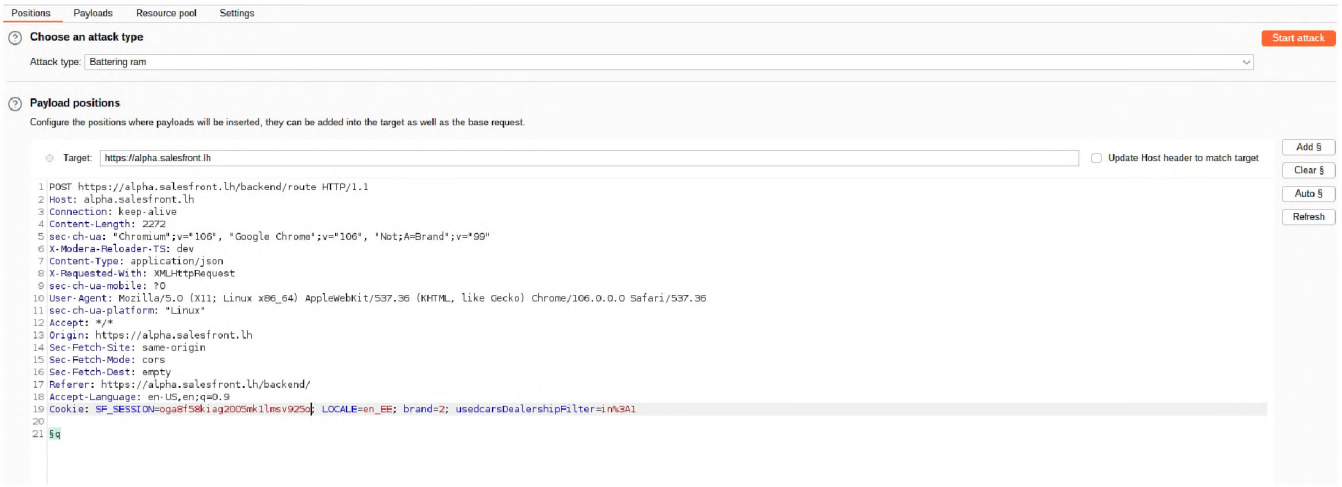


Рисунок 2.16 – Запит для перевірки знайденого ураження

Далі лишається тільки запустити атаку, та переконатися в результаті

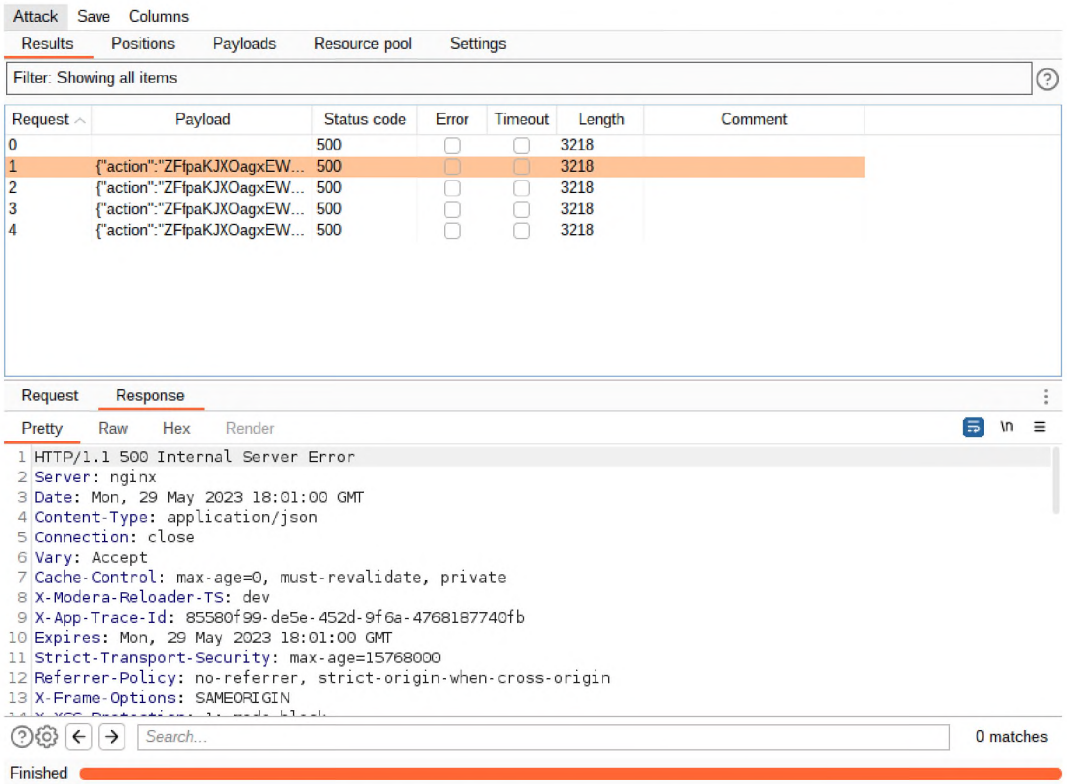


Рисунок 2.17 – Результат перевірки

З отриманих результатів видно, що проблема с буфером була виявлена, так як, при ручному відтворенні запиту помилка с кодом 500 залишилась.

2.4 Розробка комплексу заходів захисту від атак

Хоча і результат ентропії задовільний, але траплялись “аномалії” генерування токенів, для повного забезпечення контролю доступу потрібно

провести додатковий аналіз та провести впровадження нових методів захисту автентифікації.

Контроль доступу ефективний лише в надійному серверному коді або API, де зловмисник не може змінити перевірку контролю доступу або метадані.

- За винятком публічних ресурсів, потрібно забороняти доступ за замовчуванням.
- Реалізувати механізми контролю доступу один раз і повторно використовувати їх у всьому додатку, в тому числі мінімізувати використання Cross-Origin Resource Sharing (CORS).
- Модель управління доступом повинна забезпечувати право власності на записи, а не допускати, що користувач може створювати, читати, оновлювати або видаляти будь-який запис.
- Моделі доменів повинні забезпечувати дотримання вимог щодо обмежень для унікальних додатків.
- Вимкнути лістинг каталогів веб-серверів і переконатися, що метадані файлів (наприклад, .git) та файли резервних копій не знаходяться в кореневих каталогах.
- Реєструвати помилки контролю доступу, сповістити адміністраторів, коли це доречно (наприклад, повторювані помилки).
- Обмежити доступ до API та контролерів, щоб мінімізувати шкоду від автоматизованого інструментарію атак.
- Ідентифікатори сеансів зі станом повинні бути анульовані на сервері після виходу з системи. Токени JWT без статусу повинні бути недовговічними, щоб звести до мінімуму вікно можливостей для зловмисника. Для довговічних JWT потрібно дотримуватися стандартів OAuth для відкликання доступу.

Розробники та співробітники відділу контролю якості повинні передбачити функціональний блок контролю доступу та інтеграційні, регресійні тести.

Для уникнення ін’єкції, котрі були знайдені, потрібно зберігати дані окремо від команд і запитів:

- Кращим варіантом є використання безпечного API, який повністю уникає використання інтерпретатора, надає параметризований інтерфейс або мігрує до Object Relational Mapping Tools (ORM), навіть параметризовані збережені процедури можуть спричинити SQL-ін'єкції, якщо PL/SQL або T-SQL об'єднує запити і дані або виконує шкідливі дані за допомогою EXECUTE IMMEDIATE або exec();
- Використовувати позитивну перевірку вхідних даних на стороні сервера, це не є повним захистом, оскільки багато додатків вимагають спеціальних символів, наприклад, текстові області або API для мобільних додатків;
- Для будь-яких залишкових динамічних запитів екранізувати спеціальні символи, використовуючи спеціальний синтаксис екранування для цього інтерпретатора, структури SQL, такі як назви таблиць, стовпців тощо, не можна екранувати, тому назви структур, введені користувачем, є небезпечними;
- Використовувати LIMIT та інші елементи керування SQL у запитах, щоб запобігти масовому розкриттю записів у випадку SQL-ін'єкції.

Запобігання міжсайтовому скриптингу, як правило, може бути досягнуто за допомогою двох рівнів захисту:

- Кодування даних на виході
- Перевірка вхідних даних при надходженні

Кодування слід застосовувати безпосередньо перед тим, як дані, керовані користувачем, будуть записані на сторінку, оскільки контекст, в який записуються дані, визначає, яке кодування потрібно використовувати. Наприклад, значення всередині рядка JavaScript вимагають іншого типу екранування, ніж у контексті HTML.

Іноді потрібно застосувати кілька рівнів кодування у правильному порядку. Наприклад, щоб безпечно вбудувати користувацьке введення в обробник події, потрібно мати справу як з контекстом JavaScript, так і з контекстом HTML. Отже, потрібно спочатку перетворити ввід в Unicode, а потім закодувати його в HTML.

Кодування - це, найважливіша лінія захисту від XSS, але її недостатньо для запобігання XSS-вразливостей у будь-якому контексті. Також потрібно якомога суворіше перевіряти вхідні дані на етапі, коли вони вперше отримані від користувача.

Якщо користувач надсилає URL-адресу, яка буде повернута у відповідь, потрібно переконатися, що вона починається з безпечного протоколу, такого як HTTP і HTTPS. Інакше можливе використання сайту зі шкідливим протоколом, наприклад, javascript або data.

В ідеалі перевірка вхідних даних має працювати шляхом блокування невірних даних. Альтернативний підхід, який полягає у спробі очистити невірні дані, щоб зробити їх дійсними, є більш схильним до помилок, і його потрібно уникати, де це можливо.

Для перевірки вхідних даних, як правило, слід використовувати білі, а не чорні списки. Наприклад, замість того, щоб намагатися скласти список усіх шкідливих протоколів (javascript, data тощо), потрібно створити список безпечних протоколів (HTTP, HTTPS) і заборонити все, що не входить до цього списку. Це гарантує, що захист не зламається, коли з'являться нові шкідливі протоколи, і зробить його менш вразливим до атак, які намагаються завуалювати недійсні значення, щоб уникнути чорного списку.

2.5 Висновки

В спеціальному розділі було використано одну із методології для тестування на проникнення і вона показала себе дієвою. За час практичної роботи знайдено вразливості, та наведені приклади методів захисту системи. В проекті потрібно створити інфраструктуру для тестування на проникнення, або хоча б раз на пів року проводити аудит безпеки інформаційної системи.

Подальша робота може бути напрямлена на дослідження вразливостей, покращення процесу тестування, а також можлива інтеграція систем для автоматизованого тестування.

3 ЕКОНОМІЧНА ЧАСТИНА

Метою цього розділу є визначення економічної доцільності розробки комплексу заходів захисту наведених у спеціальному розділі та інтеграція з автоматизованими системами для моніторингу рівня інформаційної безпеки в системі. Для досягнення цією мети необхідно здійснити розрахунок капітальних витрат на впровадження комплексу заходів захисту та експлуатаційних витрат на реалізацію інтеграцій; економічного ефекту від впровадження інтеграцій, методів та розрахунку показників економічної ефективності, зокрема коефіцієнту повернення інвестицій та періоду окупності.

3.1 Розрахунок (фіксованих) капітальних витрат

До капітальних витрат належать витрати на розробку заходів із забезпечення інформаційної безпеки, а також витрати на придбання матеріальних та нематеріальних активів.

Витрати на впровадження методів захисту інформації на підприємстві визначаються, виходячи з трудомісткості кожної операції, та належать до капітальних витрат.

Визначення трудомісткості впровадження нових методів захисту та виправлення існуючих загроз.

Трудомісткість впровадження нових методів захисту та виправлення існуючих загроз на підприємстві визначається тривалістю кожної робочої операції, до яких належать наступні:

де $t_{ТЗ}$ – тривалість складання технічного завдання для впровадження методів захисту, $t_{ТЗ}$=70;

$t_В$ – тривалість аналізу існуючих інформаційних потоків організації, вивчення ТЗ, літературних джерел за темою тощо, $t_В$ =50;

$t_А$ – тривалість аналізу існуючих загроз безпеки інформації, $t_А$ =78;

$t_Р$ – тривалість безпечного впровадження методів захисту для підприємства, $t_М$=100;

$t_Д$ – тривалість підготовки технічної документації, $t_Д$=36.

Отже,

$$t = t_{ТЗ} + t_В + t_А + t_Р + t_Д = 70+50+78+100+36= 334 \text{ години.}$$

Розрахунок витрат на впровадження методів захисту інформації

Витрати на впровадження методів захисту інформації на підприємстві $К_{РП}$ складаються з витрат на заробітну плату спеціаліста з інформаційної безпеки $З_{ЗП}$ і вартості витрат машинного часу $З_{МЧ}$.

$$К_{РП} = З_{ЗП} + З_{МЧ}.$$

$$К_{РП} = З_{ЗП} + З_{МЧ} = 60788,78+4500,32=65289,1 \text{ грн.}$$

$$З_{ЗП} = t\ З_{ПР} = 1519*334 = 507346 \text{ грн.},$$

де t – загальна тривалість операцій, годин;

$З_{пр}$ – средньогодинна заробітна плата спеціаліста з інформаційної безпеки з нарахуваннями, грн/годину.

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

$$З_{мч} = t*С_{мч}=112*8,42=943,04 \text{ грн.},$$

де t – трудомісткість операцій на ПК для впровадження методів захисту, годин;

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

Вартість 1 години машинного часу ПК визначається за формулою:

$$С_{мч} = P * t_{нал} * С_е + \frac{Ф_{зал}+Н_а}{F_р} + \frac{К_{лпз} * Н_{апз}}{F_р}, \text{ грн.},$$

де:

Р – встановлена потужність ПК, кВт;

$С_е$ – тариф на електричну енергію, грн/кВТ*година;

$Ф_{зал}$ – залишкова вартість ПК на поточний рік, грн;

$Н_а$ – річна норма амортизації на ПК, частки одиниці;

$Н_{апз}$ – річна норма амортизації на ліцензійне програмне забезпечення частки одиниці;

$К_{лпз}$ – вартість ліцензійного програмного забезпечення, грн;

$F_р$ – річний фонд робочого часу (за 40-годинного робочого тижня це 1920)

Залишкова вартість ПК визначається виходячи з фактору терміну його експлуатації як різниця між первісною вартістю та зносом під час використання

$$С_{мч} = 0,9 * 5 * 1,44 + \frac{7000*0,3}{1920} + \frac{3200*0,5}{1900} = 8,42 \text{ грн.}$$

Для впровадження методів захисту можна використовувати стандартне обладнання, яке вже наявне на підприємстві, тому капітальні витрати не виникають.

Витрати на встановлення обладнання та налагодження системи інформаційної безпеки становитимуть 7300 грн.

Вирішення певних технічних завдань із збільшенням продуктивності та створення середовища для тестування та аналізу вразливостей потрібно скористатися послугами аутсорсингових організації, вартість послуг котрих складає 33000 грн.

Капітальні (фіксовані) витрати на проектування та впровадження проектного варіанту комплексу для підвищення інформаційної безпеки складають:

$$К = К_{рп} + К_{зпз} + К_{пз} + К_{аз} + К_{навч} + К_{н},$$

де:

$К_{рп}$ – вартість розробки проекту інформаційної безпеки та залучення для цього зовнішніх консультантів, тис. грн;

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

$К_{пз}$ – вартість створення основного й додаткового програмного забезпечення, тис. грн;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів, тис. грн;

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

$К_{н}$ – витрати на встановлення обладнання та налагодження системи інформаційної безпеки.

Отже, капітальні (фіксовані) витрати на впровадження комплексу захисту для підвищення рівня інформаційної безпеки складатимуть:

$$К = 65289,1 + 33000 + 7300 = 105589,1 \text{ грн.}$$

Річні поточні витрати на функціонування системи інформаційної безпеки складають:

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

де:

$С_в$ - вартість відновлення й модернізації системи;

$С_к$ - витрати на керування системою в цілому;

$С_{ак}$ - витрати, викликані активністю користувачів системи інформаційної безпеки.

Оскільки для впровадження нових методів захисту можливо використання старої інформаційної інфраструктури витрати на відновлення й модернізацію системи не матимуть місце.

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

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

Витрати на навчання адміністративного персоналу й кінцевих користувачів визначаються сукупною величиною витрат на організаційні заходи, яка складає 8500 грн.

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

$$С_з = З_{осн} + З_{дод}, \text{ грн.}$$

Основна заробітна плата визначається, виходячи з місячного посадового окладу, а додаткова заробітна плата – в розмірі 5-8% від основної заробітної плати.

Основна заробітна плата одного спеціаліста з інформаційної безпеки на місяць складає 60788,78 грн. Додаткова заробітна плата – 8% від основної заробітної плати. Виконання роботи щодо впровадження системи захисту інформації на підприємстві потребує залучення спеціаліста інформаційної безпеки на 0,3 ставки. Отже,

$$С_з = (60788,78*12 + 60788,78*12*0,08)*0,3= 236346,77 \text{ грн.}$$

Ставка ЄСВ для всіх категорій платників з 01.01.2021 р. складає 22%.

$$С_{єв} = 236346,77 * 0,22 = 51996,28 \text{ грн.}$$

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року ($С_{ел}$), визначається за формулою:

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

де:

Р – встановлена потужність апаратури інформаційної безпеки, (Р=0,7 кВт);

$F_р$ – річний фонд робочого часу системи інформаційної безпеки ($F_р$ = 1920 год.);

$Ц_е$ – тариф на електроенергію, ($Ц_е$ = 1,44 грн./кВт за годину).

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

$$С_{ел} = 0,7 * 1920 * 5 * 1,44 = 9676,8 \text{ грн.}$$

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки визначаються у відсотках від вартості капітальних витрат – 3%:

$$С_{тос} = 105589,1 * 0,03 = 3167,67 \text{ грн.}$$

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

$$С_к = 8500 + 236346,77 + 51996,28 + 9676,8 + 3167,67 = 309687,52 \text{ грн.}$$

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

$$С = 309687,52 \text{ грн.}$$

3.2 Оцінка можливого збитку

Для розрахунку вартості збитку можна застосувати наступну спрощену модель оцінки:

$t_П$ – час простою вузла або сегмента корпоративної мережі в наслідок атаки, 3 години;

$t_В$ – час відновлення системи після атаки персоналом, що обслуговую корпоративну мережу, 2 години;

$t_{ВИ}$ – час повторного введення загубленої інформації співробітниками атакованого вузла або сегмента корпоративної мережі, 5 годин;

$З_О$ – заробітна плата обслуговуючого персоналу (адміністраторів та ін.), 40000 грн. на місяць;

$З_С$ – заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі, 20000 грн на місяць;

$Ч_О$ – чисельність обслуговуючого персоналу (адміністраторів та ін.), 3 особи;

$Ч_С$ – чисельність співробітників атакованого вузла або сегмента корпоративної мережі, 5 осіб.;

О – обсяг продажів атакованого вузла або сегмента корпоративної мережі, 2 млн. 350 тис. грн. у рік;

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

I – число атакованих сегментів корпоративної мережі, 6;

N – середнє число атак на рік, 10.

Упущена вигода від простою атакованого вузла або сегмента корпоративної мережі становить:

$$U = П_П + П_В + V,$$

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:

$$П_П = \frac{\sum З_С}{F} * t_П = \frac{20000 * 5}{176} * 3 = 1704{,}54 \text{ грн.},$$

де:

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

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

$$П_В = П_{ВИ} + П_{ПВ} + П_{зч},$$

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 $t_{ВИ}$:

$$П_{ВИ} = \frac{\sum З_С}{F} * t_{ВИ} = \frac{20000 * 5}{176} * 5 = 2840{,}90 \text{ грн},$$

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

$$П_{ПВ} = \frac{\sum З_О}{F} * t_В = \frac{40000 * 2}{176} * 2 = 909{,}09 \text{ грн},$$

Витрати на заміни встаткування або запасних частин можуть скласти 3256 грн.

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі складуть

$$П_В = 2840{,}9 + 909{,}09 + 3256 = 7006 \text{ грн}.$$

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

$$V = \frac{O}{F_Г} * (t_П + t_В + t_{ВИ})$$

$$V = \frac{2300000}{2080} * (3 + 2 + 5) = 11057{,}69 \text{ грн.},$$

де:

$F_Г$ - річний фонд часу роботи організації (52 робочих тижні, 5-ти денний робочий тиждень, 8-ми годинний робочий день) становить близько 2080 ч.

U = 1704,54 + 7006 + 11057,69 = 19768,23 грн.

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

$$B = \Sigma_I \Sigma_N U = \Sigma_6 \Sigma_9 11057{,}69 = 597115{,}26 \text{ грн.}$$

Загальний ефект від впровадження системи ідентифікації об'єктів

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

E = B * R – C грн.,

де:

B – загальний збиток від атаки у разі перехоплення інформації, 597115,26 грн.;

R – вірогідність успішної реалізації загрози (60%);

C – щорічні витрати на експлуатацію системи інформаційної безпеки.

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

E = 597115,26*0,6 – 309687,52 = 48581,63 грн.

3.3 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки

Для встановлення економічної ефективності визначають такі показники як: коефіцієнт повернення інвестицій (ROSI) та термін окупності капітальних інвестицій (T_o).

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

$$ROSI = \frac{E}{K}, \text{ частки одиниці,}$$

де:

Е - загальний ефект від впровадження системи інформаційної безпеки грн.;

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

Коефіцієнт повернення інвестицій ROSI:

$$ROSI = \frac{48581,63}{105589,1} = 0,46, \text{ частки одиниці,}$$

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

$$\text{ROSI} > (N_{ДЕП} - N_{ІНФ})/100),$$

де:

$N_{ДЕП}$ – річна депозитна ставка, (12%);

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

Розрахункове значення коефіцієнта повернення інвестицій:

$$0,46 > (12 - 5)/100 = 0,46 > 0,07.$$

Термін окупності капітальних інвестицій T_o показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки:

$$T_O = \frac{K}{E} = \frac{1}{ROSI} = \frac{1}{0,46} = 2,17 \text{ років (27 місяців).}$$

3.4 Висновок

Впровадження запропонованих методів захисту інформації можна вважати економічно доцільними, оскільки значення коефіцієнту повернення інвестицій ROSI складає 0,46 при величині економічного ефекту 48581,63 грн.

Отримане значення коефіцієнту ROSI перевищує дохідність альтернативного вкладення коштів. Термін окупності складає 2,17 років (приблизно 27 місяців). Щорічні експлуатаційні витрати становлять 309687,52 грн, а капітальні витрати на впровадження методів захисту складуть в 105589,10 грн.

Величина економічного ефекту може бути значно більшою, якщо запропоновані методи захисту будуть розповсюджуватися і на інші системи та публічні інтеграції підприємства.

ВИСНОВКИ

Метою кваліфікаційної роботи є підвищення рівня захисту інформації в інформаційній системі ТОВ «МОДЕРА РОЗВИТОК УКРАЇНА».

Проведено аналіз діяльності підприємства, особового складу, методології тестування, можливі загрози для інформаційної безпеки. Показано, що існує потреба в проведенні тестування на проникнення.

В результаті тестування інформаційної системи встановлено, що існують вразливості, що можуть викликати втрату конфіденційних даних та можливість порушити цілісність інформаційною системи.

Запропоновано удосконалений комплекс захисту від атак, який дозволяє виправити існуючи та уникнути майбутні вразливості. Удосконалений комплекс захисту поєднує в собі декілька методологій, але за основну – OWASP. В комплексі впроваджені методи для захисту від ін’єкцій та покращення контролю доступу для користувачів.

ПЕРЕЛІК ПОСИЛАНЬ

1 Методичні рекомендації до виконання кваліфікаційних робіт бакалаврів спеціальності 125 кібербезпека / Упоряд.: О.А. Герасіна, Д.С. Тимофеєв, О.В. Кручінін, Ю.А. Мілінчук – Дніпро: НТУ «ДП» 2020.

2 Методичні рекомендації до економічної частини дипломного проекту зі спеціальності 125 кібербезпека / Упоряд.: Д.П. Пілова – Дніпро: НТУ «ДП» 2019.

3 Про основні засади забезпечення кібербезпеки України: Закон України від 05.10.17 р. URL: http://zakon.rada.gov.ua/laws/show/2163-19

4 Про захист персональних даних: Закон України від 2010 р. URL: http://zakon.rada. gov.ua/laws/show/2297-17

5 Про захист інформації в інформаційно-комунікаційних системах від 01.07.2022 р. URL: https://zakon.rada.gov.ua/laws/show/80/94-вр

6 Захист інформації. Технічний захист інформації. Порядок проведення робіт: ДСТУ 3396.1-96. Чинний від 1997-01-01. К. : ДСТСЗІ СБ України, 1996. 15 с.

7 Захист інформації. Технічний захист інформації. Терміни та визначення: ДСТУ 3396.2-96. Чинний від 1997-04-11. К.: ДСТСЗІ СБ України, 1996. 19 с.

8 Melnick J. Top 10 Most Common Types of Cyber Attacks [Електронний ресурс] / Jeff Melnick. – 2018. – Режим доступу до ресурсу: https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/

9 Офіційний сайт проекту Open Web Application Security Project [Електронний ресурс]. – Режим доступу: https://owasp.org/about/

10 Рейтинг вразливостей OWASP Top – 10 – 2021 Електронний ресурс]. – Режим доступу: https://owasp.org/Top10/

11 Veracode manual penetration testing [Електронний ресурс]. – Режим доступу: https://www.veracode.com/services/penetration-testing

12 Статья «Внедрение SQL кода» [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Внедрение_SQL-кода

13 Broken Authentication [Електронний ресурс]. – Режим доступу: https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

14 Kotowicz K. XSS ChEF: Chrome Extension Exploitation Framework [Електронний ресурс] / Krzysztof Kotowicz // IBM. – 2012. – Режим доступу до ресурсу: https://dzone.com/articles/xss-chef-chrome-extension

15 Sankar R. Burpsuite – A Beginner's Guide For Web Application Security or Penetration Testing [Електронний ресурс] / Ravi Sankar. – 2018. – Режим доступу до ресурсу: https://kalilinuxtutorials.com/burpsuite/

16 Charan H. Broken Authentication and Session Management—part □ [Електронний ресурс] / Hari Charan. – 2017. – Режим доступу до ресурсу: https://medium.com/@grep_security/broken-authentication-and-session-management-part-□-50e760c9f599.

17 Brewer J. Web Server Vulnerabilities and a Defense in Depth Strategy Using the Squid Proxy [Електронний ресурс] / Jim Brewer // GSEC Practical version 1.4b. – 2004. – Режим доступу до ресурсу: https://www.giac.org/paper/gsec/3729/web-server-vulnerabilities-defense-in-depthstrategy-squid-proxy/105970.

18 A03:2021 - Injection [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://owasp.org/Top10/A03_2021-Injection/.

19 Довідкове керівництво сканера nmap [Електронний ресурс]. – Режим доступу: https://nmap.org/man/ru/index.html

20 A10:2021 – Server-Side Request Forgery (SSRF) Injection [Електронний ресурс]. – 2021 - Режим доступу до ресурсу: https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/

21 Getting Started OWASP ZAP [Електронний ресурс] -]. – Режим доступу до русурсу: https://www.zaproxy.org/getting-started/

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	**Формат**	**Найменування**	**Кількість листів**	**Примітки**
Документація				
1	А4	Реферат	2	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	2	
4	А4	Вступ	1	
5	А4	Стан питання. Постановка задачі	17	
6	А4	Спеціальна частина	34	
7	А4	Економічний розділ	10	
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 Ґ	1	

ДОДАТОК Б. Перелік документів на оптичному носії

1 Презентація_Диплом_Сустрєтов_125-19-2.pptx

2 Диплом_Сустрєтов_125-19-2.docx

3 Диплом_Сустрєтов_125-19-2.pdf

4 Рецензія_Сустрєтова_І_О.pdf

ДОДАТОК В. Відгук керівника економічного розділу

Економічний розділ виконаний відповідно до вимог, які ставляться до кваліфікаційних робіт, та заслуговує на оцінку 90 б. («відмінно»).

Керівник розділу _____ Пілова Д.П.

(підпис) (прізвище, ініціали)

ДОДАТОК Г. Відгук керівника кваліфікаційної роботи

В І Д Г У К

на кваліфікаційну роботу студента групи 125-19-2 Сустрєтова І.О. на тему: «Аналіз вразливостей корпоративних інформаційних систем в ТОВ "МОДЕРА РОЗВИТОК УКРАЇНА" з використанням автоматизованих інструментів»

Пояснювальна записка складається зі вступу, трьох розділів і висновків, розташованих на 74 сторінках.

Мета роботи є актуальною, оскільки вона спрямована на удосконалення комплексу заходів захисту для інформаційної системи.

При виконанні роботи автор продемонстрував добрий рівень теоретичних знань і практичних навичок. На основі аналізу принципів розробки комплексу заходів захисту сформульовано задачі, вирішенню яких присвячений спеціальний розділ. У ньому було запропоновано удосконалений комплекс заходів захисту, який забезпечує виправлення вразливостей та підвищення рівня захисту системи.

Практична цінність роботи полягає у тому, що запропонований комплекс заходів захисту може бути використаний в інших інформаційних системах та не потребує великих ресурсів для впровадження його.

До недоліків роботи слід віднести недостатню проробку окремих питань.

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

В цілому робота задовольняє усім вимогам, а її автор Сустрєтов І.О. заслуговує на оцінку «добре» та присвоєння кваліфікації «Бакалавр з кібербезпеки» за спеціальністю 125 Кібербезпека.

Керівник роботи,

ст. викл. **В.О. Святошенко**

ДОДАТОК Ґ. Перевірка на запозичення

UNICHECK by Turnitin

Имя пользователя: Володимир Святошенко
ID проверки: 1015613625
Дата проверки: 15.06.2023 14:03:14 EEST
Тип проверки: Doc vs Internet + Library
Дата отчета: 15.06.2023 21:47:17 EEST
ID пользователя: 100008751

Название файла: Сустрєтов_І_О_125-19-2_Диплом (1)

Количество страниц: 74 **Количество слов:** 12739 **Количество символов:** 97731 **Размер файла:** 3.51 MB **ID файла:** 1015261261

16.5% Совпадения

Наибольшее совпадение: 7.08% с Интернет-источником (http://ir.nmu.org.ua/handle/123456789/156384)

15.8% Источники из Интернета 844 ..... Страница 76

9.69% Источники из Библиотеки 67 ..... Страница 84

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

0% Исключений

Нет исключенных источников

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 49