МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут автоматики і електротехніки

Кафедра комп'ютерних технологій та інформаційної безпеки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних технологій та інформаційної безпеки, к.т.н., доцент
_____ С.М. Нужний
«____» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

РОЗРОБКА ОРГАНІЗАЦІЙНО-ТЕХНІЧНИХ ЗАХОДІВ ДЛЯ ПРОВЕДЕННЯ ХАКАТОНУ З КІБЕРБЕЗПЕКИ

Ступінь вищої освіти: магістр
Галузь знань: 14 Електрична інженерія
Спеціальність: 141 «Електроенергетика, електротехніка та електромеханіка»
Освітньо-професійна програма: Системи технічного захисту інформації, автоматизація її обробки

Виконала: студентка 6 курсу групи 6366мз
Маркевич Вікторія Ігорівна _____

Кваліфікаційна робота містить результати самостійного виконання навчального завдання. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Керівник:
к.т.н., доцент кафедри комп'ютерних технологій та інформаційної безпеки

Сірівчук Андрій Сергійович _____

Миколаїв – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут автоматики і електротехніки

ЗАТВЕРДЖУЮ
Гарант освітньої програми, д.т.н., професор,
професор кафедри КТІБ
_____ Передерій В.І.
«14» 10 2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу

Студентка: Маркевич Вікторія Ігорівна

Курс:6

Група: 6366мз

Ступінь вищої освіти: магістр

Галузь знань: 14 Електрична інженерія

Спеціальність: 141 «Електроенергетика, електротехніка та електромеханіка»

Освітньо-професійна програма: Системи технічного захисту інформації, автоматизація її обробки

1. Тема роботи: Розробка організаційно-технічних заходів для проведення хакатону з кібербезпеки

затверджена наказом НУК від «14» 10 2025 р. № 1217-ук

2. Вихідні дані до роботи: тип хакатону – CTF; тематики що виносяться на хакатон – аналіз мережевого трафіку, аналіз трафіку HTTP, вразливість веб-застосунків

3. Перелік питань, які необхідно розробити: визначити типи та етапи провдення хакатону; провести аналіз актуальних тем для проведення хакатону; провести аналіз платформ для проведення хакатону; скласти інструкцію з підготовки завдань для заданих тематик; розгорнути обрану систему на сервері кафедри.

4. Терміни виконання завдання:
термін видачі завдання: «01- 05» вересня 2025 р.
термін подання роботи: «18» грудня 2025 р.

6. План-графік виконання кваліфікаційної роботи

п/п	Заходи	Дата		Готовність
		Навч. тиждень	Контрольна дата	
1.	Наукове стажування	1 - 8	27.10.2025	Звіт з наукового стажування
2.	Організаційні збори	9	29.10.2025	Попередній варіант змісту КР
3.	Рубіжний контроль	10	05.11.2025	Попередній варіант оглядових розділів, звіт з практика
4.	Рубіжний контроль	13	27-28.11.2025	Доповідь на 13-ій Всеукраїнській науково-технічній конференції з міжнародною участю «СПІБТ–2025»
5.	Рубіжний контроль	14	3.12.2025	1. Попередній варіант повної КР 2. Попередній варіант презентації
6.	КЕ на рівень «магістра»	15	8,9.12.2025	Складання державного екзамену зі спеціальності на ступінь магістра
7.	Перевірка на плагіат	15	10.12.2025	Електронний варіант кваліфікаційної роботи, попередній захист (робота передається студентом на кафедру для внутрішньої рецензії).
8.	Оформлення КР та супутньої документації	16	15-17.12.2025	1. Оформлена КР (в форматі pdf) У разі отримання позитивної внутрішньої рецензії, КР подається на зовнішню рецензію. 2. Оформлена презентація (в форматі pdf).
9.	Подання документів на захист	16	18.12.2025	1. Подання щодо захисту КР: тема, успішність, висновок керівника та висновок кафедри про КР. 2. Зовнішня рецензія (окремо від КР). Резюме на КР. 3. Електронний варіант презентації. 4. Електронний варіант КР на диску
10.	Захист ДР	17	22,23,24 12.2025	1. Приєднання студента до засідання кваліфікаційної комісії (конференції) у встановлений час для проведення процедури дистанційного захисту 2. Процедура захисту КР.

Керівник
к.т.н., доцент кафедри комп'ютерних технологій
та інформаційної безпеки, _____ А.С. Сірівчук

Студентка _____ В.І. Маркевич

АНОТАЦІЯ

Маркевич В.І. Розробка організаційно-технічних заходів для проведення хакатону з кібербезпеки. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття ступеня вищої освіти «магістр» за спеціальністю 141 «Електроенергетика, електротехніка та електромеханіка» галузі знань 14 «Електрична інженерія» освітньо-професійної програми «Системи технічного захисту інформації, автоматизація її обробки». – Національний університет кораблебудування імені адмірала Макарова, Миколаїв, 2025.

Пояснювальна записка: 66 с., 35 джерел.

Кваліфікаційна робота присвячена актуальній темі підвищенні кваліфікації в сфері кіберзахисту шляхом проведення спеціалізованих хакатонів, зокрема в форматі CTF.

Метою даної роботи є визначення основних завдань для проведення хакатону та виконання підґрунтя для організації хакатону в форматі CTF.

В роботі представлено етапи проведення хакатону та перспективи використання формату CTF. Перспективними тематиками для проведення змагань визначено аналіз мережевого трафіку та виявлення вразливостей в веб-застосунках.

Також було проведено аналіз платформ для проведення змагань в форматі CTF. Для проведення хакатонів було розгорнуто платформу RootTheBox.

Ключові слова: хакатон, CTF, RootTheBox, вразливість веб-сайтів, аналіз трафіку.

ANNOTATION

Markevich V.I. Development of organizational and technical measures for conducting a hackathon on cybersecurity. – Qualification work in the form of a manuscript.

Qualification work for obtaining a higher education degree “Master” in specialty 141 “Electrical power engineering, electrical engineering and electromechanics” field of knowledge 14 “Electrical engineering” of the educational and professional program “Systems of technical protection of information, automation of its processing”. – Admiral Makarov National University of Shipbuilding, Mykolaiv, 2025.

Explanatory note: 66 p., 35 sources.

The qualification work is devoted to the topical topic of improving skills in the field of cyber security by conducting specialized hackathons, in particular in the CTF format.

The purpose of this work is to determine the main tasks for conducting a hackathon and to lay the groundwork for organizing a hackathon in the CTF format.

The work presents the stages of the hackathon and the prospects for using the CTF format. Network traffic analysis and vulnerability detection in web applications were identified as promising topics for the competition.

An analysis of platforms for holding CTF competitions was also conducted. The RootTheBox platform was deployed for hackathons.

Keywords: hackathon, CTF, RootTheBox, website vulnerability, traffic analysis.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

CTF – Capture The Flag

NTA – Network Traffic Analysis

HTTP – протокол передачі гіпертексту

FTP – File Transfer Protocol

SMB – Server Message Block

RTB – RootTheBox

ВСТУП

Зростання кількості кібератак та ускладнення інформаційних систем зумовлюють підвищені вимоги до рівня підготовки фахівців з інформаційної безпеки. Традиційні теоретичні методи навчання не завжди дозволяють сформувати практичні навички, необхідні для виявлення та усунення вразливостей у реальних умовах. У зв'язку з цим особливої актуальності набувають хакатони та змагання у форматі *Capture The Flag* (*CTF*).

Формат *CTF* дозволяє поєднати теоретичні знання з практичним досвідом, моделюючи реальні сценарії виявлення та експлуатації вразливостей, а також сприяє розвитку аналітичного мислення, командної роботи та навичок прийняття рішень в умовах обмеженого часу, що робить такі заходи ефективним інструментом сучасної технічної освіти.

Об'єкт дослідження. Процес організації та проведення хакатонів у форматі *CTF* у навчальному середовищі з інформаційної безпеки.

Предмет дослідження. Методи, інструменти та програмні засоби підготовки ігрових завдань та керування змаганнями у форматі *CTF* на основі платформи *RootTheBox*.

Метою кваліфікаційної роботи є визначення основних завдань для проведення хакатону та виконання підґрунтя для організації хакатону в форматі *CTF*.

Для вирішення поставленої мети вирішено наступні завдання:

– визначити типи та етапи провдення хакатону;

– провести аналіз актуальних тем для проведення хакатону;

– провести аналіз платформ для проведення хакатону;

– скласти інструкцію з підготовки завдань для заданих тематик;

– розгорнути обрану систему на сервері кафедри.

Результати роботи частково опубліковані в наступному джерелі:

Маркевич В.І., Ковальчук К.О., Аналіз етапів проведення хакатону / Наук. керівник А.С. Сірівчук // Сучасні проблеми інформаційної безпеки на транспорті: Матеріали XIII всеукраїнської науково-технічної конференції з міжнародною участю. – Миколаїв: НУК, 2025, 66-69 сс.

Кваліфікаційна робота відповідає компетентностям:

– КФ-1 – Здатність обґрунтовано застосовувати, інтегрувати, розробляти та удосконалювати сучасні інформаційні технології, фізичні та математичні моделі, а також технології створення та використання прикладного і спеціалізованого програмного забезпечення для вирішення професійних задач у сфері інформаційної безпеки та/або кібербезпеки;

– КФ-6 – Здатність аналізувати, контролювати та забезпечувати систему управління доступом до інформаційних ресурсів згідно встановленої стратегії і політики інформаційної безпеки та/або кібербезпеки організації;

– КФ-9 – Здатність аналізувати, розробляти і супроводжувати систему аудиту та моніторингу ефективності функціонування інформаційних систем і технологій, бізнес/операційних процесів в галузі інформаційної безпеки та/або кібербезпеки організації в цілому;

– КФ-10 – Здатність провадити науково-педагогічну діяльність, планувати навчання, контролювати і супроводжувати роботу з персоналом, а також приймати ефективні рішення з питань інформаційної безпеки та/або кібербезпеки;

– КФ-15 – Здатність адмініструвати системи, мережі та системи безпеки інформації.

Дана кваліфікаційна робота може бути використана для:

– проведення студентських змагань в рамках університету;

– може бути використана для курсів підвищення кваліфікації за замовленням Миколаївської ОВА .

1 ПРИЗНАЧЕННЯ ТА ОБЛАТЬ ЗАСТОСУВАННЯ

1.1 Поняття хакатон та його область застосування

Хакатон – це захід, який зазвичай триває короткий проміжок часу, де люди збираються разом для співпраці з метою швидкого вирішення проблеми або виявлення нових можливостей [1].

Хакатони зосереджені на практичній співпраці. Команди занурюються в роботу, засукають рукави та створюють прототипи або рішення – зазвичай протягом одного-двох днів. Атмосфера інклюзивна та експериментальна, з акцентом на креативність, командну роботу та навчання через практику.

Але що справді відрізняє хакатони, так це те, як вони підвищують креативність та зростання так, як стандартна офісна рутина просто не може зрівнятися.

Концепція «хакатону» народилася в 1999 році завдяки розробникам-першопрохідцям з *OpenBSD* та *Sun Microsystems*. *OpenBSD*, організація, що займається програмним забезпеченням, орієнтованим на безпеку, придумала цей термін під час багатоденного заходу, де розробники зібралися разом, щоб «зламати» код, інтенсивно працюючи над покращенням та захистом свого програмного забезпечення. Приблизно в той же час *Sun Microsystems* представила цей термін на конференції *JavaOne* , провівши хакатон, на якому розробники запрошувалися до спільної створення нових додатків на Java.

Сам термін поєднує в собі «хак» (що стосується креативного кодування або вирішення проблем) та «марафон», що представляє тривалі, цілеспрямовані зусилля заходу. Це поєднання ідеально відображає суть хакатонів: енергійних, спільних сесій , де програмісти та новатори інтенсивно працюють протягом короткого періоду часу над вирішенням проблем або створенням прототипів.

Відтоді модель хакатону здобула світову популярність, надихаючи галузі поза межами технологічних використовувати силу швидкого генерування ідей,

командної роботи та концентрованих інновацій для вирішення викликів та просування нових ідей.

Сьогодні хакатони стали синонімом каталізатора для швидких інновацій у різних секторах, сприяючи динамічній співпраці та вирішенню проблем у структурованих середовищах з високим рівнем впливу.

Хакатони призначені не лише для технологічних гігантів чи стартапів Кремнієвої долини – вони переключили увагу на такі галузі, як автомобілебудування та косметика, доводячи свою цінність у різних умовах. У автомобільному світі хакатони спонукають учасників переосмислити сталий розвиток, наприклад, провести мозковий штурм для пошуку способів використання науки про дані для чистіших ланцюгів поставок або винайти розумнішу логістику, яка скорочує викиди CO_2. Ці заходи об’єднують не лише програмістів, а й експертів з логістики та ентузіастів екології, які прагнуть екологічнішого майбутнього.

Прикладами сучасних популярних хакатонів є [2]:

– конкурс космічних додатків *NASA* «Глобальна співпраця заради Землі та за її межами». Виклик космічних додатків є доказом того, що хакатони можуть бути буквально неземними. Зосереджуючись на вирішенні проблем, пов'язаних з космосом та науками про Землю, учасники з усього світу об'єднуються, щоб вирішувати реальні проблеми за допомогою супутникових даних, розробляти інструменти для місій або пропонувати нові способи моніторингу нашої планети. Результат? Процвітаюча мережа вчених, програмістів та великих мислителів, які співпрацюють, щоб принести на Землю рішення космічної ери;

– *AngelHack*: Інновації із соціальним підтекстом. Він відчиняє свої двері для новаторів з усього світу, проводячи як класичні хакатони, так і цілеспрямовані челенджі. Багато заходів зосереджені на актуальних питаннях, таких як різноманітність у технологіях – уявіть собі команди, які занурюються в рішення для розширення прав і можливостей жінок або ширшого соціального блага. Результатом є багатий портфель стартапів та проектів, які розмивають межу між прибутком та метою;

– *HackMIT*: Магніт для талантів та інкубатор ідей. *HackMIT* приваблює блискучих студентів з усієї планети, які прагнуть вирішувати складні проблеми в технологіях, охороні здоров'я, фінансах тощо. Спільний дух заходу заохочує сміливі експерименти – студенти часто створюють прототипи проектів, які привертають увагу далеко за межами «вежі зі слонової кістки», сигналізуючи про нові напрямки для інновацій у галузі;

– Внутрішня робота великих технологічних компаній: хакатони *Facebook* та *Google*. Усередині технологічних титанів, таких як *Facebook* та *Google*, хакатони є частиною секретного інгредієнта. Працівники по всьому світу об’єднують зусилля для мозкового штурму та створення нових функцій для користувачів (привіт, прямий ефір у *Facebook*) до досягнень у галузі штучного інтелекту та хмарних обчислень. Ці високоенергійні спринти мають хист перетворювати дикі ідеї на рішення, що визначають платформу, підживлюючи постійну еволюцію продуктів, на які мільйони людей покладаються щодня.

Корпоративні хакатони – це потужний спосіб об’єднати співробітників, зовнішніх учасників або їх поєднання для вирішення реальних проблем, стимулювання креативності та впровадження значущих інновацій. Ці заходи надають можливість використати різні точки зору, дозволяючи вашій організації розробляти передові технології, вирішувати конкретні проблеми та досліджувати нові бізнес-можливості в умовах співпраці.

Окрім надання інноваційних рішень, хакатони пропонують низку стратегічних переваг для організацій у різних галузях. Спонсорування хакатону може допомогти вам зібрати цінні відгуки та свіжі ідеї для нових продуктів, функцій чи послуг. Ці заходи також слугують платформою для сприяння розвитку співробітників, зміцнення утримання найкращих талантів та демонстрації інвестицій вашої організації в інновації та постійне вдосконалення.

Незважаючи на свою широку популярність, хакатони 2025 року пропонують поєднання переваг і потенційних підводних каменів, які організації повинні враховувати, щоб максимізувати цінність.

Ключові переваги [3]:

– розкриття креативності: Незалежно від того, чи проводяться вони технологічними гігантами, чи стартапами, що розвиваються, хакатони створюють благодатний ґрунт для нестандартного мислення та творчого вирішення проблем. Команди швидко експериментують, вдосконалюють та представляють концепції, які інакше могли б загубитися в щоденних робочих процесах;

– нетворкінг та знайомство: Завдяки присутності лідерів галузі, підприємців та спеціалістів, учасники отримують можливості для спілкування, співпраці та побудови змістовних стосунків, які можуть призвести до майбутнього партнерства або кар'єрного зростання;

– навчання через практику: Хакатони сприяють практичному навчанню, дозволяючи людям застосовувати нові інструменти, методології та технології в динамічному, швидкозмінному середовищі. Цей практичний досвід часто перетворюється на розвиток навичок після самої події.

– відчутні результати: Стислі часові рамки спонукають команди до створення практичних прототипів та рішень. Багато організацій, від фінансових установ до постачальників медичних послуг, впровадили ідеї, що народилися на хакатонах, для покращення продуктів, оптимізації процесів або виходу на нові ринки.

Потенційні недоліки:

– сталість рішень: Ідеям, що народжуються в запалі 48-годинного спринту, може бракувати глибини та вдосконалення, необхідних для довгострокового впровадження. Подальша підтримка та достатні ресурси мають вирішальне значення для перетворення прототипів на масштабовані пропозиції;

– вигорання учасників: Інтенсивність та тиск часу іноді можуть призвести до втоми, зниження креативності або відбиття бажання брати участь у тих, хто досягає успіхів завдяки більшому плануванню та рефлексії.

– проблеми інклюзивності: Незважаючи на найкращі наміри, хакатони можуть ненавмисно надавати перевагу учасникам з певним технічним досвідом

або доступом до ресурсів, іноді відтісняючи голоси нетрадиційних або недостатньо представлених груп. Створення інклюзивного, гостинного середовища та надання підтримки для всіх рівнів досвіду залишається постійним викликом.

1.2 Типи хакатону

Зі зростанням кількості хакатонів, буває важко зрозуміти всю різноманітність доступних заходів. Кожен тип має свої цілі, учасників та унікальні виклики.

1.2.1 Корпоративні хакатони

Корпоративні хакатони організовуються компаніями для заохочення інновацій та вирішення конкретних бізнес-завдань. Ці заходи часто зосереджені на внутрішніх процесах, удосконаленні продуктів або нових ринкових рішеннях. Вони є чудовою можливістю для компаній залучити зовнішні таланти та креативність співробітників.

Мета: Стимулювання бізнес-інновацій та пошук рішень для галузевих проблем.

Учасники: співробітники, фрілансери, стартапи та ентузіасти технологій.

Приклад: Роздрібна компанія проводить хакатон з розробки інструментів штучного інтелекту для персоналізованого шопінгу.

Ці хакатони часто завершуються реальними впровадженнями, даючи учасникам можливість побачити, як їхні рішення впливають.

1.2.2 Соціально-корисні хакатони

Соціальні хакатони спрямовані на вирішення нагальних суспільних проблем, таких як охорона здоров'я, освіта чи зміна клімату. Учасники працюють над проєктами, спрямованими на створення позитивного впливу на громади.

Мета: Розробка рішень соціальних та екологічних проблем.

Учасники: громадські організації, розробники, активісти та волонтери.

Приклад: Створення застосунку для ліквідації наслідків стихійних лих для оптимізації зусиль з реагування на надзвичайні ситуації.

Ці хакатони часто підтримуються некомерційними організаціями та урядовими установами, що робить їх ідеальними для тих, хто прагне зробити свій внесок у загальне благо.

1.2.3 Студентські хакатони

Студентські хакатони зосереджені на розвитку навичок та креативності молодих умів. Вони надають студентам платформу для експериментів, навчання та інновацій.

Мета: Сприяти навчанню, інноваціям та кар'єрним можливостям.

Учасники: учні старших класів або коледжів.

Приклад: Студенти розробляють навчальну гру для покращення навчального досвіду.

Ці заходи часто включають менторські сесії, що робить їх ідеальними для початківців та майбутніх фахівців у галузі технологій.

1.2.4 Тематичні хакатони

Тематичні хакатони обертаються навколо певної теми чи технології, такої як штучний інтелект, ігри чи кібербезпека.

Мета: Дослідити інновації в нішевій галузі.

Учасники: Ентузіасти та експерти з обраної теми.

Приклад: блокчейн-хакатон для розробки децентралізованих фінансових додатків.

Ці хакатони дозволяють учасникам заглибитися у спеціалізовані галузі та продемонструвати досвід у трендових технологіях.

1.2.5 Хакатони відкритих інновацій

Хакатони відкритих інновацій не мають суворих правил чи тем, що дає учасникам свободу працювати над проєктами на власний вибір.

Мета: Розвивати креативність та нестандартне мислення.

Учасники: Будь-хто, хто має ідею, включаючи розробників, підприємців та дизайнерів.

Приклад: команди, що розробляють унікальні додатки для рішень для розумного дому.

Гнучкість цих заходів робить їх центром інновацій, що залучає різноманітні таланти з різних галузей.

1.2.6 Марафонські хакатони

Марафонські хакатони – це високоінтенсивні заходи, де учасники працюють безперервно над створенням функціональних прототипів за короткий проміжок часу, який часто триває від 24 до 72 годин.

Мета: Перевірити навички вирішення проблем та витривалість під тиском.

Учасники: Високомотивовані окремі особи або команди.

Приклад: 48-годинний спринт кодування для розробки мобільного додатку для здоров'я.

Ці події часто призводять до відчутних, готових до розгортання рішень.

1.2.7 Внутрішні хакатони

Внутрішні хакатони призначені виключно для співробітників компанії, сприяючи командній співпраці та залученості. Ці заходи дозволяють організаціям виявити приховані таланти та покращити динаміку на робочому місці.

Мета: Вирішити внутрішні проблеми та підвищити моральний дух співробітників.

Учасники: Співробітники компанії-організатора.

Приклад: Фінансова компанія розробляє інструмент для оптимізації процесів адаптації клієнтів.

Внутрішні хакатони також відіграють вирішальну роль у формуванні культури безперервних інновацій.

1.2.8 Онлайн-хакатони

Онлайн-хакатони здобули популярність завдяки своїй доступності, що дозволяє учасникам з усього світу співпрацювати дистанційно.

Мета: Розширити участь та об’єднати таланти з усього світу.

Учасники: Віддалені фахівці, студенти та ентузіасти технологій.

Приклад: Глобальний хакатон для покращення інструментів віртуальної співпраці.

Онлайн-платформи забезпечують гнучкість у термінах та участі, роблячи ці заходи інклюзивними та різноманітними.

1.2.9 Хакатони для стартапів

Ці хакатони розроблені для стартапів, які прагнуть удосконалити свої бізнес-моделі, створити прототипи або отримати фінансування. Для допомоги учасникам часто залучаються ментори та інвестори.

Мета: Допомогти стартапам пришвидшити зростання та інновації.

Учасники: команди стартапів, підприємці та інвестори.

Приклад: фінтех-хакатон для створення клієнтоорієнтованих платіжних систем.

Стартапи часто отримують розголос та можливості фінансування завдяки цим заходам.

10. Хакатони *CTF* (Захоплення прапора)

Хакатони *CTF*, що в першу чергу зосереджені на кібербезпеці, передбачають вирішення проблем безпеки, таких як розшифровка кодів, виявлення вразливостей або злом змодельованих систем [4,5].

Мета: Перевірити та вдосконалити навички кібербезпеки.

Учасники: Етичні хакери та ентузіасти кібербезпеки.

Приклад: Змагання із захисту мережі від симульованих кібератак.

Ці хакатони ідеально підходять для тих, хто захоплюється етичним хакінгом та тестуванням на проникнення.

1.3 Етапи проведення

1.3.1 Етап планування

Етап планування – це те, на чому закладається основа для ефективного хакатону. Успішні події не трапляються самі по собі – вони ретельно

розробляються крок за кроком, щоб забезпечити динамічний та продуктивний досвід для всіх учасників.

Ось чого очікувати на етапі планування хакатону:

– визначення основного виклику: Перший крок – визначити переконливий виклик або можливість для учасників, яку їм потрібно вирішити;

– визначення вашої аудиторії: Далі визначте, хто має бути залучений. Вибір правильного перетину учасників допомагає наповнити захід свіжими поглядами та навичками;

– структурування ролей та обов'язків: Організація заходу означає переконатися, що кожен знає свою роль. Розподіліть ролі між фасилітаторами, наставниками, суддями та технічною підтримкою, щоб забезпечити безперебійний перебіг заходу від початку до кінця;

– складання графіка: Встановіть віхи для мозкового штурму, розробки та презентації, щоб учасники залишалися енергійними, але зосередженими протягом усього вашого швидкого інноваційного марафону;

– залучення та приваблення учасників: Після того, як логістика була розроблена, настав час поширити інформацію та залучити мотивованих фахівців з вирішення проблем. Використовуйте канали компанії, соціальні мережі та партнерські організації, щоб створити ажіотаж та забезпечити активну участь.

1.3.2 Етап проведення

На етапі проведення слід передбачити:

– вступна сесія: Підготуйте сцену захопливим привітанням, знайомством та чітким поясненням мети заходу, основних правил і критеріїв оцінювання;

– формування команди: Дайте учасникам час сформувати або зміцнити команди, обговорити ідеї та обрати напрямок свого проекту. «Лиголам» або короткі вправи з мозкового штурму можуть допомогти «розтопити лід» та заохотити співпрацю;

– робочі блоки: Розділіть основний час хакінгу на кілька цілеспрямованих періодів з додатковими перервами. Забезпечте гнучкість для команд, щоб вони

могли зануритися в розробку ідей, створення прототипів, кодування та тестування;

– проміжні демонстрації або контрольні точки: Запропонуйте командам проміжний контрольний пункт для обміну інформацією про ранній прогрес, виявлення перешкод та збору відгуків. Це також допомагає підтримувати імпульс та відновлювати енергію групи;

– заключні презентації: Зарезервуйте достатньо часу для кожної команди, щоб представити своє рішення.

– оцінювання та нагородження: Завершіть раундом оцінювання, де проекти оцінюються на основі заздалегідь встановлених критеріїв (наприклад, інновації, зручність використання, вплив). Завершіть на високій ноті сесією нагородження та заключними словами, щоб відзначити всіх учасників.

1.3.3 Підсумки

Коли ваш хакатон наближається до завершення, важливо вловити імпульс і забезпечити визнання кожного прориву та його ефективне використання. Ось що зазвичай відбувається на цьому заключному етапі:

– зворотній зв'язок та рефлексія: Команди та організатори збираються разом, щоб поділитися думками щодо заходу – що спрацювало, що можна було б покращити та які моменти варто відзначити. Ці розмови допомагають визначити сильні сторони та сфери для покращення;

– оцінка впливу: переглядаються показники успіху та результати, будь то кількість створених нових ідей, створених прототипів чи вдосконалених рішень;

– планування дій: Обговорюються та визначаються ключові наступні кроки. Перспективні концепції можуть бути обрані для подальшої розробки, прототипи можуть увійти до пілотних фаз, або команди-переможці можуть отримати ресурси для подальшого вдосконалення своїх рішень.

– відзначення досягнень: Визнання має значення – найкращі виконавці та всі учасники отримують визнання за свою креативність та відданість, що зміцнює позитивну культуру інновацій.

– постійне вдосконалення: Винесені уроки документуються для покращення структури та результатів майбутніх хакатонів, що гарантує ще більший вплив вашого наступного заходу.

Продумано завершуючи свій хакатон, ви створюєте довготривалу цінність з цього досвіду, сприяєте постійній взаємодії та закладаєте основу для стійких інновацій у вашій організації.

1.3.4 Збір остаточних показників після хакатону.

Після того, як адреналіновий сплеск від хакатону вщухне, настав час перетворити участь та креативність на практичні дані. Збір значущих показників є важливим не лише для цілей звітності, але й для того, щоб ви могли оптимізувати кожен майбутній захід.

Учасники опитування: Використовуйте опитування (Google Forms, SurveyMonkey або ваші внутрішні інструменти), щоб зібрати відгуки від учасників, суддів та спонсорів з таких питань, як залученість, навчання та загальне задоволення.

Дані про участь: Відстежуйте кількість зареєстрованих користувачів, формування команд, подані ідеї та загальний рівень завершення проектів.

Аналітика залученості: Перегляньте аналітику платформи щодо входу в систему, тривалості сеансів, частоти коментарів та активності голосування. Ці дані показують, наскільки залученими та активними у співпраці були ваші учасники.

Якість та вплив ідеї: Проаналізуйте подані ідеї на оригінальність, доцільність та відповідність вашим інноваційним цілям, використовуючи обрану вами критерій оцінювання або систему оцінювання.

Відстеження результатів: Вимірюйте, які ідеї просуваються вперед, чи переходять вони до пілотних проектів, чи повноцінних, і як вони сприяють вашому інноваційному портфелю.

Визнання та винагороди: документуйте розподіл нагород та найкращих учасників – бали, значки чи інші форми гейміфікації – щоб відзначити вплив вашої події та стимулювати майбутню залученість.

Об’єднання всього цього дає вам повну картину окупності інвестицій вашого хакатону, висвітлює сфери для покращення та надає вам ідеї для вашого наступного великого інноваційного виклику.

Висновки до розділу

У ході виконання роботи було сформовано цілісне уявлення про хакатони як ефективний інструмент розвитку практичних навичок у сфері кібербезпеки. Було проаналізовано їх природу, особливості організації та типові формати проведення, що дало змогу визначити місце та роль таких заходів у навчальному процесі та професійному розвитку спеціалістів. Особливу увагу приділено формату Capture The Flag, який дозволяє моделювати реальні кіберзагрози та забезпечує учасникам можливість відпрацьовувати практичні компетенції в максимально наближених до реальних умов сценаріях.

2 АНАЛІЗ ОСНОВНИХ ПИТАНЬ ПРОВЕДЕННЯ ХАКАТОНУ

2.1 Основні принципи провдення хакатону типу *CTF*

У комп'ютерній безпеці « Захоплення прапора» (*CTF*) – це хакатон, в якому учасники намагаються знайти текстові рядки, які називаються «прапорами», що таємно заховані у навмисно вразливих програмах або на веб-сайтах. Їх можна використовувати як для змагальних, так і для освітніх цілей. У двох основних варіантах *CTF* учасники або крадуть прапори в інших учасників (*CTF* у стилі атаки/захисту), або в організаторів (виклики у стилі небезпеки). Змішані змагання поєднують ці два стилі [6,7].

Змагання можуть включати приховування прапорів у апаратних пристроях, вони можуть проводитися як онлайн, так і особисто, а також можуть бути просунутого або початкового рівня. . *CTF* використовуються як інструмент для розвитку та вдосконалення навичок кібербезпеки, що робить їх популярними як у професійному, так і в академічному середовищі.

Два популярні формати *CTF* – це «*Jeopardy*» та «*Attack-Defense*». Обидва формати перевіряють знання учасників у сфері кібербезпеки, але відрізняються метою [8]. У форматі «*Jeopardy*» команди-учасники повинні виконати якомога більше завдань різної вартості балів з різних категорій, таких як криптографія, веб-експлойція та зворотна інженерія. У форматі «*Attack-Defense*» команди-учасники повинні захищати свої вразливі комп'ютерні системи, одночасно атакуючи системи суперника.

Ця вправа включає різноманітний спектр завдань, включаючи експлуатацію та злом паролів, але існує мало доказів того, як ці завдання перетворюються на знання кібербезпеки, якими володіють експерти з безпеки. Нещодавні дослідження показали, що завдання «Захоплення прапора» охоплювали переважно технічні знання, але не мали соціальних тем, таких як соціальна інженерія та обізнаність у сфері кібербезпеки.

CTF охоплюють широкий спектр навичок кібербезпеки, включаючи криптографію, криміналістику та мережевий захист. Члени команди можуть вибирати завдання, які розширюють їхні навички за межі їхніх сильних сторін, сприяючи зростанню в нових сферах та створюючи універсальну, всебічно збалансовану команду, краще оснащену для кіберзахисту.

Змагання з кіберзлочинності (*CTF*) вимагають від вашої команди швидкого мислення, адаптації та співпраці, як це було б під час реального кіберінциденту. Ці змагання заохочують комунікацію та стратегічне мислення, формуючи ефективну командну динаміку, яка імітує середовища з високими ставками, з якими вони зіткнуться в польових умовах.

Основні навички кіберзахисту, розроблені за допомогою *CTF*-челенджів [9].

Виявлення та аналіз загроз.

CTF моделюють реальні кіберзагрози, пропонуючи вашій команді практичний досвід у виявленні ознак компрометації в складних мережевих середовищах. Учасники відпрацьовують низку методів, таких як аналіз файлів *PCAP* (захоплення пакетів) для виявлення незвичайного мережевого трафіку та потенційної шкідливої активності.

Ще однією поширеною проблемою може бути перевірка журналів для виявлення шаблонів несанкціонованого доступу або шкідливих *IP*-адрес. За допомогою різноманітних методів виявлення члени команди вчаться виявляти ознаки компрометації. Така постійна практика зміцнює здатність вашої команди ефективно контролювати мережеву активність і швидко реагувати на потенційні порушення, посилюючи захист вашої організації на передовій.

Реагування на інциденти та судова експертиза.

CTF забезпечують необхідне навчання з реагування на інциденти, допомагаючи учасникам відпрацювати стримування та зменшення загроз у змодельованих сценаріях порушення. Завдання можуть включати дослідження скомпрометованого сервера та визначення початкової точки входу або аналіз

дампа пам'яті для виявлення прихованого шкідливого програмного забезпечення.

Окрім негайного реагування, фахівці з контролю за шкідливими системами також супроводжують вашу команду в процесі судово-медичного аналізу, навчаючи їх відстежувати джерела атак та збирати критичні докази, такі як пошук підозрілих файлів або відстеження шкідливих процесів. Ці навички життєво важливі не лише для ефективного відновлення, але й для винесення висновків з кожного інциденту з метою зміцнення захисту в майбутньому.

Виявлення та зменшення вразливостей.

Викриваючи команди поширеними вразливостями, *CTF* навчають учасників розпізнавати та усувати недоліки, перш ніж зловмисники зможуть їх використати. Проблеми цієї категорії часто пов'язані з аналізом веб-застосунків на наявність вразливостей, таких як *SQL*-ін'єкції або міжсайтовий скриптинг.

Іншим прикладом може бути перевірка коду на наявність небезпечних конфігурацій або застарілих програмних залежностей. Такий проактивний підхід посилює кіберготовність та створює сильнішу та стійкішу систему захисту.

Автоматизація та сценарії для кіберзахисту.

CTF часто передбачають завдання, які заохочують учасників використовувати скрипти та автоматизацію для ефективного вирішення проблем або оптимізації робочих процесів. Наприклад, завдання може вимагати автоматизації повторюваного завдання, такого як аналіз файлів журналів для вилучення певних шаблонів або створення скрипта для виявлення та блокування шкідливих IP-адрес.

Ці навички безцінні для вирішення складних завдань та оптимізації часу реагування в реальних сценаріях кіберзахисту, дозволяючи учасникам автоматизувати рутинні завдання та зосереджуватися на високопріоритетних загрозах.

2.2 Аналіз тематики для проведення тренувального хакатону

Для навчання з напрямку кібербезпеки в практичних навичок одними з важливих значень є аналіз вхідного та вихідного трафіку, знання принципів функціонування глобальної мережі (моделі, протоколи), вразливості Інтернет додатків тощо [10].

Для проведення навчального хакатону пропонується декілька тематик завдань:

2.2.1 Аналіз мережевого трафіку

Network Traffic Analysis (NTA) можна описати як акт дослідження мережевого трафіку для характеристики поширених портів та протоколів, що використовуються, встановлення базової лінії для нашого середовища, моніторингу та реагування на загрози, а також забезпечення максимально можливого розуміння мережі нашої організації.

Цей процес допомагає фахівцям з безпеки виявляти аномалії, включаючи загрози безпеці в мережі, завчасно та ефективно визначати загрози. Аналіз мережевого трафіку також може сприяти процесу дотримання правил безпеки. Зловмисники часто оновлюють свою тактику, щоб уникнути виявлення, та використовують легітимні облікові дані за допомогою інструментів, які більшість компаній дозволяють використовувати у своїх мережах, що ускладнює виявлення та, як наслідок, реагування для захисників. У таких випадках аналіз мережевого трафіку знову може виявитися корисним.

Наприклад, якщо ми виявляємо багато *SYN* пакетів на портах, які ми ніколи (або рідко) використовуємо в нашій мережі, ми можемо зробити висновок, що це, найімовірніше, хтось намагається визначити, які порти відкриті на наших хостах.

Навички, які необхідно надати слухачам для проходження таких типів завдань можна виділити [11-14]:

– стек *TCP/IP* та модель *OSI* – це розуміння гарантуватиме, що ми зрозуміємо, як взаємодіють мережевий трафік та хост-додатки;

– основні мережеві концепції – розуміння типів трафіку, які ми побачимо на кожному рівні, включає розуміння окремих рівнів, що складають моделі *TCP/IP* та *OSI*, а також концепцій комутації та маршрутизації. Якщо ми підключимося до мережі на магістральному з'єднанні, ми побачимо набагато більше трафіку, ніж зазвичай, і він буде разюче відрізнятися від того, що ми бачимо під час підключення до офісного комутатора;

– загальні порти та протоколи – швидке визначення стандартних портів і протоколів і функціональне розуміння того, як вони взаємодіють, забезпечать нам можливість ідентифікувати потенційно шкідливий або спотворений мережевий трафік;

– концепції *IP*-пакетів та підрівнів – базові знання про те, як взаємодіють *TCP* та *UDP*, як мінімум, гарантуватимуть, що ми розуміємо, що бачимо або шукаємо. *TCP*, наприклад, потоково-орієнтований і дозволяє нам легко відстежувати розмову між хостами. *UDP* швидкий, але не переймається повнотою, тому було б важче відтворити щось з цього типу пакета.

– інкапсуляція транспортування протоколів – кожен шар інкапсулюватиме попередній. Можливість зчитувати або аналізувати зміни цієї інкапсуляції допоможе нам швидше переміщатися по даних. Легко побачити підказки на основі заголовків інкапсуляції;

– навколишнє середовище та обладнання – визначення інструментів та засобів для проведення моніторингу.

Існує багато різних інструментів та типів обладнання, які можна використовувати для аналізу мережевого трафіку. Кожен з них пропонує різний спосіб захоплення або аналізу трафіку. Поширеними інструментами для аналізу трафіку є [15-18]:

– *tcpdump* – це утиліта командного рядка, яка за допомогою *LibPcap* захоплює та інтерпретує мережевий трафік з мережевого інтерфейсу або файлу захоплення;

– *TShark* – це аналізатор мережевих пакетів, дуже схожий на *TCPDump*. Він може захоплювати пакети з активної мережі або зчитувати та декодувати з файлу. Це варіант *Wireshark* з командним рядком;

– *Wireshark* – це графічний аналізатор мережевого трафіку. Він захоплює та декодує кадри з мережі, що дозволяє глибоко зазирнути в середовище. Він може запускати багато різних аналізаторів трафіку, щоб охарактеризувати протоколи та програми, а також надати уявлення про те, що відбувається;

– *NGrep* – це інструмент зіставлення зі шаблонами, створений для виконання функцій, подібних до *grep* для дистрибутивів *Linux*;

– *tcpick* – це сніффер пакетів командного рядка, який спеціалізується на відстеженні та повторному складанні *TCP*-потоків. Функціональність зчитування потоку та його повторного збирання назад у файл за допомогою tcpick чудова;

– *Network Taps* Відгалужувачі (*Gigamon , Niagra-taps*) – це пристрої, здатні робити копії мережевого трафіку та надсилати їх в інше місце для аналізу;

– *SIEMS* (наприклад, *Splunk*) – є центральною точкою, в якій дані аналізуються та візуалізуються. Оповіщення, судово-медичний аналіз та щоденні перевірки трафіку – все це варіанти використання *SIEM*.

Виконання аналізу мережевого трафіку.

Виконання аналізу може бути як простим, як-от спостереження за рухом трафіку в реальному часі в нашій консолі, так і складним, як збір даних одним дотиком, їх відправлення назад до *SIEM* для обробки та аналіз даних *pcap* на наявність сигнатур та сповіщень, пов’язаних із поширеними тактиками та методами.

Як мінімум, для пасивного прослуховування нам потрібно бути підключеним до сегмента мережі, який ми хочемо прослуховувати. Це особливо актуально в комутованому середовищі, де *VLAN* та порти комутатора не пересилають трафік за межі свого домену широкомовлення. З огляду на це, якщо ми хочемо захоплювати трафік з певної *VLAN*, наш пристрій захоплення має бути підключений до цієї ж мережі.

Аналіз трафіку не є точною наукою. *NTA* може бути дуже динамічним процесом і не є прямим циклом. На нього значною мірою впливає те, що ми шукаємо (помилки мережі чи шкідливі дії) та де ми маємо доступ до нашої мережі. Виконання аналізу трафіку може звестися до кількох основних кроків: отримання трафіку, зменшення шуму шляхом фільтрації, аналіз та дослідження, виявлення та оповіщення.

1) Захоплення трафіку. Щойно ми визначимося з місцем розташування, почнемо збирати трафік. Використовуйте фільтри захоплення, якщо ми вже маємо уявлення про те, що шукаєте.

2) Зменшення шуму за допомогою фільтрації. Захоплення трафіку з'єднання, особливо в робочому середовищі, може бути надзвичайно шумним. Після завершення початкового захоплення, спроба відфільтрувати непотрібний трафік з нашого поля зору може спростити аналіз. (Наприклад, трафік широкомовної розсилки та багатоадресної розсилки).

3) Аналізуйте та досліджуйте. Зверніть увагу на конкретні хости, протоколи, навіть такі специфічні речі, як прапорці, встановлені в заголовку *TCP*. Наступні питання допоможуть нам:

– Трафік зашифрований чи є звичайним текстом? Чи має бути?

– Чи можемо ми бачити, як користувачі намагаються отримати доступ до ресурсів, до яких вони не повинні мати доступу?

– Чи спілкуються між собою різні хости, які зазвичай цього не роблять?

4. Виявлення та оповіщення. На цьому етапі можуть стати в пригоді інші інструменти, такі як *IDS* та *IPS*. Вони можуть виконувати евристику та сигнатури для трафіку, щоб визначити, чи є щось у ньому потенційно шкідливим.

Для аналізу трафіка корисними є аналіз протоколів верхніх рівнів 5-7 моделі *OSI* [19,-2]:

– протокол передачі гіпертексту (*HTTP*) – це протокол прикладного рівня без збереження стану. *HTTP* дозволяє передавати дані у відкритому тексті між клієнтом і сервером через *TCP*. Клієнт надсилає *HTTP*-запит серверу, запитуючи ресурс. Встановлюється сеанс, і сервер відповідає запитуваним медіафайлом

(*HTML*, зображеннями, гіперпосиланнями, відео). *HTTP* використовує порти 80 або 8000 через *TCP* під час звичайної роботи;

– *HTTP Secure* (*HTTPS*) – це модифікація протоколу *HTTP*, розроблена для використання *Transport Layer Security* (*TLS*) або *Secure Sockets Layer* (*SSL*) зі старими програмами для захисту даних. *TLS* використовується як механізм шифрування для захисту зв'язку між клієнтом і сервером. *TLS* може обгортати звичайний *HTTP*-трафік в *TLS*, що означає, що ми можемо шифрувати всю нашу розмову, а не лише надіслані або запитувані дані. Хоча за своєю основою *HTTP*, *HTTPS* використовує порти 443 та 8443 замість стандартного порту 80. Це простий спосіб для клієнта сигналізувати серверу про те, що він бажає встановити безпечне з'єднання;

– протокол передачі файлів (*File Transfer Protocol* (*FTP*)) – це протокол прикладного рівня, який забезпечує швидку передачу даних між обчислювальними пристроями. Сам *FTP* вважається незахищеним протоколом, і більшість користувачів перейшли на використання таких інструментів, як *SFTP*, для передачі файлів через захищені канали. У цьому аспекті *FTP* унікальний, оскільки він використовує кілька портів одночасно. *FTP* використовує порти 20 та 21 через *TCP*. Порт 20 використовується для передачі даних, а порт 21 – для видачі команд, що керують сеансом *FTP*. Що стосується автентифікації, *FTP* підтримує автентифікацію користувачів, а також дозволяє анонімний доступ, якщо це налаштовано;

– блок повідомлень сервера (*Server Message Block* (*SMB*)) – це протокол, який найчастіше зустрічається в корпоративних середовищах *Windows* і дозволяє спільний доступ до ресурсів між хостами через загальні мережеві архітектури. *SMB* – це протокол, орієнтований на з’єднання, який вимагає автентифікації користувача від хоста до ресурсу, щоб гарантувати, що користувач має правильні дозволи на використання цього ресурсу або виконання дій. У минулому *SMB* використовував *NetBIOS* як механізм транспортування через порти *UDP* 137 та 138. З сучасними змінами *SMB* тепер підтримує прямий транспорт *TCP* через порт 445, *NetBIOS* через порт *TCP* 139 і навіть протокол *QUIC*.

Підхід до аналізу.

Почніть з того, що стосується лише стандартних протоколи, і поступово прокладіть собі шлях до специфічного вмісту. Це означає, що буде генеруватися трафік і про нього будуть записуватися журнали. *HTTP/S*, *FTP*, електронна пошта, а також базовий *TCP* та *UDP*-трафік – це найпоширеніші типи трафіку на які слід звертати увагу. Почніть з них і видаліть все, що не є необхідним для розслідування.

Шукайте шаблони. Чи певний хост або група хостів щодня перевіряє щось в Інтернеті одночасно? Це типова конфігурація профілю *Command and Control*, яку можна легко виявити, шукаючи закономірності в наших даних про трафік.

Перевірте все *host to host* в мережі. У стандартній конфігурації хости користувача рідко взаємодіють один з одним. Тому підозріло ставтеся до будь-якого трафіку, який виглядає подібним чином. Зазвичай хости взаємодіють з інфраструктурою для оренди *IP*-адрес, *DNS*-запитів, корпоративних служб та пошуку свого маршруту.

Звертайте увагу на особливі події. Цікаво, що, наприклад, хост, який зазвичай відвідує певний сайт десять разів на день, змінює свій шаблон і робить це лише один раз. Також варто занепокоїтися, якщо інший рядок *User-Agent* не відповідає нашим програмам або хостам, які спілкуються із сервером в Інтернеті. Також варто звернути увагу на випадковий порт, який прив'язується лише один або два рази на хості. Це може бути прохід для таких речей, як зворотні виклики *C*2, відкриття порту кимось для виконання чогось нестандартного або програма демонструє аномальну поведінку.

2.2.2 Аналіз вразливостей Інтернет додатків

Хоча більшість тестів на проникнення веб-застосунків зосереджена на компонентах серверної частини та їх функціональності, важливо також тестувати компоненти фронтенду на наявність потенційних вразливостей, оскільки ці типи вразливостей іноді можуть бути використані для отримання доступу до конфіденційних функцій (наприклад, панелі адміністратора), що може призвести до компрометації всього сервера [23-29].

2.2.2.1 Розкриття конфіденційних даних стосується доступності конфіденційних даних у відкритому тексті для кінцевого користувача. Зазвичай це знаходиться у вихідному коді сторінки на інтерфейсі веб-додатків.

Іноді ми можемо знайти дані для входу ім'я користувача або інші конфіденційні дані, приховані в коментарях вихідного коду веб-сторінки або в зовнішньому *JavaScript* коді, що імпортується. Інша конфіденційна інформація може включати розкриті посилання або каталоги, або навіть розкриту інформацію про користувача, яка потенційно може бути використана для розширення нашого доступу всередині веб-застосунку або навіть допоміжної інфраструктури веб-застосунку (веб-сервер, сервер баз даних тощо).

З цієї причини, одне з перших, що нам слід зробити під час оцінки веб-застосунку, це переглянути вихідний код його сторінки, щоб побачити, чи можемо ми виявити якісь «легкодоступні» проблеми, такі як розкриті облікові дані або приховані посилання.

Приклад. Форма входу з полями для введення імені користувача та пароля, а також кнопкою входу. Давайте подивимося на вихідний код сторінки:

```
<form action="action_page.php" method="post">

  <div class="container">
    <label for="uname"><b>Username</b></label>
    <input type="text" required>
    <label for="psw"><b>Password</b></label>
    <input type="password" required>
    <!-- TODO: remove test credentials test:test -->

    <button type="submit">Login</button>
  </div>
</form>
</html>
```

Ми бачимо, що розробники додали деякі коментарі, які забули видалити, що містять тестові облікові дані:

<!-- TODO: remove test credentials test:test -->

Цей коментар, схоже, є нагадуванням розробникам про необхідність видалення тестових облікових даних. Оскільки коментар ще не видалено, ці облікові дані можуть бути дійсними.

Використання такої інформації може надати нам додатковий доступ до веб-застосунку, що може допомогти нам атакувати компоненти серверної частини, щоб отримати контроль над сервером.

2.2.2.2 Ін'єкція *HTML*

HTML-ін'єкція відбувається, коли на сторінці відображається нефільтрований користувацький ввід. Це може бути зроблено або шляхом отримання раніше надісланого коду, наприклад, отримання коментаря користувача з бази даних бекенду, або шляхом безпосереднього відображення нефільтрованого користувацького вводу *JavaScript* на фронтенді.

Коли користувач повністю контролює відображення введених даних, він може надіслати *HTML* код, і браузер відобразить його як частину сторінки. Це може включати шкідливий *HTML* код, наприклад, зовнішню форму входу, який можна використовувати, щоб обманом змусити користувачів увійти, фактично надсилаючи їхні облікові дані для входу на шкідливий сервер для збору для інших атак.

2.2.2.3 Міжсайтовий скриптинг (*XSS*)

HTML ін’єкції часто можна використовувати для виконання атак міжсайтового скриптингу (*XSS*) шляхом введення *JavaScript* коду, який має виконуватися на стороні клієнта. Як тільки ми зможемо виконати код на машині жертви, ми потенційно зможемо отримати доступ до облікового запису жертви або навіть до її машини. *XSS* передбачає введення *JavaScript* коду для виконання

більш складних атак на стороні клієнта, а не просто введення *HTML*-коду. Існує три основні типи *XSS*:

– віддзеркалений (*Reflected*) *XSS* – виникає, коли введені користувачем дані відображаються на сторінці після обробки (наприклад, результат пошуку або повідомлення про помилку);

– збережений (*Stored*) *XSS* – виникає, коли введені користувачем дані зберігаються в базі даних сервера, а потім відображаються під час отримання (наприклад, публікації чи коментарі);

– *DOM XSS* – виникає, коли введені користувачем дані безпосередньо відображаються у браузері та записуються в *HTML* об'єкт *DOM* (наприклад, вразливе ім'я користувача або заголовок сторінки).

Ми можемо спробувати вставити наступний *DOM XSS JavaScript* код як корисне навантаження, яке має показати нам значення *cookie* для поточного користувача:

#"><img src=/ onerror=alert(document.cookie)>

Це корисне навантаження звертається до *HTML* дерева документів та отримує *cookie* значення об'єкта. Коли браузер обробляє введені дані, вони вважатимуться новими *DOM*, і команда *`our` JavaScript* буде виконана, відображаючи значення cookie у спливаючому вікні.

Зловмисник може скористатися цим, щоб викрасти сеанси *cookie* та надіслати їх собі, а потім спробувати використати значення *cookie* для автентифікації в обліковому записі жертви. Таку ж атаку можна використовувати для виконання різних типів інших атак на користувачів веб-застосунку.

2.2.4 Підробка міжсайтових запитів (*CSRF*)

Третій тип вразливості фронтенду, спричиненої нефільтрованим вводом даних користувача, – це підробка міжсайтових запитів (*CSRF*). *CSRF* атаки можуть використовувати *XSS* вразливості для виконання певних запитів та *API* викликів веб-застосунку, в якому жертва наразі автентифікована. Це дозволить

зловмиснику виконувати дії від імені автентифікованого користувача. Він також може використовувати інші вразливості для виконання тих самих функцій, таких як використання параметрів *HTTP* для атак.

Поширеною *CSRF* атакою для отримання доступу з вищими привілеями до веб-застосунку є створення *JavaScript* корисного навантаження, яке автоматично змінює пароль жертви на значення, встановлене зловмисником. Щойно жертва переглядає корисне навантаження на вразливій сторінці (наприклад, шкідливий коментар, що містить *JavaScript CSRF* корисне навантаження), *JavaScript* код виконується автоматично. Він використовує сеанс входу жертви в систему для зміни її пароля. Після цього зловмисник може увійти до облікового запису жертви та керувати ним.

CSRF також може бути використаний для атаки на адміністраторів та отримання доступу до їхніх облікових записів. Адміністратори зазвичай мають доступ до конфіденційних функцій, які іноді можна використовувати для атаки та отримання контролю над сервером (залежно від функціональності, що надається адміністраторам у певній веб-програмі). Наслідуючи цей приклад, замість використання *JavaScript* коду, який повертає файл *cookie* сеансу, ми завантажимо віддалений файл *.js*(*JavaScript*) наступним чином:

"><script src=//www.example.com/exploit.js></script>

Файл exploit.js містив би шкідливий *JavaScript* код, який змінює пароль користувача. Розробка exploit.js в цьому випадку вимагає знання процедури зміни пароля цього веб-застосунку та *API*. Зловмиснику потрібно було б створити *JavaScript* код, який би відтворював бажану функціональність та автоматично виконував її (тобто *JavaScript* код, який змінює наш пароль для цього конкретного веб-застосунку).

2.2.6 *SQL*-ін'єкція (*SQLi*)

Ще однією дуже поширеною вразливістю у веб-застосунках є *SQL Injection* вразливість. Подібно до вразливості типу «впровадження команд», ця

вразливість може виникнути, коли веб-застосунок виконує *SQL*-запит, що містить значення, взяте з введених користувачем даних.

Якщо введені користувачем дані не будуть належним чином відфільтровані та перевірені (як у випадку з *Command Injections*), ми можемо виконати ще один *SQL*-запит поряд із цим запитом, що зрештою може дозволити нам отримати контроль над базою даних та її хостинговим сервером.

2.3 Аналіз платформ для проведення змагань

2.3.1 *CTFd.*

CTFd – це відкритий фреймворк для організації та проведення змагань у форматі CTF, який розроблений з акцентом на простоту використання, гнучкість конфігурації та масштабованість. Він широко застосовується університетами, кіберспільнотами, корпоративними тренінгами та конференціями, оскільки забезпечує все необхідне для створення, управління та проведення *CTF*-заходів без потреби розробляти власну інфраструктуру з нуля. *CTFd* написано на *Python* із використанням веб-фреймворку *Flask*, що робить платформу легкою для встановлення й розгортання на власних серверах або в контейнерному середовищі *Docker [30-31]*.

Однією з ключових особливостей *CTFd* є її фокус на модульності та розширюваності. Завдяки підтримці плагінів та тем оформлення, організатори можуть не лише змінювати зовнішній вигляд платформи, але й додавати нові типи завдань, інтеграції чи автоматизовані механізми оцінювання. Такий підхід дозволяє адаптувати *CTFd* під різні формати змагань – від простих *Jeopardy*-стилів до більш складних або спеціалізованих сценаріїв.

Усі основні функції керування змаганням реалізовані через зручну веб-панель адміністратора: це реєстрація команд і користувачів, створення та редагування категорій і завдань, налаштування підказок і ваги балів, а також контроль над таблицею лідерів. *CTFd* автоматично обробляє подання та перевірку прапорів (*flags*), а також підтримує динамічну систему підрахунку

балів, рейтингові таблиці, графіки прогресу та інші візуалізації, що спрощують аналіз результатів змагання.

CTFd підтримує різноманітні функціональні можливості для підвищення зручності та безпеки: обмеження кількості спроб подання для певного завдання, автоматичне запобігання *brute-force*-атак, налаштування процесу відкриття підказок, можливість приховування результатів або заморожування таблиці результатів на певний момент. Такий набір інструментів робить платформу відповідною як для освітніх практичних занять, так і для справжніх змагань.

Окрім відкритого само-хостингу, розробники *CTFd* також пропонують керовані сервіси хостингу та комерційні рішення, включно з розширеними можливостями для корпоративного використання: підтримкою *LDAP/SAML*-автентифікації, інтеграцією з зовнішніми системами, *API*-доступом, веб-терміналами, ізольованими контейнерами для кожного учасника та технічною підтримкою з боку розробників. Це особливо корисно для організації великих заходів, тренінгів чи навчальних кіберполігонів.

Однією зі стратегічних переваг платформи є висока адаптивність та можливість інтеграції з іншими системами, включно зі сторонніми *API* та інструментами *DevOps*, що спрощує автоматичне розгортання нових завдань, синхронізацію з *CI/CD*-середовищами та інтеграцію в навчальний процес.

У підсумку, *CTFd* є потужним інструментом для організації *CTF*-змагань та практичних тренувань у сфері кібербезпеки. Його відкритий характер, багатий набір функцій, налаштування та можливість розширення роблять платформу гнучким вибором як для академічних курсів, так і для професійних змагань, що сприяє формуванню практичних навичок студентів і фахівців у реалістичних умовах.

2.3.2 *RootTheBox*.

RootTheBox – це відкрита платформа для проведення змагань у форматі *Capture The Flag* (*CTF*), яка функціонує як реальний движок для оцінювання та управління *CTF*-іграми і призначена для створення гейміфікованого середовища тестування навичок у сфері кібербезпеки. Платформа побудована на базі

відкритого коду під ліцензією *Apache* 2.0, що дозволяє її вільно адаптувати, модифікувати та розгортати для навчальних та змагальних заходів різного масштабу. Це робить *RootTheBox* привабливим вибором для університетських кіберспільнот, студентських клубів, корпоративних тренінгів та практичних курсів з інформаційної безпеки [32].

RootTheBox підтримує як командний, так і індивідуальний формат участі, що дозволяє проводити змагання для великих груп або окремих учасників, а також легко адаптувати їх під навчальні цілі різного рівня складності. Основна мета платформи – створити реалістичне ігрове поле, де учасники можуть практикувати такі навички, як тестування на проникнення, цифрова криміналістика, реагування на інциденти або *threat hunting*, збираючи «прапори» (*flags*) за проходження окремих завдань або сценаріїв.

Однією з виражених переваг *RootTheBox* є підтримка різних типів прапорів і механік змагання, включно зі статичними прапорами, регулярними виразами, часовими викликами, файлами та багатоваріантними питаннями з урахуванням регістру символів. Це дозволяє організаторам створювати різнопланові завдання, що охоплюють технічні аспекти безпеки та стимулюють критичне мислення. Система також надає можливості для налаштування штрафів, підказок, обмеженої кількості спроб та динамічного нарахування балів, що робить гру збалансованою та цікавою.

Особлива характеристика *RootTheBox* полягає в її ігрових та гейміфікованих елементах, що виходять за межі традиційних *CTF-scoreboard*: платформа підтримує анімацію табло результатів у реальному часі за допомогою веб-сокетів, має вбудований механізм «банкінгу» – внутрішньої валюти, яку гравці можуть заробляти та витрачати, наприклад, на підказки чи відкриття рівнів, а також додаткові функції, такі як «*SWAT*» для взаємодії між гравцями чи елементи *botnet*-симуляції. Ці можливості роблять змагання більш занурюючими та стратегічними, подібними до рольових ігор з елементами ресурсообміну та конкуренції.

RootTheBox також включає вбудовані інструменти для керування командами, обміну файлами та повідомленнями, а за потреби може бути інтегрована зі сторонніми чат-сервісами або експортувати дані для зовнішніх систем статистики. Платформа має налаштування для сторонніх тем, мультимовну підтримку і може бути розгорнута як у хмарних середовищах, так і локально за допомогою *Docker* або прямої інсталяції.

Водночас *RootTheBox* має й деякі обмеження та виклики: її налаштування може вимагати значного часу та технічних знань, особливо для складних сценаріїв або великих змагань, а документація іноді може бути не такою повною чи структурованою, як у деяких інших платформ. Це може створювати певну криву навчання для організаторів, які вперше працюють із відкритими CTF-фреймворками.

У підсумку *RootTheBox* виступає потужним інструментом для організації інтерактивних, настроюваних та гейміфікованих *CTF*-заходів, що сприяють розвитку практичних навичок з кібербезпеки. Її варто розглядати як ефективну платформу для навчання, тренування та проведення кіберзмагань у навчальному процесі, оскільки вона підтримує широкий спектр механік гри, різні формати участі та дає змогу адаптувати події під конкретні навчальні цілі.

2.3.3 *Mellivora*

Mellivora – це відкрита платформа-движок для організації та проведення змагань у форматі *Capture The Flag* (*CTF*), що використовується в середовищі кібербезпеки для створення, управління та проведення *CTF*-змагань. Основна реалізація *Mellivora* написана на мові *PHP* і розповсюджується під ліцензією *GPL*-3.0, що дозволяє вільне використання, модифікацію та розгортання в рамках освітніх чи тренувальних заходів. Проєкт також підтримується спільнотою розробників, а його вихідний код доступний у відкритому репозиторії, що робить платформу зручною як для малих навчальних заходів, так і для масштабних кіберзмагань [33].

Mellivora призначена для хостингу *CTF*-змагань типу *Jeopardy*, де завдання розподілені за категоріями і кожне має певну вагу балів. Платформа надає

організаторам гнучкі засоби для створення довільних категорій і викликів (*challenges*), налаштування підказок, обмежень за часом, а також можливість динамічного розкриття завдань залежно від прогресу команд. *Mellivora* підтримує як ручну, так і автоматичну оцінку поданих рішень (текстових прапорів), що дає змогу адаптувати оцінювання до різних типів практичних завдань і сценаріїв.

Серед ключових функціональних можливостей платформи – таблиця результатів (*scoreboard*) із підтримкою різних типів команд, сторінка прогресу учасників, загальний огляд доступних викликів, а також підказкова система, що сприяє кращому орієнтуванню у ході змагання. Крім того, адміністратори можуть обмежувати доступ до реєстрації за допомогою регулярних виразів для електронної пошти, застосовувати механізми *reCAPTCHA* для захисту від автоматичних ботів та використовувати двохфакторну аутентифікацію (*TOTP*) для покращення безпеки облікових записів.

Платформа забезпечує керування користувачами, адміністративну панель, де можна створювати й редагувати вміст, публікувати новини, налаштовувати меню та внутрішні сторінки, що дає змогу адаптувати її під конкретні навчальні чи змагальні сценарії. *Mellivora* також підтримує обробку великих об’ємів даних, включно з потоковим завантаженням файлів для завдань як у локальне сховище, так і на зовнішні сервіси, такі як *Amazon S*3, що розширює можливості щодо створення різноманітних практичних викликів.

Однією з переваг *Mellivora* є її легкість та висока продуктивність. Платформа є достатньо «легковажною» за ресурсами й може ефективно працювати навіть на базових серверах або в контейнерізованому середовищі (наприклад, *Docker*), що зменшує вимоги до інфраструктури та спрощує розгортання. Це особливо важливо для академічних закладів чи невеликих команд, які не мають великого технічного ресурсу для підтримки складної системи.

Варто також відзначити, що *Mellivora* має розширювані адміністративні можливості, включно зі статистичними звітами, логуванням внутрішніх подій

для відстеження помилок чи аномалій, *SMTP*-підтримкою для надсилання повідомлень учасникам та *JSON*-сумісною таблицею результатів для інтеграції з іншими сервісами чи сторонніми інструментами. Такий набір функцій робить платформу придатною не лише для окремих *CTF*-подій, а й для довготривалого використання у рамках навчальних курсів або регулярних тренувань.

Разом із тим, *Mellivora* має деякі обмеження: порівняно з деякими сучасними платформами вона може не мати настільки розвиненої візуалізації прогресу учасників чи інструментів аналітики, а також вимагає певного технічного рівня для налаштування й адміністрування, що може ускладнювати її використання початківцями. Проте для більшості базових та середніх за складністю CTF-змагань ця платформа залишається ефективним, гнучким і швидким рішенням.

Mellivora є потужним і практичним движком для проведення *CTF*-змагань, що поєднує в собі простоту розгортання, гнучкі можливості з налаштування, безліч адміністративних інструментів та відкритий код для модифікації. Завдяки цим характеристикам платформа підходить для організації кібербезпекових змагань як у навчальному процесі, так і поза ним, сприяючи розвитку практичних навичок студентів і спеціалістів у сфері інформаційної безпеки.

2.3.4 *Cyber ICE Box Platform*

Cyber ICE Box Platform – це сучасна спеціалізована онлайн-платформа для організації і проведення кібербезпекових змагань у форматі CTF, розроблена з метою практичного навчання та тренування навичок з інформаційної безпеки. Вона створена для того, щоб забезпечити студентів, молодих фахівців і учасників змагань середовищем, максимально наближеним до реальних умов кіберзахисту, де учасники можуть виявляти, ідентифікувати та аналізувати кіберзагрози, тренуватися у відлагодженому вирішенні завдань з комплексних сценаріїв атак і захисту.

Платформа *Cyber ICE Box* використовується для проведення регіональних *CTF*-змагань, таких як *Kharkiv Regional Rookie CTF*, де команди змагаються протягом тривалого часу над різноплановими завданнями у віртуальних

лабораторіях, що охоплюють різні категорії практик. Учасники повинні об’єднувати свої зусилля, здійснювати командну роботу та застосовувати знання з кібербезпеки, щоб просуватися у рейтинговій таблиці і вирішувати більш складні задачі.

Однією з фундаментальних особливостей *Cyber ICE Box Platform* є її висока гнучкість, масштабованість і безпека, що досягається завдяки використанню сучасних контейнерних технологій та оркестрації середовищ, таких як *Docker* та *Kubernetes*. Це дозволяє створювати ізольовані середовища для кожного завдання або сценарію, що гарантує стабільну роботу, незалежність ресурсів та захист від потенційних конфліктів між викликами, навіть у масштабних заходах із великою кількістю учасників.

Така архітектура не лише забезпечує високу продуктивність під час змагань, але й створює базу для реалізації реалістичних сценаріїв, де учасники стикаються з викликами, що імітують реальні кіберзагрози. Це включає завдання з аналізу мережевого трафіку, розвідки у віртуальних середовищах, експлуатації вразливостей, цифрової криміналістики та інших аспектів кібербезпеки, що сприяють комплексному розвитку компетенцій.

Крім того, *Cyber ICE Box* використовують для навчання та професійного розвитку в академічних середовищах: платформа дає змогу вивчати та практикувати технічні навички в симульованому середовищі, що є надзвичайно важливою частиною підготовки сучасного фахівця з кібербезпеки. Викладачі можуть інтегрувати *CTF*-змагання як частину курсу або як окремий навчальний компонент, стимулюючи студентів до самостійної роботи над практичними завданнями і застосування теоретичних знань у дії.

Практичний досвід, здобутий під час участі у заходах на *Cyber ICE Box*, також включає командну взаємодію, обмін ідеями та вирішення комплексних проблем у команді, що важливо для майбутніх фахівців у сфері кібербезпеки, де співпраця часто є ключовим фактором успіху.

У цілому *Cyber ICE Box Platform* виступає ефективним інструментом практичної підготовки фахівців і проведення навчальних *CTF*-змагань,

забезпечуючи безпечне, гнучке та масштабоване середовище, яке сприяє розвитку навичок захисту інформації, критичного мислення та оперативного вирішення реалістичних кіберзагроз. Її впровадження в освітній процес або як частину тренувальної інфраструктури значно підсилює практичну складову підготовки спеціалістів у галузі кібербезпеки.

2.3.5 Порівняльна характеристика

Відповідно до вищесказаного можемо створити порівняльну характеристику вказаних платформ (табл. 2.1)

Таблиця 2.1 – Порівняльна характеристика CTF платформ

Платформа / Критерій	*Cyber ICE Box Platform*	*Mellivora*	*RootTheBox*	*CTFd*
Тип системи	Платформа для організації та проведення *CTF*-змагань у реалістичних лабораторіях	Відкритий *CTF*-рушій *Jeopardy* типу	Відкритий *CTF*-рушій з гейміфікацією	Відкритий *CTF*-движок для *Jeopardy*-типу
Основне призначення	Потужні змагання з комплексними завданнями, лабораторії, кібер-симуляції	Створення та проведення базових *Jeopardy-CTF*	Створення інтерактивних, ігрових *CTF*	Організація змагань *CTF* з підтримкою автоматизованої перевірки
Архітектура	Контейнерна (*Docker/Kubernetes* / ізольовані середовища)	*PHP*-базований сервер	*Python* (*Flask*) з розширеною логікою	*Python* (*Flask*)
Гнучкість налаштування	Висока (реалістичні середовища)	Середня (категорії, прапори)	Висока (ігрові механіки, внутрішня валюта, події)	Висока (плагіни/ розширення)

Продовження таблиці 2.1

Автоматичне оцінювання	Так, інтегровано у задачі	Так, стандартне *Jeopardy*	Так	Так
Підтримка гейміфікації	Обмежена або залежить від реалізації	Немає	Так (валюта, *SWAT*, анімації)	Обмежена (через плагіни)
Підтримка команд/ індивідуальної участі	Так	Так	Так	Так
Можливість масштабування	Висока (контейнери, оркестрація)	Середня	Середня-висока	Висока
Придатність для навчального процесу	Висока (реалістичні практичні завдання)	Середня (основи *CTF*)	Висока (ігровий мотиваційний формат)	Висока (гнучкість та розширюваність)
Обмеження / недоліки	Вищі вимоги до інфраструктури	Обмежена візуалізація	Потрібні технічні налаштування	Іноді потребує додаткових плагінів

Висновки по розділу

Проведений аналіз основних аспектів організації CTF-змагань дав змогу окреслити ключові принципи роботи таких подій, визначити їхні можливості та освітню цінність. Розглянуті тематичні напрямки – аналіз мережевого трафіку, дослідження HTTP-комунікацій та виявлення вразливостей веб-застосунків – були обґрунтовані як найбільш актуальні для навчальних цілей та сучасних викликів кіберзахисту. Додатково було здійснено порівняння існуючих платформ для проведення CTF-змагань, що дозволило визначити найбільш оптимальні технічні рішення для реалізації навчального хакатону.

3 ПІДГОТОВКА ПЛАТФОРМИ ПРОВЕДЕННЯ ХАКАТОНУ

3.1 Розробка завдань хакатону

3.1.1 Аналіз трафіку

3.1.1.1 Легкий рівень завдань

Опис для учасників:

Адміністратор помітив дивний мережевий трафік на тестовому сервері. Підозрюється, що хтось передав секретну інформацію через мережу без шифрування. Вам надано файл з мережевим трафіком. Проаналізуйте його та знайдіть прихований прапор. Формат прапора: *CTF*{...}

Надані файли: *easy_traffic.pcap*

Рекомендований порядок розв'язку для учасника:

1. відкрити файл *easy_traffic.pcap* у *Wireshark*;

2. переглянути мережеві протоколи та знайти підозрілий трафік;

3. відновити переданий текст.

4. знайти прапор.

Очікувані навички від учасників:

Базові навички роботи з *Wireshark*, розуміння *HTTP* або *TCP*, вміння переглядати *payload* пакетів.

Логіка завдання:

Зловмисник передає *flag* у відкритому вигляді через *HTTP*-запит. Дані не зашифровані та легко читаються у тілі *HTTP*-пакета. У дампі присутній мінімальний фоновий трафік.

Інструкція для розробника завдання.

Крок 1. Вибір способу передачі даних. Для легкого рівня використовується *HTTP* без шифрування. *Flag* передається у параметрі *GET*-запиту або в тілі *POST*-запиту у відкритому вигляді.

Крок 2. Створення прапора. Приклад флага: *CTF*{*http_is_too_simple*}

Крок 3. Формування HTTP-запиту. Приклад GET-запиту з сформованим прапором:

GET /index.html?data=CTF{http_is_too_simple} HTTP/1.1

Host: test.local

Крок 4. Генерація трафіка. Можна використати браузер, *curl* або *Scapy* для створення одного або кількох *HTTP*-запитів. Запити зберігаються у файл *easy_traffic.pcap*.

Крок 5. Додавання мінімального шуму. Додайте кілька звичайних *HTTP*-запитів (наприклад, запит до головної сторінки сайту без параметрів), щоб трафік не виглядав порожнім, але залишався простим для аналізу.

Крок 6. Перевірка завдання.

Відкрийте *pcap* у *Wireshark*.

Використайте фільтр *http*.

Перегляньте *HTTP*-запити та знайдіть параметр, що містить flag.

Переконайтеся, що *flag* легко читається у вигляді *ASCII*-тексту.

3.1.1.2 Середній рівень завдань

Опис для учасників:

SOC-аналітики зафіксували підозрілу активність у внутрішній мережі. Є підозра, що зловмисник ексфільтрував секретні дані, маскуючи їх під звичайний мережевий трафік. Вам надано дамп мережевого трафіка. Необхідно проаналізувати його та знайти секретний рядок (*flag*), який було викрадено.

Інструкція для учасника:

– проаналізувати файл *traffic.pcap*.

– визначити, який протокол використано для ексфільтрації даних.

– відновити передані фрагменти даних.

– декодувати їх та отримати *flag*.

Очікувані навички:

Робота з *Wireshark* або *tshark*, аналіз *DNS* або *HTTP*-трафіка, розуміння кодування *Base*64, реконструкція даних із мережевих пакетів.

Логіка завдання:

Зловмисник використовує *DNS*-запити для передачі даних. Частини секрету передаються у вигляді піддоменів у *DNS*-запитах. Дані попередньо закодовані у *Base*64 та розбиті на кілька фрагментів. У дампі присутній фоновий легітимний трафік, який маскує ексфільтрацію.

Приклад правильного флага: *CTF{dns_is_not_that_innocent}*

Інструкція для розробника завдання.

Крок 1. Вибір методу ексфільтрації. Рекомендується використати *DNS*-ексфільтрацію через піддомени, оскільки вона виглядає реалістично та добре підходить для *CTF*-завдань середньої складності.

Крок 2. Створення флага та кодування прапору. Приклад прапору: *CTF{dns_is_not_that_innocent}*

Закодуйте *flag* у *Base*64, наприклад за допомогою команди:

echo -n "CTF{dns_is_not_that_innocent}" | base64

Результат кодування (приклад):

Q1RGe2Ruc19pc19ub3RfdGhhdF9pbm5vY2VudH0=

Крок 3. Розбиття даних на частини.

Розбийте *Base*64-рядок на фрагменти по 8–10 символів, наприклад:

Q1RGe2Ru

c19pc19u

b3RfdGhh

dF9pbm5v

Y2VudH0=

Крок 4. Формування *DNS*-запитів. Кожен фрагмент використовується як піддомен. Формат *DNS*-запиту:

<chunk>.exfil.attacker.com

Приклад:

Q1RGe2Ru.exfil.attacker.com

Крок 5. Генерація *PCAP*-файлу. Для генерації трафіка можна використати *Scapy* (*Python*). Створіть *DNS*-запити до публічного *DNS*-сервера (наприклад,

8.8.8.8), де ім’я запиту містить фрагмент даних. Збережіть усі пакети у файл *traffic.pcap*.

Крок 6. Додавання шуму. Додайте звичайний трафік, наприклад *DNS*-запити до популярних доменів, *HTTP*-запити, *ARP* або *NTP*. Це ускладнить аналіз та зробить завдання більш реалістичним.

Крок 7. Перевірка рішення.

– відкрийте *traffic.pcap* у *Wireshark*;

– використайте фільтр для *DNS*-трафіка;

– знайдіть підозрілі *DNS*-запити з довгими або нетиповими піддоменами;

– скопіюйте фрагменти, об’єднайте їх у правильному порядку;

– декодуйте *Base*64 та перевірте, що отриманий рядок відповідає формату прапора.

3.1.1.3 Складний рівень

Опис для учасників:

Під час розслідування складного інциденту було виявлено підозрілу активність у мережі, яка на перший погляд виглядає як звичайний трафік. Є підозра, що зловмисник використав прихований канал зв’язку для ексфільтрації секретних даних. Вам надано дамп мережевого трафіка. Знайдіть і відновіть прихований прапора. Формат флага: *CTF*{...}

Надані файли: *hard_traffic.pcap*

Завдання для учасника:

– проаналізувати всі типи трафіка у файлі *hard_traffic.pcap*.

– виявити прихований канал передачі даних.

– відновити закодовані фрагменти інформації.

– розшифрувати їх та отримати *flag*.

Очікувані навички:

Глибоке розуміння мережевих протоколів (*ICMP, DNS, TCP*), робота з *Wireshark/tshark*, аналіз нетипових полів пакетів, знання кодувань (*Hex, Base*64), базове криптографічне мислення.

Логіка завдання:

Зловмисник використовує *ICMP Echo Request* для передачі даних, але *payload* не містить читабельного тексту. Дані передаються у вигляді *XOR*-зашифрованих фрагментів, які додатково замасковані під «пінг» стандартного розміру. Частини передаються у довільному порядку, а порядок відновлюється за *ICMP sequence number*.

Інструкція для розробника завдання.

Крок 1. Вибір прихованого каналу.

Для складного рівня рекомендується використати *ICMP Echo Request/Reply* як прихований канал, оскільки такий трафік часто дозволений у мережах і рідко детально аналізується.

Крок 2. Створення прапору. Приклад флага: *CTF*{*icmp_is_more_dangerous_than_it_seems*}

Крок 3. Попередня обробка даних.

Закодуйте *flag* у *Base*64.

Зашифруйте результат за допомогою *XOR* з однобайтовим ключем (наприклад, 0x42).

Перетворіть отримані байти у *hex*-рядок або залиште у сирому вигляді.

Крок 4. Розбиття на фрагменти.

Розбийте зашифровані дані на рівні блоки (наприклад, по 8 байтів). Призначте кожному фрагменту унікальний *ICMP sequence number*, який відповідає порядку відновлення.

Крок 5. Генерація *ICMP*-трафіка.

Згенеруйте *ICMP Echo Request* пакети з нормальними інтервалами.

У *payload* кожного пакета вставте один фрагмент даних. Використовуйте реалістичні IP-адреси та розмір *payload*, щоб трафік виглядав як звичайний *ping*.

Крок 6. Додавання складного шуму. Це ускладнить виявлення прихованого каналу.

Крок 7. Перевірка рішення.

3.1.2 Аналіз веб сторінок

3.1.2.1 Загублений *HTTP*-запит

Опис для учасників:

Адміністратор веб-сервера повідомив, що один із користувачів надсилав підозрілий *HTTP*-запит до внутрішнього *API*. Пакетний дамп трафіку вдалося зберегти, але адміністратор не зміг знайти, що саме в користувацькому запиті було небезпечним. Ваше завдання – переглянути *HTTP*-трафік і знайти флаг, який був переданий у нестандартному *HTTP*-заголовку.

Файли, які надаються гравцю *lost-request.pcap*.

Інструкція для учасника:

1. відкрити файл *lost-request.pcap* у *Wireshark*;

2. відфільтрувати запити *HTTP*;

3. знайти підозрілий заголовок.

4. знайти прапор.

Інструкція для організатора CTF:

Крок 1. Створи запит з кастомним HTTP-заголовком:

curl -H "X-Flag: FLAG{http_hidden_header_42}" http://127.0.0.1/test > /dev/null

Крок 2. Додавання більше Зніми трафік:

sudo tcpdump -i lo -w lost-request.pcap port 80

Крок 6. Додавання шуму. Додайте звичайний трафік, наприклад *DNS*-запити до популярних доменів, *HTTP*-запити, *ARP* або *NTP*. Це ускладнить аналіз та зробить завдання більш реалістичним.

3.1.2.2 Декодування сторінки

Під час аналізу *HTML*-файлу було помічено дивні рядки, які не виглядають як звичайний текст. Імовірно, інформація була прихована за допомогою кодування. Проаналізуйте файл та відновіть прапор.

Надані файли: *secret.html*

Завдання для учасника:

– проаналізувати структуру *HTML*-файлу.

– знайти закодований вміст.

– декодувати знайдені дані.

– отримати *flag*.

Очікувані навички:

Робота з *HTML*, розуміння *Base*64 або *Hex*, уважність до нетипових елементів.

Логіка завдання:

Flag закодований у *Base*64 та вбудований у *HTML*-файл як значення атрибуту (*data*-*, *value, title*) або у *JavaScript*-змінній всередині <*script*>.

Інструкція для розробника завдання:

Закодуйте *flag* у *Base*64. Вставте закодований рядок у *HTML*-атрибут або у *JavaScript*-код. Не залишайте прямої підказки, але зробіть так, щоб закодований рядок виглядав нетипово для *HTML*.

3.1.2.3 Обфускація *DOM*. Складний рівень

Опис для учасників:

Веб-сторінка виглядає абсолютно звичайною, але є підозра, що вона містить приховану логіку для зберігання секретних даних. Вам надано *HTML*-файл. Проаналізуйте його, знайдіть прихований механізм та відновіть *flag*.

Надані файли: *obfuscated.html*

Завдання для учасника:

– проаналізувати *HTML*-структуру та пов’язаний *JavaScript*.

– виявити приховані або динамічно створювані елементи.

– відновити закодовані та обфусковані дані.

– розшифрувати та отримати *flag*.

Очікувані навички:

Глибоке розуміння *HTML* та *DOM*, аналіз *JavaScript*, знання *Base*64, *XOR* або простих обфускацій.

Логіка завдання:

Flag розбитий на частини та зберігається у *HTML*-атрибутах або масиві *JavaScript*. Дані додатково обфусковані (наприклад, *XOR* або реверс рядка) і

збираються лише після виконання *JavaScript*-коду, але не виводяться на сторінку.

Інструкція для розробника завдання:

Розбийте *flag* на кілька частин. Обфускуйте кожну частину. Збережіть їх у *data*-атрибутах або *JavaScript*-масиві. Напишіть *JS*-код, який збирає *flag*, але не відображає його. Учасник повинен відновити логіку вручну або через консоль.

3.2 Розгортання платформи та завантаження завдань

Для розгортання системи проведення хакатону використовується сервер кафедри з встановленим дистрибутивом *Linux Debian* [34].

Першим етапом є підготування сервера для розгортання системи для цього в командному рядку використовуються наступні команди:

sudo apt update

sudo apt upgrade -y

sudo apt install -y git python3 python3-venv python3-dev build-essential libpq-dev libffi-dev libssl-dev apache2 libapache2-mod-proxy-html libxml2-dev curl wget

де:

sudo apt update – оновлення списку пакетів;

sudo apt upgrade -y – оновлення всіх пакетів до останніх версій;

sudo apt install -y ... – встановлення необхідних пакетів:

– *git* – для клонування репозиторію *RootTheBox*

– *python3* та *python3-venv* – для створення віртуального середовища

– *python3-dev*, *build-essential, libpq-dev, libffi-dev, libssl-dev* – бібліотеки для компіляції *Python*-залежностей

– *apache2, libapache2-mod-proxy-html, libxml2-dev* – веб-сервер *Apache* та модулі для проксі

– *curl, wget* – утиліти для завантаження файлів

Для зберігання даних використовується *PostgreSQL*. Процес встановлення наступний:

sudo apt install -y postgresql postgresql-contrib

Вхід в консоль *PostgreSQL* як користувач postgres:

sudo -u postgres psql

Створюємо користувача і базу даних для *RootTheBox*:

CREATE USER rtb_user WITH PASSWORD 'strongpassword';
CREATE DATABASE rtb_db OWNER rtb_user;
\q

Клонування *RootTheBox* та створення віртуального середовища:

git clone https://github.com/moloch--/RootTheBox.git
cd RootTheBox
python3 -m venv venv
source venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt

де:

git clone – завантажує код *RootTheBox*;
python3 -m venv venv – створює віртуальне середовище *Python*.
source venv/bin/activate – активує середовище.
pip install -r requirements.txt – встановлює всі залежності *Python*.

Для збереження стандартної конфігурації *RootTheBox* копіюємо шаблон конфігураційного файлу:

cp config/ctf.config.example config/ctf.config
nano config/ctf.config

Встановлюємо параметри підключення до бази даних:

db_type = postgres
db_name = rtb_db
db_user = rtb_user
db_password = strongpassword
db_host = 127.0.0.1

Ініціалізація структури бази даних *RootTheBox* виконується наступними командами:

python3 manage.py db init
python3 manage.py db migrate
python3 manage.py db upgrade

Включаємо необхідні модулі *Apache* для проксі *HTTP, WebSocket, SSL* та перезапускаємо сервер:

sudo a2enmod proxy proxy_http proxy_wstunnel headers rewrite ssl
sudo systemctl restart apache2

Створюємо легкий контейнер з *RootTheBox*:

Dockerfile:

```
FROM python:3.11-slim
WORKDIR /app
COPY . .
RUN pip install --upgrade pip
RUN pip install -r requirements.txt
CMD ["python3", "RootTheBox.py"]
```

CMD запускає платформу при старті контейнера.

Створимо пакет з *Deployment, Service* та *ConfigMap* для *Apache Ingress*. Файл *postgres-deployment.yaml* буде мати вигляд:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rtb-postgres
spec:
  replicas: 1
  selector:
    matchLabels:
      app: rtb-postgres
  template:
    metadata:
      labels:
        app: rtb-postgres
    spec:
      containers:
      - name: postgres
        image: postgres:15
        ports:
```

```
        - containerPort: 5432
        env:
        - name: POSTGRES_DB
          value: rtb_db
        - name: POSTGRES_USER
          value: nuos_admin
        - name: POSTGRES_PASSWORD
          value: strongpassword
        volumeMounts:
        - name: pgdata
          mountPath: /var/lib/postgresql/data
      volumes:
      - name: pgdata
        emptyDir: {}
---
apiVersion: v1
kind: Service
metadata:
  name: rtb-postgres
spec:
  selector:
    app: rtb-postgres
  ports:
    - protocol: TCP
      port: 5432
      targetPort: 5432
```

Створюємо *Deployment PostgreSQL* для *RootTheBox*. Використовується користувач бази даних *nuos_admin*. *EmptyDir* використовується для простоти,

для продакшену бажано *PersistentVolume*. Файл конфігурації *rootthebox-deployment.yaml* буде мати вигляд:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rootthebox
spec:
  replicas: 1
  selector:
    matchLabels:
      app: rootthebox
  template:
    metadata:
      labels:
        app: rootthebox
    spec:
      containers:
      - name: rootthebox
        image: yourdockerhub/rootthebox:latest
        ports:
        - containerPort: 8888
        env:
        - name: DB_HOST
          value: rtb-postgres
        - name: DB_USER
          value: nuos_admin
        - name: DB_PASSWORD
          value: strongpassword
        - name: DB_NAME
```

```
        value: rtb_db
      - name: RTB_ADMIN_USER
        value: nuos_admin
      - name: RTB_ADMIN_PASS
        value: AdminStrongPassword123
---
apiVersion: v1
kind: Service
metadata:
  name: rootthebox
spec:
  selector:
    app: rootthebox
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8888
  type: ClusterIP
```

Налаштування *Apache Ingress ConfigMap* (на вузлі *Ingress*):

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: apache-ingress-config
data:
  rtb.conf: |
    <VirtualHost *:80>
        ServerName rtb.example.com
        ProxyPreserveHost On
```

ProxyRequests Off
ProxyPass / http://<ClusterIP_RootTheBox_Service>:80/
ProxyPassReverse / http://<ClusterIP_RootTheBox_Service>:80/
ErrorLog ${APACHE_LOG_DIR}/rtb_error.log
CustomLog ${APACHE_LOG_DIR}/rtb_access.log combined
</VirtualHost>

Після застосування *ConfigMap*, потрібно змонтувати її на сервер *Apache* та перезапустити службу.

Встановлення *certbot* для *HTTPS* на вузлі *Ingress*:

sudo apt install -y certbot python3-certbot-apache
sudo certbot --apache -d rtb.example.com

Certbot автоматично налаштовує *SSL* для *Apache*.

Для розгортання в *Kubernetes*. Застосовуємо базу даних:

kubectl apply -f postgres-deployment.yaml

Застосовуємо *RootTheBox*:

kubectl apply -f rootthebox-deployment.yaml

Перевірка подів:

kubectl get pods
kubectl get svc

Після цього на вузлі *Ingress Apache* підхопить *ConfigMap* та забезпечить доступ через *http://rtb.example.com* або *HTTPS*.

Ініціалізація *RootTheBox*:

kubectl exec -it <rtb-pod-name> -- /bin/bash
source venv/bin/activate
python3 manage.py db init
python3 manage.py db migrate
python3 manage.py db upgrade

Далі можна переходити за посиланням *localhost*:8888.

Висновки до розділу

Було розроблено приклади задач різного рівня складності, описано підходи до структурування та обфускації даних, а також сформовано методичні рекомендації щодо створення навчальних викликів. Після цього виконано розгортання та налаштування платформи RootTheBox на сервері кафедри, що забезпечило повноцінне середовище для проведення хакатону.

4 ТЕСТУВАННЯ СИСТЕМИ

4.1 Основні компоненти системи *RootTheBox*

4.1.1 Корпорації (групи)

Корпорація, яка є необов'язковою, – це група або набір блоків. Наприклад, корпорація може мати набір прапорців з блоку веб-сервера та блоку сервера бази даних. Адміністраторам рекомендується організовувати свої завдання з урахуванням цього (наприклад, повторне використання паролів між блоками в межах однієї корпорації). Корпорації дозволяють групувати цілі мережі машин, а не лише один блок. Якщо ви не хочете використовувати групи корпорацій, ви можете створити одну корпорацію з порожньою назвою – тоді *RTB* приховає цей шар [35].

4.1.2 Прапори (питання)

Прапор – це питання, на яке команди повинні відповісти, або певна інформація, яку потрібно отримати. Команди отримують гроші/бали після успішного захоплення прапора. Прапори можуть мати підказки або застосовувати штрафи.

4.1.3 Коробки (секції).

Блок – це набір прапорців, що представляють певний тип інформації, яку має отримати команда (рис. 4.1). Кожен блок належить корпорації та представляє хост, який команди можуть атакувати. Крім того, кожен блок може представляти питання щодо певних цифрових доказів, пов’язаних із певною категорією, наприклад, аналіз пам’яті наданого зображення.

Тип подання прапорця в коробці можна налаштувати як *Classic* або *Single Box*. У класичному режимі кожен прапор має спосіб подання. Гравець обере прапор, який він хоче захопити. В одиночному режимі є одне поле для подання, яке порівнює спробу з усіма прапорцями в коробці.

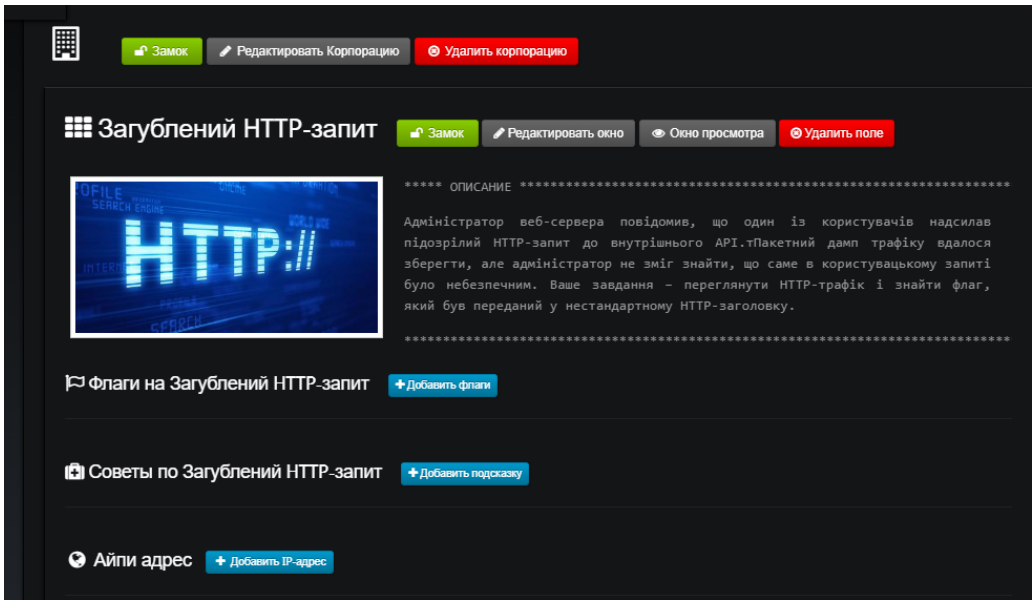

Рисунок 4.1 – Вигляд створеної коробки

4.1.4 Рівні гри.

За замовчуванням усі команди/гравці починають з 0-го рівня (рис. 4.2). Додаткові рівні можна розблокувати, захопивши певний відсоток прапорів на поточному рівні, сплативши гроші за розблокування вищих рівнів, отримавши певну кількість очок або керуючи доступом, який контролює адміністратор. Гравці не можуть бачити або надсилати прапори для рівнів, які вони не розблокували.

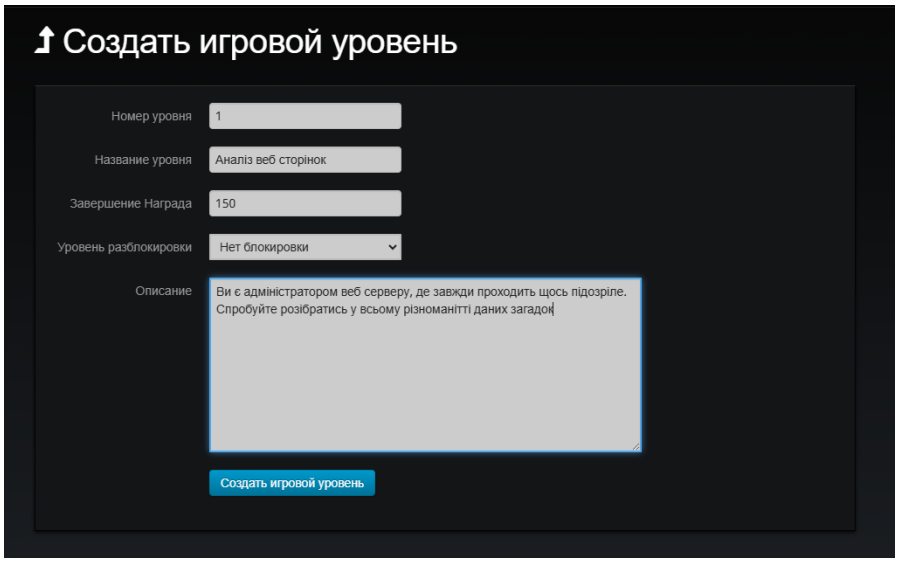

Рисунок 4.2 – Меню створення рівня

4.1.5 Категорії.

Категорії можна застосовувати до блоків, що надає гравцеві доступ до графіка навичок, що показує його прогрес у різних категоріях (рис. 4.3). Це дозволяє гравцям чіткіше бачити свої сильні сторони. Категорії необов'язкові.

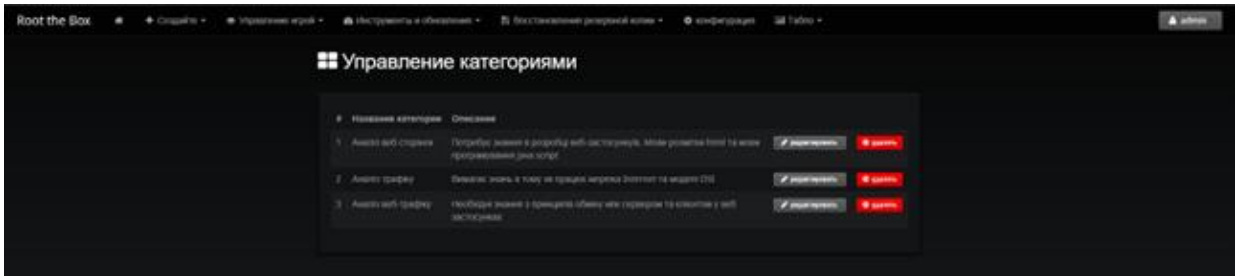

Рисунок 4.3 – Меню керування категоріями

4.1.6 Боти

Для ботів адміністратор генерує файл сміття в системі підрахунку очок. Коли ви створюєте систему, ви повинні мати можливість завантажити її копію та завантажити на цільовий(і) сервер(и). Цей файл, по суті, містить деякі випадкові дані, які є частиною доказу з нульовим розголошенням, що підтверджує, що бот має доступ до цільових машин і не працює на системі поза межами гри.

4.1.7 Ігрові матеріали

Файли, документи або програми, що використовуються гравцями в грі, можна розмістити в папці «Ігрові матеріали» (*/files/game_materials*). *RootTheBox* підтримує структуру папок, тому впорядкуйте свої матеріали відповідно до гри. Наприклад, ви можете створити каталог для інструментів, які можуть використовувати гравці (щоб вони могли завантажувати їх безпосередньо), додати документ для мережі, на яку вони збираються здійснити атаку, або перерахувати докази, які вони повинні дослідити, щоб знайти прапорець. Під час створення поля, прапорця або підказки посилайтеся на відповідні ігрові матеріали, які можуть знадобитися користувачеві. Коли опція конфігурації *use_box_materials_dir* ввімкнена, папка «Ігрові матеріали» з назвою, що відповідає імені поля, буде автоматично пов’язана з полем.

4.2 Керування системою *RootTheBox*

4.2.1 Панель адміністратора платформи *RootTheBox*

Панель адміністратора платформи *RootTheBox* призначена для керування процесом проведення змагань типу CTF. Вона забезпечує централізований доступ до функцій налаштування гри, управління користувачами та створення ігрового контенту (рис. 4.4).

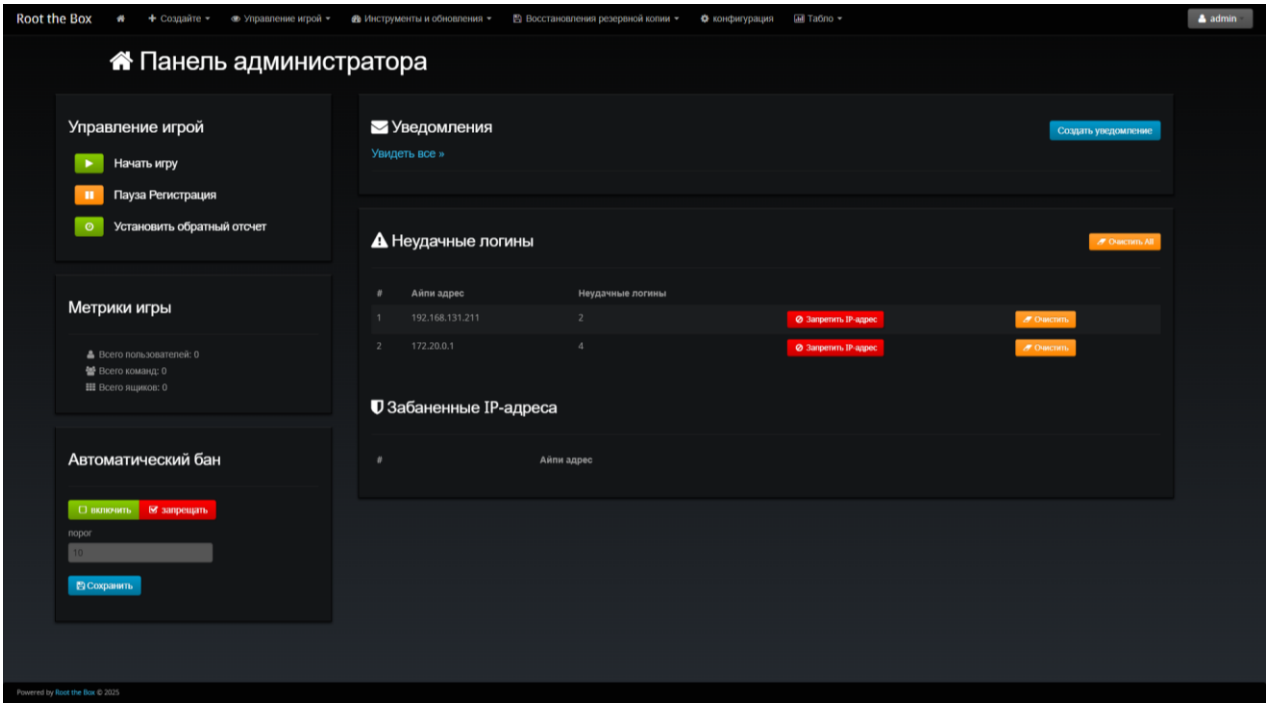

Рисунок 4.4 – Панель адміністратора

Через меню «Создать» адміністратор може створювати основні ігрові елементи, зокрема команди учасників, корпорації, категорії завдань, ігрові завдання (коробки) та відповідні флаги. Завдання описуються текстово та логічно структуруються за категоріями і рівнями складності. Перевірка правильності виконання завдань здійснюється шляхом введення прапорів.

4.2.2 Керування користувачами та командами

Панель керування користувачами дозволяє переглядати список користувачів, організовувати їх у команди, а також контролювати участь гравців

у змаганні (рис. 4.5). Адміністративні облікові записи надають розширені права доступу для керування ігровим середовищем.

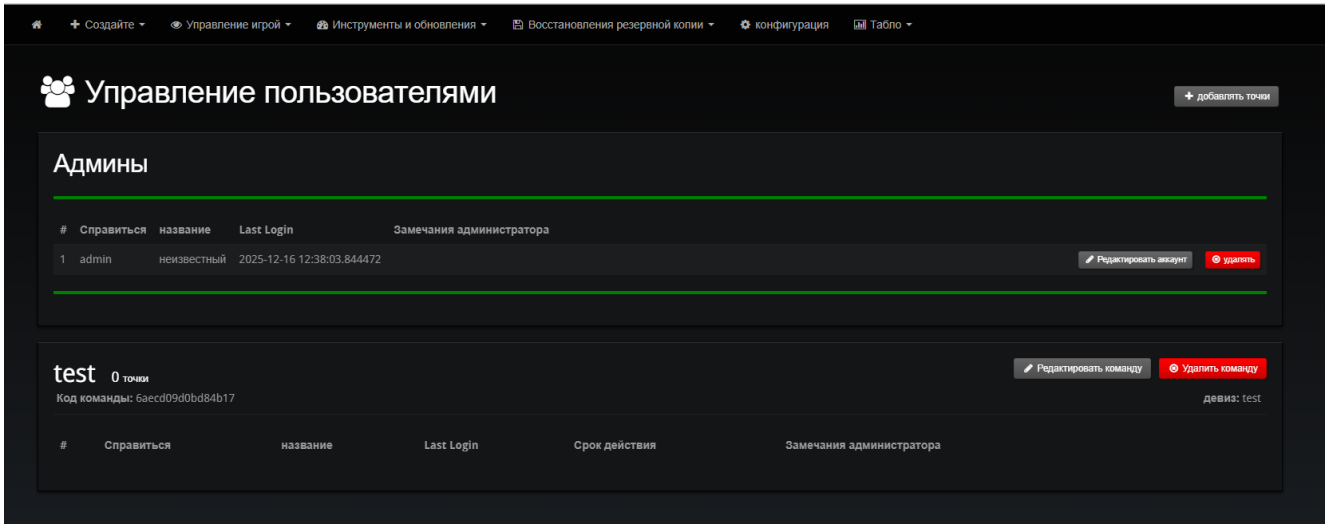

Рисунок 4.5 – Панель керування користувачами

4.2.3 Статистика та графіки

Платформа надає інструменти візуалізації статистичних даних, зокрема рейтинги команд, прогрес виконання завдань та динаміку набору балів. Графічне відображення статистики дозволяє оцінювати активність учасників та ефективність проведення змагання в режимі реального часу (рис. 4.6).

4.2.4 Налаштування та адміністрування системи

Загальні параметри роботи платформи, такі як стан гри, конфігурація бази даних та серверні налаштування, керуються через адміністративні інструменти та конфігураційні файли платформи. Це дозволяє гнучко адаптувати середовище під вимоги конкретного змагання або навчального курсу.

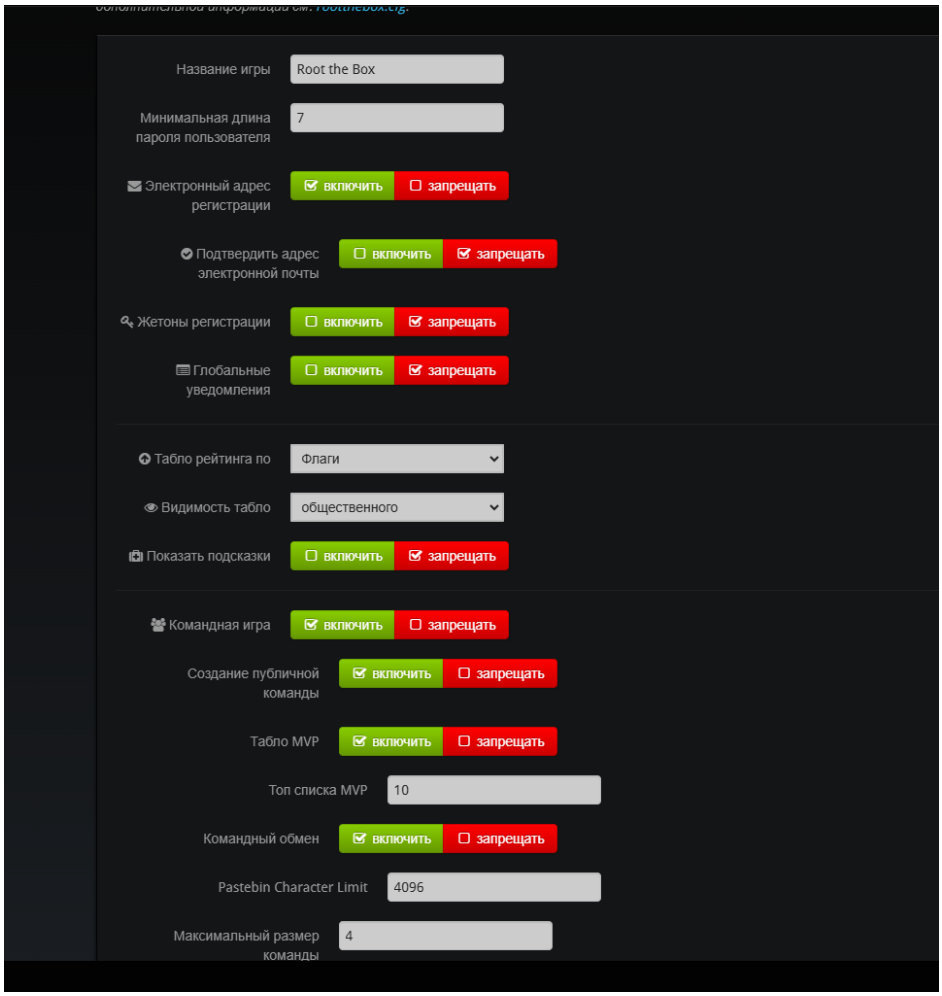

Рисунок 4.6 – Меню налаштувань гри

Таким чином, панель адміністратора *RTB* забезпечує повний набір засобів для організації та контролю *CTF*-змагань, що робить платформу доцільною для використання у навчальному процесі з інформаційної безпеки.

Висновки до розділу

Після цього виконано розгортання та налаштування платформи RootTheBox на сервері кафедри, що забезпечило повноцінне середовище для проведення хакатону. В процесі тестування платформи підтверджено її відповідність вимогам навчального CTF: зручність адміністрування, широкі можливості створення завдань, гнучкість конфігурації та наявність системи візуалізації результатів.

ВИСНОВКИ

Кваліфікаційна робота присвячена актуальному питанню – проведення хакатонів в сфері кіберзахисту. У ході виконання кваліфікаційної роботи було визначено принципи та етапи проведення хакатонів та визначено переваги використання хакатону в форматі *CTF* для задач навчання.

В рамках кваліфікаційної роботи було зроблено огляд напрямків роботи хакатону, зокрема аналіз трафіку та огляд вразливостей веб-застосунків. Також було складено інструкції для створення завдань по даним тематикам для різного рівня складності.

Для проведення *CTF* змагань було проведено аналіз існуючих платформ та розгорнуто платформу *RootTheBox* на сервері кафедри.

ПЕРЕЛІК ПОСИЛАНЬ

1. Eileen Becker What is a Hackathon? URL: The Ultimate Guide to Innovation Sprints https://www.innosabi.com/resources/post/what-is-a-hackathon-in-business?utm_source=chatgpt.com

2. Про захід хакатон URL: https://foxminded.ua/hakaton-shcho-tse/

3. Кучер Т. Що таке хакатон і чим він може бути корисним. Hight bar journal URL: https://journal.gen.tech/post/shcho-take-hakaton;

4. CTF for Beginners: What is CTF and how to get started!: веб-сайт. URL: https://dev.to/atan/what-is-ctf-and-how-to-get-started-3f04

5. Змагання CTF – ключ до успішної кар'єри у сфері ... : веб-сайт. URL: https://dou.ua/forums/topic/43810/

6. Capture the flag (CTF): веб-сайт. URL: https://uk.wikipedia.org/wiki/Capture_the_flag_(CTF)

7. CTF: веб-сайт. URL: https://www.linkedin.com/pulse/ctf-digialert

8. Types of CTF challenges: веб-сайт. URL: https://snyk.io/series/ctf/ctf-types/

9. CTFtime.org / What is Capture The Flag?: веб-сайт. URL: https://ctftime.org/ctf-wtf/

10. CTF Challenge Categories: веб-сайт. URL: https://docs.ctfd.io/events/challenge-categories/

11. Бурячок В. Л., Аносов А. О., Семко В. В., Соколов В. Ю., Складанний П. М. Технології забезпечення безпеки мережевої інфраструктури, К.: КУБГ, 2019. 218 с.

12. Marwan Al-shawi, Andre Laurent, Designing for Cisco Network Service Architectures (ARCH) Foundation Learning Guide, Fourth Edition. Cisco Press. 2017. 902 p.

13. Коробейнікова Т. І., Захарченко С. М. Комп’ютерні мережі Навчальний посібник. Львів: Видавництво Львівської політехніки, 2022. 228 с.

14. Буров, Є.В. Компютерні мережі: Підручник [Текст] // Є.В. Буров/. – Львів: «Магнолія 2006», 2008. – 262 с.

15. Sanders, C. Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems (3rd ed.). No Starch Press. 2017. 284p

16. Chappell, L. Wireshark Network Analysis (2nd ed.). Protocol Analysis Institute. 2010. 460 p. URL: https://elhacker.info/manuales/Redes/wireshark-network-analysis-second-edition.pdf

17. Sanders, C., & Smith, J. Applied Network Security Monitoring: Collection, Detection, and Analysis. No Starch Press. 2013. 672p. URL: https://studylib.net/doc/27357801/applied-network-security-monitoring--collection--detectio...

18. Leslie F. Sikos, Packet analysis for network forensics: A comprehensive survey. Forensic Science International: Digital Investigation, Volume 32, 2020. URL: https://doi.org/10.1016/j.fsidi.2019.200892.

19. Gourley, D., & Totty, B. (). HTTP: The Definitive Guide. O'Reilly Media. 2002. 658 p. URL: https://dl.ebooksworld.ir/books/HTTP.The. Definitive.Guide. Brian.Totty.David.Gourley.OReilly.9781565925090.EBooksWorld.ir.pdf

20. Andrew S. Tanenbaum, Todd Austin. Structured computer organization, 6th ed. Published by Pearson 2012, 801 p.

21. Scheible, T. (2020). Influence of HTTP Header Entries on the Forensic Analysis of Web Browser Artifacts. Conference presentation/paper, ITSec. URL: https://publikationen.uni-tuebingen.de/xmlui/bitstream/handle/10900/100434/Scientific-presentation_Scheible%20v2.pdf?sequence=1&isAllowed=y

22. Ezennaya-Gomez, Salatiel & Kiltz, Stefan & Kraetzer, Christian & Dittmann, Jana. A Semi-Automated HTTP Traffic Analysis for Online Payments for Empowering Security, Forensics and Privacy Analysis. 2021. 1-8. 10.1145/3465481.3470114.

23. Stuttard, D., & Pinto, M. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws (2nd ed.). Wiley. 2011. 770 p.

24. Mozilla Developer Network (MDN). (n.d.). Cross-site scripting (XSS). MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Glossary/Cross-site_scripting

25. Hack The Box: веб-сайт. URL: https://univd.edu.ua/uk/news/8668

26. SQL ін'єкції: що це, як працює, які типи існують, небезпечність: вебсайт. URL: https://foxminded.ua/sql-iniektsii/

27. Тестування безпеки: SQL-ін'єкції. : веб-сайт. URL: https://training.qatestlab.com/blog/technical-articles/security-testing-sql-injection/

28. Захист веб-додатків від SQL-ін'єкцій: веб-сайт. URL: https://openarchive.nure.ua/server/api/core/bitstreams/4bc35e97-c6a3-4155-bc68-70591fe4fd27/content

29. Semmy Purewal. Learning Web App Development: Build Quickly with Proven JavaScript Techniques. O'Reilly Media, Inc. 2014. 272

30. CTF101: A guide to Capture The Flag. URL: https://ctf101.org/

31. picoCTF. (n.d.). picoCTF: Writing challenges / challenge design guide. https://picoctf.org/

32. Best CTF Platforms To Learn (Capture The Flag) ... : веб-сайт. URL: https://www.craw.sg/best-ctf-platforms-to-learn/

33. Review of 5 Platforms to Run Your CTF by Cyberseclabs: веб-сайт. URL: https://www.cyberseclabs.org/review-of-5-platforms-to-run-your-ctf-bycyberseclabs/

34. RootTheBox. Instalation URL: https://github.com/moloch--/RootTheBox /wiki/ Installation

35. RootTheBox.Game Setup Tutorial URL: https://github.com/moloch--/RootTheBox/wiki/Game-Setup-Tutorial