

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Інститут післядипломної освіти

Кваліфікаційна робота бакалавра

на тему: Тестування на проникнення та уразливості інформаційної системи ОДЕКУ

Виконав студент групи КН-5
спеціальності 122 Комп'ютерні науки
Емінзаде Турал Захід огли

Керівник _____ ст. викладач
Вохменцева Т.Б.

Консультант _____ д.ф., доцент
Бучинська І.В.

Рецензент канд.техн.наук., доцент
Перелигін Б.В.

Одеса 2023

ЗМІСТ

Терміни, скорочення та умовні позначки	5
Вступ.....	7
1 Опис методології тестування проникнення в мережі інтернет	9
1.1 Характеристика способів веб-тестування на проникнення	10
1.2 Характеристика підходу до тестування	11
2 Аналіз типів тестування на проникнення	20
2.1 Опис автоматизованого та ручного пентестування.....	21
2.2 Опис цілей при pentest	22
2.3 Опис видів pen testing	23
3 Опис етапів тестування на проникнення та уразливості інформаційної системи	27
Висновки	47
Перелік джерел посилання	50
Додаток А Виконання команди wpscan --url https://odeku.edu.ua/ -e u	52
Додаток Б Виконання команди wpscan --url https://odeku.edu.ua/ -e p.....	54
Додаток В Виконання Module options (auxiliary/scanner/portscan/tcp).....	57

ТЕРМІНИ, СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Брандмауер (або Firewall) – це програмний комплекс, який служить для захисту комп'ютера від злому хакерами, а також всіляких вірусів і «троянів».

Вразливість – це термінологія, яка використовується для виявлення недоліків у системі, які можуть наразити систему на загрози безпеці.

Прикладний програмний інтерфейс (інтерфейс програмування застосунків, інтерфейс прикладного програмування, API) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

Сервер (англ. server – «служба», від англ. to serve – служити, множ. сервери) – у комп'ютерній термінології термін може стосуватися окремого комп'ютера чи програми.

Система виявлення атак (вторгнень) – програмний або апаратний засіб, призначений для виявлення фактів несанкціонованого доступу в комп'ютерну систему або мережу або несанкціонованого управління ними в основному через Інтернет.

Тестувальник – це IT-фахівець, який займається тестуванням програмного забезпечення, виявленням та усуненням помилок у програмному коді.

CMS (Content Management System) – це система керування вмістом сайту, на професійному жаргоні CMS ще називають «двигун сайту».

DevSecOps – це методика інтеграції принципів безпеки в конвейер безперервної інтеграції, безперервної поставки та безперервного розвертання.

DNS – це розподілена база даних доволі простої структури.

Pentest – це тест, який зосереджується виключно на веб-додатку, а не на всій мережі чи компанії.

SQL – це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також керування базами даних.

WAPT (Web Application Performance Testing) є засобом для тестування продуктивності, навантаження та стресового тестування вебсайтів і інтернет додатків з вебінтерфейсом.

XSS («міжсайтовий скриптинг») – тип вразливості інтерактивних інформаційних систем у вебі.

API	– Application Programming Interface.
CMS	– Content Management System.
DNS	– Domain Name System.
ERP	– Enterprise Resource Planning.
IDS	– Intrusion Detection System.
ISSAF	– Information System Security Assessment Framework.
OSSTMM	– Open Source Security Testing Methodology Manual.
OWASP	– Open Web Application Security Project .
PCI DSS	– Payment Card Industry Data Security Standard.
PTF	– Penetration Testing Framework.
SQL	– Structured query language.
WAPT	– Web Application Performance Testing.
XSS	– Cross Site Scripting.

ВСТУП

Коли мова заходить про безпеку, найчастіше впливає слово вразливість. Спочатку необхідно розібратися та пояснити різницю між вразливістю та тестуванням.

Мета кваліфікаційної роботи бакалавра полягає у виявленні потенційних слабких місць, помилок або проблем в системі, які можуть бути використані зловмисниками для злому, незаконного доступу, витоку даних або виконання інших шкідливих дій.

Вразливість – це термінологія, яка використовується для виявлення недоліків у системі, які можуть наражати систему на загрози безпеці. Сканування вразливостей дозволяє користувачеві знайти відомі слабкі місця в програмі та визначає методи виправлення та підвищення загальної безпеки програми. По суті, він з'ясовує, чи встановлені патчі безпеки, чи належним чином налаштовані системи для ускладнення атак.

Pentests здебільшого симулює системи в режимі реального часу та допомагає користувачеві з'ясувати, чи можуть до системи отримати доступ неавторизовані користувачі, якщо так, то яку шкоду можна завдати, яким даним тощо. Таким чином, сканування вразливостей є методом виявлення, який пропонує способи вдосконалення програм безпеки та гарантує, що відомі слабкі місця не з'являться знову, тоді як перевірка pentest є методом превентивного контролю, який дає загальне уявлення про існуючий рівень безпеки системи.

Хоча обидва методи мають свою важливість, це залежатиме від того, що насправді очікується в рамках тестування. Тестувальникам, вкрай важливо чітко визначити мету тестування, перш ніж приступати до тестування. Якщо чітко розуміти мету, можна дуже добре визначити, чи потрібно виконувати сканування вразливостей чи pentest.

Важливість і необхідність тестування Web App Pen:

- Pentest допомагає виявити невідомі вразливості;

- допомагає перевірити ефективність загальної політики безпеки;
- допомога в тестуванні відкритих компонентів, таких як брандмауери, маршрутизатори та DNS;
- дозволяє користувачам знайти найбільш вразливий шлях, через який можна здійснити атаку;
- допомагає знайти лазівки, які можуть призвести до крадіжки конфіденційних даних.

Якщо подивитися на поточний попит на ринку, спостерігається різке зростання використання мобільних пристроїв, що стає головним потенціалом для атак. Доступ до вебсайтів через мобільні телефони схильний до більш частих атак і, отже, компрометації даних.

Таким чином, тестування на проникнення стає дуже важливим для створення безпечної системи, якою можуть користуватися користувачі, не боячись злому чи втрати даних.

Дана кваліфікаційна робота бакалавра, складається з 120 сторінок, 14 рисунків та 5 джерел посилання.

1 ОПИС МЕТОДОЛОГІЇ ТЕСТУВАННЯ ПРОНИКНЕННЯ В МЕРЕЖІ ІНТЕРНЕТ

Методологія – це не що інше, як набір інструкцій галузі безпеки щодо того, як має проводитися тестування. Існують деякі загальноприйняті та відомі методології та стандарти, які можна використовувати для тестування, але оскільки кожна веб-програма вимагає виконання різних типів тестів, тестувальники можуть створювати власні методології, посиляючись на стандарти, доступні на ринку.

Деякі з методологій і стандартів тестування безпеки:

- OWASP (відкритий проєкт безпеки веб-додатків);
- OSSTMM (посібник з методології тестування безпеки з відкритим кодом);
- PTF (Penetration Testing Framework);
- ISSAF (система оцінки безпеки інформаційних систем);
- PCI DSS (стандарт безпеки даних індустрії платіжних карток).

Нижче наведено деякі сценарії тестування, які можна протестувати в рамках тестування проникнення веб-додатків (WAPT):

- міжсайтовий сценарій;
- SQL ін'єкція;
- порушена автентифікація та керування сесансами;
- недоліки завантаження файлів;
- атаки на кеш-сервери;
- неправильна конфігурація безпеки;
- підробка міжсайтового запиту;
- злом паролів.

Тестувальники не повинні сліпо створювати власну методологію тестування на основі наведених вище загальноприйнятих стандартів. Якщо мова йде про тестування вебсайту електронної комерції на проникнення, виникає питання, чи можна виявити всі вразливості вебсайту електронної комерції за

допомогою звичайних методів OWASP, таких як XSS, впровадження SQL тощо.

Оскільки електронна комерція працює на зовсім іншій платформі та технології порівняно з іншими вебсайтами. Для того, щоб pentest для вебсайту електронної комерції було ефективним, тестувальники повинні розробити методологію, що містить такі недоліки, як керування замовленнями, керування купонами та винагородами, інтеграція платіжного шлюзу та інтеграція системи керування вмістом.

Отже, перш ніж визначитися з методологією, упевніться, які типи вебсайтів очікується для тестування та які методи допоможуть знайти максимальну вразливість.

1.1 Характеристика способів веб-тестування на проникнення

Веб-програми можна перевірити на проникнення двома способами. Тести можуть бути розроблені для імітації внутрішньої або зовнішньої атаки.

По-перше, внутрішнє тестування на проникнення. Як випливає з назви, внутрішнє pentest виконується в організації через локальну мережу, отже, воно включає тестування веб-додатків, розміщених в інтрамережі. Це допомагає з'ясувати, чи можуть існувати вразливості в корпоративному брендмауері. Логічно вважати, що атаки можуть відбуватися лише ззовні, і часто внутрішній pentest не помічається або не надається великого значення.

В основному це включає в себе зловмисні атаки незадоволених співробітників або підрядників, які б звільнилися, але знають про внутрішню політику безпеки та паролі, атаки соціальної інженерії, симуляцію фішингових атак і атаки з використанням привілеїв користувача або неправильного використання розблокованого терміналу. Тестування в основному виконується шляхом доступу до середовища без належних облікових даних і ідентифікації.

По-друге, зовнішнє тестування на проникнення. Це атаки, які здійснюються ззовні організації та включають тестування веб-додатків, розміщених в

мережі Інтернеті. Тестери поводяться як хакери, які мало знають про внутрішню систему.

Для імітації таких атак тестувальникам надається IP-адреса цільової системи та не надається жодна інша інформація. Вони повинні здійснювати пошук і сканування загальнодоступних веб-сторінок і знаходити нашу інформацію про цільові хости, а потім скомпрометувати знайдені хости. В основному це включає тестування серверів, брандмауерів та IDS.

1.2 Характеристика підходу до тестування

Тестери pentest імітують атаки мотивованих супротивників. Для цього вони зазвичай дотримуються плану, який включає такі кроки, як:

- reconnaissance (розвідка);
- scanning (сканування);
- gaining access (отримання доступу);
- maintaining access (підтримання доступу).

На першому кроці “розвідка”, збирається якомога більше інформації про ціль із публічних і приватних джерел, щоб визначити стратегію атаки. Джерела включають пошук в Інтернеті, пошук інформації про реєстрацію домену, соціальну інженерію, ненав’язливе сканування мережі та іноді навіть занурення у смітник. Ця інформація допомагає тестувальникам pentest визначити поверхню атаки цілі та можливі вразливості. Розвідка може змінюватися в залежності від обсягу та цілей тесту пера; це може бути так само просто, як телефонний дзвінок, щоб ознайомитися з функціями системи.

Тестери, на кроці “сканування” pentest використовують інструменти для перевірки цільового вебсайту чи системи на наявність слабких місць, зокрема відкритих служб, проблем із безпекою додатків і вразливостей із відкритим кодом. Тестери pentest використовують різні інструменти на основі того, що вони знаходять під час розвідки та під час тесту.

На кроці “отримання доступу”, мотивація зловмисника може включати крадіжку, зміну або видалення даних; переміщення коштів; або просто завдає шкоди репутації компанії. Щоб виконати кожен тестовий приклад, тестери пера визначають найкращі інструменти та методи для отримання доступу до системи через слабкі місця, такі як впровадження SQL, або через шкідливе програмне забезпечення, соціальну інженерію чи щось інше.

Після того, як тестери pentest отримують доступ до цілі, їх симульована атака має залишатися на зв'язку достатньо довго, щоб досягти своїх цілей щодо викрадання даних, їх модифікації або зловживання функціональністю. Йдеться про демонстрацію потенційного впливу, це відбувається вже на четвертому кроці “підтримання доступу”.

Підхід до тестування Web Pen, можна проводити в 3 етапи:

- фаза планування (перед тестуванням);
- атаки/фаза виконання (під час тестування);
- фаза після виконання (після тестування).

Перед початком тестування бажано спланувати (фаза планування), які типи тестування будуть проводитися, як тестування буде виконуватися, визначити, чи потрібен додатковий доступ до інструментів для контролю якості тощо.

Визначення обсягу – це те саме, що й функціональне тестування, де визначається обсяг тестування перед початком тестування.

Доступність документації для тестувальників – слід переконайтися, що тестувальники мають усі необхідні документи, як-от документи з детальним описом веб-архітектури, точок інтеграції, інтеграції веб-сервісів тощо. Тестер повинен знати основи протоколу HTTP/HTTPS і знати про архітектуру веб-програми та трафік методи перехоплення.

Далі, слід перейти до визначення критеріїв успіху. На відміну від функціональних тестів, де можна отримати очікувані результати на основі вимог користувача/функціональних вимог, pentest працює за іншою моделлю. Потрібно визначити та затвердити критерії успішності або критерії проходження

тесту. Перегляд результатів попереднього тестування. Якщо попереднє тестування коли-небудь проводилося, варто переглянути результати тестування, щоб зрозуміти, які вразливості існували в минулому та які заходи були вжиті для усунення. Це завжди дає краще уявлення про тестерів. Розуміння навколишнього середовища – тестувальники повинні отримати знання про навколишнє середовище перед початком тестування. Цей крок повинен забезпечити їм розуміння брандмауерів або інших протоколів безпеки, які необхідно вимкнути для виконання тестування. Переглядачі, які підлягають тестуванню, мають бути перетворені на платформу для атаки, зазвичай це робиться шляхом зміни проксі-серверів.

На етапі атаки/фаза виконання, тестування веб-проникнення можна проводити з будь-якого місця, враховуючи той факт, що інтернет-провайдер не повинен накладати обмежень на порти та послуги. Забезпечте виконання тесту з різними ролями користувачів – тестувальники повинні забезпечити виконання тестів з користувачами з різними ролями, оскільки система може поводитися по-різному щодо користувачів з різними привілеями.

Обізнаність про те, як поводитися з постексплуатацією. Щоб повідомити про будь-яку експлуатацію, тестувальники повинні дотримуватися критеріїв успіху, визначених у рамках фази 1. Вони також повинні дотримуватися визначеного процесу звітування про вразливості, виявлені під час тестування. На цьому етапі тестувальник здебільшого з'ясовує, що потрібно зробити після того, як вони виявили, що систему зламану.

Генерація звітів про тестування. Будь-яке тестування без належної звітності мало допомагає організації, те ж саме стосується тестування на проникнення веб-додатків. Щоб забезпечити належний доступ до результатів тестування всім зацікавленим сторонам, тестувальники повинні створювати належні звіти з докладною інформацією про знайдені вразливості, методологію, використану для тестування, серйозність і місце виявлення проблеми (рис.1).

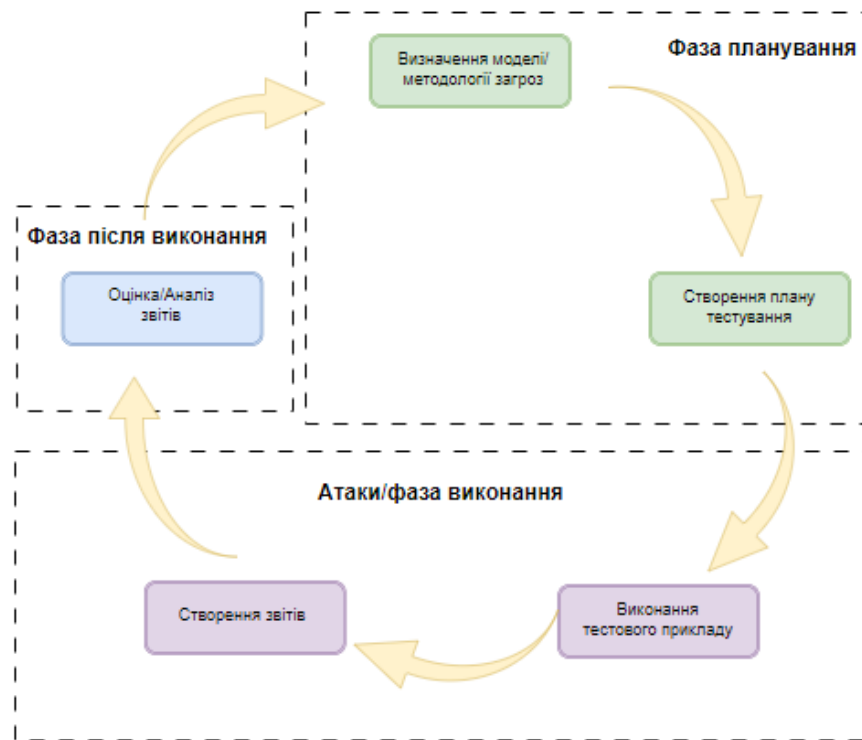


Рисунок 1 – Кроки для тестування проникнення WEB-додатків

Фаза після виконання, на цьому етапі після того, як тестування буде завершено, а звіти про тестування будуть надіслані всім зацікавленим групам, усі повинні опрацювати наступний список:

- запропонуйте виправлення – pentesting не має завершуватися лише виявленням вразливостей (Зацікавлена команда, включаючи члена QA, повинна переглянути висновки, повідомлені тестувальниками, а потім обговорити виправлення);
- повторне тестування вразливостей – після того, як виправлення виконано та реалізовано, тестувальники повинні провести повторне тестування, щоб переконатися, що виправлені вразливості не з'явилися під час повторного тестування;
- очищення – у рамках pentest тестувальники вносять зміни до налаштувань проксі-сервера, тому слід виконати очищення та повернути всі зміни.

1.3 Характеристика найпопулярніші інструменти тестування на проникнення

Оскільки автоматизація забезпечує швидкість, уникає помилок людини вручну, чудове покриття та кілька інших переваг, але, що стосується pentest, він вимагає від проведення певного ручного тестування. Тестування вручну допомагає знайти вразливості, пов'язані з бізнес-логікою, і зменшити помилкові спрацьовування [1].

Інструменти схильні давати багато хибних спрацьовувань, тому потрібне ручне втручання, щоб визначити, чи є вони справжньою вразливістю.

Інструменти тестування веб-додатків на проникнення є важливою частиною стратегії безпеки будь-якої організації. Ці інструменти імітують атаки на веб-додаток, щоб виявити вразливі місця та оцінити ефективність захисту програми. Нижче представлені найпопулярніші інструменти проникнення, які сьогодні використовуються для веб-додатків у галузі (рис.2)

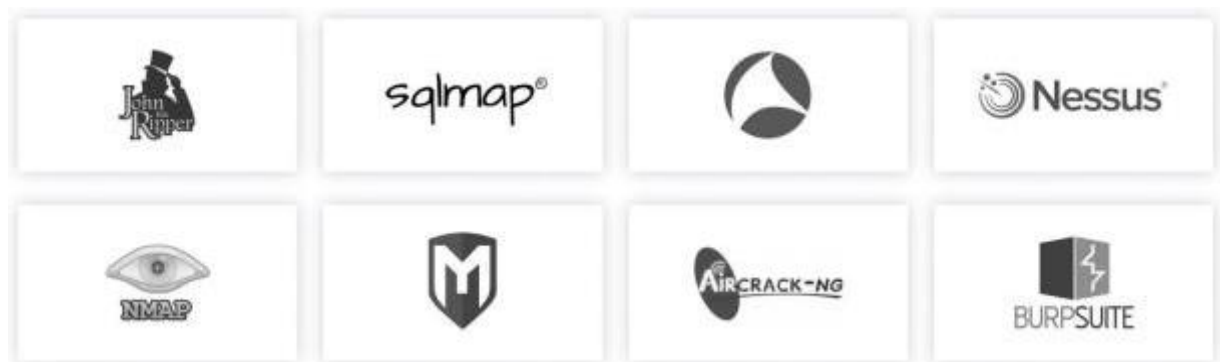


Рисунок 2 – Найпопулярніші інструменти для тестування на проникнення

John the Ripper – популярний інструмент для тестування на проникнення. Його можна використовувати для атак на паролі за допомогою словника, а також для атак методом грубої сили. Він працює, беручи текстовий файл, що містить імена користувачів і паролі, а потім запускає атаку на кожен із

них. Потім він повідомляє тестувальнику, чи знайдено пароль чи ні, і скільки разів він намагався його зламати.

SQLmap – це інструмент для тестування на проникнення, який допомагає виконувати атаки SQL-ін'єкції. Це інструмент на основі командного рядка, який автоматизує процес виявлення та використання недоліків SQL-ін'єкцій і розроблений як швидкий, ефективний і безкоштовний. Його можна використовувати проти будь-якого типу вразливості SQL-ін'єкції, включаючи сліпу ін'єкцію та ін'єкцію на основі помилок. За допомогою sqlmap можна перевіряти, чи є у вебсайтах вразливість.

Якщо вебсайт вразливий до SQL-ін'єкції, можливе:

- отримання інформації з бази даних, у тому числі дамп (всю) базу даних;
- зміна та видалення інформацію з бази даних;
- заливка шелл (бекдор) на вебсервер.

Один із сценаріїв використання sqlmap:

- отримання імені користувача та пароля з бази даних;
- пошук панелей адміністрування сайту (адмінок);
- вхід до адмінки з отриманим логіном та паролем.

SQL-ін'єкція – дуже небезпечна вразливість, яка дає зловмиснику великі можливості. За наявності вразливості атака може розвиватися за різними напрямками такими як:

- модифікація даних;
- заливка бекдору;
- використання JavaScript коду для отримання даних користувачів;
- впровадження коду для підчеплення на BeEF.

Wireshark є одним із найпопулярніших аналізаторів мережевих протоколів на даний момент, який полегшує глибоку перевірку протоколів, а також запис трафіку в реальному часі та автономний аналіз захопленого файлу. Дані можна експортувати за допомогою формату XML, PostScript, CSV або звичайного тексту для документування та подальшого аналізу.

Nessus – це сканер уразливостей допомагає тестувальникам визначати вразливості, проблеми конфігурації та навіть наявність шкідливих програм у вебдодатках. Цей інструмент, однак, не призначений для виконання операцій, але пропонує велику допомогу під час розвідки. Nessus - програма для автоматичного пошуку відомих вад в захисті інформаційних систем. Nessus є одним з багатьох сканерів вразливостей, які використовуються під час оцінок вразливостей і тестування на проникнення, включаючи шкідливі атаки. Вона здатна виявити види вразливостей, які найбільш часто зустрічаються. Наявність вразливих версій служб або доменів Помилки в конфігурації (наприклад, відсутність необхідності авторизації на SMTP-сервері). Наявність паролів за замовчуванням, порожніх, або слабких паролів. Програма має клієнт-серверну архітектуру, що сильно розширює можливості сканування.

Nmap або Network Mapper – це більше, ніж інструмент сканування та розвідки. Він використовується як для виявлення мережі, так і для перевірки безпеки. Окрім надання основної інформації про цільовий вебсайт, він також містить модуль сценаріїв, який можна використовувати для виявлення вразливостей і бекдорів, а також для виконання дій.

Metasploit виділяється серед інших інструментів тестування на проникнення для вебдодатків. Причина в тому, що це насправді фреймворк, а не конкретна програма. Його можна використовувати для створення спеціальних інструментів для конкретних завдань. Тестувальник може використовувати Metasploit для:

- вибору і налаштувань цільового експлойту;
- вибору і налаштувань корисного навантаження, яке буде використовуватися;
- вибору і налаштувань схеми кодування;
- виконання експлойту.

Зловмисники постійно розробляють нові експлойти та методи атак – програмне забезпечення для тестування на проникнення Metasploit допомагає використовувати проти них їх власну зброю. Використовуючи постійно зрос-

таючу базу даних експлойтів, можна безпечно імітувати реальні атаки на мережу, щоб навчити команду безпеки виявляти та зупиняти справжні атаки.

Aircrack-ng – це інструмент бездротової локальної мережі, який можна використовувати для відновлення ключів WEP/WPA/WPA2. Це один із найпопулярніших інструментів бездротового злому, який існує з 2002 року. Він використовується тестувальниками проникнення для перевірки безпеки бездротових мереж і пошуку слабких місць, але він також має кілька інших варіантів використання, зокрема:

- виявлення мереж, які не захищені належним чином;
- злом відкритих точок доступу Wi-Fi за допомогою слабких паролів або взагалі без шифрування;
- розшифровка трафіку в зашифрованих мережах Wi-Fi.

Aircrack-ng фокусується на різних сферах безпеки WiFi:

- моніторинг: захоплення пакетів і експорт даних у текстові файли для подальшої обробки інструментами сторонніх розробників;
- атака: повторні атаки, деавтентифікація, фальшиві точки доступу та інші за допомогою введення пакетів;
- тестування: перевірка можливостей карт WiFi та драйвера (захоплення та впровадження);
- злом: WEP і WPA PSK (WPA 1 і 2).

Усі інструменти є командним рядком, що дозволяє виконувати важкі сценарії. Багато графічних інтерфейсів скористалися цією функцією. Він працює переважно на Linux, але також на Windows, macOS, FreeBSD, OpenBSD, NetBSD, а також на Solaris і навіть eComStation 2.

Burp Suite, інструмент є універсальною платформою для тестування безпеки веб-додатків. У ньому є кілька інструментів, які можна використовувати на кожному етапі процесу тестування, включно з проксі-сервером перехоплення, програмним павуком, вдосконаленим сканером веб-додатків, інструментом порушника, інструментом повторювача та інструментом секвенсора. Burp дає повний контроль, дозволяє комбінувати просунуті ручні техні-

ки з доведеним до мистецтва автоматизмом, це робить роботу швидше, ефективнішою та приємнішою.

Burp Suite містить такі ключові компоненти:

- перехоплюючий проксі, який дозволяє інспектувати та модифікувати трафік між браузером та цільовим додатком;
- павук для програм для обходу контенту та функціональності;
- просунутий сканер веб-застосунків для автоматизованого виявлення ряду типів уразливостей;
- інструмент Intruder для виконання потужних атак користувача для пошуку і експлуатації незвичайних вразливостей;
- інструмент Repeater для маніпуляцій та повторного надсилання індивідуальних запитів;
- інструмент Sequencer для тестування хаотичних сесійних токенів (маркерів);
- можливість зберігати роботу та відновлювати робочий процес пізніше;
- розширюваність, що дозволяє легко писати власні плагіни, для виконання комплексних і високо настроюваних завдань усередині Burp.

2 АНАЛІЗ ТИПІВ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

Pentest (див. рис.3), як випливає з назви, – це тест, який зосереджується виключно на веб-додатку, а не на всій мережі чи компанії. Тестування на проникнення для веб-додатків здійснюється шляхом ініціювання імітованих атак, як внутрішніх, так і зовнішніх, щоб отримати доступ до конфіденційних даних. Перевірка дозволяє визначити будь-які недоліки безпеки всієї веб-програми та її компонентів, включаючи вихідний код, базу даних і серверну мережу). Це допомагає розробнику визначати пріоритети вразливостей і загроз веб-додатків, а також розробляти стратегії їх усунення.

Програмне забезпечення для електронної комерції, онлайн-банкінгу, охорони здоров'я, планування ресурсів підприємства (ERP), систем керування вмістом (CMS), виставлення рахунків, бухгалтерського обліку та нарахування заробітної плати зазвичай постачається у формі веб-програми. Оскільки ці веб-програми зберігають і передають конфіденційні дані, надзвичайно важливо забезпечити безпеку цих програм протягом усього життєвого циклу розробки програмного забезпечення, особливо тих, які є відкритими для всесвітньої мережі [2]. Тестування веб-проникнення, у свою чергу, важливо з наступних причин:

- визначення невідомих вразливостей;
- перевірка ефективності існуючих політик безпеки вебсайтів і мобільних додатків;
- тестування загальнодоступних компонентів, включаючи брандмауери, маршрутизатори та DNS;
- визначення найбільш вразливий шляхів для атаки;
- знаходження лазівок, які можуть призвести до крадіжки даних.



Рисунок 3 – Схема підходів до проведення тесту на проникнення

2.1 Опис автоматизованого та ручного пентестування

Автоматизоване та ручне pentest – це два різні підходи до проведення тесту на проникнення. Автоматизоване pentest передбачає використання спеціалізованих програмних інструментів для сканування системи на наявність вразливостей і здійснення атак. Цей підхід є швидким і ефективним, і він може покрити велику кількість вразливостей за короткий проміжок часу. Однак він також може давати хибні спрацьовування (тобто повідомляти про вразливості, яких насправді не існує) і може бути не в змозі ідентифікувати всі вразливості, особливо ті, для виявлення яких потрібен людський дотик.

З іншого боку, ручне pentest включає в себе кваліфікованого спеціаліста з безпеки, який вручну перевіряє систему на вразливості та використовує їх. Цей підхід повільніший і потребує більше людських зусиль, але він може бути більш ретельним і точним. Ручне pentest може виявити вразливі місця,

які автоматизовані інструменти можуть пропустити, і це дозволяє тестувальнику мислити творчо та адаптуватися до несподіваних ситуацій.

Хоча обидва підходи мають плюси та мінуси, їх можна успішно використовувати разом для створення більш ретельного тесту. Фактично, деякі компанії вважають, що поєднання двох підходів дає їм найкращі результати, об'єднуючи сильні сторони кожного методу.

2.2 Опис цілей при pentest

Залежно від цілей pentest тестувальникам надається різний рівень інформації про цільову систему або доступ до неї (рис.4). У деяких випадках команда, що перевіряє за допомогою pentest, на початку приймає один підхід і дотримується його. В інших випадках команда тестування розвиває власну стратегію, оскільки її обізнаність про систему зростає під час pentest. Є три рівні доступу до pentest:

- opaque box (непрозора коробка);
- semi-opaque box (напівпрозора коробка);
- transparent box (прозора коробка).

При рівні “непрозора коробка”, команда нічого не знає про внутрішню структуру цільової системи. Вона діє так само, як і хакери, шукаючи будь-які зовнішні слабкі місця. При рівні “напівпрозора коробка”, команда має певні знання про один або кілька наборів облікових даних. Вона також знає про внутрішні структури даних цілі, код і алгоритми. Тестери pentest можуть створювати тестові випадки на основі детальних проєктних документів, таких як архітектурні схеми цільової системи. При рівні “непрозора коробка”, тестери pentest мають доступ до систем і системних артефактів, включаючи вихідний код, двійкові файли, контейнери, а іноді навіть сервери, на яких запущена система. Цей підхід забезпечує найвищий рівень впевненості за найменший проміжок часу [3].



Рисунок 4 – Схема цілей та проведення аналізу при pentest

2.3 Опис видів pen testing

Для оптимального управління ризиками важливий комплексний підхід до pentest. Це передбачає тестування всіх областей у середовищі, що тестується.

У разі, якщо середовище яке тестується це веб-програми. Тестери перевіряють ефективність засобів контролю безпеки та шукають приховані вразливості, шаблони атак та будь-які інші потенційні прогалини в безпеці, які можуть призвести до компрометації веб-програми.

Якщо мова йде про мобільні програми. Використовується як автоматичне, так і розширене ручне тестування, тестувальники шукають уразливості у двійкових файлах додатків, що працюють на мобільному пристрої, і відпові-

дних функціях на стороні сервера. Уразливості на стороні сервера включають керування сеансами, криптографічні проблеми, проблеми з автентифікацією та авторизацією та інші поширені вразливості веб-служб.

У разі, якщо середовище яке тестується це мережа. Це тестування визначає загальні для критичних вразливостей у зовнішній мережі та системах. Експерти використовують контрольний список, який включає тестові випадки для зашифрованих транспортних протоколів, проблеми з визначенням обсягу сертифіката SSL, використання адміністративних служб тощо.

Якщо мова йде про хмарне середовище, воно значно відрізняється від традиційних локальних середовищ. Як правило, відповідальність за безпеку розподіляється між організацією, яка використовує середовище, і постачальником хмарних послуг. Через це для тестування хмарного pentest потрібен набір спеціальних навичок і досвіду для ретельного вивчення різних аспектів хмари, таких як конфігурації, API, різні бази даних, шифрування, зберігання та елементи керування безпекою.

У разі, якщо середовище яке тестується це контейнери. Вони, отримані від Docker, часто мають уразливості, які можна використати в масштабі. Неправильна конфігурація також є поширеним ризиком, пов'язаним з контейнерами та їх середовищем. Обидва ці ризики можна виявити за допомогою експертного тестування pentest.

Якщо мова йде про вбудовані пристрої (IoT), то вбудовані пристрої / пристрої Інтернету речей (IoT), такі як медичні пристрої, автомобілі, побутова техніка, обладнання для нафтових вишок і годинники, мають унікальні вимоги до тестування програмного забезпечення через їх довший життєвий цикл, віддалені місця, обмеження живлення, нормативні вимоги тощо. Експерти проводять ретельний аналіз зв'язку разом із аналізом клієнта/сервера, щоб виявити дефекти, які мають найбільше значення для відповідного сценарію використання.

У разі, якщо середовище яке тестується це мобільні пристрої. Тестувальники pentest використовують як автоматичний, так і ручний аналіз, щоб

знайти вразливості у двійкових файлах додатків, що працюють на мобільному пристрої, і відповідних функціях на стороні сервера. Уразливості у двійкових файлах програми можуть включати проблеми автентифікації та авторизації, проблеми довіри на стороні клієнта, неправильно налаштовані елементи керування безпекою та проблеми міжплатформної інфраструктури розробки. Уразливості на стороні сервера можуть включати в себе керування сеансами, криптографічні проблеми, проблеми автентифікації та авторизації та інші поширені вразливості веб-служб.

Якщо мова йде про API. Для охоплення списку Топ-10 безпеки OWASP API використовуються як автоматичні, так і ручні методи тестування. Деякі з ризиків безпеки та вразливостей, на які шукають тестувальники, включають порушену авторизацію на рівні об'єкта, автентифікацію користувача, надмірний доступ до даних, брак ресурсів/обмеження швидкості тощо.

У разі, якщо середовище яке тестується це конвеєр CI/CD. Сучасні практики DevSecOps інтегрують автоматизовані та інтелектуальні інструменти сканування коду в конвеєр CI/CD. На додаток до статичних інструментів, які знаходять відомі вразливості, автоматизовані інструменти pentest можна інтегрувати в конвеєр CI/CD, щоб імітувати те, що хакер може зробити, щоб поставити під загрозу безпеку програми. Автоматизований pentest CI/CD може виявити приховані вразливості та шаблони атак, які залишаються непоміченими за допомогою статичного сканування коду.

Універсального інструменту для перевірки pentest не існує. Натомість різні цілі вимагають різних наборів інструментів для сканування портів, сканування програм, зламів Wi-Fi або прямого проникнення в мережу. Загалом, типи інструментів для тестування пера поділяються на п'ять категорій:

- інструменти розвідки для виявлення мережеских хостів і відкритих портів;
- сканери вразливостей для виявлення проблем у мережеских службах, веб-додатках і API;

- проксі-інструменти, такі як спеціалізовані веб-проксі або загальні проксі-сервери типу “людина посередині”;
- інструменти експлуатації для досягнення системних опор або доступу до активів;
- інструменти після експлуатації для взаємодії з системами, підтримки та розширення доступу та досягнення цілей атаки.

Оскільки частота та серйозність порушень безпеки з кожним роком зростає, організації ніколи не мали більшої потреби в тому, щоб зрозуміти, як вони можуть протистояти атакам. Такі нормативні акти, як PCI DSS і HIPAA, вимагають періодичного тестування пера, щоб відповідати їхнім вимогам. З огляду на цей тиск, нижче наведені деякі переваги та недоліки цього типу техніки виявлення дефектів.

До переваг використання pentest можна віднести:

- знаходить прогалини в передових методах забезпечення безпеки, таких як автоматизовані інструменти, стандарти конфігурації та кодування, аналіз архітектури та інші легші дії з оцінки вразливостей;
- виявляє як відомі, так і невідомі недоліки програмного забезпечення та вразливості безпеки, включно з невеликими, які самі по собі не викликають особливого занепокоєння, але можуть завдати матеріальної шкоди в рамках складної моделі атаки;
- може атакувати будь-яку систему, імітуючи поведінку більшості зловмисних хакерів, імітуючи якнайближче реального ворога.

До недоліків pentest, відноситься:

- є трудомістким і дорогим;
- не забезпечує комплексної запобігання помилкам і недолікам у виробництві.

3 ОПИС ЕТАПІВ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ТА УРАЗЛИВОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ

На даний момент Kali відома в першу чергу своїми приблизно 600 інструментами пентестування з відкритим кодом, що дозволяє пентестерам легко встановлювати повний набір інструментів безпеки.

Kali Linux – це популярний дистрибутив пентестування, який підтримується Offensive Security (OffSec), приватною охоронною компанією, яка працює вже 15 років. Kali містить сканери, сніфери та багато інших інструментів для атаки.

Операційна система (ОС) може забезпечити повний пентестовий сеанс або більш специфічні атаки. Хоча існує багато інших дистрибутивів пентестування, Kali є найкращим, рекомендованим професіоналами. Більшість попередньо встановлених пакетів доступні як окремі пакети, але Kali містить і підтримує високоякісні рішення, призначені для професійного використання. Ідея цієї ОС полягає в тому, щоб мати комплексний набір інструментів, який відносно легко оновлювати, дотримуючись найкращих стандартів у галузі.

Середовище, яке буде використовуватися для тестування на проникнення на уразливості це інформаційна система – офіційний вебсайт Одеського державного екологічного університету (<https://odeku.edu.ua/>).

Для проведення даного тестування, спочатку використовується такий інструмент для пентестування як WPScan. WPScan – це сканер уразливостей WordPress, що працює за принципом “чорної скриньки”, тобто без доступу до вихідного коду. Він може бути використаний для сканування віддалених вебсайтів WordPress у пошуках проблем безпеки [4].

Перша команда яка буде використовуватись для сканування

```
wpscan --url https://odeku.edu.ua/ -e u
```

Ключ -e використовується для перерахування, u – для користувачів.

Результати проведеного сканування представлені в Додатку А. Виходячи з результатів, можна зробити наступні висновки.

```
[+] URL: https://odeku.edu.ua/ [195.138.69.231]
[+] WordPress theme in use: univer
  | Location: https://odeku.edu.ua/wp-content/themes/univer/
  | Readme: https://odeku.edu.ua/wp-content/themes/univer/readme.txt
  | Style URL: https://odeku.edu.ua/wp-content/themes/univer/style.css?
ver=5.5.11
  | Style Name: univer WordPress
  | Description: univer main css WordPress...
  | Author: Arp-solution
```

Тема яка використовується має назву univer. У звіті також представлено IP-адресу, місцезнаходження теми та файлу з стилями. Також присутній перелік користувачів, які мають доступ до Адміністративної панелі (наприклад admin.odeku, admin, admin-odeku, alex, fedorenko, timepro).

Наступна команда яка буде використовуватись для сканування

```
wpscan --url https://odeku.edu.ua/ -e p
```

Ключ -e використовується для перерахування, p – для плагінів.

Результати проведеного сканування представлені в Додатку Б. Виходячи з результатів, можна зробити наступні висновки. Після виконання наведеної вище команди можна побачити деталі плагіна, по-перше назви плагінів:

```
[+] contact-form-7
  | Location: https://odeku.edu.ua/wp-content/plugins/contact-form-7/
  | Last Updated: 2023-04-23T08:44:00.000Z
  | [!] The version is out of date, the latest version is 5.7.6
  | Version: 5.4.1 (100% confidence)
[+] location-weather
  | Location: https://odeku.edu.ua/wp-content/plugins/location-weather/
  | Latest Version: 1.3.6 (up to date)
  | Last Updated: 2023-04-13T10:31:00.000Z
```

```
| Version: 1.3.6 (100% confidence)
[+] popup-maker
| Location: https://odeku.edu.ua/wp-content/plugins/popup-maker/
| Latest Version: 1.18.1 (up to date)
| Last Updated: 2023-03-09T02:54:00.000Z
| Version: 1.18.1 (100% confidence)
[+] shortcodes-ultimate
| Location: https://odeku.edu.ua/wp-content/plugins/shortcodes-ultimate/
| Latest Version: 5.12.11 (up to date)
| Last Updated: 2023-03-29T08:41:00.000Z
| Version: 5.12.11 (80% confidence)

[+] wordpress-seo
| Location: https://odeku.edu.ua/wp-content/plugins/wordpress-seo/
| Last Updated: 2023-04-26T07:56:00.000Z
| [!] The version is out of date, the latest version is 20.6
| Version: 15.4 (100% confidence)
```

З вище описаного результату, такі плагіни як `contact-form-7` та `wordpress-seo` застарілі. Перевіривши вразливість у наведених вище плагінах, було виявлені наступні види уразливостей.

У плагіні `contact-form-7` проблема дозволяла зловмисникам обдурити захисні механізми, які відповідають за чищення імен файлів під час завантаження. В результаті хакери можуть отримати можливість завантажити шкідливий файл із довільним кодом на вразливий сервер, а потім запустити його як скрипт, щоб виконати прихований код всередині.

У плагіні `wordpress-seo`, критична уразливість, якої в тому що дозволяє користувачам з правами автора, редактора або адміністратора виконувати ін'єкцію SQL на сайті WordPress.

Для подальшого сканування, необхідно визначити IP-адресу інформаційної системи (IC) `odeku.edu.ua`, як можна побачити з рис.5 та рис.6 IP-адреси збігаються, тому для подальшого сканування буде використано саме її (195.138.69.231)

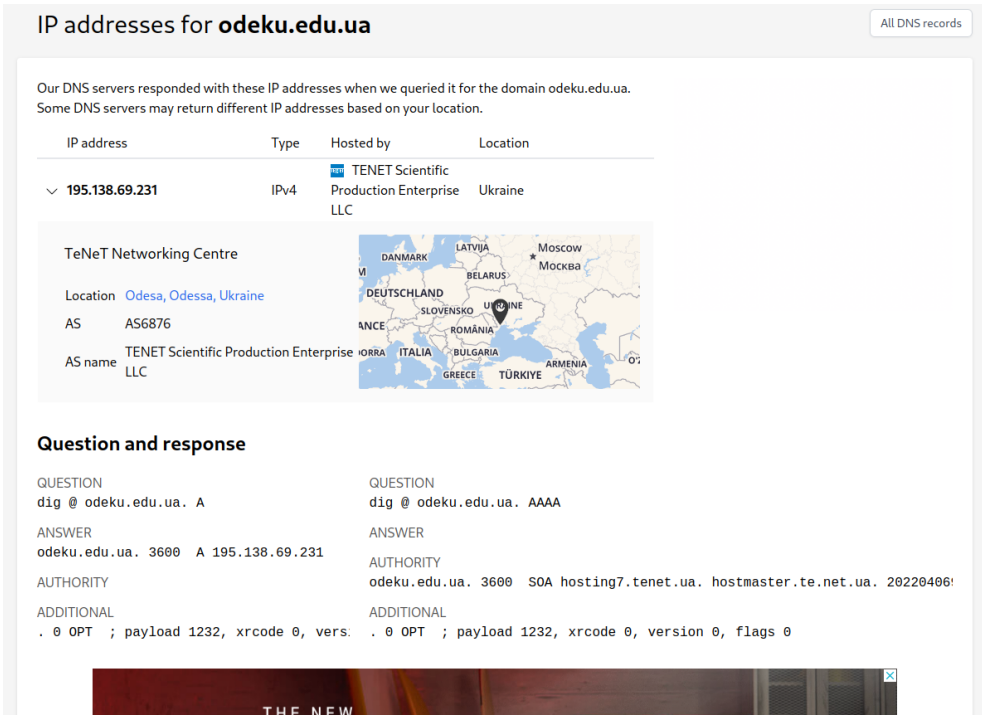


Рисунок 5 – Скріншот з мережевою інформацією з IC nslookup.io

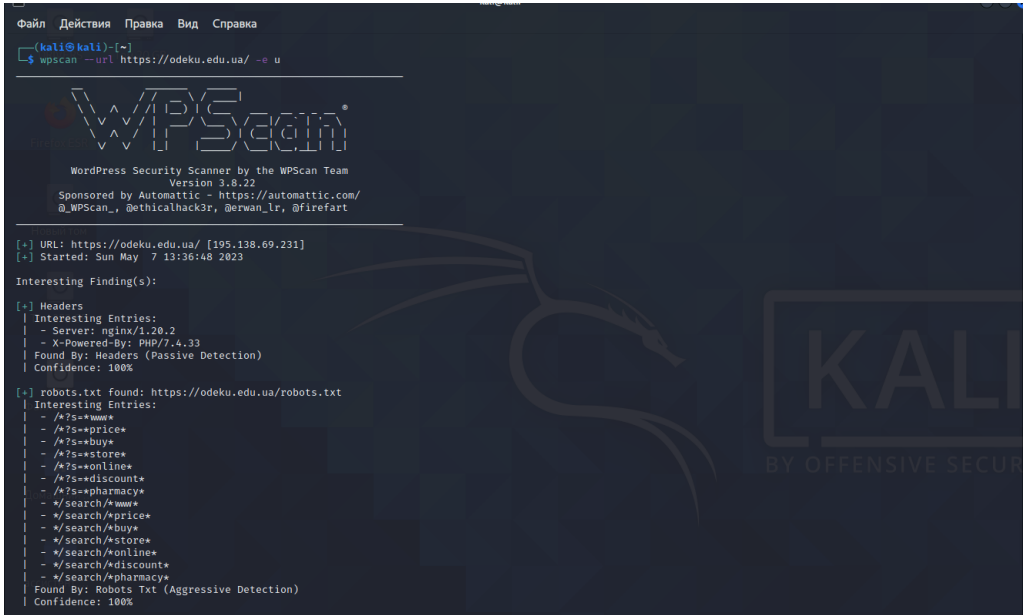


Рисунок 6 – Скріншот з зазначенням IP-адреси IC після сканування плагіном WPScan

Наступний плагін, за допомогою якого буде сканування це плагін Metasploit Framework. Metasploit Framework – це платформа з відкритим ви-

хідним кодом, яка підтримує дослідження вразливостей, розробку експлоїтів і створення спеціальних інструментів безпеки. `msfconsole` – це найбільш часто використовуваний інтерфейс типу оболонки “все в одному”, який дозволяє отримати доступ до всіх функцій Metasploit. Він має підтримку командного рядка, як у Linux, оскільки пропонує автозавершення команд, вкладки та інші ярлики `bash`. Це основний інтерфейс, який дозволить працювати з модулями Metasploit для сканування та запуску атаки на цільову машину. Metasploit має невеликі фрагменти коду, які забезпечують його основні функції. Однак перш ніж перейти до використання модулі, потрібно чітко знати наступні повторювані поняття:

- уразливість – це недолік у конструкції або коді цілі, що робить її вразливою для використання, що призводить до розкриття конфіденційної інформації;
- експлоїт – код, який використовує знайдену вразливість;
- корисне навантаження – це код, який допомагає досягти мети використання вразливості (він працює всередині цільової системи для доступу до цільових даних, наприклад, для підтримки доступу через Meterpreter або зворотну оболонку).

Існує п’ять основних модулів в плагіні Metasploit.

Перший це “auxiliary” (допоміжний), допоміжний модуль містить набір програм, таких як фазери, сканери та інструменти впровадження SQL для збору інформації та глибшого розуміння цільової системи.

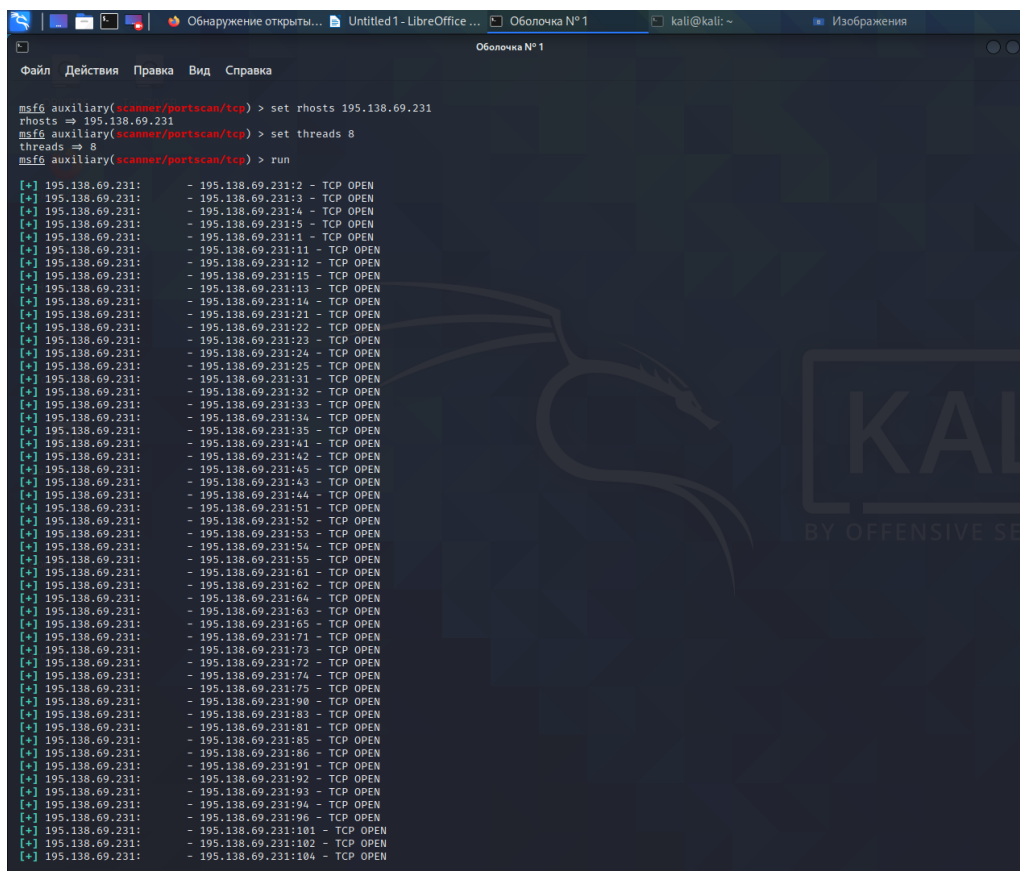
Наступний моділь “encoders” (кодери), кодери шифрують корисні навантаження/експлоїти, щоб захистити їх від антивірусних рішень на основі сигнатур. Оскільки корисні навантаження або експлоїти містять нульові або погані символи, існує висока ймовірність їх виявлення антивірусним рішенням.

Третій модуль “exploit” (експлоїт), експлоїт – це код, який використовує цільові вразливості для забезпечення доступу до системи через корисні навантаження.

Наступний модуль має назву “payload” (корисне навантаження), корисні навантаження допомагають досягти бажаної мети атаки на цільову систему. Це означає, що вони або допоможуть отримати інтерактивну оболонку, або допоможуть підтримувати бекдор, запускати команду чи завантажувати зловмисне програмне забезпечення тощо. Metasploit пропонує два типи корисних навантажень: безступеневі корисні навантаження та поетапні корисні навантаження.

П’ятий модуль “post” (допис) – це модуль після експлуатації допоможе зібрати додаткову інформацію про систему. Наприклад, це може допомогти скинути хеші паролів і шукати облікові дані користувача для бокового переміщення або підвищення привілеїв.

Нижче представлено інформацію, яка була зібрана після сканування за допомогою модуля “auxiliary” та перевірки tcp (рис.7)



```
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 195.138.69.231
rhosts => 195.138.69.231
msf6 auxiliary(scanner/portscan/tcp) > set threads 8
threads => 8
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 195.138.69.231: - 195.138.69.231:2 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:3 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:4 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:5 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:11 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:12 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:15 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:13 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:14 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:21 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:22 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:23 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:24 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:25 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:31 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:32 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:33 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:34 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:35 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:41 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:42 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:45 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:43 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:44 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:51 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:52 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:53 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:54 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:63 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:65 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:61 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:62 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:64 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:74 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:75 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:71 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:73 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:72 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:74 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:75 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:75 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:90 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:83 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:81 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:85 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:86 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:91 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:92 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:93 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:94 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:96 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:101 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:102 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:104 - TCP OPEN
```

Рисунок 7 – Скріншот з результатом сканування плагіном Metasploit

Як можна побачити, з приведеного вище скріншоту на ІС odeku.edu.ua велика кількість відкритих tcp портів. У Додатку В представлено повний звіт щодо вищевказаного сканування. Будь-який порт не більш безпечний або схильний до ризику, ніж будь-який інший порт. Порт є порт, саме від того, для чого використовується порт, і від того, наскільки безпечне використання використовується, залежить, чи є порт безпечним.

Протокол, який використовується для зв'язку через порт, службу або додаток, які споживають або генерують трафік, що проходить через порт, мають бути поточними реалізаціями та перебувати в межах періоду підтримки виробника. Вони повинні отримувати оновлення безпеки та виправлення помилок, і їх слід застосовувати своєчасно. Одні, з самих небезпечних відкритих портів, що використовуються для проведення атак є TCP 22, 80, 135, 139, 443, 445 порти. На даній ІС, відкритими залишаються 135 та 443 порти.

```
[+] 195.138.69.231:      - 195.138.69.231:21 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:22 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:23 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:53 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:135 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:161 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:443 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:4444 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:6661 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:6662 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:6664 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:6663 - TCP OPEN
[+] 195.138.69.231:      - 195.138.69.231:6665 - TCP OPEN
```

TCP є одним із основних протоколів, що використовуються для мережевого зв'язку. TCP – це протокол, орієнтований на з'єднання, який передбачає, що машини на кожній стороні каналу зв'язку розпізнають, що сеанс відкритий, і повідомлення, безсумнівно, отримані на кожній стороні асоціації. UDP – це протокол, який має менше накладних витрат, ніж підключення TCP.

За винятком того, що процедура зв'язку TCP не відрізняється від телефонного дзвінка, де обидва збори гарантують, що зв'язок, безсумнівно, отриманий як надісланий з обох сторін каналу обміну, UDP більше схожий на радіомовлення, де зв'язок передається, а не , з них відправник або бенефіціар за замовчуванням підтверджують, що кореспонденційну посилку було отримано. Програми, які використовують якість зв'язку UDP, менші накладні витрати та вищу швидкість у порівнянні з розширеною надійністю, наприклад, потокове відео та музика.

Порт 21, протокол передачі файлів, небезпечний порт FTP, на якому розміщений сервер FTP, є величезним недоліком безпеки. Багато FTP-серверів мають вразливості, які можуть дозволити анонімну аутентифікацію, бічне переміщення в мережі, доступ до підвищення привілеїв методи і – оскільки багатьма FTP-серверами можна керувати за допомогою сценаріїв – засоби розгортання міжсайтового скриптингу. Шкідливі програми, такі як Dark FTP, Ramen та WinCrash, використовували небезпечні порти та служби FTP.

Порт 22, безпечна оболонка. Облікові записи Secure Shell (SSH), налаштовані з використанням коротких, неунікальних, повторно використовуваних або передбачуваних паролів, небезпечні і можуть бути легко зламані за допомогою словника паролів. Було виявлено безліч уразливостей у минулих реалізаціях служб та демонів SSH, і вони все ще виявляються. Встановлення виправлень є життєво важливим для забезпечення безпеки за допомогою SSH.

Порт 23, Telnet – це застаріла служба, від якої слід відмовитись. Немає жодного виправдання використанню цього стародавнього та небезпечного засобу текстового спілкування. Вся інформація, яку він надсилає та отримує через порт 23, відправляється у вигляді звичайного тексту. Шифрування взагалі відсутнє. Зловмисники можуть перехопити будь-яке з'єднання Telnet та легко вибрати облікові дані для автентифікації. Вони можуть виконувати Атаки посередника шляхом впровадження спеціально створених шкідливих пакетів у немасковані текстові потоки. Віддалений зловмисник, що навіть не пройшов автентифікацію, може скористатися вразливістю переповнення бу-

фера в демоні або службі Telnet і, створюючи шкідливі пакети і вставляючи їх у текстовий потік, виконувати процеси на віддаленому сервері. Це метод, відомий як Видалене (або довільне) виконання коду (RCE).

Порт 4444, протокол керування транспортом. Кілька руткіт, чорний вхід, і троянський кінь програмне забезпечення відкриває та використовує порт 4444. Воно використовує цей порт для перехоплення трафіку та повідомлень, для власних повідомлень та для ексфільтрації даних зі зламаного комп'ютера. Він також використовується для завантаження нових корисних шкідливих даних. Шкідливе ПЗ, таке як Бластерний черв'як та його варіанти використовували порт 4444 для установки бекдорів.

Порт 6660 - 6669, Інтернет-ретранслятор. Інтернет-чат (IRC) почався у 1988 році у Фінляндії, і продовжується досі. У наші дні знадобиться чавунне економічне обґрунтування, щоб дозволити IRC-трафік у організації. За більше ніж 20 років використання IRC було виявлено та використовувалась незліченна кількість вразливостей. У UnrealIRCd У 2009 році у daemon був недолік, що робив віддалене виконання коду тривіальною справою.

Порт 161, протокол обміну невеликими мережевими повідомленнями. Деякі порти та протоколи можуть дати зловмисникам багато інформації про інфраструктуру. Порт 161 UDP привабливий для зловмисників, оскільки його можна використовувати для опитування інформації з серверів як про них самих, так і про обладнання та користувачів, які знаходяться за ними. Порт 161 використовується Простий протокол управління мережею, який дозволяє зловмисникам запитувати таку інформацію, як обладнання інфраструктури, імена користувачів, імена загальних мережеских ресурсів та іншу конфіденційну інформацію, тобто для зловмисника оперативну інформацію.

Порт 53, служба доменних імен. Зловмисникам необхідно враховувати маршрут витоку, який їх шкідливе програмне забезпечення використовуватиме для передачі даних і файлів з організації на власні сервери. Порт 53 використовувався як кращий порт ексфільтрації, оскільки трафік через Служба доменних імен рідко контролюється. Зловмисники можуть легко маскувати

вкрадені дані під трафік DNS та відправляти їх на свій підроблений DNS-сервер. Фальшивий DNS-сервер прийняв трафік та відновив дані у вихідному форматі.

Усього є 65535 портів TCP / IP (і стільки ж Протокол користувальницьких датаграм (UDP) порти).

0-1023 – Відомі порти. Вони розподіляються між службами Управління з присвоєння номерів Інтернету (IANA). Наприклад, за умовчанням SSH використовує порт 22, веб-сервери прослуховують безпечні з'єднання через порт 443 і Простий протокол передачі пошти (SMTP) трафік використовує порт 25.

1024-49151 – Зареєстровані порти. Організації можуть робити запити в IANA для порту, який буде зареєстрований для них і призначений для використання з програмою. Хоча ці зареєстровані порти називають напіврезервованими, їх слід враховувати. зарезервований. Вони називаються частково зарезервованими, тому що можливо, що реєстрація порту більше не потрібна і порт звільняється для повторного використання. Однак, незважаючи на те, що він не зареєстрований, порт все ще знаходиться в списку зареєстрованих портів. Він зберігається у готовності до реєстрації іншою організацією. Прикладом зареєстрованого порту є порт 3389. Це порт, пов'язаний із підключеннями RDP.

49152-65535 – Ефемерні порти. Вони використовують клієнтські програми на спеціальній основі. Можна використовувати їх у будь-якому написаному додатку. Зазвичай вони використовуються як локальний порт усередині комп'ютера, коли він передає дані на добре відомий або зарезервований порт на іншому пристрої, щоб запросити та встановити з'єднання [5].

Сканер портів або засіб перевірки портів – це інструменти, які використовуються для перевірки відкритих портів комп'ютерної системи в мережі. Ці інструменти сканують усю систему TCP/UDP-з'єднання та перевіряють наявність відкритих портів. Інструменти такого типу також шукають запущені служби та пов'язані з ними порти. Тож після завершення ска-

нування портів системи також можна знайти відкриті порти та запущені служби.

Наступа утиліта яка використовується для перевірки ІС, це утиліта Nmap. Nmap (Network Mapper) – це утиліта для дослідження мережі або перевірки безпеки. Вона підтримує сканування ping (визначати, які хости працюють), багато методів сканування портів, визначення версій (визначати протоколи обслуговування та версії додатків, які прослуховують порти), а також відбитки TCP/IP (ідентифікація віддаленої ОС або пристрою). Nmap також пропонує гнучку специфікацію цілей і портів, сканування приманкою/стелс, сканування sunRPC тощо. Більшість платформ Unix і Windows підтримуються як у режимі GUI, так і в режимі командного рядка. Також підтримуються кілька популярних кишенькових пристроїв, зокрема Sharp Zaurus та iPAQ.

Команда nmap -T4 використовується для визначення часу. У процесі сканування nmap надсилає пакети на цільову машину протягом певного періоду часу (інтервалу). Якщо необхідно зменшити або збільшити цей період часу, можна використовувати перемикач nmap -T але для параметра -T потрібен атрибут, тобто необхідно використовувати 1,2,3,4 відповідно до вимог. T4 має більшу швидкість, ніж T1, T2 і T3.

Виконання команди nmap -T4 -v 195.138.69.231 (рис.8)

```

kali@kali: ~
└─$ nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
kali@kali: ~
└─$ nmap -T4 -v 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 18:30 EEST
Initiating Ping Scan at 18:30
Scanning 195.138.69.231 [2 ports]
Completed Ping Scan at 18:30, 0.31s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:30
Completed Parallel DNS resolution of 1 host. at 18:30, 0.93s elapsed
Initiating Connect Scan at 18:30
Scanning srv1.tenet.hosting (195.138.69.231) [1000 ports]
Discovered open port 1025/tcp on 195.138.69.231
Discovered open port 8880/tcp on 195.138.69.231
Discovered open port 445/tcp on 195.138.69.231
Discovered open port 8080/tcp on 195.138.69.231
Discovered open port 1720/tcp on 195.138.69.231
Discovered open port 199/tcp on 195.138.69.231
Discovered open port 53/tcp on 195.138.69.231
Discovered open port 23/tcp on 195.138.69.231
Discovered open port 21/tcp on 195.138.69.231
Discovered open port 3389/tcp on 195.138.69.231
Discovered open port 113/tcp on 195.138.69.231
Discovered open port 443/tcp on 195.138.69.231
Discovered open port 995/tcp on 195.138.69.231
Discovered open port 1723/tcp on 195.138.69.231
Discovered open port 80/tcp on 195.138.69.231
Discovered open port 111/tcp on 195.138.69.231
Discovered open port 587/tcp on 195.138.69.231
Discovered open port 22/tcp on 195.138.69.231
Discovered open port 143/tcp on 195.138.69.231
Discovered open port 3306/tcp on 195.138.69.231
Discovered open port 993/tcp on 195.138.69.231
Discovered open port 5900/tcp on 195.138.69.231
Discovered open port 110/tcp on 195.138.69.231
Discovered open port 554/tcp on 195.138.69.231
Discovered open port 1072/tcp on 195.138.69.231
Discovered open port 139/tcp on 195.138.69.231
Discovered open port 135/tcp on 195.138.69.231
Discovered open port 8086/tcp on 195.138.69.231
Discovered open port 25/tcp on 195.138.69.231
Discovered open port 256/tcp on 195.138.69.231
Discovered open port 1149/tcp on 195.138.69.231
Discovered open port 1839/tcp on 195.138.69.231
Discovered open port 49999/tcp on 195.138.69.231
Discovered open port 49152/tcp on 195.138.69.231
Discovered open port 259/tcp on 195.138.69.231
Discovered open port 6005/tcp on 195.138.69.231
Discovered open port 2004/tcp on 195.138.69.231
Discovered open port 32/tcp on 195.138.69.231
Discovered open port 5101/tcp on 195.138.69.231
Discovered open port 3880/tcp on 195.138.69.231
Discovered open port 749/tcp on 195.138.69.231
Discovered open port 1272/tcp on 195.138.69.231
Discovered open port 1163/tcp on 195.138.69.231
Discovered open port 1036/tcp on 195.138.69.231
Discovered open port 1094/tcp on 195.138.69.231
Discovered open port 83/tcp on 195.138.69.231
Discovered open port 2399/tcp on 195.138.69.231
Discovered open port 89/tcp on 195.138.69.231

```

Рисунок 8 – Скріншот з результатом сканування плагіном Nmap команди `nmap -T4 -v 195.138.69.231`

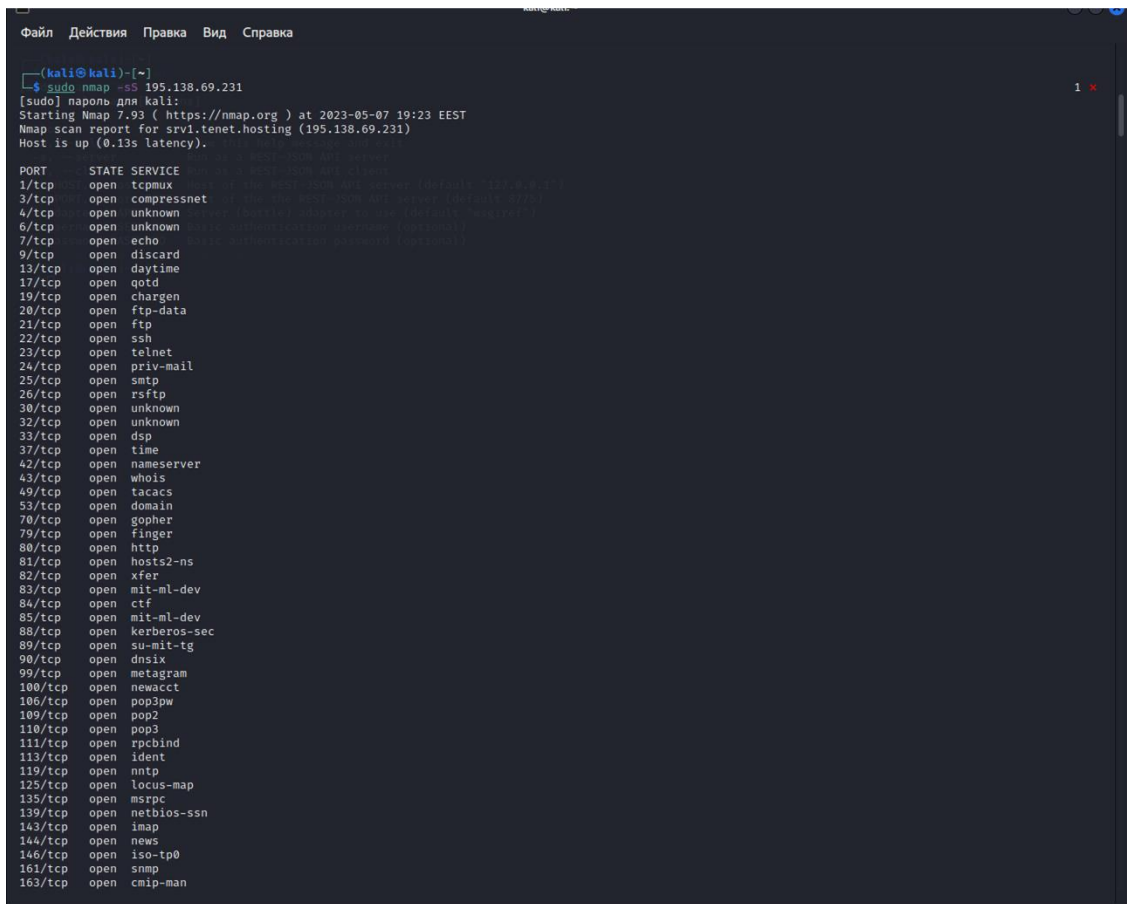
```

kali@kali: ~
└─$ nmap -T4 -v 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 18:42 EEST
Initiating Ping Scan at 18:42
Scanning 195.138.69.231 [2 ports]
Completed Ping Scan at 18:42, 0.25s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:42
Completed Parallel DNS resolution of 1 host. at 18:42, 0.20s elapsed
Initiating Connect Scan at 18:42
Scanning srv1.tenet.hosting (195.138.69.231) [1000 ports]

```

Команда `nmap -sS` для сканування TCP SYN вона вимагає привілейованого доступу та ідентифікує порти TCP. Сканування TCP SYN реалізує загальний метод ідентифікації відкритих портів без виконання тристороннього процесу рукостискання. Коли виявлено відкритий порт, рукостискання TCP

скидається перед завершенням. Таким чином, це сканування також відоме як Напіввідкрите сканування (рис.9).



```

(kali@kali) ~
└─$ sudo nmap -sS 195.138.69.231
[sudo] пароль для kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 19:23 EEST
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.13s latency).

PORT      STATE SERVICE
1/tcp    open  tcpmux
3/tcp    open  compressnet
4/tcp    open  unknown
6/tcp    open  unknown
7/tcp    open  echo
9/tcp    open  discard
13/tcp   open  daytime
17/tcp   open  qotd
19/tcp   open  chargen
20/tcp   open  ftp-data
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
24/tcp   open  priv-mail
25/tcp   open  smtp
26/tcp   open  rsftp
30/tcp   open  unknown
32/tcp   open  unknown
33/tcp   open  dsp
37/tcp   open  time
42/tcp   open  nameserver
43/tcp   open  whois
49/tcp   open  tacacs
53/tcp   open  domain
70/tcp   open  gopher
79/tcp   open  finger
80/tcp   open  http
81/tcp   open  hosts2-ns
82/tcp   open  xfer
83/tcp   open  mit-ml-dev
84/tcp   open  ctf
85/tcp   open  mit-ml-dev
88/tcp   open  kerberos-sec
89/tcp   open  su-mit-tg
90/tcp   open  dnsix
99/tcp   open  metagram
100/tcp  open  newacct
106/tcp  open  pop3pw
109/tcp  open  pop2
110/tcp  open  pop3
111/tcp  open  rpcbind
113/tcp  open  ident
119/tcp  open  nntp
125/tcp  open  locus-map
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
143/tcp  open  imap
144/tcp  open  news
146/tcp  open  iso-tp0
161/tcp  open  snmp
163/tcp  open  cmip-man

```

Рисунок 9 – Скріншот з результатом сканування плагіном Nmap
nmap -sS 195.138.69.231

```

(kali@kali) ~
└─$ sudo nmap -sS 195.138.69.231
1 x
[sudo] пароль для kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 19:23 EEST
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.13s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
53/tcp    open  domain
80/tcp    open  http
135/tcp   open  msrpc

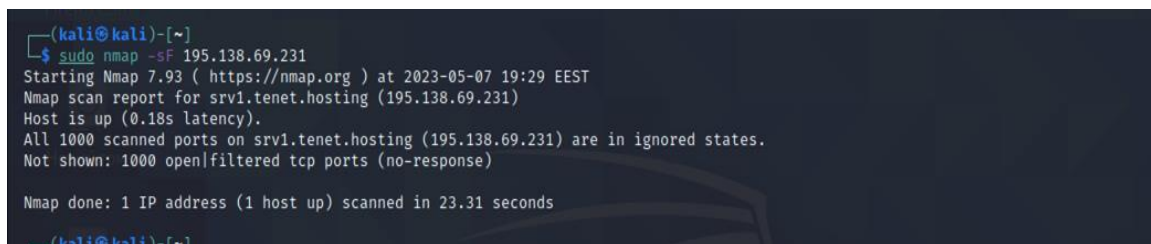
```

```

139/tcp    open  netbios-ssn
161/tcp    open  snmp
443/tcp    open  https
4444/tcp   open  krb524
6666/tcp   open  irc
6667/tcp   open  irc
6668/tcp   open  irc
6669/tcp   open  irc

```

Команда `nmap -sF` для сканування FIN. Сканування FIN надсилає пакет із прапором FIN на цільову машину, тому ці кадри є незвичайними, оскільки вони надсилаються до пункту призначення до завершення процесу тристороннього встановлення зв'язку. Якщо сеанс tcp не активний, це означає, що порт формально закрито. Якщо певний порт закрито на цільовій машині, він повертає RST-пакет у відповідь на FIN-сканування (рис.10).



```

(kali@kali)-[~]
└─$ sudo nmap -sF 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 19:29 EEST
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.18s latency).
All 1000 scanned ports on srv1.tenet.hosting (195.138.69.231) are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 23.31 seconds
(kali@kali)-[~]

```

Рисунок 10 – Скріншот з результатом сканування плагіном Nmap
`sudo nmap -sF 195.138.69.231`

```

-(kali@kali)-[~]
└─$ sudo nmap -sF ██████████
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 19:29 EEST
Nmap scan report for srv1.tenet.hosting (██████████)
Host is up (0.18s latency).
All 1000 scanned ports on srv1.tenet.hosting (██████████)
are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 23.31 seconds

```

Сканування IP-протоколу відрізняється майнером від інших сканувань `nmap`. Він шукає додаткові IP-протоколи, які використовує цільова машина, наприклад ICMP, TCP і UDP. Якщо сканувати маршрутизатор, можуть бути виявлені додаткові IP-протоколи, такі як EGP або IGP.

Команда `nmap -v` для детального режиму (рис.11). Детальний режим Nmap надає функцію отримання додаткових деталей у результатах сканування. Багатослівний режим не змінює те, що відбувається під час сканування, він лише змінює кількість інформації, яку nmap відображає на своєму виході.

```
(kali@kali)-[~]
└─$ sudo nmap -sF -v 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 19:55 EEST
Initiating Ping Scan at 19:55
Scanning 195.138.69.231 [4 ports]
Completed Ping Scan at 19:55, 0.40s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:55
Completed Parallel DNS resolution of 1 host. at 19:55, 0.13s elapsed
Initiating FIN Scan at 19:55
Scanning srv1.tenet.hosting (195.138.69.231) [1000 ports]
Completed FIN Scan at 19:55, 29.30s elapsed (1000 total ports)
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.17s latency).
All 1000 scanned ports on srv1.tenet.hosting (195.138.69.231) are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 29.95 seconds
Raw packets sent: 2020 (80.856KB) | Rcvd: 16 (704B)
```

Рисунок 11 – Скріншот з результатом сканування плагіном Nmap

`sudo nmap -sF -v 195.138.69.231`

```
(kali@kali)-[~]
└─$ sudo nmap -sF -v 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 19:55 EEST
Initiating Ping Scan at 19:55
Scanning 195.138.69.231 [4 ports]
Completed Ping Scan at 19:55, 0.40s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:55
Completed Parallel DNS resolution of 1 host. at 19:55, 0.13s elapsed
Initiating FIN Scan at 19:55
Scanning srv1.tenet.hosting (195.138.69.231) [1000 ports]
Completed FIN Scan at 19:55, 29.30s elapsed (1000 total ports)
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.17s latency).
All 1000 scanned ports on srv1.tenet.hosting (195.138.69.231) are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 29.95 seconds
Raw packets sent: 2020 (80.856KB) | Rcvd: 16 (704B)
```

Команда `nmap -p` для сканування портів (рис.12). Nmap здебільшого використовується для сканування портів, за замовчуванням він сканує всі порти, але можна сканувати один, кілька протоколів або протоколи в межах діапазону.

```
(kali@kali)-[~]
└─$ sudo nmap -p6 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 20:03 EEST
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.15s latency).

PORT      STATE SERVICE
6/tcp    open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.78 seconds

(kali@kali)-[~]
└─$ sudo nmap -p17 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 20:03 EEST
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.26s latency).

PORT      STATE SERVICE
17/tcp   open  qotd

Nmap done: 1 IP address (1 host up) scanned in 0.83 seconds

(kali@kali)-[~]
└─$ sudo nmap -p50 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 20:03 EEST
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.37s latency).

PORT      STATE SERVICE
50/tcp   open  re-mail-ck

Nmap done: 1 IP address (1 host up) scanned in 0.96 seconds

(kali@kali)-[~]
└─$ sudo nmap -p112 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 20:03 EEST
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.20s latency).

PORT      STATE SERVICE
112/tcp  open  mcidas

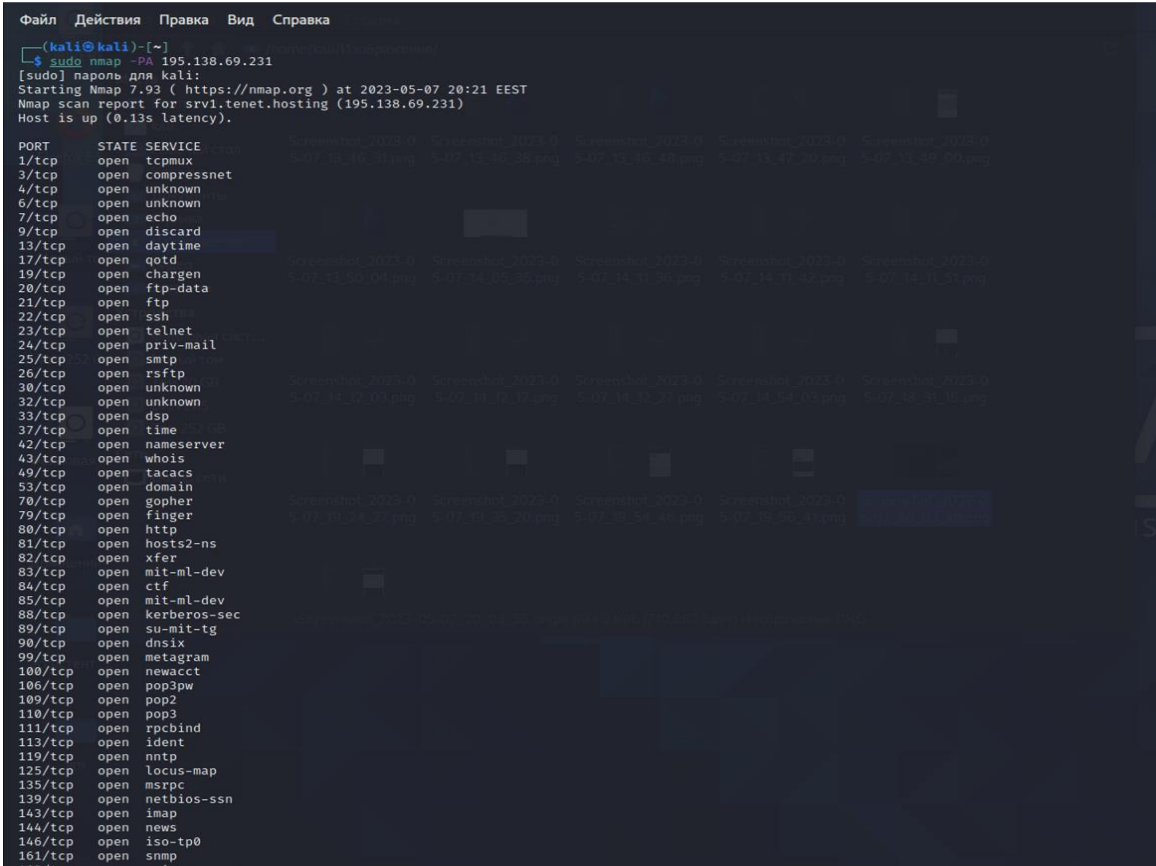
Nmap done: 1 IP address (1 host up) scanned in 1.03 seconds
```

Рисунок 12 – Скріншот з результатом сканування плагіном Nmap
`sudo nmap -p 195.138.69.231`

Команда `nmap -PE` для ICMP Echo Request Ping, Echo-запит ICMP ping надсилає Echo-запит ICMP на IP-адресу цільової машини. У звичайному типі Echo-запиту ICMP надсилається комбінація TCP і АСК-пінгу. За допомогою параметра `-PE` ICMP-ехо-запит можна вказати як метод ping `nmap` без поєднання TCP АСК-пінгу.

```
(kali@kali)-[~]
└─$ sudo nmap -PE 195.138.69.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 20:05 EEST
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.30s latency).
```


Команда `nmap -PA` для TCP ACK Ping (рис.13). Замість використання опції за замовчуванням для Echo-запиту ICMP і TCP ACK параметр `-PA` надсилає TCP ACK і відмовляється від будь-яких Echo-запитів ICMP. Це хороша альтернатива, коли використання ICMP неможливе через фільтрацію пакетів або брандмауери.



```

Файл Действия Правка Вид Справка
(kali@kali)-[~]
└─$ sudo nmap -PA 195.138.69.231
[sudo] пароль для kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 20:21 EEST
Nmap scan report for srv1.tenet.hosting (195.138.69.231)
Host is up (0.13s latency).

PORT      STATE SERVICE
1/tcp    open  tcpmux
3/tcp    open  compressnet
4/tcp    open  unknown
6/tcp    open  unknown
7/tcp    open  echo
9/tcp    open  discard
13/tcp   open  daytime
17/tcp   open  qotd
19/tcp   open  chargen
20/tcp   open  ftp-data
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
24/tcp   open  priv-mail
25/tcp   open  smtp
26/tcp   open  rsftp
30/tcp   open  unknown
32/tcp   open  unknown
33/tcp   open  dsp
37/tcp   open  time
42/tcp   open  nameserver
43/tcp   open  whois
49/tcp   open  tacacs
53/tcp   open  domain
70/tcp   open  gopher
79/tcp   open  finger
80/tcp   open  http
81/tcp   open  hosts2-ns
82/tcp   open  xfer
83/tcp   open  mit-ml-dev
84/tcp   open  ctf
85/tcp   open  mit-ml-dev
88/tcp   open  kerberos-sec
89/tcp   open  su-mit-tg
90/tcp   open  dnsix
99/tcp   open  metagram
100/tcp  open  newacct
106/tcp  open  pop3pw
109/tcp  open  pop2
110/tcp  open  pop3
111/tcp  open  rpcbind
113/tcp  open  ident
119/tcp  open  nntp
125/tcp  open  locus-map
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
143/tcp  open  imap
144/tcp  open  news
146/tcp  open  iso-tp0
161/tcp  open  snmp
163/tcp  open  cmip-map

```

Рисунок 13 – Скріншот з результатом сканування плагіном Nmap

`sudo nmap -PA 195.138.69.231`

Іноді для тестування можна отримати вебсайт або програму на основі CMS для виконання VAPT. Пентестування CMS – має свої складності, оскільки в CMS коди серверної частини здебільшого заздалегідь визначені як природа та поведінка CMS. Будь-хто може завантажити пакет CMS і створити власний вебсайт або блог за лічені секунди, не знаючи жодних знань про кодування та додаткові навички.

Тож, слід зауважити, що під час пентестування CMS доводиться боротися із попередньо визначеними кодами або статичним кодом, ідентифікатор якого розроблений експертами, такими як wordpress, drupal, joomla тощо.

Перш за все, слід відобразити ціль для структурованого перегляду. Буде краще, якщо просканувати ціль за допомогою різних інструментів, наприклад Burp, це буде чудовим варіантом. Окрім цього, можна використовувати «dirb», присутній у kali Linux, який грубо форсує URI та назву каталогу для можливого існування.

Після сканування можна шукати цікаву річ. На сьогоднішній день у CMS перерахування є найважливішою частиною, тому що відповідно до папки CMS за замовчуванням і ім'я сторінки буде однаковим. Але можливо, що розробник також включив або додав певний вид спеціальні коди відповідно до їх потреб. Тож вивчення цих деталей може виявити конфіденційну інформацію. Сканування також важливо, якщо тестуються інші CMS, такі як Modx, Exponent, Wolf CMS тощо. Оскільки стандартні інструменти доступні лише для CMS верхнього рівня, як-от Wordpress, Joomla, Drupal тощо.

Автоматизоване тестування CMS за допомогою різних інструментів і сценаріїв. Це багато доступних інструментів, які можуть допомогти швидко знайти наявні вразливості в CMS. Згідно з найкращою CMS, існують різні інструменти, доступні для WordPress, Drupal, Joomla. Використовувати їх окремо буде головним болем, тому нещодавно було випущено нові інструменти під назвою "CMSMAP", які мають усі три функціональні можливості.

Наразі розглянемо та припустимо, що цільовий домен – <https://195.138.69.231/wordpress/>

У цьому інструменті доступно багато опцій.

```
./cmsmap.py -t https://195.138.69.231 [target]/wordpress
```

Ця команда виконуватиме всі сканування, як-от отримання версії, існуючих плагінів, помилок у списку каталогів тощо.

Також можна використовувати інший інструмент. Після отримання інформації перший підхід має зосередитися на версії CMS та встановленому плагіні. Якщо версія старіша вона вразлива, та присутня певна кількість уразливостей. Іноді через деякі плагіни безпеки обраний сканер не працюватиме та зупинятиметься після виконання, тому потрібно самостійно вказати значення агента користувача за допомогою --user-agent (також шукайте інший варіант).

Типи експлоїтів мають покрокову інформацію, яку можна використати для сканування обраної цілі. Варто пам'ятати, що експлоїт може мати будь-яку версію CMS, тему/модуль/розширення, сторонню програму тощо. Необхідно вивчити всі деталі можливого експлоїту.

Панель адміністратора була б чудовим місцем, щоб знайти уразливості. Кожна CMS має своє розташування за замовчуванням для панелі адміністратора, як-от wordpress cms має site/wp-login.php, як і інші. Якщо не є можливим знайти жодної панелі адміністратора, можливо, розробник зробив якийсь розумний крок проти зловмисників, тож слід також спробувати підібрати місце розташування адміністратора за допомогою «Dirbuster» і Burp.

Щоб почистити панель адміністратора, для грубого форсування можна використовувати різні інструменти, здебільшого надають перевагу CMSMap і Burp Suite.

```
./cmsmap.py -t http://192.168.65.131/wordpress/ -u admin -p /root/wpcrack.txt
```

Ця команда за замовчуванням перейде на сторінку входу за замовчуванням wordpress і запустить підбір відповідно до параметра, -u означаю ім'я користувача/список імен користувачів & -p список паролів.

На рис.14 представлено результати виконання команди dirb <https://195.138.69.231/> /usr/share/wordlists/dirb/common.txt

```

Файл Действия Правка Вид Справка
DOWNLOADED: 0 - FOUND: 0
(kali@kali)-[~]
└─$ dirb https://195.138.69.231/ /usr/share/wordlists/dirb/common.txt

DIRB v2.22
By The Dark Raver

START_TIME: Sun May 7 20:44:46 2023
URL_BASE: https://195.138.69.231/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

GENERATED WORDS: 4612

--- Scanning URL: https://195.138.69.231/ ---
+ https://195.138.69.231/core (CODE:200|SIZE:5832)
+ https://195.138.69.231/manager (CODE:200|SIZE:11085)
=> DIRECTORY: https://195.138.69.231/phpmyadmin/
+ https://195.138.69.231/robots.txt (CODE:200|SIZE:14)

--- Entering directory: https://195.138.69.231/phpmyadmin/ ---
=> DIRECTORY: https://195.138.69.231/phpmyadmin/doc/
+ https://195.138.69.231/phpmyadmin/favicon.ico (CODE:200|SIZE:18902)
+ https://195.138.69.231/phpmyadmin/index.php (CODE:200|SIZE:9672)
=> DIRECTORY: https://195.138.69.231/phpmyadmin/js/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/libraries/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/locale/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/sql/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/test/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/themes/

--- Entering directory: https://195.138.69.231/phpmyadmin/doc/ ---
=> DIRECTORY: https://195.138.69.231/phpmyadmin/doc/html/

--- Entering directory: https://195.138.69.231/phpmyadmin/js/ ---
=> DIRECTORY: https://195.138.69.231/phpmyadmin/js/jquery/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/js/transformations/

--- Entering directory: https://195.138.69.231/phpmyadmin/libraries/ ---
=> DIRECTORY: https://195.138.69.231/phpmyadmin/libraries/config/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/libraries/dbi/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/libraries/engines/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/libraries/navigation/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/libraries/plugins/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/libraries/properties/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/libraries/rte/

--- Entering directory: https://195.138.69.231/phpmyadmin/locale/ ---
=> DIRECTORY: https://195.138.69.231/phpmyadmin/locale/az/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/locale/bg/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/locale/ca/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/locale/cs/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/locale/da/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/locale/de/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/locale/el/
=> DIRECTORY: https://195.138.69.231/phpmyadmin/locale/es/

```

Рисунок 14 – Скріншот з результатом виконання команди dirb
<https://195.138.69.231/> /usr/share/wordlists/dirb/common.txt

ВИСНОВКИ

Сьогодні веб-програми стають все більш поширеними в корпоративних, державних і державних службах внаслідок прогресу веб-технологій і зміни бізнес-середовища. Хоча веб-програми можуть зробити життя простішим і ефективнішим, деякі загрози безпеці можуть становити значні ризики для ІТ-інфраструктури організації, якщо з ними не працювати належним чином. Оскільки зараз атаки спрямовані спеціально на недоліки безпеки в дизайні веб-додатків, традиційні заходи безпеки мережі та технології можуть бути недостатніми для захисту веб-додатків від нових загроз. Разом із розробкою веб-додатків необхідно впроваджувати нові заходи безпеки, як технічні, так і адміністративні. Оскільки кількість веб-додатків продовжує зростати, важливо знати, які загрози існують для веб-додатків. Організація може стати мішенню загрози веб-програми через її вебсайт або програми. У процесі розробки організації повинні вирішити ці проблеми безпеки, запровадивши тестування на проникнення, щоб знайти вразливості в Інтернеті та захистити їх до того, як справжні зловмисники зможуть ними скористатися. Веб-програми зручні, економічно ефективні та додають цінність. Однак більшість систем відкрито для доступу в Інтернет, і дані можуть стати легко доступними для тих, хто бажає трохи дослідити. Більше того, навіть найдосконаліші веб-програми схильні до вразливостей як у дизайні, так і в конфігурації, які хакери можуть знайти та використати. Через це безпека веб-додатків має бути пріоритетом, особливо якщо вони обробляють конфіденційну інформацію.

В ідеалі тестування на проникнення може допомогти створити безпечне програмне забезпечення. Це дорогий метод, тому частоту можна дотримуватися як раз на рік.

Оскільки ручне тестування на проникнення неефективне з точки зору часу, грошей і зусиль, частіше використовують його автоматизований аналог. Веб-сканери використовуються для виконання автоматизованих тестів на проникнення в Інтернет, а тестування за допомогою автоматизованих ін-

струментів займає менше часу, ніж тестування вручну. Одне дослідження показало, що не всі інструменти тестування на проникнення в Інтернет пропонують однакові функції, і що об'єднання інструментів може надати детальну інформацію про вразливості в Інтернеті. Крім того, усі інструменти мають свої переваги та недоліки, тому вибір залежить від потреб організації чи окремої людини. Тим не менш, слід враховувати такі функції, як виявлення вразливостей, докладні звіти та сумісні операційні системи та пристрої під час вибору інструменту. Тестерам на проникнення може бути корисно переглянути представлені результати, щоб прийняти кращі рішення. Крім того, чітке розуміння обмежень і напрямків у цій галузі може бути корисним для майбутніх дослідників.

Усі порти мають бути закриті, якщо немає задокументованого, перевіреного та затвердженого економічного обґрунтування. Слід зробити те саме для відкритих сервісів. Паролі за замовчуванням необхідно змінити та замінити надійними унікальними паролями. По можливості слід використовувати двофакторну автентифікацію.

Усі служби, протоколи, мікропрограми та програми повинні, як і раніше, відповідати життєвому циклу підтримки виробників, і для них мають бути доступні виправлення безпеки та виправлення помилок.

Слід контролювати порти, які використовуються у мережі, і досліджуйте будь-які дива або незрозумілі відкриті порти. Зрозуміти, як виглядає звичайне використання порту, щоб можна було визначити незвичайну поведінку. Виконайте сканування портів та тести на проникнення.

Важливо закрити порт 23 та припинити його використання Telnet. Порти SSH можна захистити за допомогою аутентифікації з відкритим ключем та двофакторної аутентифікації. Також допоможе налаштувати мережу на використання іншого номера порту для трафіку SSH.

Якщо необхідно використовувати IRC, слід перш за все переконатися, що він знаходиться за брандмауером, і вимагайте, щоб користувачі IRC підключилися до мережі через VPN, щоб використовувати його. Не слід дозволя-

ти зовнішньому трафіку прямо потрапляти до IRC. Варто відстежувати та фільтрувати DNS-трафік. З порту 53 не повинно виходити нічого, окрім справжніх DNS-запитів.

Також слід зробити акцент на прийнятті стратегії глибокоешелонованого захисту та зробити захист багаторівневим. Використовувати міжмережеві екрани на основі хоста та мережі. Варто вимкнути проксі, які не налаштовані або які більше не потрібні для роботи.

Деякі рядки SNMP, що повертаються, містять облікові дані за замовчуванням у вигляді звичайного тексту, слід вимкнути це. Також, системному адміністратору врано видалити небажані заголовки відповідей HTTP та HTTPS та відключити банери, які за замовчанням включаються у відповіді від деякого мережного обладнання. Вони даремно видають інформацію, яка приносить користь лише зловмисникам.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Beginners Guide To Web Application Penetration Testing. URL: <https://www.softwaretestinghelp.com/getting-started-with-web-application-penetration-testing/> (дата звернення: 02.04.2023)
2. Your 2023 Guide to Web Application Penetration Testing. URL: <https://relevant.software/blog/penetration-testing-for-web-applications/> (дата звернення: 13.04.2023)
3. What are the benefits of penetration testing? URL: <https://www.synopsys.com/glossary/what-is-penetration-testing.html> (дата звернення: 06.04.2023)
4. WPScan. URL: <https://kali.tools/?p=156> (дата звернення: 02.05.2023)
5. Чому деякі мережеві порти небезпечні та як їх убезпечити? - CloudSavvy IT. URL: <https://cpab.ru/pochemu-nekotorye-setevye-porty-opasny-i-kak-ih-obezopasit-cloudsavvy-it/> (дата звернення: 27.04.2023)

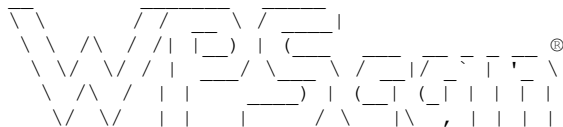
ДОДАТКИ

ДОДАТОК А

Виконання команди `wpscan --url https://odeku.edu.ua/ -e u`

[!] To see full list of options use `--hh`.

```
(kali@kali)-[~]
└─$ wpscan --url https://odeku.edu.ua/ -e u
```



WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - <https://automattic.com/>
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart

```
[+] URL: https://odeku.edu.ua/ [195.138.69.231]
[+] Started: Sun May 7 13:36:48 2023
```

Interesting Finding(s):

[+] Headers

```
| Interesting Entries:
| - Server: nginx/1.20.2
| - X-Powered-By: PHP/7.4.33
| Found By: Headers (Passive Detection)
| Confidence: 100%
```

[+] robots.txt found: <https://odeku.edu.ua/robots.txt>

```
| Interesting Entries:
| - /*?s=*www*
| - /*?s=*price*
| - /*?s=*buy*
| - /*?s=*store*
| - /*?s=*online*
| - /*?s=*discount*
| - /*?s=*pharmacy*
| - */search/*www*
| - */search/*price*
| - */search/*buy*
| - */search/*store*
| - */search/*online*
| - */search/*discount*
| - */search/*pharmacy*
| Found By: Robots Txt (Aggressive Detection)
| Confidence: 100%
```

[+] XML-RPC seems to be enabled: <https://odeku.edu.ua/xmlrpc.php>

```
| Found By: Link Tag (Passive Detection)
| Confidence: 100%
| Confirmed By: Direct Access (Aggressive Detection), 100% confidence
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
```

[+] WordPress readme found: <https://odeku.edu.ua/readme.html>

```
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
```

[+] The external WP-Cron seems to be enabled: <https://odeku.edu.ua/wp-cron.php>

```
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299
```

[+] WordPress version 5.5.11 identified (Outdated, released on 2022-10-17).

```

| Found By: Rss Generator (Passive Detection)
| - https://odeku.edu.ua/feed/, <generator>https://wordpress.org/?v=5.5.11</generator>
| Confirmed By: Meta Generator (Passive Detection)
| - https://odeku.edu.ua/, Match: 'WordPress 5.5.11'

[+] WordPress theme in use: univer
| Location: https://odeku.edu.ua/wp-content/themes/univer/
| Readme: https://odeku.edu.ua/wp-content/themes/univer/readme.txt
| Style URL: https://odeku.edu.ua/wp-content/themes/univer/style.css?ver=5.5.11
| Style Name: univer WordPress
| Description: univer main css WordPress...
| Author: Arp-solution
|
| Found By: Css Style In Homepage (Passive Detection)
| Confirmed By: Css Style In 404 Page (Passive Detection)
|
| Version: 1.0.1 (80% confidence)
| Found By: Style (Passive Detection)
| - https://odeku.edu.ua/wp-content/themes/univer/style.css?ver=5.5.11, Match: 'Version:
1.0.1'

[+] Enumerating Users (via Passive and Aggressive Methods)
Brute          Forcing          Author          IDs          -          Time:          00:00:19
<=====> (10 / 10) 100.00%
Time: 00:00:19

[i] User(s) Identified:

[+] admin.odeku
| Found By: Rss Generator (Passive Detection)
| Confirmed By: Rss Generator (Aggressive Detection)

[+] admin
| Found By: Wp Json Api (Aggressive Detection)
| - https://odeku.edu.ua/wp-json/wp/v2/users/?per_page=100&page=1
| Confirmed By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[+] admin-odeku
| Found By: Wp Json Api (Aggressive Detection)
| - https://odeku.edu.ua/wp-json/wp/v2/users/?per_page=100&page=1
| Confirmed By:
| Yoast Seo Author Sitemap (Aggressive Detection)
| - https://odeku.edu.ua/author-sitemap.xml
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[+] alex
| Found By: Wp Json Api (Aggressive Detection)
| - https://odeku.edu.ua/wp-json/wp/v2/users/?per_page=100&page=1
| Confirmed By:
| Yoast Seo Author Sitemap (Aggressive Detection)
| - https://odeku.edu.ua/author-sitemap.xml
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[+] fedorenko
| Found By: Wp Json Api (Aggressive Detection)
| - https://odeku.edu.ua/wp-json/wp/v2/users/?per_page=100&page=1
| Confirmed By:
| Oembed API - Author URL (Aggressive Detection)
| - https://odeku.edu.ua/wp-json/oembed/1.0/embed?url=https://odeku.edu.ua/&format=json
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[+] timepro
| Found By: Wp Json Api (Aggressive Detection)
| - https://odeku.edu.ua/wp-json/wp/v2/users/?per_page=100&page=1
| Confirmed By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at
https://wpscan.com/register

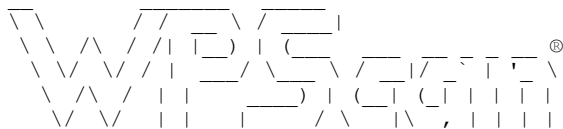
[+] Finished: Sun May 7 13:37:41 2023
[+] Requests Done: 59
[+] Cached Requests: 8
[+] Data Sent: 18.417 KB
[+] Data Received: 1.944 MB
[+] Memory used: 186.918 MB
[+] Elapsed time: 00:00:53

```

ДОДАТОК Б

Виконання команди `wpscan --url https://odeku.edu.ua/ -e p`

```
(kali@kali)-[~]
└─$ wpscan --url https://odeku.edu.ua/ -e p
```



WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - <https://automattic.com/>
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

```
[+] URL: https://odeku.edu.ua/ [195.138.69.231]
[+] Started: Sun May 7 13:38:19 2023
Interesting Finding(s):
[+] Headers
| Interesting Entries:
| - Server: nginx/1.20.2
| - X-Powered-By: PHP/7.4.33
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] robots.txt found: https://odeku.edu.ua/robots.txt
| Interesting Entries:
| - /*?s=*www*
| - /*?s=*price*
| - /*?s=*buy*
| - /*?s=*store*
| - /*?s=*online*
| - /*?s=*discount*
| - /*?s=*pharmacy*
| - */search/*www*
| - */search/*price*
| - */search/*buy*
| - */search/*store*
| - */search/*online*
| - */search/*discount*
| - */search/*pharmacy*
| Found By: Robots Txt (Aggressive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: https://odeku.edu.ua/xmlrpc.php
| Found By: Link Tag (Passive Detection)
| Confidence: 100%
| Confirmed By: Direct Access (Aggressive Detection), 100% confidence
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: https://odeku.edu.ua/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] The external WP-Cron seems to be enabled: https://odeku.edu.ua/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 5.5.11 identified (Outdated, released on 2022-10-17).
| Found By: Rss Generator (Passive Detection)
```

```

| - https://odeku.edu.ua/feed/, <generator>https://wordpress.org/?
v=5.5.11</generator>
| Confirmed By: Meta Generator (Passive Detection)
| - https://odeku.edu.ua/, Match: 'WordPress 5.5.11'

[+] WordPress theme in use: univer
| Location: https://odeku.edu.ua/wp-content/themes/univer/
| Readme: https://odeku.edu.ua/wp-content/themes/univer/readme.txt
| Style URL: https://odeku.edu.ua/wp-content/themes/univer/style.css
?ver=5.5.11
| Style Name: univer WordPress
| Description: univer main css WordPress...
| Author: Arp-solution
|
| Found By: Css Style In Homepage (Passive Detection)
| Confirmed By: Css Style In 404 Page (Passive Detection)
| Version: 1.0.1 (80% confidence)
| Found By: Style (Passive Detection)
| - https://odeku.edu.ua/wp-content/themes/univer/style.css?ver=5.5.11, Match: 'Version: 1.0.1'

[+] Enumerating Most Popular Plugins (via Passive Methods)
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:

[+] contact-form-7
| Location: https://odeku.edu.ua/wp-content/plugins/contact-form-7/
| Last Updated: 2023-04-23T08:44:00.000Z
| [!] The version is out of date, the latest version is 5.7.6
|
| Found By: Urls In Homepage (Passive Detection)
| Confirmed By:
| Urls In 404 Page (Passive Detection)
| Hidden Input (Passive Detection)
|
| Version: 5.4.1 (100% confidence)
| Found By: Query Parameter (Passive Detection)
| - https://odeku.edu.ua/wp-content/plugins/contact-form-7/includes/css/
styles.css?ver=5.4.1
| Confirmed By:
| Hidden Input (Passive Detection)
| - https://odeku.edu.ua/, Match: '5.4.1'
| Readme - Stable Tag (Aggressive Detection)
| - https://odeku.edu.ua/wp-content/plugins/contact-form-7/readme.txt

[+] location-weather
| Location: https://odeku.edu.ua/wp-content/plugins/location-weather/
| Latest Version: 1.3.6 (up to date)
| Last Updated: 2023-04-13T10:31:00.000Z
|
| Found By: Urls In Homepage (Passive Detection)
|
| Version: 1.3.6 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
| - https://odeku.edu.ua/wp-content/plugins/location-weather/readme.txt
| Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
| - https://odeku.edu.ua/wp-content/plugins/location-weather/readme.txt

[+] popup-maker
| Location: https://odeku.edu.ua/wp-content/plugins/popup-maker/
| Latest Version: 1.18.1 (up to date)
| Last Updated: 2023-03-09T02:54:00.000Z
|
| Found By: Urls In Homepage (Passive Detection)
| Confirmed By: Urls In 404 Page (Passive Detection)
|
| Version: 1.18.1 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
| - https://odeku.edu.ua/wp-content/plugins/popup-maker/readme.txt
| Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
| - https://odeku.edu.ua/wp-content/plugins/popup-maker/readme.txt

[+] shortcodes-ultimate
| Location: https://odeku.edu.ua/wp-content/plugins/shortcodes-ultimate/
| Latest Version: 5.12.11 (up to date)
| Last Updated: 2023-03-29T08:41:00.000Z
|
| Found By: Urls In Homepage (Passive Detection)
| Confirmed By: Urls In 404 Page (Passive Detection)
|

```

```
| Version: 5.12.11 (80% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
| - https://odeku.edu.ua/wp-content/plugins/shortcodes-ultimate/readme.txt
```

```
[+] wordpress-seo
| Location: https://odeku.edu.ua/wp-content/plugins/wordpress-seo/
| Last Updated: 2023-04-26T07:56:00.000Z
| [!] The version is out of date, the latest version is 20.6
|
| Found By: Comment (Passive Detection)
|
| Version: 15.4 (100% confidence)
| Found By: Comment (Passive Detection)
|- https://odeku.edu.ua/, Match: 'optimized with the Yoast SEO plugin v15.4-'
| Confirmed By: Readme - Stable Tag (Aggressive Detection)
| - https://odeku.edu.ua/wp-content/plugins/wordpress-seo/readme.txt
```

```
[!] No WPScan API Token given, as a result vulnerability data has not been output.
```

```
[!] You can get a free API token with 25 daily requests by registering at
https://wpscan.com/register
```

```
[+] Finished: Sun May 7 13:38:27 2023
[+] Requests Done: 12
[+] Cached Requests: 36
[+] Data Sent: 3.597 KB
[+] Data Received: 189.17 KB
[+] Memory used: 248.508 MB
[+] Elapsed time: 00:00:07
```

ДОДАТОК В

Виконання Module options (auxiliary/scanner/portscan/tcp)

```
$ sudo msfdb init && msfconsole
[sudo] пароль для kali:
[i] Database already started
[i] The database appears to be already configured, skipping initialization
```

```
Metasploit Park, System Security Interface
Version 4.0.5, Alpha E
Ready...
> access security
access: PERMISSION DENIED.
> access security grid
access: PERMISSION DENIED.
> access main security grid
access: PERMISSION DENIED....and...
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
```

```
      =[ metasploit v6.2.25-dev                ]
+ -- --=[ 2264 exploits - 1189 auxiliary - 404 post   ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops      ]
+ -- --=[ 9 evasion                               ]
```

```
Metasploit tip: Enable verbose logging with set VERBOSE
true
Metasploit Documentation: https://docs.metasploit.com/
```

```
msf6 > search portscan
```

```
Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/portscan/ftpbounce			normal No	FTP Bounce
Port Scanner					
1	auxiliary/scanner/natpmp/natpmp_portscan			normal No	NAT-PMP
External Port Scanner					
2	auxiliary/scanner/sap/sap_router_portscanner			normal No	SAPRouter
Port Scanner					
3	auxiliary/scanner/portscan/xmas		normal	No	TCP "XMas" Port
Scanner					
4	auxiliary/scanner/portscan/ack			normal No	TCP ACK
Firewall Scanner					
5	auxiliary/scanner/portscan/tcp			normal No	TCP Port
Scanner					
6	auxiliary/scanner/portscan/syn		normal	No	TCP SYN Port
Scanner					
7	auxiliary/scanner/http/wordpress_pingback_access			normal No	Wordpress
Pingback Locator					

Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access

```
msf6 > use 5
msf6 auxiliary(scanner/portscan/tcp) > options
```

```
Module options (auxiliary/scanner/portscan/tcp):
```

Name	Current Setting	Required	Description
CONCURRENCY	10	yes	The number of concurrent ports to check per host
DELAY	0	yes	The delay between connections, per thread, in milliseconds
JITTER	0	yes	The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.

PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
THREADS	1	yes	The number of concurrent threads (max one per host)
TIMEOUT	1000	yes	The socket connect timeout in milliseconds

```
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 195.138.69.231
rhosts => 195.138.69.231
msf6 auxiliary(scanner/portscan/tcp) > set threads 8
threads => 8
msf6 auxiliary(scanner/portscan/tcp) > run
```

```
[+] 195.138.69.231: - 195.138.69.231:1 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:2 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:3 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:4 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:5 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:11 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:14 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:12 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:16 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:13 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:22 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:21 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:23 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:24 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:27 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:31 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:32 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:33 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:34 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:35 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:41 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:43 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:42 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:47 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:44 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:51 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:52 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:55 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:53 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:54 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:61 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:63 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:62 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:66 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:65 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:73 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:71 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:72 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:74 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:75 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:81 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:82 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:84 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:83 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:86 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:91 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:92 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:93 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:94 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:95 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:101 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:102 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:103 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:104 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:106 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:111 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:112 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:116 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:113 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:114 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:121 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:122 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:126 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:123 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:127 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:131 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:133 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:132 - TCP OPEN
```



```
[+] 195.138.69.231: - 195.138.69.231:9901 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9902 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9903 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9904 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9906 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9911 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9914 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9916 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9912 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9915 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9922 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9921 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9926 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9925 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9923 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9931 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9933 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9932 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9934 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9935 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9941 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9942 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9943 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9944 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9945 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9951 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9953 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9952 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9954 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9955 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9961 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9962 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9963 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9965 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9964 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9972 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9971 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9974 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9973 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9975 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9981 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9984 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9982 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9983 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9985 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9992 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9991 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9993 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9994 - TCP OPEN
[+] 195.138.69.231: - 195.138.69.231:9995 - TCP OPEN
[*] 195.138.69.231: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > back
```