Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи магістра

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

освітньо-наукова програма *Кібербезпека*
(назва освітньої програми)

на тему: «Система автоматичного сканування мережі»

Виконавець: студентка ІІ курсу, групи КБм-21

(підпис) **Світлана САРОКА** (Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	**Підпис**
Науковий керівник	Сергій БУЧИК	
Нормоконтроль	Юрій ЩЕБЛАНІН	

Київ 2023

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Сергій ТОЛЮПА
«24» жовтня 2022 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

Здобувачки КБм-21 Сароки Світлани Олександрівни
(група) (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Система автоматичного сканування мережі

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 3 від 20.10.2022

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень	Процес розробки системи автоматичного сканування мережі.
Предмет досліджень	Методи виявлення хостів, сканування портів.
Мета	Розробка системи автоматичного сканування мережі з урахуванням можливої протидії засобів мережевого захисту.
Вихідні дані для проведення роботи	Методи скануваня мережі за допомогою модуля Nmap.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна Удосконалення процесу сканування мережі шляхом розробки системи автоматичного сканування з урахуванням можливої протидії засобів мережевого захисту.

Практична цінність Покращення процесу сканування мережі шляхом автоматизації.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Розробка плану для досягнення мети роботи	24.10.2022 – 23.01.2023
Аналіз літературних джерел	24.01.2023 – 14.02.2023
Розробка системи автоматичного сканування мережі	15.02.2023 – 24.04.2023
Оформлення і друк пояснювальної записки	25.04.2023 – 19.05.2023

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект Збереження ресурсів та часу, які б витрачалися на ручне сканування мережі, завдяки автоматизації.

Соціальний ефект Покращення безпеки мереж та захисту від потенційних кібератак шляхом автоматизації.

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ Сергій БУЧИК
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ Світлана САРОКА
(підпис) (прізвище, ініціали)

Дата видачі завдання: 24.10.2022 р.

Термін подання кваліфікаційної роботи до ЕК 19.05.2023 р.

УДК. 004.432.16

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Система автоматичного сканування мережі»: 72 сторінки основного тексту, 14 рисунків, 2 додатки та 7 таблиць. 33 літературних джерела.

Об’єкт дослідження – процес розробки системи автоматичного сканування мережі.

Мета роботи – розробка системи автоматичного сканування мережі з урахуванням можливої протидії засобів мережевого захисту.

Методи дослідження – дослідження літератури, методів виявлення хостів, сканування портів Nmap, порівняння процесів сканування.

У роботі досліджено сучасні методи виявлення хостів і сканування портів за допомогою модуля Nmap. Запропоновано використання автоматизованого сканування і методів, маскування трафіку, рандомізації затримки та хостів. Побудовано систему автоматичного сканування мережі з урахуванням можливої протидії засобів мережевого захисту.

Наукова новизна: удосконалено процес сканування мережі шляхом розробки системи автоматичного сканування з урахування можливої протидії засобів мережевого захисту.

Актуальність теми: системи автоматичного сканування мереж важливі для кібербезпеки. Вони виявляють вразливості та загрози, зменшуючи ризики доступу, атак та витоку даних. З технологічним розвитком та більшістю підключених пристроїв, системи сканування повинні оновлюватись та пристосовуватись до нових загроз, забезпечуючи постійний моніторинг та реагування на вразливості.

Ключові слова: методи виявлення хостів, сканування портів, протокол, пакет, мережа.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ARP - Address Resolution Protocol

API - Application Programming Interface

DNS – Domain Name System

DHCP - Dynamic Host Configuration Protocol

HTTP – Hyper Text Transfer Protocol

IANA - Internet Assigned Numbers Authority

ICMP - Internet Control Message Protocol

IGMP - Internet Group Management Protocol

IPv4 - Internet Protocol version 4

IPv6 - Internet Protocol version 6

IDS -Intrusion Detection System

P2P - peer-to-peer

RTT – Round Trip Time

SMTP - Simple Mail Transfer Protocol

SNMP - Simple Network Management Protocol

TTL - Time to live

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

ОС – Операційна система

ЗМІСТ

ВСТУП

Постійне удосконалення системи сканування мережі є важливою задачею у сучасному інформаційному середовищі. Використання ефективних та надійних засобів сканування мережі дозволяє виявляти потенційні проблеми та забезпечувати безпеку, стабільність і продуктивність мережевої інфраструктури. Кілька проблем, які система сканування мережі може допомогти уникнути:

- виявлення вразливостей: система сканування мережі може допомогти виявити вразливості в системах, пристроях і сервісах, що працюють у мережі. Це дає змогу адміністраторам зробити необхідні заходи з підвищення безпеки, такі як встановлення патчів, налаштування правил брандмауера та зміна стандартних паролів;
- виявлення зловмисних дій: система сканування мережі може виявляти незвичайну активність, невизначені пристрої або сумнівні підключення. Це може вказувати на можливість зламу мережі, розповсюдження шкідливого програмного забезпечення або інші кіберзагрози. Вчасне виявлення таких дій допомагає запобігти подальшим атакам та захистити мережеві ресурси;
- моніторинг трафіку: система сканування мережі може надавати детальну інформацію про трафік, що проходить через мережу. Це дозволяє виявити аномальний або надмірний трафік, що може бути зв'язаним з проблемами мережевої інфраструктури, недостатньою пропускною здатністю або атаками DDoS. В результаті, адміністратори безпеки можуть прийняти заходи для вирішення цих проблем;
- виявлення незахищених точок доступу: система сканування мережі допомагає виявити незахищені або неправильно налаштовані точки доступу до мережі, такі як відкриті Wi-Fi мережі або точки доступу зі слабкими паролями. Це дозволяє адміністраторам прийняти заходи для зміцнення безпеки, встановлення необхідних шифрувань та налаштування правил автентифікації;

- виявлення несанкціонованого доступу: система сканування мережі може виявити спроби несанкціонованого доступу до мережевих ресурсів. Це може включати спроби злому паролів, перехоплення даних або використання несанкціонованих методів доступу. Вчасне виявлення таких спроб дозволяє прийняти заходи для запобігання несанкціонованому доступу та захисту конфіденційної інформації;
- оптимізація мережевої інфраструктури: система сканування мережі надає інформацію про стан мережі, дозволяючи виявити перевантажені сегменти мережі, погану якість зв'язку або недостатню пропускну здатність. Це дозволяє адміністраторам проводити оптимізацію мережевої інфраструктури, виявляти проблеми та робити необхідні зміни для забезпечення ефективної роботи мережі.

Загалом, система сканування мережі є важливим інструментом для виявлення проблем і покращення безпеки, стабільності та продуктивності мережевої інфраструктури.

Об'єкт дослідження: процес розробки системи автоматичного сканування мережі.

Мета магістерської роботи: розробка системи автоматичного сканування мережі з урахуванням можливої протидії засобів мережевого захисту.

Предмет долідження: методи виявлення хостів, сканування портів.

Завдання магістерської роботи:

- проаналізувати методи виявлення хостів;
- проаналізувати методи сканування портів;
- розробити систему автоматичного сканування мережі;
- оцінити ефективність системи.

Наукова новизна: удосконалено процес сканування мережі шляхом розробки системи автоматичного сканування з урахування можливої протидії засобів мережевого захисту.

Актуальність теми. Тема систем автоматичного сканування мережі є досить актуальною і має значний вплив на сферу кібербезпеки. Швидкий розвиток

технологій, збільшення кількості підключених пристроїв і залежність суспільства від мережевих інфраструктур створюють нові виклики та загрози у сфері кібербезпеки.

Системи автоматичного сканування мережі дозволяють виявляти вразливості, слабкі місця та потенційні загрози в мережевій інфраструктурі. Вони допомагають ідентифікувати вразливості в операційних системах, мережевих протоколах, програмному забезпеченні та конфігураціях, що можуть бути використані зловмисниками для несанкціонованого доступу, атак або крадіжки даних.

Зловмисники постійно вдосконалюють свої методи атак і використовують нові техніки, тому системи автоматичного сканування мережі мають бути постійно оновлювані і пристосовані до нових загроз. Ці системи допомагають забезпечити постійний моніторинг мережі, виявлення нових загроз і вчасну реакцію на них.

Крім того, системи автоматичного сканування мережі є необхідним інструментом для виконання аудитів безпеки, дотримання вимог регуляторних органів і забезпечення високого рівня кібербезпеки в організаціях. Вони допомагають зменшити ризики витоку даних, псування інформації, втрати доступу до систем та інших негативних наслідків.

Апробація результатів роботи:

Buchyk S., Saroka S. Analysis of Host Detection Methods. Information Technology and Implementation (Satellite): Conference Proceedings, December 01, 2022, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor). - Kyiv: Publisher Individual entrepreneur Picha Y.V., 2022. pp. 18-20.

Сергій Бучик, Світлана Сарока. Аналіз Методів Сканування Портів Використоввуючи NMAP. Проблеми Кібербезпеки Інформаційно-Телнкомунікаційних Систем (PCSITS). VI Міжнародна Науково-Практична Конференція, 2023. с. 126-127.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ ВИЯВЛЕННЯ ХОСТІВ

1.1 Мережеві технології і протоколи

У розвідці мережі, одним із перших кроків є скорочення набору діапазонів IP-адрес до списку активних або цікавих хостів. Замість сканування кожного порту кожної окремої IP-адреси, що є повільним і не завжди необхідним, використовуються цілі сканування, які роблять хост цікавим для користувача. Наприклад, мережеві адміністратори можуть бути зацікавлені лише в хостах, які запускають певну службу, тоді як аудитори безпеки можуть цікавитися кожним окремим пристроєм з IP-адресою [1].

Використання різних методів сканування може бути зручним для різних сценаріїв. Наприклад, внутрішнім мережевим адміністраторам може вистачати використання лише ping для перевірки доступності хостів у їхній мережі. З іншого боку, зовнішні тестувальники на проникнення можуть використовувати різноманітні зонди і сканери для виявлення уразливостей та уникнення обмежень брандмауера [1].

Одним з інструментів, який часто використовується для сканування мережі і виконання розвідки, є Nmap (Network Mapper). Nmap надає широкі можливості для сканування мережі, включаючи різні методи сканування та налаштування параметрів сканування. Це один з популярних інструментів для виконання сканування мережі та аналізу активних хостів [1].

Nmap використовується мережевими адміністраторами для виявлення вразливостей сервера та оцінки продуктивності систем виявлення вторгнень і брандмауерів. Але в той же час Nmap можна використовувати як інструмент сканування для збору інформації [2].

Оскільки потреби виявлення хостів дуже різноманітні, Nmap має широкий вибір варіантів для налаштування використовуваних методів. Незважаючи на назву ping-сканування, це виходить за рамки простих пакетів exo-запитів ICMP, пов’язаних

із повсюдним інструментом ping. Користувачі можуть повністю пропустити етап ping за допомогою сканування списку (-sL) або вимкнувши ping (-Pn), або задіяти мережу за допомогою довільних комбінацій багатопортових зондів TCP SYN/ACK, UDP і ICMP. Мета зондів — отримати відповіді, які демонструють, що IP-адреса дійсно активна (використовується хостом або мережевим пристроєм). У багатьох мережах лише невеликий відсоток IP-адрес активний у будь-який момент часу. Це особливо часто трапляється з приватним адресним простором, таким як 10.0.0.0/8. Ця мережа має 16,8 мільйонів IP-адрес [1].

Головною метою Nmap є виявлення робочих та відповідаючих хостів у мережі. Це дозволяє зосередитись на реально існуючих хостах, оскільки неможливо взламати або взаємодіяти з хостом, якого не існує. Важливим джерелом інформації про мережеві хости є система доменних імен (DNS). Багато організацій призначають імена, які розкривають функціональність їх систем, надаючи додаткові вказівки. Наприклад, бездротові точки доступу можуть мати назви "wap" або "wireless", брандмауери - "fw", "firewall" або "fw-1", а розробницькі веб-сервери - "dev", "staging", "www-int" або "beta", що вказує на їхнє призначення. Іноді назви можуть відкривати інформацію про місцезнаходження або назви відділів, наприклад, у компанії може бути брандмауер офісу в Чикаго з назвою "fw.chi". Такі інформаційні розкриття можуть бути корисними при проведенні розвідки мережі [3].

За замовчуванням, Nmap використовує DNS reverse для визначення імені хоста, що відповідає кожній IP-адресі, виявленій під час сканування мережі. Це стосується хостів, які успішно відповідають на запити виявлення. Якщо виявлення хоста пропущено за допомогою параметра -Pn, то розпізнавання виконується для всіх IP-адрес, незалежно від їх відповіді. Замість використання повільних стандартних бібліотек розпізнавання DNS, Nmap використовує спеціальний розпізнавач заглушок, який дозволяє виконувати десятки запитів паралельно для ефективного і швидкого визначення імен хостів [3].

Незважаючи на те, що стандартні параметри зазвичай працюють добре, Nmap пропонує чотири варіанти керування дозволами DNS. Вони можуть суттєво вплинути на швидкість сканування та обсяг зібраної інформації [3]:

- -n (немає DNS дозволу):

Вказує Nmap ніколи не виконувати зворотне вирішення DNS для знайдених активних IP-адрес. Оскільки DNS може бути повільним навіть із вбудованим паралельним заглушкою Nmap, цей параметр зменшує час сканування.

- -R (розділення DNS для всіх цілей):

Вказує Nmap завжди виконувати зворотній дозвіл DNS для цільових IP-адрес. Зазвичай зворотній DNS виконується лише для онлайн хостів.

- --system-dns (використовує системний DNS-розпізнавач):

За замовчуванням Nmap розпізнає IP-адреси, надсилаючи запити безпосередньо на сервери імен, налаштовані на хості, а потім прослуховуючи відповіді. Багато запитів виконуються паралельно для підвищення продуктивності. Системний розпізнавач завжди використовується для сканування IPv6.

- --dns-servers <сервер1>[,<сервер2>[,...]] (сервери для використання зворотних DNS-запитів):

За замовчуванням Nmap визначає DNS-сервери (для вирішення rDNS) з файлу resolv.conf (Unix) або реєстру (Win32). Цей параметр може бути використаний, щоб вказати альтернативні сервери. Цей параметр не виконується, якщо використовується --system-dns або сканування IPv6. Використання кількох DNS-серверів часто швидше, особливо якщо обрані авторитетні сервери для цільового IP-простору. Ця опція також може покращити прихованість, оскільки запити можуть бути відхилені майже від будь-якого зворотнього DNS-сервера в Інтернеті [3].

За замовчуванням, Nmap буде включати етап перевірки наявності відповіді (ping) перед більшістю нав'язливих запитів, таких як сканування портів, визначення ОС, Nmap Scripting Engine або визначення версії. Зазвичай Nmap виконує нав’язливе сканування лише на тих машинах, які доступні на етапі сканування ping. Це значно економить час і пропускну здатність порівняно з повним скануванням кожної окремої IP-адреси. Однак цей підхід не ідеальний для всіх обставин. Бувають випадки, коли потрібно сканувати кожну IP-адресу (-Pn), а в інших випадках потрібно виконувати виявлення хоста без сканування портів (-sn). Бувають навіть випадки, коли ви хочете роздрукувати цільові хости та вийти до надсилання тестів ping (-sL) [4].

- Сканування за списком (-sL)

Сканування за списком — це форма виявлення хостів, яка просто перераховує кожен хост у вказаній мережі або мережах без надсилання жодних пакетів до цільових хостів. За замовчуванням Nmap все ще виконує зворотне вирішення DNS на хостах, щоб дізнатися їхні імена. Nmap також повідомляє загальну кількість IP-адрес у кінці. Сканування за списком є хорошою перевіркою працездатності, щоб переконатися, що є належні IP-адреси для цілей [4].

Однією з причин для попереднього сканування за списком є прихованість. У деяких випадках не бажано починати з повномасштабної атаки на цільову мережу, яка може викликати попередження IDS (Intrusion Detection System) і привернути небажану увагу. Сканування за списком є ненавязливим і надає інформацію, яка може бути корисною під час вибору окремих машин для цілі. Можливо, хоча й дуже малоймовірно, що ціль помітить усі запити зворотного DNS [4].

Сканування списку вказується за допомогою параметра командного рядка -sL. Оскільки ідея полягає в тому, щоб просто надрукувати список цільових хостів, параметри функціональності вищого рівня, такі як сканування портів, виявлення ОС або сканування ping, не можна поєднувати з -sL [4].

- Вимкнення сканування портів (-sn)

Даний параметр повідомляє Nmap не запускати сканування портів після виявлення хосту. Якщо використовується самостійно, Nmap виявляє хост, а потім друкує доступні хости, які відповіли на сканування. Це часто називають «ping-скануванням». Незважаючи на те, що сканування портів не виконано, все одно можна запитати сценарії хоста Nmap Scripting Engine (--script) і зондування traceroute (--traceroute). Сканування лише за допомогою ping є на один крок більш нав’язливим, ніж сканування списку, і часто може використовуватися для тих самих цілей. Швидко і без особливої уваги виконує легку розвідку цільової мережі. Знання кількості активних хостів є більш цінним для зловмисників, ніж список із кожними окремими IP-адресою та іменем хоста, який був наданий попереднім методом [4].

Системні адміністратори часто знаходять цей параметр цінним. Його можна легко використовувати для підрахунку доступних машин у мережі або моніторингу

доступності сервера. Це часто називають ping sweep і він є більш надійним, ніж ping широкомовної адреси, оскільки багато хостів не відповідають на широкомовні запити [4].

- Вимкнення ping (-Pn)

Інший варіант — взагалі пропустити етап виявлення. Зазвичай Nmap використовує цей етап для визначення активних машин для інтенсивного сканування. За замовчуванням Nmap виконує лише інтенсивне зондування, наприклад сканування портів, визначення версій або виявлення ОС проти хостів, які виявляються непрацюючими. Якщо вимкнути виявлення хоста за допомогою параметра -Pn, Nmap спробує виконати запитані функції сканування для кожної вказаної цільової IP-адреси. Якщо в командному рядку вказано цільовий адресний простір класу B (/16), скануються всі 65 536 IP-адрес. Належне виявлення хостів пропускається, як і під час сканування списку, але замість того, щоб зупинити та надрукувати цільовий список, Nmap продовжує виконувати запитані функції, наче кожна цільова IP-адреса активна [4].

Є багато причин для вимкнення тестів ping. Однією з найпоширеніших є інтрузивна оцінка вразливості. Можна вказати десятки різних тестів ping, намагаючись отримати відповідь від усіх доступних хостів, але все ще можливо, що активна, але жорстко захищена машина може не відповісти на жоден з цих зондів. Тому, щоб нічого не пропустити, аудитори часто виконують інтенсивне сканування, наприклад, для всіх 65 536 TCP-портів, проти кожної IP-адреси в цільовій мережі. Надсилання сотень тисяч пакетів на IP-адреси, які, ймовірно, не прослуховують, може здатися марнотратним, і це може уповільнити час сканування. Nmap має надсилати повторні передачі на кожен порт у випадку, якщо оригінальний зонд був скинутий під час передачі, і Nmap має витрачати значний час на очікування відповідей, оскільки він не має оцінки часу проходження (round-trip-time - RTT) для цих не відповідаючих IP-адрес. Однак тестувальники на проникнення готові заплатити цю ціну, щоб уникнути навіть незначного ризику втрати активних машин. Вони завжди можуть виконати швидке сканування, залишаючи масштабне сканування -Pn працювати у фоновому режимі, поки вони працюють [4].

Ще одна часта причина використання -Pn полягає в тому, що тестувальник має список машин, про які вже відомо. Тому користувач не бачить сенсу витрачати час на етап виявлення хоста. Користувач створює власний список активних хостів, а потім передає його в Nmap за допомогою параметра -iL (який означає введення даних зі списку). Ця стратегія рідко приносить користь з точки зору економії часу. Через проблеми з повторною передачею та оцінкою RTT, сканування навіть однієї IP-адреси, що не відповідає, у великому списку часто потребує більше часу, ніж для всього етапу сканування ping. Крім того, етап ping дозволяє Nmap збирати зразки RTT, які можуть пришвидшити наступне сканування портів, особливо якщо цільовий хост має суворі правила брандмауера. Хоча вказівка -Pn рідко буває корисною для економії часу, важливо, якщо деякі машини у списку блокують усі методи виявлення, які інакше були б указані [4].

1.2. Дослідження методів виявлення та ідентифікації хостів у комп'ютерних мережах

Був час, коли було легко перевірити, чи зареєстрована IP-адреса на активному хості. Потрібно було надіслати пакт exo-запиту ICMP (ping) і дочекатися відповіді. Брандмауери рідко блокували ці запити, і переважна більшість хостів відповідали на них. Така відповідь вимагається з 1989 року відповідно до RFC 1122, де чітко зазначено, що «Кожен хост повинен реалізувати функцію сервера ICMP Echo, яка отримує exo-запити та надсилає відповідні exo-відповіді» [5][6].

На жаль для мережевих дослідників, багато адміністраторів вирішили, що питання безпеки переважають над вимогами RFC, і заблокували ping-повідомлення ICMP. Але Nmap пропонує широкий спектр методів виявлення хостів, окрім стандартного exo-запиту ICMP [5][6]:

- TCP SYN Ping (-PS<список портів>)

Параметр -PS надсилає порожній TCP-пакет із встановленим прапором SYN. Типовим портом призначення є 80, але альтернативний порт можна вказати як параметр.

Прапор SYN підказує віддаленій системі, що є спроба встановити з’єднання. Зазвичай порт призначення буде закрито, а пакет RST буде надіслано назад. Якщо порт виявляється відкритим, ціль виконує другий крок тристороннього рукостискання TCP, відповідаючи TCP-пакетом SYN/ACK. Потім комп’ютер, на якому запущено Nmap, розриває з’єднання, відповідаючи RST, а не надсилаючи пакет ACK, який завершить тристороннє рукостискання та встановить повне з’єднання.

Для Nmap не має значення, відкритий чи закритий порт. Обговорена раніше відповідь RST або SYN/ACK повідомляє Nmap, що хост доступний і відповідає.

У системах Unix лише привілейований користувач root може надсилати і отримувати необроблені TCP-пакети. Для непривілейованих користувачів автоматично використовується обхідний шлях, за допомогою якого системний виклик підключення ініціюється для кожного цільового порту. Це має наслідком надсилання пакета SYN до цільового хосту для спроби встановити з’єднання. Якщо підключення повертається зі швидким успіхом або помилкою ECONNREFUSED, основний стек TCP має отримати SYN/ACK або RST, а хост позначено як доступний. Якщо спроба з’єднання не виконується, доки не мине час очікування, хост позначається як непрацюючий.

- TCP ACK Ping (-PA<список портів>)

Ping TCP ACK дуже схожий на ping SYN. Різниця полягає в тому, що прапор TCP ACK встановлено замість прапора SYN. Такий пакет ACK претендує на підтвердження даних через встановлене з’єднання TCP, але такого з’єднання не існує. Тому віддалені хости повинні завжди відповідати пакетом RST, розкриваючи своє існування в процесі.

Параметр -PA використовує той самий порт за замовчуванням, що й зонд SYN (80), а також може отримати список портів призначення в тому самому форматі. Якщо непривілейований користувач намагається це зробити або вказано ціль IPv6, використовується обхідний шлях з’єднання. Це обхідне рішення є недосконалим, оскільки підключення фактично надсилає пакет SYN, а не ACK.

Причина пропонування ping як SYN, так і ACK полягає в тому, щоб максимізувати шанси обійти брандмауери. Багато адміністраторів налаштовують

маршрутизатори та інші прості брандмауери, щоб блокувати вхідні пакети SYN, за винятком тих, які призначені для загальнодоступних служб, таких як вебсайт компанії або поштовий сервер. Це запобігає іншим вхідним підключенням до організації, дозволяючи користувачам здійснювати безперешкодні вихідні підключення до Інтернету. Цей підхід без стану займає небагато ресурсів на брандмауері/маршрутизаторі та широко підтримується апаратними та програмними фільтрами. Як лише один приклад поширеності цього методу, програмне забезпечення брандмауера Linux Netfilter/iptables пропонує зручний параметр --syn.

- UDP Ping (-PU<список портів>)

Іншим варіантом виявлення хоста є UDP ping, який надсилає UDP-пакет на вказані порти. Список портів приймає той самий формат, що й для параметрів -PS і -PA, які були розглянуті раніше. Якщо порти не вказано, за замовчуванням буде 40,125. Це за замовчуванням можна налаштувати під час компіляції, змінивши DEFAULT_UDP_PROBE_PORT_SPEC у nmap.h. За замовчуванням використовується вкрай незвичайний порт, оскільки надсилання на відкриті порти часто є небажаним для цього конкретного типу сканування.

Для більшості портів пакет буде порожнім, хоча для кількох звичайних портів, як-от 53 (DNS) і 161 (SNMP), буде надіслано корисне навантаження, що залежить від протоколу, яке, швидше за все, отримує відповідь. Параметр --data-length надсилає довільне корисне навантаження фіксованої довжини для всіх портів.

Після попадання на закритий порт на цільовій машині UDP-зонд повинен отримати у відповідь пакет недоступності порту ICMP. Це означає для Nmap, що машина готова та доступна. Багато інших типів помилок ICMP, таких як хост/мережа недоступні або TTL перевищено, свідчать про несправність або недоступність хоста. Відсутність відповіді також трактується таким чином. Якщо відкривається порт, більшість служб просто ігнорують порожній пакет і не повертають жодної відповіді. Ось чому стандартним портом тестування є 40,125, який навряд чи буде використовуватися. Кілька служб, як-от протокол Character Generator (chargen), відповідатимуть на порожній UDP-пакет і, таким чином, повідомлятимуть Nmap, що

машина доступна. Спеціальні корисні навантаження для портів, які їх мають, підвищують вірогідність того, що зонд отримує відповідь.

Основна перевага цього типу сканування полягає в тому, що воно обходить брандмауери та фільтри, які перевіряють лише TCP.

- Типи ICMP Ping (-PE, -PP і -PM)

На додаток до незвичайних типів виявлення хостів TCP і UDP, Nmap може надсилати стандартні пакети, надіслані повсюдною програмою ping. Nmap надсилає пакет ICMP типу 8 (ехо-запит) на цільові IP-адреси, очікуючи у відповідь тип 0 (ехо-відповідь) від доступних хостів. Як зазначалося вище, багато хостів і брандмауерів тепер блокують ці пакети, замість того, щоб відповідати, як того вимагає RFC 1122. З цієї причини сканування лише за протоколом ICMP рідко буває достатньо надійним проти невідомих цілей в Інтернеті. Але для системних адміністраторів, які здійснюють моніторинг внутрішньої мережі, це може бути практичним і ефективним підходом.

Стандарти ICMP (RFC 792 і RFC 950) також визначають пакети запиту мітки часу, запиту інформації та маски адреси як коди 13, 15 і 17 відповідно. Хоча нібито метою цих запитів є отримання такої інформації, як адресні маски та поточний час, їх можна легко використовувати для виявлення хостів. Nmap наразі не реалізує пакети інформаційних запитів, оскільки вони не підтримуються широко. Запити про позначку часу та маску адреси можна надсилати за допомогою параметрів -PP і -PM відповідно. Відповідь із міткою часу (код ICMP 14) або відповідь за маскою адреси (код 18) повідомляє, що хост доступний. Ці два запити можуть бути корисними, коли адміністратори спеціально блокують пакети ехо-запитів, але забувають, що інші запити ICMP можна використовувати з тією ж метою.

- Ping протоколу IP (-PO<список протоколів>)

Найновішим варіантом виявлення хоста є ping протоколу IP, який надсилає IP-пакети з указаним номером протоколу в їх IP-заголовку. Список протоколів приймає той самий формат, що й списки портів у розглянутих раніше варіантах виявлення хостів TCP і UDP. Якщо протоколи не вказано, за замовчуванням надсилається кілька IP-пакетів для ICMP (протокол 1), IGMP (протокол 2) і IP-in-IP (протокол 4).

Протоколи за замовчуванням можна налаштувати під час компіляції, змінивши DEFAULT_PROTO_PROBE_PORT_SPEC у nmap.h. Потрібно зауважити, що для ICMP, IGMP, TCP (протокол 6) і UDP (протокол 17) пакети надсилаються з відповідними заголовками протоколів, тоді як інші протоколи надсилаються без додаткових даних, окрім заголовка IP (якщо не параметр --data- параметр довжини вказано).

Цей метод виявлення хосту шукає або відповіді, що використовують той самий протокол, що й зонд, або повідомлення про недосяжність протоколу ICMP, які означають, що даний протокол не підтримується хостом призначення. Будь-який тип відповіді означає, що цільовий хост живий.

- Сканування ARP (-PR)

Одним із найпоширеніших сценаріїв використання Nmap є сканування локальної мережі Ethernet. У більшості локальних мереж, особливо тих, що використовують приватні діапазони адрес, надані RFC 1918, переважна більшість IP-адрес не використовуються в будь-який момент часу. Коли Nmap намагається надіслати необроблений IP-пакет, наприклад exo-запит ICMP, операційна система повинна визначити апаратну адресу призначення (ARP), яка відповідає цільовій IP-адресі, щоб вона могла правильно адресувати фрейм Ethernet. Це вимагає, щоб він надіслав серію запитів ARP.

- Параметри за замовчуванням

Якщо жоден із перерахованих методів виявлення хостів не вибрано, Nmap використовує параметр за замовчуванням, який еквівалентний аргументам -PE -PS443 -PA80 -PP для Windows або привілейованих (root) користувачів Unix. Це означає, що на кожну машину надсилаються exo-запит ICMP, пакет TCP SYN, пакет TCP ACK і запит на мітку часу ICMP. Винятком є те, що сканування ARP використовується для будь-яких цілей, які знаходяться в локальній мережі Ethernet. Для непривілейованих користувачів Unix значення за замовчуванням еквівалентне -PS80,443 (виклик підключення TCP до портів 80 і 443 цільових хостів).

Нижче описані позначки параметрів пов’язаних зі скануванням ping [4]:

- -v (--verbose)

У verbose режимі Nmap друкує інформацію про активні хости.

- --source-port <номер порту> (-g)

Встановлення постійного вихідного порту працює для сканування ping (TCP і UDP), як і для інших функцій Nmap. Деякі адміністратори безпеки роблять виключення з набору правил, щоб підтримувати роботу DNS (порт 53) або FTP-DATA (порт 20). Звичайно, це відкриває діру, достатньо велику для проходження ping-сканування Nmap.

- -n, -R

Даний параметр -n вимикає будь-яке розпізнавання DNS, тоді як параметр -R вмикає DNS-запити для всіх хостів, навіть непрацюючих. Поведінка за замовчуванням полягає в тому, щоб обмежити дозвіл DNS активними хостами. Ці параметри особливо важливі для сканування ping, оскільки дозвіл DNS може значно вплинути на час сканування.

- --dns-servers <сервер1>[,<сервер2>[,...]] (Сервери для зворотних DNS-запитів)

За замовчуванням Nmap намагається визначити DNS-сервери (для вирішення rDNS) з файлу resolv.conf (Unix) або реєстру (Win32). Крім того цей параметр можна використовувати, щоб вказати альтернативні сервери. Цей параметр не виконується, якщо використовується --system-dns або сканування IPv6. Використання кількох DNS-серверів часто є швидшим і більш прихованим, ніж запит лише одного.

- --data-length <довжина>

Цей параметр додає <length> випадкові байти даних до кожного пакету та працює з типами сканування TCP, UDP і ICMP (для привілейованих користувачів, які сканують IPv4). Це допомагає зробити сканування менш помітним і більш схожим на пакети, згенеровані повсюдною програмою діагностики ping. Кілька систем виявлення вторгнень (IDS), включаючи Snort, мають сповіщення про нульові ping-пакети. Ця опція дозволяє уникнути сповіщень. Значення опції 32 робить ехо-запит більш схожим на те, що він надійшов із Windows, тоді як 56 імітує стандартний ping Linux.

- --ttl <значення>

Налаштування вихідного TTL підтримується для привілейованих користувачів, які виконують сканування ping IPv4. Це може бути корисним як запобіжний захід, щоб переконатися, що сканування не поширюється за межі локальної мережі. Його також можна використовувати для ще більш переконливої імітації рідної програми ping. Деякі корпоративні мережі страждають від петель маршрутизації, які вони не можуть легко виправити. Зменшення вихідного TTL за допомогою --ttl допомагає зменшити навантаження на центральний процесор маршрутизатора, коли трапляються петлі.

- Стандартні параметри синхронізації (-T3, -T4, -T5 тощо)

Значення -T прискорюють сканування ping так само, як вони прискорюють інші функції Nmap. З помірно швидким і надійним з'єднанням між вихідною та цільовою мережами рекомендований параметр -T4 [6].

- --max-parallelism, --min-parallelism <значення>

Дані параметри впливають на те, скільки зондів може бути відкритим одночасно. З типом ping за замовчуванням (два зонди) значення паралелізму приблизно дорівнює кількості машин, сканованих паралельно.

- --min-rtt-timeout, --max-rtt-timeout, --initial-rtt-timeout <час>

Ці параметри визначають, як довго Nmap чекає на відповідь ping.

- Параметри введення (-iL <назва файлу>, -iR <номер>)

Параметри введення хосту підтримуються, як і в решті Nmap. Користувачі часто поєднують опцію input-from-list (-iL) із -Pn, щоб уникнути сканування ping хостів, про які відомо, що вони вже працюють. Параметр -iR вибирає хости випадковим чином із виділеного IP-простору Інтернету. Він приймає як аргумент кількість випадкових хостів, які можна просканувати.

- Параметри виведення (-oA, -oN, -oG, -oX тощо)

Усі типи виводу Nmap (звичайний, grepable і XML) підтримують сканування ping.

- --randomize-hosts [7]

Зміна порядку сканування хоста за допомогою цього параметра може зробити сканування менш помітним, хоча також може ускладнити відстеження результатів сканування.

- --reason

Звичайний вихід Nmap вказує, чи працює хост чи ні, але не описує, на які тести виявлення хост відповів. Для цієї деталі можна додати опцію --reason. Результати можуть заплутати виявлення хоста, оскільки Nmap не завжди перевіряє кожен тест. Він зупиняється, як тільки отримує першу відповідь. Таким чином, Nmap може повідомити про ехо-відповідь ICMP від хоста під час запуску, але потім відповідь RST може бути отримана першою під час другого запуску, і Nmap повідомить про це.

- --packet-trace

Якщо потребується більше деталей, ніж надає --reason, можна спробуввати --packet-trace. Ця опція показує кожен пакет, надісланий і отриманий Nmap, включаючи такі деталі, як порядкові номери, значення TTL і позначки TCP.

- -D <приманка1, приманка2,> [8]

Приманки повністю підтримуються для привілейованого сканування ping IPv4, маскуючи справжнього зловмисника.

Даний параметр виконує сканування-приманку, у результаті чого віддаленому хосту здається, що хост(и), які були вказані як приманка, також сканують цільову мережу. Таким чином, IDS можуть повідомити про 5–10 сканувань портів з унікальних IP-адрес, але вони не знатимуть, яка IP їх сканувала, а які були невинними приманками. Хоча це можна подолати за допомогою трасування шляху маршрутизатора, скидання відповідей та інших активних механізмів, загалом це ефективний метод приховування IP-адреси.

- -6

Сканування ping на основі підключення TCP (-PS) підтримує протокол IPv6, включаючи багатопортовий режим.

- -S <вихідна IP-адреса> [8]

У деяких випадках Nmap може не визначити вихідну адресу. У цій ситуації слід використовувати -S з IP-адресою інтерфейсу, через який можна надсилати пакети.

Іншим можливим використанням цього прапора є підробка сканування, щоб змусити цілі думати, що їх сканує хтось інший.

- -e <інтерфейс> [8]

Вказує Nmap, через який інтерфейс надсилати та отримувати пакети.

Висновок до першого розділу

Один із перших кроків у дослідженні мережі - скорочення набору IP-адрес до списку активних або цікавих хостів. Сканування кожного порту кожної окремої IP-адреси є повільним та зазвичай не завжди необхідним. Основна мета сканування полягає в тому, щоб знайти хости, які є цікавими для користувача. Мережевих адміністраторів можуть цікавити лише хости, які запускають певну службу, тоді як аудитори безпеки можуть бути зацікавлені в кожному пристрої з IP-адресою. Для адміністратора може бути зручно використовувати лише ping у своїй внутрішній мережі, тоді як зовнішній тестувальник на проникнення може використовувати широкий набір зондів, щоб уникнути обмежень брандмауера.

Nmap пропонує різні налаштування для виявлення хостів залежно від потреб користувачів. Незважаючи на назву "ping-сканування", це виходить за рамки звичайних ICMP пакетів і може бути налаштоване різними способами. Користувачі можуть пропустити етап ping за допомогою сканування списку (-sL) або вимкнути його повністю (-Pn). Вони можуть також використовувати комбінації зондів TCP SYN/ACK, UDP і ICMP для тестування активності хостів. Основна мета зондів полягає в отриманні відповідей, що свідчать про активність IP-адреси. У багатьох мережах лише невеликий відсоток IP-адрес є активним.

Головна мета Nmap - виявлення працюючих та відповідаючих хостів у мережі. Інформацію про мережеві хости можна отримати з системи доменних імен (DNS).

Багато організацій надають хостам імена, що розкривають їх функціональність. Враховуючи, що стандартні параметри зазвичай є ефективними, у даному розділі буди також розлянуті варіанти для керування правами доступу до DNS.

Також у даному розділі були розглянуті і описані методи виявлення хостів за допомогою Nmap:

- TCP ACK Ping;
- UDP Ping;
- типи ICMP Ping;
- ping протоколу IP;
- сканування ARP.

Якщо не обрано жоден з методів виявлення хостів, Nmap використовує параметр за замовчуванням, який відповідає -PE -PS443 -PA80 -PP для Windows або привілейованих користувачів Unix. Це включає exo-запити ICMP, TCP SYN, TCP ACK та запити на мітку часу ICMP. Для місцевих Ethernet мереж використовується ARP сканування. Для непривілейованих користувачів Unix використовується -PS80,443 (TCP підключення до портів 80 і 443). Далі у дипломній роботі описані позначки параметрів пов’язаних зі скануванням ping.

У данному розділі були розглянуті методи виявлення та ідентифікації хостів у комп'ютерних мережах.

РОЗДІЛ 2
АНАЛІЗ МЕТОДІВ СКАНУВАННЯ ПОРТІВ

2.1 Аналіз портів та їх станів в комп'ютерних мережах

Сканування портів використовується для виявлення відкритих портів на комп’ютері, підключеному до мережі. Сканер портів – це програмне забезпечення, створене для пошуку таких портів [9].

Комп’ютерний порт – це віртуальна точка, яка обробляє вхідні та вихідні дані й має критично важливе значення для безпеки. У великих мережах інформація, яку збирають сканери портів, може допомогти визначити потенційно вразливі місця [10].

Nmap працює з двома протоколами, які використовують порти: TCP і UDP. З’єднання для кожного протоколу ідентифікується чотирма елементами: IP-адресами джерела та призначення та відповідними портами джерела та призначення. Усі ці елементи — це просто числа, розміщені в заголовках кожного пакету, який надсилається між хостами. Протокол — це восьмибітне поле, яке вказує, який тип пакету міститься в розділі IP-даних (корисного навантаження). Наприклад, TCP — це протокол номер шість, а UDP — 17. Адреси IPv4 мають довжину 32 біти, а порти — 16 бітів. Адреси IPv6 мають довжину 128 біт [11].

Оскільки більшість популярних служб зареєстровано на добре відомий номер порту, часто можна здогадатися, які служби представляють відкриті порти. Nmap містить файл nmap-services, який містить добре відому службу для зареєстрованих номерів портів і протоколів, а також загальні порти для троянських бекдорів та інших програм, які не потребують реєстрації в Internet Assigned Numbers Authority (IANA). Nmap друкує цю назву служби для довідки разом із номером порту [10].

Оскільки поле номера порту має ширину 16 біт, значення можуть досягати 65 535. Найменше можливе значення, нуль, недійсне. API сокетів Berkeley, який визначає, як зазвичай пишуться програми для мережевого зв’язку, не дозволяє використовувати нульовий порт як такий. Замість цього він інтерпретує запит на

нульовий порт як символ підстановки, що означає, що програмісту байдуже, який використовується. Потім система вибирає доступний номер порту. Наприклад, програмісти рідко звертають увагу на те, який номер вихідного порту використовується для вихідного з'єднання. Тому вони встановлюють його на нуль і дозволяють операційній системі вибрати один [10].

Хоча нульовий порт недійсний, ніщо не заважає комусь вказати його в полі заголовка. Деякі шкідливі троянські бекдори прослуховують нульовий порт скомпрометованих систем як прихований спосіб запропонувати нелегітимний доступ, не з'являючись під час сканування більшості портів. Щоб боротися з цим, Nmap дозволяє сканувати нульовий порт, якщо він вказано явно [10].

Перший клас дійсних портів, номери від одного до 1023, відомі як зарезервовані порти. Системи Unix вимагають, щоб програми мали спеціальні (root) привілеї, щоб прив'язуватися до цих портів і прослуховувати їх. Ідея полягає в тому, щоб дозволити віддаленим користувачам вірити, що вони підключаються до дійсної служби, запущеної адміністратором, а не якимось злим, не привілейованим користувачем. Якби зареєстрований порт для SSH був 2222 замість 22, зловмисник міг би запустити на цьому порту демон SSH, який збирав паролі від усіх, хто підключався. Оскільки більшість звичайних серверних програм слухають зарезервовані порти, сканування цих портів є найефективнішим [10].

Ефемерний діапазон портів — ще один клас портів. Цей пул портів надається системою для розподілу за потреби. Коли програма вказує нульовий порт (що означає «будь-який порт»), система вибирає порт із цього діапазону. Діапазон залежить від операційної системи та зазвичай налаштовується. Він повинен містити принаймні пару тисяч портів, щоб уникнути їх виснаження, коли відкрито багато одночасних з'єднань. Сканування підключення Nmap може використовувати сотні одночасно, оскільки сканує кожен вказаний порт на кожній цільовій машині. У Linux ви можете переглянути або встановити діапазон за допомогою файлу /proc/sys/net/ipv4/ip_local_port_range [10].

- Відомі порти

Це зарезервовані порти (в діапазоні від 1 до 1023, як зазначено вище), які зареєстровані в IANA для певної служби. Відомими прикладами є порти 22, 25 і 80 для служб SSH, SMTP і HTTP відповідно [11].

- Зареєстровані порти

Ці порти належать до діапазону від 1024 до 49 151 і були зареєстровані в IANA так само, як добре відомі порти. Більшість із них не так часто використовуються, як добре відомі порти. Ключова відмінність полягає в тому, що не привілейовані користувачі можуть підключатися до цих портів і таким чином запускати служби на своєму зареєстрованому порті. Користувачі не можуть робити це на більшості платформ для добре відомих портів, оскільки вони знаходяться в зарезервованому діапазоні портів [11].

- Динамічні та/або приватні порти

IANA резервує номери портів від 49152 до 65535 для динамічного використання [14].

Топ-20 (найчастіше відкритих) портів TCP [10, 14]:

1. Порт 80 (HTTP) — якщо ви навіть не знаєте цю службу, ви читаєте не ту книгу. Це становить понад 14% відкритих портів, які ми виявили.
2. Порт 23 (Telnet) — Telnet продовжує працювати (зокрема як порт адміністрування на таких пристроях, як маршрутизатори та інтелектуальні комутатори), навіть якщо він незахищений (незашифрований).
3. Порт 443 (HTTPS) — вебсервери із шифруванням SSL використовують цей порт за замовчуванням.
4. Порт 21 (FTP) — FTP, як і Telnet, є ще одним незахищеним протоколом, який повинен припинити роботу. Навіть з анонімним FTP (щоб уникнути хвилювань щодо автентифікації), передача даних все ще піддається підробці.
5. Порт 22 (SSH) — Secure Shell, зашифрована заміна Telnet (і, у деяких випадках, FTP).
6. Порт 25 (SMTP) — простий протокол передачі пошти (також небезпечний).

7. Порт 3389 (ms-term-server) — порт адміністрування служб терміналів Microsoft.

8. Порт 110 (POP3) — протокол Post Office версії 3 для отримання електронної пошти (незахищений).

9. Порт 445 (Microsoft-DS) — для зв’язку SMB через IP зі службами MS Windows (наприклад, спільний доступ до файлів/принтерів).

10. Порт 139 (NetBIOS-SSN) — Служба сеансів NetBIOS для зв’язку зі службами MS Windows (наприклад, спільний доступ до файлів/принтерів). Це підтримується на комп’ютерах з Windows, версій яких перевищує 445.

11. Порт 143 (IMAP) — протокол доступу до повідомлень Інтернету версії 2. Незахищений протокол отримання електронної пошти.

12. Порт 53 (DNS) — система доменних імен (DNS), незахищена система для перетворення між іменами хостів/доменами та IP-адресами.

13. Порт 135 (MSRPC) — ще один поширений порт для служб MS Windows.

14. Порт 3306 (MySQL) — для зв’язку з базами даних MySQL.

15. Порт 8080 (HTTP-проксі) — зазвичай використовується для проксі-серверів HTTP або як альтернативний порт для звичайних вебсерверів (наприклад, коли інший сервер уже прослуховує порт 80 або працює непривілейованими користувачами UNIX, які можуть прив’язуватися лише до високих портів).

16. Порт 1723 (PPTP) — протокол тунелювання «точка-точка» (метод реалізації VPN, який часто потрібен для широкосмугового підключення до провайдерів).

17. Порт 111 (RPCBind) — зіставляє номери програм SunRPC із поточними номерами портів TCP або UDP.

18. Порт 995 (POP3S) — POP3 із доданим SSL для безпеки.

19. Порт 993 (IMAPS) — IMAPv2 із доданим SSL для безпеки.

20. Порт 5900 (VNC) — система спільного використання графічного робочого столу (незахищена).

Топ-20 (найчастіше відкритих) портів UDP [14, 15]:

1. Порт 631 (IPP)—протокол друку в Інтернеті.

2. Порт 161 (SNMP) — простий протокол керування мережею.

3. Порт 137 (NETBIOS-NS) — один із багатьох портів UDP для таких служб Windows, як спільний доступ до файлів і принтерів.

4. Порт 123 (NTP)—протокол мережевого часу.

5. Порт 138 (NETBIOS-DGM) — ще одна служба Windows.

6. Порт 1434 (MS-SQL-DS)—Microsoft SQL Server.

7. Порт 445 (Microsoft-DS) — ще один порт служб Windows.

8. Порт 135 (MSRPC) — ще один порт служб Windows.

9. Порт 67 (DHCPS) — сервер протоколу динамічної конфігурації хоста (видає IP-адреси клієнтам, коли вони приєднуються до мережі).

10. Порт 53 (домен) — сервер системи доменних імен (DNS).

11. Порт 139 (NETBIOS-SSN) — ще один порт служб Windows.

12. Порт 500 (ISAKMP) — асоціація безпеки в Інтернеті та протокол керування ключами використовуються для налаштування IPsec VPN.

13. Порт 68 (DHCPC) — порт клієнта DHCP.

14. Порт 520 (маршрут)—протокол інформації про маршрутизацію (RIP).

15. Порт 1900 (UPNP) — протокол Microsoft Simple Service Discovery Protocol, який дозволяє виявляти пристрої Universal plug-and-play.

16. Порт 4500 (nat-t-ike) — для узгодження проходження трансляції мережевих адрес під час ініціювання з’єднань IPsec (під час обміну ключами в Інтернеті).

17. Порт 514 (системний журнал) — стандартний демон журналу UNIX.

18. Порт 49152 (варіюється) — перший із визначених IANA динамічних/приватних портів. Жодні офіційні порти не можуть бути зареєстровані звідси до кінця діапазону портів (65536). Деякі системи використовують цей діапазон для своїх ефемерних портів, тому службам, які прив’язують порт без запиту конкретного номера, часто призначається 49152, якщо вони є першою програмою, яка це робить.

19. Порт 162 (SNMPTrap) — порт перехоплення протоколу простого мережевого керування (агент SNMP зазвичай використовує 161, а менеджер SNMP — 162).

20. Порт 69 (TFTP) — простий протокол передачі файлів.

Сканування портів — це акт віддаленого тестування багатьох портів, щоб визначити, у кому вони стані. Найцікавіший стан зазвичай відкритий, тобто програма прослуховує та приймає з'єднання через порт. Для проведення такого сканування доступно багато методів [9].

У той час як багато сканерів портів традиційно об'єднують усі порти у відкриті або закриті стани, Nmap ділить порти на шість штатів. Ці стани не є внутрішніми властивостями самого порту, але описують, як їх бачить Nmap [16].

Шість станів портів, розпізнаних Nmap [16]:

- open

Програма активно приймає з'єднання TCP або UDP-пакети на цьому порту. Їх пошук часто є основною метою сканування портів. Люди, які піклуються про безпеку, знають, що кожен відкритий порт – це шлях для нападу. Зловмисники та тестувальники хочуть використовувати відкриті порти, тоді як адміністратори намагаються закрити або захистити їх за допомогою брандмауерів, не перешкоджаючи законним користувачам. Відкриті порти також цікаві для сканування, не пов'язаного з безпекою, оскільки вони показують служби, доступні для використання в мережі. Перш ніж надто захоплюватися відкритим портом, зауважте, що можливо, програма захищена оболонкою TCP (tcpd) або сама програма налаштована на обслуговування лише затверджених IP-адрес клієнта. Такі випадки залишають більше поверхні для атаки, ніж закритий порт.

- closed

Закритий порт доступний (він отримує та відповідає на пакети зонда Nmap), але немає жодної програми, яка його слухає. Вони можуть бути корисними для демонстрації того, що хост перебуває в мережі та використовує IP-адресу (виявлення хосту або сканування ping), а також як частину виявлення ОС. Оскільки закриті порти доступні, можливо, їх варто просканувати пізніше, якщо деякі відкриються.

Адміністратори можуть захотіти заблокувати такі порти брандмауером, щоб вони відображалися у відфільтрованому стані, про що йдеться далі.

- filtered

Nmap не може визначити, чи відкритий порт, оскільки фільтрація пакетів не дозволяє його зондам досягати порту. Фільтрування може відбуватися за допомогою спеціального брандмауера, правил маршрутизатора або програмного забезпечення брандмауера на основі хоста. Ці порти засмучують зловмисників, оскільки надають дуже мало інформації. Іноді вони відповідають повідомленнями про помилку ICMP, наприклад, код 3 типу 13 (одержувач недоступний: зв'язок заборонено адміністративно), але фільтри, які просто пропускають зонди без відповіді, є набагато більш поширеними. Це змушує Nmap повторити спробу кілька разів на випадок, якщо зонд було відкинуто через перевантаження мережі, а не через фільтрацію. Цей тип фільтрації значно сповільнює сканування.

- unfiltered

Нефільтрований стан означає, що порт доступний, але Nmap не може визначити, відкритий він чи закритий. Лише сканування ACK, яке використовується для відображення наборів правил брандмауера, класифікує порти в цьому стані. Сканування нефільтрованих портів за допомогою інших типів сканування, наприклад сканування вікон, сканування SYN або сканування FIN, може допомогти визначити, чи відкритий порт.

- open|filtered

Nmap переводить порти в цей стан, якщо не може визначити, чи порт відкритий чи відфільтрований. Це трапляється для типів сканування, у яких відкриті порти не відповідають. Відсутність відповіді також може означати, що фільтр пакетів пропустив зонд або будь-яку відповідь, яку він викликав. Тож Nmap не знає напевно, чи порт відкритий, чи його фільтрують. Сканування UDP, IP-протоколу, FIN, NULL і Xmas класифікують порти таким чином.

- closed|filtered

Цей стан використовується, коли Nmap не може визначити, закрито чи відфільтровано порт. Він використовується лише для сканування в режимі очікування IP-ідентифікатора, описаного в розділі «Сканування в режимі простою TCP (-sI)».

Незважаючи на те, що Nmap намагається отримати точні результати вся його інформація базується на пакетах, які повертаються цільовими машинами (або брандмауерами перед ними). Такі хости можуть бути ненадійними та надсилати відповіді, щоб заплутати чи ввести Nmap в оману [17].

Сканування портів виконується для безпеки. Одним із головних принципів мережевої безпеки є те, що зменшення кількості та складності пропонованих послуг зменшує можливість для зловмисників проникнути. Більшість компрометацій у віддаленій мережі відбувається внаслідок використання серверної програми, яка прослуховує порт TCP або UDP. У багатьох випадках ця програма навіть не використовується цільовою організацією, але була ввімкнена за замовчуванням під час налаштування машини. Якби цю службу було вимкнено або захищено брандмауером, атаку було б зірвано [11].

Розуміючи, що кожен відкритий порт є можливістю для компрометації, зловмисники регулярно сканують цілі. Вони порівнюють цей список служб прослуховування зі своїм списком улюблених експлойтів для вразливого програмного забезпечення. Потрібен лише один матч, щоб скомпрометувати машину, створивши опору, яка часто використовується для зараження всієї мережі. Зловмисники, які менш розрізняють, на кого вони спрямовані, часто сканuватимуть лише порт за замовчуванням програми, яка може використовуватися. Це набагато швидше, ніж сканування кожного порту, хоча служба буде пропущена під час роботи на нестандартному порті. Таких зловмисників часто називають «дітками сценаріїв», тому що вони часто знають про безпеку трохи більше, ніж про те, як запустити сценарій експлойту, написаний кимось більш досвідченим. У багатьох організаціях такі зловмисники обов’язково знаходять уразливі хости. Вони можуть бути неабиякою неприємністю, хоча їх величезна кількість і невпинний удар по машинах, доступних через Інтернет, часто спонукають людей швидко виправляти системи. Це зменшує ймовірність більш серйозних цілеспрямованих атак [11].

Важливим захистом від цих зломщиків є регулярне сканування системними адміністраторами власних мереж за допомогою таких інструментів, як Nmap. Візьміть список відкритих портів і вимкніть усі служби, які не використовуються. Переконайтеся, що ті, які мають залишатися доступними, повністю виправлені, і що ви є у списку сповіщень безпеки постачальника. Слід додати правила брандмауера, де це можливо, обмежуючи доступ лише законним користувачам. Інструкції щодо посилення доступні в Інтернеті для більшості популярних програм, що ще більше зменшує можливості зломщика. Деякі адміністратори намагаються замість цього використовувати netstat, але це погано масштабується. Для цього потрібен доступ до кожної машини, а деякі мобільні машини легко пропустити. Крім того, ви не можете запустити netstat на звичайній бездротовій точці доступу, телефоні VoIP або принтері. Крім того, завжди існує ризик того, що скомпрометована машина матиме троянський netstat, який видає неправдиву інформацію. Більшість сучасних руткітів, встановлених зловмисниками, містять цю функцію. Покладатися виключно на Nmap також є помилкою. Рекомендується поєднання ретельного проектування, аудиту конфігурації та регулярного сканування [11].

Хоча безпека є найпоширенішою причиною сканування портів, адміністратори часто виявляють, що воно також підходить для інших цілей. Створення списку машин і послуг, які вони пропонують, може бути корисним для відстеження активів, проектування мережі, перевірки відповідності політикам, відстеження ліцензій на програмне забезпечення, тестування доступності, налагодження мережі тощо [11].

Файл реєстрації портів Nmap (nmap-services) містить емпіричні дані про те, як часто кожний порт TCP або UDP виявляється відкритим. Ці дані були зібрані шляхом сканування десятків мільйонів інтернет-адрес, а потім поєднання цих результатів із внутрішніми даними сканування, наданими великими підприємствами. За замовчуванням Nmap сканує 1000 найпопулярніших портів кожного протоколу, який його просять сканувати. Крім того, можна завжди вказати опцію -F (швидко), щоб сканувати лише 100 найпоширеніших портів у кожному протоколі, або --top-ports, щоб вказати довільну кількість портів для сканування [17].

Сканування портів може бути вимогливою задачею під час використання Nmap. Хоча сам по собі Nmap є швидким і ефективним, вручну налаштовувати параметри часто допомагає збільшити його продуктивність. У Nmap існує широкий спектр опцій для налаштування інтенсивності та швидкості сканування, щоб задовольнити потреби користувача [17].

Найкращі параметри сканування портів [17]:

- від -T0 до -T5

Ці шаблони часу впливають на багато змінних, пропонуючи простий спосіб налаштувати загальну швидкість Nmap від дуже повільної (-T0) до надзвичайно агресивної (-T5). Шаблон синхронізації можна поєднувати з більш детальними параметрами, описаними нижче, і найбільш детальний параметр має перевагу.

- --min-rtt-timeout, --max-rtt-timeout, --initial-rtt-timeout

Мінімальний, максимальний і початковий проміжок часу, протягом якого Nmap чекатиме на відповідь сканування порту.

- --host-timeout

Просить Nmap відмовитися від хостів, сканування яких займає більше, ніж задано.

- --min-rate, --max-rate

Встановлює кількість тестових пакетів, які Nmap надсилає за секунду, відповідно до підлоги та стелі.

- --max-retries

Визначає максимальну кількість повторних передач сканування порту на один порт.

- --min-hostgroup, --max-hostgroup

Встановлює мінімальну та максимальну кількість хостів, порти яких Nmap скануватиме паралельно.

- --min-parallelism, --max-parallelism

Обмежує мінімальну або максимальну кількість зондів сканування портів (на всіх хостах, сканованих одночасно), які Nmap може мати на розгляді.

- --scan-delay, --max-scan-delay

Просить Nmap зачекати принаймні заданий проміжок часу між надсиланням зондів на будь-який окремий хост. Затримка сканування може зростати, коли Nmap виявляє втрату пакетів, тому максимум можна вказати за допомогою --max-scan-delay.

Параметри формату виводу та детальності [17]:

Найпопулярніші параметри виведення Nmap, застосовні до сканування портів:

- -v

Підвищує рівень детальності, змушуючи Nmap друкувати більше інформації про поточне сканування. Відкриті порти відображаються по мірі їх знаходження, а також надається приблизний час завершення, якщо Nmap вважає, що сканування триватиме більше кількох хвилин.

- -d

Підвищує рівень налагодження, змушуючи Nmap друкувати подробиці про свою роботу, які можуть бути корисними для відстеження помилок або просто для розуміння того, як це працює. Вищі рівні призводять до величезних обсягів даних. Використання параметра один раз встановлює рівень налагодження на одиницю, і він збільшується для кожного додаткового -d. Або можна слідувати за -d з потрібним рівнем, як у -d5. Якщо недостатньо інформації, можна спробувати вищий рівень. Максимальний ефективний рівень - дев'ять. Якщо екран заповнений занадто великою кількістю налагоджувальних даних, можна зменшити рівень. Зменшення інтенсивності сканування, наприклад кількості сканованих портів або цілей і використовуваних функцій, також може допомогти ізолювати лише ті повідомлення налагодження, які ви хочете.

- --packet-trace

Змушує Nmap друкувати підсумок кожного надісланого або отриманого пакету. Це часто використовується для налагодження, але також є цінним способом для нових користувачів зрозуміти, що саме робить Nmap під обкладинками. Щоб уникнути друкування тисяч рядків, ви можете вказати обмежену кількість портів для сканування, наприклад -p20-30.

- -oN <ім'я файлу> (звичайний вихід)

Запис вихідних даних у звичайному форматі Nmap до <filename>. Цей формат приблизно такий самий, як стандартний інтерактивний вихід, що друкується Nmap під час виконання.

- -oX <ім'я файлу> (вихід XML)

Запис вихідних даних у форматі XML Nmap до <filename>. Звичайний (читабельний) вихід все одно буде надрукований у стандартний вихід, якщо не буде запиту скерувати XML, вказавши - як <ім'я файлу>. Це кращий формат для використання сценаріями та програмами, які обробляють результати Nmap.

- -oG <ім'я файлу> (вихід у форматі grepable)

Запис вихідних даних у так званому форматі grepable Nmap до <filename>. Цей табличний формат поміщає вивід кожного хоста в один рядок, що полегшує grep для відкритих портів, певних операційних систем, назв програм або інших даних. Звичайний вихід все одно друкуватиметься у stdout, якщо ви не попросите спрямувати туди вихід grepable, вказавши - як <ім’я файлу>. Незважаючи на те, що цей формат добре працює для аналізу простими командними рядками grep і awk, важливі сценарії та програми повинні замість цього використовувати вивід XML. Формат XML містить значну інформацію, для якої немає місця у форматі grepable, а розширюваність полегшує оновлення XML новою інформацією без поломки інструментів, які покладаються на неї.

- -oA <базова назва> (виведення в усі формати)

Для зручності можна вказати -oA <basename>, щоб зберігати результати сканування одночасно у звичайному, XML і grepable форматах. Вони зберігаються в <basename>.nmap, <basename>.xml і <basename>.gnmap, відповідно.

- --resume <ім'я файлу>

Відновлення перерваного сканування, вказавши звичайний (-oN) або grepable (-oG) вихідний файл, створений під час нещасливого сканування. Не можна використовувати параметри, окрім --resume, оскільки Nmap використовуватиме параметри, вказані у вихідному файлі. Потім він аналізує файл і відновлює сканування (і реєстрацію у файл) на хості, на якому працювало попереднє виконання Nmap, коли воно припинилося.

- -- append-output

Вказує Nmap додати результати сканування до будь-яких вказаних вихідних файлів (з такими аргументами, як -oN або -oX), а не перезаписувати їх.

- --open

Показує лише хости, які мають відкриті порти, і показує лише відкриті порти для них. Тут «відкриті порти» — це будь-які порти, які можуть бути відкритими, включаючи відкриті, відкриті|фільтровані та нефільтровані.

Параметри ухилення від брандмауера та IDS:

Nmap пропонує багато варіантів, щоб непомітно пройти повз IDS або обійти правила брандмауера.

- -6

Просить Nmap сканувати ціль за допомогою протоколу IPv6.

- -r

Nmap за замовчуванням рандомізує порядок сканування портів, щоб трохи ускладнити виявлення. Опція -r змушує сканувати їх у нумерованому порядку.

- Pn

Каже Nmap пропустити тест ping і просто просканувати кожен наданий цільовий хост.

2.2. Дослідження методів сканування портів з метою виявлення активних сервісів

Сканування TCP SYN (Stealth) (-sS) [19]:

Сканування SYN є стандартним і найпопулярнішим варіантом сканування. Його можна виконати швидко, скануючи тисячі портів за секунду у швидкій мережі, якій не перешкоджають нав’язливі брандмауери. Сканування SYN є відносно ненав'язливим і прихованим, оскільки воно ніколи не завершує TCP-з'єднання. Він також працює проти будь-якого сумісного стеку TCP, а не залежить від особливостей конкретних платформ, як це роблять FIN/NULL/Xmas, Maimon і сканування в режимі

очікування Nmap. Це також дозволяє чітко та надійно розрізняти відкриті, закриті та відфільтровані стани.

Сканування SYN можна здійснити, передавши опцію -sS до Nmap. Він вимагає привілеїв необроблених пакетів і є стандартним скануванням TCP, якщо вони доступні. Таким чином, коли Nmap запускається як root або адміністратор, -sS зазвичай опускається.

TCP Connect сканування (-sT) [20]:

Сканування підключення TCP є типом сканування TCP за замовчуванням, коли сканування SYN недоступне. Це випадок, коли користувач не має привілеїв необроблених пакетів або сканує мережі IPv6. Замість запису необроблених пакетів, як це робить більшість інших типів сканування, Nmap просить базову ОС встановити з'єднання з цільовою машиною та портом, надсилаючи системний виклик connect. Це той самий системний виклик високого рівня, який веббраузери, P2P-клієнти та більшість інших мережевих програм використовують для встановлення з'єднання. Це частина інтерфейсу програмування, відомого як Berkeley Sockets API. Замість того, щоб зчитувати необроблені відповіді пакетів з мережі, Nmap використовує цей API для отримання інформації про статус під час кожної спроби підключення. Це та сканування відмов FTP є єдиними типами сканування, доступними для непривілейованих користувачів.

Якщо сканування SYN доступне, це зазвичай кращий вибір. Nmap має менше контролю над викликом підключення високого рівня, ніж з необробленими пакетами, що робить його менш ефективним. Системний виклик завершує підключення до відкритих цільових портів, а не виконує напів відкрите скидання, яке виконує сканування SYN. Це не тільки займає більше часу та вимагає більше пакетів для отримання тієї самої інформації, але цільові машини, швидше за все, реєструватимуть з'єднання. Пристойний IDS також вловить, але більшість машин не мають такої сигналізації. Багато служб у середній системі Unix додадуть примітку до системного журналу, а іноді й загадкове повідомлення про помилку, коли Nmap підключається, а потім закриває з'єднання без надсилання даних. Інтерпретацію івдровіді Nmap на SYN-зонд можна побачити у таблиці 2.1.

Таблиця. 2.1.

Інтерпретація відповіді Nmap на SYN-зонд

Відповідь	**Присвоєний стан**
TCP SYN/ACK відповідь	open
TCP RST відповідь	closed
Відповіді не отримано (навіть після повторної передачі)	filtered
ICMP помилка недоступності (type 3, code 1, 2, 3, 9, 10, or 13)	filtered

Сканування UDP (-sU) [21]:

Хоча більшість популярних служб в Інтернеті працюють через протокол TCP, служби UDP широко розгорнуті. DNS, SNMP і DHCP (зареєстровані порти 53, 161/162 і 67/68) є трьома найпоширенішими. Оскільки сканування UDP зазвичай повільніше та складніше, ніж TCP, деякі аудитори безпеки ігнорують ці порти. Це помилка, оскільки UDP-сервіси, які можна використовувати, є досить поширеними, і зловмисники, звичайно, не ігнорують весь протокол.

Сканування UDP активується за допомогою опції -sU. Його можна поєднати з типом сканування TCP, таким як сканування SYN (-sS), щоб перевірити обидва протоколи під час одного запуску.

Сканування UDP працює, надсилаючи пакет UDP до кожного цільового порту. Для більшості портів цей пакет буде порожнім (без корисного навантаження), але для кількох більш поширених портів буде надіслано корисне навантаження, що залежить від протоколу. На основі відповіді або її відсутності порт призначається до одного з чотирьох станів, як показано у таблиці 2.2.

Таблиця. 2.2.

Інтерпретація відповіді Nmap на UDP-зонд

Відповідь	**Присвоєний стан**
Будь-яка відповідь UDP від цільового порту (незвично)	open

Продовження таблиці 2.2.

Відповідь	**Присвоєний стан**
Відповіді не отримано (навіть після повторної передачі)	open|filtered
Помилка порту ICMP недоступний (тип 3, код 3)	closed
Інші помилки недоступності ICMP (тип 3, код 1, 2, 9, 10 або 13)	filtered

Найцікавішим елементом цієї таблиці може бути відкритий|відфільтрований стан. Це ознака найбільших проблем зі скануванням UDP: відкриті порти рідко реагують на порожні зонди. Ті порти, для яких Nmap має корисне навантаження, що залежить від протоколу, з більшою ймовірністю отримають відповідь і будуть позначені як відкриті, але для решти цільовий стек TCP/IP просто передає порожній пакет програмі-прослухувачу, яка зазвичай негайно відкидає його. як недійсний. Якби відповідали порти в усіх інших станах, тоді всі відкриті порти можна було б визначити шляхом виключення. Відомо, що брандмауери та пристрої фільтрації відкидають пакети без відповіді. Отже, коли Nmap не отримує відповіді після кількох спроб, він не може визначити, чи порт відкритий чи відфільтрований.

Одна із проблем, яка стосується сканування UDP — це зробити це швидко. Відкриті та відфільтровані порти рідко надсилають будь-яку відповідь, залишаючи Nmap час очікування, а потім повторні передачі на випадок втрати зонду чи відповіді. Закриті порти часто є ще більшою проблемою. Зазвичай вони надсилають повідомлення про помилку недоступності порту ICMP. Але на відміну від RST-пакетів, які надсилаються через закриті порти TCP у відповідь на SYN або сканування з’єднання, багато хостів за замовчуванням обмежують кількість повідомлень про недоступність порту ICMP. Linux і Solaris особливо суворі щодо цього.

Nmap виявляє обмеження швидкості та відповідно сповільнює роботу, щоб уникнути переповнення мережі непотрібними пакетами, які скине цільова машина. На жаль, обмеження в стилі Linux в один пакет на секунду призводить до того, що сканування 65 536 портів займає більше 18 годин. Отже, кілька порад щодо покращення продуктивності сканування UDP [21]:

- збільшити паралелізм хостів

Якщо Nmap отримує лише одну помилку недоступності порту від одного цільового хосту за секунду, він може отримати 100/секунду, просто просканувавши 100 таких хостів одночасно. Реалізувати це можна, передавши велике значення у --min-hostgroup.

- на початку сканувати популярні порти

Зазвичай використовується дуже мало номерів портів UDP. Сканування найпоширеніших 100 портів UDP (параметр -F) завершиться швидко. Потім ви можна досліджувати ці результати, запускаючи багатоденне сканування мережі на 65 тис. портів у фоновому режимі.

- додати --version-intensity 0 до сканувань виявлення версій

Визначення версії (-sV) часто потрібне, щоб відрізнити відкриті порти UDP від відфільтрованих. Виявлення версій відбувається відносно повільно, оскільки воно передбачає надсилання великої кількості тестів, що стосуються протоколу програми, до кожного відкритого або відкритого|відфільтрованого порту, знайденого на цільових машинах. Якщо вказати --version-intensity 0, Nmap намагатиметься перевірити лише ті тести, які, швидше за все, будуть ефективними для даного номера порту. Це робиться за допомогою даних із файлу nmap-service-probes. Вплив цього параметра на продуктивність є значним.

- сканування через брандмауер

Як і у випадку з TCP, фільтри пакетів можуть значно уповільнити сканування. Багато сучасних міжмережевих екранів спрощують установку обмежень швидкості пакетів.

- використання --host-timeout, щоб пропустити повільні хости

Хости з обмеженою швидкістю ICMP можуть зайняти на порядки більше часу для сканування, ніж ті, які відповідають на кожне зондування швидким недосяжним пакетом. Якщо вказати максимальний час сканування (наприклад, 15 хвилин протягом 15 хвилин), Nmap відмовляється від окремих хостів, якщо він не завершив їх сканування протягом цього часу. Це дозволяє швидко сканувати всі відповідні хости. Потім ви можете працювати на повільних хостах у фоновому режимі.

- використання -v

З увімкненою детальністю (-v) Nmap надає приблизний час для завершення сканування кожного хосту.

Сканування TCP FIN, NULL і Xmas (-sF, -sN, -sX) [22]:

Ці три типи сканування використовують тонку лазівку в TCP RFC для розрізнення відкритих і закритих портів. Сторінка 65 RFC 793 говорить, що якщо стан цільового порту „закрито" вхідний сегмент, який не містить RST, спричиняє надсилання RST у відповідь. Потім на наступній сторінці обговорюються пакети, надіслані на відкриті порти без встановлених бітів SYN, RST або ACK, і зазначається, що: «ви навряд чи потрапите сюди, але якщо потрапите, відпустіть сегмент і поверніться».

Під час сканування систем, сумісних із цим текстом RFC, будь-який пакет, що не містить бітів SYN, RST або ACK, призведе до повернення RST, якщо порт закритий, і жодної відповіді, якщо порт відкритий. Поки жоден із цих трьох бітів не включено, будь-яка комбінація інших трьох (FIN, PSH та URG) допустима. Nmap використовує це за допомогою трьох типів сканування:

- сканування NULL (-sN)

Не встановлює жодних бітів (заголовок прапора TCP дорівнює 0)

- сканування FIN (-sF)

Встановлює лише біт TCP FIN.

- Xmas сканування (-sX)

Встановлює прапорці FIN, PSH і URG, освітлюючи пакет, як новорічну ялинку.

Ці три типи сканування абсолютно однакові за поведінкою, за винятком прапорів TCP, установлених у тестових пакетах. Відповіді обробляються, як показано у таблиці 2.3.

Таблиця. 2.3.

Інтерпретація відповіді Nmap на сканування NULL, FIN або Xmas

Відповідь	**Присвоєний стан**
Відповіді не отримано (навіть після повторної передачі)	open|filtered
Пакет TCP RST	closed

Продовження таблиці 2.3.

Відповідь	**Присвоєний стан**
Помилка недоступності ICMP (тип 3, код 1, 2, 3, 9, 10 або 13)	filtered

Ключовою перевагою цих типів сканування є те, що вони можуть проникати через певні брандмауери без контролю стану та маршрутизатори фільтрації пакетів. Такі брандмауери намагаються запобігти вхідним TCP-з'єднанням (одночасно дозволяючи вихідні), блокуючи будь-які TCP-пакети з установленим бітом SYN і очищеним ACK. Ця конфігурація досить поширена, тому команда брандмауера iptables Linux пропонує спеціальний параметр --syn для її реалізації. Сканування NULL, FIN і Xmas очищають біт SYN і, таким чином, виконують ці правила.

Ще одна перевага полягає в тому, що ці типи сканування трохи більш приховані, ніж навіть сканування SYN. Однак не можна розраховувати на це — більшість сучасних продуктів IDS можна налаштувати для їх виявлення.

Великим недоліком є те, що не всі системи дотримуються RFC 793 до букви. Деякі системи надсилають відповіді RST на зонди незалежно від того, відкритий порт чи ні.

Іншим недоліком цих сканувань є те, що вони не можуть відрізнити відкриті порти від певних відфільтрованих. Якщо фільтр пакетів надсилає повідомлення про заборону призначення ICMP, Nmap дізнається, що порт відфільтровано. Але більшість фільтрів просто скидають заборонені зонди без жодної відповіді, через що порти виглядають відкритими. Оскільки Nmap не може бути впевнений, що саме так, він позначає невідповідні порти як відкриті|відфільтровані. Додавання визначення версії (-sV) може усунути неоднозначність, як це робиться зі скануванням UDP, але це руйнує більшу частину прихованого характеру цього сканування.

Спеціальні типи сканування з –scanflags [23]:

Параметр --scanflags дозволяє розробити власне сканування, вказавши довільні позначки TCP.

Аргумент --scanflags може бути числовим значенням прапора, наприклад 9 (PSH і FIN), але використовувати символічні імена простіше. Можна об’єднати будь-

яку комбінацію URG, ACK, PSH, RST, SYN і FIN. Наприклад, --scanflags URGACKPSHRSTSYNFIN встановлює все, хоча це не дуже корисно для сканування. Порядок, у якому вони вказані, не має значення.

На додаток до визначення бажаних прапорів, можна вказати тип сканування TCP (наприклад, -sA або -sF). Цей базовий тип повідомляє Nmap, як інтерпретувати відповіді. Наприклад, сканування SYN розглядає відсутність відповіді як ознаку відфільтрованого порту, тоді як сканування FIN розглядає те саме як відкритий|фільтрований. Nmap поводитиметься так само, як і для базового типу сканування, за винятком того, що натомість використовуватиме позначки TCP, які в вкажете. Якщо ви не вкажете базовий тип, використовується сканування SYN.

Спеціальне сканування SYN/FIN [24]:

Одним із цікавих користувацьких типів сканування є SYN/FIN. Іноді адміністратор брандмауера або виробник пристрою намагатиметься заблокувати вхідні з’єднання за допомогою такого правила, як «відкинути будь-які вхідні пакети лише з установленим прапором SYN». Вони обмежують його лише прапором SYN, оскільки не хочуть блокувати пакети SYN/ACK, які повертаються як другий крок вихідного з’єднання.

Проблема цього підходу полягає в тому, що більшість кінцевих систем сприймуть початкові пакети SYN, які також містять інші прапори (не ACK). Наприклад, система відбитків пальців Nmap OS надсилає пакет SYN/FIN/URG/PSH на відкритий порт. Більше половини відбитків пальців у базі даних відповідають SYN/ACK. Таким чином, вони дозволяють сканувати порти за допомогою цього пакету та загалом також дозволяють створювати повне TCP-з’єднання. Відомо, що деякі системи навіть відповідають SYN/ACK на пакет SYN/RST! TCP RFC неоднозначно визначає, які прапори прийнятні в початковому пакеті SYN, хоча SYN/RST, безумовно, виглядає фіктивним.

Сканування TCP ACK (-sA) [25]:

Це сканування відрізняється від інших, тим, що воно ніколи не визначає відкриті (або навіть відкриті|відфільтровані) порти. Він використовується для

відображення наборів правил брандмауера, визначення того, чи мають вони статус чи ні, і які порти фільтруються.

ACK-сканування вмикається за допомогою параметра -sA. Його пробний пакет має лише встановлений прапор ACK (якщо не використовується --scanflags). Під час сканування нефільтрованих систем і відкритий, і закритий порти повертатимуть пакет RST. Потім Nmap позначає їх як нефільтровані, що означає, що вони доступні пакетом ACK, але невизначено, чи вони відкриті чи закриті. Порти, які не відповідають або надсилають певні повідомлення про помилки ICMP, позначаються як відфільтровані. Таблиця 2.4. містить повну інформацію.

Таблиця. 2.4.

Інтерпретація відповіді Nmap запит сканування ACK

Відповідь	**Присвоєний стан**
TCP RST відповідь	unfiltered
Відповіді не отримано (навіть після повторної передачі)	filtered
Помилка недоступності ICMP (тип 3, код 1, 2, 3, 9, 10 або 13)	filtered

Сканування TCP Window (-sW) [26]:

Сканування Window точно таке ж, як сканування ACK, за винятком того, що воно використовує деталі реалізації певних систем, щоб відрізнити відкриті порти від закритих, замість того, щоб завжди друкувати без фільтрів, коли повертається RST. Це робиться шляхом перевірки значення вікна TCP повернутих пакетів RST. У деяких системах відкриті порти використовують позитивний розмір вікна (навіть для пакетів RST), а закриті мають нульове вікно. Сканування вікна надсилає той самий голий тест ACK, що й сканування ACK, інтерпретуючи результати, як показано в таблиці 2.5.

Таблиця. 2.5.

Інтерпретація відповіді Nmap на ACK сканування Windows

Відповідь	**Присвоєний стан**
TCP RST відповідь	unfiltered
Відповіді не отримано (навіть після повторної передачі)	filtered
Помилка недоступності ICMP (тип 3, код 1, 2, 3, 9, 10 або 13)	filtered

Це сканування спирається на деталі впровадження меншості систем в Інтернеті, тому ви не завжди можете йому довіряти. Системи, які його не підтримують, зазвичай повертають усі порти закритими. Звичайно, можливо, що машина дійсно не має відкритих портів. Якщо більшість просканованих портів закриті, але кілька загальних номерів портів (наприклад, 22, 25 і 53) відкриті, система, швидше за все, чутлива. Іноді системи навіть демонструватимуть прямо протилежну поведінку. Якщо сканування показує 997 відкритих портів і три закриті або відфільтровані порти, то ці три цілком можуть бути справді відкритими.

Хоча це сканування не підходить для кожної ситуації, іноді воно може бути дуже корисним.

TCP сканування Маймона (-sM) [27]:

Сканування Маймона названо на честь його першовідкривача Уріеля Маймона. Він описав цю техніку в номері №49 журналу Phrack (листопад 1996). Nmap, який містить цю техніку, був випущений двома випусками пізніше. Ця техніка точно така ж, як сканування NULL, FIN і Xmas, за винятком того, що зонд FIN/ACK. Відповідно до RFC 793 (TCP), пакет RST має бути згенерований у відповідь на такий тест незалежно від того, відкритий чи закритий порт. Однак Уріель помітив, що багато систем, похідних від BSD, просто скидають пакет, якщо порт відкритий. Nmap використовує це для визначення відкритих портів, як показано в таблиці 2.6.

Таблиця. 2.6.

Інтерпретація відповіді Nmap на зонд сканування Маймона

Відповідь	**Присвоєний стан**
Відповіді не отримано (навіть після повторної передачі)	open|filtered
TCP RST пакет	closed
Помилка недоступності ICMP (тип 3, код 1, 2, 3, 9, 10 або 13)	filtered

TCP Idle Scan (-sI) [28]:

У 1998 році дослідник безпеки Антірез опублікував у списку розсилки Bugtraq новий метод сканування портів. Idle scan, як стало відомо, дозволяє повністю сліпе сканування портів. Насправді зловмисники можуть просканувати ціль, не надсилаючи жодного пакета до цільової мережі зі своєї власної IP-адреси. Замість цього розумна атака з бічного каналу дозволяє сканувати від «зомбі-хоста». Звіти IDS вказують на невинного зомбі як на зловмисника. Крім того, що цей тип сканування є надзвичайно прихованим, він дозволяє виявити довірчі відносини на основі IP між машинами.

Незважаючи на те, що сканування в режимі очікування є складнішим, ніж будь-який із розглянутих методів, не потрібно бути експертом у TCP/IP, щоб зрозуміти це. Його можна скласти з таких основних фактів:

- один із способів визначити, чи відкритий TCP-порт, — надіслати до порту пакет SYN (встановлення сеансу). Цільова машина відповість пакетом SYN/ACK (підтвердження запиту на сеанс), якщо порт відкритий, і RST (скидання), якщо порт закрито.
- Машина, яка отримує небажаний пакет SYN/ACK, відповість RST. Небажаний RST буде проігноровано.
- Кожен IP-пакет в Інтернеті має ідентифікаційний номер фрагмента (IP ID). Оскільки багато операційних систем просто збільшують це число для кожного пакета, який вони надсилають, перевірка IPID може сказати зловмиснику, скільки пакетів було надіслано з часу останнього тестування.

Поєднуючи ці ознаки, можна сканувати цільову мережу, підробляючи особу користувача так, щоб виглядало, ніби сканувала невинна машина-зомбі.

Сканування в режимі очікування крок за кроком

По суті, сканування в режимі очікування складається з трьох кроків, які повторюються для кожного порту:

- перевірка IP-ідентифікатора зомбі та запис його.
- підробка SYN-пакету від зомбі та відправка його на потрібний порт цілі. Залежно від стану порту реакція цілі може призвести до збільшення IP-ідентифікатора зомбі, а може й ні.
- знову перевірка IP-ідентифікатора зомбі. Потім стан цільового порту визначається шляхом порівняння цього нового IP-ідентифікатора з ідентифікатором, записаним на кроці 1.

Після цього процесу IP-ідентифікатор зомбі мав збільшитися на один або два. Збільшення на одиницю вказує на те, що зомбі не надсилав жодних пакетів, окрім своєї відповіді на зонд зловмисника. Ця відсутність надісланих пакетів означає, що порт не відкритий (ціль повинна була надіслати зомбі пакет RST, який був проігнорований, або взагалі нічого). Збільшення на два означає, що зомбі надіслав пакет між двома зондами. Цей додатковий пакет зазвичай означає, що порт відкритий (ціль, імовірно, надіслала зомбі пакет SYN/ACK у відповідь на підроблений SYN, який викликав RST-пакет від зомбі). Збільшення більше двох зазвичай означає поганого зомбі-господаря. Він може не мати передбачуваних ідентифікаційних номерів IP або може бути задіяний у зв’язку, не пов’язаному з неактивним скануванням.

Незважаючи на те, що те, що відбувається із закритим портом, дещо відрізняється від того, що відбувається з відфільтрованим портом, зловмисник вимірює той самий результат в обох випадках, а саме збільшення IP-ідентифікатора на 1. Тому сканування в режимі очікування не може відрізнити між закриті та відфільтровані порти. Коли Nmap фіксує збільшення IP-ідентифікатора на 1, це позначає порт як закритий|відфільтрований.

Сканування в режимі очікування – це найкраще приховане сканування. Nmap пропонує сканування-приманку (-D), щоб допомогти користувачам захистити свою особу, але це (на відміну від сканування в режимі очікування) все одно вимагає від зловмисника надіслати кілька пакетів до цілі з його справжньої IP-адреси, щоб отримати результати сканування. Одним із результатів неактивного сканування є те, що системи виявлення вторгнень зазвичай надсилають сповіщення про те, що зомбі-машина розпочала сканування проти них. Таким чином, його можна використовувати для створення іншої сторони для сканування. Майте на увазі цю можливість, читаючи сповіщення з вашого IDS.

Унікальною перевагою сканування в режимі очікування є те, що його можна використовувати для поразки певних брандмауерів і маршрутизаторів фільтрації пакетів. Фільтрування IP-адрес джерела є поширеним (хоча і слабким) механізмом безпеки для обмеження комп'ютерів, які можуть підключатися до конфіденційного хосту або мережі. Наприклад, сервер бази даних компанії може дозволяти підключення лише з публічного вебсервера, який має до нього доступ. Або домашній користувач може дозволити лише підключення SSH (інтерактивний вхід) зі своїх робочих машин.

Більш тривожний сценарій виникає, коли якийсь авторитет компанії вимагає від мережевих адміністраторів відкрити діру в брандмауері, щоб він міг отримати доступ до ресурсів внутрішньої мережі зі своєї домашньої IP-адреси. Це може статися, коли керівники не хочуть або не можуть використовувати безпечні альтернативи VPN.

Сканування в режимі очікування іноді можна використовувати для визначення цих довірчих відносин. Ключовим фактором є те, що в результатах неактивного сканування перелічуються відкриті порти з точки зору хоста-зомбі. Звичайне сканування вищезгаданого сервера бази даних може виявити відсутність відкритих портів, але виконання сканування в режимі очікування з використанням IP-адреси вебсервера як зомбі може виявити довірчі відносини, показавши порти служби, пов'язані з базою даних, відкритими.

Відображення цих довірчих відносин може бути дуже корисним для зловмисників для визначення пріоритетів цілей. Обговорений вище вебсервер може

здаватися зловмиснику звичайним, поки він не помітить його спеціальний доступ до бази даних.

Недоліком сканування в режимі очікування є те, що воно займає набагато більше часу, ніж більшість інших типів сканування. Незважаючи на оптимізовані алгоритми, 15-секундне сканування SYN може тривати 15 хвилин або більше як сканування в режимі очікування. Інша проблема полягає в тому, що повинна бути можливість підробити пакети так, ніби вони надходять від зомбі, і досягти цільової машини.

Сканування IP-протоколу (-sO) [29]:

Сканування IP-протоколів дозволяє визначити, які IP-протоколи (TCP, ICMP, IGMP тощо) підтримуються цільовими машинами. Технічно це не сканування портів, оскільки воно циклічно змінює номери протоколів IP, а не номери портів TCP або UDP. Проте він все ще використовує параметр -p для вибору сканованих номерів протоколів, звітує про результати у звичайному форматі таблиці портів і навіть використовує той самий механізм сканування, що й справжні методи сканування портів. Отже, він досить близький до сканування порту.

Крім того, що сканування протоколу є корисним саме по собі, демонструє потужність програмного забезпечення з відкритим кодом.

Сканування протоколу працює подібно до сканування UDP. Замість проходження по полю номера порту пакета UDP він надсилає заголовки IP-пакетів і виконує ітерацію по восьмибітовому полю протоколу IP. Заголовки зазвичай порожні, не містять даних і навіть належного заголовка для заявленого протоколу. Виняток зроблено для деяких популярних протоколів (зокрема TCP, UDP та ICMP). Для них включено відповідні заголовки протоколів, оскільки деякі системи не надсилають їх інакше, і оскільки Nmap уже має функції для їх створення. Замість того, щоб спостерігати за повідомленнями про недоступність порту ICMP, сканування протоколу шукає повідомлення про недоступність протоколу ICMP. У таблиці 2.7 показано, як відповіді на IP-зонди зіставляються зі станами портів.

Таблиця. 2.7.

Інтерпретація відповіді Nmap на зонд IP-протоколу

Відповідь	**Присвоєний стан**
Будь-яка відповідь у будь-якому протоколі від цільового хоста	open
Помилка протоколу ICMP недоступна (тип 3, код 2)	closed
Інші помилки недоступності ICMP (тип 3, код 1, 3, 9, 10 або 13)	filtered
Відповіді не отримано (навіть після повторної передачі)	open|filtered

Як і відкриті порти в протоколах TCP або UDP, кожен відкритий протокол є потенційним вектором експлуатації. Крім того, результати сканування протоколу допомагають визначити призначення машини та тип фільтрації пакетів. Кінцеві хости зазвичай мають відкритий лише TCP, UDP, ICMP і (іноді) IGMP, тоді як маршрутизатори часто пропонують набагато більше, включаючи протоколи, пов’язані з маршрутизацією, такі як GRE та EGP. Брандмауери та VPN-шлюзи можуть відображати протоколи, пов’язані з шифруванням, наприклад IPsec і SWIPE.

Подібно до повідомлень про недоступність порту ICMP, отриманих під час сканування UDP, повідомлення про недосяжність протоколу ICMP часто мають обмеження по швидкості. Наприклад, за секунду з вікна Linux 2.4.20 за замовчуванням надсилається не більше однієї відповіді ICMP про недосяжність. Оскільки існує лише 256 можливих номерів протоколів, це менша проблема, ніж сканування 65 536 портів UDP. Пропозиції в розділі «Прискорення сканування UDP» стосуються також прискорення сканування IP-протоколу.

Сканування протоколу використовується так само, як і більшість інших методів сканування у командному рядку. Вказується -sO на додаток до будь-яких загальних параметрів Nmap. Параметр звичайного порту (-p) використовується для вибору номерів протоколів. Або ви можна використовувати -F для сканування всіх протоколів, перелічених у базі даних nmap-protocols. За замовчуванням Nmap сканує всі 256 можливих значень.

TCP FTP Bounce Scan (-b) [30]:

Цікавою особливістю протоколу FTP (RFC 959) є підтримка так званих проксі FTP-з'єднань. Це дозволяє користувачеві підключатися до одного FTP-сервера, а потім запитувати надсилання файлів на сторонній сервер. Така функція готова для зловживань на багатьох рівнях, тому більшість серверів припинили її підтримку. Одне зі зловживань, яке дозволяє ця функція, полягає в тому, що FTP-сервер сканує порти інших хостів. Просто попросіть сервер FTP по черзі надіслати файл на кожен цікавий порт цільового хоста. У повідомленні про помилку буде описано, чи відкритий порт чи ні. Це хороший спосіб обійти брандмауери, оскільки організаційні FTP-сервери часто розміщуються там, де вони мають більше доступу до інших внутрішніх хостів, ніж будь-який старий Інтернет-хост. Nmap підтримує FTP сканування відмов з опцією -b. Він приймає аргумент у формі <ім’я користувача>:<пароль>@<сервер>:<порт>. <Server> — це ім’я або IP-адреса вразливого FTP-сервера. Як і у випадку звичайної URL-адреси, ви можете пропустити <username>:<password>, у такому випадку використовуються анонімні облікові дані для входу (користувач: анонімний пароль:-wwwuser@). Номер порту (і попередню двокрапку) також можна опустити, у цьому випадку використовується стандартний порт FTP (21) на <сервері>.

-A (параметр агресивного сканування) [31]

Ця опція включає додаткові розширені та агресивні параметри. Це дозволяє виявлення ОС (-O), сканування версій (-sV), сканування сценаріїв (-sC) і трасування (--traceroute). Суть полягає в тому, щоб увімкнути повний набір параметрів сканування без необхідності запам’ятовувати великий набір прапорців. Однак, оскільки сканування сценаріїв із набором за замовчуванням вважається нав’язливим, не варто використовувати даний параметр проти цільових мереж без дозволу. Цей параметр вмикає лише функції, а не параметри синхронізації (наприклад, -T4) або параметри докладності (-v), які вам також можуть знадобитися. Параметри, які потребують привілеїв (наприклад, root-доступ), такі як виявлення ОС і traceroute, будуть увімкнені, лише якщо ці привілеї доступні.

Висновок до другого розділу

Сканування портів використовується для виявлення відкритих портів на комп'ютері, який знаходиться в мережі. Сканер портів є програмним забезпеченням, розробленим для цієї цілі. Комп'ютерний порт є віртуальною точкою, яка обробляє вхідні та вихідні дані і має велике значення для забезпечення безпеки. Великі мережі можуть використовувати інформацію, отриману в результаті сканування портів, для виявлення потенційно вразливих місць.

У даному розділі були розглянуті, які є порти: відомі, зареєстровані і динамічні/приватні, і топ-20 найчастіше відкритих портів TCP, UDP.

Сканування портів - це процес віддаленого тестування багатьох портів з метою визначення їх стану. Один з найцікавіших станів - це відкритий стан, коли програма прослуховує та приймає з'єднання через даний порт. Існує багато методів для проведення такого сканування.

Nmap може розпізнати кілька основних станів портів. Ось деякі з них:

- Відкритий (Open): Порт відкритий і активний, програма прослуховує та приймає з'єднання через цей порт.
- Закритий (Closed): Порт закритий і неактивний, програма не прослуховує та не приймає з'єднання через цей порт.
- Фільтрований (Filtered): Nmap не може визначити точний стан порту через наявність фільтрів, таких як брандмауер або IDS/IPS, які блокують або фільтрують пакети.
- Unfiltered: Порт не має жодних фільтрів і пакети пропускаються через нього, але Nmap не може визначити його точний стан.
- Open|Filtered: Цей стан вказує на можливість того, що порт може бути відкритим, але на ньому застосовані фільтри, які блокують деякі пакети, тому Nmap не може однозначно визначити його стан.

- Closed|Filtered: Цей стан вказує на те, що порт закритий або неактивний, але на ньому також застосовані фільтри, які блокують деякі пакети, тому Nmap не може однозначно визначити його стан.

У даному розділі також були проаналізовані методи сканування портів, які можуть бути використані для виявлення відкритих портів на цільовій системі. Декілька найпоширеніших із них:

- TCP SYN сканування: Цей метод використовується для виявлення відкритих портів на системі. Він відправляє TCP SYN пакети до цільової системи і очікує відповіді. Якщо отримується відповідь SYN/ACK, то порт вважається відкритим.
- TCP Connect сканування: Цей метод встановлює повне TCP з'єднання з кожним портом, щоб перевірити його стан. Якщо встановлення з'єднання успішне, то порт вважається відкритим.
- UDP сканування: Для виявлення відкритих портів, які використовують UDP протокол, використовується цей метод. Він надсилає UDP пакети до цільової системи і очікує відповіді. Якщо отримується відповідь, то порт вважається відкритим.
- NULL, FIN та XMAS сканування: Ці методи використовуються для виявлення вразливостей в реалізації TCP стеку системи. Вони надсилають пакети з певними флагами (NULL, FIN або URG/PSH/FIN) і спостерігають за реакцією системи.

Ці методи сканування можуть бути комбіновані та налаштовуватися залежно від потреб користувача та контексту використання.

РОЗДІЛ 3
РОЗРОБКА СИСТЕМИ АВТОМАТИЧНОГО СКАНУВАННЯ МЕРЕЖІ

3.1 Програмна реалізація системи автоматичного сканування мережі

Програмна реалізація системи автоматичного сканування мережі зазвичай вимагає використання спеціалізованих мережевих бібліотек або фреймворків, які надають зручний доступ до мережевих функцій та протоколів. Існує багато мов програмування та інструментів, які можуть бути використані для розробки такої системи. Ось декілька прикладів:

1. Python: python є популярною мовою програмування для розробки систем сканування мережі. Бібліотеки, такі як Scapy, Nmap, PyShark, дозволяють взаємодіяти з мережевими пакетами, проводити сканування портів, отримувати інформацію про підключені пристрої та інше.

2. Java: java також має широке застосування у розробці систем сканування мережі. Можна використовувати бібліотеки, такі як Apache MINA або Jpcap, для реалізації мережевої функціональності.

3. C/C++: мови програмування C та C++ можуть бути використані для більш низькорівневої реалізації системи сканування мережі. Наприклад, за допомогою бібліотеки libpcap або WinPcap можна зчитувати та обробляти мережеві пакети.

4. Ruby: ruby має такі бібліотеки, як PacketFu, які дозволяють реалізувати функції сканування мережі та маніпуляції мережевими пакетами.

І це лише кілька прикладів мов програмування та бібліотек, які можуть бути використані для програмної реалізації системи автоматичного сканування мережі.

У даному випадку для програмної реалізації сканеру мережі була обрана мова python та бібліотека Nmap. Код найпростішої програми можна знайти у додатку А.

На рисунку 3.1 ми можемо побачити структурно-логічну схему найпростішого сканера. Елемент імпорту мережевих модулів, виконується задля підключення до домена. Даний модуль спеціально призначений для взаємодії з Nmap і виконання

мережевого сканування. Елемент інтерфейсу відповідає за запит домена від користувача та виведення інформації щодо сканування. Процес сканування включає в себе наступні етапи: отримання користувачем домену з елемента інтерфейсу, створення з'єднання з вказаним доменом, сканування всіх портів цього домену і передачу зібраних даних до відповідного елементу інтерфейсу для подальшого виведення. Код стандартного сканеру див. Додаток А [31].

Рисунок.3.1. Структурно-логічна схема найпростішого сканеру.

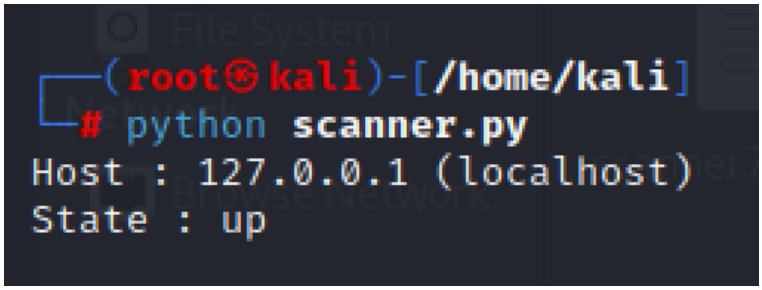

Рисунок. 3.2. Результат сканування найпростішого сканера використовуючи Nmap.

У реалізації системи, враховується можлива протидія засобам мережевого захисту шляхом використання різних параметрів сканування. До даних параметрів відносяться:

- параметр --randomize-hosts – цей параметр дозволяє випадковим чином змінювати порядок, в якому скануються хости, що може ускладнити виявлення сканування та блокування, він використовується під час прихованого сканування;

- параметр -D RND:5 для маскування трафіку використовуючи випадкові IP-адреси для входження в проксі-ланцюг, що дозволяє приховати джерело сканування і ускладнює виявлення атакуючого комп'ютера;

- параметр -T4 для помірно швидкого і надійного з’єднання між вихідною та цільовою мережами.

У вдосконаленій розробленій системі автоматичного сканування мережі процес сканування починається з введення цілі сканування. Ми це можемо побачити на рисунку 3.3.

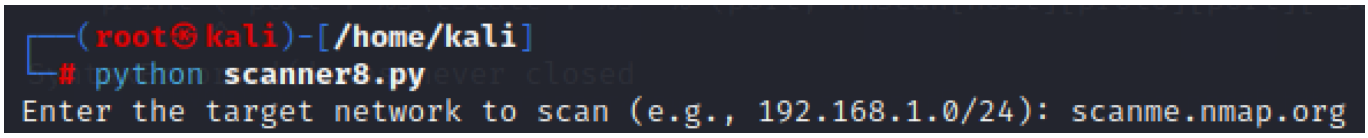

Рисунок. 3.3. Введення цілі сканування.

У наступних рядках встановлюються параметри Nmap для типу сканування. Якщо тип сканування (scan_type) дорівнює значенню «0»,, використовуються параметри для прихованого сканування (stealth). Якщо тип сканування (scan_type) дорівнює значенню «1»,, використовуються параметри для агресивного сканування (aggressive). Основна відмінність між ними полягає в рівні видимості або шуму, який створюється під час процесу сканування. Якщо тип сканування має інше значення, виводиться повідомлення про недійсний тип скану і функція завершується. Приклади можна побачити на рис. 3.4. – 3.6.

```
if scan_type == 0:
    # Set Nmap options for stealth scanning
```

```
        scan_arguments = '-sS -T4 -p 1-1000 --randomize-hosts -D RND:5'
    elif scan_type == 1:
        # Set Nmap options for aggressive scanning
        scan_arguments = '-A -T5'
    else:
        print("Invalid scan type. Please choose either 0 (stealth) or 1 (aggressive).")
        return
```

Лістинг 3.1.

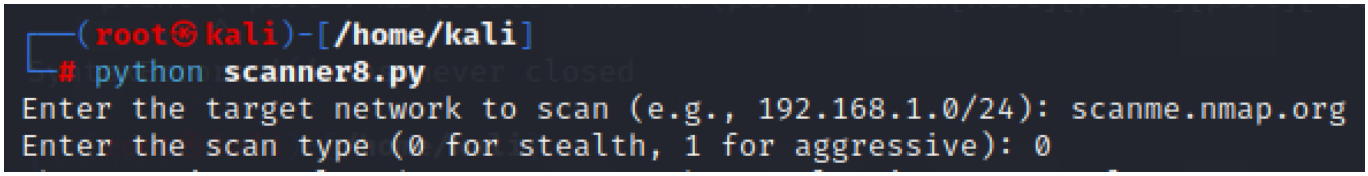

Рисунок. 3. 4. Введення прихованого типу сканування.

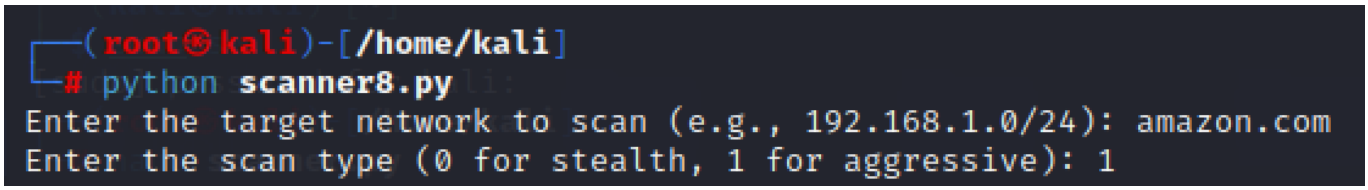

Рисунок. 3.5. Введення агресивного типу сканування.

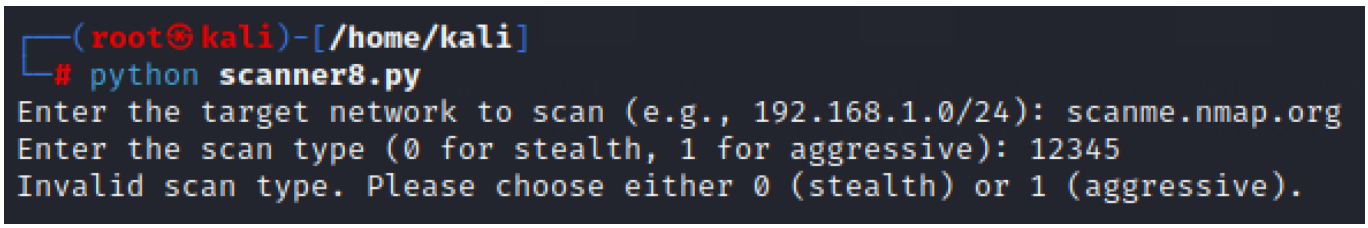

Рисунок. 3.6. Результат перевірки коду шляхом введення невірного значення для вибору типу сканування.

Приховане-сканування - це метод сканування, при якому сканер прагне залишатися непомітним. Використовуючи приховане сканування (stealth scanning), сканер старається уникати будь-яких дій, які можуть спровокувати відповідь від системи, яку він сканує. Приховане сканування може бути корисним при проведенні

високоанонімних або розвідувальних досліджень мережі. Його використовують етичні хакери і адміністратори мереж для виявлення вразливостей систем без звертання надмірної уваги адміністраторів [32].

Агресивне сканування - це метод сканування, який включає активні атакуючі дії з метою виявлення розміщення, налаштувань або вразливостей цільової системи. Агресивне сканування часто використовується зловмисниками з метою злому системи, пошуку потенційних цілей або збору інформації. Воно може бути помітним і залишати відповідні сліди в системних журналах. Такий тип сканування може викликати відповідь або спрацюювання механізмів оборони в системі, що може призвести до блокування або виявлення спроби вторгнення [33].

Наступні рядки у коді встановлюють додаткові методи уникнення виявлення. Генерується випадкова затримка між 0.5 і 2.0 секундами. Якщо тип сканування (scan_type) дорівнює значенню «0», затримка сканування прихованого типу встановлюється на згенеровану випадкову затримку (random.seed). Якщо тип сканування (scan_type) має інше значення, затримка сканування встановлюється на «0». Також виконується затримка між скануваннями за допомогою time.sleep(). це допомагає уникнути виявлення патерну сканування шляхом випадкового розподілу часу між скануваннями.

```
random.seed()
random_delay = random.uniform(0.5, 2.0)
scan_delay = random_delay if scan_type == 0 else 0
time.sleep(scan_delay)
```

Лістинг 3.2.

За результатами сканування можна побачити час сканування і зробити висновок, що хост в мережі і подивитися які порти дали відповідь.

На рисунку 3.7 результат прихованого сканування свідчить про те, що даний процес зайняв приблизно 31.55 секунди. Хост з IP-адресою 45.33.32.156 був просканований і виявлений живим та відповідаючим. Під час сканування було використано протокол TCP для перевірки відкритих портів. В результаті сканування

було виявлено два порти: порт 22 (SSH) та порт 80 (HTTP). Обидва цих порти були відкритими, що означає, що на них працюють служби або програми, доступні з пристрою сканування. Результати сканування були збережені у файлі "scan_results_20230520050822.txt"

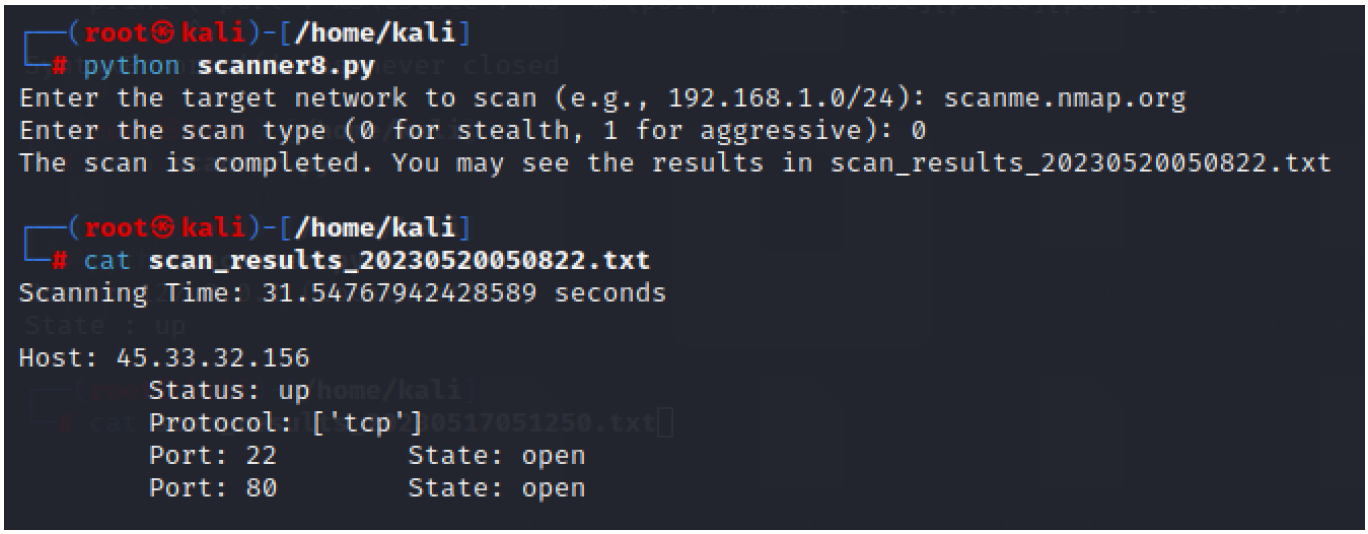

Рисунок. 3.7. Результат сканування scanme.nmap.org.

Результат сканування ресурсу ril.com (рисунок 3.8) показує що на час сканування пішло 12.9 секунд. Хост є у мережі. Під час сканування був використаний протокол TCP. У даному випадку відкритим є порт 443, HTTP Secure. Результати сканування були збережені у файлі "scan_results_20230520052843.txt"

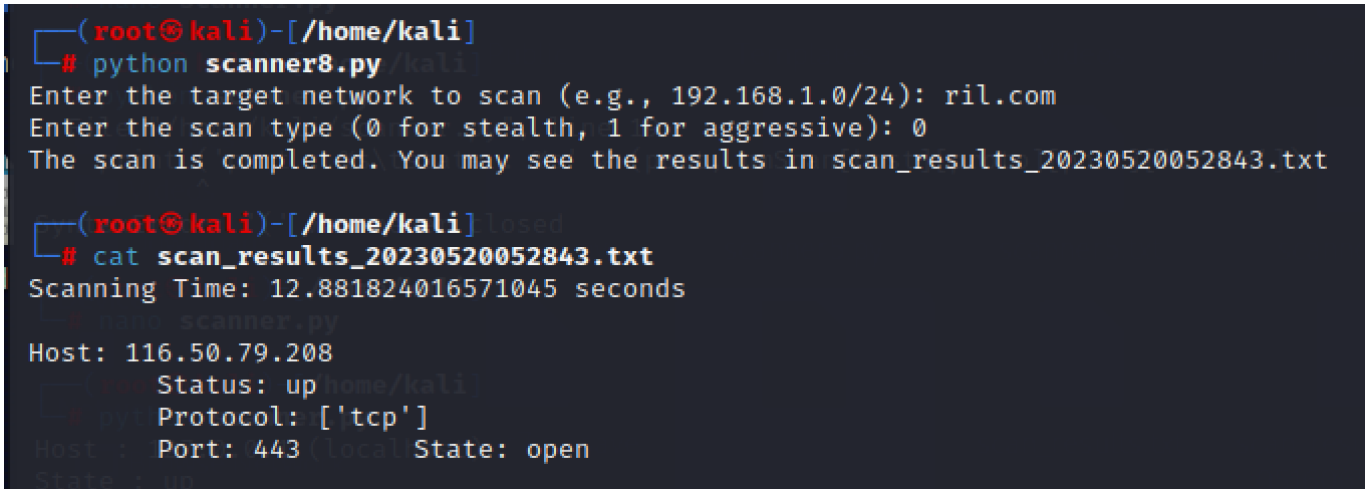

Рисунок. 3.8. Результат сканування ril.com.

```
┌──(root㉿kali)-[/home/kali]
└─# python scanner8.py
Enter the target network to scan (e.g., 192.168.1.0/24): djaweb.dz
Enter the scan type (0 for stealth, 1 for aggressive): 0
The scan is completed. You may see the results in scan_results_20230520053017.txt
┌──(root㉿kali)-[/home/kali]
└─# cat scan_results_20230520053017.txt
Scanning Time: 9.749470710754395 seconds

Host: 197.112.32.13
        Status: up
        Protocol: []
```

Рисунок. 3.9. Результат сканування djaweb.dz.

Інформація показана на рисунку 3.9 свідчить про те, що процес сканування зайняв приблизно 9.75 секунди. Цільова мережа djaweb.dz була просканована за допомогою прихованого сканування. Результати сканування були збережені у файлі "scan_results_20230520053017.txt". Хост з IP-адресою 197.112.32.13 був виявлений як доступний, що свідчить про його відповідь. Однак, під час сканування не було виявлено конкретної інформації щодо протоколу.

```
┌──(root㉿kali)-[/home/kali]
└─# python scanner8.py
Enter the target network to scan (e.g., 192.168.1.0/24): 192.168.1.10
Enter the scan type (0 for stealth, 1 for aggressive): 0
The scan is completed. You may see the results in scan_results_20230520053140.txt
┌──(root㉿kali)-[/home/kali]
└─# cat scan_results_20230520053140.txt
Scanning Time: 10.548622846603394 seconds

Host: 192.168.1.10
        Status: up
        Protocol: ['tcp']
        Port: 139       State: open
```

Рисунок. 3.10. Результат сканування одного із хостів (IP-адреса замінена) у локальній мережі. Прихований тип сканування.

Результат сканування на рисунку 3.10 був виведений за 10.55 секунд. Був проскановий один із локальних хостів (адреса замінена). Результат сканування був

збережений у файлі "scan_results_20230520054019.txt"Для сканування був використаний протокол TCP. Відкритий порт 139, який відповідає за NetBIOS.

```
┌──(root㉿kali)-[/home/kali]
└─# python scanner8.py
Enter the target network to scan (e.g., 192.168.1.0/24): amazon.com
Enter the scan type (0 for stealth, 1 for aggressive): 1
The scan is completed. You may see the results in scan_results_20230520054019.txt

┌──(root㉿kali)-[/home/kali]
└─# cat scan_results_20230520054019.txt
Scanning Time: 144.357435464859 seconds

Host: 205.251.242.103
        Status: up
        Protocol: ['tcp']
        Port: 80        State: open
        Port: 443       State: open
```

Рисунок. 3.11. Результат сканування amazon.com.

У випадку з скануванням amazon.com (рисунок 3.11) надана інформація свідчить про те, що процес сканування зайняв приблизно 144.36 секунди. Цільова мережа "amazon.com" була просканована за допомогою агресивного сканування (тип 1). Результати сканування були збережені у файлі "scan_results_20230520054019.txt". Хост з IP-адресою 205.251.242.103 був виявлений як доступний, що свідчить про його відповідь. Сканування специфічно використовувало протокол TCP для перевірки відкритих портів. У результаті сканування було виявлено два порти: порт 80 (HTTP) та порт 443 (HTTPS).

На наступному рисунку 3.12 показаний результат сканування одного із хостів у локальній мережі за агресивним типом. Результати сканування були збережені у файлі "scan_results_20230520054442.txt". Час сканування зайняв 46.35 секунди. За нього був виявлений доступний хост з IP-адресою 192.168.1.10. Під час сканування був використаний протокол TCP. Відкриті порти показані на рисунку 3.12.

```
┌──(root㉿kali)-[/home/kali]
└─# python scanner8.py
Enter the target network to scan (e.g., 192.168.1.0/24): 192.168.1.10
Enter the scan type (0 for stealth, 1 for aggressive): 1
The scan is completed. You may see the results in scan_results_20230520054442.txt

┌──(root㉿kali)-[/home/kali]
└─# cat scan_results_20230520054442.txt
Scanning Time: 46.357260942459106 seconds

Host: 192.168.1.10
        Status: up
        Protocol: ['tcp']
        Port: 135       State: open
        Port: 139       State: open
        Port: 445       State: open
        Port: 1042      State: open
        Port: 1043      State: open
```

Рисунок. 3.12. Результат сканування одного із хостів (IP-адреса замінена) у локальній мережі. Агресивний тип сканування.

На ще одному нижче наведеному рисунку 3.12 представлені результати сканування одного із хостів (IP-адреса замінена) у локальній мережі, використовуючи агресивний тип сканування. Тут можна побачити що цільовий хост доступний, оскільки він відповідає на запити. Порт 1947 перебуває у стані filtered. Це означає, що порт не повністю відкритий або доступний, і можливо, існує деяке фільтрування мережі або налаштування брандмауера, яке перешкоджає прямому доступу до нього.

```
┌──(root㉿kali)-[/home/kali]
└─# python scanner8.py
Enter the target network to scan (e.g., 192.168.1.0/24): 192.168.1.11
Enter the scan type (0 for stealth, 1 for aggressive): 1
The scan is completed. You may see the results in scan_results_20230520054308.txt

┌──(root㉿kali)-[/home/kali]
└─# cat scan_results_20230520054308.txt
Scanning Time: 42.72355794906616 seconds

Host: 192.168.1.11
        Status: up
        Protocol: ['tcp']
        Port: 1947      State: filtered
        Port: 5000      State: open
        Port: 7000      State: open
```

Рисунок. 3.13. Результат сканування одного із хостів (IP-адреса замінена) у локальній мережі. Агресивний тип сканування.

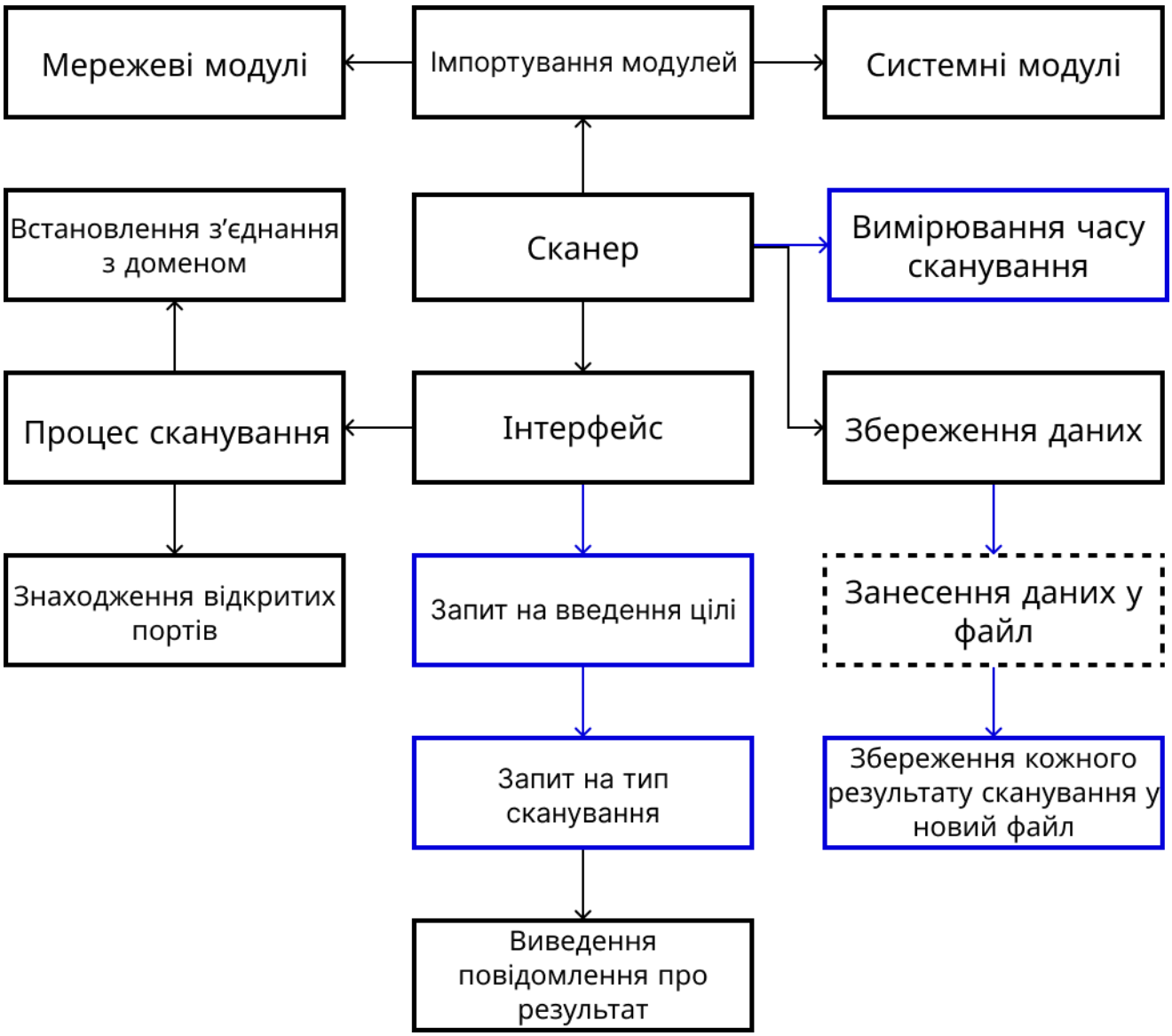

Рисунок. 3.14. Структурно-логічна схема сканеру з покращеннями.

Вище наведена структурно-логічна схема удосконаленої системи сканування мережі, яка була використана і результати якої можна побачити у рисунках 3.3 – 3.13. У даному випадку імпортується вже не тільки мережевий модуль, а і системні. У коді використовуються:

- модуль `random` для генерації випадкових даних і випадкових затримок у процесі сканування мережі;

- модуль `time`, який надає функціональність для роботи з часом, включаючи вимірювання часу та затримки;

- модуль `datetime`, який надає класи та функції для роботи з датою і часом, що можуть використовуватись для створення та форматування імен вихідних файлів.

Сканер удосконалюється за рахунок наступних опцій:

- введенню цілі сканування, шо поліпшує процес сканування, шляхом того що не потрібно кожного разу, якщо ціль звісно змінюється, змінювати її у коді;

- вибору типу сканування; прихованого або ж агресивного;

- збереження кожного нового результату сканування у окремий час із заначенням дати в імені: є своєрідним поліпшенням для зввітності;

- вимірюванням часу сканування і записом його у файл з результатами.

3.2 Оцінка ефективності системи автоматичного сканування мережі

Оцінка ефективності системи у порівнянні зі скануванням вручну за допомогою Nmap показує, система забезпечує ефективне та точне сканування мережі. Він автоматизує процес, забезпечує швидкість та точність, та дозволяє зручно обробляти результати сканування записуючи кожне наступне сканування в окремий файл.

Переваги використання системи, яка враховує можливу протидію засобам мережевого захисту відносно сканування вручну:

1. Автоматизація: вона забезпечує автоматизований процес сканування, що дозволяє легко повторювати сканування та виконувати його в автоматичному режимі. Це особливо корисно для великих мереж або сценаріїв, де потрібно виконати багато сканувань.

2. Швидкість та ефективність: система використовує певні параметри сканування, такі як «-sS», «-T4», «-p 1-1000», які можуть допомогти зменшити час сканування та зробити його більш ефективним. Це може бути корисно для швидкого виявлення вразливостей або перевірки безпеки мережі в обмежені терміни.

3. Можливий обхід засобів мережевого захисту: використання різних параметрів сканування, таких як «--randomize-hosts», «-D RND:5», дозволяє обійти деякі засоби мережевого захисту, які можуть намагатися виявити або заблокувати автоматичні сканування. Це дає можливість отримати більше інформації про мережу та вразливості, які можуть бути приховані від простого сканування вручну.

4. Скорочення часу та ресурсів: автоматизоване сканування за допомогою коду може значно скоротити час, потрібний для виконання сканування, порівняно з ручним використанням Nmap. Код може оптимізувати процес, уникнути повторення дій та забезпечити ефективніше використання ресурсів комп'ютера.

5. Консистентність та точність: використання однакових параметрів сканування у коді гарантує консистентність та точність результатів. Код може виконувати однакові сканування з однаковими параметрами без втрати даних або змін у вихідних результатах.

6. Автоматичне збереження результатів: дана програмна реалізація може автоматично зберігати результати кожного сканування у новий файл під назвою "scan_results_(дата).txt". Це спрощує збереження та аналіз результатів сканування, забезпечуючи легкий доступ до них у майбутньому.

7. Можливість інтеграції: цей код можна легко інтегрувати у інші програми або скрипти, що дозволяє використовувати його функціонал у ширшому контексті. Наприклад, його можна використовувати у скриптах автоматичного моніторингу мережі або безпекових аудитах.

Загалом, використання коду дозволяє зручно та ефективно виконувати сканування мережі, забезпечуючи автоматизацію, швидкість, можливість уникнення засобів мережевого захисту та збереження результатів.

Щодо недоліків використання коду:

Ефективність системи у порівнянні зі скануванням вручну за допомогою Nmap може бути обмежена декількома факторами:

1. Обмежені параметри: в коді використовуються певні параметри сканування, такі як `-sS` або `-A`, що можуть бути підходящими для багатьох випадків, але не враховують всі можливості, які надає сама програма nmap. Ручне використання nmap дозволяє більш гнучкий вибір параметрів для вирішення конкретних завдань сканування.

2. Відсутність динамічної адаптації: код виконує сканування з фіксованими параметрами, що не змінюються під час виконання. В деяких ситуаціях може бути

корисно мати можливість динамічно адаптувати параметри сканування в залежності від контексту чи відповідей від мережевих пристроїв.

3. Потенційні обмеження безпеки: при автоматичному скануванні з використанням коду може виникнути ризик спричинення блокування IP-адреси або виявлення засобами мережевого захисту як потенційно шкідливої діяльності. Ручне використання nmap дозволяє зберігати більший контроль над скануванням та уникати можливих проблем безпеки.

Таким чином, оцінка ефективності програми порівняно зі скануванням вручну за допомогою Nmap залежить від конкретного випадку, вимог до автоматизації та налаштувань мережевого середовища. У деяких ситуаціях ручне використання nmap може бути більш гнучким і ефективним, особливо коли потрібне точне налаштування параметрів сканування або коли безпека та уникнення виявлення є пріоритетом.

Висновок до третього розділу

У розділі були наведені програмна реалізація і структурно-логічні схеми систем автоматичного сканування мережі: найпростішої і вдосконаленої, яка враховує можливу протидію засобам мережевого захисту шляхом використання різних параметрів сканування. До таких параметрів передусім відносяться:

- --randomize-hosts;
- -D RND:5;
- -T4.

Загалом, використання коду дозволяє зручно та ефективно виконувати сканування мережі, забезпечуючи автоматизацію, швидкість, можливість уникнення засобів мережевого захисту та збереження результатів.

Описані ключові моменти коду, які також можуть допомогти потенційно допомогти уникнути бути поміченим цільовим хостом.

Було проаналізовано ефективності системи автоматичного сканування мережі і наведені переваги і недоліки її використання.

ВИСНОВКИ

Системи автоматичного сканування мережі є ефективним інструментом для виявлення вразливостей, слабких місць та потенційних загроз у мережевій інфраструктурі. Вони сприяють ідентифікації вразливостей у операційних системах, мережевих протоколах, програмному забезпеченні та конфігураціях, які можуть бути використані зловмисниками для незаконного доступу, атак або крадіжки даних.

У першому розділі було описано основні теоретичні визначення та терміни, на яких базується робота. Було проаналізовано які є методи виявлення хостів. Nmap пропонує різні налаштування для їх виявлення залежно від потреб користувачів. Головна мета Nmap - виявлення працюючих та відповідаючих хостів у мережі. Цю інформацію можна отримати з системи доменних імен. Багато організацій надають хостам імена, що розкривають їх функціональність. Враховуючи, що стандартні параметри зазвичай є ефективними, у даному розділі буди також розлянуті варіанти для керування правами доступу до DNS.

У другому розділі було розглянуто, які є порти: відомі, зарєстровані і динамічні/приватні, і топ-20 найчастіше відкритих портів TCP, UDP, і їх стани: open, closed, фільтрований filtered, unfiltered, open|filtered, і closed|filtered. Також у цьому розділі були проаналізовані методи сканування портів, які можуть бути використані для виявлення відкритих портів на цільовій системі. Декілька найпоширеніших із них, які можуть бути комбіновані та налаштовуватися залежно від потреб користувача та контексту використання:

- TCP SYN сканування;
- TCP Connect сканування;
- UDP сканування;
- NULL, FIN та XMAS сканування.

У третьому розділі було розглянуто структурно-логічні схеми найпростішого сканеру мережі і вдосконаленого, надана програмна реалізація автоматичної системи

сканування мережі, яка враховує можливу протидію засобів мережевого захисту та її оцінка у вигляді перваг і недоліків її використання.

Поставлені завдання були виконані у повному обсязі:

- проаналізовано методи виявлення хостів;
- проаналізувоно методи сканування портів;
- розроблено систему автоматичного сканування мережі;
- ефективність системи була оцінена.

Процес сканування мережі шляхом розробки системи автоматичного сканування з урахування можливої протидії засобів мережевого захисту було покращено.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nmap - інструмент дослідження мережі та сканер безпеки / портів [Електронний ресурс] / - Режим доступу до ресурсу: https://www.onworks.net/uk/programs/nmap-online.

2. Zhou C. «A Comprehensive Detection Approach of Nmap: Principles, Rules and Experiments.» // 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), - 2020. - С. 64.

3. DNS Resolution [Електронний ресурс] / - Режим доступу до ресурсу: https://nmap.org/book/host-discovery-dns.html.

4. Host Discovery Controls [Електронний ресурс] / - Режим доступу до ресурсу: https://nmap.org/book/host-discovery-controls.html.

5. Buchyk S., Saroka S. «Analysis of Host Detection Methods.» // Information Technology and Implementation (Satellite): Conference Proceedings, December 01, 2022, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor), - 2022. - С. 18-20.

6. Timing Templates [Електронний ресурс] / - Режим доступу до ресурсу: https://nmap.org/book/performance-timing-templates.html.

7. Messer, James. Secrets of Network Cartography: A Comprehensive Guide to Nmap. - 2nd ed. - 2007. - С. 192.

8. Host Discovery Strategies [Електронний ресурс] / - Режим доступу до ресурсу: https://nmap.org/book/host-discovery-strategies.html.

9. Сканування портів [Електронний ресурс] / - Режим доступу до ресурсу: https://help.eset.com/?lang=en-US.

10. What is port scanning? [Електронний ресурс] / - Режим доступу до ресурсу: https://www.avast.com/business/resources/what-is-port-scanning#mac.

11. What are the TCP/IP Well Known Port Numbers (0 to 1023) [Електронний ресурс] / - Режим доступу до ресурсу: https://www.meridianoutpost.com/resources/articles/well-known-tcpip-ports.php.

12. Eleventh Hour Linux+Exam XK0-003 Study Guide / Configuring the Base. – 2010. – P. 41-60.

13. The Official Nmap Project Guide to Network Discovery and Security Scanning [Електронний ресурс] / - Режим доступу до ресурсу: https://www.techtarget.com/searchnetworking/definition/dynamic-port-numbers.

14. Open Port Vulnerabilities List [Електронний ресурс] / - Режим доступу до ресурсу: https://blog.netwrix.com/2022/08/04/open-port-vulnerabilities-list/.

15. Commonly Open Ports [Електронний ресурс] / - Режим доступу до ресурсу: https://www.speedguide.net/ports_common.php.

16. Port Scanning Basics [Електронний ресурс] / - Режим доступу до ресурсу: https://www.uv.mx/personal/angelperez/files/2018/10/scanning_texto.pdf.

17. The Official Nmap Project Guide to Network Discovery and Security Scanning [Електронний ресурс] / - Режим доступу до ресурсу: https://nmap.org/book/port-scanning.html.

18. Preventing Malicious Hacks with Port Scanning Techniques [Електронний ресурс] / - Режим доступу до ресурсу: https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/preventing-malicious-hacks-with-port-scanning-techniques/.

19. Erikson, Jon. HACKING the art of exploitation. - 2nd ed. - San Francisco: NoStarch Press, 1977. - С. 264.

20. Messer, James. Secrets of Network Cartography: A Comprehensive Guide to Nmap. - 2nd ed. - 2007. - С. 43-46.

21. Messer, James. Secrets of Network Cartography: A Comprehensive Guide to Nmap. - 2nd ed. - 2007. - С. 29-33.

22. Messer, James. Secrets of Network Cartography: A Comprehensive Guide to Nmap. - 2nd ed. - 2007. - С. 33-40.

23. "Understanding the NMAP methodology — Part 2" [Електронний ресурс] / - Режим доступу до ресурсу: https://infosecwriteups.com/understanding-the-nmap-methodology-part-2-3d0442f1c482.

24. SYN.FIN.Scan [Електронний ресурс] / - Режим доступу до ресурсу: https://www.fortiguard.com/encyclopedia/ips/13040.

25. Messer, James. Secrets of Network Cartography: A Comprehensive Guide to Nmap. - 2nd ed. - 2007. - С. 49-52.

26. TCP Window Scan [Електронний ресурс] / - Режим доступу до ресурсу: https://nmap.org/book/scan-methods-window-scan.html.

27. What is an Nmap Maimon scan? [Електронний ресурс] / - Режим доступу до ресурсу: https://www.techtarget.com/searchsecurity/answer/What-is-an-Nmap-Maimon-scan.

28. Erikson, Jon. HACKING the art of exploitation. - 2nd ed. - San Francisco: NoStarch Press, 1977. - С. 265.

29. IP Protocol Scan [Електронний ресурс] / - Режим доступу до ресурсу: https://nmap.org/book/scan-methods-ip-protocol-scan.html.

30. TCP FTP Bounce Scan [Електронний ресурс] / - Режим доступу до ресурсу: https://nmap.org/book/scan-methods-ftp-bounce-scan.html.

31. Using the Nmap Port Scanner with Python [Електронний ресурс] / - Режим доступу до ресурсу: https://www.studytonight.com/network-programming-in-python/integrating-port-scanner-with-nmap.

32. TCP SYN (Stealth) Scan [Електронний ресурс] / - Режим доступу до ресурсу: https://nmap.org/book/synscan.html.

33. Aggressive Scan [Електронний ресурс] / - Режим доступу до ресурсу: https://www.codecademy.com/resources/docs/cybersecurity/nmap/aggressive-scan.

ДОДАТОК А

```
import nmap
# initialize the port scanner
nmScan = nmap.PortScanner()
# scan localhost for ports in range 21-443
nmScan.scan('127.0.0.1', '21-443')
# run a loop to print all the found result about the ports
for host in nmScan.all_hosts():
    print('Host : %s (%s)' % (host, nmScan[host].hostname()))
    print('State : %s' % nmScan[host].state())
    for proto in nmScan[host].all_protocols():
        print('----------')
        print('Protocol : %s' % proto)
        lport = nmScan[host][proto].keys()
        lport.sort()
        for port in lport:
            print ('port : %s\tstate : %s' % (port, nmScan[host][proto][port]['state']))[30]
```

ДОДАТОК Б

```
import nmap
import random
import time
import datetime

def scan_network(target, scan_type):
    nm = nmap.PortScanner()

    if scan_type == 0:
        # Set Nmap options for stealth scanning
        scan_arguments = '-sS -T4 -p 1-1000 --randomize-hosts -D RND:5'
    elif scan_type == 1:
        # Set Nmap options for aggressive scanning
        scan_arguments = '-A -T5'
    else:
        print("Invalid scan type. Please choose either 0 (stealth) or 1 (aggressive).")
        return

    # Additional evasion techniques
    random.seed()
    random_delay = random.uniform(0.5, 2.0)
    scan_delay = random_delay if scan_type == 0 else 0

    # Randomize delay between scans
    time.sleep(scan_delay)

    start_time = time.time()  # Start the timer
```

```
        nm.scan(hosts=target, arguments=scan_arguments)
        end_time = time.time()  # Stop the timer

        scanning_time = end_time - start_time

        timestamp = datetime.datetime.now().strftime("%Y%m%d%H%M%S")
        output_file = f"scan_results_{timestamp}.txt"

        with open(output_file, 'w') as file:
            file.write(f"Scanning Time: {scanning_time} seconds\n\n")
            for host in nm.all_hosts():
                if nm[host].state() == 'up':
                    file.write(f"Host: {host}\n")
                    file.write(f"\tStatus: {nm[host].state()}\n")
                    file.write(f"\tProtocol: {nm[host].all_protocols()}\n")

                    for protocol in nm[host].all_protocols():
                        ports = nm[host][protocol].keys()
                        for port in ports:
                            state = nm[host][protocol][port]['state']
                            file.write(f"\tPort: {port}\tState: {state}\n")
                else:
                    file.write(f"Host: {host}\tStatus: {nm[host].state()}\n")
        print(f"The scan is completed. You may see the results in {output_file}")

    # Example usage
    target_network = input("Enter the target network to scan (e.g., 192.168.1.0/24 or scanme.nmap.org): ")
    scan_type = int(input("Enter the scan type (0 for stealth, 1 for aggressive): "))
    scan_network(target_network, scan_type)
```

ДОДАТОК В

Тези наукових публікацій:

1. Buchyk S., Saroka S. Analysis of Host Detection Methods. Information Technology and Implementation (Satellite): Conference Proceedings, December 01, 2022, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor). - Kyiv: Publisher Individual entrepreneur Picha Y.V., 2022. pp. 18-20.
2. Сергій Бучик, Світлана Сарока. Аналіз Методів Сканування Портів Використоввуючи NMAP. Проблеми Кібербезпеки Інформаційно-Телнкомунікаційних Систем (PCSITS). VI Міжнародна Науково-Практична Конференція, 2023. с. 126-127.