Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи оцінки захищеності комп'ютерної мережі при тестуванні на проникнення

(тема)

Виконав:
студент II курсу, групи КСМзм-19-1
Рязанін С.Г.
(прізвище, ініціали)

Спеціальність
123 – Комп’ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп’ютерні системи та мережі
(повна назва освітньої програми)

Керівник: доц. Голубничий Д.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп’ютерна інженерія
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20___ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Рязаніну Сергію Григоровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Методи оцінки захищеності комп'ютерної мережі при тестуванні на проникнення

затверджена наказом по університету від “ 23 ” жовтня 2020 р. № 168 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 14 грудня 2020 р.

3. Вхідні дані до роботи 1) Метод тестування на проникнення: NIST SP 800-115; 2) Метод тестування на проникнення: Open Source Security Testing Methodology; 3) Метод тестування на проникнення: Information Systems Security Assessment Framework

4. Перелік питань, що потрібно опрацювати в роботі

1) огляд захищеності комп’ютерної мережі як об’єкта кібербезпеки
2) аналіз вразливостей інформаційних систем та оцінка їх критичності для комп’ютерних мереж
3) проведення експериментальних досліджень методів оцінки захищеності комп'ютерної мережі

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 17 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз сучасних підходів до проведення аудиту	27.10.20 – 09.11.20	
	кібербезпеки		
2	Вибір та обгрунтування методики дослідження	10.11.20 – 17.11.20	
3	Вибір інструментальних засобів	18.11.20 – 23.11.20	
4	Проведення експериментів	24.11.20 – 01.12.20	
5	Оформлення матеріалів атестаційної роботи	02.12.20 – 07.12.20	
6	Подання атестаційної роботи керівникові та	08.12.20 – 09.12.20	
	її попередній захист		
7	Подання атестаційної роботи на рецензування	10.12.20 – 11.12.20	

Дата видачі завдання 26 жовтня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Голубничий Д.Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 108 с., 35 рис., 7 табл., 1 дод., 28 джерел.

КОМП’ЮТЕРНА МЕРЕЖА, КІБЕРБЕЗПЕКА, КІБЕРЗАГРОЗА, ВРАЗЛИВІСТЬ, ЕКСПЛОЙТ, ТЕСТ НА ПРОНИКНЕННЯ.

Метою роботи є підвищення рівня захищеності комп’ютерної мережі за рахунок використання послуги тестування на проникнення, яка дозволяє здійснити санкціонований обхід існуючого комплексу засобів захисту власних інформаційних систем та виявити в них слабкі місця.

Об’єктом дослідження є процеси забезпечення захисту від вразливостей інформаційних систем в комп’ютерній мережі.

Предметом дослідження є методи оцінки ризику та визначення рівнів загроз інформаційній безпеці, які дозволять протидіяти кіберзагрозам на комп'ютерну мережу.

Актуальність роботи полягає в тому, що тестування на проникнення дозволяє отримати об'єктивну оцінку того, наскільки легко отримати доступ до ресурсів комп’ютерної мережі, яким способом і через які вразливості. Тестування здійснюється шляхом ідентифікації максимально можливої кількості вразливостей комп’ютерної мережі за обмежений час при заданих умовах й поточному стані та переконання ефективності виявлених слабких місць. Тестування на проникнення проводять для здобуття незалежної оцінки захищеності комп’ютерної мережі.

У роботі проведено аналіз методів тестування на проникнення та запропонований метод оцінки захищеності комп'ютерної мережі за рахунок використання експлойтів.

ABSTRACT

Master’s thesis: 108 pages, 35 figures, 7 tables, 1 appendices, 28 sources.

COMPUTER NETWORK, CYBER SECURITY, CYBER THREAT, VULNERABILITY, EXPLOIT, PENETRATION TEST.

The major goal of this thesis is to increase the level of security of the computer network through the use of penetration testing service, which allows for an authorized bypass of the existing set of means to protect their own information systems and identify vulnerabilities in them.

The object of the study is the processes of providing protection against vulnerabilities of information systems in a computer network.

The subject of the study are methods of risk assessment and determination of levels of information security threats, which will counteract cyber threats to the computer network.

The relevance of the work is that penetration testing allows to obtain an objective assessment of how easy it is to access computer network resources, in what way and through what vulnerabilities. Testing is performed by identifying the maximum possible number of computer network vulnerabilities in a limited time under the given conditions and current state and convincing the effectiveness of the identified vulnerabilities. Penetration testing is performed to obtain an independent assessment of the security of a computer network.

The work is carried out analysis of penetration testing methods and the proposed method of assessing the security of a computer network through the use of exploits.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 8
ВСТУП 9
1 АНАЛІЗ ЗАХИЩЕНОСТІ КОМП'ЮТЕРНОЇ МЕРЕЖІ ЯК ОБ'ЄКТА КІБЕРБЕЗПЕКИ 12
1.1 Технології аудиту кібербезпеки інформаційних систем 12
1.2 Класифікація погроз кібербезпеки 20
1.3 Визначення фаз проведення аудиту кібербезпеки 23
1.4 Категорії порушників кібербезпеки 26
1.5 Життєвий цикл тестування інформаційної системи на проникнення 34
1.6 Висновки за розділом 1 39
2 АНАЛІЗ ВРАЗЛИВОСТЕЙ ІНФОРМАЦІЙНИХ СИСТЕМ ТА ОЦІНКА ЇХ КРИТИЧНОСТІ ДЛЯ КОМП'ЮТЕРНИХ МЕРЕЖ 41
2.1 Класифікація атак на комп'ютерні мережі 41
2.2 Характеристика вразливостей інформаційних систем в комп'ютерних мережах 47
2.3 Методика оцінки критичності знайдених вразливостей 54
2.3.1 Підсистема збору інформації вразливостей 59
2.3.2 Підсистема формування БД експлойтів 63
2.3.3 Підсистема формування БД експлойтів 64
2.3.4 Підсистема аналізу захищеності та Metasploit Framework 64
2.4 Методика оцінки захищеності комп'ютерної мережі за рахунок використання експлойтів при тестуванні на проникнення 68
2.4.1 Підсистема формування бази даних експлойтів 72
2.4.2 Підсистема збору інформації 75
2.4.3 Підсистема аналізу захищеності 77

2.5 Тестування на проникнення до хостів комп'ютерної мережі 79
2.6 Висновки за розділом 2 80
3 МЕТОДИ ТЕСТУВАННЯ КОМП'ЮТЕРНИХ МЕРЕЖ НА ПРОНИКНЕННЯ..... 81
3.1 Підходи до проведення тестування на проникнення 81
3.2 Метод Open Source Security Testing Methodology..... 82
3.3 Метод Information Systems Security Assessment Framework 83
3.4 Метод NIST Special Publications 800-115..... 86
3.5 Метод тестування комп'ютерної мережі на проникнення 88
3.5 Висновки за розділом 3 94
ВИСНОВКИ..... 95
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 96
ДОДАТОК А Графічний матеріал атестаційної роботи 99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІБ – інформаційна безпека

ІР – інформаційний ресурс

ІС – інформаційна система

ІТ – інформаційних технологій

КМ – комп’ютерна мережа

МЕ – міжмережевий екран

ОС – операційна система

ПАЗ – підсистема аналізу захищеності

ПЗ – програмне забезпечення

ПФБДЕ – підсистема формування БД експлойтів

СВВ – система виявлення вторгнення

СЗІ – система захисту інформації

СУБД – система управління базами даних

CVE – загальні вразливості та ризики (англ., Common Vulnerabilities and Exposures)

CVSS – загальна система оцінки вразливостей (англ., Common Vulnerability Scoring System)

DSN – повідомлення про стан доставки (англ., Delivery Status Notification)

ISSAF – рамка оцінки безпеки інформаційних систем (англ., Information Systems Security Assessment Framework)

OSSTMM – методологія тестування безпеки з відкритим кодом (англ., Open Source Security Testing Methodology)

PCI DSS – стандарт безпеки даних платіжних карток (англ., Payment Card Industry Data Security Standard)

ВСТУП

Захищеність є одним з найважливіших показників ефективності функціонування інформаційної системи (ІС). Під захищеністю ІС будемо розуміти ступінь адекватності реалізованих в ній механізмів захисту інформації існуючим в даному середовищі функціонування ризикам, пов'язаним із здійсненням загроз безпеки інформації.

Під загрозами безпеці інформації традиційно розуміється можливість порушення таких властивостей інформації як конфіденційність, цілісність і доступність [5]. Арсенал програмних засобів, які використовують для аналізу захищеності ІС є досить широким.

Тест на проникнення дозволяє отримати об'єктивну оцінку того, наскільки легко отримати доступ до ресурсів комп’ютерної мережі (КМ), яким способом і через які вразливості. Тест на проникнення є моделюванням дій зловмисника по проникненню в ІС і дозволяє виявити найбільше вразливостей в захисті мережі. Тест на проникнення проводять для здобуття незалежної оцінки захищеності комп’ютерної мережі.

Федеральне бюро розслідувань США запустило у 2001 році програму попередження комп'ютерних злочинів InfraGuard, розроблену Центром захисту національної інфраструктури [25]. Однією з цілей програми стало створення захищеної від вторгнення комп’ютерної мережі для обміну інформацією між компаніями та органами забезпечення правопорядку про здійснені атаки та надання відомостей, які можуть попередить такі зазіхання.

Метою тесту на проникнення є виявлення слабких місць в захисті ІС і, якщо це можливо, і відповідає бажанню замовника, здійснити показовий злом.

Основне завдання тесту на проникнення – ідентифікація максимально можливого числа вразливостей інформаційної системи за обмежений час при певних умовах і стані ІС.

При проведенні тесту на проникнення вирішуються завдання:

- оцінка поточного стану системи захисту інформації ІС;
- виявлення вразливостей інформаційної системи;
- використання виявлених вразливостей для отримання несанкціонованого доступу чи здійснення несанкціонованого впливу на інформацію для демонстрації наявності вразливостей і існування високо ймовірної загрози інформаційної системи;
- вироблення рекомендацій щодо підвищення ефективності захисту інформації в ІС.

Таким чином, актуальність теми роботи полягає в виявленні та визначенні характеру вразливості, враховуючи, як вони можуть бути використані для злочину. В той же час тестування на проникнення використовує експерименти та оцінює тип і ступінь ризику вразливості у представленій ІС.

Спільною рисою є розробка знань та вмінь, заснованих на тих же технологіях, вони є не менш важливі як для управління ризиками, так і для військових операцій в кіберпросторі і їх використання для оборонної та розвідувальної діяльності дуже ефективне. Значущість та вплив інформаційних технологій у всій сучасній організаційно – технічній системі очевидна і постійно зростає.

Тому, метою атестаційної роботи є підвищення рівня захищеності комп’ютерної мережі за рахунок використання послуги тестування на проникнення, яка дозволяє здійснити санкціонований обхід існуючого комплексу засобів захисту власних інформаційних систем та виявити в них слабкі місця.

Об’єктом дослідження є процеси забезпечення захисту від вразливостей інформаційних систем в комп’ютерній мережі.

Предметом дослідження є методи оцінки ризику та визначення рівнів загроз інформаційній безпеці, які дозволять протидіяти кіберзагрозам на комп'ютерну мережу.

Для досягнення цієї мети потрібно вирішити наступні науково-технічні задачі:

- проведення аналізу захищеності комп’ютерної мережі як об’єкта кібербезпеки;
- проведення аналізу вразливостей інформаційних систем та оцінка їх критичності для комп’ютерних мереж;
- побудова методу оцінки захищеності комп'ютерної мережі за рахунок використання експлойтів при тестуванні на проникнення.

1 АНАЛІЗ ЗАХИЩЕНОСТІ КОМП'ЮТЕРНОЇ МЕРЕЖІ ЯК ОБ'ЄКТА КІБЕРБЕЗПЕКИ

1.1 Технології аудиту кібербезпеки інформаційних систем

Інформаційна система (ІС) – система, що реалізує автоматизований збір, обробку та маніпулювання даними, і включає: технічні засоби, програмне забезпечення, відповідний персонал і допоміжні засоби (рисунок 1.1).

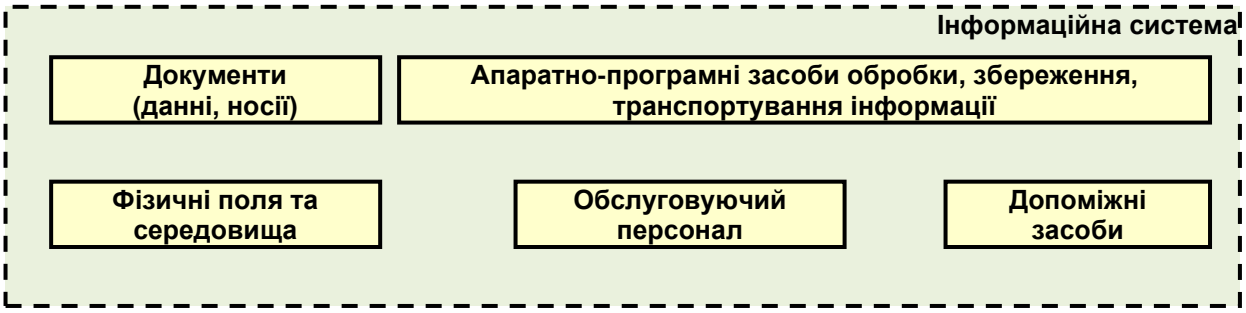


Рисунок 1.1 – Складові інформаційної системи

Кібербезпека ІС не є даністю, а створюється шляхом побудови системи захисту інформації (СЗІ) в ІС. Відповідно до документів і загальноприйнятою практикою можна виділити наступні етапи побудови СЗІ:

- визначення інформаційних ресурсів (ІР), які підлягають захисту;
- виявлення повної множини загроз кібербезпеки ІР, які підлягають захисту;
- проведення оцінки вразливості і ризиків для ІР, які підлягають захисту, при виявленій множині загроз;
- розробка проекту (плану) системи захисту інформації, що знижує за обраним критерієм ризики для ІР, які підлягають захисту, при виявленій множині загроз;

- реалізація проекту (плану) захисту інформації;
- визначення якості реалізованої системи захисту;
- здійснення контролю функціонування і управління системою захисту.

Проходження етапів необхідно в тій чи іншій мірі здійснювати безперервно і по замкнутому циклу, з проведенням відповідного аналізу стану СЗІ та уточненням вимог до неї після кожного кроку (рисунок 1.2).

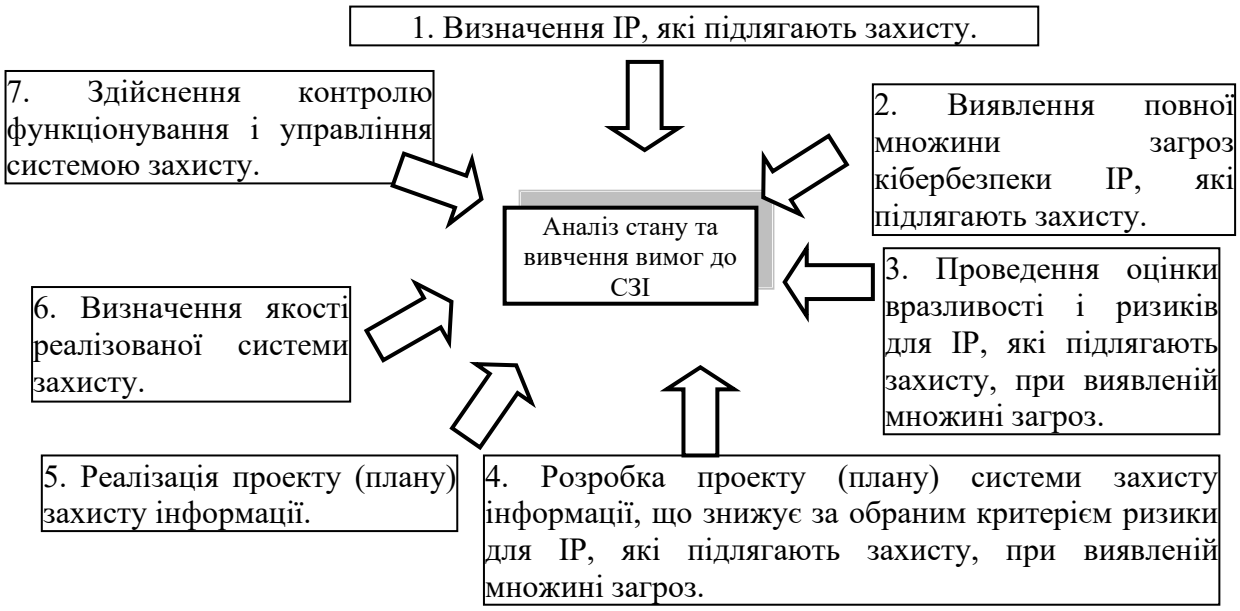


Рисунок 1.2 – Етапи побудови системи захисту інформації

Аудит кібербезпеки при створенні СЗІ доцільно здійснювати:

- на третьому етапі при оцінки вразливості ІР в складі ІС;
- на шостому етапі при визначенні якості реалізованої системи захисту;
- на сьомому етапі періодично при здійсненні контролю функціонування СЗІ.

Аудит являє собою незалежну експертизу окремих областей функціонування організації. Розрізняють зовнішній й внутрішній аудит.

Зовнішній аудит кібербезпеки ІС – це зовнішнє захід щодо ІС, підключеної до глобальної мережі Інтернет, з метою оцінки можливості подолання СЗІ ІС з боку зовнішнього зловмисника. Зовнішній аудит рекомендується проводити періодично.

Внутрішній аудит являє собою безперервну діяльність, яка здійснюється на підставі документа, зазвичай носить назву "Положення про внутрішній аудит", і відповідно до плану, підготовка якого здійснюється підрозділом внутрішнього аудиту та затверджується керівництвом організації.

Аудит кібербезпеки ІС є однією зі складових ІТ-аудита. Цілями проведення аудиту кібербезпеки є:

- аналіз ризиків, пов'язаних з можливістю здійснення загроз кібербезпеки щодо ресурсів ІС;
- оцінка поточного рівня захищеності ІС;
- локалізація вузьких місць в системі захисту ІС;
- оцінка відповідності ІС існуючим стандартам в області кібербезпеки;
- вироблення рекомендацій щодо впровадження нових та підвищення ефективності існуючих механізмів кібербезпеки ІС.

У число додаткових завдань, що стоять перед внутрішнім аудитором, крім надання допомоги зовнішнім аудиторам, можуть також входити:

- розробка політик кібербезпеки та інших організаційно-розпорядчих документів щодо захисту інформації та участь в їх впровадженні в роботу організації;
- постановка завдань для ІТ-персоналу, що стосуються забезпечення захисту інформації;
- участь в навчанні користувачів і обслуговуючого персоналу ІС питань забезпечення кібербезпеки;
- участь в розборі інцидентів, пов'язаних з порушенням кібербезпеки;
- інші завдання.

Необхідно відзначити, що всі перераховані "додаткові" завдання, які стоять перед внутрішнім аудитором, за винятком участі в навчанні, по суті аудитом не є. Аудитор за визначенням повинен здійснювати незалежну експертизу реалізації механізмів кібербезпеки в організації, що є одним з основних принципів аудиторської діяльності. Якщо аудитор бере діяльну участь в реалізації механізмів кібербезпеки, то незалежність аудитора втрачається, а разом з нею втрачається і об'єктивність його суджень, так як аудитор не може здійснювати незалежний і об'єктивний контроль своєї власної діяльності. Однак на практиці внутрішній аудитор, часом будучи найбільш компетентним фахівцем в організації в питаннях забезпечення кібербезпеки, не може залишатися осторонь від реалізації механізмів захисту.

Етап збору інформації аудита, є найбільш складним і тривалим. Це пов'язане з відсутністю необхідної документації на ІС і з необхідністю щільної взаємодії аудитора з багатьма посадовими особами організації [7].

Компетентні висновки щодо стану справ в організації з інформаційною кібербезпекою можуть бути зроблені аудитором тільки за умови наявності всіх необхідних вихідних даних для аналізу. Отримання інформації про організацію, функціонування і поточний стан ІС здійснюється аудитором в ході спеціально організованих інтерв'ю з відповідальними особами організації, шляхом вивчення технічної і організаційно-розпорядчої документації, а також дослідження ІС з використанням спеціалізованого програмного інструментарію. Зупинимося на тому, яка інформація необхідна аудитору для аналізу.

Перший пункт аудиторського обстеження починається з отримання інформації про організаційну структуру користувачів ІС і обслуговуючих підрозділів. Зазвичай в ході інтерв'ю аудитор задає опитуваним наступні питання: хто є власником інформації, хто є споживачем інформації, хто є постачальником послуг. Призначення і принципи функціонування ІС багато в чому визначають існуючі ризики і вимоги кібербезпеки, що пред'являються до системи. Тому на наступному етапі аудитора цікавить інформація про

призначення та функціонуванні ІС. Далі, аудитору потрібно більш детальна інформація про структуру ІС. Це дозволить усвідомити, яким чином здійснюється розподіл механізмів кібербезпеки за структурними елементами і рівнями функціонування ІС.

Підготовка значної частини документації на ІС зазвичай здійснюється вже в процесі проведення аудиту. Коли всі необхідні дані по ІС, включаючи документацію, підготовлені, можна переходити до їх аналізу.

Використовувані аудиторами методи аналізу даних визначаються вибраними підходами до проведення аудиту, які можуть істотно різнитися.

Перша технологія, сама складна, та базується на аналізі ризиків. Опираючись на методи аналізу ризиків, аудитор визначає для обстежуваної ІС індивідуальний набір вимог кібербезпеки, найбільшою мірою враховуючої особливості даної ІС, середовища її функціонування й існуючі в даному середовищі погрози кібербезпеки. Дана технологія є найбільш трудомістким і вимагає найвищої кваліфікації аудитора. На якість результатів аудита, у цьому випадку, сильно впливає використовувана методологія аналізу й управління ризиками і її застосовність до даного типу ІС [6].

Друга технологія, сама практична, та опирається на використання стандартів кібербезпеки. Стандарти визначають базовий набір вимог кібербезпеки для широкого класу ІС, що формується в результаті узагальнення світової практики. Стандарти можуть визначати різні набори вимог кібербезпеки, залежно від рівня захищеності ІС, що потрібно забезпечити, її приналежності (комерційна організація, або державна установа), а також призначення (фінанси, промисловості, зв'язок і т.п.). Від аудитора в цьому випадку потрібно правильно визначити набір вимог стандарту, відповідність яким потрібно забезпечити для даної ІС. Необхідна також методика, що дозволяє оцінити цю відповідність. Через свою простоту (стандартний набір вимог для проведення аудита вже заздалегідь визначений стандартом) і надійності (вимоги стандарту ніхто не спробує заперечити), описаний підхід найпоширеніший на практиці (особливо при проведенні

зовнішнього аудита). Він дозволяє при мінімальних витратах ресурсів робити обґрунтовані висновки про стан ІС [6].

Третя технологія, найбільш ефективна, та припускає комбінування перших двох. Базовий набір вимог кібербезпеки, пропонованих до ІС, визначається стандартом. Додаткові вимоги, у максимальному ступені враховуючі особливості функціонування даної ІС, формуються на основі аналізу ризиків. Цей підхід є набагато простіше першого, тому що більша частина вимог кібербезпеки вже визначена стандартом, і, у той же час, він позбавлений недоліку другого підходу, що містить у тім, що вимоги стандарту можуть не враховувати специфіки обстежуваної ІС [6].

Аудиторський звіт є основним результатом проведення аудиту. Його якість характеризує якість роботи аудитора.

Структура звіту може суттєво відрізнятися в залежності від характеру і цілей проведеного аудиту. Однак певні розділи повинні обов'язково бути присутнім в аудиторському звіті.

Він повинен, принаймні, містити опис цілей проведення аудиту, характеристику досліджуваної ІС, вказівку кордонів проведення аудиту і використовуваних методів, результати аналізу даних аудиту, висновки, узагальнюючі ці результати і містять оцінку рівня захищеності ІС або відповідність її вимогам стандартів, і, звичайно, рекомендації аудитора щодо усунення існуючих недоліків та вдосконалення системи захисту.

Аналіз ризиків – це те, із чого повинне починатися побудову будь-якої системи кібербезпеки. Він містить у собі заходу щодо обстеження кібербезпеки ІС, з метою визначення того які ресурси й від яких погроз треба захищати, а також у якому ступені ті або інші ресурси мають потребу в захисті. Визначення набору адекватних контрзаходів здійснюється в ході управління ризиками. Ризик визначається ймовірністю заподіяння збитку й величиною збитку, що наноситься ресурсам ІС, у випадку здійснення погрози кібербезпеки.

Аналіз ризиків полягає в тому, щоб виявити існуючі ризики й оцінити їхню величину (дати їм якісну, або кількісну оцінку). Процес аналізу ризиків ділиться на кілька послідовних етапів [8]:

- ідентифікація ключових ресурсів ІС;
- визначення важливості тих або інших ресурсів для організації;
- ідентифікація існуючих погроз кібербезпеки й вразливостей, що роблять можливим здійснення погроз;
- обчислення ризиків, пов'язаних зі здійсненням погроз кібербезпеки.
- Ресурси ІС можна розділити на наступні категорії:
- інформаційні ресурси;
- програмне забезпечення;
- технічні засоби (сервери, робочі станції, мережне обладнання тощо);
- людські ресурси.

У кожній категорії ресурси діляться на класи й підкласи. Необхідно ідентифікувати тільки ті ресурси, які визначають функціональність ІС і істотні з погляду забезпечення кібербезпеки [6].

Важливість (або вартість) ресурсу визначається величиною збитку, який наноситься у випадку порушення конфіденційності, цілісності або доступності цього ресурсу. Звичайно розглядаються наступні види збитку [6]:

- дані були розкриті, змінені, вилучені або стали недоступні;
- апаратури була ушкоджена або зруйнована;
- порушено цілісність програмного забезпечення.

Збиток може бути нанесений організації в результаті успішного здійснення наступних видів погроз кібербезпеки [7]:

- локальні й вилучені атаки на ресурси ІС;
- стихійні лиха;
- помилки, або навмисні дії персоналу ІС;
- збої в роботі ІС, викликані помилками в програмному забезпеченні або несправностями апаратури.

Під вразливостями звичайно розуміють властивості ІС, що роблять можливим успішне здійснення погроз кібербезпеки [7].

Величина ризику визначається на основі вартості ресурсу, імовірності здійснення погрози й величини вразливості по наступній формулі [6]:

$$\text{Ризик} = \frac{\text{вартість\_ресурсу} \cdot \text{ймовірність\_погрози}}{\text{величина\_вразливості}} . \qquad (1.1)$$

Завдання управління ризиками полягає у виборі обґрунтованого набору контрзаходів, що дозволяють знизити рівні ризиків до прийнятної величини. Вартість реалізації контрзаходів повинна бути менше величини можливого збитку. Різниця між вартістю реалізації контрзаходів і величиною можливого збитку повинна бути обернено пропорційна ймовірності заподіяння збитку.

Якщо для проведення аудита кібербезпеки обраний підхід, що базується на аналізі ризиків, то на етапі аналізу даних аудита звичайно виконуються наступні групи завдань [8]:

- аналіз ресурсів ІС, включаючи інформаційні ресурси, програмні й технічні засоби, а також людські ресурси;
- аналіз груп завдань, розв'язуваних системою, і бізнес процесів;
- побудова (неформальної) моделі ресурсів ІС, що визначає взаємозв'язку між інформаційними, програмними, технічними й людськими ресурсами, їхнє взаємне розташування й способи взаємодії;
- оцінка критичності інформаційних ресурсів, а також програмних і технічних засобів;
- визначення критичності ресурсів з обліком їх взаємозалежності;
- визначення найбільш імовірних погроз кібербезпеки відносно ресурсів ІС і вразливостей захисту, що роблять можливим здійснення цих погроз;
- оцінка ймовірності здійснення погроз, величини вразливостей і збитку, що наноситься організації у випадку успішного здійснення погроз;

- визначення величини ризиків для кожної трійки: погроза – група ресурсів – вразливість.

Перерахований набір завдань, є досить загальним. Для їхнього рішення можуть використовуватися різні формальні й неформальні, кількісні і якісні, ручні й автоматизовані методики аналізу ризиків. Суть підходу від цього не міняється.

Оцінка ризиків може даватися з використанням різних як якісних, так і кількісних шкал. Головне, щоб існуючі ризики були правильно ідентифіковані та проранжовані у відповідності зі ступенем їхньої критичності для організації. На основі такого аналізу може бути розроблена система першочергових заходів щодо зменшення величини ризиків до прийнятного рівня [6].

1.2 Класифікація погроз кібербезпеки

Погроза – сукупність умов і факторів, що визначають потенційну або реально існуючу небезпеку виникнення інциденту. Це може привести до завдання збитків функціонуванню ІС, які захищають об'єкти [3].

Погрози, по способу їхнього створення, можна умовно розділити на наступні три типи: природними, технічними, створеними людьми [4].

Природні погрози:

- стихійні лиха (опади, повені, пожежі й т п.);
- магнітні бурі - можуть привести до серйозних збоїв електронного обладнання;
- радіоактивне випромінювання – виводить із ладу напівпровідникові прилади і як наслідок електронного обладнання.

Природні погрози приводять до відмови в обслуговуванні за рахунок фізичного руйнування обладнання, втрати даних через збої системи внаслідок зазначених погроз.

Технічні погрози:

- відключення або коливання електроживлення й інших засобів забезпечення - може привести до простого відключення або виходу з ладу обладнання, як наслідок відмова в обслуговувані, втрата або перекручування даних;

- відмови й збої апаратно-програмних засобів - приводять до відмови в обслуговувані і перекручуванню й втраті даних, порушенню ліній зв'язку;

- електромагнітні випромінювання і наведення - приводять до перекручування, втраті даних;

- витоки через канали зв'язку (оптичні, електричні, акустичні) - можуть привести до несанкціонованого перехоплення, зняттю інформації.

Зовнішній порушник особа не є співробітником компанії і виробляючої дії, що порушують політику кібербезпеки компанії [4].

Внутрішній порушник, це особа, яка є співробітником компанії і створює дії, що порушують політику кібербезпеки компанії, співробітник може, як бути так і не бути користувачем системи [4].

Не навмисні дії – дії викликані цікавістю, через незнання, у тому числі політики кібербезпеки, або внаслідок помилок у роботі, без наміру заподіяти який-небудь збиток [4]:

- обслуговуючий персонал – може випадково: відключити електроживлення, висмикнути patch-корд або інший мережний кабель;

- управлінський персонал (адміністратори системи) - можливі помилки адміністрування, необережні дії, що призводять збій у системі;

- програмісти-розроблювачі – помилки при написанні програм, що приводять до переповнення буфери, виконанню довільного коду, несанкціонованому доступу та іншим вразливостям;

- користувачі системи – можуть запустити незнайому програму, що виявиться шкідливою.

Навмисні дії – усвідомлені дії, що переслідують певні незаконні цілі:

- обслуговуючий персонал – може навмисно вивести з ладу обладнання, відключити електроживлення тощо;

- управлінський персонал (адміністратори системи) – можуть скопіювати, змінити, ушкодити, знищити інформацію, або порушити працездатність засобів, де вона зберігається або обробляється, тому що має усі права доступу, може спеціально наділити якого-небудь користувача зайвими правами, тощо;

- програмісти-розроблювачі – можуть при створенні програми навмисно створити в ній "чорні входи", "троянські" і інші шкідливі функції;

- користувачі системи - можуть намагатися здійснити й здійснювати несанкціонований доступ до секретної інформації, і інформації інших користувачів, навмисно запускати шкідливі програми й навмисно порушувати нормальний режим роботи системи.

Описану вище класифікацію погроз можна наочно представити у вигляді схеми, вона наведена на рисунок 1.3.

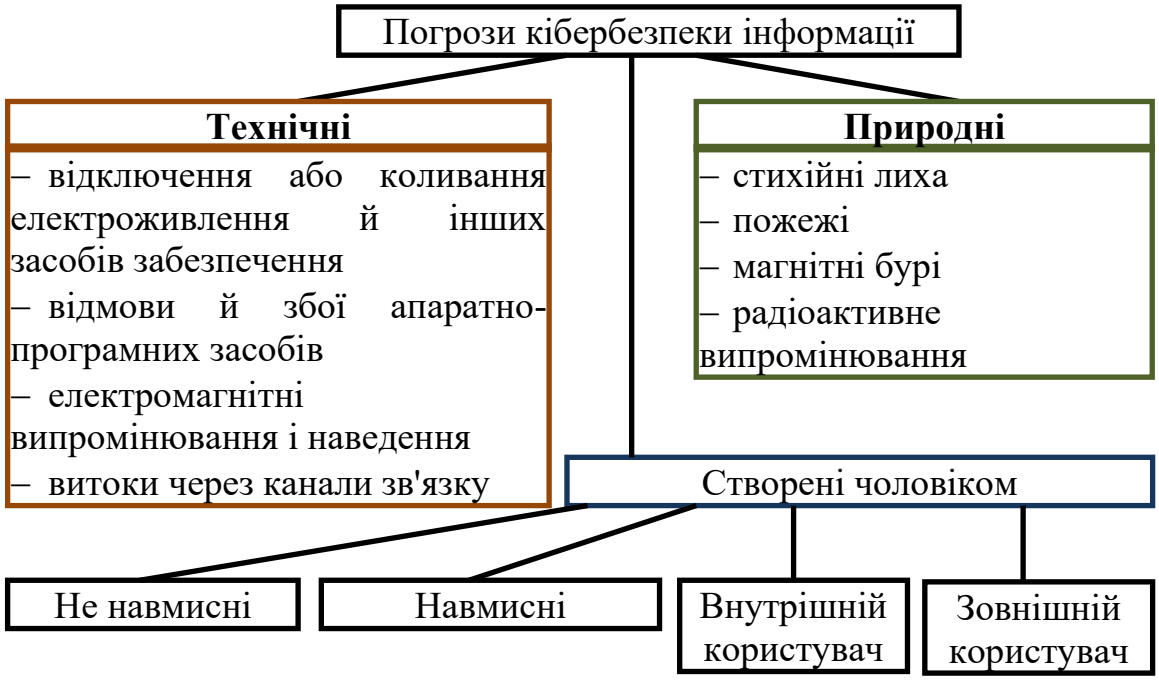


Рисунок 1.3 – Рівні погроз кібербезпеки інформаційної системи

Розглянуті погрози можуть бути реалізовані на рівнях інформаційної системи, виділимо чотири рівні [5], розподіл представлений на рисунок 1.4.

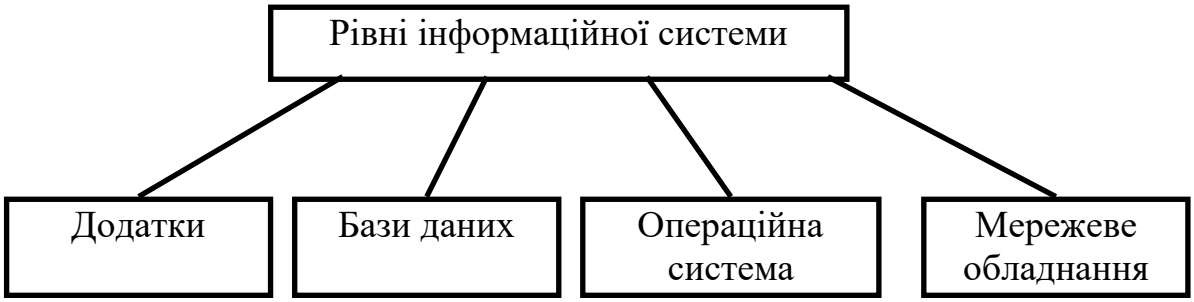


Рисунок 1.4 – Рівні інформаційної системи

Рівень прикладного програмного забезпечення (ПЗ), відповідальний за взаємодію з користувачем. Прикладом елементів ІС, що працюють на цьому рівні, можна назвати текстовий редактор Word, електронні таблиці Excel, поштову програму Outlook Express, системи MS Query тощо [5].

Рівень системи управління базами даних (СУБД), відповідальний за зберігання й обробку даних інформаційної системи. Прикладом елементів ІС, що працюють на цьому рівні, можна назвати СУБД Oracle, MS SQL Server, Sybase і навіть MS Access [5].

Рівень операційної системи (ОС), відповідальний за обслуговування СУБД і прикладного програмного забезпечення. Як приклади елементів ІС цього рівня можна привести ОС Microsoft Windows, Linux, Mac OS [5].

Рівень мережі, відповідальний за взаємодію вузлів інформаційної системи [5]. Для цього рівня характерними прикладами відповідних елементів ІС є модулі, взаємодіючі по протоколах TCP/IP, IPS/SPX або SMB/NetBIOS.

1.3 Визначення фаз проведення аудиту кібербезпеки

Аудит кібербезпеки – системний процес одержання об'єктивних якісних і кількісних оцінок про поточний стан кібербезпеки комп'ютерної мережі компанії у відповідності з визначеними критеріями та показниками кібербезпеки.

Фази проведення аудиту кібербезпеки:

- збір інформації про систему;
- мережевий аудит;
- локальний аудит (АРМ, сервера, мережевого обладнання);
- інші види аудиту.

Мережевий аудит поділяється, на:

- трасування, дослідження топології системи;
- сканування сервісів;
- інвентаризація ресурсів;
- сканування вразливостей, мережевий аудит паролів;
- перехоплення трафіку, проведення атак типу MitM (людина-посередині).

Локальний аудит поділяється, на:

- збір інформації про поточну програмно-апаратну конфігурацію персонального комп'ютера або пристрою;
- аудит локальних паролів;
- пошук залишкової інформації на носіях, контроль роботи механізмів;
- гарантованого знищення інформації.

Інші види аудиту поділяються, на:

- аудит бездротових комп'ютерних мереж;
- аудит web-вразливостей на Інтернет ресурсах.

Роботи з аудиту кібербезпеки інформаційної системи (ІС) включають в себе ряд послідовних етапів, які в цілому відповідають етапам проведення комплексних інформаційних технологій (ІТ) – аудиту автоматизованої системи, що включає в себе:

- ініціювання процедури аудиту;
- збір інформації аудиту;
- аналіз даних аудиту;
- вироблення рекомендацій;
- підготовку аудиторського звіту.

На етапі ініціювання процедури аудиту повинні бути вирішені наступні організаційні питання:

- права та обов'язки аудитора повинні бути чітко визначені і документально закріплені в його посадових інструкціях, а також у положенні про внутрішній (зовнішній) аудит;
- аудитором повинен бути підготовлений і узгоджений з керівництвом план проведення аудиту;
- у положенні про внутрішній аудит має бути закріплено, зокрема, що співробітники компанії зобов'язані сприяти й надавати аудитору всю необхідну для проведення аудиту інформацію.

На етапі ініціювання процедури аудиту мають бути визначені межі проведення обстеження. План і межі проведення аудиту обговорюються на робочому зборах, в яких беруть участь аудитори, керівництво компанії і керівники структурних підрозділів (таблиця 1.1).

Таблиця 1.1 – Системна класифікація загроз кібербезпеки інформації

Параметри класифікації	Значення параметрів	Зміст значення параметра
1	2	3
1. Види	1.1. Фізична цілісність	Знищення (спотворення)
	1.2. Логічна структура	Спотворення структури
	1.3. Зміст	Несанкціонована модифікація
	1.4. Конфіденційність	Несанкціоноване отримання
	1.5. Право власності	Привласнення чужого права
2. Природа походження	2.1. Випадкова	Відмови, збої, помилки, стихійні лиха, побічні впливи
	2.2. Навмисна	Зловмисні дії людей

Продовження таблиці 1.1

1	2	3
3. Передумови появи	3.1. Об’єктивні	Кількісна недостатність елементів системи, якісна недостатність елементів системи
	3.2. Суб’єктивні	Розвідка іноземних держав, промислове шпигунство, карні елементи, недобросовісні робітники
4. Джерела загроз	4.1. Люди	Сторонні особи, користувачі, персонал
	4.2. Технічні пристрої	Реєстрації, передачі, збереження, переробки, видачі
	4.3. Моделі, алгоритми, програми	Загального призначення, прикладні, допоміжні
	4.4. Технологічні схеми обробки	Ручні, інтерактивні, внутрішньо машинні, мережеві
	4.5. Зовнішнє середовище	Стан атмосфери, стороні шуми, побічні сигнали

Етап збору інформації аудиту є найбільш складним і тривалим. Це пов'язано з відсутністю необхідної документації на ІС і з необхідністю щільної взаємодії аудитора з багатьма посадовими особами організації.

1.4 Категорії порушників кібербезпеки

Кількість державних і комерційних структур, схильних до атак останнім часом значно збільшилася. Цьому сприяє "продуктивна робота" дійових осіб інформаційного і кіберпростір – легальних користувачів, хакерів, кіберзлочинців і кібертерористів, а також підрозділів сучасних кібервійськ.

Саме вони, будучи підкріпленими новими можливостями по злому веб-сайтів, серверів додатків і БД, здатні заподіяти не тільки прямі фінансові втрати, але і паралізувати роботу критично важливих об'єктів інфраструктури країн світу (рисунок 1.5).

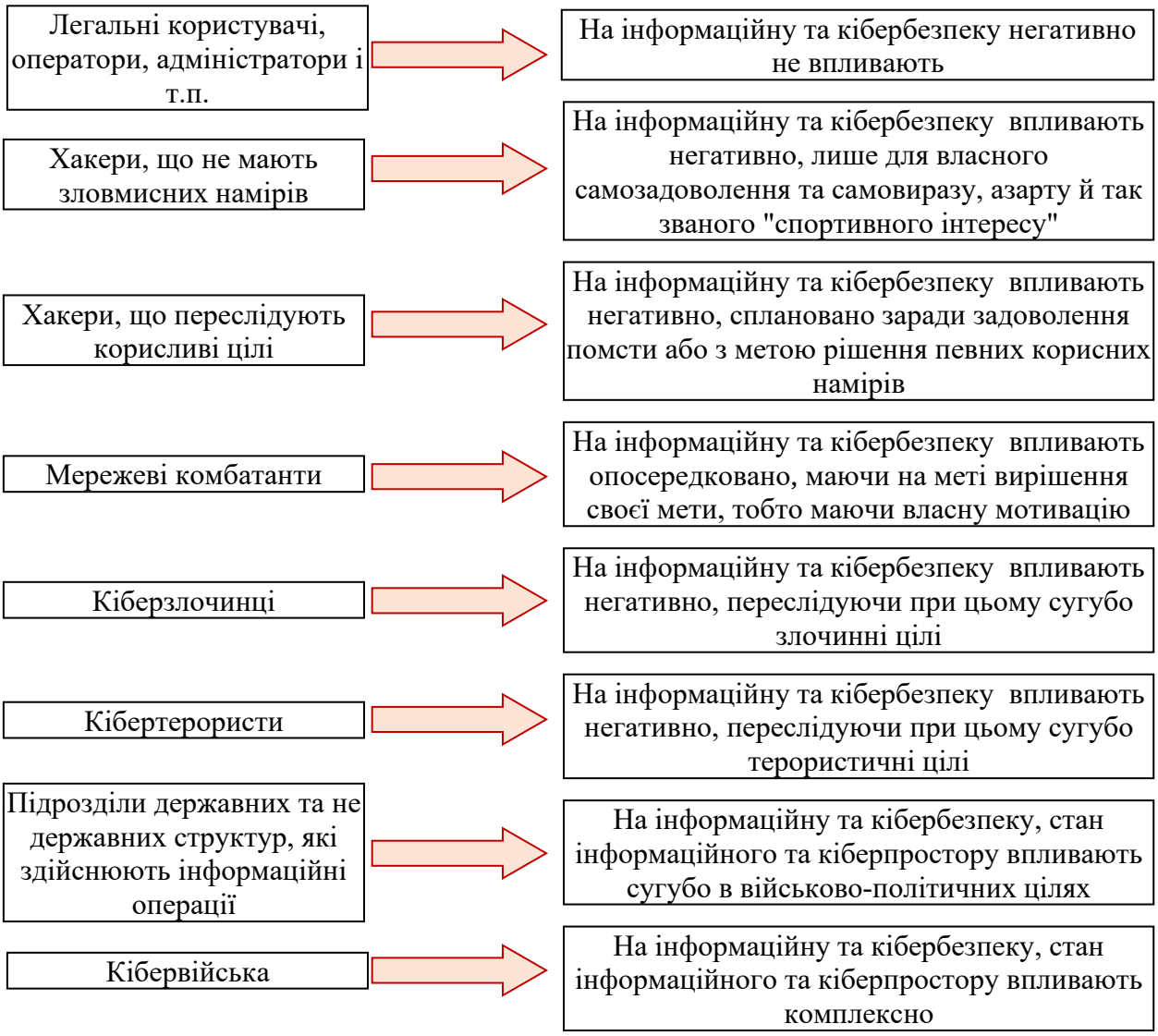


Рисунок 1.5 – Ступінь впливу на інформаційну та кібербезпеку

Причини витоку даних:

- 69% інцидентів – результат зовнішніх атак;
- 18% інцидентів – результат випадкової втрати даних;
- 9% інцидентів – результат діяльності інсайдерів;

- 3% інцидентів – результат діяльності хакерів;
- 1% інцидентів – результат операцій по кіберрозвідці.

Так наприклад, секретні служби США проінформували комітет з озброєнь сенату про загрозу безпеці США № 1.

Нею виявився хакер, який близько 200 разів зламав систему безпеки різного рівня і скопіював десятки секретних файлів, включаючи подробиці досліджень і розробок балістичних ракет. На те щоб його піймати, знадобилось 13 місяців. Хакером виявився англійський 16-річний хлопець, комп'ютерні навички котрого шкільний вчитель учитель оцінив у 4 бали. У ході судового засідання адвокат стверджував, що неповнолітній хакер не мав злого номеру і перебував під враженням від фільму "Ігри патріотів".

Якщо класифікувати види викраденого інформаційного ресурсу, то розподіл буде таким чином:

- 64% – ідентифікаційна інформація;
- 16% – фінансові дані;
- 11% – дані облікових записів;
- 5% – інтелектуальна власність;
- 4% – інше.

З огляду на викладене можна сформулювати наступну гіпотезу, що стала останнім часом об'єктивної реальністю: чим більше IT технології розвиваються і інтегруються в наше повсякденне життя, тим більш важливими і затребуваними в будь-яких сферах людської діяльності стають технологи інформаційної безпеки.

Підтвердженням цьому є статистичні дані та висновки фахівців.

Статистичні дані, оприлюднені корпорацією WASC (Web Application Security Consortium), згідно з якими уразливими до хакерських атак є більш 96% веб-сайтів, близько 74% прикладного та системного програмного забезпечення (рисунок 1.6), приблизно 68% серверних додатків (рисунок 1.7).

Рисунок 1.6 – Типи вразливостей в компонентах операційної системи

Висновки фахівців з міжнародної організації CERT (Computer Emergency Response Team), які стверджують, що кількість інцидентів в інфосфері і кількість виявлених вразливостей щорічно істотно збільшується (рисунок 1.7).

Рисунок 1.7 – Типи вразливостей в компонентах серверних додатків

При цьому і ті, й інші вважають, що як в ПО, так і в серверних додатках домінують, останнім часом, одні і ті ж уразливості типу: відмови в обслуговуванні, компрометації системи і підвищення привілеїв.

Але, досить часто буває так, що купуючи дороге антивірусне ПЗ і дорогі апаратні брандмауери, - переважна більшість замовників не отримує при цьому практично нічого, крім теоретичних доказів того, що вкладені кошти роблять їх мережі від хакерських атак більш захищеними.

Всі порушники діляться на дві категорії зовнішні та внутрішні. У свою чергу, внутрішні порушники діляться на зареєстрованих і незареєстрованних користувачів у системі. Зареєстровані користувачі діляться, відповідно до наявних прав, на простих користувачів, привілейованих користувачів і адміністраторів. Зовнішні порушники діляться на користувачів компанії і користувачів мережі Internet [4]. Для наочності класифікація порушників представлена на рисунок 1.8.

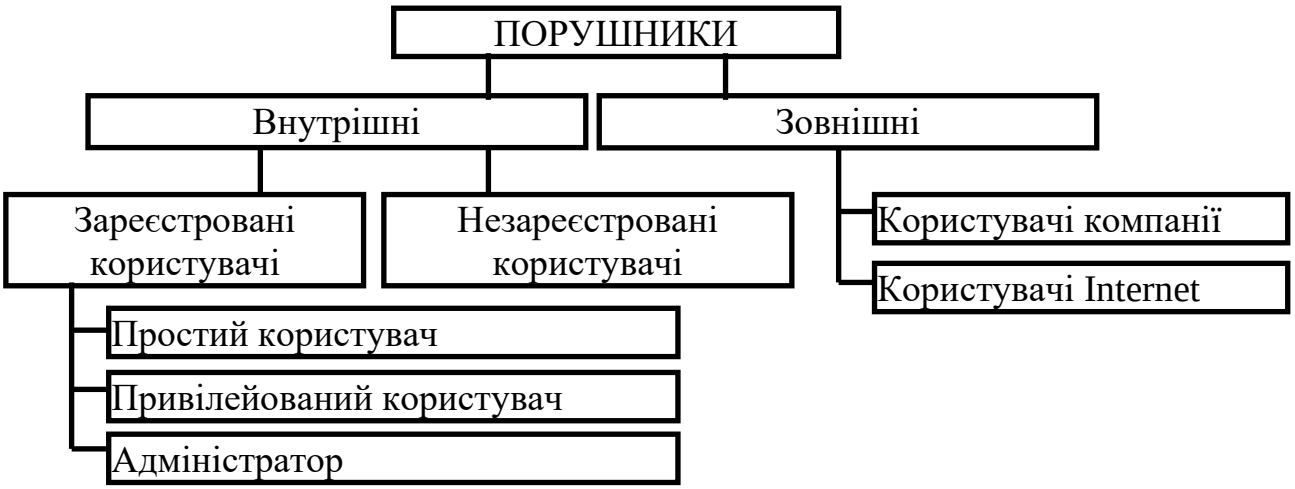


Рисунок 1.8 – Класифікація порушників

Внутрішнім порушником може бути особа з наступних категорій співробітників:

- користувачі компанії;
- обслуговуючий персонал (адміністратори КМ, системні адміністратори, інженери, що обслуговують обладнання компанії);
- співробітники-програмісти, що супроводжують системне й прикладне програмне забезпечення;

- технічний персонал (робітники підсобних спеціальностей, прибиральниці), що працює в будинках, у яких розміщується обладнання компанії;

- інші співробітники підрозділів, що мають санкціонований доступ у будинки, де розташоване обладнання передачі й обробки інформації компанії.

Передбачається, що несанкціонований доступ на об'єкти компанії сторонніх осіб виключається організаційними мірами (охорона території, організація пропускного режиму) [4].

Припущення про кваліфікації внутрішнього порушника формулюються таким чином.

Внутрішній порушник:

- є висококваліфікованим фахівцем в області розробки й експлуатації програмного забезпечення й технічних засобів, знає специфіку завдань, розв'язуваних у мережі компанії, є системним програмістом, здатним програмно модифікувати роботу операційних, у тому числі мережних операційних систем;

- правильно представляє функціональні особливості роботи мережі компанії і закономірності формування в ній масивів інформації й потоків запитів до них;

- може використати тільки штатне обладнання й технічні засоби, наявні в складі компанії.

Внутрішні порушники підрозділяються на чотири категорії (від A до D) залежно від способу доступу й повноважень доступу.

Категорія A: Не зареєстровані мережі компанії особи, що мають санкціонований доступ у приміщення з обладнанням.

Особи, що ставляться до категорії A:

- можуть мати доступ до будь-яких фрагментів інформації про термінальне й серверне обладнання компанії і встановленому на них програмному забезпеченні;

- можуть мати у своєму розпорядженні будь-які фрагменти інформації про топологію мережі (комунікаційної частини підмережі) і про використовувані комунікаційні протоколи і їх сервісах.

Категорія B: зареєстрований користувач мережі компанії, що здійснює доступ до системи по КМ і з вилученого робочого місця. Особа, що ставиться до категорії B:

- знає щонайменше одне легальне ім'я доступу (спосіб доступу);
- має всі необхідні атрибути, що забезпечують доступ до мережі компанії (наприклад, паролем);
- має інформацію про топологію КМ компанії, технічних і програмних засобах обробки інформації у компанії;
- має можливість прямого (фізичного) доступу до фрагментів технічних засобів компанії.

Категорія C: зареєстрований користувач із повноваженнями системного адміністратора [4]. Особа, що ставиться до категорії C:

- має всі можливості осіб категорії B;
- має повну інформацію про системному й прикладному програмному, забезпеченні компанії;
- має повну інформацію про технічні засоби й конфігурацію мережі компанії;
- має доступ до всіх технічних засобів обробки інформації й даним, має права конфігурування й адміністративного настроювання технічних і програмних засобів обробки інформації.

Категорія D: адміністратори обладнання (програмного забезпечення) компанії [2]. Особа, що ставиться до категорії D:

- має можливості внесення помилок, програмних "закладок", "троянських коней", вірусів у ПЗ компанії на стадії впровадження й супроводу ПЗ;
- може мати у своєму розпорядженні будь-які фрагменти інформації про топологію КМ компанії і технічних засобах обробки компанії.

Зовнішній порушник. Припущення про кваліфікації зовнішнього порушника формулюються таким чином:

- є висококваліфікованим фахівцем в області розробки й експлуатації програмного забезпечення й технічних засобів, знає специфіку завдань, розв'язуваних у мережі компанії, є системним програмістом, здатним програмно модифікувати роботу операційних, у тому числі мережних операційних систем ;

- знає мережне і канальне обладнання, протоколи передачі даних, використовуваних у компанії;

- знає особливості системного й прикладного програмного забезпечення, а також технічних засобів компанії;

- знає функціональні особливості роботи системи й закономірності формування в ній масивів інформації й потоків запитів до них.

Зовнішній порушник – це особи, що мають можливість впливати на мережу компанії і її ресурси тільки зовні. Зовнішні порушники діляться на дві категорії А и В.

Категорія А: всі користувачі компанії, які не входять в число внутрішніх користувачів компанії.

Категорія В: користувачі загальнодоступної мережі Internet.

В якості потенційного порушника кібербезпеки об'єкта захисту розглядатимемо особа або групу осіб, які перебувають або не перебувають у змові, які в результаті умисних або ненавмисних дій можуть реалізувати різноманітні загрози кібербезпеки, спрямовані на інформаційні ресурси комп'ютерної мережі та завдати моральної та/або матеріальної шкоди інтересам власника інформації.

Як загроз інформаційній безпеці розглядаються базові загрози порушення конфіденційності та цілісності інформації, а також загроза відмови в обслуговуванні інфраструктури інформаційної системи. Зведена характеристика ймовірного порушника приведена в таблиця 1.2.

Таблиця 1.2 – Характеристика ймовірного порушника

Класифікація	Характеристика
За мотиву порушення ІС	Порушення загрози цілісності, конфіденційності, доступності в корисливих чи інших цілях
За рівнем інформованості та кваліфікації порушника	Порушник має високий рівень знань в області програмування та обчислювальної техніки, проектування і експлуатації автоматизованих інформаційних систем
	Порушник має достатні знання для збору інформації, застосування відомих експлойтів і написання власного програмного забезпечення для здійснення атаки
	Порушник не є авторизованим користувачем інформаційної системи
За місцем дії	Без безпосереднього (фізичного) доступу на територію об'єкта (зовнішній порушник). Порушник діє віддалено, через мережу Інтернет

1.5 Життєвий цикл тестування інформаційної системи на проникнення

Основним заходом зовнішнього аудиту є так званий тест на проникнення або на стійкість до злому. Для проведення тестування всі методики можливо розділити на 3 групи (рисунок 1.9).

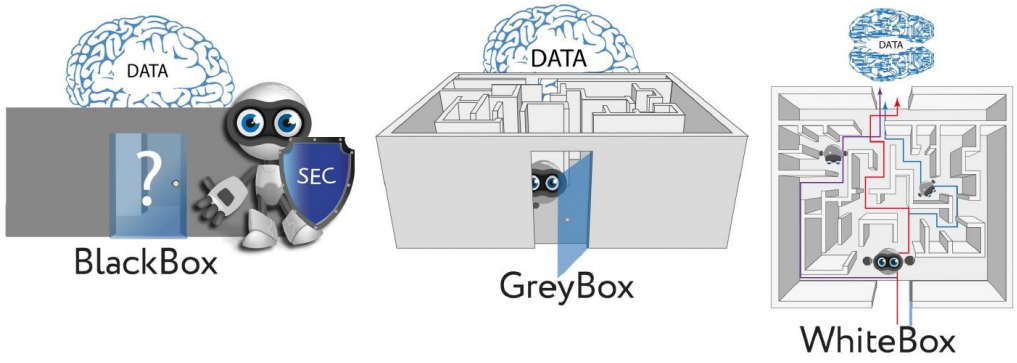


Рисунок 1.9 – Методи тестування на проникнення

BlackBox (Чорний ящик). Вихідні дані, які надаються замовником тестування дуже обмежені або повністю відсутні. Виконавець вимушений оперувати тільки інформацією, необхідною для ідентифікації самого об’єм робіт. Виконавець імітує групу хакерів, які мають тільки назва компанії і практично нульові відомості про систему, що є метою дослідження. Для

реалізації поставленого завдання йому необхідні лише діапазон зовнішніх IP-адрес і, можливо, e-mail адреси внутрішніх користувачів системи. Наприклад, область дії ІС, перечень IP-адрес тощо.

GrayBox (Сірий ящик). Вихідні дані, які надаються замовником тестування можуть містити інформацію про архітектуру комп'ютерної мережі, архітектуру сервісів та служб в неї, перелік облікових записів тощо.

WhiteBox (Білий ящик). Вихідні дані, які надаються замовником тестування містять максимально повну інформацію про комп'ютерну мережу. Виконавець має доступ до систем і повну інформацію про них. Така модель тестування використовується як частина організаційно-технічного аудиту організації IT і передбачає аналіз процесів і процедур.

Методологія тестування мережі на стійкість до злому або тестування на проникнення (Penetration Testing або Ethical hacking) має на увазі, що суб'єкт, який виконує аудит, спирається на власне розуміння того, як реалізована тестова система. Він володіє мінімумом інформації про об'єкт тестування, тому іноді такий тест називають "методом чорного ящика". Мета такого тесту – пошук способів отримання доступу до системи за допомогою інструментів і прийомів, які використовуються зловмисниками.

Справжня методологія ґрунтується на наступних стандартах:

- NIST SP800-115;
- PCI Data Security Standard (PCI DSS) 3.0;
- Penetration Testing Execution Standard Technical Guideline.

Виділяються наступні етапи при проведенні тестування на проникнення:

- підготовка до проведення тестування (проводиться на стадії укладання договору);
- збір інформації;
- аналіз вразливостей;
- експлуатація та активність після експлуатації;
- розробка звітної документації.

Відповідно до NIST SP800-115 життєвий цикл тестування показаний на рисунок 1.10.

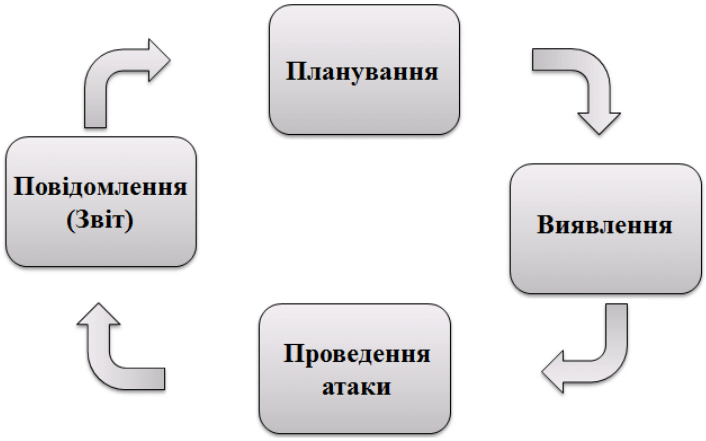


Рисунок 1.10 – Життєвий цикл тестування

У свою чергу, проведення атаки можна розбити на елементи (рисунок 1.11).

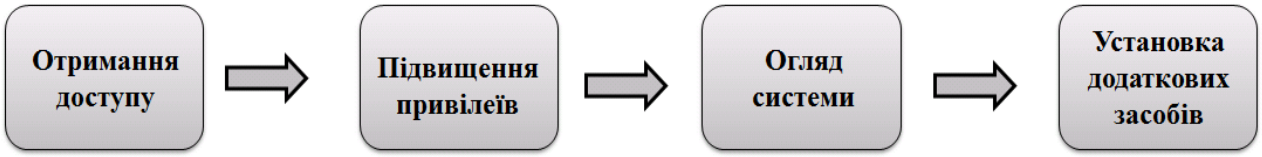


Рисунок 1.11 – Етапи тестування на проникнення

Етап збору інформації полягає в отриманні додаткової інформації про досліджувані сегменти мережевої інфраструктури, ідентифікації доступних сервісів, захисних механізмів.

В ході етапу фахівцем, який проводить тестування, здійснюється виконання наступних процедур, які дозволяють отримати додаткову інформацію про тестовані підмережі, програмне забезпечення, доступні сервіси, їх версії та іншої необхідної для проникнення інформації.

Збір "цифрових слідів". Цей етап включає зовнішній збір даних, до якого відноситься:

- збір даних про доступні підмереж, IP-адреси;
- проведення пасивної розвідки: WHOIS або Border Gateway Protocol.

Проведення активної розвідки:

- сканування портів;
- збір банерів;
- тестування SNMP;
- DNS Zone Transfers;
- SMTP Bounce Back (відправка повідомлення з невірними і/або пошкодженими даними/заголовками для збору інформації про SMTP сервісу. Так само відома як Non-Delivery Report/Receipt (NDR) або (failed) Delivery Status Notification (DSN) message;
- DNS discovery;
- reverse DNS;
- виявлення web-додатків.

Складання списку "зовнішніх цілей":

- збір даних про версії сервісів;
- ідентифікація встановлених патчів;
- пошук вразливих, "слабких" веб-додатків;
- збір інформації про доступні спробах до блокування звернень.

Збір даних про засоби захисту:

- мережевого рівня;
- рівня хоста;
- рівня програмних додатків;
- захист кінцевих користувачів.

Етап аналізу вразливостей полягає в виявлені наявності вразливостей і можливості їх експлуатації. Якщо бути більш точним, то вживають в зв'язці два терміни власне вразливості (vulnerabilities) і дефекти (exposures).

"Класичні" вразливості – це помилки в програмному забезпеченні (наприклад, допущене розробниками ПЗ переповнення буфера), які можуть бути використані хакерами для отримання несанкціонованого доступу і досягнення інших бажаних для них цілей.

"Дефект" (точніше, "схильність сторонньому впливу") - це порушення кібербезпеки, викликане неправильно вибрано параметр або конфігурацією програми, тобто провина не розробників, а адміністраторів. Типовий явний дефект - порожній або легко вгадується пароль доступу.

Насправді тільки дуже невеликий відсоток комп'ютерних злочинців досліджують і розробляють власні методи атак. Більшість зловмисників діють за допомогою опублікованих вразливостей з публічних джерел і так званих експлойтів. Використання авторитетних публічних джерел для підтвердження існування вразливостей і збору інформації про вразливість розділяється на:

- використання публічних джерел;
- підтвердження існування вразливостей (PoC, exploits);
- пошук слабких паролів, словникових паролів, часто використовуваних паролів;
- пошук інформації про вразливість на сайтах виробників мережевого обладнання;
- виконання перевірок наявності вразливостей за списком Sans Top 25.

Наприклад, використання публічних джерел:

- national vulnerability database – http://nvd.nist.gov/;
- common vulnerabilities and exposures list – https://cve.mitre.org/.

Наприклад, підтвердження існування вразливостей (PoC, exploits):

- exploit-db – http://www.exploit-db.com;
- security focus – http://www.securityfocus.com;
- packetstorm – http://www.packetstorm.com;
- security reason – http://www.securityreason.com;
- black asylum – http://www.blackasylum.com/?p=160.

Етап експлуатації та активності після експлуатації передбачає отримання доступу на основі атак, спланованих на попередньому етапі:

- підвищення привілеїв. У тому випадку, якщо на основі попередньої процедури вдалося отримати призначений для користувача рівень доступу, тестувальник робить спроби підвищити рівень своїх привілеїв до адміністратора;
- огляд системи. Подальший збір інформації для ідентифікації та пошуку нових можливостей отримання доступу до інших систем;
- встановлення додаткових засобів. Установка додаткових засобів на систему, до якої вдалося отримати доступ з метою збору інформації або подальшого проникнення.

Етап розробки звітної документації включає в себе:

- документування інформації та результатів в ході попередніх етапів;
- розробку звіту, що містить інформацію як для бізнес-підрозділів (загальна статистика, досягнуті результати), так і для технічних фахівців (подробиці виявлених вразливостей).

1.6 Висновки за розділом 1

Дослідження показують, що забезпечення кібербезпеки інформації об’єктів критичної інфраструктури держави є актуальною проблемою сьогодення і для її вирішення необхідно віднести наступні заходи:

- розробка нормативного, правового регулювання у сфері забезпечення кібербезпеки інформації в комп’ютерній мережі;
- визначення загроз кібербезпеки інформації і виявлення вразливостей в програмному і апаратному забезпеченні комп’ютерній мережі;
- оцінка реальної захищеності комп’ютерній мережі;
- розробка вимог по забезпеченню кібербезпеки інформації в комп’ютерній мережі;

- розробка та реалізація заходів по забезпеченню кібербезпеки інформації в комп’ютерній мережі;

- підготовка фахівців в області забезпечення кібербезпеки інформації в комп’ютерній мережі;

- здійснення контролю і нагляду в галузі забезпечення кібербезпеки інформації в комп’ютерній мережі;

- здійснення інформаційного, матеріально-технічного і науково-технічного забезпечення кібербезпеки інформації в комп’ютерній мережі;

- запровадження відповідного управлінського впливу щодо забезпечення кібербезпеки інформації об’єктів комп’ютерній мережі.

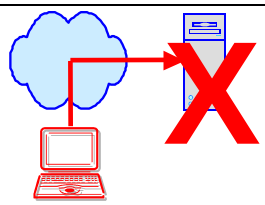
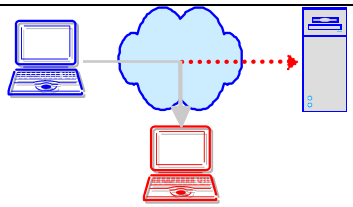
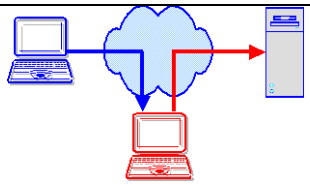
Таким чином, на основі приведених вище результатів досліджень можливо сформулювати основні завдання, рекомендації та базові пропозиції щодо підходів до забезпечення кібербезпеки інформації, яка циркулює на об'єктах комп’ютерної мережі.

2 АНАЛІЗ ВРАЗЛИВОСТЕЙ ІНФОРМАЦІЙНИХ СИСТЕМ ТА ОЦІНКА ЇХ КРИТИЧНОСТІ ДЛЯ КОМП'ЮТЕРНИХ МЕРЕЖ

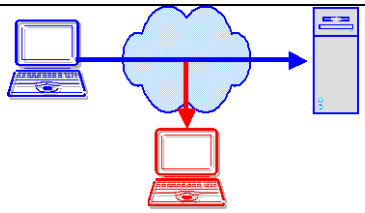
2.1 Класифікація атак на комп'ютерні мережі

Для розуміння того, як можна використовувати знайдені вразливості, необхідно систематизувати можливі віддалені атаки зловмисника на мережеві ресурси. Одна з класифікацій можливих видів віддалених мережевих атак приведена в рекомендаціях Міжнародного телекомунікаційного союзу ITU-T X.800. Security architecture for open systems interconnection for CCITT applications (таблиця 2.1).

Таблиця 2.1 – Класифікація можливих видів мережевих атак

Опис	Мнемонічна схема
1	2
Destruction. Атака на доступність руйнування. Руйнування інформації та/або мережевих ресурсів.	
Interruption. Атака на доступність. Переривання обслуговування, мережа недоступна або непридатна до використання	
Removal. Атака на доступність. Крадіжка, видалення або втрата інформації і/або інших ресурсів.	
Corruption. Атака на цілісність. Несанкціонована модифікація цінної інформації.	

Продовження таблиці 2.1

1	2
Disclosure. Атака на конфіденційність. Несанкціонований доступ до конфіденційної інформації	

Систематизація знань про атаки допомагає розробці заходів і систем захисту від них. Тому фахівці в області кібербезпеки не припиняють спроб побудови різних класифікаційних схем, які в тій чи іншій мірі сприяють розумінню процесів, що ведуть до проникнення в системи, і допомагають розробляти заходи захисту і реалізовувати системи захисту. Як приклади побудови таких класифікаційних схем розглянемо:

- списки категорій Чезвіка і Белловіна;
- матричні схеми;
- таксономічні схеми Ховарда;
- процеси Столінгса;
- онтологію мережевих атак.

Списки категорій Чезвіка і Белловіна. Атаки підрозділяються на наступні сім категорій:

- крадіжка паролів (stealing passwords) – методи отримання паролів користувачів;
- соціальна інженерія (social engineering) – використання деяких психологічних прийомів по відношенню до користувачів для отримання бажаної інформації або інформації обмеженого використання;
- помилки і потаємні ходи (bugs and backdoors) – пошук стану системи, що не відповідає специфікації, або перезапис компонент ПЗ скомпрометовані компонентами;
- відмови автентифікації (authentication failures) – видалення механізмів, що використовуються для автентифікації;

- відмови протоколів (protocol failures) – протоколи самі по собі або погано спроектовані, або погано реалізовані;

- витік інформації (information leakage) – використання таких систем, як finger або DNS, для отримання інформації, необхідної адміністраторам для належного функціонування мережі, але використовуваної атакуючим;

- відмова в обслуговуванні (denial-of-service) – зусилля для припинення можливості користувачів користуватися службами.

В даному підході також існує перекриття понять, що вимагало застосування списків всередині категорій.

Матричні схеми. Матричні схеми базуються на кількох вимірах. Для двох вимірювань розглядаються вразливості і потенційні порушники.

Матричний підхід був реалізований Лендвейром, який представив таксономію вразливостей (умов, які можуть призвести до відмови в обслуговуванні або неавторизованого доступу), що базується на трьох вимірах:

- походження (Genesis) – як вразливості знаходять свою дорогу до ПЗ;

- час впровадження (Time of introduction) – в життєвому циклі ПЗ або апаратури;

- місце розташування (Location) – розташуванню в ПЗ або апаратурі.

Цей підхід ілюструється в таблиці 2.2. Дана таксономія знайшла своє застосування при розробці систем виявлення вторгнень.

Класифікація Ховарда. Таксономія атак розроблена Ховардом (Howard) на основі аналізу статистики інцидентів CERT (Computer Emergency Response Team). Таксономічна схема Ховарда представлена на рисунку 2.1.

До переваг даної класифікації можна віднести досить хорошу опрацювання категорій. Недоліки таксономії визначаються її метою – побудовою класифікації для вже здійснених атак. Пізніше Ховард спільно з Лонгстаф розробив уточнену таксономию. Подальшим кроком в побудові таксономії атак з'явилися спроби побудови онтології атак.

Таблиця 2.2 – Матрична класифікація Лендвейра

<table>
<tr><td rowspan="6">Походження (Genesis)</td><td rowspan="4">Навмисні (Intentional)</td><td rowspan="3">Зловмисні (Malicious)</td><td rowspan="2">Троянські коні (Trojan Horse)</td><td>Некопіровані (Non-Replicating)</td></tr>
<tr><td>Копійовані(Replicating)</td></tr>
<tr><td colspan="2">Потаємні ходи (Trapdoor)</td></tr>
<tr><td colspan="2">Логічні/Часові бомби (Logic/ Time Bomb)</td></tr>
<tr><td rowspan="2">Без злого умислу (Non-Malicious)</td><td rowspan="2">Приховані канали</td><td>Постійні (Storage)</td></tr>
<tr><td>Тимчасові (Timing)</td></tr>
<tr><td colspan="3">Інші</td></tr>
<tr><td rowspan="5">Випадкові (Inadvertent)</td><td colspan="3">Помилки перевірок (Неповнота/Несумісність)</td></tr>
<tr><td colspan="3">Помилки домену (включаючи повторне використання об'єкта, залишки і помилки відкритого дійства)</td></tr>
<tr><td colspan="3">Серіалізація / Заміни (Serialization / Aliasing)</td></tr>
<tr><td colspan="3">Неадекватна ідентифікація / автентифікація (Identification/Authentication Inadequate)</td></tr>
<tr><td colspan="3">Порушення кордонів (включаючи вичерпання ресурсу і помилки порушення обмежень</td></tr>
<tr><td colspan="4">Інші помилки логіки застосування</td></tr>
</table>

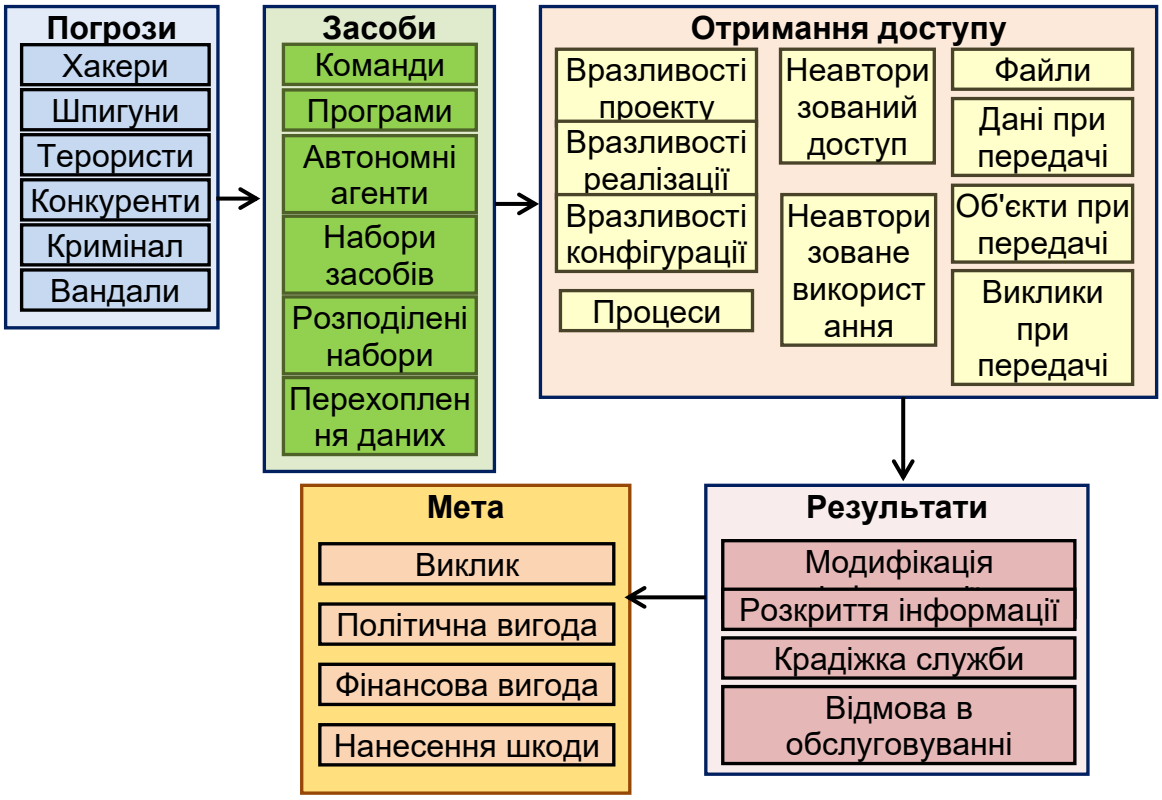


Рисунок 2.1 – Таксономія атак Ховарда

Онтологія – філософський термін, що визначає вчення про буття. Це положення простежується і в специфікаціях Міжнародної федерації з розробки інтелектуальних фізичних агентів. Розглянемо застосування онтології для побудови класифікації атак з точки зору мети атаки. Високорівневе уявлення атак включає в себе наступні властивості:

- мета атаки;
- засіб атаки;
- результат атаки;
- розташування джерела атаки.

Відповідно до цього високорівневе подання вторгнення являє собою деякий процес введення даних, який отриманий з певного місця розташування, спрямований на певний системний компонент, що використовує метод і викликає деяку системну поведінку (рисунок 2.2).

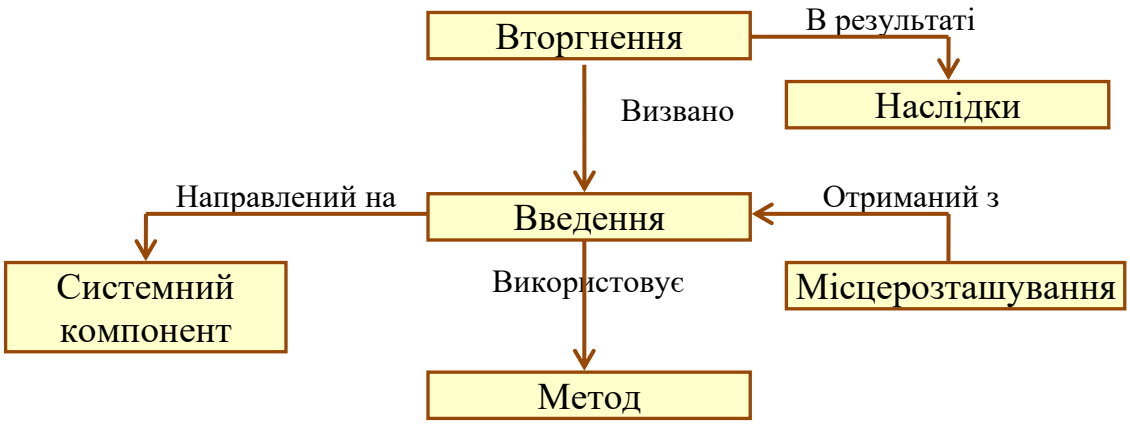


Рисунок 2.2 – Спрощена онтологія атак

Повна онтологія атак в графічній формі представлена на рисунок 2.3. Розглянемо характеристики та властивості даної онтології.

Системний компонент, який є метою атаки:

- мережевий протокол (атака використовує протоколи стека протоколів і не виходить за них);
- простір ядра (процес, що виконується як частина ОС, який або

компілюється в ядро, або завантажується модулем і використовується в ядрі);

- додаток (додаток, що виконується поза простором ядра з привілеями користувача або root);
- інші (які не вказані вище, наприклад, принтери, модеми).

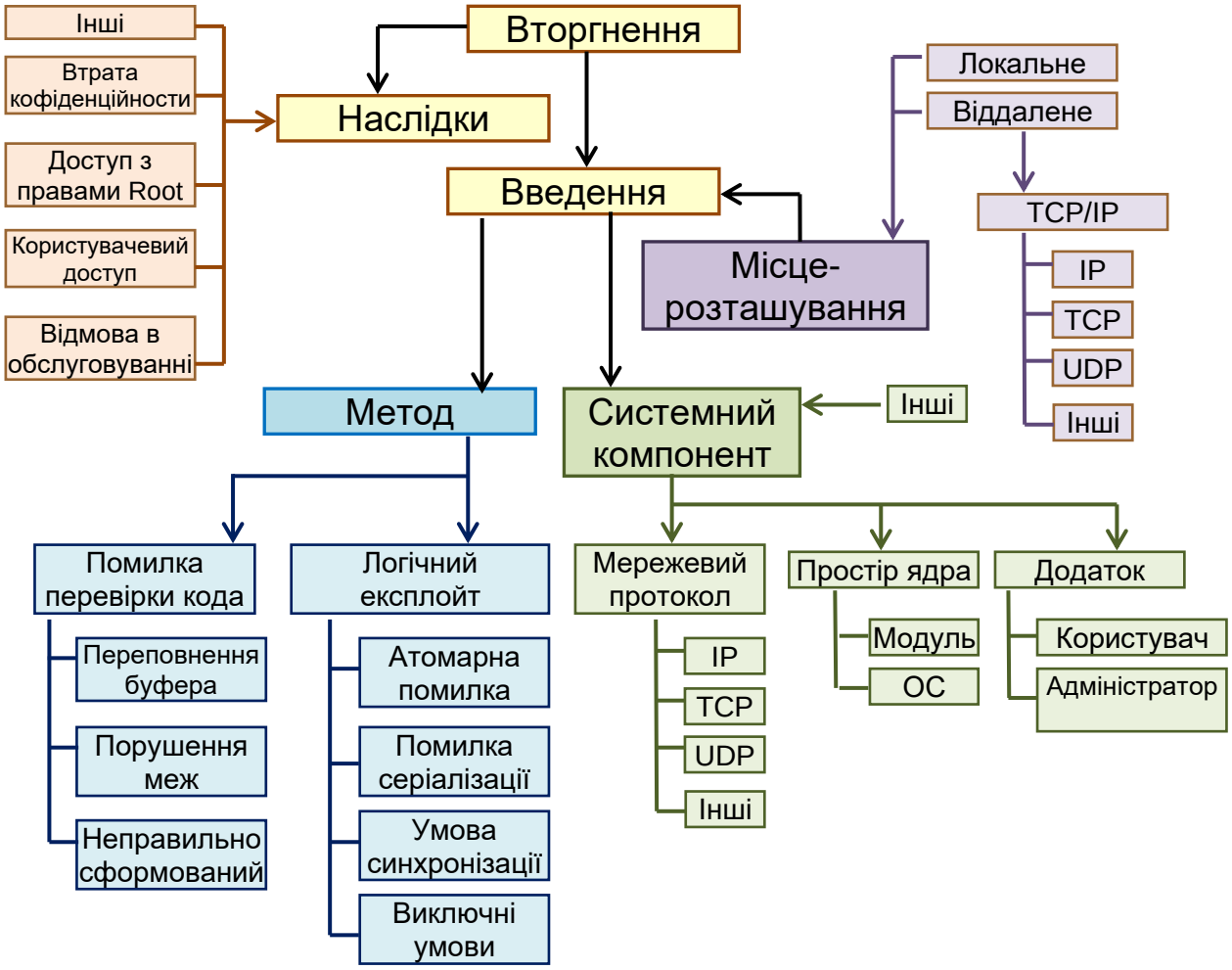


Рисунок 2.3 – Повна онтологія атак

Метод, який використовується атакуючим:

- помилка перевірки введення, яка включає в себе: переповнення буфера, порушення кордонів, неправильно сформований введення;
- логічний експлойт, що використовує вразливість і веде до зниження продуктивності і (або) компрометації системи: виняткові умови (помилки, викликані відмовою обробки виняткових умов, згенерованих функціональ-

ним модулем або пристроєм), умова синхронізації (помилки, що виникають під час тимчасового проміжку між двома операціями), помилки серіалізації в результаті неправильних операцій серіалізації, атомні помилки (помилки, що виникають, коли частково модифіковані одним процесом дані вже використовуються іншим процесом).

Наслідки – кінцевий результат атаки:

- відмова в обслуговуванні користувачів системи;
- призначений для користувача доступ (атакуючий отримує доступ до деяких служб цільової системи);
- доступ з правами root (атакуючий отримує повне управління системою);
- втрата конфіденційності (в результаті атаки користувач системи втрачає конфіденційність даних);
- інші (результат полягає в компрометації цілісності або інших небажаних характеристиках).

Місцезнаходження джерела атаки - атакуючий з'єднується через мережу або знаходиться на хості:

- віддалене (атакуючому немає необхідності "віртуально" перебувати на цілі);
- локальні (атакуючому необхідно "віртуально" бути присутнім на цілі);
- віддалено-локальні (атакуючий на різних стадіях атаки може бути як віддаленим, так і локальним).

2.2 Характеристика вразливостей інформаційних систем в комп’ютерних мережах

Вразливість інформаційної системи поділяється на наступні класи:

- вразливість кода;
- вразливість конфігурації;

- вразливість архітектури;
- організаційні вразливості;
- багатофакторні вразливості.

Принципи кібербезпеки:

- цілісність інформаційних даних означає здатність інформації зберігати початковий вигляд і структуру як в процесі зберігання, так і після неодноразової передачі. Вносити зміни, видаляти або доповнювати інформацію вправі тільки власник або користувач з легальним доступом до даних;

- конфіденційність – характеристика, яка вказує на необхідність обмежити доступ до інформаційних ресурсів для певного кола осіб. У процесі дій і операцій інформація стає доступною тільки користувачам, які включені в інформаційні системи і успішно пройшли ідентифікацію;

- доступність інформаційних ресурсів означає, що інформація, яка знаходиться у вільному доступі, повинна надаватися повноправним користувачам ресурсів своєчасно і безперешкодно;

- достовірність вказує на приналежність інформації довіреній особі або власнику, який одночасно виступає в ролі джерела інформації.

Забезпечення і підтримка кібербезпеки включають комплекс різнопланових заходів, які запобігають, відстежують і усувають несанкціонований доступ третіх осіб. Заходи ІБ спрямовані також на захист від пошкоджень, спотворень, блокування або копіювання інформації. Принципово, щоб всі завдання вирішувалися одночасно, тільки тоді забезпечується повноцінний, надійний захист.

Особливо гостро ставляться основні питання про інформаційний спосіб захисту, коли злом або розкрадання з перекручуванням інформації потягнуть за собою ряд важких наслідків, фінансових збитків.

Створений за допомогою моделювання логічний ланцюг трансформації інформації виглядає наступним чином (рисунок 2.4).

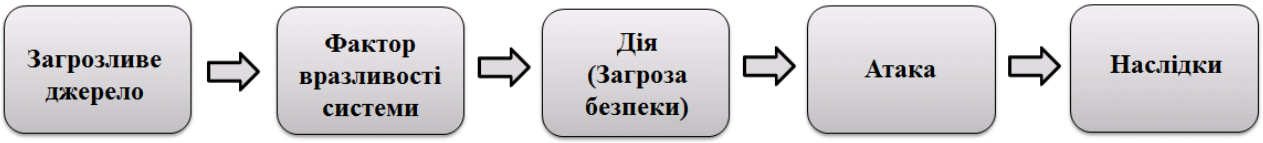


Рисунок 2.4 – Логічний ланцюг трансформації інформації

Загрози кібербезпеки проявляються не самостійно, а через можливу взаємодію з найбільш слабкими ланками системи захисту, тобто через фактори вразливості. Загроза призводить до порушення діяльності систем на конкретному об'єкті.

Основні вразливості виникають унаслідок дії наступних факторів:

- недосконаліть програмного забезпечення, апаратної платформи;
- різні характеристики будови автоматизованих систем в інформаційному потоці;
- частина процесів функціонування систем є неповноцінною;
- неточність протоколів обміну інформацією та інтерфейсу;
- складні умови експлуатації і розташування інформації.

Найчастіше джерела загрози запускаються з метою отримання незаконної вигоди внаслідок заподіяння шкоди інформації. Але можлива і випадкова дія загроз через недостатню міру захисту і масової дії загрозливого фактора.

Для оцінки небезпеки знайдених вразливостей розглянемо методику Common Vulnerability Scoring System (CVSS).

Загальна система оцінки вразливостей CVSS – це відкрита схема, яка дозволяє обмінюватися інформацією про вразливості на ІС. Система оцінки CVSS складається з 3 метрик: базова метрика, тимчасова метрика і контекстна метрика. Кожна метрика являє собою число (оцінку) в інтервалі від 0 до 10 і вектор – короткий текстовий опис зі значеннями, які використовуються для виведення оцінки. Базова метрика відображає основні характеристики вразливості. Тимчасова метрика відповідає таким

характеристикам вразливості, які змінюються з часом, а контекстна метрика – характеристикам, які є унікальними для середовища користувача.

Група базових метрик представляє основні суттєві характеристики вразливості, які не змінюються з часом і не залежать від середовища.

Група тимчасових метрик представляє такі характеристики вразливості, які можуть змінитися з часом, але не залежать від середовища.

Група контекстних метрик представляє такі характеристики вразливості, які залежать від середовища.

Метрики потрібні для того, щоб зрозуміло визначити і відобразити основні характеристики вразливості, а також розрахувати підсумкову оцінку.

CVSS є зрозумілим, прозорим і загальноприйнятим способом оцінки вразливостей ІС для керівників, виробників додатків і засобів підтримки кібербезпеки, дослідників та ін.

Група базових метрик відображає характеристики вразливості, які не змінюються з часом і не залежать від контексту (таблиця 2.3).

Таблиця 2.3 – Розшифровка базових метрик

Метрика	Можливі значення	Опис
1	2	3
Вектор доступу (AV)	AV: L (Локальний)	Для експлуатації вразливості зловмисник повинен мати локальний доступ, тобто фізичний доступ до системи або локальну обліковий запис.
	AV: A (Сусідня мережа)	Для експлуатації вразливості зловмисник повинен мати доступ до сусідньої мережі, тобто мережі, яка має загальну середу передачі з мережею, де знаходиться вразливе ПЗ.
	AV: N (Віддалений)	Для експлуатації вразливості зловмисник повинен мати доступ до вразливому ПЗ, причому цей доступ обмежений тільки величиною мережевого стека. Локального доступу або доступу з сусідньої мережі не потрібно. Такі вразливості часто називають експлуатованими віддалено.

Продовження таблиці 2.3

1	2	3
Автентифікація (AC)	AC: H (Висока)	Для експлуатації вразливості потрібні особливі умови.
	AC: M (Середня)	Для експлуатації вразливості потрібні до деякої міри особливі умови.
	AC: L (Низька)	Для експлуатації вразливості не потрібні спеціальні умови і особливі обставини.
Вплив на конфіденційність (CI)	CI: N (Ні)	Вразливість зачіпає конфіденційність системи.
	CI: P (Істотне)	Є істотне розголошення даних. Можливий доступ до деяких системних файлів, але зловмисник не отримує контролю над цією інформацією, або масштаби втрат невеликі.
	CI: C (Критичний)	Відбувається повне розголошення даних, що призводить до розкриття всіх системних файлів. Зловмисник може їх читати.
Вплив на цілісність (II)	II: N (Ні)	Експлуатація вразливості не впливає на цілісність системи.
	II: P (Істотне)	Можлива зміна деяких системних файлів або інформації, але зловмисник не отримує контроль над змінними даними, або область його впливу обмежена.
	II: C (Критичний)	Експлуатація вразливості дозволяє повністю порушити цілісність системи. Захист системи повністю втрачена, система скомпрометована. Зловмисник може змінювати будь-які файли цільової системи.
Вплив на доступність (AI)	AI: N (Ні)	Експлуатація вразливості не впливає на доступність системи.
	A: P (Істотне)	Відбуваються збої в доступності ресурсу або зменшення продуктивності.
	A: C (Критичний)	Відбувається повне відключення системи. Зловмисник може зробити ресурс повністю недоступним.

Метрики Access Vector (Вектор доступу), Access Complexity (Складність доступу) і Authentication (Автентифікація) оцінюють, як отримати доступ до вразливості і чи потрібні для експлуатації вразливості додаткові умови.

Три метрики впливу – Confidentiality Impact (Вплив на

конфіденційність), Integrity Impact (Вплив на цілісність) і Availability Impact (Вплив на доступність) – описують можливе пряме вплив на ІС в разі експлуатації вразливості.

Це вплив визначається незалежно з точки зору конфіденційності, цілісності та доступності. Це означає, наприклад, що експлуатація вразливості може викликати часткову втрату цілісності і доступності, але не впливати на конфіденційність.

На підставі метрик розраховується загальна небезпека вразливості (немає вразливості, низька, середня, висока, критична).

Найбільш повний і структурований список вразливостей має назву Common Vulnerabilities and Exposures (CVE) [11]. У ньому містяться перелік усіх загальновідомих вразливостей. Основна перевага CVE в тому, що всі вразливості в ньому систематизовані, кожна має свій номер і опис. У свою чергу більшість експлойтів посилаються саме на CVE, щоб можна було легко визначити, яку вразливість використовує той чи інший експлойт.

Експлойт (Exploit) – це програма, послідовність команд або частина програмного коду, що використовують вразливості в ПЗ для проведення атаки на ІС. Цілі атаки можуть бути найрізноманітнішими: отримання доступу, порушення стандартної функціональності системи та інше.

Експлойти можуть уявляти собою вихідний код, виконуваний модуль або навіть словесний опис того, як треба використовувати вразливість. Він може бути написаний практично на будь-якій мові програмування: C/C++, Perl, Ruby, JavaScript, Assembler. На рисунку 2.5 представлені основні складові експлойта.

Shell код - це корисне навантаження, що виконується після успішного запуску експлойта. Найчастіше адреса повернення змінюється на адресу пам'яті, де розташований цей shell код. Shell код є асемблерною командою, яка закодована у вигляді двійкового рядку. При написанні shell кода необхідно дотримуватися балансу між складністю і розміром коду, також є певні обмеження, що накладаються на shell коду (наприклад, відсутність

деяких символів). За допомогою shell кода можна виконувати різні операції, такі як відкриття socket, завантаження шкідливого програмного забезпечення, запуск командного рядка і так далі.

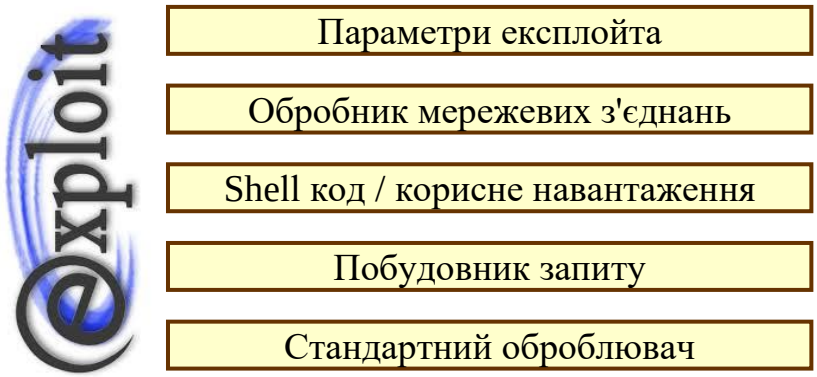


Рисунок 2.5 – Складові експлойта

Вектор ін'єкції – це покажчик або зсув, за яким впроваджується shell код і змінений адреса повернення, який вказує на це місце в пам'яті.

Будівник запиту – це код, який викликає експлойт.

Стандартний оброблювач – це оброблювач shell кода, який виконує такі операції, як зв'язок консолі з socket або створення bind shell (з'єднання з атакується машиною типу "клієнт-сервер").

Оброблювач користувачевих опцій – це призначений для користувача інтерфейс, який надає користувачеві вибирати деякі опції, такі як вибір віддаленої мети, розмір зміщення, додаткова інформації, налагодження тощо.

Для блокування експлойтів застосовуються всі можливі засоби захисту, такі як: антивіруси, міжмережеві екрани, системи виявлення та запобігання вторгнень і інші. Однак, для коректної роботи цих програм необхідна складна і детальна настройка. Адміністратор хоста не може відразу визначити, чи достатньо захищена його система, чи всі вразливості виправлені, чи правильно сконфігуровані програми, тому необхідно проводити аудит безпеки, який, в тому числі, включає в себе тестування засобів захисту інформації.

2.3 Методика оцінки критичності знайдених вразливостей

Одним із найбільш поширених інструментів для аналізу захищеності комп'ютерної мережі є сканер безпеки: програмний чи апаратно-програмний засіб, що дозволяє шляхом здійснення певних перевірок виявити схильність досліджуваного об'єкта до різноманітних вразливостей [8].

Під останніми, як правило, розуміють слабкі місця в ІС, які може привести до порушення безпеки шляхом реалізації певної загрози [8]. Сканери безпеки є зручним і простим інструментом, що допомагає своєчасно виявляти вразливості в ІС.

Спочатку сканери безпеки зародилися як інструментальні засоби, що використовувалися зловмисниками для організації атак, а потім цей інструментарій взяли на озброєння фахівці в галузі захисту інформації. Більш того, найбільш вдалі інструменти для аналізу захищеності переросли в комерційні продукти.

Сьогоднішній ринок інформаційної безпеки представлений різними сканерами безпеки, більшість з яких є орієнтованими на пошук вразливостей у певній технологічній області, наприклад (таблиця 2.4):

- безпека web-додатків (HP WebInspect, Open Source та ін.);
- безпека систем управління базами даних (СУБД) (AppSecInc AppDetective, NGSS, Safety-Lab Shadow Database Scanner);
- безпека операційних систем і мережевих застосунків (GFI LANguard Network Security Scanner, MS Baseline Security Analyzer тощо).

Перевага Penetration Testing в тому, що їх проводять люди, відповідно, вони можуть в реальному часі приймати ті чи інші рішення, змінювати стратегії, засоби атак та інше. Однак, в цьому ж і недолік даного підходу – все залежить від компетентності тих, хто ці Penetration Tests проводить.

Не можна після проведення ряду атак стверджувати, що система володіє потрібним рівнем захищеності. Адже можливо, що просто були вибрані не ті кошти або використовувалися застарілі вразливості.

Таблиця 2.4 – Аналіз програмного забезпечення оцінки вразливостей ІС

№ з/п	Назва	Критерії						
		Оцінка ризиків	Оцінка захищеності	Кросплатформеність	Зручність у користуванні	Відкритий код	Низька вартість	Відповідність міжнародним стандартам
1.	Nessus	+	+	+	+	+	+	+
2.	AccessDiver	–	–	–	+	–	–	+
3.	xSpider	–	+	–	+	–	–	–
4.	LAN guard Network Security Scanner	–	+	–	+	–	–	–
5.	Shadow Security Scanner	–	+	–	+	–	+	–
6.	Nikto	–	–	–	+	+	–	+
7.	14 Day Trial	–	–	+	+	–	+	–
8.	Windows Vulnerability Scanner	–	+	–	+	–	+	+
9.	Microsoft Baseline Security	–	+	–	+	–	+	+
10.	Cobra	+	+	–	+	–	–	+
11.	Cramm	+	+	–	+	–	–	–
12.	Calio Secura	+	+	–	+	–	–	–
13.	Octave	+	+	–	+	–	–	+
14.	Proteus enterprise	+	–	–	+	–	–	+
15.	Ra2 the art of risk	+	+	–	–	–	–	+
16.	Risk watch	+	+	–	+	–	–	–
17.	VsRisk	+	+	–	–	+	–	–

Спорідненим підходом до створення Penetration Testing є відтворенні попередньо записаного трафіку комп'ютерних атак. Налаштовується спеціальний стенд, до якого підключається тестована система. Перевагою цього підходу є можливість забезпечення багаторазової повторюваності умов експерименту за умови, що трафік комп'ютерних атак кожен раз буде

відтворюватися ідентичним чином. Відсутність у складі стенду реальних атакованих систем усуває проблему відновлення стану окремих його елементів після проведення атак. Але розглянутий спосіб тестування не дозволяє варіювати параметри атак (наприклад, кодування HTTP-запитів і використовуваний експлойтів shell-код) в процесі тестування, оскільки це вимагає серйозної модифікації вже сформованих пакетів мережевого трафіку.

Оскільки проводити тестування цільової системи вручну дуже складно, останнім часом стали з'являтися автоматизовані засоби експлойтінга. Такі програми дозволяють в автоматичному режимі виявити вразливості цільового хоста, а потім провести ряд атак для експлуатації цих вразливостей. Загальний алгоритм роботи таких засобів складається з наступних кроків:

- сканування портів і ідентифікація сервісів на досліджуваних об'єктах;
- на основі бази знань висувається припущення про наявність вразливостей в виявлених сервісах;
- перевірка можливості експлуатації передбачуваних вразливостей.

Розглянемо найпопулярніші засоби автоматичного аудиту кібербезпеки:

- Core Impact [12] - це програмний продукт для проведення тестування несанкціонованих проникнень в систему. Перевіряє досліджувані системи на наявність вразливостей. За допомогою нього можна протестувати безпеку веб-додатків, мережевих систем, кінцевих пристроїв, мобільних пристроїв, бездротових мереж і так далі. Дозволяє моделювати багатоступінчасті атаки і взаємодіяти з успішно атакований системою. Має свою БД експлойтів, яка щомісяця поповнюється власною командою розробників. З недоліків можна відзначити досить високу ціну;
- CANVAS/D2 / Nessus Bundle [13]. Збірка CANVAS (компанії Immunity) з D2 Exploit Pack (компанії Square Security) інтегровані зі сканером виявлення вразливостей Nessus дозволяють досягти аналогічного підходу,

закладеного в продукті CORE IMPACT. На відміну від свого конкурента, під платформу CANVAS, крім власних експлойтів, можливий імпорт сторонніх експлойтів. Це дозволяє CANVAS охопити набагато більше вразливостей для проведення атак в порівнянні з іншими рішеннями даного сегмента;

- SAINT Vulnerability Scanner && SAINTexploit [14]. Аналог інтегрованого CANVAS і Nessus з тією лише відмінністю, що у SAINT Vulnerability Scanner і SAINTexploit один виробник. Функціональність же практично повністю ідентична.

Всі описані вище програми є комерційними, з безкоштовних варто виділити Armitage. Потужний засіб, заснований на Metasploit, програмний комплекс під назвою Armitage. Armitage – це GUI додаток (по суті, графічна оболонка), яке спрощує і кілька автоматизує роботу з Metasploit Framework. За допомогою Armitage можна представляти досліджувані хости в візуальному режимі, автоматично підбирати експлойти для них, керувати успішно встановленими сеансами і так далі (рисунок 2.6).

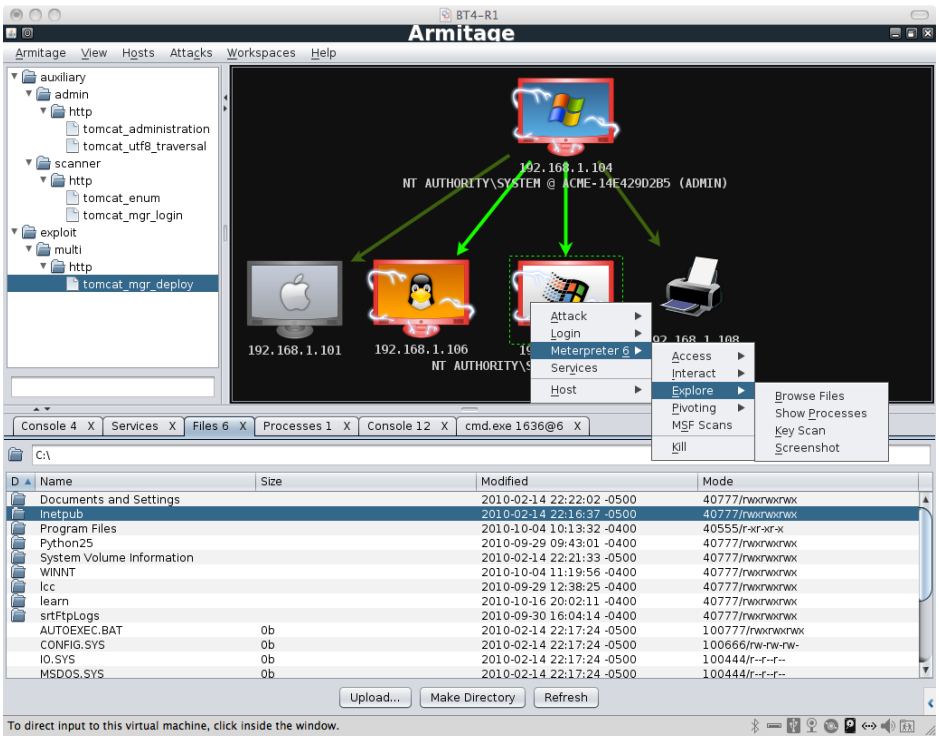


Рисунок 2.6 – Інтерфейс Armitage

Для користувачів-експертів доступна можливість віддаленого управління і спільної роботи з Metasploit (можна використовувати один і той же сеанс, мати загальну базу даних з інформацією про хостах і їх вразливості).

Виявлення хостів в Armitage відбувається за допомогою засобів сканування, закладених в Metasploit. Можна імпортувати раніше складений список хостів мережі і запустити їх сканування, щоб створити БД цілей. Armitage може уявити ці дані в графічному режимі, таким чином, видно, яка з систем в даний момент аналізується, а до якої вже отримано доступ і створено віддалене з'єднання.

Armitage автоматизує вибір найбільш придатних експлойтів для даного хоста і виконує їх перевірку (при наявності у експлойта такої опції). Також можна запустити утиліту Hail Mary, яка запускає всі наявні експлойти поспіль.

При успішному отриманні доступу до віддаленого хосту, Armitage допомагає у виборі подальших дій. До складу Armitage входить ряд інструментів, заснованих на спеціального агента Meterpreter. Meterpreter є агента, за допомогою якого можна виконувати безліч операцій після проникнення в машину. Можна здійснити підвищення прав в системі, завантажити список хеш паролів в локальну БД, переглянути файлову систему віддаленої машини, запустити оболонку командного рядка та інше.

Крім того, Armitage може створити так званий "pivot" – "опорну точку", яка дозволяє використовувати захоплені хости як платформи для організації подальших атак на інші машини і дослідження комп’ютерної мережі.

Саме можливості Armitage використаємо для створення автоматизованого тестувальника на проникнення, тобто Penetration Testing. Загальна структура такого Penetration Testing показана на рисунку 2.7.

Спочатку відбувається збір відомостей про цільовій системі, потім висуваються припущення про наявність будь-яких вразливостей в ній.

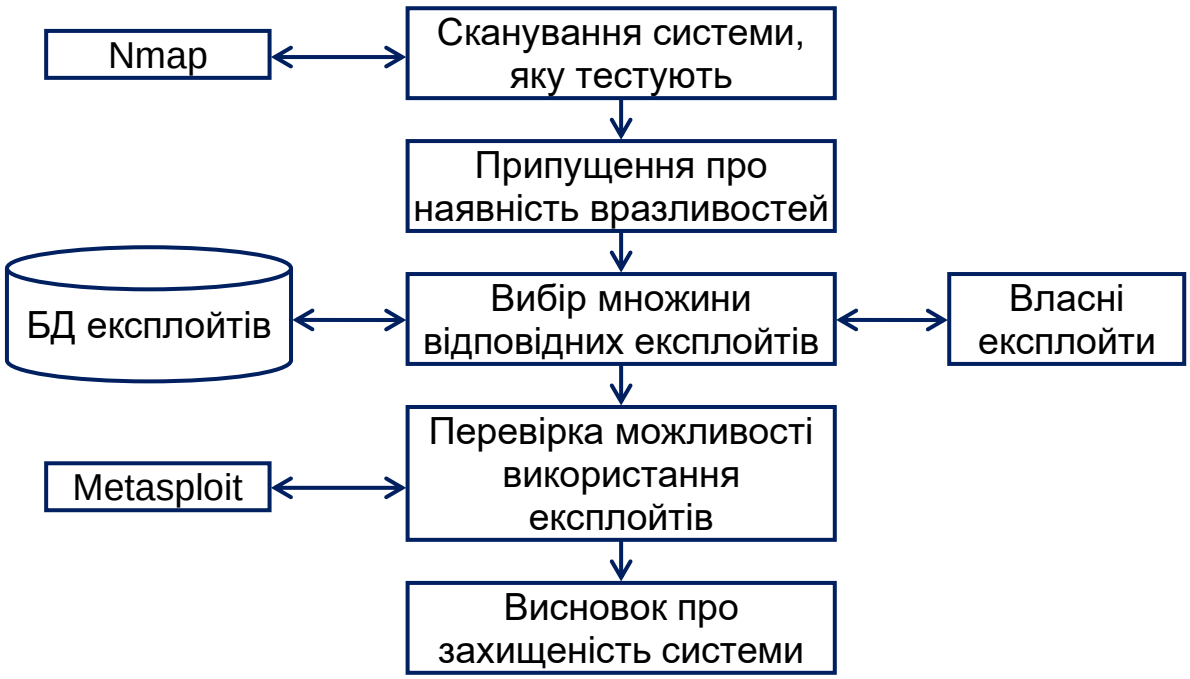


Рисунок 2.7 – Схема автоматизованого тестування на проникнення

Після цього, з БД вибираються підходящі експлойти і перевіряється їх працездатність. Якщо частина з них спрацьовує, то система вразлива. Якщо ж ні, то засоби захисту успішно відбивають атаки.

Наведена система підрозділяється на наступні взаємодіючі між собою компоненти (рисунок 2.8):

- підсистема збору інформації (ПЗІ);
- підсистема формування БД експлойтів (ПФБДЕ);
- база даних аналізатора;
- БД Metasploit Framework;
- підсистема аналізу захищеності (ПАЗ).

2.3.1 Підсистема збору інформації вразливостей

Завдання даної підсистеми є збір відомостей про цільову комп’ютерну систему. На вхід подається IP-адреса "жертви", після цього починається сканування. Основним інструментом цієї підсистеми виступає сканер Nmap.

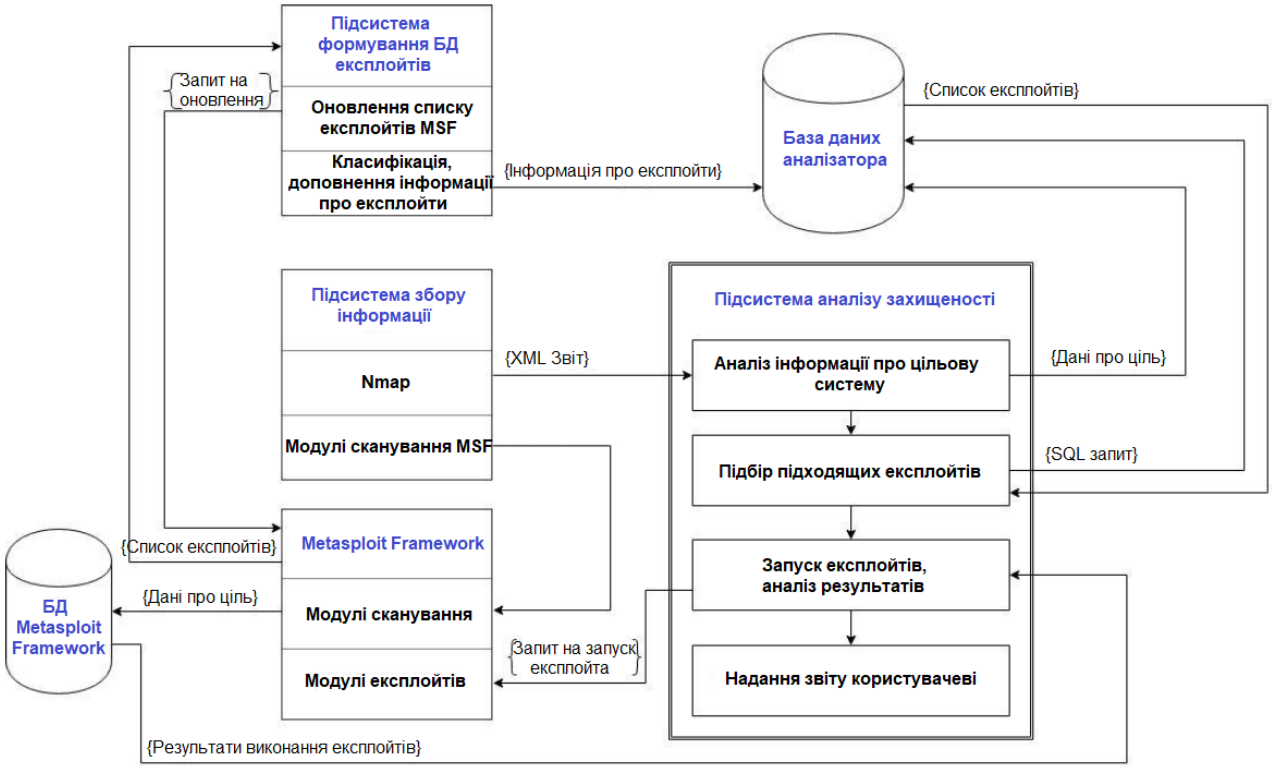


Рисунок 2.8 – Архітектура запропонованої системи тестування на проникнення

Nmap ("Network Mapper") – це утиліта з відкритим вихідним кодом, призначена для проведення підготовчих досліджень мережі або одиничних цілей і перевірки їх безпеки. Nmap посилає особливі, "неправильні" IP-пакети для того, щоб визначити, чи доступний хост, які порти відкриті, які служби (назва, версія) запущені на них, які засоби захисту використовуються. Також, Nmap дозволяє визначати операційну систему хоста (OS finger printing) аж до сімейства, версії, мови, сервісного пакета і архітектури.

На виході Nmap видає список хостів, які були просканувати і додаткову інформацію, яку вдалося витягти, для кожного з них. Найбільш суттєвою інформацією крім версії ОС є список "значущих" портів. У ньому міститься номер порту, ім'я служби, протокол і стан порту.

Результати найпростішого сканування утилітою Nmap виглядають наступним чином (рисунок 2.9).

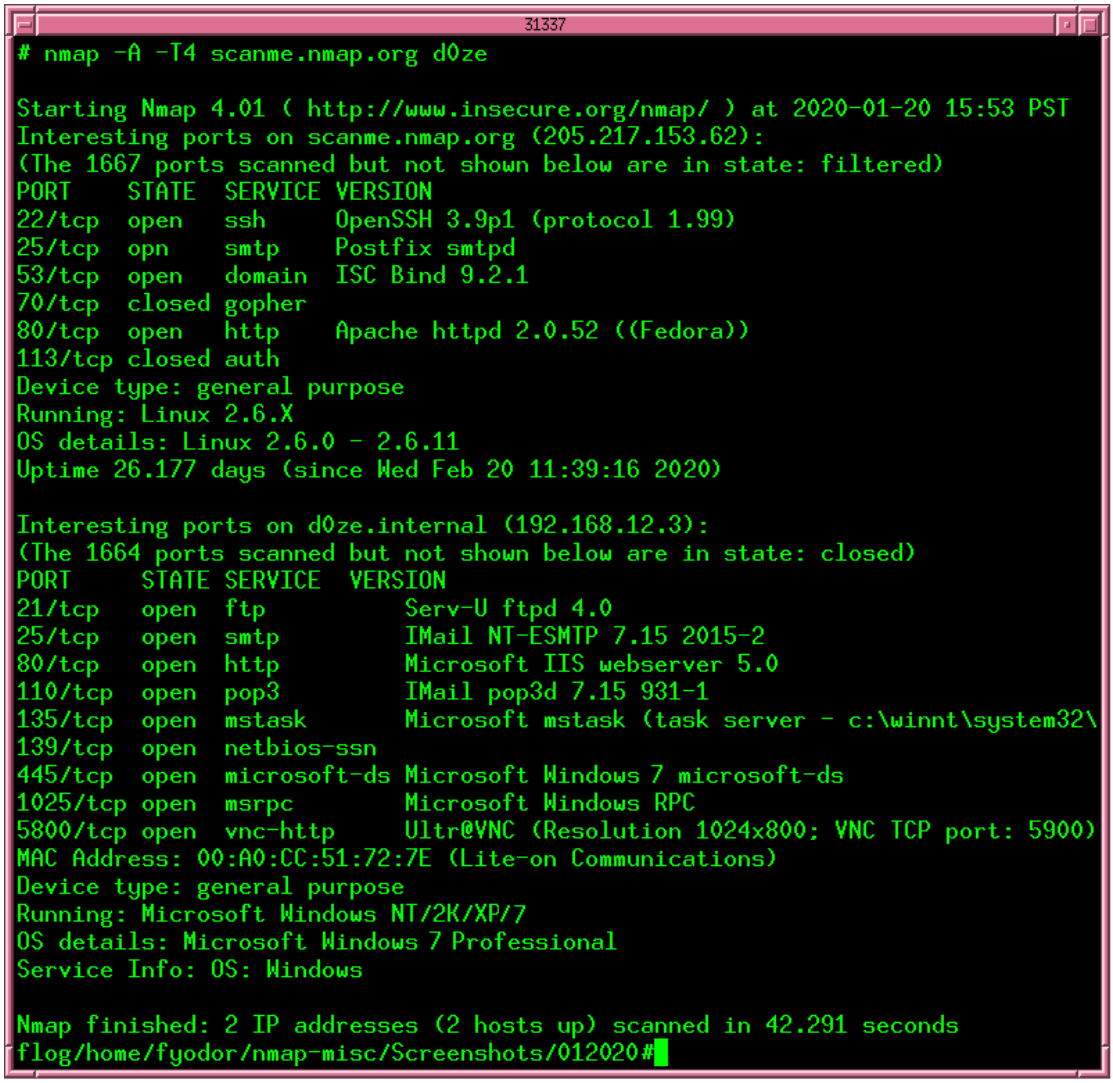


Рисунок 2.9 – Приклад збору відомостей утилітою Nmap

При проведенні пасивного збору інформації або не відбувається ніякої взаємодії з цільовою системою, або взаємодія здійснюється строго по заданих наперед правилам. Все, що має порушник – це будь-яка загальнодоступна інформація про хості і перехоплений трафік, відправлений "жертвою". Тому, пасивні методи спрямовані на різний аналіз цього трафіку. З явних переваг слід вказати абсолютну скритність таких методів, тобто адміністратор досліджуваного хоста не може виявити сам факт сканування. Однак, подібні техніки сильно обмежені і не можуть надати всю необхідну інформацію про систему.

Навпаки, при активному скануванні, відбувається відправка всіляких запитів на атакується хост і аналіз отриманих відповідей. В результаті можна отримати більш детальну і достовірну інформацію, проте велика ймовірність виявлення і блокування файерволом.

Також Nmap має кілька різних способів обходу засобів захисту інформації, таких як мережеві екрани і антивіруси.

Крім перерахованих вище методів сканування портів, існує ще багато інших. У розробленому програмному комплексі можливе використання всіх технік сканування, пропонованих Nmap.

Варто зазначити, що в залежності від ситуації, застосовуються різні типи збору інформації ("розвідку"): активну і пасивну.

Крім отримання даних про цільову системі за допомогою засобів Nmap, підсистема збору інформації використовує деякі можливості Metasploit Framework для підтвердження цієї інформації.

В MSF (Metasploit Framework) є набір модулів, призначених для проведення сканування віддалених хостів. Наступні модулі є найпоширенішими і застосовуються в реалізованої підсистемі:

- SYN сканування – для даного типу сканування використовується модуль auxiliary/scanner/portscan/syn;
- TCP сканування – модуль auxiliary/scanner/portscan/tcp;
- SMB Version сканування – визначення версії ОС на підставі інформації про службу smb. Модуль – auxiliary/scanner/smb/smb_version;
- MS SQL сканування – auxiliary/scanner/mssql/mssql_ping. Визначення відкритих портів, використовуваних Microsoft SQL Server.

Для кожного сканера можна задавати різні параметри, такі як IP-адреса досліджуваного хоста, інтерфейс, порти, які потрібно просканувати і т.д.

Звіт про скануванні Nmap вивантажується в форматі XML, потім дані з цього звіту групуються і заносяться в базу даних для подальшого використання. Результати сканування модулями MSF заносяться в БД автоматично.

2.3.2 Підсистема формування БД експлойтів

Основне завдання даного компонента – складання бази даних експлойтів на основі списку експлойтів MSF. Роботу цієї підсистеми можна описати наступними кроками.

Оновлення списку експлойтів MSF по Internet. Сервера Metasploit відправляється запит з перевіркою доступних оновлень експлойтів. У разі якщо такі оновлення присутні, відбувається завантаження і розпакування нових експлойтів.

Визначення портів і протоколів. Для кожного експлойта, доступного для запуску через Metasploit здійснюється пошук відповідних портів і протоколів. Оскільки будь-який експлойт спрямований на вразливість в якійсь конкретній службі, можна визначити, який порт відповідає даним експлойтів.

Для цього використовується документ "Service Name and Transport Protocol Port Number Registry RFC 6335", В якому наведено список сервісів і відповідних їм портів. Також цей список доповнюється інформацією з утиліти Nmap (Well Known Port List). Це файл "nmap-services", який містить перелік назв портів, їх номерів і протоколів. Також для кожного порту є дані про те, з якою ймовірністю він буде відкритий на досліджуваній системі. Даний список регулярно оновлюється і містить досить актуальну інформацію. В результаті, для кожного експлойта, обраного з Metasploit відбувається спроба поставити у відповідність порт і протокол. Якщо для експлойта не прописано в явному вигляді, на яку службу він спрямований, проводиться автоматичний аналіз його докладну розповідь, і визначення служби. Потім підбір портів повторюється.

Крім того, частина портів визначається за допомогою парсинга вихідного коду експлойтів. У деяких з них міститься вказівка портів, використовуваних експлойта за замовчуванням.

Після отримання даних про використовувані експлойта портах,

здійснюється угруповання експлойтів за типом операційної системи, для якої вони призначені (Windows, Linux, Unix) і по службам.

Вся доступна інформація про експлойтів (ОС, служба, ім'я, порти, протоколи, ранг, опис та інше) заноситься в БД PostgreSQL.

Таким чином, підсистема формування БД експлойтів займається створенням, оновленням і підтримкою БД експлойтів, яка містить всю необхідну інформацію для вибірки потрібних експлойтів для певної мети.

2.3.3 Підсистема формування БД експлойтів

Зберігає основну інформацію про експлойтів, що перевіряється хості, відкритих портах, службах і так далі. Головна мета даної БД – надавати доступ до даних про всі доступні експлойтів. З неї можна отримати список експлойтів, придатних, наприклад, для атаки хоста з ОС Windows 7 Professional, з відкритими портами 80 і 445. Крім того, можна дізнатися номер Вразливості, яку використовує той чи інший експлойт, посилання на опис цієї вразливості в різних глобальних базах вразливостей (таких як CVE), чи є він активним або пасивним, який у нього ранг та іншу корисну інформацію.

2.3.4 Підсистема аналізу захищеності та Metasploit Framework

Metasploit Framework використовується як джерело експлойтів і як інтерфейс для їх конфігурації, запуску, а також аналізу результатів їх виконання.Взаємодія з MSF здійснюється в автоматичному режимі, через консоль MSF.

Підсистема аналізу захищеності є найважливішою в програмному комплексі. В її завдання входить взаємодія з іншими підсистемами (підсистема збору інформації, формування БД експлойтів, MSF) і з БД, в якій зберігаються дані про цільову системі і експлойти, аналіз інформації про

досліджуваний хості, отриманої від підсистеми збору інформації, складання моделі цієї системи, вибір відповідних експлойтів, побудова сценаріїв атак, відправка команд на запуск експлойтів в MSF, оцінка результатів виконання атак, прийняття рішення про подальші дії і так далі.

Розглянемо більш детально роботу даної підсистеми. Всі кроки виконуються в автоматичному режимі, тобто участь користувача не потрібно.

Початок роботи підсистеми аналізу захищеності полягає в скануванні цільового хоста для формування його моделі. Для цього, ПАЗ посилає команду ПСИ із зазначенням IP-адреси цілі. ПСИ за допомогою утиліти Nmap і модулів сканування MSF проводить сканування мети і повертає звіт з результатами, що містить опис версії ОС хоста, встановленого сервісного пакету, мови, і інших даних, а також список відкритих портів і відповідних їм служб. Всю цю інформацію ПАЗ заносить в свою БД, в таблиці з певною структурою.

Паралельно з цим кроком можливий запуск оновлення БД з експлойта за допомогою підсистеми формування БД експлойтів. Відбувається з'єднання з сервером Metasploit і завантаження оновленого списку експлойтів. Потім, ПФБДЕ виконує зіставлення експлойтів і портів і створює БД експлойтів з усією доступною інформацією про них.

Після того, як модель атакується хоста складена, і БД експлойтів містить актуальну інформацію, можна переходити до підбору експлойтів. для цього ПАЗ генерує складний SQL-запит до БД і у відповідь отримує вибірку з відповідними для конкретної системи експлойта. При виборі експлойтів враховуються такі параметри:

- ОС цільового хоста, її версія, мова, сервісний пакет і будь-яка інша інформація про нього, отримана на етапі сканування;

- номера портів на хості, що мають статус "open". Подібний статус означає, що якась служба запущена і готова до прийому пакетів з даного порту. А значить, якщо в цій програмі є відома вразливість і в БД міститься експлойт, написаний для неї, то можна успішно провести атаку на систему;

- ім'я та версія служби, запущеної на відкритому порте. За цими даними також можна підібрати експлойти.

В результаті цих дій ПАЗ визначає обмежений список експлойтів, теоретично відповідних аналізованої системі. Далі необхідно впорядкувати їх і перевірити, які дійсно спрацюють і отримають доступ до системи, а які ні.

Сортування експлойтів відбувається по їх рангу, присвоєного Metasploit. Для аналізу беруться тільки ті експлойти, ранг яких вище або дорівнює 400 (GoodRanking), тобто це ті експлойти, для яких є відомості про системи і ПЗ, на які вони націлені. Експлойти з більш низьким рангом брати недоцільно, так як вони успішно спрацьовують лише в рідкісних випадках.

Крім рангу враховується "агресивність" експлойта. Все експлойти в MSF діляться на активні і пасивні. У випадку з пасивними експлойта, необхідно, щоб виповнилося одне або кілька умов на стороні атакується хоста, щоб Вразливість спрацювала. Тобто експлойт чекатиме, поки користувач на машині "жертві" виконає певні дії. Такі експлойти ПАЗ не використовує через те, що час їх виконання дуже непередбачувано, а аналіз системи бажано провести прямо зараз, а не чекати, поки користувач, наприклад, відкриє браузер. Активні ж експлойти спрацьовують (або ні) відразу після їх запуску. Саме вони ще й називаються "Агресивними". Подібне обмеження на характер експлойтів викликаний тим, що реалізовується система повинна бути повністю автоматизованої.

Далі, коли сформований відсортований список експлойтів для аналізованої системи, починається процес їх конфігурації. Для кожного експлойта задаються необхідні параметри, опції, настройки.

Основним параметром будь-якого експлойта є його корисне навантаження. Корисне навантаження – це частина експлойта, яка виконує дії, заради яких він і запускався, наприклад, копіювання даних, видалення даних, завантаження шкідливого ПЗ, відкриття з'єднання із зараженою машиною, запуск оболонки ОС та інші. Було прийнято рішення в якості корисного навантаження для всіх експлойтів вибирати Meterpreter.

Meterpreter (MP) – це розширюване, гнучке, уніфіковане корисне навантаження, яке надає широкі можливості пост-експлуатації системи в разі вдалого запуску експлойта. Зазвичай в якості корисного навантаження виступають shell коди, однак Meterpreter має ряд переваг.

По-перше, при запуску оболонки ("shell") ОС створюється окремий процес, отже, його легко виявити. По-друге, більшість СВВ досить ефективно знаходять шеллкоди по трафіку, що передається ними атакуючому, так як всі команди пересилаються у відкритому вигляді. По-третє, процес може бути обмежений командою chroot. Chroot – це операція зміни кореневого каталогу в Unix-подібних ОС. Програма, запущена з зміненим кореневим каталогом, матиме доступ тільки до файлів, що містяться в даному каталозі. Нарешті, оболонки сильно різняться в залежності від ОС, як за форматом команд, так і по набору можливостей. Розробники Meterpreter вирішили всі ці проблеми.

Проблема з породженням нового процесу відсутня, так як Meterpreter є багатоступеневим шеллкодом. Після успішного виконання експлойта здійснюється завантаження MP у вигляді DLL бібліотеки. Потім він розміщується в адресному просторі процесу, вразливість якого була проексплуатовано. Далі відбувається запуск MP на виконання у вигляді нового потоку. Таким чином,

MP і його розширення не створюють новий процес, а працюють в контексті проексплуатувати процесу у вигляді DLL бібліотеки.

Оскільки MP включає в себе досить багато можливостей по взаємодії з ОС, наприклад, завантаження і вивантаження файлів, зміна файлової системи і реєстру та інші, труднощів з chroot і доступністю стандартних функцій і програм не виникає. Також є можливість завантажувати власні DLL бібліотеки з реалізованими функціями, які потрібні були в якомусь конкретному випадку.

Ще одна відмінна риса Meterpreter – можливість міграції з процесу на процес. Таким чином, наприклад, при закритті користувачем атакується

машини процесу, в контексті якого працював MP, достатньо надіслати команду "migrate" і відбудеться зміна контексту. Можна також мігрувати в який-небудь системний процес, наприклад, explorer.exe, тоді ймовірність того, що він завершиться, сильно зменшиться. Оскільки спілкування з MP відбувається весь час через один і той же сокет, з'єднання не буде перервано. У випадку з shell кодом, можливості міграції немає.

Оскільки MP працює тільки з пам'яттю процесу і не здійснює записи на жорсткий диск комп'ютера, багатьом системам виявлення вторгнень і антивірусів виявити його не вдається. Крім того, команди MP передає в зашифрованому вигляді, тому детектувати їх не так просто.

Таким чином, після закінчення роботи ПАЗ можна дізнатися, скільки і які експлойти успішно виконалися і отримали доступ до системи, а скільки і які ні. Статистика виводиться користувачеві, із зазначенням вразливостей, які необхідно закрити на цільовій системі.

2.4 Методика оцінки захищеності комп'ютерної мережі за рахунок використання експлойтів при тестуванні на проникнення

При реалізації програмного комплексу для аналізу захищеності ІС використовувався Metasploit Framework. С допомогою нього відбувається запуск експлойтів і подальша експлуатація вразливостей.

Metasploit Project – це проект, створений для надання інформації про вразливості, що надає кошти для створення сигнатур для систем виявлення вторгнення (СВВ), написання та тестування експлойтів.

Найвідоміший продукт – Metasploit Framework (MSF) – безкоштовна платформа, призначена для створення, налагодження та запуску експлойтів. Крім цього, проект включає в себе базу shell кодів.

Можливості MSF досить широкі. Платформа надає інструмент для створення, тестування і виконання експлойтів. Для обраного конкретного експлойта можна задати корисне навантаження (payload), в залежності від

якої, в разі успішного виконання експлойта, буде вчинено ту чи іншу дію в атакується системі, наприклад, установка shell сервера (система віддаленого доступу до робочого столу комп'ютера). Також можна зашифрувати shell код, щоб приховати атаку від СВВ і систем протидії вторгненням. Metasploit Framework сумісний з деякими утилітами-сканерами, такими як Nmap і Nessus. Можна завантажувати в MSF звіти з результатами сканування хостів і використовувати цю інформацію для вибору відповідних експлойтів.

Самий базовий сценарій атаки за допомогою MSF складається з наступних кроків (рисунок 2.10):

- вибір і конфігурація експлойта.
- перевірка придатності даного експлойта для цільової системи (доступно не для всіх експлойтів).
- вибір і настройка корисного навантаження.
- вибір алгоритму шифрування, щоб СВВ не виявлено атаку.
- виконання експлойта.

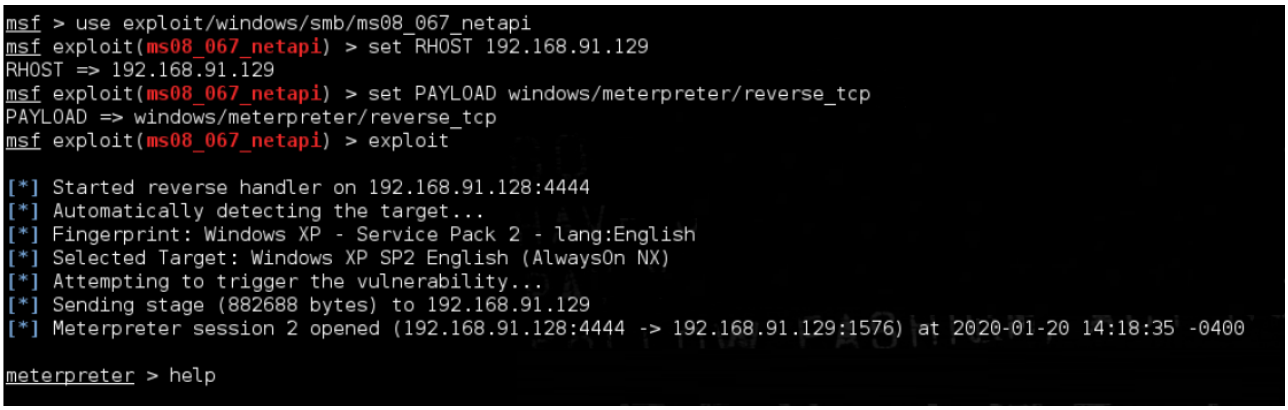


Рисунок 2.10 – Приклад успішного запуску експлойта

Metasploit Framework має модульну структуру: експлойти, корисні навантаження, мережеві сканери – все це модулі. Це дозволяє поєднувати будь-які модулі один з одним (наприклад, для експлойта можна вибрати одну з сотень варіантів корисного навантаження).

Metasploit можна встановити на Unix (в тому числі і на Linux і Mac OS X) і на Windows.

Ранжування експлойтів в Metasploit. Кожному модулю з експлойтів присвоєно ранг, заснований на потенційно можливий вплив на цільову систему. Значення рангів можуть бути наступними.

ExcellentRanking – експлойти ніколи не порушують працездатність сервісу. До таких експлойтів відносяться SQL ін'єкції, RFI, LFI. Ніяким експлойтів, що викликає пошкоджено пам'ять, не повинен бути привласнений такий ранг. Можливі поодинокі винятки (WMF Escape ()).

GreatRanking – експлойти призначені для конкретних систем, і або вміють автоматично перевіряти, підходить мета для даного експлойта чи ні, або використовують специфічний для додатка адреса повернення після перевірки версії цього додатка.

GoodRanking – експлойти призначені для конкретних систем і конкретного ПЗ.

NormalRanking – експлойти досить надійні, однак, залежать від конкретної версії ПЗ і не можуть автоматично визначати цю версію.

AverageRanking – експлойти ненадійні і складно експлуатовані.

LowRanking – експлойти практично неможливо використовувати для поширених платформ.

ManualRanking – експлойти нестабільні або складні у використанні, зазвичай представляють собою DoS атаки. Також цей ранг використовується, коли експлойт не може бути використаний без настройки користувачем (наприклад, php_eval).

Типи модулів в MSF. В Metasploit все (скрипти, файли, програми) є модулями. Виділяють 6 типів модулів:

- auxiliary – модулі, що допомагають атакуючому виконувати різні завдання, такі як сканування портів, визначення версій, аналіз мережевого трафіку;
- exploit – модулі, що містять експлойти, тобто код, який використовує

якусь вразливість в системі і дозволяє виконати корисне навантаження, наприклад, переповнення буфера або обхід автентифікації;

- payload – модулі, що містять корисну навантаження, тобто те, що має бути виконано відразу після успішного виконання експлойта, наприклад, встановлення віддаленого з'єднання, запуск сеансу meterpreter або виконання будь-яких системних команд;

- post – різні програми, які можуть бути запущені після успішного експлойтірованія і встановлення віддаленого з'єднання, наприклад, збір паролів, установка програм-шпигунів, скачування файлів і так далі;

- encoder – програми, що виконують шифрування корисного навантаження для захисту від виявлення захисними засобами;

- nop – генератори NOP. NOP – це інструкція на мові Асемблер, яка нічого не робить. Машинний код даної інструкції різниться для кожного типу архітектури системи. Зазвичай NOP інструкції використовуються для приведення розміру виконуваних файлів до визначеного розміру.

MSF можна використовувати під час проведення Pent test для створення звітів разом з іншими системами автоматичного виявлення вразливостей. За допомогою Metasploit можна визначити, чи є знайдені вразливості дійсно небезпечними і чи можна їх використовувати для отримання доступу до системи.

Крім того, MSF можна використовувати для тестування нових експлойтів. Для цього потрібно налаштувати локальний сервер з вразливістю, яку використовує експлойт. Таким чином, можна швидко перевірити ефективність створеного експлойта.

Також, MSF прекрасно підходить для перевірки коректності налаштувань різних СВВ на випадок мережевих атак.

Щоб почати працювати з Metasploit Framework, необхідно запустити дві служби: postgresql і metasploit. Служби запускаються командою "service <Ім'я_служби> start ". Для того щоб можна було легко стартувати їх з основної програми, був написаний невеликий скрипт (приклад 2.1):

```
#!/Bin/bash
service postgresql start
service metasploit start
```

Приклад 2.1 – Запуск служб postgresql і metasploit в MSF

Створення будь-яких сторонніх процесів в Java здійснювалося за допомогою класу ProcessBuilder. Так, наприклад, щоб запустити наведений вище скрипт, виконуються наступні команди (приклад 2.2):

```
ProcessBuilder pb = new ProcessBuilder ("/root/services.sh");
Process process = pb.start();
process.waitFor();
```

Приклад 2.2 – Запуск сторонніх процесів в MSF

Запуск цього скрипта відбувається кожного разу при старті програмного комплексу. Процес реалізації був також розділений по підсистемах програмного комплексу. Почнемо опис з підсистеми формування БД експлойтів.

2.4.1 Підсистема формування бази даних експлойтів

Першим кроком при створенні БД експлойтів є оновлення списку експлойтів Metasploit Framework. Процес оновлення полягає в оновленні самого MSF і виконується за допомогою команди "msfupdate". Виклик команди знову ж відбувається через методи класу ProcessBuilder.

Після того, як MSF оновився, з нього вивантажується список експлойтів. Робиться це в такий спосіб: оскільки експлойти зберігаються в певній директорії ("/usr/share/metasploit-framework/modules/exploits/-<назва_ОС>/<назва_служби> /"), можна рекурсивно відкривати папки з експлойта і заносити в таблицю їх імена, ОС і служби, для яких вони призначені.

Крім того, для кожного експлойта відбувається спроба зіставити йому порт і протокол. Для цього готується спеціальний файл зі списком служб і відповідних їм портів і протоколів.

Файл формується з документа "Service Name and Transport Protocol Port Number Registry". У ньому містяться такі дані: ім'я служби, номер порту, протокол транспортного рівня, опис служби. Крім цього документа, файл доповнюється записами з файлу "nmap-services", використовуваного утилітою Nmap. Він також містить перелік портів, служб і протоколів. Його структура досить проста. У ньому є три колонки, розділені пробілами. У першій зберігається ім'я служби, в другій – номер і протокол, розділені знаком "/", в третій-ймовірність того, що даний порт буде відкритим на досліджуваній системі.

В результаті, з двох джерел збирається один, загальний файл "services.txt". Всього в ньому близько 12000 записів. Нижче наведено фрагмент цього файлу (рисунок 2.11). З цього файлу і відбувається визначення портів для експлойтів.

Рисунок 2.11 – Фрагмент файлу "services.txt"

Таким чином, після виконання всіх цих дій, виходить таблиця зі стовпцями: "ОС", "Служба", "Ім'я експлойта", "Порт", "Протокол". Потім інформація про кожного експлойтів доповнюється з таблиць MSF - "module_details" і "module_platforms". У них зберігаються ранги експлойти, їх опису, типи (активний / пасивний), платформи, для яких вони написані та інше. Далі отримана таблиця заноситься в БД експлойтів (рисунок 2.12).

Спілкування з СУБД PostgreSQL відбувається за допомогою драйвера JDBC. Зв'язок відбуватися за такою схемою: є єдиний інтерфейс, до якого підключається драйвер для роботи з PostgreSQL, після цього можна передавати запити БД.

	id	os	service	name	fullname	port	protocol	file	refname	textname	rank	description	privileged	disclosure_dat	default_ta	default	stance	ready
	integer	text	character vary	text	text	character	character	text	text	text	integer	text	boolean	timestamp with	integer	text	character	boolean
391	1014	linux	http	mutiny_subnetmask_exec	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	Mutiny F	600	This mod	TRUE	2012-10-22	1		passive	TRUE
392	1014	unix	http	mutiny_subnetmask_exec	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	Mutiny F	600	This mod	TRUE	2012-10-22	1		passive	TRUE
393	974	php	http	nas4free_php_exec	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	NAS4Free	500	NAS4Free	TRUE	2013-10-30	0		aggressi	TRUE
394	990	windows	http	netwin_surgeftp_exec	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	Netwin S	400	This mod	FALSE	2012-12-06			aggressi	TRUE
395	990	unix	http	netwin_surgeftp_exec	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	Netwin S	400	This mod	FALSE	2012-12-06			aggressi	TRUE
396	1023	unix	http	op5_license	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	OP5 lice	600	This mod	TRUE	2012-01-05	0		aggressi	TRUE
397	1065	linux	http	op5_welcome	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	OP5 welc	600	This mod	TRUE	2012-01-05	0		aggressi	TRUE
398	1065	unix	http	op5_welcome	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	OP5 welc	600	This mod	TRUE	2012-01-05	0		aggressi	TRUE
399	1044	java	http	openfire_auth_bypass	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	Openfire	600	This mod	TRUE	2008-11-10	0		aggressi	TRUE
400	1044	linux	http	openfire_auth_bypass	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	Openfire	600	This mod	TRUE	2008-11-10	0		aggressi	TRUE
401	1044	windows	http	openfire_auth_bypass	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	Openfire	600	This mod	TRUE	2008-11-10	0		aggressi	TRUE
402	983	unix	http	openmediavault_cmd_exec	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	OpenMedi	600	OpenMedi	TRUE	2013-10-30	0		aggressi	TRUE
403	983	linux	http	openmediavault_cmd_exec	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	OpenMedi	600	OpenMedi	TRUE	2013-10-30	0		aggressi	TRUE
404	1063	php	http	openx_backdoor_php	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	OpenX Ba	600	OpenX Ad	FALSE	2013-08-07	0		aggressi	TRUE
405	1056	java	http	opmanager_socialit_file_upload	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	ManageEr	600	This mod	TRUE	2014-09-27	0		aggressi	TRUE
406	1072	windows	http	oracle_reports_rce	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	Oracle F	500	This mod	FALSE	2014-01-15	0		aggressi	TRUE
407	1072	linux	http	oracle_reports_rce	exploit/multi/ht	80/8008	sctp/tcp	/usr/sh	multi/ht	Oracle F	500	This mod	FALSE	2014-01-15	0		aggressi	TRUE

Рисунок 2.12 – Частина таблиці з експлойта

При закінченні формування/оновлення БД експлойтів користувачеві виводиться статистика експлойтів (скільки їх міститься в БД) (рисунок 2.13).

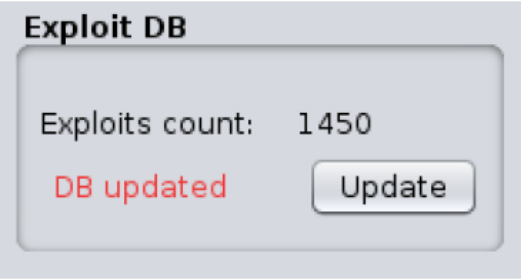


Рисунок 2.13 – Частина таблиці з експлойта

2.4.2 Підсистема збору інформації

На вхід підсистемі подається IP-адреса цільової системи і профіль сканування Nmap. Користувач може обрати такі варіанти, як показано на рисунку 2.14.

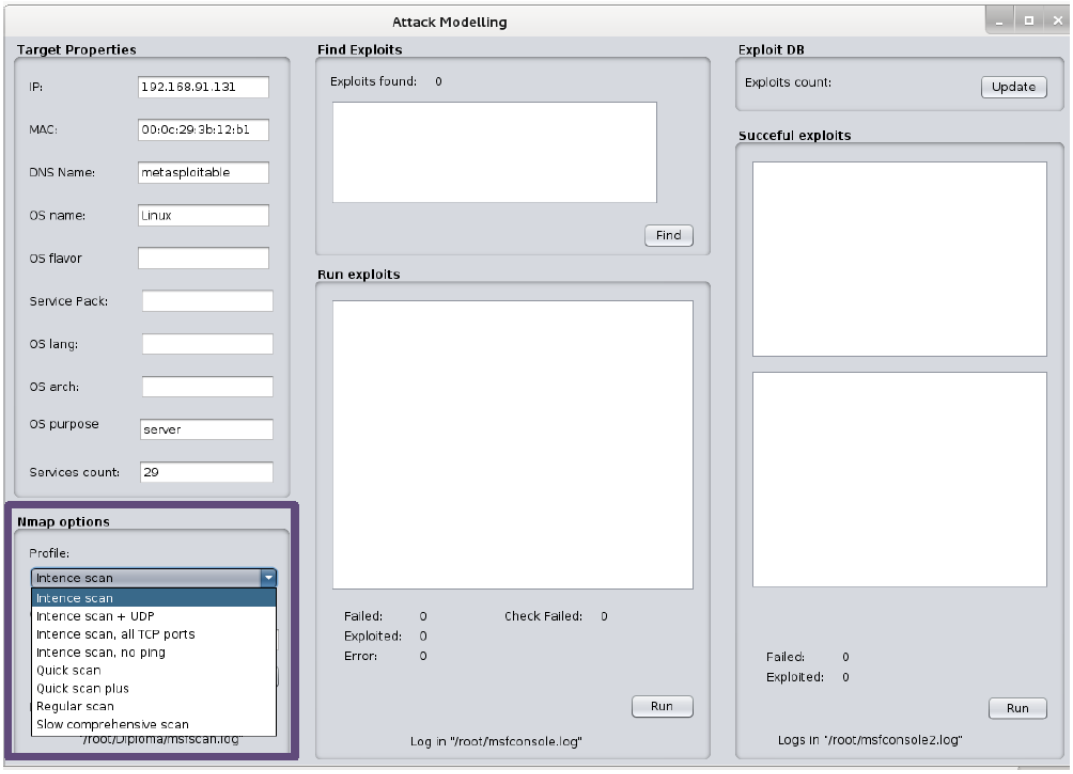


Рисунок 2.14 – Вибір режиму сканування Nmap

Параметрами в цьому випадку виступають:

- Intense scan – досить швидке сканування, перевіряється більшість TCP портів, визначаються запущені служби;
- Intense scan plus UDP – додається UDP і TCP SYN сканування;
- Intense scan, all TCP ports – перевірка всіх портів (з 1 по 65535). За замовчуванням перевіряється лише першу 1000 портів
- Intense scan, no ping – використовується опція -Pn, тобто передбачається, що хост точно активний і немає необхідності надсилати йому echo запити;

- Quick scan – дуже швидке сканування за рахунок того, що перевіряється тільки 100 основних TCP портів;

- Quick scan plus – те саме, що і "Quick scan", але ще додається розпізнавання версій запущених служб;

- Regular scan – всі параметри виставлені за замовчуванням (TCP SYN сканування, першу 1000 портів, ICMP echo запити для визначення стану хоста);

- Slow comprehensive scan – найдокладніший і тривалий варіант сканування.

Після вибору профілю для сканування утилітою Nmap, формується набір параметрів Nmap.

Наприклад, для Intense scan він буде таким: "-T4 -A -v -O". Потім ці параметри подаються на вхід Nmap і відбувається запуск сканування.

Дані з результатами (інформацією про вузол, відкритих портах і запущених службах) заносяться в БД (рисунок 2.15, 2.16).

	id [PK] serial	created_at timestamp	address inet	mac character varying(255)	c	name character	state character	os_name character	os_flavor character	os_sp character	os_lang	arch character	workspace_ integer	updated_a timestamp	purpose text	ir	c	s	v	note_coun integer	vuln_coun integer	service_count integer	host_detai integer	exploit_att integer	cred_coun integer
1	15	2020-01-	192.168.0	0:0c:29:9e:a2:a9	'	APO3WXP5	alive	Windows				x86	1	2015-05-	client					15	1	175	0	134	0

Рисунок 2.15 – Таблиця з даними про хості

42	46	15	2020-01-	135	tcp	open	msrpc	2020-01-	Microsoft Windows RPC
43	47	15	2020-01-	139	tcp	open	netbios-	2020-01-	''
44	48	15	2020-01-	445	tcp	open	microsof	2020-01-	Microsoft Windows microsoft-ds

Рисунок 2.16 – Таблиця з портами і службами

Після проведення сканування Nmap, проводиться дослідження хоста засобами Metasploit.

Виробляється запуск модуля auxiliary/scanner/portscan tcp – TCP сканування портів. Результати так само заносяться в БД.

По завершенні роботи підсистеми збору інформації у вікні програми заповнюються поля, пов'язані з цільовою системою (рисунок 2.17).

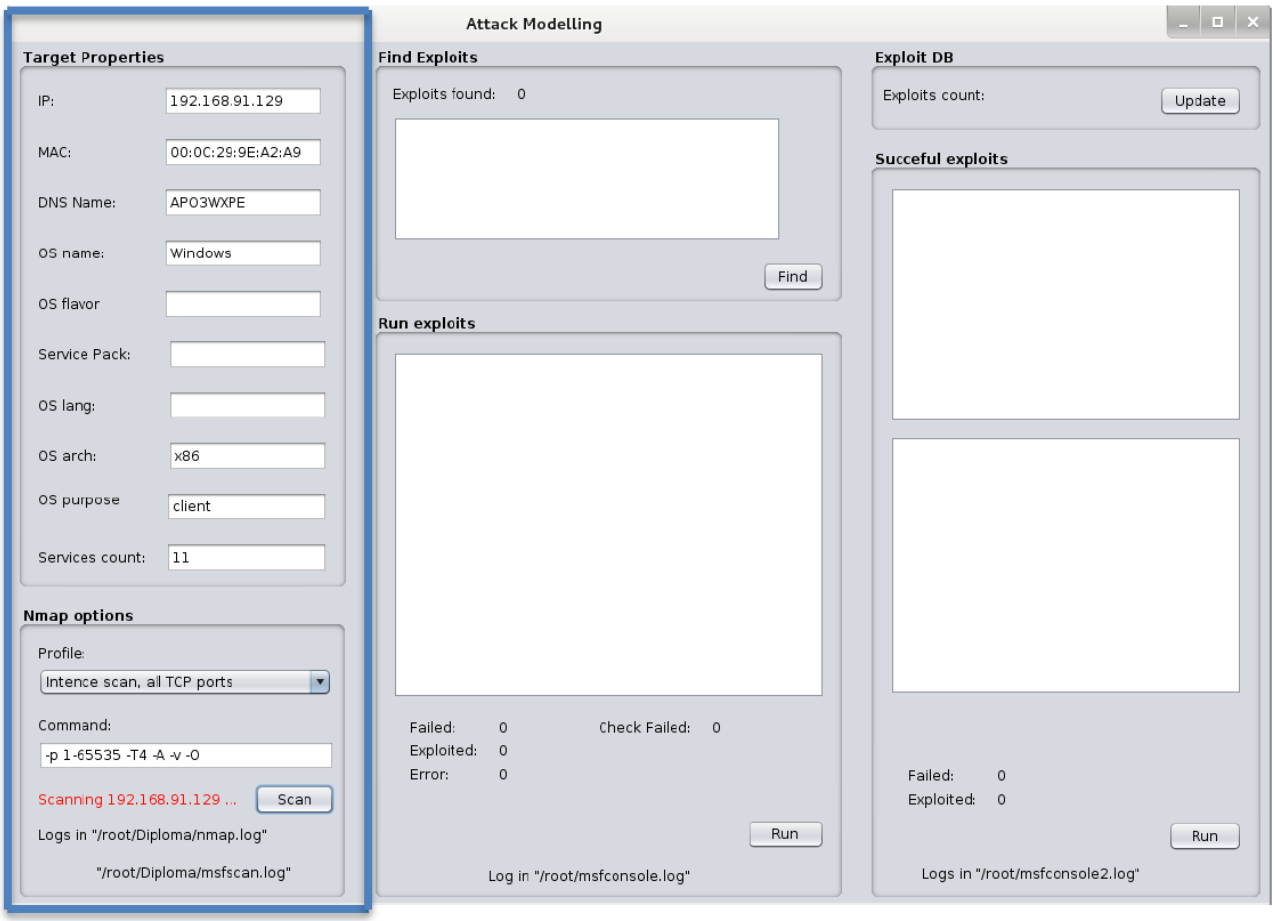


Рисунок 2.17 – Результати роботи підсистеми збору інформації в КМ

Вся підготовча робота на цьому етапі зроблено, можна переходити до основної частини – пошук експлойтів і спроба їх використання.

2.4.3 Підсистема аналізу захищеності

Підсистема аналізу захищеності повинна обмінюватися командами з Metasploit Framework, а точніше – з командним рядком MSF - msfconsole.

Варто відзначити, що при створенні будь-якого процесу (Nmap, msfconsole) проводиться перенаправлення його вихідного потоку в файл за

допомогою методу redirectOutput класу ProcessBuilder. Таким чином, можна легко вести протоколювання всіх процесів, які запускаються.

Після того, як буде складено список потенційно небезпечних для системи експлойтів, можна приступати до тестування. За це відповідає друга складова ПАЗ – функція запуску експлойтів. Спочатку відбувається створення окремого процесу консолі Metasploit. На вхід msfconsole подається заздалегідь створений файл з інструкціями установки параметрів експлойтів. Щоб не задавати параметри, наприклад, RHOST - IP-адреса цільової системи або LHOST -IP-адреса атакуючої системи, для кожного експлойта окремо, застосовується спеціальна команда "setg" - завдання глобальних параметрів. Таким чином, досить задати всі параметри експлойтів один раз, при запуску консолі MSF.

Щоб визначити, коли експлойт завершив свою роботу (успішно чи ні), проводиться підрахунок записів в таблиці exploit_attempts в БД Metasploit Framework. У цю таблицю Metasploit автоматично заносить всі спроби виконання експлойтів. Якщо число записів збільшилася, значить експлойт виконався. За певним полю в цій таблиці можна визначити, з яким результатом він завершився. Залежно від Якщо число записів збільшилася, значить експлойт виконався. За певним полю в цій таблиці можна визначити, з яким результатом він завершився. Залежно від Якщо число записів збільшилася, значить експлойт виконався. За певним полю в цій таблиці можна визначити, з яким результатом він завершився. Залежно від того, чи вдалося отримати доступ до системи чи ні, експлойт заноситься в один з двох списків і запускається наступний.

В результаті, користувач отримує звіт про проведене тестування системи. Формується перелік успішно виконаних експлойтів і відповідних їм вразливостей, які адміністратору системи необхідно ліквідувати. Також виводиться статистика по успішним і неуспішним експлойтів, скільки всього було протестовано і які (рисунок 2.18).

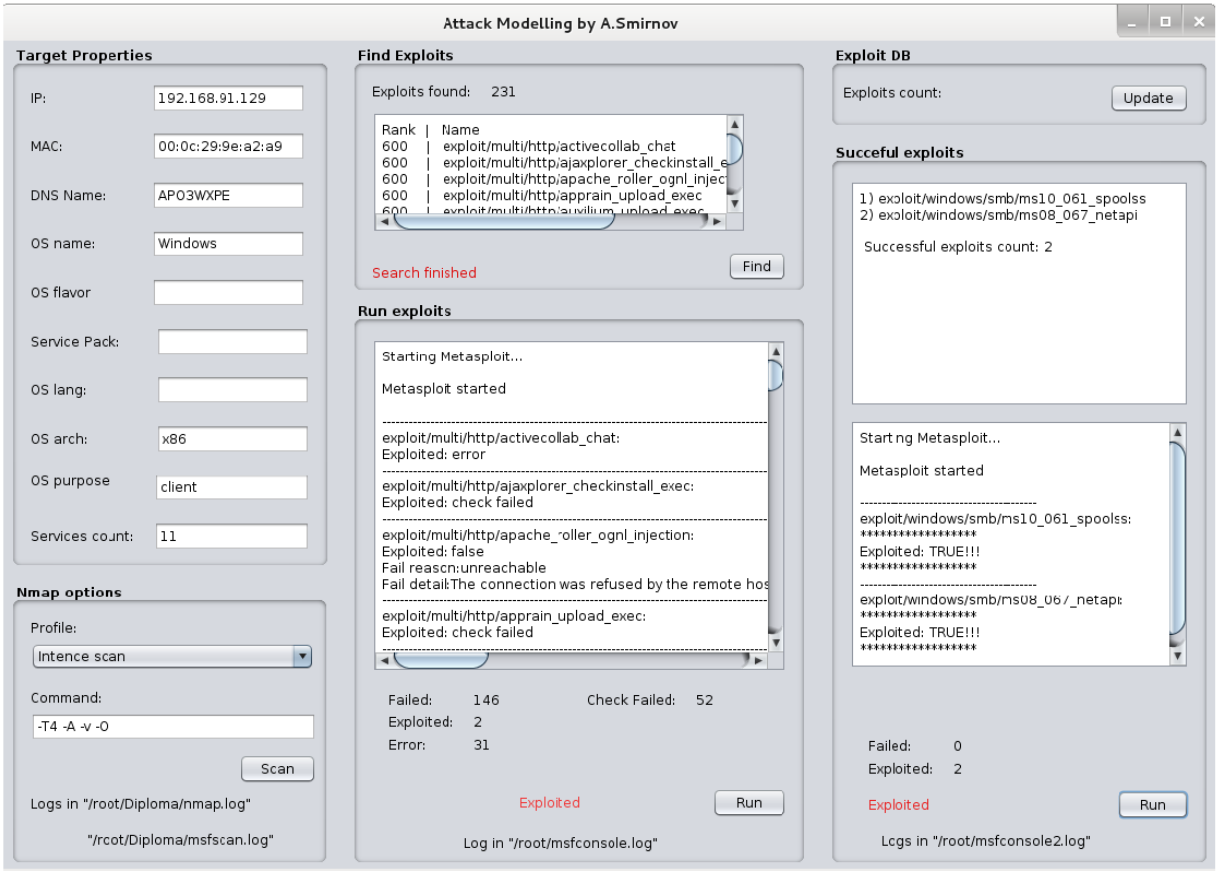


Рисунок 2.18 – Результати тестування операційної системи

2.5 Тестування на проникнення до хостів комп'ютерної мережі

Проведемо тестування декількох операційних систем (Windows, Linux, Mac OS). Частина з них спеціально призначена для тестування нових засобів безпеки, інші ж є звичайними системами. Результати наведені в таблиця 2.5.

Після установки засобів захисту жодному з раніше успішно виконаних експлойтів не вдалося отримати доступ до системи. Це пов'язано з тим, що всі, що запускаються експлойти загальнодоступні, а, отже, актуальні засоби захисту мають їх сигнатури в своїх БД.

Однак, існують методи обходу засобів захисту: по-перше, кодування корисного навантаження, по-друге, зміна сигнатури самого експлойта.

Таблиця 2.5 – Результати аналізу різних ОС на хостах комп'ютерної мережі

Операційні системи	Кількість виявлених запущених служб і відкритих портів	Кількість відібраних експлойтів для цільової системи	Кількість успішних експлойтів	Час виконання	
				Загальний, хв. сек.	Нормований, сек/exploit.
Windows 10	11/11	45/1450	1/1	3 хв 02 сек	3,75
Windows 7	10/10	39/1450	3/3	2 хв 30 сек	3,84
Windows Vista	9/9	33/1450	2/3	3 хв 06 сек	5,6
Windows XP SP3	9/9	231/1450	1/1	20 хв 02 сек	5,19
Windows Server 2008 R2	10/10	37/1450	2/2	2 хв 46 сек	3,98
Windows Server 2003	4/4	95/1450	3/4	5 хв 30 сек	3,45
Mac OS X High Sierra	4/4	37/1450	1/1	5 хв 30 сек	3,75
Mac OS X Sierra	5/5	65/1450	1/1	4 хв 27 сек	4,23
Linux Mint 17	2/2	84/1450	0/0	4 хв 34 сек	3,25
CentOS 7	1/1	8/1450	1/1	0 хв 41 сек	3,17
Metasploitable 2	30/30	283/1450	7/7	25 хв 45 сек	5,43

2.6 Висновки за розділом 2

Проведено аналіз вразливостей інформаційних систем та оцінка її критичності для комп’ютерної мережі Аналіз надав можливість визначити ефективний підхід і використати його з метою оптимізації параметрів систем захисту.

3 МЕТОДИ ТЕСТУВАННЯ КОМП'ЮТЕРНИХ МЕРЕЖ НА ПРОНИКНЕННЯ

3.1 Підходи до проведення тестування на проникнення

Крім навичок використання величезної кількості технік і інструментів, аудитор для реалізації pentest повинен розуміти всі нюанси технічної та організаційної складової ІБ, володіти навичками соціальної інженерії, дотримуватися певних методів:

- Open Source Security Testing Methodology Manual (OSSTMM);
- Information System Security Assessment Framework (ISSAF);
- NIST Special Publication 800-115: Technical Guide to Information Security Testing and Assessment;
- Penetration Testing Model;
- Payment Card Industry Data Security Standard (PCI DSS);
- Penetration Testing Execution Standard.

Також аудитор повинен визначити вектори атак, які можуть бути спрямовані насамперед на:

- користувачів корпоративних систем (соціальна інженерія);
- зовнішній периметр мережі (периметр IP-адрес і web-сайтів);
- бездротові мережі IEEE 802.11 (Wi-Fi), 802.15 (Bluetooth) і 802.16 (Wi-Max);
- переносні комп'ютери і мобільні пристрої.

За місцем розташування аудитора щодо мережевого периметра корпоративної системи pentest може бути внутрішнім, зовнішнім або комплексним (рисунок 3.1).

Зовнішній pentest передбачає насамперед тестування зовнішнього периметра мережі, web-сайтів і спецдодатків і т.д. Внутрішній – орієнтований головним чином на внутрішні ресурси.

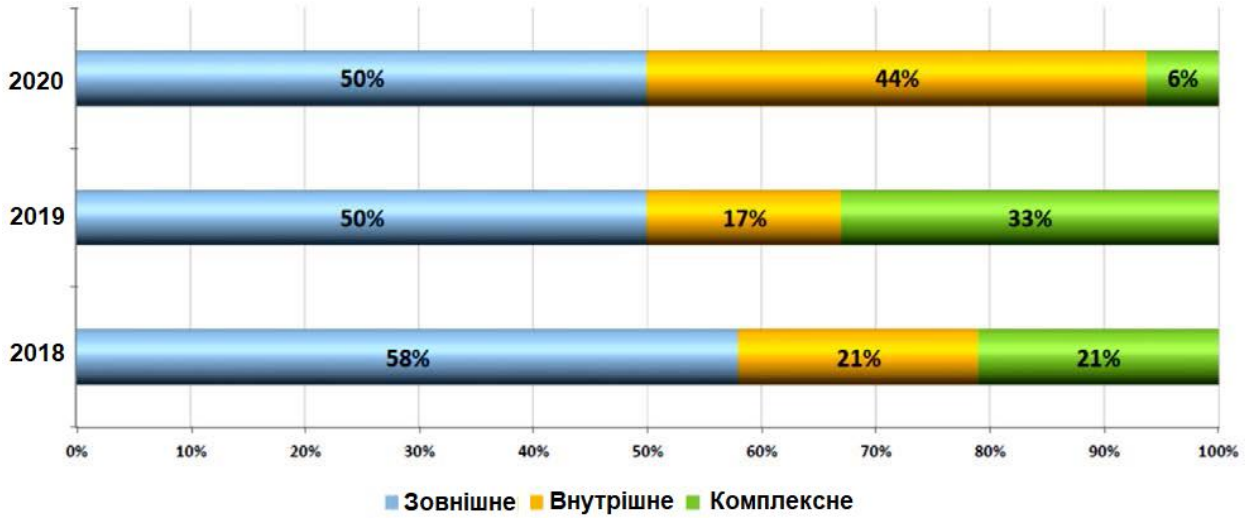


Рисунок 3.1 – Підходи до проведення pentest

3.2 Метод Open Source Security Testing Methodology

Методика Open Source Security Testing Methodology (OSSTMM) є досить формалізованим і добре структурованим документом для тестування мережі.

Документ має так звану "Карту безпеки" – візуальний показник безпеки. На карті (рисунок 3.2) вказуються основні області безпеки, які включають в себе набори елементів, які повинні бути протестовані на відповідність методиці:

- інформаційна кібербезпека;
- тестування процесу кібербезпеки;
- тестування технології Інтернет-кібербезпеки;
- тестування кібербезпеки каналів зв'язку;
- тестування кібербезпеки бездротових технологій;
- тестування фізичної кібербезпеки.

Рисунок 3.2. ілюструє розділи тестування безпеки з методики тестування безпеки з відкритим кодом. Це також показує, що кожен розділ перекривається і містить елементи всіх інших розділів.

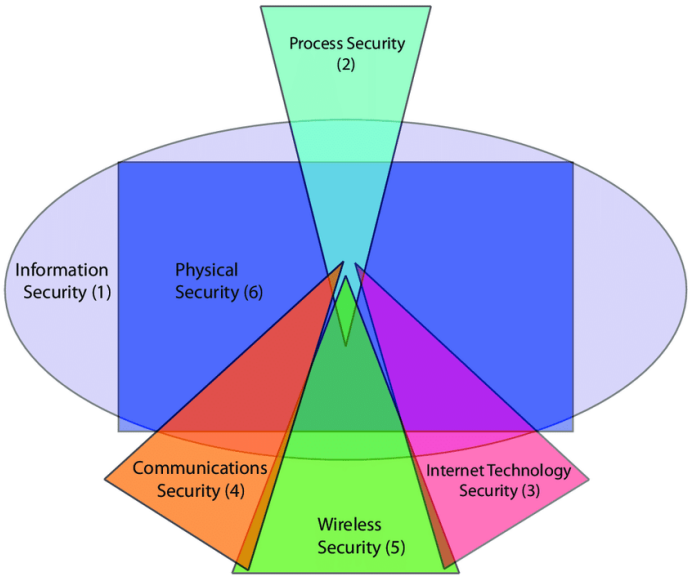


Рисунок 3.2 – Структура розділів методики OSSTMM

Приклад такої взаємодії розділів показаний на рисунку 3.3. В OSSTMM здійснюється перевірка на помилки:

- перевірка наявності втратою пакетів на маршруті до цільової мережі;
- вимірювання часу відгуку на пакет;
- вимірювання відносини прийом / відповідь по цільовій мережі;
- вимірювання кількості втрачених пакетів і отриманих відмов при зв'язку з цільовою мережею.

Мінусами методики вважається формалізованість і відсутність додаткового опису до вимог. Дану методику можна використовувати як на етапі оцінки продукту для можливості використання на підприємстві, так і на етапі розробки для перевірки окремих можливостей і функцій. Також методику можна використовувати, як шаблон розробки.

3.3 Метод Information Systems Security Assessment Framework

В якості альтернативи, Information Systems Security Assessment Framework (ISSAF) організовує дані навколо "критеріїв оцінки", кожен з яких був складений експертами в кожній сфері застосування рішень безпеки.

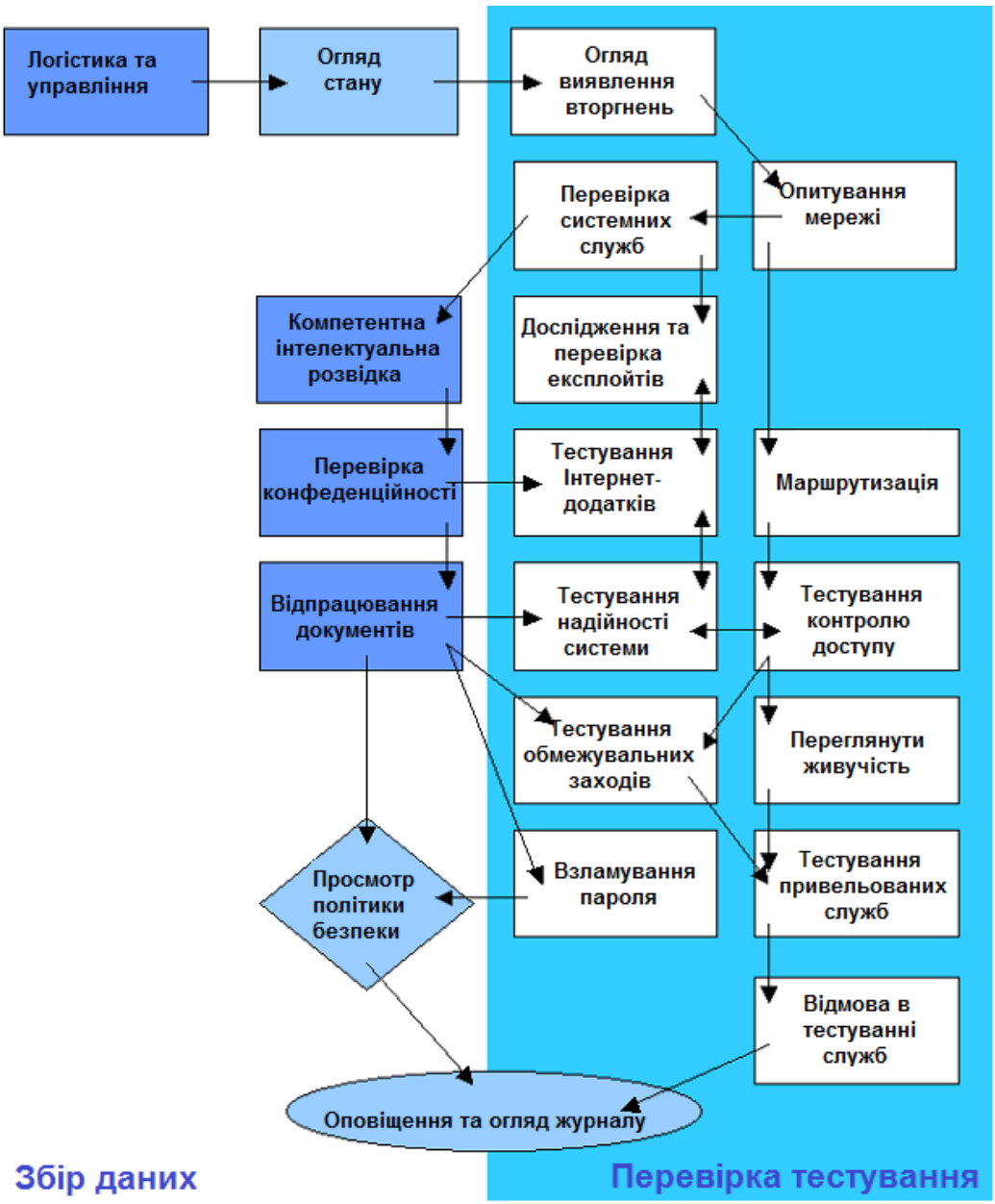


Рисунок 3.3 – Приклад взаємодії розділів в методиці OSSTMM

Payment Card Industry Data Security Standard (PCI DSS) був розроблений радою провідних компаній-емітентів кредитних і дебетових карт і він служить в якості керівництва для організацій, які обробляють, зберігають і передають дані про власників карт. Саме під цей стандарт був розроблений PCI-пентестінг.

Кількість методів і фреймворків досить велика, вони великі й досить різноманітні. Як вже було сказано, вибір між ними буде залежати від

розуміння потреб вашої компанії і знання необхідних стандартів безпеки. Але роблячи все правильно, ви зможете захищати свої системи набагато більш ефективно, заздалегідь знаючи, де і як вони можуть вийти з ладу. Безцінна інформація для тих, хто вміє нею користуватися. Розроблено Open Information Systems Security Group для наступних інструментів менеджменту і внутрішніх контрольних перевірок:

- оцінка політик і процедур кібербезпеки організації для звітності про їх відповідність індустріальним IT-стандартам, які можуть застосовуватися законам, а також нормативним вимогам;
- оцінка залежності бізнесу від інфраструктури IT-послуг;
- проведення оцінки вразливостей і тестів на проникнення для виділення вразливостей в системі, які можуть привести до потенційних ризиків інформаційних активів;
- вказівка моделей оцінки по доменах безпеки;
- знаходить та усуває неправильних конфігурацій;
- ідентифікація та рішення ризиків, пов'язаних з технологіями;
- ідентифікація та рішення ризиків, пов'язаних з персоналом або бізнес-процесами;
- посилення існуючих процесів і технологій;
- надання кращих практик і процедур для підтримки ініціатив безперервності бізнесу;
- документ охоплює величезну кількість питань, пов'язаних з інформаційною безпекою. Присутні глави, що описують оцінку безпеки;
- роутерів, антивірусних систем і багато іншого.

Документ охоплює величезну кількість питань, пов'язаних з інформаційною безпекою. Присутні глави, що описують оцінку безпеки МЕ, роутерів, антивірусних систем і багато іншого.

Також присутній глава "Оцінка безпеки міжмережевого екрана (МЕ)", де описується, які бувають МЕ, яким функціями вони повинні володіти і захист від чого вони не можуть надати.

Безпосередньо методологія включає в себе 4 етапи:

- визначення ME;
- визначення загальних неправильних конфігурацій;
- тестування загальних атак на ME;
- тестування продукції за специфічними питань.

Також в розділі додається докладні рекомендації з тестування. Описано не тільки утиліти, якими можна провести тестування, а й вказівки по їх використанню і які реакції можна отримати в результаті тестування з певними параметрами.

Дану методику рекомендується застосовувати для перевірки кінцевого продукту або перевірки загальної надійності мережі.

3.4 Метод NIST Special Publications 800-115

Методика NIST Special Publications 800-115 Technical Guide to Information Security Testing and Assessment створена і підтримується підрозділом NIST-CSRC, центром з комп'ютерної безпеки, який об'єднує фахівців федеральних служб, університетів, найбільших IT-компаній США. У документі присутні такі розділи як:

- огляд тестування і експертизи безпеки;
- огляд методів;
- визначення мети і техніки аналізу;
- техніки оцінки вразливостей мети;
- планування оцінки безпеки;
- виконання оцінки безпеки;
- пост-тестові заходи.

У розділі "Техніки оцінки вразливостей мети", в якості однієї з можливих варіантів описуються тести на проникнення, а саме фази і логістика тестів.

За даним документом тести на проникнення, на додаток до стандартних їх можливостям, можна застосовувати для визначення:

- наскільки добре система переносить реально існуючі моделі атак;
- зразкового рівня складності, який необхідно подолати атакуючому;
- додаткових заходів протидії, які могли б послабити загрози на адресу системи;
- здатності захищає систему на виявлення атак і забезпечення відповідної реакції на них.

Виділяються наступні фази тестів на проникнення:

- планування. На даному етапі визначаються правила, затверджується і документується управління, визначаються мети тесту. Здається основа для успішного тестування на проникнення.
- дослідження. Даний етап включає в себе 2 частини.

Перша частина – старт тестування і збереження інформації, що збирається і сканируемой інформації. Для ідентифікації потенційних цілей проводиться визначення мережевих портів і сервісів.

Друга частина – аналіз вразливостей. Здійснюється порівняння сервісів, додатків, ОС сканується хоста с базами вразливостей (автоматичний процес для сканерів вразливостей) і з власними знаннями аудитора про вразливість. Аудитор може використовувати свої власні бази - або відкриті бази вразливостей - для ручного визначення вразливостей. Ручна обробка дозволяє виявити нові вразливості, але істотно уповільнює процес.

Атака. Етап перевірки раніше визначених вразливостей шляхом їх експлоітірованія. Якщо атака вдала, то вразливість перевіряється і визначаються заходи зниження загроз безпеки.

Звіт. Проводиться під час трьох вищеописаних фаз. Під час фази "Планування" розробляється план оцінки. Під час фаз "Дослідження" і "Атака" зберігаються лог-файли і створюються періодичні звіти. На закінчення тесту, створюється звіт, як правило, для опису вразливостей, вказівки оцінок ризиків.

3.5 Метод тестування комп'ютерної мережі на проникнення

Щоб уберегтися від зайвих втрат державні та комерційні структури застосовують в даний час таку послугу в області інформаційної безпеки, як "Тестування на проникнення" (Pentest).

Вона передбачає здійснення санкціонованого обходу існуючого комплексу засобів захисту власних IT систем та мереж з метою виявити в них слабкі місця (шляхом ідентифікації максимально можливої кількості вразливостей за обмежений час при заданих умовах і поточний стан) і переконатися в їх ефективності.

Суть тесту: в ході pentest роль зловмисника відіграє фахівець, який повинен здійснити атаку на веб-сервер, сервер додатків або баз даних, персонал або корпоративну мережу, визначити рівень захищеності, виявити уразливості, ідентифікувати найбільш ймовірні шляхи злому і визначити на скільки добре працюють засоби виявлення і захисту ІС від атак на комп'ютерну мережу.

Тестування на проникнення є складовою частиною етичного хакінгу – процесу, орієнтованого на пошук і виявлення вразливостей ІБ, а також на проведення контрольованих атак, спрямованих як на окремі IT системи – CMS, CRM, ERP і інтернет клієнт-банк, так і на інфраструктуру об'єкта інформаційної діяльності в цілому.

При аналізі вразливостей елементи pentest можуть використовуватися для оцінки використовуваного в IT системах (мережах) програмного і апаратного забезпечення на предмет спроби їх експлуатації для проникнення в систему.

При проведенні аудиту ІБ елементи pentest можуть використовуватися для оцінки ефективності реалізації таких захисних механізмів, як "захист від злоякісного коду", "забезпечення мережевої безпеки" і інших видів атак. Головне завдання аудитора полягає в тому, щоб знайти відповіді на такі питання: "як простіше потрапити в систему, порушити її працездатність або

що-небудь отримати" і "яка може бути мінімальна ціна злому".

В ході атестації об'єктів інформатизації елементи pentest можуть використовуватися для демонстрації на практиці того, що невідповідність вимогам стандартів або інших нормативно-правових документів з безпеки інформації може призвести до успішної компрометації системи.

Пропонується структурно-логічна схема проведення pentest у вигляді V етапів (рисунок 3.4):

- планування тесту;
- ідентифікація вразливостей;
- спроба експлуатації вразливостей;
- створення та надання звіту;
- усунення вразливостей.

Відповідно до запропонованої структурно-логічної схеми розробимо алгоритм проведення тестування на проникнення (рисунок 3.5).

Перший етап складається в отриманні попередньої інформації про мережу замовника і плануванні проведення тесту на проникнення.

Рисунок 3.4 – Структурно-логічна схема проведення pentest

Пасивні методи:

- Google Hacking & Google Cache;
- Shodan і WHOIS інформація;

- Wayback Machine;
- публікації про компанію та її співробітників в ЗМІ;
- сайти пошуку роботи, прес-релізи інтеграторів;
- сторінки співробітників в соціальних мережах, блоги і форуми;
- пошук в фізичному смітті компанії - Dumpster Diving.

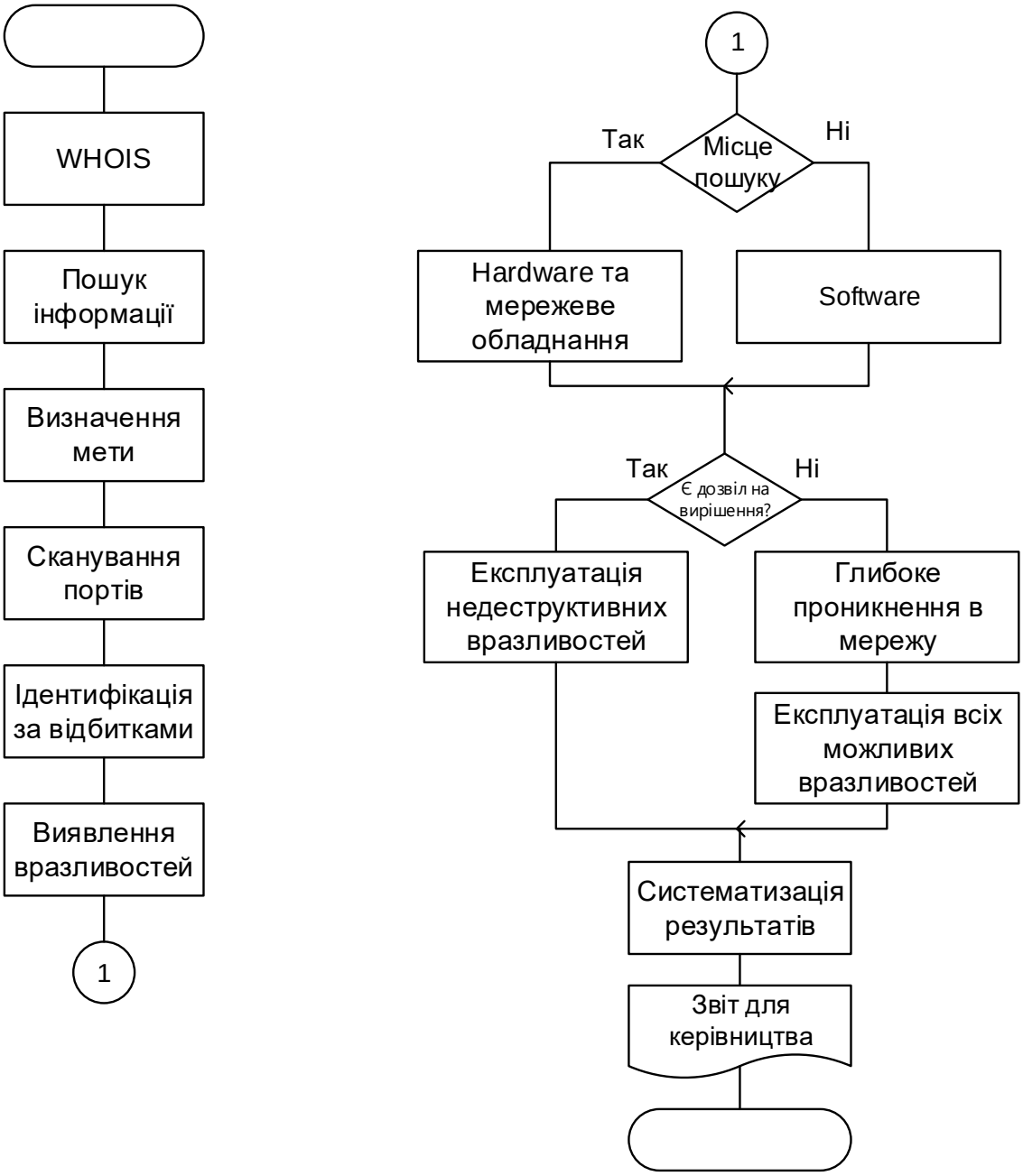


Рисунок 3.5 – Загальний алгоритм проведення pentest

Активні методи:

- Ping Sweep;
- Fingerprint;
- сканування портів;
- NetBios Enumeration і SNMP Enumeration;
- LDAP Enumeration і NTP Enumeration;
- SMTP Enumeration і DNS Enumeration;
- соціальна інженерія.

Результатом першого етапу може бути – формування карти мережі, визначення типів пристроїв, ОС і додатків шляхом оцінки їх реакції на зовнішній вплив

Другий етап. Полягає у пошуку, ідентифікації та перевірці можливості використання вразливостей мережевих служб і додатків. При цьому перевіряється наявність і можливість використання таких вразливих місць, як:

- SQL Injection (використання операторів SQL);
- Source code injection (виконання довільного коду);
- OS Commanding (виконання команд ОС);
- Client-side Attacks (атак на клієнтів);
- Cross-Site Scripting, XSS (міжсайтового виконання сценаріїв);
- Content Spoofing (підміни вмісту);
- Buffer Overflow (переповнення буфера);
- механізмів авторизації і аутентифікації та інше.

Перевірка проводиться як вручну, так і з використанням різних сканерів уразливості від компаній MaxPatrol, Nessus, OpenVAC та інших.

Третій етап складається в експлуатації вразливостей. Отримавши перелік можливих вразливостей аудитор проводить їх експлуатацію. Методи та інструментарій вибираються індивідуально для кожного типу вразливості. Особлива увага при цьому приділяється:

- підбору паролів в різних мережевих сервісах;

- проведення атак типу "людина посередині" для перехоплення паролів користувачів.

За погодженням із замовником при тестуванні на проникнення, додатково, може проводитися перевірка:

- базових робіт по контролю захищеності бездротових мереж;
- зовнішнього периметра і відкритих ресурсів на можливість DoS (DDoS) атак, а також оцінки ступеня стійкості мережевих елементів і можливого збитку при їх проведенні;
- стійкості мережі, шляхом моделювання атак на протоколи канального рівня STP, VTP, CDP, ARP;
- стійкості маршрутизації, шляхом моделювання фальсифікації маршрутів і проведення DoS (DDoS) атак проти використовуваних протоколів маршрутизації;
- мережевого трафіку, з метою отримання, наприклад, паролів користувачів, конфіденційних документів та ін .;
- можливості отримання зловмисником НСД до конфіденційної інформації або інформації обмеженого доступу замовника (проводиться перевіркою прав доступу до різних IP замовника з привілеями, отриманими на різних етапах тестування) і.т.д.

Четвертий етап грунтується на оформленні звіту. Отримана в ході аналізу вразливостей і спроб їх експлуатації інформація документується і аналізується з метою вироблення рекомендацій у формі звіту, направленого на покращення захищеності IT систем (мереж).

Найбільш оптимальною структурою звіту є його розбиття на три рівні: для вищого керівництва, для менеджерів ІБ і для технічних фахівців (рисунок 3.6). Звіт повинен містити:

- методику проведення тесту;
- висновки для керівництва, що містять загальну оцінку рівня захищеності;
- опис виявлених недоліків системи управління ІБ;

- опис ходу тестування з інформацією по всіх виявлених вразливостей і результатами їх експлуатації;
- рекомендації щодо усунення виявлених вразливостей.

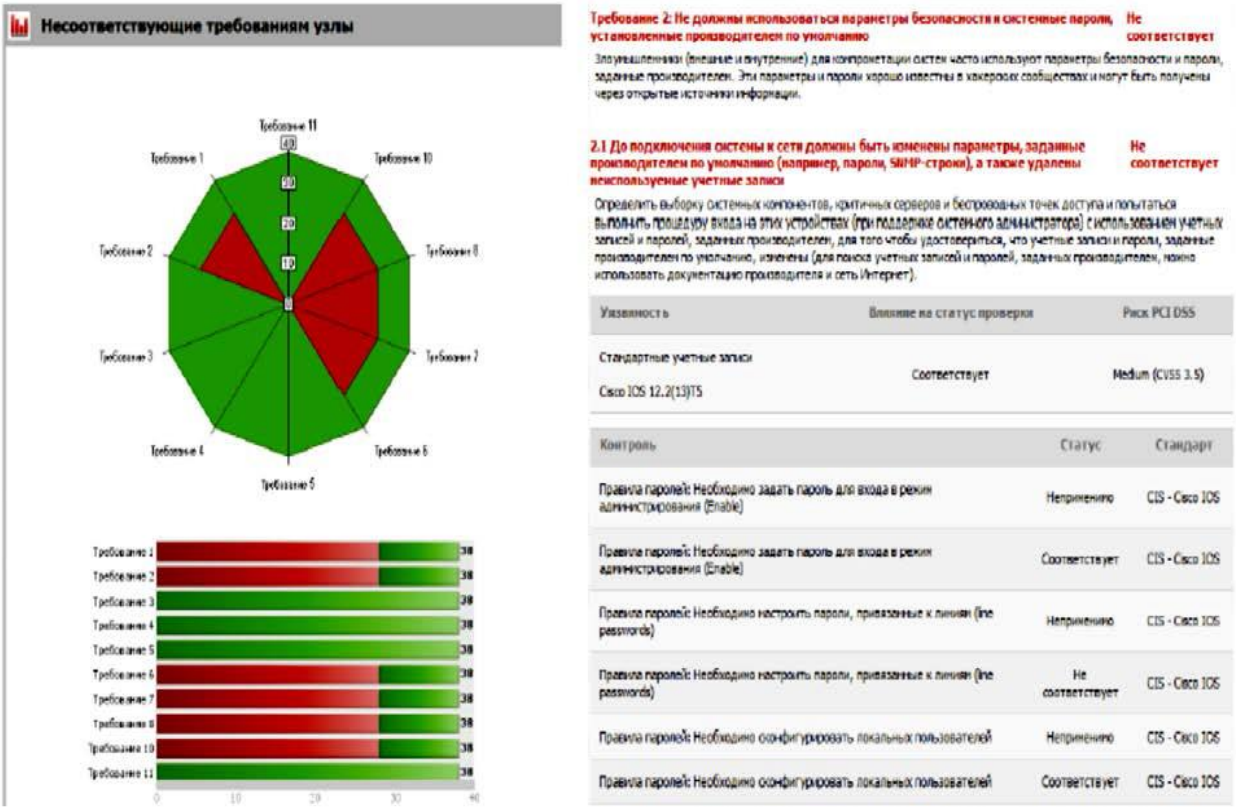


Рисунок 3.6 – Варіант звіту про результати pentest, виконаного за методикою NIST SP 800-115

Критерієм завершення тесту на проникнення є отримання:

- доступу до внутрішньої мережі з боку мережі Інтернет;
- доступу до певного сегмента мережі (наприклад, сегмент АСУ ТП);
- привілеїв в основних інфраструктурних та інформаційних системах / сервісах (Active Directory, мережеве обладнання, СУБД, ERP і т.п.);
- доступу до певних інформаційних ресурсів;
- доступу до певної інформації (наприклад, електронна пошта директора);
- першого серйозного збою, викликаного діями аудитора.

На п'ятому етапі після проведення тесту можливі залишкові сліди тесту, так звані артефакти, які необхідно усунути.

3.5 Висновки за розділом 3

Проведений аналіз існуючих методів і засобів оцінки вразливостей інформаційних систем дозволив виявити їх недоліки і формалізувати завдання щодо розробки більш ефективного засобу.

У результаті багатокритеріального аналізу встановлено, що усі ці засоби не є досконалими і мають певні обмеження щодо практичного застосування для розв’язання різного роду завдань ІБ.

ВИСНОВКИ

В результаті даної роботи були проаналізовані підходи до проведення аудиту кібербезпеки за рахунок технології тестування проникнення. Таким чином, на основі приведених результатів досліджень можливо сформулювати основні завдання, рекомендації та базові пропозиції щодо підходів до забезпечення безпеки інформації, яка циркулює в комп'ютерних мережах.

Проведено аналіз вразливостей інформаційних систем та оцінка її критичності для комп'ютерних мереж. Аналіз дав можливість визначити найбільш ефективний підхід і використати його з метою оптимізації параметрів систем захисту.

Проведено аналіз існуючих методів тестування засобів захисту інформації, обрані допоміжні інструменти, з використанням яких була розроблена архітектура системи моделювання атак і реалізований прототип програмного засобу.

Розроблену систему можна вдосконалити в декількох напрямках:

- можливість аудиту цілої комп'ютерної мережі, а не тільки окремого одного хоста;
- комбінування декількох мережевих сканерів;
- автоматизація процесів після успішної експлуатації системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Стюарт М. Секрети хакерів. Безпека мереж – готові рішення [Текст] / М. Стюарт, Д. Скрембрей, Д. Курц. – Москва: "Вільямс", 2002. – 529 с.

2. Безпека інформаційних технологій. Критерії оцінки безпеки інформаційних технологій //Гостехкомиссия України. [Електронний ресурс] – Режим доступу: http://www.fstec.ua.

3. Безпека інформаційних технологій. Міжмережні екрани корпоративного рівня // Інфосистеми. [Електронний ресурс] – Режим доступу: http://www.fstec.ua.

4. Лукацкий А. В. Мережева безпека перехід на апаратний рівень [Електронний ресурс] – Режим доступу: http://bezpeka.ua.

5. Лукацкий А.В. Виявлення атак [Текст] / А.В. Лукацкий. – СПб.: БХВ-Петербург, 2003. – 608 с.

6. Аудит безпеки інформаційних систем. [Електронний ресурс] – Режим доступу: http://www.isaca.ru/security/Pubs.

7. Аналіз захищеності автоматизованих систем. [Електронний ресурс] – Режим доступу: http://www.isaca.ru/security/Pubs.

8. Єршов В.А. Мультисервісні телекомунікаційні мережі [Текст] / В.А. Єршов, Н.А. Ковалив. – Москва : Вид-во МГТУ, 2013. – 432 с.

9. Курило А.П. Аудит інформаційної безпеки [Текст] / А.П. Курило, С.Л. Зефіров, В.Б. Голованов та ін. – Москва : "БДЦ-прес", 2006. – 304 с.

10. Лепихин В.Б. Порівняльний аналіз сканерів безпеки. [Електронний ресурс] – Режим доступу: http://www.itsecurity.ru/edu/actions/2008-pentest.zip.

11. CVE. [Електронний ресурс] – Режим доступу: https://cve.mitre.org.

12. Core Impact Pro. Comprehensive multi-vector penetration testing. [Електронний ресурс] – Режим доступу: http://www.coresecurity.com/core-impact-pro.

13. CANVAS, D2 and Tenable Nessus Professional Feed Bundle.

[Електронний ресурс] – Режим доступу: http://www.immunityinc.com/-products/canvas/d2-nessus-bundle.html.

14. SAINT. Vulnerability management, penetration testing, configuration assessment and compliance. [Електронний ресурс] – Режим доступу: http://www.saintcorporation.com.

15. Intro to host scanning with NMAP, AMAP, HPING3, XPROBE2, TCPDUMP. [Електронний ресурс] – Режим доступу: http://www.securitytube.net/video/4008.

16. Nmap preset scans. Options and scan types explained. [Електронний ресурс] – Режим доступу: http://www.securesolutions.no/zenmap-preset-scans.

17. Nmap techniques for avoiding firewalls. [Електронний ресурс] – Режим доступу: https://pentestlab.wordpress.com/2012/04/02/nmap-techniques-for-avoiding- firewalls.

18. Port scanning techniques. [Електронний ресурс] – Режим доступу: http://nmap.org/book/man-port-scanning-techniques.html.

19. Port scanning with Nmap. [Електронний ресурс] – Режим доступу: http://my.safaribooksonline.com/book/networking/security/9781593272883/3dot-intelligence-gathering / active_information_gathering.

20. Service name and transport protocol port number registry. [Електронний ресурс] – Режим доступу: http://www.iana.org/assignments/-service-names-port-numbers/service-names-port-numbers.xhtml.

21. Understanding and customizing Nmap data files. [Електронний ресурс] – Режим доступу: http://nmap.org/book/nmap-services.html.

22. Meterpreter в справі: хитрі прийоми через MSF. [Електронний ресурс] – Режим доступу: https://xakep.ru/2011/03/22/54887.

23. Chroot, практика. [Електронний ресурс] – Режим доступу: http://linuxru.org/tips/173.

24. Metasploit, MeterpreterClient. [Електронний ресурс] – Режим доступу: http://en.wikibooks.org/wiki/Metasploit/MeterpreterClient.

25. Welcome to InfraGard. [Електронний ресурс] – Режим доступу: https://www.infragard.org/.

26. Голубничий Д.Ю. Технології аудиту кібербезпеки інформаційних систем / Д.Ю. Голубничий, О.В. Коломійцев, В.Ф. Третяк, С.Г. Рязанін // Scientific Collection «InterConf», (36): with the Proceedings of the 7th International Scientific and Practical Conference «Challenges in Science of Nowadays» (November 26 - 28, 2020) in Washington, USA: EnDeavours Publisher, 2020. – Pp. 333 – 342.

27. Голубничий Д.Ю. Аналіз вимог безпеки щодо адміністрування комп'ютерних систем та мереж організації на основі процесного підходу / Д.Ю. Голубничий, В.П. Коцюба, О.В. Сєвєрінов, О.І. Соловйова, Ю.О. Семеренко // The 4th International Scientific and Practical Conference «Scientific Research in XXI Century» (May 16-18, 2020). Ottawa, Canada: Methuen Publishing House, 2020. – Pp. 333 – 342.

28. Голубничий Д.Ю. Оцінка складності методів виявлення атак / А.В. Власов, В.Ф. Третяк, Д.М. Запара, І.Ю. Жукова // Scientific Collection «InterConf», (37): with the Proceedings of the 1st International Scientific and Practical Conference «Recent Scientific Investigation» (December 6-8, 2020). – Oslo, Norway: Dagens naeringsliv forlag, 2020. – Pp. 1061 – 1070.