Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)
Кафедра Інфокомунікаційної інженерії імені В.В. Поповського
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Аналіз процесного підходу до аудиту Linux-подібних серверних операційних систем
(тема)

Виконав:
студент 2 курсу, групи АМСЗІм-18-1
Стрілець А.М.
(прізвище, ініціали)

Спеціальність: 125 Кібербезпека
(код і повна назва спеціальності)
Тип програми: освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма: Адміністративний менеджмент у сфері захисту інформації
(повна назва освітньої програми)

Керівник: професор кафедри ІКІ ім. В.В. Поповського

Шостко І.С.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ Лемешко О.В.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)
Кафедра Інфокомунікаційної інженерії імені В.В. Поповського
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 125 Кібербезпека
(код і повна назва)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Адміністративний менеджмент у сфері захисту інформації
(повна назва)

ЗАТВЕРДЖУЮ
Зав. кафедри_____
(підпис)
«____» _____2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студенту Стрільцю Андрію Миколайовичу .
(прізвище, ім'я, по батькові)

1. Тема роботи: Аналіз процесного підходу до аудиту Linux-подібних серверних операційних систем
затверджена наказом по університету від «17» березня 2020р. №465 Ст.
2. Термін подання студентом роботи до екзаменаційної комісії 10.05.2020 р.
3. Вихідні дані до роботи: методи проведення експертного аудиту інформаційної безпеки серверних операційних систем, аналіз методів проведення аудиту інформаційної безпеки, політики управління інформаційною безпекою, комп'ютер з операційною системою CentOS 8, веб-сервер Apache
4. Перелік питань, що потрібно опрацювати в роботі.
1) Аудит інформаційної безпеки
2) Розробка рекомендацій щодо проведення аудиту linux-подібних операційних систем
3) Розробка рекомендацій щодо проведення аудиту веб-серверів
4) Приклад аудиту інформаційної безпеки сервера

5. Перелік графічного матеріалу із зазначенням креслень, плакатів, комп'ютерних ілюстрацій: Демонстраційний матеріал у вигляді ppt-презентації

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		(підпис)	(дата)
Основна частина	професор Шостко Ігор Світославович		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	17.02.2020	Виконано
2	Збір матеріалів для дослідження	28.02.2020	Виконано
3	Розробка 1 розділу	15.03.2020	Виконано
4	Розробка 2 розділу	20.03.2020	Виконано
5	Розробка 3 розділу	26.04.2020	Виконано
6	Розробка 4 розділу	01.05.2020	Виконано
7	Оформлення атестаційної роботи	10.05.2020	Виконано

Дата видачі завдання 17 лютого 2020 року

Студент _____ Стрілець А.М.
(підпис) (прізвище, ініціали)

Керівник роботи _____ професор Шостко І.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 79 с., 11 рис., 2 таблиці, 21 джерело.

LINUX, CENTOS, ЗАХИСТ ІНФОРМАЦІЇ, АУДИТ, ПРАВА ДОСТУПУ, ОПЕРАЦІЙНА СИСТЕМА, ВРАЗЛИВОСТІ, ЗАГРОЗИ, ВЕБ-СЕРВЕР.

Об’єкт дослідження – процес проведення аудиту інформаційної безпеки Linux-подібних серверних операційних систем.

Предмет дослідження – методологія проведення аудиту інформаційної безпеки Linux-подібних серверних операційних систем.

Мета роботи – підвищення якості інформаційної безпеки Linux-подібних серверних операційних систем за допомогою проведення ефективного аудиту інформаційної безпеки.

Методи дослідження – аналіз, синтез та порівняння.

У наш час корпоративні мережі відіграють важливу роль в успішній діяльності будь-яких організацій. Вони значно підвищують ефективність роботи співробітників організації. Тому виникає необхідність в забезпеченні ефективної і надійної системи захисту корпоративних мереж.

У роботі розглядається аналіз процесного підходу проведення аудиту інформаційної безпеки Linux-подібних серверних операційних систем (ОС), оскільки саме сервер є найважливішим компонентом корпоративної мережі. Проведення аудиту дозволяє виявити прогалини в системі захисту сервера, які можуть призвести до втрати конфіденційної інформації чи завдати іншої шкоди.

На основі отриманих даних, розроблено комплекс рекомендацій направлених на усунення виявлених прогалин і на надійне забезпечення захисту інформації.

ABSTRACT

The report contains: 79 p., 11 fig, 2 tables, 21 sources.

LINUX, CENTOS, INFORMATION PROTECTION, AUDIT, ACCESS RIGHTS, OPERATING SYSTEM, VULNERABILITIES, THREATS, WEB SERVER.

The object of the study is the process of conducting an information security audit for Linux-like server operating systems.

The subject of the study is the methods and means of conducting information security audits of Linux server operating systems.

The purpose of the work is to improve the quality of information security of Linux server operating systems by conducting an effective information security audit.

Nowadays, corporate networks play an important role in the success of any organization. They greatly increase the efficiency of the organization's employees. Therefore, there is a need to provide an effective and reliable corporate network security system.

The work examines the process of conducting an audit of information security of Linux-like server operating systems (OS), since the server itself is an essential component of the corporate network. The auditing audit allows you to isolate the gaps in the server security system, which can lead to loss of confidential information or other damage.

On the basis of the received data, a set of recommendations aimed at eliminating the identified gaps and the reliable security of information was developed.

ЗМІСТ

Перелік скорочень, умовних позначень, символів, одиниць і термінів…..... 7
Вступ……. 8
1 Аудит інформаційної безпеки…….. 10
1.1 Поняття аудиту інформаційної безпеки, його цілі і завдання…….. 10
1.2 Етапи проведення аудиту інформаційної безпеки…….. 12
1.3 Процесний підхід в забезпечені інформаційної безпеки……. 16
2 Розробка рекомендацій щодо проведення аудиту linux-подібних операційних систем……..…….. 18
2.1 Керування обліковими записами та управління паролем…….. 18
2.2 Безпека файлів і елементи управління…….. 28
2.3 Мережна безпека…… 35
2.4 Журнали аудиту…… 43
2.5 Моніторинг безпеки та загальний контроль……. 45
2.6 Інструменти та технології…… 47
3 Розробка рекомендацій щодо проведення аудиту веб-серверів…….. 49
3.1 Основи веб-аудиту…….. 49
3.2 Рекомендації щодо проведення аудиту веб-сервера……. 50
4 Приклад аудиту інформаційної безпеки сервера……... 53
4.1 Дослідження системи з точки зору аудиту…… 53
4.2 Створення чек-листа аудиту…….. 59
4.3 Звіт проведення аудиту…… 69
Висновки……. 76
Перелік джерел посилання ……..…….. 78

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ІБ – інформаційна безпека
ІС – інформаційна система
ОС – операційна система
ACL – Access Control Lists
CERT – Computer Emergency Response Team
CIS – Center for Internet Security
CITR – Computer Incident Response Team
CMS – Content Management System
DMZ – Demilitarized Zone
DoS – Denial of Service
FTP – File Transfer Protocol
ICMP – Internet Control Message Protocol
IIS – Internet Information Services
LDAP – Lightweight Directory Access Protocol
LMD – Linux Malware Detect
NASL – Nessus Attack Scripting Language
NFS – Network File System
NIS – Network Information Services
NNTP – Network News Transfer Protocol
NOC – National Occupational Classification
PAM – Pluggable Authentication Modules
RCP – Remote Copy
RSH – Remote Shell
SFTP – Secure File Transfer Protocol
SSH – Secure Shell
TCP Transmission Control Protocol
UDP – User Datagram Protocol
UID – Unique Identifier
XST – Cross-Site Tracing

ВСТУП

Удосконалення інформаційно-комунікаційних технологій зробило доступним величезний обсяг інформації. Але в той же самий час, доступність створює значні ризики для інформаційних систем (ІС), і особливо, для критичних ресурсів, які вони підтримують. Зараз ми можемо спостерігати, що кількість складних атак на інформаційні системи збільшується в геометричній прогресії. Кількість нових атак, а також їх складність буде зростати прямо пропорційно, тому як будуть розвиватися інформаційно-комунікаційні технології. Цей факт зумовлює необхідність забезпечення безпеки інформаційних систем організації.

Незважаючи на значні досягнення в галузі інформаційної безпеки, багато інформаційних систем все ще вразливі до внутрішніх або зовнішніх атак. У цьому контексті надзвичайно важливим є належне управління безпекою шляхом постійної ідентифікації основних активів та їх вразливостей, а також загроз та атак, яким вони піддаються. Однією із стратегій для досягнення цієї мети є проведення регулярних аудитів безпеки інформаційної системи, оцінка ефективності управління інформацією та аналіз існуючих політик управління інформаційною безпекою.

Аудит інформаційної безпеки – це системний процес отримання об’єктивних оцінок поточного стану інформаційної безпеки організації (підприємства) відповідно до визначених критеріїв інформаційної безпеки (ІБ), який включає комплексне обстеження різних середовищ функціонування інформаційної системи, проведення тестування на вразливості, аналіз і оцінку стану захищеності системи, формування звіту і розробку відповідних рекомендацій [1].

Перевірка стану інформаційної безпеки організації допомагає сформувати єдиний погляд на проблеми безпеки компанії серед спеціалістів різних напрямків, дає можливість дати оцінку всім основним рівням забезпечення інформаційної безпеки компанії: організаційний, нормативно-правовий, технологічний і апаратно-програмний. Також аудит безпеки інформаційної системи допомагає дати оцінку ефективності здатності організації захищати свої цінні або критичні активи.

Беручи до уваги вищезазначене, можна впевнено заявити, що проведення аудиту інформаційної безпеки є досить актуальним питанням в наш час, у зв’язку

з чим темою даної атестаційної роботи було обрано дослідження аудиту Linux-подібних серверних операційних систем (ОС) як однієї із ключових складових інформаційної системи.

Аудит безпеки операційної системи необхідний, особливо коли її використовують кілька користувачів або коли система є частиною мережі компанії. Перш ніж приступити до аудиту безпеки, ви повинні ознайомитися з основами аудиту IT-безпеки, основною метою якого є забезпечення захисту інформаційних активів і правильний розподіл інформації серед уповноважених осіб. Щоб зробити кращий вибір при виборі операційної системи, а безпека є найбільш важливим фактором, вам необхідно знати процедури кожної операційної системи для створення, реєстрації та складання звітів про перевірки стану безпеки. Нарешті, необхідно скласти список, що порівнює найбільш важливі функції безпеки операційних систем і вибирає краще рішення на його основі.

В даній роботі були створені рекомендації для збільшення ефективності методології проведення аудиту Linux-подібних серверних операційних систем. Оскільки сервер є одним із найважливіших елементів інформаційної системи, то аудит серверів є важливим завданням для забезпечення безпеки на рівні платформи в IT-інфраструктурі і забезпечення правильного налаштування безпеки сервера Linux. Система Linux має свою власну конфігурацію безпеки і систему управління, для задоволення вимог безпеки в корпоративному середовищі. Системний адміністратор повинен налаштувати систему Linux, щоб отримати більше гарантій безпеки від системи, а аудитори ІС повинні перевірити конфігурацію системи Linux відповідно до стандартів аудиту, щоб переконатися, що на підприємстві встановлена безпечна система.

1 АУДИТ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

1.1 Поняття аудиту інформаційної безпеки, його цілі і завдання

В сучасному світі інформаційні системи відіграють вирішальну роль в забезпеченні якісного виконання бізнес-процесів як комерційних, так і державних підприємств. Разом з тим повсюдне використання ІС для зберігання, обробки і передачі інформації призводить до підвищення актуальності проблем, пов’язаних з їх захистом. Підтвердженням цьому служить той факт, що останнім часом спостерігається зростання числа злочинів, пов’язаних з порушенням основних принципів інформаційної безпеки: доступності, цілісності і конфіденційності інформації.

Незважаючи на розвиток різноманітних засобів захисту, таких як мережеві екрани, системи автентифікації і розмежування доступу, кількість атак зловмисників на серверні операційні системи, як одні з ключових елементів ІТ-інфраструктури організації, зростає з кожним роком. Збільшення числа атак на сервери призводить до необхідності розробки і застосування засобів та систем захисту. Для того щоб гарантувати ефективний захист від інформаційних атак зловмисників, компаніям необхідно мати об’єктивну оцінку поточного рівня безпеки ІС. Саме для цих цілей застосовується аудит безпеки, різні аспекти якого розглядаються в рамках цієї роботи.

ІТ-аудит являє собою перевірку засобів управління всередині ІТ-інфраструктури. Також ІТ-аудит можна вважати процесом збору та оцінки доказів для визначення того, чи захищає комп’ютерна система активи, підтримує цілісність даних, дозволяє ефективно досягти організаційних цілей та ефективно використовувати ресурси. Його можна здійснювати спільно з аудитом фінансової звітності та внутрішнім аудитом. Мета ІТ аудиту відрізняється від аудиту фінансової звітності, оскільки останній дотримується стандартної практики бухгалтерського обліку, а аудит ІТ оцінює структуру і ефективність внутрішнього контролю системи.

Цілі та завдання аудиту інформаційної безпеки бувають різні, в залежності від типу і виду аудиту. Цілі аудиту визначає замовник аудиту, це може бути керівник організації, керівник служби ІБ або інша уповноважена особа на основі

специфіки, бізнес-цілей і діяльності організації. Всі цілі і завдання перерахувати неможливо, але можна виділити найосновніші і найбільш розповсюдженні.

Цілями проведення аудиту інформаційної безпеки можуть бути [3]:

- перевірка поточного стану інформаційної безпеки організації;
- демонстрація надійності організації, здатності виступати в якості стійкого партнера, здатного забезпечити захист інформаційних ресурсів та систем;
- виявлення недоліків і визначення напрямків розвитку системи захисту інформації;
- прогнозування і мінімізація ризиків, загроз і вразливостей, а також управління їх впливом на бізнес-процеси організації;
- оцінка і встановлення ступеня відповідності ІБ організації обраними критеріями аудиту ІБ;
- оцінка відповідності ІБ організації існуючим вимогам і стандартам в області інформаційної безпеки, нормативним документам або політикам безпеки;
- контроль ефективності фінансування в забезпечення інформаційної безпеки організації;
- оцінка рівня ефективності системи захисту інформації після її впровадження;
- розслідування інциденту, що стався, пов’язаного з порушенням інформаційної безпеки в організації;
- розробка рекомендацій щодо вдосконалення ІБ;
- розробка рекомендацій щодо впровадження нових та вдосконалення існуючих методів забезпечення інформаційної безпеки.

Завдань, які можуть вирішуватися в ході проведення аудиту інформаційної безпеки також може буди велика кількість, ми виділимо серед них основні [4]:

- оцінка критичності активів;
- класифікація загроз ІБ;
- визначення ймовірності реалізації загроз;
- визначення рівня небезпеки і актуальності загроз;
- оцінка ризиків ІБ;
- виявлення значимих вразливостей, загроз і ризиків інформаційної безпеки і шляхів їх реалізації;

- аналіз структури, функцій і використовуваних технологій в організації;
- аналіз бізнес-процесів;
- аналіз поточного стану інформаційної безпеки;
- розробка політик безпеки та інших організаційно-розпорядчих документів щодо захисту інформації;
- розробка рекомендацій за підсумками аудиту ІБ організації.

1.2 Етапи проведення аудиту інформаційної безпеки

Аудит інформаційної безпеки, незалежно від форми та методів його проведення, складається з чотирьох основних етапів, котрі наведені на рис. 1.1.

Рисунок 1.1 – Етапи проведення аудиту

Перший етап аудиту, це планування і незважаючи на те, що планування зосереджено на початку аудиту, воно представляє з себе ітеративний процес, що виконується протягом усього аудиту. Це тому що результати попередньої оцінки забезпечують основу для визначення ступеня і типу подальшого тестування. Якщо аудитори отримують докази того, що певні контрольні процедури неефективні, вони можуть визнати за необхідне переглянути свої попередні висновки та інші рішення з планування, прийняті на основі цих висновків.

На початку планування ІТ-аудитор повинен зібрати інформацію про наступні аспекти об’єкта перевірки.

1) Функція організації та робоче середовище. Сюди слід включити загальне розуміння різних ділових практик та функцій, що стосуються аудиту,

типів інформаційних систем, що підтримують діяльність, а також робоче середовище. Розуміння організації допомагає вирішити, як проводити аудит, з якою частотою, коли, як і в якій мірі.

2) Структура організації. IT-аудитору необхідно зрозуміти ієрархію організації, а також структуру та ієрархію IT-відділу.

3) Критичність IT-систем. IT-системи можна віднести до категорії критично важливих систем. Критично важливі системи – це системи, вихід зі строю яких матиме дуже серйозні наслідки для організації.

4) Характеристика апаратного та програмного забезпечення. Розуміння апаратного обладнання організації в цілому та IT-системи зокрема є критично важливим для аудитора. Ця інформація забезпечує аудитору розуміння пов'язаних з цим ризиків. Хоча світ рухається до стандартизованого обладнання, все ж існують відмінності, і кожен тип апаратного забезпечення має свої вразливості, які потребують конкретного контролю. Аудитор також повинен оцінити процес придбання та технічного обслуговування обладнання, а також зрозуміти тип програмного забезпечення, яке використовується в організації. Аудитору необхідно зібрати інформацію про операційні системи (ОС), прикладні системи та системи управління базами даних, що використовуються в організації.

5) Характер та ступінь ризиків, що впливають на системи. Визначення ризиків, що впливають на систему є дуже важливим кроком при плануванні, оскільки отримані результати впливають на весь процес проведення аудиту інформаційної безпеки.

Аудитор може зібрати необхідну інформацію шляхом:

- дослідження довідкових матеріалів, включаючи публікації організації, щорічні звіти та незалежні аудиторські і аналітичні звіти (якщо такі існують);
- ознайомлення з довгостроковими стратегічними планами;
- опитування ключового персоналу для розуміння бізнес-питань;
- відвідування ключових об'єктів організації.

Інформація яка необхідна аудитору, визначатиметься характером організації та рівнем деталізації, на якому виконується аудиторська перевірка. Знання про організацію повинні включати, бізнес, фінанси, та ризики з якими зустрічається організація. Аудитор також повинен включати ступінь, в якій організація покладається на аутсорсинг для досягнення своїх цілей. Він повинен використовувати цю інформацію для виявлення потенційних проблем,

формулювання цілей та обсягу роботи.

На другому етапі відбувається визначення цілей і обсягу аудиту. Здійснюється це за допомогою визначення ймовірних ризиків інформаційної системи. Управління ризиками є основною вимогою сучасних IT-систем, де важлива безпека. Його можна визначити як процес виявлення ризику, оцінки ризику і прийняття заходів щодо зниження ризику до прийнятного рівня. Три цілі безпеки будь-якої організації: конфіденційність, цілісність і доступність.

Окремо можна виділити кроки, яких можна дотримуватися для підходу, заснованого на оцінці ризику, для складання плану аудиту:

- провести інвентаризацію інформаційних систем, що використовуються в організації, і класифікувати їх;
- визначте, яка із систем впливає на критичні функції чи активи;
- оцініть, які ризики впливають на ці системи та ступінь тяжкості їх впливу на бізнес;
- на підставі вищезгаданої оцінки визначають пріоритет аудиту, ресурси, графік та частоти.

Аудитору, як правило, слід зробити попередню оцінку контролю та розробити план аудиту на основі цієї оцінки. Спираючись на ці оцінки для планування аудиторських тестів, аудитор може уникнути витрачання ресурсів на тестування, які явно не є ефективними. Хоча важливо чітко встановити цілі аудиту для початку детального аудиту, необхідно розуміти, що в ході аудиту ці цілі можуть зазнати змін або додаткових розробок.

Далі наведено перелік деяких загальних цілей аудиту інформаційної безпеки:

- огляд контролю IT-систем, щоб отримати впевненість у їх адекватності та ефективності;
- оцінка продуктивності системи або конкретної програми;
- огляд безпеки IT-систем;
- дослідження процесів розробки системи та процедури, що дотримуються на різних етапах, що в них беруть участь.

Цілі та сфера аудиту можуть охоплювати більше ніж один аспект вищезазначених напрямків.

Третій етап полягає в проведенні аналізу отриманої інформації з метою оцінки реального рівня захищеності інформаційної системи, що перевіряється.

Також необхідно отримати компетентні, відповідні та обґрунтовані докази, які підтверджують судження та висновки аудитора щодо діяльності та функцій організації, апаратних та технічних засобів. Методи, що використовуються аудитором для аналізу даних визначаються обраними підходами до проведення аудиту, які можуть сильно відрізнятися.

Методи збору даних повинні бути ретельно підібрані. Аудитори повинні чітко розуміти обрані методи та процедури. Для збору аудиторських доказів зазвичай застосовуються наступні методи.

1) Інтерв'ю. Аудитори можуть використовувати інтерв'ю для отримання як якісної, так і кількісної інформації під час роботи зі збору доказів. Наприклад, системні аналітики та програмісти можуть бути опитані для кращого розуміння функцій та елементів управління, вбудованих у систему.

2) Анкети. Традиційно анкети використовуються для оцінки контролю в системах. Аудитори також можуть використовувати анкети для визначення слабких областей системи під час збору доказів. Аналогічно, анкети можуть бути використані для визначення областей в інформаційній системі, де існує потенційна неефективність.

3) Блок-схеми. Блок-схеми управління показують, наявність елементів керування в системі і де ці елементи існують в системі.

Вони мають три основні цілі аудиту:

- розуміння – побудова блок-схеми управління висвітлює ті сфери, де аудитори не мають розуміння ні самої системи, ні контролю в системі;
- оцінювання – досвідчені аудитори можуть використовувати блок-схеми управління для розпізнавання моделей, які виявляють або сильні сторони управління, або слабкість контролю в системі;
- комунікація – аудитори можуть використовувати блок-схеми управління, щоб передавати свої розумінням системи та пов'язані з нею елементи управління іншим.

Четвертий етап проведення аудиту ІБ полягає в створенні рекомендацій щодо дій пов'язаних з усуненням виявлених вразливостей. Також аудитори повинні зафіксувати аудиторські докази в робочих документах, включаючи наступне:

- планування та підготовка обсягу та завдань аудиту;
- програма аудиту;

- зібрані докази, на основі яких зроблені висновки;
- всі робочі документи, що стосуються аудиту;
- звіти та дані, отримані в системі безпосередньо аудитором або надані персоналом (аудитор повинен забезпечити, щоб ці звіти містили джерело звіту, дату та час та умови, за яких вони отримані).

У різних пунктах документації аудитор може додавати свої зауваження та роз'яснення щодо проблем, сумнівів та потреби в додатковій інформації. Аудитор повинен пізніше повернутися до цих коментарів і додати зауваження та посилання на те, як і де вони були вирішені.

Проект і підсумкові звіти по аудиту повинні бути частиною аудиторської документації.

1.3 Процесний підхід в забеспечені інформаційної безпеки

Однією з концепцій управління являється процесний підхід. Вся сутність цієї концепції полягає в тому, що вся діяльність складається з набору процесів, і щоб управляти, необхідно управляти цими процесами. Такий підхід є одним з основних елементів покращення якості.

Найчастіше процес має наступне визначення. Процес – це сукупність взаємопов'язаних і взаємодіючих видів діяльності, які перетворять входи у виходи. Але на мою думку, до цього визначення потрібно добавити, що дії процесу повинні бути систематичними і не випадковими.

Процесний підхід був розроблений і застосовується з метою створення горизонтальних зв'язків в організаціях. При такому підході, підрозділи та працівники, які беруть участь в одному процесі, можуть самостійно координувати роботу в рамках процесу і вирішувати проблеми без участі керівництва. Процесний підхід дає можливість швидше вирішувати виникаючі питання і впливати на результат.

Зазвичай підходи забезпечення ІБ будувалися за принципом чек-листа. Суть такого підходу полягає в тому, що забезпечення безпеки полягало в виконанні певного набору пунктів, які містяться в загально визначених рекомендаціях. Але такий підхід не зможе задовольнити всі потреби сучасного бізнесу, враховуючи динаміку з якою з'являються нові загрози ІБ, потрібно швидко реагувати на будь-які зміни. Досягти цього більшою мірою можливо саме при реалізації процесного

підходу.

Процесний підхід передбачає наявність ключових елементів, без яких він не може бути запроваджений в організації. До таких ключових елементів відносяться:

- вхід процесу;
- вихід процесу;
- ресурси;
- власник процесу;
- споживачі і постачальники процесу;
- показники процесу.

У процесному підході будь-яка діяльність, яка виконується або для управління якої використовуються ресурси організації, вважається процесом, що перетворює входи на виходи. Це методологія (підхід), що ідентифікує процеси в організації так, щоб їх взаємозв'язки могли бути зрозумілі, видимі і вимірні, а підсумкова сукупність процесів розумілася б як єдина система реалізації цілей діяльності організації.

На рис. 1.2 зображена схема процесу аудиту, на зображені ми бачимо, що на вході маємо все необхідне для проведення перевірки (програма і план аудиту, документація, обладнання), а на виході отримаємо результати заради яких виконувався аудит [1]. Потрібно зауважити, що процес аудиту повинен буди періодичним.

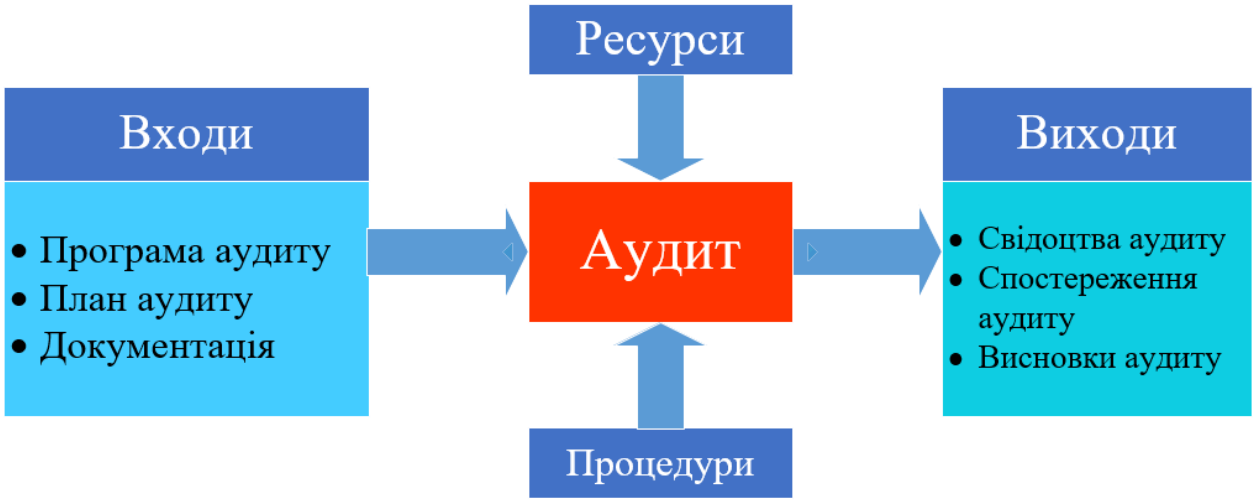

Рисунок 1.2 – Процес аудиту

2 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ПРОВЕДЕННЯ АУДИТУ LINUX-ПОДІБНИХ ОПЕРАЦІЙНИХ СИСТЕМ

Перевірка стану безпеки сервера є основною частиною забезпечення безпеки на рівні платформи в IT-інфраструктурі та забезпечення правильної конфігурації сервера Linux. Система Linux має особисту систему конфігурації та керування безпекою для вирішення вимог захисту в корпоративному середовищі. Системному адміністратору потрібно відповідно налаштувати систему, щоб отримати додаткову гарантію безпеки від системи, а перевіряючому ІС слід перевірити чи відповідають налаштування системи Linux стандартам інформаційної безпеки, щоб забезпечити безпеку системи [5].

В цьому розділі мною представлені основні напрямки та рекомендації по проведенню аудиту безпеки серверних операційних систем.

2.1 Керування обліковим записом та управління паролем

Перш ніж розпочати перевірку, я рекомендую аудитору визначити, чи система використовує лише свій локальний файл пароля (/etc/passwd) або якусь додаткову форму централізованого управління обліковим записом, наприклад Network Information Services (NIS) або Lightweight Directory Access Protocol (LDAP). Я не буду намагатися вказувати команди для кожної можливої централізованої системи управління обліковими записами. У якості прикладу мною будуть розглянуті деталі для отримання інформації з NIS, яка є найпоширенішою з цих систем. Якщо використовується інший інструмент, наприклад NIS + або LDAP, вам знадобиться переглянути документацію для цих систем, щоб визначити еквівалентні команди.

Для того, щоб ефективно виконати аудиту інформаційної безпеки операційної системи, я рекомендую виконати наступні кроки, для кожного з кроків я описав як виконати перевірку.

1) Оцінити процедури створення облікових записів користувачів та переконатися, що облікові записи створюються лише тоді, коли є потреби в цьому. Також рекомендую оцінити як відбувається процес вимкнення облікових записів в разі потреби.

Якщо ефективні засоби управління не встановлені для надання та видалення доступу до сервера, це може призвести до того, що користувачі матимуть зайвий доступ до системних ресурсів. А це, в свою чергу, ставить під загрозу цілісність та безпеку сервера.

В якості одного з методів перевірки можете опитати системних адміністраторів та переглянути процедури створення облікових записів. Цей процес повинен включати певну форму перевірки того, що кожен користувач має відповідний доступ.

Щоб здійснити перевірку, я раджу взяти зразок облікових записів з файлу паролів і переглянути докази того, що вони були затверджені належним чином до їх створення. Також потрібно взяти зразок облікових записів із файлу паролів та підтвердити їх відповідність, досліджуючи та розуміючи функцію роботи власників облікових записів.

Також необхідно дослідити процес видалення облікових записів, коли доступ більше не потрібен, наприклад після звільнення працівників. В якості перевірки цього процесу я пропоную отримати зразок облікових записів із файлу паролів та переконатися, що вони належать активним працівникам та чи не змінилися посадові місця цих працівників з моменту створення облікового запису.

2) Необхідно переконатися, що всі ідентифікатори користувачів у файлі паролів унікальні.

Якщо двоє користувачів мають однаковий ідентифікатор користувача UID (Unique Identifier), вони можуть повністю отримувати доступ до файлів та каталогів один одного і можуть «вбивати» процеси один одного, навіть якщо вони мають різні імена користувачів. Операційна система використовує UID для ідентифікації користувача. Він просто відображає ім’я користувача на відповідний UID у файлі пароля.

Використовуючи команду «more /etc/passwd» (команда для перегляду файлів) ви переконаєтесь у відсутності дублікатів UID для локальних облікових записів. Якщо у вашій системі використовується NIS, то необхідно виконати команду «ypcat passwd», також раджу її використовувати для того, щоб можна було вивчити UID NIS.

Використання наступної команди перерахує всі UID, що повторюються, у файлі локального пароля:

#cat /etc/passwd | awk -F: '{print $3}' | uniq –d.

3) Потрібно переконатися, що паролі приховані, і по можливості використовувати надійні хеші.

Щоб система функціонувала належним чином, файл паролів повинен бути доступний для читання всім, тому що імена користувачів потрібні дуже багатьом нешкідливим програмам. Оскільки файл легко прочитати будь-якому користувачеві, то і зашифровані варіанти паролів теж доступні, а значить, будь-який хакер зможе запустити підбір паролів і чекати заповітної години «X», коли буде знайдена потрібна комбінація. Щоб захистити паролі, у всіх сучасних версіях Linux їх ховають в файл «/etc/shadow», який доступний для читання тільки користувачу root.

Також розглянемо форму шифрування паролів. Підпрограма шифрування, що традиційно використовується для паролів Unix, є відносно слабкою формою шифрування по сучасним стандартам, і максимальна ефективна довжина пароля становить вісім символів. Я рекомендую використовувати хеш-кодування MD5, які важко зламати і в яких можна ввести більше восьми символів для пароля.

Щоб визначити, чи використовується файл тіньового пароля, раджу використати команду «more /etc/passwd». Потрібно переглянути поля пароля для всіх облікових записів. Якщо кожен обліковий запис має «*», «x» або якийсь інший поширений символ у ньому, то система використовує «тіньовий» файл пароля. Як говорив раніше файл «тіньового» пароля буде розміщений за адресою «/etc/shadow» для більшості систем. Системи, що використовують NIS, створюють деякі особливі проблеми, які ускладнюють використання прихованих паролів, а старі системи взагалі не можуть приховувати ці паролі.

4) Оцінити права доступу до файлу паролів і файлу «тіньових» паролів.

Якщо користувач може змінити вміст цих файлів, він зможе додавати та видаляти користувачів, змінювати паролі користувачів або стати суперкористувачем, змінюючи свій UID на 0. Якщо користувач може прочитати вміст файлу «тіньового» пароля, він може скопіювати зашифровані паролі та спробувати їх розшифрувати.

Для перевірки я пропоную переглянути права доступу до цих файлів за допомогою команди «ls –l». Файл «/etc/passwd» має бути доступний для запису лише для root, а файл «/etc/shadow» також має бути доступним для читання лише користувачеві root.

5) Оцінити надійність паролів в системі.

Напевно не є таємницею, що коли в системі є легкі паролі, існує велика ймовірність того, що зловмисник зможе зламати обліковий запис, отримуючи таким чином несанкціонований доступ до системи та її ресурсів. Тому в ході аудиту, обов'язково рекомендую перевірити налаштування складності пароля, щоб знизити ризик атак на пароль методом «грубої сили» (анг. Brute force) або підбором по словнику. Для налаштування складності пароля раджу застосовувати модулі аутентифікації Pluggable Authentication Modules (PAM). Перевірити відповідні налаштування можна у файлі конфігурації:

vi /etc/pam.d/system-auth.

Також необхідно переконатися, що в системі відсутні або заблоковані облікові записи, що дозволяють увійти в систему без введення пароля. Для цього рекомендую використати наступну команду:

cat /etc/shadow | awk -F: ($2==””){print $1}’.

6) Оцінити використання засобів керування паролем.

Важливо періодично змінювати паролі з двох основних причин. По-перше, якщо пароль не змінювати, зловмисник з копією зашифрованих або хешованих паролів матиме необмежену кількість часу для здійснення офлайн-атаки злому. По-друге, той, хто вже має несанкціонований доступ, зможе зберегти цей доступ на невизначений термін.

Підчас перевірки необхідно перевірити системні налаштування, які забезпечують контроль за старінням пароля.

Одним із параметрів які потрібно перевірити в ході аудиту, це налаштування терміну закінчення дії пароля. Щоб перевірити термін дії пароля раджу використовувати команду «change». Ця команда виводить детальну інформацію терміну дії пароля, максимальний та мінімальний вік паролю, а також дату його останньої зміни. Мінімальний вік важливий для того, щоб користувач не міг змінити свій пароль, а потім негайно змінити його на попереднє значення. Обов'язково потрібно виконати налаштування цих параметрів і порівняти їх з політикою безпеки IT вашої компанії.

Також потрібно заборонити користувачам використовувати один пароль декілька разів підряд. Щоб виконати це необхідно відкрити файл «/etc/pam.d/common-password» (для Ubuntu/Debian/Linux Mint) і додати наступний рядок в розділ «auth»: «auth sufficient pam_unix.so likeauthnullok». Потім для заборони використовувати останні шість паролів потрібно додати наступний рядок: «Password sufficient pam_unix.so nullokuse_authtok md5 shadow remember=6». Після збереження файлу, система буде зберігати історію про попередні шість паролів, і якщо який-небудь користувач буде намагатися оновити пароль, використовуючи будь-який з останніх шести, він отримає повідомлення про помилку.

Як правило, кореневий обліковий запис не підлягає автоматичному старінню, щоб запобігти можливості блокування облікового запису. Однак я рекомендую встановити ручний процес для періодичної зміни пароля відповідно до політики компанії. Необхідно переглянути процес зміни цього пароля та знайти докази того, що цей процес виконується. Також раджу обов’язково перевірити процес, який використовуються системними адміністраторами для документування та передачі root паролів, оскільки вони, ймовірно, передаються між членами команди.

7) Впевнитись, що існують процеси, які використовуються системними адміністраторами для встановлення початкових паролів для нових користувачів та передачі цих паролів.

Коли створюються нові облікові записи користувачів, системний адміністратор повинен призначити цьому користувачеві початковий пароль. Якщо цей пароль легкий, він може дозволити зламати обліковий запис, що призведе до несанкціонованого доступу до сервера та його ресурсів. Якщо початковий пароль передається не захищеним каналом, він також може дозволити іншим переглядати пароль та отримувати несанкціонований доступ до облікового запису.

В якості перевірки рекомендую опитати системного адміністратора та переглянути документацію, щоб зрозуміти механізм, який використовується для створення початкових паролів. Також обов’язково переконайтесь, що цей механізм призводить до складних паролів котрі відповідають політиці ІБ вашої компанії.

Також я б порадив переглянути канали, які використовуються для передачі нових паролів користувачам. Та переконатись, що незашифровані передачі не використовуються. Часто буває корисно, щоб користувач змінив свій пароль

відразу при першому вході. Щоб визначити, чи це робиться чи ні, необхідно опитати системного адміністратора. Зберіть письмові докази вище вказаного.

8) Необхідно переконатись, що кожний обліковий запис пов’язаний і може бути легко відстежено для конкретного співробітника.

Якщо власник облікового запису не є очевидним, це буде перешкоджати судовому розслідуванню щодо неналежних дій, вчинених цим обліковим записом. Якщо кілька людей використовують обліковий запис, неможливо встановити відповідальність за дії, що виконуються цим обліковим записом.

Для перевірки рекомендую переглянути вміст файлу паролів. Власник кожного облікового запису повинен бути очевидним, з ім’ям користувача або іншим унікальним ідентифікатором (наприклад, таким як номер співробітника), використовуваним в якості імені користувача або поміщеним в поле «GECOS».

Поставте питання системному адміністратору про облікові записи, які є загальними, наприклад, облікові записи гостя або додатка. Якщо такі облікові записи потрібні, вони повинні бути налаштовані з обмеженими оболонками і/або так, щоб користувач не міг безпосередньо увійти в них. В цілях безпеки рекомендую, щоб користувач спочатку увійшов в систему під своїм ім’ям, а потім використовував su або sudo, щоб отримати доступ до загального облікового запису.

9) Оцінити доступ до облікових записів суперкористувача root та інших привілейованих облікових записів.

Обліковий запис з доступом до кореневого рівня має можливість робити будь-що із системою, включаючи видалення всіх файлів та вимкнення системи. Тому бажано доступ до цієї здатності звести до мінімуму. Інші облікові записи можуть існувати в системі для адміністрування конкретних програм, а також повинні бути жорстко контрольовані, щоб запобігти зриву системи.

Для виявлення привілейованих користувачів рекомендую переглянути вміст файлів паролів та визначити усі облікові записи з UID 0. Тому що, всі облікові записи з UID 0 система обробляє так, як якщо б це був обліковий запис root.

Також я б рекомендував, переглянути файл пароля на наявність інших привілейованих облікових записів. Ви, ймовірно, знайдете потенційних кандидатів під час виконання кроку під номером вісім. За допомогою інтерв’ю, можна визначити хто знає паролі до цих облікових записів, і оцінити відповідність.

Багато середовищ використовують sudo або подібний інструмент, що дозволяє користувачам виконувати певні функції з підвищеними привілеями. Це корисний спосіб дозволити користувачеві виконувати конкретні обов'язки системного адміністрування без надання користувачеві повного root доступу. Тому бажано навіть для користувачів, яким потрібен повний кореневий доступ, налаштувати sudo так, щоб дозволити користувачеві виконувати всі команди з кореневим доступом, дозволяючи користувачеві виконувати системне адміністрування зі свого власного облікового запису замість входу в обліковий запис root. Це корисно для цілей аудиту.

Якщо sudo використовується, раджу переглянути файл «/etc/sudoers», щоб оцінити здатність користувачів запускати команди від користувача root (та інших конфіденційних облікових записів) за допомогою команди sudo. Sudo може бути використаний для надання конкретним користувачам можливості запускати певні команди так, як ніби вони були «root» (або будь-який інший обліковий запис). Це як правило, краще, ніж надати користувачам повний кореневий доступ.

Основний формат запису у файлі «/etc/sudoers», наведений на рис. 2.1.

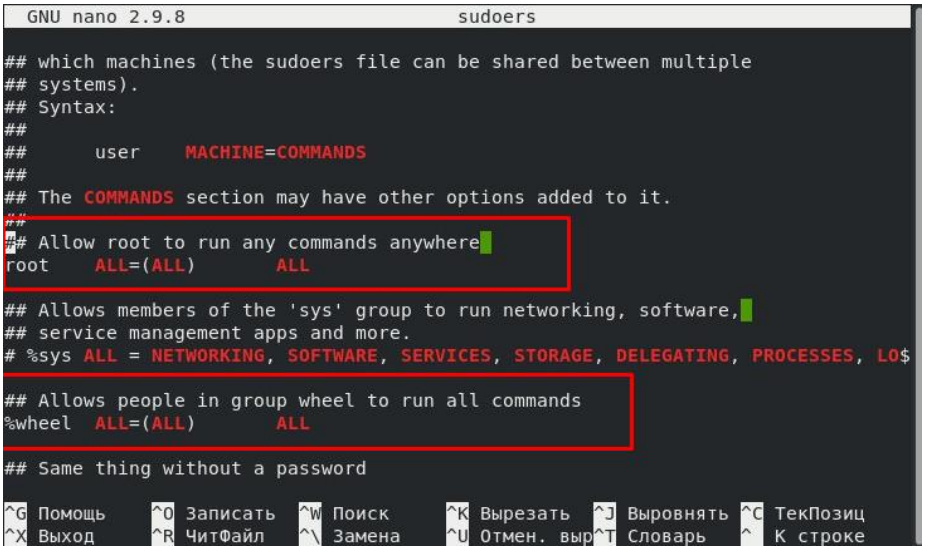

Рисунок 2.1 – Зміст файлу «/etc/sudoers»

У цьому прикладі користувачеві andrey будуть надані привілеї користувача root у всіх системах, а членам групи «wheel» дозволено виконувати будь-яку команду.

Також необхідно переконатися, що записи у файлі «/etc/sudoers» періодично перевіряють та корегують. Якщо ваше середовище велике (складається з великої кількості серверів), рекомендую, краще реалізувати певну форму

централізованого файлу sudoers, на яку посилаються всі системи, а не намагатися підтримувати файл у кожній окремій системі.

10) Оцінити використання груп та визначити обмеженість їх використання.

Ця інформація стане основою для оцінки дозволів файлів на наступних кроках. Якщо всі користувачі розміщені в одній або двох великих групах, то дозволи групових файлів не відіграють великої ролі. Однак якщо користувачі розміщені у вибіркових, продуманих групах, дозволи групових файлів є ефективними елементами управління.

Для перевірки необхідно переглянути вміст файлів «/etc/group» та «/etc/passwd» та пов’язані з ними централізовані файли (наприклад, NIS), раджу використовувати команду «more» (more /etc/passwd) і для NIS команду «ypcat» (ypcat passwd і ypcat group).

Подивіться на файли паролів та груп, щоб отримати уявлення про призначення групи, оскільки призначення файлу основної групи користувача з файлу пароля не повинні бути включені в файл групи. Іншими словами, якщо користувач призначений в групу «користувачі» у файлі «/etc/passwd», немає необхідності вказувати його або її в якості члена цієї групи в файлі «/etc/group». Тому, щоб отримати повний список всіх членів групи «користувачі», потрібно визначити, хто був призначений цій групі в файлі «/etc/group», а також визначити, хто був призначений цій групі в файлі /etc/passwd (поряд з будь-якими NIS, LDAP і т. д., еквівалентами, що використовуються у вашому середовищі). Важливо відзначити, що група не повинна бути вказана в файлі групи, щоб існувати. Тому необхідно ідентифікувати всі ідентифікатори груп GID (Group Identifiers) в файлі паролів і визначити членство в цих групах. Якщо ви використовуєте файл групи для ідентифікації всіх груп в системі, ви можете не отримати повну картину.

11) Оцінити використання паролів на рівні груп.

Паролі на рівні групи дозволяють людям стати членами груп, з якими вони не асоціюються. Створення пароля на рівні групи створює ще один вектор атаки на систему, створюючи можливість для користувачів зламати паролі на рівні групи та ескалацію своїх привілеїв. Для перевірки я б порадив переглянути вміст файлів групи, використовуючи команду «more /etc/group» для локального файлу та команду «ypcat group» для NIS. Якщо у групах є щось інше, ніж загальний символ (наприклад, «*» або навіть нічого) у полі пароля (друге поле для кожного запису), використовуються паролі. Якщо використовуються паролі на рівні групи,

раджу звернутися до системних адміністраторів, щоб зрозуміти мету та значення використання таких паролів, і перегляньте процес обмеження знань про ці паролі.

Щоб шукати паролі в файлі «/etc/group», я раджу використати наступну команду під час аудиту:

#awk -F: '{if($2!="" && $2!="x" && $2!="*")print "A password is set for group "$1" in /etc/group\n"}' /etc/group.

12) Оцінити безпеку каталогів у шляху за замовчуванням, який використовується системним адміністратором при додаванні нових користувачів. Та оцінити використання «поточного каталогу» на шляху.

Шлях користувача містить набір каталогів, які слід шукати щоразу, коли користувач вводить команду, не вводячи повне ім’я шляху. Наприклад, припустимо команда «ls» у вашій системі знаходиться за адресою «/bin/ls». Щоб виконати цю програму і переглянути дозволи в каталозі «/home», ви можете ввести «/bin/ls /home». Ввівши точне місце розташування файлу, ви використовуєте повне ім’я шляху. Однак ми рідко робимо це. Натомість норма – це виконати команду «ls /home». У цьому випадку шлях користувача – це механізм пошуку файлу, який повинен бути виконаний.

Наприклад, скажімо, що ваш шлях виглядає так:

/usr/bin:/usr/local/bin:/bin.

Це означає, що при введенні команди операційна система спочатку шукатиме файл з таким іменем у «/usr/ bin». Якщо файл там не існує, його буде шукати в «/usr/local/bin». Якщо він все ще не знайде файл з таким ім’ям, він буде шукати в «/bin». Таким чином, у нашому прикладі ми спробували виконати команду ls, яка знаходиться в «/bin». Система спочатку шукатиме файл, який називається ls в каталозі «/usr/bin». Оскільки в цьому каталозі немає файлу, він буде шукати в директорії «/usr/local/bin». Оскільки файлу там також немає, він буде знаходитись в /bin. Файл котрий називається ls знаходиться в директорії «/bin» вашої системи, тому операційна система спробує виконати цей файл. Якщо дозволи на цей файл дають вам дозвіл, вам буде дозволено запускати програму.

Зловмисники, які мають дозволи на запис в каталог на шляху користувача, можуть виконувати підробку імен файлів. Наприклад, якщо каталог, який містить

команда «ls» не захищений, зловмисник може замінити команду. Крім того, якщо «поточний каталог» (тобто те, в якому каталозі знаходиться користувач під час виконання команди) або інший незахищений каталог розміщений на початку шляху користувача, зловмисник може розмістити свою власну версію команди «ls» в одному з таких і ніколи не звертатися до реальної команди «ls».

Через все це, потрібно встановити права доступу на папки на шляху таким чином, щоб вони належали користувачу або системі, а також не повинні бути доступні для запису групі чи іншим користувачам.

Найпростіший спосіб переглянути власний шлях – це введення команди «echo $PATH» у командному рядку. Налаштування за замовчуванням для шляхів користувачів можна знайти в файлах «/etc/default/login», «/etc/profile», або в одному з файлів у «/etc/skel». Потрібно запитати у системного адміністратора, де зберігаються налаштування за замовчуванням, якщо ви не впевнені. Якщо користувач змінив свій шлях, це робиться, як правило, в одному з файлів домашнього каталогу, тому подивіться на вміст таких файлів як «.login», «.profile», «.cshrc», «.bash_login», тощо. Для швидкого способу пошуку – в домашньому каталозі користувача раджу використати наступну команду:

#grep "PATH =". *.

Домашній каталог користувача можна визначити, переглянувши його запис у файлі паролів.

Як тільки ви дізнаєтесь ім’я файлу, який містить шлях, обов’язково перегляньте вміст файлу, використовуючи команду «more». Каталоги повинні бути доступні для запису лише обліковими записами користувачів та системним. Оцінити безпеку домашніх каталогів та файлів конфігурації. Вони мають бути доступні для запису тільки власником.

Файли конфігурації користувача – це в основному будь-який файл, розташований у домашній директорії користувача, який починається з крапки «.». Який зазвичай називається dot-файлом. Ці файли визначають середовище користувача, і якщо третя сторона може змінити їх, то вона може отримати привілейований доступ до облікового запису. Наприклад, коли користувач вперше входить у систему, виконуються команди в межах його «.login», «.profile», «.bashrc» або іншого файлу (залежно від оболонки). Якщо зловмисник в змозі змінити один з цих файлів, він може вставити довільні команди, і користувач

виконає ці команди при наступному вході. Наприклад, можуть бути виконані команди, які копіюють оболонку користувача в інший файл і роблять його Set User Identifier (SUID). Тоді зловмисник зможе виконати цей новий файл і «стати» цим користувачем. Доступ до цих файлів також дозволяє зловмиснику змінити шлях користувача або створити шкідливі псевдоніми для загальних команд шляхом зміни цих файлів. Інші конфігураційні файли, такі як «.cshrc» та «.kshrc», виконуються при вході в систему, коли запускається нова оболонка або коли хтось використовує команду su для переходу на обліковий запис користувача. Можливість вставляти довільні команди в ці файли призводить до аналогічного ризику, що і для файлів «.login» та «.profile». Тому обов'язково раджу встановити права доступу до цих файлів таким чином, щоб тільки власникові вони були доступні.

Ще один конфігураційний файл, який слід заблокувати, це файл «.rhosts». Цей файл забезпечує надійний доступ (доступ без використання пароля) до облікового запису користувача з конкретних облікових записів інших систем. Людина, яка може змінити цей файл, може отримати доступ до облікового запису користувача.

Доступ до домашнього каталогу користувача також потрібно заблокувати. Якщо зловмисник має права доступу до каталогу, він матиме змогу видалити будь-який з конфігураційних файлів користувача та замінити їх на власні версії.

Місце розташування домашніх каталогів користувачів можна отримати із записів облікового запису у файлі паролів. Рекомендую використовувати команду «ls –ld» у кожному каталозі для перегляду дозволів каталогу та дозволів на файли (включаючи конфігураційні файли) всередині каталогу.

2.2 Безпека файлів і елементи управління

Важливо, щоб в системі були правильно налаштовані права доступу до всіх файлів та каталогів у системі. Для здійснення ефективної перевірки прав доступу мною були сформовані наступні рекомендації.

1) Оцінити права доступу до файлів для об'єктивної вибірки критичних файлів і пов'язаних з ними каталогів.

Якщо важливі файли не захищені належним чином, дані в цих файлах можуть бути змінені або видалені невідповідними користувачами. Це може

привести до збою системи або несанкціонованого розкриття і зміни конфіденційної інформації.

Для перевірки раджу використати команду «ls –l», за її допомогою можна вивчити дозволи на критичні системні файли та пов’язані з ними каталоги. Як правило, найважливіші файли в операційних системах Unix та Linux містяться в наступних каталогах:

- /bin, /usr/bin, /sbin, /usr/sbin, and/or /usr/local/bin (програми, які інтерпретують команди та керують такими речами, як зміна паролів);
- /etc (файли, що містять інформацію, таку як паролі, членство в групі та довірені хости та файли, які контролюють виконання різних демонів);
- /usr та /var (містять різні журнали обліку).

Для цих каталогів і файлів, що в них містяться, необхідно поставити під сумнів необхідність надання доступу на запис кому-небудь, окрім персоналу системного адміністратора.

Крім того, інші важливі файли даних (наприклад, файли, що містять ключові дані програм та конфіденційну інформацію про компанію), ймовірно, будуть у системі, яку ви ревізуєте, і повинні бути захищені. Раджу провести інтерв’ю з системним адміністратором щоб визначити їх.

Для зручності використання та отримання повного уявлення про файлову систему рекомендую попросити системного адміністратора запустити команду «ls –alR» (рекурсивний перелік файлів) для всієї файлової системи та розмістити результати у файл для вас. Потім ви можете переглянути вміст цього файлу. Системний адміністратор повинен зробити це, тому що лише супервайзер може отримати доступ до вмісту всіх каталогів.

2) Знайти відкриті каталоги (каталоги з дозволом, встановленим на drwxrwxrwx) в системі та визначити, чи повинен у них бути встановленій sticky bit.

Якщо каталог відкритий, кожен може видалити файли в каталозі та замінити їх своїми власними файлами з однойменною назвою. Це іноді підходить для «/tmp» каталогів та інших сховищ для некритичних даних, однак це не рекомендується для більшості каталогів.

Sticky bit – додатковий атрибут файлів або каталогів в операційних системах сімейства UNIX. Використовується в основному для каталогів, щоб захистити у них файли. З такого каталогу користувач може видалити лише ті файли,

власником яких він є. Прикладом може служити каталог «/tmp», в якому запис дозволений для всіх користувачів, але не бажане видалення чужих файлів. Установка атрибута здійснюється за допомогою утиліти «chmod» [6].

Для перевірки я б радив вивчити права доступу до каталогів, з файлу отриманому на попередньому кроці, та знайти відкриті каталоги. Щоб знайти лише каталоги з дозволами world-write, рекомендую скористатися командою:

#find / -type d –perm -777.

Для будь-яких виявлених таких каталогів обговоріть функцію цих каталогів із системним адміністратором та визначте відповідність відкритих дозволів.

3) Оцінити безпеку всіх файлів SUID в системі, особливо тих, які SUID мають root.

Файли SUID дозволяють користувачам виконувати їх під привілеями іншого UID. Іншими словами, під час виконання цього файлу операційна система «робить вигляд», що користувач, який його виконує, має привілеї UID, який належить файлу. Наприклад, кожному користувачеві потрібна можливість оновлювати файл паролів, щоб періодично змінювати паролі. Однак не було б розумно встановлювати права доступу до файлу паролів, щоб дозволити доступ до world-write, оскільки це дасть можливість кожному користувачеві додавати, змінювати та видаляти облікові записи. Команда «passwd» була створена таким чином, щоб надати користувачам можливість оновлювати свої паролі, не маючи можливості змінювати решту файлів паролів. Файл «passwd» належить root і встановлений біт SUID (- rwsr-xr-x), це означає, що коли користувачі виконують його, вони роблять це, використовуючи привілеї «root».

Якщо файл SUID доступний для запису будь-кому, крім власника, можливо, що його обліковий запис може бути скомпрометований. Інші користувачі можуть змінити запущену програму для виконання довільних команд під UID власника файлу. Наприклад, команда може бути вставлена так, що оболонка власника копіюється в файл і перетворюється в SUID. Потім, коли зловмисник запустить цю скопійовану оболонку, він буде працювати так, як якщо б він був власником файлу SUID, що дозволить зловмисникові виконати будь-яку команду, використовуючи рівень привілеїв захопленого облікового запису.

Повний список файлів SUID пропоную переглянути за допомогою наступної команди:

#find / -perm -u+s.

Зверніть увагу, що результати цієї команди не будуть повними, якщо вона не запущена кимось з правами суперкористувача.

4) Необхідно переконатися, що всі файли мають законного власника у файлі «/etc/passwd».

Кожен раз, коли файл створюється, йому присвоюється власник. Якщо згодом цей обліковий запис буде видалено, UID цього облікового запису все ще буде вказаний як власник файлу, якщо право власності не буде перенесено на дійсний обліковий запис. Якщо інший обліковий запис буде створено пізніше з тим самим UID, власнику цього облікового запису за визначенням буде надано право власності на ці файли.

Наприклад, припустимо, що Andrey (UID 226) створює файл «/UIB/file». UID 226 (Andrey) вказаний як власник цього файлу. Потім Andrey звільняється, а його обліковий запис видаляється. Однак право власності на його файл не передається. Операційна система все ще вважає UID 226 власником цього файлу, хоча той UID більше не відображає користувача у файлі пароля. Через кілька місяців Ivana приймають на роботу і йому присвоюють UID 226. Тепер система вважає Ivana власником файлу «/UIB/file», і вона має повні привілеї щодо цього файлу. Якщо файл «/UIB/file» файл містить конфіденційну інформацію, це може бути проблемою. Щоб уникнути цієї проблеми, перед видаленням облікового запису системні адміністратори повинні видалити всі файлами, що належать цьому обліковому запису, або передати право власності.

Для перевірки раджу попросити системного адміністратора виконати команду «quot» (яку повинен виконувати супервайзер). Ця команда покаже всіх власників файлів у системі. Отриманий список необхідно переглянути і переконатись, що для кожного запису відображається ім’я користувача, а не UID. Якщо з’явиться UID, це означає, що у файлі паролів для цього UID немає запису, що означає, що файл пароля не зміг перетворити UID в ім’я користувача. Якщо користувач буде доданий пізніше до файлу паролів з цим UID, він має право власності на ці файли.

5) Необхідно впевнитись, що команда «chown» не може використовуватися користувачами для компрометації облікових записів користувачів.

Ця команда дозволяє користувачам передавати право власності на свої файли комусь іншому. Якщо користувач може передати SUID-файл іншому користувачеві, він зможе виконати цей файл і «стати» власником.

Багато версій Unix дозволяють виконувати «chown» лише суперкористувачам. Щоб визначити, чи є ці елементи керування на машині, яку ви ревізуєте, рекомендую виконати наступні дії:

- переглянути файл пароля і визначити, де знаходиться ваша оболонка (це ймовірно, буде щось на кшталт «/bin/csh» або «/usr/bin/sh»);
- виконати команду «cp <shell file name> ~/myshell» створивши копію вашого файлу оболонки в вашому домашньому каталозі;
- запустити команду «chmod 4777 ~/myshell», щоб створити новий виконуваний файл оболонки SUID;
- обрати іншого користувача з файлу паролів, до якого потрібно передати право власності, бажано колеги-аудитори;
- виконати команду «chown <new owner name>~/myshell», буде спроба передати право власності на файл іншому користувачеві;
- виконати команду «ls -l ~/myshell», щоб побачити, чи перевели ви право власності успішно, і якщо так, чи SUID-біт також передано;
- якщо біт SUID передано іншому власнику, необхідно виконати файл, ввівши «/myshell»;
- виконати команду «whoami», це повинно показати, що ви зараз інший користувач і прийняли його обліковий запис.

6) Оцінити значення umask за замовчуванням для сервера.

Umask визначає, які дозволи матимуть нові файли та каталоги за замовчуванням. Якщо umask за замовчуванням не встановлений належним чином, користувачі можуть ненавмисно надати доступ до своїх файлів і каталогів. За замовчуванням має бути надійне створення файлів. Привілеї можуть бути послаблені на основі потреб та свідомих рішень користувачів (на відміну від того, що вони не знають, що їхні нові файли та каталоги не захищені).

Umask для всіх користувачів за умовчанням встановлюється в файлах «/etc/.profile» або в файлів «/etc/.bashrc». Однак часто найпростішим тестом є

просто переглянути значення umask для вашого власного облікового запису, тому що це зазвичай буде представлення значення за замовчуванням для всіх нових користувачів. Це можна зробити за допомогою команди «umask».

Umask в основному віднімає привілеї, коли файли та каталоги створюються з використанням модульного формату дозволів файлів і припускаючи, що за замовчуванням усі файли та каталоги створюються повністю відкритими дозволами (777). Іншими словами, якщо umask 000, всі нові файли та каталоги будуть створені з дозволами 777 (777 мінус 000), що означає повний доступ для власника, групи та інших користувачів.

Як мінімум, для системи за замовчуванням, я б рекомендував встановити дозволи 027 (дозвіл на запис для груп, а всім іншим користувачам доступ заборонено) або 037 (для груп запис/ виконання і для інших доступ заборонений).

7) Дослідити crontabs системи, особливо root, на наявність незвичайних або підозрілих записів.

Cron виконує програму в заданий час. Це в основному рідний спосіб системи Unix або Linux, який дозволяє вам планувати завдання. Cron можна використовувати для створення «бомб» уповільненої дії для компрометації облікового запису. Наприклад, якщо зловмиснику вдалося скомпрометувати обліковий запис користувача, він може створити завдання, яке копіює оболонку користувача вночі і робить її SUID, а потім видаляє цю копію оболонки через 15 хвилин. Тоді зловмисник може щодня відновлювати доступ до облікового запису протягом цього періоду часу, але інструменти моніторингу безпеки не виявлять його, якщо тільки інструменти не запускаються у тому п’ятнадцятихвилинному вікні.

Я рекомендую переглянути файли crontab, вони повинні розташовуватися в каталозі «/usr/spool/cron/crontabs» або «/var/spool/cron/crontabs». Виконуючи команду «ls –l» в цих каталогаг, ви зможете переглянути вміст. Кожен обліковий запис із cronab матиме свій файл у цьому каталозі. Вміст цих файлів можна переглядати за допомогою команди «more». Це дозволить вам побачити команди, які виконуються, та графік їх виконання. Виходячи з дозволів на файли, вам може знадобитися адміністратор, щоб відобразити вміст файлів crontabs. Також, залежно від рівня ваших знань Unix, вам може знадобитися допомога адміністратора в інтерпретації вмісту цих файлів.

8) Оцінити безпеку файлів, на які є посилання в записах crontab, особливо root. Раджу переконатися, що записи відносяться до файлів, які

належать і доступні для запису тільки власнику crontab, і що ці файли знаходяться в каталогах, які належать і доступні для запису тільки власнику crontab.

Cron все запускає так, як ніби їх запускає власник crontab, незалежно від власника виконуваного файлу. Якщо хтось, крім власника crontab, може записувати в файл, що виконується crontab, несанкціонований користувач може отримати доступ до цих облікових записів, змінивши виконувану програму, щоб змусити власника crontab виконувати довільні команди (такі як копіювання оболонки власника cron і створення SUID). Наприклад, якщо в crontab root є запис, який виконує файл «/home/Andrey/flash», і цей файл належить «Andrey», то «Andrey» має можливість додати будь-яку команду, яку він хоче, у флеш-файл, щоб виконати цю команду при наступному запуску cron.

Якщо crontab виконує файл, який знаходиться в каталозі, який не є захищеним, це дозволить іншим користувачам видалити запущену програму та замінити її на власну, що знову може призвести до того, що власник crontab виконує довільні команди. Тому рекомендую переглянути вміст crontab кожного користувача. Команда «ls –l» повинна виконуватися для кожного файлу, який виконується в crontab, і команда «ls –ld» команда повинна бути виконана для кожного каталогу, що містить ці файли.

9) Рекомендую дослідити заплановані роботи системи на предмет незвичних або підозрілих записів.

Atjobs – це разові завдання, які планується виконувати в певний час у майбутньому. Вони функціонують так само, як і завдання в cron (за винятком того, що вони виконуються лише один раз) і також можуть використовуватися для створення «бомб» уповільненої дії.

Atjobs повинні знаходитись у каталозі «/usr/spool/cron/atjobs» або «/var/spool/cron/atjobs». Виконуючи команду «ls –l» в цьому каталозі, ви можете отримати вміст каталогу. Вміст цих файлів можна переглядати за допомогою команди «more». Це дозволить вам побачити команди, які виконуються, та графік виконання. На основі дозволів на файли вам може знадобитися адміністратор, щоб відобразити вміст роботи. Також, залежно від рівня ваших знань Unix, вам може знадобитися допомога адміністратора в інтерпретації вмісту файлів.

2.3 Мережна безпека

Оскільки в наш час комп’ьютер майже завжди знаходиться в мережі, тому дуже важливо перекотися що він захищений. Для проведення аудиту мережевої безпеки, я б порадив виконати наступні кроки.

1) Визначити, які мережеві сервіси включені в систему, і підтвердити їх необхідність з адміністратором системи. Для необхідних служб переглянути та оцінити процедури оцінювання вразливих місць, пов’язаних з цими службами, та збереження їх виправлень.

Щоразу, коли віддалений доступ дозволений (тобто коли ввімкнено послугу мережі), він створює новий потенційний вектор атаки, тому збільшує ризик несанкціонованого входу в систему. Враховуючи це, рекомендовано, щоб послуги мережі повинні бути включені лише тоді, коли в них є необхідність. Нові вразливі місця в безпеці часто виявляються та передаються спільноті Unix/Linux. Якщо системний адміністратор не знає про ці сповіщення, і якщо він не встановлює патчі безпеки, в системі можуть існувати відомі вразливості безпеки, забезпечуючи вектор для компрометації системи.

Перевірку рекомендую здійснювати використовуючи команду «netstat –an», за її допомогою потрібно шукати рядки, що містять «LISTEN» або «LISTENING». Це порти Transmission Control Protocol (TCP) та User Datagram Protocol (UDP), на яких хост доступний для вхідних з’єднань. Також можна скористатися утилітою «LSOF», ввівши команду «lsof –i».

Коли отримаєте список ввімкнених служб, потрібно обговорити цей список із системним адміністратором, щоб зрозуміти необхідність кожної служби. Багато служб ввімкнено за замовчуванням. Будь-які служби, які не потрібні, рекомендую відключити їх.

Для того щоб бути на крок попереду і завжди бути в курсі нових вразливостей для ввімкнених служб та появи нових релізів патчів для усунення цих вразливих місць, рекомендую моніторити новини від організації Computer Emergency Response Team (CERT), вона відноситься до загальних джерел оголошень про виявленні нові вразливості. CERT охоплює гучні вразливості, але вам також слід отримувати сповіщення від вашої ОС та постачальників додатків програмного забезпечення, щоб забезпечити належне покриття. Інформація про цей процес може бути зібрана за допомогою інтерв’ю та огляду документації.

Якщо вам потрібно перевірити певний патч або версію пакета, ви можете переглянути встановлені пакети та виправлення за допомогою наступних команд:

- rpm -q –a (Red Hat або інші дистрибутиви, що використовують RPM);
- dpkg –list (Debian та пов’язані з ним дистрибутиви) покажуть версії встановлених пакетів;
- showrev -p (Solaris) перерахує застосовані патчі; вони можуть бути перехресними посиланнями на патчі, перелічені в рекомендаціях щодо безпеки від Sun.

Також потрібно розглянути конфігурацію служб, а не лише те, чи вони дозволені. Правильна конфігурація певних служб, таких як Network File System (NFS), File Transfer Protocol (FTP), а також ті, які дозволяють довірений доступ та кореневий вхід. Ведення обмежень не дозволяє нам детально описати правильну конфігурацію кожного потенційного сервісу (плюс постійно виявляються нові вразливості). Ось чому використання інструменту мережевого сканування є критично важливим компонентом ефективного аудиту. Такий інструмент дозволить швидко виявляти нові вразливості в системі.

2) Рекомендую запустити засіб сканування мережевих вразливостей для перевірки поточних вразливостей в середовищі.

Це забезпечить короткий знімок поточного рівня безпеки системи (з точки зору послуг мережі). В наш час постійно з’являються нові вразливості мережі, і створити програму статичного аудиту, яка надасть сучасний портрет вразливих місць, який слід перевірити, нереально. Тому інструмент сканування, часто оновлюється, і є найбільш реалістичним механізмом розуміння поточного стану безпеки машини.

Незважаючи на те, що багато з цих інструментів призначені для забезпечення безперебійної роботи і не вимагають доступу до системи, я б радив завжди інформувати відповідний IT-персонал (наприклад, системного адміністратора, мережеву команду і відділ IT-безпеки) про те, що ви плануєте запускати інструмент, а потім отримати їх схвалення і призначити час для запуску інструменту. Інструменти сканування можуть несподівано взаємодіяти з портом і викликати збої, тому важливо, щоб інші знали про ваші дії.

3) Перевірити використання довіреного доступу за допомогою файлів /etc/hosts.equiv та файлів користувача .rhosts. Ви повинні переконатися, що

довірчий доступ не використовується або, якщо він вважається абсолютно необхідним, обмежується наскільки це можливо.

Довірений доступ дозволяє користувачам отримувати віддалений доступ до системи без використання пароля. Зокрема, файл «/etc/hosts.equiv» створює довірчі відносини з конкретними комп'ютерами, в той час як файл «.rhosts» створює довірчі відносини з конкретними користувачами на певних машинах.

Наприклад, якщо у системі «Linux» є файл /etc/hosts.equiv, який відображає машину «Ubuntu» як довірений хост, то будь-який користувач із обліковим записом, що використовує те саме ім'я користувача в обох системах, зможе отримати доступ до «Linux» від «Ubuntu» без використання пароля. Таким чином, якщо ім'я користувача «Andrey» існує на обох машинах, власник облікового запису «Andrey» у «Ubuntu» зможе отримати доступ до облікового запису «Andrey» на «Linux» без використання пароля. Тут треба врахувати, що ключовим є ім'я облікового запису. Якщо у Ivana Ivaniva є обліковий запис на обох машинах, але на одному є ім'я облікового запису «Ivan», а в іншого – «Ivann», тоді довірчі відносини не працюватимуть. Операційна система не визнає їх одним і тим же обліковим записом.

Якщо система, яку ви ревізуєте, має довірчі відносини з іншими машинами, безпека довіряючої системи залежить від безпеки довіреної системи. Якщо облікові записи, яким довіряють, є скомпрометованими, то, за визначенням, облікові записи в системі, яку ви ревізуєте, також будуть скомпрометовані. Це так, тому що доступ до надійної машини забезпечує доступ до довіряючої машини. Рекомендую уникати такого роду залежності, якщо це взагалі можливо.

Довірений доступ також можна використовувати для обходу контролю над загальними обліковими записами. Як обговорювалося раніше, загальні облікові записи можуть бути заблоковані таким чином, щоб необхідно скористатися Su або Sudo для доступу. Однак, якщо користувач має доступ до загального облікового запису за допомогою одного з цих механізмів, а потім створює файл «.rhosts» для цього облікового запису, надає довірений доступ до свого особистого облікового запису, користувач зможе обійти необхідність використання Su або Sudo щоб отримати доступ до облікового запису.

Під час перевірки, в першу чергу раджу усунути довірений доступ. Якщо стане очевидним, що це неможливо в середовищі що перевіряється, то необхідно виконати дії направлені на зменшення можливих ризиків.

Для зменшення ризиків рекомендую дослідити вміст фалу «/etc/hosts.equiv» та будь-які файли «.rhosts» у системі. Вміст файлу «/etc/hosts.equiv» можна переглянути за допомогою команди «more /etc/hosts.equiv». Щоб знайти файл «.rhosts», вам потрібно буде переглянути вміст домашнього каталогу кожного користувача через команду «ls –l» (розташування домашніх каталогів користувача можна знайти у файлі паролів), щоб побачити, чи існує файл «.rhosts». Вміст знайдених файлів «.rhosts» можна переглянути за допомогою команди more. Якщо дозволи на доступ до файлів обмежують перегляд вмісту цих файлів, вам знадобиться системний адміністратор, щоб виконувати ці команди.

Обговоріть вміст цих файлів із системним адміністратором, щоб зрозуміти необхідність бізнесу для кожного запису. Рекомендовано видалити зайві записи або бажано взагалі виключити використання довіреного доступу. Для важливих довірчих відносин обговоріть можливість використання надійних ключів Secure Shell (SSH), яка, як правило, є кращою альтернативою «hosts.equiv» та «.rhosts».

Також необхідно переконатись, що жоден з файлів не містить знаку «+». Цей символ визначає всі системи в мережі як надійні та дає їм змогу ввійти в систему без використання пароля (якщо на довірчому сервері є рівнозначне ім’я користувача).

Якщо знак «+» існує у файлі «/etc/hosts.equiv file», то будь-який користувач (крім «root») у будь-якій системі мережі, який має те саме ім’я користувача, як і будь-який з облікових записів у довіряючій системі, зможе отримати доступ облікового запису без використання пароля. Якщо знак «+» існує у файлі «.rhosts», будь-який користувач у будь-якій системі мережі, який має те саме ім’я користувача, як власник файлу .rhosts, зможе отримати доступ до облікового запису без використання пароля. Сюди входить і обліковий запис «root», тому файл «.rhosts» з позначкою «+» у домашньому каталозі root зазвичай є дуже поганою ідеєю.

Рекомендую для будь-яких законних та необхідних довірчих відносин визначити, чи впевнений адміністратор, що кожна система, до якої надається довірений доступ, така ж безпечна, як і система, що перевіряється. Як було сказано раніше, безпека системи залежить від безпеки будь-якої системи, якій довіряють. адміністратори, як правило, не повинні надавати надійний доступ до систем, які вони не контролюють. У цьому випадку необхідно вжити заходів для забезпечення безпеки та цілісності систем, яким довіряють, або шляхом

проведення власних сканувань безпеки, або шляхом проведення інтерв'ю з системним адміністратором довіреної системи.

В ході аудити необхідно впевнитись, що файли «/etc/hosts.equiv» і «.rhosts» надійно захищені (використовуючи команду ls –la). Файл «/etc/hosts.equiv» повинен належати системному обліковому запису (наприклад, «root») і доступний для запису тільки для цього облікового запису. Якщо інші можуть записати у цей файл, вони можуть перелічити неавторизовані машини у списку надійних хостів. Файли «.rhosts» повинні належати тому обліковому запису, у домашньому каталозі якого вони знаходяться, і його слід записувати лише цим обліковим записом. Якщо користувач може записати у файл «.rhosts» іншого користувача, цей користувач може зробити себе чи когось іншого, довіривши вхід в акаунт цього користувача з іншої машини.

Також рекомендую переконатися, що в записах використовується повне доменне ім'я для довірених систем (наприклад, «trast.mlb.com» замість «trast»). Запис, який не використовує повне доменне ім'я, може бути машиною з тим же ім'ям хоста, але іншим доменом.

Для виявлення та перегляду будь-якого нового надійного доступу, встановленого в системі, потрібно дослідити процеси, які використовуються системними адміністраторами. Вони повинні виявляти та переглядати будь-які нові записи і фйлі «.rhosts» чи записи та будь-які нові записи «/etc/hosts.equiv».

4) Оцінити використання доступу через SSH-ключі.

Довірений доступ через SSH-ключі концептуально такий же, як і довірений доступ через файли «.rhosts», обговорені на попередньому етапі, і, як правило, більш рекомендований, якщо потрібен довірений доступ. Він дозволяє користувачам віддалено отримувати доступ до системи через SSH без використання пароля, створюючи довірчі відносини з конкретними користувачами на певних машинах.

Щоб встановити довірчі відносини за допомогою ключів SSH, користувач створює (або, швидше за все, використовує команду генерації ключів SSH) підкаталог у своєму домашньому каталозі на довіреній машині під назвою «.ssh» та розміщує два файли у цьому каталозі: «id_rsa» є приватним ключем і «id_rsa.pub» є відкритим ключем (якщо DSA використовується замість RSA, замініть rsa на dsa в цих іменах). Потім користувач розміщує текст з файлу відкритого ключа у файл під назвою «authorized_keys2» у підкаталозі «.ssh» домашнього каталогу на машині, до якої користувач хоче отримати доступ. Після

цього користувач зможе отримати доступ до довіряючої машини (машини, на якій він створив файл «authorized_keys2» у своєму домашньому каталозі) із надійної машини (машина, що містить файли відкритого та приватного ключа користувача) через SSH без використання пароля.

Під час перевірки потрібно вивчити зміст файлу «authorized_keys2», і переконатися що він є санкціонованим.

Обов'язково необхідно переконатися, що файли «authorized_keys2» та пов'язані з ними «.ssh» підкаталоги належним чином захищені(використовуючи команду ls –la). Вони повинні належати тому обліковому запису, в чиєму домашньому каталозі вони знаходяться і повинні бути доступні для запису тільки цьому обліковому запису. Раджу переконатися, що всі файли «id_rsa» в системі і пов'язаних з нею підкаталогах «.ssh» захищені належним чином.

Якщо користувач може прочитати файл приватного ключа іншого користувача, він може використовувати цю інформацію, щоб скомпрометувати іншого користувача і отримати доступ до довірчих відносин, які користувач встановив з іншими серверами.

5) Якщо вхід по FTP включений і дійсно необхідний, рекомендую переконатися, що він контролюється належним чином.

Вхід через FTP дозволяє будь-якому користувачеві мережі отримувати файли або надсилати файли до обмежених каталогів. Він не вимагає використання пароля, тому його слід правильно контролювати.

Щоб визначити, чи ввімкнено вхід по FTP, вивчіть вміст файлів паролів. Якщо ви бачите обліковий запис «ftp» у файлі пароля то послуга FTP увімкнена, анонімний FTP доступний у системі. Після входу FTP-користувача він обмежується лише тими файлами та каталогами в домашній директорії облікового запису «ftp», яка вказана у файлі пароля ftp. Обліковий запис «ftp» повинен бути відключений у файлі паролів і не повинен мати дійсну оболонку.

Використовуючи FTP, користувач стає користувачем «ftp». Якщо у «ftp» є власні файли та каталоги, кожен, хто використовує FTP, може змінити дозволи файлу на все, що належить «ftp». Тому потрібно впевнитись, що каталог FTP (/ftp) належить та записується лише «root», а не «ftp».

Рекомендую дослідити дозволи каталогу /ftp та підкаталогів (використовуючи команду ls):

- у каталозі «/ftp/pub» повинен бути встановлений stick біт, щоб люди не могли видаляти файли в каталозі;
- каталог «/ftp» та інші його підкаталоги повинні бути встановлені з дозволами, принаймні такими ж обмежувальними, як і «dr-xr-xr-x» так що користувачі не можуть видаляти та замінювати файли в каталогах.

Також корисно буде переконатися, що у файлі «/ftp/etc/passwd» немає записів користувачів (лише «ftp») або паролів (виконавши команду more у файлі). В іншому випадку будь-хто в мережі може бачити імена користувачів на сервері та використовувати їх для атаки на систему.

Зловмисники можуть перенести великі файли в каталоги /ftp та заповнити файлову систему (здійснити атаку відмови у наданні послуги або запобігти запису журналів аудиту). Рекомендую системному адміністратору розглянути можливість розміщення квоти на файли користувача «ftp» або розміщення домашнього каталогу «/ftp» в окремій файловій системі.

6) Дослідити використання захищених протоколів.

Окремі протоколи (наприклад, Telnet, FTP, remote shell (rsh), rlogin, and remote copy (rcp)) передають всю інформацію у вигляді тексту, включаючи UID та паролі. Це може дозволити комусь отримати цю інформацію шляхом підслуховування в мережі.

Тому рекомендую переглянути список увімкнених служб та визначити, чи є серед них telnet, ftp, rsh, rlogin та rcp. Якщо вони ввімкнені, то за допомогою інтерв'ю з системним адміністратором визначте можливість їх вимкнення або заміни захищеними (зашифрованими) альтернативами. Telnet, rsh та rlogin можуть бути замінені SSH; FTP може бути замінений протоколом Secure File Transfer Protocol (SFTP) або протоколом Secure Copy (SCP); і rcp також можна замінити SCP.

7) Оцінити використання файлів «.netrc».

Файли «.netrc» використовуються для автоматизації входу в систему. Якщо конфіденційний пароль розміщено в одному з цих файлів, цей пароль може бути відкритий для інших користувачів системи.

Наприклад за допомогою наступної команди можна здійснити пошук та друк вмісту всіх файлів «.netrc» в системі. Вам, можливо, знадобиться системний адміністратор, щоб запустити цю команду для пошуку:

#find / -name '.netrc' -print -exec more {} \.

Для будь-яких знайдених файлів «.netrc» перегляньте їх зміст. Якщо доступ для читання обмежений, вам буде потрібен системний адміністратор, щоб зробити це за вас. Знайдіть вказівки на паролі, що розміщуються в цих файлах. Якщо ви знайдете їх, перегляньте права доступу до файлів за допомогою команди «ls –la» і переконайтеся, що ніхто крім власника не може читати файл. Навіть якщо права доступу до файлу заблоковані, будь-який користувач з правами суперкористувача зможе прочитати файл, тому краще взагалі не використовувати ці файли. Однак, якщо вони існують і вони необхідні, аудитор повинен переконатися, що вони були максимально захищенні.

8) Переконатись, що інформаційний банер попередження відображається, коли користувач підключається до системи.

Офіційне повідомлення про вхід – це попередження, яке відображається кожного разу, коли хтось намагається підключитися до системи. Це попередження повинно відображатися перед фактичним входом у систему, і в основному містить наступну інформацію: «Вам заборонено користуватися цією системою, якщо не має дозволу». Повідомлення подібного роду може знадобитися для притягнення до відповідальності зловмисників у суді.

Для перевірки потрібно ввійти у свій обліковий запис за допомогою кожного доступного механізму, що забезпечує доступ до оболонки, наприклад, Telnet або SSH. Визначити, чи відображається попереджувальний банер. Текст цього банера часто міститься у таких файлах, як «/etc/issue» та «/etc/sshd_config» (або /etc/openssh/sshd_config). Рекомендую, щоб текст для цього застережливого банера розроблявся спільно з юридичним відділом компанії.

2.4 Журнали аудиту

В системних журналах зберігається вся інформація про все, що відбувається у системі, тому їх перевірка е дуже важливою частиною проведення аудиту. Для детальної перевірки цих файлів рекомендую виконати наступні кроки.

1) Переглянути елементи контролю для запобігання прямих «root» входів.

Зазвичай декілька людей знають пароль «root», якщо їм дозволяється входити в аккаунт безпосередньо як root, відповідальності за дії, виконані цим

обліковим записом, не існує. Якщо невідповідні дії виконуються в обліковому записі root, не буде можливості відстежити ці дії до конкретного користувача. Тому рекомендую, щоб користувачі спочатку входили в систему під своїм іменем, а потім використовувати команди su або sudo для доступу до облікового запису root.

При перевірці потрібно дослідити журнал «wtmp» (виконавши команду more для «/usr/adm/wtmp», «/var/adm/wtmp», або «/etc/wtmp», залежно від типу системи), щоб перевірити відсутність прямих root входів. Або рекомендую використати команду «last», вона може використовуватися для перегляду вмісту цього файлу в більшості систем.

Також потрібно переглянути налаштування для запобігання прямого root входу через telnet і rlogin. У системах Linux файл «/etc/securetty» можна використовувати для запобігання прямих входів як root. Вміст файлу повинен міститись у всіх терміналах, яким дозволений прямий «root» вхід. Файл повинен існувати, але бути порожнім. Іноді системному адміністратору необхідно дозволити прямий «root» вхід з консольного терміналу. Це можна робити, якщо консоль знаходиться в надійному місці. Вміст цього файлу можна переглянути за допомогою виконання команди «more /etc/securetty».

Обов’язково необхідно переглянути налаштування SSH, для запобігання прямих root входів. Для цього використовується файл «/etc/sshd_config» або «/etc/openssh/sshd_config». Потрібно переглянути вміст цих файлів, та знайти параметр «PermitRootLogin». Якщо цей параметр встановлюється значення «no», «root» реєстрація не дозволяється. Якщо цього параметра немає або встановлено значення «yes», реєстрація «root» дозволена.

2) Дослідити журнали команд su та sudo, щоб переконатися, що при використанні цих команд вони реєструються з датою, часом та користувачем, який вводив команду.

Команда su – це інструмент, який часто використовують зловмисники, щоб спробувати проникнути в обліковий запис користувача. А команда sudo дозволяє авторизованим користувачам виконувати певні команди так, ніби вони «root». Через це критично важливо щоб для обох команд відбувалась реєстрація, для забезпечення підзвітності та допомоги у розслідуванні.

Тому рекомендую особливо уважно дослідити зміст журналу su. Цей журнал буде можна знайти в «/usr/adm/sulog», «/var/adm/sulog», або «/var/log/auth.log». Під час перегляду, потрібно впевнитись, що цей файл існує та

в ньому зберігається інформація використання su(наприклад, хто виконував команду, на який обліковий запис вони перейшли, дата та час виконання команди).

3) Дослідити журнали реєстрацій подій.

Журнали реєстрації подій. Система проколювання системних подій, вважається однією із найбільших переваг ОС Linux. Беручи до уваги те, що цій операційній системі відбувається велика кількість процесів, то це і справді дуже важливо. Іноді при налаштуванні нового сервісу нічого не працює і не зрозуміло, в чому саме причина. Або виникає необхідність стежити за спробами несанкціонованого доступу до системи. У будь-якому випадку журнали дозволять знайти причину проблеми або отримати необхідну інформацію.

Тому під час проведення аудиту, переглядаємо журнали подій, а також визначаємо періодичність та регулярність їх аналізу. Рекомендую використовувати централізоване зберігання журналів подій на виділеному сервері. Це може перешкодити зловмисникам легко змінювати локальні журнали. В ході аудиту також потрібно впевнитись, що політика збереження журналів подій враховує вимоги чинного законодавства.

Більшість лог-файлів міститься в директорії / var/log:

- /var/log/syslog – глобальний системний журнал, в якому пишуться повідомлення з моменту запуску системи, від ядра Linux, різних служб, виявлених пристроях, мережевих інтерфейсів і багато іншого;
- /var/log/auth.log – журнал системи авторизації (логіни та механізм перевірки автентичності);
- /var/log/yum.log – журнал установки / видалення пакетів з використанням yum;
- /var/log/faillog – записи невдалих спроб входу в систему і їх граничного числа для кожної облікового запису;
- /var/log/maillog або /var/log/mail.log – журнал поштового сервера;
- /var/log/wtmp – журнал входу в систему (час реєстрації і тривалість роботи всіх користувачів системи);
- /var/run/utmp – інформація про активних користувачів, в даний момент часу;
- /var/log/lastlog – записи про попередні входах в систему.

4) Оцінити безпеку збереження журналів реєстрації подій.

Якщо журнали аудиту не захищені, то неавторизовані користувачі можуть змінити їх вміст, що призведе до зниження корисності журналів під час розслідувань. Крім дослідження цих журналів раджу налаштувати процедуру зберігання журналів протягом достатнього періоду часу, оскільки в разі необхідності адміністратор може бути не в змозі розслідувати невідповідні дії та інші системні проблеми, якщо це необхідно.

Щоб визначити термін зберігання журналів потрібно опитати системного адміністратора. Як правило, бажано зберігати ці журнали від трьох до шести місяців, щоб забезпечити отримання інформації під час розслідувань.

2.5 Моніторинг безпеки та загальний контроль

Щоб перевірити систему моніторингу безпеки, я пропоную виконати ряд рекомендацій, котрі дозволять зробити перевірку більш повною та ефективнішою.

1) Оцінити процедури адміністратора системи для моніторингу стану безпеки в системі.

Якщо системний адміністратор не має процесів для моніторингу безпеки, можуть існувати дірки в безпеці, а інциденти з безпекою можуть статися без його відома.

Для перевірки потрібно опитати системного адміністратора та переглянути будь-яку відповідну документацію, щоб зрозуміти методи моніторингу безпеки. Можна виконувати різні методи моніторингу безпеки. Хоча всі їх не потрібно виконувати, але повинен існувати певний рівень моніторингу, який повинен відповідати критичності системи та властивому ризику системи що перевіряється. Наприклад, веб-сервер котрий знаходиться в Demilitarized Zone (DMZ) повинен мати більш надійний моніторинг безпеки, ніж сервер друку у внутрішній мережі.

Далі розглянемо чотири основні рівні моніторингу. Потенційні інструменти для здійснення таких типів моніторингу обговорюються в підрозділі «Інструменти та технології» цього розділу.

Перший рівень це сканування вразливостей мережі Це, мабуть, найважливіший тип моніторингу безпеки в більшості середовищ. Він відстежує можливі вразливості, які можуть дозволити будь-кому, отримати доступ до системи або порушити цілісність системи. Оскільки ці вразливості можуть використовувати будь-хто в мережі, вам потрібно знати про них і закривати їх.

Другий – сканування вразливостей на основі хоста Це сканування на вразливості, які дозволять особі, яка вже є в системі, ескалацію своїх привілеїв (наприклад, використання облікового запису root), отримання невідповідного доступу до конфіденційних даних (наприклад, через погано встановлені дозволи файлів), або порушити систему.

Третій рівень заключається у виявленні вторгнень. Цей моніторинг виявляє несанкціонований вхід (або спроби несанкціонованого входу) у систему. Інструменти моніторингу базової лінії (такі як Tripwire) можуть використовуватися для виявлення змін критичних файлів, а засоби моніторингу журналів можуть використовуватися для виявлення підозрілих дій через системні журнали.

Останій полягає в запобіганні вторгнень. Моніторинг такого типу виявляє спробу атаки і зупиняє її до того, як вона скомпрометує систему.

Якщо моніторинг безпеки здійснюється, необхідно оцінити частоту моніторингу та якість, з якою він проводиться. Потрібно знайти докази того, що засоби моніторингу безпеки фактично використовуються та діють. Для цього необхідно переглянути останні результати та визначити чи були вони розслідувані. Отримані результати решти аудиту використовуйте при виконанні цієї оцінки. Наприклад, якщо ви знайшли суттєві проблеми в зоні, яка нібито контролюється, це може призвести до питань ефективності цього моніторингу.

2) Якщо ви проводите аудит великого середовища Unix/Linux (на відміну від однієї або двох ізольованих систем), необхідно визначити, чи існує стандартна збірка для нових систем і чи має вона адекватні параметри безпеки. Здійсніть аудит новоствореної системи.

Одним з найкращих способів поширення безпеки в усьому середовищі є забезпечення правильної побудови нових систем. Таким чином, у міру розгортання нових систем необхідно впевнитись, що вони спочатку мають відповідний рівень безпеки.

За допомогою інтерв’ю з системним адміністратором необхідно визначити методологію, що використовується для побудови та розгортання нових систем. Якщо використовується стандартна збірка, проведіть аудит новоствореної системи, використовуючи кроки описані в цьому розділі.

2.6 Інструменти та технології

Спільнота вільного програмного забезпечення надала численні цінні інструменти, якими аудитор може скористатися для підвищення як точності, так і ефективності своєї роботи. Нижче мною відібрані та описані деякі найпоширеніші інструменти для аудиту * nix систем з кількома порадами щодо їх використання.

1) Nessus – це сканер мережевих вразливостей з відкритим вихідним кодом, який описує окремі загрози і потенційні атаки. Значні можливості Nessus включають в себе [10]:

- сумісність з комп'ютерами і серверами всіх розмірів;
- виявлення дірок в безпеці на локальних або віддалених хостах;
- виявлення відсутніх оновлень безпеки і патчів;
- імітація атак для виявлення вразливостей;
- виконання тестів безпеки в ізольованому середовищі;
- заплановані перевірки безпеки.

2) Network Mapper (NMAP) – інструмент для дослідження мережі та перевірки безпеки. NMAP може бути зручним способом перевірити наявність відкритих портів на сервері, не запускаючи сканер уразливості, наприклад Nessus, також можливо, перевірити правила брандмауера на основі хоста.

3) John the Ripper. Це утиліта для виявлення слабких паролів користувачів операційних систем. При цьому в даний час версія Jumbo (з розширеним функціоналом від спільноти) перетворилася в потужний комбайн, який може підбирати паролі для різноманітних архівів, документів, електронних гаманців і багато чого ще [11].

4) Tiger. Виконує аудит безпеки, автоматично скануючи машину на наявність поганих файлів налаштувань, змінених програм і інших потенційних проблем захисту. Tiger контролює такі аспекти системи [12]:

- записи планувальника cron;
- поштові псевдоніми;
- експорт файлових систем NFS;
- записи демона inetd / xinetd;
- змінні оточення PATH;
- файли rhosts і .netrc;

- права доступу до певних файлів і каталогів.

Tiger також переглядає файлові системи з метою виявлення незвичайних файлів і перевірки шляхів, записаних в файлах, виявлених в результаті попередніх перевірок. Під час роботи програма робить записи в файл про всі виявлені слабкі місця в системі. Можна подивитися цей файл, виправити знайдені похибки в системі захисту, а потім запустити інструментарій знову, для того щоб перевірити, чи все було зроблено правильно. Можна навіть здійснити настройку таким чином, щоб Tiger запускався періодично і відсилав результати своєї роботи за e-mail.

5) Linux Malware Detect (LMD). LMD – це сканер шкідливих програм для Linux. Однією з головних особливостей LMD є виявлення шкідливого вмісту в файлах web-сайтів, на відміну від безлічі інших антивірусів, які розроблені з урахуванням виявлення загроз на рівні операційної системи. Крім того він використовує власну базу сигнатур [14].

6) Nikto – це сканер з відкритим вихідним кодом для веб-серверів, він виконує комплексні тести щодо серверів за кількома напрямками, включаючи понад 6700 потенційно небезпечних файлів / програм, перевірка на застарілі версії більше 1250 серверів і проблеми, специфічні для версій більш ніж 270 серверів. Сканер також перевіряє елементи конфігурації сервера, такі як присутність декількох індексних файлів, серверні опції HTTP (HyperText Transfer Protocol)і намагається визначити ім’я і версії веб-сервера і програмного забезпечення.Він надає повну систему звітів для перегляду поточних і попередніх результатів сканування, підтримує оповіщення по електронній пошті після кожного виконання сканування і багато інших корисних функцій [15].

7) Netcat – утиліта Unix, що дозволяє встановлювати з’єднання TCP і UDP, приймати звідти дані і передавати їх. Незважаючи на свою корисність і простоту, багато хто не знає способи її застосування і незаслужено обходять її стороною [16].

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ПРОВЕДЕННЯ АУДИТУ ВЕБ-СЕРВЕРІВ

3.1 Основи веб-аудиту

Мало технологічних винаходів змінювали наше життя так само швидко, як і веб-додатки. Веб-інтерфейс виріс з статичних сторінок в неймовірно інтерактивну суміш можливостей, якими управляє армія креативних програмістів. У 1991 року в США був встановлений перший веб-сервер для зв'язку з комп'ютером NeXT в Швейцарії.

Власне Web сервер – це програмне забезпечення, яке здійснює взаємодію по HTTP протоколу з браузерами: прийом запитів, пошук вказаних файлів і передача їх вмісту, запуск CGI-додатків і передача клієнту результатів їх виконання[17].

Безпека Web сервера являє собою лише невеликий компонент загальної системи безпеки хоста Internet. Під словами «злом сервера» найчастіше мається на увазі заміна або модифікація сторінок Web сервера – найбільш видовищний прояв атаки на сервер, хоча насправді може виявитися лише побічним явищем захоплення управління всім хостом.

У той же час існують проблеми безпеки, характерні саме для Web серверів. Скориставшись ними, зловмисник може отримати стартовий майданчик для подальшого проникнення в систему (тому як і раніше залишається в силі рекомендація винести Web сервер, як і всі інші служби, які потребують зовнішнього доступу, на окрему машину, по можливості ізольовану від внутрішньої мережі).

Повний веб-аудит – це аудит трьох основних компонентів, включаючи операційну систему сервера, веб-сервер та веб-додаток. Перший компонент, який ми обговорюємо – це базова платформа або операційна система, на якій встановлені і працюють веб-сервер і додаток (аудит операційної системи ми розглянули в попередньому розділі). Далі йде сам веб-сервер, такий як Internet Information Services (IIS) або Apache, який використовується для розміщення веб-програми. Нарешті, ми охоплюємо аудит веб-додатку. Веб-додаток для наших цілей включає в себе відповідні середовища розробки, такі як ASP.NET, Java, Python або PHP, та Content management system (CMS), такі як Drupal, Joomla або WordPress [17].

3.2 Рекомендації щодо проведення аудиту веб-сервера

1) Перевірити, що доступ до веб-сервера дозволений лише відповідним протоколам та портам.

Мінімізація кількості протоколів та портів, дозволених отримати доступ до веб-сервера, зменшує кількість векторів атак, доступних для компрометації сервера.

Тому за допомогою системного адміністратора необхідно переконатися, що що доступ до сервера дозволений лише необхідним протоколам. Зверніть увагу на будь-які додаткові елементи управління, такі як правила брандмауера або Access Control Lists (ACL), щоб обмежити протоколи та порти, дозволені для доступу до веб-сервера.

2) Переконатись, що облікові записи, що дозволяють отримати доступ до веб-сервера, належним чином контролюються та мають надійний пароль.

Неправильно керовані або використані облікові записи можуть забезпечити легкий доступ до веб-сервера, минаючи інші додаткові засоби безпеки, що запобігають атакам.

Взагалі рекомендую видалити з сервера або повністю відключити облікові записи котрі не використовуються.

Особливо свою увага потрібно звернути на обліковий запис «root». На хостах з операційною системою Linux повинен суворо контролюватися і ніколи не використовуватися для прямого віддаленого адміністрування. І взагалі сервером і веб-сервером завжди повинна застосовуватись сильна політика щодо облікових записів та паролів.

3) Дослідити реєстрацію подій, що веб-сервері.

Реєстрація аудиторських подій допомагає адміністраторам вирішувати багато різноманітних проблем. Тому необхідно переконатися в тому, чи відбувається збереження системних подій, наприклад, невдалі спроби входу. В ідеалі ці журнали повинні бути переміщені та захищені на іншому місці, ніж веб-сервер. Файли журналів також повинні регулярно архівуватися та аналізуватися.

4) Дослідити, розширення скриптів та впевнитись, що вони відображаються відповідним чином.

Скрипти можуть дозволити зловмиснику виконати код на свій вибір, потенційно компрометуючи веб-сервер. Переконайтеся у веб-адміністратора, що

розширення скриптів, які не використовуються веб-сервером, прив’язані до обробника 404 сторінки або просто заборонені.

5) Перевірити протоколи і порти, за допомогою яких здійснюється доступ до веб-сервера.

Мінімізація кількості протоколів та портів, дозволених отримати доступ до веб-сервера, зменшує кількість векторів атак, доступних для компрометації сервера.

В якості перевірки рекомендую обговорити та переконатись з допомогою адміністратора, що доступ до сервера дозволений лише необхідним протоколам. Наприклад, стек TCP/IP на сервері повинен бути посилений, щоб дозволити лише відповідні протоколи. Зверніть увагу на будь-які додаткові елементи управління, такі як правила брандмауера або мережеві списки контролю доступу (ACL), щоб обмежити протоколи та порти, дозволені для доступу до веб-сервера. Загалом, доступ до веб-сервера повинен мати лише TCP на портах 80 (HTTP) і 443 (SSL). Крім того, в деяких випадках буде корисно переглянути узгоджені шифри, дозволені транзакціями SSL.

6) Дослідити елементи управління файлів, каталогів та віртуальних каталогів.

Невідповідні елементи керування файлами та каталогами, які використовуються веб-сервером та системою загалом, дозволяють зловмисникам отримати більше інформації. Наприклад, утиліти віддаленого адміністрування збільшують ймовірність компрометації веб-сервера.

В якості перевірки рекомендую переконатися, що файли та каталоги мають відповідні дозволи, особливу увагу потрібно звернути на файли і каталоги що мастять наступну інформацію:

- контент;
- скрипти;
- системні файли (наприклад,% windir% \ system32 або каталоги веб-сервера);
- набори інструментів, утиліт та програмне забезпечення.

4 ПРИКЛАД АУДИТУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ СЕРВЕРА

Мета цього практичного прикладу – продемонструвати, як проводити аудит веб-сервера, виконуючи як огляд робочих процедур безпеки, так і оцінку вразливостей безпеки веб-сервера. Ця практика складається з чотирьох основних етапів, які включають дослідження, розробку формального і повторюваного контрольного списку аудиту, проведення аудиту на живих веб-серверах і розробку звіту, призначеного для управління.

Контрольний список аудиту був розроблений з використанням особистого досвіду, існуючих контрольних списків аудиту, а також безкоштовних і відкритих інструментів безпеки.

4.1 Дослідження системи з точки зору аудиту

Об’єктом перевірки, є фінансовою установою середнього розміру, з великою кількістю конфіденційної інформації. Будемо посилатися на неї, використовуючи вигадане ім’я «Intra». В Intra ми перевіримо три критично важливих веб-сервера, на яких працюють критично важливі сервіси, такі як корпоративний веб-сайт, додаток для ділових партнерів і пошта. Ці веб-сервери розташовані в демілітаризованій зоні між двома міжмережевими екранами. Перед зовнішнім брандмауером також знаходиться маршрутизатор, який допомагає фільтрувати частину трафіку від зовнішнього брандмауера, щоб захистити від атак типу Denial of Service (DoS), заміни пакетів і невикористовуваних портів. На рис. 4.1, зображена базова архітектура мережі.

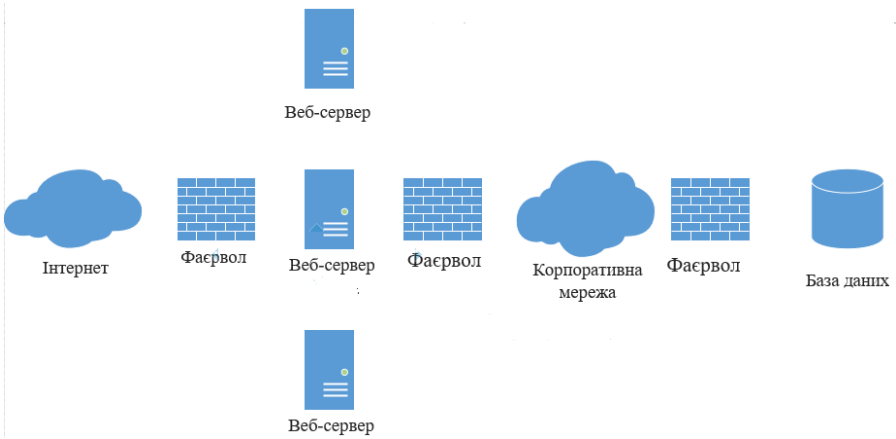

Рисунок 4.1 `– Базова архітектура мережі

Сервери розташовані на операційні системі CentOS 8. Для мережевих пристроїв, таких як брандмауери, маршрутизатори і комутатори, в основному використовується пристрої Cisco. Intra також використовує Snort для виявлення вторгнень і має центр керування мережею (NOC) для моніторингу трафіку, також є група реагування на комп'ютерні інциденти (CIRT) для обробки і розслідування підозрілих дій.

Почтовий сервер забезпечує доступ до електронної пошти для віддалених користувачів. За допомогою додатка, ділові партнери входять в цю систему і вводять і подають запит на різноманітну інформацію про клієнтів. Корпоративний веб-сайт Intra в першу чергу є інформаційним веб-сайтом, який містить інформацію про послуги та продукти Intra. Intra активно використовує свій корпоративний веб-сайт в маркетингових цілях для просування своїх продуктів і послуг.

Безпечна конфігурація цих систем не допустить несанкціонований доступ до конфіденційних даних і допоможе захистити від відомих вразливостей безпеки, таких як атаки переповнення буфера і неправильної конфігурації операційних систем і веб-додатків.

На цих веб-серверах розміщуються додатки, доступні будь-кому в Інтернеті. Оскільки зовнішні брандмауери зазвичай пропускають трафік HTTP / HTTPS для зв'язку з веб-серверами, веб-атаки стали дуже частим явищем. Це створює надзвичайно високий ризик для будь-якої компанії, яка покладається на веб-додатки для ведення бізнесу. Загроза безпеки також залишається постійною, тому що тисячі користувачів знаходяться на цих веб-серверах постійно.

Оскільки нові вразливості виявляються щодня, стовідсоткової гарантія безпеки не існує. Те, що сьогодні здається безпечним, завтра може бути абсолютно небезпечним. Важливо, щоб компанії робили безпеку пріоритетом і продовжували оновлювати виправлення безпеки та оновлювати політики і процедури в міру необхідності. Загальна мета безпеки для цих систем полягає в тому, щоб забезпечити безпечне налаштування операційної системи і додатки та наявність сучасних виправлень для захисту від відомих вразливостей безпеки.

Кожна з трьох систем має різні типи ризиків, пов'язаних з ними. Використовуючи інформацію з плану безперервності бізнесу, Intra вважає, що ці сервери містять свої найбільш критичні і високоризикові додатки. Почтовий сервер зберігає більшу частину електронної пошти організації, багато з якої містить інформацію, що стосується їх бізнесу і клієнтів. Якби цей сервер був

скомпрометований, зловмисник мав би доступ до корпоративних електронних листів і потенційно комерційної і конфіденційної інформації. Ця інформація може бути використана для соціальної інженерії, крадіжки особистих даних і інших узгоджених атак проти організації.

Якщо корпоративний веб-сайт Intra буде зіпсований, ділова репутація фірми може бути поставлена під загрозу, і вона може втратити довіру з боку своїх клієнтів і ділових партнерів. Intra також може зазнати великих фінансових втрат.

Якщо буде зламано додаток, то зловмисник може потенційно отримати доступ до внутрішньої бази даних, яка зберігає конфіденційну про клієнта, таку як контактна інформація клієнта, номери рахунків та іншу особисту фінансову інформацію.

Працюючи з персоналом Intra, була розроблена таблиця 4.1 «Аналіз зовнішніх загроз», в якій аналізуються і ідентифікуються різні типи зовнішніх загроз, з якими може зіткнутися фірма, а також був визначений рівень ймовірність атаки.

Таблиця 4.1 – Аналіз зовнішніх загроз

№	Загроза	Опис	Ймовірність	Коментарі
1	2	3	4	5
1	Хакери	Особи за межами системи, які можуть спробувати проникнути і використовувати мережеві системи Intra в пошуках обчислювальних ресурсів, інформації або фінансової вигоди.	Середня	Intra– це доволі велика фінансова установа, яку можна легко знайти в Інтернеті. Адекватні заходи захисту системи і процедури ескалації допомагають знизити цей ризик.
2	Шкідливий код	Комп'ютерний вірус або програма, призначена для порушення роботи комп’ютера.	Висока	Користувачі мають можливість завантажувати файли і файли з Інтернету і отримувати віруси по електронній пошті.

Продовження таблиці 4.1

1	2	3	4	5
3	Конкуренти	Інші фінансові установи на тій же арені в пошуках цінної інформації про клієнтів або секретів фірми.	Висока	Intra є не найменшою фінансовою установою яка давно існує, і тому може стати цінним інформаційним ресурсом для конкурентів.
4	Співробітники (колишні та дійсні)	Співробітники, які можуть мати проблеми всередині компанії, або які могли недавно піти через внутрішній конфлікт.	Середня	Колишні та дійсні співробітники можуть завдати проблем якщо не контролювати доступ до критичних систем і даних. Вплив зменшується відповідно до типу і ступеня доступу співробітника, а також надійною політикою керуванням доступом і обліковими записами.
5	Постачальники або підрядники	Постачальники та підрядники можуть забезпечувати обслуговування системи і / або брати участь в адмініструванні. У деяких випадках послуги можуть надаватися через віддалений доступ або пряме підключення до корпоративної мережі.	Середня	Постачальники і підрядники зазвичай мають привілейований доступ до систем і можуть ненавмисно викликати збої в обслуговуванні, якщо їх не контролювати належним чином. Конкретні рекомендації та обмеження допомагають зменшити ймовірність такого типу подій.

Для пошуку інформації пов’язаної з аудитом операційних систем та аудитом веб-серверів, використовувалась пошукова система Google. Крім того, я дослідив наступні джерела пов’язаних з аудитом безпеки і сайти інформаційної безпеки.

1) Система автоматизації аудиторської діяльності AuditNET. Система AuditNET пропонує комплексну автоматизацію аудиторської діяльності та застосовується для вирішення завдань. Це стандартизація методології та документації аудиторії, зберігання архіву робочих документів, відстеження постійних файлів клієнтів і управління договірними взаємозв’язками, управління структурою компанії, ведення картотеки співробітників, навчання робочих моментів, аналіз, планування та управління оперативною службою, планування та контроль проведення перевірок, контроль якості роботи, підготовка, планування та проведення аудиту.

2) Center for Internet Security (CIS). Центр інтернет-безпеки (CIS) є некомерційною організацією, яка розробляє власні контрольні показники і рекомендації, які дозволяють організаціям удосконалювати свої програми забезпечення безпеки та відповідності вимогам. Ця ініціатива спрямована на створення базових рівнів конфігурації безпеки систем, які зазвичай зустрічаються у всіх організаціях [20].

3) Information Systems Audit and Control Association (ISACA). Асоціація ISACA – міжнародний провайдер знань, сертифікацій, спільнот і навчання в областях аудиту та безпеки інформаційних систем, корпоративного керівництва і управління інформаційними технологіями та IT-ризиків і відповідністю вимогам. Асоціація фокусується на аудиті, безпеки та управлінні IT, а також надає сертифікації [21];

- аудитора інформаційних систем (Certified Information Systems Auditor – CISA);
- менеджера інформаційної безпеки (Certified Information Security Manager – CISM);
- з управління корпоративними IT (Certified in the Governance of Enterprise IT – CGEIT);
- з управління ризиками використання інформаційних систем (Certified in Risk and Information Systems Control – CRISC).

4) Open Source Security Testing Methodology Manual (OSSTMM). OSSTMM – формалізоване і добре структуроване керівництво для проведення тестування мережі.

Дана методологія дозволяє провести повноцінне тестування і стандартизацію мережі. Містить «карту безпеки», яка використовується в якості візуального показника. На карті вказуються основні галузі безпеки, які включають в себе набори елементів, що підлягають тестуванню:

- інформаційна безпека;
- тестування процесу безпеки;
- тестування технології веб-безпеки;
- тестування безпеки каналів зв’язку;
- тестування безпеки бездротових технологій;
- тестування фізичної безпеки.

5) SANS Security Policy Project. Це дослідницький проект спільноти SANS, котрий містить великий репозиторій готових політик безпеки на різні випадки життя, які розповсюджуються безкоштовно. Також в ньому можна знайти цікаві посилання на ресурси, присвячені розробці політик безпеки.

Я виявив, що існує цілий ряд ресурсів для аудиту веб-серверів як з точки зору безпеки, так і з точки зору оцінки вразливості. Однак під час мого дослідження я не зміг знайти жодного ресурсу, який би об’єднував обидва в одному аудиті. При технічній перевірці і перевірці конфігурації я в основному спирався на критерії безпеки, розроблені Center for Internet Security, і Open Source Security Testing Methodology Manual. Я гадаю, що комбінація цих двох ресурсів забезпечила надійну методологію як для безпечного налаштування систем, так і для їх періодичного аудиту на наявність вразливостей безпеки.

З точки зору процесів або процедур, які можуть вплинути на безпеку системи, таких як контроль доступу та керування виправленнями, я розробив безліч контрольних списків аудиту з AuditNet. Однак я не знайшов ніяких контрольних списків, які можна було б просто використовувати без особливих налаштувань. Наприклад, я не знайшов ніяких контрольних списків або процедур для управління виправленнями безпеки, тому я склав контрольний список управління змінами для задоволення потреб мого аудиту безпеки.

4.2 Створення чек-листа аудиту

В даній роботі здійснюється аудит безпеки трьох критично важливих веб-серверів. Ці сервери розташовані в демілітаризованій зоні між двома міжмережевими екранами і фізично розташовані в центрі обробки даних з обмеженим доступом.

В роботі розроблений чек-лист для здійснення аудиту, який складається з вісімнадцяти кроків та розділений на три окремі категорії.

1) Планування та адміністрування.
2) Перевірка операцій безпеки.
3) Тестування вразливостей.

Планування і адміністрування, а також перевірка операцій безпеки повинні проводитися з ключовими IT-фахівцями компанії, такими як директор з безпеки, мережеві інженери і системні адміністратори. А деякі етапи аудиту повинні відбуватися в неробочий час, через існування потенційного впливу тестування на роботу системи.

Метою даного аудиту є виявлення відомих слабких сторін і вразливостей конфігурації безпеки у зазначених веб-серверів. Крім того, в ході аудиту будуть розглянуті важливі процедури безпеки, які повинні бути прийняті для підтримки зусиль щодо забезпечення безпеки і для захисту конфіденційності, цілісності та доступності інформації. У таблиці 4.2 наведений чек-лист проведення аудиту.

Таблиця 4.2 – Контрольний чек-лист аудиту

№	Мета перевірки	Рекомендації щодо аудиту	Ризики
1	2	3	4
ПЛАНУВАННЯ ТА АДМІНІСТРУВАННЯ			
1	Визначте масштаб проекту, оцінки вразливостей і визначте критично важливі сервери, які будуть перевірятися.	Опитайте персонал і переглянути відповідні документи, щоб отримати докладне уявлення про бізнес-цілі, ризики для досягнення цих цілей і засобах управління, які знижують ризик.	Розуміння бізнес-цілей клієнта, пов’язаних з ним ризиків і заходів безпеки є невід’ємним компонентом аудиту і безпосередньо вплине на результати аудиту.

Продовження таблиці 4.2

1	2	3	4
1		Необхідно визначити і належним чином задокументувати бізнес-цілі і критичні системи.	Без чіткого визначення масштабу проекту, можливо, що проект не буде завершено вчасно або що очікування не будуть повністю виконані.
2	Збір інформації про веб-сервери (встановлені сервіси, версія додатків, IP-адреса, схема мережі, відкриті порти і т.д.).	Провести інтерв'ю з співробітниками, щоб зібрати всю необхідну інформацію про веб-сервери. Документація веб-сервера, така як схема мережі, інформація про відкриті порти, IP-адреси, повинні бути задокументованими і бути актуальними.	Без цих документів існує ймовірність того, що перевірки можуть ненавмисно вплинути на сервери, які виходять за межі області аудиту або взагалі належать іншій організації.
3	Дослідити політики безпеки і процедури, які стосуються системи що перевіряється.	Провести інтерв'ю з клієнтом, щоб зібрати відповідні правила і процедури. Визначити, чи існують політики інформаційної безпеки. Переглянути політики і переконайтеся, що вони: - реалізовані і здійсненні; - короткі і легкі для розуміння; - існує баланс захисту з продуктивністю.	Політики і процедури інформаційної безпеки є інструкціями для співробітників про те, як захистити цінні інформаційні та технологічні ресурси компанії, і є фундаментальним компонентом будь-якої програми інформаційної безпеки.

Продовження таблиці 4.2

1	2	3	4
3		Політика також повинна: - містити причини, за якими потрібна; - містити інформацію, на що поширюється; - визначати обов'язки; - містити інструкцію розглядання порушеннь.	
4	Визначити час і дату тестування вразливостей мережі і скоординувати їх с керівництвом.	Обговорити і скоординувати з ключовим персоналом, час і дату виконання тестування вразливостей мережі, яке буде мати найменший вплив і переривати бізнес-операції. Потрібне офіційне підтвердження часу і дати тестування.	Веб-сервери клієнта можуть бути надзвичайно чутливими в залежності від типу бізнесу. Зазвичай тестування вразливості мережі слід проводити в непікові години (наприклад в ночі і вихідних). В офіційному документі повинен бути запис про затверджений час і дату тестування.
5	Попередити про тестування всіх кого воно буде стосуватися (NOC, системних адміністраторі, і т.д.).	Завчасне повідомити зацікавлені сторони, такі як клієнтська група реагування на інциденти, системні адміністратори, і аналітики з виявлення вторгнень. Листи повинні бути відправлені контактною особою, щоб завчасно повідомити сторони про тестування	Без попереднього повідомлення про заплановані тести на вразливості IT-персонал може не знати про тестування. Електронні листи повинні зберігатися в запису на випадок виникнення проблем з командами безпеки.

Продовження таблиці 4.2

1	2	3	4
ПЕРЕВІРКА ОПЕРАЦІЙ БЕЗПЕКИ			
6	Перевірити існування політики і процедури для контролю та управління логічним доступом до систем.	Переконатися, що політика контролю доступу існує і включає наступне. - визначає, хто відповідає за контроль доступу і надання облікового запису; - визначає, як виконується підготовка облікового запису; - вимагає, щоб користувачам було надано мінімальний рівень доступу, необхідний для їх роботи; - вимагає, щоб облікові записи періодично перевірялися на достовірність.	Без формальної процедури контролю доступу облікові записи користувачів не можуть бути додані, змінені або видалені своєчасно. Застарілі облікові записи користувачів можуть залишатися активними в системі на невизначений термін і можуть використовуватися для доступу до потенційно конфіденційної інформації.
7	Повинні бути передбачені політики та процедури для своєчасного моніторингу, тестування і розгортання виправлень безпеки.	Переконатися, що процедура управління виправленнями існує і включає наступне: - визначає, хто відповідає за управління виправленнями; - визначає, як контролюється доступність виправлень і перевіряються нові виправлення;	Без офіційно задокументованої політики управління виправленнями, критичні виправлення безпеки можуть не застосовуватися своєчасно, що робить системи уразливими для відомих недоліків, які можуть призвести до компрометації системи або втрати інформації.

Продовження таблиці 4.2

1	2	3	4
7		- вимагає, щоб виправлення було перевірено на критичність; - визначає, як координувати і виконувати тестування і розгортання виправлень, зазвичай за допомогою процедури управління змінами; - потрібно, щоб зберігалася документація для управління виправленнями; - вимагає, щоб системи опитувалися на регулярній основі, щоб переконатися, що до них застосовані останні виправлення.	
8	Політики і процедури реєстрації підозрілої активності для аналізу.	Впевнитись, що політика і процедура ведення журналу існують включають наступне: - вимагає, щоб ведення журналу було включено; - визначає, як протоколювання захищено від несанкціонованого доступу; - визначає, як протоколювання зберігається і архівується.	Файли журналів допомагають постійно відстежувати події, що відбуваються в системах, і часто можуть забезпечити протидію атаці. Журнали також мають вирішальне значення при проведенні розслідувань інцидентів безпеки. Без реєстрації може бути важко виявити потенційних зловмисників.

Продовження таблтці 4.2

1	2	3	4
9	Повинні бути передбачені політики та процедури для моніторингу та виявлення підозрілої активності.	Переконатися, що політика і процедура моніторингу існують і включають наступне: - визначає, хто відповідає за моніторинг; - визначає, як виконується моніторинг; - потрібно, щоб моніторинг виконувався на основі 24x7x365; - потрібна наявність такого механізму, як система виявлення вторгнень (IDS) або автоматизовані сценарії для аналізу журналів і відправки повідомлень про будь-які підозрілі дії.	Без формальних процедур моніторингу серверів важко підтримувати постійний моніторинг IT-середовища. Також більш імовірно, що важливі попередження безпеки пропущені або неправильно витлумачені.
10	Повинні бути передбачені політики та процедури для реагування на інциденти безпеки і належного управління ними.	Переконатися, що процедура обробки інцидентів існує і включає наступні кроки: - визначте інциденти і оцініть ситуацію; - задокументувати інцидент в офіційному звіті про інцидент; - усунути проблему, усунувши причину інциденту;	Без офіційного плану реагування на інциденти в сфері безпеки, який дозволяв би ефективно і дієво реагувати на інциденти в сфері безпеки, збої в роботі підприємства можуть бути тривалими і, можливо, повторюватися.

Продовження таблиці 4.2

1	2	3	4
10		- проаналізувати інцидент, щоб виявити першопричини і вжити відповідних заходів для захисту від подібних інцидентів в майбутньому; - зберегти докази шляхом резервного копіювання скомпрометованої системи і ведення інших записів.	Наявність процедури реагування на інциденти безпеки може допомогти зменшити і обмежити негативний вплив інциденту безпеки і може допомогти організації в судовому переслідуванні осіб, які вчинили атаки. Оскільки кількість інцидентів комп'ютерної безпеки зростає, важливо бути готовим реагувати на них ефективно і дієво в разі обходу превентивних заходів безпеки.
11	Повинні бути передбачені політики та процедури для перерозподілу або виведення з експлуатації серверів.	Переконатися, що політика і процедура перерозподілу або виведення з експлуатації існують і включають наступне: - визначає, хто відповідає виведення з експлуатації серверів; - критерії, які визначають, які системи можуть бути перерозподілені; - процес безпечного видалення даних з серверів або утилізації обладнання.	Сервери зазвичай містять конфіденційні дані, і з ними слід поводитися обережно. Якщо інформація не буде надійно видалена, неавторизоване особа може отримати доступ до даних.

Продовження таблиці 4.2

1	2	3	4
ТЕСТУВАННЯ ВРАЗЛИВОСТЕЙ			
12	Інформація про сервер, така як ім’я і версія додатка, повинна відповідати інформації, наданої на кроці №2.	Netcat. За допомогою даної утиліти можна робити наступне: - сканувати порти; nc -vnz «ip-адреса» 20-24 - перенаправляти порти; - здійснювати збір банерів сервісів; nc –vv «ip-адреса» 80 - слухати порти; - завантажувати і викачувати файли; - створити міні-чат.	Якщо ім’я і версія веб-додатки не відповідають інформації, наданої клієнтом, можливо, сервер невірний, банер був змінений або додаток було змінено без відома клієнта.
13	Переконайтеся, що на цільових веб-серверах прослуховуються тільки необхідні порти.	Nmap. Щоб перевірити наявність відкритих портів на веб-сервері, введіть наступну команду: nmap –sS –P0 –O –v –oN «ім’я файлу» «IP адреса» Внаслідок відбудеться сканування, перевірка часто використовуваних портів, сканування, навіть якщо ping (ICMP) не включений, спроба визначити операційну систему веб-сервера, сканування в докладному режимі і виведення результатів в текстовий файл.	Непотрібні порти ніколи не повинні бути відкриті, тому що це дає зловмиснику додаткові можливості проникнути в систему і отримати доступ до даних.

Продовження таблиці 4.2

1	2	3	4
14	Автоматичне сканування машину на наявність поганих файлів налаштувань, змінених програм і інших потенційних проблем захисту.	Tiger. Утиліта переглядає файлові системи з метою виявлення незвичайних файлів і перевірки шляхів, записаних в файлах, виявлених в результаті попередніх перевірок. Під час роботи програма робить записи в файл про всі виявлені слабкі місця в системі. Зараз програмний пакет переріс в Tiger Analytical Research Assistant і є сучасною інтерпретацією даного інструменту.	Вразливості, виявлені на веб-сервері, можуть привести до витоку інформації, несанкціонованого доступу до конфіденційної інформації або недоступності сервера.
15	Перевірка операційної системи і вразливостей, пов’язаних з налаштуванням.	Nessus. Перед початком перевірки потрібно оновити базу зі скриптами перевірок для цього необхвдно ввести наступне команду: nessus-update-plugins Також перед початком сканування в налаштуваннях плагінів необхідно, включити все крім небезпечних плагінів (які можуть викликати нестабільність роботи сервера). Якщо знаєте, що служба не працює, вимкніть цей плагін.	Вразливості, виявлені на веб-сервері, можуть привести до витоку інформації, несанкціонованого доступу конфіденційної інформації або недоступності сервера.

Продовження таблиці 4.2

1	2	3	4
16	Аналізувати і зіставляти результати сканування.	Використовуючи результати сканерів, проведіть аналіз ризику потенційних вразливостей. Рейтинги ризику повинні враховувати ряд факторів, таких як потенційний вплив на організацію і ймовірність виникнення. Порівняйте результати з інформацією, наданої клієнтом, щоб переконатися, що вони є точними.	Оскільки відомо, що інструменти аудиту безпеки і сканери вразливостей генерують помилкові спрацьовування, тому важливо ретельно аналізувати результати на предмет достовірності і точності.
17	Виконайте ручні і ненав’язливі тести, щоб перевірити результати.	На підставі результатів, отриманих сканерами, деякі прості тести ручної перевірки можуть бути виконані з використанням базових інструментів, таких як веб-браузер, telnet, FTP або netcat. Наприклад, для перевірки результатів, пов’язаних з відкритими портами в системі, ви можете використовувати telnet або netcat для підключення за IP-адресою і портом. Також можна зустрітися з системними адміністраторами для перевірки результатів.	Тому що інструменти аудиту безпеки та сканери вразливості, як відомо іноді генерують помилкові результати стану безпеки, тому іноді важливо перевірити, інформацію отриману за допомогою цих інструментів.

Продовження таблиці 4.2

1	2	3	4
18	Запустіть відповідні експлойти в контрольованому середовищі для перевірки вразливостей.	З письмового дозволу клієнта відповідні операції, такі як переповнення буфера, можуть бути запущені з контрольованого лабораторного середовища на серверах, щоб перевірити, чи дійсно вони вразливі.	Експлойти можуть привести до нестабільної роботи системи.

4.3 Звіт проведення аудиту

Здійснивши перевірку відповідно розробленому чек-листу, можна зробити висновок, що загалом, схоже, що Intra вживає активних кроків у розумінні та управлінні ризиками інформаційної безпеки. В ході аудиту виявлені ризики можна вирішити через довгострокові ініціативи, або негайно, коли це було можливо (наприклад, видалення непотрібних служб). В даний час загальна безпека IT-інфраструктури Intra виглядає відносно стабільною з кількома зонами слабкого контролю.

Грунтуючись на проведеній перевірці, було виявлено кілька можливостей для управління, щоб більш чітко визначити методи забезпечення безпеки мережі і зміцнити безпеку IT-інфраструктури. Керівництву рекомендується розглянути наступні ключові зауваження.

1) На серверах працюють старі версії додатків з відомими уразливими.

2) Сервери мають вразливі конфігурації файлів або неправильні дозволи.

3) На серверах працюють вразливі служби, які в разі експлуатації можуть дозволити зловмиснику виконати довільний код або отримати повний контроль над системою.

4) Повний набір формалізованих політик і процедур інформаційної безпеки не існує.

Під час перевірки був виявлений ряд вразливостей, відповідно до якого були вказані можливі ризики від виявлених вразливостей та рекомендації щодо їх усунення.

1) Вразливість. IT-фахівці Intra надали політику і процедуру для підготовки і керування обліковими записами користувачів. В політиці вказано, що відділ кадрів повинен здійснювати розсилку щотижневих повідомлень про звільнених співробітників всім менеджерам і системним адміністраторам для видалення облікових записів. Цей процес використовується для всіх облікових записів користувачів, включаючи облікові записи привілейованого рівня, такі як доступ адміністратора або root. Крім того, облікові записи періодично перевіряються для перевірки дійсності і належного розподілу прав.

Щоб перевірити цей елемент управління, я вибрав оціночну вибірку з десяти користувачів привілейованого рівня і визначив одного користувача, у якого не було задокументованого доказу на його адміністративний доступ до систем.

Ризик. Колишні та дійсні співробітники можуть завдати проблем якщо не контролювати їх доступ до критичних систем і даних.

Рекомендації. Дотримуватися розробленої політики та процедур по контролю і управлінню доступом до системи. І потрібно періодично перевіряти чи дотримуються ці політики.

Витрати. Вартість усунення цієї вразливості мінімальна. Потрібно періодично перевіряти чи дотримуються політики та процедур по контролю і управлінню доступом до системи.

2) Вразливість. У Intra не існує офіційної процедури управління виправленнями. За результатами сканування Nikto і Nessus я виявив, що на декількох серверах і додатках не встановлені останні версії виправлень.

Ризик. Без офіційно задокументованої політики і процедури управління виправленнями, критичні виправлення безпеки можуть не застосовуватися своєчасно, що робить системи уразливими для відомих недоліків, які можуть призвести до компрометації системи або втрати конфіденційної інформації.

Рекомендації. Розробити політики і процедури управління виправленнями. Моніторингу оновлень рекомендацій з безпеки, включаючи CERT і SecurityFocus, і розгортанні виправлень, якщо це може бути застосовано до IT-інфраструктури підприємства.

Витрати. Вартість усунення цієї вразливості мінімальна. Доручити IT фахівцям розробити політику управління виправленнями.

3) Вразливість. Вразливість «HTTP TRACE / TRACK Methods Enabled» зображена на рис. 4.2.

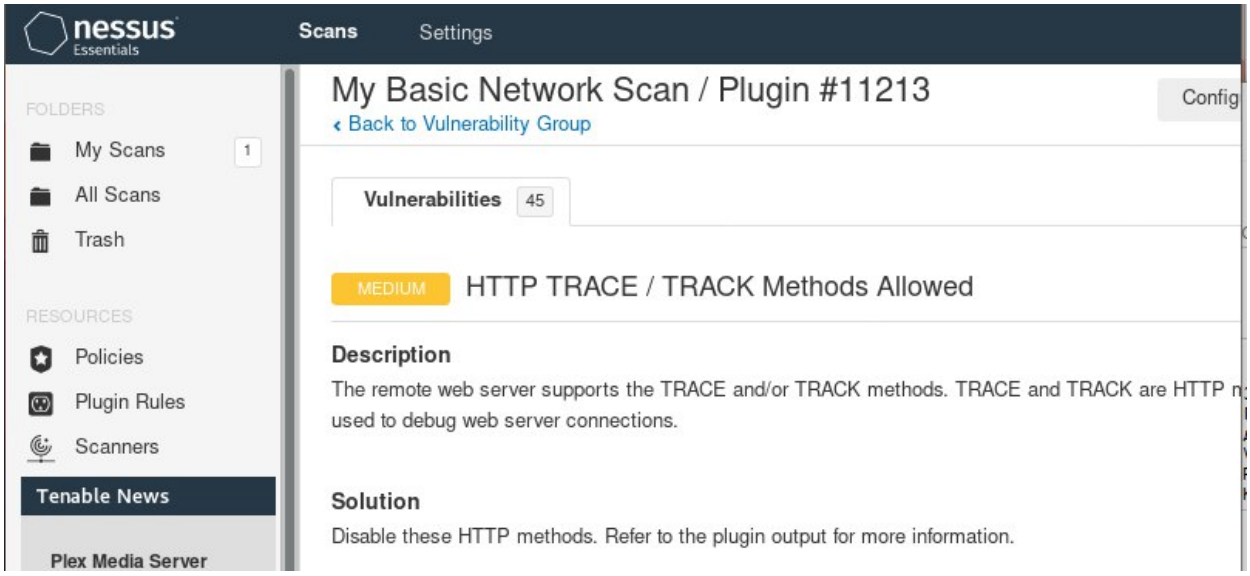

Рисунок 4.2 – Вразливість пов’язана з методами TRACE і TRACK

Ризик. Віддалений веб-сервер підтримує методи TRACE та / або TRACK. TRACE і TRACK – це методи HTTP, які використовуються для налагодження з’єднань з веб-сервером. Вразливості, які можуть виникати при використанні цих методів, пов’язані з Cross-Site Tracing (XST). Цей клас вразливостей зловмисник може використовувати для крадіжки cookie або іншої конфіденційної інформації (наприклад, облікових записів), що зберігаються в заголовку «Authorization» за допомогою міжсайтового скриптинга.

Рекомендації. Вимкнути методи HTTP TRACE / TRACK.

Витрати. Вартість усунення цієї вразливості мінімальна. Оскільки потрібно лише внести зміни в конфігураційні файли.

4) Вразливість. Відкритий 21 порт, на рис. 4.3 наведено результат сканування за допомогою команди «nmap».

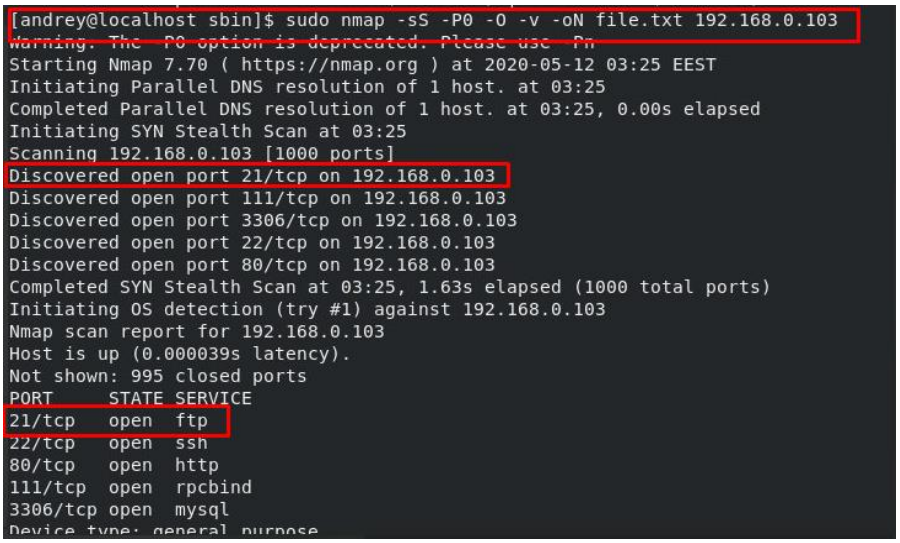

Рисунок 4.3 – Результат використання Nmap

Ризик. Служби я які не використовуються потрібно вимкнути, тому створює додатковий вектор для проведення.

Рекомендації. Потрібно вимкнути порт. В разі потреби рекомендується використовувати більш безпечні рішення для передачі файлів, такі як SCP або SFTP.

Витрати. Вартість усунення цієї вразливості мінімальна. Порт FTP повинен бути заблокований на брандмауері, оскільки він не використовується.

5) Вразливість. На веб-сервері, версія Apache не остання і має відомі вразливості, пов’язані з нею.

Ризик. Версія Apache, що працює на віддаленому хості, становить 2.4.37. Версію можна дізнатися за допомогою команди «httpd -v», результат виконання команди наведений на рис. 4.4.

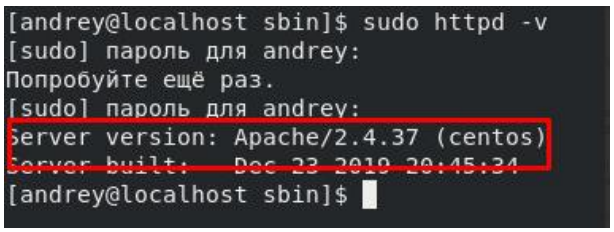

Рисунок 4.4 – Версія Apache

Таким чином, на нього впливають численні уже відомі вразливості. Перша вразливість, це вразливість пов’язана з помилкою в функції mod_auth_digest. Цей баг дає користувачеві, що має обліковий запис у системі, можливість авторизуватися на сервері під чужим ім’ям і обійти встановлені обмеження

доступу. Більш детальну інформацію про вразливість можна знайти за її номером «CVE-2019-0217». Ще одна вразливість яка може існувати в цій версії, це вразливість в функції mod_ssl і зачіпає Apache HTTP Server версій 2.4.37 і 2.4.38. Цей баг пов'язаний з некоректною обробкою сертифіката безпеки TLS 1.3 і допускає несанкціонований доступ через клієнта, що підтримує перевірку автентичності після рукостискання. Більш детальну інформацію про вразливість можна знайти за її номером «CVE-2019-0215».

Потрібно зауважити, що Nessus не перевіряв ці проблеми, а натомість покладався лише на номер версії програми, про яку повідомляється. Також Nessus пропонує методи усунення вразливостей, які зображенні на рис. 4.5.

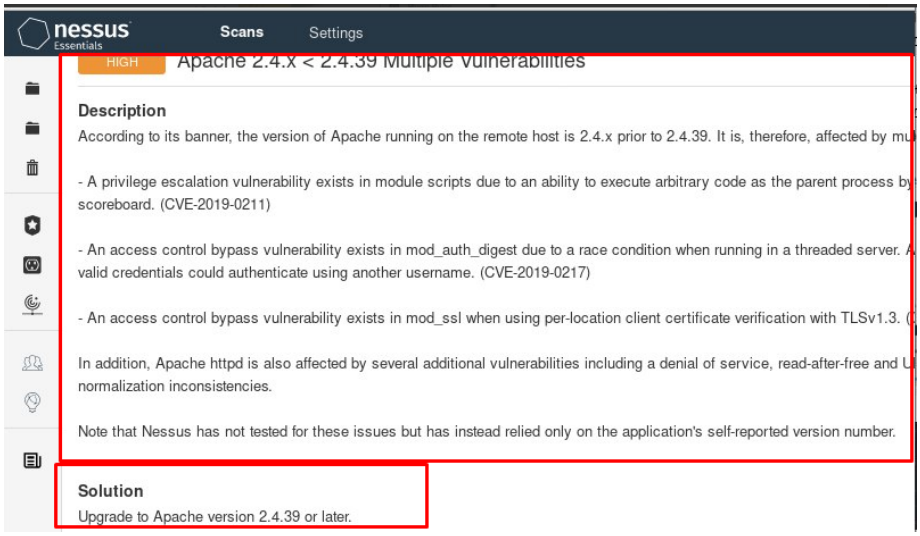

Рисунок 4.5 – Вразливість знайдена за допомого сканера вразливостей Nessus

Рекомендації. Оновіть до останньої версії Apache.

Крім того, рекомендовано підписатися на розсилки з безпеки, такі як Buqtraq, SANS, CERT, які допоможуть визначити більш ранні версії програмного забезпечення, які мають вразливості.

Витрати. Вартість усунення цієї вразливості мінімальна.

6) Вразливість. На всіх трьох серверах файл «~/.bash_history» користувача root є доступним і розкриває останні команди, виконані користувачем «root». Встановлені дозволи для файлу (рис. 4.6). «~/.bash_history», зображені на рис. 4.6.

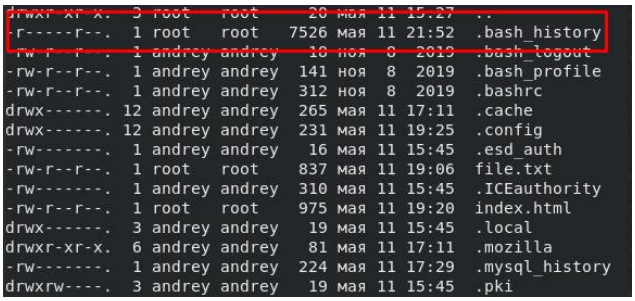

Рисунок 4.6 – Права доступа до файлу .bash_history

Ризик. Це витік інформації, якою зловмисник може потенційно скористатися, щоб запустити більш небезпечні та складніші атаки.

Рекомендації. Налаштуйте сервер так, щоб «~/.bash_history» був недоступним. Щоб запобігти подібним проблемам у майбутньому, слід ретельно перевірити веб-сервери перед розгортанням та періодично після розгортання.

Витрати. Вартість усунення цієї вразливості мінімальна. Веб-сервер повинен бути налаштований так, щоб не дозволяти читати, писати або виконувати цей файл нікому окрім користувача «root».

7) Вразливість. Відсутній файл «/etc/cron.allow». Вразливість була виявлена за допомогою утиліти Tiger. Результат сканування наведений на рис. 4.7 Вона виконує аудит безпеки, автоматично скануючи машину на наявність поганих файлів налаштувань, змінених програм і інших потенційних проблем захисту[13].

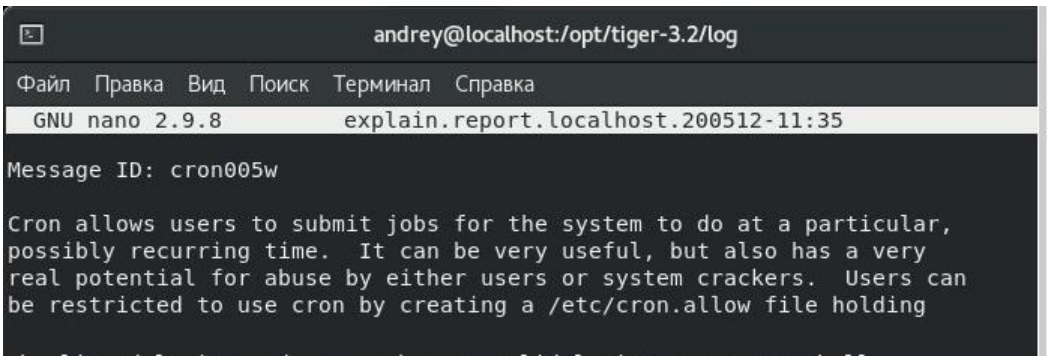

Рисунок 4.7 Результат сканування за допомогою програми Tiger

Ризик: Cron дозволяє користувачам відправляти завдання для системи в заданий час. Це може бути дуже корисно, але також може призвести до зловживань з боку користувачів або зломщиків системи.

Рекомендації: Користувачі можуть бути обмежені у використанні cron, для цього необхідно створити файл «/etc/cron.allow», який повинен містити тільки системних адміністраторів.

Витрати: Вартість усунення цієї вразливості мінімальна. Треба доручити системому адміністратору створити файл «/etc/cron.allow», і проконтролювати кому буде дозволено управляти плануванням.

8) Вразливість. Встановленні не відповідні дозволи для «/var/log/wtmp»(записується інформація про вхід і тривалість роботи всіх користувачів), «/var/run/utmp»(інформація про активних користувачів) та «/var/log/massenge»(глобальний журнал в якому є інформація про всі служби). На рис. 4.8 зображено знайдену вразливість за допомогою сканування програмою Tiger.

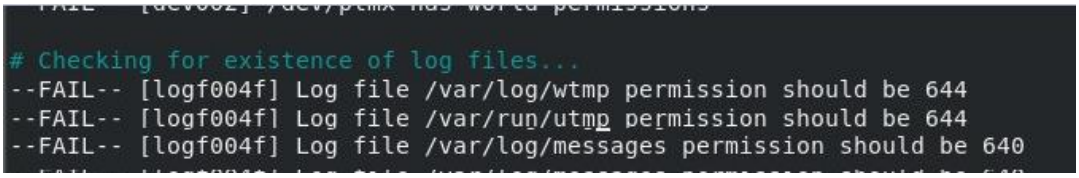

Рисунок 4.8 – Результат сканування за допомогою програми Tiger

Ризик. Якщо встановлені не відповідні права доступу на log-файли, то це може допомогти зловмиснику зібрати необхідну інформацію для проведення більш складної атаки, або допоможе видалити інформацію про неправомірні дії.

Рекомендації Встановити відповідні дозволи, щоб до файлів мав лише системний адміністратор.

Витрати: Вартість усунення цієї вразливості мінімальна. Треба доручити системному адміністратору змінити права доступу до цих файлів.

Результати проведеного аудиту стосувалися питань пов’язаних з конфігурацією критичних елементів ІС (в нашому випадку, це сервери) та процедурних документів пов’язаних із забезпеченням інформаційної безпеки системами. Виявлені вразливості можна виправити за мінімальної кількості зусиль та витрат, використовуючи наявний персонал та ресурси. Оскільки використовувались засоби здійснення перевірок та сканування системи, більшість з яких безкоштовні та відкриті, то системним адміністраторам можна періодично самостійно здійснювати перевірки і сканування системі за допомогою цих засобів.

Важливо також зазначити, що навіть якщо всі виявлені вразливості були усунені, це не означає, що системи захищені. Оскільки системи, що підлягають аудиту – це веб-сервери, доступ до яких може отримати будь-хто в Інтернеті, загроза та ризик завжди постійні, оскільки щодня виявляються нові вразливості.

ВИСНОВКИ

Підводячи підсумки атестаційної роботи, можна впевнено заявити, що аудит, як і професійне обслуговування серверів, може знадобитися майже всім комерційним та державним організаціям. Оскільки в наш час сервери є навіть у невеликих підприємствах, на яких зберігаються бази клієнтів, і лише ця інформація є конфіденційною і потребує захисту. Зараз майже у всіх фірм є власні веб-сайти, які є критичним активом і містять конфіденційну інформацію і саме вони повинні мати надійні засоби захисту. В результаті чого організації повинні встановити адекватні політики безпеки та правильно налаштувати свою систему, так щоб захистити свої цінні та критичні ресурси від випадкових або навмисних атак і забезпечити їх нормальну роботу. Досягти цього допоможе аудит інформаційної безпеки, він не лише допоможе покращити загальний стан захищеності вашої інформаційної системи, а навіть збільшить її ефективність. Саме тому аудит ІБ організації чи окремих її компонентів, останнім часом набуває все більшої актуальності і є обов’язковим заходом по забезпеченню безпеки інформації будь-якої організації. Саме тому зараз можемо спостерігати зростання кількості організацій, які надають послугами з проведення аудиту інформаційної безпеки.

Аудит інформаційної безпеки підприємства повинен проводитися експертами, котрі мають кваліфікацію підтверджену міжнародними сертифікатами, і мають великий досвід проведення аудиту і робіт в сфері інформаційної безпеки, як в організаційних, так і в практичних областях. Через це вартість проведення аудиту є немаленькою. Великі комерційні чи державні організації, для яких втрата цінної чи конфіденційної інформації, призведе до великих втрат, як фінансів так репутації. Наприклад, на серверах банку знаходиться конфіденційна інформація (паспортні дані, ідентифікаційні номера, номера телефонів та інша) про всіх клієнтів банку, і втрата цих даних призведе до великих фінансових втрат. Тому великі організації постійно здійснюють аудит ІБ, як за допомогою зовнішніх організацій так і самостійно, оскільки часто такі організації мають власні відділи ІБ. Як було вище сказано у невеликих організаціях також є цінна і конфіденційна інформація, яка потребує захисту. Але через велику вартість вони не можуть собі дозволити власний відділ ІБ, або постійно користуватися послугами зовнішніх аудиторів.

В таких випадках потрібно здійснювати перевірку безпеки самостійно. Оскільки власні відділи ІБ, є далеко не в усіх організаціях, але майже в усіх, навіть у не великих організаціях, є системні адміністратори, які відповідають за правильне функціювання всієї інформаційної системи. Але нажаль зараз важко знайти гідну інформацію про проведення аудиту інформаційної безпеки. Тому метою даної атестаційної роботи, було вирішення проблеми пов’язаної з відсутністю детальної інформації про аудит ІБ. Таки чином у роботі був проаналізований, розроблений і розглянутий процесний підхід проведення аудиту інформаційної безпеки Linux-подібних операційних систем.

В результаті виконання атестаційної роботи були розроблені рекомендації з проведення аудиту Linux-подібних операційної системи, із зазначенням як саме виконати перевірку, на прикладі аудиту веб-сервера. Описаний підхід допомагає вирішити питання пов’язанні з неправильним налаштуванням операційної системи та її сервісів. Використовуючи матеріал котрий наведений в роботі, системні адміністратори зможуть здійснювати проміжний аудит інформаційної безпеки власних організації і цим самим збільшити рівень захищеності системи. Запропонована структура проведення аудиту ІБ повинна допомогти організаціям зрозуміти, що саме потрібно для захисту сервера і визначити слабкі сторони з точки зору захисту. Також потрібно зазначити, що проведення такого аудиту не дає стовідсоткової гарантії того, що ваша система буде надійно захищена. Оскільки здійснюється аудит лише одного елемента ІС, крім того кожного дня з’являються нові вразливості. Але розроблений процес не вимагає великих фінансових затрат, і якщо здійснювати аудит регулярно, то це значно збільшить рівень захищеності серверу.

Наукова новизна роботи полягає у визначенні рекомендацій та виборі методів реалізації процесу аудиту Linux-подібних операційних систем, шляхом розробки рекомендацій щодо проведення аудиту.

Практична значущість виконаної атестаційної роботи полягає у підвищенні ефективності проведення аудиту інформаційної безпеки Linux-подібних операційних систем самостійно.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аудит інформаційної безпеки – основа ефективного захисту підприємства [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://www.intuit.ru/studies/courses/600/456/lecture/10226.

2. Стрілець А.М. Аналіз програмних засобів захисту Linux-подібних серверних операційних систем / А.М. Стрілець, І.С. Добринін // Харків, ХНУРЕ, Матеріали XXII міжнародного молодіжного форуму «Радіоелектроніка і молодь в XXI столітті». Том 3 –2018. – С.182-183.

3. Стрілець А.М. Дослідження характеристик системи біометричної ідентифікації по райдужній оболонці ока / В.Г. Чернікова, А. М. Стрілець // Харків, ХНУРЕ, Матеріали для всеукраїнської науково-практичної конференції здобувачів вищої освіти та молодих учених «Перспективи розвитку інфокомунікацій та інформаційно-вимірювальних технологій». – 2019. – С. 150-157.

4. Стрілець А. М. Аудит інформаційної безпеки телекомунікаційних мереж з хмарною технологією / С.О Скирда, А. М. Стрілець, В. Г. Чернікова // Харків, ХНУРЕ, Матеріали XXIV міжнародного молодіжного форуму «Радіоелектроніка і молодь в XXI столітті». Том 4 – 2020. – С. 227-228.

5. Стандартні права (SUID, SGID, Sticky bit) в Unix/Linux [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://linux-notes.org/standartny-e-prava-unix-suid-sgid-sticky-bity/.

6. Аудит безпеки сервера [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://ispserver.ru/help/server-security-audit.

7. Кравчук Д. І. Аудит безпеки корпоративних інформаційних систем / Д. І. Кравчук, В. І. Коркушко. – Київ: Еко-трендз, 2015. – 700 с.

8. Немеет Е Unix і Linux. Керівництво системного адміністратора. Для професіоналів / Е. Немеет, Т. Хейн – Київ: МК-Пресс, 2007. – 925 с.

9. Nessus [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://searchnetworking.techtarget.com/definition/Nessus.

10. Перевірка сайту на наявність шкідливого коду за допомогою Linux Malware Detect [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://freehost.com.ua/faq/articles/proverka-sajta-na-nalichie-vredonosnogo-koda-s-pomoschju-linux-malware-detect-i-clamav/.

11. Інструкції з використання John the Ripper [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://hackware.ru/?p=411.

12. Аудит безпеки засобами Tiger [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: http://securos.org.ua/audit-bezopasnosti-sredstvami-tiger/.

13. Інструкція з використання сканера веб-серверів Nikto [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://hackware.ru/?p=3443.

14. Корисні трюки при роботі з netcat [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://habr.com/ru/post/336596/.

15. Стрілець А. М. Дослідження методів захисту біометричного шаблону райдужної оболонки ока / В. Г. Чернікова, С. О. Скирда, А. М. Стрілець // Харків, ХНУРЕ, Матеріали XXIV міжнародного молодіжного форуму «Радіоелектроніка і молодь в XXI столітті». Том 4 –2020. – С. 205-206.

16. Стрілець А. М. Методи тестування засобів захисту інформації / А. М. Стрілець, С.О Скирда, В. Г. Чернікова // Харків, ХНУРЕ, Матеріали XXIV міжнародного молодіжного форуму «Радіоелектроніка і молодь в XXI столітті». Том 4 –2020. – С. 217-218.

17. Ситнов А. А. Особенности аудита информационной безопасности бизнес-систем / А. А Ситнов, С. В. Попов – Київ: Аудитор,2015. – 201 с.

20. CIS Benchmarks: кращі практики, гайдлайни і рекомендації з інформаційної безпеки [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://habr.com/ru/post/338532/.

21. ISACA Information Systems Audit and Control Association [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: http://tadviser.ru/8330220/ISACA.