Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНКО
«____» червня 2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 Кібербезпека
(код і назва спеціальності)
освітній ступень бакалавр
освітня програма Кібербезпека
(назва освітньо-професійної програми)
на тему: Спеціалізовані дистрибутиви UNIX-подібних операційних систем для використання у галузі інформаційної безпеки

Виконавець: студент IV курсу, групи КБ-42

_____ **Роман КАРАКОША**
(підпис) (Ім'я, ПРІЗВИЩЕ)

	Ім'я, прізвище	**Підпис**
Керівник	Іван ПАРХОМЕНКО	

Нормоконтроль	Лариса МИРУТЕНКО	

Київ 2023

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

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

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студенту **КБ-42** (група) **Каракоші Роману Романовичу** (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Спеціалізовані дистрибутиви UNIX-подібних операційних систем для використання у галузі інформаційної безпеки

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №3 від 20.10.2022 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

UNIX-подібні операційні системи, спеціалізовані утиліти

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Ознайомитись з архітектурою UNIX-подібних систем, провести аналіз варіантів застосування таких ОС для фахівців з ІБ, проаналізувати рішення безпеки у таких ОС, розробити власний дистрибутив ОС спеціаліста з ІБ та його модель безпеки.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Зкомпонований дистрибутив на базі UNIX-подібних ОС

з використанням спеціалізованих утиліт та з впровадженою моделлю безпеки.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 24 жовтня 2022 року

Завдання видав _____ (підпис) Іван ПАРХОМЕНКО (Ім'я, ПРІЗВИЩЕ)

Завдання прийняла до виконання _____ (підпис) Роман КАРАКОША (Ім'я, ПРІЗВИЩЕ)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	24.10.2022 – 01.11.2022	*виконано*
2	Аналіз відкритих джерел	01.11.2022 – 15.11.2022	*виконано*
3	Обґрунтування вибору рішення	15.11.2022 – 29.11.2022	*виконано*
4	UNIX-подібні операційні системи. Пропрієтарне та вільне ПЗ	29.11.2022 – 13.12.2022	*виконано*
5	Спеціалізовані UNIX-подібні ОС. Minix, OpenBSD, Linux	13.12.2022 – 13.02.2023	*виконано*
6	Розробка плану зі створення дистрибутиву Linux для ІБ.	13.02.2023 – 27.02.2023	*виконано*
7	Розробка вимог та моделі безпеки	27.02.2023 – 03.04.2023	*виконано*
8	Виділення необхідних компонентів та характеристик дистрибутиву	03.04.2023 – 17.04.2023	*виконано*
9	Розробка дистрибутиву. Використання системи на практиці та документація	17.04.2023 – 01.05.2023	*виконано*
10	Аналіз роботи. Аспекти використання системи	01.05.2023 – 15.05.2023	*виконано*
11	Оформлення пояснювальної записки	15.05.2023 – 29.05.2023	*виконано*
12	Підготовка до захисту кваліфікаційної роботи	29.05.2023 – 12.06.2023	*виконано*

Завдання видав _____ (підпис) Іван ПАРХОМЕНКО (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Роман КАРАКОША (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 12 червня 2023 року

РЕФЕРАТ

Кваліфікаційна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, одного додатку, має 106 сторінки основного тексту та 34 рисунка. Список використаних джерел містить 35 найменувань і займає 4 сторінки.

Метою роботи є розробка спеціалізованого дистрибутиву Linux та моделі безпеки для нього.

Об'єктом дослідження є процес організації захисту UNIX-подібних операційних систем.

Предметом дослідження є механізми та методи забезпечення безпеки UNIX-подібних систем.

Методи дослідження кваліфікаційної роботи:

- аналіз;
- порівняння;
- моделювання;
- опис.

Практичною цінністю є розроблений дистрибутив Linux для використання в галузі інформаційної безпеки.

Ключові слова*:* операційна система, UNIX, UNIX-подібна операційна система, архітектура ОС, спеціалізовані дистрибутиви, Linux, інформаційна безпека, кібербезпека, ОС загального призначення.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ОС	-	Операційна система
ПЗ	-	Програмне забезпечення
API	-	Прикладний програмний інтерфейс
POSIX	-	Portable Operating System Interface
IP	-	Internet Protocol
FOSS	-	Free Open Source Software
FLOSS	-	Free / Libre Open Source Software
I/O	-	Input / Output
FUSE	-	Filesystem in Userspace
CVE	-	Common Vulnerabilities and Exposures
ME	-	Management Engine
UEFI	-	Unified Extensible Firmware Interface
SaaS	-	Software as a Service
FS	-	File System
ARM	-	Advanced RISC Machine
RISC	-	Reduced Instruction Set Computer
SSD	-	Solid State Disk
GNU	-	GNU’s Not UNIX
CVE	-	Critical Vulnerabilities and Exposures
NTP	-	Network Time Protocol
DoS	-	Denial of Service

ВСТУП

В контексті сучасної системи освіти України, фахівцями зі сфери інформаційних технологій прийнято використовувати закриті (а також часто платні) програмні продукти. Студентам та викладачам часто доводиться використовувати неліцензовані (так звані “піратські”) копії програм, що є порушенням авторських прав та ризиком власної безпеки, а сама пропрієтарність ПЗ та закритість його коду несе велику кількість проблем та незручностей. Для академічного процесу серед таких є: неможливість аналізу вихідного коду програм, неможливість порівнянь вихідного коду та зкомпільованого бінарного коду, відсутність методів зручного дослідження поведінки програм при незадокументованих сценаріях, неможливість модифікувати вихідний код програм для академічних та власних цілей, потреба (в багатьох випадках) отримувати платні копії ліцензованого ПЗ, можливі проблеми із правоохоронними органами при використанні неліцензованих копій ПЗ.

Головною причиною мати за мету замінити пропрієтарне ПЗ на вільне та ПЗ, що розповсюджується за вільними ліцензіями в учбовому процесі є нормативно-правова законодавча база, що гарантує авторське право в Україні, та з якою можна ознайомитись на веб-сайті Державної системи правової охорони інтелектуальної власності, а саме:

- Конституція України [1].
- Цивільний кодекс України [2].
- Закон України “Про авторське право і суміжні права” [3].

Дана робота, що була виконана в рамках учбової програми, затвердженої кафедрою КБЗІ, що включає проведення практичних робіт, першочергово спрямована на дослідження можливостей заміни закритого, пропрієтарного ПЗ на вільне та відкрите для використання у академічних цілях, а в другу чергу на аналіз історії та сучасного ринку UNIX-подібних систем, які використовувались або можуть використовуватися в галузі інформаційної безпеки. Цю задачу можна розбити на наступні завдання:

- Дослідження історії, архітектури та концепцій UNIX та UNIX-подібних операційних систем.
- Дослідження та аналіз сучасних UNIX-подібних операційних систем, що можуть використовуватися в галузі інформаційної безпеки.
- Аналіз питань, сценаріїв та можливостей використання UNIX-подібних операційних систем для фахівців у галузі інформаційної безпеки.
- Розробка дистрибутиву Linux, що можна використати для академічних та професійних цілей в сфері інформаційної безпеки.

Серед проблем, що адресує дана робота, є наступні:

- Повсюдне використання неліцензованого ПЗ у рамках навчального процесу через відсутність регуляції та підтримки студентів на учбових закладів зі сторони держави та від розробників або власників ліцензій пропрієтарного ПЗ.
- Часте використання застарілого ПЗ та методик вирішення проблем кібербезпеки, що розглядаються під час учбового процесу.
- Великий акцент на використання студентами закритих операційних систем (в першу чергу Windows), що націлені більше на звичайного користувача, ніж на спеціалістів.
- Відсутність розуміння студентами сучасних трендів та концепцій програмного забезпечення та операційних систем, таких як: відкрите та вільне ПЗ, гнучкі спеціалізовані ОС, хмарні технології та контейнеризація, методи організації захищених та надійних ОС, модульність ПЗ.
- Незвичність студентів спеціальності “Кібербезпека” користування відкритими операційними системами GNU/Linux, їх інструментарієм та можливостями.

Таким чином в якості практичної частини цієї роботи було поставлено за мету вирішити проблему забезпечення студентів зручною програмною платформою для вивчення практичних аспектів дисципліни “Кібербезпека”. Цей проект націлений на створення сучасного та відкритого, безкоштовного, зручного та компактного

інструменту для фахівців з кібербезпеки, що можна буде використовувати під час навчання, а також в інших професійних та повсякденних цілях.

Тому *метою роботи* є розробка спеціалізованого дистрибутиву Linux та моделі безпеки для нього.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- Проаналізувати архітектуру UNIX-подібних операційних систем.
- Дослідити наявні спеціалізовані UNIX-подібні операційні системи, що можна використати в галузі інформаційної безпеки, та їх складові.
- Розглянути наявне прикладне програмне забезпечення для таких систем.
- Дослідити наявні операційні системи та прикладне ПЗ, що можна використати для власного дистрибутиву спеціаліста з інформаційної безпеки.

Об'єктом дослідження є процес організації захисту UNIX-подібних операційних систем.

Предметом дослідження є механізми та методи забезпечення безпеки UNIX-подібних систем.

Методи дослідження кваліфікаційної роботи:

- аналіз
- порівняння;
- моделювання;
- опис.

Практичною цінністю є розроблений дистрибутив Linux для використання в галузі інформаційної безпеки.

РОЗДІЛ 1
ОПЕРАЦІЙНІ СИСТЕМИ: UNIX ТА UNIX-LIKE. ПРОПРІЄТАРНЕ ТА ВІЛЬНЕ ПЗ

1.1 Операційні системи та їх роль в сучасному світі. Вільне та пропрієтарне ПЗ

Операційна система (ОС) – це програмне забезпечення, що слугує інтерфейсом між користувачем та апаратним забезпеченням комп’ютера, головною задачею якого є управління системними ресурсами, що включають апаратні та програмні.

ОС можна також характеризувати як певний рівень абстракції від апаратного забезпечення, на якому відбувається обробка дій користувача, управління файлами, пам’яттю та процесами.

В сучасному світі будь-який користувач інформаційних технологій користується ОС. Вона є основою функціонування комп'ютерів, серверів, мобільних пристроїв, вбудованих систем та інших пристроїв для різноманітних задач. Кожен користувач програм користується операційною системою – програміст, графічний дизайнер, бухгалтер, спеціаліст з кібербезпеки.

Від устрою, або архітектури, операційної системи залежить те, які програми можуть бути запущені на ній, яким чином оброблюється доступ до системних ресурсів, методи взаємодії користувача з нею, тощо. Таким чином, питання вибору операційної системи є важливим для кожного користувача комп’ютерів, будь то настільний персональний комп’ютер, ноутбук, сервер чи смартфон. ОС, особливо пропрієтарна, за тих чи інших причин диктує користувачеві те, як він буде нею користуватися, що на ній робити та встановлювати.

Пропрієтарне та вільне ПЗ в сфері операційних систем: пропрієтарні ОС зазвичай зроблені для звичайних користувачів, яким в першу чергу важливо користуватися інструментами (тобто програмами), але не розуміти їх принцип роботи. Також цим користувачам зазвичай пріорітетно якомога швидше почати

роботу за комп’ютером, тому як самі операційні системи, так і програми для них часто вже заздалегідь налаштовані, а також в принципі розроблені в обмежених рамках в цілях спрощення та уніфікації. Отже, такі операційні системи зазвичай слугують перешкодами для так званих “power” користувачів, які розуміють, або бажають розуміти різні процеси як в операційних системах, так і в програмному забезпеченні. Для розробників та інших IT спеціалістів та ентузіастів закритий код пропрієтарних ОС зумовлює багато незручностей як через неможливість переглянути (або навіть змінити) повний вихідний код, так і через обмеженість певної екосистеми, в яку інтегрована ця ОС. Неповність або відсутність документації тих чи інших процесів майже гарантує неможливість самостійного (або навіть колективного) їх зрозуміння в повній мірі, а сама концепція закритого коду передбачає відсутність контролю користувачів над своєю системою та кодом, що на ній виконується разом з усіма вихідними потенційними ризиками безпеці та конфіденційності.

З іншого боку стоїть ПЗ з відкритим кодом та вільне ПЗ, які приймають ідеї загальнодоступного, безкоштовного, вільно розповсюджуваного ПЗ, вихідний код якого всі можуть переглядати, аналізувати та видозмінювати для власних потреб. Тісно з цим розвинулася ідеологія модульного ПЗ, що є основою більшості, якщо не всіх, ОС з відкритим кодом, суттю якої є просте правило: для конкретної задачі є конкретна програма, яка повинна якісно виконувати свою роботу. Таким чином в світі ПЗ з відкритим кодом та вільного ПЗ має місце бути широке розмаїття різноманітних програм для будь яких потреб, які досить просто підтримувати, та до яких вносять зміни або пропонують покращення мільйони людей [4], [5].

Для сфери відкритого ПЗ, а також для ОС з відкритим кодом, великий вклад зробив Лінус Торвальдс – засновник Git, архітектор та започатківець ядра Linux. Помилково говорять, що він є розробником найпопулярнішої ОС в світі, оскільки Linux, хоча це тільки ядро ОС, стоїть на більшості смартфонів, серверів та суперкомп’ютерів світу. А якщо взяти до уваги, що дистрибутиви Linux, а також iOS, iPad OS та Mac OS X всі є UNIX-like ОС, то можна сказати, що такі операційні системи домінують на всіх ринках комп’ютерних систем.

1.2 UNIX та UNIX-like. Філософія UNIX

Кен Томпсон, один із розробників передуючої їй Multics, що була розроблена для мейнфрейму GE, разом із Деннісом Рітчі – творцем мови програмування C (Сі), повністю переписали Multics для міні-комп'ютера PDP-7. Це започаткувало розробку нової ОС з власною файловою структурою, результатом чого стала нова ОС UNICS (пізніше перейменована в UNIX), яка також містила інтерпретатор команд та утиліти для PDP-7. Потім Рітчі та Томпсон переписали ядро UNIX на мові C, що зробило його код легко підтримуваним та портативним, що було важливою особливістю цієї ОС серед інших того часу, які були написані на асемблері.

З часом багато інших людей зробили свій вклад в розробку та розвиток UNIX та їй подібних систем, які створили сімейство UNIX-подібних ОС. Таким чином, термін "UNIX-like" посилається на системи, які унаслідували від UNIX та її похідних наступні характеристики: переносимість (портативність), мультизадачність, підтримка багатьох користувачів, розділення часу (ОС з розділенням часу, time-sharing OS); ядро, що є шлюзом/інтерфейсом із апаратним забезпеченням, сервіси якого доступні лише через спеціальні функції, які називаються системними викликами; мінімалістичний та модулярний підхід до розробки ПЗ. Сама UNIX філософія може бути сформована наступним чином: "Пишіть програми, які виконують одну задачу, та роблять це якісно. Пишіть програми, щоб вони працювали одна з одною. Пишіть програми, щоб вони вміли працювати з потоками тексту, тому що це універсальний інтерфейс". В таких ОС існують загальні бібліотеки функцій, якими разом із системними викликами користуються прикладні програми. Також там є інтерпретатор команд, що називають shell, який дозволяє викликати інші програми.

Серед інших важливих концепцій, що пішли або були унаслідовані від UNIX є наступні:

- Everything is a file (все є файлом): для обробки вводу/виводу до та з різних ресурсів, таких як документи, директорії, пристрої (диски, принтери, монітори тощо), термінали та процеси, система використовує прості потоки байтів, що надає можливість абстрагуватися від цих ресурсів та використовувати єдиний інтерфейс

(API - інтерфейс прикладного програмування) для їх обробки. Таким чином використовується єдиний набір базових команд для читання та запису з та на диски, периферійні чи мережеві пристрої тощо.

- Дозволи файлової системи: усі файли мають набір правил (флагів, або просто “дозволів” – від англ. permission) для управління доступом до них.
- Групи користувачів: розділ користувачів на групи дозволяє реалізувати управління та розмежування їх доступу до файлів чи команд.
- Кореневий доступ, кореневий користувач (root user, root access): користувач з кореневим доступом має повний доступ до системи [6], [7], [8], [9].

1.3 Спадщина UNIX: UNIX-подібні операційні системи. Спеціалізовані ОС

1.3.1 Minix та мікроядерна архітектура. Різниця між мікроядром та монолітним ядром. Механізми безпеки. Простір користувача та ядра. Minix 3 в сучасності

Minix – це ОС з мікроядерною архітектурою, початковою ціллю розробки якої було надання студентам прикладу для вивчення теорії операційних систем та будови комп’ютера.

Тут потрібно зробити відступ про архітектури ядра: монолітне та мікроядерне.

Як відомо, ядро є основою операційної системи. В більшості систем пам’ять розділяється на дві частини: user space (простір користувача) та kernel space (простір ядра). В користувацькому просторі зазвичай виконуються прикладні програми та виконується взаємодія з користувачем. В просторі ядра виконується привілейоване ядро системи та його розширення, а також, часто, драйвери. Ядро слугує інтерфейсом між користувачем та апаратним забезпеченням. В користувацькому просторі програми мають обмежений доступ до пам’яті, зазвичай тільки до тої, яка виділена для цих програм. Доступ до функціоналу ядра реалізований за допомогою інтерфейсу ядра, що називається системними викликами. Це є основою для захисту пам’яті та розмежування доступу. У ядра ж є повний доступ до пам’яті, як і до всього апаратного

забезпечення, що відповідає кільцю 0 (режим ядра) моделі кілець захисту (в архітектурі x86 їх передбачено 4). Код простору користувача виконується на кільці 3, тобто зовнішньому і, відповідно, найменш привілейованому. Інші два кільця передбачені для драйверів пристроїв. Кільце 1 іноді використовується гіпервізорами, а в деяких імплементаціях цієї моделі кільце 2 використовують його для користувацьких програм, що мають доступ до вводу/виводу [10]. На рис. 1.1 зображені кільця безпеки процессорів архітектури x86.

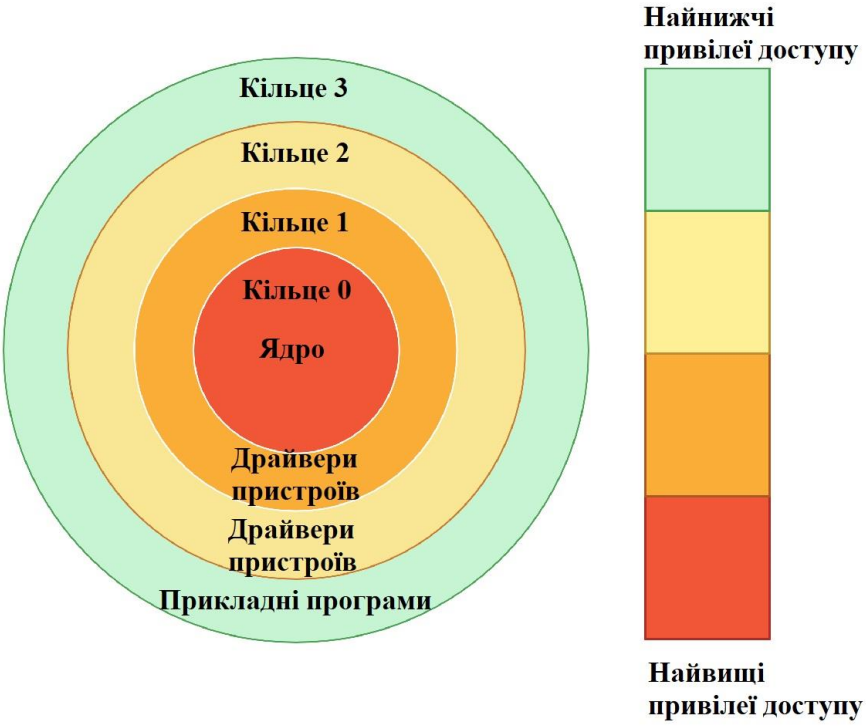


Рисунок 1.1 – Кільця безпеки процессорів x86

У монолітному ядрі, прикладом якого є Linux, є розділення лише на Kernel та User простори. В першому виконується код самого ядра, більшості його розширень та драйверів. Деякі розширення ядра працюють у просторі користувача, найбільшими прикладами чого є userspace I/O system (система вводу/виводу простору користувача), а також filesystem in userspace (FUSE – файлова система простору користувача). Важливо пам’ятати, що ядро – це єдиний процес, що працює в єдиному адресному просторі. Іншими словами, усі сервіси ядра існують в цьому адресному просторі, до якого вони всі мають повний доступ. Головною перевагою монолітного ядра є швидкий час виконання програм. Серед недоліків важливо вказати більший об’єм,

більша складність розширення ядра, а також те, що при збоях будь-яких сервісів ядра система виходить із ладу [11], [12], [13], [14], [15], [16].

У 2018 році на конференції Asia-Pacific Systems Conference (Азійсько-Тихоокеанська системна конференція) було надано документ, в якому стверджувалося, що 40% критичних вразливостей зі списку CVE (Common Vulnerabilities and Exposures) для Linux вдалося повністю запобігти при використанні формально перевіреного мікроядра (при чому 29% критичних вразливостей можна запобігти просто при використанні мікроядра), а лише 4% в повному обсязі залишилися у такій системі [17].

Мікроядро займається лише базовою міжпроцесовою комунікацією, управлінням віртуальною пам’яттю, адресним простором та потоками. Сервери, такі як драйвери девайсів, файловий та стек протоколів (мережевий стек), а також міжпроцесова комунікація прикладних програм виконуються у просторі користувача. Сервери у мікроядрерній архітектурі ОС є, по суті, демонами (сервісами), яким ядро надає ті чи інші права. Перевагою такої архітектури є більша краща безпека та надійність, оскільки всі сервери є розділеними та мають власні адресні простори, а отже, наприклад, драйвери не можуть викликати збій системи, або отримати доступ до ядра. Також об’єм мікроядра є меншим. Додавання функціоналу до системи на основі мікроядра не потребує перекомпіляції. Мікроядерна архітектура наслідує принцип мінімальних привілеїв, суть якої в тому, щоб код отримував лише мінімальний набір привілеїв, необхідний для виконання програми.

Отже, за архітектурою ядро Minix є мінімальним, а файловий сервер, сервер процесів, а також кожен драйвер пристроїв виконуються як процеси в режимі користувача. Таким чином ця ОС добре ізолює процеси один від одного, обмежує доступ процесів до ядра та його функцій, а також пам’яті, а отже зменшує ризики та збитки при компрометації або виходу із ладу процесів та драйверів.

Ядро оброблює переривання, примітивні процеси та міжпроцесову комунікацію, а також планує виконання задач. Також ядро Minix підтримує API (прикладний програмний інтерфейс), що складається з приблизно 30 викликів, якими можуть користуватися авторизовані сервери та драйвери. Програми рівня

користувача не можуть їх використовувати, проте можуть утилізувати системні виклики POSIX, що відправляють повідомлення до серверів.

В серверах закладений основний функціонал операційної системи. Найважливішим сервером є так званий “сервер реінкарнації”, який відповідає за контроль працездатності інших серверів, драйверів та програм шляхом періодичних опитувань. В разі їх несправності, цей сервер автоматично перезавантажує процеси.

До сфери безпеки можна віднести Minix 3 як систему, що була розроблена а) з ціллю бути відмовостійкою завдяки визначенню та виправленню помилок під час роботи системи без потреби втручання користувача та б) для використання у вбудованих системах, або системах з малим об’ємом ресурсів [18], [19].

Якщо бажання використання Minix 3 як відмовостійкої системи як засіб забезпечення інформаційної безпеки є цілком зрозумілим, використання цієї ОС у вбудованих системах потрібно окремо виділити: з 2015 року Intel Management Engine (Intel ME) як основу програмної бази використовує Minix 3.

Intel ME – автономна підсистема, що була вбудована (або може бути вбудована) в усі чіпсети Intel починаючи з 2008 року. Вона весь час активна, принаймні доти, поки материнська плата отримує живлення, що включає батарею CMOS. Management Engine має повний доступ до усіх системних ресурсів, включаючи пам’ять, мережевий адаптер, периферійні пристрої. За офіційним описом Intel, “ME включає, але не обмежується наступним функціоналом: послуги позаполосного керування (OOB) з низьким енергоспоживанням; службу ліцензування можливостей (CLS); захист від крадіжки; захищений аудіо-відеошлях (PAVP)”.

Серед модулей Intel ME, що використовуються для безпеки, можна виділити наступні: Intel Boot Guard (IBG) та Secure Boot. Перший використовується для уникнення виконання комп’ютером “фірменних” (firmware), UEFI образів, що не були випущені та підписані офіційними виробниками. Secure Boot – це протокол, що слугує для запобігання завантаження UEFI драйверів або завантажувачів ОС (OS boot loader), що не були підписані дозволеними цифровими підписами [20], [21], [22], [23].

Таким чином модифікована пропрієтарна версія Minix 3 є однією з найпоширеніших ОС, що встановлена умовно на всіх процесорах Intel, що були випущені після 2015 року.

1.3.2 OpenBSD – перший серед ОС загального призначення у сфері безпеки

OpenBSD – це відкрита, зфокусована на безпеці UNIX-подібна ОС, базована на BSD. За описом з офіційного веб-сайту: “Наші зусилля зосереджені на переносимості, стандартизації, коректності, проактивній безпеці та інтегрованій криптографії”. В цілях проекту визначено наступне: “Ми прагнемо зробити наше програмне забезпечення надійним і безпечним та заохочуємо компанії використовувати такі деталі, які вони хочуть”, а також “OpenBSD вірить у надійну безпеку. Ми прагнемо бути НОМЕРОМ ОДИН в індустрії безпеки (якщо ми ще не там). Наша відкрита модель розробки програмного забезпечення дозволяє нам мати більш безкомпромісний погляд на підвищення безпеки, ніж більшість інших постачальників”.

Розробники OpenBSD перші в індустрії прийняли концепт повної відкритості проблем безпеки. Важливою рисою цієї ОС є те, що вона поставляється у режимі “Secure by Default” – безпечна за замовчуванням. Це вмотивовано ідеєю того, що неможливо очікувати від користувачів OpenBSD миттєво стати експертами безпеки, тому одразу після встановлення у системі вимкнені усі необов’язкові сервіси. Таким чином, під час вивчення цієї ОС, користувач власноруч навчиться вмикати та налаштовувати різні системні сервіси та ознайомиться з аспектами їх безпеки.

На даний момент OpenBSD вважається однією з найбільш безпечних ОС загального призначення. Її визнають як безпечнішу за інші ОС сімейства BSD, дистрибутиви Linux, Microsoft Windows та Mac OS.

Серед заходів безпеки OpenBSD можна виділити наступні: А) Передналаштоване розділення привілеїв: вбудований веб-сервер виконується від імені обмеженого користувача, який має доступ лише до файлів веб серверу та навіть не може отримувати доступ до оболонки shell для виконання команд. Також веб-

сервер виконується в так званій “chroot-jail”, що для процесу веб-серверу означає обмежений доступ лише до передвизначеної root-директорії та її субдиректорій, таким чином створюється sandbox (пісочниця) для процесу, щоб ізолювати його від доступу до зовнішніх файлів; Б) Write XOR Execute – адресний простір процесу або ядра може бути записуваним або виконуваним, але не в одночас. Деякі Linux дистрибутиви лише недавно включили такий функціонал; В) Guard Pages – нечитаємі та незаписуємі сторінки в пам’яті в кінці кожної сторінки “реальної” пам’яті задля виявлення переповнення буферу; Г) Рандомізація адресного простору: код не повинен знаходитись в одному й тому ж місці пам’яті кожного разу, коли програма запускається; Д) Pledge та Unveil: системні виклики, що покликані обмежити доступ до інших системних викликів та файлової системи. В парі вони існують лише в OpenBSD, тобто це унікальна особливість безпеки. Програми зазвичай починають роботу з більшою кількістю привілеїв, ніж їм буде потрібно під час роботи. За допомогою Pledge процес “обіцяє”, що не буде використовувати ніякі системні виклики, окрім зазначених. Якщо процес порушить “обіцянку”, він буде одразу припинений. До того ж, обмеження до системних викликів можна лише посилювати, але не полегшувати. Unveil використовується для визначення директорій та файлів, до яких програма буде мати доступ. Таким чином програма не зможе отримати доступ до файлів, потребу у доступі до яких не було попередньо задекларовано [24].

Цю операційну систему можна використовувати як для настільного ПК та робочої станції (завдяки достатній підтримці потрібного для цього ПЗ та драйверів), так і для більш спеціалізованих задач, таких як будь-які сервери (поштовий, веб тощо), фаєрвол та роутер з фаєрволом та інших.

Отже, OpenBSD є лідером серед ОС загального призначення у сфері безпеки, особливо завдяки заводським налаштуванням та унікальним засобам. Також часто розробники цього проекту просувають нові технології та тренди безпеки, які потім приймаються всією індустрією. Постійні аудити коду та повна відкритість також допомагають якомога швидше знаходити та адресувати проблеми безпеки, що є запорукою та причиною успіху цієї системи в своїй сфері. Завдяки пріорітетам

“безпека перед усім”, “простота та мінімалізм”, а також відмінній документації та проактивній позиції розробників, ця ОС заслужено має свою репутацію.

1.3.3 SmartOS – наслідник Solaris від Sun Microsystems (нині Oracle)

Ця відкрита (FOSS – free and open source) операційна система була розроблена як своє хмарне рішення компанією Joyent, що займалася сервісами та програмним забезпеченням. Ця ОС може бути використана як гіпервізор віртуальних машин, легкий хост контейнерів, або у багатокористувацьких розгортаннях (multi-tenancy) у “Програмному забезпеченні як послузі” (Software as a Service, SaaS). SmartOS надає повну ізоляцію контейнерів, віртуалізацію та розмежування мережі для кожного контейнера (унікальний IP для окремого), безпечну, надійну та масштабовану файлову систему та багато іншого. Цю ОС можна також охарактеризувати як легковажний контейнер ОС та гіпервізор, оптимізований для безпеки.

Історію виникнення SmartOS можна зобразити ланцюжком “(BSD) – (Sun OS + System III) – (System V) – Solaris – OpenSolaris – Illumos – SmartOS”, таким чином вона містить багато технологій та рішень, що були розроблені цими операційними системами. Включає в себе ядро Illumos, технології OpenSolaris, зокрема Zones – гіпервізор типу 2 (системного рівня), віртуалізацію KVM (Kernel-based Virtual Machine) і bhyve (BSD hypervisor), а також підтримує апаратну віртуалізацію [25], [26].

Важливою особливістю SmartOS є технології, зокрема ті, що були перейняті від попередників, та їх тісна інтеграція. Серед них, окрім вище зазначених, є:

- Solaris Fault Management Architecture (FMA) – Архітектура Управління Несправностями Solaris – надає автоматичну діагностику несправного апаратного забезпечення, а також може вживати заходи для виправлення цих несправностей [27].
- Service Management Facility (SMF) - Об’єкт (центр) управління сервісами – автоматично діагностує та виправляє неполадки програмного рівня.

- ZFS – Zettabyte File System, файлова система зетабайтів – Продвинута файлова, яку описують як “Останнє слово в сфері файлових систем”, що відзначається стабільністю, швидкістю та безпекою. Серед особливостей ZFS є:
  - ❖ Pooled storage – об’єднане сховище (інтегроване управління томами) – ZFS комбінує в собі функції файлової системи та управління логічними дисками, що означає, що, на відміну від інших файлових систем, ZFS може поширюватися на пул дисків, а також можна просто додати сховище до пулу під’єднавши диск – ZFS автоматично займається розділенням (partitioning) та форматуванням дисків.
  - ❖ Copy-on-write – механізм копіювання при записі, що використовується для оптимізації роботи з оперативною пам’яттю або файлами на диску.
  - ❖ Snapshots – снапшоти, знімки – за допомогою Copy-on-write, ZFS використовує снапшоти для відслідковування змін у системі. Знімок містить в собі оригінальну файлову систему, а “пряма” (live) файлова система містить всі зміни, що були зроблені після створення знімку.
  - ❖ Верифікація цілісності даних.
  - ❖ Автоматичне відновлення (англ. repair).
  - ❖ Широкі ліміти на розмір файлу та об’єм сховища (16 екзбібайтів [2^{64} байтів] та 256 тріліонів йобібайтів [2^{128} байтів]), безлімітна кількість файлів у системі, 2^{48} у директорії.
- DTrace – це комплексний фреймворк динамічного трасування для пошуку та вирішення проблем (troubleshooting) у ядрі та прикладному ПЗ на системах організацій (виробничих системах, production systems) в реальному часі. Він надає спостережність всього програмного стеку (software stack) в одному інструменті, що надає широкі можливості обстежувати роботу програмного забезпечення. Його використовують для визначення вузьких місць в продуктивності систем, а також відлагодження несподіваної та непередбачуваної поведінки ПЗ [28].

- Crossbow – набір технологій для віртуалізації мережі. Включає в себе екземпляри (інстанції) IP-стеку, віртуальні інтерфейси мережі, або технологія інтерфейсу псевдо-мережі (VNIC – virtual network interface card), управління потоком (трафіком) мереж.

Поверхнево оглядаючи SmartOS, можна виділити наступні причини використання цієї системи:

- Швидкодія
- Спостережність
- Надійність

В сфері кібербезпеки, для технічного захисту інформації серед цих пунктів найважливішим є спостережність. В документі НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу» серед функціональних критеріїв захищено вказано спостереженість і виділено наступне: “Ідентифікація і контроль за діями користувачів, керованість комп'ютерною системою становлять предмет послуг спостереженості і керованості”. Спостережність – це оцінка того, чи дають наявні результати вимірювання стану системи адекватну інформацію про систему, або, іншими словами, оцінка відповідності внутрішнього стану системи за її вихідними даними. Також для кібербезпеки є важливими аспектами системи її довіреність (trustworthiness) та, як її складова, надійність (reliability).

В даній ОС інструментом надання спостережності є DTrace, що дозволяє спостерігати за всім прикладним ПЗ, а надійності – ZFS, що, наприклад, за допомогою контрольних сум перевіряє цілісність даних.

1.3.4 GNU / Linux – найрізноманітніше сімейтво операційних систем

Linux – монолітне, модульне ядро з відкритим кодом, оригінально розроблене Лінусом Торвальдсом. Після вивчення MINIX він хотів зробити відкриту ОС в якості власного хоббі-проекту. В нього також було багато питань до MINIX, тому Торвальдс

переписав деякі її компоненти на свій смак. З часом все з’являлося всі більше бажаючих зробити свій вклад у Linux, тому це переросло в серйозний проект.

Сьогодні Linux є одним із найпопулярніших ядер операційних систем, одночасно будучи першим у сфері операційних систем загального призначення, суперкомп’ютерів та серверів.

ОС GNU є частиною проекту GNU, головними цінностями якого є вільне ПЗ. Річард Сталлман, засновник цього проекту, хотів створити повністю вільну ОС з відповідним ПЗ. Його ціллю було надати користувачам можливість вільно вивчати, ділитися, модифікувати та поширювати свої варіації ПЗ, яким вони користуються. На момент 1992 року, під час розробки цієї ОС було створено більшість основних утиліт, проте ще не було створено ядро. Того самого року ядро Linux було випущено під вільною ліцензією GNU GPLv2, яке в той самий час було розроблено за допомогою інструментарію GNU, в тому числі GCC (GNU Compiler Collection) та бібліотеки GNU libc. Таким чином, з’явилася перша можливість запуску ОС, що повністю складалася з вільного ПЗ. Як наслідок, GNU офіційно прийняла ядро Linux, як складову своєї операційної системи. Хоча в цьому визначенні є свої нюанси, загалом прийнято називати більшість ОС, базованих на ядрі Linux, та які використовують програмну базу GNU “GNU/Linux”, або “GNU + Linux”, часто спрощено Linux, хоча це не вважається повністю коректним.

Нині операційні системи сімейства Linux набули досить широкого розповсюдження, в тому числі та в більшості своїй в ролі системи загального призначення. Існує велика кількість відгалужень (fork-ів) для різних потреб, в тому числі і для сфери кібербезпеки [5].

1.4 Спеціалізовані дистрибутиви Linux для використання у сфері кібербезпеки

1.4.1 Пентестинг, цифрова криміналістика, етичний хакінг, зворотня інженерія, дослідження безпеки

Kali Linux – найпопулярніший та продвинутий дистрибутив.

Це похідний від Debian дистрибутив, створений для задач інформаційної безпеки. Він надає інструментарій (понад 600 інструментів), конфігурації та засоби автоматизації для пентестингу, аудиту та дослідження безпеки, зворотьої інженерії та управління та аналізу вразливостей, збору інформації. В дистрибутиві містяться пропрієтарні рішення.

Серед переваг, крім широкого вибору інструментів, зазначають наступні:

- Багатомовна підтримка.
- Модифіковане ядро, зокрема для ін’єкцій (зокрема для вправ з бездротовими мережами).
- Широка підтримка бездротових пристроїв.
- Заходи безпеки: розробка у довіреному та безпечному середровищеі, підписані пакети та протоколи.
- Відкрита для модифікацій система.
- Підтримка архітектур ARM.

Kali Linux має відмінну репутацію серед дистрибутивів для пентестингу та дослідження інформаційної безпеки та містить найширший набір інструментарію на ринку. Недоліками є відносно високі потреби до системних ресурсів: бажано мати мінімум 2 ГБ оперативної пам’яті та 20 ГБ пам’яті диску (іноді рекомендують до 50 ГБ вільного місця), проте для деяких інструментів рекомендовано від 8 ГБ оперативної пам’яті, зокрема для використання Burp Suite. Для Live-CD в рекомендаціях визначено накопичувачі об’ємом від 8 ГБ. Іноді можна спостерігати сповільнення системи. Мінімальні вимоги з конфігурацією SSH серверу без робочого стола становлять 128 МБ оперативної пам’яті (рекомендовано 512 МБ) та 2 ГБ дискового простору.

Серед основних інструментів, що надає Kali Linux, є:

- Burp Suite – для пентестингу веб-застосунків.
- Wireshark – для аналізу мережевих протоколів.
- Aircrack-ng – для зламу бездротових мереж.

- Hydra – для онлайн брут-форс зламу паролів.
- Maltego – для збору інформації.
- John – офлайн аналог Hydra.
- Metasploit Framework – для експлойтування слабких місць безпеки.
- Owasp-zap – для пошуку вразливостей прикладного ПЗ.
- Nmap – мережевий сканер.
- Sqlmap – для експлойтування вразливостей SQL за допомогою SQL-ін’єкцій.

Parrot OS – пентестинг, інформаційна безпека та анонімність.

Метою розробки ParrotOS було створення ОС для пентестингу, етичного хакінгу, пошуку, оцінок та виправлення вразливостей, комп’ютерної криміналістики та анонімного браузингу (web-browsing, перегляд мережевих ресурсів). Містить в собі особливості Frozenbox (свого попередника, розробники якого тепер працюють над ParrotOS) та Kali Linux. Головною особливістю є функціонал для анонімності, що дозволяє приховати ідентичність користувача при перегляді інтернет ресурсів, таким чином залишаючи його відносно непомітним, що особливо корисно під час кібер-контратак проти спроб зламу.

Серед переваг виділяють:

- Комбінацію різних інструментів, що також присутні у Kali, разом з інструментарієм для анонімності та конфіденційності.
- Зручне налаштовуване середовище робочого столу, зручний інтерфейс та навігація.
- Відносна легковажність порівняно з аналогами.

ParrotOS більш серйозно ставиться до власної безпеки користувача, ніж аналоги, зокрема шляхом обеження активності ОС за допомогою її віртуального, sandboxed (пісочниця) середовища.

Системні вимоги ParrotOS Security Edition становлять 1 ГБ оперативної пам’яті та 20 ГБ дискового простору, а Home Edition та Architect потребують лише 16 ГБ

дискового простору та 1 ГБ RAM. Для Raspberry Pi потрібно 512 МБ RAM та 8 ГБ дискового простору.

Як простіший аналог можна зазначити BlackBox Linux, що більш зфокусований на пентестингу, містить зручне меню та надає короткі описи для усіх інструментів.

ArchStrike – репозиторій для перетворення своєї Arch Linux в професійний інструмент інформаційної безпеки.

Хоча він і розповсюджується також у вигляді дистрибутиву, головна частина проекту лежить у його репозиторії. Він якісно організований та модерований, що дозволяє швидко та зручно обрати пакети для своїх потреб. Застарілі та не оновлювані тривалий час пакети видаляються з репозиторію, тому користувач може бути впевнений, що встановлює актуальні інструменти.

ArchStrike хвалять за зручність та можливість вибору лише тих пакетів, що потрібні користувачам. Також репозиторій містить понад 5000 інструментів для використання та аналізу експлойтів, зловмисного ПЗ, мережевих протоколів, а також соціальної інженерії, спуфінгу, брут-форсу, криміналістики, DDoS атак та пошуку вразливостей.

CAINE – Computer-Aided Investigative Environment – Комп’ютеризоване середовище розслідування.

Оснований на Ubuntu, цей дистрибутив був створений як частина проекту для ПЗ для цифрової криміналістики, яке, як результат, було організоване за допомогою дружнього для користувача графічного інтерфейсу. В CAINE можна знайти інструменти для роботи з та аналізу баз даних, пам’яті, мережі та інші корисні для цифрової криміналістики.

Містить в собі наступні інструменти:

- The Sleuth Kit та Autopsy – бібліотека та набір утиліт для витягування інформації з різних накопичувачів, слугуючи основою для кримінального аналізу комп’ютерних систем. Autopsy фактично є графічним інтерфейсом для цього інструменту.

- RegRipper – для добування та розбору (parsing) інформації з файлів, що зберігаються на пристрої.
- Tinfoleak – аналіз Twitter постів та аккаунтів.
- Wireshark – для аналізу мережевих протоколів.
- PhotoRec – відновлення видалених файлів з дисків, пам'яті цифрової камери та інших запам'ятовуючих пристроїв.
- Fsstat – відображення статистичної інформації для знімків, файлових систем та зберігаючих пристроїв.

Як інструмент криміналістики, цей дистрибутив можливо буде корисним для спеціалістів з кібербезпеки, працюючих на правоохоронні органи. Він розроблений специфічно для отримання та аналізу інформації з власних пристроїв, що корисно при зборі доказів. Також CAINE можна використовувати для зрозуміння процесу отримання інформації з пристроїв, що корисно в академічних цілях, а також для розробки контр-заходів для забезпечення конфіденційності.

Варто зазначити аналог – *DEFT (Digital Evidence & Forensic Toolkit)*, в перекладі "Інструментарій для збору цифрових доказів і криміналістики". Хоча оригінальний сайт не працює, а проект не підтримується, він все ще являється якісним продуктом. Дистрибутив є зручним та дружнім для користувача, відомий за поширені можливості виявлення апаратного забезпечення та перевірки цілісності. Містить одні з найкращих прикладних програм з відкритим кодом для реагування на інциденти та цифрової криміналістики. Може також використовуватися для відновлення пошкоджених дисків.

Network Security Toolkit – Інструмент для перевірки безпеки мережі.

Цей інструментарій був розроблений для надання простого та зручного доступу до найкращих програм з відкритим кодом для сфери безпеки мереж. Містить більшість з топу 125 інструментів безпеки за версією insecure.org. Має продвинутий веб інтерфейс користувача, наданий для зручного системного та мережевого адміністрування, навігації, автоматизації, моніторингу та аналізу мереж, налаштувань та інших функцій. Також може використовуватися для моніторингу та аналізу

безпеки мереж віртуальних серверів та машин у виробництві. Розповсюджується в форматі Live-CD.

Samurai Web Testing Framework – інструмент для перевірки безпеки веб додатків.

Має репутацію найкращого інструмента для веб пентестингу. Містить інтерфейс командного рядка для простої інтеграції з іншими інструментами, такими як Burp Suite. Підтримує різні методи аутентифікації, надає можливості перевірити усі можливі способи отримання доступу до інтернет ресурсів, серед яких GET/POST запити, виклики JavaScript тощо, допомагаючи автоматизувати рутину. Розповсюджується у вигляді віртуальної машини та вихідного коду. Більшість пакетів є ПЗ з відкритим кодом, проте містить і пропрієтарні рішення.

Fedora Security Lab – лабораторія безпеки Fedora.

Fedora Security Lab надає безпечне тестове середовище для роботи над криміналістикою, аудитом безпеки, відновленням системи, а також для вивчення методологій тестування безпеки в університетах та інших організаціях. Містить чистий інтерфейс та налаштовуване меню, що надає усі потрібні інструменти для проходження потрібного тесту або процесу відновлення пошкодженої системи. Розповсюджується в форматі Live-CD образу.

Серед інструментів, що йдуть в пакеті з Fedora Security Lab є наступні:

- EtherApe – графічний монітор мережі.
- Ettercap – пакет для атак типу MiTM – Man-in-the-Middle, атака посередника.
- Medusa – брут-форсер з широким функціоналом.
- Scap-workbench – графічний SCAP (Security Content Automation Protocol – протокол автоматизації управління даними безпеки) сканер.
- Skipfish – активний інструмент аналізу безпеки веб-додатків.
- Sqlninja – інструментарій тестування SQL-ін’єкцій веб-додатків.
- Yersinia – мережевий інструмент для експлуатування вразливостей в мережевих протоколах.

- Інші, такі як Nmap та Wireshark.

1.4.2 Особиста безпека, конфіденційність та анонімність

Qubes OS – відкрита, орієнтована на безпеку система для робочого столу з використанням віртуалізації.

Вважається одним з найбезпечніших Linux дистрибутивів завдяки можливості виконання програм в ізольованому одна від одної середовищі. Окрім безпечної системи робочого столу, функціонал розмежування Qubes OS корисний також для управління мережевим стеком, фаєрволом (в тому числі для серверу) та для інших потреб користувача. Також ця ОС містить якісну документацію.

В Qubes OS використовується технологія Split GPG (GNU Privacy Guard – програма для шифрування інформації та створення електронних підписів), що реалізує концепт схожий на смарт карту з вашими власними GPG ключами, але роль смарт карти виконує окремий “куб” – віртуальна машина. Таким чином недовірений домен, наприклад Thunderbird (клієнт пошти, новин, RSS, чатів, а також менеджер персональної інформації), може делегувати всі криптографічні операції (такі як шифрування, дешифрування, підписи) іншому, більш довіреному, ізольованому від мережі домену. Таким чином компрометація Thunderbird не призведе до викрадення всіх ваших ключів.

На рис. 1.2 та рис. 1.3 надані схеми роботи Qubes OS, де можна побачити переваги з точки зору безпеки, які надає ця ОС.

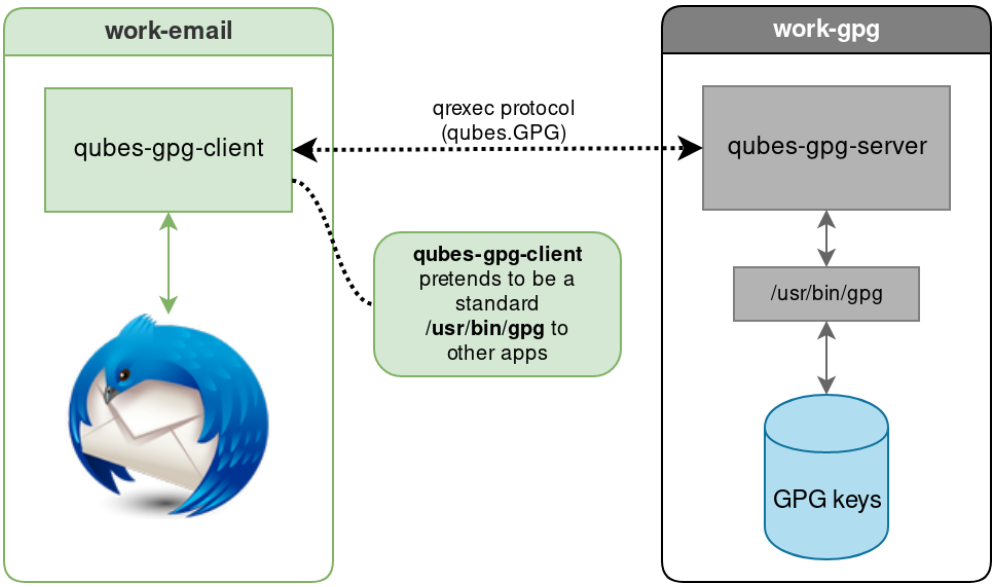


Рисунок 1.2 – Схема розділених GPG

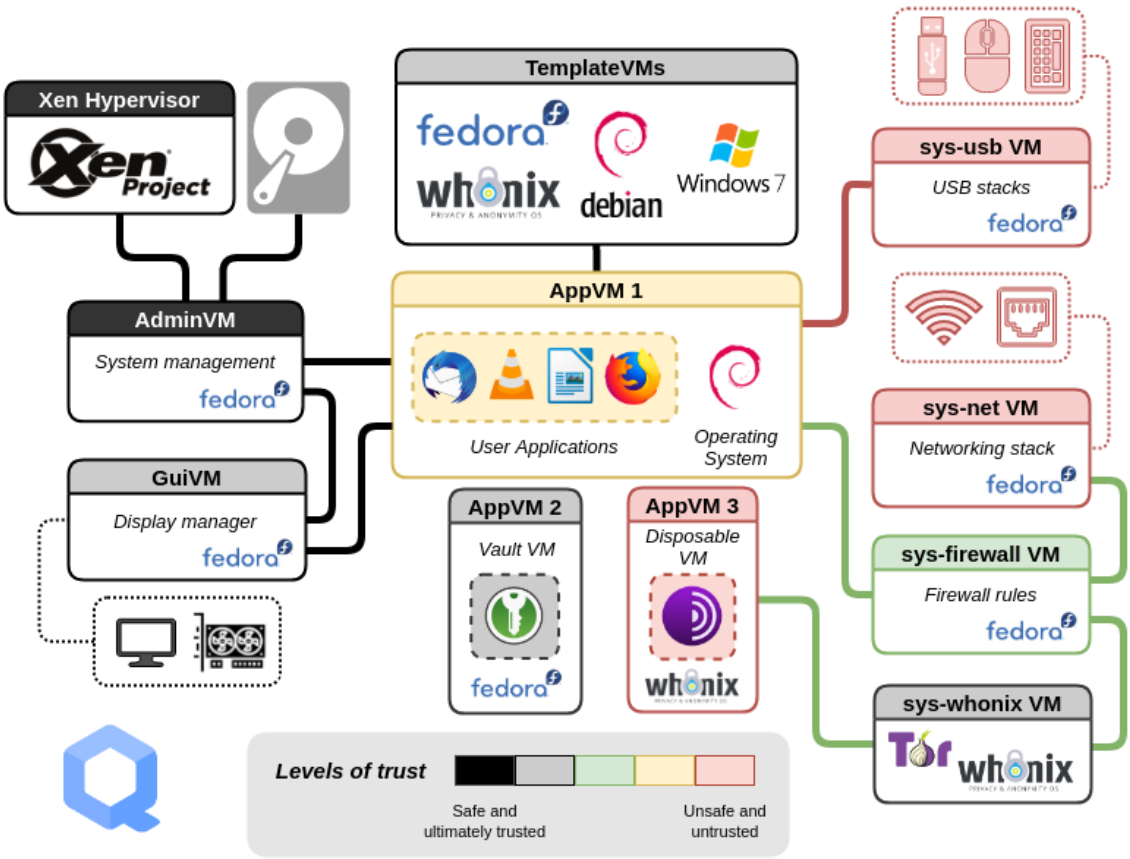


Рисунок 1.3 – Приклад схеми роботи Qubes OS

Tails (The Amnesic Incognito Live System) – анонімна система Linux.

Це портативна ОС що надає захист користувачеві від цензури та відстежування, та головна ціль якої – дотримання анонімності та конфіденційності користувача. Для анонімності в ній використовується мережа Tor. Будь які спроби підключення до Інтернету без мережі Tor блокуються. Також дії користувача не залишають слідів на комп'ютері, оскільки Tails працює як Live-CD система, а отже не залишає ніяких файлів на запам'ятовуючих пристроях. Щоб зберегти власні файли, користувачеві потрібно це зробити в окремій зашифрованій папці. Tails має широку підтримку спільноти та широку документацію.

Серед програмних засобів Tails можна виділити:

- Tor Browser – захищений анонімний браузер з модулем uBlock.
- Thunderbird – для шифрованої пошти.
- KeePassXC – менеджер паролів.
- LibreOffice – вільний офісний пакет.
- OnionShare – обмін файлів через мережу Tor.
- Metadata Cleaner – очищувач метаданих файлів.

Окремо важливо зазначити ОС Whonix, яка в початковому стані є чимось середнім між Qubes та Tails: для безпеки використовуються 2 віртуальні машини (одна окремо як проксі для перенаправлення трафіку через Tor), і весь трафік також перенаправляється через Tor. Для найкращої безпеки та анонімності одночасно краще використовувати Qubes з кожним контейнером налаштованим на трафік через Tor.

Discreete Linux – ізольоване середовище для захисту інформації від шпигунського ПЗ та троянів.

Головною метою Discreete Linux надати локалізоване, безпечне середовище для обробки чутливої інформації без загроз її витоку. Це виконується за допомогою трьох головних ідей: "Закрити входи", "Запобігти поширенню", "Закрити виходи".

"Закрити входи" означає закрити можливі шляхи зловмиснику зайти в систему. Це досягається за допомогою: запобігання мережевих атак, а саме шляхом роботи системи в повністю ізольованому від мереж режимі – цей функціонал був видалений з коду ядра; запобігання використання ATA дисків – в разі, коли локальний диск може

бути попередньо заражений вірусом, що стосується не тільки шкідливих програм, що зберігаються на диску, а й зловмисної прошивки (firmware); файлових систем без прав на виконання – оскільки від'єднуємі накопичувачі являються єдиним входом до системи, потрібно запобігти найбільш поширеній моделі атаки на ізольовані системі від мережі з використанням інфекованих портативних носіїв.

Пункт "Запобігти поширенню" розроблений для сценаріїв, де зловмисне ПЗ все ж таки дісталося цільової системи. Для такого випадку розроблені та/або імплементовані наступні функції: незмінна система, що є read-only (тільки для читання) – вся система зберігається в оперативній пам'яті, зміни до якої не зберігаються після виключення комп'ютера; використання тільки підписаних командою Discreete Linux Team модулів ядра; запобігання отримання root-привілеїв.

Реалізація пункту "Закрити виходи" тісно йде з першим пунктом, оскільки шляхи входу і виходу із системи однакові – за допомогою мережі, внутрішніх та зовнішніх дисків та інших носіїв інформації. Таким чином зловмисний код при дбалій моделі оперативної безпеки (OPSEC) та чутлива інформація не вийдуть за межі зараженої, або заражених систем.

На сайті дистрибутиву вказано, що він більше націлений на викривачів, різного роду активістів, журналістів, юристів та інших, хто може стати ціллю спрямованих кібератак та шпигунства. Таким чином він ставить за мету бути простим у вивченні, використанні та впроваджувати механізми захисту від недосвідчених користувачів, що могли б порушити безпеку системи. Єдиним шляхом роботи з системою Discreete, а також методом імпорту інформації для обробки ті її обміну являються USB-накопичувачі. В системі наявні криптографічні модулі для шифрування даних.

Особиста думка: через деякі непрямі "червоні флаги", такі як: відсутність власного підфоруму Reddit, як і майже повна відсутність обговорень на форумах цієї ОС, припинена розробка дистрибутиву та відсутність активності з боку розробників з 2016 року, відносно мала кількість інформації про ОС в просторі Інтернет і, як наслідок, низький статус перевіреності та довіри до цього дистрибутиву, невідповідна назва веб-сайту назві дистрибутиву, а також дуже заохочуючий дизайн цього сайту можуть вказувати на зловмисний вміст системи. Це може бути так званим "honeypot"-

ом для недосвідчених користувачів, ймовірно бажаючих приховати свою нелегальну діяльність, або просто для компрометації даних тих користувачів, яким є що приховувати. Перед використанням дистрибутиву рекомендується прискіпливо проаналізувати співвідношення можливі ризики та ймовірних переваг від використання системи, а також роздивитись інші, більш популярні рішення з аналогічним функціоналом. До того ж, при такій моделі загроз, на яку зорієнтовано дистрибутив, можливо є більш долучним власноруч проаналізувати шляхи досягнення тих цілей, яких намагається досягнути Discreete Linux.

Висновки до розділу 1

В даному розділі було проаналізовано та описано UNIX-подібні операційні системи, їх архітектуру, а також технології та програмні рішення для організації захисту цих систем. Були досліджені такі аспекти, як мікроядерна архітектура та монолітне ядро, простір користувача та простір ядра, віртуалізація і контейнеризація, системні виклики, аспекти файлових систем та різні технології, що використовуються у проаналізованих спеціалізованих UNIX-подібних операційних системах.

Також були розглянуті спеціалізовані дистрибутиви Linux для використання у різних аспектах інформаційної безпеки, таких як пентестинг, цифрова криміналістика, етичний хакінг, зворотня інженерія, дослідження безпеки, особиста безпека, конфіденційність та анонімність.

РОЗДІЛ 2
РОЗРОБКА МОДЕЛІ БЕЗПЕКИ ТА ВИМОГ ДЛЯ СПЕЦІАЛІЗОВАНОГО ДИСТРИБУТИВУ LINUX

2.1 Опис завдань з розробки дистрибутиву

Практична частина даної роботи спрямована на розробку програмної платформи для навчання фахівців з кібербезпеки. Серед завдань були поставлені наступні:

- Розробка плану та структури роботи.
- Формулювання та обґрунтування вимог, визначення мети та завдань роботи.
- Аналіз вимог та розробка концепції.
- Розробка та реалізація програмної частини.
- Налаштування системи.
- Формування рекомендацій щодо подальшого розвитку системи.

2.2 Формулювання мети, цілей та планування розробки

Мета – розробка спеціалізованого робочого інструменту фахівця у галузі кібербезпеки для виконання наступних завдань:

- Забезпечення студентів та фахівців з кібербезпеки набором програмного забезпечення для навчання, відпрацювання навичок, розвитку знань та вмінь у галузі інформаційних технологій та кібербезпеки і вирішення професійних завдань.
- Заміна пропрієтарного ПЗ на відповідне йому вільне для використання у сферах кібербезпеки та освіти.
- Забезпечення можливості підтримки актуальності, динамічного розвитку та модернізації програмних додатків та компонентів операційної системи у відповідності до стану розвитку галузі та потреб спеціалістів з кібербезпеки.

Навчальна операційна система для фахівців у галузі інформаційної безпеки повинна бути спрямована на підвищення рівня технічної підготовки та автоматизацію практичних навичок і рутин користувачів у цій галузі. Вона має забезпечувати якісне навчання у дистанційному режимі, інтуїтивний та доступний інтерфейс з можливістю детального налаштування, запровадити більш якісне середовище під час самостійного опрацювання різноманітних модулів та завдань, а також вказати на доцільність розробки централізованої системи оцінювання та контролю успішності навчального прогресу окремих користувачів.

Завдання щодо розробки дистрибутиву:

- Визначення вимог до програмного забезпечення фахівця в галузі кібербезпеки.
- Розробка моделі безпеки.
- Вибір програмної бази (дистрибутиву ОС) для створення робочого інструменту фахівця у галузі кібербезпеки.
- Вибір прикладного програмного забезпечення для надання достатнього інструментарію фахівцям з кібербезпеки.
- Забезпечення можливості розвитку та довгострокової підтримки актуальності системи.

Цілі дистрибутиву:

- Розробка та впровадження моделі безпеки ОС загального призначення спеціаліста з кібербезпеки.
- Забезпечення інструментів для покращення рівня безпеки ОС та використання їх при реалізації моделі безпеки ОС.
- Забезпечення функціоналу та інструментарію для виконання різних завдань для отримання навичок інформаційної безпеки.
- Забезпечення зручності та ефективності у використанні для користувачів різних рівнів підготовки та досвіду роботи з комп'ютерами. Цю ціль можна виконати як за допомогою зручного інтерфейсу та доступної документації, так і написанням інструкцій для виконання тих чи інших завдань.

- Забезпечення надійності та стабільності роботи системи.
- Надання можливості розширюваності, оновлення та розвитку системи.
- Надання можливості користувачам отримати теоретичні знання та практичні навички в області ІБ шляхом створення спеціалізованого дистрибутиву з програмним забезпеченням як платформи для навчання.
- Забезпечення інструменту з можливостями для самостійної підготовки та підвищення кваліфікації фахівців у галузі ІБ.

2.3 Розробка моделі безпеки ОС

2.3.1 Опис моделі безпеки

Дана модель безпеки описує цілі, політики та етапи їх розробки і впровадження, механізми та заходи безпеки, модель загроз та модель управління інцидентами, які можливо та бажано використовувати для забезпечення цілісності, конфіденційності та доступності ОС. В ній наданий широкий спектр аспектів безпеки, які можливо використати для досягнення підвищеного рівня безпеки системи. Варто зазначити, що ця модель розроблялась без прив’язки до будь-яких стандартів безпеки, а отже для визначення рівня безпеки, який надасть повна реалізація даної моделі на реальній системі, потрібно проводити аудит безпеки за обраними стандартами.

Також в даній моделі не розглянуто аспекти фізичної безпеки та проведення тренингів безпеки, оскільки фізична безпека не відноситься безпосередньо до ОС, а більше стосується апаратного забезпечення, на якому вона розташована. Аспект проведення тренингів, з іншого боку, вважається автоматично виконаним з огляду на те, що ця модель розробляється для ОС спеціаліста з кібербезпеки, тобто цільова аудиторія повинна бути обізнаною або знаходитись в процесі вивчення інформаційної безпеки.

Також ця модель в першу чергу спрямована на ОС з єдиним користувачем, тому впровадження групових політик є опціональним, проте фунціонал для сценаріїв з кількома користувачами є бажаним.

Розробка даної моделі безпеки велась з використанням публікацій NIST [29] та [30], проте тільки в контексті зрозуміння міжнародних та американських стандартів, тобто ніякої прив’язки до цих документів зроблено не було. Також ця модель розроблялась з огляду на джерела [31] та [32], в яких розглянуто безпеку Linux.

2.3.2 Мета та цілі моделі безпеки

Головною метою моделі безпеки є формулювання заходів щодо конфіденційності, цілісності та доступності даних та системи. Це повинно досягатися наступними шляхами:

- Забезпечення конфіденційності: організація управління доступом до конфіденційної інформації. Це включає захист від несанкціонованого доступу, витоку інформації та недостатньої захищеності даних.
- Забезпечення цілісності: організація заходів щодо забезпечення захисту від несанкціонованої зміни, пошкодження або знищення інформації. Це включає виявлення та захист від вторгнень, вірусів, зловмисного програмного забезпечення та інших загроз цілісності даних, а також забезпечення можливості резервного копіювання та відновлення інформації.
- Забезпечення доступності: забезпечення належного функціонування системи та доступу до ресурсів для авторизованих користувачів. Це включає захист від атак типу відмова обслуговування (DoS), а також забезпечення інструментів для організації доступу до системи в разі виникнення проблем доступності.
- Управління інцидентами безпеки: виявлення, відстеження та відповідь на інциденти безпеки.
- Управління загрозами: Оцінка, ідентифікація та керування загрозами, пов'язаними з безпекою операційної системи. Включає проведення аудиту безпеки,

виявлення та усунення вразливостей, впровадження політик безпеки та керування інцидентами.

- Забезпечення відповідності вимогам: впровадження заходів, спрямованих на перевірку відповідності системи вимогам та політикам безпеки.

2.3.3 Етапи розробки та впровадження політик безпеки

1. Аналіз поточного стану безпеки, оцінка системи:

❖ Оцінка наявних заходів безпеки: оцінка наявних механізмів та заходів безпеки, таких як налаштування фаєрволу, облікові записи користувачів, шифрування даних тощо.

❖ Аудит безпеки: Використання інструментів, таких як OpenSCAP, для проведення аудиту системи згідно з встановленими стандартами та політиками безпеки.

2. Розробка політик безпеки:

❖ Визначення цілей безпеки: визначення конкретних цілей безпеки, які потрібно досягти, враховуючи особливості вашої системи та потенційні загрози.

❖ Створення політик безпеки: розробка документованих політик безпеки, які визначають правила, процедури та стандарти, що мають бути дотримані для забезпечення безпеки системи.

3. Розробка плану впровадження:

- Визначення пріоритетів: встановлення пріоритетів щодо впровадження політик безпеки, визначення критичних аспектів, які потребують негайної уваги.
- Розподіл відповідальності: визначення відповідальних осіб або команди, які будуть відповідати за впровадження політик безпеки та контроль їх дотримання.

4. Реалізація політик безпеки:

- Виконання технічних заходів: налаштування системи згідно з встановленими політиками безпеки, встановлення необхідного програмного забезпечення та інструментів безпеки.
- Навчання та свідомості користувачів: Проведення навчання користувачів щодо правил безпеки, впровадження свідомості щодо потенційних загроз та заходів безпеки.

5. Моніторинг та аудит:

- Встановлення системи моніторингу: встановлення і налаштування системи моніторингу безпеки, яка дозволить виявляти потенційні загрози та невідповідності політикам безпеки.
- Проведення регулярних аудитів: Виконання регулярних аудитів безпеки системи, щоб переконатися в дотриманні політик безпеки та виявити можливі вразливості.

6. Оновлення та вдосконалення:

- Постійне оновлення політик безпеки: періодичне оглядання та оновлення політик безпеки з урахуванням нових загроз та вимог безпеки.
- Впровадження вдосконалень: впровадження вдосконалень у систему безпеки на основі виявлених вразливостей, недоліків або досвіду з реагування на інциденти.

2.3.4 Політики безпеки

1. Загальна політика безпеки: визначає набір принципів, процедур та вимог, що стосуються безпеки операційної системи. Вона охоплює всі аспекти безпеки, включаючи управління загрозами, управління інцидентами та захист інформації. Ця політика спрямована на забезпечення високого рівня безпеки ОС та її користувачів.

Мета: метою загальної політики безпеки ОС є:

- Забезпечення конфіденційності, цілісності та доступності інформації, що зберігається та обробляється в ОС.

- Запобігання несанкціонованому доступу до системи та недозволеним змінам конфігурації.
- Виявлення, відстеження та вирішення інцидентів безпеки, що виникають в ОС.

Положення:

➢ Управління загрозами:

❖ Визначення методів інвентаризації активів, оцінки загроз та встановлення прийнятних рівнів ризику.

❖ Розробка та впровадження заходів мінімізації загроз, включаючи контролі та захист активів, перевірку сторонніх постачальників та інші стратегії зниження ризиків.

➢ Управління інцидентами:

❖ Визначення процедур виявлення, реєстрації, розслідування та реагування на інциденти безпеки.

❖ Розробка плану відновлення після інциденту та впровадження необхідних заходів для мінімізації впливу інцидентів на систему.

➢ Управління доступом: визначення політик і процедур управління доступом до ресурсів ОС, включаючи ідентифікацію та аутентифікацію користувачів, контроль прав доступу та аудит доступу.

➢ Захист інформації: встановлення механізмів шифрування, захисту мережевого зв'язку та захисту даних від несанкціонованого доступу або втрати.

➢ Постійне вдосконалення: встановлення процесів перегляду, аудиту та оновлення політики безпеки ОС для відповідності постійно мінливим загрозам та вимогам.

2. *Політика розмежування доступу:* визначає принципи та правила для розмежування доступу до різних ресурсів та функцій операційної системи. Ця політика спрямована на забезпечення конфіденційності, цілісності та доступності даних і функцій, а також запобігання несанкціонованому доступу та зловживанням.

Мета: забезпечити належне розмежування доступу до системних ресурсів та функцій шляхом встановлення відповідних обмежень, правил та механізмів контролю, що допоможе уникнути неповноважного використання, зловживання та витоку конфіденційної інформації.

Положення:

- Встановлення рівня доступу: визначити рівень доступу до ресурсів та функцій системи на основі ролей, обов'язків та потреб користувачів.
- Механізми ідентифікації, аутентифікації та авторизації: використовувати механізми ідентифікації, аутентифікації та авторизації для перевірки ідентифікації користувачів та надання відповідного рівня доступу.
- Принципи найменших привілеїв: застосовувати принципи найменших привілеїв, що передбачають надання користувачам та процесам лише необхідних привілеїв для їх потреб.
- Захист конфіденційної інформації: визначити правила та обмеження для доступу до конфіденційної інформації та ресурсів, забезпечити їх захист від неповноважного копіювання, змін або передачі.
- Засоби захисту: використовувати ефективні механізми захисту від несанкціонованого доступу, такі як паролі, шифрування, біометричні методи аутентифікації та технічні засоби.
- Індивідуальні облікові записи: забороняти спільне використання облікових записів та передачу доступу до особистих облікових записів.
- Моніторинг та оцінка: забезпечити регулярний моніторинг та оцінку ефективності політики розмежування доступу та впроваджувати відповідні корективи для забезпечення високого рівня безпеки.
- Парольна політика: регулярно оновлювати паролі та використовувати сильні парольні політики.

3. Політика мережевої безпеки: визначає принципи та правила, що стосуються безпеки аспектів операційної системи, пов'язаних з її взаємодією з мережею. Ця політика спрямована на захист користувача та системи під час взаємодії з мережевими ресурсами та іншими аспектами використання мережевих інтерфейсів.

Мета: забезпечення безпеки операційної системи та її користувача, а також мінімізація ризиків зовнішніх атак та несанкціонованого доступу до системи.

Положення:

- Застосовувати захист мережевого трафіку шляхом використання мережевих протоколів з шифруванням, фаєрволів та інших механізмів для запобігання несанкціонованому доступу та атакам.
- Використовувати захищені мережеві з'єднання, такі як віртуальні приватні мережі (VPN), для забезпечення безпеки під час віддаленого доступу до системи та передачі даних.
- Регулярно оновлювати програмне забезпечення для усунення вразливостей, що може використати зловмисник для отримання несанкціонованого доступу або інших порушень цілісності, конфіденційності та доступності.
- Забезпечити моніторинг мережі та реагування на підозрілу або небезпечну активність, використовуючи системи виявлення вторгнень та інші інструменти моніторингу.
- Проводити регулярну оцінку та аудит безпеки мережі з метою виявлення потенційних загроз та вдосконалення політики мережевої безпеки.

4. *Політика управління загрозами та ризиками:* в цій політиці визначені вимоги до визначення, аналізу, оцінки та контролю загроз безпеці, що разом формують модель загроз. Це включає в себе такі етапи, як ідентифікація загроз, оцінка їх імовірності та наслідків, розробка стратегій мінімізації ризиків та впровадження заходів безпеки для зменшення ймовірності реалізації загроз або їхнього впливу на систему. Спрямоване на забезпечення адекватного рівня безпеки і зменшення вразливостей системи шляхом впровадження заходів безпеки, моніторингу та аналізу загроз, планування заходів у разі виникнення інцидентів та постійного вдосконалення процесів безпеки.

Мета: ідентифікація, аналіз, оцінка та контроль загроз безпеці з ціллю забезпечення адекватного рівня захисту та зменшення вразливостей системи. Політика спрямована на розробку та впровадження стратегій мінімізації ризиків та заходів безпеки, а також на постійне вдосконалення процесів безпеки.

Положення:

- Розробити стратегії та плани мінімізації ризиків, включаючи впровадження заходів безпеки, які зменшують імовірність виникнення інцидентів та їхній вплив на систему.
- Забезпечити виконання регулярних оновлень для операційної системи та програмного забезпечення з метою усунення вразливостей та запобігання експлуатації потенційних загроз.
- Встановити механізми резервного копіювання та відновлення для забезпечення надійності даних.
- Здійснювати моніторинг та аналіз загроз безпеці, оновлення стратегій мінімізації ризиків на основі нових загроз або змін в інфраструктурі.
- Планувати заходи у разі виникнення інцидентів, включаючи відновлення після інциденту та забезпечення безперебійності діяльності системи.
- Здійснювати оцінку ефективності заходів безпеки та процесів управління загрозами та ризиками і вносити необхідні виправлення та вдосконалення.
- Проводити аудит системи для виявлення вразливостей та потенційних загроз.

5. *Політика аудиту та моніторингу системи:* визначає принципи та процедури, які стосуються аудиту та моніторингу безпеки операційної системи. Ця політика спрямована на забезпечення постійного контролю, виявлення вразливостей та потенційних загроз, а також вчасну реакцію на події, що відбуваються в системі.

Мета: забезпечення постійного контролю та відстеження активності в операційній системі з метою виявлення потенційних загроз безпеці, вразливостей та аномальних активностей. Вона спрямована на забезпечення відповідності встановленим правилам безпеки, виявлення порушень та вчасну реакцію на інциденти.

Положення:

- Встановлення подій для аудиту: визначення списку подій, які підлягають аудиту, включаючи входи, виходи, зміни конфігурації, доступ до ресурсів, спроби несанкціонованого доступу та інші важливі події.

- Конфігурація аудиту: встановлення параметрів аудиту, включаючи рівень деталізації, обсяг зберігання журналів аудиту та механізми захисту аудитованих даних.
- Моніторинг активності: використання інструментів моніторингу для постійного відстеження активності в системі, включаючи мережевий трафік, журнали подій, системні ресурси та інші важливі параметри.
- Аналіз та виявлення загроз: використання аналітичних методів та інструментів для аналізу зібраних даних та виявлення потенційних загроз безпеці, аномальних патернів поведінки та інших важливих ознак несправностей або атак.
- Реагування на інциденти: визначення процедур реагування на виявлені інциденти безпеки, включаючи відповідність інцидентам, ізоляцію та відновлення системи, повідомлення відповідним сторонам та проведення розслідувань.

6. *Політика виявлення та реагування на вразливості:* визначає процедури та відповідальності, пов'язані з виявленням, оцінкою та реагуванням на вразливості в операційній системі. Ця політика спрямована на попередження та ефективне усунення вразливостей, забезпечення безпеки системи та захисту від можливих атак.

Мета: забезпечення систематичного виявлення, оцінки та реагування на вразливості в операційній системі з метою зменшення ризиків злому, несанкціонованого доступу та інших потенційних загроз безпеці. Вона спрямована на постійне оновлення системи та програмного забезпечення для виправлення виявлених вразливостей та забезпечення адекватного рівня захисту.

Положення:

- Моніторинг вразливостей: встановлення механізмів та інструментів для постійного моніторингу вразливостей в операційній системі та використання баз даних вразливостей для виявлення нових загроз.
- Активний пошук вразливостей: перевірка захисту системи за допомогою інструментів тестування на проникнення.
- Оцінка та класифікація вразливостей: розробка процедур для оцінки важливості та критичності виявлених вразливостей, що дозволить приділити пріоритетну увагу усуненню найбільш значимих загроз.

- Впровадження заходів усунення вразливостей: розробка процедур для внесення необхідних змін у систему та програмне забезпечення з метою усунення виявлених вразливостей.
- Реагування на критичні вразливості: розробка процедур для негайного реагування на виявлення критичних вразливостей, включаючи прийняття невідкладних заходів для захисту системи та забезпечення її безпеки.
- Оновлення та перевірка вразливостей: встановлення процедур регулярного оновлення системи та програмного забезпечення з метою виправлення виявлених вразливостей та перевірки ефективності цих заходів після внесення змін.

7. *Політика безпеки та захисту даних:* визначає принципи та процедури, що стосуються захисту конфіденційності, цілісності та доступності даних, зокрема в операційній системі. Ця політика спрямована на запобігання несанкціонованому доступу до даних, втрати чи пошкодженню інформації, а також забезпечення відповідного рівня захисту для забезпечення конфіденційності та цілісності даних.

Мета: метою політики безпеки та захисту даних є забезпечення високого рівня захисту даних в операційній системі. Це включає розробку та впровадження політик, процедур та технічних засобів для захисту даних від несанкціонованого доступу, втрати, пошкодження та несанкціонованого використання.

Положення:

- Класифікація та оцінка даних: встановлення процедур для класифікації даних за рівнем конфіденційності та визначення відповідних заходів захисту для кожної категорії даних. Оцінка загроз пов'язаних з даними та визначення пріоритетів заходів захисту.
- Логічний захист даних: розробка та впровадження механізмів автентифікації, авторизації та аудиту доступу до даних. Це включає встановлення паролів, ролей користувачів, шифрування даних та інших технічних заходів для забезпечення контролю доступу та захисту даних від несанкціонованого використання.
- Резервне копіювання та відновлення даних: розробка процедур регулярного резервного копіювання даних та планів відновлення в разі втрати або

пошкодження інформації. Впровадження засобів для забезпечення доступності та цілісності даних під час відновлення.

- Застосовування шифрування для захисту конфіденційної інформації.
- Встановлення механізмів контролю доступу до даних та заборона несанкціонованого доступу до них.

8. Політика безпеки додатків: визначає принципи та процедури, які стосуються безпеки додатків, які використовуються в операційній системі. Ця політика спрямована на забезпечення безпечного використання, установку та взаємодії з додатками, а також на попередження вразливостей та небажаних наслідків, пов'язаних з використанням ненадійних або шкідливих додатків.

Мета: забезпечення безпечного та надійного середовища для використання додатків в операційній системі. Це включає розробку та впровадження процедур, механізмів та контролів, які зменшують ризики експлуатації вразливостей, проникнення шкідливого програмного забезпечення та несанкціонованого доступу до системи через додатки.

Положення:

- Сертифікація та автентифікація додатків: встановлення процедур для перевірки та сертифікації додатків перед їх встановленням. Забезпечення автентичності джерела додатків, щоб уникнути встановлення шкідливого або ненадійного програмного забезпечення.
- Обмеження привілеїв додатків: встановлення механізмів для обмеження привілеїв, доступу та поведінки додатків у системі. Використання принципу найменших привілеїв та контролю доступу для запобігання небажаним діям та зловживанням.
- Моніторинг та виявлення вразливостей: розробка процедур та механізмів для постійного моніторингу додатків на вразливості та небажані дії. Використання системи виявлення вторгнень та механізмів аналізу коду для ідентифікації потенційних загроз та вразливостей.

- Оновлення безпеки: встановлення процедур регулярного оновлення додатків, включаючи установку оновлень безпеки та виправлення вразливостей. Актуалізація додатків для забезпечення їх надійності та безпеки.

9. *Політика забезпечення актуальності системи, регулярності та довіреності оновлень:* встановлює вимоги щодо регулярності оновлень операційної системи та її програмного забезпечення з метою забезпечення захисту від вразливостей та підтримки її надійності. Вона спрямована на забезпечення вчасного впровадження оновлень та застосування безпечних конфігурацій.

Мета:

- Забезпечити можливість якомога швидшого застосування оновлень для оперативного усунення вразливостей.
- Запобігти використанню відомих вразливостей шляхом постійного оновлення системи.
- Підтримувати безпеку шляхом створення постійно змінюваної системи за допомогою використання моделі швидких оновлень.

Положення:

- Регулярні оновлення системи та її компонентів:
  - Встановлювати оновлення ОС та її компонентів якомога швидше після їх випуску.
  - Автоматизувати процес оновлення, використовуючи інструменти для керування оновленнями.
  - Періодично перевіряти наявність нових оновлень.
  - Використовувати актуальні версії програмного забезпечення з офіційних джерел або надійних джерел постачальників.
  - Встановлювати оновлення програмного забезпечення безпосередньо після їх випуску або відповідно до визначеного графіку.
  - Використовувати механізми автоматичного оновлення для мінімізації загроз.
- Документування та відповідальність:

- ❖ Вести журнал оновлень та патчів, включаючи дати, версії та джерела.
- ❖ Документувати процедури, політики та вимоги з оновлення та патчування для майбутнього посилання.

Для всіх політик було розроблено наступний загальний план впровадження:

1. Визначення механізмів та заходів для виконання політики.
2. Розгортання та налаштування механізмів та інструментів необхідних для виконання політики.
3. Перевірка дійсності, правильності та ефективності впровадження політики безпеки.

2.3.5 Механізми безпеки

- Ідентифікація та аутентифікація:
  - ❖ Впровадження механізмів аутентифікації користувачів та системних облікових записів.
  - ❖ Вимоги до паролів, використання багатофакторної аутентифікації.
  - ❖ Механізми керування доступом на основі ролей та привілеїв.
- Керування доступом та авторизація:
  - ❖ Визначення правил та політик доступу до файлів та ресурсів системи.
  - ❖ Управління групами та ролями, призначення привілеїв.
  - ❖ Використання різних рівнів доступу (наприклад, адміністратор, користувач, гість)
- Шифрування та захист даних:
  - ❖ Встановлення політик щодо шифрування даних в спокої та під час передачі.
  - ❖ Захист конфіденційної інформації, включаючи шифрування дискового простору, файлів та комунікаційних каналів.
- Моніторинг та аудит:

- ❖ Встановлення механізмів аудиту подій безпеки та ведення журналів.
- ❖ Аналіз та моніторинг активності системи для виявлення вторгнень та аномальної активності, аналіз журналів подій.
- ❖ Зовнішнє тестування системи за допомогою інструментів пентестингу.
- ❖ Сповіщення про безпекові події та інциденти.

- Захист від вразливостей:
  - ❖ Встановлення регулярних оновлень і патчів для операційної системи та програмного забезпечення.
  - ❖ Використання системи моніторингу вразливостей для виявлення потенційних загроз і вразливостей.
  - ❖ Використання програмного забезпечення, що реалізує механізми захисту від шкідливих програм, такі як антивірус, IDS та IPS.
- Резервне копіювання та відновлення:
  - ❖ Регулярне створення резервних копій важливих даних і системних налаштувань.
  - ❖ Перевірка резервних копій на цілісність та доступність для відновлення.
  - ❖ Планування та проведення тестових випробувань процедур відновлення.
- Управління загрозами, ризиками та інцидентами.

2.3.6 Модель загроз

Ідентифікація активів:

- Операційна система та її складові:
  - ❖ Процеси та сервіси (демони).
  - ❖ Ядро системи (та драйвери, як його складова).

- Програмні додатки та інше програмне забезпечення:
  - ❖ Мережеві інструменти та протоколи.
  - ❖ Інструменти аналізу ПЗ.
  - ❖ Інструменти розробки.
  - ❖ Утиліти.
  - ❖ Інше ПЗ.
- Дані та інформація:
  - ❖ Конфіденційні дані (особисті дані, паролі).
  - ❖ Конфігураційні файли.
  - ❖ Журнали подій.
- Користувачі:
  - ❖ Адміністратори системи (root та sudoers).
  - ❖ Користувачі з додатковими привілеями.
  - ❖ Звичайні користувачі.
- Інфраструктура безпеки:
  - ❖ Файрволи.
  - ❖ Антивірусне та антишпигунське ПЗ.
  - ❖ IDS та IPS.
  - ❖ Системи моніторингу та захисту системи, IDS та IPS.

Аналіз загроз:

- Операційна система та її складові:
  - ❖ Загроза: вразливості в операційній системі можуть бути використані зловмисниками для отримання несанкціонованого доступу або виконання шкідливого коду.
  - ❖ Можливі заходи безпеки: регулярне оновлення операційної системи та встановлення патчів, використання механізмів ідентифікації, аутентифікації та авторизації, використання сильних паролів, обмеження привілеїв користувачів.
- Програмні додатки та інше програмне забезпечення:

- ❖ Загроза: наявність вразливостей в програмному забезпеченні може призвести до несанкціонованого доступу до системи або втрати конфіденційності даних.
- ❖ Можливі заходи безпеки: регулярне оновлення програмного забезпечення, використання механізмів шифрування, перевірка вихідного коду на вразливості перед виконанням, налаштування відповідних прав доступу.

- Дані та інформація:
  - ❖ Загроза: несанкціонований доступ до конфіденційних даних може призвести до втрати приватності або використання даних від несанкціонованих осіб.
  - ❖ Можливі заходи безпеки: застосування механізмів шифрування для конфіденційних даних, забезпечення резервного копіювання та відновлення даних, контроль доступу до даних, використання сильних паролів.
- Користувачі:
  - ❖ Загроза: відсутність політики розмежування доступу та привілеїв користувачів може призвести до втрати контролю над системою або зловживання привілеями.
  - ❖ Можливі заходи безпеки: обмеження доступу до адміністративних привілеїв, використання різних облікових записів для різних завдань, надання доступу лише необхідним користувачам.
- Інфраструктура безпеки:
  - ❖ Загроза: недостатня захищеність інфраструктури безпеки може призвести до несанкціонованого доступу або втрати контролю над системою.
  - ❖ Можливі заходи безпеки: встановлення та налаштування фаєрволу для контролю мережевого трафіку, використання антивірусного та антишпигунського ПЗ, встановлення систем моніторингу та виявлення

вторгнень, обмеження привілеїв для запобігання зміни налаштувань безпеки та відключення елементів інфраструктури безпеки.

- Зовнішня мережева інфраструктура:
  - ❖ Загроза: атаки з підміною мережевого трафіка, його прослуховування і аналіз та інші атаки типу Man-in-the-Middle.
  - ❖ Можливі заходи безпеки: шифрування трафіку, використання асиметричних ключів, перевірка сертифікатів, багатофакторна аутентифікація, примусове використання HTTPS, налаштування безпеки зовнішньої інфраструктури.

Управління загрозами:

1. Прийняття заходів безпеки: розробка та виконання стратегій та заходів безпеки для зменшення загроз (встановлення оновлень, налаштування політик безпеки, моніторинг активності, встановлення антивірусного, антишпигунського програмного забезпечення, IDS, IPS).
2. Відстеження вразливостей: постійне сканування та оновлення вразливостей системи, впровадження виправлень та патчів.
3. Керування інцидентами: розроблення планів дій для реагування на інциденти безпеки, виявлення та відновлення після інцидентів.

Моніторинг та оцінка:

1. Моніторинг системи: встановлення механізмів моніторингу безпеки системи (наприклад, журналів подій, моніторинг мережі) для виявлення можливих атак та незвичайної активності.
2. Аудит безпеки: регулярне проведення аудиту безпеки для перевірки ефективності заходів безпеки, виявлення потенційних вразливостей та виявлення аномалій.
3. Оновлення та вдосконалення: постійне оновлення та вдосконалення моделі загроз на основі нових загроз та викликів.

2.3.7 Модель управління інцидентами безпеки

Виявлення інцидентів:

- Моніторинг безпеки: встановлення систем моніторингу та спостереження за подіями, які можуть вказувати на можливі безпекові порушення.
- Виявлення загроз: використання систем детекції вторгнень (IDS), аналізу журналів подій, мережевих сканерів тощо для виявлення потенційних загроз.

Реагування на інциденти:

- Установлення процедур реагування: розроблення документованих процедур, які описують кроки реагування на різні типи інцидентів безпеки.
- Ізоляція та обмеження: прийняття заходів для ізоляції та обмеження поширення інциденту, включаючи відключення компрометованих систем, заборону доступу тощо.

Розслідування та аналіз інцидентів:

- Збір доказів: збір і аналіз доказів, пов'язаних з інцидентом, включаючи журнали подій, системні файли, мережевий трафік та інші відповідні дані.
- Аналіз причин: визначення причин інциденту, виявлення уразливостей системи, людських помилок або інших факторів, які сприяли виникненню інциденту.

Відновлення та реагування

- Відновлення послідовності роботи: відновлення нормального функціонування системи після інциденту, включаючи відновлення даних, конфігурацій та сервісів.
- Постійне вдосконалення: оцінка ефективності заходів, які були прийняті під час реагування на інцидент, та вдосконалення процедур інцидентного менеджменту на основі набутих досвіду.

Звітність та документування

- Створення звітів: підготовка звітів про інциденти, включаючи опис подій, застосовані заходи, виявлені уразливості та рекомендації щодо запобігання подібним інцидентам у майбутньому.

- Документування процедур: документування процедур реагування на інциденти, аналізу причин, відновлення та інших процесів управління інцидентами безпеки.

2.4 Розробка вимог до дистрибутиву, ПЗ та його функціоналу

2.4.1 Загальні вимоги до ПЗ

Будь-яке ПЗ, що є частиною плану розробки дистрибутиву ОС інформаційного захисту у межах даної роботи, має виконувати наступні вимоги:

1. Модель розповсюдження: ПЗ має розповсюджуватися безкоштовно та належати до групи Free/Libre and Open Source Software (FLOSS, FOSS), тобто розповсюджуватися за однією з відкритих ліцензій (типу GNU GPL), або мати статус Public Domain (програмне забезпечення загального користування, тобто авторські права відсутні), або за будь-якою іншою ліцензією, в межах якої будуть дані наступні свободи або права для використання даного ПЗ у дистрибутиві:
   - ❖ Свобода запуску програми з у будь-яких не зловмисних цілях.
   - ❖ Свобода вивчення роботи програми та її модифікації.
   - ❖ Свобода поширення копій вихідного та виконуваного коду, або дозвіл на обмежене поширення вихідного та виконуваного коду в межах проекту.
   - ❖ Свобода покращення програми та випуску покращень в публічний доступ, або дозвіл на покращення програми та випуску покращень в обмежений доступ для цілей проекту.
2. Наявність відомостей про першоджерела, автора ПЗ, а також відповідних посилань. Джерело ПЗ повинно бути довіреним.
3. Наявність документації.
4. Безпечність: ПЗ має відповідати вимогам безпеки, зокрема код ПЗ має бути перевірений на шкідливі фрагменти (включаючи небезпечні бібліотеки) та на вразливості.

5. Доведеність відповідності ПЗ вимогам до нього.
6. Відповідність вимогам до функціоналу програмних додатків.

2.4.2 Вимоги до ОС

1. Функціональні вимоги

1.1. Програмна сумісність: операційна система повинна мати можливість запуску та роботи зі спеціалізованим програмним забезпеченням, необхідним для роботи фахівця з кібербезпеки.

1.2. Апаратна сумісність: Операційна система повинна бути сумісна з широким спектром апаратних пристроїв та програмного забезпечення, що забезпечує безперебійну роботу та високу ступінь сумісності з програмним забезпеченням фахівців з кібербезпеки.

1.3. Робота з даними: операційна система повинна достатньо надійно здійснювати обробку та зберігання інформації, а також підтримувати збереження резервних копій даних та можливість відновлення даних з резервних копій.

1.4. Мережеві функції: ОС повинна надавати користувачеві можливості під’єднання до провідних та безпровідних мереж. Повинна бути підтримка мережевих протоколів та можливість налаштування мережі.

1.5. Безпека: Операційна система повинна мати інструменти забезпечення достатнього рівня безпеки різними засобами, такими як шифрування файлів, налаштування безпеки мережі, перевірка цілісності файлів, контроль прав доступу тощо. Серед яких можна виділити:

- ❖ Аутентифікація: підтримка засобів аутентифікації.
- ❖ Шифрування: підтримка шифрування файлів та папок з можливістю вибору алгоритму шифрування.
- ❖ Розмежування доступу: можливість налаштування прав доступу до файлів та папок.

- ❖ Мережева безпека: встановлення файрволу з можливістю налаштування правил доступу до мережевих ресурсів.
- ❖ Активний захист: підтримка антивірусного захисту із забезпеченням постійного оновлення баз даних.
- ❖ Інструментарій для сканування вразливостей.
- ❖ LSM – Linux Security Modules.

1.6. Модель оновлень: Rolling-Release без необхідності компіляції на стороні користувача.

2. Нефункціональні вимоги

2.1. Об’єм дистрибутиву: не більше 8-10 ГБ – для того, щоб було ще 6-8 ГБ для програмних додатків та іншого ПЗ, що в сумі становить 16 ГБ пам’яті, що є одним з найпопулярніших об’ємів пам’яті USB Flash-носіів.

2.2. Інтерфейс: Операційна система повинна мати як мінімум базовий, незаплутаний, зручний та інтуїтивно зрозумілий користувацький інтерфейс в першу чергу для збільшення продуктивності користувача, в тому числі для спрощення використання інструментів кібербезпеки та налаштування системи.

2.3. Продуктивність: Операційна система повинна працювати швидко та ефективно, забезпечуючи при цьому високий рівень функціональності та стабільності роботи.

2.4. Підтримка: Операційна система повинна забезпечувати довгострокову підтримку та регулярні оновлення для забезпечення безпеки та актуальності системи.

2.5. Орієнтація: Операційна система має бути загального призначення та орієнтована на десктопи або робочі станції, але не на сервери. До того ж, вона повинна бути функціональною та зручною для використання, не потребувати багато часу на налаштування з нуля та постійну підтримку. Користувач не повинен постійно налагоджувати систему після

встановлення неперевірених оновлень, натомість система та/або її розробники повинні дбати про стабільність.

2.5 Розробка концептуальної моделі дистрибутиву

2.5.1 Категорії прикладного ПЗ

- Інструменти соціальної інженерії: маніпуляція, переконання, вплив, введення в оману, фішинг.
- Інструменти експлуатації: хакерство, шкідливе програмне забезпечення, набори експлойтів, програми-вимагачі, руткіти.
- Реверс-інжиніринг: декомпіляція, дизасемблювання, налагодження, аналіз коду.
- Аналіз та пошук вразливостей: сканування, тестування на проникнення, оцінка ризиків, моделювання загроз, управління вразливостями.
- Аналіз веб-додатків: сканери вразливостей, нечітке тестування, аналіз вихідного коду, брандмауери веб-додатків, динамічний аналіз.
- Сніфінг та спуфінг: перехоплення пакетів, ARP-спуфінг, DNS-спуфінг, мережеве сканування, сніффінг Wi-Fi.
- Оцінка баз даних: SQL-ін'єкції, відбитки баз даних, вилучення даних, сканування вразливостей баз даних, оптимізація запитів.
- Бездротові атаки: Злам Wi-Fi, несанкціоновані точки доступу, злом Bluetooth, прослуховування бездротових мереж, глушіння.
- Системні сервіси: управління демонами, моніторинг процесів, резервне копіювання системи, відновлення системи, аналіз системних журналів.
- Збір інформації: OSINT, розвідка, стеження, аналіз соціальних мереж, сканування портів.
- Атаки на паролі: перебір, атаки за словниками, райдужні таблиці, злам паролів, кейлоггери.

- Захист системи та користувача, анонімність, робота з даними: розмежування доступу, контроль трафіку, інструменти приховування, резервне копіювання, архівація та шифрування.
- Мультимедіа: редагування зображень, відеомонтаж, аудіомонтаж, анімація, 3D-моделювання.
- Розробка: мови програмування, дизайн програмного забезпечення, тестування програмного забезпечення, контроль версій, гнучка розробка.
- Інтернет: перегляд веб-сторінок, поштові клієнти, соціальні мережі, онлайн-співпраця, обмін файлами.
- Офісне ПЗ: обробка текстів, електронні таблиці, програми для створення презентацій, поштові клієнти, управління проектами.

2.5.2 Утвердження додатків за визначеними категоріями

1. *Обов'язкове ПЗ:*

1.1. Спеціалізоване ПЗ для вивчення інформаційної безпеки:

1.1.1. Фаєрволи:

- iptables або firewalld, як стандартне та базове рішення.
- pfSense (як додаткове рішення) – відкрите рішення для фаєрволів та роутерів у вигляді окремої ОС.
- OpenWrt (для поглибленого вивчення) – Операційна система OpenWrt дозволяє налаштувати мережу та забезпечити безпеку на роутерах.

1.1.2. Анонімні мережі:

- Tor
- Захищені браузери:
- Brave – гнучкий, орієнтований на приватність браузер, що намагається передавати якомога менше інформації про користувача та підтримувати його анонімність. Дозволяє блокувати

скрипти, трекери та cookies. Має вбудований захист від знімка відбитків. Використовує мережу Tor в приватних вкладках.

1.1.3. Програми для шифрування даних:

- GNUPG, також відомий як GNU Privacy Guard, є вільним ПЗ для шифрування та підпису електронної пошти та інших форматів даних. Він дає можливість створювати цифрові підписи, шифрувати та розшифровувати файли та повідомлення з використанням стандартів шифрування
- LUKS: дані можна безкоштовно і зручно захистити засобом LUKS, що функціонує на Linux і багатьох інших ОС.

1.1.4. Інструменти для аудиту безпеки:

- Nmap (збір інформації): інструмент Nmap сканує мережі, знаходить вразливості в системах і пристроях, які зловмисники можуть використовувати для атаки мережі або отримання конфіденційної інформації.
- SQLmap (збір інформації, оцінка баз даних, тестування та експлуатація вразливостей): даний інструмент з відкритим кодом надає широкі можливості для тестування на проникнення, що дозволяє автоматизувати процес пошуку та експлуатації вразливостей SQL-серверів типу “SQL-ін’єкція”. Підтримує велику кількість СУБД, 6 технік SQL-ін’єкцій та багато іншого.
- Metasploit (експлойтування): Metasploit є програмою для тестування на проникнення, яка використовує різноманітні експлоіти і патчі для виявлення слабких місць у програмному забезпеченні та доступу до системи віддалено.
- OpenVAS (як додаткове потужне рішення для збору інформації, пошуку мережевих вразливостей та аналізу веб-додатків): безкоштовна система сканування вразливостей OpenVAS

допомагає виявляти проблеми безпеки в комп'ютерних системах і мережах.

- Nikto (аналіз веб-додатків, пошук вразливостей, аналіз баз даних): це відкрите програмне забезпечення для виявлення вразливостей веб-серверів і автоматичного сканування безпеки веб-додатків. З допомогою Nikto можна виявити різні вразливості веб-сайтів, такі як SQL-ін'єкції, крос-сайт скриптінг та інші атаки, що можуть бути використані для зламу систем.
- Hydra (атаки на паролі). Hydra є інструментом для тестування на проникнення, який дозволяє автоматизувати атаки на системи автентифікації та парольні атаки на SSH, FTP, Telnet, HTTP тощо.
- Aircrack-ng (бездротові атаки, перевірка безпеки мереж): набір інструментів для вивчення даних про бездротові мережі.
- Suricata (безпека мереж): безкоштовна система моніторингу безпеки мереж, IDS та IPS, яка використовується для моніторингу мереж і виявлення потенційно шкідливої активності.
- Wireshark (сніфінг, збір інформації): Wireshark розроблений для аналізу мережевого трафіку.

1.1.5. Соціальна інженерія:

- Social Engineering Toolkit – фреймворк для тестування на проникнення з використанням соціальної інженерії. Дозволяє проводити веб-атаки, фішингові атаки, масову атаку поштою, а також створювати корисне навантаження (шкідливе програмне забезпечення).

1.1.6. Реверс-інжиніринг:

- Edb-debugger – дебагер з графічним інтерфейсом для бінарних файлів Linux.

- diStorm3 – легкий, швидкий та зручний інструмент дизасемблювання інструкцій для архітектур x86/AMD64.
- Ghidra – потужний фреймворк для зворотнього інжинірингу, що було розроблено NSA.
- JD-GUI – декомпілятор Java коду. Він бере файли .class Java та перебудовує вихідний код, що їх створив. Хоча це не надає ідеальної реконструкції, це надає достатньо зручні умови для аналізу програм Java.
- Apktool – інструмент реверс інжинірингу Android програм. Він дозволяє декодувати ресурси до майже оригінальної форми та відтворити їх після деяких налаштувань. Також дозволяє виконувати пошаговий дебагінг коду.

1.1.7. Програми для знищення даних:

- Secure Delete: За допомогою Secure Delete ви можете безпечно видаляти файли і папки з комп'ютера за допомогою простого виконуваного файлу. Цей інструмент є безкоштовним і простим у використанні, його можна отримати з різних джерел в Інтернеті
- DBAN - Darik's Boot and Nuke - дозволяє безпечно очистити дані з жорстких дисків, SSD і USB-накопичувачів. Допоможе тим, у кого зберігаються чутливі дані на пристроях при продажу або поверненні. Це ISO-образ, що монтується на завантажувальну флешку. На операційну систему можна встановити утиліту nwipe, що є форком nwipe, яка міститься в ПЗ DBAN.

1.1.8. Віртуальні приватні мережі (VPN):

- OpenVPN — це система віртуальної приватної мережі (VPN), яка реалізує методи створення безпечних з’єднань «точка-точка» або «сайт-сайт» у маршрутизованих або мостових конфігураціях і засоби віддаленого доступу. Він реалізує як клієнтські, так і серверні програми.

- Програми для резервного копіювання даних:
- Rsync, як стандартне базове рішення для всіх основних потреб
- Rclone (для хмар): Rclone: є вільним програмним забезпеченням, яке дозволяє керувати файлами в різних хмарних сервісах і мережевих сховищах даних, таких як Google Drive, Dropbox, Amazon S3, OneDrive та інші. Користувачам дозволяється виконувати багато операцій з Rclone, такі як копіювання, переміщення, синхронізація та шифрування файлів та каталогів
- Amanda (аналог Rclone): За допомогою Amanda можна безкоштовно зберігати резервні копії на локальних та віддалених серверах. Цей інтерфейс забезпечує зручну настройку та планування процесу резервного копіювання.
- Restic (базове рішення для всіх потреб) – якщо потрібна сумісність з Windows, BSD та Mac OS.

1.1.9. Плагіни та розширення браузера

- HTTPS Everywhere - це безкоштовне розширення для браузерів, яке забезпечує перехід на зашифрований протокол HTTPS на сайтах з підтримкою HTTPS.
- uBlock Origin: це безкоштовне розширення для браузерів, яке дозволяє блокувати рекламу та шкідливі скрипти на веб-сайтах.
- NoScript: це безкоштовне програмне забезпечення для веб-браузерів, яке дає можливість користувачам контролювати виконання JavaScript, Java, Flash та інших вбудованих сценаріїв на веб-сторінках.

1.2. *ПЗ для виконання повсякденних задач:*

1.2.1. Офісні пакети:

- LibreOffice – для слідування політиці відкритого ПЗ та форматів, використання файлів формату ODF – Open Document

Format, що не є пропрієтарним, хоча є підтримка і форматів Microsoft Office з деякими нюансами.

- OnlyOffice, як безкоштовна та відкрита альтернатива Microsoft Office з високою сумісністю. Команда розробників невелика. Оновлення виходять рідко, деякого функціоналу не вистачає. Проте це є найкращою вільною альтернативою Microsoft Office.

1.2.2. Браузер:

- Firefox (як популярний браузер з відкритим кодом і широкою базою розширень для нього)
- Waterfox (як альтернатива, що є форком з повністю відкритим кодом та відсутнім непотрібним функціоналом)
- Midori (як легковажне рішення): маючи відкриту кодову базу, Midori - це швидкий і легкий веб-браузер

1.2.3. Менеджери паролів:

- KeePassXC: Дозвольте KeePassXC безпечно зберігати ваші паролі та автоматично заповнювати їх у ваших улюблених програмах, щоб ви могли забути про них.

2. *Додаткове ПЗ*

2.1. Антивірусне ПЗ:

- ClamAV - безкоштовний антивірус для ОС Linux, FreeBSD, OpenBSD, NetBSD, Solaris, macOS та Windows.

Поштовий клієнт:

- Thunderbird. Thunderbird є безкоштовним email-клієнтом з крос-платформовою підтримкою, який дозволяє надсилати та отримувати електронні листи, керувати контактами та календарем. Thunderbird - це крос-платформовий email-клієнт з численними перевагами.

2.5.3 Визначення потреб та вимог до розробленого дистрибутиву

Основні потреби як до базового дистрибутиву, так і до дистрибутиву, що буде результатом розробки проекту, є наступними:

- Доступність: Дистрибутив повинен бути легко доступним, наприклад, шляхом завантаження з Інтернету або з фізичного носія.
- Надійність: Дистрибутив повинен бути надійним, тобто забезпечувати правильну та стабільну роботу програмного забезпечення.
- Безпека: Дистрибутив повинен мати постійні оновлення безпеки, а також повинна бути доступна можливість встановлювати та налаштовувати додаткові заходи безпеки.
- Сумісність: Дистрибутив повинен підтримувати сучасне обладнання для персональних комп’ютерів.
- Документація: Дистрибутив повинен мати достатньо документації, щоб користувачі могли зрозуміти, як встановлювати та використовувати програмне забезпечення, що входить у дистрибутив, а також могли проводити дослідницьку роботу.
- Доступне налаштування: Дистрибутив повинен бути доступним та зрозумілим для налаштування, щоб користувачі могли швидко встановити та налаштувати програмне забезпечення на своєму комп'ютері.
- Підтримка: Дистрибутив повинен мати підтримку та оновлення для програмного забезпечення, яке входить у нього, щоб забезпечити безперебійну роботу та захист від загроз безпеки.
- Відповідність вимогам: Дистрибутив повинен відповідати всім сформованим вимогам до ПЗ.

2.5.4 Практичні висновки з потреб та вимог до ОС

Надійність та безпека: мінімум – базування на дистрибутиві, розробник та підтримувач якого достатньо якісно виконує функції контролю якості, що включає як стабільність, так і безпеку ПЗ. Додатково – власна автоматична (та/або автоматизована) система перевірки якості дистрибутиву або ПЗ на стабільність (та безпеку, якщо можливо).

Сумісність: потрібно визначити список архітектур (ISA – Instruction Set Architecture), які повинен підтримувати дистрибутив. Мінімум – x86-64 (amd64), яка використовується в більшості сучасних ПК. Додатково – будь-які інші архітектури, в тому числі, бажано, щоб підтримувалась ARM архітектура, на якій базовано процесори Apple.

Підтримка: Найкращим варіантом буде дистрибутив, що оновлюється за моделлю Rolling-release. Переваги такого рішення:

- Ніколи не застаріває: Оновлення надходять безперервно, що забезпечує операційну систему оновленою і безпечною. Це включає в себе і підтримку новіших драйверів, що забезпечує кращу та ширшу сумісність з комп’ютерним обладнанням.
- Покращення без збоїв: Оновлення можуть бути надані частково, тобто один пакет програмного забезпечення може бути оновлений без залежності від інших, що знижує ризик непередбачуваних збоїв.
- Більша свобода: Модель rolling release дозволяє користувачам обирати, коли і як оновлювати свою операційну систему.
- Спрощене управління залежностями: У моделі rolling release немає необхідності вручну вирішувати залежності між пакетами програмного забезпечення, оскільки це робиться автоматично.
- Більша підтримка: Багато розробників віддають перевагу моделі rolling release, тому це може означати, що операційна система отримує більше уваги від розробників та має кращу підтримку.

2.6 Дослідницька робота з підбору базового дистрибутиву

2.6.1 Критерії до базового дистрибутиву

В ході даної роботи була розглянута велика кількість дистрибутивів Linux. Важливо зазначити, що за потреби, більшість дистрибутивів підходить для виконання поставлених цілей, оскільки їх завжди можна модифікувати бажаним чином. Проте, завжди стоїть питання доцільності витрати ресурсів на те, що може вже бути представлено у якомусь дистрибутиві. Таким чином, для вибору бази для цього проекту було проведено дослідження, в рамках якого були оцінені різні дистрибутиви на предмет їх різних основних компонентів та характеристик, таких як система управління пакетами, механізми відкату системи та оновлень, рівень підтримки, механізми управління системою та безпекою тощо.

Основні критерії, які роздивлялись при виборі ОС:

- Призначення: десктоп, розробка ПЗ, пентестинг, системне адміністрування, аналіз ПЗ.
- Сумісність: основні сучасні мікроархітектури для сучасних комп'ютерів та ноутбуків, що підтримують архітектуру набору команд x86-64.
- Надійність: потрібна достатня надійність системи, або можливість власноруч забезпечити її у плані оновлень та роботи під час виконання основних функцій.
- Підтримка та оновлення: дистрибутив повинен отримувати оновлення без великих затримок, а також ці оновлення повинні бути достатньо стабільними.
- Величина спільноти: бажано, щоб у дистрибутиву була достатньо велика спільнота, аби забезпечити зворотній зв'язок з розробниками для полагодження будь-яких проблем з дистрибутивом, а також широку інформаційну базу з питань щодо дистрибутиву.
- Доступність документації: документація повинна бути достатньо місткою та об'ємною.
- Доступна, але розширена установка та налаштування: дистрибутив повинен надавати доступний, проте місткий механізм встановлення (або має бути

можливість власноруч зробити такий механізм), а також користувач повинен не обмежуватися у налаштуваннях.

- Рівень знань користувачів: дистрибутив не має бути спрямований лише на професійних користувачів Linux, аби студент, що переходитиме на цю операційну систему, мав змогу відносно легко інтегруватися у систему, але одночасно система не має бути орієнтована на новачків.
- Модель оновлень (релізів): Rolling-Release. Ця модель має велику кількість переваг над класичною моделлю релізів, серед яких:
    - Відсутність версій ОС, оскільки оновлення до компонентів надходять поступово, що дає можливість отримувати їх вчасно, що веде до відсутності накопичувального ефекту оновлень, під час впровадження яких можливі технічні проблеми, а також нуліфікує потребу в перевстановленні системи під час виходу наступної “мажорної” версії системи.
    - Стабільність: хоча це залежить від особливостей дистрибутиву, в цілому відсутність накопичувальних оновлень веде до більшої стабільності системи через менший ризик конфліктів та помилок під час оновлення системи.
    - Актуальність: користуючись дистрибутивом, що оновлюється за моделлю Rolling Release, у вас завжди буде доступ до найсвіжіших версій ПЗ без потреби в ручній компіляції коду, отже проблема пакетів, створених для специфічної версії ОС відсутня. Це також стосується драйверів для комп’ютерних компонентів.

2.6.2 Порівняння дистрибутивів

Серед розглянутих дистрибутивів є 6 незалежних та 3 похідних від Arch (Manjaro, Artix Linux, Endeavour OS). Точніше, OpenSUSE Tumbleweed є похідним від SUSE, але фактично він розробляється як незалежний.

Тому спочатку розглянемо основні проблеми та недоліки Arch Linux та дистрибутивів, базованих на Arch:

- Нестабільність базового дистрибутиву: через політику “найшвидших оновлень перед усе”, оновлення досить часто призводять до помилок та проблем із сумісністю. Похідні дистрибутиви, які намагаються компенсувати цей аспект, ризикують мати проблеми із залежностями при використанні репозиторіїв Arch.
- Складність налаштування та підтримки: Arch Linux призначений для досвідчених користувачів Linux, які мають досвід у роботі з командним рядком та налаштуванні системи вручну.
- Складність та нестабільність Arch та дистрибутивів, базованих на Arch, часто вимагає втручання досвіченого користувача, або довгого ручного пошуку рішення проблем. До того ж, відкат системи та ПЗ часто є достатньо складним та незручним процесом у цих дистрибутивах. Критичні проблеми після оновлень для багатьох користувачів будуть означати необхідність перевстановлення системи.
- Низький рівень абстракції: Arch Linux надає користувачеві прямий доступ до системи, що може бути складним для користувачів, які шукають високорівневий інтерфейс та зручність в користуванні.
- Високий рівень вимог до часу: Установка та налаштування Arch Linux може зайняти більше часу, ніж установка інших дистрибутивів.
- Слабкий функціонал пакетного менеджеру pacman: хоча він вважається одним із найбільш швидких, це через те, що він майже не несе ніякого додаткового функціоналу, що міг би бути спрямований на забезпечення більшої стабільності системи, або на кращі можливості відкату або виправлення помилок із залежностями. До того ж, в багатьох дистрибутивах немає стандартних налаштувань очистки кешу pacman, що досить швидко призведе до засмітнення системи.
- Накопичення оновлень у тривалий проміжок часу майже гарантовано буде означати проблемний процес оновлення системи. Таким чином, користувачеві необхідно достатньо часто оновлювати систему, щоб вона не зламалась.

- Відсутність підтримки для користувачів похідних від Arch дистрибутивів на форумах Arch: там заборонені питання щодо цих дистрибутивів, згадка про те, що ви користуєтесь похідним дистрибутивом, може призвести до бану. Це є проблемою тому, що часто користувачі похідних дистрибутивів звертаються до документації та офіційного форуму Arch, оскільки аналоги для їх дистрибутивів є не такими повноцінними та популярними. Таким чином, користувачам похідних дистрибутивів часто буде складніше знайти швидку відповідь на їх запитання [33].

Manjaro спрямований на переорієнтацію Arch на графічний інтерфейс та простоту використання. Таким чином, дистрибутив обрав шлях дружності до користувача, а таким чином і обмежень його доступу до глибоких налаштувань системи.

Цей дистрибутив можуть рекомендувати початківцям, проте як тільки користувачі почнуть поглиблюватись у Linux, можуть з’являтися проблеми, які було б легше вирішити на Arch Linux, або яких би там взагалі не було. Так чи інакше, Manjaro не сильно турбується про безпеку та стабільність системи, а в порівнянні з Arch здебільшого лише спрощує базове користування системою.

Також слід зазначити, що базова інсталяція Manjaro містить значну кількість передвстановленого ПЗ, що може бути небажаним або просто непотрібним.

Endeavour OS, з іншого боку, не так радикально намагається перетягнути Arch на сторону графічного інтерфейсу, і взагалі не дуже модифікує Arch, проте значно покращує досвід використання системи одразу після встановлення. Так, у дистрибутиві зверху базового Arch додали декілька власних інструментів (інсталятор Calamares, Welcome App), а також найважливіших для початку роботи з ОС, серед яких браузер Firefox, Yay – допоміжник для користування репозиторієм AUR, фаєрвол FirewallD, інсталятор Nvidia, утиліта Downgrade, що корисна для відкату до попередніх версій пакетів, та декілька інших. Також розробники Endeavour OS не втручаються в процес оновлення та не відстроковують його для перевірки пакетів на стабільність. Все це разом робить дистрибутив максимально близьким до Arch, проте більш зручним та менш потребуючим часу на базове налаштування [34].

Artix Linux можна просто охарактеризувати як “Arch Linux з вирізаним systemd”. Це дистрибутив для тих, хто не любить систему ініціалізації та управління службами systemd, проте хоче користуватись усіма іншими перевагами Arch Linux, або кому потрібна легша та більш ефективна ОС. Для більшості користувачів цей дистрибутив не надасть ніяких переваг, окрім трохи швидшого запуску та роботи системи. Натомість дистрибутив привносить незручності, такі як проблеми з сумісністю усього ПЗ з репозиторіїв Arch, яке залежить від systemd, як і взагалі початкова відсутність деяких репозиторіїв у pacman, що вимагає деякого налаштування, а також гірша документація.

Як висновок, можна сказати наступне: не важливо, користуєтесь ви Arch Linux або похідним дистрибутивом, час від часу потрібно буде власноруч щось налаштовувати та виправляти. Дистрибутиви, які адресують проблеми, які вкладені в архітектуру Arch, зазвичай їх повноцінно не виправляють, або мають якісь інші нюанси. І як би себе не позиціонував дистрибутив, базований на Arch, повноцінно орієнтованим на стабільність або на зручність користувача він не буде, і недосвіченому Linux-користувачеві іноді буде складно підтримувати систему, або просто незручно. Це дистрибутив для бувалих користувачів Linux з достатніми знаннями, які хочуть мати повністю гнучку систему для своїх потреб, або для тих, хто хоче на практиці ці знання здобути.

Перейдемо до незалежних дистрибутивів: Gentoo, Void Linux, NixOS, openSUSE Tumbleweed, Solus.

Gentoo: Це дистрибутив, створений для дуже досвідчених користувачів, або навіть експертних, яким може знадобитись максимальна оптимізація ПЗ під конкретне апаратне забезпечення та під конкретні потреби. Весь дистрибутив базується на компіляції коду на стороні користувача. Цитуючи офіційний сайт, можна описати можливості дистрибутиву: “Gentoo може стати ідеальним захищеним сервером, робочою станцією для розробки, професійним настільним комп’ютером, ігровою системою, вбудованим рішенням або чимось іншим – усім, що захочете”. Система потребує великих витрат часу на підтримку, в першу чергу через потребу в перекомпіляції коду під час оновлень, тому користувачі рекомендують робити це не

частіше ніж раз у тиждень, бажано тоді, коли не потрібно використовувати систему. Деякі пакети також надаються у формі бінарних файлів, що зручно для випадків, де компіляція коду займає багато часу, наприклад Firefox або LibreOffice, але це забирає можливість оптимізації.

Таким чином, Gentoo є складною у підтримці та налаштуванні системою, що націлена на повну оптимізацію для конкретних цілей, що потребує значних витрат часу. Для більшості користувачів, які хочуть використовувати цей дистрибутив на персональних комп'ютерах, ці оптимізації будуть досить незначними для продуктивності системи через відносно надлишкову потужність сучасного апаратного забезпечення. Також процес відкату оновлень є досить складним, або просто потребує значних витрат часу.

Nix OS: Багато користувачів називають цей дистрибутив логічним продовженням Gentoo в плані управління пакетами та ідеології системи. Проте краще охарактеризувати ідею дистрибутиву як "система у файлах конфігурації" – цей дистрибутив відомий за його переносимість та відтворюваність: файл конфігурації " /etc/nixos/configuration.nix" є, по суті, визначенням вашої інсталяції ОС.

Nix OS позиціонує себе як надійний дистрибутив, і не просто так: по-перше, це immutable система, що означає дуже високу стабільність системи, а також надає можливості відкату або просто використання іншого зображення системи. По-друге, потужний пакетний менеджер Nix робить процес обробки пакетів повністю передбачуваним та надійним завдяки функціональному підходу та концепції транзакцій – при непередбачуваних інцидентах під час оновлення, система повернеться до стабільного стану. До того ж, Nix гарантує, що оновлення одного пакету не зламає інший [35].

Також слід зазначити, що на відміну від Gentoo, у Nix є можливість користуватися онлайн-кешем зкомпільованих пакетів, якщо користувач не бажає компілювати код на своїй системі.

Отже, Nix OS є дуже потужним дистрибутивом для досвідчених користувачів, проте потребує витрат часу на вивчення власної мови Nix та налаштування. Цей дистрибутив може подарувати дуже зручну екосистему для користувачів багатьох

пристроїв, на яких встановлена ця ОС, або для будь-яких інших сфер, де потрібна переносимість та відтворюваність. Методи управління пакетами та конфігурацією системи у Nix OS є передовими, що є великою перевагою над іншими дистрибутивами.

Void Linux: Це мінімальний дистрибутив по типу Arch, проте без його AUR та Systemd. Він надає усю базу, що вам потрібна, для створення та налаштування ОС під свої потреби. Цей дистрибутив є відомим за свою можливість вибору між бібліотеками мови С: glibc та musl. Пакетний менеджер дистрибутиву є написаним з нуля, та відзначається зручністю та швидкістю. Офіційний репозиторій містить достатньо пакетів, але є і репозиторій спільноти для більш широкого вибору. Void також відомий за зручність розробникам робити свій вклад у дистрибутив.

Отже, Void Linux є легким та швидким дистрибутивом, який надає зручний інструментарій для досвідчених користувачів та розробників. Багато користувачів звітували також про його надійність і стабільність, проте потрібно пам’ятати, що цільова аудиторія цього дистрибутиву – це досить досвідчені користувачі.

OpenSUSE Tumbleweed: Дистрибутив який розробляється паралельно з пропрієтарною SUSE Linux Enterprise. За довгий час свого існування цей дистрибутив заробив репутацію дуже одного із, якщо не найбільш найдійного дистрибутиву, що оновлюється за моделлю Rolling Release. Цей дистрибутив є зручним як для звичайних, так і для досвідчених користувачів. Нахил в сторону графічного інтерфейсу дає можливість швидко та інтуітивно налаштувати ОС, а велика кількість інструментів гарантує повноту можливостей для адаптації під свої потреби. OpenSUSE та SUSE Linux Enterprise розробляються окремо, проте напрацювання часто переходять від однієї до іншої, що в результаті дає глибоко розроблену та відшліфовану ОС.

Серед заходів безпеки, за замовчуванням openSUSE Tumbleweed має достатньо жорсткі налаштування фаєрволу та встановлений модуль безпеки AppArmor, що надає широкі можливоті контролю над кожною програмою.

Цей дистрибутив також достатньо стабільний, щоб оновлюватися лише раз на кілька тижнів, або навіть значно рідше. Незручністю буде великий об’єм оновлень, проте це є ціною за високу надійність та свіжість ПЗ.

Серед інших переваг є широка підтримка різних середовищ робочого столу, надійна файлова система за замовчуванням BTRFS з вбудованим функціоналом снапшотів системи, автоматизоване тестування оновлень перед їх випуском, як у Nix OS, проте openSUSE має ширшу інфраструктуру для цього, а також душе широкий репозиторій пакетів.

Незручностю буде використання пропрієтарних кодеків та драйверів, які за філософією openSUSE не встановлюються за замовчуванням, тому це потрібно буде робити власноруч, а також потреба завантажувати більш об’ємні оновлення, ніж, наприклад, у Arch.

Найбільше цей дистрибутив підходить для системних адміністраторів та розробників ПЗ, проте це не означає, що він буде незручним для звичайних користувачів.

Solus: Це дистрибутив Linux, який був розроблений з метою забезпечити користувачам простий та ефективний досвід використання операційної системи. Його часто називають “Windows від світу Linux”, проте лише в контексті його зручності для користувача. Solus отримав значну популярність в середовищі Linux-користувачів та розробників, завдяки своєму користувальницькому інтерфейсу та інтегрованим власним рішенням, які зробили його особливим дистрибутивом.

Solus постачається з великою кількістю інтегрованих інструментів, що дозволяє користувачам працювати з різними форматами файлів та різними додатками без додаткових встановлень. Серед передвстановлених програм є: файловий менеджер Files, браузер Firefox, медіаплеєри GNOME MPV та Rhythmbox, а також менеджер програм Software Center.

Цей дистрибутив здебільшого націлений на звичайних користувачів персональних комп’ютерів, або на тих, хто працює з операційною системою, як з базою для встановлення інших програм, але не для глибоких налаштувань та

кастомізації. Також, оскільки це достатньо новий дистрибутив, у його репозиторії немає так багато пакетів, як у інших дистрибутивів.

Таким чином, щоб підсумувати всю інформацію, підведемо рейтинг дистрибутивів за шкалою “від зручного, графічно орієнтованого та налаштованого дистрибутиву, до дистрибутиву для power-користувачів”:

1. Solus
2. Manjaro
3. OpenSUSE Tumbleweed
4. Endeavour OS
5. Arch / Artix
6. Void Linux
7. Gentoo, NixOS

2.7 Визначення необхідних та бажаних компонентів і характеристик дистрибутиву

Серед необхідних компонентів та характеристик для даного дистрибутиву було вирішено виділити наступні:

- Система ініціалізації та управління службами Systemd – бажано та рекомендовано мати, щоб уникнути ризику несумісності ПЗ через його залежність від цієї системи.
- Інструменти для відновлення та/або відкату системи, а також для підтримки її стабільності: як для відкату до попередньої версії окремих пакетів, блокування для завантаження непотрібних версій пакетів, так і для відкату до попередньої версії ядра, або до попереднього снапшоту системи.
- Підтримка контейнеризації та віртуалізації.
- Працездатна та зручна графічна оболонка.
- Підтримка або можливість використання пропрієтарних драйверів для апаратного забезпечення.

- Можливості користувача гнучко налаштовувати та адмініструвати системою.
- Інтегроване рішення безпеки для мандатного управління доступом за допомогою Linux Security Modules типу AppArmor або SELinux.
- Модель оновлень: Rolling-Release та її варіації. Для ОС загального призначення (навіть спеціалізованої), важливу роль грає зручність. Проблеми ОС, що розповсюджуються за моделлю Point-Release лежать в тому, що після виходу нової “мажорної”, або глобальної версії ОС, для користувачів, що не оновились, по-перше, програмні додатки не отримують глобальних оновлень фукнціоналу, а все ПЗ (включаючи систему та додатки) гарантовано лише отримують оновлення безпеки, які виходять відтерміновано через технічні причини, а по-друге, однією з головних проблем переходу на нову “мажорну” версію є, як правило, необхідність перевстановлення ОС та відсутність сумісності деякого ПЗ з новою ОС. Таким чином, використовуючи ОС з моделлю оновлень Rolling, користувач позбавлюється цих незручностей, а також, як правило, отримує як оновлення функціоналу, так і безпеки ПЗ раніше, ніж інші.

2.8 Утвердження базової ОС

З огляду на всю попередньо викладену інформацію, в тому числі вимоги та аналіз необхідних та бажаних характеристик ОС, було вирішено обрати openSUSE Tumbleweed, як базовий дистрибутив, оскільки вона найкраще підходить для цілі створення ОС загального призначення спеціаліста з ІБ, зокрема за наступних причин:

Стабільність: openSUSE Tumbleweed має дуже сильну репутацію серед Rolling-Release дистрибутивів, при чому багато користувачів звітують про кращу стабільність, ніж у деяких Point-Release дистрибутивів (в першу чергу в контексті оновлень).

Безпека: модель розробки дистрибутивів openSUSE – спільнота з підтримкою від компанії SUSE, а вона, у свою чергу, розробляє та продає різні рішення для

бізнесу. Ця компанія дуже серйозно ставиться до безпеки, і це видно у всіх дистрибутивах openSUSE. Існує теорія, що в сучасному світі для ОС робочого столу швидкий цикл оновлень може надавати більшу безпеку, тому що чим новіша система, тим менше в ній відомих дір безпеки.

Висновки до розділу 2

В даному розділі було розроблено модель безпеки, вимоги та концептуальну модель для спеціалізованого дистрибутиву Linux, а також визначено цілі та мету дистрибутиву.

В даній частині роботи було також розроблено вимоги до ПЗ, визначено категорії ПЗ для спеціаліста з кібербезпеки, підібрано програмні додатки, визначено необхідні характеристики та компоненти операційної системи, критерії до базового дистрибутиву, виконано дослідницьку роботу з підбору базового дистрибутиву та утверджено базову систему.

РОЗДІЛ 3
КОМПОНУВАННЯ ДИСТРИБУТИВУ ТА ВПРОВАДЖЕННЯ МОДЕЛІ БЕЗПЕКИ

Дана частина роботи стосується:

- Розробки програмного забезпечення відповідно до вимог та концептуальної моделі, тобто формування дистрибутиву.
- Конфігурування та налаштування інфраструктури проекту, тобто налаштування ОС та встановлення ПЗ.
- Документації встановленого ПЗ та виконаних налаштувань.

Фактично ця частина є підсумовуванням виконання вимог, що були сформовані в наступних пунктах розділу 2 даної кваліфікаційної роботи:

- "Розробка вимог до операційної системи та налаштувань".
- "Розробка моделі безпеки".
- "Розробка концептуальної моделі для програмних додатків".
- "Утвердження додатків за визначеними категоріями".
- "Розробка концептуальної моделі для операційної системи".
- "Утвердження базової ОС. Визначення моделі розповсюдження і встановлення дистрибутиву та моделі підтримки".

3.1 Інтегроване програмне забезпечення

- Інструменти соціальної інженерії: Social Engineering Toolkit.
- Інструменти експлуатації: Metasploit.
- Реверс-інжиніринг: Edb-debugger, diStorm3, Ghidra, JD-GUI, Apktool.
- Аналіз та пошук вразливостей: Nikto.
- Аналіз веб-додатків: Nikto.
- Сніфінг та спуфінг: Wireshark, mitmproxy.
- Оцінка баз даних: SQLmap.

- Бездротові атаки: Aircrack-ng.
- Системні сервіси: firewalld, iptables.
- Збір інформації: Nmap, SQLmap, Nikto, Wireshark.
- Атаки на паролі: Hydra.
- Захист системи та користувача, анонімність, робота з даними: iptables, Tor, Brave, GNUPG, LUKS, Suricata, Secure Delete, Nwipe, OpenVPN, Rsync, Rclone, Restic, KeePassXC.
- Аудит безпеки: Lynis, OpenSCAP.
- Мультимедіа: різні мультимедійні (в тому числі пропрієтарні) кодеки з репозиторію Packman.
- Розробка: Python, GCC compiler
- Інтернет: Brave, Firefox
- Офісне ПЗ: LibreOffice, OnlyOffice
- Додаткове ПЗ: Thunderbird

3.2 Теоретичні аспекти використання системи

Даний дистрибутив повинен забезпечувати спеціаліста з кібербезпеки набором інструментів для захисту системи та інформації, для активної перевірки інших систем на проникнення та для виконання звичайних повсякденних задач. Захист системи можливо організувати шляхом впровадження розробленої моделі безпеки за допомогою передвстановлених інструментів.

За допомогою цього дистрибутиву можна виконувати ряд таких задач спеціаліста з кібербезпеки, як: дослідження процесів аудиту безпеки, пентестингу, аналізу програмного забезпечення та реверс-інжинірингу, аналіз вразливостей сервісів, баз даних, веб-додатків та мереж, огранізація анонімності, захист даних, розробка, організація безпеки системи.

Модель безпеки можна організувати за допомогою різних встановлених інструментів, таких як вибіркове керування доступом (DAC) за допомогою списків

контролю доступу (ACL), інтегрованих модулей безпеки Linux (LSM – Linux Security Modules), а саме AppArmor та SELinux, що реалізують мандантне управління доступом та інших. Моніторинг системи можна виконати за допомогою логування, інструментів аудиту типу OpenSCAP або Lynis, системи IDS та IPS Suricata та інших інструментів.

Захист цілісності, конфіденційності та доступності даних можна організувати за допогою: можливостей файлової системи BTRFS, таких як снапшоти (в тому числі read-only) і контрольні суми, шифрування програмами LUKS та GPG, резервного копіювання з використанням Rsync, Rclose або Restic, менеджеру паролів KeePassXC, повного видалення даних Nwipe або Secure Delete.

Перевірку інших систем можна виконувати за допомогою різних встановлених інструментів для пентестингу, таких як Metasploit, Wireshark, Nikto, Nmap, SQLmap, Hydra, Aircrack-ng, Social Engineering Toolkit та mitmproxy.

Забезпечити анонімність допоможуть браузер Brave, рішення для віртуальних мереж OpenVPN, а також за допомогою Tor.

3.3 Впровадження політик безпеки

3.3.1 Аналіз поточного стану безпеки та ризиків. Налаштування безпеки

Аналіз інфраструктури та ризиків: сервіси.

Після запуску системи, можна відкрити консоль та перевірити список сервісів за допомогою команди “sudo service -s”, де параметр “-s” відповідає за “status all”. Там будуть усі сервіси, що є активними, а також такі, що мають статус “exited”, тобто ті, які виконали свою роботу, припинені та не потребують повторного запуску. Для зручності виведемо лише справді активні сервіси у зручному форматі командою “sudo service -s | grep running | nl”, результат відображений на рис. 3.1:

```
admin@localhost:~> sudo service -s | grep running | nl
     1    auditd.service                    loaded active running Security Auditing Service
     2    avahi-daemon.service              loaded active running Avahi mDNS/DNS-SD Stack
     3    chronyd.service                   loaded active running NTP client/server
     4    cron.service                      loaded active running Command Scheduler
     5    cups.service                      loaded active running CUPS Scheduler
     6    dbus.service                      loaded active running D-Bus System Message Bus
     7    display-manager.service           loaded active running X Display Manager
     8    firewalld.service                 loaded active running firewalld - dynamic firewall daemon
     9    getty@tty1.service                loaded active running Getty on tty1
    10    irqbalance.service                loaded active running irqbalance daemon
    11    ModemManager.service              loaded active running Modem Manager
    12    NetworkManager.service            loaded active running Network Manager
    13    nscd.service                      loaded active running Name Service Cache Daemon
    14    polkit.service                    loaded active running Authorization Manager
    15    postfix.service                   loaded active running Postfix Mail Transport Agent
    16    power-profiles-daemon.service     loaded active running Power Profiles daemon
    17    rtkit-daemon.service              loaded active running RealtimeKit Scheduling Policy Service
    18    snap.cups.cups-browsed.service    loaded active running Service for snap application cups.cups-browsed
    19    snap.cups.cupsd.service           loaded active running Service for snap application cups.cupsd
    20    snapd.service                     loaded active running Snap Daemon
    21    sshd.service                      loaded active running OpenSSH Daemon
    22    systemd-journald.service          loaded active running Journal Service
    23    systemd-logind.service            loaded active running User Login Management
    24    systemd-udevd.service             loaded active running Rule-based Manager for Device Events and Files
    25    udisks2.service                   loaded active running Disk Manager
    26    upower.service                    loaded active running Daemon for power management
    27    user@1000.service                 loaded active running User Manager for UID 1000
    28    vboxclient.service                loaded active running VirtualBox guest VMSVGA resize client
```

Рисунок 3.1 – Активні сервіси

За потреби аналогічно можна вивести сервіси зі статусом “exited”, проте такої необхідності немає.

Також можна переглянути сервіси, що включені до автозапуску при завантаженні системи командою “systemctl list-unit-files --type=service | grep enabled” (рис 3.2):

```
admin@localhost:~> systemctl list-unit-files --type=service | grep enabled
apparmor.service                          enabled         enabled
appstream-sync-cache.service              enabled         enabled
auditd.service                            enabled         enabled
avahi-daemon.service                      disabled        enabled
bluetooth.service                         enabled         disabled
chronyd.service                           enabled         disabled
cron.service                              enabled         enabled
cups.service                              masked          enabled
display-manager.service                   enabled         enabled
drkonqi-coredump-processor@.service       enabled         enabled
firewalld.service                         enabled         disabled
getty@.service                            enabled         enabled
irqbalance.service                        enabled         enabled
issue-generator.service                   enabled         enabled
kbdsettings.service                       enabled         enabled
lvm2-monitor.service                      enabled         enabled
mcelog.service                            enabled         enabled
ModemManager.service                      enabled         enabled
NetworkManager-dispatcher.service         enabled         disabled
NetworkManager-wait-online.service        enabled         disabled
NetworkManager.service                    enabled         disabled
nscd.service                              enabled         enabled
postfix.service                           enabled         enabled
purge-kernels.service                     enabled         enabled
smartd.service                            enabled         enabled
snapd.apparmor.service                    enabled         disabled
snapd.service                             enabled         disabled
sshd.service                              enabled         disabled
systemd-pstore.service                    enabled         enabled
systemd-remount-fs.service                enabled-runtime disabled
vgauthd.service                           enabled         enabled
vmblock-fuse.service                      enabled         enabled
vmtoolsd.service                          enabled         enabled
YaST2-Firstboot.service                   enabled         enabled
YaST2-Second-Stage.service                enabled         enabled
```

Рисунок 3.2 – Автозапуск сервісів при завантаженні системи

Серед цих сервісів окремо можна виділити bluetooth, який в більшості випадків не потрібен користувачу, а тому як потенційний ризик його краще вимкнути командами “sudo systemctl stop bluetooth.service” та “sudo systemctl disable bluetooth” (рис. 3.3):

```
admin@localhost:~> sudo systemctl stop bluetooth.service
admin@localhost:~> sudo systemctl disable bluetooth.service
Removed "/etc/systemd/system/dbus-org.bluez.service".
Removed "/etc/systemd/system/bluetooth.target.wants/bluetooth.service".
```

Рисунок 3.3 – Відключення автозавантаження bluetooth

Розглянемо деякі активні сервіси, що можуть нести потенційні загрози безпеці, функціонал яких не є обов’язковим для користувача, тому які можна вимкнути або додатково налаштувати:

- Avahi: дозволяє програмам надавати та виявляти сервіси та хости в локальній мережі, не вимагаючи введення жодних спеціальних налаштувань. За відсутності потреби в функціоналі цього сервісу, його можна вимкнути. Оскільки у цього сервісу є доступ до мережі, він може бути площею для мережевих атак. На сайті “Common Vulnerabilities and Exposures” (загальні вразливості та ризики, CVE’s) можна знайти різні історичні вразливості, наприклад CVE-2017-6519 (DoS атака), деталі якої наведені на рис 3.4:

– CVSS Scores & Vulnerability Types

CVSS Score	**6.4**
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	None (There is no impact to the integrity of the system)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of Service Obtain Information
CWE ID	346

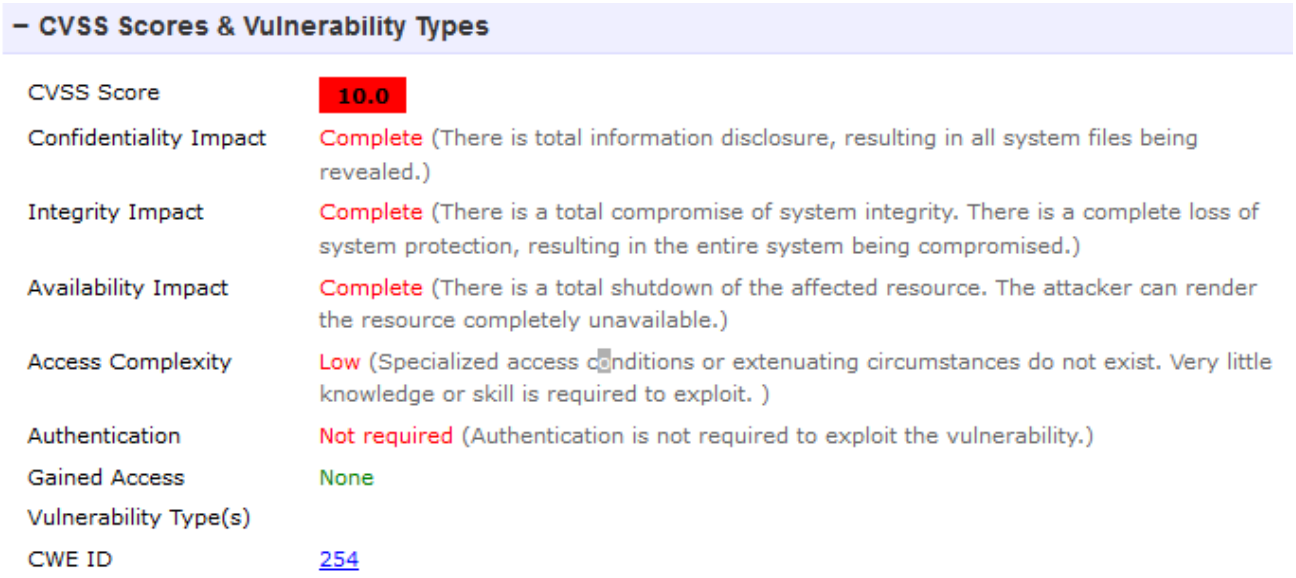
Рисунок 3.4 – Вразливість CVE-2017-6519

- Chronyd: це гнучка реалізація протоколу мережевого часу Network Time Protocol (NTP). Також цей сервіс можна використовувати як сервер NTPv4 для синхронізації часу з іншими серверами у тій же мережі. Серверний функціонал сервісу може нести ризики бути об’єктом DoS-атак з різними наслідками. Як приклад такої загрози можна навести CVE-2015-1853, подробиці на рис. 3.5:

CVSS Score	**4.0**
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	None (There is no impact to the integrity of the system)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	???
Gained Access	None
Vulnerability Type(s)	Denial Of Service
CWE ID	CWE id is not defined for this vulnerability

Рисунок 3.5 – Вразливість CVE-2015-1853

- Cron: це планувальник задач, що слугує для виконання задач у фоновому режимі у вказаний час. Crontab може бути використаний для планування зловмисних задач, наприклад для реінфекції системи. Тому можна налаштувати доступ користувачів до цієї утиліти, а також можна проводити аудит запланованих задач.
- Cups: сервер друку. Комп'ютер із запущеним сервером CUPS є мережевим вузлом, який приймає завдання на друк від клієнтів, обробляє їх і відправляє на відповідний принтер. Як і усі сервіси, що мають доступ до мережі, являє собою ризик. У 2015 було знайдено критичну вразливість CVE-2015-1158 з максимальним рівнем загрози (10), в результаті використання якої можна було виконувати довільний код. Подробиці CVE-2015-1158 наведені на рис. 3.6:



– CVSS Scores & Vulnerability Types

CVSS Score	10.0
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	
CWE ID	254

Рисунок 3.6 – Критична вразливість CVE-2015-1158

Серед інших сервісів можна визначити такі групи:

- Системні сервіси та менеджмент ресурсів: dbus (механізм міжпроцесової комунікації), getty (менеджер терміналів), irqbalance (розподіл апаратних переривань між процесорами та ядрами), ModemManager (управління мобільними модемами), NetworkManager (менеджер мереж), nscd (кеш для найпоширеніших запитів служби імен, таких як passwd, group, hosts), power-profiles-daemon (для обробки профілів живлення системи через D-Bus), rtkit-daemon (планування процесів рівня користувача в реальному часі, тобто розподіл ресурсів), sshd (OpenSSH), systemd-journald (системні журнали), systemd-logind (менеджер входів в систему), systemd-udevd (менеджер пристроїв), udisks2 (управління зберігаючими пристроями), upower (управління, інформація про пристрої живлення).
- Сервіси безпеки: auditd (аудит системи), firewalld (фаєрвол), polkit (засіб управління правами додатків).
- Інші: postfix (поштовий сервер), snapd (управління пакетами Snap), vboxclient (гістьові утиліти VirtualBox).

Управління загрозами: налаштування сервісів.

Після того, як було проаналізовано активні сервіси, потрібно впровадити політики безпеки шляхом налаштувань цих сервісів:

- Відключення Avahi (рис. 3.7):

```
admin@localhost:~> sudo systemctl stop avahi-daemon
[sudo] password for root:
Warning: Stopping avahi-daemon.service, but it can still be activated by:
  avahi-daemon.socket
admin@localhost:~> sudo systemctl disable avahi-daemon
Removed "/etc/systemd/system/sockets.target.wants/avahi-daemon.socket".
Removed "/etc/systemd/system/multi-user.target.wants/avahi-daemon.service".
Removed "/etc/systemd/system/dbus-org.freedesktop.Avahi.service".
```

Рисунок 3.7 – Відключення сервісу Avahi

- Відключення серверного функціоналу Chrony:

Перевіримо, чи визначений порт 0 у файлі “/etc/chrony.conf”. Якщо строки з портом немає, або порт є іншим – тоді потрібно змінити значення на 0, або додати строку “port 0”. Результат перевірки на рис. 3.8:

```
admin@localhost:~> sudo grep -w 'port' /etc/chrony.conf
admin@localhost:~>
```

Рисунок 3.8 – перевірка мережевих налаштувань chrony

Як видно, строки з текстом “port” немає. Відкриємо редактор nano та додамо строку “port 0” (рис. 3.9):

```
# Add sourcedir needed by NetworkManager DHCP dispatcher
sourcedir /run/chrony-dhcp

port 0

                        [ Wrote 55 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
```

Рисунок 3.9: Налаштування chrony

Тепер перевіримо, чи зберіглись зміни (рис. 3.10):

```
admin@localhost:~> sudo grep -w 'port' /etc/chrony.conf
port 0
```

Рисунок 3.10 – Перевірка змін налаштувань chrony

Після цих дій можна перезавантажити службу командою “sudo systemctl restart cronyd” і бути впевненими, що тепер служба працює з встановленими налаштуваннями.

- Налаштування Cron:

Для кращої безпеки, було вирішено обмежити доступ до служби Cron тільки root користувачу. Для цього нам потрібно, щоб існував файл “cron.allow”, який слугує “білим списком” користувачів. Спочатку перевіримо, які файли налаштувань Cron вже існують (рис. 3.11):

```
admin@localhost:~> ls /etc/ | grep cron
cron.d
cron.daily
cron.deny
cron.hourly
cron.monthly
crontab
cron.weekly
```

Рисунок 3.11 – Перевірка файлів налаштувань Cron

Створимо файл “cron.allow” та запишемо в нього строку “root” командою “sudo nano /etc/cron.allow” (рис. 3.12):

Рисунок 3.12 – Налаштування Cron

Для перевірки існуючих запланованих задач можна використати команду “sudo less /etc/crontab” (рис. 3.13):

```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
MAILTO=root
#
# check scripts in cron.hourly, cron.daily, cron.weekly, and cron.monthly

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
@hourly         root      run-parts /etc/cron.hourly
@daily          root      run-parts /etc/cron.daily
@weekly         root      run-parts /etc/cron.weekly
@monthly        root      run-parts /etc/cron.monthly
```

Рисунок 3.13 – Заплановані задачі Cron

Як видно, запланованих задач немає.

- Відключення Cups:

Щоб виключити Cups, достатньо ввести команди “sudo systemctl stop cups” та “sudo systemctl disable cups” (рис. 3.14):

```
admin@localhost:~> sudo systemctl stop cups
admin@localhost:~> sudo systemctl disable cups
Removed "/etc/systemd/system/sockets.target.wants/cups.socket".
Removed "/etc/systemd/system/multi-user.target.wants/cups.service".
Removed "/etc/systemd/system/multi-user.target.wants/cups.path".
Removed "/etc/systemd/system/printer.target.wants/cups.service".
```

Рисунок 3.14 – Відключення Cups

Проте, існують також сервіси snap.cups.cupsd та snap.cups.cups-browsed, що теж можна відключити (рис 3.15):

```
admin@localhost:~> sudo systemctl stop snap.cups.cupsd
admin@localhost:~> sudo systemctl disable snap.cups.cupsd
Removed "/etc/systemd/system/multi-user.target.wants/snap.cups.cupsd.service".
admin@localhost:~> sudo systemctl stop snap.cups.cups-browsed
admin@localhost:~> sudo systemctl disable snap.cups.cups-browsed
Removed "/etc/systemd/system/multi-user.target.wants/snap.cups.cups-browsed.service".
```

Рисунок 3.15 – Відключення Snap-версій Cups

Варто звернути увагу, що ці сервіси можуть бути запущеними іншими сервісами. Якщо немає ризику похитнути роботу системи або якихось додатків, то можна виконати “маскування” сервісу Cups (рис. 3.16):

```
admin@localhost:~> sudo systemctl mask cups
[sudo] password for root:
Created symlink /etc/systemd/system/cups.service → /dev/null.
admin@localhost:~> sudo systemctl mask snap.cups.cupsd
Failed to mask unit: File /etc/systemd/system/snap.cups.cupsd.service already exists.
admin@localhost:~> sudo systemctl mask snap.cups.cups-browsed
Failed to mask unit: File /etc/systemd/system/snap.cups.cups-browsed.service already exists.
```

Рисунок 3.16 – Маскування сервісу Cups

Основний сервіс було успішно замасковано, а інші, як виявилось, уже були замасковані.

Додатково можна відключити такі сервіси, як ModemManager та sshd, функціоналом яких не передбачається. В цьому випадку вирішено було не відключати їх, оскільки ModemManager може бути корисним для використання мобільної мережі для підключення до інтернету, а sshd для використання OpenSSH.

Управління ризиками: налаштування фаєрволу.

Для початку перевіримо статус роботи firewalld, результат показаний на рис. 3.17:

```
admin@localhost:~> systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: disabled)
     Active: active (running) since Sat 2023-06-17 08:47:52 EEST; 5min ago
       Docs: man:firewalld(1)
   Main PID: 1013 (firewalld)
      Tasks: 2 (limit: 4915)
        CPU: 326ms
     CGroup: /system.slice/firewalld.service
             └─1013 /usr/bin/python3 /usr/sbin/firewalld --nofork --nopid
```

Рисунок 3.17 – Статус фаєрволу

Коли ми впевнились, що він працює, почнемо роботу з фаєрволом:

- Виведемо список існуючих зон та дізнаємось, яка з них є за замовчуванням (рис. 3.18):

```
admin@localhost:~> sudo firewall-cmd --get-zones
block dmz docker drop external home internal nm-shared public trusted work
admin@localhost:~> sudo firewall-cmd --get-default-zone
public
```

Рисунок 3.18 – Список зон та зона за замовчуванням

За потреби, можна дізнатись список сервісів, які покриває фаєрвол (рис. 3.19):

```
admin@localhost:~> sudo firewall-cmd --get-services
[sudo] password for root:
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 b
acula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet
 bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-
collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-r
egistry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa
-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gp
sd grafana gre http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins k
admin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-control-plane kube-co
ntrol-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-scheduler
 kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-tls light
ning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh moun
td mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp
nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s
postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster
 quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-
dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-sy
nc squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syncthing-relay synergy syslog syslog-tls t
elnet tentacle tftp tigervnc tigervnc-https tile38 tinc tor-socks transmission-client upnp-client vdsm vnc-serv
er warpinator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp
 wsdd wsman wsmans x11 xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
```

Рисунок 3.19 – Список сервісів, що покриває firewalld

За допомогою команди “sudo firewall-cmd --get-default-zone” можна дізнатись зону за замовчуванням. В нашому випадку це зона “public”. Роздивимось її налаштування на рис. 3.20:

```
admin@localhost:~> sudo firewall-cmd --zone=public --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Рисунок 3.20 – Налаштування публічної зони

Як видно, в публічній зоні дозволений сервіс ssh, порт якого визначено у файлі “/etc/firewalld/services/ssh.xml” та за замовчуванням є портом 22. Вимкнути цей сервіс в зоні назавжди можливо командою “sudo firewall-cmd --permanent --zone=public --

remove-service=ssh", або без ключа "--permanent", якщо це потрібно лише тимчасово. Після цього потрібно ввести команду "sudo firewall-cmd --reload", щоб зберегти зміни. Якщо дії успішно виконались, буде виведене відповідне повідомлення "success". Після цього можна перевірити список сервісів у зоні "public", і там буде лише dhcpv6-client, без ssh (рис. 3.21):

```
admin@localhost:~> sudo firewall-cmd --zone=public --remove-service=ssh
[sudo] password for root:
success
admin@localhost:~> sudo firewall-cmd --zone=public --permanent --remove-service=ssh
success
admin@localhost:~> sudo firewall-cmd --reload
success
admin@localhost:~> sudo firewall-cmd --zone=public --list-services
dhcpv6-client
```

Рисунок 3.21 – Успішне видалення дозволеного сервісу ssh

Для домашньої зони можна додати порти 80/tcp та 443/tcp, або інші. У випадку потреби, можливо увійти в режим "паніки", в якому будуть блокуватися усі з'єднання, командою "sudo firewall-cmd --panic-on".

3.3.2 Управління ризиками: налаштування автоматичних оновлень

Автоматичні оновлення можна просто налаштувати за допомогою YaST:

- Завантажуємо пакет yast-online-update-configuration.
- Відкриваємо його в YaST та налаштовуємо (рис. 3.22):

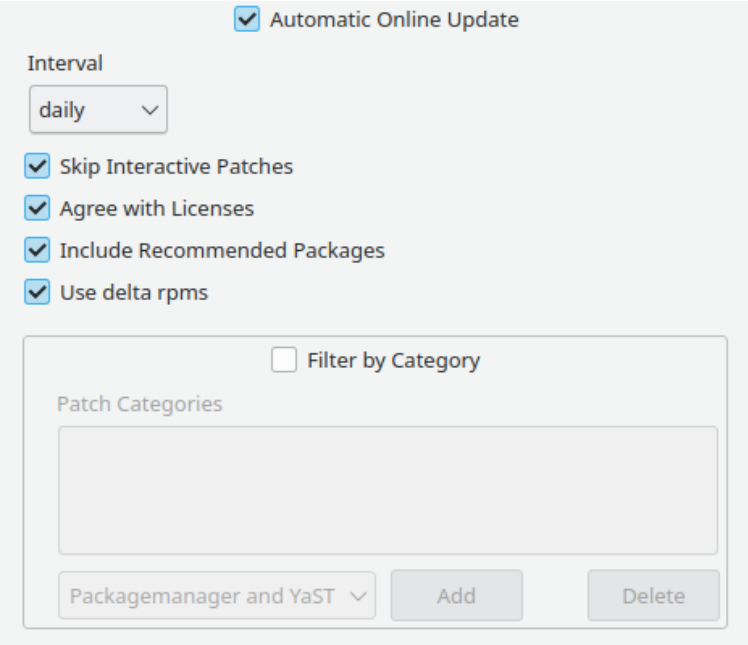


Рисунок 3.22 – Базові налаштування автоматичних оновлень

В даних налаштуваннях було обрано: автоматично оновлювати ПЗ кожен день, пропускати інтерактивні оновлення (для того, щоб ті оновлення, що не потребують втручання користувача, встановлювалися в першу чергу, а потім користувач міг вручну оновити інші пакети), згоджуватися з ліцензіями (або їх змінами) для пришвидшення та автоматизації процесу, включити рекомендовані пакети (тобто додаткові пакети, які потрібно встановлювати як залежності для оновлених пакетів), а також використовувати дельта RPM-пакети (пакети, які мають меншу вагу, проте потребують більше процесорних ресурсів – для пришвидшення завантажень).

Перед додаванням задачі автоматичних оновлень можна також відкрити додаткові налаштування, де обираються політики оновлень репозиторіїв. Там можна налаштувати пріорітети репозиторіїв, включення їх до списку оновлень та інші функції.

3.3.3 Управління інцидентами: налаштування системи IDS / IPS

В дану ОС було інтегровано систему виявлення та запобігання вторгненням Suricata. Налаштуємо її:

- Завантаження правил: можна завантажити набір правил Emerging Threats Open Ruleset для Suricata вручну, проте він автоматично завантажується утилітою suricata-update, тому достатньо просто ввести команду “sudo suricata-update”, що завантажить та оновить всі стандартні набори правил.
- Створимо заплановану задачу для оновлення бази правил кожної години. Для цього виконаємо послідовно наступні дії:

1. Виконаємо команду “sudo crontab -e”.
2. Запишемо у файл строку “12 0 * * * suricata-update” та збережемо зміни. Таким чином ми заплануємо задачу, що буде виконуватися кожен день о 12:00.

Для кращої автоматизації бажано використовувати anacron, щоб гарантувати виконання завдань у випадках, коли система вимкнена у час, на який заплановано задачу. Для того, щоб заплановані задачі Cron користувачем root стабільно виконувались, достатньо стандартних налаштувань anacron, які автоматично встановлюються після завантаження цієї утиліти.

Для підтримки перезавантажень правил під час роботи Suricata (в live-режимі), до кінця файлу “/etc/suricata/suricata.yaml” можна додати наступні строки (рис. 3.23):

Рисунок 3.23 – Включення перезавантажень правил під час роботи

Після цих дій можна додати власні правила. Для прикладу візьмемо просте правило, яке реагує на SSH з’єднання до не-SSH портів. Для цього виконаємо наступні дії:

1. Дізнаємось свою IP адресу командою “ip -brief address”
2. Відкриємо файл “/var/lib/suricata/rules/local.rules” та вставимо в нього наступну строку: “alert ssh any any -> 192.168.0.102 !22 (msg:"SSH TRAFFIC on non-SSH port"; flow:to_client, not_established; classtype: misc-attack; target: dest_ip; sid:1000000;)”, де 192.168.0.102 – наша IP адреса.

Аналогічно можна додати правила для нестандартного трафіку http:

“alert http any any -> 192.168.0.102 !80 (msg:"HTTP REQUEST on non-HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000002;)”, або TLS: “alert tls any any -> 192.168.0.102 !443 (msg:"TLS TRAFFIC on non-TLS HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000004;)”.

3. Тепер у файлі “/etc/suricata/suricata.yaml” потрібно додати користувацький список правил. Потрібно знайти строку номер з налаштуванням “rule-files” та дописати строку “- local.rules” (рис. 3.24):

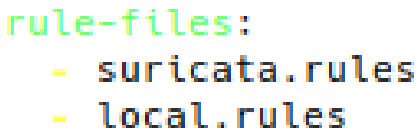


Рисунок 3.24 – Включення користувацького файлу правил

- Перевіримо на вірність налаштування Suricata: для цього потрібно виконати команду “sudo suricata -T -c /etc/suricata/suricata.yaml -v”. Успішна перевірка виглядає приблизно так (рис. 3.25):

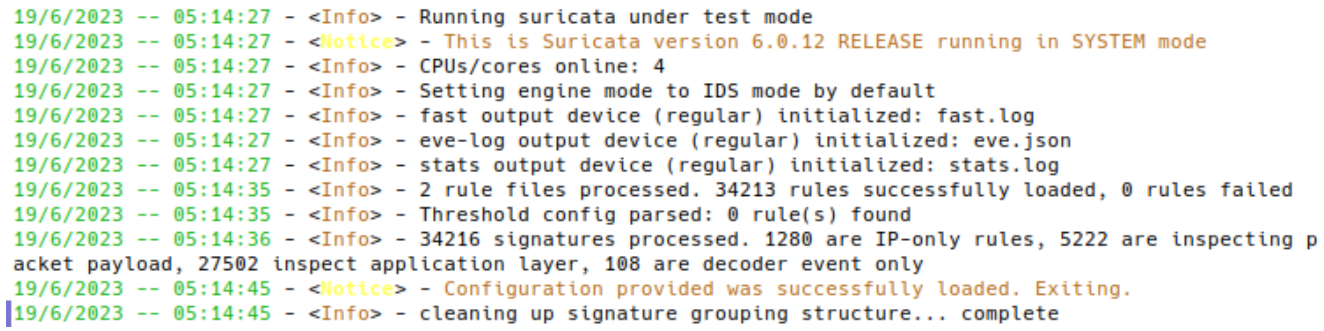


Рисунок 3.25 – Успішна перевірка правил Suricata

На даний момент наші правила налаштовані лише на виявлення інцидентів. Щоб змінити налаштування правил для реалізації функціоналу IPS, достатньо налаштувати їх на режим “drop” або “reject” шляхом заміни першого слова “alert” потрібних правил.

- Переналаштуємо Suricata для роботи в режимі NFQUEUE, тобто на високорівневій взаємодії з фаєрволом для інтеграції у вже налаштовані правила

firewalld: до цього, Suricata була налаштована на низькорівневий режим AF-PACKET, де використовувався безпосередній доступ до мережевого інтерфейсу.

1. Потрібно включити налаштування nfqueue в файлі конфігурації: Для цього достатньо розкоментувати строки “nfq:”, “mode: accept” та “fail-open: yes” у файлі “/etc/suricata/suricata.yaml” (рис. 3.26):

```
nfq:
  mode: accept
#  repeat-mark: 1
#  repeat-mask: 1
#  bypass-mark: 1
#  bypass-mask: 1
#  route-queue: 2
#  batchcount: 20
  fail-open: yes
```

Рисунок 3.26 – Налаштування Suricata для режиму NFQUEUE

2. Тепер у файлі “/etc/sysconfig/suricata” в строку “SURICATA-OPTIONS:” потрібно додати параметри “-q 0 -D”, щоб Suricata працювала як демон в режимі NFQ.

3. Далі треба перезавантажити Suricata командою “sudo systemctl restart suricata”. Після перезавантаження перевіримо статус Suricata командою “sudo systemctl status suricata”, повинно вивести подібне повідомлення (рис. 3.27):

```
[sudo] password for root:
● suricata.service - Suricata Intrusion Detection and Prevention Tool
     Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset: disabled)
     Active: active (running) since Mon 2023-06-19 06:13:20 EEST; 5min ago
       Docs: man:suricata(1)
    Process: 17066 ExecStart=/usr/bin/suricata -c /etc/suricata/suricata.yaml $SURICATA_OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 17066 (code=exited, status=0/SUCCESS)
      Tasks: 12 (limit: 4915)
        CPU: 18.555s
     CGroup: /system.slice/suricata.service
             └─17070 /usr/bin/suricata -c /etc/suricata/suricata.yaml -q 0 -D
```

Рисунок 3.27 – статус активної Suricata

- Налаштування фаєрволу firewalld для перенаправлення трафіку через Suricata: для цього потрібно ввести 3 команди:
  - ❖ “firewall-cmd --direct --add-rule ipv4 filter INPUT 0 -j NFQUEUE”

- “firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -j NFQUEUE”
- “firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -j NFQUEUE”

Замість цього можна написати shell-скрипт з цим кодом, який будемо запускати, коли потрібно налаштувати фаєрвол для перенаправлення трафіку через Suricata. Варто відмітити, що ці команди змінюють налаштування фаєрволу до наступного його перезавантаження для зручності.

Тепер Suricata готова постійно працювати в активному режимі. За потреби сервіс можна виключити за допомогою утиліти systemctl, проте потрібно не забути перезавантажити фаєрвол, щоб відмінити перенаправлення трафіку.

Лог-файли знаходяться за розташуванням “/var/log/suricata/”.

3.3.4 Аудит системи

Lynis – це потужний інструмент для аудиту, покращення безпеки системи та перевірки відповідності вимогам, пентестингу та виявлення вразливостей.

Розпочати роботу з Lynis можна просто виконавши команду “sudo lynis audit system”. За потреби можливо також виконати аудит віддаленої системи або docker-образів.

Після сканування даної системи, в лог-файл за розташуванням “/var/log/lynis.log” було занесено 8452 строки. Короткий підсумок зазначає рівень “загартування”, тобто безпеки системи, у 82% за шкалою стандартного тесту lynis. Висновок у логах: “Система виглядає достатньо загартованою”. Скріншот фрагменту логів з підсумками наведено на рис. 3.28:

```
Hardening index : [82] [################    ]
Hardening strength: System seem to be decent hardened
====
Checking permissions of /usr/share/lynis/include/tool_tips
File permissions are OK
Tool tips: enabled
================================================================
Tests performed:     257
Total tests:         452
Active plugins:      0
Total plugins:       0
```

Рисунок 3.28 – Висновок у лог-файлі з аудиту системи

Під час сканування для кожного етапу формуються оцінки трьох рівнів: зелений означає, що все добре, жовтий – є спірні моменти, або не всі умови виконані, червоний сповіщує, що є проблема, з якою потрібно якомога швидше розібратись. Серед перевірених дозволів файлів не було жодного червоного статусу, а більшість спірних питань стосувались сервісу Cron. Результат сканування цих дозволів представлено на рис. 3.29:

```
- Starting file permissions check
  File: /boot/grub2/grub.cfg                                  [ OK ]
  File: /etc/at.deny                                          [ SUGGESTION ]
  File: /etc/cron.allow                                       [ SUGGESTION ]
  File: /etc/cron.deny                                        [ OK ]
  File: /etc/crontab                                          [ OK ]
  File: /etc/group                                            [ OK ]
  File: /etc/group-                                           [ OK ]
  File: /etc/hosts.allow                                      [ SUGGESTION ]
  File: /etc/hosts.deny                                       [ SUGGESTION ]
  File: /etc/issue                                            [ OK ]
  File: /etc/issue.net                                        [ OK ]
  File: /etc/passwd                                           [ OK ]
  File: /etc/passwd-                                          [ OK ]
  File: /etc/hosts.equiv                                      [ OK ]
  Directory: /root/.ssh                                       [ OK ]
  Directory: /etc/cron.d                                      [ SUGGESTION ]
  Directory: /etc/cron.daily                                  [ SUGGESTION ]
  Directory: /etc/cron.hourly                                 [ SUGGESTION ]
  Directory: /etc/cron.weekly                                 [ SUGGESTION ]
  Directory: /etc/cron.monthly                                [ SUGGESTION ]
```

Рисунок 3.29 – Аудит розмежування доступу до файлів

В кінці сканування програма надає рекомендації з покращення безпеки, які системний адміністратор повинен проаналізувати та впровадити відповідні рішення. Фрагмент таких рекомендацій наведено на рис. 3.30:

```
Set a password on GRUB boot loader to prevent altering boot configuration (e.g. boot in single user mode without password)
  https://cisofy.com/lynis/controls/BOOT-5122/

Consider hardening system services [BOOT-5264]
- Details  : Run '/usr/bin/systemd-analyze security SERVICE' for each service
  https://cisofy.com/lynis/controls/BOOT-5264/

If not required, consider explicit disabling of core dump in /usr/etc/security/limits.conf file [KRNL-5820]
  https://cisofy.com/lynis/controls/KRNL-5820/

Check PAM configuration, add rounds if applicable and expire passwords to encrypt with new values [AUTH-9229]
  https://cisofy.com/lynis/controls/AUTH-9229/

When possible set expire dates for all password protected accounts [AUTH-9282]
  https://cisofy.com/lynis/controls/AUTH-9282/

Look at the locked accounts and consider removing them [AUTH-9284]
  https://cisofy.com/lynis/controls/AUTH-9284/

Disable drivers like USB storage when not used, to prevent unauthorized storage or data theft [USB-1000]
  https://cisofy.com/lynis/controls/USB-1000/

Disable drivers like firewire storage when not used, to prevent unauthorized storage or data theft [STRG-1846]
  https://cisofy.com/lynis/controls/STRG-1846/

Split resolving between localhost and the hostname of the system [NAME-4406]
  https://cisofy.com/lynis/controls/NAME-4406/

Check your resolv.conf file and fill in a backup nameserver if possible [NETW-2705]
  https://cisofy.com/lynis/controls/NETW-2705/

Determine if protocol 'dccp' is really needed on this system [NETW-3200]
  https://cisofy.com/lynis/controls/NETW-3200/

Determine if protocol 'sctp' is really needed on this system [NETW-3200]
  https://cisofy.com/lynis/controls/NETW-3200/
```

Рисунок 3.30 – Фрагмент рекомендацій з покращення безпеки

Для отримання додаткової інформації щодо згенерованих рекомендацій, потрібно взяти TEST-ID, що вказаний в квадратних дужках після опису рекомендації, та виконати команду “lynis show details TEST-ID”. Прикладом візьмемо тест AUTH-9282 (рис. 3.31):

```
Performing test ID AUTH-9282 (Checking password protected account without expire date)
Test: Checking Linux version and password expire date status
Result: found one or more accounts without expire date set
Account without expire date: admin
Suggestion: When possible set expire dates for all password protected accounts [test:AUTH-9282] [details:-] [solution:-]
```

Рисунок 3.31 – Подробиці щодо рекомендацій після тестування

3.3.5 Практичні рекомендації

В ході впровадження політик безпеки було проаналізовано та налаштовано деякі сервіси, оглянуто базові налаштування фаєрволу, налаштовано систему IDS/IPS Suricata та виконано аудит системи за допомогою Lynis. Для впровадження вищого рівня безпеки було розроблено наступні рекомендації:

Для запобігання несанкціонованому доступу рекомендується проаналізувати критичні дані та процеси, до яких потрібно обмежити доступ, та розробити і впровадити більш широкі політики доступу використовуючи такі інструменти та методики, як:

- Фаєрволи, фільтрація мережевого трафіку, заборона сканування портів.
- Налаштування вибіркового керування доступом (DAC) за допомогою утиліт chmod, chown, chgrp.
- Налаштування мандатного керування доступом (MAC), наприклад за допомогою профілів AppArmor.
- Створення та налаштування гістьового користувача з мінімальними привілеями для ситуацій, коли потрібно лише користуватися мінімальним списком ПЗ, або для сценаріїв користування ОС в публічних місцях.
- Налаштування обмеженої shell.
- Використання контейнеризації та chroot-jail для програмних додатків, у яких підвищений ризик компрометації, мають повільний цикл оновлень або які потенційно можуть стати вектором атак.
- Використання засобу контролю системних привілеїв PolicyKit.
- Налаштування за замовчуванням на доступ до сервісів за принципом білого списку.

- Глибокі налаштування зв’язки firewalld-Suricata для більш якісного контролю трафіку та запобігання втручанням у систему.
- Встановлення антивірусного ПЗ.
- Інші налаштування на базі звітів з аудиту системи.

Для зниження ризиків потрібно перевіряти джерела програмного забезпечення, постійно його оновлювати (можливо налаштувати систему автоматичних оновлень після запуску системи) та налаштовувати привілеї додатків, а також максимально зменшити кількість програмних додатків задля зменшення площі атак та, як наслідок, об’єму інформації, що підлягає моніторингу. Бажано впровадити парольні політики, що контролюватимуть якість паролів, а також вимагатимуть періодичного їх оновлення.

Для моніторингу безпеки системи потрібно використовувати інструменти аудиту типу Lynis, OpenSCAP та інших, що автоматично перевіряють налаштування системи відповідно обраним політикам та надають звіт відповідності. Лог-файли бажано автоматизовано та вручну аналізувати, а також можна зберігати їх резервні копії на окремому сервері.

3.4 Розробка напрямів підтримки та розвитку

Дана частина роботи спрямована на розробку можливих напрямів розвитку дистрибутиву та покращень до системи.

Важливим елементом навчальної операційної системи є документація, яка повинна бути легко доступною та зрозумілою для користувачів. Документація має містити опис функціональності системи, інструкції з користування та розв'язання можливих проблем, а також матеріали для навчання та підвищення кваліфікації.

- Таким чином, одним з пунктів розвитку має стати, як мінімум, забезпечення наявності доступної документації на англійській мові і, бажано, на українській мові. Англомовна документація вже присутня та інтегрована в дану ОС розробником.

- Можливим доповненням стане створення та інтеграція україномовної документації. Інтеграція посилань на неї у систему.
- Покращення політик безпеки, моделі загроз, оперативної безпеки.
- Постійне проведення аудиту системи та впровадження покращень безпеки.
- Оцінка надійності, стабільності та безпеки системи, доведення відповідності достатньому рівню цих показників за допомогою розширеного аудиту.
- Створення та інтеграція україномовної документації. Інтеграція посилань на неї у систему, або створення централізованого сховища або ресурсу документації та інструкцій на українській мові.
- Оцінка ПЗ, аналіз вимог до ПЗ, переоцінка цих аспектів, доведення відповідності ПЗ вимогам.
- Формування списку вправ, які потрібно відпрацьовувати, щоб забезпечити можливість підготовки до міжнародних сертифікаційних іспитів у галузі ІБ.
- Переоцінка моделі оновлення, можливе впровадження власної курованої схеми.
- Система контролю оновлень та відслідковування звітів про несправні оновлення, щоб запобігти встановленню некорректно працюючих пакетів.

Доступність: мінімум – створення ISO образів, їх вчасне оновлення (за потреби), та їх розповсюдження з використанням сервісів хостингу. Додатково – хостинг попередніх версій дистрибутиву, веб-сайт. З цього випливає, що для проекту є потрібним:

- Створення або підбору готового рішення для автоматичного або автоматизованого створення ISO образів.
- Створення інфраструктури для розповсюдження дистрибутиву, а саме у вигляді як мінімум власного веб-серверу, додатково з веб-сайтом.
- Бажано мати інфраструктуру для автоматизованого оновлення цих ISO образів на веб-сервері, наприклад за моделлю “автоматичне створення оновленого

ISO-образу – ручна/автоматична перевірка образу з підтвердженням працездатності, автоматичне завантаження оновленої версії на веб-сервер після підтвердження працездатності”.

Висновки до розділу 3

В даному розділі було описано інтегроване програмне забезпечення, визначені теоретичні аспекти використання дистрибутиву, проведено практичне впровадження розроблених політик безпеки та зроблені практичні рекомендації щодо покращення безпеки системи. Також додатково було розроблено напрями підтримки та розвитку системи.

ВИСНОВКИ

В ході даної кваліфікаційної роботи було розроблено вимоги, модель безпеки та зкомпоновано спеціалізований дистрибутив Linux, що налаштований та містить додатки для вивчення спеціальності “Кібербезпека”.

У першому розділі було проаналізовано архітектурні рішення UNIX та UNIX-подібних ОС, а також розглянуті спеціалізовані ОС, що можна використовувати в сфері інформаційної безпеки. Серед аспектів операційних систем з точки зору кібербезпеки було розглянуто відкритість вихідного коду, архітектури ядра, модель кілець безпеки, простір користувача та ядра, контейнеризацію.

Друга частина роботи стосується теоретичної частини розробки дистрибутиву Linux для спеціаліста з ІБ. Було створено план розробки, сформовано вимоги, розроблено модель безпеки, виділено необхідні та бажані компоненти і характеристики ОС, проаналізовано категорії ПЗ, що повинні увійти в дистрибутив та на основі цих категорій розроблено список додатків, проаналізовано існуючі дистрибутиви для формування програмної основи.

Третій розділ відповідає практичній реалізації розроблених вимог та концепцій. В ньому описано процеси компонування програмних додатків, впровадження моделі безпеки, розроблено рекомендації для подальшого покращення безпеки системи, розглянуто теоретичні сценарії використання дистрибутиву.

В процесі виконання даної кваліфікаційної роботи були виконані наступні завдання:

- Досліджено історію, архітектуру та концепції UNIX та UNIX-подібних операційних систем.
- Досліджено та проаналізовано ринок UNIX-подібних операційних систем, що можуть використовуватися в галузі інформаційної безпеки.
- Проаналізовано питання, сценарії та можливості використання спеціалізованих операційних систем для фахівців у галузі інформаційної безпеки.

- Розроблено вимоги та модель безпеки для операційної системи спеціаліста з ІБ.
- Зкомпоновано дистрибутив Linux, що можна використати для академічних та професійних цілей в сфері інформаційної безпеки.
- Виконані налаштування та проведений аудит безпеки операційної системи.
- Сформовані рекомендації для подальшого покращення захисту системи.
- Описані теоретичні аспекти використання сформованого дистрибутиву.
- Сформовано напрями підтримки та розвитку системи.

Всі поставлені задачі було виконано в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. КОНСТИТУЦІЯ УКРАЇНИ [Електронний ресурс]. – 1996. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/254%D0%BA/96-%D0%B2%D1%80#Text.

2. ЦИВІЛЬНИЙ КОДЕКС УКРАЇНИ [Електронний ресурс]. – 2003. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/435-15#Text.

3. ЗАКОН УКРАЇНИ "Про авторське право і суміжні права" [Електронний ресурс]. – 1994. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/3792-12#Text.

4. The ideology of Open-Source and its related concepts. [Електронний ресурс] – Режим доступу до ресурсу: https://greyhatlinux.medium.com/the-ideology-of-open-source-and-its-related-concepts-5425d5f9906c.

5. ОСНОВИ ОПЕРАЦІЙНИХ СИСТЕМ / А. С. Авраменко, В. С. Авраменко. – Черкаси, 2018. – 524 с.

6. Linux – The Complete Reference, 2008. – 830 c. – (McGraw-Hill).

7. The Art of Unix Programming [Електронний ресурс]. – 2003. – Режим доступу до ресурсу: http://www.catb.org/esr/writings/taoup/html/.

8. In UNIX Everything is a File [Електронний ресурс] – Режим доступу до ресурсу: https://hackmd.io/@jkyang/unix-everything-is-a-file.

9. The History of the UNIX operating system (Part 1) [Електронний ресурс] – Режим доступу до ресурсу: https://www.redhotcyber.com/en/post/the-history-of-the-unix-operating-system-part-1/.

10. Protection ring [Електронний ресурс] – Режим доступу до ресурсу: https://lsi.vc.ehu.eus/pablogn/docencia/ISO/2%20Llamadas%20al%20Sistema,%20Kernel/varios/Protection%20ring.pdf

11. A Comprehensive Study of Kernel (Issues and Concepts) in Different Operating Systems [Електронний ресурс]. – 2021. – Режим доступу до ресурсу:

https://www.researchgate.net/publication/351424523_A_Comprehensive_Study_of_Kernel_Issues_and_Concepts_in_Different_Operating_Systems.

12. Research of an architecture of operating system kernel based on modularity concept [Електронний ресурс]. – 2010. – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S0895717709003409.

13. The history of documented Unix facilities [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://dspinellis.github.io/unix-history-man/.

14. Linux Kernel Development (3rd Edition) – RR Donnelley, Crawfordsville, Indiana: Addison-Wesley, 2010. – 440 с.

15. Understanding Linux Kernel, 2006. – 923 с. – (O'Reilly).

16. Evolution of the Unix System Architecture: An Exploratory Case Study [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.researchgate.net/publication/332826685_Evolution_of_the_Unix_System_Architecture_An_Exploratory_Case_Study.

17. Microkernels Really Do Improve Security [Електронний ресурс] – Режим доступу до ресурсу: https://microkerneldude.org/2018/08/23/microkernels-really-do-improve-security/.

18. Towards Secure and Reliable Firewall Systems based on Minix 3 [Електронний ресурс] – Режим доступу до ресурсу: https://subs.emis.de/LNI/Proceedings/Proceedings170/85.pdf.

19. An Introduction to MINIX [Електронний ресурс] – Режим доступу до ресурсу: https://www.linuxjournal.com/article/10754.

20. A Coprocessor-based Introspection Framework via Intel Management Engine [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.researchgate.net/publication/350704741_A_Coprocessor-based_Introspection_Framework_via_Intel_Management_Engine.

21. What Is Intel Management Engine Components And How To Disable It [Електронний ресурс] – Режим доступу до ресурсу: https://www.itechtics.com/intel-management-engine-components/.

22. Intel Management Engine, Explained: The Tiny Computer Inside Your CPU [Електронний ресурс] – Режим доступу до ресурсу: https://www.howtogeek.com/334013/intel-management-engine-explained-the-tiny-computer-inside-your-cpu/.

23. WHAT YOU NEED TO KNOW ABOUT THE INTEL MANAGEMENT ENGINE [Електронний ресурс] – Режим доступу до ресурсу: https://hackaday.com/2017/12/11/what-you-need-to-know-about-the-intel-management-engine/.

24. What is OpenBSD? Overview, Latest Features, & Security Considerations [Електронний ресурс] – Режим доступу до ресурсу: https://www.liquidweb.com/blog/what-is-openbsd/.

25. Get Smart with SmartOS [Електронний ресурс] – Режим доступу до ресурсу: https://www.admin-magazine.com/Articles/SmartOS-Cool-Cloud-Platform-Rises-from-the-Ashes-of-Solaris.

26. New SmartOS: Ready to Serve as Next VM or Container Host [Електронний ресурс] – Режим доступу до ресурсу: https://thenewstack.io/a-new-release-of-smartos-is-available-and-ready-to-serve-as-your-next-virtual-machine-or-container-host/.

27. Getting to know the Solaris Fault Management Architecture (FMA) [Електронний ресурс] – Режим доступу до ресурсу: https://prefetch.net/presentations/SolarisFaultManagement_Presentation.pdf.

28. DTrace Topics: Introduction [Електронний ресурс] – Режим доступу до ресурсу: https://www.brendangregg.com/Slides/dtrace_topics_intro.pdf.

29. Developing Cyber-Resilient Systems: A Systems Security Engineering Approach [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v2r1.pdf

30. Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations [Електронний ресурс] // NIST. – 2021. – Режим доступу до ресурсу: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-171r2.pdf.

31. Architectural design for a secure Linux operating system [Електронний ресурс] – Режим доступу до ресурсу:

https://www.researchgate.net/publication/323354090_Architectural_design_for_a_secure_Linux_operating_system.

32. Linux Security: A Survey [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.researchgate.net/publication/335795125_Linux_Security_A_Survey.

33. Arch Linux review: Is it worth installing? [Електронний ресурс] – Режим доступу до ресурсу: https://www.infoworld.com/article/3017521/arch-linux-review-is-it-worth-installing.html.

34. Endeavour OS Review [Електронний ресурс] – Режим доступу до ресурсу: https://www.slant.co/options/35116/~endeavour-os-review.

35. NixOS: the good, the bad, and the ugly [Електронний ресурс] – Режим доступу до ресурсу: http://www.willghatch.net/blog/2020/06/27/nixos-the-good-the-bad-and-the-ugly/.