МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій,
обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Шакури Івана Юрійовича
(прізвище, ім'я, по батькові здобувача освіти)

УДК 004:451:004.056

КВАЛІФІКАЦІЙНА РОБОТА

Система контролю цілісності критичних інформаційних ресурсів для операційної системи Linux

(тема роботи)

КБ-23-м 125-«Кібербезпека та захист інформації»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня магістр

кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Корченко Анна Олександрівна
(прізвище, ім'я, по батькові)
д.т.н, професор
(науковий ступінь, вчене звання)

Житомир – 2024

Висновок кафедри _____
за результатами попереднього захисту:
_____

Протокол засідання кафедри
_____
№ ____ від «_____» _____ 20____ р.

Завідувач кафедри _____
_____ _____
_____
(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)
«_____» _____ 20____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)
(прізвище ,ім'я, по батькові)
кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____
за шкалою ECTS _____
за національною шкалою _____

Секретар ЕК
_____ _____
_____
(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)

АНОТАЦІЯ

Шакура І.Ю. Система контролю цілісності критичних інформаційних ресурсів для операційної системи Linux. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня магістр за спеціальністю 125 – кібербезпека та захист інформації. – Поліський національний університет, Житомир, 2024. У кваліфікаційній роботі було створено систему контролю цілісності критичних інформаційних ресурсів у вигляді програмного коду на мовах програмування BASH та Python.

Кваліфікаційна робота:

Ключові слова: інформаційні ресурси, Linux, CentOS, inotify, BASH, Python, Telegram бот.

SUMMARY

Shakura I.Y. Integrity Control System for Critical Information Resources in the Linux Operating System. – Qualification Work in Manuscript Form. Qualification work for obtaining the Master’s degree in specialty 125 – Cybersecurity and Information Protection. – Polissya National University, Zhytomyr, 2024. The qualification work presents a system for controlling the integrity of critical information resources developed in the form of program code using BASH and Python programming languages.

Qualification work:

Keywords: information resources, Linux, CentOS, inotify, BASH, Python, Telegram bot.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

DNS	Сервер доменних імен
API	Application programming interface
HTTP	Гіпертекстовий протокол передачі даних
RHEL	Red Hat Enterprise Linux
HIDS	Системи виявлення вторгнень
SUID	Біт супер користувача

ВСТУП

Актуальність теми. Адміністрування Linux-систем потребує високої уважності, оскільки навіть незначна помилка або вразливість може мати серйозні наслідки для безпеки системи. Зростаючі кіберзагрози та складність сучасних IT-інфраструктур збільшують ризик того, що адміністратор може пропустити важливі зміни або атаки на критичні файли системи. Відсутність оперативного реагування на несанкціоновані модифікації може призвести до захоплення контролю над системою зловмисниками, що загрожує втратою даних та компрометацією безпеки.

Наукова новизна. Удосконалено систему контролю цілісності критичних інформаційних ресурсів, за рахунок використання вбудованих функцій ядра Linux, утиліти inotify-tools, Telegram-ботів, BASH-скриптів та Python-скриптів, що дало можливість керувати та відстежувати критично важливі файли з використанням мінімуму системних ресурсів та сторонніх пакетів програм.

Метою кваліфікаційної роботи є удосконалення систем контролю цілісності критичних інформаційних об’єктів в операційній системі Linux, шляхом використання вбудованих функцій ядра операційної системи, BASH-скриптів та програмування на Python.

Задачі поставлені в кваліфікаційній роботі:

1) Аналіз систем контролю цілісності інформаційних ресурсів для різних операційних систем.
2) Розробка системи контролю цілісності критичних інформаційних ресурсів операційної системи Linux.
3) Експериментальне дослідження системи контролю цілісності критичних інформаційних ресурсів.

Об’єктом дослідження є процес здійснення контролю за критичними інформаційними ресурсами операційної системи Linux.

Предметом дослідження є методи та системи забезпечення контролю цілісності інформаційних ресурсів для операційних систем.

Практична цінність. Розроблено алгоритмічне та програмне забезпечення контролю цілісності критичних інформаційних ресурсів за допомогою функцій ядра Linux, BASH та Python.

За темою кваліфікаційної роботи було опубліковано наукові публікації, а саме:

- Шакура І.Ю., КОНТРОЛЬ ЦІЛІСНОСТІ КРИТИЧНИХ ІНФОРМАЦІЙНИХ РЕСУРСІВ ОПЕРАЦІЙНОЇ СИСТЕМИ LINUX. Матеріали IX Всеукраїнської науково-практичної конференції молодих науковців «Litteris et Artibus: Нові горизонти».Кременець, листопад 12, 2024. с.108;

- Шакура І.Ю., УПРАВЛІННЯ ПРАВАМИ ДОСТУПУ В LINUX-СИСТЕМАХ: ВІД КЛАСИЧНИХ МЕТОДІВ ДО СУЧАСНИХ РІШЕНЬ. Матеріали VI Міжнародної наукової конференції «Розвиток наукової думки постіндустріального суспільства: сучасний дискурс». Хмельницький, листопад 1, 2024. с. 360;

- Шакура І.Ю., ЗАХИСТ КРИТИЧНИХ ІНФОРМАЦІЙНИХ РЕСУРСІВ ТА БЕЗПЕКА ОПЕРАЦІЙНОЇ СИСТЕМИ LINUX. Матеріали VIII Міжнародної наукової конференції «Здобутки та досягнення прикладних та фундаментальних наук XXI століття». Біла Церква, листопад 22, 2024. с. 312;

Структура та обсяг роботи. Дипломна робота складається зі вступу, трьох розділів основної частини, висновків та списку використаних джерел.

Розділ 1. ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ СИСТЕМИ КОНТРОЛЮ ЦІЛІСНОСТІ КРИТИЧНИХ ОБ'ЄКТІВ

1.1 Перегляд аналогічних систем

Webmin — це веб-інструмент для адміністрування Unix-подібних серверів і служб, що встановлюється близько 1 000 000 разів щороку. Він дозволяє налаштовувати ключові елементи операційної системи, такі як користувачі, дискові квоти, служби, конфігураційні файли, а також контролювати програми з відкритим кодом, наприклад, BIND DNS Server, Apache HTTP Server, PHP та MySQL.

Webmin забезпечує збір логів, який дозволяє переглядати системні логи через різні модулі. Однак, його можливості в цій сфері обмежені, оскільки він не підтримує глибокий збір логів з різних джерел. Щодо моніторингу критичних файлів та моніторингу процесів, Webmin дозволяє переглядати активні процеси, але не пропонує детального аналізу або тривожних сповіщень при виявленні підозрілої активності.

У контексті сповіщення, Webmin не надає автоматизованих функцій для сповіщення адміністратора про критичні зміни чи події. Також програма не має функцій автоматичної протидії або автоматичного виправлення, що обмежує її можливості у випадку виявлення загроз.

Таким чином, хоча Webmin є корисним інструментом для базового адміністрування, його функціонал стосується лише основних аспектів моніторингу та управління, без розширених можливостей для захисту даних і виявлення загроз. [1]

The RHEL web console — це зручний веб-інтерфейс, розроблений на основі проєкту Cockpit, для адміністрування системи. Він дозволяє адміністраторам виконувати ключові завдання, такі як перевірка системних служб, управління дисковим простором, налаштування мережі, а також моніторинг проблем з мережею.

Серед його можливостей — збір логів, що дає змогу переглядати основні системні логи RHEL у веб-інтерфейсі, проте консоль не забезпечує глибокого

аналізу чи збору даних з різних джерел. Що стосується моніторингу процесів, ця функція реалізована, але без розширених опцій, що обмежує можливості адміністраторів у виявленні проблем. Однак, консоль не надає автоматизованих сповіщень про критичні зміни в системі, що може зменшити реакцію на потенційні загрози. Також відсутні функції автоматичної протидії або автоматичного виправлення, які б підвищили загальний рівень захисту системи.

Таким чином, RHEL web console є корисним інструментом для виконання базових адміністративних завдань, проте її можливості в моніторингу та захисті інформації залишаються обмеженими. [2]

AIDE — це хост-система для виявлення атак і вторгнень для систем Linux. Це недорогий інструмент, який можна використовувати для перевірки цілісності системи.

Це має допомогти адміністратору розпізнати, чи було змінено файли чи каталоги системи щодо їхнього вмісту та/або їхніх властивостей, наприклад дозволів файлової системи, контексту SELinux, розширених атрибутів тощо.

Слабкі сторони AIDE та подібних систем виявлення вторгнень на основі хоста

- Програма, файл(и) конфігурації, база даних і файл журналу розташовані локально на відповідному хості.
- Зловмисники, які можуть змінювати локальні файли, потенційно також можуть змінювати файли, що належать AIDE.
- Існує ризик того, що перевірки цілісності не є надійними, якщо сама IDS була скомпрометована.[3]

Система виявлення вторгнень (HIDS) Samhain забезпечує перевірку цілісності файлів, моніторинг і аналіз журналів, а також виявлення руткітів, моніторинг портів, виявлення підозрілих виконуваних файлів з правами SUID та прихованих процесів.

Samhain призначений для моніторингу кількох хостів із різними операційними системами, підтримуючи централізоване ведення журналів і адміністрування, хоча його можна використовувати і як окрему програму на

одному хості. Інструмент має обмежену функцію збору логів і може відправляти результати перевірки цілісності до системного журналу або бази даних. Хоча Samhain не виконує прямого моніторингу процесів, він здатний виявляти змінені файли, що може свідчити про активність сторонніх процесів. [4]

Observium — це платформа для моніторингу та управління мережею, яка надає інформацію про стан і продуктивність мережевих пристроїв у реальному часі. Вона автоматично виявляє мережеві пристрої та служби, збирає показники продуктивності та надсилає сповіщення у разі виявлення проблем. Завдяки зручному веб-інтерфейсу користувачі можуть переглядати поточні та історичні дані про стан мережі, що допомагає адміністраторам швидко виявляти та усувати проблеми.

Observium підтримує широкий спектр пристроїв і платформ, включаючи Cisco, Windows, Linux, HP, Juniper, Dell, FreeBSD, Brocade, Netscaler, NetApp та багато інших. Хоча Observium не збирає логи безпосередньо, він відображає інформацію про пристрої, які надсилають логи через протокол SNMP. Програма орієнтована на мережевий моніторинг і не має функцій для моніторингу окремих процесів на сервері.[5]

Monitorix — це безкоштовний легкий інструмент моніторингу системи з відкритим кодом, призначений для моніторингу якомога більшої кількості служб і системних ресурсів. Він був створений для використання на робочих серверах Linux/UNIX , але завдяки своїй простоті та малому розміру його можна використовувати також на вбудованих пристроях . Він складається в основному з двох програм: збирача під назвою monitorix, який є демоном Perl, який запускається автоматично, як і будь-яка інша системна служба, і сценарію CGI під назвою monitorix.cgi. Monitorix включає власний вбудований HTTP-сервер (який за замовчуванням прослуховує порт 8080/TCP) для перегляду статистичних графіків, тож вам не доведеться встановлювати сторонній веб-сервер, щоб використовувати його. Monitorix не збирає логи у традиційному розумінні, але може відображати статистику, зокрема дані з системних логів (на

кшталт використання диска та мережі). Monitorix підтримує базовий моніторинг системних процесів та ресурсів, що дозволяє переглядати інформацію про CPU, пам'ять і навантаження.[6]

Auditd (Linux Auditing System) — це інструмент для збору та запису аудиторських журналів, вбудований у ядро Linux. Він відстежує події на рівні файлової системи, що дозволяє моніторити різноманітні дії користувачів і процесів, забезпечуючи деталізовану інформацію про доступ до файлів, зміни в них, а також дії з критичними каталогами. Цей інструмент фіксує відомості, такі як дата і час події, ідентифікатор користувача, який виконав дію, і конкретні зміни, що були внесені. Завдяки цьому адміністраторам надається можливість виявляти підозрілу активність і проводити аналіз ситуації. Auditd виконує такі функції, як моніторинг критичних файлів, збір логів і сповіщення про важливі події, проте не має механізмів для автоматичної протидії або виправлення. [10]

Auditd дозволяє гнучко налаштовувати правила моніторингу, які можна адаптувати до потреб системи. Наприклад, адміністратори можуть встановлювати правила для моніторингу змін у певних критичних файлах, контролювати запуск системних команд або слідкувати за процесами, що можуть загрожувати безпеці. Журнали, які створює Auditd, можуть переглядатися за допомогою спеціальних утиліт, таких як ausearch і aureport. Ausearch дозволяє проводити пошук конкретних подій за фільтрами, такими як користувач чи файл, а aureport зводить інформацію до загальних звітів, що дає можливість отримати цілісне уявлення про безпеку системи. [11]

Tripwire Enterprise — це потужне рішення для моніторингу цілісності файлів (File Integrity Monitoring, FIM) і керування конфігурацією безпеки (Security Configuration Management, SCM). Воно забезпечує комплексний контроль за змінами в системі в реальному часі та виявлення загроз. Tripwire виконує функції моніторингу критичних файлів, надає можливість сповіщення

про несанкціоновані зміни та фіксує дані для подальшого аналізу. Спеціалісти з комплаєнсу можуть активно зміцнювати систему, завдяки функціям моніторингу, що не лише виявляють несанкціоновані зміни, але й забезпечують негайні сповіщення про них.

Tripwire Enterprise підтримує розширену інтеграцію з іншими системами безпеки та IT-інфраструктурою, що дозволяє реалізувати цілісний підхід до захисту даних і управління конфігураціями. Інструмент дозволяє налаштовувати правила для конкретних подій, здійснювати перевірку критичних систем і контролювати конфігурацію мережевих пристроїв, серверів і хмарних середовищ. Крім того, Tripwire Enterprise має гнучкі налаштування звітності, що дозволяє адміністраторам отримувати зведені дані про стан системи та вчасно реагувати на потенційні загрози, підвищуючи загальну безпеку корпоративного середовища. [12]

1.2 Призначення системи

Основні функції для систем контролю цілісності:

- Логування.
- Сповіщення та звітність адміністратору.
- Моніторинг змін у файловій системі.
- Моніторинг процесів.
- Забезпечення цілісності даних.
- Автоматизоване реагування на інцидент.
- Відновлення після інцидентів.

Система, що розробляється, призначена для адміністраторів серверів, особливо невеликих серверних рішень, де немає можливості постійного контролю за станов файлів вручну. Основною проблемою, яку вона вирішує, є автоматична протидія несанкціонованому редагуванню або пошкодженню важливих системних файлів. У разі виявлення таких змін система здатна

автоматично відновити файли до їх початкового стану. Окрім цього, при будь-якій спробі втручання в цілістність важливих файлів система автоматично надсилає повідомлення адміністратору, щоб він міг негайно вжити заходів для подальшого захисту системи.

Однією з функцій може бути відстеження системних ресурсів, таких як оперативна пам’ять чи дисковий простір, але цей функціонал не є обов’язковим, оскільки у багатьох Linux-дистрибутивах є власні монітори продуктивності.

Системи контролю цілісності створюються для того, щоб переконатися, що:

- Обладнання працює.
- Сервер запущений і працює.
- Критичні інформаційні ресурси сервера є цілісними та не піддавались редагуванню.
- Жодні вузькі місця ресурсів не сповільнюють роботу.
- Системні адміністратори отримують сповіщення, коли KPI не відповідає вказаній метриці.[7]

1.3 Аналіз сучасних загроз та механізмів безпеки

Згідно з даними Організації економічного співробітництва та розвитку (ОЕСР), українські компанії стикаються з такими основними проблемами цифрової безпеки:

1. **Збільшення кількості кібератак**

Кількість кібератак значно зросла від початку повномасштабного вторгнення РФ в Україну. Збитки, завдані українським підприємствам кіберзлочинністю у 2022 році, склали 1 млрд грн, що на 96% більше, ніж у 2021 році.

2. **Безпека даних**

Протягом 2021-2023 років проблеми з безпекою даних зросли на 14%.

3. **Недостатній захист персональних даних**

Українське законодавство окремо не регламентує процедуру повідомлення суб'єктів персональних даних або правоохоронні органи про порушення безпеки персональних даних, що може знижувати довіру клієнтів до бізнесу.

4. **Вразливість до кіберзлочинів**

У 2018 році 31% підприємств стикалися з кіберзлочинами, що є зростанням з 24% у 2016 році. Компанії частіше стикаються зі зривом бізнес-процесів, здирництвом або порушенням прав інтелектуальної власності, а також з політично вмотивованими атаками.

5. **Підтримка МСП**

Щоб підвищити технологічну захищеність українських підприємств, Міністерство цифрової трансформації, Офіс з розвитку підприємництва та експорту, національний проєкт Дія.Бізнес та Проєкт USAID «Кібербезпека критично важливої інфраструктури України» запустили Програму з кібердіагностики бізнесу. Ініціатива має допомогти українським компаніям зрозуміти, у якому стані перебуває кібербезпека на підприємстві, а також надати рекомендації щодо покращення їх стійкості в цифровому середовищі.»[8]

Критично важними інформаційними ресурсами в Linux є системні файли і каталоги, серед яких варто виділити /boot, /dev, /etc; бінарні файли оболонок, такі як bash, sh, zsh; конфігураційні файли для оболонок: для bash — /etc/profile, .bash_login, .profile, для csh — /etc/csh.cshrc, .rcshrc, для zsh — /etc/zsh, .zlogin, .zshenv. Також важливими є файли журналів системних подій, наприклад /dev/log, /var/log, access_log, auth.log, user.log, yum.log, а також бінарні файли для Docker і Kubernetes, зокрема /usr/local/bin/docker, /usr/local/bin/kubectl, runc, /usr/local/bin/notary. Це лише деякі з прикладів важливих файлів та каталогів,

що потребують пильної уваги для підтримання надійності та безпеки системи.[13]

Щоб захистити ці критично важливі ресурси в Linux, необхідно впровадити хоча б базовий комплекс заходів безпеки. Для наочності розглянемо захист на основі серверного дистрибутиву CentOS. Хоча CentOS вважається одним з найбезпечніших дистрибутивів, він не є абсолютно захищеним від можливих загроз. Тому важливо вжити заходів для забезпечення максимальної безпеки вашого середовища CentOS, так само як і для будь-якого іншого сервера або служби, які ви адмініструєте.

Першим кроком є забезпечення безпеки логінів через SSH, оскільки Secure Shell (SSH) є стандартним протоколом для віддаленого керування в CentOS. Неправильна конфігурація може призвести до серйозних проблем безпеки. Далі, важливо налаштувати брандмауер, оскільки CentOS використовує Firewalld, який дозволяє динамічно коригувати правила без відключення активних з’єднань. Регулярні оновлення системи також мають велике значення для захисту від нових вразливостей — використовуйте команду `yum update` для встановлення останніх оновлень.

Крім того, варто встановити та налаштувати SELinux (Security-Enhanced Linux), який надає механізми контролю доступу за допомогою політик безпеки. Він вже попередньо налаштований у CentOS. Не забувайте про системи виявлення вторгнень, такі як AIDE (Advanced Intrusion Detection Environment) або RKHunter, які можуть виявити підозрілу активність і попередити про можливі порушення. І на завершення, регулярне створення резервних копій може стати рятівним колом у разі критичних ситуацій; для автоматизації цього процесу можна використовувати інструменти, такі як rsync або Bacula.[14]

Таблиця 1.1 Аналіз існуючих аналогів

№	Назва	Моніторинг критичних файлів	Моніторинг процесів	Збір логів	Сповіщення	Автоматич на протидія	Автоматичне виправлення

1	Webmin	-	+	+	Email	-	-
2	The RHEL web console	-	+	+	Email	-	-
3	AIDE	+	-	-	Email, логи, текстові файли	-	-
4	Samhain	+	-	+	Email, логи,	-	-
5	Observium	-	-	-	Email	-	-
6	Monitorix	-	+	-	Email	-	-
7	Auditd	+	-	+	-	-	-
8	Tripwire	+	-	+	Email, месенджери	-	-

Висновки до першого розділу

У першому розділі було проведено аналіз існуючих систем моніторингу та управління серверами, а також розглянуто основні загрози та механізми безпеки, з якими стикаються сучасні компанії. Проведений огляд показав, що існуючі системи, такі як Webmin, The RHEL Web Console, AIDE, Samhain, Observium та Monitorix, Auditd, Tripwire мають свої переваги, проте не всі вони здатні забезпечити автоматичне відновлення важливих файлів або надійний захист від несанкціонованих змін у системі.

Розроблювана система вирішує цю проблему, забезпечуючи автоматичну протидію несанкціонованому редагуванню або пошкодженню критичних файлів та їх відновлення. Важливим аспектом є також автоматичне сповіщення адміністратора про спроби втручання, що дозволяє оперативно реагувати на загрози. У порівнянні з іншими системами, розроблювана система виділяється своєю здатністю ефективно забезпечувати цілісність важливих файлів та спрощувати управління безпекою серверів.

Загалом, результати аналізу підкреслюють важливість автоматизації процесів моніторингу та захисту даних, що робить нову систему цінним інструментом для адміністраторів, особливо у невеликих організаціях, де обмежені ресурси на постійний контроль і управління серверною інфраструктурою.

Розділ 2. СТВОРЕННЯ СИСТЕМИ КОНТРОЛЮ ЦІЛІСНОСТІ КРИТИЧНИХ ІНФОРМАЦІЙНИХ РЕСУРСІВ

2.1 Налаштування inotify на слідкування за критичними файлами в дистрибутиві CentOS

Проект CentOS — це зусилля вільного програмного забезпечення, керовані спільнотою, що зосереджені на створенні багатофункціональної базової платформи для спільнот з відкритим кодом.[8]

Inotify API забезпечує механізм моніторингу файлової системи події. Inotify можна використовувати для моніторингу окремих файлів або для монітор каталогів. Коли каталог відстежується, inotify буде події повернення для самого каталогу та для файлів усередині каталог.[9]

Окрім стандартного модулю ядра inotify, буде використовуватись утиліта inotify-tools. Inotify-tools є набором комбінованих програм для Linux, щоб забезпечити простий interface для inotify.[15]

Список файлів, що підлягатимуть моніторингу:

/etc/passwd — у цьому файлі зберігатиметься важлива інформація, яка потрібна під час входу в системи Gnu/Linux.. Іншими словами, там буде зберігатися інформація, що стосується облікових записів користувачів.[16]

/etc/shadows — містить захешовані паролі користувачів та іншу інформацію, пов’язану з безпекою користувачів.

/**etc/group** — файл, що містить інформацію про групи в системі та їх учасників.[17]

/etc/fstab — файл конфігурацій, що містить інформацію про диски та розділи, які будуть вмонтовані під час завантаження системи.[18]

/etc/hosts — використовується для ручного задання ip-адрес для доменних імен, що дозволяє локально дозволити доступ до DNS-імені без запиту до DNS-серверу. Файл має більший пріоритет ніж звернення до DNS-серверів.[19]

/etc/resolv.conf — визначає DNS-сервери, які використовує система для дозволу доменних імен.[20]

/etc/hostname — ім’я хоста системи, що відображається в командній строці та використовується для мережевої взаємодії.[21]

/etc/sysctl.conf — файл конфігурацій параметрів ядра, таких як мережеві налаштування, управління пам’яттю та параметрами безпеки. Навички користування цією утилітою є необхідними для системного інженера, щоб досягти оптимальних показників роботи системи.[22]

/etc/rc.local — запускається при старті системи і часто використовується для виконання кастомних скриптів або команд, які потрібно виконувати автоматично.[23]

/etc/crontab — файл для налаштування періодичних задач за допомогою cron, що дозволяє запускати скрипти або команди за розкладом.[24]

/etc/sudoers — це файл, який адміністратори Linux використовують для призначення системних прав користувачам системи. Це дозволяє дії користувачів, що можна робити, а що не можна.[25]

/etc/ssh/sshd_config — файл налаштувань для сервера SSH, що використовується для налаштувань доступу по SSH і управлінні безпекою віддалених підключень.[26]

/var/log/syslog — основний системний лог, в який запусуються загальні події та помилки, що відбуваються в системі.[27]

/var/log/dmesg — зберігає журнал повідомлень ядра, записанних під час завантаження та під час роботи системи.[28]

/boot/grub2/grub.cfg — файл конфігурацій завантажувача GRUB, що містить параметри завантаження ядра та інші налаштування, використовувані для завантаження системи.[29]

Скрипт ,що буде відповідати за моніторинг цих файлів та записувати логи матиме наступний вигляд:

```
#!/bin/bash

# Список файлів для мониторингу
FILES_TO_MONITOR=(
```

```
    "/etc/passwd"
    "/etc/shadow"
    "/etc/group"
    "/etc/fstab"
    "/etc/hosts"
    "/etc/resolv.conf"
    "/etc/hostname"
    "/etc/sysctl.conf"
    "/etc/rc.local"
    "/etc/crontab"
    "/etc/sudoers"
    "/etc/ssh/sshd_config"
    "/var/log/messages"
    "/var/log/boot.log"
    "/boot/grub2/grub.cfg"
)

# Папка для зберігання логів
LOG_DIR="/home/user/watch_logs"
mkdir -p "$LOG_DIR"

# Функція для мониторинга одного файлу
monitor_file() {
    local file_path="$1"
    local file_name=$(basename "$file_path")

    # Запуск inotifywait и запис подій в окремий лог-файл
    inotifywait -m -e open -e modify -e move -e delete -e delete_self -e moved_from -e close_write -e close_nowrite "$file_path" |
    while read -r line; do
```

```
        echo "$(date '+%Y-%m-%d %H:%M:%S') $line" >> "$LOG_DIR/inotify_${file_name}_$(date '+%Y-%m-%d').log"
    done
}

# Запуск мониторингу для кажного файла в фоновому процесі
for file in "${FILES_TO_MONITOR[@]}"; do
    monitor_file "$file" &
done

# Очікування всіх процесів
wait
```

Усі ці команди будуть розміщені в спеціально відведеному BASH-файлі (startup_logs.sh), який запускатиметься при завантаженні або перезавантаженні системи. Такий спосіб створення системи моніторингу критично важливих інформаційних ресурсів без використання сторонніх програм, що потребують нагляду та прав при використанні, підвищує безпеку системи. Програми, які встановлюються в систему, мають недолік — вони можуть містити вразливості, які хакери можуть використати при нападі на систему. Використовуючи вбудовані можливості самої системи Linux, можна зменшити таку ймовірність.

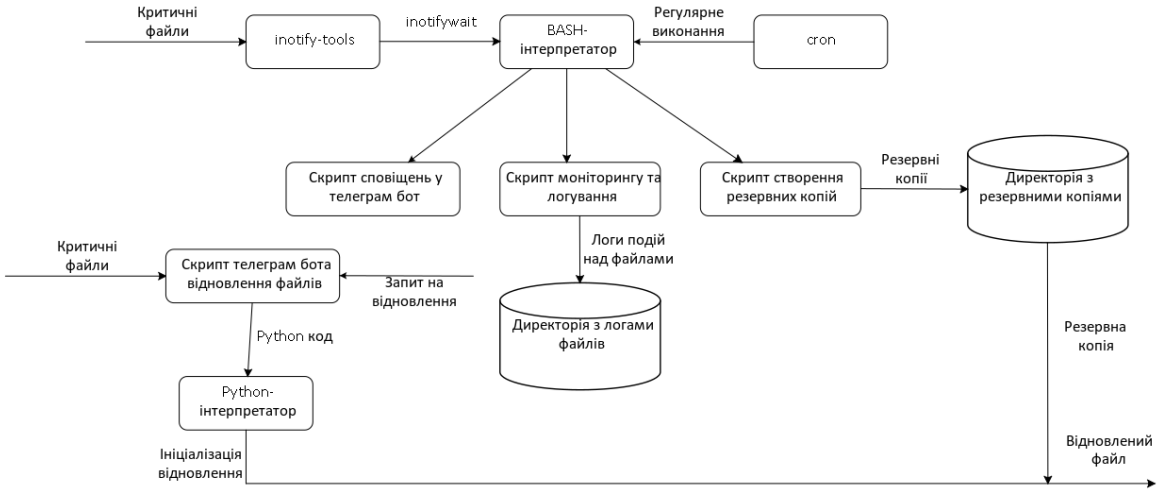

Рисунок 2.1 – Структурне рішення

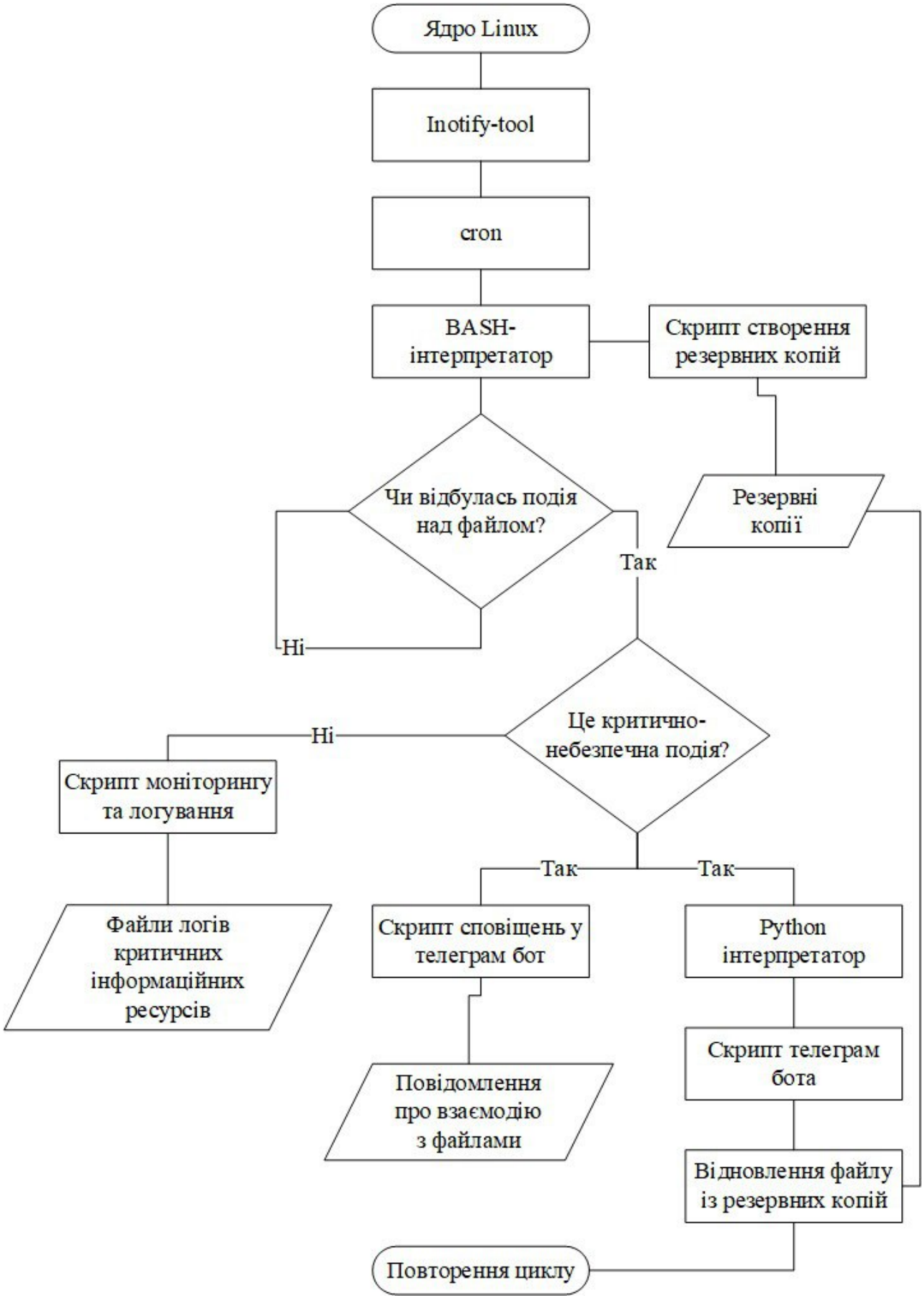

Рисунок 2.2 – Алгоритм роботи системи контролю цілісності

2.2 Повідомлення адміністратора про найважливіші дії над критичними файлами

Створивши ще один скрипт-файл та додавши в нього telegram API для роботи з ботами, можна створити систему, що буде повідомляти уповноважену особу про найнебезпечніші дії з критичними інформаційними ресурсами, такими як модифікація, видалення та переміщення.

```
#!/bin/bash

# Parameters for sending notifications to Telegram
BOT_TOKEN="7753372278:AAESXqaS-B_T-ywK6uPlrgJvj-SqZ1ZQpak"
CHAT_ID="875791968"

# Email parameters
EMAIL_TO="dfyz.ifrehf@gmail.com"
EMAIL_SUBJECT="Notification from Bash Script"

# Function to send messages to Telegram
send_to_telegram() {
    local message="$1"
    curl -s -X POST "https://api.telegram.org/bot$BOT_TOKEN/sendMessage" \
        -d chat_id="$CHAT_ID" \
        -d text="$message"
}

# Function to send messages to email
send_to_email() {
    local message="$1"
    echo "$message" | mail -s "$EMAIL_SUBJECT" "$EMAIL_TO"
}
```

```
# List of files to monitor
FILES_TO_MONITOR=(
    "/etc/passwd"
    "/etc/shadow"
    "/etc/group"
    "/etc/fstab"
    "/etc/hosts"
    "/etc/resolv.conf"
    "/etc/hostname"
    "/etc/sysctl.conf"
    "/etc/rc.local"
    "/etc/crontab"
    "/etc/sudoers"
    "/etc/ssh/sshd_config"
    "/var/log/messages"  # Replacement for syslog
    "/var/log/boot.log"   # Replacement for dmesg
    "/boot/grub2/grub.cfg"
)

# Start inotifywait for each file
inotifywait -m -e modify -e move -e delete "${FILES_TO_MONITOR[@]}" |
while read path action file; do
# Skip modify event for /var/log/messages
    if [[ "$path$file" == "/var/log/messages" && "$action" == "MODIFY" ]];
then
        continue
    fi
    # Create message with timestamp
```

```
    message="$(date '+%Y-%m-%d %H:%M:%S') - Event: $action. File: $file at $path"

    # Log the event to a file
    echo "$message" >> "/home/user/watch_logs/inotify_log_$(date '+%Y-%m-%d').log"

    # Send the message to Telegram
    send_to_telegram "$message"

    # Send the message to Email
    send_to_email "$message"
done
```

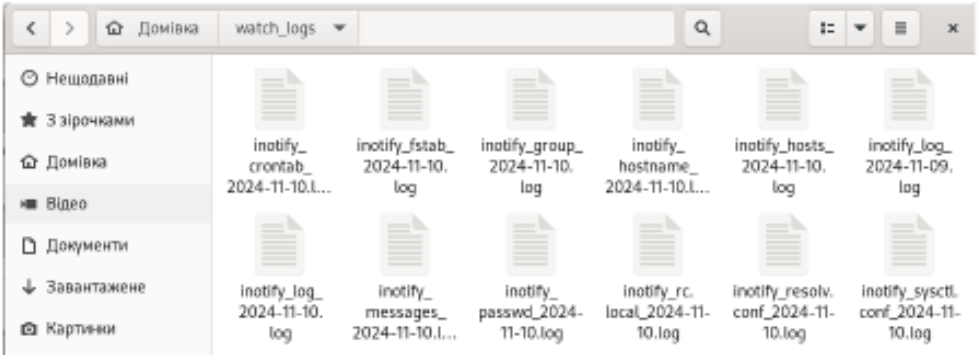

Рисунок 2.3 – Логи створені скриптом

Файли логів збираються в різні файли, відповідно до тих за якими ведеться моніторинг (рис. 2.3). Це потрібно для покращення читабельності логів, оскільки деякі файли генерують занадто багато подій при їх моніторингу, тому для загального покращення читабельності вони розбиті на різні файли.

Такий спосіб повідомлення адміністратора вимагає додаткового встановлення утиліти curl, яку можна встановити на дистрибутив CentOS за допомогою команди ***sudo yum install curl***.

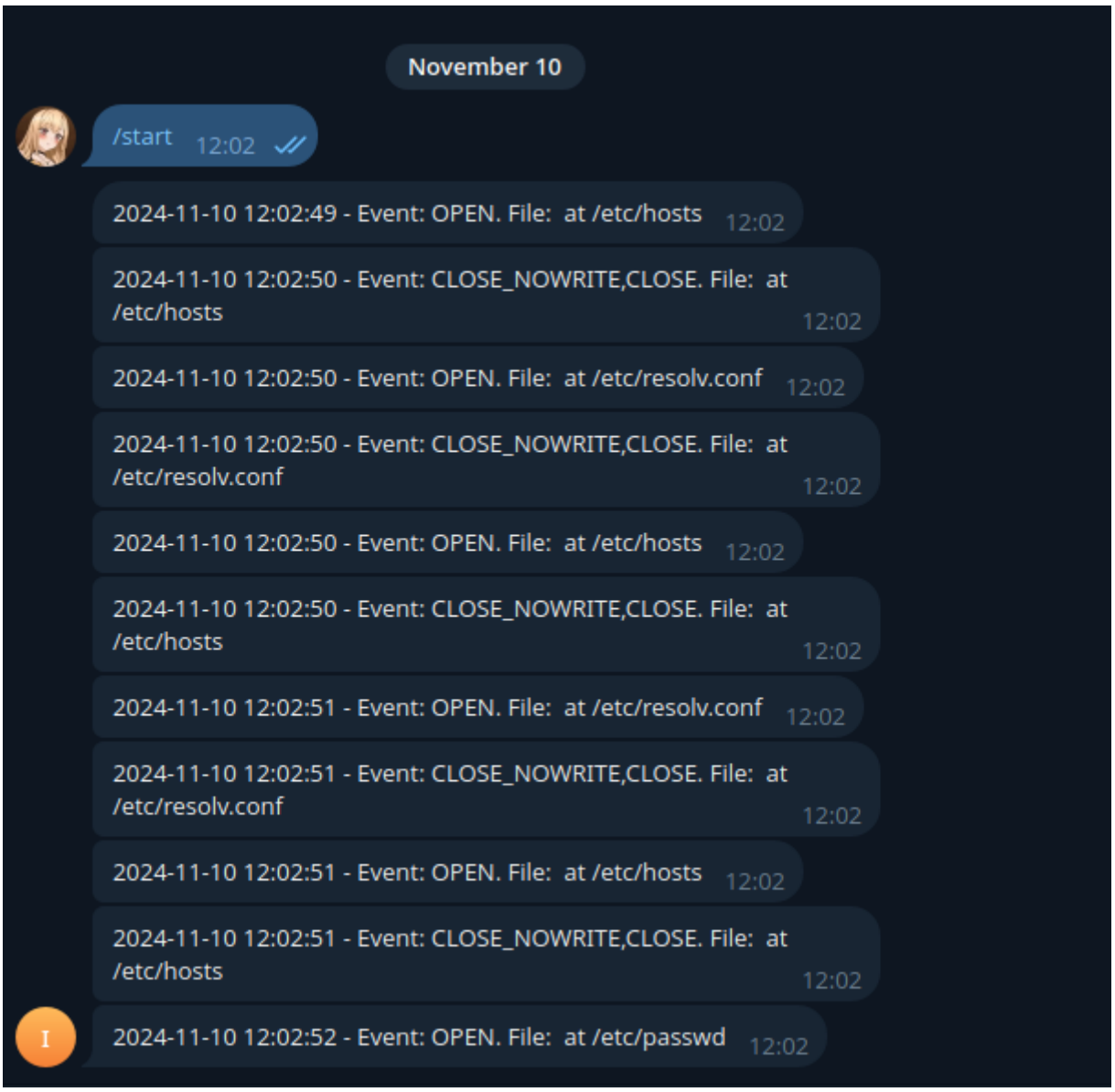

Рисунок 2.4 – Приклад роботи скрипта сповіщення

Важливою частиною системи контролю цілісності є також створення та відновлення системи з бекапів. Bash-скрипт для створення бекапів має наступний вигляд:

```
#!/bin/bash
# Директория для зберігання резервних копій
backup_dir="/path/to/.backups"

# Список файлів для резервного копіювання
files_to_backup=("/etc/passwd"
    "/etc/shadow"
    "/etc/group"
```

```
    "/etc/fstab"
    "/etc/hosts"
    "/etc/resolv.conf"
    "/etc/hostname"
    "/etc/sysctl.conf"
    "/etc/rc.local"
    "/etc/crontab"
    "/etc/sudoers"
    "/etc/ssh/sshd_config"
    "/var/log/messages"  # Replacement for syslog
    "/var/log/boot.log"   # Replacement for dmesg
    "/boot/grub2/grub.cfg")

# Створюємо резервну копію з міткою часу
timestamp=$(date '+%Y%m%d')
for file in "${files_to_backup[@]}"; do
    filename=$(basename "$file")
    cp "$file" "$backup_dir/${filename}_$timestamp.bak"
done
```

Завдяки утиліті cron цей скрипт виконується щоденно в 10 годин дня.

Для відновлення файлів із бекапів використовується інший телеграм бот(Рис. 2.5)

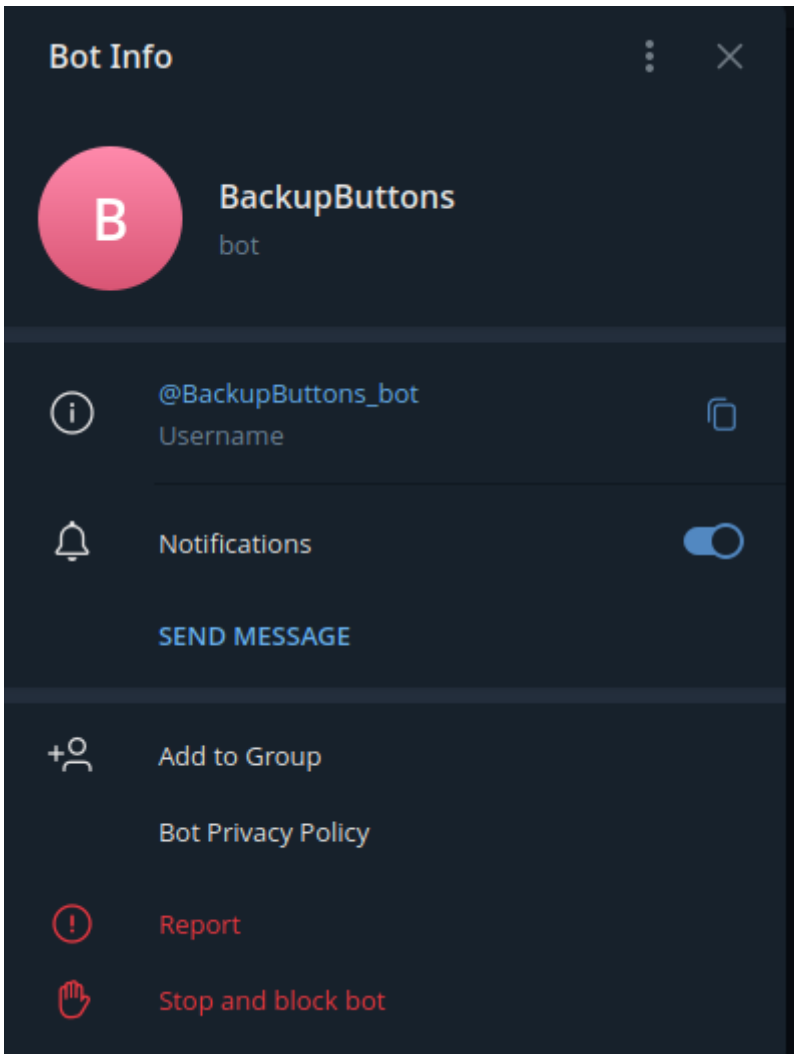

Рисунок 2.5 — Телеграм бот з кнопками для відновлення файлів з бекапів

Резервні копії критично важливих файлів повинні бути максимально ізольовані від звичайних користувачів, тому потрібно забрати доступ до директорії з бекапами у кожного користувача окрім root. Для цього потрібно зробити root користувача власником директорії та змінити права доступу за допомогою таких команд(Рис. 2.5):

sudo chown root /home/user/.backups #зміна власника

sudo chmod 700 /home/user/.backups #зміна прав доступу

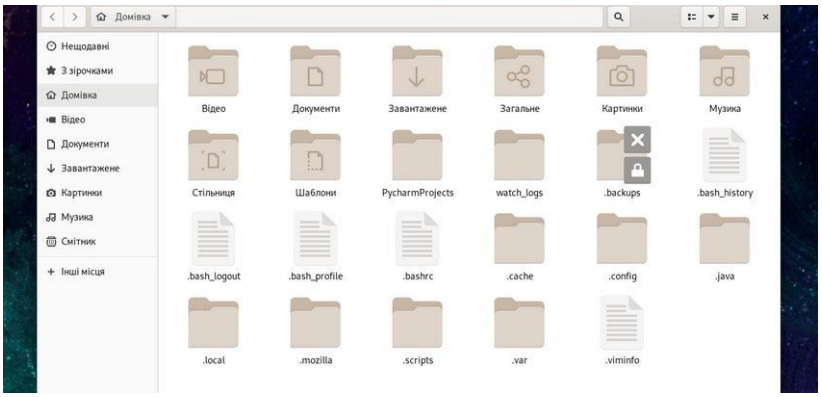

Рисунок 2.6 — Директорія з резервними копіями

2.3 Відновлення файлів з резервних копій за допомогою телеграм боту

Взаємодія з файлами резервних копій відбувається за допомогою другого телеграм боту (Рис. 2.7).

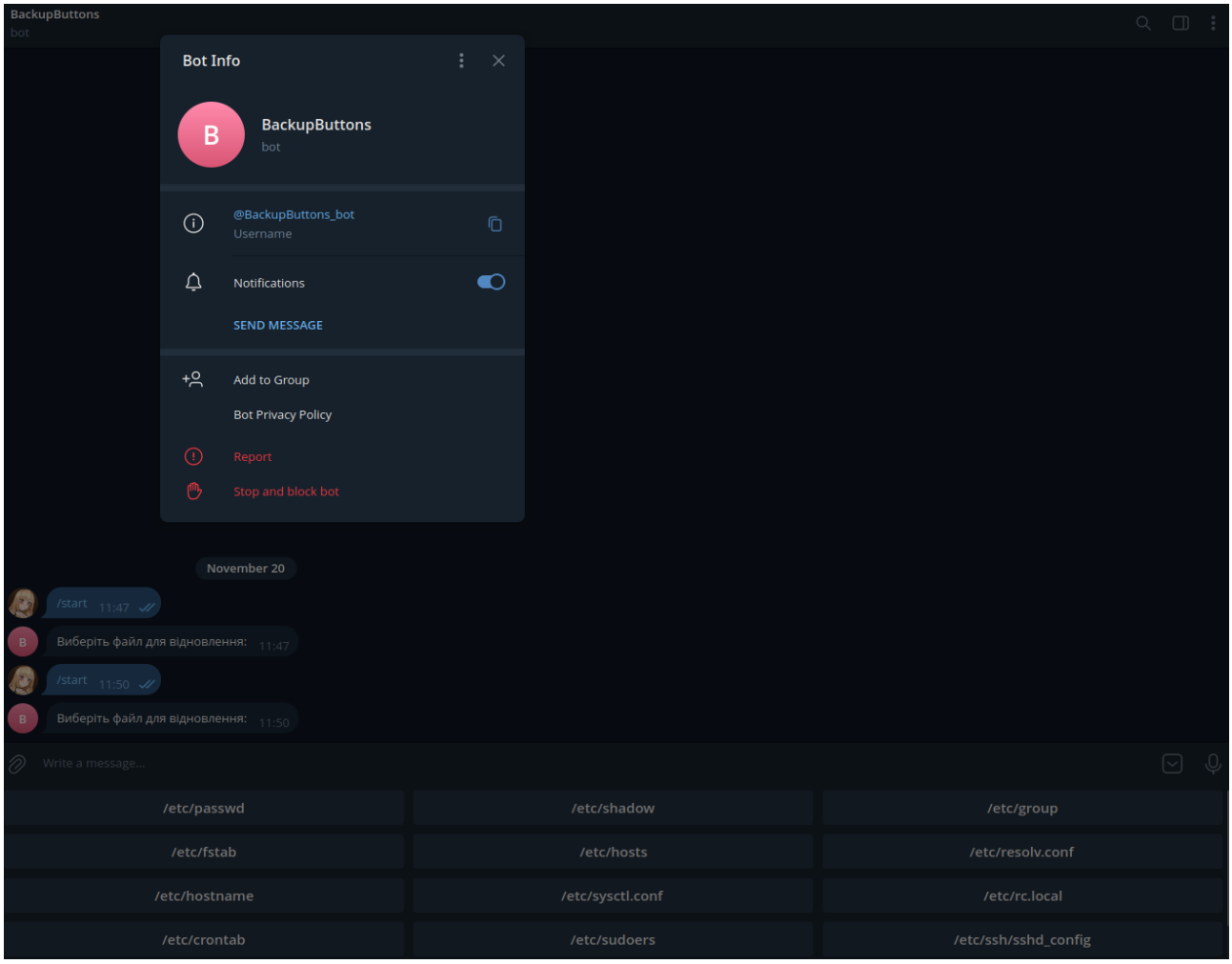

Рисунок 2.7 — Телеграм бот для відновлення файлів з резервних копій

Функціонал бота реалізовано на мові Python з наступним вмістом:

```
import telebot
import shutil
import os
```

```
from telebot import types
from datetime import datetime, timedelta

bot = telebot.TeleBot('7903074248:AAGzLCoEcwB_akBj3wX8EEVC2x5GpsI7ZbU')

files = ["/etc/passwd",
    "/etc/shadow",
    "/etc/group",
    "/etc/fstab",
    "/etc/hosts",
    "/etc/resolv.conf",
    "/etc/hostname",
    "/etc/sysctl.conf",
    "/etc/rc.local",
    "/etc/crontab",
    "/etc/sudoers",
    "/etc/ssh/sshd_config",
    "/var/log/messages",
    "/var/log/boot.log",
    "/boot/grub2/grub.cfg"]

def recover(file):
    try:
        # Формуємо шлях з інформацією
        target_directory = os.path.basename(file)
        timestamp = datetime.now().strftime('%Y%m%d')
        source_file = f"/home/user/.backups/{target_directory}_{timestamp}.bak"

        # Перевіряємо, чи існує шлях
        while not os.path.exists(source_file):
            date_object = datetime.strptime(timestamp, '%Y%m%d') - timedelta(days=1)
            timestamp = date_object.strftime('%Y%m%d')
            target_file = file

        # Копіюємо вміст файлу
        if target_file:
            shutil.copy(source_file, target_file)

    except Exception as e:
        print('Error')


@bot.message_handler(commands=['start'])
def start(message):
```

```
    buttons = types.ReplyKeyboardMarkup(resize_keyboard=True)
    for i in range(0, len(files), 3):
        row = [types.KeyboardButton(file) for file in files[i:i + 3]]
        buttons.row(*row)


    bot.send_message(
        message.chat.id,
        "Виберіть файл для відновлення:",
        reply_markup=buttons
    )
    bot.register_next_step_handler(message, choice)

def choice(message):
    for i in range(len(files)):
        if message.text == files[i]:
            recover(files[i])

bot.polling(non_stop=True)
```

2.4 Використання утиліти cron для автоматичного запуску скриптів моніторингу, телеграм ботів та створення бекапів

Для додавання запису в cron можна використати вбудований в систему редактор nano. Для безперешкодної роботи всіх скриптів вони дадаються у cron від імені root користувача.

Автозапуск скриптів при старті системи можна налаштувати за шаблоном @reboot /path/to/script.sh (Рис. 2.7). Для того щоб відкрити заплановані задачі в крон потрібно в терміналі виконати наступну команду:

```
su root
crontab -e
```

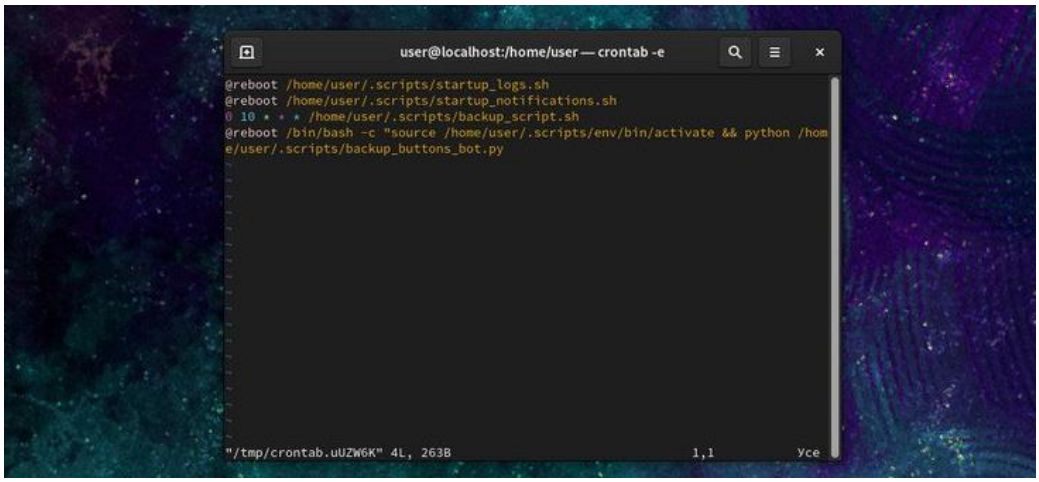

Рисунок 2.8 — Вміст команд на автозапуску в cron

Висновки до другого розділу

Було розроблено систему контролю цілісності критичних інформаційних ресурсів операційної системи Linux. Для її побудови використовувався дистрибув CentOS встановлений як Server + GUI. Скрипти були написані на мові BASH та python(функціонал телеграм ботів). Наведено рисунки, що демонструють робото здатність системи а також код із самих скриптів з пояснюючими коментарями. Безперешкодна робота роботи скриптів при старті системи забезпечується cron задачами від root користувача, саме тому перед виконанням команди ***crontab -e*** потрібно зайти в root користувача за допомогою команди ***su root.***

Розроблена система забезпечує ефективний контроль над критично важливими файлами для працездатності системи Linux, використовуючи мінімум обчислювальних ресурсів. Вона також мінімізує ризики появи потенційних вразливостей у системі, оскільки не потребує постійно активних сервісів, що працюють на портах сервера.

Розділ 3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ КОНТРОЛЮ ЦІЛІСНОСТІ КРИТИЧНИХ ІНФОРМАЦІЙНИХ РЕСУРСІВ

3.1 Експериментальна демонстрація системи контролю цілісності

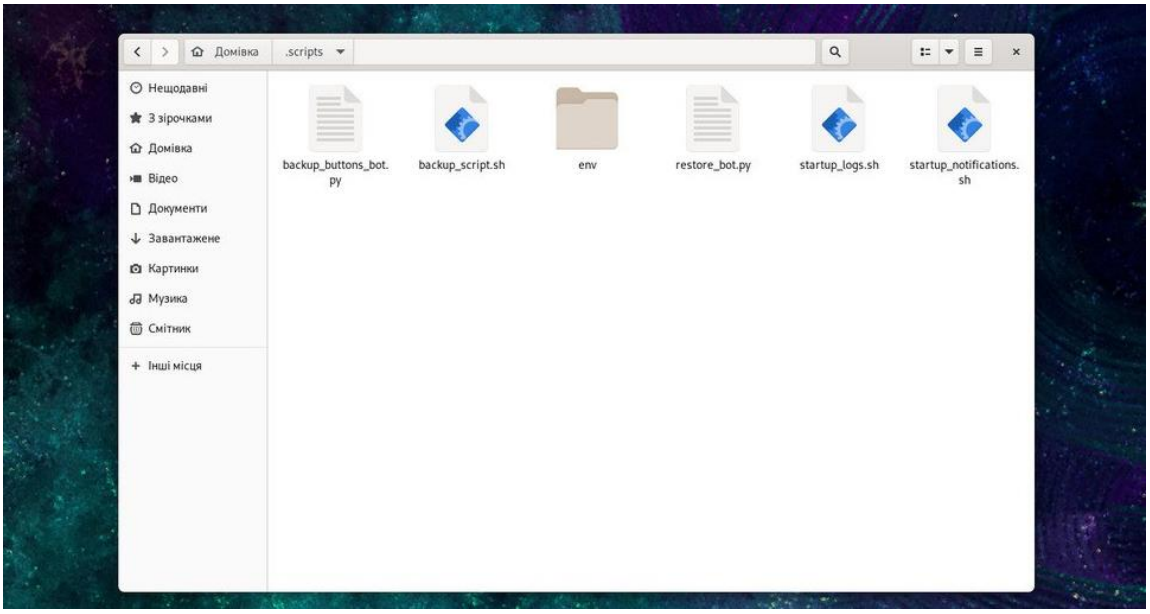

Рисунок 3.1 – Директорія зі скриптами системи

Файли із директорії зі скриптами (Рис 3.1) виконують весь необхідний для системи функціонал, такий як ведення логів для кожного файлу із датою в назві файлу та вмісту кожного файлу (Рис 3.2).

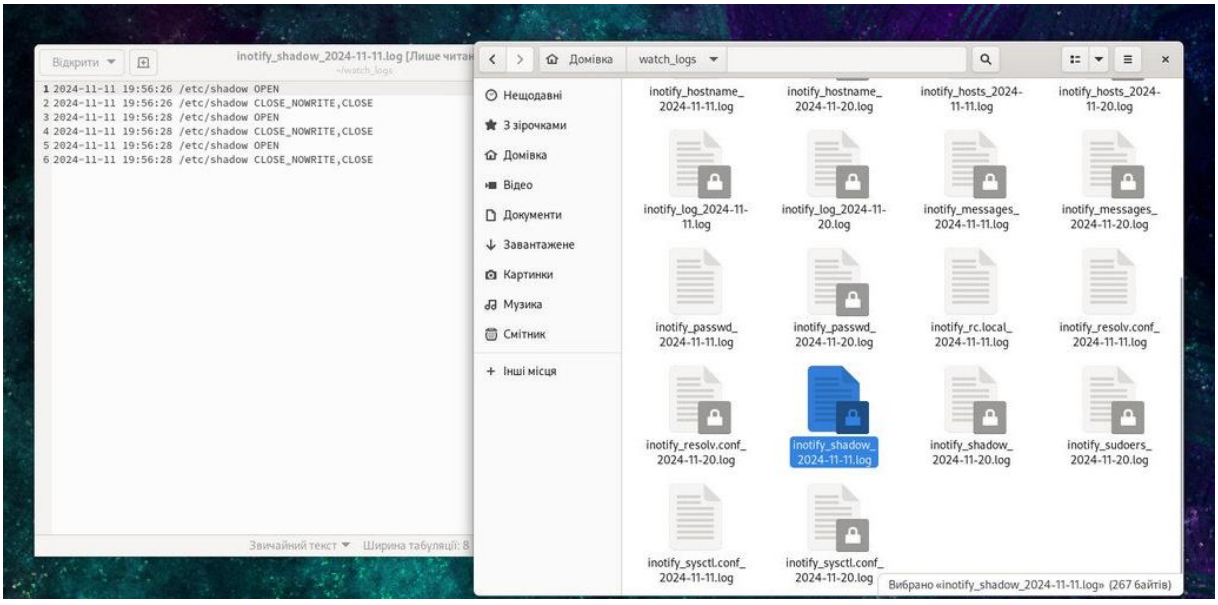

Рисунок 3.2 — Файли логів та їх вміст

Приклад роботи бота та вигляд повідомлень що він надсинає наведено далі (Рис 3.3), а приклад роботи бота з кнопками було наведено у минулому розділі та продемонстровано(Рис. 2.6).

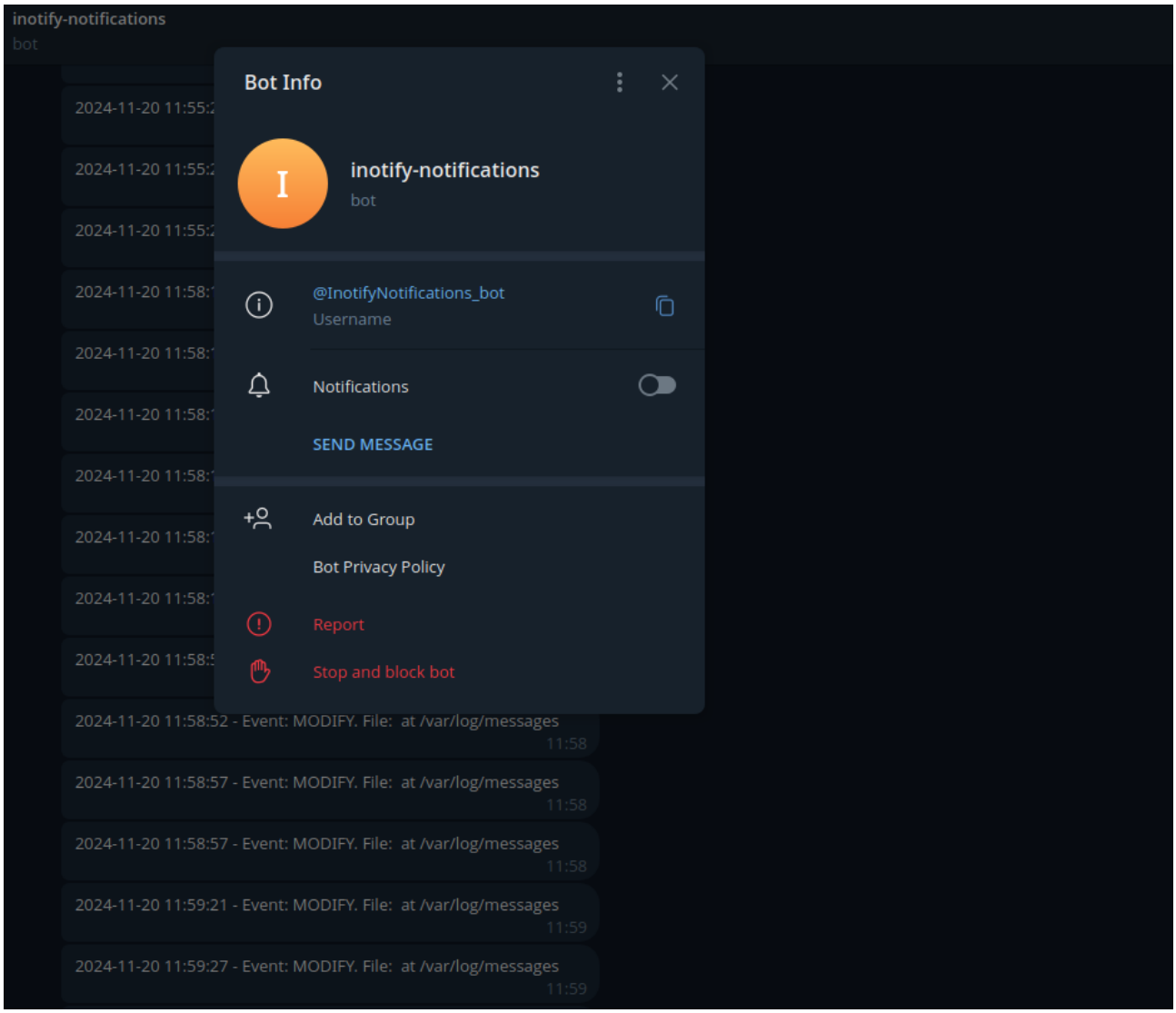

Рисунок 3.3 – Телеграм бот зі сповіщеннями подій

3.2 Порівняльний аналіз запропонованого рішення з відомими

Webmin чудова програма, яка має великий функціонал але цей функціонал в контексті захисту критичних інформаційних об’єктів є надлишковим. Веб-інтерфейс що надається цим застосунком потребує входу від root користувача і світиться у світ через 10000 порт, що несе за собою ряд потенційних небезпек. У випадку компрометації пароля від root користувача, зловмисник отримає занадто зручний графічний засіб для керування системою. (рис. 3.4).

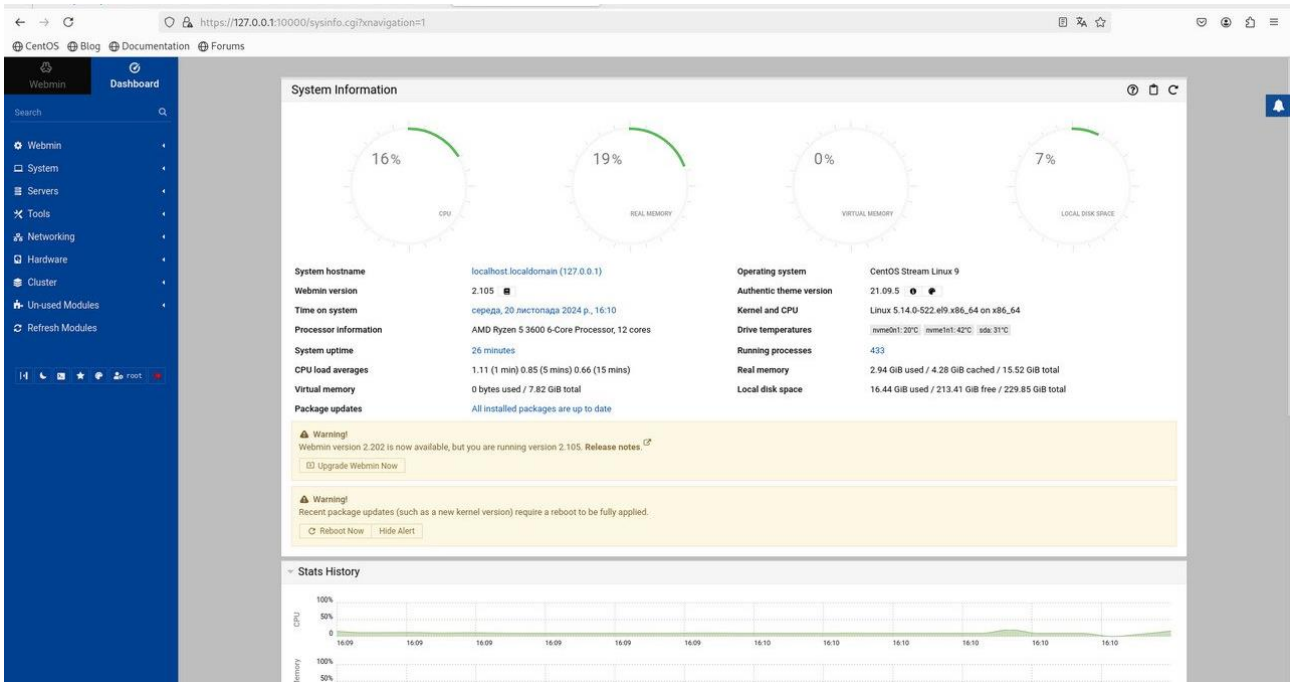

Рисунок 3.4 – Веб-інтерфейс webmin

The RHEL web console має ті ж самі мінуси що і аналог вказаний вище, відрізняється лише тим що світиться в мережу на порті 9090 та має можливість входу не від root користувача. До обох застосунків важливо додати що вони вимагають встановлення з великою кількістю залежностей що потребує від адміністратора знань, а від комп'ютерного пристрою фізичної пам'яті.

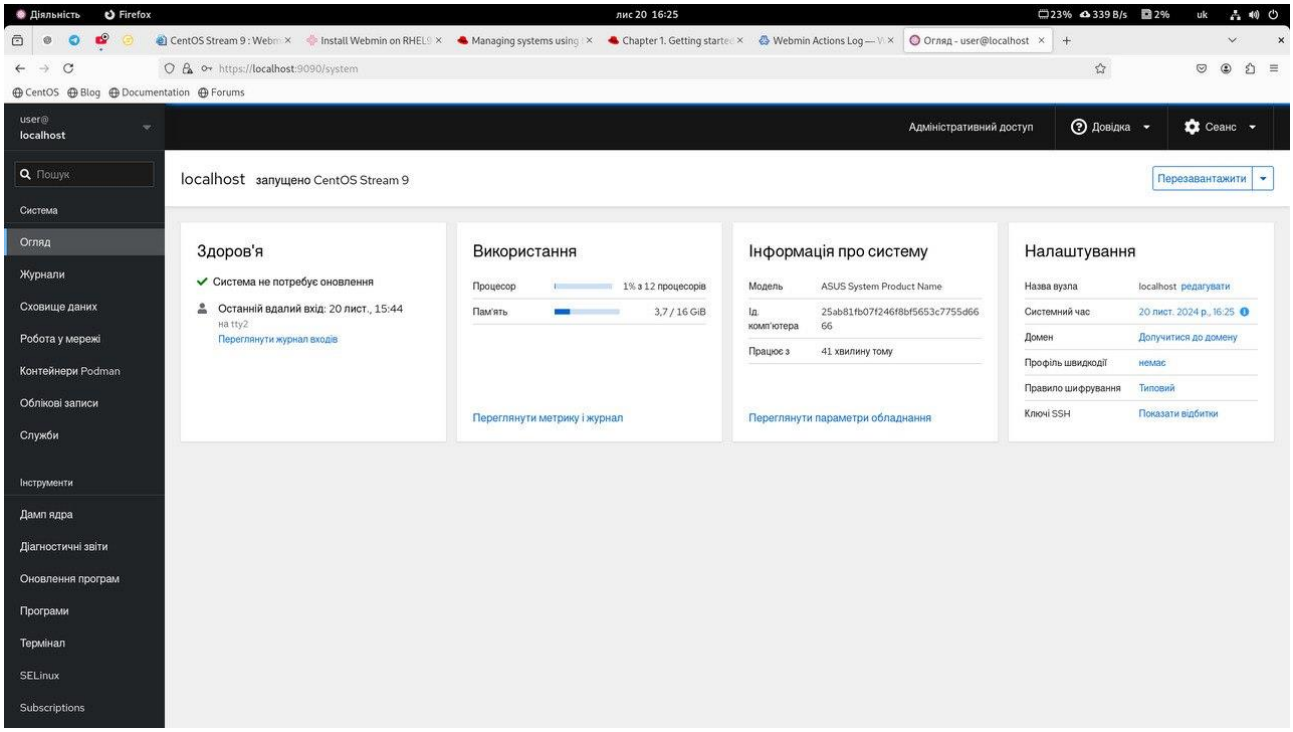

Рисунок 3.6 – Веб-інтерфейс The RHEL web console

AIDE загалом схожа на розроблену систему, але не має достатньої автоматизації і функціоналу по виправленню при вторгненні.

Samhain є слабоактуальною, оскільки останнє оновлення цієї програми було 4 роки тому і остання підтримувана версія дистрибутива CentOS 8 (Рис. 3.7).

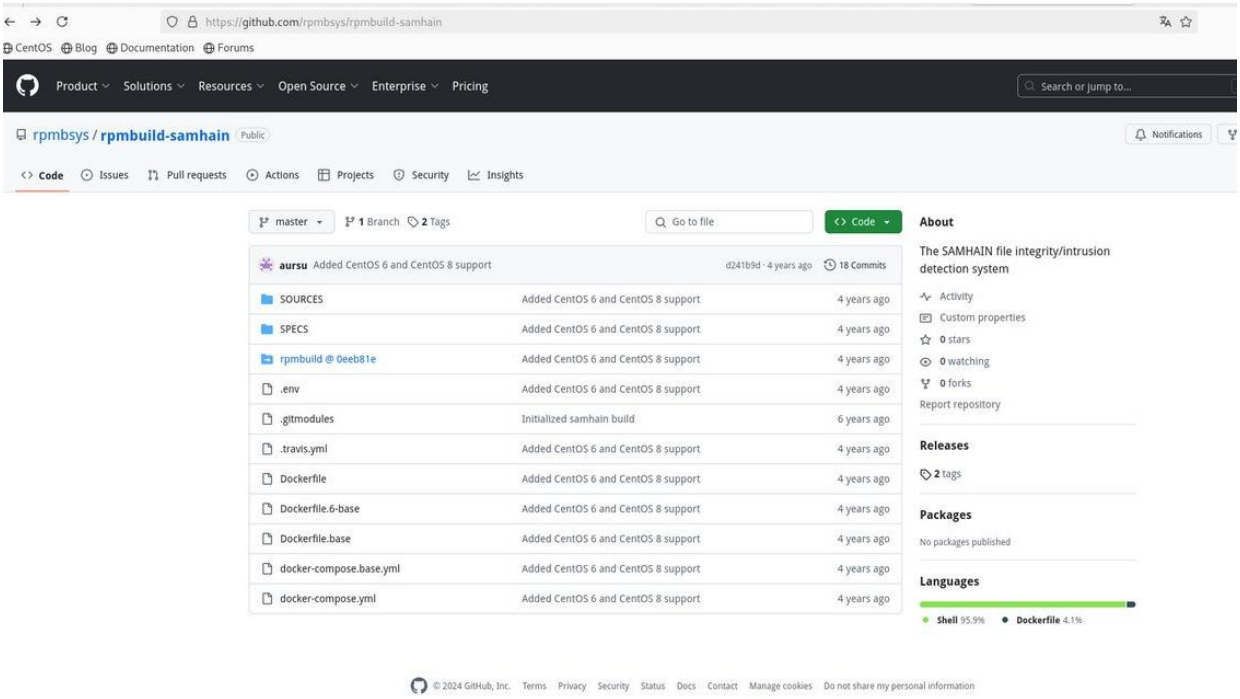

Рисунок 3.7 – GitHub сторінка Samhain

Observium має досить великий список залежностей та вимагає від користувача освоєння свого функціоналу. Через свій функціонал має недолік у вигляді високого рівня навантаження на сервер.

Monitorix тяжко встановлюється оскільки вимагає великої кількості залежностей таких як інтерпретатор мови програмування Perl та додаткових утиліт для нього, а отриманий функціонал це досить детальний моніторинг системних ресурсів та ще одна потенційна вразливість у вигляді постійно активного порту 8080 (Рис. 3.8).

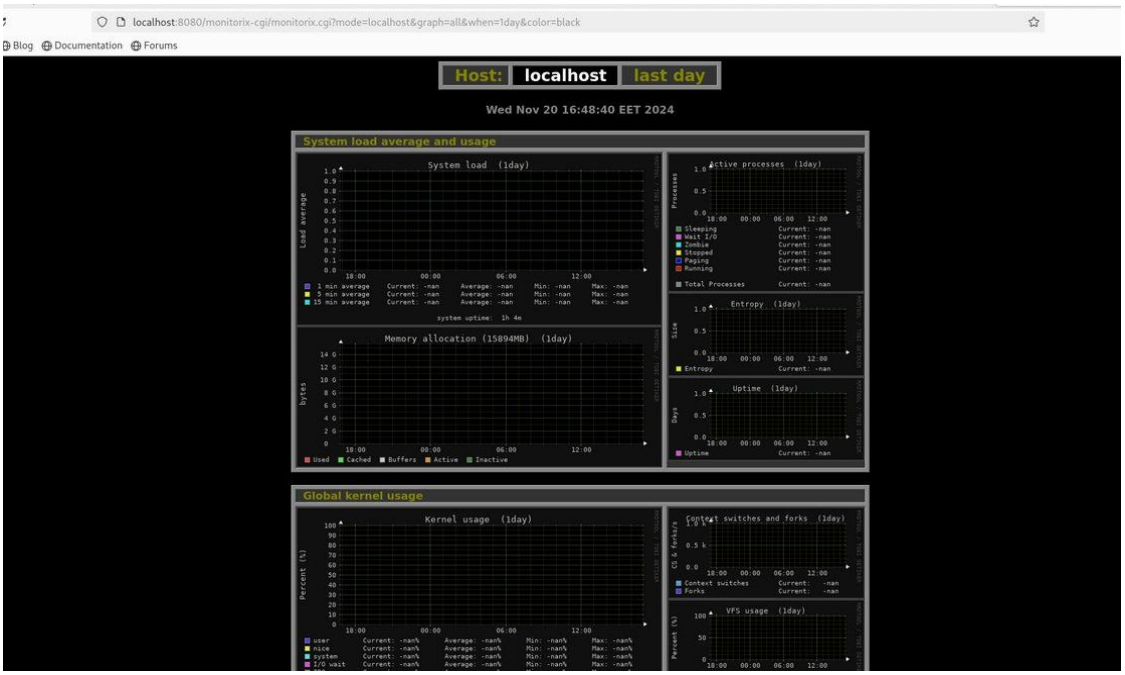

Рисунок 3.8 – Веб-інтерфейс Monitorix

Auditd є просто вбудованою утилітою в дистрибутив CentOS 9 і відповідає лише за ведення логів, що мають майже не читабельний вигляд (Рис. 3.9).

Рисунок 3.9 – Файл логів зібраний за допомогою Auditd

Останній із протестованих систем контролю цілісності — Tripwire. Ця система найбільш подібна до розробленої і з досліджених недоліків знайдено тільки складність налаштувань, що призводить до великої кількості помилкових спрацювань та високе навантаження на систему.

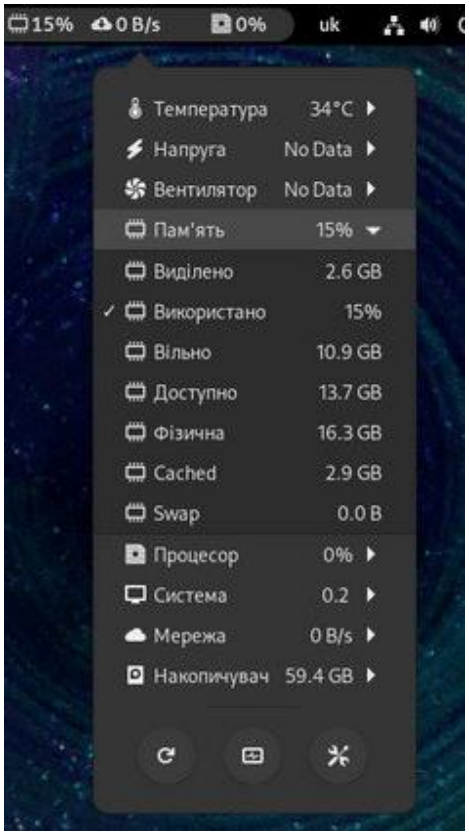

Рисунок 3.10 – Використання системних ресурсів

Удосконалена система контролю цілісності критичних інформаційних ресурсів на базі інструменту ядра Linux inotify-tools використовує близько 5% оперативної пам’яті (операційна система використовує 10%) із загальних 16 ГБ, що становить приблизно 800 МБ (Рис. 3.10). Такий рівень навантаження було досягнуто завдяки реалізації лише необхідного функціоналу. Робота системи продемонстрована на прикладі критично важливого файлу passwd.

```
2 nobody:!*:19735::::::
3 dbus:!*:19735::::::
4 bin:!*:19735::::::
5 daemon:!*:19735::::::
6 mail:!*:19735::::::
7 ftp:!*:19735::::::
8 http:!*:19735::::::
9 systemd-coredump:!*:19735::::::
10 systemd-network:!*:19735::::::
11
```

Рисунок 3.11 – Модифікований файл

Під час такої модифікації у файлах логів записуються всі події, що відбуваються над файлами. Окрім цього, до Telegram-бота надсилаються сповіщення з відповідною інформацією. (Рис. 3.12)

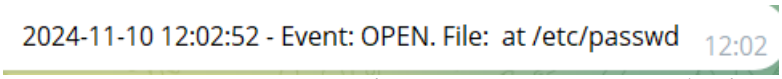

Рисунок 3.12 – Сповіщення в телеграм боті

Натиск кнопки (Рис. 3.13) з потрібним файлом в іншому телеграм боті відновлює вміст файлу до модифікації.(Рис. 3.14)

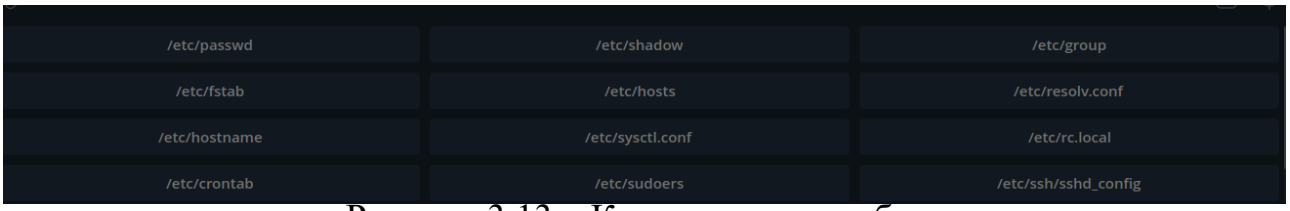

Рисунок 3.13 – Кнопки телеграм бота

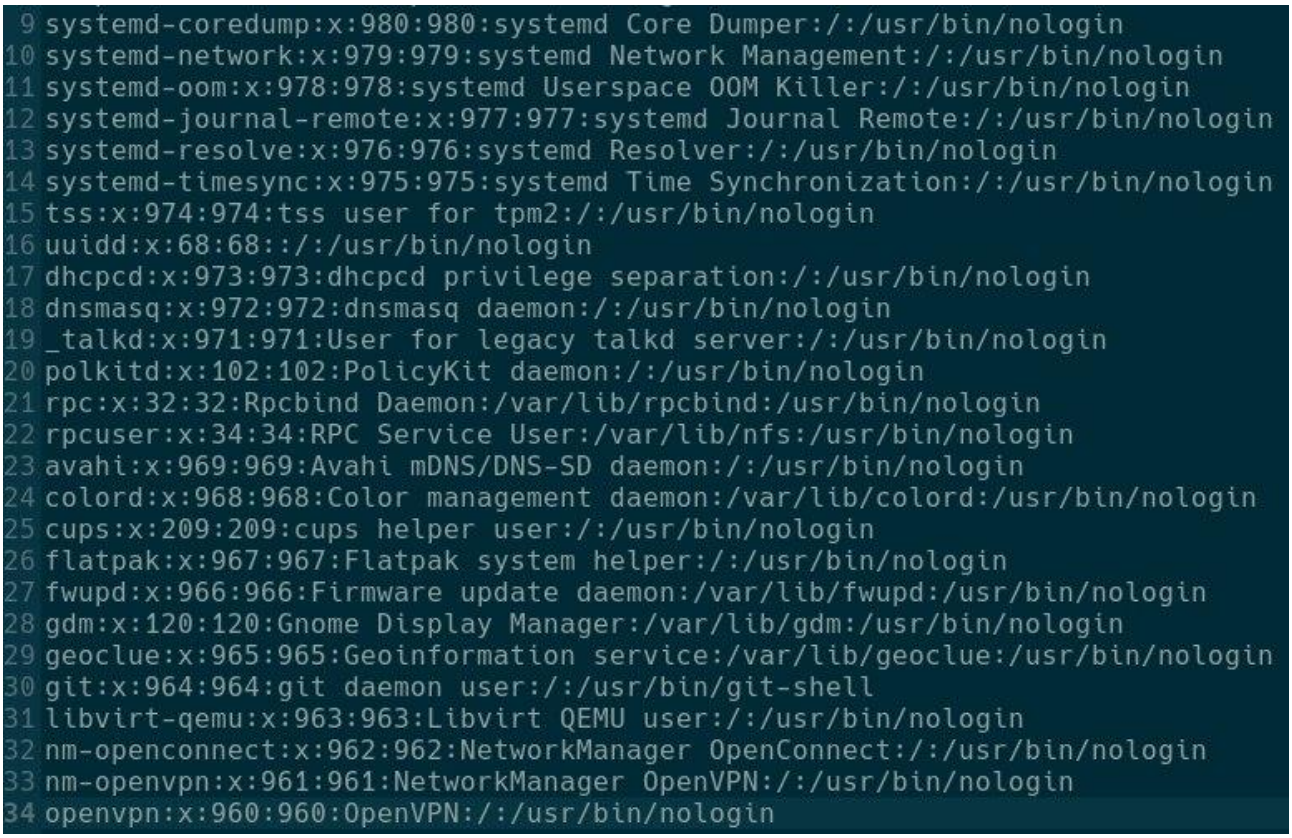

Рисунок 3.14 – Відновлений вміст файлу

Таблиця 3.1 Порівння аналогів зі створеною системою

№	Назва	Моніторинг критичних файлів	Моніторинг процесів	Збір логів	Сповіщення	Автоматична протидія	Автоматичне виправлення
1	Webmin	-	+	+	Email	-	-
2	The RHEL web console	-	+	+	Email	-	-
3	AIDE	+	-	-	Email, логи, текстові файли	-	-
4	Samhain	+	-	+	Email, логи,	-	-
5	Observium	-	-	-	Email	-	-
6	Monitorix	-	+	-	Email	-	-

7	Auditd	+	-	+	-	-	-
8	Tripwire	+	-	+	Email, месенджери	-	-
9	Удосконалена система	+	-	+	Телеграм, Email	-	+

Висновки до третього розділу

У цьому розділі було детально представлено розроблену систему та проведено практичне порівняння з існуючими аналогами на всіх етапах — від встановлення до використання. Виявлено низку недоліків аналогів:

- значна кількість залежностей
- високі вимоги до системних ресурсів
- недостатність, або надлишковість функціоналу.

Запропонована удосконалена система контролю цілісності критичних інформаційних ресурсів операційної системи Linux поєднує оптимальний функціонал із низькими вимогами до ресурсів. Зокрема, експериментально визначено, що система споживає лише 800 МБ оперативної пам’яті залежно від навантаження та мінімально впливає на інші апаратні компоненти (близько 5%).

Експериментальне тестування аналогів дозволило оцінити їхній функціонал і вплив на систему, що підтвердило конкурентоспроможність запропонованого рішення.

ВИСНОВОК

В кваліфікаційній роботі розроблено систему контролю цілісності критичних інформаційних ресурсів операційної системи Linux на базі дистрибутива CentOS та вбудованих функцій в систему з мінімальним використанням сторонніх програм та утиліт. Використання такої системи контролю забезпечує мінімальне використання системних ресурсів на фоні аналогічних застосунків, що встановлюються на систему та потребують оновлень. Такі застосунки можуть мати вразливості в собі, що може зашкодити працездатності системи, тому використання функцій самого Linux дистрибутива є пріоритетним та раціональним рішенням.

Проведено аналіз існуючих систем контролю цілісності критичних інформаційних ресурсів та порівняння їх функціоналу, що дало розуміння про особливості роботи існуючих аналогів, та напрямок для подальшої розробки власної системи.

Удосконалена система контролю використовує такі інструменти, як Python, BASH і Telegram API дозволяє ефективно виконувати свої завдання з мінімальним споживанням системних ресурсів і не містить надлишкового функціоналу, що міг би становити загрозу безпеці та цілісності операційної системи.

Експериментальне тестування розробленої системи та існуючих аналогів демонструє наявний функціонал аналогів, їх недоліки, а також ілюструє принцип роботи удосконаленої системи контролю цілісності критичних інформаційних ресурсів із використанням рисунків.

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. What is Webmin? : веб-сайт. URL: https://webmin.com/ (дата звернення: 30.09.2024).
2. Управління системами за допомогою веб-консолі RHEL 9 : веб-сайт. URL: https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/managing_systems_using_the_rhel_9_web_console/index (дата звернення: 30.03.2024).
3. Introduction to the Advanced Intrusion Detection Environment (AIDE): веб-сайт. URL: https://www.opensourcerers.org/2024/04/15/introduction-to-the-advanced-intrusion-detection-environment-aide/ (дата звернення: 30.09.2024).
4. Samhain: веб-сайт. URL: https://la-samhna.de/samhain/ (дата звернення: 30.09.2024).
5. Network monitoring with intuition: веб-сайт. URL: https://www.observium.org/ (дата звернення: 30.09.2024).
6. Welcome to the Monitorix project Take control over your small server: веб-сайт. URL: https://www.monitorix.org/ (дата звернення: 30.09.2024).
7. Linux System Monitoring Fundamentals: веб-сайт. URL: https://www.linode.com/docs/guides/linux-system-monitoring-fundamentals/ (дата звернення: 30.09.2024).
8. The CentOS Project: веб-сайт. URL: https://www.centos.org/ (дата звернення: 10.10.2024).
9. inotify(7) — Linux manual page: веб-сайт. URL: https://man7.org/linux/man-pages/man7/inotify.7.html (дата звернення: 25.10.2024).
10. Definition: What Is auditd?: веб-сайт. URL: http://surl.li/xwverh (дата звернення: 25.10.2024).
11. Configure Linux system auditing with auditd: веб-сайт. URL: https://www.redhat.com/en/blog/configure-linux-auditing-auditd (дата

звернення: 25.10.2024).

12. Tripwire’s Integrity Management and Cybersecurity Solutions: веб-сайт. URL: https://www.tripwire.com/ (дата звернення: 28.10.2024).
13. Top 9 file integrity monitoring (FIM) best practices: веб-сайт. URL: https://sysdig.com/blog/file-integrity-monitoring/ (дата звернення: 10.10.2024).
14. Securing Your CentOS 7 Environment: A Step-by-Step Guide: веб-сайт. URL https://tuxcare.com/blog/securing-your-centos-7-environment-a-step-by-step-guide/ (дата звернення: 10.10.2024).
15. Package: inotify-tools (4.23.9.0-2 and others): веб-сайт. URL: https://packages.debian.org/eu/sid/inotify-tools (дата звернення: 5.11.2024).
16. /etc/passwd, що це за файл і для чого він?: веб-сайт. URL: https://uk.ubunlog.com/%d1%82%d0%be%d1%89%d0%be-passwd/ (дата звернення: 5.11.2024).
17. Створення користувачів та груп: веб-сайт. URL: http://surl.li/tgvpsv (дата звернення: 5.11.2024).
18. /etc/fstab: веб-сайт. URL: https://wiki.gentoo.org/wiki//etc/fstab/uk (дата звернення: 5.11.2024).
19. Файл hosts: веб-сайт. URL: https://tuthost.ua/uk/faq/fajl-hosts/ (дата звернення: 9.11.2024).
20. Налаштування DNS клієнта у Linux, файл “resolv.conf”: веб-сайт. URL: https://resk.ua/ua/dns-resolv-conf/ (дата звернення: 9.11.2024).
21. How to Set or Change System Hostname in Linux: веб-сайт. URL: https://runcloud.io/blog/change-hostname-in-linux (дата звернення: 9.11.2024).
22. Що таке sysctl: веб-сайт. URL: https://freehost.com.ua/ukr/faq/articles/chto-takoe-sysctl/ (дата звернення: 9.11.2024).
23. rc.local: веб-сайт. URL: https://www.ipfire.org/docs/pkgs/rc-local (дата звернення: 9.11.2024).
24. Crontab: веб-сайт. URL: https://www.sciencedirect.com/topics/computer-

science/crontab#:~:text=crontab%20files%20(cron%20table)%20tells,store%20their%20own%20schedule%20files. (дата звернення: 9.11.2024).

25. Explain Sudoers file Configuration in Linux: веб-сайт. URL: https://heshandharmasena.medium.com/explain-sudoers-file-configuration-in-linux-1fe00f4d6159#:~:text=The%20sudoers%20file%20is%20a,local%20users%2C%20network%20users). (дата звернення: 9.11.2024).
26. sshd_config - How to Configure the OpenSSH Server?: веб-сайт. URL: https://www.ssh.com/academy/ssh/sshd_config (дата звернення: 9.11.2024).
27. Classic SysAdmin: Viewing Linux Logs from the Command Line: веб-сайт. URL: http://surl.li/zxruvn (дата звернення: 9.11.2024).
28. Red Hat Enterprise Linux - What Are the Log Files That Are Located in /var/log and What Do They Do?: веб-сайт. URL: http://surl.li/zkddzu (дата звернення: 9.11.2024).
29. Configuration File Structure: веб-сайт. URL: https://documentation.suse.com/es-es/sles/12-SP5/html/SLES-all/cha-grub2.html#:~:text=%2Fgrub2%2Fgrub.-,cfg,directly%20from%20the%20file%20system. (дата звернення: 9.11.2024).

Збірники тез	«Litteris et Artibus: Нові горизонти».Кременець, листопад 12, 2024.
	«Розвиток наукової думки постіндустріального суспільства: сучасний дискурс». Хмельницький, листопад 1, 2024.
	«Здобутки та досягнення прикладних та фундаментальних наук XXI століття». Біла Церква, листопад 22, 2024.