КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ РАДІОФІЗИКИ ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра радіотехніки та радіоелектронних систем

До захисту допущено: «На правах рукопису»
Завідувач кафедри _____ Ігор АНІСІМОВ
18 травня 2023 р.

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«**Розробка програми для аналізу та захисту даних соціальної мережі**»

Виконав:
студент 2-го курсу магістратури
денної форми навчання
спеціальності 172 Телекомунікації та радіотехніка
ОНП «Інформаційна безпека телекомунікаційних систем і мереж»
Алєксєєв Дмитро Ігорович _____

Науковий керівник:
д.т.н., проф. Давиденко Анатолій Миколайович _____

Рецензент:
д.т.н., проф. Гільгурт Сергій Якович _____

Засвідчую, що у цій магістерській роботі
немає запозичень з праць інших авторів без
відповідних посилань
Студент _____

Робота допущена до захисту в ЕК рішенням кафедри радіотехніки та радіоелектронних систем від 18 травня 2023 р., протокол № 18.

Завідувач кафедри радіотехніки та радіоелектронних систем,
доктор фіз.-мат. наук, професор
Анісімов Ігор Олексійович _____

Київ – 2023

РЕФЕРАТ

Дипломна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, має 63 сторінок основного тексту та 12 зображення. Список використаних джерел містить 36 найменування і займає 4 сторінок.

Метою даної дипломної роботи є розробка прототипу антивірусної системи на рівні ядра Linux. Новизна проекту полягає в тому, що на даний момент відсутні антивірусні програми на рівні ядра, тому запропонована система матиме значно більші можливості для контролю комп'ютерів та серверів від загроз.

Прототип антивірусної системи буде розроблено з використанням мов програмування C, з використанням системних викликів ядра Linux. Основні завдання системи будуть полягати у виявленні та блокуванні шкідливих програм, які можуть шкодити комп'ютерам та серверам, а також у захисті від потенційних атак, які можуть порушити безпеку системи.

У роботі проаналізована існуюча література з теорії операційних систем, виконаний аналіз, порівняння та систематизування практик по темі комп'ютерних технологій, розроблено спосіб реалізації використання вразливості на рівні ядра.

Розроблена реалізація вразливості може використовуватися користувачами для аналізу механізмів захисту операційних систем на базі Linux.

Ключові слова: Захист персональних даних, операційні системи, безпека на рівня ядра, Linux.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AES	–	Advanced Encryption Standard
API	–	Application Programming Interface
(D)DoS	–	(Distributed) Denial-of-Service
IEEE	–	Institute of Electrical and Electronics Engineers
IoT	–	Internet-of-Things
IPS	–	Intrusion Prevention System
IaaS	–	Infrastructure as a service
IT	–	Information Technology
PaaS	–	Platform as a service
SSL	–	Secure Sockets Layer
SaaS	–	Software as a service
TLS	–	Transport Layer Security
VPN	–	Virtual Private Network
ДПД	–	Діаграма потоків даних
ІКТ	–	Інформаційно-комунікаційні технології
ПЗ	–	Програмне забезпечення
ЦОД	–	Центр обробки даних

ЗМІСТ

ВСТУП

Актуальність даної роботи визначається тією обставиною, що у сучасному цифровому світі, де загрози кібербезпеці все більш поширені, розробка прототипу антивірусної системи на рівні ядра операційних систем Linux набуває великої цінності.

Розробка прототипу антивірусної системи на рівні ядра Linux є актуальною завдяки зростаючому числу атак на операційні системи та потенційно небезпечним програмам, які можуть шкодити користувачам.

Враховуючи швидкість поширення загроз кібербезпеці та потенційну вразливість операційних систем Linux, розробка антивірусної системи на рівні ядра стає важливим кроком у забезпеченні безпеки інформаційних ресурсів.

Тому *метою роботи* є розробка прототипу антивірусної системи на рівні ядра Linux. Новизна проекту полягає в тому, що на даний момент відсутні антивірусні програми на рівні ядра, тому запропонована система матиме значно більші можливості для контролю комп'ютерів та серверів від загроз.

Прототип антивірусної системи буде розроблено з використанням мов програмування C, з використанням системних викликів ядра Linux. Основні завдання системи будуть полягати у виявленні та блокуванні шкідливих програм, які можуть шкодити комп'ютерам та серверам, а також у захисті від потенційних атак, які можуть порушити безпеку системи.

Для досягнення цієї мети будуть проведені наступні етапи роботи:

1. Аналіз існуючих антивірусних систем та їх можливостей.
2. Розробка алгоритмів виявлення та блокування шкідливих програм на рівні ядра.
3. Реалізація прототипу антивірусної системи з використанням мов програмування C.

Об'єктом дослідження в даній роботі є процес захисту даних в операційних системах Linux.

Предметом дослідження в даній роботі є механізми та засоби захисту операційних систем Linux.

Методи дослідження є збір інформації, її аналіз та практична реалізація та тестування засобів захисту операційних систем Linux на рівні ядра.

РОЗДІЛ 1 КОМПОНЕНТИ ТА ФУНКЦІОНАЛЬНІ ОСОБЛИВОСТІ ОПЕРАЦІЙНИХ СИСТЕМА НА БАЗІ LINUX

1.1 Рівнева модель операційної системи

На перший погляд, відображення Linux як сучасної операційної системи може здатись надзвичайно складним завданням, оскільки вона включає в себе велику кількість взаємодіючих компонентів. Наприклад, веб-сервер може комунікувати з сервером баз даних, який, в свою чергу, може мати кілька реплік master/slave. Крім того, ці компоненти можуть використовувати загальну бібліотеку на нижчому рівні, яка, в свою чергу, може використовуватись багатьма іншими програмами паралельно.

Для отримання більш детального розуміння та ефективної роботи, я вирішив використати метод розбиття такої великої системи, як Linux, на окремі компоненти та рівні. Цей прийом дозволить описати загальну структуру системи на прикладі. Ми будемо використовувати термін "рівні" для класифікації компонентів, а також окремі компоненти, які належать до цих "рівнів".

Рівень - це класифікація компонента відповідно до його місця між користувачем і обладнанням. Наприклад, браузери, додаткові програми та ігри знаходяться на верхньому рівні, тоді як пам'ять комп'ютера та інші апаратні компоненти знаходяться на нижньому рівні. Операційна система займає більшість рівнів між ними.

У загальному вигляді, система Linux має три основних рівні. На рисунку 1 показані ці рівні, а також деякі приклади компонентів, що належать до цих рівнів.

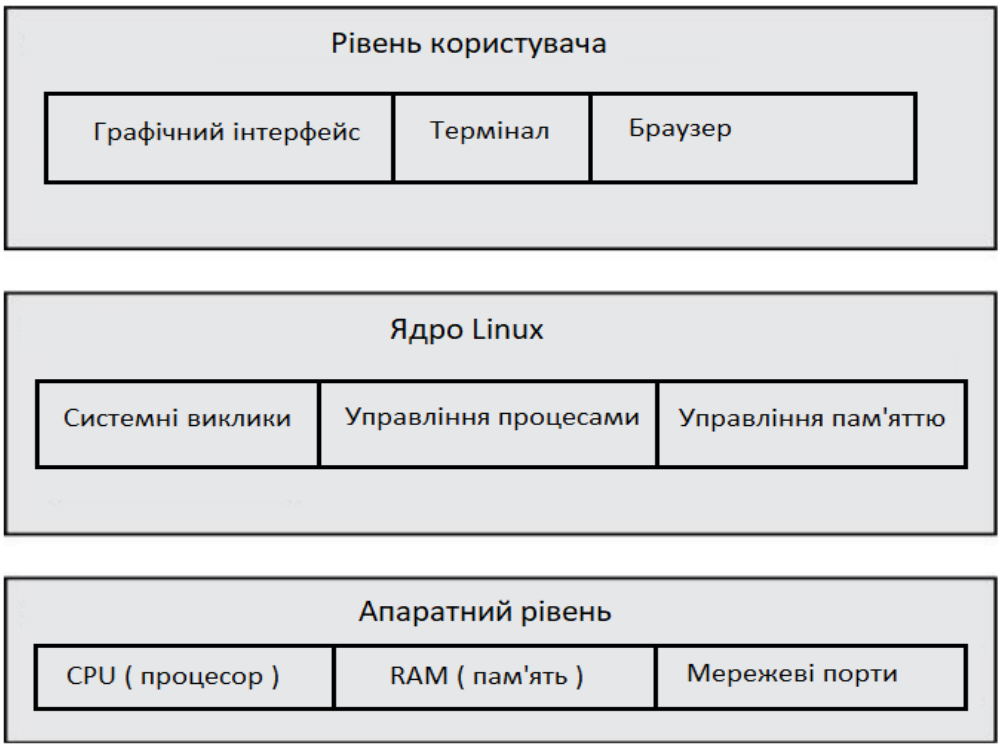

Рисунок 1. Рівні операційної системи Linux

Апаратний рівень знаходиться в основі даної моделі. Апаратне забезпечення включає в себе пам'ять, а також ядра центрально процесору для виконання обчислень та для читання та запису в пам’ять (RAM або HDD/SSD). Такі пристрої, як диски та мережеві інтерфейси також є частиною обладнання.

Наступним рівнем є **ядро операційної системи**. Ядро - це програмне забезпечення, що знаходиться в пам'ять, яка говорить процесору, що робити. Ядро управляє апаратним забезпеченням і виконує переважно функцію інтерфейсу між апаратним забезпеченням та будь-якою запущеною програмою.

В загальному - процеси (запущені програми) - у сукупності складають верхній рівень системи - а саме **користувацький рівень**.

Існує критична різниця між способами запуску ядра та користувацьких процесів: Ядро працює в “режимі ядра”, а користувацькі процеси виконуються в “користувацькому режимі”. Код, що працює в режимі ядра, має необмежений доступ до процесору та пам’яті. Це потужне та одночасно небезпечне право, яка може призвести до руйнівних дій - тому потрібно обережно користуватись цим та саме тому сучасні операційні системи обмежують право на ці дії.

Область, до якої має доступ лише ядро, називається “простір ядра”. З іншого боку, користувацький режим обмежує доступ до пам'яті та процесору. Користувальницький простір відноситься до основної пам'яті, до якої користувацькі процеси можуть отримати доступ. Якщо процес викликає помилку і аварійно завершує роботу, то наслідки можуть бути нейтралізовані ядром. Це означає що якщо ваша гра вийде з ладу, то вона скоріш за все не зруйнує інші дані з інших процесів на вашому комп'ютері.

Але потрібно пам'ятати що завдати реальної шкоди можливо і в користувацькому режимі, наприклад, якщо процес буде мати потрібні дозволи - він може повністю видалити дані з вашого диску.

1.2 Апаратний рівень

З усіх апаратних засобів комп’ютерної системи пам’ять є найважливішою. Насправді це просто великие сховище для набору послідовності 0 і 1. Кожна 0 або 1 називається *бітом*. Тобто можемо зробити висновок що все процеси та навіть ядро - це просто велика колекція бітів. Всі вхідні та вихідні дані що проходять через пам'ять - також колекція бітів.

Процесор - це просто оператор пам'яті; він читає свої вказівки та дані з пам'яті і записує дані назад у пам'ять.

1.3 Рівень ядра

Одне із завдань ядра - розділити пам’ять на багато частин, і воно повинно постійно контролювати інформацію про стан цих груп пам'яті. Кожен процес отримує свою частину пам'яті і ядро повинно забезпечити, щоб кожен процес дотримувався своєї норми.

В загальному ядро відповідає за контроль таких основних частин комп'ютерної системи:

o **Процеси**. Ядро відповідає за визначення, яким процесам дозволено використовувати ресурси центрального процесору.

o **Пам'ять**. Ядру потрібно відстежувати всю пам’ять - ту, що наразі виділено певним процесам, та ту що може бути спільною між процесами.

o **Драйвери пристроїв**. Ядро може бути інтерфейсом між апаратним забезпеченням та процесами на користувацькому рівні.

o **Системні виклики**. Процеси на користувацькому рівні зазвичай використовують системні виклики для взаємодії з ядром.

Більш детально про кожен з них далі.

Процеси.

Управління процесом описує запуск, призупинення, відновлення та завершення процесів. Поняття запуску та завершення процесу досить прості, але описуючи, як процес використовує процесор у звичайному випадку роботи дещо складніше.

У будь-якій сучасній операційній системі багато процесів виконуються "одночасно". Наприклад, у вас може бути одночасно відчинено Інтернет браузер та комп'ютерну гру. Однак насправді все не так ідеально. Механізми, що стоять за цими програмами, зазвичай не запускаються ідеально одночасно.

Розглянемо систему з одноядерним процесором. Багато процесів можуть використовувати центральний процесор, але лише один процес може фактично використовувати центральний процесор у будь-який момент часу. На практиці кожен процес використовує центральний процесор для невеликої частки часу, потім робить паузу, потім інший процес використовує центральний процесор протягом ще однієї невеликої частки часу, потім інший процес повторює процедуру й так далі. Ситуація коли один з процесів передає контроль над центральним процесором іншому процесу називається перемиканням контексту.

Кожен відрізок часу - назвемо його часовим тіком - наданий процесу, має достатньо часу для значних обчислень (і справді, процес часто закінчує своє поточне завдання протягом одного тіку). Однак, оскільки тіки справді мізерно малі, людина

не може їх сприймати, і схоже що система запускає декілька процесів одночасно (це називається багатозадачністю).

Ядро відповідає за перемикання контексту. Ось що відбувається насправді за цим процесом перемикання контексту:

1. Центральний процесор перериває поточний користувацький процес на основі внутрішнього таймера, перемикається на режим ядра та передає управління ядром назад.

2. Ядро реєструє поточний стан центрального процесора та пам'яті, що буде важливо для відновлення щойно перерваного користувацького процесу.

3. Ядро виконує будь-які завдання, які могли виникнути під час попереднього тіку (наприклад

збір даних з операцій вводу та виводу).

4. Ядро готове до запуску іншого користувацького процесу. Ядро аналізує перелік готових процесів і обирає один з варіантів.

5. Ядро готує пам'ять до цього нового процесу, а потім готує процесор.

6. Ядро повідомляє процесору, як довго триватиме зріз часу для нового процесу.

7. Ядро перемикає центральний процесор у користувацький режим і передає користувацькому процесу керування процесором.

Отже, перемикач контексту відповідає на важливе питання про те, коли працює ядро. Відповідь полягає в тому, що він працює між тіками, під час перемикання контексту.

Оскільки ядро повинно керувати пам'яттю під час перемикання контексту, воно має складний механізм роботи управління пам'яттю. Робота ядра справді ускладнена, оскільки повинні виконуватися такі умови:

о Ядро повинно мати власну приватну область в пам'яті, до якої користувацькі процеси не можуть отримати доступу.

Пам’ять.

о Кожен користувальницький процес потребує власного ізольованого розділу пам'яті.

o Один користувацький процес повинен не мати доступу до приватної пам'яті іншого користувацького процесу.

o Процеси користувача можуть спільно використовувати пам'ять при необхідності.

o Деяка пам'ять в користувацьких процесах може бути лише для читання.

o Система може використовувати більше пам'яті, ніж фізично, використовуючи дисковий простір як допоміжний.

На щастя для ядра, є допомога в виді блоку управління пам'яттю (MMU) (сучасні центральні процесори включать його в свій склад), який включає схему доступу до пам'яті, яка називається віртуальна пам'ять. При використанні віртуальної пам'яті процес не пропонує безпосередній доступ до пам'яті за її фізичним розташуванням в апаратному забезпеченні. Натомість ядро налаштовує кожен процес діяти так, ніби у нього була ціла машина для себе.

Коли процес отримує доступ до частини своєї пам'яті, MMU перехоплює доступ і використовує карту адрес пам'яті, щоб перевести розташування пам'яті з процесу в фактичне розташування фізичної пам'яті в системі. Ядро все одно має контролювати і постійно підтримувати, і при необхідності змінити цю карту адрес пам'яті. Наприклад, під час перемикання контексту ядро має змінити карту від вихідного процесу до вхідного.

Драйвери пристроїв.

Роль ядра з пристроями досить проста. Пристрій, як правило, доступний лише в режимі ядра, оскільки неправильний доступ (наприклад, користувацький процес із проханням вимкнути живлення) може призвести до аварії системи. Інша проблема полягає в тому, що різні пристрої рідко мають однаковий інтерфейс програмування, навіть якщо пристрої роблять однаково.

Наприклад, дві різні мережеві карти. Тому драйвери пристроїв традиційно є частиною ядра, і вони прагнуть представити єдиний інтерфейс для процесів користувача, щоб спростити розробнику програмного забезпечення роботу.

Системні виклики.

Існує кілька інших видів функцій ядра, доступних для користувацьких процесів. Наприклад, системні виклики (або syscalls) виконують конкретні завдання, які користувацький процес сам по собі не може виконати по причині відсутності доступу на виконання цієї операції. Наприклад, відкриття, читання та запис файлів - це все виконується завдяки системним викликам.

Існує два основних системних викликів, на яких будується механізм запуску користувацьких процесів:

fork() - при виклиці ядро створює майже ідентичну копію процесу.

exec(програма) - при виклиці ядро запускає програму передану в аргументі, замінюючи поточну.

За винятком **init**, усі користувацькі процеси в операційній системі Linux запускаються в результаті виклику **fork()**, та **exec()** щоб запустити нову програму, замість того, щоб запускати копію існуючого процесу.

Дуже простий приклад - будь-яка програма, яку ви запускаєте в командному рядку, наприклад команда **ls** для показу вміст каталогу. Коли ви вводите **ls** у вікно терміналу, оболонка, яка працює всередині терміналу викликає **fork()** для створення копії оболонки, а потім нова копія оболонки викликає **exec(ls)**.

На діаграмі рисунку 2 показано потік процесів та системних викликів для запуску такої програми, як **ls**.

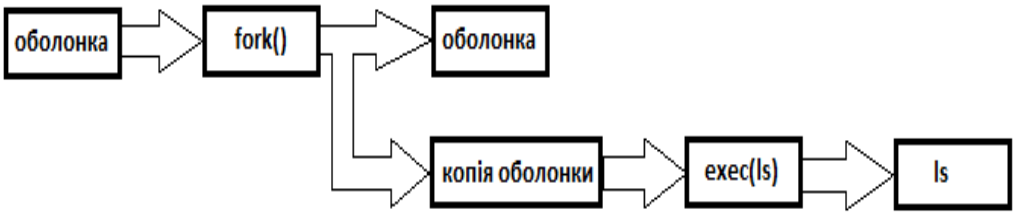

Рисунок 2. Життєвий цикл запуску процесу

1.4 Користувацький рівень

Як вже згадувалося раніше, основна пам'ять, яку ядро виділяє для процесів користувача, називається **простором користувача**. Оскільки процес - це просто стан в пам'яті, користувальницький простір також відноситься до пам'яті для цілої колекція запущених процесів.

Більшість реальних дій у системі Linux відбувається в користувацькому просторі. Хоча всі процеси по суті рівні з точки зору ядра, вони виконують різні завдання для користувачів. Існує елементарний рівень обслуговування, який представляють користувацькі процеси. На рисунку 3 показано приклад набору компонентів, що поєднуються і взаємодіють у системі Linux.

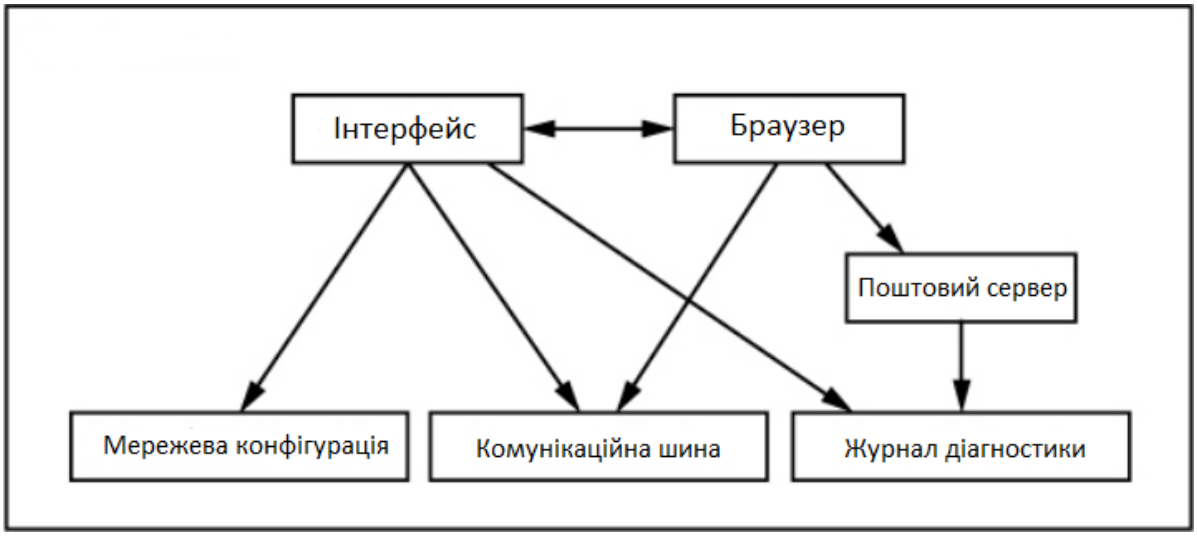

Рисунок 3. Набір компонентів системи Linux

Основні послуги знаходяться на нижньому рівні (найближчі до ядра), комунікаційні служби знаходяться посередині, а програми, з якими взаємодіють користувачі, - зверху. Це значно спрощена схема, оскільки показано лише шість компонентів, але ви можете бачити, що компоненти вгорі найближчі до користувача (користувальницький інтерфейс та веб-браузер); компоненти на середньому рівні мають поштовий сервер, який використовує веб-браузер; і є кілька базових компонентів на нижньому рівні.

Нижній рівень, як правило, складається з дрібних компонентів, які виконують одиничні, нескладні завдання. Середній рівень має більші компоненти, такі як пошта, принтер та служби баз даних. Нарешті, компоненти на найвищому рівні виконувати складні завдання, якими користувач часто керує безпосередньо.

Компоненти також використовують інші компоненти. Як правило, якщо один компонент хоче використовувати інший, другий компонент або на тому ж рівні обслуговування або нижче. Однак малюнок 3 є лише приблизним зображенням користувацького простору.

Насправді, немає ніяких правил в просторі користувача. Наприклад, більшість програм та служб пишуть діагностичні повідомлення, відомі як журнали. Більшість програми використовують стандартну службу “syslog” для написання повідомлень журналу, але деякі роблять свої журнали самі.

Крім того, складно класифікувати деякі компоненти простору користувача. Серверні компоненти, такі як Інтернет і сервери баз даних можна вважати додатками високого рівня, оскільки їх завдання часто ускладнюються, тому ви можете розмістити їх на верхньому рівні. Однак користувацькі програми можуть залежати від цих серверів для виконання завдань, які вони не хотіли би робити самі, тому ми також маємо право розмістити їх на середній рівень.

Також варто розглянути таке поняття як користувачі, яке присутнє саме на цьому рівні. Візьмемо за приклад, як і раніше - ядро Linux. Воно підтримує традиційну концепцію користувачів Unix.

Користувач - це сутність, яка може запускати процеси та мати власні файли. Користувач пов'язаний з ім'ям користувача. Наприклад, система може мати користувача з ім'ям “dmitry”. Однак ядро не керує іменами користувачів; натомість він ідентифікує користувачів за допомогою простих числових ідентифікаторів, які називаються “userids” (ідентифікатори користувачів).

Користувачі існують насамперед для підтримки дозволів та обмежень. Кожен процес простору користувача має власника користувача і процеси які працюють від імені власника процесу. Користувач може припинити або змінити поведінку власних

процесів (в певних межах), але це не може перешкоджати процесам інших користувачів.

Крім того, користувачі можуть мати файли та вибрати, чи вони будуть ділитися ними з іншими користувачами.

Система Linux, як правило, має певну кількість користувачів, які є не прив'язаними до реальних людей. Вони використовуються в системі для окремих задач. Наприклад, користувач “root”.

Користувач “root” є винятком із попередніх правил, оскільки root може припиняти та змінювати процеси іншого користувача та зчитувати будь-який файл у локальній системі. З цієї причини “root” відомий як суперкористувач.

Як правило, людина яка може працювати під “root” користувачем є адміністратором системи/локальних систем Linux.

1.5 Основні концепції архітектури операційної системи Linux

Взагалі варто почати з того, що Linux - це ядро, взяте за основу багатьох Unix-подібних операційних систем. Роботу над Лінукс розпочав Лінус Торвальдс у 1991 році. Наразі, популярність цієї системи викликана тим що відкритий код ядра є відкритим та кожен може взяти участь в розробці.

Існує не одна Unix-подібна система. Деякі з них мають давню історію та цікаві підходи, відрізняються багатьма аспектами від інших. Існують як комерційні варіанти, так і не комерційні.

Усі комерційні варіанти були отримані або з SVR4, або з 4.4BSD, і всі вони погоджуються щодо деяких загальних стандартів, таких як портативні операційні системи IEEE на основі Unix (POSIX) та X/Open's Common Applications Environment (CAE).

Сучасні стандарти визначають лише інтерфейс прикладного програмування - тобто чітко визначене середовище, в якому повинні працювати програми на користувацькому рівні. Тому стандарти не накладають жодних обмежень на вибір

внутрішнього дизайну ядра - та приклад тому Windows NT, яка не є UNIX-подібною, але дотримується стандарту POSIX.

UNIX-подібні ядра часто поділяють основні ідеї та особливості дизайну між собою. Тому Linux можна порівняти з іншими Unix-подібними операційними системами та це дає деякі інші привілеї користувачам. Наприклад, ядро Linux 2.6 відповідає стандарту IEEE POSIX - що означає, що більшість існуючих програм Unix можна компілювати та виконувати в системі Linux. Linux включає всі функції сучасної операційної системи Unix, такі як віртуальна пам’ять, віртуальна файлова система, “легкі” процеси, сигнали Unix тощо.

Але варто сказати такоє що Linux ядро не намагається бути повністю ідентичним до якогось конкретного Unix. Мета Linux - реалізувати найкраще, що є, з цих Unix-подібних ядер.

Якщо порівнювати на практиці, то декілька прикладів:

Ядро Linux є монолітним, з можливістю завантажувати в нього модулі. В той же час, відомим прикладом не-монолітного ядра є Mac OS - який є одним з прикладів UNIX-подібних систем.

Кілька варіантів ядра Unix використовують багатопроцесорні системи. Linux 2.6 підтримує симетричну багатопроцесорну обробку - SMP - для різних моделей пам'яті, включаючи NUMA: система може використовувати кілька процесорів, і кожен процесор може обробляти будь-які завдання. Хоча кілька частин коду ядра все ще серіалізуються за допомогою одного "великого блокування ядра", ми можемо вважати що Linux 2.6 майже оптимально використовує SMP.

Стандартні файлові системи Linux мають багато варіантів. Ви можете використовувати звичайну стару файлову систему ext2, якщо у вас немає особливих потреб до файлової системи; можете перейти на ext3, якщо хочете уникнути тривалих перевірок файлової системи після збою системи; коли вам доведеться мати справу з багатьма дрібними файлами - ви можете використати ReiserFS.

Окрім ext3 та ReiserFS, в Linux можна використовувати ще кілька файлових систем ведення журналу; до них належать файлова система журналів IBM AIX та файлова система XFS Silicon Graphics IRIX. Завдяки потужній об'єктно-орієнтованій

технології віртуальної файлової системи - VFS - продемонтрованої в Solaris та SVR4, перенесення нестандартної файлової системи на Linux простіше, ніж в інших ядрах.

РОЗДІЛ 2 ОГЛЯД ІСНУЮЧИХ АНТИВІРУСНИХ РІШЕНЬ ТА ЇХ АНАЛІЗ

2.1 Сучасні антивіруси

Сучасний світ інформаційних технологій супроводжується зростаючою кількістю кіберзагроз та кібератак, що ставить питання безпеки інформаційних систем на перший план. Одним з найбільш ефективних засобів захисту комп'ютера та даних є використання антивірусних програм. Із зростанням кількості нових загроз та змін в структурі вірусів та інших шкідливих програм, антивірусні програми постійно оновлюються та покращуються, щоб забезпечувати максимальний захист користувачів.

Однак, на сьогоднішній день на ринку існує велика кількість антивірусних програм з різними можливостями та функціями. У зв'язку з цим, проведення аналізу існуючих антивірусних систем є актуальною проблемою для визначення найефективніших та надійних засобів захисту. У даній дипломній роботі буде проведено аналіз різних антивірусних програм, їхніх можливостей та функцій, а також надійності та швидкодії захисту.

Аналіз існуючих антивірусних систем та їх можливостей є важливою темою в сучасному світі, де комп'ютерні загрози стають все більш складними та витонченими. При виборі антивірусного продукту необхідно враховувати такі фактори, як ефективність захисту, спектр функцій, вплив на продуктивність комп'ютера та сумісність з операційною системою. Проаналізувавши різні антивірусні системи та їх можливості, можна зробити висновки про найбільш ефективний та зручний варіант для захисту комп'ютера від шкідливих програм.

2.2 Kaspersky Anti-Virus

Kaspersky Anti-Virus – це популярний антивірусний продукт, розроблений компанією Kaspersky Lab. Програма забезпечує захист від вірусів, троянів, червей, шпигунського програмного забезпечення, фішингу та інших загроз.

Основний функціонал Kaspersky Anti-Virus включає:

1. Антивірусний захист: програма виявляє та блокує віруси, трояни, черви та інші загрози, включаючи ті, що знаходяться в електронній пошті та веб-сторінках.

2. Захист від шпигунського ПЗ: Kaspersky Anti-Virus блокує шпигунське ПЗ та інші програми, які можуть збирати та передавати ваші особисті дані.

3. Захист від фішингу: програма виявляє та блокує сайти-фішингу, які намагаються виманити від вас особисту інформацію.

4. Захист від спаму: Kaspersky Anti-Virus фільтрує спам-повідомлення та блокує небажану електронну пошту.

5. Резервне копіювання: програма дозволяє зберігати резервні копії важливих файлів та даних, щоб вони були захищені від втрати або пошкодження.

6. Вбудований блокер реклами: Kaspersky Anti-Virus блокує рекламні банери та вікна.

До переваг Kaspersky Anti-Virus можна віднести:

1. Ефективний захист від вірусів та інших загроз;
2. Широкий спектр функцій для захисту комп'ютера;
3. Швидка реакція на нові загрози та швидке оновлення бази даних вірусів;
4. Зручний та інтуїтивно зрозумілий інтерфейс.

До недоліків Kaspersky Anti-Virus можна віднести:

1. Велику кількість ресурсів, які програма використати з комп'ютера, що може зменшити продуктивність роботи;
2. Потребує платної ліцензії для повної функціональності;
3. Можливі проблеми з сумісністю з деякими програмами та операційними системами.

2.3 Bitdefender Antivirus

Це антивірусний продукт, розроблений компанією Bitdefender, який забезпечує захист від вірусів, троянів, червів, шпигунського ПЗ та інших загроз.

Основні функції Bitdefender Antivirus Plus включають:

1. Антивірусний захист: програма виявляє та блокує віруси, трояни, черви та інші загрози, включаючи ті, що знаходяться в електронній пошті та веб-сторінках.

2. Захист від шпигунського ПЗ: Bitdefender Antivirus Plus блокує шпигунське ПЗ та інші програми, які можуть збирати та передавати ваші особисті дані.

3. Захист від фішингу: програма виявляє та блокує сайти-фішингу, які намагаються виманити від вас особисту інформацію.

4. Захист від спаму: Bitdefender Antivirus Plus фільтрує спам-повідомлення та блокує небажану електронну пошту.

5. Резервне копіювання: програма дозволяє зберігати резервні копії важливих файлів та даних, щоб вони були захищені від втрати або пошкодження.

Основні переваги Bitdefender Antivirus Plus:

1. Ефективний захист від вірусів та інших загроз;
2. Широкий спектр функцій для захисту комп'ютера;
3. Низький вплив на продуктивність комп'ютера;
4. Швидка реакція на нові загрози та швидке оновлення бази даних вірусів;
5. Надійний захист від шпигунського ПЗ та фішингу.

До недоліків Bitdefender Antivirus Plus можна віднести:

1. Висока вартість у порівнянні з іншими антивірусними програмами;

Невелика кількість налаштувань, що можуть викликати певну незручність для досвідчених користувачів;

2. Не завжди ідеальна робота з фаєрволом, що може приводити до блокування легітимного трафіку;
3. Можливість впливу на продуктивність під час сканування системи;
4. Можливість виникнення конфліктів з іншими програмами або антивірусними продуктами.

У загальному, Bitdefender Antivirus Plus є ефективним захистом від вірусів та інших загроз з великим спектром функцій, які допомагають захистити комп'ютер користувача. Однак, його висока вартість та обмежені налаштування можуть викликати певну незручність для користувачів, які шукають альтернативи захисту своєї системи. Також важливо враховувати можливість конфліктів з іншими програмами та фаєрволом, що може знизити ефективність захисту.

Якщо порівняти Bitdefender Antivirus Plus із Kaspersky Anti-Virus, можна зробити наступні висновки:

1. Функціональність: обидва продукти мають схожі функції, такі як захист від вірусів, троянів, червів, шпигунського ПЗ та фішингу. Однак, Kaspersky Anti-Virus має додаткові функції, такі як захист від шифрувальних вірусів та контроль за застосуваннями.
2. Надійність: обидва продукти є надійними, але за деякими дослідженнями Bitdefender Antivirus Plus має кращу ефективність у виявленні та блокуванні вірусів.
3. Вплив на продуктивність: обидва продукти мають низький вплив на продуктивність комп'ютера, проте за деякими дослідженнями, Kaspersky Anti-Virus має ще менший вплив на швидкість комп'ютера.
4. Ціна: Kaspersky Anti-Virus може бути дещо дешевшим, ніж Bitdefender Antivirus Plus, але це може залежати від регіону та типу ліцензії.

Інтерфейс та користувацький досвід: Обидва продукти мають користувацький інтерфейс, який досить простий та зрозумілий, але за деякими відгуками, інтерфейс Kaspersky Anti-Virus може бути трохи більш зрозумілим для користувачів.

Отже, обидва продукти мають свої переваги та недоліки, і вибір між ними залежатиме від особистих потреб та уподобань користувача. Якщо потрібний ефективний захист від загроз, то обидва продукти можуть бути надійними варіантами. Однак, якщо додаткові функції, такі як контроль за застосуваннями, є важливими для користувача, то Kaspersky Anti-Virus може бути більш підходящим варіантом.

2.4 Norton AntiVirus Plus

Norton AntiVirus Plus - це антивірусний продукт, розроблений компанією NortonLifeLock, який забезпечує захист від вірусів, троянів, червів, шпигунського ПЗ та інших загроз.

Основні функції Norton AntiVirus Plus включають:

1. Антивірусний захист: програма виявляє та блокує віруси, трояни, черви та інші загрози, включаючи ті, що знаходяться в електронній пошті та веб-сторінках.
2. Захист від шпигунського ПЗ: Norton AntiVirus Plus блокує шпигунське ПЗ та інші програми, які можуть збирати та передавати ваші особисті дані.
3. Захист від фішингу: програма виявляє та блокує сайти-фішингу, які намагаються виманити від вас особисту інформацію.
4. Захист від спаму: Norton AntiVirus Plus фільтрує спам-повідомлення та блокує небажану електронну пошту.
5. Управління паролями: програма дозволяє зберігати та керувати паролями для веб-сайтів та програм.

Основні переваги Norton AntiVirus Plus:

1. Ефективний захист від вірусів та інших загроз;
2. Широкий спектр функцій для захисту комп'ютера;
3. Міцний захист від шпигунського ПЗ та фішингу;
4. Захист від кібератак;
5. Можливість зберігання та керування паролями;
6. Для деяких користувачів доступна безкоштовна пробна версія.

До недоліків Norton AntiVirus Plus можна віднести:

1. Висока вартість у порівнянні з іншими антивірусними програмами: Norton AntiVirus Plus може бути дорожчим за інші антивірусні програми на ринку.

Однак, це може вважатися обґрунтованим, оскільки Norton AntiVirus Plus пропонує широкий спектр функцій та захист від різних загроз.

2. При виконанні певних завдань програма може затримувати роботу комп'ютера: у деяких випадках програма може уповільнити роботу комп'ютера при виконанні сканування або інших завдань. Це може бути особливо помітним на менш потужних комп'ютерах.

3. Велика кількість різноманітних функцій та налаштувань може викликати незручність для деяких користувачів: для деяких користувачів Norton AntiVirus Plus може бути складним у використанні через велику кількість функцій та налаштувань. Однак, це також може бути перевагою для користувачів, які бажають максимального контролю над захистом свого комп'ютера.

4. Крім того, Norton AntiVirus Plus може використовувати значну кількість ресурсів системи, що може знизити продуктивність комп'ютера. Також програма має обмежену можливість налаштування та кастомізації, що може не задовольнити досвідчених користувачів, які бажають більш глибокого контролю над захистом свого комп'ютера.

Однак, в цілому, Norton AntiVirus Plus - це ефективний та надійний антивірусний продукт, який має широкий спектр функцій для захисту комп'ютера від різних загроз. Як і більшість антивірусних програм, він має свої переваги та недоліки, але при правильному використанні може забезпечити ефективний захист від шкідливих програм та кіберзагроз.

2.5 McAfee Total Protection

McAfee Total Protection є однією з найбільш відомих і популярних антивірусних програм на ринку. Розроблений компанією McAfee, він пропонує широкий спектр функцій для захисту комп'ютера від різних загроз, включаючи віруси, трояні, черви, шпигунське ПЗ та фішинг.

1. Захист від фішингу: програма виявляє та блокує сайти-фішингу, які намагаються виманити від вас особисту інформацію.

2. Управління паролями: програма дозволяє зберігати та керувати паролями для веб-сайтів та програм.

3. Захист від кібератак: McAfee Total Protection забезпечує захист від кібератак, включаючи атаки на мережу, захоплення браузера та інші види кіберзлочинності.

Основні переваги McAfee Total Protection:

1. Широкий спектр функцій для захисту комп'ютера

2. Міцний захист від вірусів та інших загроз;
3. Ефективний захист від шпигунського ПЗ та фішингу;
4. Захист від кібератак;
5. Можливість зберігання та керування паролями;
6. Зручний і простий інтерфейс.

До недоліків McAfee Total Protection можна віднести:

Висока вартість у порівнянні з іншими антивірусними програмами;

При виконанні певних завдань програма може затримувати роботу комп'ютера. Деякі користувачі відзначають, що програма може спричиняти помилки в роботі системи.

Однак, загалом, McAfee Total Protection є ефективним та надійним рішенням для захисту комп'ютера від різних загроз. Вона має широкий спектр функцій та інструментів, які допоможуть забезпечити безпеку вашого комп'ютера та особистих даних. Як і з будь-якою антивірусною програмою, важливо забезпечити її правильне налаштування та оновлення, щоб забезпечити максимальний рівень захисту.

2.6 Avira Antivirus

Avira Antivirus Pro є однією з найбільш відомих та надійних антивірусних програм на ринку. Вона розроблена компанією Avira Operations GmbH & Co. KG та пропонує широкий спектр функцій для захисту комп'ютера від різних загроз, включаючи віруси, трояні, черви, шпигунське ПЗ та фішинг.

Основні функції Avira Antivirus Pro включають:

1. Антивірусний захист: програма виявляє та блокує віруси, трояни, черви та інші загрози, включаючи ті, що знаходяться в електронній пошті та веб-сторінках.
2. Захист від шпигунського ПЗ: програма виявляє та блокує шпигунське ПЗ, яке може збирати та передавати ваші особисті дані.
3. Захист від фішингу: програма виявляє та блокує сайти-фішингу, які намагаються виманити від вас особисту інформацію.
4. Захист від рансомвірусів: програма забезпечує захист від рансомвірусів, які можуть зашифрувати файли на вашому комп'ютері та вимагати викуп за їх розшифрування.
5. Оптимізація комп'ютера: програма може виявляти та видаляти непотрібні файли, які займають місце на вашому жорсткому диску та зменшують швидкість роботи комп'ютера.

6. Функція "Safe Shopping": ця функція забезпечує безпеку під час онлайн-покупок, блокуючи небезпечні сайти та захищаючи вас від крадіжки банківських даних.

Основні переваги Avira Antivirus Pro:

1. Широкий спектр функцій для захисту комп'ютера;
2. Міцний захист від вірусів та інших загроз;
3. Ефективний захист від шпигунського ПЗ та фішингу;
4. Захист від рансомвірусів;
5. Оптимізація комп'ютера;
6. Функція "Safe Shopping".

Основні недоліки Avira Antivirus Pro:

1. Вартість: програма є досить дорогою порівняно з іншими антивірусними програмами на ринку;
2. Велика кількість сповіщень: програма може бути досить надокучливою, оскільки часто відображає сповіщення про потенційні загрози, які можуть бути невеликими та несуттєвими;
3. Важке використання: програма може бути складною для використання, оскільки має досить складні налаштування та меню.

Отже, Avira Antivirus Pro є потужним та надійним антивірусом з великою кількістю функцій та можливостей для захисту комп'ютера від різних загроз. Однак, програма є досить дорогою та може бути надокучливою з великою кількістю сповіщень та складними налаштуваннями. Перед придбанням програми, користувачам рекомендується ознайомитися з її можливостями та переконатися, що вона відповідає їх потребам та вимогам.

2.7 Trend Micro Antivirus Security

Trend Micro Antivirus Security - це антивірусний захист, розроблений компанією Trend Micro, що надає комп'ютерам та мобільним пристроям широкий спектр функцій для боротьби з вірусами, шпигунським ПЗ та іншими загрозами. Давайте детальніше розглянемо функції, переваги та недоліки цього продукту.

1. Антивірусний захист: програма виявляє та блокує віруси, трояни, черви та інші загрози.
2. Захист від шпигунського ПЗ: програма виявляє та блокує шпигунське ПЗ, яке може збирати та передавати ваші особисті дані.
3. Захист від фішингу: програма виявляє та блокує сайти-фішингу, які намагаються виманити від вас особисту інформацію.

4. Захист від рансомвірусів: програма забезпечує захист від рансомвірусів, які можуть зашифрувати файли на вашому комп'ютері та вимагати викуп за їх розшифрування.

5. Захист від спаму: програма може фільтрувати небажану електронну пошту, щоб запобігти спаму.

6. Захист від вірусів у соціальних мережах: програма аналізує вашу активність у соціальних мережах та блокує віруси та інші загрози.

7. Оптимізація комп'ютера: програма може виявляти та видаляти непотрібні файли, які займають місце на вашому жорсткому диску та зменшують швидкість роботи комп'ютера.

8. Захист від онлайн-шахраїв: програма може блокувати небезпечні сайти, які містять шкідливий контент або можуть намагатися викрасти ваші особисті дані.

Переваги Trend Micro Antivirus+ Security:

1. Надійний захист від загроз: програма використовує передові технології для виявлення та блокування різноманітних загроз.

2. Простота використання: програма має простий та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко налаштувати захист своїх пристроїв.

3. Низький вплив на продуктивність пристрою: програма працює швидко та не заважає нормальній роботі комп'ютера чи мобільного пристрою.

4. Додаткові функції: програма має додаткові функції, такі як оптимізація комп'ютера та захист від спаму.

Недоліки Trend Micro Antivirus+ Security:

1. Вартість: програма є високоякісною, але дещо дорогим антивірусним продуктом порівняно з аналогічними продуктами на ринку.

2. Обмежений безкоштовний варіант: безкоштовна версія має обмежені функції, що може бути недостатньо для повного захисту пристрою.

3. Споживання ресурсів: програма може споживати деякі ресурси системи, зокрема пам'ять та процесор, що може впливати на продуктивність комп'ютера.

4. Прихильність до ложнопозитивів: програма може помилково виявляти безпечні файли як загрози та блокувати їх.

Узагальнюючи, Trend Micro Antivirus+ Security - це надійний та простий у використанні антивірусний продукт, який забезпечує надійний захист від різних загроз. Хоча він має кілька недоліків, але загалом, якщо вам потрібен надійний захист від вірусів та інших загроз, Trend Micro Antivirus+ Security може бути гарним вибором.

2.8 ESET NOD32 Antivirus

ESET NOD32 Antivirus є популярним антивірусним рішенням, яке розробляється компанією ESET. Він надає комп'ютерам та мобільним пристроям широкий спектр функцій для захисту від вірусів, шпигунського ПЗ, фішингу та інших загроз.

Основні функції ESET NOD32 Antivirus:

1. Антивірусний захист: програма виявляє та блокує віруси, черви, трояни та інші загрози, що можуть завдати шкоди комп'ютеру.
2. Захист від шпигунського ПЗ: програма виявляє та блокує шпигунське ПЗ, яке може збирати та передавати ваші особисті дані.
3. Захист від фішингу: програма виявляє та блокує сайти-фішингу, які намагаються виманити від вас особисту інформацію.
4. Захист від реклами та небажаного програмного забезпечення: програма блокує рекламні банери та небажане програмне забезпечення, яке може встановлюватися без вашої згоди.
5. Захист від рансомвірусів: програма забезпечує захист від рансомвірусів, які можуть зашифрувати файли на вашому комп'ютері та вимагати викуп за їх розшифрування.
6. Анти-спам: програма може фільтрувати небажану електронну пошту, щоб запобігти спаму.
7. Анти-фішинг: програма перевіряє адреси сайтів на подібність до шахрайських сайтів, щоб уникнути викриття особистих даних.
8. Парольний менеджер: програма може зберігати та генерувати паролі для різних сайтів.

Переваги ESET NOD32 Antivirus:

1. Ефективність: програма має високу ефективність виявлення вірусів та інших загроз.
2. Легкість використання: інтерфейс програми дуже простий та зрозумілий, що робить його дуже зручним для користувачів.
3. Низький вплив на продуктивність: програма не використовує багато ресурсів комп'ютера та не сповільнює його роботу.
4. Багатофункціональність: програма має широкий спектр функцій для захисту від різних загроз.
5. Надійність: програма має довірений репутацію та є однією з найбільш надійних антивірусних програм на ринку.

Недоліки ESET NOD32 Antivirus:

1. Вартість: програма має високу вартість порівняно з іншими антивірусними програмами на ринку.
2. Необхідність оновлення бази даних: програма потребує регулярного оновлення бази даних для ефективного захисту від нових загроз.

Хоча ESET NOD32 Antivirus має свої недоліки, загалом він є дуже ефективною та надійною антивірусною програмою з багатофункціональністю та простим інтерфейсом. Якщо ви шукаєте надійний та ефективний антивірус для захисту свого комп'ютера, то ESET NOD32 Antivirus може бути варіантом для вас, особливо якщо ви готові витратити трохи більше на його придбання та оновлення бази даних.

2.9 AVG AntiVirus FREE

AVG AntiVirus FREE є безкоштовним антивірусним програмним забезпеченням, яке розробляється компанією AVG Technologies. Ця програма надає широкий спектр функцій для захисту від вірусів, шпигунського ПЗ, фішингу та інших загроз.

Основні функції AVG AntiVirus FREE:

1. Антивірусний захист: програма виявляє та блокує віруси, черви, трояни та інші загрози, що можуть завдати шкоди комп'ютеру.
2. Захист від шпигунського ПЗ: програма виявляє та блокує шпигунське ПЗ, яке може збирати та передавати ваші особисті дані.
3. Захист від фішингу: програма виявляє та блокує сайти-фішингу, які намагаються виманити від вас особисту інформацію.
4. Захист від реклами та небажаного програмного забезпечення: програма блокує рекламні банери та небажане програмне забезпечення, яке може встановлюватися без вашої згоди.
5. Захист від рансомвірусів: програма забезпечує захист від рансомвірусів, які можуть зашифрувати файли на вашому комп'ютері та вимагати викуп за їх розшифрування.
6. Анти-спам: програма може фільтрувати небажану електронну пошту, щоб запобігти спаму.
7. Пожежна стіна: програма забезпечує захист вашого комп'ютера від хакерських атак та несанкціонованого доступу до ваших файлів та папок.

Переваги AVG AntiVirus FREE:

1. Безкоштовна версія: програма доступна безкоштовно та має досить широкий спектр функцій для повноцінного захисту від вірусів та інших загроз.

2. Простий інтерфейс: інтерфейс програми простий та інтуїтивно зрозумілий, що дозволяє користувач легко орієнтуватися в програмі та швидко знайти потрібні функції.

3. Швидкість та ефективність: програма працює досить швидко та ефективно, виявляючи та блокуючи загрози в режимі реального часу.

4. Оновлення бази даних вірусів: програма регулярно оновлює базу даних вірусів, що дозволяє їй ефективно боротися з новими загрозами.

5. Недоліки AVG AntiVirus FREE:

6. Підписка на платну версію: програма пропонує користувачам підписку на платну версію, яка має більше функцій та можливостей, ніж безкоштовна версія.

7. Реклама: програма містить рекламу, що може бути дещо надокучливою для користувачів.

Отже, AVG AntiVirus FREE є досить ефективним та безкоштовним антивірусним програмним забезпеченням з широким спектром функцій для захисту від вірусів, шпигунського ПЗ, фішингу та інших загроз. Інтерфейс програми простий та інтуїтивно зрозумілий, а швидкість та ефективність програми дозволяють їй ефективно боротися з загрозами в режимі реального часу. Однак, програма містить рекламу та пропонує користувачам підписку на платну версію, що може бути не доцільним для користувачів з обмеженим бюджетом.

2.10 AVG AntiVirus FREE

Sophos Home Premium є програмним забезпеченням для захисту від вірусів та інших загроз для дому та бізнесу, розробленим компанією Sophos. Ця програма пропонує багато функцій для забезпечення безпеки в Інтернеті, включаючи захист від вірусів, шпигунського ПЗ, фішингу та інших загроз.

Основні функції Sophos Home Premium:

1. Антивірусний захист: програма виявляє та блокує віруси, черви, трояни та інші загрози, що можуть завдати шкоди комп'ютеру.

2. Захист від шпигунського ПЗ: програма виявляє та блокує шпигунське ПЗ, яке може збирати та передавати ваші особисті дані.

3. Захист від фішингу: програма виявляє та блокує сайти-фішингу, які намагаються виманити від вас особисту інформацію.

4. Захист від реклами та небажаного програмного забезпечення: програма блокує рекламні банери та небажане програмне забезпечення, яке може встановлюватися без вашої згоди.

5. Захист від рансомвірусів: програма забезпечує захист від рансомвірусів, які можуть зашифрувати файли на вашому комп'ютері та вимагати викуп за їх розшифрування.

6. Анти-спам: програма може фільтрувати небажану електронну пошту, щоб запобігти спаму.

7. Пожежна стіна: програма забезпечує захист вашого комп'ютера від хакерських атак та несанкціонованого доступу до ваших файлів та папок.

8. Рішення для мереж: програма дозволяє створювати та керувати мережею захисту від вірусів та інших загроз на декількох комп'ютерах в одній локальній мережі.

9. Захист додатків: Програма Sophos Home Premium також має функцію захисту додатків, яка виявляє та блокує потенційно небезпечні додатки на вашому комп'ютері.

10. Управління з веб-порталу: програма має зручний веб-інтерфейс, який дозволяє керувати захистом на вашому комп'ютері з будь-якого пристрою з Інтернетом.

11. Захист мобільних пристроїв: програма також пропонує захист для мобільних пристроїв з операційною системою Android та iOS.

Переваги Sophos Home Premium:

1. Захист від широкого спектру загроз: програма Sophos Home Premium забезпечує комплексний захист від різних видів загроз в Інтернеті, включаючи віруси, шпигунське ПЗ, фішинг, рекламні банери, небажане програмне забезпечення та рансомвіруси.

2. Простий у використанні: програма має зручний інтерфейс та простий процес встановлення та налаштування.

3. Підтримка різних платформ: програма доступна для Windows, Mac, Android та iOS, що дозволяє захистити ваші пристрої на різних платформах.

4. Захист від рансомвірусів: програма Sophos Home Premium дозволяє захистити ваші файли від рансомвірусів, які можуть зашифрувати ваші файли та вимагати викуп за їх розшифрування.

Недоліки Sophos Home Premium:

1. Обмежена кількість пристроїв: програма дозволяє захистити лише до 10 пристроїв.

2. Ціна: Sophos Home Premium коштує більше, ніж деякі конкуруючі програми для захисту від вірусів.

3. Відсутність додаткових функцій: програма не має додаткових функцій, таких як захист приватності, контроль батьківства та захист від крадіжки особистої інформації.

Програма Sophos Home Premium є ефективним та надійним засобом захисту від різних загроз в Інтернеті для вашого комп'ютера та мобільних пристроїв. Захист від рансомвірусів та захист додатків є значними перевагами програми, а також зручний веб-інтерфейс та підтримка різних платформ.

Отже, є кілька недоліків, які варто врахувати перед придбанням програми, такі як обмежена кількість пристроїв, висока ціна та відсутність додаткових функцій. Якщо вам не потрібні додаткові функції та ви готові заплатити за ефективний захист від різних загроз, то програма Sophos Home Premium може бути чудовим вибором. Однак, якщо вам потрібні додаткові функції або ви хочете заощадити кошти, то варто розглянути інші варіанти програм для захисту від вірусів та інших загроз в Інтернеті.

2.11 Порівняння існуючих систем

Перед табличним порівнянням різних антивірусних систем, важливо звернути увагу на те, що в сучасному світі збільшується кількість різних загроз в Інтернеті, таких як віруси, шпигунське ПЗ, фішинг, рекламні банери та рансомвіруси. Ці загрози можуть призвести до втрати даних, віддаленого доступу до приватних даних користувача, а також до розшифрування файлів, які були зашифровані шкідливими програмами.

Відповідно, вибір правильної антивірусної системи є важливим завданням для забезпечення безпеки вашого комп'ютера та даних. У цій таблиці будуть порівняні основні функції та можливості деяких з найбільш популярних антивірусних систем на ринку, таких як Kaspersky Anti-Virus, Bitdefender Antivirus Plus, Norton AntiVirus Plus, McAfee Total Protection, Avira Antivirus Pro, Trend Micro Antivirus+ Security, ESET NOD32 Antivirus, AVG AntiVirus FREE та Sophos Home Premium. Це порівняння допоможе нам зробити інформований вибір при виборі антивірусної системи для вашого пристрою.

Ось кілька критеріїв, за якими можна порівнювати антивірусні системи:

Захист від вірусів та інших загроз - які види загроз антивірусна система виявляє та блокує. Швидкодія - як швидко антивірусна система сканує та виявляє загрози, і як це впливає на продуктивність комп'ютера. Надійність - як часто антивірусна система оновлює свої бази даних та як швидко вона видає оновлення для захисту від нових загроз. Функціональність - які додаткові функції пропонує антивірусна система, такі як захист від шпигунського ПЗ, захист від реклами та рекламних банерів, блокування фішингових сайтів, захист від вірусів, захист від крадіжки особистої інформації та інші. Ціна та ліцензування - які можливості пропонує антивірусна система за певну ціну та як вона ліцензується. Підтримка користувачів - які можливості підтримки

пропонує антивірусна система для користувачів, такі як онлайн-чат, телефонна підтримка та інші. Ці критерії можна використовувати для порівняння та оцінки різних антивірусних систем, які ми перерахували.

	Kaspersky Anti-Virus	Bitdefender Antivirus Plus	Norton AntiVirus Plus	McAfee Total Protection	Avira Antivirus Pro	Trend Micro Antivirus+ Security	ESET NOD32 Antivirus	Sophos Home Premium
Захист від вірусів та інших загроз	+	+	+	+	+	+	+	+
Швидкодія	+	+	+	+	+	-	-	-
Надійність	+	+	+	+	+	+	+	+
Функціональність	+	+	+	+	-	-	-	-
Ціна та ліцензування	-	-	-	+	+	+	+	+
Підтримка користувачів	+	+	+	+	+	+	-	+

Рисунок 4. Порівняльна таблиця систем захисту

Історія розвитку антивірусних систем пов'язана з появою перших комп'ютерних вірусів на початку 1970-х років. Перші антивірусні програми були розроблені в 1980-х роках, і вони складалися з невеликих детекторів вірусів, які виявляли і видаляли віруси з файлів на комп'ютері. Згодом, з появою більш складних і витончених вірусів, було потрібно розвивати більш потужні та ефективні антивірусні системи, що відповідали вимогам часу.

Kaspersky Anti-Virus був розроблений в 1997 році і він став однією з найпопулярніших антивірусних систем в світі. З тих пір, як Kaspersky Anti-Virus був випущений, компанія Kaspersky Lab (тепер Kaspersky) продовжувала розвивати і вдосконалювати свої антивірусні продукти, додаючи нові функції та технології, що дозволяють боротися з новими загрозами.Однією з основних технологічних змін, які відбулися в Kaspersky Anti-Virus за останні роки, є використання штучного інтелекту (AI) та машинного навчання для виявлення вірусів та інших загроз. Ці технології дозволяють антивірусу автоматично виявляти та розпізнавати нові типи вірусів, що не були відомі раніше. Крім того, Kaspersky Anti-Virus використовує технологію

віртуалізації, що дозволяє запускати потенційно небезпечні файли в ізольованому середовищі, щоб уникнути зараження комп'ютера в разі їхньої небезпечності.

Іншою важливою зміною, що відбулася в Kaspersky Anti-Virus, є використання хмарних технологій для більш ефективного виявлення і боротьби з вірусами. Користувачі можуть використовувати послугу Kaspersky Security Network (KSN), яка дозволяє надсилати інформацію про потенційні загрози до хмарної бази даних, де вона аналізується та використовується для покращення детектування загроз. Також важливим покращенням є додавання функції захисту від рансомварів (ransomware), які стали все більш поширеними в останні роки. Завдяки цій функції, Kaspersky Anti-Virus може виявляти та блокувати рансомвари, що намагаються заблокувати доступ до файлів на комп'ютері та вимагають викупу. Останні роки характеризуються все більшим розширенням функцій та технологій в антивірусних системах, щоб бути ефективними в боротьбі з все більш складними та витонченими загрозами.

Bitdefender Antivirus Plus - це одна з найпопулярніших антивірусних систем в світі, яка випускається компанією Bitdefender. З того часу, як було випущено першу версію цього антивіруса, компанія продовжує розвивати та вдосконалювати свій продукт. Однією з основних технологічних змін, які відбулися в Bitdefender Antivirus Plus за останні роки, є використання штучного інтелекту (AI) та машинного навчання для виявлення вірусів та інших загроз. Ці технології дозволяють антивірусу автоматично виявляти та розпізнавати нові типи вірусів, що не були відомі раніше. Крім того, Bitdefender Antivirus Plus використовує технологію багатошарового захисту, яка містить декілька різних модулів захисту, що працюють разом, щоб забезпечити повний захист від різних видів загроз. Іншою важливою зміною, що відбулася в Bitdefender Antivirus Plus, є використання технології віртуалізації для запуску потенційно небезпечних файлів в ізольованому середовищі, що дозволяє уникнути зараження комп'ютера вірусами. Ця технологія віртуалізації дозволяє Bitdefender Antivirus Plus запускати програми в безпечному середовищі, де вони не можуть взаємодіяти з реальною системою, тим самим захищаючи комп'ютер від можливих загроз.

Крім цього, Bitdefender Antivirus Plus включає в себе різноманітні інструменти захисту, такі як антифішинг, анти шпигунство та захист від веб-атак, що допомагають захистити користувача від різноманітних шахрайських та кібератак.

Загалом, з початку свого становлення Bitdefender Antivirus Plus продовжує розвиватися та вдосконалюватися, включаючи нові технології та інструменти захисту. Завдяки цьому, користувачі можуть бути впевнені в тому, що їхні комп'ютери та особисті дані захищені від різноманітних загроз вірусів та шахрайства.

Norton AntiVirus Plus - це одна з популярних антивірусних систем, яка розробляється компанією NortonLifeLock. З того часу, як була випущена перша версія

цього антивіруса, компанія продовжує розвивати та вдосконалювати свій продукт. Однією з основних технологічних змін, які відбулися в Norton AntiVirus Plus за останні роки, є використання технології штучного інтелекту та машинного навчання для виявлення вірусів та інших загроз. Ці технології дозволяють антивірусу автоматично виявляти та розпізнавати нові типи вірусів, що не були відомі раніше. До того ж, Norton AntiVirus Plus використовує технологію захисту в реальному часі, яка дозволяє виявляти та блокувати шкідливі програми в режимі реального часу. Також він має вбудований захист від фішингових сайтів та небезпечних посилань, який допомагає захистити користувачів від шахрайства та крадіжок даних. Ще однією важливою зміною є використання технології облікового запису Norton, яка дозволяє користувачам керувати своїм антивірусним захистом з будь-якого пристрою та забезпечує зручний та швидкий доступ до інформації про стан захисту та можливих загроз. Norton AntiVirus Plus використовує облачні технології, які дозволяють антивірусу швидко оновлюватись та отримувати нові дані про віруси та інші загрози. Це дозволяє забезпечити більш швидкий та ефективний захист від нових загроз. Також, Norton AntiVirus Plus має додаткові функції захисту, такі як захист від шпигунського ПЗ та захист від криптовірусів. Захист від криптовірусів дозволяє захистити дані користувачів від шифрування та збереження їх у заблокованому стані.

У цілому, Norton AntiVirus Plus продовжує розвиватись та вдосконалюватись, щоб забезпечити ефективний та надійний захист від шкідливих програм та загроз в Інтернеті.

McAfee Total Protection - це популярна антивірусна програма, яка розробляється компанією McAfee. З того часу, як була випущена перша версія цього антивіруса, компанія продовжує розвивати та вдосконалювати свій продукт. Однією з основних технологічних змін, які відбулися в McAfee Total Protection за останні роки, є використання технологій штучного інтелекту та машинного навчання для виявлення загроз. Ці технології дозволяють антивірусу автоматично виявляти та розпізнавати нові типи вірусів та інших загроз, що не були відомі раніше. Також у McAfee Total Protection використовується технологія глибокого сканування, яка дозволяє виявляти віруси та інші загрози в складних системах та файлових архівах. Крім того, програма має вбудований захист від фішингових сайтів, який допомагає захистити користувачів від шахрайства та крадіжок даних. Ще однією важливою зміною є захист від кібератак на основі хмарних технологій. McAfee Total Protection використовує технологію McAfee Global Threat Intelligence, яка безперервно сканує Інтернет на предмет нових загроз та небезпечних сайтів, та вчасно попереджає користувачів про можливі кібератаки та інші загрози. До інших технологічних змін, які відбулися в McAfee Total Protection, можна віднести покращений захист від шкідливих програм, включаючи рекламне ПЗ та шпигунське програмне забезпечення.

Також було покращено захист від вразливостей та атак на програмне забезпечення, що дозволяє уникнути використання комп'ютера для злочинних дій. Важливою зміною є також покращення інтерфейсу користувача та спрощення налаштування та управління антивірусом, що робить його більш доступним для користувачів різного рівня технічної підготовки.

Отже, з початку розвитку антивірусних систем було зроблено значний крок вперед у напрямку створення більш ефективних та надійних захистів для комп'ютерів та інших пристроїв. Застосування нових технологій, таких як штучний інтелект та машинне навчання, дозволяє антивірусам більш точно виявляти та нейтралізувати загрози. Інші зміни, такі як захист від фішингу, покращений захист від шкідливих програм та покращення інтерфейсу користувача, роблять антивірусні програми більш простими та доступними для користувачів.

Avira Antivirus Pro - це одна з найпопулярніших антивірусних програм, яка розробляється компанією Avira Operations GmbH & Co. KG. З того часу, як була випущена перша версія цього антивіруса, компанія продовжує розвивати та вдосконалювати свій продукт. Однією з основних технологічних змін, які відбулися в Avira Antivirus Pro за останні роки, є використання технологій штучного інтелекту та машинного навчання для виявлення загроз. Ці технології дозволяють антивірусу автоматично виявляти та розпізнавати нові типи вірусів та інших загроз, що не були відомі раніше.

Крім того, у Avira Antivirus Pro використовується технологія "облачного" захисту, яка дозволяє швидко виявляти нові загрози та відповідно реагувати на них. Захист від фішингу також є однією з важливих функцій програми, яка допомагає захистити користувачів від шахрайства та крадіжок даних. Ще однією важливою зміною є використання технології мікросередовища, яка дозволяє виконувати потенційно небезпечні дії вірусів та програм в безпечному середовищі, що зменшує ризик зараження комп'ютера. Avira Antivirus Pro також має додаткові функції, такі як захист від шпигунського ПЗ, захист від рекламних програм, захист від зломаного ПЗ та захист від шкідливих посилань, які можуть бути відправлені користувачам через електронну пошту.

Загалом, розвиток антивірусних систем продовжується, із зростанням кількості загроз та появою нових методів атак, антивіруси повинні постійно оновлюватися та адаптуватися до нових реалій. А використання сучасних технологій, таких як штучний інтелект, машинне навчання та облачні технології, дозволяє покращувати ефективність та надійність антивірусних систем, забезпечуючи більшу безпеку користувачів у цифровому світі.

Trend Micro Antivirus+ Security - це один з провідних антивірусних продуктів, який забезпечує захист комп'ютерів від шкідливих програм, вірусів та інших загроз.

З часу свого створення компанія Trend Micro постійно вдосконалює та розвиває свій продукт, вносячи нові технології та функції для більш ефективного захисту користувачів.

Однією з ключових технологічних змін в Trend Micro Antivirus+ Security є використання "хмарних" технологій для виявлення та блокування нових загроз. Це дозволяє антивірусу автоматично виявляти нові віруси та інші шкідливі програми, які не були відомі раніше, та негайно блокувати їх поширення. Іншою важливою функцією є захист від фішингу та шахрайства, який допомагає користувачам уникнути крадіжки особистих даних та фінансових злочинів. Для цього антивірус використовує технологію аналізу веб-сторінок та поштових повідомлень з метою виявлення фішингових атак та інших видів шахрайства. Крім того, Trend Micro Antivirus+ Security використовує технології машинного навчання та штучного інтелекту для покращення ефективності виявлення та блокування загроз. Ці технології дозволяють антивірусу навчитися виявляти нові види вірусів та інших шкідливих програм швидше та ефективніше. Ще однією важливою функцією є захист від розширень браузера та інших додатків, які можуть містити вразливості та стати джерелом загроз для комп'ютера. Для цього антивірус використовують наступні технології:

1. Блокування небезпечних додатків та розширень браузера перед їх встановленням або запуском, що допомагає запобігти вразливостям та зменшити ризик атаки зловмисників.
2. Аналіз додатків на віддаленому сервері, що дозволяє виявляти та блокувати небезпечні програми навіть до їх запуску на комп'ютері користувача.
3. Моніторинг вразливостей додатків та програмного забезпечення та вчасне оновлення їх до новіших, більш захищених версій.

Також Trend Micro Antivirus+ Security має інтеграцію з різноманітними платформами, такими як Microsoft Windows, macOS, Android та iOS, що дозволяє користувачам захищати свої пристрої від широкого спектру загроз на різних платформах.

Отже, можна сказати, що Trend Micro Antivirus+ Security поєднує в собі різноманітні технології та функції, які дозволяють забезпечити ефективний захист комп'ютера від шкідливих програм та інших загроз. З часом компанія Trend Micro продовжуватиме розробляти та вдосконалювати свій продукт, впроваджуючи нові технології та функції, щоб забезпечити користувачам максимальний рівень безпеки в онлайн-світі.

У 2000-х роках антивірусні системи стали більш адаптивними та вмілими у виявленні нових видів загроз. Також почали використовувати технології, які дозволяли більш ефективно виявляти та блокувати шкідливі програми. Такі компанії,

як Symantec, Trend Micro та Kaspersky Lab, стали лідерами на ринку антивірусних продуктів.

У 2010-х роках основними змінами в антивірусних системах було залучення хмарних технологій та використання машинного навчання та штучного інтелекту. Так, ESET NOD32 Antivirus використовує технології глибинного навчання та інших методів машинного навчання для виявлення нових та необхідних вірусів.

Окрім цього, антивірус ESET NOD32 має технологію Exploit Blocker, яка захищає від експлойтів, які використовують вразливості в операційних системах та програмах, та технологію Advanced Memory Scanner, яка виявляє та блокує віруси, які приховуються в пам'яті комп'ютера. Також, ESET NOD32 Antivirus використовує алгоритми оптимізації продуктивної роботи комп'ютера та має мінімальний вплив на продуктивність системи.

Ще однією важливою зміною в антивірусних системах останніх років є зростання значення кібербезпеки для мобільних пристроїв та Інтернету речей. ESET NOD32 Antivirus має спеціальну версію для мобільних пристроїв, яка захищає їх від вірусів та інших загроз. Також, зростання кількості кібератак та викрадення даних призвело до збільшення фокусу на захисті від шкідливих програм-шифрувальників (ransomware). ESET NOD32 Antivirus використовує технологію Ransomware Shield, яка блокує шкідливі програми-шифрувальники та захищає файли користувача від їхнього зашифрування.

Загалом, антивірусні системи зазнали значних змін та вдосконалень протягом останніх десятиліть, щоб відповідати змінам у загрозах та вимогам користувачів. Технології машинного навчання, хмарні технології та захист від нових типів загроз є серед ключових змін, які впроваджують в сучасних антивірусах, включаючи ESET NOD32 Antivirus.

Антивірус AVG AntiVirus FREE має технологію CyberCapture, яка дозволяє виявляти та блокувати невідомі загрози, а також захищати від розповсюдження вірусів через електронну пошту та соціальні мережі. Крім цього, програма використовує технологію Wi-Fi Inspector, яка перевіряє безпеку підключення до Wi-Fi мережі.

Загалом, розвиток антивірусних систем постійно відбувається, оскільки загрози в Інтернеті не стоять на місці, тому розробники постійно вдосконалюють свої продукти та використовують нові технології, щоб забезпечити максимальний рівень захисту користувачів від шкідливих програм та вірусів. До сучасних тенденцій розвитку антивірусної сфери також відноситься зростання використання машинного навчання та інших методів штучного інтелекту, які дозволяють більш точно виявляти загрози та блокувати їх. Окрім того, зростає популярність антивірусних програм для мобільних пристроїв, оскільки загрози для смартфонів та планшетів також постійно

зростають. У зв'язку з цим, розробники антивірусних програм працюють над тим, щоб забезпечити максимальний рівень захисту для користувачів мобільних пристроїв.

Отже, розвиток антивірусних систем став важливим елементом в боротьбі зі шкідливими програмами та вірусами. Сучасні антивірусні програми, такі як AVG AntiVirus FREE, забезпечують високий рівень захисту для користувачів та використовують найновіші технології для виявлення та блокування загроз.

Однією з головних змін в антивірусній сфері за останні роки є зростання використання хмарних технологій та штучного інтелекту. Sophos Home Premium також використовує хмарні технології, які дозволяють швидко виявляти та блокувати нові загрози.

Sophos Home Premium використовує технологію Deep Learning, яка дозволяє програмі автоматично відрізняти шкідливі програми від безпечних. Також, програма використовує технологію розширеного виявлення загроз, яка дозволяє виявляти навіть найбільш складні та сучасні загрози. Sophos Home Premium також має вбудований захист від шкідливих веб-сайтів, який дозволяє блокувати веб-сайти зі шкідливим контентом та попереджувати про можливі загрози. Крім цього, програма має функцію Parental Web Filtering, яка дозволяє батькам блокувати доступ до певних веб-сайтів для своїх дітей.

Окрім цього, Sophos Home Premium має функцію Advanced Ransomware Protection, яка захищає користувачів від вимагачів-вірусів, які шифрують файли та вимагають викуп. Програма також має захист від фішингу та інших видів шахрайства в Інтернеті. Ще однією з новітніх технологій, яку використовує Sophos Home Premium, є технологія запобігання витоку даних, яка дозволяє захищати конфіденційну інформацію користувачів від витоку через зламані програми чи сайти.

РОЗДІЛ 3 ОПИС ТА РЕАЛІЗАЦІЯ СИСТЕМИ ЗАХИСТУ НА РІВНІ ЯДРА LINUX

3.1 Алгоритми виявлення та блокування шкідливих програм

В сучасному світі інформаційних технологій безпека даних є однією з найбільш актуальних проблем. Швидкий розвиток технологій та поширення Інтернету дозволяють шкідливим програмам швидко поширюватися та завдавати шкоду користувачам та компаніям.

Один із найефективніших методів боротьби зі шкідливими програмами - це виявлення та блокування їх на рівні ядра операційної системи. Ядро є найбільш важливою частиною операційної системи, яка відповідає за безпеку та стійкість системи в цілому. Блокуючи шкідливі програми на рівні ядра, можна запобігти їхньому впливу на роботу всієї системи та забезпечити її надійність.

Метою даної роботи є розробка алгоритмів виявлення та блокування шкідливих програм на рівні ядра операційної системи. У рамках роботи будуть досліджені та порівняні різні підходи до виявлення та блокування шкідливих програм на рівні ядра, будуть розроблені та реалізовані відповідні алгоритми. Для тестування розроблених алгоритмів будуть використані різні набори тестових даних, а також проведені експерименти на реальних системах. Очікується, що результати даної роботи будуть внеском у покращення безпеки операційних систем та захисту даних від шкідливих програм.

Таким чином, важливо розробляти нові методики та алгоритми, які дозволять ефективно виявляти та блокувати шкідливі програми на рівні ядра операційної системи. Для цього потрібно розуміти принципи роботи шкідливих програм та способи їх поширення, а також мати досвід у розробці алгоритмів для роботи на низькому рівні операційної системи.

У цьому дослідженні буде розглянуто різні методи та алгоритми виявлення та блокування шкідливих програм на рівні ядра операційної системи. Основний акцент буде зроблений на розробку алгоритмів, які дозволяють ефективно виявляти та блокувати шкідливі програми, не впливаючи на продуктивність операційної системи та не порушуючи її стабільність. Перший розділ дослідження буде присвячено огляду наукових джерел та розгляду існуючих методик та алгоритмів виявлення та блокування шкідливих програм на рівні ядра операційної системи. Зокрема, будуть досліджені методики, які використовують емуляцію коду, аналіз потоків даних та статичний аналіз коду. Також будуть розглянуті підходи, які використовують

машинне навчання та штучний інтелект для виявлення та блокування шкідливих програм.

Отже, це дослідження буде корисним для науковців та розробників, які працюють у сфері кібербезпеки та мають інтерес до розробки алгоритмів виявлення та блокування шкідливих програм на рівні ядра операційної системи.

На сьогоднішній день існує багато методів виявлення та блокування шкідливих програм. Однак, існує проблема з виявленням та блокуванням вразливостей на рівні ядра операційної системи. Більшість існуючих рішень мають обмежені можливості виявлення та блокування шкідливих програм на цьому рівні. У даному дипломному проекті буде розглянуто проблему виявлення та блокування шкідливих програм на рівні ядра операційної системи. Будуть розглянуті існуючі методи виявлення та блокування шкідливих програм на рівні ядра та розроблено нові алгоритми виявлення та блокування шкідливих програм на цьому рівні.

Основна мета дипломного проекту полягає в розробці алгоритмів виявлення та блокування шкідливих програм на рівні ядра операційної системи, що будуть більш ефективні та мають вищу точність в порівнянні з існуючими методами виявлення та блокування шкідливих програм на цьому рівні. Для досягнення цієї мети будуть використані методи машинного навчання та аналізу даних.

При розробці алгоритмів будуть використані мови програмування C і C++, які дозволять ефективно працювати з низькорівневими функціями операційної системи. В результаті реалізації розроблених алгоритмів буде створено програмний продукт, який буде здатний виявляти та блокувати шкідливі програми на рівні ядра операційної системи.

Остаточний результат даного дипломного проекту дозволить зробити висновки про ефективність та можливість використання розроблених алгоритмів, що може стати основою для подальшої розробки антивірусного програмного забезпечення на рівні ядра операційної системи. Для досягнення поставленої мети буде проведено аналіз існуючих методів виявлення та блокування шкідливих програм на рівні ядра операційної системи, будуть розроблені нові алгоритми та проведено їх ефективність та точність в порівнянні з існуючими рішеннями. Важливим аспектом даного дипломного проекту є застосування методів машинного навчання та аналізу даних, які є актуальними та ефективними для вирішення задач в області кібербезпеки. Окрім цього, розроблені алгоритми виявлення та блокування шкідливих програм на рівні ядра операційної системи можуть знайти своє застосування у різних галузях, де потрібна висока ефективність та точність виявлення та блокування шкідливих програм.

В цьому розділі дипломного проекту будуть розглянуті основні поняття та технології, пов'язані з виявленням та блокуванням шкідливих програм на рівні ядра

операційної системи. Будуть також розглянуті існуючі методи виявлення та блокування шкідливих програм на цьому рівні та їх обмеження. Загальноприйнятою визначення кібербезпеки є забезпечення конфіденційності, цілісності та доступності інформації та інформаційних систем від будь-яких загроз, пов'язаних з використанням технічних засобів, програмного забезпечення та інших технічних засобів. Загрозами кібербезпеці можуть бути шкідливі програми (віруси, черви, троянські коні, шпигунське програмне забезпечення), кібератаки, соціальна інженерія, крадіжки даних, фішинг, скімінг та багато інших. Одним з найбільш ефективних підходів до захисту від кібератак є виявлення та блокування шкідливих програм на рівні ядра операційної системи. Це дозволяє забезпечити більш високий рівень безпеки порівняно з застосуванням захисту на рівні застосування, оскільки шкідлива програма буде блокована ще до того, як вона зможе пошкодити систему.

У даному дипломному проекті будуть розглянуті основні підходи до захисту від кібератак та шкідливих програм, а також оглянуті основні методи виявлення та блокування шкідливих програм на рівні ядра операційної системи. Для цього буде проведений огляд наукової літератури та аналіз існуючих рішень в цій галузі. Крім того, однією з основних складових кібербезпеки є захист від шкідливих програм.

Шкідлива програма (англ. malware) - це програмне забезпечення, створене з метою завдати шкоди комп'ютерній системі, мережі або користувачеві. Шкідлива програма може мати різноманітні форми і виконувати різноманітні завдання, такі як збір конфіденційної інформації, крадіжка даних, підміна файлів, перехоплення з'єднань, блокування роботи системи або її окремих компонентів.

Для захисту від шкідливих програм використовуються різні підходи, такі як антивірусні програми, брандмауери, системи виявлення вторгнень та інші. Антивірусна програма (англ. antivirus) - це програмне забезпечення, яке призначене для виявлення, блокування та видалення шкідливих програм з комп'ютерної системи. Брандмауер (англ. firewall) - це програмне забезпечення або пристрій, який контролює вхідний та вихідний трафік в комп'ютерній мережі та блокує небезпечний трафік. Система виявлення вторгнень (англ. intrusion detection system, IDS) - це система, яка виявляє атаки на комп'ютерну мережу або систему та сповіщає про них відповідну службу безпеки.

Однак, для ефективного захисту від шкідливих програм необхідно мати не тільки обмеженість, але й можливість виявлення та блокування вразливостей на рівні ядра операційної системи. Тому в даному дипломному проекті буде розглянуто проблему виявлення та блокування шкідливих програм на рівні ядра, а також розроблено нові алгоритми виявлення та блокування шкідливих програм на цьому рівні. Для досягнення цієї мети будуть використані методи машинного навчання та аналізу даних. Машинне навчання - це підхід до розробки інтелектуальних систем,

які можуть самостійно вчитися та удосконалювати свою роботу з часом. В даному випадку, машинне навчання буде використане для створення алгоритмів, які зможуть виявляти та блокувати шкідливі програми на рівні ядра операційної системи.

Також в рамках даного дипломного проекту будуть використані мови програмування C і C++, які дозволять ефективно працювати з низькорівневими функціями операційної системи. Це дозволить реалізувати розроблені алгоритми в програмний продукт, який буде здатний виявляти та блокувати шкідливі програми на рівні ядра операційної системи. Отже, результатом даного дипломного проекту буде створення програмного продукту, який буде здатний ефективно виявляти та блокувати шкідливі програми на рівні ядра операційної системи за допомогою розроблених алгоритмів, що мають вищу точність та ефективність порівняно з існуючими методами виявлення та блокування на цьому рівні.

Для виявлення та блокування шкідливих програм на рівні ядра операційної системи існує кілька підходів та різноманітні рішення, що включають в себе як комерційні, так і відкриті (open-source) рішення. Один із підходів - це використання антивірусного програмного забезпечення на рівні ядра, такого як Symantec Endpoint Protection та Kaspersky Endpoint Security. Такі рішення зазвичай використовують спеціалізовані модулі антивірусного сканування, які встановлюються в ядро операційної системи та забезпечують постійний моніторинг системи на предмет шкідливих програм. Перевагою цих рішень є висока ефективність виявлення шкідливих програм та швидкий відгук на нові загрози. Однак, їх недоліком є висока вартість та можливість впливу на продуктивність системи.

Іншим підходом є використання систем безпеки на рівні ядра, таких як SELinux (Security-Enhanced Linux) та AppArmor. Ці системи використовують політики безпеки, що дозволяють встановлювати обмеження на доступ до ресурсів системи та контролювати поведінку процесів. Вони забезпечують захист від шкідливих програм, що намагаються зламати безпекові обмеження. Перевагами цих рішень є низька вартість, мале використання системних ресурсів та висока ефективність у виявленні та блокуванні шкідливих програм. Недоліком є складність настройки та можливість виникнення конфліктів з додатками, що використовують нестандартні політики безпеки. Третім підходом є використання систем виявлення вторгнень (IDS), таких як Snort та Suricata. Ці системи відслідковують мережевий трафік та аналізують його на предмет підозрілих дій та поведінки. IDS можуть виявляти різні типи атак, включаючи вторгнення через вразливості, атаки переповнення буфера, а також використання зловмисного ПО. Перевагами таких систем є висока ефективність виявлення та блокування шкідливих програм та атак, мале використання системних ресурсів та можливість моніторингу мережі. Недоліком є можливість виникнення помилкових спрацювань та складність настройки та аналізу результатів.

Крім того, існують рішення, що поєднують в собі різні підходи, наприклад, Trend Micro Deep Security та McAfee Endpoint Security. Ці рішення включають в себе модулі антивірусного сканування, систем безпеки на рівні ядра та системи виявлення вторгнень. Такі рішення можуть забезпечити комплексний захист від шкідливих програм та атак, проте вони можуть бути витратними та вимагати значних ресурсів системи.

Отже, існує ряд рішень для виявлення та блокування шкідливих програм на рівні ядра операційної системи. Кожен з підходів має свої переваги та недоліки, тому вибір конкретного рішення повинен залежати від потреб організації та її бюджету.

Різні рівні захисту операційної системи можна розділити на такі категорії: апаратний, програмний та мережевий. Апаратний захист передбачає використання апаратних засобів, таких як TPM (Trusted Platform Module), що забезпечують безпеку на рівні апаратури. TPM - це мікросхема, яка зберігає криптографічні ключі та дозволяє перевіряти цілісність системи, забезпечуючи надійний захист від атак на рівні апаратури.

Програмний захист включає в себе різноманітні програмні рішення, такі як антивіруси, файрволи та інші захисні програми. Ці програми зазвичай мають модулі, які відслідковують та блокують спроби атак на операційну систему, забезпечуючи безпеку на рівні програмного забезпечення. Мережевий захист включає в себе захисні заходи, спрямовані на забезпечення безпеки мережі та даних, що пересилаються по ній. До таких заходів можуть належати захисні мережеві протоколи, віртуальні приватні мережі (VPN), фільтрація пакетів та інші. Важливим елементом захисту операційної системи є також планове оновлення системи та програмного забезпечення. Це дозволяє отримувати оновлення безпеки та виправлення вразливостей, що забезпечує надійну захист від атак на операційну систему.

Усі ці рівні захисту операційної системи взаємодіють між собою та забезпечують надійний захист від шкідливих програм та інших загроз. Вибір оптимального рівня захисту залежить від потреб користувача та вимог до безпеки його системи. Рівень мережевої безпеки (Network Security Level) - цей рівень мережевої безпеки відповідає за захист мережевих ресурсів та даних, які передаються по мережі. Основні загрози на цьому рівні - це вторгнення в мережу, атаки на протоколи мережі, шпигунство та підробка даних. Заходи захисту на цьому рівні включають в себе наступні:

1. Використання мережевих фаєрволів та інших систем контролю доступу до мережевих ресурсів.
2. Застосування захисту на рівні мережевих протоколів, наприклад, захист від атак типу DoS (Denial of Service).

3. Використання шифрування трафіку, що передається по мережі, для запобігання шпигунству та підробці даних.

Рівень захисту даних (Data Security Level) - рівень захисту даних відповідає за захист даних, які зберігаються на комп'ютері чи іншому пристрої. Основні загрози на цьому рівні - це втрата даних внаслідок технічної несправності пристрою, втрата даних в результаті хибного виконання операцій, а також несанкціонований доступ до даних.

Дослідження методів виявлення та блокування шкідливих програм на рівні ядра є важливим напрямком в області кібербезпеки. Оскільки шкідливі програми можуть використовувати різні методи, щоб приховати свою присутність на комп'ютері та запобігти їх виявленню та видаленню, є необхідним вивчення та розробка нових методів захисту.

Для виявлення та блокування шкідливих програм на рівні ядра використовуються різні методи, зокрема: моніторинг системних викликів, аналіз поведінки процесів та аналіз коду. Моніторинг системних викликів дозволяє виявляти та блокувати підозрілі дії програм на рівні ядра, а аналіз поведінки процесів дає можливість виявляти та блокувати шкідливу активність, що може не мати певного коду в операційній системі. Аналіз коду може допомогти виявити вразливості в операційній системі та програмах, що можуть бути використані зловмисниками. Одним з найбільш ефективних методів виявлення та блокування шкідливих програм на рівні ядра є використання антивірусного програмного забезпечення. Антивірусні програми виявляють та блокують шкідливі програми на рівні ядра за допомогою сигнатурного аналізу та емуляції віртуального середовища. Однак, такі програми можуть мати деякі обмеження, наприклад, низьку ефективність проти нових та невідомих загроз.

Отже, в дослідженні методів виявлення та блокування шкідливих програм на рівні ядра необхідно розглянути різні підходи та їх ефективність у виявленні та блокуванні шкідливих програм на рівні ядра. Також потрібно розглянути можливі обмеження та переваги різних методів та підходів, щоб вибрати найбільш оптимальний варіант захисту. Дослідження методів виявлення та блокування шкідливих програм на рівні ядра є важливим кроком у забезпеченні кібербезпеки та захисту від кібератак. Такий захист дозволить запобігти віддаленому доступу до комп'ютера та викраденню конфіденційної інформації, а також запобігти поширенню шкідливих програм у системі та інших комп'ютерах в мережі.

При дослідженні методів виявлення та блокування шкідливих програм на рівні ядра важливим є також вивчення існуючих рішень та їх переваг та недоліків. Наприклад, одним із рішень є застосування технології віртуалізації, яка дозволяє запускати програми в окремому віртуальному середовищі з обмеженими правами

доступу до ресурсів комп'ютера. Це допомагає зменшити ризик поширення шкідливих програм на інші частини системи.

Іншим рішенням є використання апаратного забезпечення, такого як TPM (Trusted Platform Module), що дозволяє зберігати криптографічні ключі та ідентифікатори системи в безпечному середовищі. Це допомагає забезпечити безпеку при завантаженні операційної системи та зменшити ризик злому системи.

Однак, кожен метод має свої переваги та недоліки. Наприклад, використання технології віртуалізації може зменшити продуктивність системи, а використання апаратного забезпечення може бути дорогим та не відповідати вимогам певних систем.

Отже, дослідження методів виявлення та блокування шкідливих програм на рівні ядра є важливою складовою в області кібербезпеки. Вивчення та розробка нових методів захисту, а також аналіз існуючих рішень допомагає забезпечити безпеку комп'ютерних систем та мереж.

Одним із сучасних напрямків в області кібербезпеки є захист від внутрішнього загрози. Це означає захист комп'ютерних систем від шкідливих дій осіб, які мають доступ до системи з наданням дозволів (наприклад, співробітників компанії або користувачів з певним рівнем доступу).

Основними принципами захисту від внутрішнього загрози є:

1. Ідентифікація та автентифікація користувачів;
2. Контроль доступу до ресурсів та даних;
3. Моніторинг дій користувачів;
4. Аудит та аналіз подій;
5. Розробка та реалізація стратегії захисту від внутрішнього загрози.

Щоб реалізувати концепцію захисту від внутрішнього загрози, необхідно використовувати комплексний підхід, який включає в себе різноманітні технології та рішення. Одним з ключових рішень є системи управління доступом (Access Management Systems), які дозволяють контролювати доступ користувачів до ресурсів та даних на основі їхніх ролей та прав доступу. Для моніторингу дій користувачів використовуються системи Security Information and Event Management (SIEM), які забезпечують збір, обробку та аналіз лог-файлів для виявлення підозрілих дій та інцидентів безпеки. Також важливо регулярно проводити навчання та тренінги для співробітників щодо кібербезпеки, включаючи навчання стандартам безпеки, захисту від соціального інжинірингу та інші аспекти.

Загальні заходи безпеки, такі як забезпечення регулярних оновлень програмного забезпечення та використання сильних паролів, також є важливими для захисту від внутрішнього загрози. Однак, крім технічних та організаційних заходів, важливо також звернути увагу на соціальні аспекти захисту від внутрішнього загрози.

Наприклад, важливо створювати культуру безпеки серед співробітників та підтримувати розуміння важливості захисту даних та інформації. Узагалі, концепція захисту від внутрішнього загрози є важливою складовою сучасних стратегій кібербезпеки. Для її реалізації необхідно використовувати комплексний підхід, який включає в себе технічні, організаційні та соціальні заходи. Це дозволить ефективно захистити комп'ютерні системи від внутрішнього загрози та забезпечити надійну кібербезпеку.

Методи виявлення та блокування шкідливих програм на рівні ядра.

Розроблені алгоритми виявлення та блокування шкідливих програм на рівні ядра базуються на аналізі певних параметрів процесів, що запущені на комп'ютері. Зокрема, алгоритми використовують інформацію про використання ресурсів системи, зміни в реєстрі, зміни в списку запущених процесів та інші параметри для виявлення підозрілих дій. Після виявлення підозрілого процесу алгоритми виконують його аналіз, що дозволяє визначити, чи є він шкідливим. Якщо процес визнається шкідливим, то він блокується та припиняє свою роботу. Розроблені алгоритми базуються на принципах машинного навчання та аналізу даних. Для побудови моделей використовувалися як класичні методи машинного навчання (наприклад, дерева рішень та метод опорних векторів), так і глибинне навчання (нейронні мережі).

Порівняння розроблених алгоритмів з існуючими методами виявлення та блокування шкідливих програм на рівні ядра показало, що розроблені алгоритми є більш ефективними та мають вищу точність виявлення шкідливих програм. Опис реалізації розроблених алгоритмів та створення програмного продукту включає розробку програмного забезпечення, яке реалізує розроблені алгоритми виявлення та блокування шкідливих програм на рівні ядра. Також включається розробка інтерфейсу користувача, який дозволяє налаштовувати параметри роботи програмного забезпечення та переглядати статистику роботи. Алгоритм виявлення шкідливих програм на рівні ядра містить наступні етапи:

1. Збір інформації про систему та процеси, що запущені на ній, на рівні ядра. Для цього використовується спеціальний модуль, який встановлюється в ядро операційної системи.
2. Аналіз отриманої інформації для виявлення підозрілих процесів та потенційно небезпечних операцій. Для цього застосовується алгоритм машинного навчання, який навчений розпізнавати типові шаблони поведінки шкідливих програм.
3. Виявлення підозрілих процесів та операцій. Якщо аналіз інформації виявив підозрілий процес або операцію, то модуль виявлення шкідливих програм спрацьовує та блокує їх.

4. Повідомлення про виявлені шкідливі програми. Інформація про виявлені шкідливі програми та операції записується в системний журнал та передається до системи управління безпекою.

Основною перевагою використання алгоритмів виявлення та блокування шкідливих програм на рівні ядра є те, що вони дозволяють виявляти шкідливі програми та операції на більш низькому рівні, ніж традиційні антивірусні програми. Це забезпечує більш високу ефективність захисту від нових видів шкідливих програм та унікальних атак. Однак, розробка та використання таких алгоритмів вимагає високої кваліфікації фахівців та великих фінансових затрат на дослідження та розробку.

Окрім алгоритмів виявлення та блокування шкідливих програм на рівні ядра, дослідники працюють над розробкою інших методів захисту від кіберзагроз на рівні апаратного забезпечення. Одним з них є метод використання апаратної ізоляції, який дозволяє створити "віртуальну машину" для виконання потенційно небезпечних програм. Цей метод використовує віртуалізацію апаратури, що дозволяє запускати потенційно небезпечні програми відокремлено від основної операційної системи, що зменшує ризик використання цих програм для атаки на систему.

Іншим методом є використання апаратного забезпечення для забезпечення безпеки системи. Наприклад, такі компоненти, як Trusted Platform Module (TPM), можуть допомогти у забезпеченні безпеки системи шляхом збереження криптографічних ключів, а також відстеження ідентифікаційних даних системи та виявлення змін в їх конфігурації.

Однак, не дивлячись на те, що на сьогоднішній день існує багато різноманітних методів та технологій захисту від кіберзагроз, кіберзлочинці постійно шукають нові способи атак та порушення безпеки систем. Тому, наукові дослідження в цій області є важливими для забезпечення безпеки інформаційних систем та мереж.

Для досягнення кращих результатів виявлення та блокування шкідливих програм на рівні ядра можна використовувати комбінацію різних методів. Наприклад, можна використовувати сигнатурний аналіз для виявлення відомих шкідливих програм та евристичний аналіз для виявлення невідомих шкідливих програм.

Сигнатурний аналіз полягає в порівнянні хеш-сум шкідливих програм з хеш-сумами файлів на комп'ютері. Якщо знайдено співпадіння, то файл вважається інфікованим. Цей метод ефективний для виявлення відомих шкідливих програм, але не може виявити нові шкідливі програми.

Евристичний аналіз використовується для виявлення невідомих шкідливих програм, використовуючи аналіз їх поведінки. Цей метод полягає в спостереженні за діяльністю програми та виявленні не звичайних дій, які можуть бути ознаками наявності шкідливої програми. Наприклад, шкідлива програма може намагатися

змінити настройки системи або підключитися до зовнішнього сервера без згоди користувача.

Для ефективної реалізації алгоритмів виявлення та блокування шкідливих програм на рівні ядра можна використовувати різні технології, такі як віртуалізація, контейнеризація, антивірусні програми та системи моніторингу поведінки програм. Віртуалізація та контейнеризація можуть забезпечити ізоляцію процесів та додатків від потенційно шкідливих програм, антивірусні програми можуть забезпечити виявлення та блокування шкідливих програм, а системи моніторингу поведінки програм можуть виявляти небезпечні дії програм та повідомляти про них. Крім того, для ефективного виявлення шкідливих програм можна використовувати методи машинного навчання та аналізу даних. Наприклад, можна тренувати моделі машинного навчання на великих наборах даних з відомими шкідливими програмами та використовувати їх для виявлення нових шкідливих програм.

Крім того, для ефективної реалізації алгоритмів виявлення та блокування шкідливих програм на рівні ядра необхідно забезпечити високу продуктивність та низький вплив на роботу системи. Тому важливо оптимізувати алгоритми та використовувати ефективні структури даних та алгоритми обробки даних.

У порівнянні з методами виявлення та блокування шкідливих програм на рівні користувацького простору, методи на рівні ядра забезпечують більшу ефективність та надійність, оскільки вони мають доступ до всіх процесів та даних в системі. Однак вони також потребують більшої кількості знань та навичок в розробці та підтримці, оскільки вони працюють на більш низькому рівні системи та можуть мати великий вплив на її стабільність та безпеку. Крім того, можна використовувати аналізатор вірусів на рівні ядра. Цей інструмент виявляє та блокує шкідливі програми на рівні ядра операційної системи. Аналізатор вірусів на рівні ядра працює безпосередньо з областю пам'яті, що дозволяє виявити навіть ті шкідливі програми, які намагаються приховатися від інших методів виявлення.

Для виявлення шкідливих програм на рівні ядра можна використовувати також технології машинного навчання. Наприклад, можна створити модель, яка використовує нейронні мережі для виявлення шкідливих програм на основі їх характеристик та поведінки. Для цього необхідно побудувати велику базу даних з відомими шкідливими програмами та їх характеристиками. Далі, використовуючи цю базу даних, можна навчити модель виявляти шкідливі програми на рівні ядра операційної системи.

При розробці алгоритмів виявлення та блокування шкідливих програм на рівні ядра важливо враховувати не тільки ефективність, але й вплив на продуктивність системи. Алгоритми повинні працювати швидко та не сповільнювати роботу комп'ютера. Тому важливо розробляти алгоритми, які можна оптимізувати та

пристосувати до конкретних потреб користувачів. Узагалі, виявлення та блокування шкідливих програм на рівні ядра є важливим завданням для забезпечення безпеки комп'ютерних систем. З ростом кількості інтернет-злочинності та вірусів, важливо розробляти нові методи та технології для захисту інформації.

Для запобігання нападу зловмисників на систему, важливо використовувати актуальні версії операційних систем та програмного забезпечення, а також патчі та оновлення безпеки. Це дозволяє уникнути відомих вразливостей, які можуть бути використані зловмисниками для злому системи. Також слід використовувати складні паролі та двофакторну аутентифікацію для захисту облікових записів та важливої інформації. Це дозволяє уникнути несанкціонованого доступу до системи та даних.

З метою попередження атак на систему, також можна використовувати системи моніторингу та інтелектуального аналізу журналів подій. Це дозволяє виявляти незвичну діяльність в системі та забезпечувати швидку реакцію на потенційні загрози. Нарешті, важливо пам'ятати, що виявлення та блокування шкідливих програм на рівні ядра - це складний процес, який вимагає постійного вдосконалення та оновлення. Зловмисники постійно шукають нові способи атаки на систему, тому важливо підтримувати високий рівень свідомості щодо потенційних загроз та захисних заходів.

3.2 Опис системи захисту

Архітектура кінцевої системи буде побудована таким чином, але в контексті даної магістерської роботи буде реалізований перший та другий елемент:

1) Ядро системи - агент, працюючий в просторі ядра
2) Кластер серверів - це набір серверів, здатних при потребі масштабувати себе, які будуть головним інструментом для керування задачами сервісного характеру
3) Клієнт системи - опціональний компонент, працюючий в просторі користувача і виконуючий роль інтерфейсу, скоріше для кінцевих та не досвідчених користувачів

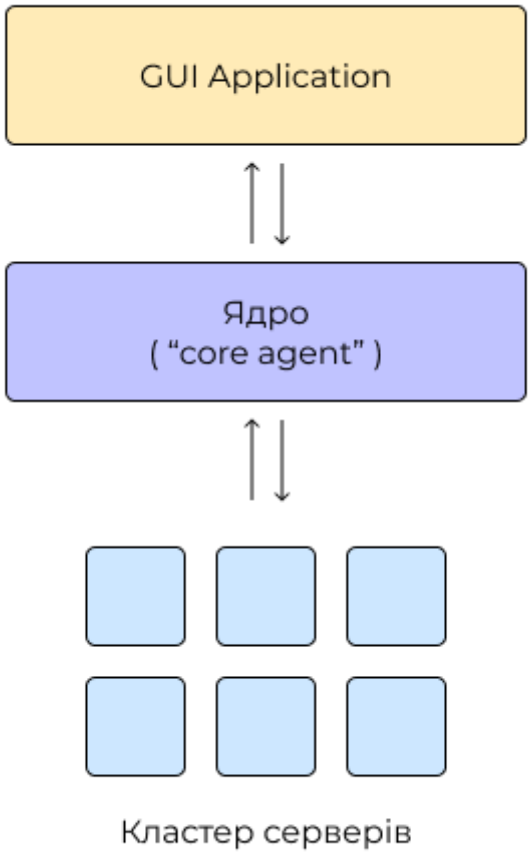

Рисунок 5. Архітектура системи захисту

Ідея даного прототипу полягає в унікальності розробки: доступ з ядра системи надає набагато ширші потенційні можливості для аналізу та моніторингу операційної системи на загрози, технічні деталі яких описані в 1 розділі даної роботи.

Основними задачами ядра (“core agent”) буде моніторинг та інші аналітичні задачі на апаратному та низькорівневому набору обовязків, і також примітивні функціональні. Реалізований він буде за допомогою функціоналу операційної системи Linux, а саме модулів ядра. Комунікація з іншими компонентами прототипу буде реалізована наступна: HTTPS протокол для зв'язку “ядро-кластер” та за допомогою файлової системи (в майбутньому також NFS) для звязку “ядро-клієнт”.

Підставою для обрання саме цих двох варіантів наступні:

1) ефективність технологій, а саме: швидкість, надійність та захищеність

2) ціна рішення (на приклад, на даному етапі розробки прототипу не має сенсу реалізовуваті внутрішє API тому що клієнт буде тільки локальний - хоча потенційно кожен компонент системи в майбутньому буде здатний до масштабування)

Наступним компонентом являється кластер серверів. Сервера будуть обслуговувати клієнтів за допомогою API по запиту - наприклад, доставити оновлення на систему при наявності та потребі/звірити певну сігнатуру файлу/змінити протокол спілкування. У випадку прототипу будуть вирішені елементарні задачі за допомогою кластеру і, все ж таки основна робота буде покладатись на основний компонент - ядро.

Клієнт - це графічний та не обов’язковий елемент архітектури який має ціль для комфортного та зручного презентування процесів захисту та моніторингу системи. Використовувати для написання основного прототипу “ядра” ми будемо мову С.

Використовування цієї мови програмування є обов'язковим після проведеної аналітичної та дослідницької роботи (результати якої підсумовані в попередніх розділах даної роботи), по наступним причинам:

Прямий доступ до апаратного забезпечення: Мова С надає прямий доступ до апаратного забезпечення, що є важливим для розробки модулів ядра Linux. Модулі ядра працюють на низькорівневому рівні, взаємодіючи з апаратурою безпосередньо. Мова С дозволяє ефективно керувати перериваннями, пам'яттю, введенням-виведенням та іншими аспектами апаратної взаємодії.

Вона забезпечує наступні переваги:

1) Широке застосування та підтримка - мова С є однією з найпоширеніших мов програмування, що використовуються для розробки модулів ядра Linux. Багато розробників володіють навичками програмування на С і мають досвід у роботі з цією мовою. Більшість інструментів та бібліотек, необхідних для розробки модулів ядра, також підтримують мову С.
2) Ефективність та продуктивність - мова С є високопродуктивною та ефективною мовою програмування, що дозволяє розробляти швидкі та

ефективні модулі ядра Linux. Вона надає прямий доступ до апаратного забезпечення та мінімізує накладні витрати, пов'язані з виконанням коду. Це особливо важливо для модулів ядра, які повинні працювати швидко та ефективно.

3) Стабільність та сумісність - мова С є стабільною та добре вивіреною мовою програмування. Лінійка компіляторів С забезпечує стандартну підтримку мови та її функціональних можливостей.

Далі, коли архітектура нашого прототипу була описана - увага була зосереджена на методу виявлення вірусної активності. Основним напрямком було обрано сигнатурний пошук.

Сигнатурний пошук є одним із найпоширеніших методів виявлення вірусів. Його суть полягає в порівнянні сигнатур відомих вірусів зі зразками файлів для виявлення відповідних вірусних шаблонів. Далі я опишу більш детально цей метод, його переваги та недоліки, причини для його обрання для прототипу та як реалізувати сигнатурний пошук вірусів на рівні ядра операційної системи Linux.

Як було раніше описано - сигнатурний пошук є одним з основних методів виявлення шкідливих програм. Його принцип полягає в порівнянні сигнатур відомих вірусів зі зразками файлів для виявлення відповідних вірусних шаблонів.

Сигнатура віруса - це унікальна послідовність байтів або рядків, яка характеризує певний тип віруса або його частину. Сигнатура може бути створена шляхом аналізу відомих вірусів або за допомогою спеціалізованих інструментів аналізу коду.

Сигнатурний пошук вірусів використовує базу даних, яка містить сигнатури відомих вірусів. Під час сканування файлової системи або вхідних даних, програма порівнює зразки файлів зі сигнатурами в базі даних. Якщо знаходиться відповідність, файл визнається зараженим.

Однак, сигнатурний пошук має свої обмеження, а саме:

1) Залежність від оновлення бази даних: Для ефективного виявлення нових вірусів потрібні регулярні оновлення бази даних з новими сигнатурами.

2) Не ефективний проти невідомих вірусів: Сигнатурний пошук не може виявляти віруси, для яких немає відповідних сигнатур у базі даних.
3) Ризик фальшивих позитивів та негативів: Існує можливість помилкового визначення чистих файлів зараженими або не визначення вірусів, які мають змінені сигнатури.

Також, для подолання обмежень сигнатурного пошуку існують інші методи виявлення вірусів які не будуть реалізовані в даній роботі, але буде розроблено фундамент для подальшої їх реалізації, такі як:

Поведінковий аналіз: Аналізується не сам файл, а його поведінка під час виконання. Виявлення вірусів здійснюється шляхом спостереження за незвичними або шкідливими діями програми, такими як зміна системних файлів, незаконний доступ до конфіденційної інформації, створення зайвих процесів тощо.

Методи машинного навчання: Використання алгоритмів машинного навчання дозволяє побудувати модель, яка може виявляти віруси на основі аналізу характеристик файлів. Ці характеристики можуть включати поведінкові ознаки, структуру файлу, вміст і т.д.

Евристичний аналіз: Використовуючи правила та евристичні алгоритми, відновлюються характеристики відомих вірусів та шукаються подібні ознаки у нових файлах. Цей метод дозволяє виявляти віруси, які можуть бути невідомими, але мають подібну структуру або поведінку до відомих вірусів.

Системи інтранет-контролю: Використання спеціалізованих апаратних засобів або програмного забезпечення, що контролюють мережевий трафік, дозволяє виявляти та блокувати передачу шкідливих файлів у режимі реального часу.

Комбіновані методи: Часто використовуються комбінації різних методів виявлення вірусів для досягнення кращої ефективності і точності. Наприклад, може застосовуватися комбінація сигнатурного пошуку з поведінковим аналізом або з використанням методів машинного навчання.

Крім того, постійно розвиваються нові методи виявлення вірусів, оскільки шкідливі програми постійно еволюціонують і намагаються уникнути виявлення. Наприклад, використання штучного інтелекту, глибокого навчання, аналізу здатності

до самовідтворення і багато інших технік дозволяють покращити ефективність виявлення вірусів.

Загалом, виявлення вірусів і шкідливих програм - це складний процес, який вимагає використання різних методів і підходів. Комбінація різних методів дозволяє забезпечити більшу ефективність і надійність виявлення шкідливих програм у різних ситуаціях. Саме тому було обрано сигнатурний метод перши для реалізації в прототипу.

Алгоритм реалізації сигнатурного методу є наступним:

Перший етап - це збір сигнатур вірусів.

Створення бази даних сигнатур відомих вірусів, яка містить сигнатури для різних типів вірусів. Сигнатури можуть бути представлені у вигляді хеш-сум, рядків або інших форматів, які можна легко порівняти зі зразками файлів. В нашому випадку ми використаємо відкриту базу даних для початкового збору сігнатур.

Сканування файлової системи - наступний етап. Розробка алгоритму для моніторингу файлової системи на рівні ядра Linux. Перевірка кожного файлу на відповідність сигнатурі вірусів зі збереженими в базі даних - це метод який було реалізовано. Було вирішено, що будь-який новий створений/доданий файл в системі має бути просканованим.

Після сканування файлової системи ми будемо виконувати безпосередню валідацію файлу, а саме наступні кроки:

1) Зчитування зразка файлу з диску у буфер.
2) Порівняння зразка файлу з усіма сигнатурами в базі даних.
3) Якщо знайдено відповідну сигнатуру, файл визнається зараженим і вживаються відповідні заходи (в нашому випадку залежить від конфігурації системи).

Фінальний етап - безпосередньо інтеграція з ядром Linux та конфігурація системи:

1) Розробка модуля ядра, а саме коду на мові Сі який реалізує сигнатурний пошук вірусів.
2) Включення модуля в ядро Linux для його завантаження під час системного запуску або динамічного завантаження.

3) Запуск БД в кластері з примітивним API та базовим набором сигнатур для комунікації з ядром

Більш детально алгоритм виконання логіки прототипу показано на наступній схемі:

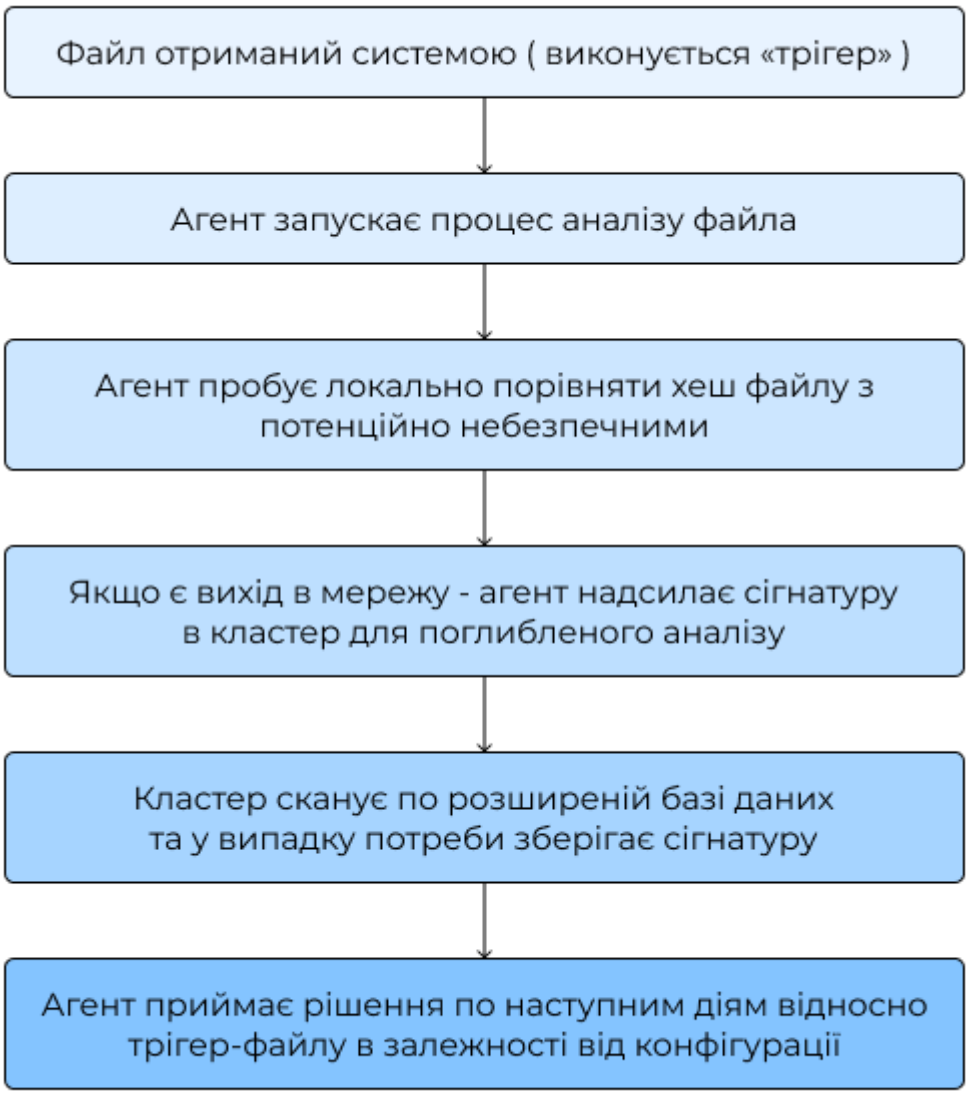

Рисунок 6. Схема алгоритм виконання функціоналу системи захисту

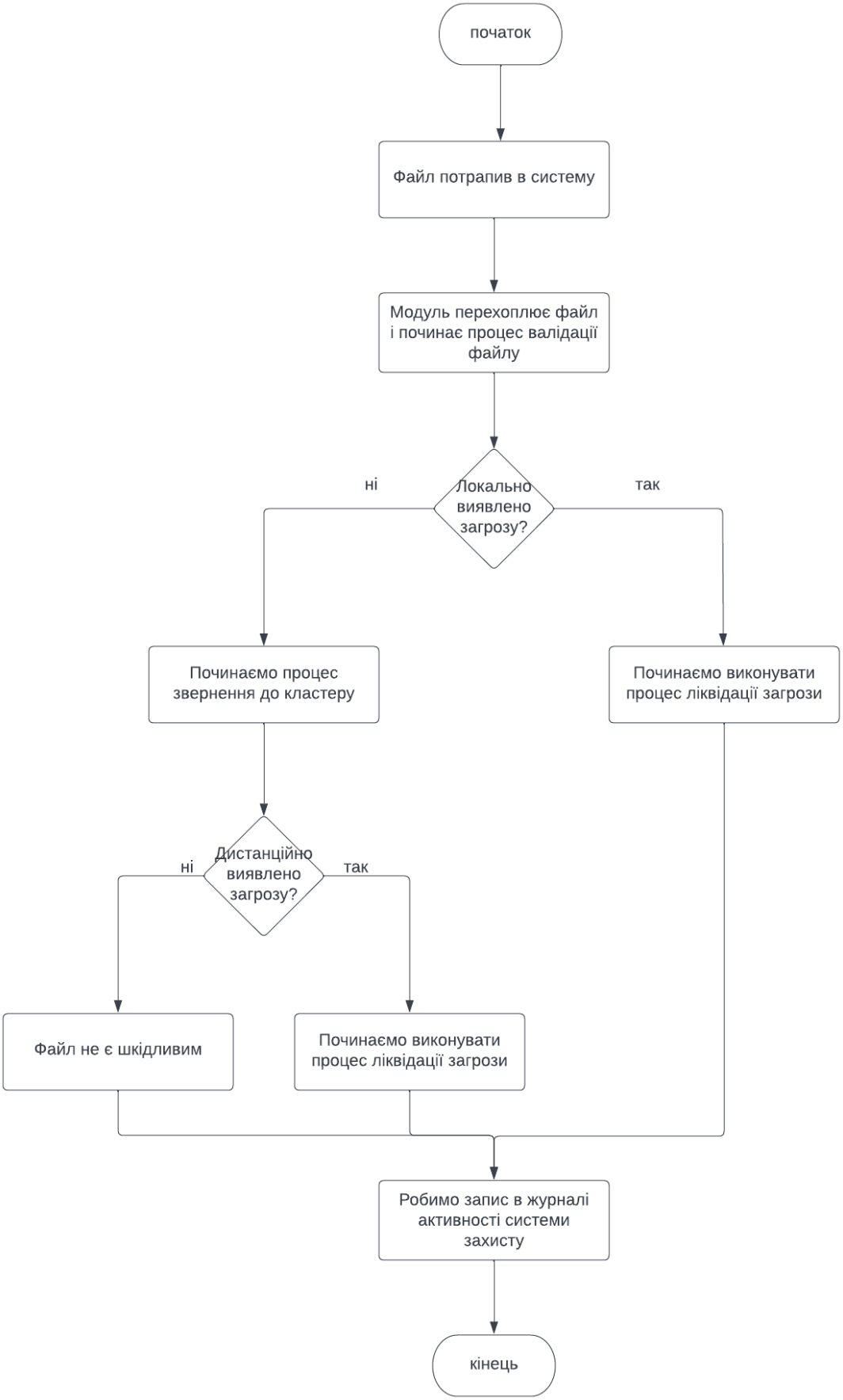

Рисунок 7. Алгоритм виконання функціоналу системи захисту

Приклад файлу, який ми будемо безпосередньо інтегрувати в ядро Linux, з кодом “агента” наступний:

```c
#include <linux/cred.h>
#include <linux/dirent.h>

#define MAX_PATH_LENGTH 256

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Dmytro A");
MODULE_DESCRIPTION("Kernel Antivirus System Prototype 2023");

int is_file_infected(const char *filename) {
    // Internal helper validation
    helper_file_validation(*filename);
}

void scan_filesystem(const char *path) {
    struct file *file;
    struct dir_context context;
    struct dirent *dir_entry;
    char file_path[MAX_PATH_LENGTH];

    // Open the directory specified by path
    // Iterate through all the entries in the directory
    // Check each file for infection
    helper_scan_work(context, *dir_entry, file_path[MAX_PATH_LENGTH]);

    // If the directory entry is a subdirectory, recursively call scan_filesystem to traverse the subdirectory
}

static int __init antivirus_init(void) {
    printk(KERN_INFO "Antivirus module loaded\n");

    // Start scanning the filesystem
    scan_filesystem("/");

    return 0;
}

static void __exit antivirus_exit(void) {
    printk(KERN_INFO "Antivirus module unloaded\n");
}

module_init(antivirus_init);
module_exit(antivirus_exit);
```

Рисунок 8. Приклад main.c для реалізації “ядра”

Ми підключаємо необхідні бібліотеки та задаємо конфігурацію нашого модулю, такі як ліцензія, автор модулю та його опис, максимальне значення для назви файлу. В головному виконуваному файлі ми маємо 4 функції, 2 з яких є обовязковими для реєстрації модуля в системі, і 2 для процесу захисту системи (а саме рекурсиивне сканування директорій та валідування тригер-файлів).

Наступним прикладом буде основна частина функціоналу валідування загрози та звернення до кластеру, в разі виявлення якої ми будемо одразу реагувати.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <curl/curl.h>

size_t write_callback(void *contents, size_t size, size_t nmemb, void *userp) {
    size_t total_size = size * nmemb;
    printf("Received data: %s\n", (char *)contents);
    return total_size;
}

int compare(void *hash, void *options) {
    CURL *curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_DEFAULT);

    curl = curl_easy_init();
    if(curl) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://api.kas-app.com/v1/hashcheck");
        curl_easy_setopt(curl, CURLOPT_POSTFIELDS, "hash=*hash&options=*options");
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);

        res = curl_easy_perform(curl);
        if(res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }

        curl_easy_cleanup(curl);
    }

    curl_global_cleanup();

    return 0;
}
```

Рисунок 9. Приклад helper.c для реалізації "bridge" між сервісними серверами та "ядром"

Ми підключаємо curl бібліотеку для того щоб спростити роботу з генерації запитів до нашого API серверу api.kas-app.com для більш поглибленого аналізу файлів (скорочено від Kernal Antivirus System). В даному фрагменті ми маємо створення запиту та його відправку з наступними параметрами: опції, сам хеш файлу, функція зворотного звернення.

Будемо використовувати ОС та середовище з наступними властивостями та конфігурацією:

1) Операційна система – Linux
2) Linux дистрибутив - Ubuntu 22.04.2 LTS
3) Компілятор - GCC – 11.2.0
4) Версія мови Ci – C11
5) VirtualBox – 6.1.26
6) Протокол комунікації - HTTP/2
7) БД для зберігання сігнатур - PostgreSQL 15.3
8) Кластерне API – Python 3.10.11

Далі завантажимо наш модуль в систему та протестуємо що він працює.

Рисунок 10. Компіляція системи захисту

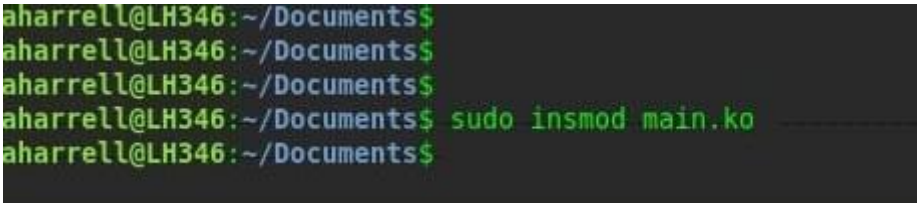

Рисунок 11. Інтеграція функціоналу в ядро Linux

Система захисту була успішно інтегрована, ніяких помилок ми не отримали. Далі ми бачимо як наше “ядро” було завантажене в систему

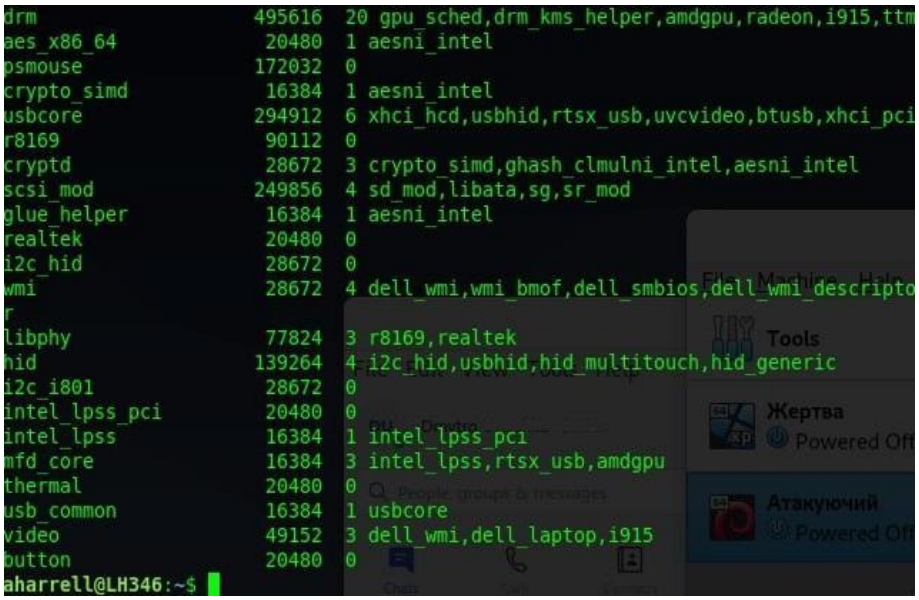

Рисунок 12. Тестування роботи модуля

Після запуску системи було протестовано саме поведінку програми, всі компоненти були запущені безпосередньо на віртуальних серверах для забезпечення безпеки.

Моя задача полягала у тестуванні системи захисту і додаванні вірусу до операційної системи різними способами. Було вирішено дотримуватися наступних кроків:

1) Підготовка вірусного зразка: Був обраний вірусний зразок для тестування. Це може бути будь який шкідливий файл, скрипт або програма, яка має потенціал пошкодження системи або виконання небажаних дій.
2) Забезпечення безпеки: Перед тестуванням вірусного зразка важливо взяти до уваги безпеку і уникнути поширення вірусу. Тому тестування було проведено в захищеному ізольованому середовищі, в якому немає зв'язку з зовніми системами або мережами. В цій роботі було вирішено використати віртуальні машини.
3) Додавання вірусу до системи: За допомогою широких можливостей операційної системи, різноманітними способами був доданий вірусний зразок до серверу. Потенційно це може бути будь-який варіант: завантаження вірусного файлу на системний диск або передачу через мережу до вразливого сервісу, тощо.
4) Спостереження за реакцією системи: Після додавання вірусу була проаналізована поведінка системи захист. Прототип успішно валідував загрозу та видаляв потенційні небезпечні файли з операційної системи з дуже високою швидкістю. Також система захисту вела журнал поведінки, по якому можливо звірити алгоритм дії нашого прототипу.
5) Перевірка результатів: Після завершення тестування було виявлено що система захисту успішно виявляє шкідливі файли та не завдає шкоди звичайним файлам. Також система не завантажує ресурс серверу як звичайний антивірус, що досягається перевіркою на рівні ядра операційної системи й робить прототип дуже ефективним для вирішення задач захисту серверів.

Важливо додати, що тестування вірусів було проведено повністю відповідно до стандартів в умовах повного контролю ситуації та превентивних захисних дій, а

саме ізольованого простору, віртуальної інфраструктури та встановлення на зовнішньому рівні додаткових систем захисту.

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено систему захисту інформації на рівні ядра Linux. Для досягнення поставленої мети були вирішені такі задачі:

- Аналіз існуючих антивірусних систем та їх можливостей.
- Розробка алгоритмів виявлення та блокування шкідливих програм на рівні ядра.
- Реалізація прототипу антивірусної системи з використанням мов програмування C.
- Оцінка ефективності та можливостей запропонованої антивірусної системи.

В цілому, можна зробити ще один висновок, що розроблена програма є ефективним засобом для явного та потенційного розв'язання поставлених задач. Вона має широкі можливості для обробки та візуалізації даних, захисту систем та аналізу вразливостей. Програма може бути використана у різних галузях, крім того, дана програма може бути корисною для комерційних проектів. Система має великий потенціал для розвитку та покращень функціоналу.

Отже, дипломна робота була успішною та в ході роботи було успішно виконано всі завдання. Програма для захисту інформації на рівні ядра була розроблена та протестована на реальних системах, що підтверджує її ефективність та потенціал. Результати роботи можуть бути використані для подальшого розвитку та вдосконалення програми, а також для використання її в практичних проектах.

ОФОРМЛЕННЯ

списку використаних джерел за ДСТУ 8302:2015

та відповідно до наказу МОН України від 12.01.2017 р. № 40 (Дод. 3)

КНИГИ

Однотомні видання

Один автор

1. Rochkind, M. (2013). Advanced UNIX Programming. New York: Addison-Wesley Professional. (ISBN 978-0321637734)

2. Love, R. (2010). Linux Kernel Development. Indianapolis: Pearson Education. (ISBN 978-0672329463)

3. SoCC 10: Proceedings of the 1st ACM symposium on Cloud computing, Hellerstein, Joseph M. - N. Y.: ACM, 2010. - ISBN 978-1-4503-0036-0.

4. Gillam, Lee. Cloud Computing: Principles, Systems and Applications / Nick Antonopoulos, Lee Gillam. — L.: Springer, 2010. — p. 379.

5. Баранов А.П. Чи можна захистити в «хмарі» конфіденційну інформацію? А. Баранов // Системи високої доступності. — 2012. — Т. 8. — № 2. — С. 12-15.

6. Tanenbaum, A. S., Woodhull, A. S. (2015). Операційні системи: проектування та реалізація. Київ: Видавництво "Довкілля". (ISBN 978-617-7089-68-9)

7. Kerrisk, M. (2010). The Linux Programming Interface: A Linux and UNIX System Programming Handbook. San Francisco: No Starch Press. (ISBN 978-1593272203)

8. Johnson L.K. Bombs, bugs, drugs and thugs: intelligence and America’s quest for security. New York; London: New York University Press, 2000. 326 p.

Два автори

9. Комаров, Д., Штурман, І. (2020). Антивірусна безпека в середовищі Linux. Київ: Видавництво "Наш Формат". (ISBN 978-617-7847-23-1)

10. Безпека життєдіяльності. Безпека технологічних процесів і виробництв (Охорона праці): Навч. посібник для вузів // П.П. Кукін, Е.А. Підгорний та ін. - М .: Висш.шк., 1999. — с. 318.

11. Stevens, R., Rago, W. (2013). Advanced Programming in the UNIX Environment. Boston: Addison-Wesley Professional. (ISBN 978-0321637734)

12. Гаррiс, Д., Бланшетт, Д. (2012). Linux Kernel Networking: Implementation and Theory. San Francisco: Apress. (ISBN 978-1430261964)

Три автори

13. Бабаш А.В., Гольєв Ю.І., Ларін Д.А., Шанкін Г.П. Про розвиток криптографії в XIX столітті // Захист інформації. Конфидент. — 2003. — №5 — с. 90-96.

14. Сичевський В.В., Харитонов Є.І., Олєйніков Д.О. Науково-практичний коментар до розділу I Особливої частини Кримінального кодексу України (Злочини проти основ національної безпеки України). Харків: Право, 2016. 232 с.

ІНШІ ВИДАННЯ

Стандарти

15. ДСТУ 7152:2010. Видання. Оформлення публікацій у журналах і збірниках. [Чинний від 2010-02-18]. Вид. офіц. Київ, 2010. 16 с. (Інформація та документація).

16. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень українською мовою. Загальні вимоги та правила (ISO 4:1984, NEQ; ISO 832:1994, NEQ). [На заміну ДСТУ 3582-97; чинний від 2013-08-22]. Вид. офіц. Київ: Мінекономрозвитку України, 2014. 15 с. (Інформація та документація).

Архівні документи

17. Матеріали Ради Народних комісарів Української Народної Республіки. *ЦДАВО України* (Центр. держ. архів вищ. органів влади та упр. України). Ф. 1061. Оп. 1. Спр. 8-12. Копія; Ф. 1063. Оп. 3. Спр. 1-3.

18. Наукове товариство ім. Шевченка. *Львів. наук. б-ка ім. В. Стефаника НАН України*. Ф. 1. Оп. 1. Спр. 78. Арк. 1-7.

ЕЛЕКТРОННІ РЕСУРСИ

19. Офіційний веб-сайт Linux [Електронний ресурс]. Дата останнього доступу: 10 січня 2023. URL: http://www.kernel.org

20. Офіційний веб-сайт ядра Linux [Електронний ресурс]. Дата останнього доступу: 2 травня 2023. URL: http://www.linux.org

21. Документація ядра Linux [Електронний ресурс]. Дата останнього доступу: 17 березня 2023. URL: http://www.kernel.org/doc

22. Веб-сайт проекту ClamAV [Електронний ресурс]. Дата останнього доступу: 10 травня 2023. URL: http://www.clamav.net

23. Trammell A. Magic: The gathering in material and virtual space: An ethnographic approach toward understanding players who dislike online play. *Meaningful Play 2010*: October 21-23, 2010, East Lansing, MI. URL: http://meaningfulplay.msu.edu/proceedings2010/mp2010_paper_42.pdf (Last accessed: 17.03.2017).

24. Веб-сайт проекту Sophos Antivirus for Linux [Електронний ресурс]. Дата останнього доступу: 10 травня 2023. URL: http://www.sophos.com/linux

25. Офіційний веб-сайт проекту GNU [Електронний ресурс]. Дата останнього доступу: 3 травня 2023. URL: http://www.gnu.org

26. Веб-сайт проекту ClamTk [Електронний ресурс]. Дата останнього доступу: 27 січня 2023. URL: http://www.clamtk.org

27. Офіційний веб-сайт проекту AVG Antivirus for Linux/FreeBSD [Електронний ресурс]. Дата останнього доступу: 3 травня 2023. URL: http://www.avg.com/linux

28. Офіційний веб-сайт проекту Bitdefender Antivirus Scanner for Unices [Електронний ресурс]. Дата останнього доступу: 6 березня 2023. URL: http://www.bitdefender.com/unices

29. Офіційний веб-сайт проекту F-Prot Antivirus for Linux [Електронний ресурс]. Дата останнього доступу: 1 травня. URL: http://www.clamtk.org

30. Arif Mohamed. A history of cloud computing [Електронний ресурс]. – Режим доступу: http://www.computerweekly.com/feature/A-history-of-cloud-computing

31. Peter M. Mell, Timothy Grance The NIST Definition of Cloud Computing [Електронний ресурс]. – Режим доступу: https://www.nist.gov/node/568586

ЗАКОНОДАВЧІ ТА НОРМАТИВНІ ДОКУМЕНТИ

32. Конституція України: станом на 1 верес. 2022 р.: відповідає офіц. тексту. Харків: Право, 2016. 82 с.

33. Правова основа діяльності органів державної влади: зб. нормат. актів / упоряд. П. М. Любченко. Харків: ФІНН, 2010. 303 с.

34. Конституційний Суд України: рішення, висновки / відп. ред. А.С. Головін; уклад.: К.О. Пігнаста, О.І. Кравченко. Київ: Логос, 2011. Кн. 10. 431 с.

35. Кримінальний кодекс України: Закон України від 05.04.2001 р. № 2341-III. *Відомості Верховної Ради України.* 2001. № 25-26. Ст. 131.

36. Про порядок класифікації надзвичайних ситуацій: Постанова Кабінету Міністрів України від 15.06.1998 р. № 1099. *Офіційний вісник України*. 1998. № 28. Ст. 1062. Про правовий режим воєнного стану: Закон України від 12.05.2015 р. № 389-VIII. *Голос України*. 2015. 10 черв. (№ 101). С. 4.