МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНЕ НЕКОМЕРЦІЙНЕ ПІДПРИЄМСТВО «ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
Кафедра Комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
Аліна САВЧЕНКО
«_____» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)
ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ
«МАГІСТРА»
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ «ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

Тема: «Інтелектуальна система аналізу мережевого трафіку»

Виконавець: Галкіна Марія Валеріївна

Керівник: проф. Зіатдінов Юрій Кашафович

Нормконтролер: Ігор РАЙЧЕВ

Київ 2024

ДЕРЖАВНЕ НЕКОМЕРЦІЙНЕ ПІДПРИЄМСТВО «ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Факультет *Комп'ютерних наук та технологій*

Кафедра *Комп'ютерних інформаційних технологій*

Галузь знань, спеціальність, освітньо-професійна програма: 12 «Інформаційні технології», 122 «Комп'ютерні науки», «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Аліна САВЧЕНКО

«_____» _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студентки

Галкіної Марії Валеріївни

(прізвище, ім'я, по батькові)

1. **Тема:** «Інтелектуальна система аналізу мережевого трафіку», затверджена наказом в.о. ректора від 06.09.2024р. за №1782/ст.
2. **Термін виконання роботи:** з 07.09.2024р. по 02.12.2024 р.
3. **Вихідні дані до роботи:** методологія побудови процесу аналізу мережевого трафіку.
4. **Зміст пояснювальної записки (перелік питань, що підлягають розробці):** характеристики мережевого трафіку, огляд методів до аналізу мережевого трафіку, розгляд та розробка інтелектуальної системи, надання рекомендацій щодо використання системи аналізу мережевого трафіку.
5. **Перелік обов'язкового графічного матеріалу:** інформативні рисунки, діаграми, таблиці.

6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Аналіз предметної області, постановка задачі на виконання кваліфікаційної роботи.	07.09.2024 – 11.09.2024	
2.	Збір і аналіз наукових джерел.	12.09.2024 – 13.09.2024	
3.	Проведення консультацій з науковим керівником.	14.09.2024 – 15.09.2024	
4.	Написання Розділу 1 пояснювальної записки.	16.09.2024 – 30.09.2024	
5.	Написання Розділу 2 пояснювальної записки.	01.10.2024 – 13.11.2024	
6.	Написання Розділу 3 пояснювальної записки.	14.11.2024 – 19.11.2024	
7.	Редагування пояснювальної записки.	20.11.2024 – 26.11.2024	
8.	Робота над доповіддю та презентацією. Передзахист кваліфікаційної роботи.	27.11.2024 – 29.11.2024	
9.	Підготовка і передача матеріалів кваліфікаційної роботи керівнику для написання відгуку та секретарю ДЕК.	30.11.2024 – 02.12.2024	

7. Дата видачі завдання: «07» вересня 2024 р.

Керівник дипломної роботи _____ Юрій ЗІАТДІНОВ

(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Марія ГАЛКІНА

(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи “Інтелектуальна система аналізу мережевого трафіку” виконана на 79 сторінках і містить 8 рисунків та 9 таблиць. Список бібліографічних посилань складається з 23 найменувань.

МЕРЕЖЕВИЙ ТРАФІК, ПРОТОКОЛ, АНАЛІЗ МЕРЕЖЕВОГО ТРАФІКУ, ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ.

Об’єкт дослідження - мережевий трафік та процеси його аналізу для виявлення аномалій та кіберзагроз.

Предмет дослідження - інтелектуальні системи аналізу мережевого трафіку, засновані на алгоритмах машинного навчання та штучного інтелекту.

Методи дослідження: аналіз літератури та наукових джерел з тематики мережевого трафіку та інтелектуальних систем. Моделювання мережевого трафіку для тестування системи.

Мета роботи. Розробка інтелектуальної системи для аналізу мережевого трафіку з виявлення аномалій та підвищення ефективності кібербезпеки.

Актуальність теми дослідження визначається зростаючою важливістю кібербезпеки в умовах сучасного світу. У зв'язку з постійним збільшенням обсягу даних, що передаються через мережі, і складністю атак на інформаційні системи, традиційні методи захисту не завжди забезпечують необхідний рівень безпеки. Інтелектуальні системи аналізу мережевого трафіку дозволяють виявляти аномалії, відстежувати підозрілі дії та оперативно реагувати на потенційні загрози. Вони застосовують методи штучного інтелекту та машинного навчання для автоматизації процесу аналізу, що підвищує ефективність і точність виявлення небезпек.

Такі системи є необхідними в умовах постійного зростання складності кіберзагроз, і їх розробка та впровадження можуть значно підвищити рівень

захищеності мережевої інфраструктури як у приватних, так і в державних установах.

Подальший розвиток - отримані результати можуть бути використані для розробки та впровадження ефективних систем кіберзахисту в реальних мережах. Вони також сприятимуть підвищенню точності виявлення загроз, оптимізації моніторингу мережевого трафіку та забезпеченню безпеки інформаційних систем

ЗМІСТ

ВСТУП 7
РОЗДІЛ 1.ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ ЗА ДОПОМОГОЮ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ 8
1.1. Основи аналізу мережевого трафіку: поняття, методи та задач 8
1.2. Інтелектуальні системи в аналізі мережевого трафіку 16
1.3. Алгоритми машинного навчання та штучного інтелекту для аналізу мережевого трафіку 25
1.4. Висновки до розділу 1 37
РОЗДІЛ 2.ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ 39
2.1. Аналіз та засоби моніторингу мережевого трафіку 39
2.2. Моделювання мережевого трафіку 49
2.3. Функціонально-вартісний аналіз 60
2.4. Висновки до розділу 2 66
РОЗДІЛ 3.РЕКОМЕНДАЦІЇ ЩОДО ВИКОРИСТАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ 68
3.1. Оптимізація параметрів системи для підвищення точності аналізу 68
3.2. Використання системи в різних типах мереж 70
3.3. Рекомендації щодо подальшого розвитку та інтеграції системи з іншими рішеннями безпеки 73
3.4. Висновки до розділу 3 76
ВИСНОВКИ 77
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ 78

ВСТУП

У сучасному світі стрімкий розвиток інформаційних технологій і збільшення мережевого трафіку створюють нові виклики в області інформаційної безпеки. Кіберзагрози стають все більш досконалими, і необхідно впроваджувати ефективні системи для аналізу та виявлення аномалій у мережевих даних. Традиційні підходи, засновані на статичних правилах і підписах, не завжди адекватно реагують на нові типи атак або швидкі зміни в архітектурі мережі.

В останні роки активно розвиваються методи аналізу мережевого трафіку, засновані на інтелектуальних алгоритмах, таких як машинне навчання і штучний інтелект. Використання таких технологій значно підвищує точність і швидкість виявлення аномалій, забезпечує адаптацію до нових загроз і підвищує рівень захисту корпоративних систем.

Метою дослідження є розробка інтелектуальних систем аналізу мережевого трафіку, які виявляють потенційні загрози, аналізують поведінку користувачів і попереджають про можливі кіберінциденти. Основна увага буде приділена впровадженню інноваційних алгоритмів і створенню системної архітектури, що відповідає останнім вимогам до продуктивності, масштабованості та інтеграції.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ ЗА ДОПОМОГОЮ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

1.1.Основи аналізу мережевого трафіку: поняття, методи та задач

Мережевий трафік є фундаментальним елементом сучасних інформаційних систем та мереж. Це потік даних, який передається через комп'ютерні мережі між різними пристроями: комп'ютерами, серверами, маршрутизаторами, мобільними пристроями та іншими учасниками мережевої інфраструктури. Основна функція мережевого трафіку полягає у забезпеченні передачі інформації між віддаленими системами, що дозволяє обмінюватися даними в реальному часі або за запитом користувачів.

Мережевий трафік – це обсяг даних, який передається через мережу за певний проміжок часу. Він включає в себе всі види інформації, що пересилаються через мережу: файли, відео, аудіо, веб-запити, електронні листи тощо. Трафік вимірюється у кількості переданих біт або байтів за секунду (bps, bytes per second) і є одним з основних показників функціонування мережі. Його аналіз дозволяє оцінити навантаження на мережу, виявити аномалії або можливі загрози, а також оптимізувати роботу мережі.

Мережевий трафік можна класифікувати за різними критеріями, одним із яких є масштаб і характер передачі даних. Існують три основні типи трафіку: локальний, глобальний та внутрішньомережевий.

Локальний трафік – це потік даних, який циркулює в межах локальної мережі (Local Area Network, LAN). Локальна мережа зазвичай обмежена невеликим географічним простором, наприклад, офісом, будівлею або домашньою мережею. Основними перевагами локального трафіку є висока швидкість передачі даних та низька затримка, оскільки всі пристрої, що обмінюються інформацією, знаходяться поруч. Локальні мережі часто використовуються для спільного доступу до файлів, принтерів та інших ресурсів між користувачами однієї організації.

Глобальний трафік передається через широкомасштабні мережі (Wide Area Network, WAN), які охоплюють великі географічні відстані. Це можуть бути міжміські, міжнародні або навіть міжконтинентальні мережі. Приклади глобального трафіку включають обмін даними через Інтернет, між різними філіями корпорацій або через VPN-з’єднання. Глобальний трафік зазвичай має більшу затримку та нижчу швидкість порівняно з локальним через більші відстані і кількість проміжних вузлів, через які проходять дані. Проте сучасні технології, такі як оптичні кабелі та супутникові системи, дозволяють значно підвищити ефективність передачі таких даних.

Внутрішньомережевий трафік, або VLAN (Virtual Local Area Network) трафік, використовується для передачі даних між сегментами мережі, що належать до однієї фізичної інфраструктури, але логічно відокремлені один від одного. Це дозволяє створити віртуальні сегменти всередині однієї фізичної мережі, наприклад, для різних відділів в організації, які мають різні вимоги до безпеки або продуктивності. Внутрішньомережевий трафік дозволяє розподіляти ресурси ефективніше, підвищувати безпеку мережі та зменшувати навантаження на фізичні компоненти.

Мережевий трафік є невід'ємною складовою сучасного цифрового світу. Локальний, глобальний та внутрішньомережевий трафік виконують різні функції у системах передачі даних і забезпечують ефективну роботу як локальних, так і глобальних мереж. В умовах зростання кіберзагроз і збільшення обсягів переданих даних аналіз і оптимізація мережевого трафіку

є важливим аспектом у забезпеченні надійної роботи мережевої інфраструктури.

Обсяг мережевого трафіку визначає кількість даних, які передаються через мережу за певний проміжок часу. Він вимірюється в байтах, кілобайтах, мегабайтах або гігабайтах і є важливим показником навантаження на мережу. Обсяг трафіку може варіюватися в залежності від типу даних, що передаються, та інтенсивності їх обміну. Наприклад, передача великих відеофайлів або завантаження програмного забезпечення створює значний обсяг трафіку, що може вплинути на швидкість і ефективність мережі. Аналіз обсягу трафіку дозволяє виявити перевантаження мережі, оптимізувати використання ресурсів та планувати потреби у пропускній здатності.

Швидкість мережевого трафіку характеризує швидкість передачі даних через мережу і вимірюється в біт/секунду (bps), кілобіт/секунду (kbps), мегабіт/секунду (Mbps) або гігабіт/секунду (Gbps). Швидкість є критичним фактором для визначення ефективності мережі, оскільки вона впливає на те, як швидко дані досягають призначення. Висока швидкість передачі даних забезпечує швидкий доступ до інформації, зменшує затримки і покращує загальний досвід користувачів. Однак, швидкість мережі може бути обмежена різними факторами, такими як пропускна здатність каналу, затримки передачі даних і наявність вузьких місць у мережевій інфраструктурі. Регулярний моніторинг швидкості трафіку допомагає виявити та усунути проблеми, що впливають на продуктивність мережі.

Протоколи мережевого трафіку визначають правила і формати для обміну даними між пристроями в мережі. Вони забезпечують коректну передачу та прийом інформації і гарантують, що дані будуть інтерпретовані правильно. Існує безліч протоколів, кожен з яких має специфічні функції:

- TCP (Transmission Control Protocol). Забезпечує надійну, орієнтовану на з'єднання передачу даних, контролюючи їх доставку і коригуючи помилки. TCP гарантує, що всі пакети даних будуть доставлені в правильному порядку і без помилок;

- UDP (User Datagram Protocol). Працює без встановлення з'єднання і надає менш надійну, але швидшу передачу даних. UDP використовуються для додатків, де важлива швидкість передачі, наприклад, у відеоконференціях і онлайн-іграх;

- HTTP/HTTPS (Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure). Протоколи, що забезпечують передачу даних між веб-серверами і браузерами. HTTPS є захищеною версією HTTP, яка використовує шифрування для захисту переданих даних;

- FTP (File Transfer Protocol). Протокол для передачі файлів між клієнтом і сервером, що дозволяє завантажувати і завантажувати великі обсяги даних;

- ICMP (Internet Control Message Protocol). Використовується для передачі контрольних повідомлень, наприклад, для перевірки доступності мережевих пристроїв через команди ping.

Обсяг, швидкість і протоколи мережевого трафіку є основними характеристиками, які визначають ефективність і продуктивність мережі. Розуміння цих характеристик є ключовим для управління мережами, забезпечення їх безперебійної роботи і оптимізації процесів обміну даними. Своєчасний моніторинг і аналіз цих параметрів допомагає підтримувати високу якість мережевих послуг, забезпечувати надійність і безпеку інформаційних систем, а також відповідати на виклики, що постають у світі сучасних технологій.

Аналіз мережевого трафіку є критично важливим для підтримки ефективності та безпеки сучасних інформаційних систем. В умовах зростаючої складності і обсягу даних, а також зростання кіберзагроз, важливо мати інструменти та методи для всебічного моніторингу мережі. Серед основних завдань аналізу мережевого трафіку виділяються виявлення аномалій, оптимізація роботи мережі, моніторинг безпеки та визначення вузьких місць у мережі. Це дозволяє забезпечити безперебійність функціонування мережі та захист від потенційних загроз.

Однією з ключових задач аналізу мережевого трафіку є виявлення аномалій та підозрілої активності. Аномалії можуть вказувати на різні проблеми, від технічних збоїв до потенційних атак. Виявлення відхилень від нормальної поведінки трафіку дозволяє оперативно реагувати на несанкціоновані дії або порушення, що можуть загрожувати цілісності системи.

Аналіз трафіку допомагає виявити нетипові патерни або різке збільшення обсягу трафіку, які можуть бути ознаками атаки типу DDoS (розподілена атака на відмову в обслуговуванні) або інших форм зловмисного втручання. Системи моніторингу часто використовують алгоритми машинного навчання для автоматичного виявлення аномалій, що дозволяє зменшити залежність від людського фактору і підвищити швидкість реакції на загрози.

Оптимізація роботи мережі є ще однією важливою задачею аналізу мережевого трафіку. Зростання обсягу даних та навантаження на мережу може призвести до зниження швидкості передачі даних, затримок і збоїв. Ефективний аналіз мережевого трафіку дозволяє виявити проблеми, які можуть вплинути на продуктивність мережі, такі як перевантаження каналів, неефективне використання ресурсів або неправильна конфігурація мережевих пристроїв.

Завдяки аналізу, мережеві адміністратори можуть здійснювати оптимізацію параметрів мережі, таких як маршрутизація трафіку, балансування навантаження та управління пропускною здатністю. Це забезпечує більш ефективне використання ресурсів і покращує загальний рівень продуктивності мережі.

Моніторинг безпеки і попередження кіберзагроз є критично важливими аспектами аналізу мережевого трафіку. В умовах зростання кількості і складності кіберзагроз, необхідно забезпечити постійний контроль за мережевим трафіком для своєчасного виявлення і нейтралізації потенційних загроз.

Системи моніторингу безпеки використовують різноманітні методи для ідентифікації кіберзагроз, включаючи аналіз поведінки трафіку, виявлення шкідливих програм і перевірку на наявність уразливостей. Раннє виявлення загроз дозволяє вжити необхідні заходи для запобігання їх реалізації, таких як блокування зловмисного трафіку, ізоляція уражених частин мережі або оновлення систем безпеки.

Визначення вузьких місць у мережі є важливою частиною аналізу мережевого трафіку. Вузькі місця (bottlenecks) – це ділянки мережі, які обмежують її продуктивність або швидкість передачі даних. Вони можуть бути викликані недостатньою пропускною здатністю каналів зв'язку, перевантаженням серверів або обмеженнями у мережевих пристроях.

Аналіз мережевого трафіку дозволяє точно визначити місця, де відбувається затримка або зниження продуктивності, що дозволяє здійснювати необхідні корективи. Це може включати модернізацію обладнання, перепланування архітектури мережі або оптимізацію конфігурації для підвищення ефективності та зменшення затримок.

Задачі аналізу мережевого трафіку, такі як виявлення аномалій, оптимізація роботи мережі, моніторинг безпеки та визначення вузьких місць, є критично важливими для підтримки стабільності і безпеки інформаційних систем. Ефективне виконання цих завдань забезпечує не лише безперебійність роботи мережі, але й захист від потенційних загроз, підвищення продуктивності і зниження витрат на обслуговування мережевої інфраструктури. В умовах швидкого розвитку технологій і зростання обсягу даних, ці завдання стають ще більш актуальними, що робить їх незамінними для сучасного управління мережами.

Аналіз мережевого трафіку є критичним компонентом управління інформаційними системами, що дозволяє забезпечити їхню ефективність і безпеку. Сучасні методи аналізу трафіку охоплюють різноманітні підходи, серед яких можна виділити пасивний і активний моніторинг, а також аналіз на основі сигнатур і аномалій. Кожен з цих методів має свої особливості та

застосування, що дозволяє ефективно виявляти проблеми та забезпечувати безперебійність роботи мережі.

Пасивний моніторинг трафіку – це метод, при якому аналіз даних здійснюється без втручання у процес їх передачі. Він включає в себе збір і аналіз інформації про трафік, що вже проходить через мережу, без додаткового навантаження на систему.

Дзеркальні порти (port mirroring) і мережеві зонд (network probes) є основними інструментами пасивного моніторингу. Дзеркальний порт дозволяє копіювати трафік з одного порту комутатора на інший порт, куди підключається моніторингове обладнання або програмне забезпечення. Це дає можливість аналізувати дані без впливу на сам процес їх передачі.

Мережеві зонд, з іншого боку, це спеціалізовані пристрої або програми, які підключаються до мережі для збору і аналізу трафіку. Вони можуть бути обладнані різними сенсорами для моніторингу параметрів трафіку, таких як затримка, пропускна здатність і пакетні втрати. Пасивний моніторинг допомагає виявляти проблеми, не порушуючи звичайного функціонування мережі, що робить його ідеальним для безперервного контролю.

Активний моніторинг передбачає активну участь у процесі аналізу, включаючи введення додаткових пакетів у мережу для перевірки її роботи.

Один з основних методів активного моніторингу – тестування пропускної здатності мережі. Це дозволяє перевірити, скільки даних може бути передано через мережу за певний проміжок часу, що є критично важливим для виявлення можливих вузьких місць. Інструменти для тестування пропускної здатності можуть бути використані для симуляції навантаження і вимірювання фактичної швидкості передачі даних.

Активний моніторинг також включає тестування на втрати пакетів. У процесі тестування додаються спеціально сформовані пакети в мережу, і моніторингова система фіксує, які пакети були успішно передані, а які загублені або спотворені. Це дозволяє виявити проблеми з якістю зв’язку і своєчасно вжити заходів для їх усунення.

Аналіз на основі сигнатур передбачає виявлення атак і аномалій за відомими шаблонами або сигнатурами. Цей метод використовує заздалегідь визначені характеристики, що відповідають певним типам загроз або аномалій, для виявлення потенційних проблем у мережевому трафіку.

Антивірусні та мережеві системи безпеки, що використовують аналіз на основі сигнатур, можуть швидко ідентифікувати відомі атаки або зловмисні програми шляхом порівняння трафіку з базами даних сигнатур. Цей метод є дуже ефективним для виявлення вже відомих загроз, але його обмеженням є неспроможність виявляти нові або невідомі атаки, які не мають визначених сигнатур.

Аналіз на основі аномалій порівнює нормальну поведінку мережевого трафіку з аномальними відхиленнями, що можуть свідчити про проблеми або загрози.

Цей метод включає створення профілю нормальної поведінки трафіку, а потім постійний моніторинг і порівняння фактичного трафіку з цим профілем. Виявлені відхилення від нормального профілю можуть бути ознаками аномалій або атак. Однак, цей метод може призвести до великої кількості помилкових спрацьовувань, тому важливо мати добре налаштовану систему порівняння для зменшення помилкових тривог.

Методи аналізу мережевого трафіку – пасивний і активний моніторинг, а також аналіз на основі сигнатур і аномалій – забезпечують всебічний підхід до моніторингу та управління мережами. Пасивний моніторинг дозволяє безперервно контролювати трафік без порушення його передачі, активний моніторинг допомагає виявити проблеми і перевірити пропускну здатність мережі, а аналіз на основі сигнатур і аномалій допомагає забезпечити безпеку і виявити потенційні загрози. Поєднання цих методів дозволяє забезпечити ефективність і надійність сучасних мережевих інфраструктур.

Отже, аналіз мережевого трафіку є критично важливим для підтримки функціонування та безпеки сучасних інформаційних систем. Основи цього

аналізу охоплюють поняття мережевого трафіку, його характеристики, методи моніторингу та ключові задачі, що визначають його ефективність.

Мережевий трафік представляє собою дані, що передаються через мережу, і включає інформацію про те, як ці дані переміщуються між пристроями. Важливими аспектами є обсяг трафіку (кількість даних, що передаються), швидкість передачі (швидкість, з якою дані досягають призначення), і протоколи (правила для обміну даними між пристроями).

Аналіз мережевого трафіку є комплексним процесом, що включає як технічні, так і стратегічні аспекти. Застосування різних методів і вирішення ключових задач забезпечують ефективне управління, оптимізацію і захист мережевих ресурсів, що є основою стабільної і безпечної роботи сучасних інформаційних систем.

1.2. Інтелектуальні системи в аналізі мережевого трафіку

У сучасному світі інформаційних технологій, де дані стають новою валютою, забезпечення ефективного і безпечного управління мережами є критично важливим. Інтелектуальні системи, які використовують передові технології штучного інтелекту та машинного навчання, відіграють ключову роль у цьому процесі. Розглянемо визначення інтелектуальних систем, їх роль в аналізі мережевого трафіку та зростаючу значимість цих систем у сучасних мережах.

Інтелектуальні системи – це складні комп'ютерні системи, які використовують алгоритми штучного інтелекту (ШІ) та машинного навчання для виконання завдань, що вимагають інтелектуальних зусиль, подібних до людських. Вони здатні обробляти та аналізувати великі обсяги даних, вчитися на основі минулого досвіду, адаптуватися до нових умов і робити прогнози. Інтелектуальні системи включають в себе різноманітні підходи, такі як нейронні мережі, алгоритми кластеризації, системи рекомендацій і багато

інших технологій, що дозволяють автоматизувати складні процеси і приймати рішення на основі даних.

Аналіз мережевого трафіку є критичним для забезпечення стабільності і безпеки інформаційних систем. Інтелектуальні системи в цьому контексті виконують кілька важливих ролей:

1. Виявлення аномалій та загроз. Інтелектуальні системи використовують алгоритми машинного навчання для виявлення нетипових патернів у мережевому трафіку, що можуть вказувати на аномалії або зловмисні атаки. Завдяки здатності обробляти великі обсяги даних у реальному часі, ці системи можуть швидко виявити аномальні поведінкові патерни, які вказують на потенційні загрози, такі як атаки типу DDoS, шкідливі програми або несанкціонований доступ.

2. Автоматизація моніторингу і реагування. Інтелектуальні системи здатні автоматизувати процеси моніторингу і реагування на інциденти, що значно знижує навантаження на адміністратора мережі. Вони можуть самостійно ініціювати дії, такі як блокування підозрілих IP-адрес або перепланування маршрутизації трафіку, що допомагає швидше реагувати на інциденти без втручання людини.

3. Прогнозування навантаження і оптимізація ресурсів. На основі аналізу історичних даних та тенденцій, інтелектуальні системи можуть прогнозувати майбутнє навантаження на мережу і пропонувати рішення для оптимізації ресурсів. Це дозволяє забезпечити ефективну роботу мережі навіть у умовах змінюваного навантаження або зростання обсягу даних.

Значимість інтелектуальних систем у сучасних мережах постійно зростає через кілька факторів:

1. Збільшення обсягів даних. Сучасні мережі обробляють величезні обсяги даних, що ускладнює їх ефективний аналіз за допомогою традиційних методів. Інтелектуальні системи дозволяють ефективно обробляти і аналізувати ці дані, надаючи можливість виявлення важливих трендів і загроз;

2. Зростання складності атак. Атаки на мережі стають дедалі складнішими і хитрішими. Інтелектуальні системи, завдяки своїм можливостям до адаптації і навчання, можуть швидше виявляти нові типи загроз і реагувати на них більш ефективно, ніж традиційні системи безпеки;

3. Необхідність в реальному часі. У сучасному світі, де час реагування на загрози може бути критично важливим, інтелектуальні системи забезпечують швидкий аналіз і своєчасну реакцію, що допомагає запобігти або зменшити вплив можливих інцидентів.

Інтелектуальні системи для аналізу мережевого трафіку використовують передові технології для забезпечення ефективного моніторингу, виявлення загроз і оптимізації ресурсів у мережах. Основними компонентами таких систем є машинне навчання, штучний інтелект та системи обробки великих даних. Розглянемо ці компоненти детальніше та їх внесок у забезпечення безпеки та ефективності мереж.

Машинне навчання (ML) є ключовим компонентом інтелектуальних систем, що використовують алгоритми для автоматичного вивчення і вдосконалення на основі даних. Основні алгоритми машинного навчання включають:

- Алгоритми класифікації. Використовуються для віднесення даних до певних категорій. Наприклад, алгоритми підтримки векторних машин (SVM) та дерева рішень можуть класифікувати мережеві запити як нормальні або аномальні;

- Алгоритми кластеризації. Групують дані на основі їх подібності. Кластеризація може використовуватися для виявлення нових типів аномалій шляхом групування схожих поведінкових патернів;

- Алгоритми регресії. Вимірюють і прогнозують числові значення. У контексті мережевого трафіку, регресія може використовуватися для прогнозування майбутнього навантаження на мережу на основі історичних даних.

Машинне навчання дозволяє системам автоматично адаптуватися до нових загроз і змінюваних умов у мережі. Завдяки здатності навчатися на історичних даних, ML-алгоритми можуть:

- Виявляти аномалії. Моделі машинного навчання можуть навчитися розпізнавати нормальну поведінку мережі і виявляти відхилення, що можуть вказувати на зловмисну активність або технічні проблеми;

- Покращувати точність аналізу. Завдяки постійному оновленню і вдосконаленню моделей, системи, що використовують машинне навчання, можуть зменшити кількість помилкових тривог і підвищити точність виявлення загроз.

Штучний інтелект (ШІ) охоплює ширший спектр технологій, що включають не лише машинне навчання, але й інші підходи, такі як обробка природної мови та системи розпізнавання образів. У контексті мережевої безпеки, ШІ грає кілька важливих ролей:

- Аналіз поведінки. ШІ-системи можуть здійснювати глибокий аналіз поведінки користувачів і пристроїв, допомагаючи виявити аномалії, які не можуть бути виявлені традиційними методами;

- Автоматизоване реагування. ШІ може автоматично ініціювати захисні заходи у відповідь на виявлені загрози, що значно знижує час реагування та покращує загальний рівень безпеки мережі.

Штучний інтелект активно використовується для виявлення аномалій у мережевому трафіку. Приклади застосування включають:

- Системи виявлення вторгнень (IDS). Використовують ШІ для аналізу трафіку та виявлення підозрілих активностей, таких як аномальні запити або незвичні патерни передачі даних;

- Розпізнавання шкідливого програмного забезпечення. ШІ-системи аналізують поведінку програм і порівнюють її з відомими зразками шкідливого ПО, виявляючи нові або модифіковані загрози.

Системи обробки великих даних грають ключову роль у зборі та аналізі даних у реальному часі. Вони дозволяють ефективно управляти величезними

обсягами інформації, що надходить з мережевих пристроїв та датчиків. Основні етапи збору і обробки даних включають:

- Збір даних. Інструменти для збору даних забезпечують отримання інформації з різних джерел, таких як мережеві зонд, системи моніторингу та реєстратори подій;

- Обробка даних. Використання технологій, таких як Hadoop та Spark, дозволяє обробляти великі обсяги даних з високою швидкістю, забезпечуючи можливість аналізу в реальному часі.

Для ефективної обробки великих даних використовуються різноманітні інструменти та технології, зокрема:

- Hadoop. Платформа для зберігання та обробки великих обсягів даних на розподілених системах;

- Spark. Інструмент для швидкої обробки даних у реальному часі, що забезпечує високу швидкість і ефективність обробки даних;

- Elasticsearch. Система для пошуку і аналізу даних, що дозволяє здійснювати швидкий пошук і візуалізацію великих обсягів даних.

Інтелектуальні системи, що використовують машинне навчання, штучний інтелект і технології обробки великих даних, забезпечують ефективний аналіз мережевого трафіку, що дозволяє підвищити безпеку, оптимізувати ресурси і забезпечити стабільність мережевих систем. Здатність автоматично адаптуватися до нових умов і загроз, а також ефективно обробляти величезні обсяги даних, робить ці системи незамінними для сучасних інформаційних технологій.

В умовах швидкого розвитку інформаційних технологій та зростання обсягів даних, інтелектуальні системи стають незамінними для ефективного управління мережевим трафіком. Завдяки передовим методам машинного навчання, автоматизації процесів моніторингу та прогнозування навантаження, ці системи забезпечують підвищену безпеку та оптимізацію ресурсів у мережах.

Аналіз аномалій на основі машинного навчання є одним з найважливіших методів, що використовуються в інтелектуальних системах. Машинне навчання дозволяє створювати моделі, здатні виявляти нетипові або підозрілі патерни в мережевому трафіку, які можуть свідчити про атаки або технічні проблеми. Основні алгоритми, такі як методи класифікації, кластеризації та регресії, допомагають системам навчатися на історичних даних, виявляти аномалії, що відрізняються від нормального поведінкового профілю мережі. Наприклад, моделі, засновані на нейронних мережах або методах підтримки векторних машин (SVM), можуть розпізнавати складні патерни атак, такі як розподілені атаки відмови в обслуговуванні (DDoS) або несанкціоновані спроби доступу.

Автоматизація моніторингу і реагування є ще однією важливою функцією інтелектуальних систем. Системи автоматизованого моніторингу використовують алгоритми для постійного спостереження за мережевим трафіком, що дозволяє оперативно виявляти загрози і аномалії. Інструменти, такі як системи управління подіями та інформацією безпеки (SIEM), забезпечують автоматизовану обробку подій і сповіщення, що дозволяє зменшити навантаження на адміністратора мережі. У випадку виявлення підозрілих активностей, інтелектуальні системи можуть автоматично ініціювати реакцію, таку як блокування IP-адрес або обмеження доступу до певних ресурсів, що дозволяє зменшити потенційний вплив загроз.

Прогнозування навантаження і оптимізація ресурсів є ще однією ключовою функцією інтелектуальних систем. На основі аналізу історичних даних і трендів, інтелектуальні системи можуть прогнозувати майбутнє навантаження на мережу і забезпечувати оптимізацію ресурсів для підвищення ефективності. Для цього використовуються методи регресії та часового ряду, які дозволяють оцінювати тенденції і підготовлювати мережу до можливих змін в навантаженні. Інтелектуальні системи також можуть автоматично регулювати ресурси, такі як пропускна здатність і обсяг пам'яті,

в залежності від прогнозованого навантаження, що забезпечує стабільну роботу мережі навіть у періоди пікового навантаження.

Таким чином, інтелектуальні системи, що використовують методи машинного навчання, автоматизації моніторингу та прогнозування навантаження, надають потужні інструменти для ефективного управління мережевим трафіком. Вони дозволяють не лише виявляти і реагувати на загрози в реальному часі, але й забезпечують оптимізацію ресурсів і підвищення загальної безпеки і ефективності мережевих систем.

Інтелектуальні системи, що використовуються для аналізу мережевого трафіку, представляють собою важливий інструмент для сучасних інформаційних технологій. Завдяки впровадженню передових технологій, таких як машинне навчання і штучний інтелект, ці системи забезпечують підвищену точність у виявленні загроз, зменшення кількості помилкових тривог і автоматизацію процесів моніторингу. Однак, як і будь-яка технологія, інтелектуальні системи мають свої обмеження, зокрема, залежність від якості даних, високі вимоги до обчислювальних ресурсів і проблеми з інтерпретацією результатів.

Одна з основних переваг інтелектуальних систем полягає в їхній здатності підвищувати точність виявлення загроз. Завдяки алгоритмам машинного навчання, які навчаються на величезних обсягах історичних даних, ці системи можуть ефективно розпізнавати складні патерни атак і аномалій, які традиційні системи безпеки можуть пропустити. Наприклад, алгоритми можуть виявляти нові або модифіковані форми шкідливих програм, що не входять до бази відомих загроз, забезпечуючи таким чином проактивний захист.

Інтелектуальні системи також сприяють зменшенню кількості помилкових тривог. Завдяки навчанню на великих обсягах даних і адаптації до змінних умов мережі, ці системи можуть точніше визначати, які події є справжніми загрозами, а які є звичайними або незначними відхиленнями. Це дозволяє знизити навантаження на адміністратора мережі, зменшуючи

кількість хибних сповіщень і фальшивих тривог, що, в свою чергу, підвищує загальну ефективність системи безпеки.

Автоматизація є ще однією важливою перевагою інтелектуальних систем. Вони можуть автоматично виконувати завдання, такі як моніторинг трафіку, виявлення аномалій і навіть ініціювання заходів реагування. Це не лише знижує необхідність ручного втручання, але й дозволяє системі швидше реагувати на загрози. Автоматизація процесів забезпечує більш швидке виявлення і нейтралізацію атак, що підвищує загальний рівень безпеки мережі.

Одним із значних обмежень інтелектуальних систем є їхня залежність від якості даних. Технології машинного навчання і штучного інтелекту ефективні лише тоді, коли дані, на яких вони навчаються, є точними і репрезентативними. Низька якість даних, включаючи шум або неповноту, може призвести до неточних або ненадійних результатів, що може негативно вплинути на ефективність системи.

Інтелектуальні системи часто мають високі вимоги до обчислювальних ресурсів. Обробка великих обсягів даних і виконання складних алгоритмів машинного навчання потребує значних обчислювальних потужностей і пам'яті. Це може бути проблемою для організацій з обмеженими ресурсами або для тих, хто впроваджує ці системи на великих масштабах. Високі витрати на апаратне забезпечення і електроенергію можуть стати значною перешкодою для впровадження таких систем.

Проблеми з інтерпретацією результатів є ще однією суттєвою перешкодою. Інтелектуальні системи, зокрема ті, що використовують складні алгоритми машинного навчання, можуть генерувати результати, які важко зрозуміти і інтерпретувати для людини. Це може ускладнити розуміння того, чому система виявила певну подію як загрозу або як вона зробила своє рішення. Труднощі в інтерпретації результатів можуть перешкоджати ефективному прийняттю рішень і знижувати загальну впевненість у системі.

Інтелектуальні системи для аналізу мережевого трафіку пропонують значні переваги, включаючи підвищення точності виявлення загроз,

зменшення кількості помилкових тривог і автоматизацію процесів. Проте, вони також стикаються з певними обмеженнями, такими як залежність від якості даних, високі вимоги до обчислювальних ресурсів і проблеми з інтерпретацією результатів. Розуміння цих переваг і обмежень є важливим для ефективного впровадження і використання інтелектуальних систем у сфері мережевої безпеки, що допомагає забезпечити більш надійний і ефективний захист інформаційних систем.

Отже, інтелектуальні системи стали важливим елементом сучасного аналізу мережевого трафіку, надаючи потужні інструменти для забезпечення безпеки та оптимізації мереж. Основними складовими цих систем є технології машинного навчання, штучного інтелекту та обробки великих даних.

Завдяки машинному навчанню, ці системи можуть ефективно виявляти аномалії і загрози, аналізуючи великий обсяг історичних даних і розпізнаючи складні патерни атак, які не завжди помітні традиційними методами. Це підвищує точність виявлення загроз і зменшує кількість помилкових тривог, що, в свою чергу, покращує загальну ефективність мережевої безпеки.

Автоматизація процесів моніторингу і реагування є ще однією суттєвою перевагою. Інтелектуальні системи дозволяють автоматично здійснювати моніторинг трафіку, виявляти підозрілі активності і швидко реагувати на загрози без потреби в ручному втручанні. Це забезпечує оперативність у реагуванні на інциденти та зменшує навантаження на адміністратора мережі.

Однак інтелектуальні системи не позбавлені обмежень. Їх ефективність залежить від якості даних, які використовуються для навчання моделей. Низька якість даних може призвести до неточних результатів і зниження надійності системи. Високі вимоги до обчислювальних ресурсів також можуть бути перешкодою для впровадження таких систем, особливо для організацій з обмеженими ресурсами. Крім того, складність інтерпретації результатів може ускладнити розуміння роботи системи і прийняття обґрунтованих рішень.

В цілому, інтелектуальні системи для аналізу мережевого трафіку пропонують значні переваги в підвищенні безпеки і ефективності мереж,

проте їх впровадження і використання повинні враховувати можливі обмеження і виклики. Систематичний підхід до оцінки і налаштування цих технологій може допомогти максимізувати їхню ефективність і забезпечити надійний захист інформаційних систем.

1.3. Алгоритми машинного навчання та штучного інтелекту для аналізу мережевого трафіку

В епоху цифрових технологій, коли обсяги мережевого трафіку зростають в геометричній прогресії, забезпечення безпеки і ефективності мереж стає дедалі складнішим завданням. Інтелектуальні системи, що базуються на машинному навчанні (ML) і штучному інтелекті (AI), відіграють ключову роль у цьому процесі, надаючи потужні інструменти для аналізу та управління мережевим трафіком. Ці технології дозволяють не лише підвищити точність виявлення загроз, але й автоматизувати багато процесів, що забезпечують належний рівень безпеки і ефективності мереж.

Машинне навчання і штучний інтелект радикально змінили підходи до аналізу мережевого трафіку, забезпечуючи можливість ефективного оброблення великих обсягів даних і виявлення складних патернів, які можуть бути невидимі для традиційних методів. Машинне навчання дозволяє системам автоматично вдосконалюватися на основі даних без потреби у явному програмуванні. Це особливо корисно для виявлення нових або модифікованих форм атак, які не входять до бази відомих загроз.

Штучний інтелект, у свою чергу, включає в себе широкий спектр технологій, таких як глибоке навчання, яке використовує багатошарові нейронні мережі для обробки і аналізу складних даних. Це дозволяє розпізнавати більш тонкі і складні патерни, що є критично важливим для захисту від сучасних загроз, які постійно еволюціонують.

Існує кілька ключових типів алгоритмів машинного навчання та штучного інтелекту, які використовуються в аналізі мережевого трафіку:

1. Класифікаційні алгоритми – ці алгоритми призначені для класифікації мережевого трафіку за різними категоріями. Наприклад, алгоритми логістичної регресії, дерева рішень та методи підтримки векторних машин (SVM) використовуються для виявлення аномалій і загроз шляхом порівняння даних з відомими шаблонами. Нейронні мережі, включаючи глибокі нейронні мережі, також використовуються для класифікації, завдяки їхній здатності навчатися на складних даних і розпізнавати нетипові патерни;

2. Кластеризаційні алгоритми – ці алгоритми групують дані в кластери на основі їхньої подібності. Метод K-середніх, ієрархічна кластеризація та DBSCAN є популярними інструментами для виявлення аномалій і непередбачуваних шаблонів у мережевому трафіку. Кластеризація дозволяє виявити підозрілі групи трафіку, які можуть бути ознаками атаки або несанкціонованого доступу;

3. Регресійні алгоритми – використовуються для прогнозування майбутніх значень на основі історичних даних. Лінійна та поліноміальна регресія допомагають передбачити навантаження на мережу і її ресурси, що дозволяє оптимізувати використання ресурсів і запобігати перевантаженням;

4. Алгоритми навчання без нагляду – ці алгоритми, такі як метод головних компонент (PCA), використовуються для зменшення розмірності даних і виявлення структурних патернів без попередньо відомих міток. Вони корисні для виявлення нових аномалій або трендів у мережевому трафіку, які можуть бути неочевидними при використанні традиційних методів;

5. Глибоке навчання – це підхід до машинного навчання, який використовує багатошарові нейронні мережі для обробки і аналізу складних даних. Конволюційні нейронні мережі (CNN) та рекурентні нейронні мережі (RNN) є прикладами технік глибокого навчання, які дозволяють розпізнавати складні патерни і взаємозв'язки у мережевому трафіку. Ці алгоритми здатні досягати високих рівнів точності у виявленні загроз і аномалій.

Машинне навчання і штучний інтелект радикально змінили підходи до аналізу мережевого трафіку, надаючи інструменти для ефективного

управління великими обсягами даних і виявлення складних загроз. Завдяки класифікаційним, кластеризаційним, регресійним алгоритмам та методам глибокого навчання, ці системи можуть забезпечувати точний аналіз, автоматизацію процесів і прогностичні можливості, що є критично важливими для сучасних мереж. Однак успішне впровадження цих технологій вимагає ретельного підходу до якості даних, обчислювальних ресурсів і інтерпретації результатів.

Машинне навчання є однією з найбільш динамічно розвиваючих галузей у сфері комп'ютерних наук, і класифікаційні алгоритми відіграють ключову роль у цій дисципліні. Класифікаційні алгоритми дозволяють автоматично відносити дані до певних категорій або класів на основі їхніх характеристик. Вони знаходять широке застосування в багатьох областях, включаючи фінансову аналітику, медичну діагностику, обробку природної мови і, звісно, аналіз мережевого трафіку.

Логістична регресія є одним з найпростіших і найбільш використовуваних класифікаційних алгоритмів. Її основна мета — передбачити ймовірність належності об'єкта до певного класу на основі незалежних змінних. Цей алгоритм використовує логістичну функцію (сигмоїдну функцію) для перетворення вхідних даних у ймовірність, яка потім класифікується на основі заданого порогового значення. Логістична регресія є ефективною для двокласових задач, але її можна розширити для багатокласових проблем за допомогою методів, таких як One-vs-Rest.

У контексті аналізу мережевого трафіку, логістична регресія може бути використана для класифікації трафіку як нормального або аномального на основі різних характеристик, таких як обсяг даних, час передачі, тип пакета тощо.

Дерева рішень є ще одним популярним класифікаційним алгоритмом, який використовує дерево у вигляді графа для прийняття рішень. Кожен вузол дерева представляє тест на певну ознаку, а кожне гілля — результат тесту. Листові вузли дерева представляють фінальні рішення або класи. Алгоритм

побудови дерева рішень (наприклад, C4.5 або CART) включає розбиття даних на підмножини таким чином, щоб максимізувати інформацію, яка отримується з кожного тесту.

Цей підхід є зрозумілим і інтуїтивно зрозумілим, що робить його корисним для інтерпретації результатів. У сфері аналізу мережевого трафіку, дерева рішень можуть допомогти в ідентифікації та класифікації типів трафіку, а також у виявленні потенційних загроз на основі різних ознак.

Метод підтримки векторних машин (SVM) є потужним класифікаційним алгоритмом, який створює гіперплощину в багатовимірному просторі, що максимізує розмежування між різними класами даних. SVM намагається знайти таку гіперплощину, яка забезпечить найкраще можливе розділення між класами, що є особливо корисним для задач з високою розмірністю.

Цей метод є ефективним для складних класифікаційних завдань, де інші алгоритми можуть виявитися недостатньо точними. У аналізі мережевого трафіку, SVM може бути використаний для виявлення нетипового або шкідливого трафіку, що відрізняється від нормального патерну, завдяки його здатності працювати з великими обсягами даних і високими розмірами ознак.

Нейронні мережі, особливо глибокі нейронні мережі (deep neural networks), є одними з найбільш потужних інструментів для класифікації даних. Вони складаються з багатьох шарів вузлів (нейронів), які імітують роботу людського мозку. Кожен шар обробляє дані та передає результати наступному шару, що дозволяє нейронним мережам навчатися складним патернам і взаємозв'язкам у даних.

Глибокі нейронні мережі є особливо корисними в завданнях, де дані мають складну структуру або велику кількість ознак. В контексті аналізу мережевого трафіку, нейронні мережі можуть використовуватися для виявлення складних аномалій і атак, таких як DDoS-атаки або нові форми шкідливого програмного забезпечення, завдяки їхній здатності моделювати складні взаємозв'язки у даних.

Класифікаційні алгоритми машинного навчання, такі як логістична регресія, дерева рішень, метод підтримки векторних машин (SVM) і нейронні мережі, є потужними інструментами для аналізу мережевого трафіку. Кожен з цих алгоритмів має свої сильні сторони і підходить для різних типів класифікаційних завдань. Вибір алгоритму залежить від специфіки задачі, обсягу і якості даних, а також вимог до точності і швидкості обробки. Інтеграція цих методів у систему аналізу мережевого трафіку дозволяє ефективно виявляти загрози, оптимізувати використання ресурсів і забезпечувати надійну безпеку мережі.

Машинне навчання постійно розвивається, і кластеризаційні, регресійні алгоритми, а також алгоритми навчання без нагляду займають важливе місце в аналітиці даних. Ці алгоритми дозволяють розпізнавати патерни, прогнозувати значення і виявляти структури в даних, що є критично важливим для багатьох сфер, включаючи аналіз мережевого трафіку.

Алгоритм K-середніх є одним з найбільш відомих і використовуваних методів кластеризації. "K" у назві вказує на кількість кластерів, на які потрібно розділити дані. Алгоритм працює наступним чином: спочатку вибираються випадкові центри кластерів (центроїди), після чого кожен об'єкт даних призначається найближчому центроїду. Після цього центри кластерів пересуваються до середніх значень всіх точок, що належать до відповідного кластеру. Процес повторюється до тих пір, поки центри не стабілізуються або зміни між ітераціями стануть незначними.

K-середніх добре підходить для задач, де дані можуть бути чітко розділені на кілька кластерів, проте його ефективність може знижуватися, коли класи мають різні розміри або форми. У аналізі мережевого трафіку, цей метод може допомогти в групуванні схожих типів трафіку або виявленні патернів, що повторюються.

Ієрархічна кластеризація створює ієрархічну структуру кластерів у вигляді дерева або дендрограми. Існують два основні підходи до ієрархічної кластеризації: агломеративний (знизу вгору) і дивізивний (згори вниз).

Агломеративний підхід починається з того, що кожен об'єкт є окремим кластером, і поступово об'єднує найбільш схожі кластери, поки не буде досягнуто бажаної кількості кластерів. Дивізивний підхід починається з одного великого кластеру, який поступово розбивається на менші кластери.

Ієрархічна кластеризація корисна, коли потрібно зрозуміти, як об'єкти зв'язані між собою на різних рівнях деталізації. Вона дозволяє виявити структуру даних без попередньо визначеної кількості кластерів. У сфері мережевого трафіку це може допомогти у виявленні ієрархії різних типів трафіку або структури атак.

DBSCAN є алгоритмом кластеризації, заснованим на щільності. Він класифікує точки даних в кластери на основі щільності навколо кожної точки, а також визначає точки як шум або викиди, якщо їхня щільність нижча за поріг. Основні параметри алгоритму — радіус області (ε) і мінімальна кількість точок для утворення кластера (MinPts).

DBSCAN особливо корисний для виявлення кластерів різних форм і розмірів і є стійким до викидів, що робить його придатним для задач, де дані мають складну структуру або включають шум.

Лінійна регресія є основним методом для моделювання відношень між незалежними змінними та залежною змінною, використовуючи лінійну функцію. Вона намагається знайти лінійну залежність, яка найкраще підходить для даних, мінімізуючи суму квадратів відхилень між фактичними і передбаченими значеннями.

Лінійна регресія застосовується в багатьох сферах для прогнозування числових значень, таких як обсяг мережевого трафіку на основі історичних даних. Це дозволяє планувати ресурси і виявляти потенційні проблеми.

Поліноміальна регресія розширює концепцію лінійної регресії, використовуючи поліноми в якості функцій для моделювання більш складних залежностей. Це дозволяє вловлювати нелінійні зв'язки між змінними, що робить її корисною для задач, де лінійна регресія не забезпечує точних прогнозів.

У мережевому аналізі поліноміальна регресія може бути використана для моделювання складних патернів трафіку і прогнозування навантаження на мережу, враховуючи нелінійні залежності.

Алгоритми асоціації використовуються для виявлення частих шаблонів або асоціацій між ознаками в даних. Один з найбільш відомих алгоритмів асоціації – алгоритм Apriori, який знаходить часті елементи та асоціації між ними, виходячи з заданих порогів підтримки та довіри.

В контексті мережевого трафіку ці алгоритми можуть використовуватися для виявлення частих шаблонів трафіку або спільних характеристик між різними типами атак.

Метод головних компонентів (PCA) є технікою для зменшення розмірності даних, яка дозволяє перетворити великий набір змінних у менший набір компонентів, що зберігають найбільшу частину варіації даних. PCA працює шляхом знаходження основних компонентів, які представляють основні напрямки варіації в даних.

У мережевому аналізі PCA може допомогти у виявленні основних патернів у трафіку і спрощенні даних для подальшого аналізу або моделювання, що може підвищити ефективність алгоритмів класифікації та прогнозування.

Кластеризаційні, регресійні алгоритми та алгоритми навчання без нагляду є потужними інструментами для аналізу даних, що дозволяють вирішувати різноманітні задачі, від виявлення патернів до прогнозування значень. Кожен з цих методів має свої особливості та підходить для різних типів даних і завдань. Інтеграція цих алгоритмів у системи аналізу мережевого трафіку дозволяє створювати більш точні, ефективні та адаптивні рішення для управління і захисту інформаційних систем.

Штучний інтелект (ШІ) значно змінив підходи до аналізу даних і прийняття рішень, зокрема завдяки розвитку глибоких нейронних мереж та їх численних варіацій. Ці алгоритми дозволяють вирішувати складні задачі, що виходять за межі можливостей традиційних методів. Глибокі нейронні мережі,

конволюційні нейронні мережі, рекурентні нейронні мережі, а також методи підкріпленого навчання стали основними інструментами в розробці сучасних інтелектуальних систем.

Глибокі нейронні мережі (Deep Learning) є підходом до машинного навчання, який використовує багатошарові нейронні мережі для виявлення складних патернів у великих обсягах даних. Глибокі нейронні мережі мають велику кількість шарів (глибин), що дозволяє їм ефективно вивчати абстрактні ознаки з даних. Ці мережі виявилися надзвичайно потужними у задачах класифікації, регресії, розпізнавання образів та багатьох інших.

У сфері аналізу мережевого трафіку, глибокі нейронні мережі використовуються для виявлення аномалій і загроз, таких як зловмисні атаки, за рахунок здатності моделювати складні залежності та патерни в даних. Вони можуть навчатися на великій кількості історичних даних, щоб розпізнати нові, ще не зафіксовані загрози.

Конволюційні нейронні мережі (CNN) є спеціалізованим типом глибоких нейронних мереж, які були розроблені для обробки даних з сітковою структурою, таких як зображення. CNN використовують конволюційні шари, які дозволяють мережам автоматично виявляти просторові патерни і характеристики у зображеннях.

В аналізі мережевого трафіку CNN можуть бути адаптовані для обробки та класифікації даних, представлених у вигляді матриць або тензорів, що дозволяє виявляти аномалії і патерни в даних, представлених у структурованій формі.

Рекурентні нейронні мережі (RNN) є ще одним важливим класом нейронних мереж, призначених для обробки послідовностей даних. Вони здатні зберігати інформацію про попередні стани завдяки своїй рекурентній структурі, що дозволяє їм бути ефективними в задачах, де важлива історія подій.

Однією з популярних варіацій RNN є LSTM (Long Short-Term Memory), яка вирішує проблему "забування" у стандартних RNN шляхом введення

механізму, що дозволяє зберігати важливу інформацію протягом довгих періодів. LSTM добре підходять для обробки часових рядів, таких як мережевий трафік, де послідовність подій має значення для розуміння контексту.

Нейронні мережі для обробки природної мови (NLP) спеціалізуються на роботі з текстовими даними. Вони використовуються для виконання завдань, таких як розпізнавання тексту, переклад, аналіз настрою і багато іншого. Моделі NLP базуються на різних архітектурах, таких як рекурентні нейронні мережі і трансформери, які дозволяють обробляти текстову інформацію з урахуванням контексту і семантики.

Трансформери є архітектурою, яка революціонізувала обробку природної мови завдяки здатності обробляти текст з урахуванням контексту на всіх рівнях. BERT (Bidirectional Encoder Representations from Transformers) є прикладом трансформера, який використовує двонаправлене кодування для кращого розуміння контексту слів у реченнях.

BERT і його варіації демонструють високу ефективність у завданнях, що включають обробку і аналіз тексту, і можуть бути адаптовані для аналізу логів і повідомлень у мережах для виявлення аномалій або підозрілих активностей.

Методи підкріпленого навчання (Reinforcement Learning) є класом алгоритмів, де агент навчається через взаємодію з середовищем, отримуючи винагороди або покарання в залежності від своїх дій. Основною метою є навчитися приймати оптимальні рішення для максимізації загальної винагороди.

У контексті безпеки мереж, підкріплене навчання може використовуватися для адаптивного управління системами безпеки, де агент може оптимізувати свої стратегії реагування на загрози і аномалії, вчитися з попередніх інцидентів і підвищувати ефективність захисту мережі.

Алгоритми штучного інтелекту, такі як глибокі нейронні мережі, конволюційні і рекурентні нейронні мережі, моделі обробки природної мови, а також методи підкріпленого навчання, забезпечують потужні інструменти

для аналізу складних даних і вирішення задач, що стоять перед сучасними інформаційними системами. Вони дозволяють створювати ефективні системи для виявлення аномалій, прогнозування загроз і оптимізації ресурсів. Розуміння і використання цих алгоритмів є ключовим для розвитку інтелектуальних систем у різних галузях, включаючи кібербезпеку та управління мережами.

Аналіз мережевого трафіку є критично важливою частиною забезпечення безпеки та оптимізації мереж. Сучасні алгоритми машинного навчання і штучного інтелекту (ШІ) забезпечують потужні інструменти для вирішення різних задач, включаючи виявлення аномалій і загроз, прогнозування навантажень і ресурсів, а також автоматизацію процесів моніторингу та реагування. Розглянемо, як різні типи алгоритмів допомагають у цих сферах.

Виявлення аномалій і загроз є одним з основних завдань у забезпеченні мережевої безпеки. Класифікаційні алгоритми, такі як логістична регресія, дерева рішень і метод підтримки векторних машин (SVM), дозволяють розпізнавати відомі шаблони атак і аномалій в мережевому трафіку. Класифікаційні алгоритми навчаються на історичних даних, щоб відрізняти нормальний трафік від аномального, і можуть швидко виявляти потенційні загрози в реальному часі.

Кластеризаційні алгоритми, такі як K-середніх та DBSCAN, допомагають виявити нові, ще не відомі аномалії, кластеризуючи дані за схожістю. Наприклад, якщо новий тип атаки має специфічний патерн трафіку, кластеризаційні алгоритми можуть виявити ці нові патерни, навіть якщо вони не входять до навчальних даних.

Глибокі нейронні мережі грають особливу роль у виявленні складних і невідомих атак завдяки своїй здатності автоматично вивчати складні патерни в даних. Глибокі мережі можуть виявляти більш тонкі аномалії, які можуть бути невидимі для традиційних методів, завдяки своїй здатності виявляти нелінійні зв'язки в даних.

Прогнозування навантажень і ресурсів є важливим для планування та управління мережами. Регресійні алгоритми, такі як лінійна і поліноміальна регресія, використовуються для прогнозування навантажень на мережу, базуючи свої прогнози на історичних даних про трафік. Лінійна регресія підходить для простих завдань, де є лінійний зв'язок між параметрами, в той час як поліноміальна регресія може впоратися з більш складними і нелінійними трендами.

Глибоке навчання пропонує потужні методи для прогнозування майбутніх трендів. Глибокі нейронні мережі можуть аналізувати великий обсяг історичних даних, щоб знайти складні патерни і тренди, що дозволяє точніше прогнозувати навантаження і потреби в ресурсах. Це допомагає в управлінні навантаженням, оптимізації ресурсів і забезпеченні стабільної роботи мережі.

Автоматизація процесів моніторингу і реагування є критично важливою для швидкого реагування на загрози. Алгоритми штучного інтелекту можуть автоматизувати ці процеси за рахунок їх здатності виявляти загрози та аномалії у реальному часі і здійснювати миттєві дії для їх нейтралізації.

Методи підкріпленого навчання дозволяють створювати адаптивні системи безпеки, які можуть навчатися з попередніх інцидентів і постійно вдосконалювати свої стратегії реагування. Ці системи можуть автоматично регулювати параметри безпеки, такі як рівень фільтрації трафіку або політики доступу, щоб ефективно реагувати на нові загрози.

Інші алгоритми ШІ можуть автоматично здійснювати моніторинг трафіку, визначати аномалії та генерувати попередження без необхідності постійного втручання людини. Це не тільки знижує навантаження на фахівців з безпеки, але й забезпечує швидше реагування на потенційні загрози.

Алгоритми машинного навчання і штучного інтелекту забезпечують важливі можливості для аналізу мережевого трафіку, допомагаючи вирішувати задачі виявлення аномалій і загроз, прогнозування навантажень і ресурсів, а також автоматизації процесів моніторингу і реагування. Вони

дозволяють створювати більш ефективні, адаптивні і швидкі рішення для управління і захисту мереж, що є критично важливим в умовах постійно зростаючих кіберзагроз і складності інформаційних систем.

Отже, аналіз мережевого трафіку за допомогою алгоритмів машинного навчання і штучного інтелекту надає потужні інструменти для підвищення безпеки і ефективності мереж. Застосування цих алгоритмів дозволяє зокрема виявляти аномалії і загрози, прогнозувати навантаження і ресурси, а також автоматизувати процеси моніторингу і реагування.

Класифікаційні алгоритми, такі як логістична регресія, дерева рішень і метод підтримки векторних машин (SVM), забезпечують ефективну ідентифікацію відомих загроз і аномалій, базуючись на історичних даних. Кластеризаційні алгоритми, включаючи K-середніх і DBSCAN, допомагають виявляти нові або невідомі патерни аномального трафіку.

Глибокі нейронні мережі, завдяки своїй здатності автоматично виявляти складні патерни, мають значний вплив на виявлення складних і нових атак. Конволюційні нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), включаючи їх варіації, такі як LSTM, спеціалізуються на обробці структурованих даних та послідовностей, що робить їх надзвичайно корисними для аналізу трафіку і прогнозування майбутніх трендів.

Методи підкріпленого навчання дозволяють автоматизувати адаптивне управління безпекою, оптимізуючи стратегії реагування на загрози на основі попереднього досвіду.

Загалом, алгоритми машинного навчання і штучного інтелекту забезпечують значні переваги в аналізі мережевого трафіку, включаючи підвищення точності виявлення загроз, прогнозування навантажень і ресурсів, а також автоматизацію процесів моніторингу і реагування. Ці інструменти дозволяють забезпечити високий рівень захисту мереж і оптимізації ресурсів, що є критично важливим у сучасному цифровому середовищі.

1.4. Висновки до розділу 1

У розділі 1 було розглянуто теоретичні основи аналізу мережевого трафіку за допомогою інтелектуальних систем, включаючи основи поняття, методи, задачі та роль інтелектуальних систем у цьому процесі.

Мережевий трафік є ключовим аспектом сучасних інформаційних технологій, що забезпечує безперебійний обмін даними між різними пристроями та системами. Для ефективного управління, моніторингу та оптимізації мережі важливо розуміти основні характеристики мережевого трафіку, такі як обсяг, швидкість та протоколи. Ці характеристики відіграють критичну роль у забезпеченні стабільності та ефективності мережі.

Інтелектуальні системи відіграють важливу роль в аналізі мережевого трафіку, надаючи інструменти для виявлення загроз, автоматизації процесів моніторингу і оптимізації ресурсів. Їх значимість у сучасних мережах зростає завдяки здатності ефективно обробляти великі обсяги даних, адаптуватися до нових загроз і забезпечувати швидкий і точний аналіз у реальному часі. У світлі зростання обсягів даних і складності кіберзагроз, інтелектуальні системи стають незамінним елементом сучасних інформаційних технологій, забезпечуючи стабільність, безпеку і ефективність мережевих інфраструктур.

Алгоритми машинного навчання і штучного інтелекту є основою сучасних інтелектуальних систем для аналізу мережевого трафіку. Їх здатність обробляти великі обсяги даних, виявляти складні патерни та адаптуватися до нових загроз забезпечує значні переваги в порівнянні з традиційними методами. Глибокі нейронні мережі, конволюційні та рекурентні нейронні мережі, а також методи підкріпленого навчання забезпечують потужні інструменти для моніторингу, прогнозування і реагування на інциденти.

Отже, теоретичні основи аналізу мережевого трафіку, інтелектуальні системи та алгоритми машинного навчання і штучного інтелекту утворюють комплексний підхід до забезпечення безпеки та ефективності мереж. Вони дозволяють значно підвищити точність і швидкість реагування на загрози, а

також оптимізувати використання ресурсів. Ці інструменти є невід'ємною частиною сучасних систем управління мережами і їх безпекою.

РОЗДІЛ 2
ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ

2.1.Аналіз та засоби моніторингу мережевого трафіку

Основним засобом для відстеження повідомлень, які передаються між хостами, є пакетний сніфер. Цей інструмент "перехоплює" трафік, що проходить через мережу, дозволяючи захоплювати повідомлення, що надсилаються або отримуються комп'ютером. Пакетні сніфери зазвичай зберігають і візуалізують вміст полів протоколів у захоплених повідомленнях. Оскільки це пасивна програма, сніфер не надсилає пакети самостійно, а лише аналізує трафік, який передається через комп'ютер. Отримані пакети не призначаються безпосередньо сніферу, але він отримує їх копії для подальшого аналізу.

Пакетний сніфер є інструментом, як програмним, так і апаратним, призначеним для захоплення, реєстрації та аналізу мережевого трафіку. Ці інструменти допомагають ідентифікувати, класифікувати та усувати неполадки в мережі, беручи до уваги тип програми, джерело та призначення даних. Сучасні сніфери пакетів часто використовують API, такі як pcap для Unix-подібних систем або libcap для Windows, для збору мережевого трафіку. Після захоплення даних, ці інструменти дозволяють детально аналізувати інформацію, що допомагає точно виявити і виправити проблеми у мережі, запобігаючи їх повторенню.

Щоб повною мірою оцінити роль сніферів, важливо зрозуміти, як відбувається маршрутизація даних в Інтернеті. Коли ви надсилаєте електронний лист, відкриваєте веб-сторінку або передаєте файл, інформація розбивається на тисячі маленьких частин, званих пакетами даних. Ці пакети подорожують через стек протоколів TCP/IP, який складається з чотирьох рівнів:

1. Рівень прикладних протоколів;
2. Рівень протоколу управління передачею (TCP);
3. Рівень інтернет-протоколу (IP);
4. Апаратний рівень.

На першому етапі пакет переходить через рівень програми, де йому присвоюється номер порту. Потім дані передаються на рівень IP, де пакету присвоюється цільова IP-адреса. Після цього пакет готовий до передачі через Інтернет, а апаратний рівень перетворює його в мережеві сигнали. При досягненні пункту призначення, дані для маршрутизації, такі як номер порту та IP-адреса, видаляються, і пакет продовжує свій шлях через стек протоколів нової мережі, щоб у кінцевому результаті зібратися у початкову форму.

Пакетні сніфери працюють шляхом перехоплення даних про мережевий трафік під час його проходження через дротові або бездротові мережі та збереження цих даних у файлах, що називається захопленням пакетів. На відміну від звичайних комп'ютерів, які зазвичай ігнорують трафік від інших пристроїв, сніфери спеціально розроблені для виявлення і запису такого трафіку. Для цього мережева карта інтерфейсу (NIC) — пристрій, що з'єднує комп'ютер з мережею, — повинна бути переведена в режим перехоплення. Це дозволяє комп'ютеру фіксувати і обробляти всі пакети, що проходять через мережу.

Якість захоплення пакетів залежить від типу мережі. У дротових мережах можливості сніфера визначаються конфігурацією мережевих комутаторів, які відповідають за з'єднання декількох пристроїв. Це може

означати, що сніфер здатний бачити трафік на всій мережі або тільки на певній її частині. В бездротових мережах ситуація може бути складнішою, оскільки сніфери зазвичай можуть захоплювати дані тільки з одного каналу одночасно, якщо комп'ютер не обладнаний кількома бездротовими інтерфейсами.

На ринку існує безліч інструментів для збору мережевого трафіку, як платних, так і безкоштовних. Незважаючи на те, що всі ці інструменти базуються на загальних принципах перехоплення даних, вони значно відрізняються за своїми функціональними можливостями і рівнем деталізації. Багато інструментів з відкритим кодом пропонують просту конструкцію, що дозволяє ефективно збирати дані з мінімальним впливом на систему. Такі безкоштовні інструменти можуть бути корисні для швидкої діагностики і основного моніторингу.

Однак з широким вибором продуктів може бути важко вибрати відповідний сніфер. Безкоштовні версії можуть бути достатніми для базових потреб, але платні рішення часто пропонують розширені аналітичні функції і глибший аналіз. Платні інструменти зазвичай забезпечують глибокий аналіз пакетів (DPI) і можуть створювати інтуїтивно зрозумілі графіки та діаграми, що демонструють детальні показники мережевої активності. Хоча ці інструменти можуть бути дорогими, їх функціональність часто виправдовує витрати, надаючи більш точну і детальну інформацію для управління і аналізу мережевого трафіку.

TCPDUMP є популярним інструментом для аналізу мережевих пакетів, що використовує інтерфейс командного рядка (CLI) і є сумісним з платформами Unix та Linux. Розроблений у 1987 році в Національній лабораторії Лоуренса Берклі, він був опублікований декілька років потому.

Основною складовою частиною TCPDUMP є бібліотека libpcap, написана на мові програмування C, яка забезпечує збору інформації про мережу. Libpcap пропонує інтерфейс для численних Unix-подібних систем, таких як FreeBSD і Linux. Для платформи Windows аналогічну функціональність забезпечує WinDump, який використовує WinPcap — порт

бібліотеки libpcap для Windows. Бібліотека libpcap була розроблена як універсальний API для роботи з різними програмами та усунення залежності від системи для модулів збору даних. TCPDUMP є основним інструментом для перехоплення та аналізу мережевих пакетів.

Процес роботи TCPDUMP включає такі етапи:

1. Читання або запис захоплених даних у форматі PCAP з мережі через CLI команди;

2. Фільтрація пакетів за заданими критеріями;

3. Виведення на екран захоплених даних згідно з заданими параметрами.

Завдяки використанню CLI, TCPDUMP є компактним і портативним інструментом для аналізу мережевого трафіку, що дозволяє адміністраторам мереж здійснювати віддалений доступ до мережевих пристроїв.

Основне обмеження TCPDUMP полягає в тому, що він не надає графічного інтерфейсу користувача (GUI) для аналізу захоплених даних, а лише командний рядок (CLI). Це означає, що всі дані відображаються у текстовому форматі, що дозволяє зручно користуватися ним через віддалене з'єднання, наприклад, через Telnet.

Серед інших недоліків TCPDUMP варто відзначити:

- Обмеження в аналізі трафіку. TCPDUMP може аналізувати тільки трафік, заснований на протоколах TCP;

- Обмежена інформація. Якщо IP-адреса у пакеті підроблена, TCPDUMP не здатний виявити чи повідомити про це;

- Недоступність заблокованих пакетів. Пакети, що блокуються брандмауером, не будуть відображені в захоплених даних.

Wireshark, розроблений Джеральдом Комбсом наприкінці 1997 року, є популярним інструментом для аналізу мережевого трафіку і моніторингу даних. Спочатку цей інструмент називався Ethereal, але з травня 2006 року його назва була змінена на Wireshark. Це програмне забезпечення з відкритим кодом, яке є безкоштовним та має графічний інтерфейс користувача. Воно написано мовою C і розповсюджується під ліцензією GNU General Public

License (GPL). Wireshark підтримує різні платформи, включаючи Unix-подібні системи, такі як Mac OS X, Linux, Solaris, а також Microsoft Windows. Для командного рядка є версія Wireshark під назвою TShark, яка дозволяє користувачам працювати з програмою через командний рядок, подібно до TCPDump, але з додатковими можливостями графічного інтерфейсу.

Wireshark використовується для захоплення пакетів даних у мережі та перегляду вже збережених файлів з даними. Підтримується формат файлів «PCAP» для захоплення пакетів. Інструмент дозволяє переглядати дані в байтовому та шістнадцятковому форматах, а також аналізувати різні типи пакетів і протоколів.

Wireshark має трисекційний інтерфейс:

- Панель списку пакетів показує різні пакети, що містять інформацію про номер кадру, дату, час, IP-адреси призначення та джерела, протоколи верхнього рівня, довжину пакету та інші дані, що відображаються з кольоровим маркуванням;

- Панель деталей пакета надає детальні відомості про вибраний пакет у вигляді деревоподібної структури, що відображає протоколи на різних рівнях, такі як TCP, UDP, ICMP, HTTP та інші;

- Панель даних або байтова панель відображає необроблені дані пакету в шістнадцятковому та ASCII форматах, показуючи текстові дані та кодування.

Цей інтерфейс дозволяє користувачам здійснювати детальний аналіз мережевого трафіку та зручний перегляд даних.

Colasoft — це аналізатор мережевого трафіку з закритим вихідним кодом, спеціально розроблений для платформи Windows. Цей інструмент використовується адміністраторами мереж для усунення несправностей, моніторингу та діагностики мережевого трафіку. Colasoft надає безкоштовні інструменти, включаючи Capsa, які пропонують простоту використання, реальний час аналізу пакетів і потужний криміналістичний аналіз протоколів, а також забезпечують цілодобовий моніторинг мережі.

Capsa дозволяє відкривати кілька інтерфейсів одночасно в одному сеансі та пропонує графічні інтерфейси і матричні представлення даних. Інструмент здійснює глибокий аналіз пакетів, надаючи можливість створювати звіти, журнали та отримувати сповіщення у вигляді голосових повідомлень та електронних листів у версіях з ліцензією. Графічний інтерфейс Colasoft дозволяє відображати захоплену інформацію у формі графіків і матриць, детально демонструючи характеристики кожного протоколу, що використовується в мережі.

Для порівняння трьох методів мережевого аналізу, таких як Wireshark, TCPDump і Colasoft, розглянемо кілька ключових параметрів, включаючи наявність вихідного коду, підтримувані протоколи, операційні системи, підтримка формату PCAP, інтерфейс користувача, вартість, декодування, виявлення аномальних пакетів і можливість відновлення TCP потоку.

Таблиця 2.1 ілюструє порівняння між цими інструментами. Жоден з них не ідеально відповідає всім критеріям, але це порівняння допомагає виявити переваги та недоліки кожного інструмента, що може сприяти вдосконаленню їх функцій. Розглянемо якісні та кількісні параметри інструментів:

Colasoft виділяється завдяки своєму візуально орієнтованому підходу до аналізу, надаючи детальну статистику захоплених пакетів та інформацію про протоколи з графіками та матричними представленнями. Це забезпечує зручність в діагностиці мережевих проблем завдяки функціям, таким як звіти, журнали і поглиблений аналіз. Colasoft також ефективно реконструює TCP потоки і пропонує розширене декодування трафіку, що допомагає в комплексному моніторингу мережі. Інструмент має дружній графічний інтерфейс, зручний для користувачів, який підтримує одночасну роботу з кількома інтерфейсами. Графічне відображення даних дозволяє легко аналізувати різні аспекти мережі, що робить Colasoft зручним і потужним інструментом для мережевого моніторингу.

Wireshark, з іншого боку, пропонує велику функціональність з графічним інтерфейсом, підтримуючи численні протоколи і дозволяючи

детальний аналіз пакетів. Проте, він обмежений у функціях графічного інтерфейсу і не підтримує одночасну роботу з кількома інтерфейсами.

TCPDump є простішим інструментом, орієнтованим на командний рядок, що робить його менш інтуїтивно зрозумілим для користувачів, але потужним для базового аналізу. Хоча він є легким і портативним, його функціональність обмежена в порівнянні з більш складними інструментами, такими як Wireshark і Colasoft.

Таблиця 2.1 демонструє різницю в характеристиках між Wireshark, TCPDump і Colasoft, і допомагає вибрати найбільш відповідний інструмент для конкретних завдань у мережевому аналізі.

Таблиця 2.1

Порівняльні характеристики між Wireshark, TCPDump та Colasoft Packet Sniffing Tools

Параметри	Сніферні інструменти		
	TCPDump	WireShark	Colasoft
Відкритий код	+	+	-
Якими операційними системами підтримується	Linux(WINDum для		
Число підтримки протоколів	Windows)	Linux, Windows	Windows
Користувацький інтерфейс	TCP/IP	БІльш ніж 300	300
Вартість	CLI	CLI і GUI	GUI
Libpcap основа	Безкоштовно	Безкоштовно	999 $
Визначення прихований даних	+	+	-
Використання місця на диску	-	+	+
Відображення в додатку шару протоколу	484КБ	449МБ для Unix 89МБ для Windows	32МБ
	Сніферні інструменти		
	TCPDump	WireShark	Colasoft
Декодування протоколу	Лише Hex, ASCII	Лише Hex, ASCII	EBDIC, Hex, ASCII

Завершення табл. 2.1

Відновлення TCP			
потоку	-	+(але лише форматоване)	+
Виявлення ненормальних даних	-	-(лише створює попередження)	+
Кілька інтерфейсів	-	-	+
Сповіщення знаходження	-	-	+
Відновлення HTTP			
веб сторінки	-	-(показує актуальний контент трафіку індивідуально)	-(показує лінки контенту трафіка індивідуально)
Мережева комунікаційна матрична мапа	-	-	+
Оцінка критичного та некритичного для бізнесу трафіку	-	+(за допомогою створення нових фільтрів та пошуку)	+(вбудована)
UDP трафік	-	+	+

У порівнянні з Wireshark, Colasoft надає більший рівень мережевої безпеки завдяки системі сповіщень про події, які можуть надходити у вигляді аудіо сигналів або електронних листів. Однак Colasoft має певні обмеження: він підтримує лише 300 протоколів, що значно менше порівняно з Wireshark, який працює з понад 1100 протоколами.

TCPDump є легким і економічним інструментом для захоплення пакетів, займаючи всього 484 КБ пам'яті для установки. Для порівняння, Wireshark

потребує 18 МБ для початкової інсталяції і займає 81 МБ на диску в Windows та 449 МБ в Linux після завершення інсталяції. Colasoft займає 32 МБ. Таким чином, Wireshark є значно витратнішим за обсягом пам’яті.

Wireshark, будучи інструментом з відкритим вихідним кодом, дозволяє будь-якому користувачеві завантажувати та вдосконалювати свій код. Завдяки цьому багато розробників у світі можуть налаштовувати та покращувати його функціональність. У порівнянні з цим, Colasoft розробляється лише командою компанії Capsa, що обмежує можливості налаштування. Таким чином, Wireshark є відмінним вибором для тих, хто хоче глибше зрозуміти програмування і налаштування мереж, оскільки він також підтримує різні платформи, включаючи Linux, Solaris, OS X і Windows.

Крім того, деякі дослідники вдосконалюють Wireshark, зокрема для виявлення атак типу Denial of Service (DoS), таких як атака ping flood, яка намагається перевантажити пристрій жертви великою кількістю команд ping.

Кожен з інструментів мережевого аналізу – Wireshark, TCPDump і Colasoft – має свої унікальні особливості, хоча всі вони володіють загальними характеристиками для роботи з мережевими властивостями. При порівнянні цих інструментів важливими є різноманітні якісні та кількісні параметри, такі як підтримка протоколів, відкритість вихідного коду, сумісність з платформами, використання бібліотеки libpcap, підтримка формату PCAP, типи інтерфейсу, вартість, методи декодування, здатність виявляти аномальні пакети, можливості матричного моніторингу мережі і реконструкція TCP-потоків.

Коласофт вирізняється перевагами в області графічних і матричних звітів, надаючи детальніше візуальне представлення даних. Wireshark, з іншого боку, є інструментом з відкритим вихідним кодом, що дозволяє користувачам розробляти та налаштовувати його функціональність відповідно до власних потреб. Цей інструмент сумісний з різними платформами, такими як Linux і Windows. На відміну від цього, Colasoft працює лише на платформах Windows.

TCPDump відзначається своєю легкістю і економічним використанням пам’яті, що робить його ідеальним для віддаленого моніторингу мережі через командний рядок. Однак, в плані підтримки протоколів, Wireshark є найкращим варіантом завдяки підтримці понад 1000 протоколів, що робить його відмінним для моніторингу різноманітних мереж, включаючи ті, що використовують відео та аудіо додатки. У порівнянні з цим, Colasoft підтримує близько 300 протоколів, а TCPDump обмежений лише протоколами TCP/IP, не підтримуючи UDP.

Отже, аналіз мережевого трафіку є критично важливим для забезпечення ефективності, безпеки та стабільності комп'ютерних мереж. Для цього використовуються різноманітні інструменти, кожен з яких має свої унікальні особливості, переваги та обмеження.

TCPDump – це легкий і портативний інструмент командного рядка для захоплення і аналізу мережевого трафіку. Він підтримує тільки TCP/IP і має обмежені можливості візуалізації даних, що робить його менш зручним для детального аналізу в порівнянні з іншими інструментами. TCPDump має невеликий розмір та низькі вимоги до пам’яті, але його текстовий інтерфейс обмежує користувача в можливостях глибокого аналізу.

Wireshark є одним з найпопулярніших і найпотужніших інструментів для аналізу мережевого трафіку. Його переваги включають підтримку понад 1000 протоколів і наявність графічного інтерфейсу, що дозволяє детально аналізувати трафік у зручному візуальному форматі. Wireshark має відкритий вихідний код, що дозволяє його налаштовувати та розширювати відповідно до потреб користувачів. Проте, його вимоги до пам’яті є значними, і інструмент працює з різними платформами, такими як Linux і Windows.

Colasoft пропонує потужний графічний інтерфейс для аналізу мережевого трафіку, що включає матричні та графічні звіти. Хоча Colasoft обмежений платформою Windows і підтримує менше протоколів (близько 300), він надає детальні звіти, зручний візуальний інтерфейс і функції для моніторингу в реальному часі. Це робить його корисним для адміністраторів

мереж, які потребують детальної інформації про мережевий трафік у зручному форматі.

Загалом, кожен інструмент має свої сильні сторони та обмеження. Вибір відповідного інструменту для аналізу мережевого трафіку залежить від конкретних потреб, таких як підтримка протоколів, типи інтерфейсу, вимоги до пам'яті та вартість. TCPDump є економічним і легким, Wireshark забезпечує потужний аналіз з відкритим вихідним кодом, а Colasoft пропонує зручний графічний інтерфейс для детального моніторингу.

2.2. Моделювання мережевого трафіку

Існуючі генератори трафіку зосереджені на підтримці статистичних характеристик мережевого трафіку, використовуючи три основні підходи. Однак для нашої теми, важливо не тільки зберігати статистичні розподіли, а й виявляти аномалії та аналізувати поведінку мережевих потоків у режимі реального часу.

Генератори трафіку на рівні пакетів, як iPerf, працюють на основі розміру пакету та часу між пакетами, де користувач визначає статистичні параметри. Проте для нашої системи, орієнтованої на аналіз реального трафіку, важливо враховувати, що такі генератори, як iPerf, обмежені у відтворенні реальних потоків. Їх використання часто обмежене лише тестами продуктивності, не враховуючи детальних особливостей різноманітних потоків, які спостерігаються у великих мережах, як у корпоративних середовищах.

Генератори на рівні додатків, такі як Surge, імітують поведінку конкретних додатків, створюючи послідовності запитів, які повторюють шаблони трафіку цих додатків. Однак ці інструменти створюють лише один тип трафіку, що може не відображати всі можливі варіанти, що присутні у реальних мережах. Для нашої інтелектуальної системи важливо створити більш універсальний аналізатор, який здатен працювати з різними типами

трафіку, аналізуючи не лише статистичні, а й поведінкові характеристики потоку.

Генератори на рівні потоку, як Harpoon, моделюють потоки на основі восьми розподілів TCP і UDP, забезпечуючи більш точне відтворення інтернет-трафіку. Однак такі генератори все ще мають обмежену кількість параметрів для аналізу, що не дозволяє комплексно оцінювати різноманітність мережевого трафіку. У нашій системі необхідно подолати ці обмеження, розширюючи аналіз на більшу кількість характеристик, зокрема з урахуванням поведінкових патернів і взаємодій між різними рівнями мережі.

Дутта та ін. запропонували підхід для моделювання поведінки ботів, що імітують дії реальних користувачів, і продемонстрували його ефективність у виявленні вторгнень. Хоча їхній підхід базується на заздалегідь визначених правилах, в нашій роботі ми плануємо використовувати методи машинного навчання для автоматизації процесу виявлення та аналізу реальної поведінки користувачів. Це дозволить нашій інтелектуальній системі аналізу мережевого трафіку не тільки розпізнавати шаблони на основі попередньо заданих даних, але й адаптуватися до нових загроз та аномалій у трафіку.

Використання генераторів мережевого трафіку за останні роки значно збільшилось, зокрема через зростаючі вимоги до захисту конфіденційної інформації, які обмежують доступ до реальних даних. Синтетичні дані стають основним інструментом для моделювання трафіку, оскільки вони можуть бути створені на основі реальних даних, але без загрози порушення конфіденційності. Синтетичні дані мають ту перевагу, що вони побудовані на реальних розподілах, тому практично не відрізняються від вихідних. Це особливо корисно в ситуаціях, коли анонімізовані дані недостатньо точні або неефективні для аналітичних цілей, тоді як синтетичні дані дозволяють отримувати подібні результати без ризику витоку персональних даних.

Корисність синтетичних даних, підтверджена численними дослідженнями, відкриває можливості для їх застосування в нових галузях, зокрема в аналізі мережевого трафіку. У нашій роботі ми вивчаємо можливість

використання синтетичних даних для моделювання мережевого трафіку, що дозволить розширити можливості нашої інтелектуальної системи. Основна ідея полягає в тому, щоб створити генеративну модель на основі реальних даних від обмеженої кількості користувачів, які погодилися на збір даних, і на основі цих даних генерувати ширший та більш різноманітний трафік.

Згенерований трафік, який відображає ключові особливості вихідних даних, включаючи різноманіття користувачів, типи додатків і мережеві властивості, буде корисним для кількох завдань. Зокрема, його можна використовувати для аналізу вищого рівня, класифікації потоків і виявлення аномалій. Наші синтетичні дані можуть стати основою для тестування систем в реальних умовах, де потрібно моделювати поведінку мережі при різних навантаженнях. Це дозволить нашій системі аналізувати продуктивність і поведінку різних мережевих компонентів у корпоративних середовищах. Наприклад, згенерований трафік може бути використаний для оцінки пропускної здатності систем або в симульованих середовищах для моделювання реалістичного фоново трафіку.

Важливо, що наш підхід дозволить постійно створювати нові умови для тестування, що зменшить ризик того, що система не впорається з непередбаченою поведінкою трафіку. Таким чином, наша інтелектуальна система аналізу мережевого трафіку буде здатна не тільки виявляти загрози, але й адаптуватися до нових викликів, що виникають у реальних умовах використання мережевих інфраструктур.

Існуючі рішення здебільшого зосереджені на генеруванні трафіку, який статистично відповідає реальним даним, але, наскільки нам відомо, жодна система не здатна зберігати послідовність моделей мережевої активності. Для розуміння важливості збереження цих послідовностей можна розглянути такий приклад: завантаження PDF-файлу з певного сайту зазвичай відбувається після того, як користувач переглядає електронну пошту в Gmail. Хоча аналіз записаного трафіку може показати, що ця послідовність є статистично значущою, жоден із сучасних генераторів штучного трафіку не

зберігає цієї взаємопов’язаної поведінки. Через це система виявлення вторгнень (IDS), що оцінюється за допомогою згенерованого трафіку, може не виявити певні загрози або не ідентифікувати потенційні джерела атак, як це було б у випадку аналізу реального трафіку.

Це підводить нас до основної мети нашого дослідження: створити інтелектуальну систему для аналізу мережевого трафіку, яка здатна не тільки генерувати реалістичний трафік, але й зберігати послідовності дій користувачів. В рамках нашого дослідження ми пропонуємо створення генератора мережевого трафіку (Network Traffic Generator, NTG), який дозволить не тільки моделювати реалістичний фоновий трафік, а й підтримувати важливі послідовності мережевої активності. Такий підхід дозволить краще моделювати сценарії мережевої поведінки та забезпечити більш точну оцінку мережевих загроз.

NTG буде використовувати сотні параметрів для того, щоб зберегти якомога більше характеристик вихідного трафіку, що дозволить зробити систему універсальною та не прив'язаною до конкретного протоколу чи додатку. Потоки пакетів будуть кластеризовані для виявлення подібних мережевих дій, після чого методи машинного навчання допоможуть створити моделі послідовної поведінки. Ці моделі будуть інтегровані у згенерований трафік, дозволяючи системі не тільки аналізувати статистичні дані, а й ідентифікувати поведінкові закономірності, що можуть вказувати на аномалії або потенційні загрози .

Прототип генератора мережевого трафіку відіграє важливу роль у нашій роботі над інтелектуальною системою аналізу мережевого трафіку. Цей прототип складається з кількох ключових фаз, які включають взаємопов'язані компоненти, що забезпечують ефективний аналіз і генерацію трафіку.

На першому етапі здійснюється попередня обробка мережевого трафіку, що передбачає фільтрацію та нормалізацію даних, зібраних з мережі. Це дозволяє усунути шум та аномалії, які можуть спотворити результати подальшого аналізу. Після цього проводиться кластеризація потоків для

подібних дій, що дозволяє групувати трафік за схожими характеристиками або поведінкою. Цей крок є критично важливим для виявлення шаблонів, які можуть свідчити про типові або аномальні активності в мережі.

Наступним етапом є моделювання послідовностей діяльності, де на основі кластеризованих даних створюються моделі, що імітують послідовність дій користувачів. Це дозволяє прогнозувати, які дії можуть бути здійснені в майбутньому, що важливо для попередження можливих загроз. Завершальною фазою є генерація трафіку на основі створених моделей, що дає змогу тестувати стійкість системи та її реакцію на різні сценарії. Це значно покращує наші можливості в аналізі та управлінні мережевою безпекою.

Кожен з цих етапів тісно пов'язаний із загальною метою нашої системи – забезпечити глибокий аналіз мережевого трафіку для виявлення та нейтралізації потенційних загроз, що сприяє підтриманню високого рівня безпеки в інформаційних системах зображено на рисунку 2.1.

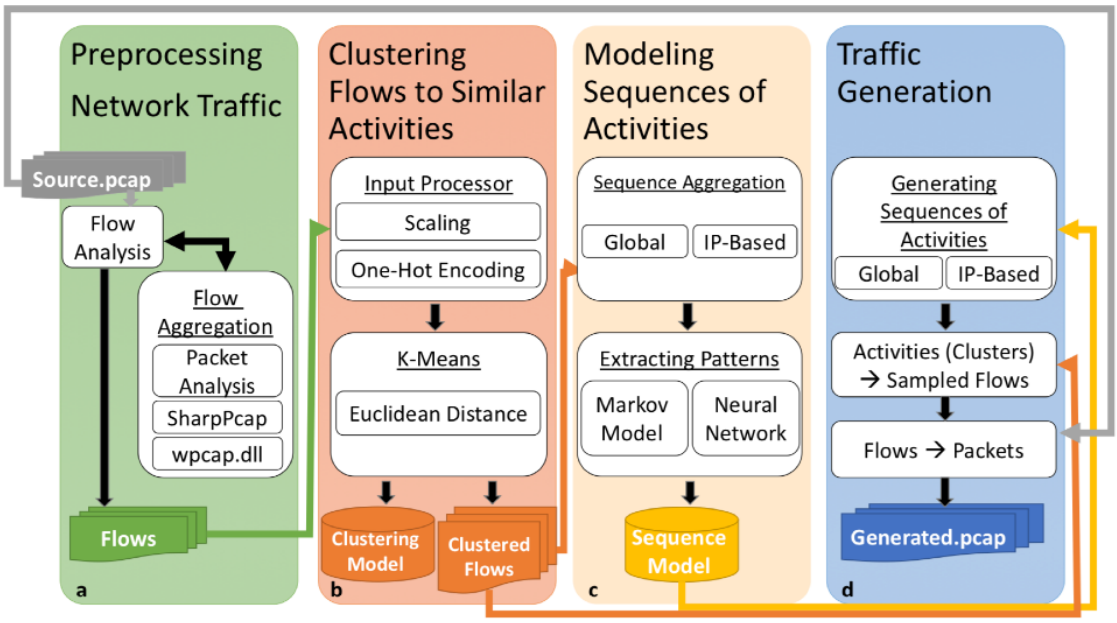

Рисунок 2.1. Фази мережевого трафіку

Вихідний трафік спочатку проходить етап попередньої обробки, під час якого дані агрегуються в більш узагальнені потоки. Наступним кроком є

кластеризація цих потоків, що дозволяє групувати їх за типами мережевої діяльності. Далі ми розробляємо модель послідовності, яка навчається на основі виявлених шаблонів у послідовностях мережевих дій. Для цього використовуються кілька методів, що підтримують різноманітні підходи до аналізу даних.

При попередній обробці ми акцентуємо увагу на двох найбільш вживаних протоколах прикладного рівня – DNS і HTTP. Однак, завдяки модульній архітектурі, прототип можна легко адаптувати для інтеграції додаткових протоколів шляхом вилучення нових функцій. Після обробки записи потоків підлягають подальшому аналізу для виявлення подібних дій, що дозволяє покращити точність класифікації.

На етапі моделювання послідовностей діяльності використовуються моделі Маркова або нейронні мережі, що дозволяє створювати точні прогнози на основі отриманих даних. Завершальний етап полягає у використанні навченої моделі для генерації штучного мережевого трафіку, що імітує реальні сценарії використання. Цей підхід відкриває нові можливості для тестування та оптимізації мережевої безпеки, що є критично важливим у сучасному цифровому середовищі.

Основна сутність, яку ми розглядаємо, — це потік трафіку, що визначається як послідовність пакетів, переданих між двома вузлами під час одного сеансу. Ці два вузли формують унікальний 4-ту пакету, що складається з вихідних та цільових IP-адрес, а також портів. Сеанс TCP розпочинається з успішного handshake-у і завершується таймаутом або отриманням пакета з прапорцем RST або FIN від одного з вузлів. Оскільки в рамках цього протоколу вузли обмінюються пакетами по черзі, для кожного напрямку сеансу генеруються окремі потоки. У випадку з UDP сеанс включає всі пакети, що передаються від клієнта до сервера, до моменту досягнення встановленого часу простою або максимальної тривалості з’єднання, що забезпечує формування одного потоку на сеанс.

Ми визначили 205 унікальних функцій, які формують поведінку мережевого трафіку, витягуючи їх з трьох різних рівнів: рівня пакету, рівня потоку та рівня застосування. Особливості на рівні пакету витягуються з кожного пакета в транспортному шарі. Ми виділили п’ятнадцять атрибутів для TCP та одну функцію для UDP. Кожна необроблена функція перетворюється на набір агрегативних ознак, які описують специфіку пакетів у заданому потоці. Наприклад, "розмір пакету" деталізується до одинадцяти агрегованих характеристик, таких як середній розмір пакету, ентропія розміру пакету тощо.

Функції рівня потоку, як очікується, витягуються безпосередньо з самого потоку, і ми виділили шість таких функцій; одним із прикладів є кількість пакетів у потоці. Функції на рівні застосування стосуються специфічних програм і ілюструють особливості, що стосуються тільки певних застосунків. Наприклад, агент HTTP користувача виділяється для потоків, що відповідають HTTP-додаткам. Ми витягли вісім функцій на основі DNS, вісім функцій на основі HTTP та сім функцій на основі SSL.

Таблиця 2.2 містить приклади функцій, які спостерігаються в різних протоколах мережевого трафіку.

Таблиця 2.2

Функцції на різних рівнях протоколів

Level	Features
Flow	time of day, packet interarrival time, the number of packets
TCP	time to live, seq num, ack num
UDP	checksum invalid
DNS	additional records, canon names, response count
HTTP	cookie, unique content types, bytes

Вивчаючи різні варіанти агрегації моделей послідовностей мережевих дій, ми стикнулися з важливими питаннями. Чи варто враховувати послідовність дій, виконуваних усіма учасниками мережі, чи краще зосередитися на послідовностях дій, які здійснюються між парами

комунікаційних IP-адрес? Для цього ми аналізуємо обидва підходи агрегації: перший, глобальний, об'єднує весь трафік в одну послідовність, а другий, на основі IP, розглядає кожну пару комунікаційних IP-адрес як окрему послідовність. У рамках цього дослідження ми вивчаємо, який із двох підходів буде більш доцільним для наших цілей.

Послідовність s представляє собою впорядкований перелік мережевих дій, де кожна дія ідентифікується через відповідний кластер, що пов'язаний з потоками, що здійснюються послідовно: $s = \{c_1 \rightarrow c_2 \rightarrow \ldots \rightarrow c_i\}$, де $i \in K$. У разі глобальної агрегації всі потоки трафіку об'єднуються в одну послідовність, тоді як у випадку агрегації на основі IP потоки, що належать до однієї послідовності, ділять спільні джерела та цільові IP-адреси. Час першого пакета в кожному потоці t_i визначає порядок дій у послідовності, а час переходу між двома послідовними діями становить $t_{i+1} - t_i$. При цьому можливо, що послідовні стани будуть однаковими (наприклад, $c_3 \rightarrow c_2 \rightarrow c_2$).

На рис. 2.2 представлено відмінності між агрегацією послідовностей у глобальному масштабі та за IP-адресою. У глобальній агрегації всі потоки формують єдину послідовність (жовта стрілка вказує їхній порядок), в той час як в агрегації на основі IP для кожної унікальної пари вихідних і цільових IP-адрес створюється окрема послідовність. У нашому наборі прикладів є дві унікальні пари IP, які призводять до формування двох послідовностей: перша охоплює три потоки, що пройшли з 1.1.1.2 до 2.2.2.1 (позначені червоним кольором), а друга складається з двох потоків, які пройшли від 2.2.2.3 до 2.2.2.1 (позначено синім кольором). У агрегації на основі IP ми також додаємо початкові та кінцеві стани для кожної послідовності, що дозволяє вибирати початковий стан та прогнозувати кінець послідовності (вивчення розмірів послідовностей).

Для того, щоб знайти ефективний алгоритм вилучення моделей послідовностей, ми досліджуємо два підходи: модель Маркова та модель нейронної мови.

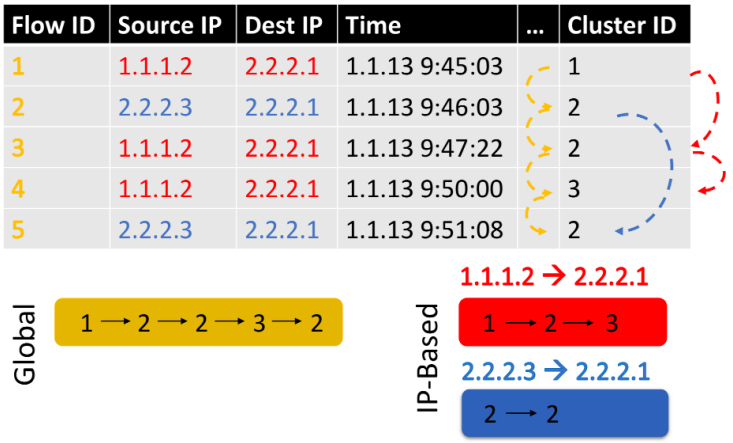

Рисунок 2.2. Ілюстрація двох підходів до агрегації послідовностей: глобальна агрегація та агрегація на основі IP-адрес

У випадку глобальної агрегації (жовтий) усі потоки трафіку об'єднуються в єдину послідовність. Натомість в агрегації на основі IP (червоний і синій) для кожної унікальної пари IP-адрес джерела і призначення формується окрема послідовність. Ця послідовність складається з ідентифікаторів кластерів, що пов'язані з відповідними потоками, і впорядкована відповідно до часу надходження першого пакета в кожному потоці.

Ми пропонуємо генеративний метод, що відповідає навченим моделям. Зокрема, нова послідовність мережевих дій, яка представлена через ідентифікатори кластерів, формується шляхом "склеювання" коротких фрагментів мережевих дій, що перекриваються, які часто зустрічаються в навчальних даних. Правила для визначення ймовірності наступної мережевої активності залежать від обраного методу (модель Маркова або модель нейронної мови) і мають дещо неявний характер.

Для генерації мережевого трафіку спочатку створюються послідовності дій (кластери); потім потоки, відповідні кожному кластеру, відбираються з

вихідного трафіку, а пакети цих потоків передаються у порядку, визначеному згенерованими послідовностями. У глобальній агрегації послідовностей модель навчається на потоках усіх учасників мережі, в результаті чого формується єдина глобальна послідовність, що імітує мережеві дії всіх однорангових учасників, упорядковані за часом. Завершення послідовності відбувається, коли досягається приблизний час закінчення, тому розмір послідовності значною мірою залежить від розрахункових часів передачі, які базуються на статистичних даних про часи переходу, отриманих із вихідного трафіку.

В агрегації на основі IP формуються послідовності для кожної пари IP-адрес (адреси регенеруються), і ці послідовності завершуються, коли досягається стан завершення. При використанні моделі нейронної мови контекст m попередніх кластерів необхідно оновлювати в кожній ітерації для прогнозування наступного кластеру; у випадку з моделлю Маркова попередній кластер також оновлюється відповідним чином. Алгоритм генерації трафіку описано в формальному вигляді нижче (рисунок 2.3).

```
1  D* ← ExtractFlows(D);
2  D* ← AttachClusters(D*, M_Clusters);
3  t_start, t_end ← FirstTime(D), LastTime(D);
4  trans ← ExtractHist(TransitionTimes(D*), bins = √|D*|); // transition times between flows
5  if using global aggregation then
6      c ← SampleFirstCluster(M_Sequences);
7      t ← t_start;
8      while t < t_end do
9          f ← SampleFlow(D*,c);
10         t ← t+ Sample(trans);
11         context ← Update (context,c);
12         c ← PredictCluster(M_Sequences,context);
13         Add GetPackets(f,D) to results set;
14 else
       // If using IP-based aggregation
15     for each pair IP_src,IP_dest in D do
16         context ← Update (context,start);
17         c ← PredictCluster(M_Sequences,context);
18         IP*_src,IP*_dest ← GenerateIPs();
19         t ← t_start;
20         while c! = end do
21             f ← SampleFlow(D*,c);
22             t ← t+ Sample(trans);
23             Add SetIPs(GetPackets(f,D),IP*_src,IP*_dest) to results set;
24             context ← Update (context,c);
25             c ← PredictCluster(M_Sequences,context);
```

Рисунок 2.3. Алгоритм мережевого трафіку

Цей алгоритм приймає на вхід дані мережевого трафіку (D), які використовуються для відновлення записаних пакетів при створенні нового трафіку. Крім того, він враховує модель кластеризації (MClusters) та модель послідовностей (MSequences).

Розрахунки були проведені на двох умовних колекціях мережевого трафіку, характеристики яких наведені в таблиці 3.5. Як видно з таблиці, ці дві колекції мають суттєво різні характеристики: у них різний час збору, і трафік EMC містить майже втричі більше потоків. Однак потоки в трафіку EMC значно коротші, ніж у трафіку BGU, і містять значно менше пакетів.

Додатково, хоча трафік EMC має приблизно половину від загальної кількості послідовностей у порівнянні з трафіком BGU, він демонструє у шість разів більше комунікаційних пар IP-адрес. Це ускладнює структуру трафіку, принаймні в контексті агрегації на основі IP.

Таблиця 2.3

Різні характеристики трафіку, використовуваного для експериментів

	BGU	**EMC**
Розмір	20.1 Гб	2.79 Гб
Кількість пакетів	193 50 505	8 700 370
Кількість потоків	190 462	549 350
Source IPs	1 329	633
Destination IPs	1 309	435
Source та Destination	3 322	24 760
Мінімальний час	1.10.2013 1:05:57	13.06.2013 07:38:27
Максимальний час	2.10.2013 1:11:07	13.06.2017 12:59:14
Розмір послідовності	57 721 потоків	22 367 потоків

У результаті проведеного аналізу двох колекцій мережевого трафіку було виявлено суттєві відмінності в їх характеристиках. Колекція трафіку EMC демонструє значно більшу кількість потоків, але ці потоки є коротшими

і містять менше пакетів у порівнянні з колекцією BGU. Хоча трафік EMC має приблизно половину послідовностей у порівнянні з BGU, він показує у шість разів більше комунікаційних пар IP-адрес. Ці фактори ускладнюють агрегацію на основі IP, вказуючи на необхідність врахування різноманітних характеристик трафіку під час моделювання мережевих дій.

2.3. Функціонально-вартісний аналіз

Оцінка основних параметрів, що впливають на створення скрипту для отримання мережевих метаданих та їх відбитків, таких як JA3 і HASSH, проводиться на основі файлів захоплення пакетів (.pcap) або в режимі живого мережевого трафіку. Розробка програмної реалізації виконана за допомогою мови програмування Python та середовища розробки PyCharm від компанії JetBrains.

Основна функція F_0 виконує роль скрипту для отримання мережевих метаданих.

Виходячи з конкретних цілей, виділяються ключові функції програми:

- F_1 – вибір джерела вхідного трафіку;
- F_2 – вибір підтримуваних протоколів;
- F_3 – вибір кількості параметрів для виведення даних.

Кожній з функцій відповідає кілька варіантів реалізації:

- Функція F1:
  1. поточний мережевий трафік;
  2. трафік з файлу захоплених пакетів.
- Функція F2:
  1. протокол SSL/TLS;
  2. протокол SSH;
  3. протокол RDP;
  4. протокол HTTP.
- Функція F3:

1. стандартне виведення даних (мінімальна кількість параметрів);

2. розширене виведення даних (більша кількість параметрів, ніж мінімальна).

На основі цих варіантів формується морфологічна карта системи (рисунок 2.4), яка слугує основою для побудови позитивно-нагативної матриці варіантів основних функцій.

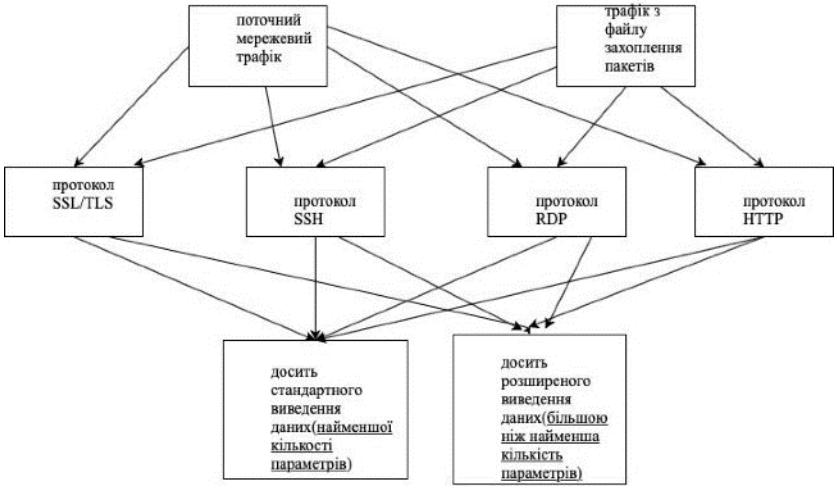

Рисунок 2.4. Морфологічна карта

Виходячи з наведеної карти, була створена позитивно-негативна матриця (таблиця 2.4).

Таблиця 2.4

Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Недоліки	Переваги
F1	A	Потреба додаткового аналізу відповідного трафіку по мірі його надходження	Більш проста можливість отримання трафіку

Завершення табл. 2.4

	B	Функціонал зчитування трафіку з файлу захоплених пакетів	Відсутність потреби аналізу помірі надходження(фіксований об’єм даних)
F2	A	Складна здатність масштабування. Поверхневий аналіз.	Низькі часові витрати для обробки даних
	B	Розширені часові витрати для аналізу. Потреба великих ресурсів	Можливість глибокого аналізу пакетів. Легко масштабуються
F3	A	Недостатня кількість функціоналу	Низька ціна
	B	Висока вартість	Можливість використання більшої кількості функціоналу

Для опису прототипу програмного додатку використовуються параметри X1 – X5. Спираючись на інформацію, надану в літературі, ми визначили мінімальні, середні та максимально допустимі значення (таблиця 2.5).

Таблиця 2.5

Система параметрів додатку

Назва параметру	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Продуктивність мови програмування	X1	Оп/мс	4000	8000	16000
Орієнтовна кількість програмного коду	X2	рядків	4000	1500	1000
Середня ціна при аналізі за певним критерієм	X3	Кількість автомобілів	5	3	0
Середнє навантаження	X4	Кількість годин на добу	0	10	24

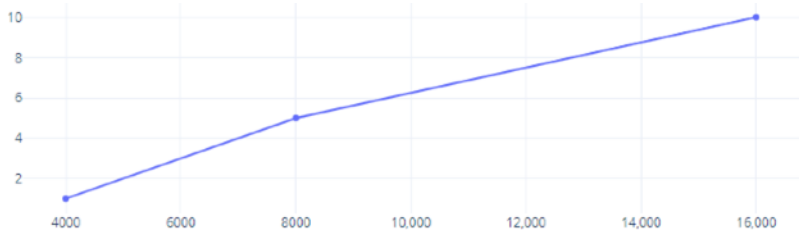

Рисунок 2.5. Бальна оцінка продуктивності мови програмування

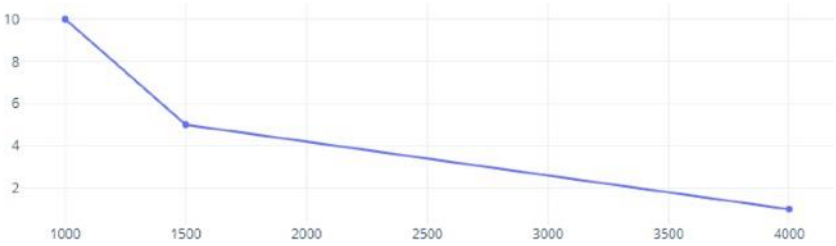

Рисунок 2.6. Бальна оцінка орієнтовної кількісті програмного коду

Рисунок 2.7. Бальна оцінка середньої ціни при аналізі за певним критерієм

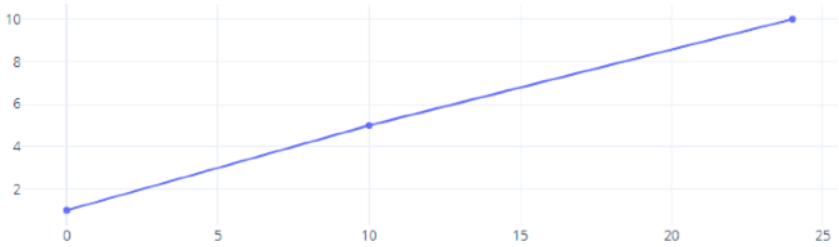

Рисунок 2.8. Бальна оцінка середнього навантаження

Важливість параметрів оцінюється за допомогою методів попарного порівняння, де ранги варіюються від 1 до 5. Результати цієї оцінки представлені в таблицях 2.6 і 2.7.

Таблиця 2.6

Результати оцінки параметрів

Познач. параметра	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
	1	2	3	4	5	6	7			
X1	3	2	4	3	3	2	3	20	2.5	6.25
X2	4	4	3	4	4	4	4	27	9.5	90.25
X3	1	3	2	2	2	3	1	14	-3.5	12.25
X4	2	1	1	1	1	1	2	9	-8.5	72.25
Разом	10	10	10	10	10	10	10	70	0	181

За найбільший ранг приймаємо 4

Таблиця 2.7

Попарне зрівняння параметрів

Параметри	Експерти							Підсумкова	Числове
	1	2	3	4	5	6	7	оцінка	значення
X1,X2	<	<	>	<	<	<	<	<	0.5
X1,X3	>	<	>	>	>	<	>	>	1.5
X1,X4	>	>	>	>	>	>	>	>	1.5
X2,X3	>	>	>	>	>	>	>	>	1.5
X2,X4	>	>	>	>	>	>	>	>	1.5
X3,X4	>	>	>	>	>	>	>	>	1.5

Визначимо коефіцієнт конкординації :

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 181}{7^2(4^3 - 4)} = 0.72 > W = 0{,}67.$$

Оскільки коефіцієнт конкордації перевищує нормативне значення, результати оцінки визнаються достовірними. Розрахунок вагомості окремих параметрів представлений у таблиці 2.8.

Таблиця 2.8

Розрахунок вагомості параметрів.

Параметри	Параметри				Перший крок		Другий крок		Третій крок	
	X1	X2	X3	X4	b_i	Кві	b_i^1	$\hat{E}_{\hat{a}^3}^1$	b_i^1	$\hat{E}_{\hat{a}^3}^1$
X1	1.0	0.5	1.5	1.5	5	0.3	16.75	0.2757	60.75	0.273495
X2	1.5	1.0	1.5	1.5	5.5	0.33	22	0.36214	80.125	0.36072
X3	0.5	0.5	1.0	1.5	3.5	0.21	12.5	0.205761	46.125	0.207653
X4	0.5	0.5	0.5	1.0	2.5	0.15	9.5	0,1563	35.125	0.158132
Всього:					16.5	1	60.75	1	222.125	1

Таблиця 2.9

Розрахунок показників рівня якості варіантів реалізації

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	а	1	8	0.276	2.88
F2	а	9000	4.2	0.36	1.1592
	б	12000	6.6	0.36	1.8216
F3	а	1500	5	0.207	0.78
	б	3000	3.3	0.207	0.5148
F4	а	8	3.3	0.158	0.6765

Обчислимо коефіцієнти якості для кожного з варіантів розробки:

- КК1 = 2.28 + 1.15 + 0.78 + 0.6765 = 4.88;
- КК2 = 2.28 + 1.82 + 0.78 + 0.6765 = 5.56;
- КК3 = 2.28 + 1.15 + 0.51 + 0.6765 = 4.62;
- КК4 = 2.28 + 1.82 + 0.52 + 0.6765 = 5.29.

Згідно з проведеними розрахунками, найкращим варіантом є останній, оскільки він має найвищий коефіцієнт технічного рівня.

2.4. Висновки до розділу 2

У цьому розділі було проведено всебічний аналіз практичної реалізації інтелектуальної системи для аналізу мережевого трафіку. Ми розглянули основні аспекти моніторингу мережевої активності, зокрема засоби, які забезпечують ефективний збір і обробку даних про трафік. Застосовані технології моделювання мережевого трафіку дозволили виявити закономірності та аномалії, що є критично важливими для забезпечення безпеки мережі.

У ході функціонально-вартісного аналізу було оцінено витрати та вигоди, пов’язані з реалізацією системи, що підкреслює її ефективність і доцільність впровадження. Результати аналізу показали, що інтелектуальна система має потенціал для підвищення рівня захисту мереж і оптимізації ресурсів.

Загалом, проведене дослідження доводить, що інтеграція сучасних методів аналізу мережевого трафіку здатна істотно поліпшити процеси моніторингу, виявлення загроз та управління мережевими ресурсами. У наступних розділах буде детально розглянуто подальший розвиток системи та її практичні застосування.

РОЗДІЛ 3
РЕКОМЕНДАЦІЇ ЩОДО ВИКОРИСТАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ

3.1.Оптимізація параметрів системи для підвищення точності аналізу

Оптимізація параметрів інтелектуальної системи аналізу мережевого трафіку є важливим етапом для підвищення ефективності та точності роботи системи. Це дозволяє зменшити кількість хибних спрацьовувань, збільшити швидкість виявлення загроз та покращити загальну продуктивність системи. Основні аспекти оптимізації охоплюють налаштування апаратних ресурсів, вибір алгоритмів для обробки даних, калібрування правил фільтрації та інші важливі параметри.

Оптимізація апаратного забезпечення, на якому працює система аналізу трафіку, відіграє критичну роль. Якщо система має недостатні обчислювальні ресурси (процесор, оперативну пам'ять), це може призвести до затримок у аналізі трафіку та зниження продуктивності. Основні кроки включають:

- Збільшення пропускної здатності. Для того щоб система могла обробляти великий обсяг трафіку без затримок, потрібно збільшити потужність мережевих інтерфейсів;

- Оптимізація використання пам'яті. Забезпечення достатньої кількості оперативної пам'яті для безперервної роботи аналізу та обробки даних у реальному час;

- Паралельна обробка. Використання багатоядерних процесорів і розподілених обчислень для прискорення аналізу.

Інтелектуальна система аналізу трафіку використовує різноманітні алгоритми для обробки та аналізу мережевих пакетів.

Для покращення точності система може використовувати алгоритми поведінкового аналізу, що дозволяють виявляти нетипову поведінку в мережі. Вибір алгоритмів класифікації (наприклад, алгоритми машинного навчання) може вплинути на точність виявлення аномалій.

Важливим аспектом є правильне налаштування порогових значень для визначення загроз. Надто низькі пороги можуть призвести до численних хибних спрацювань, а надто високі – до пропуску загроз. Необхідно проводити тестування та калібрування цих значень на основі реальних даних.

Налаштування алгоритмів для відсівання непотрібного або безпечного трафіку. Це дозволяє системі зосередитися на потенційно шкідливих пакетах і знизити кількість хибно позитивних спрацьовувань.

Правила фільтрації визначають, які типи трафіку аналізуються як потенційні загрози. Оптимізація цих правил є критично важливою для зменшення навантаження на систему та підвищення її точності.

Використання динамічних систем, які адаптуються до змін у поведінці мережевого трафіку. Це дозволяє системі постійно оновлювати свої правила для виявлення нових типів загроз. Оптимізація аналізу вмісту пакетів дозволяє виявляти загрози навіть у зашифрованому трафіку. Це досягається шляхом використання розширених алгоритмів, таких як Deep Packet Inspection (глибокий аналіз пакетів).

Для інтелектуальної системи аналізу трафіку важливо забезпечити мінімальні затримки при обробці даних, що досягається за допомогою:

- Буферизація та оптимізація черг. Ефективна робота з буферами для запобігання втраті пакетів при високому навантаженні;

- Розподілені обчислення. Використання розподілених обчислень або хмарних технологій для паралельного аналізу великих обсягів трафіку.

Останнім, але дуже важливим аспектом є впровадження механізмів самонавчання та адаптації. Використання технологій машинного навчання та штучного інтелекту дозволяє системі покращувати свої алгоритми на основі отриманих даних:

- Адаптивні моделі машинного навчання. Здатність системи автоматично оновлювати моделі виявлення загроз на основі нових даних та загроз;

- Зворотній зв'язок від користувачів. Система повинна отримувати та обробляти зворотний зв'язок від адміністраторів мережі для подальшого покращення своїх моделей і правил.

Постійне тестування системи на основі реальних мережевих сценаріїв та даних є ключовим етапом оптимізації. Це дозволяє виявляти слабкі місця, удосконалювати налаштування та гарантувати, що система працює з максимальною ефективністю. Проведення тестів із використанням як реальних, так і симуляційних даних для оцінки ефективності виявлення. Встановлення регулярних оновлень для усунення вразливостей та додавання нових функцій.

Оптимізація параметрів інтелектуальної системи аналізу мережевого трафіку є комплексним процесом, що охоплює як технічні, так і аналітичні аспекти. Від правильного налаштування ресурсів і алгоритмів залежить точність виявлення загроз, ефективність системи та її здатність швидко реагувати на нові виклики у мережевому середовищі.

3.2.Використання системи в різних типах мереж

Інтелектуальні системи аналізу мережевого трафіку можуть використовуватися в різноманітних типах мереж, кожен з яких має свої особливості та вимоги до безпеки й ефективності роботи системи. Використання таких систем варіюється залежно від структури мережі, її масштабів, пропускної здатності та типу трафіку. Правильне налаштування й

адаптація інтелектуальної системи до конкретної мережі дозволяє забезпечити оптимальну продуктивність та надійний захист від загроз.

У локальних мережах (LAN) інтелектуальні системи аналізу трафіку зазвичай використовуються для забезпечення безпеки в організаціях чи офісах, де велика кількість пристроїв підключається до однієї фізичної або бездротової інфраструктури. В локальних мережах зазвичай немає великих обсягів зовнішнього трафіку, але існує ризик внутрішніх загроз або проникнення через незахищені пристрої. Інтелектуальні системи в таких мережах можуть аналізувати поведінку користувачів і пристроїв, виявляти аномальні дії, такі як несанкціонований доступ до ресурсів чи спроби встановити з’єднання з підозрілими зовнішніми серверами. Основний акцент у таких мережах робиться на внутрішній безпеці, захисті від витоків інформації та контролі доступу до важливих даних.

У глобальних мережах (WAN), які охоплюють великі географічні області, завдання інтелектуальних систем стає складнішим через необхідність обробки великого обсягу трафіку, який проходить між різними регіональними офісами чи підрозділами організації. WAN можуть бути підключені до Інтернету, що підвищує ризик зовнішніх атак, таких як DDoS, фішинг або вторгнення в мережу через незахищені канали зв'язку. В такій мережі інтелектуальні системи повинні фокусуватися на аналізі маршрутизації трафіку, захисті точок виходу в Інтернет, а також забезпеченні безпеки передавання даних між сегментами мережі. Важливою є можливість інтеграції системи з іншими рішеннями безпеки, такими як VPN або фаєрволи, для забезпечення комплексного захисту всієї мережі.

У хмарних мережах (Cloud) інтелектуальні системи аналізу трафіку грають ключову роль, оскільки все більше організацій переміщують свої дані й сервіси в хмару. У таких середовищах захист даних та контроль доступу стають критичними завданнями. Хмарна інфраструктура зазвичай включає безліч користувачів та додатків, що працюють одночасно, що підвищує ризики виникнення вразливостей і кібератак. Інтелектуальні системи повинні

забезпечувати захист як від внутрішніх загроз (від користувачів із неналежними правами доступу), так і від зовнішніх атак (таких як злом хмарних акаунтів чи проникнення через загальнодоступні мережі). Оптимізація систем в хмарних мережах також передбачає можливість гнучкого масштабування ресурсів для аналізу великого обсягу даних і динамічного навантаження на інфраструктуру.

У мережах мобільного зв'язку (3G, 4G, 5G) використання інтелектуальних систем стає дедалі важливішим через збільшення кількості пристроїв, що підключаються до мережі. У таких мережах, де основними користувачами є смартфони та інші мобільні пристрої, існує високий ризик атак на окремі пристрої, фішингових атак або поширення шкідливого ПЗ через незахищені мобільні додатки. Інтелектуальні системи повинні мати можливість аналізувати високошвидкісний мобільний трафік та виявляти загрози, пов'язані з мобільними мережами. Одним з головних викликів є швидка передача великого обсягу даних у мобільних мережах, що вимагає оптимізованої роботи системи для швидкого виявлення аномалій без створення затримок у мережі.

У безпровідних мережах (WLAN) інтелектуальні системи можуть використовуватися для моніторингу та забезпечення безпеки пристроїв, що підключаються через Wi-Fi. В таких мережах існують специфічні загрози, такі як несанкціоноване підключення до мережі, атаки типу «людина посередині» (Man-in-the-Middle), підслуховування трафіку чи використання слабких паролів для доступу до мережі. Інтелектуальна система аналізу трафіку може допомогти виявляти підозрілі дії, такі як спроби зламати паролі до Wi-Fi чи підключення до шкідливих точок доступу, а також забезпечити шифрування трафіку та захист від витоку конфіденційних даних.

В корпоративних мережах інтелектуальні системи зазвичай інтегруються з іншими системами безпеки, такими як системи виявлення вторгнень (IDS) та запобігання вторгнень (IPS), а також з рішеннями для управління інформаційною безпекою (SIEM). Це дозволяє забезпечити

багаторівневий захист та більш глибокий аналіз усіх аспектів роботи мережі, включаючи аналіз логів, контроль доступу до ресурсів, виявлення аномальної активності та реагування на інциденти безпеки в режимі реального часу. Такі системи можуть забезпечувати цілісний моніторинг усіх сегментів корпоративної мережі, що дозволяє вчасно виявляти загрози, координувати реагування на інциденти та забезпечувати відповідність нормативним вимогам.

Таким чином, інтелектуальні системи аналізу мережевого трафіку можуть бути адаптовані та оптимізовані для використання в різних типах мереж залежно від їх специфіки. Важливою є гнучкість цих систем, можливість масштабування та інтеграція з іншими рішеннями для забезпечення комплексного захисту мережі та ефективного реагування на нові загрози.

3.3.Рекомендації щодо подальшого розвитку та інтеграції системи з іншими рішеннями безпеки

Інтелектуальні системи аналізу мережевого трафіку є важливими інструментами для виявлення загроз і забезпечення безпеки в сучасних мережах. Однак для підвищення їх ефективності та здатності протидіяти складним і новим видам атак необхідно постійно вдосконалювати їх функціональні можливості та інтегрувати з іншими системами захисту. Це дозволить створити єдину екосистему безпеки, здатну ефективніше реагувати на кіберзагрози та захищати мережі різного масштабу. Нижче представлені основні рекомендації щодо розвитку та інтеграції інтелектуальних систем аналізу трафіку.

Однією з головних рекомендацій є інтеграція інтелектуальної системи з рішеннями для управління інформаційною безпекою (SIEM). Системи SIEM збирають, об'єднують і аналізують дані з різних джерел, включаючи журнали подій (лог-файли), системи моніторингу трафіку, а також інші компоненти

безпеки. Інтеграція з SIEM дозволить забезпечити більш повний огляд ситуації в мережі, оскільки інформація з інтелектуальної системи аналізу трафіку буде збагачуватися даними з інших джерел. Це допоможе швидше і точніше ідентифікувати загрози, побудувати контекст навколо інцидентів безпеки та покращити реагування на атаки.

Інша важлива рекомендація полягає в впровадженні механізмів автоматичного реагування на загрози. Оскільки сучасні атаки можуть бути дуже швидкими, а деякі загрози здатні завдати шкоди мережі за лічені секунди, інтеграція з системами автоматичного реагування, такими як рішення для запобігання вторгнень (IPS), є критично важливою. Інтелектуальна система повинна мати змогу не тільки виявляти загрози, але й автоматично ініціювати контрзаходи, такі як блокування підозрілих IP-адрес, закриття сесій або ізоляція скомпрометованих пристроїв. Це зменшить навантаження на адміністраторів безпеки та забезпечить більш оперативне реагування на інциденти.

Також доцільним є розвиток можливостей використання штучного інтелекту (AI) та машинного навчання (ML) для підвищення точності виявлення загроз та мінімізації хибнопозитивних результатів. Використання AI/ML дозволить системі краще адаптуватися до нових, раніше невідомих атак, шляхом аналізу великих обсягів трафіку і виявлення аномалій у поведінці. Інтеграція цих технологій з існуючими системами дасть змогу створити гнучкіші моделі аналізу, що навчатимуться на нових даних і покращуватимуть свою ефективність з часом.

Окрему увагу варто приділити інтеграції з хмарними рішеннями для безпеки, зокрема хмарними фаєрволами та рішеннями для захисту даних (CASB). У зв'язку з ростом використання хмарних технологій, забезпечення безпеки в таких середовищах стає все більш актуальним. Інтелектуальні системи аналізу трафіку повинні бути здатні працювати не лише у фізичних мережах, але й у хмарних інфраструктурах, виявляючи загрози, специфічні для таких середовищ. Інтеграція з хмарними системами безпеки забезпечить

комплексний захист для організацій, які використовують гібридні або повністю хмарні рішення для зберігання та обробки даних.

Крім того, варто розглянути можливість впровадження засобів аналізу зашифрованого трафіку. Сучасні кіберзагрози часто використовують шифрування для маскування своєї активності, що ускладнює їх виявлення за допомогою традиційних методів аналізу трафіку. Розробка технологій для аналізу зашифрованих даних або застосування розширених методів розшифрування трафіку (без порушення конфіденційності користувачів) допоможе виявляти приховані загрози, що використовують SSL/TLS-з'єднання.

Ще одним важливим аспектом є створення єдиного центру моніторингу та реагування (SOC), який інтегрував би різні рішення безпеки, включаючи інтелектуальні системи аналізу трафіку, системи виявлення та запобігання вторгнень (IDS/IPS), фаєрволи, антивірусні рішення та SIEM. Така інтеграція дозволить створити централізовану платформу для моніторингу подій безпеки, що зменшить час на виявлення і реагування на інциденти, а також покращить координацію між різними компонентами системи захисту.

Окрім цього, рекомендується розширювати функціональність системи через модулі та плагіни. Це дозволить системі швидко адаптуватися до нових загроз і вимог безпеки. Плагіни можуть включати підтримку нових протоколів, інтеграцію з додатковими рішеннями безпеки, а також додавання специфічних функцій для аналізу окремих типів трафіку (наприклад, VoIP, відео-трафіку тощо). Такий підхід дозволить системі залишатися актуальною в умовах швидких змін у світі кібербезпеки.

Для успішного розвитку системи важливо також забезпечити регулярне оновлення сигнатур і баз даних загроз. Це дозволить системі виявляти нові загрози, які постійно з'являються в кіберпросторі. Крім того, розробка відкритих API для інтеграції з базами даних загроз від сторонніх постачальників дозволить отримувати найсвіжішу інформацію про нові атаки та вразливості.

З огляду на все вищесказане, подальший розвиток інтелектуальних систем аналізу трафіку має зосереджуватися на інтеграції з іншими рішеннями безпеки, автоматизації процесів реагування на загрози, використанні штучного інтелекту для адаптації до нових видів атак і забезпеченні захисту в хмарних середовищах. Це дозволить створити ефективну, гнучку та надійну систему захисту мереж від сучасних кіберзагроз.

3.4.Висновки до розділу 3

У розділі 3 було розглянуто ключові аспекти використання інтелектуальних систем аналізу мережевого трафіку, що спрямовані на підвищення їх ефективності та адаптивності до сучасних викликів кібербезпеки. Оптимізація параметрів системи дозволяє значно покращити точність аналізу, мінімізуючи хибні спрацювання і збільшуючи продуктивність. Використання системи в різних типах мереж потребує адаптації до специфіки кожного середовища – від локальних та глобальних мереж до хмарних і мобільних інфраструктур. Крім того, подальший розвиток систем передбачає їх інтеграцію з іншими рішеннями безпеки, впровадження автоматичних механізмів реагування, використання штучного інтелекту та машинного навчання для аналізу загроз, а також забезпечення безпеки в хмарних середовищах. Усі ці заходи спрямовані на створення комплексної та ефективної системи захисту мережевої інфраструктури, яка здатна швидко адаптуватися до нових загроз і забезпечувати надійну безпеку

ВИСНОВКИ

У процесі дослідження теми було всебічно розглянуто інтелектуальні системи аналізу мережевого трафіку, їх особливості та перспективи подальшого розвитку. Теоретичні основи аналізу мережевого трафіку, включаючи поняття, методи та задачі, дозволили глибше зрозуміти ключові механізми та важливість цих систем у сучасному кіберсередовищі. Окрему увагу було приділено ролі інтелектуальних систем, які використовують алгоритми машинного навчання та штучного інтелекту для підвищення ефективності виявлення загроз і аналізу трафіку.

Практична частина роботи продемонструвала реальні підходи до моделювання трафіку та моніторингу мереж, а також можливості функціонально-вартісного аналізу для оцінки економічної доцільності впровадження таких систем у різних середовищах. Це підтверджує важливість комплексного підходу до вибору та налаштування інструментів для аналізу мережевого трафіку.

У третьому розділі були розроблені рекомендації щодо подальшого розвитку інтелектуальних систем та їх інтеграції з іншими рішеннями безпеки. Важливими аспектами є оптимізація параметрів систем, адаптація до різних типів мереж, а також впровадження нових технологій, таких як штучний інтелект і машинне навчання, для підвищення точності виявлення загроз.

Таким чином, проведене дослідження демонструє, що інтелектуальні системи аналізу мережевого трафіку є важливими елементами сучасної кібербезпеки. Їх подальший розвиток та інтеграція з іншими засобами захисту дозволять забезпечити більш ефективну боротьбу з новими кіберзагрозами та покращити безпеку мережевих інфраструктур різного рівня.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пакетні сніфери. URL: https://www.inter-nauka.com/uploads/public/15895669733068.pdf (дата звернення 13.09.2024)

2. Mike Cloppert. An Overview Of Protocol Reverse-Engineering. URL: https://digitalforensics.sans.org/blog/2012/07/03/an-overview-of-protocol- reverse-engineering (дата звернення 13.09.2024)

3. Що таке сніфер? URL: https://www.avg.com/en/signal/what-is-sniffer (дата звернення: 13.09.2024)

4. Stephen Northcat, Judy Nowak, Network Security Discovery. 3rd ed. New York: Sams Publishing, 2003. 356 p.

5. Orebaugh Angela ,Wireshark network protocol analyzer, 2006. 450 p.

6. The Bro Network Security Monitor. URL: http://www.bro.org/ (дата звернення 13.09.2024)

7. Tcpdump. URL: http://www.tcpdump.org/ (дата звернення 13.09.2024

8. Wireshark Display Filter Reference. URL: https://www.wireshark.org/

9. docs/dfref/, (дата звернення 13.09.2024)

10. Мережеві моделі OSI и TCP/IP. URL: http://www.quizful.net/post/osi_tcpip_layers (дата звернення 13.09.2024)

11. Colasoft Packet Player. URL: http://www.colasoft.com/packet_player/ (дата звернення 13.09.2024)

12. Microsoft SMB Protocol and CIFS Protocol Overview, URL: https://msdn.microsoft.com/en-us/library/aa365233(VS.85).aspx, (дата звернення 13.09.2024)

13. Documentation to DCE/RPC, URL: http://www.dcerpc.org/documentation/, (дата звернення 13.09.2024)

14. Oink: a Collaboration of C/C++ Tools for Static Analysis and Source- to-Source Transformation. URL: http://daniel-wilkerson.appspot.com/oink/ index.html/, (дата звернення 13.09.2024)

15. J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-Means Clustering Algorithm. New York: Wiley, 1979. 108p.

16. Arjuna Sathiaseelan and Tomasz Radzik. Improving the performance of TCP in the case of packet reordering. France: IEEE. 2004. 73p.

17. Аналітика SAS для IoT // електрон. текст. дані URL: https://www.sas.com/ru_ua/software/analytics-iot.html (дата звернення: 13.09.2024)

18. Архітектура і технології IoT // електрон. текст. дані URL: https://learn.ztu.edu.ua/pluginfile.php/68838/mod_resource/content/2/%D0%9B1.pdf (дата звернення: 13.09.2024)

19. Чотири етапи та архітектури Інтернету речей // електрон. текст. дані URL: https://medium.com/datadriveninvestor/4-stages-of-iot-architectureexplained-in-simple-words-b2ea8b4f777f (дата звернення: 13.09.2024)

20. The advantages and disadvantages of Internet Of Things // електрон. текст. дані URL: https://e27.co/advantages-disadvantages-internet-things20160615/ (дата звернення: 13.09.2024) 7. Welcome on Star // електрон. текст. дані URL: https://www.onstar.com/ web/portal/termsconditions / (дата звернення: 13.09.2024)

21. IEEE 1451 Smart Transducer Interface Standards // електрон. текст. дані URL: www.nist.gov/el/isd/ieee/IEEE 1451.cfm (дата звернення: 71 13.09.2024)

22. Filjar. R. ECall: Automatic notification of a road traffic accident / K.Vidovic, P.Britvic, M. Rimac // MIPRO. 2011. C. 600-605.

23. Using GPS / TS Wey, MH Lin, NT Hu A// Display technology. 2011. C. 45-53.