МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

МИХАЙЛИШИН Дмитро Андрійович

Система моніторингу безпеки кінцевих пристроїв/ Host security monitoring system

Спеціальність 125 – Кібербезпека
Освітньо-професійна програма – Кібербезпека
Кваліфікаційна робота

Виконав студентка групи КБзм -21
Д. А.Михайлишин

Науковий керівник
к.т.н., доцент Н.Г. Яцків

Кваліфікаційну роботу
допущено до захисту:

«____» _____ 2022 р.

Завідувач кафедри
_____ **В.В.Яцків**

ТЕРНОПІЛЬ - 2022

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 - Кібербезпека
освітньо-професійна програма –Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ В.В.Яцків
«____» _____ 2021 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

МИХАЙЛИШИН Дмитро Андрійович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Система моніторингу безпеки кінцевих пристроїв / Host security monitoring system
керівник роботи к.т.н., доцент Н.Г. Яцків

затверджені наказом по університету від 31 грудня 2021 року № 606

2. Строк подання студентом закінченої випускної кваліфікаційної роботи 16 листопада 2022 року.

3. Вихідні дані до кваліфікаційної роботи: завдання на випускну кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
- провести аналіз існуючих систем виявлення вторгнень;
- порівняти види систем виявлення вторгнень;
- провести аналіз методів виявлення вторгнень;
- розгорнути систему безпеки кінцевих пристроїв.

5. Перелік графічного матеріалу у роботі.
Загальна схема систем виявлення вторгнень
Принцип роботи сигнатурних IDS
Інтерфейси програм HIDS та NIDS.
Схеми роботи систем HIDS.

6. Консультанти розділів кваліфікаційної роботи

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11 жовтня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз систем безпеки кінцевих пристроїв	12.2021 р. – 03.2022 р.	
2	Методи виявлення вторгнень	03.2022 р. – 05.2022 р.	
3	Розгортання та дослідження системи безпеки	05.2022 р. – 11.2022 р.	

Студент _____ Михайлишин Д.А.
(підпис)

Керівник роботи _____ к.т.н., доцент Н.Г. Яцків
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Система моніторингу безпеки кінцевих пристроїв» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека» освітньо-професійної програми «Кібербезпека» написана обсягом 76 сторінки і містить 19 ілюстрацій, 2 додатки та 23 джерел за переліком посилань.

Метою кваліфікаційної роботи є дослідження систем моніторингу безпеки кінцевих пристроїв.

Методи досліджень. Для розв’язання поставлених задач у кваліфікаційній роботі використано: методи аналізу загроз, методи виявлення загроз на основі сигнатур, методи виявлення загроз на основі аномалій.

Результати дослідження: розгорнуто та досліджено систему моніторингу безпеки кінцевих пристроїв.

Результати роботи можуть успішно застосовуватися при реалізації систем моніторингу безпеки кінцевих пристроїв на основі HIDS та NIDS.

Ключові слова: СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ХОСТІ, МЕРЕЖЕВІ СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ, ВИЯВЛЕННЯ НА ОСНОВІ СИГНАТУР, ВИЯВЛЕННЯ НА ОСНОВІ АНОМАЛІЙ.

АНОТАЦІЯ

The qualification work on the topic "Host security monitoring system" for obtaining the Master's degree in the specialty 125 "Cybersecurity" of the educational and professional program "Cybersecurity" is written in the volume of 76 pages and contains 19 illustrations, 2 appendices and 23 sources according to the list of references. The purpose of the qualification work is to study security monitoring systems of end devices.

Research methods. In order to solve the set tasks, threat analysis methods , methods of detecting threats based on signatures, methods of detecting threats based on anomalies were used in the qualification work.

Research results: the security monitoring system of end devices has been deployed and investigated.

The results of the work can be successfully applied in the implementation of end device security monitoring systems based on HIDS and NIDS.

Keywords: HOST-BASED INTRUSION DETECTION SYSTEM, NETWORK INTRUSION DETECTION SYSTEMS, SIGNATURE-BASED DETECTION, ANOMALITY-BASED DETECTION.

ЗМІСТ

ВСТУП

Актуальність роботи. Ландшафт загроз стає дедалі різноманітнішим, а системи, які використовуються для атак, стають більш досконалими, ніж будь-коли раніше. У 2018 році підприємства та організації будь-якого розміру та в усіх галузях зіткнулися з серйозними витоками даних (з витоком інформації зазнали Aadhar — 1,1 млрд користувачів; myFitnessPal — 150 млн; Quora — 100 млн; Facebook — 29 млн і багато інших). Однією з найбільших жертв став Marriot . Один несанкціонований доступ призвів до викрадення 500 мільйонів особистих даних. Відповідно до «Звіту про розслідування витоку даних за 2018 рік », понад 73% порушень було скоєно сторонніми особами [1].

Немає сумнівів що системи моніторингу безпеки кінцевих пристроїв є важливими для забезпечення безпеки сучасних організацій і мережевого трафіку. Ці засоби захисту використовуються для захисту обмеженого доступу до мережі організації. Що стосується систем виявлення вторгнень, існує два різних типи; хост-орієнтовані (HIDS) і мережево-орієнтовані системи (NIDS). Мережевий IDS аналізує мережевий трафік на наявність будь-яких вторгнень і видає сповіщення, тоді як HIDS відстежує поведінку хостів на наявність будь-якої підозрілої активності, перевіряючи події у мережі [2].

Відомо, що в комп’ютерній системі можуть відбуватися зловмисні чи аномальні дії, що робить наявність системи виявлення вторгнень на хості (HIDS) надзвичайно важливою.

Мета і завдання дослідження. Метою роботи є підвищення ефективності систем виявлення вторгнень.

Для досягнення даної мети ставились наступні завдання:

- провести аналіз існуючих систем виявлення вторгнень;
- порівняти види систем виявлення вторгнень;
- провести аналіз методів виявлення вторгнень;
- розгорнути систему безпеки кінцевих пристроїв.

Об’єкт дослідження – процеси виявлення вторгнень з використанням інструменту OSSEC.

Предмет дослідження – методи та алгоритми виявлення вторгнень.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: методи аналізу загроз, методи виявлення загроз на основі сигнатур, методи виявлення загроз на основі аномалій.

Наукова новизна одержаних результатів. Розроблено та протестовано спосіб моніторингу параметрів, які не включені за замовчуванням у OSSEC, зокрема: контроль використання дискового простору, середнє навантаження та виявлення використання USB накопичувача.

Практичне значення отриманих результатів. Встановлено та налаштовано систему безпеки кінцевих пристроїв з використанням інструменту OSSEC.

Публікації та апробація КР.

1. Михайлишин Д.А., Цаволик Т.Г., Драпак В.І. Система моніторингу безпеки кінцевих пристроїв. Матеріали наукової конференції «Автоматизація та комп’ютерно-інтегровані технології» (АКІТ - 2022), Тернопіль, 2022. – С. 107–109.

2. Михайлишин Д.А. Методи виявлення вторгнень в комп’ютерні мережі. Матеріали наукової конференції «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2022), Тернопіль, 2022. – С.68-69.

1 АНАЛІЗ СИСТЕМ БЕЗПЕКИ КІНЦЕВИХ ПРИСТРОЇВ

1.1 Принцип роботи системи виявлення вторгнень на хості

Система виявлення вторгнень на хості (HIDS) схожий на використання розумних камер безпеки у домі; якщо зловмисник увірветься у будинок, камера почне записувати та надішле сповіщення в реальному часі на мобільний пристрій. Виявлення вторгнень стало важливим методом захисту для мереж, щоб боротися зі слабкими місцями безпеки, властивими будь-якій системі у якій є людський елемент. Незалежно від того, наскільки сильною є ваша політика доступу користувачів, хакери завжди можуть їх обійти, за допомогою соціальної інженерії обманом змусивши співробітника розкрити облікові дані доступу [3].

Хакери з доступом можуть займати корпоративну систему роками, не будучи виявленими. Цей тип атаки називається розвинена стала загроза (Advanced Persistent Threat далі APT). IDS спеціально спрямовані на викорінення APT.

Моніторинг безпеки кінцевих пристроїв виконується за допомогою системи виявлень на основі хоста (Host-based intrusion detection system), далі будемо називати HIDS.

Розглянемо детальніше функції та призначення система HIDS. HIDS головним чином зосереджується на моніторингу та аналізі файлів журналу з метою виявлення аномалій і неавторизованих змін на основі попередньо визначених політик і набору правил [4]. На рисунку 1.1 приведена загальна схема системи виявлень на основі хоста.

HIDS не запобігає вторгненням або атакам, як це робить IPS(Intrusion Prevention System), різниця між ними в тому що IPS це система керування, а IDS система моніторингу.

Хоча HIDS можна встановити на точках мережі, таких як маршрутизатори або сервери, вони не можуть контролювати на рівні мережі.

З іншого боку, NIDS (виявлення вторгнення в мережу) можна встановити в точках перетину мережі та контролювати трафік.

HIDS не фільтрує вхідний/вихідний трафік на основі правил, як це робить брандмауер або монітор пропускної здатності.

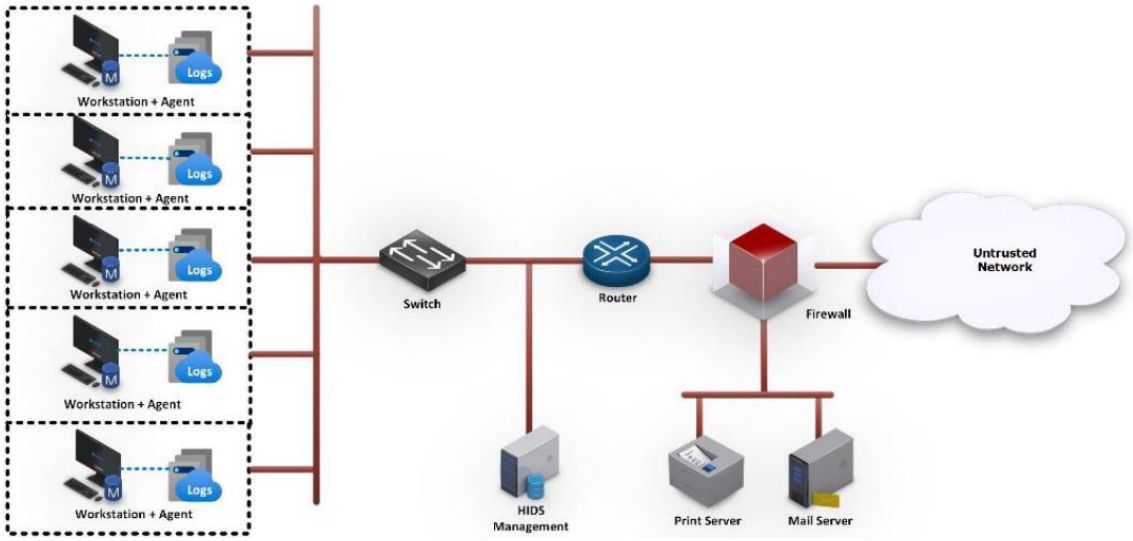

Рисунок 1.1 – Загальна схема HIDS

HIDS не призначена для зупинки атак, проте вона найкраща для виявлення всіх типів атак і делегування «профілактики» комусь іншому.

Можливості HIDS.

1. Виявлення аномалій.
2. Виявлення сигнатурних атак.
3. Атаки нульового дня.
4. Моніторинг трафіку на хості.
5. Контроль цілісності файлів.
6. Аналіз журналу.
7. Відповідність та аудит.
8. Система оповіщення та тривог.

Інструмент HIDS зосереджений на моніторингу журналів. Більшість програм створюють журнальні повідомлення, і збереження цих записів у файлах дає змогу шукати в них з часом і виявляти ознаки вторгнення. Однією з великих проблем зі збором кожного повідомлення журналу у вашій системі є те, що ви отримаєте великий обсяг даних .

Зберігання журнальних повідомлень систематичним чином допомагає визначити потрібний файл для отримання даних за програмою та датою. Отже, перший крок до отримання суттєвої інформації з нашої системи журналювання – це організувати імена файлів і структуру каталогів нашого сервера файлів журналів.

Наступним кроком у впровадженні HIDS є автоматичне виявлення. HIDS шукатиме в журналах певні події, які, схоже, могли зафіксувати зловмисну активність. Це ядро інструменту HIDS, а метод виявлення, який визначає, які записи отримати, встановлюється політиками та базою правил .

Спеціальна система виявлення вторгнень відстежує трафік на наявність зловмисних дій або порушень політики. Він виявляє відомі атаки за конкретними діями, які вони виконують (сигнатури). HIDS аналізує трафік (подібно до аналізатора мережевого трафіку) і поведінку, яка відповідає цим сигнатурам, у реальному часі на хості. Вона також може виявляти незвичайні моделі використання за допомогою методів виявлення аномалій. Коли HIDS знаходить збіг, вона запускає тривогу та сповіщає адміністратора [5, 6].

Багато HIDS дозволяють написати власні правила генерації сповіщень . Однак те, що ми дійсно шукаємо, при розробці системи безпеки, — це набір попередньо написаних правил, які включають досвід експертів із безпеки, які пишуть програмне забезпечення [7].

Система HIDS настільки хороша, наскільки хороші політики, які вона надає. Не можна очікувати, що ми будемо встигати за всіма останніми методами атак, водночас приділяючи час повсякденним завданням нашої роботи, і немає сенсу намагатись знати все, якщо ми можемо отримати рішення, надані нам інструментом HIDS [8].

1.2 Відмінності між HIDS та NIDS

HIDS перевіряє певну поведінку на основі хоста (на рівні кінцевої точки), зокрема, які програми використовуються, до яких файлів здійснюється доступ і яка інформація зберігається в журналах ядра.

Мережеві системи виявлення вторгнень (NIDS) є частиною ширшої категорії, тобто систем виявлення вторгнень. Іншим типом IDS є система виявлення вторгнень на основі хоста або HIDS. Системи виявлення вторгнень на основі хосту приблизно еквівалентні елементу керування інформацією безпеки SIEM [9].

Терміни SIEM і NIDS дуже похожі. SIEM означає Security Information and Event Management. Область SIEM — це комбінація двох уже існуючих категорій захисного програмного забезпечення. Керування інформацією про безпеку (SIM) і керування подіями безпеки (SEM).

У той час як мережеві системи виявлення вторгнень переглядають дані в реальному часі, системи виявлення вторгнень на основі хосту перевіряють файли журналу в системі. Перевага NIDS полягає в тому, що ці системи працюють негайно [10]. Переглядаючи мережевий трафік, вони можуть швидко вжити заходів. Однак багато дій зловмисників можна помітити лише за серією дій. Хакери навіть можуть розділяти шкідливі команди між пакетами даних щоб приховати їх. Оскільки NIDS працює на рівні пакетів, він менш здатний виявляти стратегії вторгнення, які поширюються між пакетами [11].

HIDS перевіряє дані подій після їх збереження в журналах(логах) . Запис записів у файли журналу створює затримки у відповідях. Однак ця стратегія дозволяє інструментам для аналізу виявляти дії, які відбуваються в кількох точках мережі одночасно. Наприклад, якщо той самий обліковий запис користувача використовується для входу в мережу з розосереджених географічних місць, а працівник, призначений для цього облікового запису, не працює в жодному з цих місць, тоді його обліковий запис було зламано [12].

Зловмисники знають, що файли журналів(файли логів) можуть викрити їхню діяльність, тому видалення записів журналу є захисною стратегією, яку використовують хакери. Таким чином, захист файлів журналів є важливим елементом системи HIDS.

І в NIDS, і в HIDS є свої переваги. Наприклад NIDS дає швидкі результати. Однак цим системам потрібно вчитися на звичайному трафіку мережі, щоб запобігти сповіщенню про «помилкові спрацьовування». Особливо на перших тижнях роботи в мережі інструменти NIDS мають тенденцію надмірно виявляти вторгнення та створювати потік попереджень, які виявляють регулярну активність. З одного боку, ви не хочете відфільтровувати попередження та ризикувати пропустити дії зловмисників. Однак, з іншого боку, надто чутливий NIDS може випробовувати терпіння команди адміністраторів мережі помилковими спрацьовуваннями.

HIDS дає повільнішу відповідь, але може дати точнішу картину діяльності зловмисників**,** оскільки вона може аналізувати записи подій із широкого спектру джерел реєстрації. NIDS перевіряє потік даних між комп'ютерами, часто відомий як мережевий трафік . Вони в основному відстежують мережу на предмет незвичної активності. У результаті NIDS може ідентифікувати зловмисника до того, як він зможе здійснити злам, тоді як HIDS діє як другий рівень захисту та вживає заходів на рівні кінцевої точки, якщо систему зламано.

Нам потрібно застосувати підхід SIEM і розгорнути як NIDS, так і HIDS, щоб надійно захистити свою мережу.

NIDS використовує два основні методи виявлення [13]:

1) виявлення аномалій;

2) виявлення на основі сигнатур.

Стратегії на основі сигнатур виникли на основі методів виявлення, які використовували антивірусні програми. Програма сканування шукає моделі використання мережевого трафіку, включаючи послідовності байтів і типи пакетів , які регулярно використовуються для атак [14].

Підхід на основі аномалій порівнює поточний мережевий трафік із нормальною активністю . Отже, ця стратегія захисту вимагає фази навчання, яка встановлює модель нормальної діяльності. Прикладом такого типу виявлення може бути кількість невдалих спроб входу. Можна очікувати, що користувач-людина кілька разів неправильно введе пароль, але запрограмована спроба вторгнення методом грубої сили(брутфорсу) використає багато комбінацій паролів, що змінюються швидко. Це дуже простий приклад. У реальних умовах моделі діяльності, які шукає підхід на основі аномалій, можуть бути дуже складними комбінаціями різних дій [15].

1.3 Аналіз HIDS з відкритим кодом

1.3.1 SolarWinds Security Event Manager

SolarWinds Security Event Manager (раніше відомий як Log & Event Manager) представлений постачальником як потужний і відзначений нагородами SEM. Це локально розгорнутий інструмент, який збирає, консолідує та аналізує журнали та події з брандмауерів, пристроїв і програм IDS/IPS, комутаторів, маршрутизаторів, серверів, журналів операційної системи та інших програм [16, 21].

Основою програми є виявлення загроз, автоматизований аналіз інцидентів і реагування на них, а також звітування про відповідність IT-інфраструктури.

Особливості SolarWinds (рисунок 1.2):

Інтегрована звітність про відповідність.

Автоматизоване усунення загроз.

Криміналістичний аналіз.

Контроль цілісності файлів.

USB моніторинг.

Пересилати необроблені дані журналу подій.

Програма може виявляти загрози на одному хості або в усій мережі.

SEM постачається з системою аналізу кіберзагроз для виявлення підозрілих дій і вжиття обґрунтованих дій.

SEM виходить за межі можливостей звичайного HIDS, і надає деякі практичні можливості.

Інструмент виконує потужний аналіз подій у реальному часі, сповіщає вас або може активно реагувати.

На рисунку 1.2 зображено інтерфейс SEM.

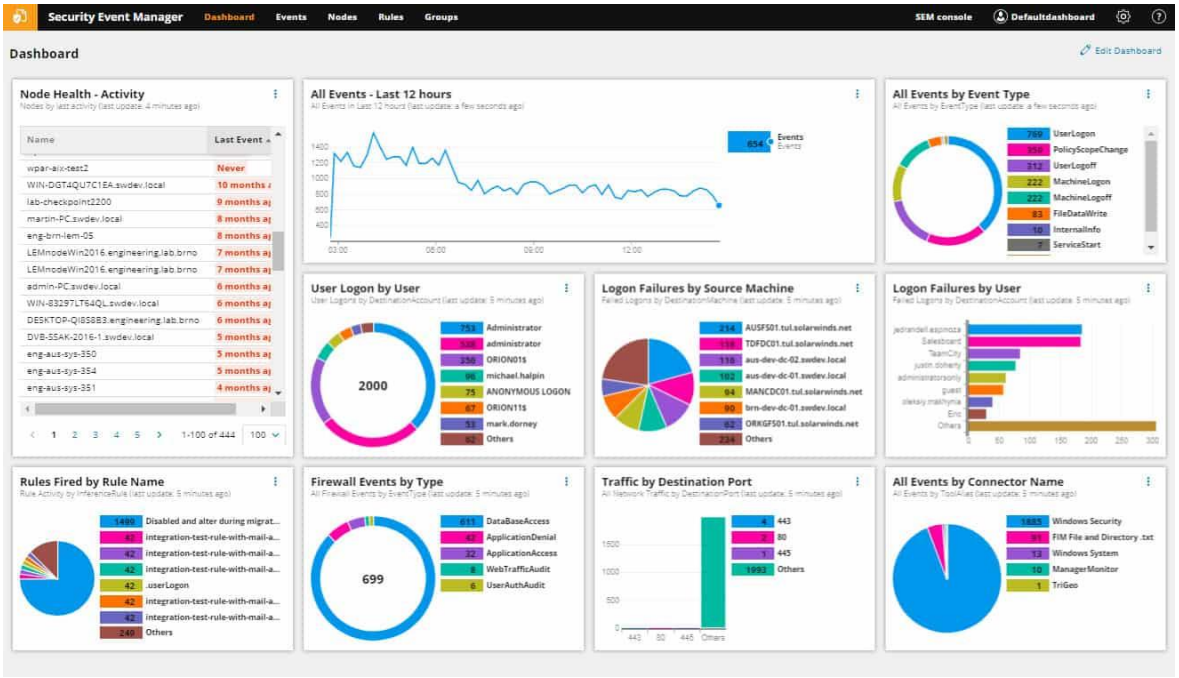

Рисунок 1.2 – Інтерфейс SEM

Він може блокувати IP-адреси, змінювати привілегії, блокувати USB, вбивати програми тощо.SEM може допомогти нам підтвердити відповідність наступним стандартам HIPAA, PCI DSS, SOX тощо.

Зразу доступно кілька готових шаблонів звітів, для того щоб зробити аудит і відповідність набагато доступнішими.

Перевагою є те що SIEM, орієнтована на підприємство, із широким спектром інтеграцій, в програмі просте фільтрування журналів, не потрібно вивчати спеціальну мову запитів. Десятки шаблонів дозволяють адміністраторам почати використовувати SEM з невеликим налаштуванням або кастомізацією.

Інструмент історичного аналізу допомагає знаходити аномальну поведінку та викиди в мережі. Основним недоліком є те що SEM створений для професіоналів, тому потрібен час щоб розібратись та вивчити платформу.

1.3.2 Papertrail

Інша програма від цього ж розробника SolarWinds називається Papertrail. Це агрегатор журналів, який централізує зберігання файлів журналу. Papertrail може керувати журналами подій Windows, повідомленнями Syslog, файлами журналів сервера Apache, повідомленнями програми Ruby on Rails, а також сповіщеннями маршрутизатора та брандмауера. Повідомлення можна переглядати в реальному часі на системній панелі, коли вони переміщуються до файлів журналу. Окрім керування файлами журналів, інструмент містить утиліти аналітичної підтримки [17]. На рисунку 1.3 зображено інтерфейс Papertrail.

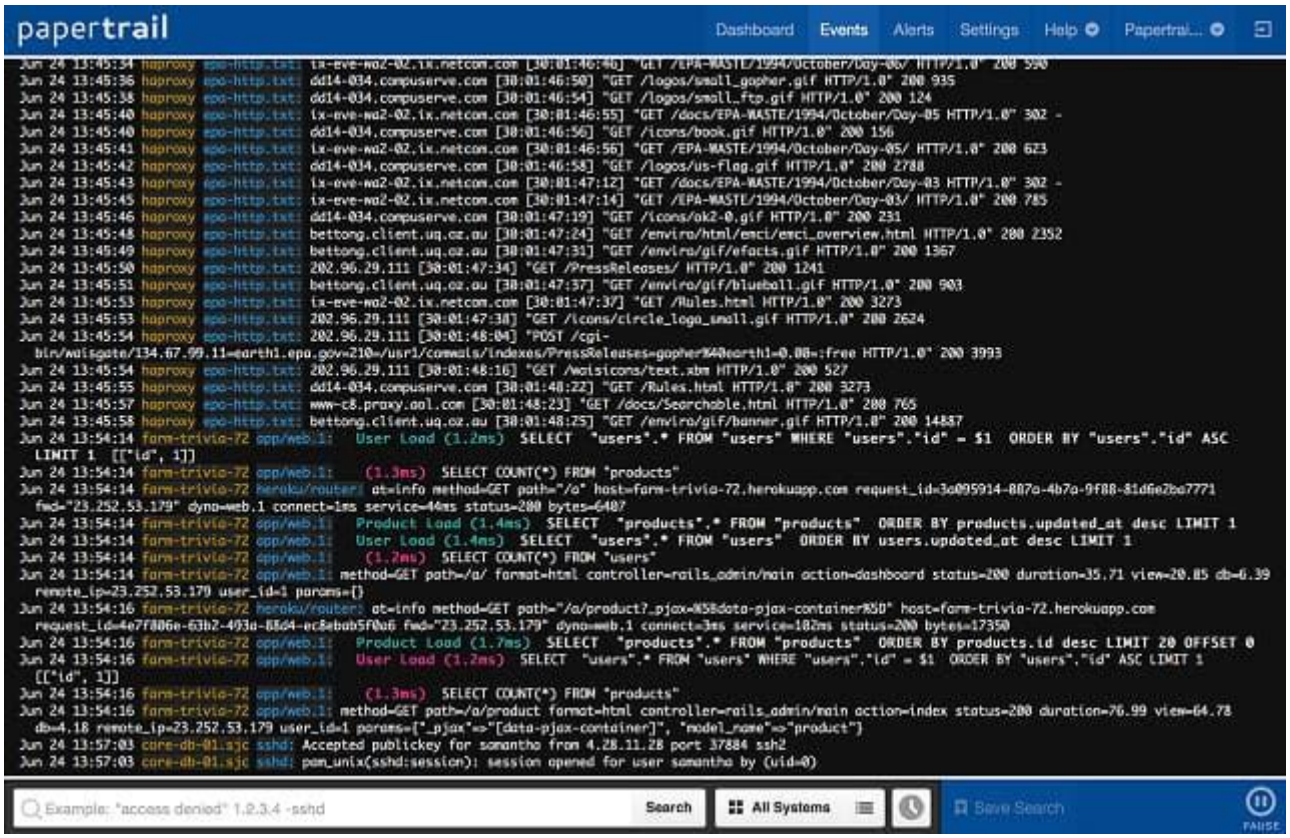

Рисунок 1.3 – Інтерфейс Papertrail

Особливості Papertrail:

– хмарний;

- лог сервер і консолідатор журналів;
- працює як на основі аномалії, так і на основі підпису;
- виконує резервне копіювання та архівування журналів.

Дані журналу шифруються як під час передачі, так і під час спокою, а доступ до файлів журналу захищено автентифікацією. Ваші файли зберігаються на сервері Papertrail, а SolarWinds піклується про резервне копіювання та архівування, тож ви можете заощадити гроші на покупці, управлінні та обслуговуванні файлових серверів.

Papertrail використовує як методи виявлення аномалій, так і методи на основі сигнатур, і ви отримуєте переваги від оновлень політики, отриманих від загроз, спрямованих на інших клієнтів Papertrail. Ви також можете створити власні правила виявлення.

Перевагами Papertrail є:

хмарний сервіс який допомагає масштабувати збір журналів без інвестицій в нову інфраструктуру;

шифрування даних як під час передачі, так і в стані спокою;

резервне копіювання та архівування виконується автоматично та є частиною послуги;

papertrail використовує виявлення як на основі сигнатур так і на основі аномалій для максимально ретельного моніторингу;

є безкоштовна версія;

Недоліком можна вважати потребу витрачення часу для того щоб вивчити всі функції та параметри.

1.3.3 ManageEngine Event Log Analyzer

Аналізатор журналу подій ManageEngine є одночасно HIDS і NIDS . Модуль керування журналами збирає та зберігає повідомлення системних логів та SNMP [18]. Також зберігаються метадані про кожне повідомлення системних логів. Інтерфейс ManageEngine зображено на рисунку 1.4.

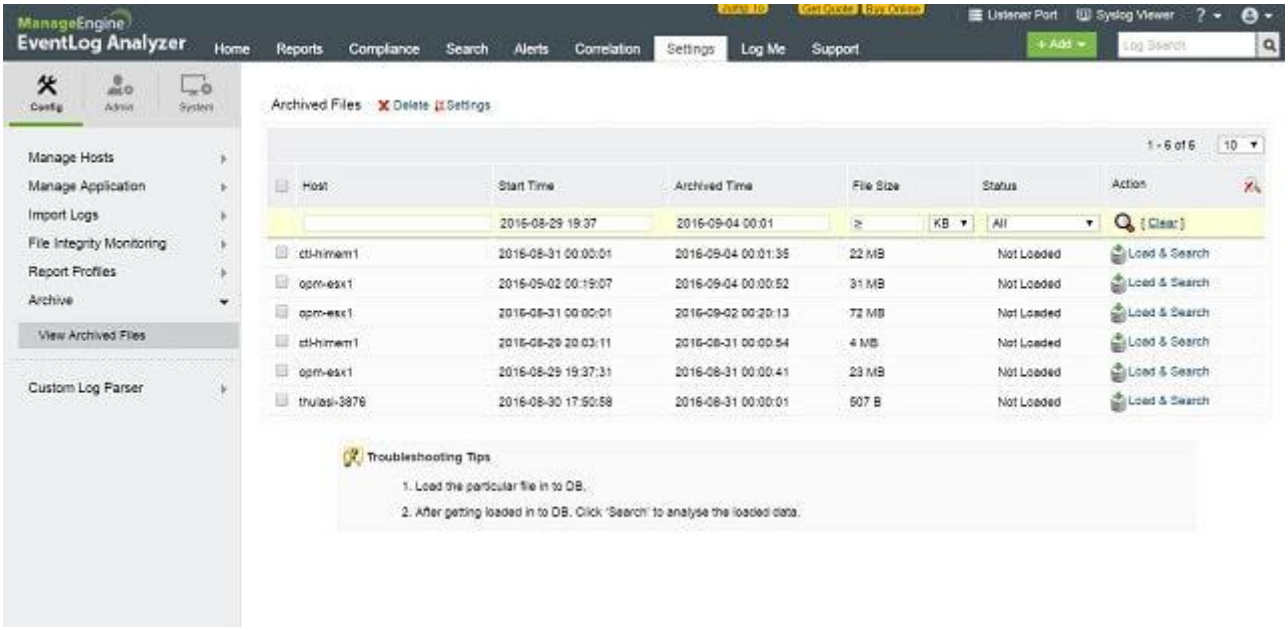

Рисунок 1.4 – інтерфейс ManageEngine

Особливості:

1. Локальна версія для систем Windows та Linux;

2. Збір та аналіз логів;

3. Аудит відповідності.

Логи захищені стисненням та шифруванням, а доступ захищено автентифікацією. Резервні копії можна відновити автоматично коли аналізатор виявляє підробку логів.

Інформаційну панель можна налаштувати, і різні екрани та функції можна призначити різним групам користувачів.

Звітність включає перевірки відповідності стандартам PCI DSS, FISMA та HIPAA, зокрема. Також можна активувати сповіщення про відповідність системи.

Перевагами ManageEngine є:

− настроювані інформаційні панелі, які чудово працюють для мережевих операційних центрів;

− кілька каналів сповіщень забезпечують сповіщення команд через SMS, електронну пошту чи інтегровану програму;

− mroanageEngine використовує виявлення аномалій, щоб допомогти технікам у їхній повсякденній роботі;

− моніторинг цілісності файлів, який може діяти як система раннього попередження про програми-вимагачі, крадіжки даних і проблеми з доступом.

− функції перевірки журналів дають змогу адміністраторам створювати звіти для судових справ або розслідувань.

З недоліків є функція пошуку, яку можна було б покращити зокрема використання операторних функцій таких як символи підстановки тощо.

Аналізатор журналу подій працює на Windows або Linux і може інтегруватися з інструментами керування інфраструктурою ManageEngine. Доступна безкоштовна версія цього інструменту, що дозволяє використовувати лише до 5 джерел журналів. Також доступна для завантаження 30-денну безкоштовну пробну версію.

1.3.4 OSSEC

OSSEC – це безкоштовний HIDS з відкритим кодом, створений Trend Micro. Він також включає функції моніторингу системи, які зазвичай приписуються NIDS. Це дуже ефективний обробник логів даних, але він не має інтерфейсу користувача [3, 19]. Більшість користувачів розміщують плагіни Kibana або Graylog на інтерфейс OSSEC. На рисунку 1.5 зображено інтерфейс OSSEC.

Особливості:

1) програма безкоштовна;
2) доступні функції NIDS;
3) додавання власного інтерфейсу.

Цей інструмент організує зберігання журналу логів та захистить файли від підробки. Виявлення вторгнень базується на відхиленнях і реалізується через « політики ». Ці набори правил можна отримати безкоштовно від спільноти користувачів.

Програмне забезпечення OSSEC можна встановити на Windows , Linux , Unix або Mac OS . Воно відстежує журнали подій Windows, а також реєстр. Також охороняє обліковий запис root на Linux , Unix і Mac OS . Підтримка надається безкоштовно від спільноти активних користувачів або можна придбати у Trend

Micro професійний пакет підтримки. На рисунку 1.6 зображено загальну схему роботи OSSEC.

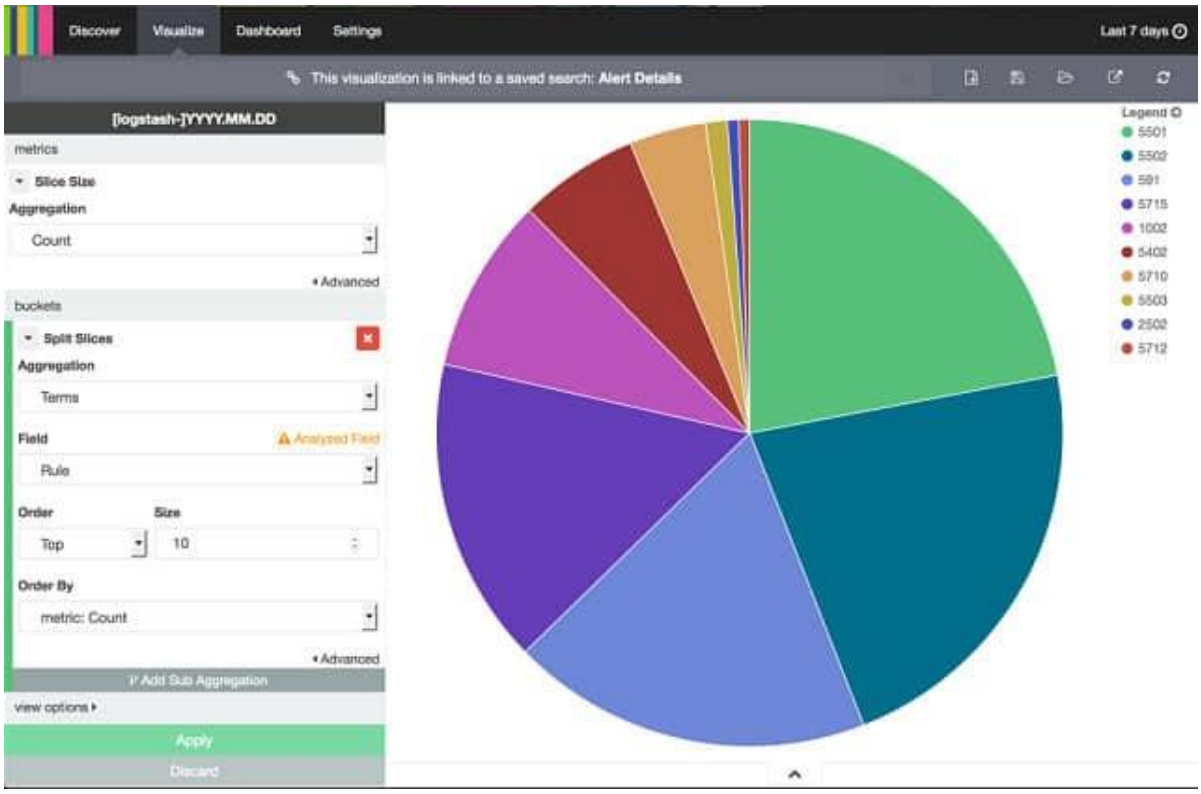

Рисунок 1.5 — інтерфейс OSSEC

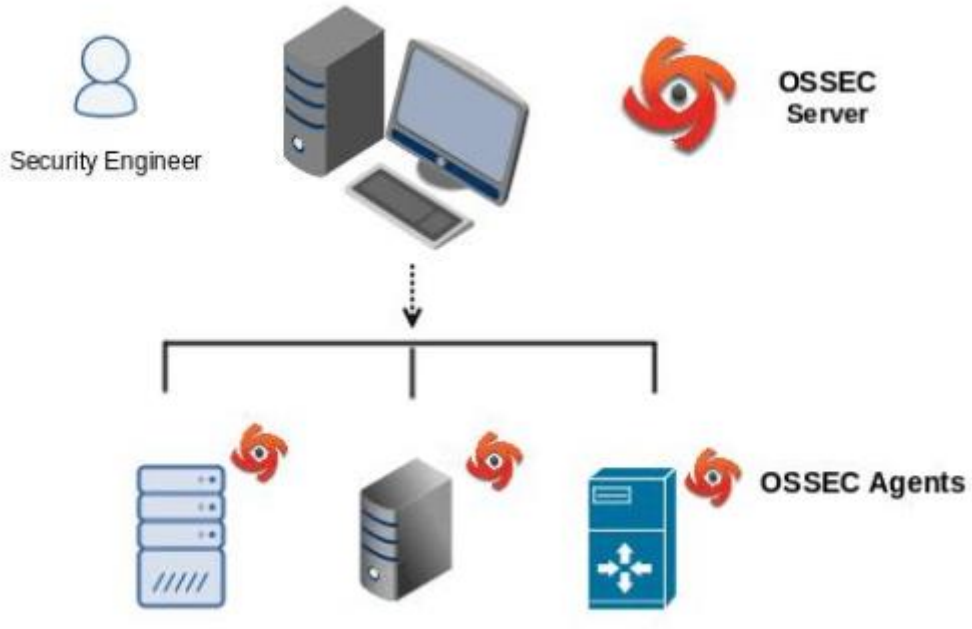

Рисунок 1.6 – Загальна схема OSSEC

Переваги:

Може використовуватися в широкому діапазоні операційних систем, Linux, Windows, Unix і Mac;

Може працювати як комбінація SIEM і HIDS;

Легко кастомізувати;

Створені спільнотою шаблони дозволяють адміністраторам швидко розпочати роботу.

Недоліки:

Для подальшого аналізу потрібні додаткові інструменти, такі як Graylog і Kibana;

У версії з відкритим вихідним кодом немає платної підтримки.

1.3.5 Sagan

Sagan – це безкоштовна система HIDS , яка доступна для систем Unix , Linux і Mac OS . Вона здатна збирати повідомлення журналу подій Windows , навіть якщо він не працює в Windows [4, 20]. Ви можете розподілити обробку Sagan, щоб зменшити навантаження на ЦП вашого сервера логів. Система використовує методи виявлення аномалій і сигнатур.

Особливості:

Безкоштовне використання;

Виявлення на основі аномалій та сигнатур;

Правила автоматичної відповіді.

Ви можете налаштувати автоматичні дії при виявленні вторгнення. Інструмент має кілька унікальних функцій, яких не вистачає в деяких більш популярних HIDS. До них відноситься функція геолокації IP, яка дозволить нам надсилати сповіщення, коли діяльність різних IP-адрес відстежується до одного географічного джерела. Інструмент також дозволяє встановлювати пов’язані з часом правила для ініціювання сповіщень . Система була розроблена для сумісності з Snort , яка є мережевою системою виявлення, що надає можливості

Saga NIDS у поєднанні з мережевим збирачем даних. Sagan включає засіб виконання скриптів, який робить його IPS.

Переваги:

– безкоштовний інструмент аналізу логів;

– сумісний з іншими інструментами з відкритим кодом, такими як Zeek і Snort;

– містить локатор IP-адрес, який може надати геополітичну інформацію про адреси.

Недоліки:

– редоступна для Windows;

– це більше інструмент HIDS, ніж традиційний IDS;

– складна в освоєнні для нових користувачів.

У пізніших випусках додано файл журналу автентифікації для систем на базі debian і нестандартну мітку часу Sophos UTM для analysisd (3.6.0). На рисунку 1.7 зображено інтерфейс Sagan.

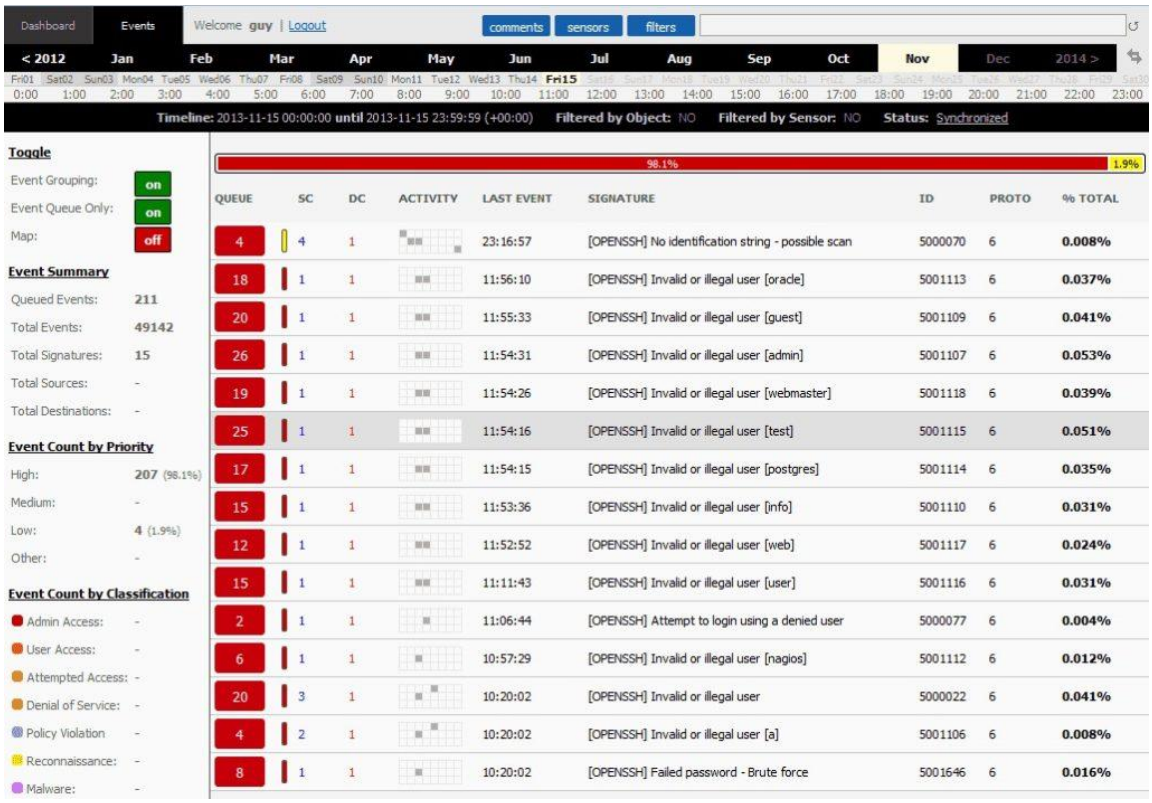

Рисунок 1.7 – інтерфейс Sagan

1.3.6 Splunk

Splunk пропонує функції як HIDS, так і NIDS. Базовий пакет цього інструменту є безкоштовним для використання , і він не містить жодних мережевих сповіщень про дані, тому це чистий HIDS. Якщо потрібен HIDS на основі аномалій , то базовий пакет дуже хороший варіант[4]. Найкраще видання Splunk називається Splunk Enterprise , а також є версія Software-as-a-Service (SaaS), яка називається Splunk Cloud . Між безкоштовною версією та версією Enterprise знаходиться Splunk Light, яка має певні обмеження щодо обслуговування. Існує також онлайн-версія Splunk Light під назвою Splunk Light Cloud . Інтерфейс Splunk зображено на рисунку 1.8.

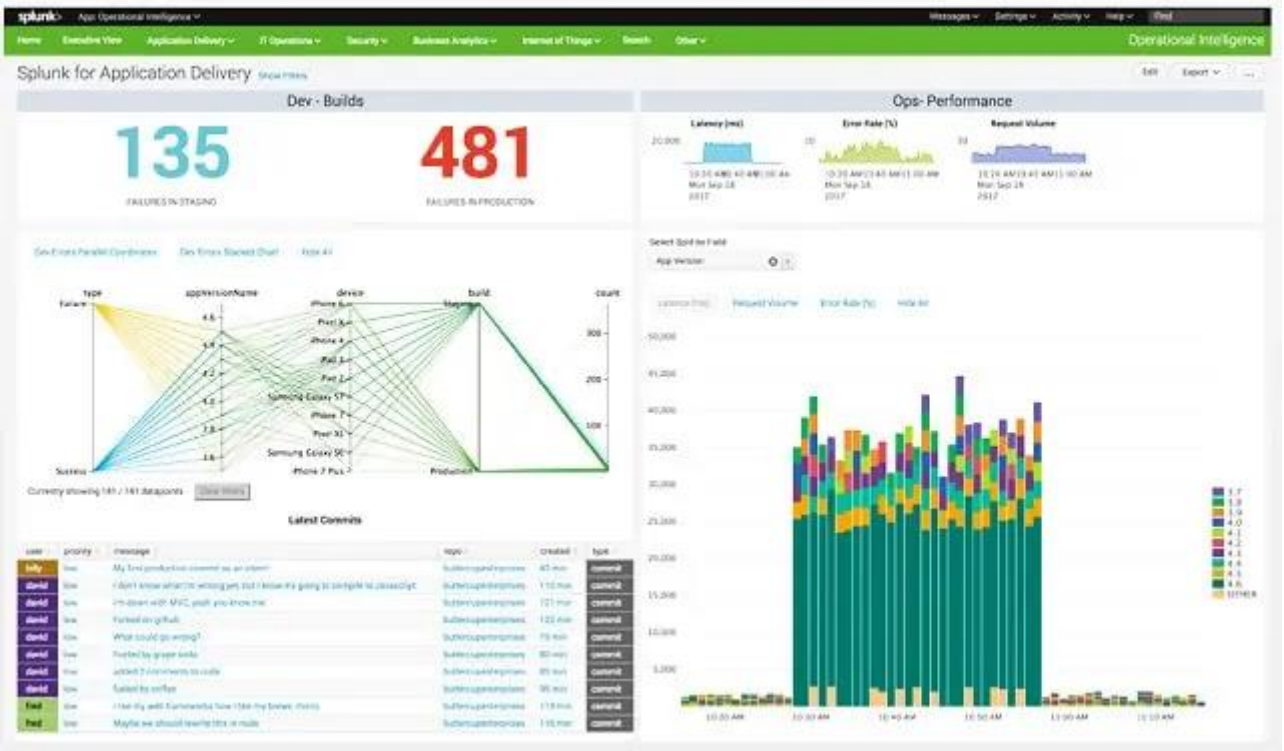

Рисунок 1.8 – Інтерфейс Splunk

Особливості:

- інструмент аналізу даних;
- SIEM доповнення;
- локальна та хмарна версія.

Splunk має функції автоматизації робочого процесу, які роблять його системою запобігання вторгненням. Цей модуль називається Adaptive Operations Framework і він пов’язує автоматизовані сценарії для ініціювання сповіщень.

Автоматизація вирішення виявлених проблем доступна лише з преміум версією Splunk.

Інформаційна панель Splunk дуже приваблива у візуалізації даних, таких як лінійні графіки та кругові діаграми. Система містить аналізатор даних у всіх версіях Splunk. Це дає змогу переглядати записи, узагальнювати, сортувати та здійснювати пошук у них, а також відображати їх у вигляді графіків.

Переваги:

– може використовувати аналіз поведінки для виявлення загроз, які не виявляються через журнали;

– чудовий користувальницький інтерфейс – візуально красивий та з легкими параметрами налаштування;

– легке визначення пріоритетів подій;

– орієнтований на підприємство;

– доступний для Linux і Windows.

Недоліки:

– щоб дізнатися ціну, потрібно зв’язатися з відділом продажів;

– більше підходить для великих підприємств, задорогий для малого бізнесу;

– використовує спеціальну мову обробки пошуку (SPL) для запитів, що ускладнює навчання.

Усі версії Splunk працюють у Windows, Linux і macOS. На даний момент доступні: 30-денна безкоштовну пробна версія Splunk Light, 60-денна безкоштовна пробна версія Splunk Enterprise і 15-денна безкоштовна пробна версія Splunk Cloud.

1.4 Обгрунтування вибіру рішення NIDS

1.4.1 CrowdStrike Falcon Intelligence

Хоча NIDS зазвичай відстежує проходження мережевого трафіку, CrowdStrike Falcon Intelligence працює на кінцевих точках, перехоплюючи трафік, щойно він надходить на пристрій. Теоретично ця резиденція повинна зробити Falcon Intelligence системою виявлення вторгнень на основі хоста. Однак служба працює з даними в режимі реального часу, а не шляхом читання файлів логів, тому це NIDS [5, 22].

Особливості:

– збирає трафік з кінцевих точок;

– опція керованої служби безпеки;

– правила Yara та Snort.

Falcon Intelligence може адаптувати інші системи NIDS шляхом створення правил, які можна запускати в Yara та Snort. Ці правила генеруються в службі Falcon Intelligence, а потім перевіряються та коригуються вручну експертом з кібербезпеки в штаб-квартирі CrowdStrike. Інтерфейс CrowdStrike Falcon Intelligence зображено на рисунку 1.9.

Більшість процесів Falcon Intelligence автоматизовані. Однак найвищий тарифний план включає виділеного аналітика з кібербезпеки. Цей план називається Falcon Intelligence Elite. Проміжний план, який включає індивідуальне сканування Інтернету на предмет згадок про вашу компанію, називається Falcon Intelligence Premium. Базовий план відомий як Falcon Intelligence і включає автоматичний пошук даних про загрози на кожній кінцевій точці мережі.

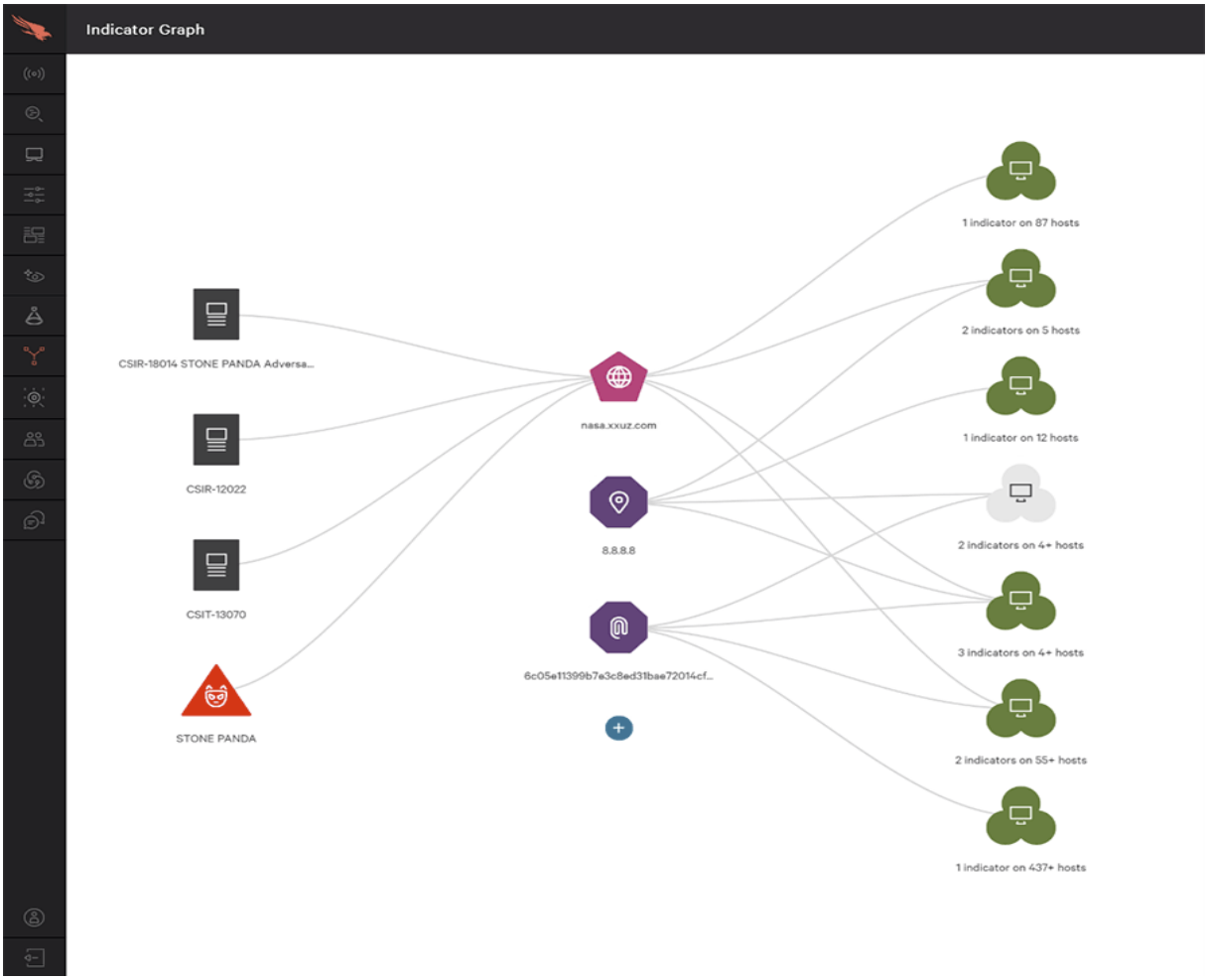

Рисунок 1.9 – Інтерфейс CrowdStrike Falcon Intelligence

Переваги:

– не покладається лише на логи, а використовує сканування процесів щоб негайно знайти загрози;

– може і HIDS і NIDS одночасно;

– відстежує та сповіщає про аномальну поведінку, чим довше він стежить за мережею тим точніші сповіщення;

– можна встановити локально, або в хмарі;

– не сповільнює роботу сервера чи пристрої кінцевих користувачів;

До недоліків можна назвати лише маленький випробувальний період (15 днів).

1.4.2 Security Onion

Дуже хороший варіант для Linux. Security Onion це проект з відкритим кодом, який підтримується спільнотою.

Програмне забезпечення для виявлення вторгнень працює на Ubuntu та було взято з інших утиліт аналізу мережі. Наприклад в Security Onion інтегровано Snort, Zeek та Suricata.

Переваги:

– колекція інструментів;
– безкоштовний;
– HIDS та NIDS.

Функціональність підтримується компанією OSSEC, а інтерфейс це система Kibana. Добре відомі інструменти як EKSA, NetworkMiner, Snorby, Squert, Squil та Xplico також включені в Security Onion.

Утиліта містить широкий спектр інструментів аналізу та використовує методи виявлення сигнатур і аномалій. Незважаючи на те, що повторне використання існуючих інструментів означає, що Security Onion отримує переваги від усталеної репутації своїх компонентів, оновлення елементів у пакеті може бути складним [5]. Інтерфейс Security Onion зображено на 1.10.

Перевагами програми є безкоштовне програмне забезпечення з відкритим кодом, висока деталізація призначена для аналізу на рівні криміналістики, та вбудований аналізатор пакетів і параметри відтворення трафіку.

З недоліків можна згадати про те що програма доступна лише для Linux, та необхідність використання Kibana для візуалізації через незручний інтерфейс для користувача.

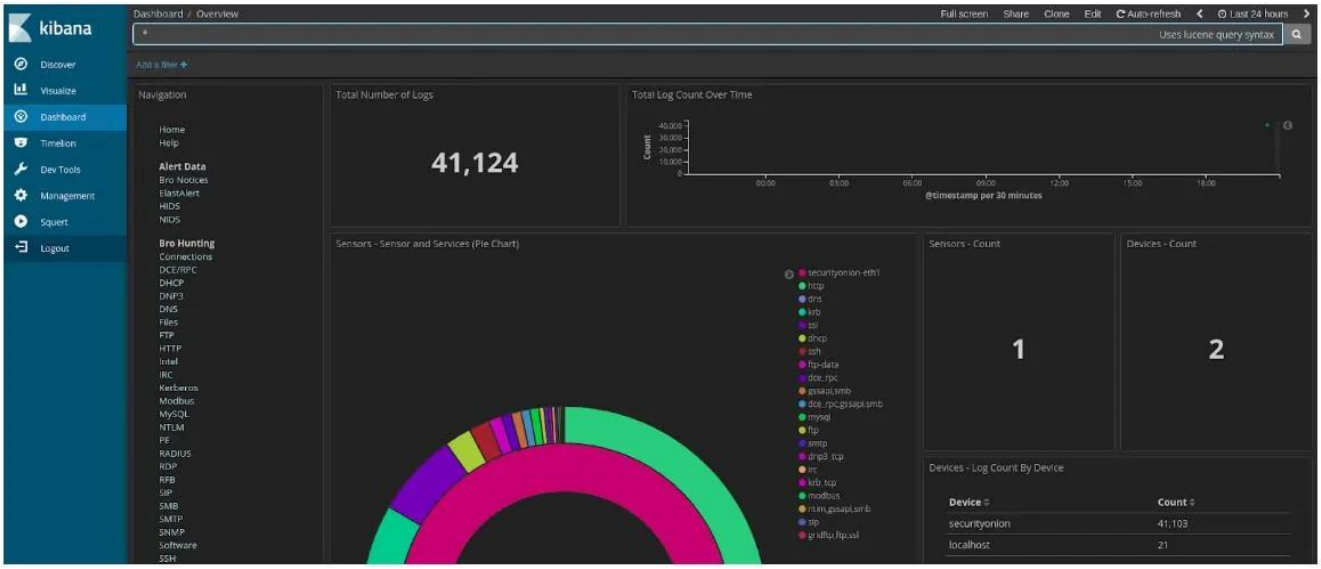

Рисунок 1.10 – Kibana з Security Onion

1.4.3 Snort

Snort це безкоштовний для використання проект з відкритим кодом що належить Cisco Systems. На сьогоднішній день є одним з найкращих NIDS.

Доступний на Linux, Unix та Windows.

Особливості:

– безкоштовна для використання;

– велика спільнота користувачів;

– сніффер пакетів.

Система аналізу пакетів збирає копії мережевого трафіку, та аналізує. Але можливості Snort не обмежуються лише цим, інструмент має інші режими, і один з них – виявлення вторгнень. В цьому режимі Snort використовує базові політики, які є основою правил виявлення вторгнень [5]. На рисунку 1.11 зображено інтерфейс Snort.

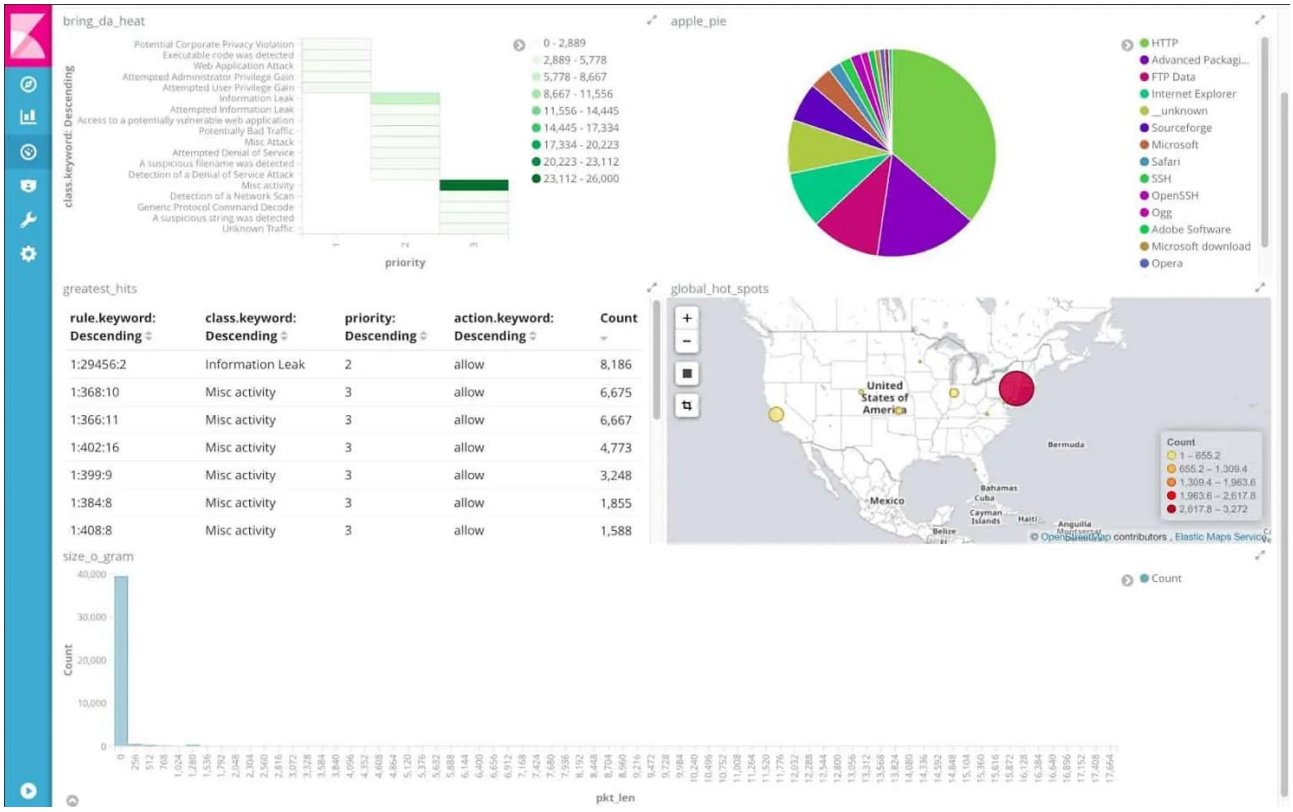

Рисунок 1.11 – Інтерфейс Snort

Базові політики роблять Snort гнучким та адаптивним. Потрібно точно налаштувати політики відповідно до звичних дій нашої мережі, для того щоб зменшити випадки помилкових спрацьовувань. Можна написати власні базові політики, але це не обов'язково, оскільки можна завантажити пакет з веб-сайту Snort. Також доступний форум користувачів Snort де можна поставити різні питання, та обмінятись інформацією.

З переваг є повністю безкоштовний і відкритий код, велика спільнота яка ділиться новими наборами правил і конфігураціями для розгортання адміністраторами у своїх система та підтримка аналізу пакетів який аналізує трафік та сканує логи.

З недоліків підтримка лише від спільноти користувачів, важкіша для освоєння та потрібно більше налаштувань щоб усунути помилкові спрацьовування

Досвідчені користувачі безкоштовно надають власні поради та уточнення на форумі іншим. Також можна безкоштовно отримати більше основних політик

від спільноти. Завдяки тому що Snort використовує дуже багато людей, на форумах завжди є нові базові політики та ідеї.

1.4.4 Suricata

Suricata це NIDS який працює на прикладному рівні, що забезпечує видимість кількох пакетів. Це безкоштовний інструмент, який має схожі можливості до Zeek. Хоча ці системи виявлення на основі сигнатур працюють на програмному рівні, вони все ще мають доступ до деталей пакетів, що дозволяє програмі обробки отримувати інформацію на рівні протоколу з заголовків пакетів [5]. Це включає шифрування даних, дані транспортного рівня, та інтернет рівня.

Ключовими рисами є [9]:

– аналіз прикладного рівня;

– безкоштовне використання;

– сумістність з Snort.

Цей IDS також використовує методи виявлення аномалій. Окрім пакетних даних, Suricata може перевіряти сертифікати TLS, HTTP-запити та транзакції DNS. Інструмент також може витягувати сегменти з файлів на бітовому рівні для виявлення вірусів. На рисунку 1.12 зображений інтерфейс Suricata.

Переваги:

– збирає дані на прикладних рівнях, надаючи їм унікальну видимість там, де такі продукти, як Snort, не бачать;

– дуже ефективно аналізує та збирає пакети протоколів;

– може контролювати кілька протоколів і перевіряти цілісність сертифікатів у TLS, HTTP та SSL;

Сумісний з іншими інструментами, які використовують формат правила VRT.

Недоліки:

– вбудовані сценарії можуть бути простішими у використанні;

– хоча Suricata безкоштовний, він не має такої великої спільноти, як інструменти, такі як Snort або Zeek;

– важка навігація на.домашній інформаційній панелі.

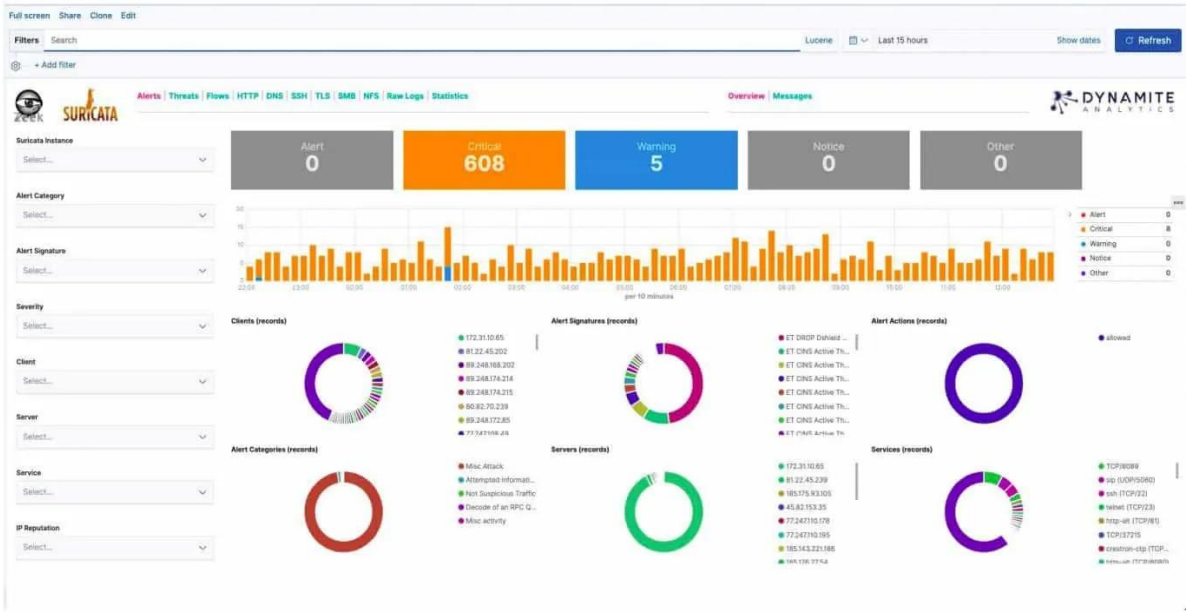

Рисунок 1.12 — Інтерфейс Suricata

Suricata є одним із багатьох інструментів, які сумісні з структурою даних Snort. Він здатний реалізувати базові політики Snort. Великою перевагою цієї сумісності є те, що спільнота Snort також може дати вам поради щодо прийомів, які можна використовувати з Suricata. Інші інструменти, сумісні із Snort, також можуть інтегруватися з Suricata. До них належать Snorby, Anaval, BASE та Squil.

1.4.5 Zeek

Великою перевагою Zeek від Snort є те що він працює на прикладному рівні. Цей NIDS широко використовують наукові та академічні спільноти.

Zeek це система на основі сигнатур, а також він використовує методи виявлення аномалій. Він здатний виявляти шаблони на рівні бітів, які вказують на зловмисну активність у мережевих пакетах [5].

Процес виявлення відбувається в два етапи. Першим з них керує Event Engine. Оскільки дані оцінюються на рівні вищому за пакетний, аналіз неможливо виконати миттєво. Має бути певний рівень буферизації, щоб достатню кількість пакетів можна було оцінити разом. Отже Zeek трохи

повільший за типові NIDS на рівні пакетів, але все одно виявляє зловмисну активність швидше, ніж HIDS. Зібрані дані оцінюються за допомогою сценаріїв політики, що є другим етапом процесу виявлення.

Переваги:

– NIDS з можливістю налаштування, розроблений спеціально для професіоналів у сфері безпеки;

– підтримує аналіз трафіку прикладного рівня, а також сканування на основі журналу;

– використовує виявлення сигнатур і сканування аномальної поведінки для виявлення відомих і невідомих загроз;

– підтримує автоматизацію за допомогою сценаріїв, що дозволяє адміністраторам легко створювати сценарії для різних дій.

Недоліки:

– доступно лише для Linux, Mac та Unix;

– не дуже зручний, потребує глибоких знань NIDS, SIEM тощо;

– краще підходить для спеціалістів та дослідників.

Можна налаштувати автоматичне виконання дій виправлення за допомогою сценарію політики. Це робить Bro системою запобігання вторгненням. Програмне забезпечення для виявлення вторгнень можна встановити на Linux, Unix і Mac OS.

Zeek можна розгорнути в поєднанні з Dynamite-NSM , безкоштовним монітором безпеки мережі, щоб розширити його можливості та скористатися перевагами розширеного графічного відображення даних журналу.

2 МЕТОДИ ВИЯВЛЕННЯ ВТОРГНЕНЬ

2.1 Методи виявлення на основі сигнатур

Виявлення на основі аналізу сигнатур було першим методом, застосованим для виявлення вторгнень. Сигнатурні системи виявлення вторгнень (Далі SIDS) засновані на методах зіставлення шаблонів для пошуку відомої атаки; вони також відомі як виявлення на основі знань або виявлення неправильного використання (Khraisat et al., 2018). Найвідоміші SIDS доступні на даний момент – Snort, Suricata, Netstat та Bro .У SIDS методи зіставлення використовуються для пошуку попереднього вторгнення. Іншими словами, коли сигнатура вторгнення збігається з сигнатурою попереднього вторгнення, яка вже існує в базі даних сигнатур, спрацьовує сигнал тривоги. Для SIDS журнали хоста перевіряються, щоб знайти послідовності команд або дій, які раніше були визначені як зловмисне програмне забезпечення. Відповідно, в цьому головний недолік IDS які працюють на основі виявлення на основі сигнатур, якщо по якійсь причині база недоступна, мережа стає вразливою. Також якщо нова атака нова і ще не внесена в базу, є ризик того що загроза не буде виявлена [16].

Традиційні підходи до SIDS перевіряють мережеві пакети та намагаються зіставити їх із базою даних сигнатур. Але ці методи не можуть ідентифікувати атаки, які охоплюють кілька пакетів. Оскільки сучасне зловмисне програмне забезпечення є більш складним, може знадобитися витягти інформацію про підпис із кількох пакетів. Для цього IDS має відкликати вміст попередніх пакетів. Що стосується створення підпису для SIDS, то загалом існувала низка методів, у яких підписи створювалися як кінцеві машини (Meiners та інш., 2010), шаблони формальних мовних рядків або семантичні умови (Lin та інш., 2011).

SIDS здатні відстежувати шаблони чи стани. Шаблони – це ті сигнатури, які зберігаються в базі, що постійно оновлюється. Стани – це будь-які дії всередині системи.

Початковий стан системи – нормальна робота, відсутність атак. Після успішної атаки система переходить у скомпрометований стан, тобто зараження

пройшло успішно. Кожна дія (Наприклад встановлення з’єднання за протоколом, що не відповідає політиці безпеки компанії, активізація ПЗ тощо) здатна змінити стан. Рисунок 2.1 демонструє принцип роботи підходів сигнатурних IDS. Основна ідея полягає в тому, щоб побудувати базу даних сигнатур вторгнення та порівняти поточний набір дій з існуючими сигнатурами та подати сигнал тривоги, якщо знайдено відповідність. Наприклад, правило у формі «if: antecedent -then: consequent» може призвести до «if (IP-адреса джерела=IP-адреса призначення), тоді позначити як атаку». Тому сигнатурні IDS відстежують не дії, а стан системи [22].

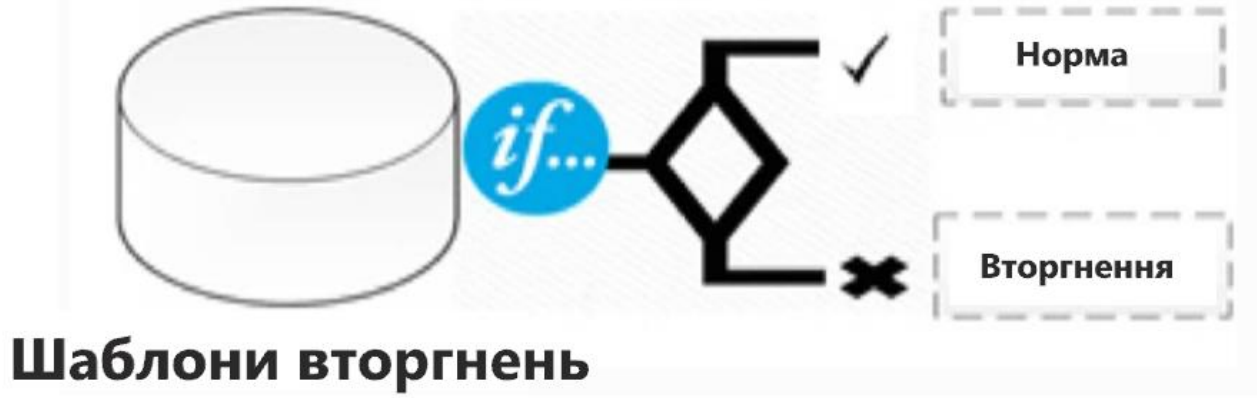

Рисунок 2.1 – Принцип роботи сигнатурних IDS

Зростаюча кількість атак нульового дня (Symantec, 2017) поступово зробила методи SIDS менш ефективними, оскільки не існує попереднього підпису для будь-яких таких атак. Поліморфні варіанти шкідливого програмного забезпечення та зростання кількості цілеспрямованих атак можуть ще більше підірвати адекватність цієї традиційної парадигми. Потенційним вирішенням цієї проблеми було б використання методів боротьби що використовують IDS які базуються на виявленні на основі аномалій(Далі AIDS), які працюють шляхом профілювання того, що є прийнятною поведінкою, а не того, що є аномальним, як описано в наступному розділі.

2.2 Виявлення на основі аномалій

Підхід на основі аномалій, з іншого боку, спрямований на виявлення нових (невідомих) атак шляхом моделювання дій, які вважаються нормальними в межах системи та виявлення потенційних атак від поведінки, яка відхиляється від відомої нормальної моделі поведінки, через швидкий розвиток шкідливих програм, основний підхід полягає у використанні машинного навчання. Метою машинного навчання в цьому контексті є класифікація події (наприклад, нормальний стан або напад/вторгнення). Перші події, зафіксовані з навколишнього середовища зберігаються в базі даних під час процесу. Набір функцій витягується та зберігається у наборі даних для кожної події в базі даних. Потім набір даних використовується в алгоритмі машинного навчання, щоб вивести шаблон і створити модель, яка представляє таку поведінку щоб створити модель правдоподібної діяльності, з якою потім порівнюють нові поведінки. Для правильної роботи таких систем виявлення загроз, потрібний тестовий період навчання. Адміністраторам рекомендується протягом кількох місяців повністю вимкнути сигнали тривоги, щоб система навчалась. Розглянемо види навчань системи AIDS. Існує 3 основних напрямки навчань системи AIDS. Ці три напрямки, а також їх підкласи показані на рисунку 2.2.

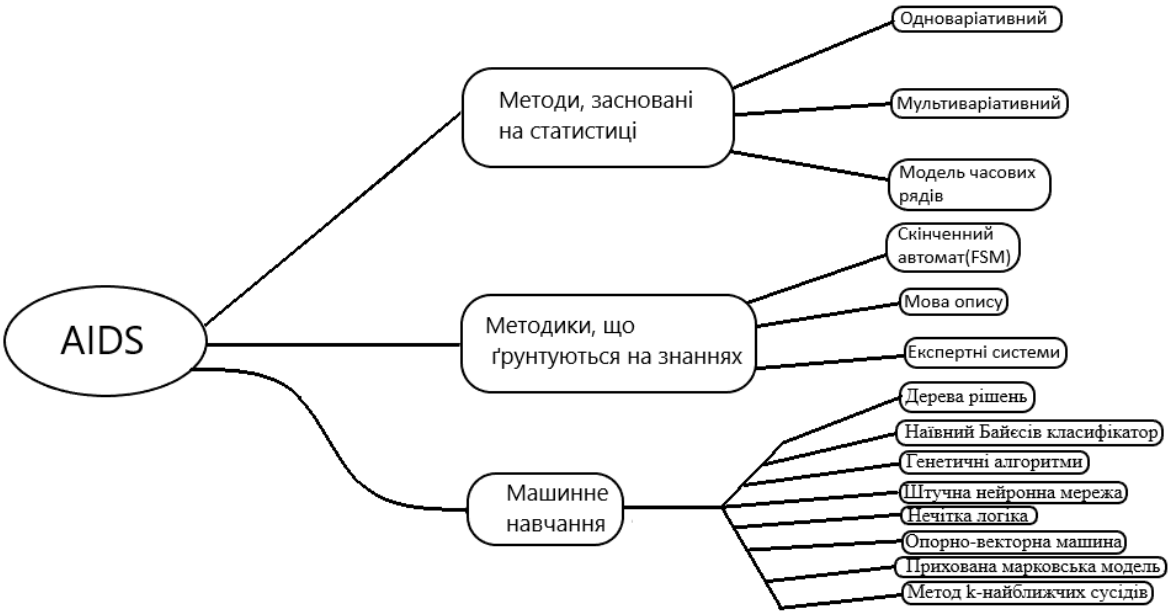

Рисунок 2.2 – Напрямки навчань системи AIDS

2.2.1 Методи на основі статистики

IDS на основі статистики будує модель розподілу для нормального профілю поведінки, потім виявляє події з низькою ймовірністю та позначає їх як потенційні вторгнення. Статистичні AIDS по суті враховують статистичні показники, такі як медіана, середнє значення, мода та стандартне відхилення пакетів. Іншими словами, замість того, щоб перевіряти трафік даних, відстежується кожен пакет, що означає відбиток потоку. Статистичний AIDS використовується для виявлення будь-яких типів відмінностей у поточній поведінці від нормальної поведінки. Статистичні IDS зазвичай використовують одну з наступних моделей.

Одноваріативний (Univariate): «Uni» означає «one», тобто дані мають лише одну змінну. Цей метод використовується, коли статистичний нормальний профіль створюється лише для одного показника поведінки в комп'ютерних системах. Однофакторний IDS шукає аномалії в кожному окремому показнику. Мультиваріативний (Multivariate): базується на зв'язках між двома чи більше показниками, щоб зрозуміти взаємозв'язки між змінними. Ця модель буде корисною, якщо експериментальні дані покажуть, що кращої класифікації можна досягти за допомогою комбінацій корельованих показників, а не аналізувати їх окремо. Є та ін. вивчити багатовимірний метод контролю якості для виявлення вторгнень шляхом створення довгострокового профілю нормальної діяльності. Основна проблема для багатовимірних статистичних ідентифікаторів полягає в тому, що важко оцінити розподіли для даних великої розмірності.

Модель часових рядів: часовий ряд – це серія спостережень, проведених протягом певного інтервалу часу. Нове спостереження є ненормальним, якщо ймовірність його появи в цей час є занадто низька. Придатність цієї методики була підтверджена змодельованими експериментами (Qingtao & Zhiging в 2005, Viinikka в 2009).

2.2.2 Методи, що грунтуються на знаннях

Цю групу методів також називають методом експертної системи. Цей підхід вимагає створення бази знань, яка відображає законний профіль трафіку. Дії, що відрізняються від цього стандартного профілю, розглядаються як вторгнення. На відміну від інших класів AIDS, модель стандартного профілю зазвичай створюється на основі людських знань у термінах набору правил, які намагаються визначити нормальну діяльність системи. Основною перевагою методів, заснованих на знаннях, є можливість зменшити хибнопозитивні тривоги, оскільки система має знання про всі нормальні поведінки. Однак у динамічно змінюваному обчислювальному середовищі цей тип IDS потребує регулярного оновлення даних щодо очікуваної нормальної поведінки, що є трудомістким завданням, оскільки зібрати інформацію про всі нормальні поведінки дуже важко. Скінченний автомат (FSM): Finite State Machine — це обчислювальна модель, яка використовується для представлення та керування потоком виконання. Як правило модель представлена у вигляді станів, переходів і дій. FSM перевіряє дані історії. Наприклад, будь-які варіації вхідних даних відзначаються, і на основі виявленої варіації відбувається перехід. FSM може орієнтуватись на нормальну поведінку системи, а будь-яку відхилення від цього, буде розцінюватись як атака. Мова опису: вона визначає синтаксис правил, які можна використовувати для визначення характеристик визначеної атаки. Правила можуть бути побудовані мовами опису, такими як N-граматики та UML(Універсальна мова моделювання). Експертна система: експертна система містить ряд правил, які визначають атаки. В експертній системі правила зазвичай визначаються вручну інженером знань, який працює у співпраці з експертом домену.

Аналіз сигнатур: це найперша техніка, застосована в IDS. Вона опирається на просту ідею зіставлення рядків. Під час зіставлення рядків вхідний пакет перевіряється слово за словом із чітким підписом. Якщо підпис збігається, виникає сповіщення. Якщо ні, інформація в трафіку потім зіставляється з наступним підписом у базі даних підписів.

2.2.3 Машинне навчання.

Машинне навчання це процес отримання знань із великої кількості даних. Моделі машинного навчання (приклад на рисунку 2.3) складаються з набору правил, методів, або складних «функцій передачі», які можна застосовувати для пошуку цікавих моделей даних або для розпізнавання чи прогнозування поведінки. Методи машинного навчання широко застосовуються у області AIDS. Кілька алгоритмів, таких як кластеризація, нейронні мережі, правила асоціації, дерева рішень, генетичні алгоритми та методи k-найближчих сусідів використовують для вивчення наборів даних про вторгнення.

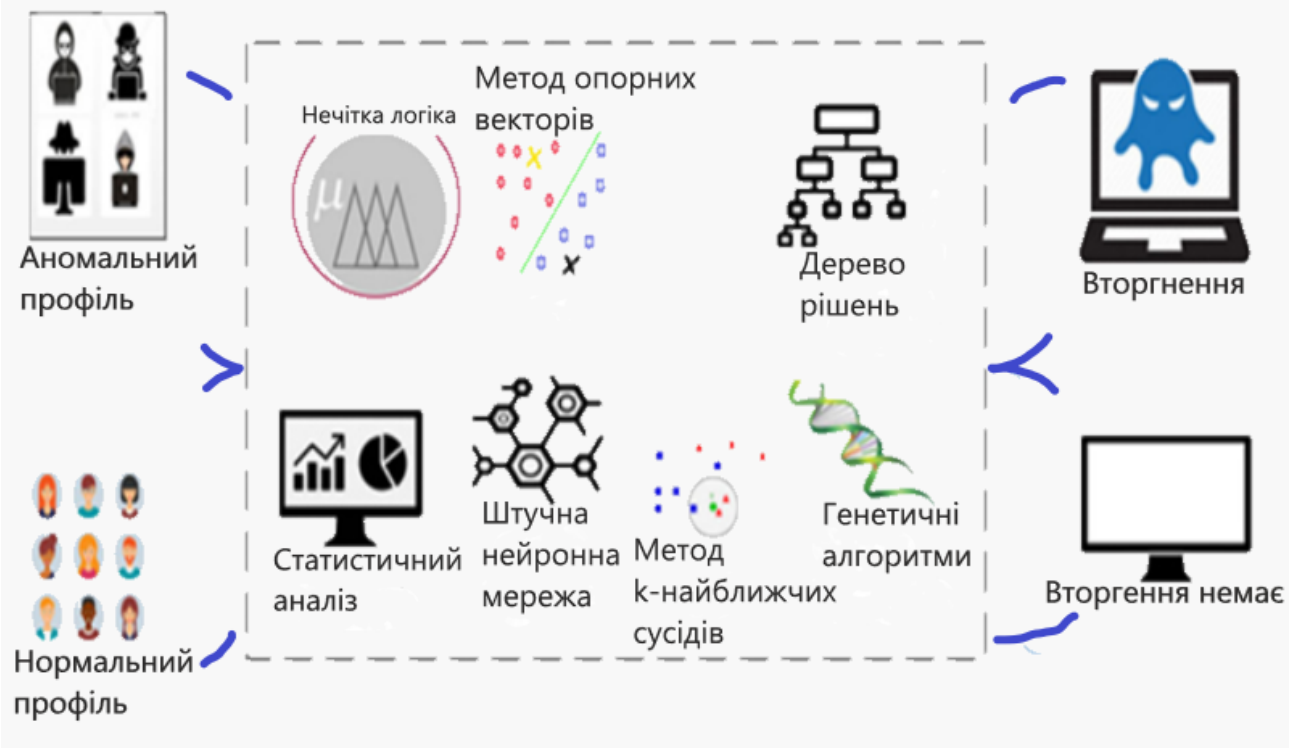

Рисунок 2.3 – Машинне навчання AIDS

Метою використання методів машинного навчання є створення IDS з підвищеною точністю та меншими вимогами до людських знань. За останні кілька років кількість AIDS , які використовують методи машинного навчання, зростає. Основним напрямком IDS на основі досліджень машинного навчання є виявлення шаблонів і створення системи виявлення вторгнень на основі набору даних. Загалом існує два типи методів машинного навчання: контрольовані та

неконтрольовані. Технології IDS на основі контрольованого навчання виявляють вторгнення за допомогою позначених навчальних даних. Підхід до навчання під наглядом зазвичай складається з двох етапів, а саме навчання та тестування. На етапі навчання ідентифікуються відповідні функції та класи, а потім алгоритм вивчає ці зразки даних. У керованому навчанні IDS кожен запис є парою, що містить джерело даних мережі або хоста та пов'язане вихідне значення (тобто мітку), а саме вторгнення або звичайне. Далі можна застосувати вибір функцій для усунення непотрібних функцій. Використовуючи навчальні дані для вибраних функцій, далі використовується методика навчання під наглядом, щоб навчити класифікатор вивчати внутрішній зв'язок, який існує між вхідними даними та позначеним вихідним значенням. Зараз уже досліджено широкий спектр методик навчання під наглядом, кожна з яких має свої переваги та недоліки. На етапі тестування навчена модель використовується для класифікації невідомих даних у клас вторгнення або звичайний. Потім результуючий класифікатор стає моделлю, яка, враховуючи набір значень ознак, передбачає клас, до якого можуть належати вхідні дані. На рисунку 2.4 показано загальний підхід до застосування методів класифікації.

Рисунок 2.4 – Модель класифікації.

Існує багато методів класифікації, таких як дерева рішень, системи на основі правил, нейронні мережі, опорно-векторні машини(ОВМ), наївний баєсів класифікатор та метод k-найближчого сусіда[15]. Кожен метод використовує матод навчання для побудови моделі класифікації. Однак відповідний

класифікаційний підхід повинен не лише обробляти навчальні дані, але й точно ідентифікувати клас записів, яких він ніколи раніше не бачив. Створення моделей класифікації з надійною здатністю до узагальнення є важливим завданням алгоритму навчання.

2.2.3.1 Контрольоване навчання в системі виявлення вторгнень.

Дерева рішень: вони складаються з трьох основних компонентів. Перший компонент – вузол рішення, який використовується для ідентифікації тестового атрибута. Другий – це гілка, кожна гілка предсталяє собою можливе рішення на основі значення тестового атрибуту. Третій – листок, який містить клас, до якого належить екземпляр. Існує багато різних алгоритмів дерев рішень, включаючи ID3 (Квінлан, 1986), C4.5 (Квінлан, 2014) і CART (Брейман, 1996).

Наївний класифікатор Байєса: цей підхід базується на застосуванні принципу Байєса з надійними припущеннями незалежності серед атрибутів. Наївний Байєсів класифікатор відповідає на таке питання як «яка ймовірність того, що відбувається певний вид атаки, враховуючи спостережувану діяльність системи?» застосовуючи формули умовної ймовірності. Він покладається на ознаки, які мають різну ймовірність появи в атаках і в нормальній поведінці. Наївний Байєсів класифікатор є однією з найпоширеніших моделей в IDS через її простоту використання та ефективність обчислень. Однак система не працює належним чином, якщо припущення незалежності невірне. Також результати досліджень показують, що наївна модель Байєса має знижену точність для великих наборів даних.

Генетичні алгоритми (GA). Генетичні алгоритми — це евристичний підхід до оптимізації, заснований на принципах еволюції. Кожне можливе рішення представлено у вигляді серії бітів (генів) або хромосом, і якість рішень покращується з часом завдяки застосуванню операторів відбору та відтворення, упереджених на користь кращих рішень. У застосуванні генетичного алгоритму до проблеми класифікації вторгнень, як правило, існує два типи кодування хромосом: один відповідно до методу кластеризації для генерації бінарного

хромосомного кодування; інший – визначення центру кластера (матриці прототипу кластеризації) за допомогою цілочисельної кодуючої хромосоми. Мюррей в 2014 році використовував GA для розробки простих правил для мережевого трафіку. Кожне правило преставлено геномом, а первинна популяція геномів – це кілька випадкових правил. Кожен геном складається з різних генів, які відповідають таким характеристикам, як джерело IP, IP адреса, джерело порту, та 1 тип протоколу.

Штучна нейронна мережа (Artificial Neural Network (далі ANN)). ANN є одним з найпоширеніших методів машинного навчання, який успішно виявляє різні шкідливі програми. Найпоширенішим методом навчання під наглядом є алгоритм зворотнього поширення(backpropagation algorithm). Алгоритм зворотнього поширення оцінює градієнт помилки мережі щодо її модифікованої ваги. Проте для IDS на основі ANN точність виявлення, особливо для менш частих атак, все ще потребує покращення. Навчальний набір даних для менш частих атак невеликий порівняно з більш частими атаками, і це ускладнює для ANN правильне вивчення властивостей цих атак. Як наслідок, точність виявлення нижча для менш частих атак. У сфері інформаційної безпеки може бути завдано величезної шкоди, якщо низькочастотні атаки не будуть виявлені. Наприклад якщо атаки User to Root(U2R) уникають виявлення, кіберзлочинець може отримати права авторизації користувача root і таким чином, виконувати зловмисні дії на комп’ютерах жертви. ANN часто страждають від локальних мінімумів, тому навчання може зайняти дуже багато часу. Перевагою ANN є те, що одним або декількома прихованими шарами він здатний створювати дуже нелінійні моделі, які фіксують складні зв’язки між вхідними атрибутами та мітками класифікації. Отже ANN є потужними інструментами для багатьох завдань класифікації, включаючи IDS.

Нечітка логіка: ця техніка базується на ступенях невизначеності, а не на типовій істинній чи хибній булевій логіці, на основі якої створено сучасні ПК. Таким чином, це простий спосіб прийти до остаточного висновку на основі нечітких, неоднозначних, галасливих, неточних або відсутніх вхідних даних. З

нечіткою областю нечітка логіка дозволяє екземпляру належати, можливо частково, до кількох класів одночасно. Таким чином, нечітка логіка є хорошим класифікатором для проблем IDS, оскільки сама безпека включає нечіткість, а межа між нормальним і ненормальним станами не ідентифікована. Крім того, проблема виявлення вторгнення містить різні числові характеристики в зібраних даних і кілька похідних статистичних показників. Побудова IDS на основі числових даних із жорсткими пороговими значеннями створює велику кількість помилкових тривог. Діяльність, яка незначно відхиляється від моделі, не може бути розпізнана, також незначна зміна нормальної активності може викликати помилкові тривоги. За допомогою нечіткої логіки можна змоделювати цю незначну аномалію, щоб підтримувати низькі помилкові показники. Elhag та ін. в 2015 році показали, що за допомогою нечіткої логіки частота помилкових тривог при визначенні вторгнення може бути зменшена. Вони окреслили групу нечітких правил для опису нормальних і ненормальних дій у комп'ютерній системі, а також механізм нечіткого висновку для визначення вторгнень.

Опорно-векторні машини (Support Vector Machines(далі SVM)). SVM – це дискримінаційний класифікатор, визначений гіперплощиною розщеплення. SVM використовують функцію ядра для відображення навчальних даних у просторі з більшою розмірністю, щоб вторгнення класифікувалося лінійно. Також SVM добре відомі своєю здатністю до узагальнення і в основному цінні, коли кількість атрибутів велика, а кількість точок даних невелика. Різні типи роздільних гіперплощин можуть бути досягнуті шляхом застосування ядра, наприклад лінійного, поліноміального, радіально-базисної функції Гауса(RBF) або гіперболічного тангенса[17]. У наборах даних IDS, багато функцій є зайвими або менш впливовими на розділення точок даних на правильні класи. Тому вибір функцій слід враховувати під час навчання SVM. SVM також можна використовувати для класифікації у кілька класів.

Прихована Марковська модель(Hidden Markov Model(далі HMM)): HMM – це статистична модель Маркова у якій система, що моделюється, вважається марковським процесом з невидимими даними. У техніці навчання під наглядом

прихована Марковська модель навчається на відомі функції шкідливого програмного забезпечення (наприклад, послідовність коду операцій), і після завершення етапу навчання, навчена модель застосовується для оцінки вхідного трафіку. Потім оцінка порівнюється з попередньо визначеним пороговим значенням, а оцінка, що перевищує порогове значення, вказує на шкідливе програмне забезпечення. Так само, якщо оцінка нижча за порогове значення, трафік визначається як нормальний.

Метод k-найближчих сусідів (K-Nearest Neighbors(далі KNN)). KNN є типовим непараметричним класифікатором, що застосовується в машинному навчанні. Ідея цих методів полягає в тому, щоб назвати нерозмічену вибірку даних класом її k найближчих сусідів (де k є цілим числом, що визначає кількість сусідів, яку слід враховувати). KNN можна належним чином застосувати як еталон для всіх інших класифікаторів, оскільки він забезпечує хорошу ефективність класифікації в більшості IDS.

2.2.3.2 Неконтрольоване навчання в системі виявлення вторгнень

Неконтрольоване навчання – це форма техніки машинного навчання, яка використовується для отримання цікавої інформації з вхідних наборів даних без міток класу. Точки вхідних даних зазвичай розглядаються як набір випадкових змінних. Потім для набору даних створюється модель щільності з’єднання. У контрольованому навчанні вихідні мітки надаються та використовуються, щоб навчити машину отримувати необхідні результати для невидимої точки даних, тоді як у неконтрольованому навчанні мітки не даються, натомість дані автоматично групуються в різні класи в процесі навчання. У контексті розробки IDS, засоби неконтрольованого навчання використовують механізм для виявлення вторгнень за допомогою немаркованих даних для навчання моделі.

2.2.4 Категорії аномалій

Хоча такий підхід дозволяє виявити невідомі типи атак, у нього є свої недоліки, такі як хибні спрацьовування, наприклад невідома раніше законна діяльність також може бути класифікована як шкідлива.

Система аналізує роботу мережі в даний момент, порівнює з аналогічним періодом та виявляє аномалії. Цей різновид IDS чимось схожий з відстеженням станів, тільки має більше охоплення.

Аномалії діляться на три категорії:

- статистичні;
- аномалії протоколів;
- аномалії трафіку.

Статистичні аномалії виявляються коли система IDS складає профіль штатної активності (об’єм вхідного/вихідного трафіку, запущені програми і т.д.) та порівнює його з поточним профілем. Наприклад для компанії характерне зростання трафіку в будні на 90%. Якщо трафік раптом зросте не на 90, а на 900% то система повідомить про загрозу.

Для виявлення аномалій протоколів IDS система аналізує комунікаційні протоколи, їх зв’язки з користувачами, додатками і складає профілі. Наприклад веб сервер повинен працювати на порті 80 для HTTP і 443 для HTTPS. Якщо для передачі інформації по HTTP або HTTPS буде використовуватись інший порт, IDS про це повідомить.

Також IDS здатні виявити аномалії, будь-яку небезпечну чи навіть загрозливу активність в мережевому трафіку. Для прикладу, розглянемо DoS атаку. Якщо спробувати виконати таку атаку «в лоб», її розпізнає та зупинить навіть брандмауер. Хакери можуть надсилати пакети з різних адрес (DDoS), що складніше виявити. IDS технології дозволяють аналізувати мережевий трафік, і завчасно запобігти подібним атакам. Існують різні типи джерел даних, які розглядаються для виявлення вторгнень на основі хосту. Файли системного журналу містять інформацію про попередження, помилки та збої системи. Дані системного аудиту створюються програмами та містять детальнішу інформацію,

ніж системні журнали, пов’язану із сеансами користувача (наприклад, дії командного рядка, час входу, підвищення привілеїв тощо). Збір даних обох типів є дорогим. З цієї причини дані системних викликів, які не потребують попередньої обробки, наразі є більш популярним джерелом інформації. Трасування системного виклику – це послідовність усіх системних викликів, створених процесом або програмою за певний проміжок часу. Нарешті, реєстр Windows і файлові системи також використовуються як джерела інформації, хоча й рідше.

Застосування методів на основі аномалій ґрунтується на наявності даних одного з вищезазначених типів. Зараз є кілька доступних наборів даних, які також полегшують порівняння різних методів. Набір даних ADFA Linux (ADFA-LD12), представлений Creech і Hu17, використовувався для оцінки методів машинного та глибокого навчання в багатьох дослідницьких роботах і є набором даних системного виклику, зібраним у середовищі Linux. Крім того, Haider та інші. представили два набори даних на базі Windows, ADFA-WD і ADFA-WD:SAA, обидва складаються з відбіркових даних аудиту. Нещодавно та ж дослідницька група представила синтетичний набір даних під назвою NGIDS-DS (набір даних IDS наступного покоління), який складається як з мережевого трафіку, так і з даних журналів хост-системи, що відображає критичні кіберінфраструктури різних підприємств. Крім того, набір даних AWSCTD містить дані системного виклику, створені на хості Windows (включаючи аргументи системного виклику та значення, що повертаються). У деяких випадках набори даних, які в основному призначені для використання для NIDS, наприклад NSL-KDD, також використовувались для розробки та оцінки систем HIDS. Проблема для підходів до виявлення аномалій полягає в тому, що вони зазвичай дають високу частоту помилкових тривог (FAR), що означає, що відносно велика кількість нормальних послідовностей даних характеризується як аномальні. Цей показник оцінки найчастіше називають частотою помилкових позитивних результатів (FPR).

2.3 Порівняння методів виявлення вторгнень

Виявлення на основі сигнатур та на основі аномалій, це два основні методи виявлення загроз які використовують системи виявлення вторгнень, щоб попередити адміністраторів мережі про ознаки загрози.

Виявлення на основі сигнатур зазвичай найкраще використовувати для визначення відомих загроз. Воно працює за допомогою попередньо запрограмованого списку відомих загроз, та їх індикаторів компрометації (indicators of compromise) далі IOC. IOC може бути певною поведінкою яка зазвичай передує зловмисній мережевій атаці, хешам файлів, шкідливим доменам, відомим послідовностям байтів, або навіть вмісту заголовків тем електронних листів. Оскільки SIDS, відстежує пакети, що проходять мережею, вона порівнює ці пакети з базою даних відомих IOC або сигнатур атаки, щоб виявити будь-яку підозрілу поведінку.

Виявлення на основі аномалій може попередити вас про підозрілу поведінку, яка ще невідома. Замість пошуку відомих загроз, AIDS використовує машинне навчання, щоб навчити систему виявлення розпізнавати нормалізований базовий рівень. Базовий рівень відображає звичайну поведінку системи, а потім вся мережева активність порівнюється з цим базовим рівнем. Також замість того щоб шукати відомі IOC, AIDS просто визначає будь-яку незвичайну поведінку щоб викликати сповіщення.

За допомогою AIDS, все, що не відповідає існуючому нормальному режиму, наприклад спроба користувача ввійти в систему поза стандартним робочим часом, додавання нових пристроїв до мережі без авторизації або потік нових IP адрес, які намагаються встановити з’єднання з мережею — підніме хвилю помилкових спрацьовувань. Недоліком є те, що багато нешкідливих дій буде позначено просто як нетипові. Підвищена ймовірність помилкових спрацьовувань може вимагати додаткового часу та ресурсів для перевірки всіх сповіщень про потенційні загрози.

У той же час цей потенційний недолік робить метод виявлення вторгнень на основі аномалій здатним виявляти експлойти нульового дня, що не може виявити метод виявлення на основі сигнатур. Виявлення на основі сигнатур обмежується списком з відомих існуючих загроз. З іншого боку, він також має високу швидкість обробки та більшу точність для відомих атак. Ці два методи мають переваги та недоліки, тому найкращим рішенням буде використовувати їх в тандемі [17].

2.4 Важливість та безпека лог-файлів

Обсяг логів і повідомлень про події може бути величезним, і виникає спокуса просто ігнорувати їх. Однак ризик судового розгляду через розголошення даних або збиток, який може бути завдано бізнесу через втрату даних , означає, що неспроможність захистити дані зараз може зруйнувати бізнес. Питання безпеки та захисту даних тепер включені в вимоги контрактів, і існує багато стандартів, яких зараз дотримуються галузі, щоб заспокоїти зацікавлених сторін і зберегти бізнес у безпеці. Відповідність стандартам цілісності даних включає вимоги щодо обслуговування логів.

Залежно від того, який стандарт реалізує компанія, нам потрібно буде зберігати файли журналу протягом кількох років. Отже, керування файлами логів тепер стало важливою вимогою для бізнесу.

Задля безпеки логів необхідно підтримувати їх цілістність. Повідомлення про події можуть ідентифікувати спроби вторгнення, тому файли логів є цілями для хакерів. Зловмисник може замести свої сліди, проводячи маніпуляції в файлах логів, щоб видалити записи які можуть містити сліди вторгення. Таким чином лог сервер, який постійно створює резервні копії файлів журналу та перевіряє наявність неавторизованих змін, є важливим елементом для дотримання високих стандартів безпеки даних [1].

Системи HIDS не можуть ефективно захистити ресурси системи, якщо її вихідна інформація скомпрометована. Захист файлів логів також поширюється на систему автентифікації мережі. Жодна автоматизована система захисту файлів журналу не зможе відрізнити авторизований доступ до файлу журналу від несанкціонованого без моніторингу безпеки дозволів користувача.

Системи виявлення вторгнень на основі хосту – не єдині методи захисту від вторгнень. Вони діляться на дві категорії. HIDS – один із цих секторів, інший – мережеві системи виявлення вторгнень (NIDS). Приклад мережі з NIDS зображений на рисунку 2.5.

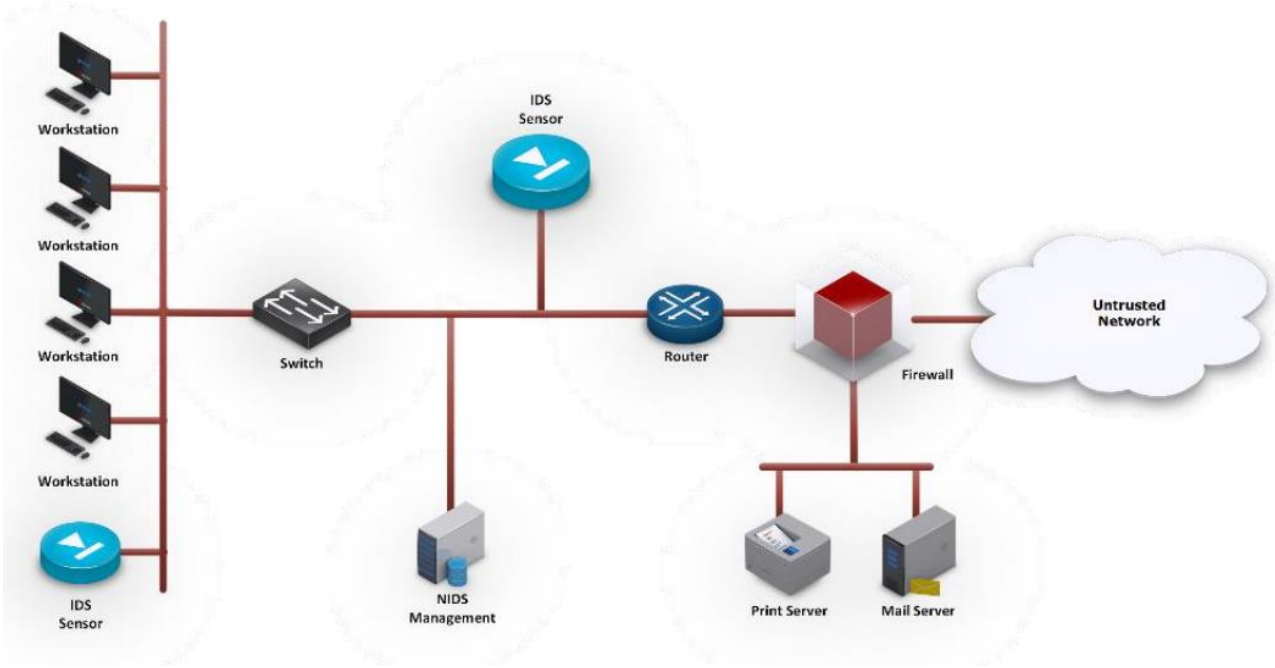

Рисунок 2.5 – Типова мережева топологія NIDS

І HIDS, і NIDS перевіряють системні повідомлення. Це стосується як перегляду журналу, так і повідомлень про події. Однак NIDS також перевіряє пакетні дані під час їх проходження по мережах. Емпіричне правило, яке розподіляє відповідальність за виявлення вторгнень між цими двома методологіями, полягає в тому, що NIDS захоплює живі дані для виявлення, а HIDS перевіряє записи у файлах .

Перевага NIDS полягає в тому, що вони пропонують швидшу відповідь, ніж HIDS. Щойно в мережі виникає підозріла подія, NIDS має виявити це та

повідомити про це. Однак хакери підступні та постійно коригують свої методи, щоб уникнути виявлення. Деякі моделі дій стають очевидними як зловмисні, лише якщо розглядати їх у ширшому контексті .

Питання що краще, HIDS чи NIDS, не є великою проблемою, тому що насправді потрібні обидва [16].

Аналізуючи історію активності, HIDS виявляє моделі активностей які відбуваються з часом. Проте навіть у мережах середнього розміру обсяг логів, які щодня генеруються, може бути дуже великим тому важливо вибрати ефективний інструмент сортування та пошуку.

HIDS який краще виявляє загрози, але працює повільніше не варто використовувати, тому що нові записи постійно накопичуються, і швидкий HIDS часто може показати себе краще за будь-який інший. Більшість системних адміністраторів ідуть на компроміс та обирають швидкість, замість якості, однак якщо інструмент HIDS є одночасно швидким та якісним, то це найкраще рішення.

3 РОЗГОРТАННЯ ТА ДОСЛІДЖЕННЯ СИСТЕМИ БЕЗПЕКИ

3.1 Установка та налаштування OSSEC

На ОС Ubuntu 22.04 установимо необхідні пакети для роботи OSSEC[10]:

```
apt-get install make gcc libssl-dev
```

Завантажимо OSSEC з офіційного сайту github:

```
wget https://github.com/ossec/ossec-hids/archive/3.1.0.tar.gz
```

Отримуємо результат:

```
--2022-11-12 08:30:35--  https://github.com/ossec/ossec-hids/archive/3.1.0.tar.gz
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/ossec/ossec-hids/tar.gz/refs/tags/3.1.0 [following]
--2022-11-12 08:30:36--  https://codeload.github.com/ossec/ossec-hids/tar.gz/refs/tags/3.1.0
Resolving codeload.github.com (codeload.github.com)... 140.82.121.10
Connecting to codeload.github.com (codeload.github.com)|140.82.121.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: '3.1.0.tar.gz'

    [     <=>                                                                 ] 1,886,469   2.05MB/s   in 0.9s

2022-11-12 08:30:37 (2.05 MB/s) - '3.1.0.tar.gz' saved [1886469]
```

Розпакуємо завантажений файл:

```
tar -xvzf 3.1.0.tar.gz
```

Перейдемо до директорії яку ми розпакували:

```
cd ossec-hids-3.1.0/
```

Запустимо скрипт інсталяції:

```
sh install.sh
```

Обираємо такі значення:

(en/br/cn/de/el/es/fr/hu/it/jp/nl/pl/ru/sr/tr) [en]: en

OSSEC HIDS v3.1.0 Installation Script - http://www.ossec.net

You are about to start the installation process of the OSSEC HIDS.

You must have a C compiler pre-installed in your system.

- System: Linux ip-172-31-43-187 5.4.0-1045

- User: root

- Host: ip-172-31-43-187

-- Press ENTER to continue or Ctrl-C to abort. --

1- What kind of installation do you want (server, agent, local, hybrid or help)? local

- Local installation chosen.

2- Setting up the installation environment.

- Choose where to install the OSSEC HIDS [/var/ossec]: /var/ossec

- Installation will be made at /var/ossec .

3- Configuring the OSSEC HIDS.

3.1- Do you want e-mail notification? (y/n) [y]: y

- What's your e-mail address? user@email.com

- We found your SMTP server as: mx01.mail.com.

- Do you want to use it? (y/n) [y]: y

--- Using SMTP server: mx01.mail.com.

3.2- Do you want to run the integrity check daemon? (y/n) [y]: y

- Running syscheck (integrity check daemon).

3.3- Do you want to run the rootkit detection engine? (y/n) [y]: y

- Running rootcheck (rootkit detection).

3.4- Active response allows you to execute a specific

command based on the events received. For example,

you can block an IP address or disable access for

a specific user.

More information at:

http://www.ossec.net/en/manual.html#active-response

- Do you want to enable active response? (y/n) [y]: y
- Active response enabled.
- By default, we can enable the host-deny and the
firewall-drop responses. The first one will add
a host to the /etc/hosts.deny and the second one
will block the host on iptables (if linux) or on
ipfilter (if Solaris, FreeBSD or NetBSD).
- They can be used to stop SSHD brute force scans,
portscans and some other forms of attacks. You can
also add them to block on snort events, for example.
- Do you want to enable the firewall-drop response? (y/n) [y]: y
- firewall-drop enabled (local) for levels >= 6
- Default white list for the active response:
- 127.0.0.53
- Do you want to add more IPs to the white list? (y/n)? [n]:

3.6- Setting the configuration to analyze theNow Ossec interface is Ready. following logs:

-- /var/log/auth.log
-- /var/log/syslog
-- /var/log/dpkg.log
-- /var/log/apache2/error.log (apache log)
-- /var/log/apache2/access.log (apache log)
- If you want to monitor any other file, just change
the ossec.conf and add a new localfile entry.
Any questions about the configuration can be answered
by visiting us online at http://www.ossec.net .

5- Installing the system

- Running the Makefile

CC external/cJSON/cJSON.o

LINK libcJSON.a

RANLIB libcJSON.a

cd external/zlib-1.2.11/ && ./configure && make libz.a

Checking for gcc...

Checking for shared library support...

Building shared library libz.so.1.2.11 with gcc.

Checking for size_t... Yes.

Запускаємо ossec-control:

/var/ossec/bin/ossec-control start

Отримуємо результат:

```
Starting OSSEC HIDS 2.9.3 (by Trend Micro Inc.)...
Deleting PID file '/var/ossec/var/run/ossec-remoted-1297.pid' not used...
ossec-csyslogd already running...
2022/11/12 08:46:25 ossec-maild: INFO: E-Mail notification disabled. Clean Exit.
Started ossec-maild...
ossec-execd already running...
ossec-analysisd already running...
ossec-logcollector already running...
Started ossec-remoted...
ossec-syscheckd already running...
ossec-monitord already running...
Completed.
```

Інсталюємо веб інтерфейс OSSEC:

git clone https://github.com/ossec/ossec-wui.git

Результат:

```
Cloning into 'ossec-wui'...
remote: Enumerating objects: 205, done.
remote: Total 205 (delta 0), reused 0 (delta 0), pack-reused 205
Receiving objects: 100% (205/205), 217.04 KiB | 0 bytes/s, done.
Resolving deltas: 100% (69/69), done.
```

Виконаємо наступні команди, для налаштування облікових даних для входу в OSSEC

mv ossec-wui /srv

cd /srv/ossec-wui

./setup.sh

Вводимо ім’я користувача, пароль, та логін користувача веб-сервера

```
/srv/ossec-wui# ./setup.sh
trap: SIGHUP: bad trap
Setting up ossec ui...
Username: admin
New password:
Re-type new password:
Adding password for user admin
Enter your web server user name (e.g. apache, www, nobody, www-data, ...)
www-data
Setup completed successfully.
```

Налаштовуємо Apache2:

```
vim /etc/apache2/sites-available/ossec-wui.conf
```

Вказуємо такі значення:

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
ServerName test
ServerAlias www.test
DocumentRoot /srv/ossec-wui/
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
<Directory/srv/ossec-wui/>
Options FollowSymLinks
AllowOverride All
Require all granted
</Directory>
</VirtualHost>
```

Вмикаємо сайт:

```
a2ensite ossec-wui.conf
```

Вмикаємо режим перезапису:

```
a2ensite rewrite
```

Перезапускаємо apache2:

```
systemctl restart apache2
```

3.2 Встановлення агента на компютері з ОС Windows.

Для початку необхідно встановити OSSEC Server, для цього переходимо на офіційний сайт https://ossec.github.io/downloads.html, обираємо безкоштовний тарифний план, та завантажуємо версію OSSEC Agent Windows як зображено на рисунку 3.1[8].

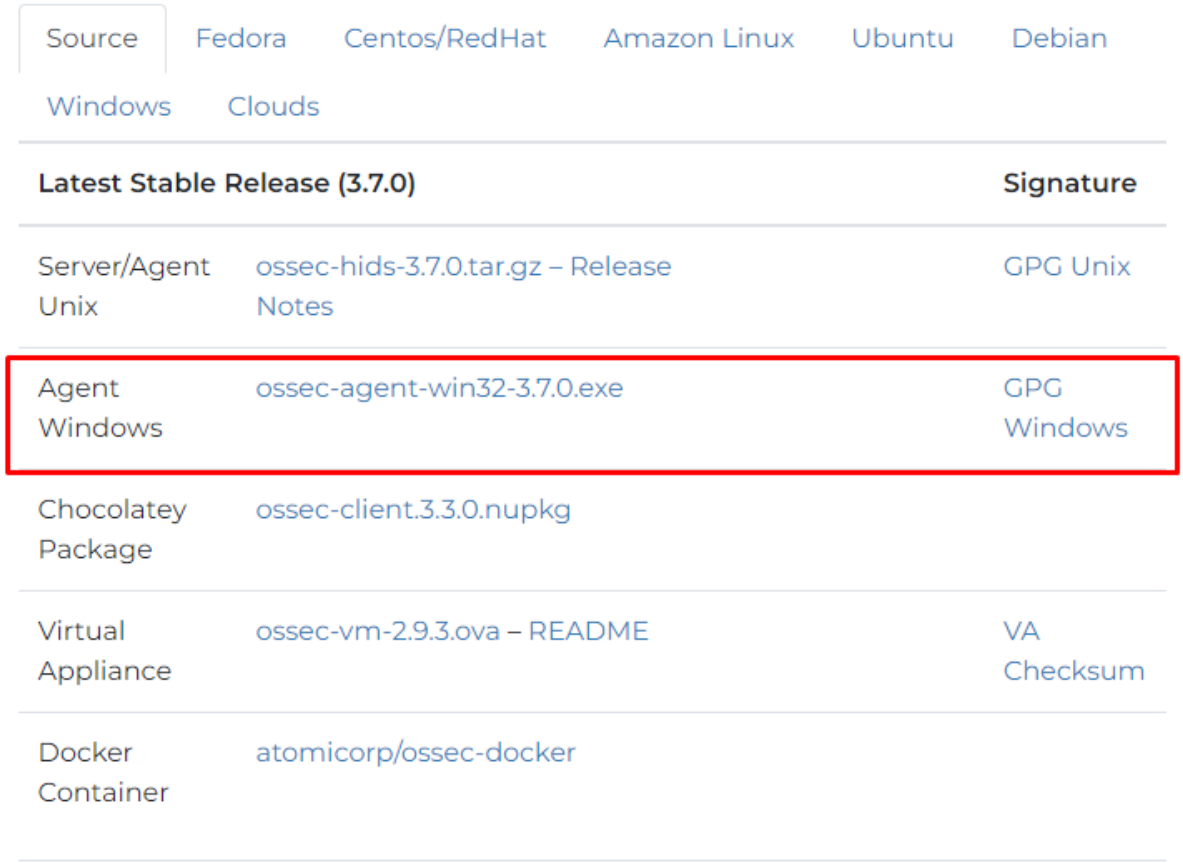

Рисунок 3.1 – Доступні версії OSSEC

Тепер в терміналі нашого сервера OSSEC вводимо команду /var/ossec/bin/manage_agents.

Результат:

```
[root@ossec-server ossec]# /var/ossec/bin/manage_agents


****************************************
* OSSEC HIDS v2.9.2 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: A
```

Обираємо A, агент менеджер запитає ім’я, можна будь-яке. Далі, він запитає IP адресу. Після цього він отримує унікальний ідентифікатор. Ідентифікатор має складатися з числа, що містить максимум вісім цифр. Менеджер агентів також пропонує ідентифікатори для нових агентів.

Нарешті, він запитує підтвердження всієї наданої інформації, потім він додає всю інформацію про агента до /var/ossec/etc/client.keys і повертається до головного меню. Результат:

```
- Adding a new agent (use '\q' to return to the main menu).
  Please provide the following:
   * A name for the new agent: agentd
   * The IP Address of the new agent: 10.10.50.2
   * An ID for the new agent[002]:
Agent information:
   ID:002
   Name:agentd
   IP Address:10.10.50.2

Confirm adding it?(y/n):
```

Тепер нам необхідно видобути ключ клієнта. В головному меню обираємо опцію E. Відобразиться список агентів:

```
Choose your action: A,E,L,R or Q: E

Available agents:
   ID: 002, Name: agentd, IP: 10.10.50.2
Provide the ID of the agent to extract the key (or '\q' to quit):
```

Вводимо повний ідентифікатор агента, який ми додали. Менеджер відобразить весь ключ. Копіюємо його в буфер обміну, оскільки він знадобиться нам пізніше.

```
Provide the ID of the agent to extract the key (or '\q' to quit): 002

Agent key information for '002' is:
MDAyIGFnZW50ZCAxMC4xMC41MC4yIGEzODE2ODFhYzMxMTAzMzI4NTc3NzJhOTc4NGNkOWVhYTAyMjdlNDViYTBhODU4YjE4Y2ZlZDExNjYwMmJlYjI=

** Press ENTER to return to the main menu.
```

Результат приведені на рисунку 3.2.

Далі на Windows запускаємо OSSEC Agent Manager, вводимо ip адресу нашого сервера та ключ. Далі клацаємо вкладку керування, та перезапускаємо наш агент.

3.3 Процес моніторингу

OSSEC показує логи в вигляді журналу. Однак деяка інформація недоступна в файлах журналу, але якщо нам побрібно її моніторити, то є рішення. Для того щоб усунути цю прогалину OSSEC додали можливість відстежувати вихідні дані команд через OSSEC і розглядати дані цих команд так само, як логи.

Приклади конфігурації.

Використання дискового простору (df-h).

Для того щоб контролювати використання дискового простору, нам потрібно налаштувати роботу cron, щоб виводити дані в файл журналу (/var/log/df.log) і налаштувати OSSEC для перегляду цих даних.

Починаючи з OSSEC версії 2.3, можна контролювати команди безпосередньо в OSSEC за наступною конфігурацією (/var/ossec/etc/ossec.conf)[9]:

```
<localfile>
        <log_format>command</log_format>
        <command>df -h</command>
</localfile>
```

Коли будь-який розділ диску заповниться на 100%, ми отримаємо приблизно таке сповіщення:

```
** Alert 1257451341.28290: mail - ossec,low_diskspace,
2022 Nov 22 16:02:21 (home-ubuntu) 192.168.0.0->df -h

Rule: 531 (level 7) -> "Partition usage reached 100% (disk space monitor)."
Src IP: (none)
User: (none)
ossec: output: 'df -h': /dev/sdb1 24G 12G 11G 100% /var/backup
```

Середнє навантаження (uptime).

Для того щоб контролювати середнє навантаження, необхідно налаштувати OSSEC для моніторингу команди «uptime» і сповіщень коли середнє навантаження більше ніж 2 (для прикладу).

```
<localfile>
  <log_format>command</log_format>
  <command>uptime</command>
</localfile>
```

І в правилі (/var/ossec/rules/local_rules.xml):

```
<rule id="100101" level="7" ignore="7200">
  <if_sid>530</if_sid>
  <match>ossec: output: 'uptime': </match>
  <regex>load averages: 2.</regex>
  <description>Load average reached 2..</description>
</rule>
```

Ця функція має багато можливостей для експериментів щоб додати свої ідеї для моніторингу та правил.

Сповіщення коли вивід команди змінюється.

Якщо потрібно сповіщення коли змінюються логи чи вивід команди, в OSSEC є параметр <check_diff /> для правил.

Наприклад, створимо правило для сповіщення, коли на сервері відкривається новий порт у режимі прослуховування.

Спочатку налаштуємо OSSEC для виконання команди, додавши наступне до ossec.conf: netstat -tan grep listen

```
<localfile>
  <log_format>full_command</log_format>
  <command>netstat -tan |grep LISTEN|grep -v 127.0.0.1</command>
</localfile>
```

Після цього додаємо правило, яке сповіщатиме про зміну результату:

```
<rule id="140123" level="7">
  <if_sid>530</if_sid>
  <match>ossec: output: 'netstat -tan |grep LISTEN</match>
  <check_diff />
  <description>Listened ports have changed.</description>
</rule>
```

Перший раз подія зберігатиметься у внутрішній базі даних. Кожного разу коли OSSEC отримує ту саму подію, він порівнюватиме її з тою яку зберегло у внутрішній базі даних, і сповіщатиме лише про зміни виводу.

Приклад результату після запуску прослуховування порту 23456:

OSSEC HIDS Notification.

2022 Nov 22 16:36:30

Received From: XYZ->netstat -tan |grep LISTEN|grep -v 127.0.0.1

Rule: 140123 fired (level 7) -> "Listened ports have changed."

Portion of the log(s):

ossec: output: 'netstat -tan |grep LISTEN|grep -v 127.0.0.1':

tcp4 0 0 *.23456 *.* LISTEN

tcp4 0 0 *.3306 *.* LISTEN

tcp4 0 0 *.25 *.* LISTEN

Previous output:

ossec: output: 'netstat -tan |grep LISTEN|grep -v 127.0.0.1':

tcp4 0 0 *.3306 *.* LISTEN

tcp4 0 0 *.25 *.* LISTEN

Вявлення використання USB-накопичувачів за допомогою OSSEC. Розглянемо процес з використанням нової функції :xml:`check_diff`.

Для початку налаштуємо агент Windows для моніторингу запису реєстру USBSTOR за допомогою команди reg:

```
<agent_config os="windows">
  <localfile>
    <log_format>full_command</log_format>
    <command>reg QUERY HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR</command>
  </localfile>
</agent_config>
```

Далі створюємо локальне правило для цієї команди:

```
<rule id="140125" level="7">
  <if_sid>530</if_sid>
  <match>ossec: output: 'reg QUERY</match>
  <check_diff />
  <description>New USB device connected</description>
</rule>
```

Через кілька хвилин з’явиться каталог /var/ossec/queue/diff/[agent_name]/[rule_id]. Коли хтось додасть новий пристрій, ми отримаємо таке сповіщення.

```
** Alert 1268687754.35062: mail  - local,syslog,
2022 Nov 22 16:55:54 (xx-netbook) any->reg QUERY
HKLMSYSTEMCurrentControlSetEnumUSBSTOR
Rule: 140125 (level 7) -> 'New USB device connected'
Src IP: (none)
User: (none)
ossec: output: 'reg QUERY
HKLMSYSTEMCurrentControlSetEnumUSBSTOR':! REG.EXE VERSION 3.0

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTOR
HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_&Prod_USB_Flash_Memory&Rev_5.00
HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_Generic&Prod_Flash_Disk&Rev_8.0
HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_Hitachi&Prod_HTS543225L9A300&Rev_
HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_LEXAR&Prod_JD_FIREFLY&Rev_1100
HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_SAMSUNG&Prod_HM160JC&Rev_0000
```

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_Sony&Prod_DSC&Rev_1.00

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_TomTom&Prod_ONE_XXL_IQ_Rts

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_USB_2.0&Prod_USB_Flash_Drive&Rev_0.00

Previous output:

ossec: output: 'reg QUERY
HKLMSYSTEMCurrentControlSetEnumUSBSTOR':

! REG.EXE VERSION 3.0

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTOR

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_&Prod_USB_Flash_Memory&Rev_5.00

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_Generic&Prod_Flash_Disk&Rev_8.07

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_Hitachi&Prod_HTS543225L9A300&Rev_

HKEY_LOCAL_ACHINESYSTEMCurrentControlSetEnumUSBSTORDisk&
Ven_SAMSUNG&Prod_HM160JC&Rev_0000

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_Sony&Prod_DSC&Rev_1.00

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_TomTom&Prod_ONE_XXL_IQ_Rts

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetEnumUSBSTORDisk
&Ven_USB_2.0&Prod_USB_Flash_Drive&Rev_0.00

ВИСНОВКИ

В кваліфікаційній роботі розв’язано актуальну задачу підвищення ефективності систем виявлення вторгнень. При цьому отримано наступні результати.

Проведено аналіз існуючих систем виявлення вторгнень, що дало можливість виявити їхні переваги та недоліки.

Порівняно види систем виявлення вторгнень, що дало можливість впевнитись в ефективності їх одночасного використання.

Проведено аналіз методів виявлення вторгнень, на основі яких побудовані системи виявлення вторгнень.

Встановлено та налаштовано систему безпеки кінцевих пристроїв з використанням інструменту OSSEC.

Розроблено та протестовано алгоритм моніторингу параметрів, які не включені за замовчуванням у OSSEC, зокрема: контроль використання дискового простору, середнє навантаження та виявлення використання USB накопичувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dora Tudor. What Is a Host Intrusion Detection System (HIDS) and How It Works. [Електронний ресурс]. Режим доступу: https://heimdalsecurity.com/blog/host-intrusion-detection-system-hids/

2. Marc Wilson. Best Host-Based Intrusion Detection Systems (HIDS) Tools & Software. [Електронний ресурс]. Режим доступу: https://www.pcwdld.com/host-based-intrusion-detection-systems-hids-tools-and-software#wbounce-modal

3. Evan Klein. Top 5 open-source HIDS systems. [Електронний ресурс]. Режим доступу: https://logz.io/blog/open-source-hids/

4. Stephen Cooper. Host-Based Intrusion Detection Systems Explained – 6 Best HIDS Tools for 2022. [Електронний ресурс]. Режим доступу: https://www.comparitech.com/net-admin/hids-tools-software/#HIDS_vs_NIDS

5. Stephen Cooper. 10 Best Network Intrusion Detection Systems Software & NIDS Tools [Електронний ресурс]. Режим доступу: https://www.comparitech.com/net-admin/nids-tools-software/#What_is_the_difference_between_NIDS_and_HIDS

6. Harman Singh. Host-based Intrusion Detection System – Overview and HIDS vs NIDS.[Електронний ресурс]. Режим доступу: https://thecyphere.com/blog/host-based-ids/

7. Ansam Khraisat , Iqbal Gondal, Peter Vamplew, Joarder Kamruzzaman. Survey of intrusion detection systems: techniques, datasets and challenges. 2-11.

8. Windows Agent Installation manual OSSEC. [Електронний ресурс]. Режим доступу: https://www.ossec.net/docs/docs/manual/installation/installation-

9. OSSEC Process Monitoring Manual. [Електронний ресурс]. Режим доступу: https://www.ossec.net/docs/docs/manual/monitoring/process-monitoring.html

10. Step by Step Guide to Install OSSEC HIDS on Ubuntu 20.04 LTS [Електронний ресурс]. Режим доступу:

https://www.hackerxone.com/2021/09/19/step-by-step-guide-to-install-ossec-hids-on-ubuntu-20-04-lts/

11. Host-based intrusion detection system [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/Host-based_intrusion_detection_system

12. Network Intrusion Detection System (NIDS) [Електронний ресурс]. Режим доступу: https://netacea.com/glossary/network-intrusion-detection-system-nids/

13. What Does Network-based Intrusion Detection System (NIDS) Mean? [Електронний ресурс]. Режим доступу: https://www.techopedia.com/definition/12941/network-based-intrusion-detection-system-nids#what-does-network-based-intrusion-detection-system-nids-mean

14. Анатолій Лазар. IDS – що це таке? Система виявлення вторгнень (IDS) як працює? [Електронний ресурс]. Режим доступу: https://poradumo.com.ua/49510-ids-sho-ce-take-sistema-viiavlennia-vtorgnen-ids-iak-pracuye/

15. S. S. Tirumala, H. Sathu and A. Sarrafzadeh. Free and open source intrusion detection systems: A study. *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2015, pp. 205-210, doi: 10.1109/ICMLC.2015.7340923

16. Intrusion Detection System (IDS): Signature vs. Anomaly-Based [Електронний ресурс]. Режим доступу: https://www.n-able.com/blog/intrusion-detection-system

17. Panos Panagiotou, Notis Mengidis, Theodora Tsikrika, Stefanos Vrochidis, Ioannis Kompatsiaris. Host-based Intrusion Detection Using Signature-based and AI-driven Anomaly Detection Methods, 1998, pp. 38-41, doi: https://doi.org/10.11610/isij.5016

18. Janis Griffin. What Is an Intrusion Detection System (IDS)? [Електронний ресурс]. Режим доступу: https://logicalread.com/intrusion-detection-system/

19. Intrusion Detection System (IDS). [Електронний ресурс]. Режим доступу: https://www.geeksforgeeks.org/intrusion-detection-system-ids/

20. Garrett Gross. Intrusion Detection Techniques, Methods & Best Practices. [Електронний ресурс]. Режим доступу: https://cybersecurity.att.com/blogs/security-essentials/intrusion-detection-techniques-methods-best-practices

21. Михайлишин Д.А., Цаволик Т.Г., Драпак В.І. Система моніторингу безпеки кінцевих пристроїв. Матеріали наукової конференції «Автоматизація та комп'ютерно-інтегровані технології» (АКІТ - 2022), Тернопіль, 2022. – С. 107–109.

22. Михайлишин Д.А. Методи виявлення вторгнень в комп'ютерні мережі. Матеріали наукової конференції «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ - 2022), Тернопіль, 2022. – С.68-69.

ДОДАТОК А.

Копія публікацій

ЗАХІДНОУКРАЇНСЬКИЙНАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТЕФАНИКА
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ
НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ
НАДВІРНЯНСЬКИЙ КОЛЕДЖ НТУ
ГАЛИЦЬКИЙ КОЛЕДЖ ІМ. В. ЧОРНОВОЛА

Проблемно-наукова міжгалузева конференція

АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

(АКІТ – 2022)

21—23 лютого 2022 року

Тернопіль

АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

БЕЗПЕКА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 004.056

Михайлишин Д.А.[1], Цаволик Т.Г.[1], Драпак В.І.[1]

Західноукраїнський національний університет

СИСТЕМА МОНІТОРИНГУ БЕЗПЕКИ КІНЦЕВИХ ПРИСТРОЇВ

Вступ. Ландшафт загроз стає дедалі різноманітнішим, а системи, які використовуються для атак, стають більш досконалими, ніж будь-коли раніше. У 2018 році підприємства та організації будь-якого розміру та в усіх галузях зіткнулися з серйозними витоками даних (з витоком інформації зазнали Aadhar – 1,1 млрд користувачів; myFitnessPal – 150 млн; Quora – 100 млн; Facebook – 29 млн і багато інших). Однією з найбільших жертв став Marriot. Один несанкціонований доступ призвів до викрадення 500 мільйонів особистих даних. Відповідно до «Звіту про розслідування витоку даних за 2018 рік», понад 73% порушень були скоєні сторонніми особами [1].

Найкращі системи виявлення вторгнень у хост з відкритим кодом допомагають компаніям відстежувати порушення безпеки та шахрайство. Очікується, що світовий ринок систем виявлення вторгнень на базі хостів зросте з 4,8 мільярда доларів США у 2020 році до 6,2 мільярда доларів США у 2025 році.

З безперервним розвитком Інтернету кібербезпека стала необхідністю як для великих і відомих організацій, так і для малих підприємств і окремих осіб. Системи виявлення вторгнень (IDS) вважаються ефективним способом виявлення та запобігання загрозам кібербезпеки. Однак системам виявлення та запобігання вторгненням не приділялося достатньо уваги та обізнаності, особливо серед малих підприємств і окремих осіб. Отже, вибір і розгортання IDS є важливим у зв'язку з тим, що цей предмет вважається технічно складним, дорогим і трудомістким процесом.

Мета: дослідження програмних засобів моніторингу безпеки кінцевих пристроїв.

1. Функції та призначення HIDS

Системи моніторингу безпеки кінцевих пристроїв є важливими для забезпечення безпеки сучасних організацій і мережевого трафіку. Ці засоби захисту використовуються для захисту обмеженого доступу до мережі організації. Що стосується систем виявлення вторгнень, існує два різних типи; хост-орієнтовані (HIDS) і мережево-орієнтовані системи (NIDS). Мережевий IDS аналізує мережевий трафік на наявність будь-яких вторгнень і видає сповіщення, тоді як HIDS відстежує поведінку хостів на наявність будь-якої підозрілої активності, перевіряючи події у мережі.

Моніторинг безпеки кінцевих пристроїв виконується за допомогою системи виявлень вторгнень на основі хоста (Host-based intrusion detection system, HIDS), далі будемо називати HIDS. HIDS схожий на використання розумних камер безпеки у домі; якщо зловмисник увірветься у ваш будинок, камера почне записувати та надішле сповіщення в реальному часі на ваш мобільний пристрій [2].

Виявлення вторгнень стало важливим методом захисту для мереж, щоб боротися зі слабкими місцями безпеки, властивими будь-якій системі у якій є людський елемент. Незалежно від того, наскільки сильною є ваша політика доступу користувачів, хакери завжди можуть їх обійти, за допомогою соціальної інженерії обманом змусивши співробітника розкрити облікові дані доступу. Хакери з доступом можуть займати корпоративну систему роками, не будучи виявленими. Цей тип атаки називається розвинена стала загроза (Advanced Persistent Threat далі APT). IDS спеціально спрямовані на викорінення APT.

HIDS головним чином зосереджується на моніторингу та аналізі файлів журналу з метою виявлення аномалій і неавторизованих змін на основі попередньо визначених політик і набору правил.

HIDS призначена для [3]:

1) виявлення аномалій;
2) виявлення сигнатурних атак;
3) виявлення атаки нульового дня;
4) моніторинг трафіку на хості;
5) контроль цілісності файлів;
6) аналізу журналу;
7) відповідності та аудит;
8) система оповіщення та тривог.

На рисунку 1 приведена загальна схема системи виявлення вторгнень на основі хоста.

Рисунок 1 – Загальна схема HIDS

2. Недоліки та переваги HIDS

HIDS не запобігає вторгненням або атакам, як це робить IPS. Хоча HIDS можна встановити на точках мережі, таких як маршрутизатори або сервери, вони не можуть контролювати на рівні мережі.

З іншого боку, NID (виявлення вторгнення в мережу) можна встановити в точках перетину мережі та контролювати трафік.

HIDS не фільтрує вхідний/вихідний трафік на основі правил, як це робить

брандмауер або монітор пропускної здатності.

HIDS не призначена для зупинки атак, проте вона найкраща для виявлення всіх типів атак і делегування «профілактики» комусь іншому.

3. Принцип роботи HIDS

Інструмент HIDS зосереджений на моніторингу журналів. Більшість програм створюють журнальні повідомлення, і збереження цих записів у файлах дає змогу шукати в них з часом і виявляти ознаки вторгнення. Однією з великих проблем зі збором кожного повідомлення журналу у вашій системі є те, що ми отримаємо великий обсяг даних який потребує ресурсів сховища.

Зберігання журнальних повідомлень систематичним чином допомагає визначити потрібний файл для отримання даних за програмою та датою. Отже, перший крок до отримання суттєвої інформації з системи журналювання – це організувати імена файлів і структуру каталогів сервера файлів журналів.

Наступним кроком у впровадженні HIDS є автоматичне виявлення. Воно шукатиме в журналах певні події, які, швидше за все, могли зафіксувати зловмисну активність. Це ядро інструменту HIDS, а метод виявлення, який визначає, які записи отримати, встановлюється політиками та базою правил .

Спеціальна система виявлення вторгнень відстежує трафік на наявність зловмисних дій або порушень політики. Вона виявляє відомі атаки за конкретними діями, які вони виконують (сигнатури). HIDS аналізує трафік (подібно до аналізатора мережевого трафіку) і поведінку, яка відповідає цим сигнатурам, у реальному часі на хості. Вона також може виявляти незвичайні моделі використання за допомогою методів виявлення аномалій. Коли HIDS знаходить збіг, вона запускає тривогу та сповіщає адміністратора. Багато систем HIDS дозволяють написати власні правила генерації сповіщень . Однак те, що ми дійсно шукаєте, для нашої системи безпеки, — це набір попередньо написаних правил, які включають досвід експертів із безпеки, які пишуть програмне забезпечення.

Система HIDS настільки хороша, наскільки хороші політики, які вона надає. Не можна очікувати, що ми будемо встигати за всіма останніми методами атак, водночас приділяючи час повсякденним завданням нашої роботи, немає сенсу намагатись знати все, якщо ми можемо отримати рішення, надані нам інструментом HIDS.

Висновки. В роботі розглянуто роботу системи HIDS, визначено її переваги та недоліки.

Перелік використаних джерел.

1. Top 5 open-source HIDS systems. Evan Klein [Електронний ресурс].- Режим доступу: https://logz.io/blog/open-source-hids/

2. Host-Based Intrusion Detection Systems Explained – 6 Best HIDS Tools for 2022. Stephen Cooper. [Електронний ресурс]. - Режим доступу: https://www.comparitech.com/net-admin/hids-tools-software/

3. S. S. Tirumala, H. Sathu and A. Sarrafzadeh. Free and open source intrusion detection systems: A study. 2015 International Conference on Machine Learning and Cybernetics (ICMLC), 2015, pp. 205-210, doi: 10.1109/ICMLC.2015.7340923.

ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
ГАЛИЦЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМ. В. ЧОРНОВОЛА

КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ (КБКІТ – 2022)

науково-практична конференція
молодих вчених, аспірантів та студентів

29–31 серпня 2022
Тернопіль

УДК 004.056

Михайлишин Д.А.

Західноукраїнський національний університет

МЕТОДИ ВИЯВЛЕННЯ ВТОРГНЕНЬ В КОМПЮТЕРНІ МЕРЕЖІ

Вступ. Система виявлення вторгнень (Intrusion detection system, IDS) – це, як правило, програмне забезпечення або апаратний пристрій, який відстежує вхідний і вихідний мережевий трафік на наявність ознак зловмисної діяльності або порушень політики безпеки. Системи виявлення вторгнень і продукти IDS часто порівнюють із сигналізацією, яка сповіщає вас про будь-які дії, які можуть поставити під загрозу ваші дані або мережу.

Залежно від типу системи виявлення вторгнень, яка була розгорнута, різні продукти IDS поводитимуться по-різному. Наприклад, мережева с истема виявлення вторгнень (Network intrusion detection system, NIDS) стратегічно розмістить датчики в кількох місцях у самій мережі. Потім ці датчики відстежуватимуть мережевий трафік, не створюючи проблем з продуктивністю або вузьких місць. Системи виявлення вторгнень на основі хостів (Host-based intrusion detection system, HIDS) працюють на певних пристроях і здатні контролювати трафік лише для цих конкретних пристроїв і хостів [1, 2].

Що стосується методів виявлення, то HIDS і NIDS можуть використовувати підхід на основі сигнатур або на основі аномалій. Деякі продукти IDS навіть можуть поєднувати обидва методи виявлення для більш комплексного підходу.

Мета: Порівняння методів виявлення вторгнень, які використовують HIDS та NIDS.

1. Виявлення на основі сигнатур

Виявлення на основі аналізу сигнатур було першим методом, застосованим для виявлення вторгнень. IDS такого типу працює по схожому з антивірусним програмним забезпеченням принципу. Вони аналізують сигнатури, і порівнюють їх з базою яка повинна постійно оновлюватись, для забезпечення коректної роботи. Відповідно, в цьому головний недолік IDS які працюють на основі виявлення на основі сигнатур, якщо по якійсь причині база недоступна, мережа стає вразливою. Також якщо нова атака нова і ще не внесена в базу, є ризик того що загроза не буде виявлена [2].

Сигнатурні IDS здатні відстежувати шаблони чи стани. Шаблони – це ті сигнатури, які зберігаються в базі, що постійно оновлюється. Стани - це будь-які дії всередині системи.

Початковий стан системи – нормальна робота, відсутність атак. Після успішної атаки система переходить у скомпрометований стан, тобто зараження пройшло успішно. Кожна дія(Наприклад встановлення з'єднання за протоколом, що не відповідає політиці безпеки компанії, активізація ПЗ тощо) здатна змінити стан. Тому сигнатурні IDS відстежують не дії, а стан системи. NIDS частіше відстежують шаблони, а HIDS – стан системи.

2. Виявлення на основі аномалій

Виявлення на основі аномалій були введені для того щоб виявляти невідомі атаки, через швидкий розвиток шкідливих програм, основний підхід полягає у використанні машинного навчання, щоб створити модель правдоподібної діяльності, з якою потім порівнюють нові поведінки. Для правильної роботи таких систем виявлення загроз, потрібний тестовий період навчання. Адміністраторам рекомендується протягом кількох місяців повністю вимкнути сигнали тривоги, щоб система навчалась. Після тестового періоду вона готова до роботи.

Хоча такий підхід дозволяє виявити невідомі типи атак, у нього є свої недоліки, такі як хибні спрацьовування, наприклад невідома раніше законна діяльність також може бути класифікована як шкідлива.

Система аналізує роботу мережі в даний момент, порівнює з аналогічним періодом та виявляє аномалії. Цей різновид IDS чимось схожий з відстеженням станів, тільки має більше охоплення.

Аномалії діляться на три категорії:

- статистичні;
- аномалії протоколів;
- аномалії трафіку.

Статистичні аномалії виявляються коли система IDS складає профіль штатної активності (об'єм вхідного/вихідного трафіку, запущені програми і т.д.) та порівнює його з поточним профілем. Наприклад для компанії характерне зростання трафіку в будні на 90%. Якщо трафік раптом зросте не на 90, а на 900% то система повідомить про загрозу.

Для виявлення аномалій протоколів IDS система аналізує комунікаційні протоколи, їх зв'язки з користувачами, додатками і складає профілі. Наприклад веб сервер повинен працювати на порті 80 для HTTP і 443 для HTTPS. Якщо для передачі інформації по HTTP або HTTPS буде використовуватись інший порт, IDS про це повідомить.

Також IDS здатні виявити аномалії, будь-яку небезпечну чи навіть загрозливу активність в мережевому трафіку. Для прикладу, розглянемо DoS атаку. Якщо спробувати виконати таку атаку «на пряму», її розпізнає та зупинить навіть брандмауер. Хакери можуть надсилати пакети з різних адрес (DDoS), що складніше виявити. IDS технології дозволяють аналізувати мережевий трафік, і завчасно запобігти подібним атакам.

Висновок. В роботі розглянуто два методи виявлення вторгнень, дізнались в чому полягають їх відмінності, недоліки та переваги.

Перелік використаних джерел.

1. 1.Intrusion Detection System (IDS): Signature vs. Anomaly-Based [Електронний ресурс]. Режим доступу: https://www.n-able.com/blog/intrusion-detection-system

2. What is a Host Intrusion Detection System(HIDS) and how it works. Dora Tudor. [Електронний ресурс]. Режим доступу: https://heimdalsecurity.com/blog/host-intrusion-detection-system-hids/