МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Західноукраїнський національний університет

Факультет комп’ютерних інформаційних технологій

Кафедра кібербезпеки

ЛАПШИН Ярослав Романович

Алгоритм поглибленого аналізу загроз на основі інструментів SysFlow / The algorithm for in-depth threat analysis based on SysFlow tools

спеціальність 125 Кібербезпека

освітньо-професійна програма - Кібербезпека

кваліфікаційна робота

Виконав студент групи КБм-21

Я.Р. Лапшин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Науковий керівник:

д.т.н., професор В.В. Яцків

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кваліфікаційну роботу

допущено до захисту

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ р.

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.В. Яцків

підпис

Тернопіль – 2021

РЕЗЮМЕ

Кваліфікаційна робота на тему «Алгоритми поглибленого аналізу загроз на основі інструментів SysFlow» зі спеціальності 125 «Кібербезпека» написана обсягом 71 сторінок і містить 25 рисунок, 1 таблицю, 2 додатки та 60 джерел за переліком посилань.

Метою роботи є розробка алгоритмів поглибленого аналізу загроз на основі інструментів SysFlow.

Методи досліджень. Для розв’язання поставлених задач у випускній кваліфікаційній роботі використано методи: теорію поглибленого аналізу загроз, теорію алгоритмів (для аналізу розроблених методів та алгоритмів), алгоритмах теорії графів, об’єктно-орієнтоване програмування.

Результати дослідження: на основі запропонованих алгоритмів досліджено алгоритми поглибленого аналізу загроз на основі SysFlow.

Проведено тестування розроблених програмних налаштувань для поглибленого аналізу загроз на основі інструментів SysFlow, що підтвердило ефективність розроблених налаштувань.

Орієнтовні напрямки розвитку досліджень: у подальшому буде проведено більше досліджень для розвитку даної теми.

Ключові слова: АНАЛІЗ ЗАГРОЗ, ПОГЛИБЛЕНИЙ АНАЛІЗ ЗАГРОЗ, АЛГОРИТМИ, SYSFLOW, ІНСТРУМЕНТИ SYSFLOW.

RESUME

Qualification work on "Algorithms for in-depth threat analysis based on SysFlow tools" in the specialty 125 "Cybersecurity" is written in the length of 71 pages and contains 25 figures, 1 table and 60 sources from the list of links.

The aim of the work is to develop algorithms for in-depth threat analysis based on SysFlow tools.

Research methods. The following methods were used in the final qualification work to solve the set tasks: theory of in-depth threat analysis, theory of algorithms (for analysis of developed methods and algorithms), graph theory algorithms, object-oriented programming.

Research results: based on the proposed algorithms, in-depth threat analysis algorithms based on SysFlow were investigated.

The developed software settings for in-depth threat analysis based on SysFlow tools were tested, which confirmed the effectiveness of the developed settings.

Indicative directions of research development: in the future more research will be conducted for the development of this topic.

Keywords: THREAT ANALYSIS, IN-DEPTH THREAT ANALYSIS, ALGORITHMS, SYSFLOW, SYSFLOW TOOLS.

# 

# Зміст

[ВСТУП](#_c189fbik1nk4) 5

[1 АНАЛІЗ ЗАГРОЗ ТА ЙОГО ОСОБЛИВОСТІ](#_hnxpynmsterv) 7

[1.1 Аналіз загроз. Життєвий цикл аналізу загроз](#_27erq89tgerf) 7

[1.2 Типи аналізу загроз](#_2a9r1z3oxmof) 12

[1.3 Аналіз загроз інформаційної безпеки](#_5blkct11fbo4) 15

[1.4 Поглиблений аналіз загроз кібербезпеки та дослідження його впливу](#_eg7ksp4np9no) 17

[1.5 Класифікація загроз](#_fjzjtkayjkpd) 18

[1.6 Машинне навчання для поглибленого аналізу загроз](#_yw6o2gpct28m) 22

[2 ПОГЛИБЛЕНИЙ АНАЛІЗ БЕЗПЕКИ ВНУТРІШНІХ МЕРЕЖ НА ОСНОВІ FLOW-ПРОТОКОЛУ](#_ck5ja6aymuij) 25

[2.1 Поглиблений аналіз загроз за допомогою Flow - протоколів](#_1hus6v235ks4) 25

[2.2. Інструменти для аналізу Flow-протоколів](#_j5fj8w9oesj5) 35

[2.3 SysFlow як засіб моніторингу системи та мережі](#_ma1d4x1gvg7v) 40

[3 ПОГЛИБЛЕНИЙ АНАЛІЗ ЗАГРОЗ З ВИКОРИСТАННЯМ SYSFLOW](#_btyae3pzjfxv) 43

[3.1 Основні елементи SysFlow](#_kut8pvjskadc) 43

[3.2 Інструменти SysFlow для поглибленого аналізу загроз](#_6ly18f4ehjy3) 46

[3.3 Алгоритм моніторингу SysFlow потоків](#_1wtsoy1qpkq3) 49

[3.4 Алгоритм розкриття атаки на основі SysFlow](#_meer3ls94ong) 52

[3.5 Алгоритм моніторингу локальних і хмарних робочих навантажень та збору даних аналітики безпеки за допомогою SysFlow та ElasticSearch.](#_o2qffa27doy2) 54

[ВИСНОВКИ](#_hrj7udz83xqp) 66

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ](#_bavielkmoba1) 67

# 

# ВСТУП

Кіберзагрози зростають щодня по мірі того, як технічний прогрес в області штучного інтелекту та інтелектуальних систем полегшує необхідність у більш досконалих здібностях для обходу захищених систем. З цих причин керівники організацій повинні провести ретельний і докладний аналіз загроз, щоб знати ступінь схильності їх бізнесу або підприємств кібератакам. Основною метою аналізу кіберзагроз є отримання результатів, що використовуються для ініціалізації або підтримки контррозвідувальних розслідувань.

Поглиблений аналіз загроз допомагає організаціям отримувати цінні знання про ці загрози, створювати ефективні механізми захисту і знижувати ризики, які можуть завдати фінансовий і репутаційних збиток. Аналіз загроз - це можливість прогнозування майбутніх атак, яким піддається організація, щоб вона могла активно адаптувати свій захист і запобігати майбутні атаки.

Жодна організація не застрахована від кіберзлочинності. Хакери націлюються на підприємства, уряд, установи або навіть на окремих осіб, що володіють цінною інформацією. Основною метою аналізу загроз дозволити експертам з безпеки краще зрозуміти як може думати та що може зробити зловмисник. Отримані дані можуть показати на план, метод, мотиви, цілі та процедури, які були використані для здійснення атаки. Фактично, це призводить до покращення засобів безпеки, відносячи до них оцінку, розпізнавання, а також тривалість відповіді на ситуацію.

Кваліфікаційна робота присвячена дослідженню питань, пов'язаних з SysFlow, який перетворює необроблену інформацію про системні події в поняття, що описує поведінку процесів та їх взаємозв'язку з контейнерами, файлами та мережами. Щоб заповнити ці сліпі зони, SysFlow вивчає нові технології моніторингу, які забезпечують всебічне уявлення про поведінку хмарного робочого навантаження. Наочність служб і додатків має вирішальне значення для створення надійної системи безпеки та можливості зниження ризиків кібербезпеки.

Завданням дослідження є розглянути методи та алгоритми поглибленого аналізу загроз на основі інструментів SysFlow.

Метою кваліфікаційної роботи є розробка алгоритмів поглибленого аналізу загроз на основі інструментів SysFlow. Забезпечення системи моніторингу та аналізу безпеки мережі на основі інструментів SysFlow. Система може збирати, зберігати, отримувати доступ, вимірювати й аналізувати дані мережевого потоку, використовувати підходи машинного навчання з контекстом ролі хоста та поведінки користувачів. Представлені результати показують, що дана технологія може дозволити розробити систему безперервного моніторингу та аналізу безпеки мережі в реальному часі з використанням алгоритмів поглибленого аналізу загроз.

Кінцеві результати містять найкращі практичні рекомендації про те, як використовувати захисні засоби контролю для забезпечення цілісності, доступності та конфіденційності, не зачіпаючи умови функціональності і зручності використання.

# 1 АНАЛІЗ ЗАГРОЗ ТА ЙОГО ОСОБЛИВОСТІ

## 1.1 Аналіз загроз. Життєвий цикл аналізу загроз

Цифрові технології сьогодні лежать в основі майже кожної галузі. Автоматизація та велика зв'язаність, які вони надають, зробили революцію у світових економічних та культурних установах, але вони також принесли ризик у вигляді кібератак. Інформація про загрози - це знання, які дозволяють запобігти або пом'якшити ці атаки. Заснована на даних, система аналізу загроз надає контекст — наприклад, хто на вас нападає, які їхні мотиви та можливості, а також які ознаки компрометації у ваших системах слід шукати, — що допомагає приймати обґрунтовані рішення щодо вашої безпеки.

Аналіз загроз - це знання, засновані на фактичних даних, включаючи контекст, механізми, показники, наслідки та рекомендації, орієнтовані на конкретні дії, щодо існуючої або виникаючої загрози. Ця інформація може бути використана для прийняття рішень щодо реакції суб'єкта на цю загрозу або небезпек.

Чому важлива інформація про загрози? Сьогодні індустрія [кібербезпеки стикається з численними проблемами](https://www.recordedfuture.com/cybersecurity-issues-challenges/) — все більш наполегливими і підступними [учасниками загроз](https://www.recordedfuture.com/threat-actor-types/), щоденним потоком даних, повних сторонньої інформації, і помилковими сигналами тривоги в декількох зв'язаних системах безпеки, а також серйозною нестачею кваліфікованих фахівців.

Деякі організації намагаються включити [потоки даних про загрози](https://www.recordedfuture.com/threat-intelligence-feeds/) у свою мережу, але не знають, що робити з усіма цими додатковими даними, що збільшує навантаження на аналітиків, у яких може не бути інструментів для вирішення того, що розставляти пріоритети, а що ігнорувати. Рішення для аналізу кіберзагроз може вирішити кожну з цих проблем. Кращі рішення використовують машинне навчання для автоматизації збору та обробки даних, інтеграції з існуючими рішеннями, збору неструктурованих даних з різних джерел, а потім пов'язують точки зору, надаючи інформацію про показники компрометації (IOC) і тактиці, методах і процедурах (TTP) учасників загроз. Інформація про загрози піддається дії — вона своєчасна, забезпечує контекст і може бути зрозуміла людьми, відповідальними за прийняття рішень.

Розвідка кіберзагроз широко вважається сферою діяльності елітних аналітиків. Насправді це підвищує цінність різних функцій безпеки для організацій усіх розмірів.

Коли інформація про загрози розглядається як окрема функція в рамках широкої парадигми безпеки, а не як важливий компонент, який доповнює всі інші функції, в результаті багато людей, які могли б отримати найбільшу вигоду з інформацією про загрози, не мають доступу до неї, коли вони її потребують.

Оперативні групи по забезпеченню безпеки, як правило, не в змозі обробляти одержувані попередження. Інформація про загрози інтегрується з вже використовуваними рішеннями щодо забезпечення безпеки, що допомагає автоматично визначати пріоритети й фільтрувати попередження та інші загрози. Команди управління вразливими можуть більш точно розставляти пріоритети у найбільш важливих вразливостей, маючи доступ до зовнішньої інформації і контексту, що надаються службою розвідки загроз. Та запобігання шахрайства, аналіз ризиків і інші процеси забезпечення безпеки високого рівня збагачуються розумінням поточного ландшафту загроз, що надається розвідкою загроз, включаючи ключові відомості про учасників загроз, їх тактиці, методах і процедурах і багато чого іншого з джерел даних в Інтернеті.

Компанії виявляють все більшу стурбованість на рахунок цифрових атак. Враховуючи зростання кількості витоків даних, інформація про які все частіше потрапляє у випуски новин, це не дивно. Однак багато керівників підприємств не знають, з чого розпочати будувати захист свого бізнесу та його цифрових активів. Перш за все ви повинні мати уявлення про деякі основні поняття, коли йдеться про кібербезпеку.

Розглянемо один із ключових аспектів кібербезпеки – аналіз загроз.

Аналіз загроз – це процес, який виявляє та аналізує кіберзагрози, націлені на підприємства або бізнес. Ключове слово "аналіз". Аналіз загроз - це ретельне вивчення великих обсягів даних, які досліджуються контекстуально, щоб виявити реальні проблеми та розгорнути рішення, спеціально призначене для усунення виявленої проблеми.

Визначення аналізу загроз часто спрощують або плутають з іншими термінами кібербезпеки. Найчастіше це стосується понять «дані про загрози» та «аналіз погроз». Дані про загрози це список можливих загроз. Його можна представити у вигляді стрічки новин на сторінці в Facebook - такий перелік потенційних проблем, що постійно оновлюється. Але пости у Facebook нічого не означають, поки не їх не прочитати і не поєднаєте ці знання з попередніми постами. Також працює аналіз загроз: IT-фахівці чи складні інструменти читають загрози та аналізують їх, а потім за допомогою отриманих раніше знань встановлюють, чи реальна загроза, і якщо так, то що з цим робити.

Чому аналіз загроз такий важливий? Аналіз загроз є важливою частиною будь-якої екосистеми кібербезпеки. Кваліфікована програма аналізу кіберзагроз або системи інформування про загрози, націлені на вашу організацію, допоможе запобігти витоку даних. Маючи добре структуровану програму аналізу кіберзагроз, компанія або організація може виявляти кіберзагрози та не допускати витоку конфіденційної інформації. Виявляючи та аналізуючи загрози, програма виявляє шаблони, що використовуються хакерами, та допомагає підприємствам впроваджувати заходи безпеки для захисту від майбутніх атак.

Розглянемо, як створюється інформація про кіберзагрози. Необроблені дані – це не те ж саме, що аналізовані. Аналіз кіберзагроз - це готовий продукт, який виходить в результаті шести етапного циклу збору, обробки та аналізу даних. Цей процес являє собою цикл, оскільки в ході аналізу даних виявляються нові питання та прогалини в знаннях, що призводить до встановлення нових вимог до збору даних. Ефективна розвідувальна програма є ітеративною, з часом вона стає все більш досконалою. Щоб максимізувати цінність аналізу загроз, дуже важливо визначити варіанти використання і визначили свої цілі, перш ніж робити що-небудь ще.

Життєвий цикл аналізу загроз складається з таких основних етапів: планування та керівництво, збір, обробка, аналіз, поширення інформації та зворотний зв'язок (рисунок 1.1).

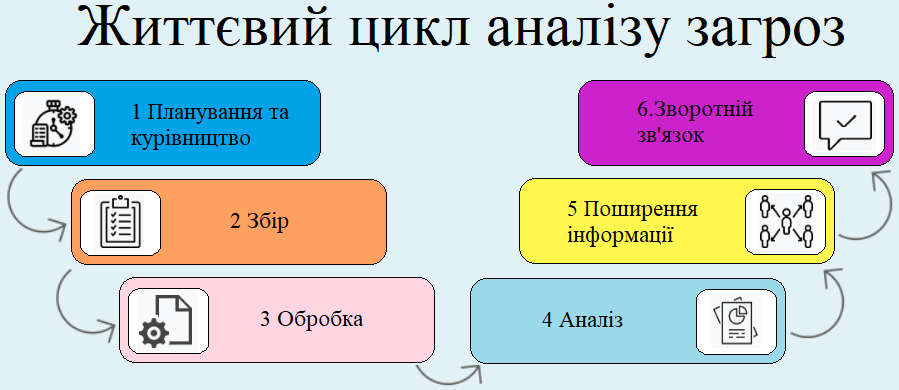


Рисунок 1.1 - Життєвий цикл аналізу загроз

1. Планування та керівництво. Перший крок до отримання інформації про реальні загрози - задати правильне завдання. Зазвичай слід уникати завдань, що фокусуються на одному факті, події або діяльності. Визначте пріоритетність своїх цілей в області аналізу на основі таких факторів, як те, наскільки тісно вони відповідають основним цінностям вашої організації, наскільки великий вплив матиме підсумкове рішення і наскільки вони будуть чутливими до часу.

Одним з важливих керівних факторів на цьому етапі є розуміння того, хто буде споживати готовий продукт і отримувати вигоду з нього — буде інформація передана команді аналітиків з технічними знаннями, яким потрібен швидкий звіт про новий експлойт, або керівнику, який шукає широкий огляд тенденцій для обґрунтування своїх рішень про інвестиції в безпеку на наступний квартал?

2. Збір. Наступним кроком є збір вихідних даних, що відповідають вимогам, встановленим на першому етапі. Найкраще збирати дані з широкого спектру джерел — внутрішніх, таких як журнали мережевих подій і записи минулих відповідей на інциденти, і зовнішніх, з відкритої мережі, темної мережі та технічних джерел.

Дані про загрози зазвичай розглядаються як списки, такі як шкідливі IP-адреси, домени і хеші файлів, але вони також можуть включати інформацію про вразливості, таку як особиста інформація клієнтів, вихідний код і веб-сайтів текст з джерел новин або соціальних мереж.

3. Обробка. Як тільки всі необроблені дані будуть зібрані, вам необхідно відсортувати їх, впорядкувавши за допомогою тегів метаданих і відфільтрувавши надлишкову інформацію, помилкові спрацьовування і негативи. Сьогодні навіть невеликі організації щодня збирають дані про мільйони зареєстрованих подій і сотень тисяч показників. Це занадто багато для людей — аналітиків, щоб ефективно обробляти дані.

4. Аналіз. Наступний крок - розібратися в оброблених даних. Мета аналізу полягає у пошуку потенційних проблем безпеки і інформуванні відповідних команд у форматі, що відповідає вимогам розвідки, викладених на етапі планування і керівництва. Аналіз загроз може приймати різні форми в залежності від початкових цілей і цільової аудиторії, але ідея полягає в тому, щоб перевести дані у формат, зрозумілий аудиторії. Це може варіюватися від простих списків загроз до рецензованих звітів.

5. Поширення інформації. Потім готовий продукт розподіляється серед ймовірних споживачів. Щоб інформація про загрози була дієвою, вона повинна потрапити до потрібних людей у потрібний час. Також необхідно відстежувати, щоб між одним циклом передачі і наступним зберігалася безперервність і дані не було втрачено.

6. Зворотний зв'язок. Завершальний крок - це коли цикл аналізу проходить повний цикл, що робить його тісно пов'язаним з початковою фазою планування та напрями. Після отримання готового продукту розвідки той, хто зробив первинний запит, розглядає його і визначає, чи були дані відповіді на їх питання. Це визначає цілі та процедури наступного циклу збору розвідувальної інформації, що, знову ж таки, робить документацію і безперервність необхідною.

## 1.2 Типи аналізу загроз

Як показує життєвий цикл аналізу загроз, кінцевий продукт буде виглядати по-різному в залежності від початкових вимог, джерел інформації та цільової аудиторії. Може бути корисно розділити інформацію про загрози на кілька категорій на основі цих критеріїв.

Відомості про загрози часто поділяються на три підкатегорії:

* стратегічний аналіз — широкі тенденції, як правило, призначені для нетехнічної аудиторії;
* тактичний аналіз — загальні відомості про тактику, методи і процедури суб'єктів загроз для більш технічної аудиторії;
* оперативний аналіз — технічні відомості про конкретні атаки та вразливості.

Стратегічний аналіз загроз. Стратегічний аналіз загроз надає широкий огляд на загрози націлені на організації. Він призначений для інформування про рішення високого рівня, прийнятих керівниками та іншими особами, що приймають рішення в організації, як таке, утримання, як правило, менш технічне і представлено в доповідях або брифінгах. Хороший стратегічний аналіз повинен забезпечувати розуміння таких областей, як ризики, пов'язані з певними напрямками дій, загальні закономірності тактики і цілей суб'єктів загроз, а також геополітичні події та тенденції.

Загальні джерела інформації для аналізу стратегічних загроз включають:

- політичні документи національних підприємств або неурядових організацій;

- новини місцевих та національних засобів масової інформації, галузевих і тематичних публікацій або інших тематичних видань;

- технічні документи, звіти про дослідження та інший вміст, підготовлене організаціями безпеки;

Створення потужної розвідки стратегічних загроз починається з постановки цілеспрямованих, конкретних питань для визначення вимог до розвідки. Для цього також потрібні аналітики, що володіють знаннями, які виходять за рамки типових навичок в області кібербезпеки, зокрема, глибоке розуміння соціально-політичних і ділових концепцій.

Хоча кінцевий продукт не є технічним, для створення ефективної стратегічної розвідки потрібні глибокі дослідження з використанням величезних обсягів даних, часто на декількох мовах. Це може зробити початковий збір і обробку даних занадто складними для виконання вручну, навіть для тих рідкісних аналітиків, які володіють необхідними мовними навичками, технічною освітою і навичками роботи. Рішення для аналізу загроз, що автоматизує збір і обробку даних, допомагає знизити цю навантаження і дозволяє аналітикам, не володіє таким великим досвідом, працювати більш ефективно.

Тактичний аналіз загроз. Тактичний аналіз загроз описує тактику, методи і процедури (TTP) учасників загроз. Це повинно допомогти захисникам зрозуміти, в конкретних термінах, як їх організація може піддаватися нападу, і найкращі способи захисту від цих нападів або пом'якшення їх наслідків. Зазвичай він включає технічний контекст і використовується персоналом, що безпосередньо беруть участь у захисті організації, таким як системні архітектори, адміністратори і співробітники служби безпеки.

Звіти, що створюються постачальниками систем безпеки, часто є найпростішим способом отримання тактичної інформації про загрози. У звітах про загрози можна знайти інформацію про вектори атаки, інструменти та інфраструктуру, які використовують зловмисники, включаючи відомості про те, які вразливості є мішенню і які експлойти використовують зловмисники, а також про те, які стратегії та інструменти вони можуть використовувати, щоб уникнути або затримати виявлення.

Тактична інформація про загрози повинна використовуватися для інформування про удосконалення існуючих засобів контролю та процесів безпеки і прискорення реагування на інциденти.

Оперативний аналіз загроз.Оперативний аналіз - це знання про кібератаки, події або вразливості в кампаніях. Він дає спеціальні відомості, які допомагають груп реагування на інциденти зрозуміти характер, наміри і терміни конкретних атак.

Оскільки це зазвичай включає технічну інформацію — інформацію про те, який вектор атаки використовується, які вразливості використовуються або які області управління та управління використовуються, — цей вид аналізу також називається аналізом технічних загроз. Поширеним джерелом технічної інформації є канали даних про загрози, які зазвичай фокусуються на одному типі індикатора, наприклад, на хешах шкідливих програм або підозрілих доменах.

Але якщо вважати, що технічний аналіз загроз суворо заснований на технічній інформації, такої як потоки даних про загрози, то технічна та оперативна розвідка загроз не є повністю синонімічними — швидше, це діаграма Венна [19] з величезними перекриттями. Інші джерела інформації про конкретні атаки можуть надходити з закритих джерел, таких як перехоплення повідомлень групи загроз, або шляхом проникнення, або злому цих каналів зв'язку.

Отже, існує кілька перешкод для збору такого роду розвідданих:

- Групи загроз доступу можуть обмінюватися даними по закритим і зашифрованих каналах або вимагати підтвердження особи. Існують також мовні бар'єри для груп загроз, розташованих у зарубіжних країнах.

- Шум – Може бути важко або неможливо вручну зібрати достовірну інформацію з великих джерел, таких як чати і соціальні мережі.

- Заплутування – щоб уникнути виявлення, групи загроз можуть використовувати тактику заплутування, наприклад, використовуючи кодові імена.

Рішення для аналізу загроз, що ґрунтуються на процесах машинного навчання для автоматизованого збору даних у великих масштабах, можуть вирішити багато з цих проблем при розробці ефективної оперативної інформації про загрози. Наприклад, рішення, що використовує обробку різних мов, може збирати інформацію з іншомовних джерел, не вдаючись до людського досвіду для розшифрування.

## 1.3 Аналіз загроз інформаційної безпеки

Багато компаній вже користуються програмами інформування про загрози, і це є хорошою новиною. Щоб досягти успіху, організаціям слід вибирати такого постачальника рішень для забезпечення кібербезпеки, який має для цього всі необхідні інструменти, а не лише окремі.

На що звертати увагу у програмі аналізу загроз? Управління загрозами вимагає повного огляду усіх активів компанії. Для цього потрібна програма, яка відстежує активність, виявляє проблеми та надає дані, необхідні для прийняття обґрунтованих рішень для захисту компанії. Програма, яка матиме доступ до вашої системи, виявлятиме в ній слабкі місця, пропонуватиме заходи захисту та забезпечуватиме цілодобовий контроль. Кібербезпека – це не одне рішення на всі випадки життя, тому не слід погоджуватись на пропозиції, що обіцяють такий підхід.

Аналіз загроз інформаційної безпеки дозволяє виділити складові сучасних комп'ютерних загроз – їх джерела та рушійні сили, способи та наслідки реалізації. Аналіз виключно важливий для отримання всієї необхідної інформації про інформаційні загрози, визначення потенційної величини шкоди, як матеріальної, так і нематеріальної, та вироблення адекватних заходів протидії.

При аналізі загроз інформаційній безпеці використовуються три основні методи (рисунок 1.2.) [6]:



Рисунок 1.2 – Основні аналітичні методи оцінки інформаційних загроз

Розглянемо наведені методи докладніше. Пряма експертна оцінка. Метод експертних оцінок полягає в тому, що параметри загроз задаються експертами. Експерти визначають переліки параметрів, що характеризують загрози інформаційній безпеці, та надають суб'єктивні коефіцієнти важливості кожного параметра. У деяких випадках зручніше для вибору найкращого фактора (об'єкта, альтернативи) спочатку зробити безпосередню оцінку, а потім усі фактори ранжувати. Нехай, наприклад, m експертів оцінили (за шкалою від 0 до 100) до напрямів досліджень з погляду важливості їх задля досягнення певної мети. Коли в експертній оцінці беруть участь кілька експертів, зазвичай намагаються отримати усереднену оцінку (вагу) для кожного з об'єктів, що порівнюються. Для цього нормовані оцінки кожного об'єкта підсумовуються, а потім сума поділяється на кількість експертів.

Статистичний аналіз – це аналіз інформаційних загроз на основі накопичених даних про інциденти інформаційної безпеки, зокрема, про частоту виникнення загроз певного типу, їх джерела та причини успіху чи невдачі реалізації. Наприклад, знання частоти появи загрози дозволяє визначити ймовірність виникнення за певний проміжок часу. Для ефективного застосування статистичного методу потрібна наявність досить великої за обсягом бази даних про інциденти. Потрібно відзначити ще одну вимогу: при використанні об'ємних баз необхідні інструменти узагальнення даних та виявлення у базі вже відомої та нової інформації.

Факторний аналіз грунтується на виявленні факторів, які з певною ймовірністю ведуть до реалізації загроз та тих чи інших негативних наслідків. Такими факторами можуть бути наявність привабливих для кіберзлочинців інформаційних активів, вразливості інформаційної системи, високий рівень вірусної активності у зовнішньому середовищі тощо. Оскільки сучасні інформаційні системи впливають безліч чинників, зазвичай використовується багатофакторний аналіз.

Під час аналізу загроз інформаційної безпеки найефективніше застосовувати комплекс різних аналітичних методів. Це значно підвищує точність оцінки.

## 1.4 Поглиблений аналіз загроз кібербезпеки та дослідження його впливу

Коли питання стосується цифрової безпеки, найбільш важливий елемент є поглиблений аналіз загроз. Це пов’язано з тим, що він передає важливу інформацію, необхідну для забезпечення безпеки ІТ-інфраструктури компанії. Можливість аналізувати великі обсяги даних з різних джерел майже в режимі реального часу дає аналітикам безпеки перевагу перед традиційними підходами до забезпечення безпеки. За даними Fundera [41], ринку фінансових рішень для малого бізнесу, статистика загроз кібербезпеки приводить компанії до тями й змушує їх зосереджувати свою увагу на питаннях інформаційної безпеки, з якою більшість із нас зустрічається постійно.

За даними 2021 року: – 43% усіх кібератак сконцентровані на малому бізнесі. – 60% малих підприємств, які стали жертвами кібератак, вимушені закритися впродовж п’яти - шести місяців. – Кіберзлочинність обходиться малому і середньому бізнесу більш ніж у 2,2 млрд. доларів на рік. – Число нових комп’ютерних порушень у галузі кібербезпеки зросло на 424%. – Охорона здоров’я – одна з галузей, найбільш схильна до кібератак. – 66% малих підприємств схвильовані можливостями атаки кіберзловмисниками. – 14% малих підприємств оцінили власні можливості зменшувати кіберзагрози і кібератаки як високоефективні. – 47% малих підприємств не розуміють, як захистити себе від кібератак. – 66% підприємств найбільше стурбовані компрометацією даних клієнтів і конфіденційністю їх даних. – Лише у 1 з 4 малих підприємств присутній персонал який вирішує питання інформаційної безпеки. Це всього 25% з усіх компаній. – 22% малих підприємств зашифровують свої дані.

Людський фактор і збої систем становлять понад 50% порушень інформаційної безпеки. Тобто найбільшою загрозою та основним ризиком є людина – працівник компанії або користувач. Також апаратне та програмне забезпечення, яке періодично стає сумнівним та неактуальним у досить короткий відрізок часу.

## 1.5 Класифікація загроз

Загрозами кібербезпеки називаються потенційні джерела ненадійних інцидентів, які можуть нанести шкоду ресурсам інформаційної системи. Усі вразливості безпеки, спрямовані проти програмних та технічних засобів інформаційної системи, в кінцевому підсумку впливають на безпеку інформаційних ресурсів та призводять до порушення основних властивостей, які зберігають та обробляють інформацію. Як правило, вразливості інформаційної безпеки розрізняються по способу їх реалізації.

Виходячи з цього можна виділити наступні основні класи загрози безпеки, спрямованих проти інформаційних ресурсів:

- загрози, реалізовані або впливом на програмне забезпечення та конфігураційну інформацію системи, або за допомогою некоректного використання системного та прикладного програмного забезпечення;

- загрози, пов'язані з виходом із ладу технічних засобів системи, приводячи до повного чи часткового руйнування інформації, зберігається та обробляється в системі;

- загрози, обумовлені людським фактором і пов'язані з некоректним використанням співробітників програмного забезпечення або з впливом на технічні засоби, у більшому ступені залежать від дій і "особливостей" моральної поведінки співробітників;

- загрози з використанням програмних засобів. Найближчий багаточисленний клас загроз конфіденційності, цілісності та доступності інформаційних ресурсів, пов’язаних з отриманням внутрішніх та зовнішніх налаштувачів логічного доступу до інформації з користувацькими можливостями, наданими загально системним та прикладним програмним забезпеченням. Більшість розглянутих у цьому класі загроз реалізується шляхом локальних чи видалених атак на інформаційні ресурси системи внутрішніх та зовнішніх наглядачів. Результатом здійснення цих загроз стає несанкціонований доступ до даних, керуючої інформації, що зберігається на робочому місці системи адміністратора, конфігураційної інформації технічних засобів, а також до інформації, переданої по каналам зв'язку.

У цьому класі виділяються наступні основні загрози:

- загрози технічними засобами;

- несанкціонований доступ до додатків;

- впровадження шкідливого програмного забезпечення;

- зловживання системними ресурсами;

- відмова від підтвердження авторства переданої інформації;

- збої системного та мережевого програмного забезпечення;

- збої прикладного програмного забезпечення.

Найбільш вразливі з точки зору захисту інформаційних ресурсів так названі критичні комп'ютерні системи. Під критичними комп'ютерними системами розглядають складні комп'ютеризовані організаційно - технічні та технічні системи, блокування порушення функціонування яких потенційно призводить до втрати контролю над власними системами, втрати здатності подолати загрозу, руйнування системи фінансового зв'язку, дезорганізації систем енергетичного та інформаційно - транспортного державного управління, глобальним екологічним і техногенним катастрофам.

При зміні способу зберігання інформації з паперового виду на цифровий, постало головне питання - як цю інформацію захистити, адже дуже багато факторів впливає на збереження конфіденційних даних. Щоб організувати безпечне зберігання даних, необхідно провести аналіз загроз для правильного проектування схем інформаційної безпеки.

Термінологія та підходи до класифікації. Організація забезпечення безпеки повинна мати комплексний характер і ґрунтуватися на глибокому аналізі можливих негативних наслідків. При цьому важливо не впустити будь-яких істотних аспектів. Аналіз негативних наслідків передбачає обов'язкову ідентифікацію можливих джерел загроз, факторів, що сприяють їхньому прояву і, як наслідок, визначення актуальних загроз безпеці інформації [2].

Під час такого аналізу необхідно переконатися, що всі можливі джерела загроз ідентифіковані та зіставлені з джерелами загроз усі можливі фактори (вразливості), властиві об'єкту захисту, всім ідентифікованим джерелам та факторам зіставлені загрози безпеці інформації.

Виходячи з даного принципу, моделювання та класифікацію джерел загроз та їх проявів, доцільно проводити на основі аналізу взаємодії логічного ланцюжка (рисунок 1.3):

Джерело загрози – фактор (вразливість) – загроза (дія) – наслідки (атака).

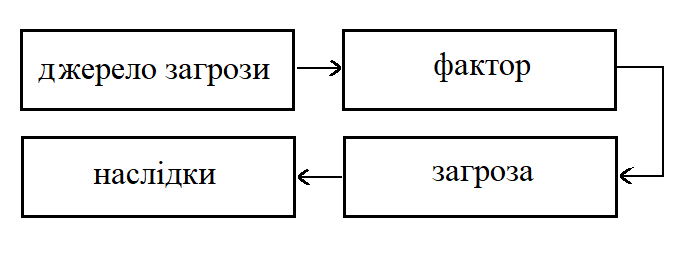


Рисунок 1.3 – Ланцюжок оцінки джерел загроз

Під цими термінами розуміється.

Джерело загрози – це потенційні антропогенні, технічні чи стихійні носії загрози безпеці.

Загроза (дія, Threat) - це можлива небезпека (потенційна або реально існуюча) вчинення будь-якого діяння (дія або бездіяльності), спрямованого проти об'єкта захисту (інформаційних ресурсів), що завдає шкоди власнику, працівнику або користувачу, що піддається небезпеці спотворення та втрати інформації.

Фактор (вразливість, Vulnerability) - це притаманні об'єкту інформатизації причини, що призводять до порушення безпеки інформації на конкретному об'єкті та зумовлені недоліками процесу функціонування об'єкта інформатизації, властивостями архітектури автоматизованої системи, протоколами обміну та інтерфейсами, що використовуються програмним забезпеченням та апаратною платформою, умовами експлуатації.

Наслідки (атака, Attack) – це можливі наслідки реалізації загрози (можливі дії) при взаємодії джерела загрози через наявні фактори (вразливості).

Як видно з визначення, атака – це завжди пара «джерело – фактор», що реалізує загрозу і призводить до шкоди. При цьому аналіз загроз передбачає проведення аналізу можливої шкоди та вибору методів встановлення загроз безпеці інформації.

Ефективне [управління вразливими](https://www.recordedfuture.com/vulnerability-management-prioritization/) означає перехід від підходу "виправляти все і завжди" – підходу, якого ніхто ніколи не зможе досягти, – до пріоритизації вразливостей на основі фактичного ризику. Хоча число вразливостей і загроз збільшується з кожним роком, дослідження показують, що більшість загроз націлені на одну і ту ж невелику частку вразливостей. Учасники загроз також працюють швидше – тепер між оголошенням нової вразливості і появою засобів, націлених на неї, в середньому проходить всього п'ятнадцять днів. Це має два наслідки:

Є лише два тижні, щоб виправити або виправити системи від нового вразливостей. Якщо ви не можете виправити ситуацію в ці терміни, складіть план щодо пом'якшення шкоди.

Якщо нова вразливість не буде використана протягом двох тижнів — трьох місяців, вона навряд чи коли-небудь буде використана, її виправлення може мати менший пріоритет.

Аналіз загроз допомагає виявити вразливості, які становлять реальний ризик для вашої організації, виходячи за рамки оцінки CVE, шляхом об'єднання даних внутрішнього сканування вразливостей, зовнішніх даних і додаткової інформації про властивості учасників загроз. Завдяки цим записам користувачі виявляють на 22 відсотки більше реальних загроз, перш ніж вони нададуть серйозний вплив.

## 1.6 Машинне навчання для поглибленого аналізу загроз

Обробка даних сьогодні здійснюється у таких масштабах, які потребують комплексної автоматизації. Об'єднуючи дані з безлічі різних типів джерел можна сформувати максимально достовірну картину. Методи машинного навчання в основному використовують чотири способи покращення збору та аналізу даних – структурувати дані за категоріями, аналізувати текст кількома мовами, надавати оцінки ризику та створювати прогнозні моделі:

Структурувати дані на сутності та події. Онтологія має відношення до того, як ми поділяємо концепції і як ми групуємо їх разом. В науці про дані, онтологію представляють категорії сутностей на основі їхніх імен, властивостей і відносин один до одного, що полегшує їх сортування по ієрархій наборів. Наприклад, клас, контейнер, процес, файл - це окремі об'єкти, які також підпадають під більш широку категорію "компоненти системи".

Якщо суті являють собою спосіб сортування фізично різних понять, то події сортують за перебігом часу. Записані майбутні події не залежать від часу в якому вони відбулися – щось на подобі "об'єкт записана у файл", "об'єкт запишуть у файл", "об'єкт записали у файл" і "об'єкт був записана у файл" - усі вони являються однією подією. Онтологія та події забезпечують зручний пошук за категоріями, дозволяючи аналітикам зосередитися на більш широкій картині, а не на необхідності вручну сортувати дані самостійно.

Структурування тексту на декількох мовах з допомогою обробки основної мови.

Завдяки обробці природної мови сутності і події можуть виходити за рамки простих ключових слів, перетворюючи неструктурований текст з джерел на різних мовах в структуровану базу даних.

Машинне навчання, що управляє цим процесом, може відділяти рекламу від основного контенту, класифікувати текст по категоріям, таким як проза, журнали даних або код і усувати неоднозначність між об'єктами з однаковими іменами (наприклад, "Apple", компанія і "яблуко", фрукти), використовуючи контекстні підказки в навколишньому тексті.

Таким чином, система може щоденно аналізувати текст з мільйонів документів на декількох різних мовах – завдання, для виконання якої потрібно непрактично велика і кваліфікована команда фахівців-аналітиків. Така економія часу допомагає командам ІТ-безпеки працювати на 32 відсотки ефективніше з записаним майбутнім.

Класифікувати події та об'єкти, допомагаючи аналітикам-фахівцям розставляти пріоритети попередження атак і загроз.

Машинне навчання і статистична методологія використовуються для подальшого сортування об'єктів і подій за важливістю — наприклад, шляхом присвоєння оцінок ризику шкідливих об'єктів.

Оцінки ризику розраховуються за допомогою двох систем: одна заснована на правилах, заснованих на людської інтуїції і досвіді, а інша заснована на машинному навчанні, навченому на вже перевіреному наборі даних. Класифікації, такі як оцінки ризику, надають як судження ("ця подія є критичною"), так і контекст, що пояснює оцінку ("оскільки кілька джерел підтверджують, що ця IP-адреса є шкідливим"). Автоматизація класифікації ризиків економить час аналітиків на сортуванні помилкових спрацьовувань і визначенні пріоритетів, допомагаючи співробітникам ІТ-служби безпеки, витрачати понад 40 відсотків менше часу на складання звітів.

Машинне навчання також може генерувати моделі, які передбачають майбутнє, часто набагато точніше, ніж будь-які людські аналітики, спираючись на глибокі масиви даних, раніше здобутих і класифікованих. Це особливо сильне застосування машинного навчання за "законом великих чисел" — по мірі того, як ми будемо продовжувати використовувати все більше джерел даних, ці прогнозуючі моделі будуть ставати все більш і більш точними.

Таким чином машинна обробка даних про загрози може значно скоротити час і витрати які використовуються для аналізу даних (рисунок 1.6.). Машинне навчання є більш універсальним і здатне вирішувати широке коло проблем.

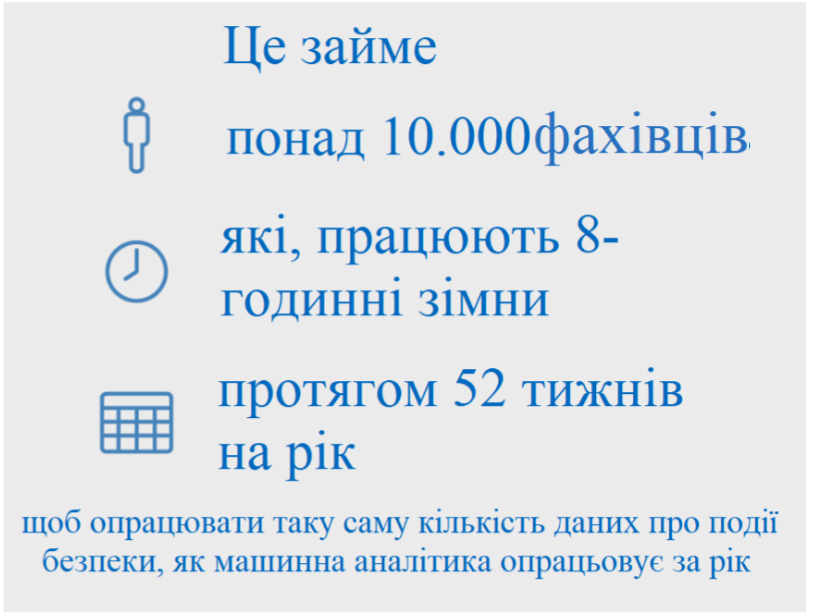


Рисунок 1.6 - Використання машинного аналізу загроз

# 2 ПОГЛИБЛЕНИЙ АНАЛІЗ БЕЗПЕКИ ВНУТРІШНІХ МЕРЕЖ НА ОСНОВІ FLOW-ПРОТОКОЛУ

## 2.1 Поглиблений аналіз загроз за допомогою Flow - протоколів

Коли мова заходить про поглиблений аналіз безпеки внутрішньої корпоративної або відомчої мережі, то у багатьох виникає асоціація з контролем витоків інформації та впровадженням DLP-рішень (DLP - Data Loss Prevention**)**. Якщо спробувати уточнити питання і запитати, як виявляти атаки у внутрішній мережі, то відповіддю буде, як правило, системи аналізу атак (intrusion analysis systems, IAS). І те, що було єдиним варіантом ще 10-20 років тому, то сьогодні стає анахронізмом. Існує більш ефективний, а місцями і єдиний можливий варіант аналізу та моніторингу внутрішньої мережі — використовувати flow-протоколи, спочатку призначених для пошуку мережевих проблем (troubleshooting), але з часом вони перетворилися в дуже цікавий інструмент безпеки. Ось про те, які flow-протоколи бувають і які з них краще допомагають виявляти мережеві атаки, де найкраще впроваджувати аналіз та моніторинг flow - протоколів, на що звернути увагу при розгортанні такої схеми, ми і поговоримо.

Розглядаючи дані для моніторингу інфраструктури на мережевому рівні можна виділити три ключові джерела (рисунок 2.1):

- "сирий" трафік, який захоплюється і подається на аналіз певним системам аналізу;

- події з мережевих пристроїв, через які проходить трафік;

- інформація про трафік, отримана по одному з flow-протоколів.

Захоплення сирого трафіку — найпопулярніший у фахівців з безпеки варіант, тому що він історично з'явився і самим першим. Звичайні мережеві системи виявлення атак (самою першою комерційною системою виявлення атак була NetRanger від компанії Wheel Group, куплена в 1998-му році Cisco) якраз і займалися захопленням пакетів (а пізніше і сесій), в яких шукали певні сигнатури ("вирішальні правила"), які сигналізують про атаки. Зрозуміло, аналізувати сирий трафік можна не тільки з допомогою IDS, але і за допомогою інших засобів (наприклад, Wireshark, tcpdump або функціонал NBAR2 в Cisco IOS), але їм зазвичай не вистачає бази знань, яка відрізняє засіб ІБ (інформаційної безпеки) від звичайного ІТ-інструменту.

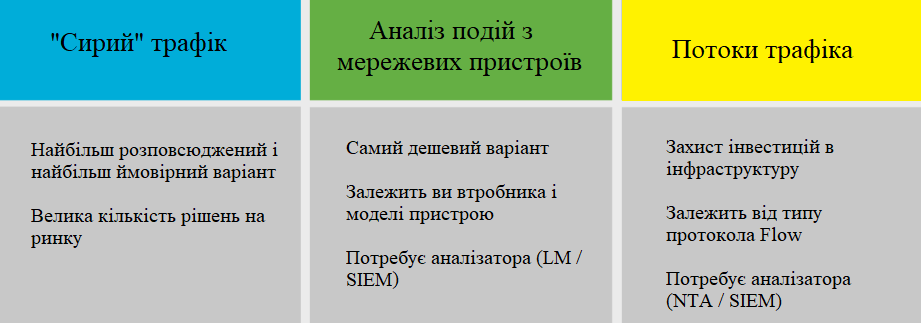


Рисунок 2.1- Три ключові джерела даних для моніторингу інфраструктури на мережевому рівні

Отже, системи виявлення атак. Самий старий і самий популярний метод виявлення мережевих атак, який непогано справляється зі своїм завданням на мережі, але відстає у сучасних комутованих та програмно - визначених мережах. У випадку з мережею, побудованою на базі звичайних комутаторів, інфраструктура сенсорів виявлення атак стає занадто великою — вам доведеться ставити по сенсору на кожне з'єднання з вузлом, атаки на який ви хочете моніторити. Будь який виробник, звичайно, буде радий продати вам сотні і тисячі сенсорів, але думаю ваш бюджет не витримає таких витрат. Можна сказати, що навіть в Cisco не змогли це зробити, хоча, здавалося б, питання ціни не стоїть перед ними, тому що це їхнє власне рішення. Крім того, виникає питання, а як підключати сенсор в такому варіанті? А якщо сам сенсор буде виведений з ладу? Вимагати наявності bypass-модуля в сенсорі? Використовувати розгалужувачі? Все це здорожує рішення і робить його непідйомним для компанії будь-якого масштабу.

Можна спробувати "повісити" сенсор на SPAN/RSPAN/ERSPAN-порт і направити на нього трафік з потрібних портів комутатора. Цей варіант частково знімає проблему, описану в попередньому абзаці, але ставить іншу — SPAN-порт не може прийняти абсолютно весь трафік, який в нього буде направлятися — йому не вистачить пропускної спроможності. Прийдется чимось жертвувати. Або частина вузлів залишити без моніторингу (тоді попередньо треба виставити вузлам пріоритети), або направляти не весь трафік з мережі, а лише певного типу. У будь-якому випадку можна пропустити якісь атаки. Крім того, SPAN-порт може бути зайнятий під інші потреби. У підсумку, доведеться переглянути існуючу топологію мережі і внести, в неї корективи, щоб охопити по максимуму вашу мережу наявними у вас числом сенсорів (і узгодити це з ІТ).

А якщо мережа використовує асиметричні маршрути? А якщо впроваджений або планується до впровадження SDN (Software Defined Network програмно - конфігурована мережа)? А якщо потрібно моніторити віртуальні машини або контейнери, трафік яких взагалі не доходить до фізичного комутатора? Ці питання не люблять виробники традиційних IDS, тому що не знають, як відповісти на них. Можливо, вони будуть схиляти вас до того, що всі ці модні технології – хайп і він вам не потрібен. Можливо, вони будуть говорити про необхідність почати з малого. А може бути вони скажуть, що вам треба поставити потужну мясорубку центр мережі і направити весь трафік у неї з допомогою балансувальників. Який би варіант не пропонували, потрібно самим чітко зрозуміти, наскільки він підходить. І тільки після цього приймати рішення про вибір підходу до моніторингу ІБ мережевої інфраструктури. Повертаючись же до захоплення пакетів, цей метод продовжує залишатися дуже популярним і важливим, але його основне призначення — контроль кордонів; кордонів між вашою організацією та мережею Інтернет, кордонів між ЦОД (Центру обчислення даних) та іншою мережею, кордонів між АСУТП ([Автоматизована система керування технологічним процесом](https://uk.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B8%D0%BC_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D0%BC)) і корпоративним сегментом. У цих місцях класичні IDS/IPS рішення, мають право на існування і непогано справляються з поставленими завданнями.

Перейдемо до другого варіанту. Аналіз подій, що надходять з мережевих пристроїв, теж може бути використаний для цілей виявлення атак, але не як основний механізм, так як він дозволяє виявляти тільки невеликий клас вторгнень. До того ж йому властива деяка реактивність — атака спочатку має відбутися, потім вона повинна бути зафіксована мережевим пристроєм, який тим чи іншим способом буде сигналізувати про проблеми з ІБ. Способів таких кілька. Це може бути syslog, RMON або SNMP. Останні два протоколи для мережевого моніторингу в контексті ІБ застосовуються тільки якщо необхідно виявити DoS-атаку на саме мережеве обладнання, так як за допомогою RMON і SNMP можна, наприклад, відстежувати завантаження центрального процесора пристрою або його інтерфейсів. Це один з найбільш "дешевих" (syslog або SNMP є у всіх), але й найменш ефективний з усіх способів моніторингу ІБ внутрішньої інфраструктури – багато атак просто приховані від нього. Зрозуміло їм не треба нехтувати і той же аналіз syslog допомагає своєчасно ідентифікувати зміни в конфігурації самого пристрою, компрометацію саме його, але виявляти атаки на всю мережу він не дуже підходить.

Третій варіант – це аналіз інформації про трафік, що проходить через пристрій, який підтримує один з декількох flow-протоколів. В даному випадку, незалежно від протоколу, інфраструктура роботи з потоками обов'язково складається з трьох компонентів.

- Генерація або експорт flow. Ця роль звичайно покладається на маршрутизатор, комутатор або інше мережеве обладнання, яке, пропускаючи через себе мережевий трафік, дозволяє виділяти з нього ключові параметри, які потім передаються на модуль збору. Наприклад, в Cisco протоколу Netflow підтримується не тільки на маршрутизаторах і комутаторах, включаючи і віртуальні та промислові, але і на бездротових контролерів, міжмережевих екранах і навіть серверах.

- Збір flow. Враховуючи, що в сучасній мережі зазвичай більше одного мережевого пристрою, виникає завдання збору та консолідації потоків, яка вирішується за допомогою так званих колекторів, які проводять обробку отриманих потоків і потім передають їх для аналізу.

- Аналіз flow. Аналізатор бере на себе основні інтелектуальні завдання й, застосовуючи до потоків різні алгоритми, робить ті чи інші висновки. Наприклад, в рамках ІТ-функції такий аналізатор може виявляти вузькі місця в мережі або аналізувати профіль завантаження трафіку для подальшої оптимізації мережі. А для ІБ такий аналізатор може виявляти витоку даних, поширення шкідливого коду або DoS-атаки.

Не варто думати, що така триланкова архітектура занадто складна – всі інші варіанти (крім, системи моніторингу, що працюють з SNMP і RMON) також працюють згідно з нею. Генератор даних для аналізу, в якості якого виступає мережевий пристрій або окремо стоїть сенсор, система збору сигналів тривоги і система управління усією інфраструктурою моніторингу. Останні два компоненти можуть бути об'єднані в рамках одного вузла, але в більш-менш великих мережах вони зазвичай рознесені по двох, як мінімум, вузлах з метою забезпечення масштабування і надійності.

На відміну від аналізу пакетів, що базується на вивченні заголовка і тіла кожного пакету даних і складаються з цих сесій, аналіз потоків спирається на збір метаданих про мережевий трафік. Хто, коли, скільки, де, куди і як ось на ці питання, відповідає аналіз мережевої телеметрії з допомогою різних протоколів flow (рисунок 2.2.). Спочатку вони використовувалися для аналізу статистики та пошуку ІТ-проблем в мережі, але потім, у міру розвитку аналітичних механізмів, їх стало можливим застосовувати до тієї ж телеметрії і для цілей безпеки.

Тут варто ще раз зазначити, що аналіз потоків не замінює і не скасовує захоплення пакетів. Кожен з цих методів мають свою область застосування. Але в контексті даної статті, саме аналіз потоків краще всього підходить для моніторингу внутрішньої інфраструктури. У вас є мережеві пристрої (і не важливо, працюють вони в програмно-визначається парадигмі або згідно з статичними правилами), які атака оминути не може. Сенсор класичної IDS вона може обійти, а мережеве пристрій, що підтримує flow-протокол, немає. У цьому перевага цього методу.

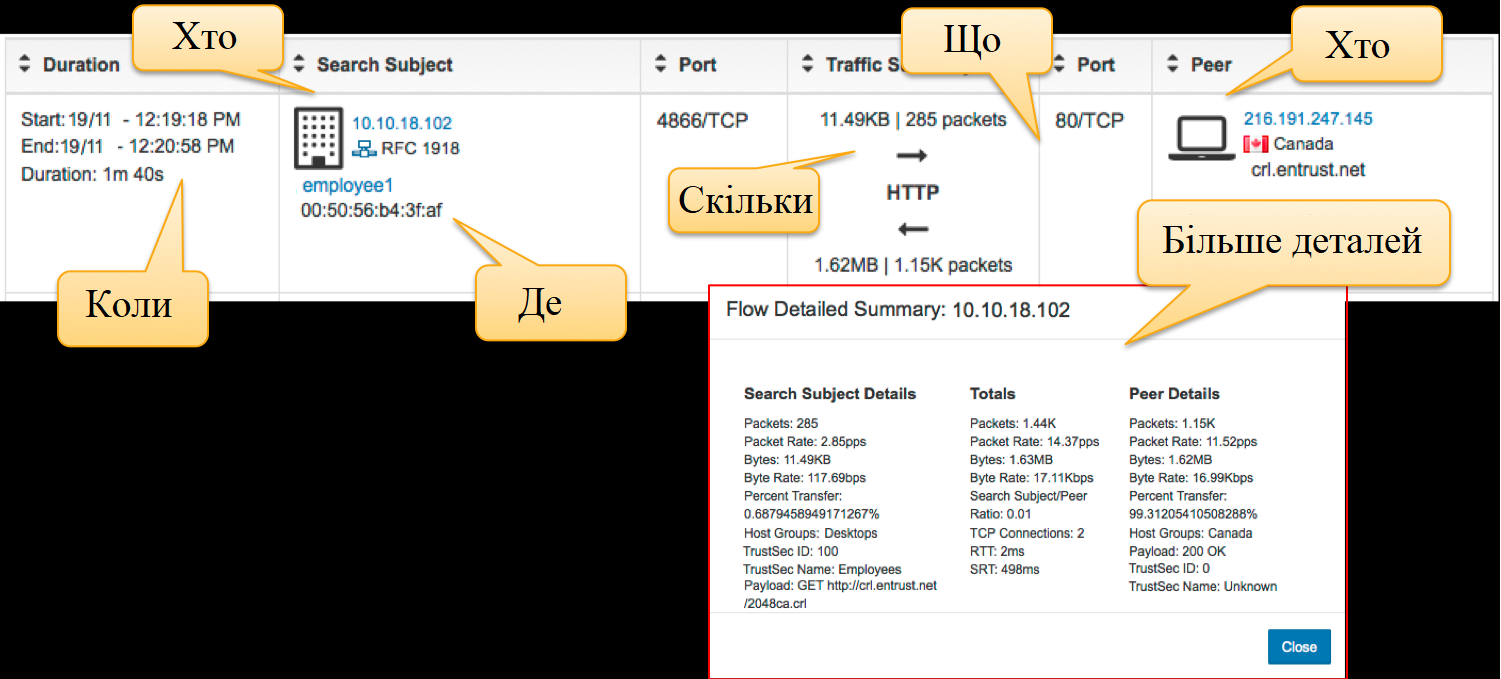


Рисунок 2.2 – Аналіз мережевої телеметрії з допомогою протоколів flow

З іншого боку, якщо потрібна доказова база для правоохоронних органів або власної групи розслідування інцидентів, без захоплення пакетів у цьому випадку не обійтися – мережева та системна телеметрія не є копією трафіку, який можна використовувати при зборі доказів; вона потрібна для оперативного виявлення і прийняття рішень в галузі ІБ. З іншого боку, використовуючи аналіз телеметрії, можна "писати" не весь мережевий трафік, а лише той, який бере участь в атаці. Засоби аналізу телеметрії в цьому плані добре доповнюють традиційні механізми захоплення пакетів, даючи команду на вибіркове захоплення і зберігання. В іншому випадку вам доведеться мати колосальну інфраструктуру зберігання.

Уявімо мережа, що працює на швидкості 250 Мбіт/с. Якщо ви захочети зберігати весь цей обсяг, то вам знадобиться сховище на 31 Мб для однієї секунди передачі трафіку, 1,8 Гб – для однієї хвилини, 108 Гб – для однієї години, і 2,6 Тб – для однієї доби. Для зберігання добових даних з мережі з пропускною здатністю 10 Гбіт/сек вам знадобиться сховище на 108 Тб. Але ж деякі регулятори вимагають зберігати дані з безпеки роками. Це потребує колосального сховища даних. Запис «на вимогу», яку вам допомагає реалізувати аналіз потоків, допомагає скоротити ці значення на порядки. До речі, якщо говорити про співвідношення обсягу записуваних даних мережевої телеметрії та повного захоплення даних, то воно складає приблизно 1 до 500. Для тих же наведених вище значень, зберігання повної розшифровки всього денного трафіку складе 5 і 216 Гб відповідно (можна навіть на звичайну флешку записати).

Якщо у засобів аналізу "сирих" мережевих даних метод їх захоплення майже не відрізняється від вендора до вендору, то у випадку з аналізом потоків ситуація інша. Існує кілька варіантів flow-протоколів, про відмінності у яких необхідно знати саме в контексті безпеки. Найпопулярнішим є протоколу Netflow, розроблений компанією Cisco. Існує кілька версій цього протоколу, що відрізняються за своїми можливостями та об'ємом записуваної інформації про трафік. Поточна версія – дев'ята (Netflow v9), на базі якої був розроблений промисловий стандарт Netflow v10, також відомий як IPFIX. Сьогодні більшість мережевих вендорів підтримує саме Netflow або IPFIX в своєму обладнанні. Але є і різні інші варіанти flow-протоколів – SysFlow, sFlow, jFlow, cFlow, rFlow, NetStream і т. п., з яких більш популярним є sFlow. Саме він найчастіше підтримується вітчизняними виробниками мережного обладнання зважаючи на простоту реалізації.

У чому ключові відмінності між ними, які стандарти вони використовують і в яких випадках вони знадобляться більше всього. По-перше, Netflow має поля, які настроюються користувачем навідміну від фіксованих полів у sFlow. А по-друге, і це найголовніше в нашому випадку, sFlow збирає так звану вибіркову телеметрію, на відміну від невибіркової у Netflow і IPFIX. В чому ж між ними різниця? На відміну від них SysFlow - це новий формат системної телеметрії і набір інструментів для моніторингу поведінки системи для масштабованої аналітики безпеки, відповідності вимогам і продуктивності. SysFlow кодує подання системних дій в компактний формат, який записує, як програми взаємодіють зі своїм середовищем. Він пов'язує поведінку процесів з діями по доступу до мережі і файлів, надаючи більш багатий контекст для аналізу.

Вирішивши ознайомитися з книгою можна виділити три варіанти досягнення поставленої мети – прочитати книгу повністю, пробігти її очима, зупиняючись на кожній п’ятій чи десятій сторінці, або спробувати знайти переказ ключових понять та концепцій у якомусь блозі або пошуковому сервісі. Так ось глибока телеметрія — це читання кожної сторінки мережевого трафіку, тобто аналіз метаданих по кожному пакету. Вибіркова телеметрія — це вибіркове вивчення трафіку в надії, що в обраних варіантах виявиться те, що вам потрібно. В залежності від швидкості каналу вибіркова телеметрія буде віддавати для аналізу кожен 64-ий, 200-ий, 500-ий, 1000-ий, 2000-ий або навіть 10000-ий пакет (таблиця 2.1) [24].

Таблиця 2.1 - Кількість пакетів які аналізуються в залежності від швидкості каналу.

|  |  |
| --- | --- |
| Швидкість каналу | Вибірка |
| 10 мбіт/с | 1 з 200 |
| 100 мбіт/с | 1 з 500 |
| 1 гбіт/с | 1 з 1000 |
| 10 гбіт/с | 1 з 10000 |

У основі моніторингу ІБ це означає, що вибіркова телеметрія добре підходить для виявлення DDoS-атак, сканування, поширення шкідливого коду, але може пропустити атомарні або багатопакетні атаки, не потрапили в вибірку, відправлену для аналізу. У глибокій телеметрії таких недоліків немає і з її допомогою спектр виявлення атак набагато ширший. Ось невеликий перелік подій, які можна виявляти за допомогою засобів аналізу мережевої телеметрії. Зрозуміло, що який-небудь open source аналізатор Netflow вам цього не дозволить, так як його основна задача — зібрати телеметрію і провести над нею базовий аналіз з точки зору ІБ. Для виявлення на базі flow загроз ІБ необхідно оснастити аналізатор різними движками і алгоритмами, які і будуть на базі стандартних або користувальницьких полів Netflow виявляти проблеми кібербезпеки, збагачувати стандартні дані зовнішніми даними від різних джерел Threat Intelligence (розвідка загроз) і т. п. Тому якщо у вас є вибір, то зупиняйте його на Netflow або IPFIX. Але навіть якщо ваше обладнання працює тільки з sFlow, як у вітчизняних виробників, то навіть у цьому випадку ви можете отримати з нього користь в контексті безпеки.

Наступне питання, – де впроваджувати підтримку flow для цілей безпеки? Насправді питання поставлено не зовсім коректно. На сучасному обладнанні підтримка flow-протоколів є майже завжди. Тому питання можна перефразувати інакше – де найефективніше збирати телеметрію з точки зору безпеки? Відповідь буде досить очевидною — на рівні доступу, де можна побачити 100% всього трафіку, де буде детальна інформація за хостами (MAC, VLAN, ID інтерфейсу), де можна відстежувати навіть P2P-трафік між хостами, що критично для виявлення сканування та поширення шкідливого коду. На рівні ядра частина трафіку може просто не відображатися, а на рівні периметра можна побачити навіть не чверть усього мережевого трафіку. Але якщо з якихось причин в мережі завелися сторонні пристрої, які дозволяють зловмисникам "входити і виходити", минаючи периметр, то аналіз телеметрії з нього вам нічого не дасть. Тому для максимального охоплення рекомендується включати збір телеметрії саме на рівні доступу. При цьому, варто відзначити, що навіть якщо говорити про віртуалізацію або про контейнери, то в сучасних віртуальних комутаторах також часто зустрічається підтримка flow, що дозволяє контролювати трафік і там.

Але що робити якщо все-таки обладнання, фізичне або віртуальне, не підтримує flow-протоколи? Або його включення заборонено (наприклад, у промислових сегментах для забезпечення надійності)? Або його включення призводить до високого завантаження центрального процесора (таке буває на застарілому обладнанні)? Для вирішення цього завдання існують спеціалізовані віртуальні сенсори (flow sensor), які по суті є звичайними розгалужувачами, що пропускають через себе трафік і транслюють його у вигляді потоків на модуль збору. Правда, в цьому випадку можна отримати проблеми, які призводять до захоплення пакетів. Тобто треба розуміти не тільки переваги технології аналізу потоків, але і її обмеження.

Ще один момент, який важливо пам'ятати, говорячи про засоби аналізу потоків стосується до звичайних засобів генерації подій безпеки застосовується метрика EPS (event per second, подій в секунду), то до аналізу телеметрії цей показник непридатний, він замінюється на FPS (flow per second, потоків в секунду). Як і у випадку з EPS, вирахувати заздалегідь його не можна, але можна оцінити приблизну кількість потоків, що генерує той або інший пристрій в залежності від його завдання. В Інтернеті можна знайти таблиці з приблизними значеннями для різних типів корпоративних пристроїв і умов, що дозволить вам оцінити, які ліцензії вам потрібні для засобів аналізу і яка буде їх архітектура. Справа в тому, що сенсор IDS обмежений певною пропускною здатністю, яку він "витягне", так і колектор потоків має свої обмеження, які треба розуміти. Тому у великих, територіально-розподілених мережах колекторів зазвичай кілька. Наприклад для моніторингу мережі всередини Cisco число колекторів – 21 [15]. І це на мережу, розкидану по п'яти материках і налічує близько півмільйона активних пристроїв.

Збору даних про потоки зазвичай виділяють найменше уваги з усіх етапів типової настройки моніторингу потоків, хоча не оптимальне налаштування може призвести до серйозної, і часто непомітної втрати даних. Що ускладнює визначення розмірів колектора потоку, так це те, що в аномальних ситуаціях кількість записів вхідного потоку може бути подвоєна або навіть більше. Тому необхідні значні надлишкові налаштування та калібрування, щоб гарантувати, що дані не будуть втрачені. Втрата даних як частина перевантаження колектора потоку може бути наслідком переповнення як ядра, так і буфера програми, а також перевантаження дискового вводу-виводу.

Буфери ядра зберігають повідомлення до того, як їх отримує процес збору. Буфери програми є частиною самого колектора потоків і можуть використовуватися для будь-якої проміжної обробки перед збереженням даних потоку. Буфери можна налаштувати до певної міри; збільшення розмірів буфера дозволяє тимчасово зберігати більше даних, але буде марно, якщо наступні елементи системи стають вузьким місцем.

На практиці дисковий ввід - вивід буде вузьким місцем у таких ситуаціях, тому збільшення буферів дає лише обмежені переваги. Багато збирачів потоків застосовують архівування даних до даних потоку за замовчуванням. Увімкнення архівування даних залежить від ємності обробки та зберігання системи, яка виконує функцію колектора потоку. Як правило, можна порівняти час, необхідний для зберігання блоку даних потоку як у архівованому, так і в неархівованому форматі. Якщо запис архівованих даних відбувається швидше, підсистема зберігання стає вузьким місцем у системі збору даних, і архівування даних, яке вимагає інтенсивного використання ЦП, має бути увімкнена. В іншому випадку, якщо запис не архівованих даних відбувається швидше, вузьким місцем є потужність обробки, і архівування даних слід вимкнути. Це емпіричне правило підтверджується nftest, що є частиною nfdump, який виконує ці тести автоматично та надає пораду щодо того, вмикати чи ні архівування даних.

## 

## 2.2. Інструменти для аналізу Flow-протоколів

В якості системи аналізу загроз можна використати рішення Cisco Stealthwatch (рисунок 2.3) [30], яке спеціально орієнтоване на розв'язання завдань безпеки. Це одне найбільш повнофункціональних рішень для моніторингу та аналізу мережевого трафіку, виявлення мережевих загроз та реагування на них, в якому використовуються дані корпоративної телеметрії з наявної мережевої інфраструктури. Це рішення підвищує ефективність виявлення загроз, прискорює реагування на них і полегшує сегментацію мережі завдяки багаторівневому машинному навчанню і моделюванню об'єктів.

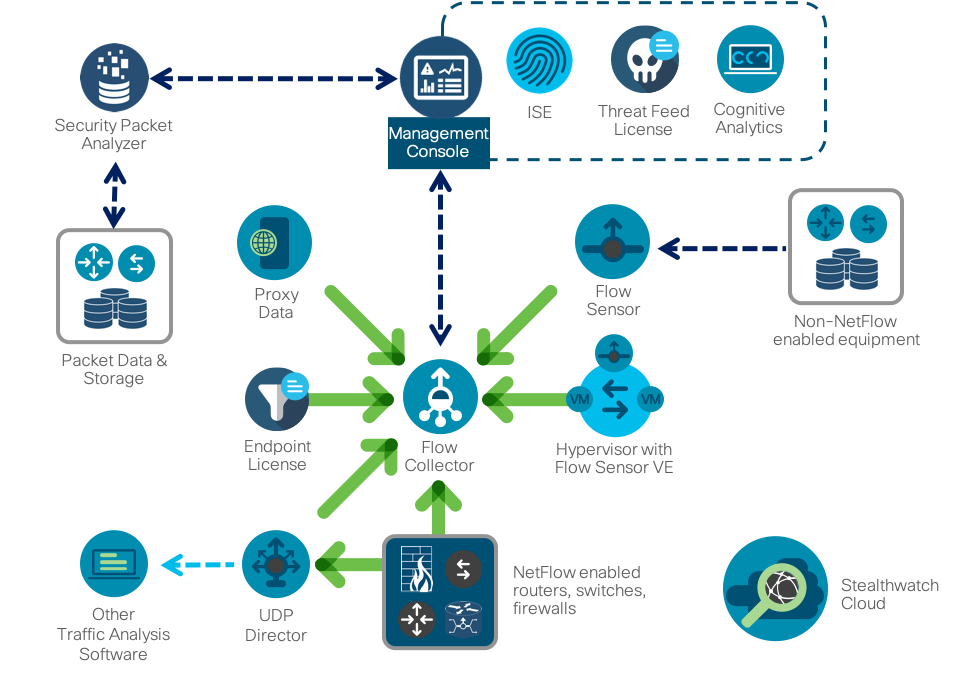
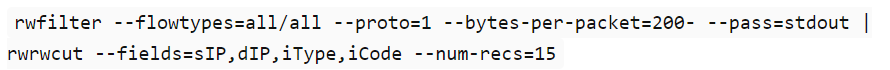


Рисунок 2.3 – Архітектура Cisco Stealthwatch Enterprise

За допомогою розширеної аналітики подій ви завжди знатимете, хто і що робить у вашій мережі. У нього є багато вбудованих механізмів виявлення аномальної, підозрілої, і явно шкідливої активності, що дозволяє виявляти широкий спектр різних загроз – від криптомайнинга до витоків інформації, від поширення шкідливого коду до шахрайства. Як і більшість аналізаторів потоків Stealthwatch побудований за трирівневою схемою генератор – колектор – аналізатор, але він доповнений низкою цікавих особливостей, які важливі в контексті розглянутого матеріалу.

По-перше, він інтегрується з рішеннями по захопленню пакетів (наприклад, Cisco Security Packet Analyzer) [27], що дозволяє записувати вибрані мережеві сесії для подальшого глибокого розслідування та аналізу. По-друге, спеціально для розширення завдань безпеки був розроблений спеціальний протокол nvzFlow, який дозволяє "транслювати" активність додатків на кінцевих вузлах (серверах, робочих станціях і т. п.) в телеметрію і передавати її на колектор для подальшого аналізу. Якщо у своєму початковому варіанті Stealthwatch працює з будь яким flow-протоколом (sFlow, rFlow, Netflow, SysFlow IPFIX, cFlow, jFlow, NetStream) на рівні мережі, то підтримка nvzFlow дозволяє проводити кореляцію даних ще і на рівні вузлів, тим самим. підвищуючи ефективність всієї системи і відображати більше атак, ніж звичайні мережеві аналізатори потоків. Зрозуміло, що говорячи про системи аналізу flow - потоків з точки зору безпеки, ринок не обмежений єдиним рішенням від Cisco. Можна використовувати як комерційні, так і безкоштовні, або умовно безкоштовні рішення.

SiLK – це набір інструментів (the System for Internet-Level Knowledge)[49] для аналізу трафіку, розроблений американським CERT/CC і який підтримує, в контексті сьогоднішньої статті, SysFlow, Netflow, IPFIX, sFlow, jFlow, cFlow і з допомогою різних утиліт (rwfilter, rwcount, rwflowpack та ін) проводити різні операції над мережевою телеметрією з метою виявлення в ній ознак несанкціонованих дій. Але треба відзначити кілька важливих моментів. SiLK – це інструмент командного рядка і проводити оперативний аналіз, весь час вводячи команди виду (виявлення ICMP-пакетів розміром більше 200 байт), що не є зручно:



Графічний інтерфейс SiLK можна використовувати, але він не але він не сильно поможе, оскільки вирішує лише функцію візуалізації, а не замінює аналітика. На відміну від комерційних рішень, у які вже закладена солідна аналітична база, алгоритми виявлення аномалій, workflow і т. п., відсутні у випадку з SiLK все це потрібно будете зробити самостійно, що потребує від трохи більших компетенцій, ніж від використання вже готового до роботи інструментарію. Це недобре і непогано – це є особливістю майже будь-якого безкоштовного інструменту, який виходить з того, що ви знаєте, що і робите, він вам тільки допоможе в цьому (комерційний інструментарій менш залежний від компетенцій його користувачів, хоча теж припускає, що аналітики розуміють хоча б основи проведення мережевих розслідувань та моніторингу). Але повернемося до SiLK. Алгоритм роботи аналітика з ним виглядає наступним чином:

- Формулювання гіпотези. Користувач повинен розуміти, що буде шукати всередині мережевої телеметрії, знати унікальні атрибути, за якими будемо виявляти ті чи інші аномалії або загрози.

- Побудова моделі. Сформулювавши гіпотезу, користувач програмує її з допомогою того ж Python, Shell або інших інструментів, що не входять в SiLK.

- Тестування. Настає черга перевірки правильності гіпотези, яка підтверджується або спростовується за допомогою утиліт SiLK, що починаються з 'rw', 'set', 'bag'.

- Аналіз реальних даних. В промисловій експлуатації SiLK допомагає виявляти щось і аналітик повинен відповісти на питання «чи знайшли ми те, що передбачали?», «Чи це відповідає нашій гіпотезі?», «Як знизить число помилкових спрацьовувань?», «Як поліпшити рівень розпізнавання?» і т. п.

- Поліпшення. На фінальному етапі покращується зроблене раніше — створюються шаблони, поліпшується і оптимізується код, переформульовується і уточнюється гіпотеза та ін.

Цей алгоритм можна застосувати до будь-якого аналізатору трафіку, зменшивши кількість помилок аналітика і підвищуючи оперативність виявлення інцидентів. Наприклад, у SiLK мережеву статистику можна збагатити зовнішніми даними по шкідливим IP за допомогою власноруч написаних скриптів, а в Cisco Stealthwatch – це вбудована функція, яка відразу відображає сигнал тривоги, якщо в мережевому трафіку зустрічається взаємодія з IP-адресами з чорного списку.

Також можна використати ПЗ аналізу flow ELK[44], яке є умовно безкоштовне і складається з трьох ключових компонентів – Elasticsearch (індексація, пошук та аналіз даних), Logstash (ввід/вивід даних)і Kibana (візуалізація). На відміну від SiLK, де все доводиться писати самому, у ELK вже є багато готових бібліотек/модулів (частина платні, частина - ні), які автоматизують аналіз мережевої телеметрії. Наприклад, фільтр GeoIP в Logstash дозволяє зв'язати спостережувані IP-адреси до їх географічного місця розташування (у того ж Stealthwatch це вбудована функція).

У ELK також досить багато користувачів, які дописують необхідні компоненти для цього рішення з моніторингу. Наприклад, щоб працювати з Netflow, IPFIX, sFlow або SysFlow можна скористатися модулем elastiflow, потрібен не тільки Sysflow.

Даючи більше оперативності збору flow-даних і пошук в ньому, у ELK на поточний момент відсутня багата вбудована аналітика щодо виявлення аномалій і загроз мережевої телеметрії. Тобто, вам доведеться самостійно описувати моделі порушень і потім вже користуватися ним в бойовій системі (вбудованих моделей там немає).

Є звичайно і більш наворочені розширення для ELK, в яких вже закладені деякі моделі виявлення аномалій в мережевій телеметрії, але такі розширення коштують грошей і тут вже питання писати аналогічну модель самому, купувати її реалізацію для свого засобу моніторингу або купити готове рішення класу Network Traffic Analysis (рисунок 2.4.) [46].

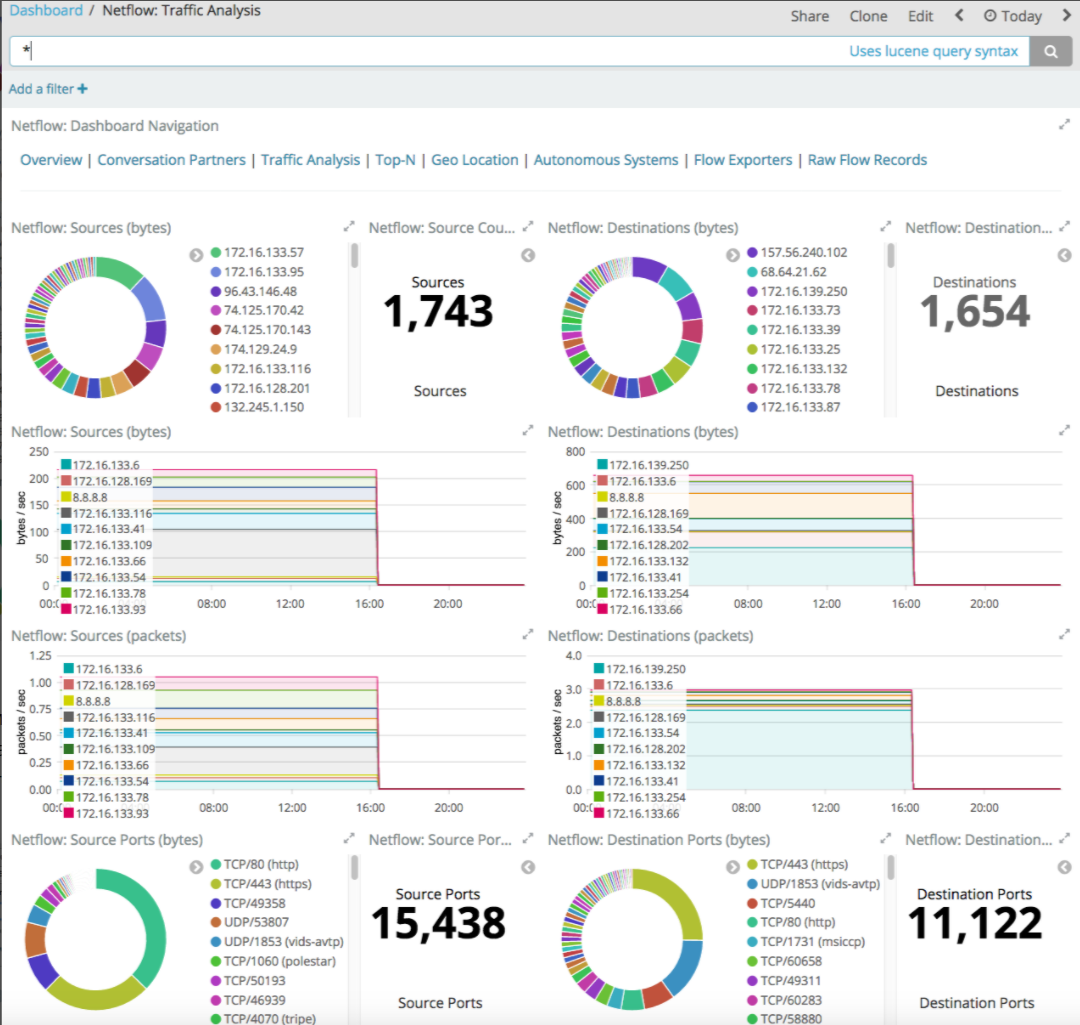


Рисунок 2.4 – Network Traffic Analysis як готове рішення для моніторингу мережі

## 

## 2.3 SysFlow як засіб моніторингу системи та мережі

Основою конвеєра телеметрії є новий формат даних, називається SysFlow, який перетворює необроблену інформацію про системні події в поняття, що описує поведінку процесів та їх взаємозв'язку з контейнерами, файлами та мережами. Цей об'єктно-реляційний формат дуже компактний, але при цьому забезпечує широку видимість в хмарах контейнерів. Також було створено декілька API, які дозволяють користувачам обробляти SysFlow з допомогою їх улюблених наборів інструментів.

SysFlow – це відкрита специфікація для телеметрії на рівні системних подій. Основна мета SysFlow - створити стандартний і розширений формат даних для аналізу безпеки та продуктивності обчислювальних робочих навантажень. Відкритий стандарт дозволить дослідникам і практикам легше працювати над спільним форматом даних і зосередитися на аналізі з використанням програмного забезпечення з відкритим вихідним кодом.

SysFlow розроблений з нуля для інтеграції, як з платформами з відкритим вихідним кодом, так і з платформами комерційної аналітики з допомогою відкритого формату серіалізації та пов'язаних бібліотек. ІТ-організації також можуть використовувати набір повторно використовуваних компонентів і API-інтерфейсів, щоб спростити розгортання датчиків телеметрії. Крім того, формат і бібліотеки відкритої серіалізації SysFlow забезпечують інтеграцію з платформами з відкритим вихідним кодом (наприклад, ELK, Apache Spark, scikit-learn) та користувацькими аналітичними микросервисами.

SysFlow – це компактний відкритий формат даних, який перетворює представлення системної діяльності на об’єктно-реляційне відображення, що фіксує, як програми взаємодіють з їх середовищем — аналогічно тому, як NetFlow підсумовує мережеві комунікації. Однак, на відміну від NetFlow, який фіксує лише мережеві взаємодії, SysFlow пов’язує поведінку мережі з процесами та інформацією про доступ до файлів, забезпечуючи більш багатий контекст для аналізу. Цей додатковий контекст сприяє глибшому аналізу ланцюжків знищення атак, який дає менше помилкових спрацьовувань і вищий рівень виявлення, ніж традиційні підходи, засновані на мережі.

Основна мета SysFlow полягає в тому, щоб перевести необроблені дані системних викликів в більш семантичні процеси, що сприяють значному скороченню обсягу даних для довгострокового зберігання даних криміналістичної експертизи, що має вирішальне значення для аналізу безпеки. За допомогою об'єктно-реляційної моделі сутностей, подій і потоків ми дозволяємо користувачам SysFlow налаштовувати необхідну ступінь деталізації збору і фільтрації даних, щоб полегшити більшість типів аналізу в рамках великих даних.

Також можна планувати та проводити докладний аналіз мережевого трафіку, а також отримувати різні часові звіти про використання ресурсів пропускної спроможності за допомогою декількох кліків мишки. Накопичення даних дозволить визначати пікові періоди пропускної спроможності та коригувати політики для більшої керованості процесів у вашій мережі. У ряді випадків це дозволить заощадити на підвищенні пропускної спроможності.

Агент SysFlow може бути розгорнутий в конфігураціях експорту S3 (пакетна обробка) або rsyslog (обробка подій) (рисунок 2.5) [13]. У пакетної конфігурації SysFlow експортує зібрану телеметрію у вигляді файлів трасування (пакетів записів SysFlow) у будь-яку службу зберігання об'єктів, сумісну з S3.

У конфігурації пакетної обробки SysFlow експортує зібрану телеметрію у вигляді подій, що передаються в складальник rsyslog. Це розгортання дозволяє створювати власні крайні конвеєри і пропонує вбудований механізм політики для фільтрації, збагачення та оповіщення про записи SysFlow.

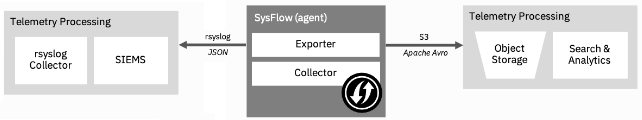


Рисунок 2.5 – Варіанти запуску датчиків SysFlow

# 3 ПОГЛИБЛЕНИЙ АНАЛІЗ ЗАГРОЗ З ВИКОРИСТАННЯМ SYSFLOW

## 3.1 Основні елементи SysFlow

Сутності (елементи) – це компоненти системи, які задіяні в моніторингу. До них відносяться контейнери, процеси і файли. Також підтримується спеціальний об'єкт сутності, званий заголовком, у якому зберігається інформація про версії SysFlow, і унікальний ідентифікатор, який представляє хост або віртуальну машину, яка контролюється експортером SysFlow. Заголовок завжди є першим записом, що з'являється у файлі SysFlow. Всі інші сутності містять позначки часу ідентифікатор об'єкта і стан. Мітка часу використовується для вказівки часу, в який об'єкт був експортований файл SysFlow.

Ідентифікатори об'єктів дозволяють подіям і потокам посилатися на об'єкти без дублювання інформації, що зберігається в кожному записі. Ідентифікатори об'єктів не обов'язково повинні бути глобально унікальними в просторі і часі. Фактично, єдиною вимогою унікальності є те, що ніякі два об'єкти, керовані експортером SysFlow, не можуть мати один і той же ідентифікатор одночасно. Сутності завжди записуються у двійковий вихідний файл перед будь-якими подіями, і потоки, пов'язані з ними, експортуються. Оскільки сутності експортуються першими, кожна подія і потік зіставляються з сутністю (з тим же ідентифікатором), яка знаходиться найближче до нього у файлі. Крім того, кожний двійковий вихідний файл повинен бути автономним, що означає, що всі сутності, на які посилаються потоки/події, повинні бути присутніми в кожному створеному файлі SysFlow.

Стан – це перерахування, вказує, чи об'єкт був записаний на диск. Стан може мати одне із трьох значень: створений, модифікований, повторюваний. Кожна сутність визначена нижче з рекомендаціями по використанню ідентифікаторів об'єктів на основі того, що використовується в поточній реалізації експортера SysFlow.

Сутність заголовок – це об'єкт, який відображається в початку кожного двійкового файлу SysFlow. Він містить поточну версію SysFlow, підтримувану в файлі, і ідентифікатор експортера.

Сутність процес являє собою запущений процес у системі. Він містить важливу інформацію про процеси, включаючи pid (ідентифікатор процесу) хоста, час створення, oid (ідентифікатор об'єкта), а також посилання на батьківський ідентифікатор. Коли об'єкт процесу експортується в файл SysFlow, всі його батьківські процеси повинні бути експортовані до процесу, а також в об'єкт - контейнер процесу. Процеси експортуються в файл SysFlow тільки в тому випадку, якщо експортується подія або потік, пов'язаний з цим процесом або будь яким з його потоків. Потоки явно не експортуються в об'єкт процесу, але представлені в подіях і потоках через поле ідентифікатора потоку. Нарешті, сутність процесу необхідно експортувати у файл тільки один раз, якщо вона не була змінена подією або потоком.

Сутність файл являє собою файлові ресурси в системі, включаючи файли, каталоги, сокети unix і канали.

Події являють собою важливі індивідуальні моделі поведінки об'єкта, які виділяються самі по собі через їх важливості, рідкісності або тому, що важливо підтримувати порядок операцій. Щоб управляти подіями і їх різними атрибутами, вони поділяються на три різні категорії: події процесів(PE Process Events), файлові події (FE File Events) і мережеві події (NE Network Events). Вони будуть описані більш докладно пізніше. Кожна подія і потік містять ідентифікатор об'єкта процесу, мітку часу, ідентифікатор потоку і набір прапорів операцій. Ідентифікатор об'єкта становить сутність процесу, в якому сталася подія, в той час, як ідентифікатор потоку вказує який процесу був пов'язаний з подією.

Ідея SysFlow полягає в тому, щоб дозволити користувачеві налаштувати необхідну ступінь деталізації даних системного рівня на основі обмежень ресурсів та вимог до аналізу даних. Таким чином, поведінка може бути розбита на окремі події або об'єднана в більш дрібні агреговані об'ємні потоки. Поточна версія специфікації описує події і потоки в трьох ключових областях поведінки: Файли, Мережі і Процеси. На рисунку 3.1 [1] показані ці події і потоки з їх атрибутами і відносинами до сутностей, які більш детально описані в наступному підрозділі.

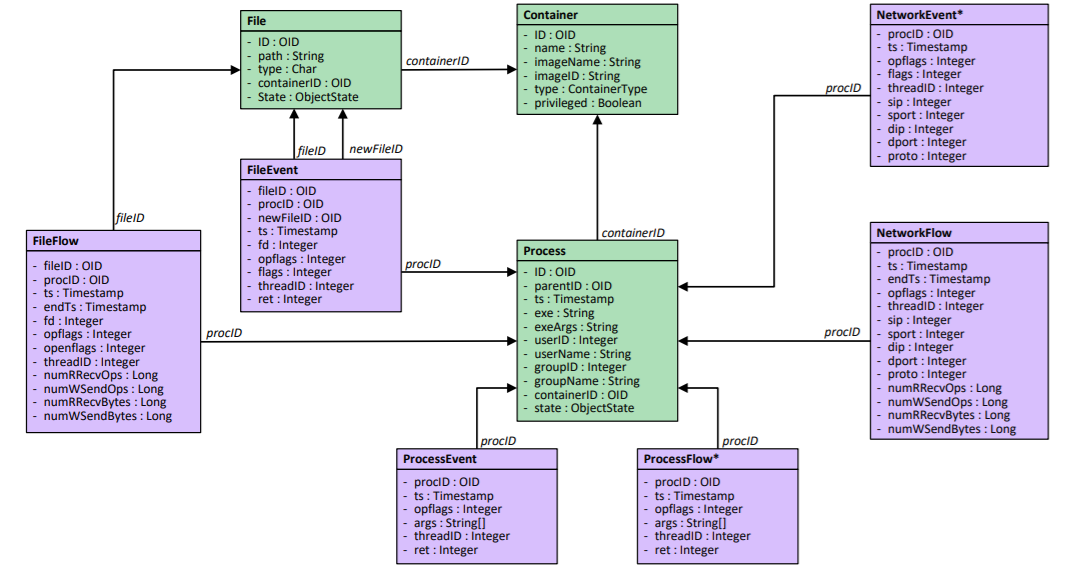


Рисунок 3.1 – Об'єктно-реляційна модель SysFlow

SysFlow кодує подання системних подій в компактний формат, який записує, як програми взаємодіють зі своїм середовищем. Він пов'язує поведінку процесів з діями, надає доступ до мережі і файлів, надаючи більш багатий контекст для аналізу. Цей додатковий контекст полегшує більш глибоке уявлення робочих навантажень вузлів і контейнерів та забезпечує потік варіантів використання захисту хмарних робочих навантажень, включаючи захист цілісності середовища виконання контейнерів, пошук загроз і криміналістичну експертизу. Хоча телеметрія інформації про системні події не нова, поточні монітори збирають дані з деталізацією системних викликів, генеруючи величезні обсяги даних, які обмежують аналітику простими підходами, заснованими на правилах. SysFlow різко знижує швидкість збору даних на порядки і перетворює події в поведінку, яка дозволяє застосовувати криміналістичні програми та комплексні підходи до аналізу.

Тим не менш, SysFlow дозволяє ІТ-організаціям застосовувати аналітику з допомогою графічної візуалізації до поверхневих шаблонів, що виходить за рамки порівняно простого підходу, заснованого на правилах. Наприклад, підхід SysFlow дозволить легше виявити взаємозв'язок між різними подіями, складовими атаки на кібербезпеку, а також згодом визначити, які контрзаходи слід використовувати для створення відповідної реакції на ланцюжок знищення атаки. Це також повинно істотно знизити ступінь втоми команд кібербезпеки від відстеження хибно - позитивних попереджень. ІТ-організації також можуть використовувати набір повторно використовуваних компонентів і API-інтерфейсів, щоб спростити розгортання датчиків телеметрії. ІТ-організації також можуть скористатися перевагами розширювального механізму політики, який може використовувати настроювані політики безпеки, описані декларативною мовою введення, які потім можна звірити з записами, отриманими SysFlow.

## 3.2 Інструменти SysFlow для поглибленого аналізу загроз

Конвеєр телеметрії SysFlow – це основа для [моніторингу](https://securityonline.info/terraform-aws-secure-baseline/) хмарних робочих навантажень і для створення аналітики продуктивності і безпеки. Мета цього проекту - створити все необхідне для системної телеметрії, щоб користувачі могли зосередитися на написанні та спільному використанні аналітики на масштабованої загальній платформі з відкритим вихідним кодом. Основою конвеєра телеметрії є новий формат даних, званий SysFlow, який перетворює вихідну інформацію про системні події в абстракцію, що описує поведінку процесів та їх взаємозв'язку з контейнерами, файлами та мережами. Цей об'єктно-реляційний формат дуже компактний, але при цьому забезпечує широку видимість в хмарах контейнерів. Ми також створили кілька API, які дозволяють користувачам обробляти SysFlow з допомогою їх улюблених наборів інструментів.

Структура SysFlow складається з таких інструментів:

- sf - api надає схему SysFlow і програмні API golang, python і C++;

- sf - колектор відстежує і збирає інформацію про системні виклики і події з хостів і експортує їх у форматі SysFlow з допомогою серіалізації об'єктів Apache Avro;

- sf - процесор забезпечує оптимізований по продуктивності механізм політики для обробки, збагачення, фільтрації подій SysFlow, генерації попереджень та експорту опрацьованих даних в різні цільові об'єкти;

- sf - експортер експортує трасування SysFlow в системи зберігання, сумісні з S3, для архівування;

- sf - розгортач містить пакети для розгортання SysFlow, включаючи Docker, Helm і OpenShift;

- sysflow - це сховище документації і засіб відстеження проблем для платформи SysFlow.

Колектор SysFlow (sf - колектор). Колектор SysFlow відстежує і збирає інформацію про системні виклики і події з хостів і експортує їх у форматі SysFlow з використанням серіалізації об'єктів Apache Avro. SysFlow переводить інформацію про системні виклики в об'єктно-реляційну форму більш високого порядку, яка моделює, як контейнери, процеси і файли взаємодіють зі своїм середовищем за допомогою потоків управління процесами, і мережевих файлових операцій.

Колектор SysFlow в даний час побудований на ядрі Sysdig і вимагає, щоб зонд Sysdig пасивно збирав системні події і перетворював їх у SysFlow. В результаті колектор підтримує потужні можливості фільтрації Sysdig.

Експортер SysFlow (sf - експортер). Експортер SysFlow використовується для експорту трасування SysFlow в сховища об'єктів, сумісні з S3 (рисунок 3.2).

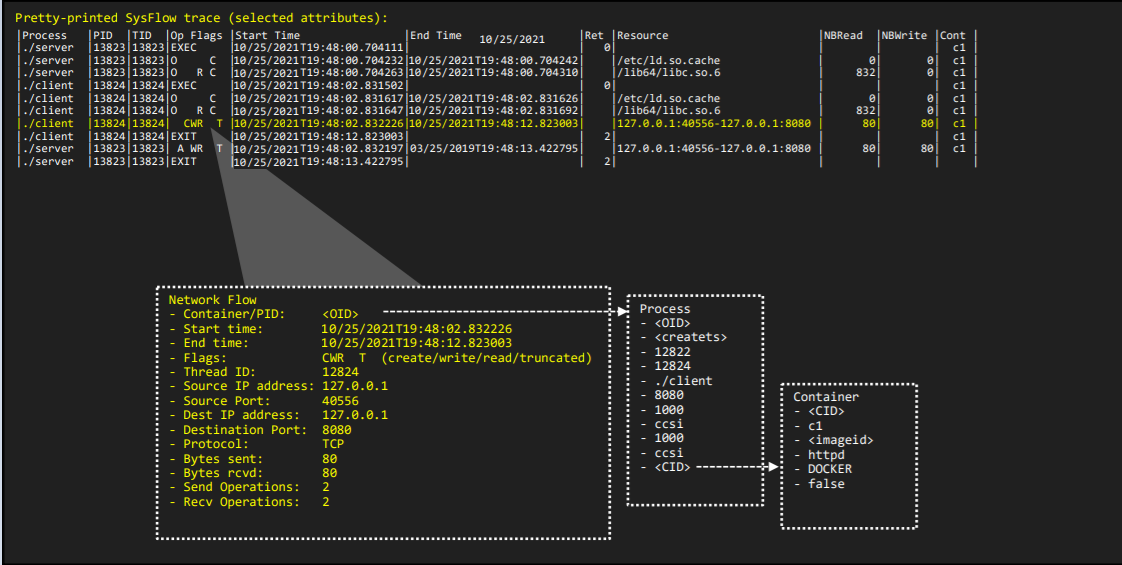


Рисунок 3.2 - Трасування потоків в SysFlow експортері

Процесор SysFlow (sf-процесор) - це полегшений конвеєр крайньої аналітики, який може обробляти і збагачувати дані SysFlow (рисунок 3.3) [3]. Процесор написаний на мові go і дозволяє користувачам створювати і настроювати різні конвеєри, використовуючи набір вбудованих і користувальницьких плагінів і драйверів. Плагіни конвеєра - це об'єкти розробник - споживач, які слідують інтерфейсу і передають дані один одному за заздалегідь визначеними каналами в багатопотоковому середовищі. А драйвер, навпаки представляє джерело даних, який передає дані в модулі. В даний час процесор підтримує два вбудовані драйвери, включаючи один, який зчитує sysflow з файлу, і інший, який зчитує потоковий sysflow через сокет домену. Плагіни і драйвери настроюються за допомогою файлу JSON.

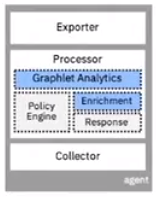


Рисунок 3.3 – Архітектура sf-процесора

Основний вбудований плагін - це механізм політики, який може застосовувати логічні правила для фільтрації, оповіщення або семантично позначати записи sysflow, використовуючи декларативну мову на основі синтаксису [правил Falco](https://falco.org/docs/rules/) з кількома доданими розширеннями (докладніше про це пізніше).

## 3.3 Алгоритм моніторингу SysFlow потоків

Алгоритм моніторингу SysFlow потоків складається з декілька етапів (рисунок 3.4) [19]. Перший етап - це спостереження за пакетами, на якому пакети захоплюються з точки спостереження та попередньо обробляються. Точками спостереження можуть бути, збирачі або процесори.

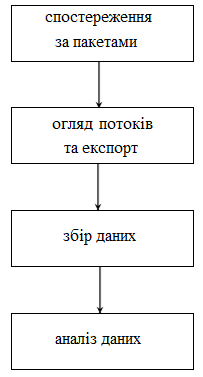


Рисунок 3.4 - Алгоритм моніторингу за потоками

Другим етапом є огляд потоку та експорт, який складається як із процесу огляду, так і з процесу експорту. У рамках процесу вимірювання пакети об’єднуються в потоки, які визначаються як набори IP-пакетів, що проходять через точку спостереження в мережі протягом певного інтервалу часу, так що всі пакети, що належать до певного потоку, мають набір загальних властивостей (рисунок 3.5) [36].

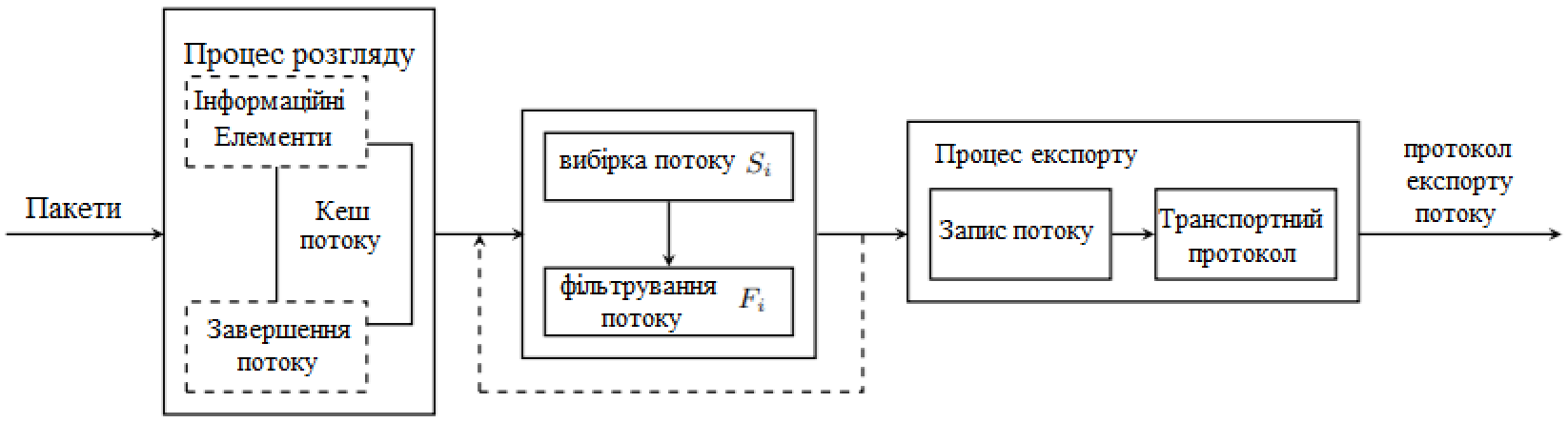


Рисунок 3.5 – Процес огляду та експорту потоку

Після того, як потік вважається завершеним, процес експорту експортує запис потоку, що означає, що запис поміщається в датаграму розгорнутого протоколу експорту потоку. Записи потоку визначені про кожен конкретний потік, який спостерігався в точці огляду, яка може включати як характерні властивості потоку (IP-адреси та номери портів), так і обчислювальні властивості (пакети та лічильники байтів). Їх можна представити, як записи або рядки в типовій базі даних з одним стовпцем за властивістю. Зазвичай використовуються SCTP, TCP, UDP транспортні протоколи. Процеси огляду та експорту на практиці тісно пов'язані.

Третій етап – збір даних. Його основними завданнями є прийом, зберігання та попередня обробка потокових даних, згенерованих у попередньому етапі. Поширені операції попередньої обробки включають агрегацію, фільтрацію, архівованих даних та генерацію підсумків.

Завершальним етапом є аналіз даних. У дослідницьких цілях аналіз даних часто носить пошуковий характер (аналіз вручну), тоді як в операційних середовищах функції аналізу часто інтегруються в етап збору даних (як ручний, так і автоматизований). Загальні функції аналізу включають кореляцію та агрегацію, профілювання, класифікації та характеристики трафіку, виявлення аномалій і вторгнень, пошук архівних даних для криміналістичних чи інших дослідницьких цілей. Зауважте, що сутності в рамках представленої архітектури є концептуальними і можуть бути об'єднані або розділені різними способами, як показано на рисунку 3.6 [37] Можна виділити дві основні відмінності:

Перше і найважливіше, етапи спостереження за пакетами та вимірювання потоку часто об’єднуються в одному пристрої, який зазвичай називають пристроєм для експорту потоку або експортером потоку. Коли експортер потоку є спеціальним пристроєм, його часто називають датчиком потоків. У цьому порядку пакети зчитуються безпосередньо з контрольованого зв’язку або отримуються через механізми пересилання пристрою пересилання пакетів.

Однак, особливо в дослідницьких середовищах, де аналізуються дані, захоплення пакетів може відбуватися на абсолютно окремому пристрої, і тому його не слід вважати невід’ємною частиною етапу аналізу потоків.

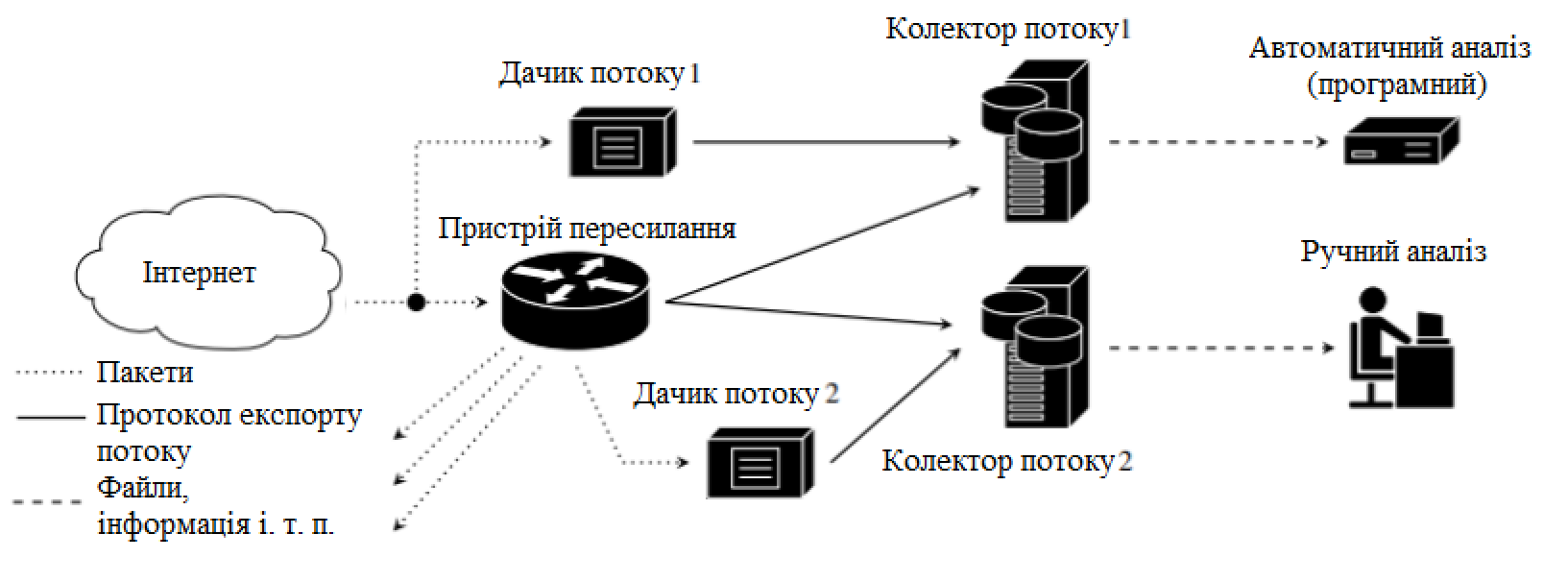


Рисунок 3.6 – Різні налаштування контролю за потоками

Друга відмінність від того, полягає в тому, що кілька експортерів потоків можуть експортувати потоки на декілька пристроїв для зберігання та попередньої обробки, які зазвичай називають колекторами потоків. Після попередньої обробки дані потоку доступні для аналізу, який може бути як автоматичним (наприклад, за допомогою приладів), так і ручним.

## 3.4 Алгоритм розкриття атаки на основі SysFlow

Щоб проілюструвати переваги підходу поглибленого аналізу загроз на основі SysFlow, на рисунку 3.7 показано, як SysFlow можна використовувати для розкриття цільової атаки, в ході якої кіберзлочинець видаляє дані з хмарного сервісу [21]. Під час розвідки (крок 1) зловмисник виявляє вразливий node.js сервер, який піддається атаці віддаленого виконання коду, що використовує вразливість в модулі node.js. Зловмисник експлуатує систему за допомогою шкідливого програмного навантаження (крок 2), яка захоплює node.js сервер і завантажує скрипт на python з віддаленого сервера (крок 3). Скрипт зв'язується з своїм сервером керування та управління (крок 4), а потім починає сканування системи на наявність конфіденційних ключів, в кінцевому підсумку отримуючи доступ до конфіденційної бази даних клієнтів (крок 5). Атака завершується, коли дані видаляються з сайту (крок 6).

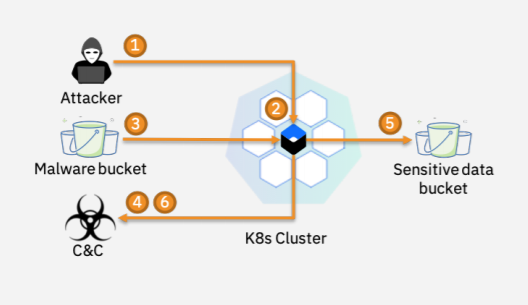


Рисунок 3.7 – Проведення атаки для видалення даних з хмарного сховища

У той час, як сучасні засоби моніторингу будуть фіксувати тільки потоки відключених подій, SysFlow може підключати об'єкти кожного етапу атаки в системі. Наприклад, виділене трасування SysFlow точно відображає етапи ланцюжка знищення атаки (рисунок 3.8) [21]: захоплення процесу node.js, а потім взаємодія з віддаленим сервером шкідливих програм на порту 2345 для завантаження і виконання шкідливого сценарію (exfil.py), який в кінцевому підсумку виконується і запускає взаємодію з сервером керування та управління на порту 4444 для отримання конфіденційної інформації з бази даних клієнтів порту на 3000. У цьому прикладі показано переваги застосування аналізу потоків до системної телеметрії.

SysFlow забезпечує видимість в середовищі хоста, надаючи відносини між контейнерами, процесами, файлами і кінцевими точками мережі у вигляді окремих подій і потоків.

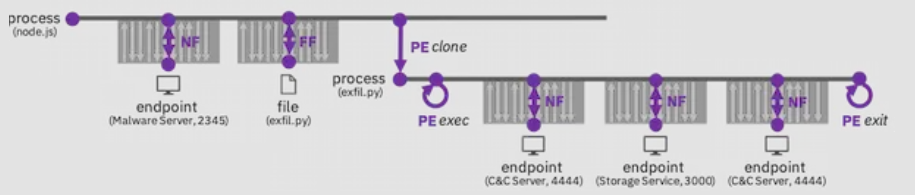


Рисунок 3.8 – Ланцюжок знищення атаки і візуальне представлення об'єктів і дій, захоплених SysFlow

Під час виявлення атаки, процес node.js клонується і виконується в новому процесі ці завдання записуються як події процесу (PE). Таким чином, цей клон насправді буде іншою подією, звідти він виконує таку саму подію як основний процес, а потім починає взаємодіяти з набором мережі та файлів. Коли процес взаємодіє з кінцевою точкою мережі або записується у файл, ці взаємодії фіксуються та підсумовуються з допомогою компактних записів потоків файлів (FF) і мережевих потоків (NF). Це дозволяє підсумувати кількість байтів і кількість операцій, які відбулися при взаємодії з цією конкретною точкою. Ця кінцева точка знаходиться на порту 4444. Аналогічним чином це можна зробити із файлами, які були записані, і прочитати та узагальнити інформацію про цей конкретний запис.

Результатом є візуальне представлення об'єктів і дій, захоплених SysFlow, яка дозволяє точно міркувати і швидко отримувати інформацію, що належить до безпеки, дозволяючи автоматизованим засобам захисту швидко виявляти атаки і реагувати на них навіть до того, як зловмисники можуть завершити свої місії.

## 3.5 Алгоритм моніторингу локальних і хмарних робочих навантажень та збору даних аналітики безпеки за допомогою SysFlow, ElasticSearch та Kibana.

SysFlow виявляє потенційно шкідливу активність у контейнерних середовищах, таких як Docker, Kubernetes і RedHat OpenShift, шляхом збору фактичних даних і отримання ознак компрометації. За допомогою інтеграції SysFlow у ELK можна побачити, як використовується SysFlow для моніторингу локальних і хмарних робочих навантажень і збору даних аналізу загроз за допомогою об'єднувача ElasticSearch.

У цьому прикладі розгортання буде використано:

- збір подій телеметрії, які відображаються у запущених контейнерах;

- застосування політики до необроблених подій і створення сповіщень;

- перевести як сповіщення, так і вихідні події в Elastic Common Schema (ECS);

- збереження результатів в ELK для гнучкого запиту з допомогою ElasticSearch і візуалізації за допомогою Kibana.

З точки зору тактики MITRE ATT&CK (база знань про тактику і методи противника) [57], можна пропустити фазу початкового доступу і зосередитись переважно на методах виявлення та контролю в цьому випадку використання. Можна припустити момент часу, коли зловмисник вже закріпився в системі і може виконувати команди. Наступним кроком зловмисника буде спроба провести розвідку навколишнього середовища або встановити додаткові інструменти, необхідні для атаки. Стек телеметрії робить ці кроки видимими, генеруючи сповіщення для характерних команд (наприклад, відображення вмісту та завантаження файлів) або надання доступу до повних даних телеметрії для кореляції подій. У цьому прикладі розгортається конвеєр SysFlow, який збирає дані телеметрії і створює два потоки: потік попереджень та потік необроблених подій SysFlow. Конвеєр перетворює дані з обох потоків в ECS і записує результат до двох окремих індексів ElasticSearch, які можна переглянути за допомогою Kibana. Для запуску сценарію виконується команда: make run. Ця команда запускає всі необхідні контейнери docker, і виводить вихідні дані (рисунок 3.9).

Атрибут make використовується для управління, створенням, запуском, зупинкою і очищенням репозиторію. Основний файл Makefile знаходиться в кореневому каталозі робочого каталогу. У всіх підкаталогах є виділені файли створення. Кожен файл Makefile підтримує чотири цілі: build, run, stop та clear.

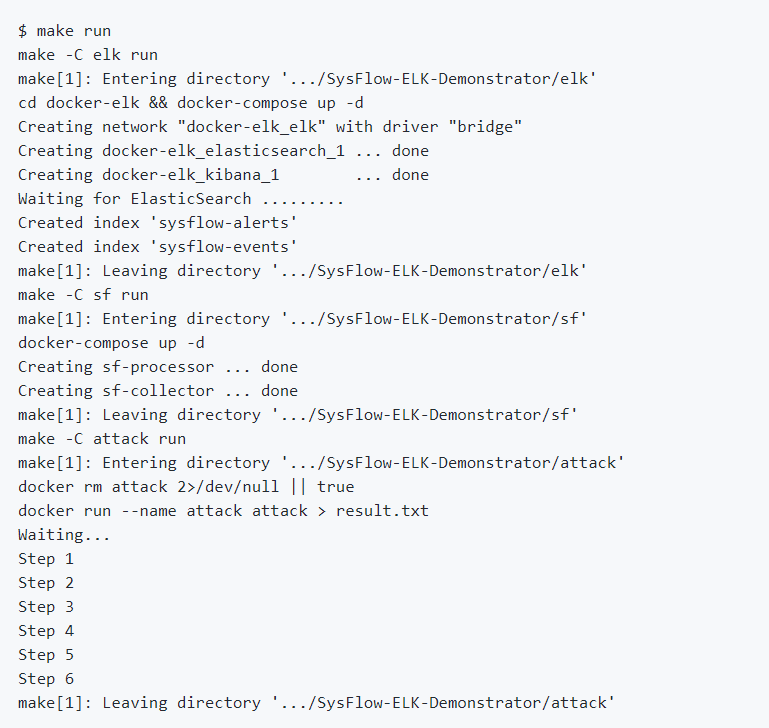


Рисунок 3.9 – Запуск усіх docker - контейнерів

Існує два типи результатів аналізу даних: сповіщення та необроблені події, які можна переглядати за допомогою Kibana (плагін для візуалізації даних SysFlow у Elasticsearch).

Сповіщення. Стек телеметрії генерує сповіщення, збагачені тегами техніки MITRE ATT&CK і причинами сповіщень. За замовчуванням стек телеметрії прикладу записує ці попередження в індекс ElasticSearch sysflow-alerts. Переглянути повідомлення у цьому індексі можна в браузері Kibana за адресою http://localhost:5601. В систему можна зайти використовуючи облікові дані ELK за замовчуванням.

Для перегляду даних з індексів ElasticSearch Kibana потрібен шаблон індексу. Якщо шаблону індексу ще немає, його необхідно створити або використати наступну команду для створення шаблону індексу, адаптовану до даних ECS SysFlow:

curl -u "<ELK username>:<ELK password>" -XPOST http://localhost:5601/api/saved\_objects/index-pattern -H 'Content-Type: application/json' -H 'kbn-version: 7.13.2' -d '{"attributes":{"title":"sysflow-alerts","timeFieldName":"event.start", "fieldFormatMap":"{\"event.start\":{\"id\":\"date\_nanos\"}}"}}'

Як тільки шаблон індексу буде створено необхідно перейти на панель "Detect", вибравши меню в лівому верхньому куті. На панелі "Detect" виберіть шаблон sysflow-alerts в полі змінити шаблону індексу в лівому верхньому кутку. У правому верхньому кутку поруч із кнопкою "Refresh" потрібно встановити відповідний діапазон часу, наприклад, "Last 15 minutes". У полі "Available fields" зліва вибирати поля попередження, які потрібно переглянути. Для початку потрібно оглянути event.reason,tags, і process.command\_line. Після цього виведуться результати сповіщень SysFlow (рисунок 3.10).

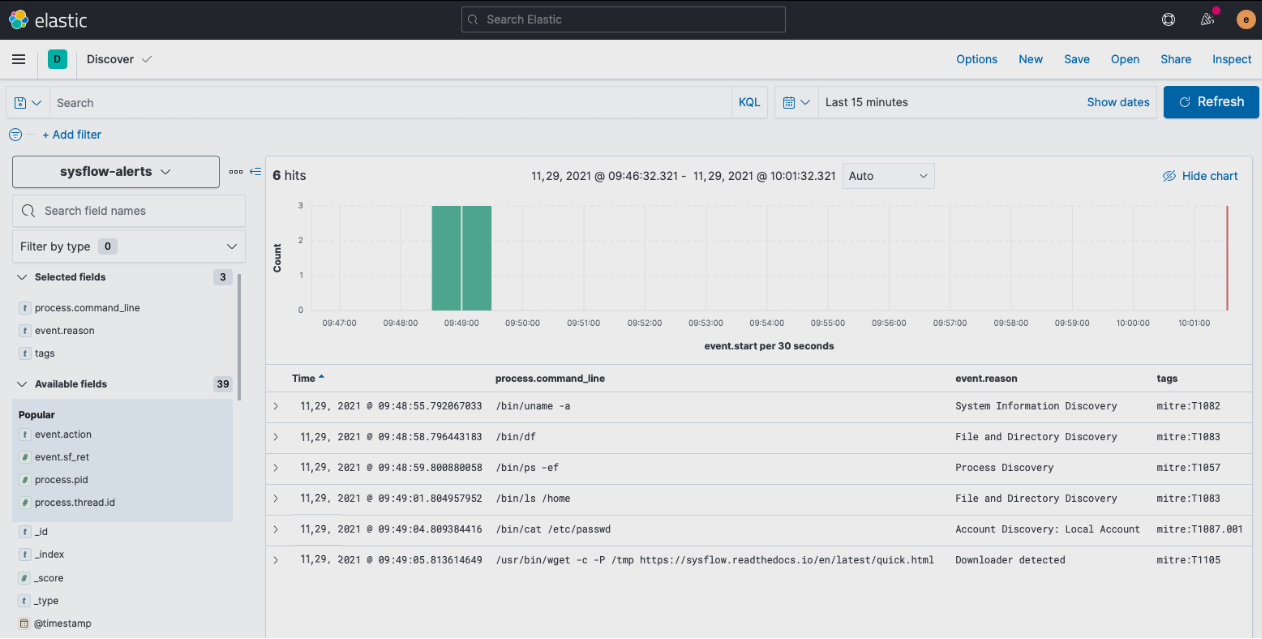


Рисунок 3.10 – Сповіщення і їх причини

Необроблені Події. Одночасно з формуванням попереджень стек телеметрії також перетворює необроблені події SysFlow в запис ECS і зберігає їх в індексі sysflow-events. Шаблон індексу для відображення даних з цього індексу також знадобиться. Рекомендовано дотримуватися процедури, описаної в пункті Сповіщення, щоб створити окремий шаблон індексу sysflow-events. Після створення шаблон індексу відображається в полі вибору на панелі виявлення Kibona.

Записи ECS в sysflow-events містять безліч різних атрибутів. Набір доступних атрибутів залежить від типу події (event.sf\_type). У полі "Доступні поля" зліва можна вибрати будь-яку комбінацію атрибутів для відображення. Результат зображення індексу подій показано на рисунку 3.11.

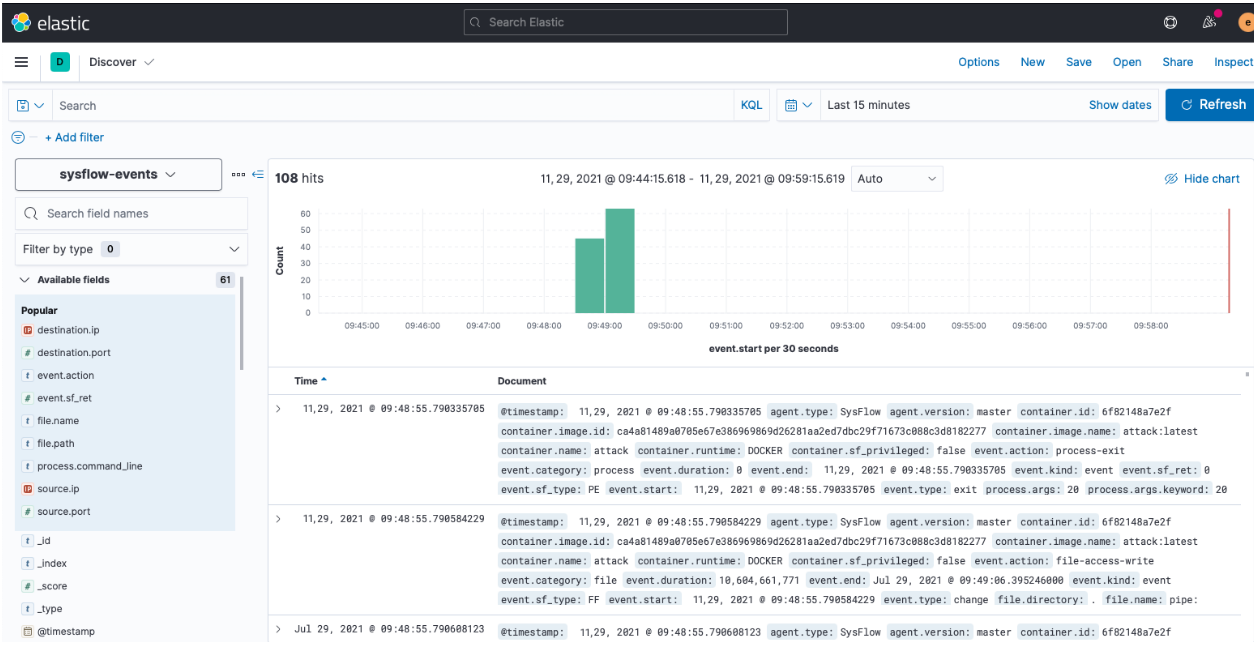


Рисунок 3.11 – Необроблені події

Контейнери docker-elk створюються з стека ELK. Вони були започатковані docker-compose. Sf-контейнери відбуваються з стека SysFlow. Вони також розпочинаються в docker-compose. Ці чотири контейнери підключені до однієї мережі докера docker-elk, створеної ELK процедурою встановлення стека. Таким чином, процесор SysFlow може "бачити" кінцеву точку ElasticSearch під ім'ям контейнера в docker, і тоді не потрібно вказувати зовнішній адресу. Ці контейнери співпрацюють, як показано на рисунку 3.12

Після виконання приклад сценарію не забудьте зупинити його. Ви можете зупинити весь стек, запустивши команду make stop. Результатом стане інформація про 5 контейнерів, які беруть участь в цій системі (рисунок 3.13).

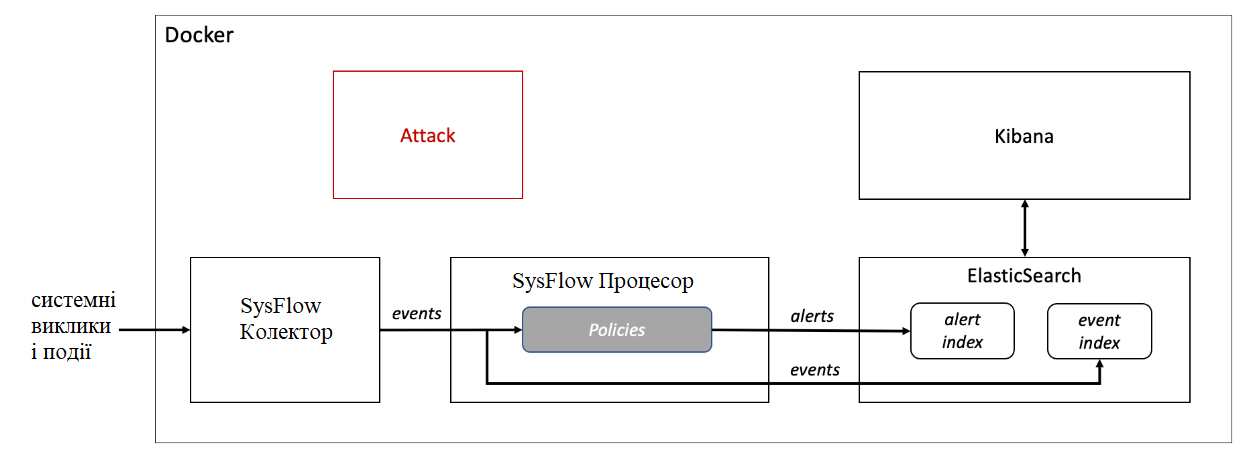


Рисунок 3.12 - Контейнери docker-elk

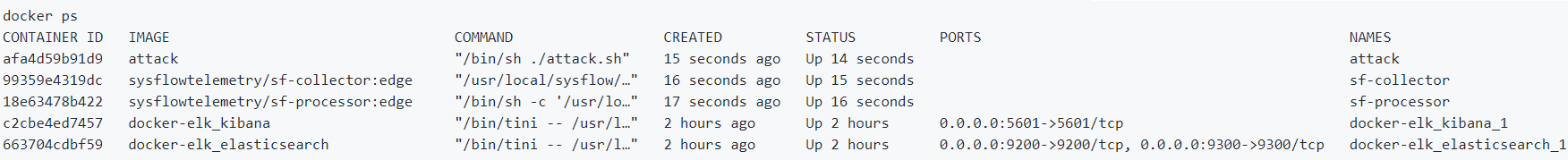


Рисунок 3.13 – Інформація про контейнери doker-elk після зупинки сценарія

Кожен контейнер має своє спеціальне призначення:

1. Attack. Результат атаки записується у файл attack/result.txt.
2. sf-колектор відстежує і збирає інформацію про системні виклики і події з контейнерів і експортує їх у форматі SysFlow з використанням равлізаці об'єктів Apache Avro. У цьому випадку складальник застосовує фільтрацію першого етапу, задану за допомогою змінної середовища в файлі sf/docker-compose.yml. Умова зчитується з визначень середовища в sf/.env. Цей фільтр приглушує всі події, що відносяться до вузла docker, контейнерів SysFlow або контейнерів ELK. Складальник пересилає виявлені події процесору, використовуючи доменний сокет UNIX, який підключається до обох контейнерів з використанням загального тома.
3. sf-процесор - це легкий конвеєр прикордонної аналітики, який може обробляти і збагачувати дані SysFlow. Поведінка процесора визначається специфікацією конвеєра, що складається з набору вбудованих і користувальницьких плагінів і драйверів. Наш конвеєр виконує генерацію попереджень на основі правил і збагачення тегами MITRE[57]. Його кінцева точка кодує дані в ECS і поміщає їх в ElasticSearch.
4. docker-elk\_elasticsearch\_1 - це екземпляр ElasticSearch з одним вузлом, що надається стеком ELK в docker.
5. docker-elk\_kibana\_1 - це інтерфейс Kibana, що надається стеком ELK в docker.

В основі цієї інтеграції лежить процесор SysFlow. Процесор SysFlow - це легко настроюваний механізм обробки даних, який зчитує дані, надані sf-колектором, і відправляє їх по конвеєру в кінцевий пункт призначення. Ecs Весь процес контролюється налаштуванням конвеєра.

Налаштування конвеєра являє собою документ JSON, який складається з декількох модулів і описує, як ці модулі з'єднуються один з одним. У даному випадку цей документ описує дерево обробки. Це пов'язано з тим, що кожен плагін може передавати більше одного вихідного каналу. У цьому розгортанні цей розгалужувач виконується засобом читання SysFlow, який живить два механізми політики, які, в свою чергу, обслуговують двох експортерів.

Визначення конвеєра процесорів для розгортання в цьому прикладі наведено нижче. У наведеній нижче конфігурації конвеєра показано, як налаштувати конвеєр, який буде зчитувати потік sysflow і відправляти записи в механізм політики, який під час виконання, ініціює сповіщення за допомогою набору механізмів що зберігаються у файлі з форматом *yaml*. Приклад конвеєра з такою конфігурацією наведено на рисунку 3.14.

Цей конвеєр визначає три вбудовані плагіни:

* sysflowreader: це універсальний плагін для зчитування даних SysFlow, який кешує їх та представляє об’єкти SysFlow обробнику для опрацювання. У цьому випадку ми використовуємо обробник elk, але можна використати користувальницькі обробники;
* policyengine: це механізм політики, який приймає правила SysFlow в якості вхідних записів, які представляють попередження, або відфільтровані події SysFlow в залежності від режиму механізму політики. Ці правила вказані в синтаксисі Falco[17]. Якщо вони збігаються, він створює сповіщення на основі подій джерела SysFlow.
* exporter: бере запис з механізму політики і експортує їх в ElasticSearch, системний журнал, файл чи термінал в форматі JSON або у форматі еластичної загальної схеми (ECS). Зверніть увагу, що власні плагіни можуть бути створені для експорту в інші формати реалізації та транспортні протоколи.

# 

Рисунок 3.14 - Конфігурація конвеєра телеметрії SysFlow

Кожен плагін має набір загальних атрибутів, які присутні у всіх плагінах, і набір атрибутів, що настроюються для деяких конкретних плагінів (рисунок. 3.15.).

Загальні атрибути наступні:

* *processor* (обов'язково): ім'я плагіну процесора. Процесори повинні реалізовувати інтерфейс sf-процесора; ім'я - це значення, яке має бути повернено GetName()функцією, визначеною в об'єкті процесора.
* *handler* (не обов'язково): ім'я об'єкта обробника, який буде використовуватися для процесора. Обробники повинні реалізовувати інтерфейс sf-обробника.
* *in* (обов'язково): вхідний канал (тобто канал golang) об'єктів, які передаються підключеному модулю.
* *out* (необов'язково): вихідний канал (тобто канал golang) для об'єктів, які виштовхуються з плагіна і переходять на наступний плагін в послідовності конвеєра.

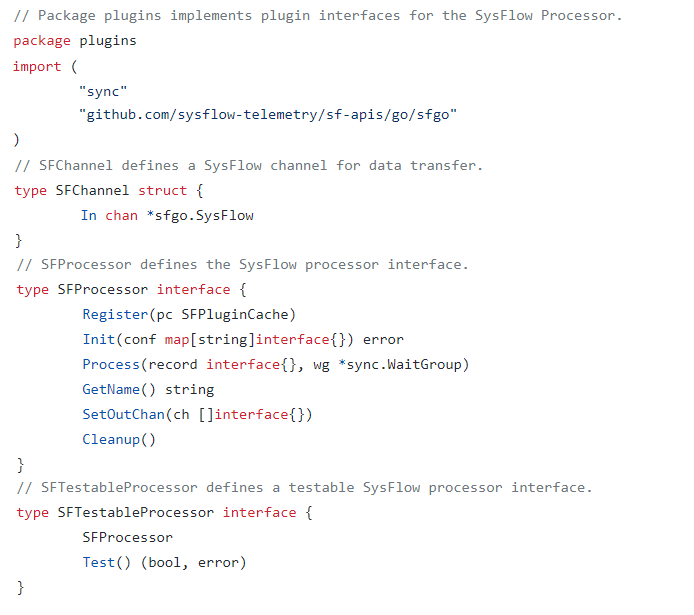


Рисунок 3.15 - Реалізація пакетів плагінів для sf-процесора

Можна змінити будь-який з атрибутів плагіна допомогою змінної середовища. Це особливо корисно при роботі з процесором в якості контейнера, де вам, можливо, доведеться розгорнути процесор на декількох вузлах і мати атрибути, які змінюються для кожного вузла. Якщо задана змінна середовища, вона заміщує параметр всередині файлу конфігурації.

Sf-канали моделюються як об'єкти потоку, які мають in-атрибут, який представляє певний канал об'єктів golang. Синтаксис sf-каналу в конвеєрі такий [ім'я каналу] [тип каналу]. Де тип каналу - це мітка, зазначена типом каналу при ідентифікації плагіна, а ім'я каналу - унікальний ідентифікатор поточного екземпляра каналу. Ім'я та тип вихідного каналу в одному плагіні повинні співпадати з іменем та типом вхідного каналу наступного плагіна в послідовності конвеєра.

Плагін має тільки один вхідний канал, але він може вказувати декілька вихідних каналів. Це дозволяє визначати конвеєри, які розділяють дані на декілька модулів приймача. Хоча завжди повинен бути один зчитувач SysFlow, діючий в якості точки входу в конвеєр, конфігурація конвеєра може вказувати механізми політики, що передають дані різним експортерам, або зчитувач SysFlow, що передає дані різним механізмам політики. Як правило, конвеєри утворюють дерево, а не лінійну структуру.

Плагін policyengine ("processor": "policyengine") керується набором правил. Ці правила вказані в YAML - файлі, який використовує той же синтаксис, що і правила Falco проекту. Специфікація плагіна механізму політики вимагає наступних атрибутів:

* policies (обов'язково): Шлях до файлу специфікації правил YAML.
* mode (необов'язково): Режим механізму політики.

Плагін exporter ("processor": "exporter") складається з двох модулів, кодувального модуля для перетворення даних у відповідний формат і транспортного модуля для відправки даних до цілі. Кодувальники призначені для конкретної мети, тобто для них експортер може використовуватися як певний набір кодерів. У конфігурації експортера транспортний модуль вказується через параметр експорту. Кодувальник вибирається за допомогою формату параметра. Формат за замовчуванням json. Null використовується для налагодження процесора і не експортує ніяких даних. Якщо для параметра експорт встановлений додатковий параметр file.path від дозволяє вказати цільовий файл на який будуть експортовані дані.

Механізм політики використовує як власний так і розширений синтаксис визначення правил Falco. Механізм політики записуються в один або кілька yaml файлів і зберігаються в каталозі, зазначеному у файлі конфігурації конвеєра під policies атрибутом плагіна механізму політики. Правила містять такі поля: правило: назва правила; опис: текстовий опис правила.

При генерації попереджень стек телеметрії також записує всі необроблені події в ElasticSearch. Хоча ця частина конвеєра створює значно більше даних, вона забезпечує дуже докладне уявлення про поведінку конкретних кроків атаки.

Sysflow - alerts містить попередження, яке вказує на виявлення процесу, з якого були завантажені дані потоку. Щоб переглянути трасування подій цього конкретного процесу, можна використати індекс sysflow-events. Для цього на панелі "Detect" в браузері Kibana необхідно вибрати шаблон sysflow-events в полі "Select" в лівому верхньому кутку вікна браузера. Перейдіть на панель "Detect" в Kibana і виберіть шаблон sysflow-events в полі "Вибір" в лівому верхньому кутку вікна браузера.

Sysflow-events містить необроблені події attack контейнера. Це пов'язано з умовою фільтра, використовуваним складальником SysFlow, яке приглушує всі події вузла, складальника, процесора і ELK, щоб зменшити кількість необроблених подій в індексі. Якщо у docker додати більше контейнерів, їх трасування SysFlow також буде видно в індексі sysflow-events.

Щоб переглянути трасування подій зловмисника, додайти умови фільтрації в Kibana, необхідно натиснути на посилання + add filter в лівому верхньому кутку. У вікні Редагування фільтра натисніть на поле для Вибору поля, виберіть атрибут ECS process.name. Виберіть оператора в полі "Select operator" поряд з полем "Field". Нарешті, натисніть на поле для вибору значення нижче, виберіть зі списку імена процесів і натисніть кнопку Зберегти. На панелі "Discover" клацніть заголовок стовпця "Time", щоб відобразити події процесу в хронологічному порядку. У полі "Available fields" зліва виберіть event.action, file.path, source.ip, source.port, destination.ip та destination.port. Атрибут event.action показує тип події. Оскільки існують загальні атрибути ECS, які заповнюються для всіх типів подій (наприклад, time, event.action), і атрибути, що відносяться до інших типів подій (всі інші в нашому прикладі), деякі клітинки будуть мати пусте значення (-). На рисунку 3.16 можна бачити, що відбувається під час атаки.

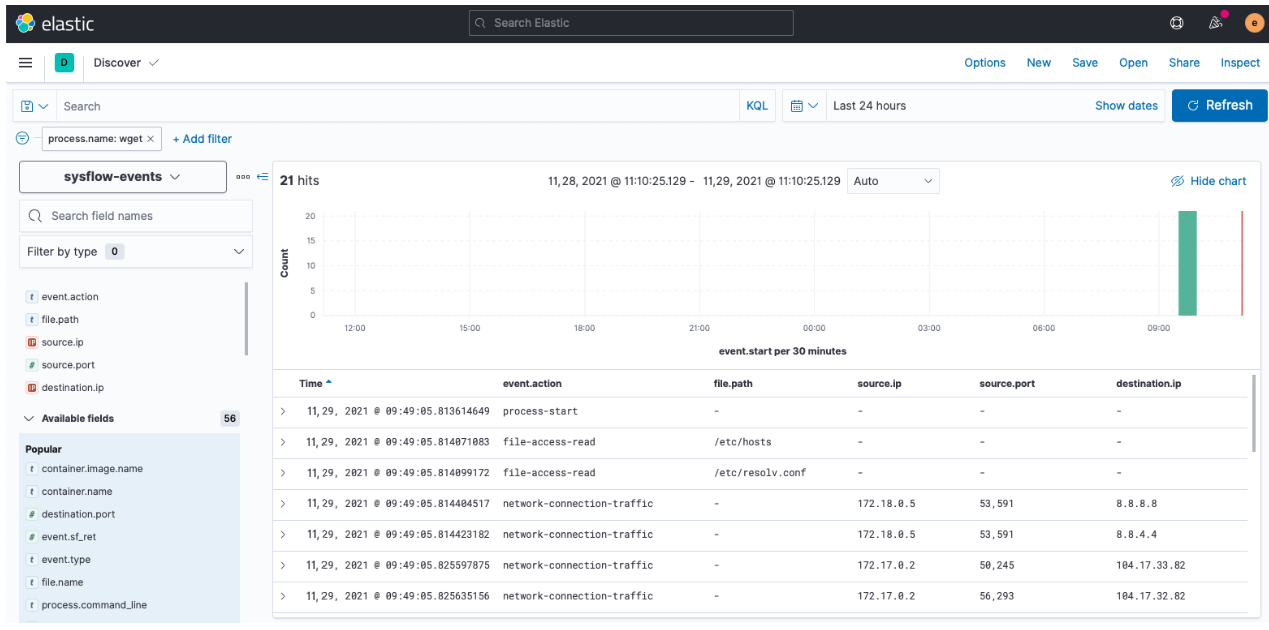


Рисунок 3.16 - Результат аналізу атаки

Можна чітко розрізнити три типи подій:

1. Події процесу (PE): Перша подія - це початок процесу. Якщо прокрутити список до кінця, можна побачити завершення останньої події process-exit. Також можна вибрати process.cmd\_line відображення повної команди -c -P /tmp https://sysflow.readthedocs.io/en/latest/quick.html яка була виконана.
2. Потоки файлів (FF): Їх можна класифікувати як file-access-read якщо дані були тільки прочитані, file-access-write якщо дані були записані, або просто file-access(не передані). Приклад показує, що деякі файли були прочитані. Існують додаткові атрибути (sf\_file\_acion.\*), які містять детальну інформацію про передачу даних з файлу або у файл. Якщо прокрутити вниз до кінця списку, також з'явиться file-access-write подія, яка вказує, що дані були записані в /tmp/quick.html
3. Мережеві потоки (NF): Подія network-connection-traffic показує тільки кінцеві точки підключення. Також видно мережевий трафік для вирішення DNS і до місця призначення 104.17.33.82, який був IP-адресою sysflow.readthedocs.io на момент створення трасування SysFlow. Більш детальну інформацію про трафік можна дізнатися за допомогою атрибутів ECS source.bytes, destination.bytes network.bytes, які містять кількість байтів, відправлених в кожному напрямку і в цілому.

# ВИСНОВКИ

Під час виконання випускної кваліфікаційній роботи було отримано такі результати:

1. Проаналізовано об’єкт досліджень – поглиблений аналіз загроз, що показало актуальність поставленої задачі. Досліджений життєвий цикл аналізу загроз та типи поглибленого аналізу загроз. Проаналізовано вплив поглибленого аналізу загроз.

2. Зроблено поглиблений аналіз безпеки внутрішніх мереж на основі flow–протоколів, досліджено SysFlow, як засіб моніторингу систем та мереж.

3. Проаналізовано основні інструменти та елементи SysFlow для поглибленого аналізу загроз та для чого вони використовуються. Було показано як SysFlow можна використовувати для розкриття цільової атаки з метою видалення даних з хмарного сховища.

4. Реалізовано алгоритм моніторингу SysFlow потоків. Досліджено взаємозв’язок усіх етапів моніторингу потоків .

5. Програмно реалізований алгоритм моніторингу локальних і хмарних робочих навантажень та збору даних аналітики безпеки. Проведено візуалізацію отриманих даних за допомогою гнучких запитів Elasticsearch та візуалізацію браузера Kibana, що дало змогу SysFlow виявляти потенційну шкідливу активність.

6. Проведено тестування реалізованих алгоритмів, що підтвердило ефективність SysFlow, як засобу для поглибленого аналізу загроз.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SysFlow Specification URL: https://sysflow.readthedocs.io/en/latest/spec.html.
2. Penetration Testing: SysFlow: Cloud-native system telemetry pipeline URL: https://securityonline.info/sysflow-cloud-native-system-telemetry-pipeline/.
3. IBM Open Sources SysFlow Monitoring Platform URL: https://devops.com/ibm-open-sources-sysflow-monitoring-platform/.
4. SysFlow: Scalable system telemetry for improved security analytics URL: https://research.ibm.com/blog/sysflow.
5. Збірник наукових праць "Спеціальні телекомунікаційні системи та захист інформації" випуск 2(26) 2014 с.87-97 Шаповал М. В. Порівняльний аналіз методів захисту даних pos-термінального трафіку.
6. Анализ угроз информационной безопасности URL: https://arinteg.ru/articles/analiz-ugroz-informatsionnoy-bezopasnosti-27291.html
7. Proactive Defense: Understanding the 4 Main Threat Actor Types URL: https://www.recordedfuture.com/threat-actor-types/
8. What Is Threat Intelligence? Why Is Threat Intelligence Important? URL: https://www.recordedfuture.com/threat-intelligence/
9. How Strategic Threat Intelligence Informs Better Security Decisions URL: https://www.recordedfuture.com/strategic-threat-intelligence/
10. How to Empower Your SOC With Security Intelligence URL: https://www.recordedfuture.com/empower-security-operations/
11. How Does Golang Channel Works URL: https://levelup.gitconnected.com/how-does-golang-channel-works-6d66acd54753
12. By Joseph Muniz, Gary McIntyre, Nadhem AlFardan: Security Operations Center: Building, Operating, and Maintaining your SOC: Published 2015 by Cisco Press 68-97
13. Телеметрія URL: https://uk.wikipedia.org/wiki/Телеметрія
14. Quick Start. Starting the collection probe. Deployment options. URL: https://sysflow.readthedocs.io/en/latest/quick.html#deployment-options
15. Как Cisco мониторит безопасность в своей внутренней сети? URL: https://habr.com/ru/company/cisco/blog/348532/
16. SysFlow telemetry slack URL: https://sysflow-telemetry.slack.com/
17. Falco rules URL: https://falco.org/docs/rules/
18. B. Claise, B. Trammell, and P. Aitken, “Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information,” RFC 7011 (Internet Standard), Internet Engineering Task Force, Septe
19. Діаграма Венна URL: https://uk.wikipedia.org/wiki/Діаграма\_Венна
20. T. Zseby, E. Boschi, N. Brownlee, and B. Claise, “IP Flow Information Export (IPFIX) Applicability,” RFC 5472 (Informational), Internet Engineering Task.
21. SysFlow: Scalable System Telemetry for Improved Security Analytics URL: https://resources.sei.cmu.edu/asset\_files/Presentation/2020\_017\_001\_644826.pdf
22. SysFlow: Cloud-native system telemetry pipeline URL: https://github.com/sysflow-telemetry/sysflow
23. M. Burkhart, D. Schatzmann, B. Trammell, E. Boschi, and B. Plattner, “The Role of Network Trace Anonymization under Attack,” SIGCOMM Computer Communication Review, vol. 40, no. 1, pp. 5–11.
24. Презентація з Payment Security 2019 про боротьбу з атаками Cisco Systems URL: https://www.slideshare.net/lukatsky/ss-154755507
25. IBM Financial Crimes Insight: About graph analytics URL: https://www.ibm.com/docs/en/fci/1.1.0?topic=analytics-about-graph
26. SysFlow Pricing, Alternatives & More 2021 URL: https://www.capterra.com/p/144802/SysFlow/
27. Cisco Security Packet Analyzer. Intelligent Packet Capture for Faster Response URL: https://www.cisco.com/c/en/us/products/security/security-packet-analyzer/index.html
28. Машинне навчання проти прогнозуваної аналітики URL: https://uk.education-wiki.com/5277960-machine-learning-vs-predictive-analytics
29. Systems Flow, Inc URL: http://www.sysflow.com/
30. Cisco Secure Network Analytics (Stealthwatch) URL: https://www.cisco.com/c/ru\_ru/products/security/stealthwatch/index.html#
31. netmattersweb/sysflow-ee:latest sysflowtelemetry's URL: https://hub.docker.com/layers/netmattersweb/sysflow-ee/latest/images/sha256-771d391f7160d41a6088e3041e91e5e54479457ff136ad4ff519c3c80ddde3c6?context=explore
32. Грайворонський М.В. Сучасні підходи до забезпечення кібернетичної безпеки/ Матеріали ХVII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики», НТУУ «КПІ», 2015 р.,
33. Go by Example: Channels URL: https://gobyexample.com/channels
34. Network flow problem URL: https://en.wikipedia.org/wiki/Network\_flow\_problem
35. Network Flow Algorithms for Structured Sparsity Julien Mairal , Rodolphe Jenatton , Guillaume Obozinski , Francis Bach. INRIA Rocquencourt, LIENS: Adv. Neural Information Processing Systems, 2010
36. B. Trammell and E. Boschi, “An Introduction to IP Flow Information Export (IPFIX),” IEEE Communications Magazine, vol. 49, no. 4, pp. 89–95, 2011.
37. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, March 2008.
38. ECS Field Referenceedit URL: https://www.elastic.co/guide/en/ecs/1.7/ecs-field-reference.html
39. Juniper Secure Analytics Configuring DSMs Guide. SysFlow URL: https://www.juniper.net/documentation/us/en/software/jsa7.4.2/jsa-dsm/topics/concept/jsa-dsm-sysflow.html
40. Що таке аналіз загроз в кібербезпеці? URL: https://datami.ua/shho-take-analiz-zagroz-v-kiberbezpetsi/
41. Cyber Security threats Statistics URL: https://www.fundera.com/resources/small-business-cyber-security-statistics/
42. Kibana Guide [7.15] Quick start URL: https://www.elastic.co/guide/en/kibana/current/get-started.html
43. SysFlow: Docs, Tutorials, Reviews. Openbase URL: https://openbase.com/python/sysflow
44. sysflow-telemetry and sf-collector Go PKGoto GithubPK URL: https://githubhelp.com/sysflow-telemetry/sf-collector
45. The Complete Guide to the ELK Stack URL: https://logz.io/learn/complete-guide-elk-stack/
46. Rick Hofstede, Pavel Celeda, Brian Trammell, Idilio Drago, Ramin Sadre, Anna Sperotto and Aiko Pras "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX" Venue/Journal/Book: IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, 2014
47. DLP надійне запобігання витокам інформації компаній. URL: https://www.eset.com/ua/support/information/entsiklopediya-ugroz/dlp/
48. N. Duffield, “Sampling for Passive Internet Measurement: A Review,” Statistical Science, vol. 19, no. 3, pp. 472–498, 2004.
49. SiLK, the System for Internet-Level Knowledge URL: https://www.oreilly.com/library/view/network-security-through/9781449357894/ch05.html
50. https://fitsmallbusiness.com/cybersecurity-statistics/
51. B. Li, J. Springer, G. Bebis, and M. H. Gunes, “A Survey of Network Flow Applications,” Journal of Network and Computer Applications, vol. 36, no. 2, pp. 567–581, 2013.
52. Cisco Systems, Inc., “Cisco IOS Flexible NetFlow Commands,” December 2010, accessed on 2 May 2014. URL: http://www.cisco.com/en/US/docs/ios/fnetflow/ command/reference/fnf 01.pdf
53. R. Hofstede, V. Bartos, A. Sperotto, and A. Pras, “Towards Real- ˇ Time Intrusion Detection for NetFlow/IPFIX,” in Proceedings of the 9th International Conference on Network and Service Management, CNSM’13, 2013, pp. 227–234
54. WHAT IS CYBER THREAT INTELLIGENCE? URL: https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/
55. B. Trammell, E. Boschi, L. Mark, T. Zseby, and A. Wagner, “Specification of the IP Flow Information Export File Format,” RFC 5655 (Proposed Standard), Internet Engineering Task Force, October 2009.
56. Живко З. Інформаційні загрози: суть і проблеми / З. Живко, М. Живко // Безпека та захист інформації в інформаційних системах: тези доповідей ІІ Міжнародної НПК. — К., 2017. — C. 116—118.
57. MITRE ATT&CK URL: https://attack.mitre.org/
58. Дереко В.Н. Теоретико-методологічні засади класифікації загроз об’єктам інформаційної безпеки / В. Дереко // Інформаційна безпека людини, суспільства, держави. – 2015. – № 2 (18). – С. 16–22,
59. IBM SysFlow 2020: Exit of the tool URL: https://www.tadviser.ru/index.php/product:IBM\_SysFlow
60. SysFlow: Scalable System Telemetry for Improved Security Analytics URL: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=642299