НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«До захисту допущено»
В.о. завідувача кафедри

_____ М.В.Грайворонський
(підпис)

“___” _____2019 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

на тему: Виявлення р-2-р ботнетів за допомогою SIEM-систем _____

_____

Виконав (-ла): студент (-ка) _4__ курсу, групи __ФБ-52_____
(шифр групи)

Фіц Владислав _____
(прізвище, ім’я, по батькові) (підпис)

Керівник к. ф.-м. н., доц. кафедри ІБ Грайворонський М. В.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент_____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)
«___»_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту

_____Фіц Владислав_____
(прізвище, ім'я, по батькові)

1. Тема роботи Виявлення р-2-р ботнетів за допомогою SIEM-систем_____ ___
_____,
науковий керівник роботи к. ф.-м. н., доц. кафедри ІБ Грайворонський М. В. ___ ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «____ » 2019 р. № _____

2. Термін подання студентом роботи 10 червня 2019 р.

3. Вихідні дані до роботи _____
_____
_____

4. Зміст роботи_____
_____
_____
_____
_____

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) _____
_____
_____
_____

6. Дата видачі завдання _____

РЕФЕРАТ

Кваліфікаційна робота містить: 56 сторінок, 15 рисунків, 2 таблиці та 11 джерел.

У цій роботі було розглянуто роботу SIEM-систем на прикладі Elasticsearch, а також її використання в якості детектора p-2-p ботнету.

Проаналізовано більшість відомих на сьогодні архітектур ботнетів, методів їх виявлення.

Було розглянуто різні підходи при отриманні даних, аналізі, виділенні з них дійсно важливих, обробці цих даних та аналізі отриманих результатів.

Порівняно з результатами, отриманими при інших умовах та з інших джерел.

Основною метою було дослідження роботи SIEM-системи в якості ботнет-детектора.

Завданням роботи було налаштування системи отримування логів, їх візуалізація, створення моделі виявлення ботнету на основі відомих датасетів ботнетів. Порівняння роботи моделі в залежності від зміни вхідних даних.

Об’єктом дослідження було використання SIEM-систем задля більш ефективного виявлення наявності ботнету.

Предметом дослідження було порівняння даних, отриманих з різних джерел, та їх перевірка при різних параметрах.

Отримані результати можуть бути використані для більш точного написання правил SIEM-систем, що зможе значно підвищити їх ефективність.

КЛЮЧОВІ СЛОВА: БОТНЕТ, P-2-P БОТНЕТ, SIEM-СИСТЕМИ, АНАЛІЗ ДАНИХ, ПОРІВНЯННЯ РЕАЛЬНИХ ДАНИХ З ЛАБОРАТОРНИМИ.

РЕФЕРАТ

Квалификационная работа содержит 56 страниц, 15 рисунков, 2 таблицы и 11 источников.

В этой работе была рассмотрена работа SIEM-систем на примере Elasticsearch, а также ее использование в качестве детектора p-2-p ботнета.

Проанализированы большинство известных на сегодня архитектур ботнетов, методов их обнаружения.

Были рассмотрены различные подходы при получении данных, анализе, выделении из них действительно важных, обработке этих данных и анализе полученных результатов.

Сравнено с результатами, полученными при других условиях и из других источников.

Основной целью было исследование работы SIEM-системы в качестве ботнет-детектора.

Задачей работы было настройка системы получения логов, их визуализация, создание модели обнаружения ботнета на основе известных датасетов ботнетов. Сравнение работы модели в зависимости от изменения входных данных.

Объектом исследования было использование SIEM-систем для более эффективного выявления наличия ботнета.

Предметом исследования было сравнение данных, полученных из различных источников, и их проверка при различных параметрах.

Полученные результаты могут быть использованы для более точного написания правил SIEM-систем, сможет значительно повысить их эффективность.

КЛЮЧЕВЫЕ СЛОВА: БОТНЕТ, P-2-P БОТНЕТ, SIEM-СИСТЕМЫ, АНАЛИЗ ДАННЫХ, СРАВНЕНИЕ РЕАЛЬНЫХ ДАННЫХ С ЛАБОРАТОРНЫМИ.

ABSTRACT

The qualifying paper contains: 56 pages, 15 figures, 2 tables and 11 sources.

In this work, the work of SIEM-systems was considered in the example of Elasticsearch, as well as its use as a p-2-p botnet detector.

Most of the well-known botnets' architectures and methods of their detection are analyzed.

Different approaches were taken into consideration in obtaining data, analyzing, distinguishing them from really important ones, processing these data and analyzing the results.

Compared to the results obtained under other conditions and from other sources.

The main goal was to study the work of the SIEM system as a botnet detector.

The task of the work was to set up a logging system, to visualize it, and to create a botnet detection model based on known botnets datasets. Comparison of model work depending on the change of input data.

The object of the study was to use SIEM-systems to more effectively detect the presence of botnets.

The subject of the study was to compare the data obtained from different sources and check them at various parameters.

The results can be used to more accurately write the rules of SIEM-systems, which can significantly increase their effectiveness.

KEYWORDS: BOTNET, P-2-P BOTNET, SIEM-SYSTEMS, ANALYSIS OF DATA, REAL AND LABORATORY DATA COMPARISON.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення

ІБ – Інформаційна безпека

C&C – Command and Control

IRC – Internet Relay Chat

P-2-P – Peer-to-Peer

DGS – Domain Generation System

IP – Internet Protocol address

DDoS – Disturbed Denial-of-Service

SIEM – Security information and event management

СУБД – Система управління базами даних

IDS – Intrusion Detection System

IPS – Intrusion Prevention System

NAT – Network Address Translation

ВСТУП

З безперервним розвитком інформаційних технологій та покращенням рівня життя та комфортності, комп’ютерні системи були інтегровані майже у всі аспекти людського життя. Інформація стала товаром, який можна придбати, продати, обміняти. При цьому вартість інформації часто в сотні разів перевершує вартість комп'ютерної системи, в якій вона зберігається.

Сучасні методи обробки, передачі та накопичення інформації сприяли появі загроз, пов'язаних з можливістю втрати, перекручування та розкриття даних, які адресовані або належать кінцевим користувачам. Саме недостатня захищеність та широка поширеність таких систем створюють передумови для пошуку вразливостей та подальшого використання останніх в корисних цілях. Прикладом можуть слугувати ботнети.

Традиційні ботнети використовують централізовану архітектуру. Це означає що існує єдиний сервер який віддає команди всім своїм ботам. Проблема виникає коли вдається заблокувати доступ до сервера, це означає, що ботмайстер втрачає контроль над всією бот-мережею. Новітні ботнети застосовують розподілену архітектуру, тобто, фактично кожен заражений вузол в мережі може виступати сервером передачі команд на виконання або не виконання тих або інших дій. Як наслідок, знищення такого ботнету завдання не з простих, а отже моніторинг таких ботнетів є важливим завданням для аналітиків.

Актуальність роботи

Широка поширеність дешевих пристроїв з виходом в інтернет робить їх потенційними учасниками ботнету, навіть без їх відома. Кількість ботів об’єднаних в одну мережу у 2030 буде 125 мільярдів. Володіючи такими ресурсами, у світі де абсолютно все залежатиме від інтернету, зловмисник зможе атакувати абсолютно будь яку інфраструктуру наносячи колосальні

збитки. А в силу ускладнення архітектури ботнетів, часто винуватці мають можливість уникати відповідальності, а отже не понесуть абсолютно ніякого покарання.

Метою роботи є дослідження роботи SIEM-систем в якості ботнет детектора.

Завданням роботи є налаштування системи отримування логів, їх візуалізація, створення моделі виявлення ботнету на основі відомих датасетів ботнетів. Порівняння роботи моделі в залежності від зміни вхідних даних.

1 ОСНОВНІ ТЕРМІНИ ТА ВИЗНАЧЕННЯ

Основні поняття та терміни даної роботи пов’язані з ботнетом та SIEM-системами.

1.1 Ботнет

Термін ботнет виник як поєднання слів «робот» (robot) та «мережа» (net). З практичної точки зору ботнет це сукупність скомпрометованих комп'ютерів (ботів), заражених однаковим шкідливим програмним забезпеченням, що об’єднані в мережу і віддалено керовані зловмисником - ботмайстром. Зазвичай всі заражені машини отримують команди від однієї машини. На рисунку 1.1 наведено узагальнену схему ботнету.

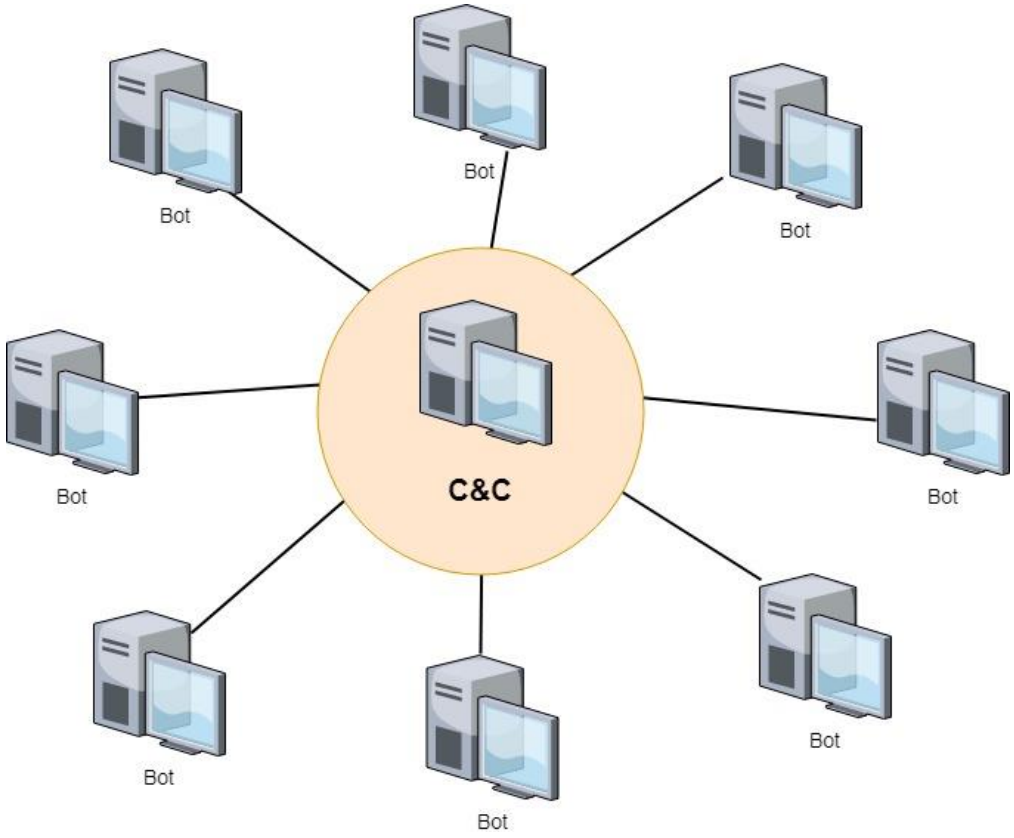

Рисунок 1.1 – Загальна схема ботнету

Саме для ефективного адміністрування та оновлення ботнету, ботмайстер мусить створити інфраструктуру каналу зв’язку для надсилання команд ботам і отримання від них результатів на запити. Такий канал зазвичай називається каналом управління та контролю (C&C).

1.1.1 Поняття ботнету

Перші, традиційні ботнети використовують централізовану, клієнт-серверну архітектуру, більш нові – розподілену (децентралізовану), одноранговy (Peer-to-Peer). Обидві архітектури мають свої переваги та недоліки. Завдяки своїй простоті централізований ботнет широко використовується багатьма сімействами ботнетів. Найбільш відомими підходами є Internet Relay Chat(IRC) та Hypertext Transfer Protocol(HTTP).

IRC – це система чату, яка забезпечує миттєвий обмін повідомленнями «один-до-одного» та «один-до-багатьох» через інтернат. На рисунку 1.2 наведено приклад схеми IRC-ботнету. Користувачі можуть підключитись до визначеного каналу в мережі IRC і спілкуватись з групами інших користувачів. З часом цей канал почав використовуватись IRC-ботами для зловмисних дій. Зараження відбувалось в основному через використання сімейства троянів BackDoor.IRC.Bot, боти яких з'єднувалися з виділеним IRC-сервером, підключалися до певного чат-каналу і починали «слухати» його в очікуванні вхідних команд. Це ускладнювало виявлення та аналіз даних для знаходження адреси сервера. Сама адреса розшифровувалась лише піж час безпосередньої роботи бота. Першими цілями IRC-ботів були атаки на інших користувачів IRC, а також на IRC-сервери.

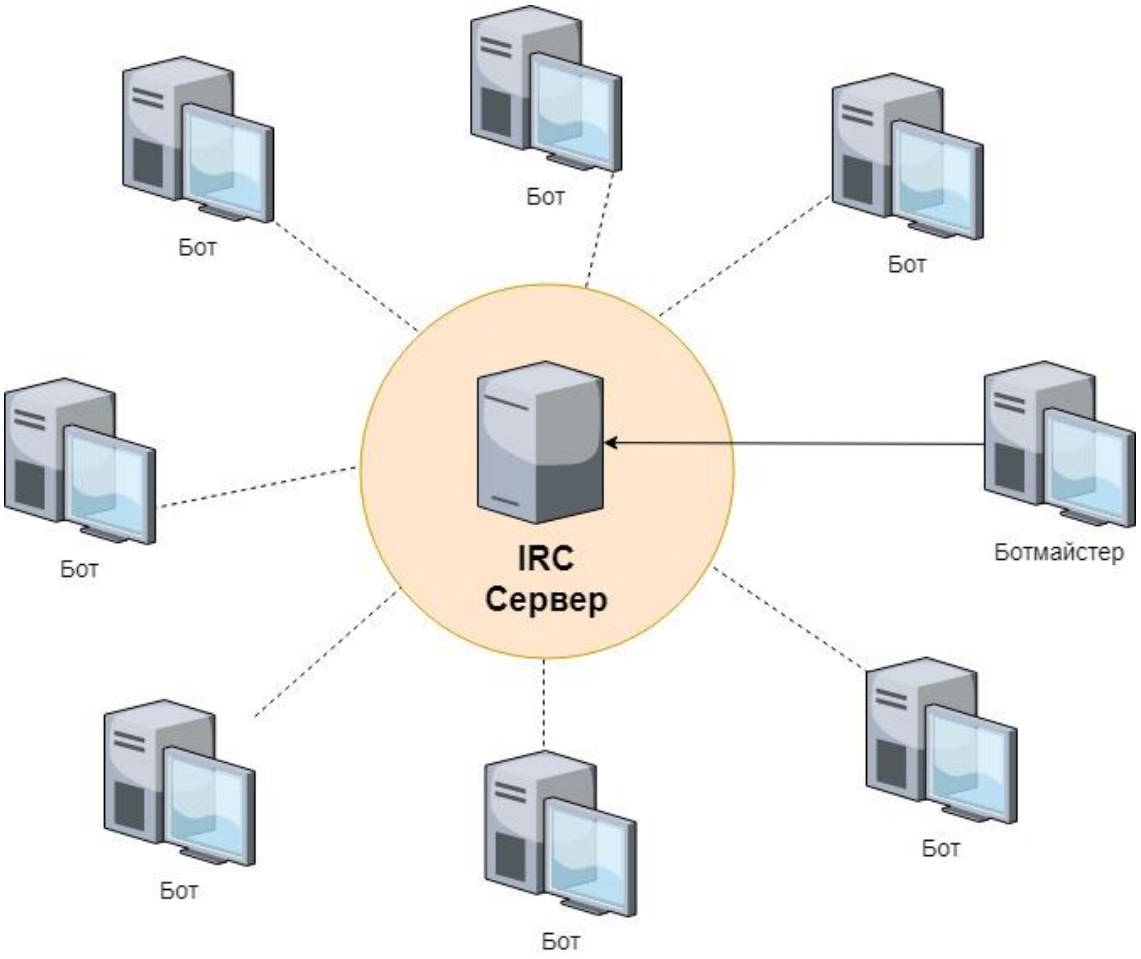

Рисунок 1.2 – Схема IRC-ботнету

Зіркоподібна структура перших ботнетів породжувала істотний архітектурний недолік: при відключенні або перехопленні C&C сервера вся бот-мережа ставала недієздатною, і робота зловмисників виявлялася даремною [1].

Наступним кроком стала поява технології генерації доменних імен C&C серверів (DGS, domain generation system). Ця система була створена для покращення боротьби з антивірусами і для підвищення противиявних властивостей ботнетів.

Використовуючи спеціальний алгоритм, що генерує C&C адресу сервера за визначеною схемою, ботмайстри знайшли новий спосіб боротьби з виявленням сервера керування, вони перестали використовувати статичні

списки адрес останніх. Алгоритм генерував масив потрібних для перевірки адрес. З отриманого масиву адрес, бот, по черзі відправляв запити та очікував на конкретно визначену відповідь. Якщо відповідь підтверджувалась – адреса сервера ставала адресою сервера керування. Відомі випадки, коли адреси були комбінацією цифр, латинських символів та доменів першого рівня .com або .org. Володіючи алгоритмом за яким генеруються адреси, ботмайстер з легкістю може реєструвати нові домени для керуючих серверів. Така властивість дозволяє легко обходити проблему блокування C&C сервера, просто перенісши його на новий IP.

Для підвищення стійкості від різноманітних зовнішніх впливів використовується механізм розбиття ботмережі на окремі, незалежні підмережі, що керуються своїми власними серверами.

Подальшим етапом еволюції ботнетів було впровадження зашифрованих протоколів обміну даними, використання цифрового підпису для підтвердження автентичності керуючого сервера, що значно ускладнювало механізми комунікації ботнету та керуючого сервера, але і ускладнювало їх виявлення та знешкодження.

Подальший розвиток та збільшення ефективності методів виявлення серверів керування, змушували ботмайстрів переходити на більш складні архітектури C&C, такі, що базувались на децентралізації.

Ботнети P-2-P (Peer-To-Peer), або пірингові однорангові ботнети.

Ідеєю роботи такого ботнету є відмова від використання єдиного сервера керування. Боти такої мережі спілкуються з усіма сусідніми зараженими машинами, передаючи команди від себе до сусіда, що зображено на рисунку 1.3. Такий принцип називається децентралізованим. Вивівши конкретну машину з ладу неможливо знищити весь ботнет. Як приклад, для створення децентралізованого ботнету спочатку використовувався вірус Win32.Sector.

Він може завантажувати та запускати на зараженій машині будь які програми, зупиняти роботу антивірусів і забороняти користувачам доступ в інтернет. [1]

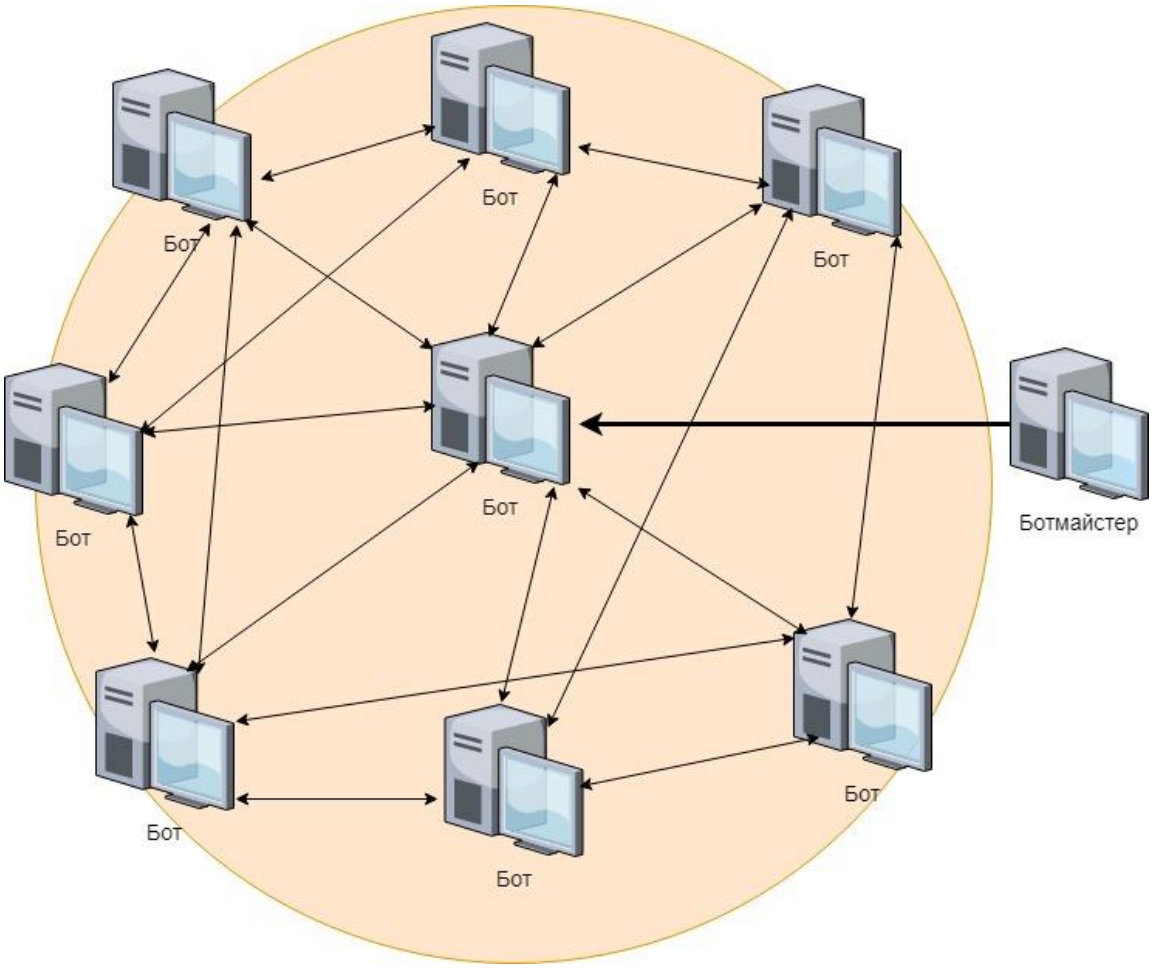

Рисунок 1.3 – Схема р-2-р-ботнету

Для створення ботнету змішаного типу, що використовувався для виконання веб-інжектингу існує троян Trojan.Dridex.49. Він використовує шифрування при обміні даними по інтернету, вбудовуючись в процеси комп’ютера-жертви для сталої роботи. Такого типу ботнет дозволяв викрадати конфіденційну інформацію, отримувати дистанційний доступ та дані систем пов’язаних з банківським обслуговуванням.

Для забезпечення підвищеної стабільності роботи використовувалась складна двошарова проксі мережа, що у поєднанні з одноранговою архітектурою ботнету робили зв’язок з C&C сервером максимально надійним.

Заражений вірусом комп’ютер повинен був визначити для себе одну з трьох можливих ролей, в залежності від рівня розміщення бота у ієрархії мережі.

• роль «Bot» - цю роль отримують боти, які розміщені у внутрішній мережі(локальній) без виходу в інтернет. Тобто для стабільної роботи в якості ботів їм потрібно підтримувати зв’язок з ботами ролі «Node».

• роль «Node» - цю роль отримують боти, що розміщені на межі локальної мережі та інтернету. Основною їх задачею є виконання функції певного моста для передачі команд та даних від «Admin node» до «Bot» та в зворотному напрямку.

• роль «Admin Node» - цю роль отримують боти, що мають доступ до інтернету і можуть безпроблемно здійснювати зв’язок між собою та з сервером керування.

Останнім часом для підвищення захищеності ботнету активно почали використовуватись цифрові ключі, якими боти обмінюються між собою. [2]

Іншим прикладом еволюції сучасних p-2-p-ботнетів є використання тунелювання в побудові та керуванні останніх. Деякі види ботнетів спеціалізованих на банківській сфері, наприклад такі, що використовують троян Zeus, запускаючись на комп’ютері-жертві, намагаються підключитись до ботів з роллю «Node», отримуючи їх адреси з власного вбудованого списку у файлі конфігурацій. Якщо це їм не вдається, далі в роботу вступає виконання алгоритму DGS, що генерує список можливих адрес серверів, на яких містяться дані про відкриті «Node» вузли. Згенеровані адреси вказують саме на сервери-посередники, що містять лише дані про адреси вузлів другого рівня, а не на керуючі сервери. Такі посередники з легкістю видають списки

всім ботам які до них звертаються, а основною ідеєю їх існування є ускладнення виявлення істинного C&C сервера. Ботмайстер, задля ускладнення локалізації та ідентифікації сервера керування, користується тунелювання, не передаючи команди безпосередньо на сервер. Саме «Node» вузли в цій ботмережі виступають в якості серверів тунелювання, максимально заплутуючи аналіз структури та ієрархії ботнету.

1.1.2 Призначення ботнету

Використання ботнету напряму зв’язано з отриманням фінансової вигоди, зокрема їх активне використання спостерігається у сфері комерційної діяльності, використовуючи різні типи атак, такі як: збір конфіденційної інформації, фішинг, пошуковий спам, спам у повідомленнях пошти, DDoS та інші. [3]

DDoS (від англ. Distributed Denial-of-Service) – це тип атаки, що базується на методі масивного відправлення запитів на ціль ураження, наприклад комп’ютер-жертву або сайт-жертву, що призводить до перевантаження ресурсів сервера або комп’ютера. На рисунку 1.4 зображено схему DDoS-атаки. Перенавантажений хост переходить в стан, коли він перестає приймати та опрацьовувати масові вхідні запити в тому числі і від легітимних користувачів. Такий стан речей називається відмовою в обслуговуванні. [1]

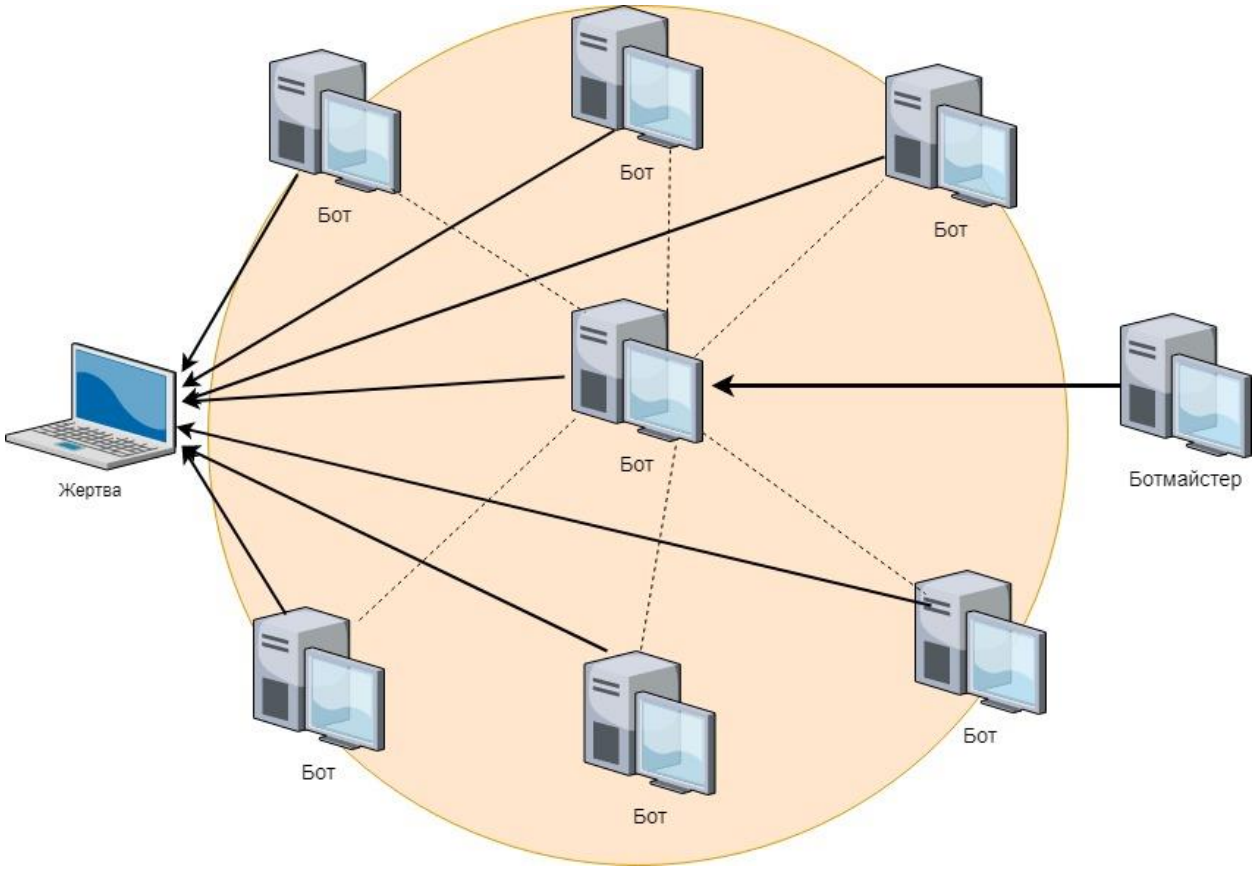

Рисунок 1.4 – Схема DDoS-атаки

Для переходу в стан початку атаки або її закінчення, ботнети, що використовуються для проведення атак відмови в обслуговуванні користуються будь якими з наявних і підтримуваних ними протоколів. Сучасні ботнети здатні виконувати DDoS-атаки різноманітних типів, зокрема:

• SYN Flood – атака, основною ідеєю якої є відправлення великого потоку особливим чином створених пакетів запитів на вузол-жертву, до стану, коли останній перестане реагувати на будь які запити.

• UDP Flood – атака, основною ідеєю якої є попереднє встановлення з сервером-жертвою з’єднання по UDP протоколу, з подальшим надсиланням останньому великої кількості «сміттєвих» дейтаграм.

• Ping Flood – атака, коли бот, використовуючи ICMP протокол, формує echo-запит. Цей запит надсилається на вузол-жертву використовуючи певний часовий інтервал.

• DNS Amplification – атака, при якій бот використовує DNS-сервери, надсилаючи на них величезну кількість запитів, очікуючи відповідь. Сервер не справляючись з вхідним потоком переходить до стану відмови в обслуговуванні.

• NTP Amplification – атака, основною ідеєю якої є масована відправка запитів на NTP-сервери, що призводить до стану відсутності реакції на вхідні запити.

Використання DDoS-атак в сучасному розумінні стає дедалі частішим способом недобросовісної конкуренції, методом вимагання фінансової винагороди в атакованих компаній, або як один з активних методів кібертероризму. З огляду на умови, більшість компаній не наважуються боротись з зловмисниками, надаючи перевагу швидкому вирішенню проблеми, просто заплативши останнім. Зазвичай самостійна ліквідація наслідків дорівнює величезним збиткам компанії.

Не можливо обходити стороною також такий факт використання ботнету як систему по збору конфіденційної інформації, до якої належать дані про банківські рахунки, номери кредитних карт, будь яка фінансова інформація, паролі до поштових скриньок, месенджерів, FTP-серверів.

За даними «Лабораторії Касперського», основним комерційним завданням наявних нині ботнетів, такими за яке ботмайстри отримують винагороду, є розсилка спам-пошти. До них належить до 80% завдань всього ботнет ринку. Щодо інших важливих властивостей ботнетів, то таким є збір конфіденційних даних, зокрема даних банківських рахунків. Більшість таких ботнетів по завершенні своєї роботи виконують операції по повній утилізації операційної системи на комп’ютері-жертві. Такі дії дозволяють на певний час

відкласти виявлення користувачем-жертвою ознак зловмисної діяльності з його банківськими даними, допоки останній не звернеться до банку з вимогою заблокування подальших транзакцій з його рахунків.

1.2 Поняття SIEM

ІТ системи великих підприємств постійно піддаються ризику атак з боку зловмисників. Впевнено можна сказати, що з плином часу складність цих атак зростає.

Зараження комп'ютера і діяльність шкідливого ПЗ відбувається непомітно для користувача, що ускладнює виявлення проблеми та її рішення. В сучасних ботнетах взаємодія з ботмайстром відбувається по зашифрованому каналу зв'язку через відкриті мережеві порти (80, 443 і т.д.).

Останнім часом розробникам шкідливого коду і операторам ботнетів стали доступні сервіси, які дозволяють автоматично відслідковувати виявлення нових зразків шкідливого ПЗ за допомогою різних антивірусних систем, а також вносити незначні зміни в код шкідливого ПЗ, роблячи вже створені антивірусні сигнатури неактуальними. Це значно знижує ефективність антивірусних систем і підвищує ризики для стабільності інфраструктури і безпеки корпоративних даних.

Сучасний розвиток ІБ спрямований більшою мірою на захист від проникнення зловмисника в мережу компанії, а також на припинення несанкціонованого доступу до цінних ресурсів. Однак, якщо зловмисник застосує принципово новий спосіб атаки, то існуючі засоби захисту вже не зможуть його зупинити. Іншим важливим фактом є неможливість своєчасно відстежити момент проникнення зловмисника в мережу підприємства та його дії, якщо вони не носять відкрито деструктивний характер. Відповідно, на

даний момент, всі процедури, пов'язані з розслідуванням таких дій, запускаються, тільки коли (якщо) їх все ж виявили. На жаль, на той час мінімізувати збиток вже не вийде.

Для того щоб боротися зі складними атаками, підприємству потрібні не тільки засоби захисту інформації, а й система, здатна відстежувати аномальну поведінку користувачів і IT-систем, а також своєчасно сповіщати про це профільних співробітників, здійснюючи проактивний підхід до роботи системи захисту корпоративних ресурсів.

Управління інформацією та подіями безпеки (SIEM) - це метод в управлінні безпекою, що поєднує в собі функціональність SIM (управління інформацією безпеки) та SEM (управління подіями безпеки), що зображено на рисунку 1.5. Основною ідеєю є створення єдиної системи управління безпекою. [4]

До SIM (Security Information Management) частини належить система, що базується на аналізі даних на основі відхилень від правил безпеки, та даних зібраної статистики.

SEM (Security Event Management) частина відповідає за захист в режимі реального часу. До основних функцій якої належить збір, перевірка на кореляцію подій інформаційних потоків, генерація повідомлень, що несуть превентивну дію.

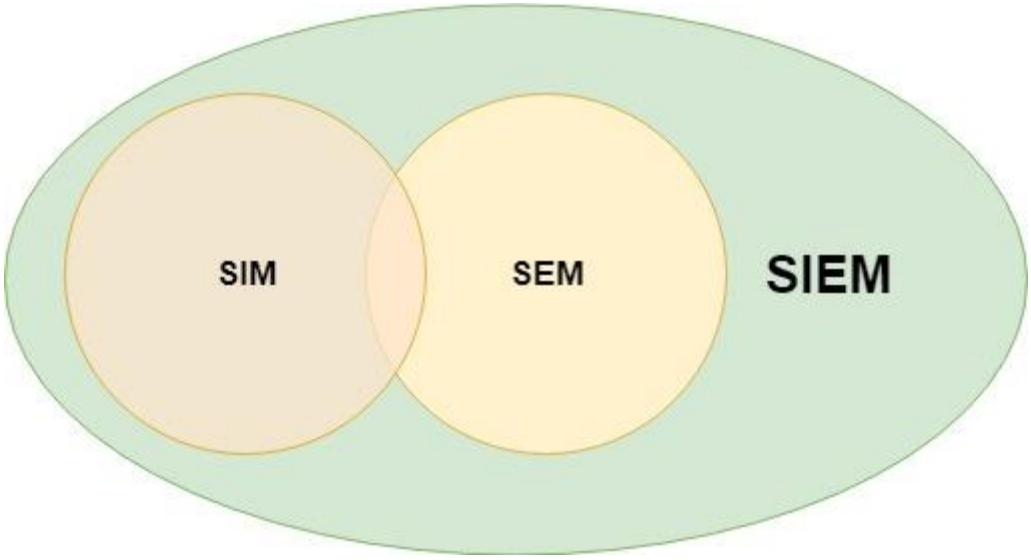

Рисунок 1.5 – Схема SIEM-системи

На практиці дані продукти часто накладаються один на одного, виконуючи однакові або дуже схожі функції, що спричинено неоднозначністю в розумінні конкретних меж функціональності кожної.

1.2.1 Механізм роботи SIEM

До основних завдань, що повинні виконуватись SIEM системами належать:

- Об’єднання, зберігання та первинна обробка журналів логів від різних джерел. Зазвичай такими джерелами виступають мережеві пристрої, програми, журнали логів ОС, засоби захисту.
- Надання інструментарію для активного аналізу даних журналів подій, та розбору виявлених інцидентів.
- Можливість проведення обробки даних у відповідності до написаних правил перевірки та можливість виявлення кореляції цих даних.

- Апарат з можливістю автоматичного сповіщення спеціаліста про виявлення підозрілої активності на основі написаних правил.

В грубому наближенні основним принципом роботи SIEM систем є циклічне виконання однакових дій: збір, аналіз, сповіщення. Система налаштована на збір відомостей з різних джерел, далі відбувається аналіз отриманих даних в режимі «реального часу», по-суті аналізуються дані останньої хвилини або навіть більшого проміжку, залежить від налаштувань, якщо щось підозріле було знайдено, формується інцидент і за бажанням відбувається сповіщення або певні превентивні заходи. Також відбувається систематизація баз даних, аналіз поведінки користувачів на основі попереднього моніторингу, виявлення більш точних особливостей передумов критичних подій.

Одним з ключових засобів, які використовує SIEM, є засіб збору та агрегації логів. Інформація надходить з безлічі різних джерел - мережевого обладнання, засобів захисту інформації, програм, СУБД та ін. Інформація потрапляє в SIEM або напряму з систем відслідковування (як правило це syslog, що використовується для інформації про стан системи), або через парсинг лог-файлів. В останньому випадку на систему, за якою слідкують, ставиться SIEM-агент, який відправляє інформацію на сервер агрегації даних.

Незважаючи на те, що для сучасних систем написано дуже багато конекторів, на підприємствах можуть використовуватися різні "самописні" додатки, які також необхідно відстежувати. Для включення їх в контур моніторингу існує можливість написання власних правил парсингу (за умови, звичайно, що додаток надсилає інформацію про свій стан в лог-файли). У загальному випадку інформація про ту чи іншу подію містить дані про час, код події, а також набір довільних полів даних, частина яких зустрічається дуже часто (адреса джерела події, ім'я користувача та інші.). Задавши в правилах

парсингу формат логів і призначення полів даних, ми зможемо використовувати ці дані в SIEM-системі.

Одним з ключових показників продуктивності SIEM-системи є кількість оброблюваних подій (наприклад, рядків в лог-файлах) за одиницю часу. Чим більша кількість систем відстежується, тим вищий цей показник, і, відповідно, серйозніші вимоги до SIEM-системі в частині процесингу і зберігання цієї інформації. Для територіально-розподілених компаній дуже важливим є питання масштабування системи, так як джерела даних можуть знаходитися на віддалених майданчиках і використовувати "вузькі" канали зв'язку. В цьому випадку використовуються модулі «препроцесингу», які розгортаються на віддалених майданчиках, збирають інформацію і відправляють її на сервер агрегації в оптимізованому вигляді. Після обробки інформація стає доступна спеціалісту, який може її використовувати для розслідування інцидентів, статистичного аналізу роботи IT-систем або аудиту.

При підключенні великої кількості джерел даних виникає питання аналізу одержуваної інформації. Ручна обробка потоку даних навіть не великої системи (3-5 тисяч подій в хвилину) неможлива, тому для моніторингу роботи систем використовуються різні способи кореляції подій, що забезпечують автоматизований аналіз вхідних подій і сповіщають спеціаліста лише в разі явної необхідності. Можна виділити два основних способи кореляції:

- кореляція на базі правил (Rule based correlation);

- кореляція без використання правил (Rule-lesscorrelation).

Кожен з цих способів має як переваги, так і певні недоліки.

Класичним є спосіб кореляції подій на базі правил. В цьому випадку при впровадженні системи ми описуємо певну послідовність логічних дій, які характеризують дії зловмисника. Наприклад, велика кількість спроб входу в будь-яку систему, що нагадує процес підбору пароля, яка закінчується успішним входом в цю систему. Перевагою цього способу є висока точність

виявлення зловмисника. До недоліків можна віднести необхідність періодичного оновлення правил кореляції і неможливість реагувати на послідовність подій, що не описана в правилах.

Кореляція без використання правил використовує ризик-орієнтований підхід, подібний до того, що використовується в банківських скорингових системах. В рамках цього підходу всі події мають певний рейтинг, і, коли рівень рейтингу послідовності подій з одним параметром (IP-адреса призначення, IP-адреса джерела, користувач та інші.) перевищує встановлений поріг, відбувається сповіщення про підозрілу подію. Перевагою цього способу є відсутність залежності від написаних правил і можливість виявлення нових векторів атак. До недоліків відноситься менший рівень точності, а також більша ймовірність помилкового спрацювання.

У деяких SIEM-системах з'явилася можливість збору і зберігання копій мережевого трафіку, що дозволяє значно підвищити ефективність майбутнього аналізу та розслідування можливих інцидентів. Після виявлення підозрілої активності в мережі підприємства, мережевий трафік між зловмисником і цільовою системою представляє особливий інтерес. Він дає аналітику широкі можливості для розслідування інцидентів і виявлення нанесеної шкоди. [4]

До основних джерел даних SIEM-систем належать різні корпоративні, тобто в основному комерційні, організації. Зокрема такі, що використовують різного роду системи контролю та аналізу трафіку та даних системних журналів, наприклад:

- Системи контролю доступу та аутентифікації, що використовуються для аналізу процесу отримання доступу до збору інформації.
- Журнали логів серверів та тонких клієнтів, що перевіряють відповідність отриманих прав доступу і політик ІБ.

- DLP-системи, що надсилають дані про несанкціоновану передачі даних за межі контрольованої мережі, а також дані про порушення пов’язані з отриманням привілеїв.
- Антивірусні платформи використовуються для попередження загроз, пов’язаних з роботою баз даних, програмного забезпечення, зміни політик конфіденційності та заміні файлів конфігурацій. А також про загрози пов’язані з використанням шкідливого коду.
- Системи веб-фільтрації аналізують, групують і надсилають дані, про шкідливі або заборонені з різних причин ресурси, що їх відвідує користувач.
- Ресурси IDS / IPS потрібні для передачі даних отриманих від аналізу трафіку, а саме дані про зміну прав доступу та будь які мережеві атаки.
- Міжмережеві екрани використовуються для отримання та передачі даних про шкідливе програмне забезпечення та небезпечні інциденти, що трапились.
- Устаткування мережі. Аналізує дані трафіку мережі, а також намагається контролювати використання користувачами інформаційних потоків.

Використовуючи наявні механізми кореляції в SIEM системах, а також механізми автоматизації процесів, можна доволі просто налаштувати безперервну роботу SIEM-системи. Саме правильне налаштування дозволяє підрозділам відповідальним за роботу системи, економити час на простих задачах, залишаючи його більше для вирішення важливих та особливо критичних загроз, працювати не безпосередньо з вхідними даними, а з інцидентами, ефективно виявляти аномальну поведінку, запобігаючи можливим фінансовим ризикам. Така робота системи дозволяє набагато краще виявляти різного роду збої та ризики в роботі інформаційних систем. [5]

В реальному житті, основною функцією SIEM-системи залишається збір та первинна обробка даних журналів подій від різного роду джерел, що є безпосередньою зоною відповідальності SIM, щодо SEM частини, то від неї використовуються лише правила кореляції, проте і вони рідко оновлюються.

1.3 Використання SIEM для виявлення ботнету

Використання заданих критеріїв безпеки для нормального функціонування системи дозволяє аналізуючи різного роду отримані дані, з легкістю виявляти певні аномалії та відхилення . Оскільки, кількість джерел інформації, зазвичай, доволі велика, в них набагато легше виявляти нові підозрілі поведінки дій користувачів, або комп'ютерних систем, а отже вони ідеально підходять і для виявлення та аналізу діяльності ботнетів. Маючи точний набір даних роботи ботнету можна з легкістю визначити його структуру, знайти IP-адреси ймовірних C&C-серверів, або ефективно боротись з останніми.

Для ефективного виявлення даних ботнету, SIEM-системи повинні проводити безперервний аналіз даних.

До методів безсигнатурного аналізу, що використовуються SIEM-системами належать статичні, кореляційні, методи базовані на певних правилах, графах та інші. [5]

• Statistical – один з складних типів безсигнатурного методу аналізу, базований на кореляції подій. Основною ідеєю є обчислення статистичного зв'язку між двома або більше різними змінними.

• RBR Rule-based (pattern based)- метод, в якому відношення між подіями визначаються на основі попередньо заданих правал, сформованих аналітиками.

• CBR Codebook based. – це метод, в якому кореляція проводиться за відповідними векторами з попередньо заданої матриці подій.

• MBR model based reasoning – метод, що базується на абстракції об'єктів і використовує модель для спостереження за ними.

• Graph based – метод, що використовує властивість залежності між системними компонентами, будує на їх основі граф, та використовує його для знаходження основної причини проблеми.

• Neural network based – метод, основна ідея якого полягає у використанні машинного навчання, а саме нейронних мереж. Мережа тренується для виявлення аномалій в потоці подій.

Сигнатурні методи відрізняються від безсигнатурних тим, що вони налаштовані на виявлення збігів за допомогою деякої кількісті одночасно спрацьованих правил. Кожне, окреме правило розробляється під окрему проблему, але такий спосіб означає, що по одному інциденту можливе спрацьовування кількох правил одночасно. До складу правила входить тригер, що потрібен для спрацювання у разі виконання його внутрішніх умов. У разі спрацювання тригер виконує спеціально прописаний сценарій. Також тригер містить лічильник, що потрібний для підрахунку спрацювань останнього. [6]

Прикладом роботи сигнатурного методу є виявлення ботнет-DDoS-атаки.

DDoS - це атака, основна ціль якої введення комп’ютера-жертви в стан, коли легальні користувачі системи не можуть отримати доступ до наданих їм системних ресурсів. DDoS-атака можливо реалізувати використовуючи різні

підходи. Наприклад, для SYN-флуд атаки характерна передача особливих TCP пакетів. В умову спрацьовування тригера записують максимальне число «напіввідкритих» з'єднань з однієї IP адреси, за певний час (сесію).

Якщо DDoS-атака використовує HTTP-флуд, то в тригер записується максимальна кількість з’єднань на 80 порт, зазвичай це середньостатистичне за певний проміжок часу нормальної роботи, а також кількість одночасних процесів Apache.

Я в першому так і в другому випадку, основною задачею тригера буде очікування збігу обставин для спрацювання його внутрішніх умов.

Реакція тригера в більшості випадків дозволяє точно визначити наявність атаки, а отже більш конкретно визначити набір даних, що причетні до цієї атаки, що в свою чергу спрощує роботу по виявленню ботнету.

Висновок до розділу 1

Розглянувши в першій частині розділу поняття ботнету, його будови та методів використання, можна з впевненістю сказати, що дана технологія це довгий час буде використовуватись зловмисниками в якості грізної сили при виконанні будь яких протиправних дій в інформаційному просторі. Ботнети можуть застосовувати безліч методів зараження та зомбування комп’ютерів-жертв, при цьому ці техніки безперестанно удосконалюються на мутують, стають дедалі складнішими та витонченішими. Вже сьогодні застосування однієї технології захисту не може гарантувати хоч якийсь захист від усіх вихідних загроз. Ботнети – складні мережі об’єднаних навколо зловмисника машин-ботів. Нові ботнети несуть нові загрози, боротьба з якими з часом стає все складнішою, вимагаючи дедалі більших матеріальних та людських ресурсів.

Найбільшу загрозу нині представляють децентралізовані типи ботнетів. В минулому боти підключались до каналу загальної передачі команд, слухали його і реагували на отримані завдання, зараз такі системи перетворились в абсолютно нові структуру, в яких відсутній єдиний центр збору, а отже відсутня одна з найбільших проблем ботнетів минулого. Тепер не можливо закрити центр керування, хоча б лише через те що ніхто точно не може сказати де він, кожен рядовий бот одночасно виступає сервером керування. Можливо в майбутньому еволюція ботнетів поверне на шлях використання вразливості XSS.

Аналізуючи дані минулих часів, та знайомлячись з сучасним положенням справ можна припустити, що дослідження нових методів виявлення та ліквідації ботнетів найближчим часом займатиме провідні позиції у сфері інформаційної безпеки.

Щодо SIEM, то система, що дозволяє отримувати нагальну та повну інформацію про стан IT-інфраструктури підприємства беззаперечно буде одним з основних та найефективніших методів в боротьбі з добре відомими та абсолютно новими атаками та загрозами. Властивість гнучкого налаштування, наявність величезної кількості додаткових інструментів, що значно полегшують роботу з даними сприяє дедалі ширшому впровадженню останніх на малих та середніх підприємствах, що в свою чергу сприяє більш ефективній роботі систем з даними реального світу користувачів, дозволяючи виявляти загрози практично в момент їх першої появи в мережі.

Безумовно, використовуючи SIEM-системи, можна набагато ефективніше та легше виявляти структури ботнетів, але це не означає, що останні зникнуть. Ботнети будуть лише розвиватись, покращуватимуть методи проникнення, шифрування, маскування і єдиним ефективним засобом боротьби проти них стануть системи, що будуть безперервно відслідковувати поведінку, виявляти аномальну діяльність, безперестанно оновлюватись, це

будуть аналоги сучасних SIEM-систем, людський фактор в яких буде мінімальним.

2 МЕТОДИ ВИЯВЛЕННЯ БОТНЕТУ

З отриманого із SIEM-системи набору даних, виділити потрібні та правильно їх обробити можна різними способами

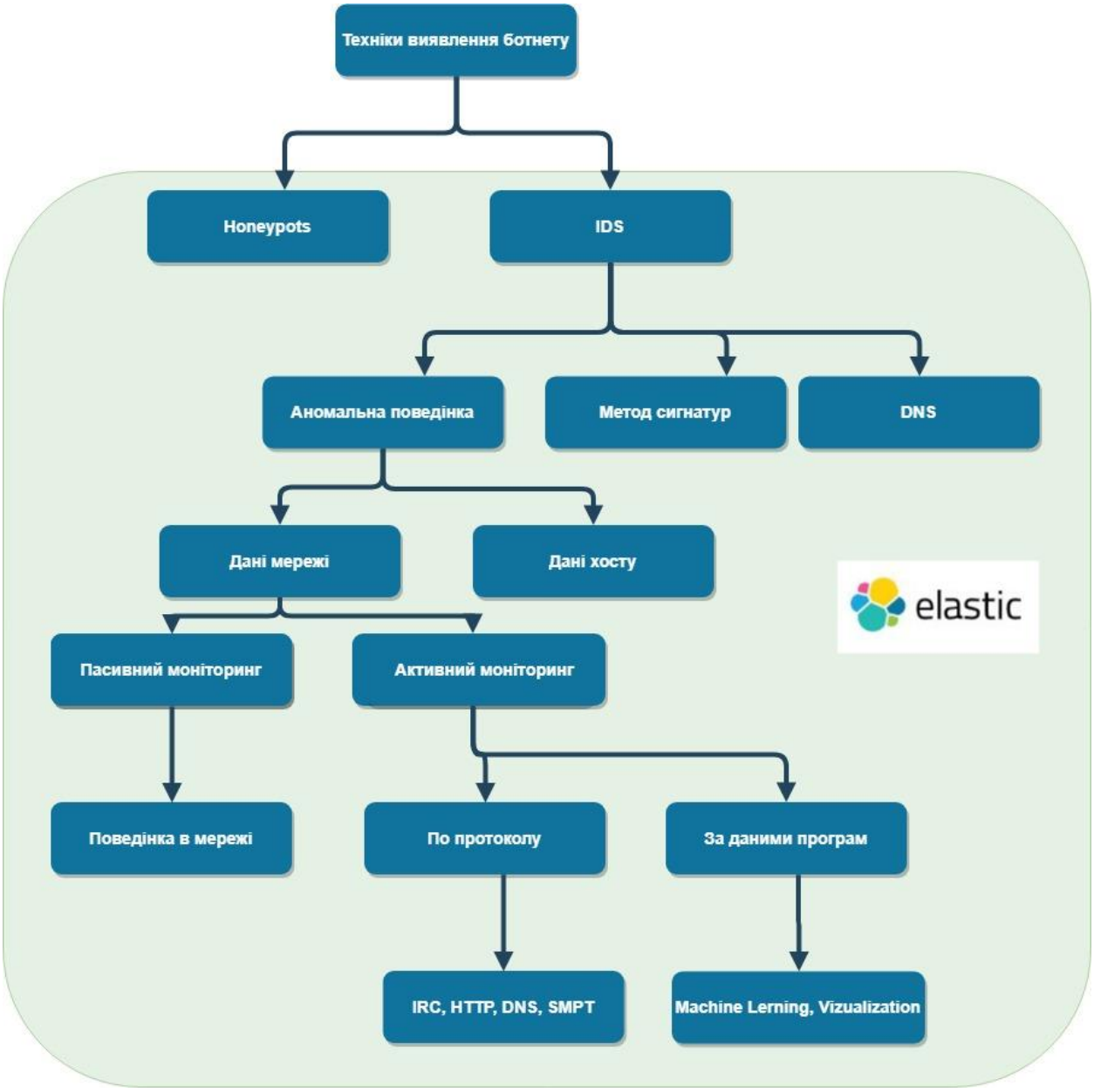

Рисунок 2.1 – Схема сучасних методів виявлення ботнету

В останні роки спостерігається зростання інтересу до методів виявлення та профілактики ботів. Звичайно, важливим є дізнатися про те, як бот заражає комп'ютери, це спонукає до вирішення проблем в безпеці комп’ютерних систем, проте, якщо хост вже заражений, найбільш важливими стають методи, що найточніше зможуть виявити заражену машину, перш ніж вона буде використана для виконання шкідливих дій. Існує низка підходів для виявлення ботнету. Ці підходи можна класифікувати на основі сигнатур, аномалій поведінки мережевого трафіку, DNS та інтелектуального аналізу даних. Деякі вчені поділяють системи виявлення P-2-P-ботнету на три типи: інтелектуальний аналіз даних, машинне навчання та поведінку мережі та аналіз трафіку. Також існує класифікація методів детектування ботнету на основі методу машинного навчання з автоматичним навчанням та без нього.

2.1 Виявлення на основі сигнатур

Це підхід для виявлення ботнету шляхом співставлення всіх вхідних даних на хості або мережі з базою даних, що містить відомі дані сигнатур ботнетів, або скоріше сигнатур трафіку, характерних для ботнетів. Головною перевагою такої системи є точність ідентифікації при виявленні відомих ботнетів. Як наслідок - низький відсоток помилкових спрацювань, проте, такий підхід не зможе виявити ботнет, сигнатура якого не була раніше вписана в базу даних. Таким чином, системи виявлення на основі сигнатур самі по собі потребують постійних оновлень, щоб підтримувати ефективність у виявленні нових типів ботнету.

2.2 Використання Honeypots

Як тільки розпочався моніторинг ботнетів дослідники використовували honeypots для відслідковування централізованих ботнетів. Honeypots або honeynet - це машина або мережа машин, призначена для того, щоб здаватися звичайним користувачем в очах зловмисників. Honeypots вважаються не складним інструментом у використанні, оскільки для проведення моніторингу не потрібно попереднє знання шкідливого програмного забезпечення або протоколів зв'язку.

Шкідливе ПЗ, що заражає honeypot, зв'язується з C&C, наприклад, IRC, щоб повідомити про зараження ботмайстру. За допомогою моніторингу мережевого трафіку, що генерується honeypot, можна виявити деталі C&C, наприклад, IRC-сервери. Крім того, інформація про інші заражені машини, які відправляють звіти C&C, може бути отримана шляхом перевірки журналів зв'язку C&C.

Проблеми, що виникали в процесі роботи з honeypots, сприяли активному розвитку таких систем моніторингу бот-мереж як сканери та сенсори. Обидва надають можливість звичайному користувачеві повністю керувати процесом моніторингу. Як приклад повного контролю, користувач може самостійно налаштувати систему так, щоб вибірково відповідати або надсилати нові команди іншим ботам. Також існує можливість активного моніторингу передачі повідомлень між ботами, що дозволяє ефективно встановлювати зв’язки з іншими ботами, значно збільшуючи охоплення мережі.

2.3 Сканери

Через децентралізовану архітектуру P-2-P-ботнетів, боти мають можливість сканувати мережу на наявність активних ботів-сусідів. Зазвичай, така операція проводиться у випадку, коли число активних сусідів в їх NL(лист сусідів) падає нижче деякої встановленої норми. Саме спостереження за даним процесом використовується сканером, що є своєрідною комп'ютерною програмою, реалізованою для імітації поведінки звичайного бота, який не відповідає на запити сусідів, проте сам активно здійснює запити. Метою сканування є представлення точного вигляду ботнету шляхом отримання інформації про всі заражені хости, а також про взаємозв'язки між ними. Кожне сканування створює певного роду карту, що складається з орієнтованого графа, який є графічним зображенням виявленого бота разом з усіма його сусідами.

2.4 Сенсори

Через широке використання мережевих пристроїв, які допускають спільне використання IP-адрес на багатьох машинах, наприклад NAT, бот-майстри стикаються з проблемою коли мають два різних класи пристроїв: superpeers і non-superpeers. Оскільки однорангові вузли всередині NAT-мережі недоступні напряму, ботнети використовують дворівневу структуру мережі. Вона потрібна для того, щоб всі боти, пов’язані в мережу, могли залишатись зв'язаними між собою і з ботмайстром. Боти, чия роль не superpeers, надіються лише на те, що боти з іншою роллю зможуть підключитись до основного ботнету та отримати звідти команди, в протилежному випадку вони просто випадають як учасники ботмережі. Команди від ботмайстра отримуються

шляхом запитів до superpeers на наявність оновлень. Таким чином superpeers-боти можуть передавати будь-яку інформацію від бот-майстрів далі у внутрішню мережу, така структура дозволяє обходити проблеми NAT.

На основі такого принципу дворівневої мережі можливо побудувати систему моніторингу ботнетів. Канг з співавторами запропонували механізм, званий датчиками, для перерахування структурованих P-2-P-ботнетів. Датчики мають пряму маршрутизацію і розгортаються з використанням стратегічних ідентифікаторів DHT, призначених для перехоплення запитів маршрутів інших ботів. Так як запити були ініційовані самими ботами, датчики можуть розпізнати non-superpeers бота на основі перехоплених повідомлень запитів. На відміну від сканування, сенсори можуть виявляти як superpeers, так і non-superpeers.

2.5 Метод аномалій

Основною ідеєю даного методу є виявлення ботнету на основі даних аномальної поведінки мережевого трафіку, наприклад нестандартно високу затримку в мережі, трафік на нестандартних портах, великі об’єми мережевого трафіку, нестандартна поведінка системи. Все це може вказувати на підозрілу активність, що зазвичай є наслідком роботи в мережі різного роду ботнетів, таких як спамери. [7]

Методи виявлення на основі аномалій можна розділити на дві категорії: виявлення на основі хоста та виявлення на основі мережі.

Техніка виявлення на основі хоста -це метод, основна ідея якого полягає у відстежуванні та аналізі роботи внутрішніх компонент хоста, не використовуючи дані про зовнішню мережеву діяльність. Такий підхід дозволяє відстежувати та виявляти будь яку незвичну, підозрілу поведінку

окремого комп’ютера, наприклад надмірне використання ресурсів для обробки даних сумнівного походження , та відразу ж повідомляти про неї адміністратора з безпеки або безпосередньо користувача. Використовується спосіб порівняння образу системи до інциденту та після, співставляючи однакові та змінені фрагменти. Зміна або видалення критичних або важливих файлів, є першою ознакою підозрілої поведінки, що відразу повинна бути відправлена на розслідування адміністратору. [8]

Техніка виявлення на основі мережі - це метод, що базується на виявленні ботнету активно відстежуючи та аналізуючи дані мережі.

2.6 Активний моніторинг

Метод, основною ідею якого є здатність вводити певні текстові пакети в мережу, та очікувати відповіді на них. В залежності від введених пакетів, адміністратор може визначати хто керує хостом, реальний користувач або ж бот. Іноді такий спосіб призводить до надмірного збільшення трафіку в мережі. Найкраще цей метод зарекомендував себе у виявленні ботнетів на основі IRC .

2.7 Пасивний моніторинг

Метод, що базується на детальному аналізі трафіку даних в мережі, шукаючи будь яку підозрілу активність, яка може бути наслідком роботи ботнету або сервера керування. Такий метод не збільшує навантаження на мережу, адже не вмішується в роботу останньої, а просто аналізує отримані дані.

Зазвичай боти намагаються отримати доступ до керуючого сервера використовуючи DNS-запити. Вони шукають конкретний сервер, що зазвичай надається постачальником динамічного DNS. На основі даних про DNS-пакети можна побудувати систему виявлення, та аналізу на аномальну поведінку даних мережі. Оскільки ботнети часто користуються DNS, наприклад для збору відомостей про інших ботів, для запуску атак та оновлення, їх можна легко виявляти.

2.8 Машинне навчання

Найновітніший метод виявлення ботнету, що може включати в себе всі попередні пункти. Базується на математичних алгоритмах, спроектованих таким чином, що прийнявши на вхід масив даних, на виході отримають структуровану систему ймовірностей, або дерево, що з високою ймовірністю візуалізує топологію ботнету. Цей метод може використовуватись в мережах для виявлення ботнету на стадії, коли той ще не несе ніякої загрози.

Висновок до розділу 2

В даному розділі розглянуто різні сучасні методи моніторингу ботнетів, зокрема: honeypots, сенсори, сканери та інші. Honeypots широко використовуються для виявлення ботнету, оскільки вони не потребують складного налаштування, проте вони обмежені в своїх можливостях. Сканери, навпаки, можуть детально визначати взаємозв’язок між ботами в мережі, проте не здатні виявити тих, що знаходяться за NAT. На противагу сканерам можна використовувати сенсори. Вони здатні виявляти ботів за NAT, проте гірше справляються з задачею визначення взаємозв’язку між ботами.

Найкращим варіантом вважається одночасне використання і сканерів і сенсорів. Інші методи виявлення побудовані на аналізі отриманих даних, зокрема, відхиленні цих даних від деякої норми. Методи машинного навчання базуються на певному періоді навчання машини, коли вона визначає основні характеристики даних, а далі на основі отриманих результатів перевіряє їх кореляцію з даними отриманими з реальної системи. Саме метод на основі машинного навчання вважається одним з найперспективніших напрямів виявлення ботнету на сьогоднішній день.

3 ПРАКТИЧНЕ ВИЯВЛЕННЯ P-2-P БОТНЕТУ ЗА ДОПОМОГОЮ SIEM-СИСТЕМ

3.1 Розгортання середовища збору даних

Робота з отриманими даними журналів може бути доволі складним завданням, враховуючи те, що вони містять велику кількість не фільтрованої інформації про безпеку, продуктивність та використання програм та безпосередньо хоста. Ефективна робота, аналіз та управління цими даними потребують попередньої їх обробки, а це в свою чергу займає велику кількість і часу і ресурсів.

Одним з найкращих варіантів для отримання та первинної обробки даних мережі може використовується ELK Stack. Це поєднання трьох OpenSource проектів: Elasticsearch, Logstash і Kibana з різними можливими надбудовами. [9]

Elasticsearch - це відкрита програма, що реалізує повнотекстовий механізм пошуку та аналізу, заснована на пошуковій системі Apache Lucene. Вона є головним елементом ELK Stack. Особливостями її є централізоване зберігання даних, що потрібно для того, щоб полегшити роботу по пошуку та перевірці різного роду аномалій або невідповідностей. В Elasticsearch реалізовано механізм, що дозволяє виконувати та комбінувати різного роду пошуки, наприклад пошук за структурою, гео, метричний та інші. Побудова програми здійснена на мові програмування Java, яка в свою чергу дозволяє розгортати її на різних платформах. Саме мультиплатформеність найбільш сприяє дослідженню великих об’ємів даних на різних ОС на високій швидкості.

Logstash - це механізм обробки даних журналів на стороні сервера, що одночасно виконує функцію приймача дані з різного роду джерел, далі перетворює їх до певного прийнятного виду, а потім відправляє на основну обробку до Elasticsearch. Вхідні дані часто розпорошені по великій кількості систем у різних форматах, саме Logstash сприяє приведенню такого різноманіття до однакового вихідного формату. Вбудовані механізми дозволяють легко отримувати дані журналів, метрик, веб-додатків, сховищ даних і різних сервісів AWS, в безперервному режимі. Logstash має фреймворк, що містить більше 200 плагінів, які дозволяють комбінувати та організовувати отримання даних з різного роду вхідних джерел, фільтруючи та форматуючи їх на виході.

Kibana - це ще одна платформа призначена для візуального аналізу та побудови різноманітних графіків та діаграм, що працює безпосередньо з Elasticsearch. Kibana використовується для полегшеного пошуку, перегляду та взаємодії з даними, що були проіндексовані та зберігаються в Elasticsearch. Вона надає можливість легко виконувати розширений аналіз даних і візуалізувати їх в різних діаграмах, таблицях та картах. Простий, зрозумілий інтерфейс на основі браузера значно спрощує роботу та дозволяє швидко створювати динамічні панелі, потрібні для підвищення ефективності роботи з даними. Зазвичай вони відображають зміни в Elasticsearch в режимі реального часу.

Рисунок 3.1 – Схема ELK Stack

Beats - це програми, що представляють собою набір ресурсоефективних функцій, які використовуються в якості первинних збирачів даних журналів. Вони діють як агенти, встановлюючись на різних серверах наявної інфраструктури по збору логів або метрик.

Beats існують в різних формах. Це можуть бути програми для збору даних файлів журналів (Filebeat), мережних даних (Packetbeat), метрик серверів (Metricbeat) або будь-яких інші типів даних. Дані практично будь якого роду можуть бути зібрані Beats, завдяки безперервному оновленню наявних та розробці нових типів Beats розробниками Elastic та спільнотою. Після збору дані надсилаються або безпосередньо в Elasticsearch, або в Logstash для додаткової обробки та форматування. [10]

Beats може бути встановлений практично на будь-яку операційну систему.

Також до ELK стеку в роботі було включено систему виявлення та попередження мережевих вторгнень Suricata.

Suricata – це система виявлення загроз, яка володіє широкою функціональністю. Вона може виступати в ролі системи виявлення вторгнень (IDS), або як система запобігання вторгненням (IPS). Також її можливо використовувати як систему моніторингу мережевої безпеки.

Suricata може налаштовуватись як IDS на основі хосту для моніторингу трафіку тільки одного комп’ютера, або ж в якості пасивного IDS для моніторингу всього трафіку, що проходить через мережу. При виявленні шкідливої активності, можна налаштувати функцію сповіщення аналітика. В якості активної вбудованої IDS та IPS Suricata використовується для моніторингу вхідного та вихідного трафіку. [11]

Suricata надає певний перелік вбудованих правилами безпеки, які можна доповнювати будь якими власними за бажанням. Ці правила

генерують дані логів або події безпеки, і для ефективного зчитування, прийому, зберігання та аналізу використовується ELK Stack

3.2 Збір та підготовка даних

Для виконання завдання по збору даних було розгорнуто наступні експериментальні хости:

10.10.198.25 – для Elasticsearch та Kibana

10.10.198.21 – для Logstash

10.10.198.11 – для Suricata

10.10.198.1 – для робочої станції Windows

Спрощена таксономія руху зібраних даних виглядає наступним чином:

- На робочих станціях встановлюються Beats, для збору заданих у файлах конфігурацій типів логів. Ці дані перенаправляються до хоста, що стоїть вище в ієрархії обробки та фільтрації, тобто до logstash
- Хост зі встановленою на ньому Suricata, на основі прописаних правил, виявляє підозрілу діяльність, формує логи та відправляє їх вище – на logstash.

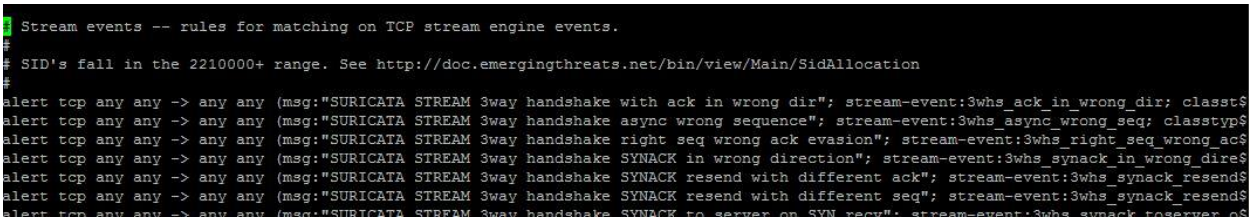

Рисунок 3.2 – Правило Suricata

Ефективність роботи таких систем напряму залежить від кількості прописаних правил, тому найкращим варіантом було б весь час оновлювати останні на основі даних про нові типи загроз.

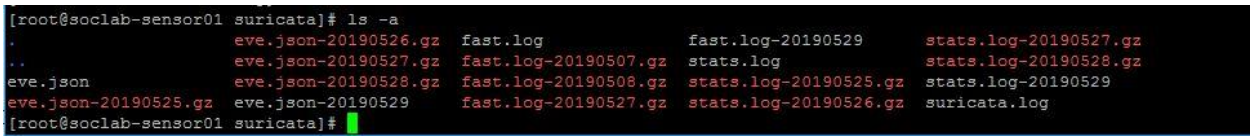

Рисунок 3.3 – Приклад отриманих даних від Suricata

- **suricata.log:** тут містяться дані, зібрані при запуску самої Suricata
- **stats.log:** тут містяться дані про статистику мережевого трафіку
- **fast.log:** дані про підозрілу активність, виявлену при роботі Suricata
- **eve.json:** тут містять дані трафіку локальної мережі в форматі JSON-повідомлень, а також сповіщення, надіслані до fast.log у форматі JSON

- Дані, що приходять до Logstash також піддаються певному фільтруванню та обробці, що визначена в файлах конфігурації та правилах logstash.

```
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
  beats {
    port => 5044
  }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
    #user => "elastic"
    #password => "changeme"
  }
}
```

Рисунок 3.4 - правило передачі даних від Beats до Elasticsearch з додаванням відомостей про версію та час

- Наступним кроком є обробка логів безпосередньо на Elasticsearch. Аналіз даних та подальше сповіщення про певні події можливе завдяки використанню правил. Правила пишуться у форматі yaml.

```
name: Brute force detection
index: kolide-*

type: frequency
timeframe:
  hours: 1
aggregation:
  hours: 1
aggregation_key: ip
num_events: 8
filter:
- query:
    query_string:
      query: "_type:logs AND error_code:535 AND NOT initial_connection:true"

query_key: [ip, failed_login]
include: ["ip","failed_login","password_hash"]
top_count_keys: ["ip","failed_login"]
```

Рисунок 3.5 - правило для виявлення діяльності, що нагадує брутфорс

Знаючи поведінку ботнету, можна написати правила, що повністю перекриють його діяльність і зможуть точно визначити його наявність в отриманих даних (сигнатурний підхід).

Інший метод виявлення ботнету базується на аномальній поведінці. Для цього ELK Stack може використовувати Kibana, сервіс, що дозволяє легко візуалізувати отримані дані, наочно виявити аномальну поведінку певних полів, використовуючи різні фільтри, схеми та графіки.

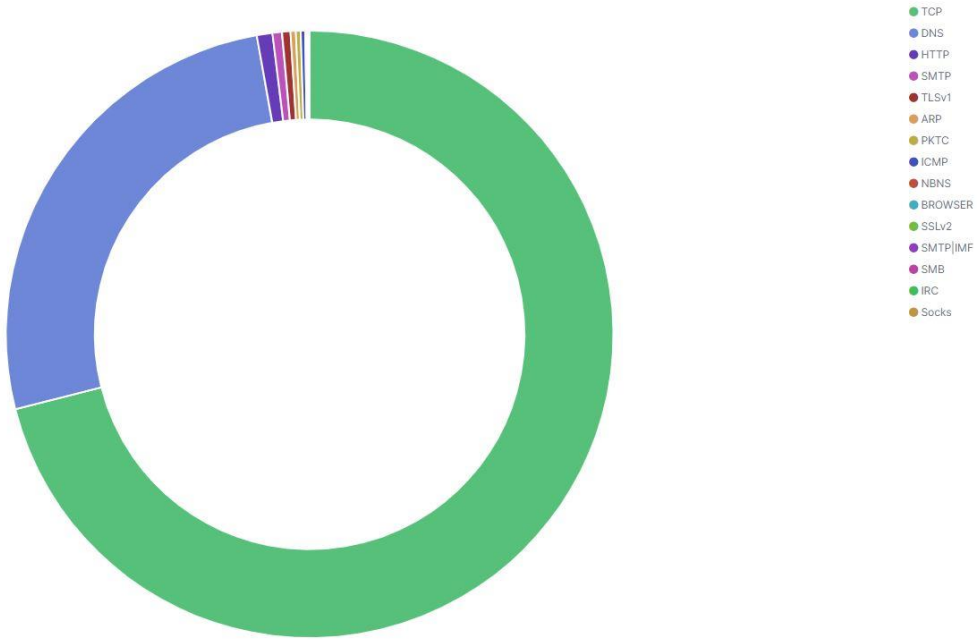

Рисунок 3.6 - дані про використання різного роду протоколів на одному хості, за останні 5 хв

Використовуючи можливості Kibana аналітик може переглядати різні параметри вхідних даних практично в реальному часі:

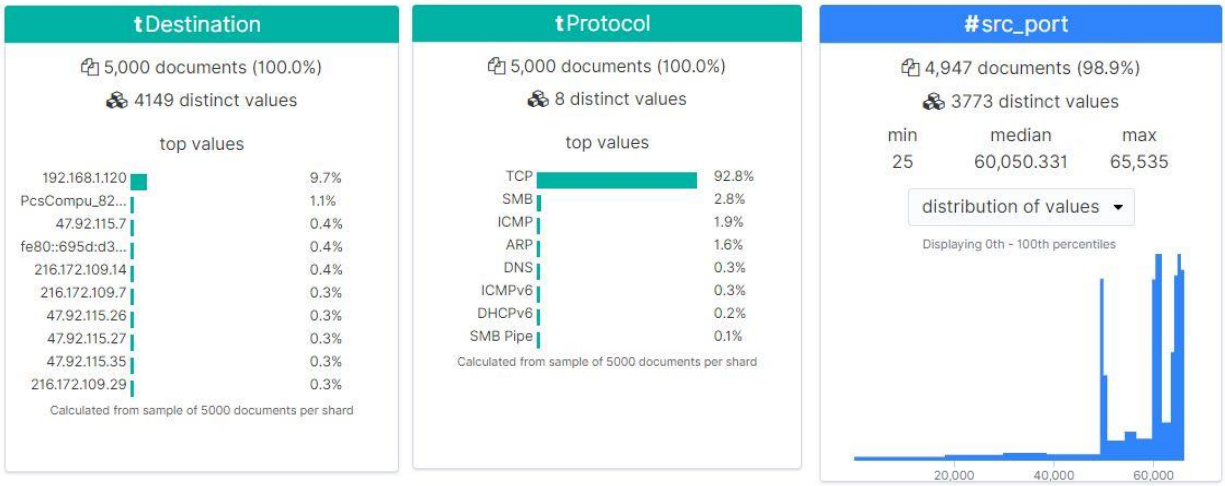

Рисунок 3.7 – Приклад роботи засобів візуалізації Kibana в реальному часі

Це дозволяє проаналізувати поведінку ботнету в момент його діяльності.

Машинне навчання може використовуються для виявлення аномалій, як в режимі реального часу так і при обробці збережених даних ботнету. ELK Stack надає можливість виявлення аномального трафіку використовуючи вбудовану систему аналізу

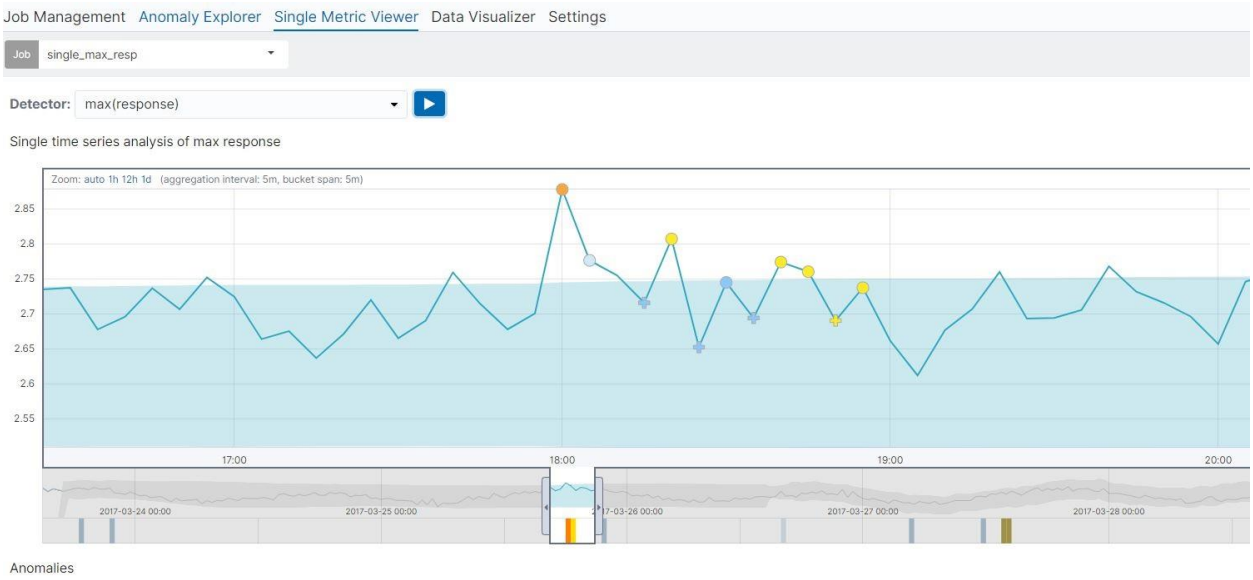

Рисунок 3.8 –Приклад роботи системи по виявленню аномалій

Така система порівнює вхідні дані з даними, що були в Elasticsearch раніше, і сповіщає аналітика про відхилення трафіку від норми.

Проблемою такого способу виявлення ботнету є відсутність даних про попередню поведінку ботнету, тобто не можливість заздалегідь передбачити атаку, а також відсутність системи у відкритому доступі.

3.3 Використання ML для аналізу трафіку SIEM-систем

Машинне навчання є поширеним методом виявлення бот-мереж. Комунікаційна інформація, яка може використовуватись в бот-мережі, стає параметром функції машинного навчання. Аналізуючи поведінку параметрів, ці функції можуть виявляти нову комунікаційну інформацію, а отже знаходити нові ботнети.

Характерні особливості для ботнету отримуються шляхом аналізу, навчання і тестування даних. Щоб підвищити ефективність і гнучкість функцій по виявленню ботнетів в процесі машинного навчання, необхідно підбирати функції з найкращою відповідністю до даних, які найбільше впливають на виявлення ботнетів. В P-2-P ботнетах кожен вузол може використовуватися в якості як джерела так і розповсюджувача інформації, що підвищує складність їх виявлення.

Результати роботи алгоритмів машинного навчання, наприклад алгоритмів дерев рішень, використовуються для аналізу типової поведінки ботнету. На основі цих результатів можуть будуватись правила для сигнатурного методу виявлення ботнетів.

В роботі використовується система машинного навчання з відкритим вихідним кодом Weka. Це набір алгоритмів машинного навчання для інтелектуального аналізу даних. Цей інструмент використовувався для

порівняння продуктивності та точності алгоритму на різних датасетах та при різних параметрах вхідних даних.

Серед різних класифікаторів одним з основних алгоритмів при виявленні ботнетів служить класифікатор дерева рішень , який, як випливає з назви, побудований у формі дерева. В якості класифікатора використовується дерево рішень J48, реалізоване на JAVA на основі алгоритму C4.5. Класифікатор використовує параметри для побудови моделі і класифікує вхідні дані відповідно до їх характеристик.

Алгоритм C4.5, розроблений Россом Куінланом, для генерації дерева рішень з метою класифікації. Він став популярним після того, як зайняв перше місце в статті під назвою «10 кращих алгоритмів інтелектуального аналізу даних», опублікованій в лекційних нотатках Springer з комп'ютерних наук (LNCS) в 2008 році.

В якості тестових модулів використовувались два набори даних: Zeus та Waledac, що знаходяться у відкритому доступі. Набір даних реального трафіку був отриманий з Elasticsearch.

В якості параметрів вхідних даних для тренування були вибрані Source, Destination, Protocol, Time та Length.

Створення моделі проводилось технікою перехресного підтвердження (Cross-validation). Тобто набір даних розділявся на 10 частин, з подальшим вибором однієї частини для тесту та 9 для тренування. В кінці-кінців отримують 10 результатів оцінки які усереднюються. Виконавши 10-кратну перехресну перевірку WEKA запускає алгоритм навчання 11 раз для всього набору даних, щоб отримати якнайкращу модель.

Для оцінки ефективності запропонованого методу такі показники, як точність і коефіцієнт помилкових спрацьовувань.

TPR – правда позитивна (шкідливе, правильно виявлене як шкідливе)

TNR – правда негативна (звичайне, правильно виявлене як звичайне)

FPR – брехня позитивна (звичайне, виявлене як шкідливе)

FNR – брехня негативна (шкідливе, виявлене як звичайне)

ACC – точність (вказує відсоток правильних прогнозів для всіх екземплярів)

Результат тестування даних отриманих з Elasticsearch та даних відкритих датасетів ботнетів наведено в таблиці 3.1

Таблиця 3.1 Дані, отримані при роботі з J48

Модель	TPR	TNR	FPR	FNR	ACC		
Zeus	0,978	0,987	0,013	0,022	0,9855	Лабораторні дані	
Waledac	0,994	0,996	0,004	0,006	0,9952		
Zeus	0,971	0,987	0,013	0,029	0,9843	Дані Elasticsearch	
Waledac	0,945	0,996	0,004	0,055	0,9735		
Zeus	0,963	0,987	0,013	0,037	0,9829	без Source	
Waledac	0,96	0,996	0,004	0,04	0,9804		
Zeus	0,934	0,996	0,004	0,066	0,9857	без Destination	
Waledac	0,688	0,996	0,008	0,312	0,8615		

Також було проведено дослідження на іншому алгоритмі.

Таблиця 3.2 Дані, отримані при роботі з SVM

	TPR	TNR	FPR	FNR	ACC
Waledac	0,956	0,896	0	0	0,926

Як бачимо з таблиці, TPR нашої моделі трохи зменшився, проте він все одно залишився на хорошому рівні. Це говорить про те, що вага різних параметрів даних різна, що Source та Destination відіграють одну з головних ролей у виявленні ботнетів. Також на результат могла вплинути наявність в даних, отриманих з Elasticsearch, певної кількісті логів, які належать не ботнетам, а іншим системам, що користуються розподіленою архітектурою.

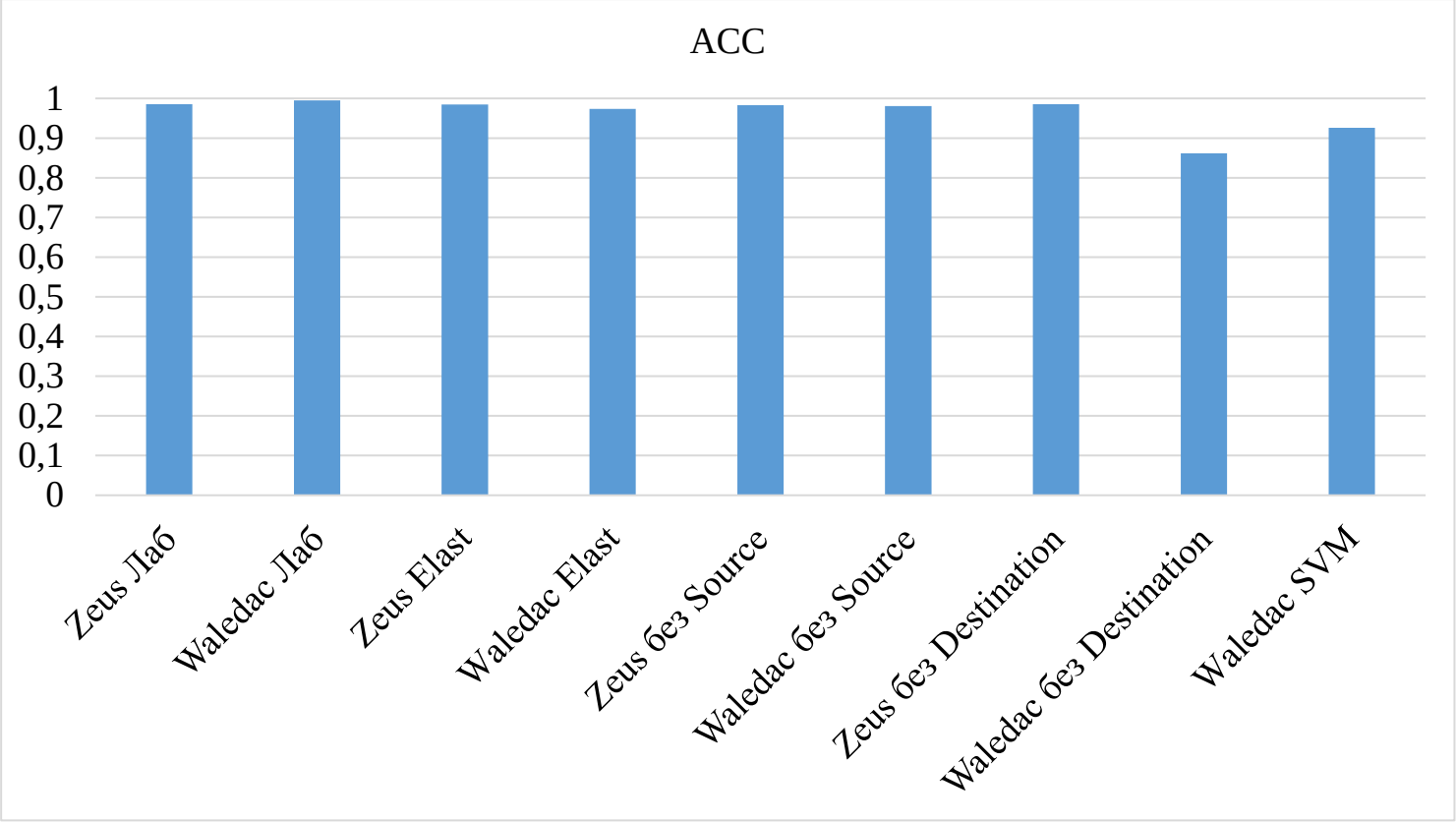

Рисунок 3.9 – Графік отриманих результатів точності

З рисунка 3.9 видно, що дані отримані в результаті тестувань трохи відрізняються. При використанні всього набору параметрів результат кращий,

проте проблемою такого методу є колосальне збільшення часу на навчання та тренування. Менш вибагливі умови перевірки дають гірший результат, проте працюють швидше.

Результат, все ж позитивний, а отже такий метод можна вважати ефективним способом виявлення p-2-p ботнету. Оскільки обидва алгоритми це дерева, то їх можна побудувати, та візуально прослідкувати структуру, виявивши, в деяких випадках хости вищих рівнів, процедури передачі команд.

Використовуючи дані тестів машинного навчання, можна зрозуміти поведінку ботнету, а отже створювати правила напряму в Elasticsearch та Suricata, прискорюючи виявлення останніх.

Висновок до розділу 3

Даний розділ було присвячено розгортанню системи по збору даних, а також їх первинній обробці, фільтрації, підготовці до використання в Elasticsearch та подальшій обробці методами машинного навчання.

Було представлено різного роду налаштування, що використовувались для цього. Отримані дані були проаналізовані вбудованими в Elasticsearch та Kibana інструментами. Також було проведено тренування та тестування на різних датасетах отриманих з відкритих джерел та з реального комерційного проекту. Було порівняно коефіцієнти отриманих даних в залежності від різних вхідних умов.

Дослідження проводились на основі отриманих даних, проте результат який був отриманий можна використати при написанні правил для виявлення ботнету в реальному часі. Саме знання про вагу кожного з вхідних параметрів, або їх комбінацій, дозволяють створювати тести та правила, що мають збалансованість між точністю та швидкістю.

ВИСНОВКИ

Через велику поширеність засобів, слабко захищених від різного роду вразливостей, питання про підвищену небезпеку з боку бот-мереж постає особливо гостро. Одним з можливих шляхів вирішення може стати повсюдне впровадження систем безперервного аналізу трафіку на наявність аномалій, будь яких відхилень від норми, не стандартної поведінки.

Ці засоби не дають абсолютної гарантії, але в комбінації, наприклад з машинним навчанням, можуть дуже ефективно протидіяти будь яким загрозам. SIEM-системи були створені для таких робіт, їх основною задачею є виявлення, своєчасне попередження та можливе знешкодження протиправної діяльності.

Еволюція ботнетів тягне за собою активний розвиток засобів здатних ним протидіяти, в тому числі і SIEM-систем, що можна побачити на прикладі використання Elasticsearch. Активні дослідження комбінації SIEM та машинного навчання дозволить створити якомога кращий варіант для виявлення сучасних ботнетів. Вибираючи з набору розкиданих вхідних даних лише найважливіші, можна добитись найкращих результатів по ефективності та часу роботи. Це означає, що інтеграція SIEM-систем на прикладі Elasticsearch для виявлення ботнету є доцільною та має велику практичну цінність.

Засоби машинного навчання дозволяють з високою ймовірністю виявляти ботнети, проте дуже часто це супроводжується великими часовими тратами. Дослідження в цій області спрямовані на знаходження компромісу між величиною моделі та часом потрібним для її ефективної роботи.

Отримані результати встановлюють залежність точності виявлення від вхідних параметрів, а це в свою чергу дозволяє впровадити кращі моделі в процес виявлення ботнету в режимі реального часу.

В майбутніх дослідженнях на цю тему потрібно дослідити роботу моделі використовуючи різні алгоритми, перевірити більшу кількість комбінацій параметрів даних, визначити залежність точності моделі від часу роботи та від величини пакету моделі та пакету для тестування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Валентин Холмогоров. "Pro Вирусы". [Електронний ресурс] – Режим доступу до ресурсу: https://wm-help.net/lib/b/book/1202642470/5

2. "Ботнет. Как и зачем создаются ботнеты. Средства защиты от ботнетов.". [Електронний ресурс] – Режим доступу до ресурсу: http://kz-cert.kz/ru/botnets

3. Алексей Дрозд. "Обзор SIEM-систем на мировом и российском рынке". [Електронний ресурс] – Режим доступу до ресурсу: https://www.anti-malware.ru/analytics/Technology_Analysis/Overview_SECURITY_systems_global_and_Russian_market

4. "СРАВНЕНИЕ SIEM-СИСТЕМ". WIRED. [Електронний ресурс] – Режим доступу до ресурсу: https://searchinform.ru/products/siem/sravnenie-siem-sistem/

5. "Управление событиями информационной безопасности (SIEM)". [Електронний ресурс] – Режим доступу до ресурсу: http://www.in4sec.com.ua/upravlenie-soby-tiyami-informatsionnoj-bezopasnosti-siem/

6. Борисов В. И. "О ПРИМЕНЕНИИ СИГНАТУРНЫХ МЕТОДОВ АНАЛИЗА ИНФОРМАЦИИ В SIEM-СИСТЕМАХ". [Електронний ресурс] – Режим доступу до ресурсу: http://docplayer.ru/41066049-O-primenenii-signaturnyh-met...v-siem-sistemah.html

7. "Botnet detection". WIRED. Retrieved 2017-05-24. [Електронний ресурс] – Режим доступу до ресурсу: http://jpdias.me/botnet-lab//countermeasures/detection.html

8. "Host-based intrusion detection systems". [Електронний ресурс] – Режим доступу до ресурсу: https://gradesfixer.com/free-essay-examples/host-based-intrusion-detection-systems/

9. Ritvik Khanna. "How to use Elasticsearch, Logstash and Kibana to visualise logs in Python in realtime". [Електронний ресурс] – Режим доступу до ресурсу: https://www.freecodecamp.org/news/how-to-use-elasticsearch-logstash-and-kibana-to-visualise-logs-in-python-in-realtime-acaab281c9de/

10. KARTHIK "Tutorial : Visualize historical data with ELK stack". [Електронний ресурс] – Режим доступу до ресурсу: https://www.upnxtblog.com/index.php/2018/08/09/tutorial-visualize-historical-data-with-elk-stack/

11. Daniel Berman. "Network Security Monitoring with Suricata, Logz.io and the ELK Stack". [Електронний ресурс] – Режим доступу до ресурсу: https://logz.io/blog/network-security-monitoring/