ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

Пояснювальна записка

до бакалаврської роботи
на тему:

«ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА ВТОРГНЕННЯ В СИСТЕМУ НА БАЗУ SURICATA»

Виконав студент 4 курсу, групи БСД-43
спеціальності 125 Кібербезпека
освітньо-професійної програми «Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)
Компанець О.С.
(прізвище та ініціали)
Керівник Бойко А.О.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)
Нормоконтролер
(прізвище та ініціали)

КИЇВ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Інститут ННІЗІ
Кафедра Інформаційної та кібернетичної безпеки
Ступінь вищої освіти Бакалавр
Спеціальність 125 Кібербезпека
Освітня програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКБ
_____ Гайдур Г.І.
“___”_____ 2023 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Компанцю Олександру Сергійовичу
(прізвище, ім’я, по батькові)

1. Тема бакалаврської роботи: «Дослідження методів та засобів виявлення та реагування на вторгнення в систему на базі Suricata»
керівник бакалаврської роботи Бойко Анна Олександрівна, асистент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «___ » _____2023 року № ____.

2. Строк подання студентом бакалаврської роботи 28.05.2023 р.

3. Вихідні дані до бакалаврської роботи
Системи виявлення та реагування на вторгнення;
Система Suricata;
наукова та технічна література, експлуатаційна документація, нормативні документи, міжнародні стандарти.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Актуальність проблеми виявлення та реагування на вторгнення в систему.
2. Актуальні методи захисту.
3. Методи та засоби виявлення та реагування на вторгнення.
4. Рекомендації щодо використання методів Suricata.

5. Перелік графічного матеріалу
1. Тема бакалаврської роботи.
2. Об’єкт, предмет, мета та наукові завдання дослідження.
3. Результати аналізу систем виявлення та захисту.
4. Результати аналізу методів та засобів виявлення та реагування на вторгення.
5. Рекомендації щодо виявлення та реагування в систему на базі Suricata.
7. Висновки за результатами роботи.

6. Дата видачі завдання 15.02.2023р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів бакалаврської роботи	Строк виконання етапів бакалаврської роботи	Примітка
1.	Підбір науково-технічної літератури.	15.02.2023 р.	
2.	Аналіз наукової та технічної літератури з питань теми бакалаврської роботи.	01.03.2023 р.	
3.	Аналіз методів та засобів виявлення та реагування на вторгення в систему.	21.03.2023 р.	
4.	Розроблення методики виявлення та реагування на вторгення з використанням Suricata System.	11.04.2023 р.	
5.	Оформлення результатів дослідження.	12.05.2023 р.	
6.	Підготовка доповіді до захисту.	28.05.2023 р.	

Студент Компанець О.С.
(підпис) прізвище та ініціали

Керівник бакалаврської роботи Бойко А.О.
(підпис) прізвище та ініціали

ВІДГУК РЕЦЕНЗЕНТА

на бакалаврську роботу

студента Компанця Олександра Сергійовича
на тему: « Дослідження методів та засобів виявлення та реагування на вторгнення в систему на базі Suricata»

Актуальність: Зростаюча кількість кібератак та загроз безпеці даних вимагають постійного вдосконалення методів захисту та виявлення вторгнень. У контексті постійно зростаючої складності кібератак, дослідження методів та засобів виявлення та реагування на вторгнення стає критично важливим завданням. Використання Suricata може допомогти виявити вторгнення на ранніх стадіях, що дозволяє ефективно реагувати на них та забезпечувати безпеку системи. Усі ці аспекти дослідження мають практичне значення для підвищення безпеки інформаційних систем. Враховуючи постійний розвиток загроз та атак, ця тема залишається актуальною і вимагає подальших досліджень і вдосконалення методологій та технологій виявлення та реагування на вторгнення.

Позитивні сторони:

1. Робота дозволяє досить повно оцінити загальний стан, характеристику, сутність та структуру поставленої проблеми.
2. Досліджено важливе питання розробки методики виявлення та реагування на вторгення з використанням Suricata Systems.
3. Текст викладено достатньо грамотно, послідовно.

Недоліки:

1. У бакалаврській роботі не приведено детальний опис розробки методики.
2. Слід було провести аналіз додаткової технічної літератури по темі бакалаврської роботи.

Висновок: Враховуючи недоліки, бакалаврська робота заслуговує оцінку **добре**, а студент **Компанець О.С.** – присвоєння кваліфікації: бакалавр з кібербезпеки за спеціалізацією Інформаційна та кібернетична безпека.

Якість роботи	
Виконано на замовлення підприємства	
Виконано за тематикою НДР	
Виконано з макетом	
Виконано з застосуванням ЕОМ та МПТ	√
Має практичну цінність	√
Проект-частина комплексної теми	

Підпис рецензента (_____)

Підпис засвідчую
Підпис особи, що засвідчує (_____)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

ПОДАННЯ
ГОЛОВІ ДЕРЖАВНОЇ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ БАКАЛАВРСЬКОЇ РОБОТИ

Направляється студент Компанець О.С до захисту бакалаврської роботи
(прізвище та ініціали)
спеціальності 125 Кібербезпека
освітньо-професійної програми Інформаційна та кібернетична безпека
(шифр і назва спеціальності)
на тему: «Дослідження методів та засобів виявлення та реагування на вторгнення в систему на базі Suricata».
Бакалаврська робота і рецензія додаються.

Директор інституту _____ Савченко В.А.
(підпис) (прізвище та ініціали)

Довідка про успішність

Компанець О.С. за період навчання в інституті
(прізвище та ініціали студента)
ННІЗІ з 2019 року по 2023 рік повністю виконав навчальний план за напрямом підготовки, спеціальністю з таким розподілом оцінок за:
національною шкалою: відмінно ____%, добре ____%, задовільно ____%;
шкалою ECTS: A ____%; B ____%; C ____%; D ____%; E ____%.

Секретар інституту, факультету (відділення) _____ Берестяна Т.В.
(підпис) (прізвище та ініціали)

Висновок керівника бакалаврської роботи

Студент ***Компанець О.С.*** обрав тему роботи, метою якої було дослідження методів та засобів виявлення та реагування на вторгнення в систему на базі Suricata. Перелік використаних джерел свідчить про вміння студентом розбиратись в наукових питаннях та застосовувати їх при дослідженнях. Під час виконання бакалаврської роботи Компанець О.С. показав добру теоретичну та практичну підготовку. Роботу виконував сумлінно, акуратно та вчасно за планом.
Все це дозволяє оцінити виконану бакалаврську роботу студента Компанця Олександра Сергійовича на оцінку **«добре»** та присвоїти йому кваліфікацію: бакалавр з кібербезпеки за спеціалізацією Інформаційна та кібернетична безпека.

Керівник бакалаврської роботи _____ Бойко А.О.
(підпис) (прізвище та ініціали)
“____” _____ 2023 року

Висновок кафедри про бакалаврську роботу

Бакалаврська робота розглянута. Студент Компанець О.С.
(прізвище та ініціали)
Завідувач кафедри Інформаційної та кібернетичної безпеки
(назва)

_____ Гайдур Г.І.
(підпис) (прізвище та ініціали)

“____” _____ 2023 року

РЕФЕРАТ

Текстова частина бакалаврської роботи: 46 сторінок, 10 джерел.

Об'єкт дослідження – Процес виявлення та запобігання вторгнень.

Предмет дослідження – Методи та засоби виявлення та реагування на вторгнення в систему.

Мета роботи – Огляд системи Suricata та її можливостей, а також оцінка її ефективності як інструменту безпеки.

Методи дослідження – Розробити рекомендаціїї щодо розробки методики виявлення та реагування на вторгнення в систему на базі Suricata. Опрацювання літератури за даною темою, аналіз експлуатаційної документації, міжнародних стандартів та їх порівняння, проведення тестів.

Для проведення цього дослідження будуть використані різні методи, включаючи огляд літератури про існуючі дослідження та документацію щодо системи Suricata. Це передбачає збір інформації з академічних журналів, офіційних документів та інших відповідних джерел, щоб отримати розуміння особливостей, можливостей та обмежень системи.
Загалом, це дослідження надасть цінну інформацію про систему Суріката та її ефективність як інструменту безпеки. Поєднуючи теоретичні та практичні методи дослідження, цей нарис запропонує всебічний огляд системи та її потенціалу для використання в операціях з мережевої безпеки.

БЕЗПЕКА МЕРЕЖІ, СИСТЕМИ ВИЯВЛЕННЯ, СИСТЕМИ ЗАПОБІГАННЯ ВТОРНЕНЬ, ЗАГРОЗИ, ЗАХИСТ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ОС - операційна система

IDS - Intrusion Detection System

IPS - Intrusion Prevention System

OISF - Open Information Security Foundation

IoT - Internet of Things

HIDS – Host-based intrusion detection system

ICMP - Internet Control Message Protocol

SQL - Structured Query Language

SMTP - Simple Mail Transfer Protocol

FTP – File transport protocol

SIEM - Security information and event management

SRL - Suricata Rules Language

NIDS - Network Intrusion Detection System

ВСТУП

У сучасну цифрову епоху ми покладаємося на комп'ютерні системи майже в усіх аспектах нашого життя - від особистого спілкування до критично важливої інфраструктури. Однак разом з цим зростає ризик кібератак, в тому числі системних вторгнень. Системне вторгнення - це несанкціонований доступ до комп'ютерної системи, який може бути спричинений різними факторами, такими як шкідливе програмне забезпечення, соціальна інженерія або мережеві вразливості.

Наслідки системного вторгнення можуть бути серйозними, включаючи крадіжку конфіденційної інформації, фінансові втрати і навіть порушення роботи критично важливої інфраструктури. Тому важливо мати надійні методи і засоби виявлення та реагування на системні вторгнення, щоб мінімізувати ризик таких атак.

Виявлення та реагування на системні вторгнення є складним завданням, яке вимагає спеціалізованих знань та інструментів. Одним з найпопулярніших інструментів, що використовується в цьому контексті, є система виявлення вторгнень (IDS)[9]. IDS - це програмне забезпечення, яке відстежує мережевий трафік для виявлення підозрілої активності, що може свідчити про вторгнення в систему. Після виявлення, IDS може попередити системного адміністратора або вжити автоматичних заходів для блокування або пом'якшення атаки.

Хоча на ринку є кілька комерційних рішень IDS, багато організацій обирають рішення з відкритим вихідним кодом через їхню гнучкість та економічну ефективність. Одним з таких рішень з відкритим кодом, яке набуло популярності в останні роки, є Suricata.

Suricata - це IDS та система запобігання вторгненням IPS з відкритим вихідним кодом, яка використовується для моніторингу мережевого трафіку та виявлення потенційних загроз. Спочатку вона була розроблена OISF, а зараз підтримується спільнотою розробників з усього світу.

Suricata використовує кілька методів виявлення, включаючи виявлення на основі сигнатур, виявлення на основі аномалій і виявлення на основі протоколів. Виявлення на основі сигнатур передбачає порівняння мережевого трафіку з базою даних відомих сигнатур загроз. Виявлення на основі аномалій передбачає виявлення трафіку, який відхиляється від нормального, що може свідчити про нову або невідому загрозу. Виявлення на основі протоколів передбачає аналіз трафіку на рівні протоколів для виявлення підозрілої активності.

На додаток до можливостей виявлення, Suricata також надає ряд механізмів реагування, включаючи блокування або оповіщення про підозрілий трафік. Ефективність цих механізмів реагування залежить від їх конфігурації та інтеграції з іншими інструментами безпеки.

Враховуючи критичну роль IDS у виявленні та реагуванні на системні вторгнення, важливо постійно досліджувати і розробляти нові методи і засоби для підвищення їх точності та ефективності. Дослідники зосереджені на різних аспектах досліджень IDS, включаючи розробку нових методів виявлення, оптимізацію існуючих, вдосконалення механізмів реагування та зменшення кількості помилкових спрацьовувань і хибних спрацьовувань.

Одним з напрямків досліджень є використання алгоритмів машинного навчання для підвищення точності виявлення аномалій. Алгоритми машинного навчання можуть аналізувати великі обсяги даних мережевого трафіку і виявляти закономірності, які можуть свідчити про вторгнення в систему. Дослідники також вивчають використання апаратного прискорення і розподіленої обробки для підвищення продуктивності рішень IDS, таких як Suricata.

Ще одним напрямком досліджень є оптимізація конфігурації IDS для мінімізації помилкових спрацьовувань та хибних спрацьовувань. Хибні спрацьовування стосуються ситуацій, коли звичайний трафік позначається як підозрілий, тоді як хибні негативні спрацьовування стосуються ситуацій, коли реальні загрози не виявляються. Як хибні спрацьовування, так і хибні

негативні результати можуть суттєво вплинути на ефективність IDS і потребують ретельного контролю.

Отже, дослідження методів і засобів виявлення та реагування на системні вторгнення є критично важливими для підтримки безпеки комп'ютерних систем. Суріката стала потужним інструментом виявлення та реагування на системні вторгнення, і дослідники зосередилися на розробці нових методів виявлення, оптимізації існуючих, вдосконаленні механізмів реагування та зменшенні помилкових спрацьовувань і хибних спрацьовувань. Оскільки загроза системних вторгнень продовжує зростати, дослідження рішень IDS, таких як Suricata, залишатимуться критично важливою сферою досліджень.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ЗАХИСТУ

1.1 Історія та сучасний стан інформаційної безпеки

Інформаційна безпека є критично важливою сферою, яка постійно розвивається і зазнала значних змін за останні кілька десятиліть. Історія інформаційної безпеки бере свій початок з перших днів розвитку комп'ютерів, коли заходи безпеки були зосереджені в першу чергу на фізичній безпеці. У ті часи комп'ютерні кімнати замикали, а обладнання захищали, щоб запобігти несанкціонованому доступу. Однак, у міру того, як комп'ютерні мережі зростали в розмірах і ускладнювалися, загрози безпеці ставали все більш витонченими, і інформаційна безпека повинна була розвиватися, щоб протистояти цим загрозам.

У 1970-х роках з'явилася концепція контролю доступу, що дозволила системним адміністраторам обмежувати доступ до комп'ютерних систем і конфіденційних даних. У 1980-х роках з'явилися технології шифрування, які дозволили безпечно передавати дані мережами. Ці технології мали важливе значення для забезпечення захисту конфіденційних даних від несанкціонованого доступу або підробки.

У 1990-х роках поява Інтернету та широке використання комп'ютерів у бізнесі та повсякденному житті принесли нові загрози безпеці, зокрема комп'ютерні віруси, шкідливе програмне забезпечення та мережеві атаки. Це призвело до необхідності більш надійних заходів інформаційної безпеки, і організації почали інвестувати в технології та протоколи безпеки для захисту своїх мереж і даних [10]. У цей час брандмауери та антивірусне програмне забезпечення стали популярними, оскільки вони могли виявляти і запобігати проникненню в мережу шкідливих програм та іншого шкідливого програмного забезпечення.

На початку 2000-х років з'явилася концепція кібербезпеки, яка зосередилася на захисті комп'ютерних мереж, систем і даних від несанкціонованого доступу, використання, розкриття, порушення, модифікації або знищення. Кібербезпека

охоплює широкий спектр заходів, включаючи управління ризиками, аналіз загроз, оцінку вразливості, реагування на інциденти та навчання з питань безпеки. Зі зростанням складності кібератак стала очевидною потреба у фахівцях з кібербезпеки.

Сучасний стан інформаційної безпеки - це стан постійної еволюції та адаптації. Ландшафт загроз продовжує розвиватися, і організації повинні постійно оновлювати свої заходи безпеки, щоб не відставати від новітніх загроз. Розвиток хмарних обчислень, інтернету речей (IoT) та інших новітніх технологій створив нові виклики для безпеки, а також нові можливості для інновацій у сфері безпеки.

Сьогодні фахівці з безпеки використовують широкий спектр технологій і методологій для захисту конфіденційних даних і мереж. Системи виявлення вторгнень (IDS), системи запобігання вторгненням (IPS), шифрування даних і безпечні методи кодування - це лише деякі з багатьох методів, що використовуються для захисту мереж і даних від несанкціонованого доступу. Фахівці з безпеки також покладаються на найкращі практики, такі як контроль доступу, сегментація мережі та тренінги з підвищення обізнаності щодо безпеки, щоб мінімізувати ризик порушень безпеки.

На додаток до цих технічних заходів, інформаційна безпека також значною мірою спирається на політику та системи управління, такі як ISO 27001 та NIST Cybersecurity Framework. Ці документи містять рекомендації щодо управління та захисту конфіденційних даних і гарантують, що організації дотримуються відповідних законів і нормативних актів.

Загалом, історія інформаційної безпеки - це історія постійного розвитку та адаптації до нових загроз і технологій. Сучасний стан інформаційної безпеки характеризується складністю та витонченістю, з широким спектром технологій, методологій та фреймворків, що використовуються для захисту конфіденційних даних та мереж. Оскільки ландшафт загроз продовжує розвиватися, потреба в ефективних заходах інформаційної безпеки буде тільки зростати. Завдання організацій полягає в тому, щоб залишатися пильними і адаптувати свої заходи безпеки, щоб випереджати новітні загрози.

1.2 Типи загроз інформаційній безпеці

Для захисту від широкого спектру загроз інформаційній безпеці важливе значення має не лише організаційний та процедурний контроль, а й навчання та інформування користувачів. Організаціям також важливо регулярно оцінювати свій стан безпеки та виявляти вразливі місця, які можуть бути використані зловмисниками.

Ландшафт загроз інформаційній безпеці постійно розвивається, і постійно з'являються нові загрози. З розвитком технологій неминуче виявлятимуться нові вразливості та вектори атак. Тому для організацій вкрай важливо бути в курсі останніх загроз і заходів безпеки, щоб забезпечити захист своїх конфіденційних даних і систем.

Крім того, важливо відзначити, що загрози інформаційній безпеці не обмежуються зовнішніми зловмисниками. Інсайдери, такі як співробітники або підрядники, також можуть становити значний ризик для організацій. Тому організаціям дуже важливо впроваджувати надійний контроль доступу та заходи моніторингу для запобігання та виявлення внутрішніх загроз.

Загалом, типи загроз інформаційній безпеці є різноманітними та складними. Для ефективного захисту від них організації повинні застосовувати багатогранний підхід, що включає технічні, політичні та процедурні засоби контролю, а також навчання та підвищення обізнаності користувачів. Залишаючись в курсі останніх загроз і заходів безпеки, організації можуть забезпечити захист своїх конфіденційних даних і систем перед обличчям постійно мінливого ландшафту загроз.

Рис. 1.1. Основні загрози інформаційній безпеці

Шкідливе програмне забезпечення

Шкідливе програмне забезпечення - це тип програмного забезпечення, призначеного для завдання шкоди комп'ютерним системам, мережам або пристроям (рисунок 1.1.). До шкідливих програм належать, зокрема, віруси, хробаки, троянські програми та програми-вимагачі. Шкідливе програмне забезпечення може поширюватися через вкладення електронної пошти, шкідливі веб-сайти або заражене програмне забезпечення. Після встановлення в системі шкідливе програмне забезпечення може викрасти конфіденційні дані, пошкодити файли або взяти систему під свій контроль. Атаки шкідливих програм можуть завдати значної шкоди організаціям, призводячи до втрати даних, фінансових втрат і простою системи.

Рис. 1.2. Фішинг

Фішинг

Фішинг - це різновид соціальної інженерії, який використовує підроблені електронні листи або веб-сайти, щоб обманом змусити користувачів надати конфіденційну інформацію, таку як імена користувачів, паролі або номери кредитних карток (рисунок 1.2.). Фішингові атаки можуть бути дуже ефективними, оскільки вони часто виглядають як такі, що надходять із законних джерел, і користувачі можуть не усвідомлювати ризиків. Отримавши цю інформацію, зловмисники можуть використовувати її для крадіжки особистих даних, шахрайства або отримання доступу до конфіденційних систем чи даних.

Атаки на відмову в обслуговуванні (DoS)

Атаки на відмову в обслуговуванні (DoS-атаки) призначені для перевантаження системи або мережі трафіком, що робить її недоступною для користувачів. DoS-атаки можуть здійснюватися за допомогою бот-мереж - мереж інфікованих пристроїв, якими можна керувати віддалено. DoS-атаки можуть спричинити значні перебої в роботі організації, що призводить до втрати продуктивності та доходів.

DDoS-атака (розподілена відмова в обслуговуванні)

DDoS-атака залучає кілька комп'ютерів, часто розподілених по різних місцях або мережах, які спільно використовуються для запуску атаки. Ці комп'ютери, як

правило, скомпрометовані шкідливим програмним забезпеченням або контролюються зловмисником за допомогою ботнету - мережі скомпрометованих пристроїв, що перебувають під його управлінням. Зловмисник координує ці комп'ютери для одночасного надсилання величезного обсягу трафіку або запитів, щоб перевантажити цільову систему або мережу.

DDoS-атаки складніше протидіяти, оскільки вони передбачають використання розподіленої інфраструктури, що ускладнює ідентифікацію та ефективне блокування всіх джерел атак. Зловмисник може змінювати вектори атаки, використовувати підміну IP-адреси, щоб приховати справжнє джерело, і динамічно змінювати шаблони атаки, що ускладнює розрізнення легітимного трафіку від зловмисного.

Таким чином, хоча і DoS, і DDoS-атаки спрямовані на порушення доступності комп'ютерних систем або мереж, DoS-атака передбачає наявність одного джерела атаки, тоді як DDoS-атака передбачає наявність декількох розподілених джерел, які працюють разом, щоб перевантажити ціль. DDoS-атаки, як правило, складніші, їх важче нейтралізувати, і вони здатні завдати більш значної шкоди через свою розподілену природу (рисунок 1.3).

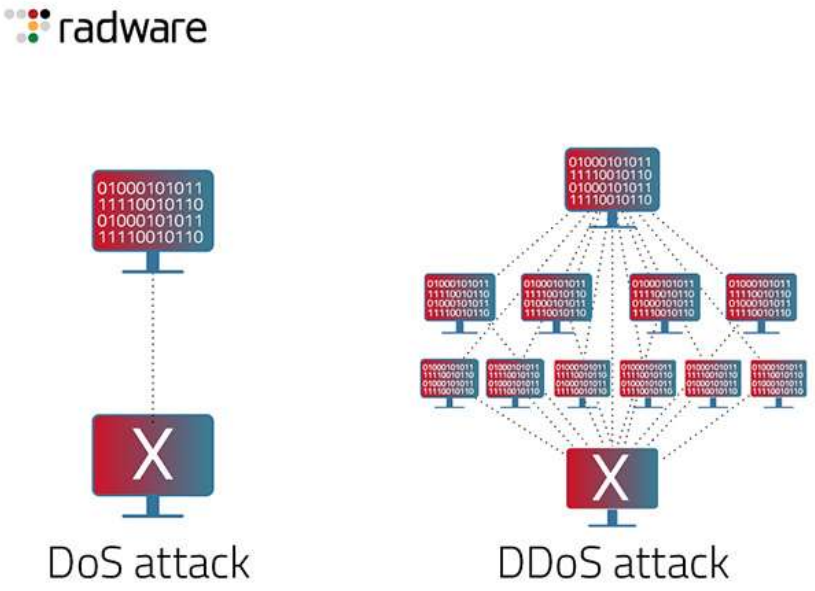

Рис. 1.3. Різниця між Dos та DDos атаками

Внутрішні загрози

Внутрішні загрози виникають, коли працівники, підрядники або інші інсайдери використовують свій доступ до конфіденційних систем або даних у зловмисних цілях. Внутрішні загрози можуть бути навмисними або випадковими, і вони можуть завдати значної шкоди організаціям. Інсайдери можуть викрасти конфіденційні дані, пошкодити системи або порушити роботу. Внутрішні загрози може бути важко виявити, оскільки інсайдери можуть мати законний доступ до конфіденційних систем або даних.

Складні постійні загрози (APT)

Сучасні постійні загрози (APT) - це довготривалі цілеспрямовані атаки, спрямовані на отримання доступу до конфіденційних даних або систем. Зазвичай APT здійснюються висококваліфікованими зловмисниками, які використовують складні методи, щоб уникнути виявлення (рисунок 1.4). APT-атаки важко виявити та нейтралізувати, і вони можуть завдати значної шкоди організаціям, включаючи крадіжку інтелектуальної власності або конфіденційних даних.

Рис. 1.4. Сучасні постійні загрози

Фізичні загрози

Фізичні загрози інформаційній безпеці включають крадіжку або пошкодження обладнання, такого як сервери, комп'ютери або мобільні пристрої. Фізичні загрози також можуть включати несанкціонований доступ до фізичних місць, де зберігаються конфіденційні дані. Фізичні загрози може бути важко запобігти, оскільки вони вимагають фізичних заходів безпеки, таких як замки, сигналізація та камери спостереження.

Експлойти нульового дня

Експлойти нульового дня - це вразливості в програмному або апаратному забезпеченні, про які не знає виробник або розробник. Зловмисники можуть використовувати ці вразливості, щоб отримати доступ до конфіденційних систем або даних. Експлойти нульового дня буває важко виявити та усунути, оскільки може не існувати патчів або оновлень для усунення вразливості.

Атаки на веб-додатки

Атаки на веб-додатки націлені на вразливості у веб-додатках, такі як SQL-ін'єкції або міжсайтовий скриптинг. Ці вразливості можуть бути використані для отримання доступу до конфіденційних даних або систем. Атаки на веб-додатки можуть бути дуже ефективними, оскільки вони можуть бути автоматизовані, а зловмисники можуть атакувати декілька систем одночасно.

1.3 Системи виявлення вторгнень

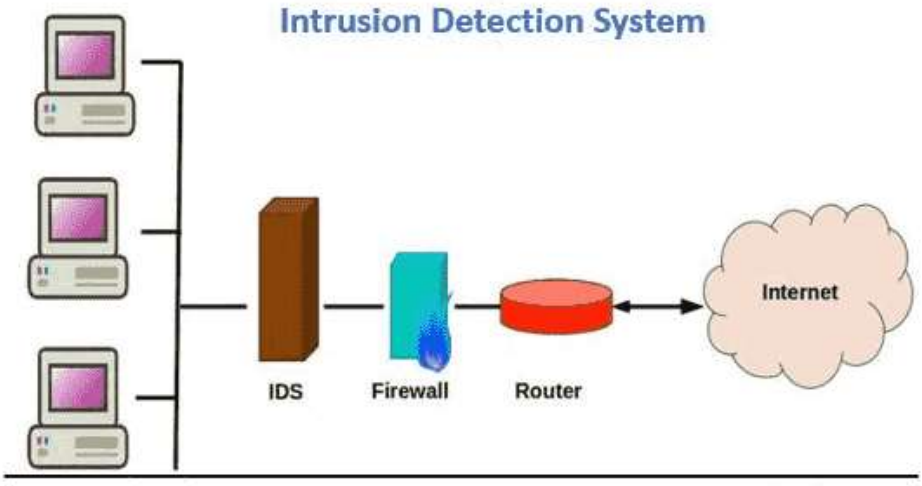

Рис. 1.5. Ілюстрація знаходження системи виявлення

Системи виявлення вторгнень (IDS) є важливим компонентом інформаційної безпеки, призначеним для виявлення та оповіщення про підозрілу активність в мережі або системі (рисунок 1.5). Основною функцією IDS є моніторинг мережевого трафіку та системних журналів для виявлення ознак потенційних вторгнень або атак. IDS можна розділити на два типи: мережеві системи виявлення вторгнень (NIDS) і системи виявлення вторгнень на основі хостів (HIDS).

NIDS призначені для моніторингу мережевого трафіку в режимі реального часу в пошуках ознак потенційних загроз, таких як шкідливе програмне забезпечення, спроби несанкціонованого доступу або підозрілі шаблони активності. NIDS можуть бути розгорнуті в різних точках мережі, в тому числі на периметрі мережі, в окремих сегментах мережі або на критично важливих серверах. Аналізуючи мережевий трафік і порівнюючи його з відомими сигнатурами атак або поведінковими аномаліями, NIDS може попереджати команди безпеки про потенційні загрози, дозволяючи своєчасно реагувати і пом'якшувати їх.

HIDS, з іншого боку, призначені для моніторингу активності в окремих системах або хостах, шукаючи ознаки потенційних загроз, таких як спроби несанкціонованого доступу, зміни в критично важливих системних файлах або незвичайні шаблони активності. HIDS зазвичай працюють як програмні агенти на окремих системах, відстежуючи системні журнали та іншу системну активність. Порівнюючи активність системи з відомими сигнатурами атак або поведінковими аномаліями, HIDS може сповіщати команди безпеки про потенційні загрози, що дозволяє вчасно реагувати та пом'якшувати їх наслідки.

На додаток до традиційних IDS, існують також новіші системи, які включають машинне навчання і штучний інтелект для підвищення точності виявлення і зменшення кількості помилкових спрацьовувань. Ці системи, відомі як системи виявлення вторгнень на основі поведінки, аналізують моделі поведінки користувачів і систем для виявлення потенційно зловмисних дій, які можуть бути не виявлені традиційними системами виявлення на основі сигнатур.

Хоча IDS можуть бути ефективними у виявленні потенційних загроз, вони не є безвідмовними і можуть давати помилкові спрацьовування і помилкові негативні

результати. Хибні спрацьовування виникають, коли IDS сповіщає про легітимну активність, тоді як хибні негативні спрацьовування виникають, коли IDS не може виявити фактичну зловмисну активність. Щоб зменшити ризик хибних спрацьовувань і помилкових відмов, IDS слід ретельно налаштовувати і регулярно контролювати точність роботи.

Отже, системи виявлення вторгнень є критично важливим компонентом інформаційної безпеки, призначеним для виявлення та оповіщення про підозрілу активність у мережі або системі. Відстежуючи мережевий трафік і системну активність на наявність ознак потенційних загроз, IDS може попередити команди безпеки про потенційні атаки, що дозволяє вчасно відреагувати на них і пом'якшити їх наслідки. Однак, щоб забезпечити ефективність IDS, їх необхідно ретельно налаштовувати і контролювати, щоб зменшити ризик хибних спрацьовувань і помилкових відмов.

1.4 Системи захисту від вторгнень

Системи захисту від вторгнень (IPS) є ключовим компонентом інформаційної безпеки, призначеним для запобігання та пом'якшення наслідків вторгнень і атак в мережі або системі. У той час як системи виявлення вторгнень (IDS) зосереджені на виявленні підозрілої активності, IPS - на запобіганні заподіянню шкоди від цієї активності.

IPS працюють шляхом моніторингу мережевого трафіку і системної активності на предмет відомих сигнатур атак або поведінкових аномалій, подібно до IDS. Однак, замість того, щоб просто сповіщати про потенційні загрози, IPS відіграють активну роль у запобіганні цим загрозам від заподіяння шкоди. Це може включати блокування або карантин мережевого трафіку, припинення шкідливих процесів або блокування спроб несанкціонованого доступу.

IPS можуть бути розгорнуті в різних точках мережі, в тому числі на периметрі мережі, в окремих сегментах мережі або на критично важливих серверах (рисунок 1.6). Відстежуючи та блокуючи підозрілу активність, IPS може допомогти

запобігти широкому спектру атак, включаючи зараження шкідливим програмним забезпеченням, атаки на відмову в обслуговуванні та спроби несанкціонованого доступу.

Як і IDS, IPS можуть також включати в себе машинне навчання і штучний інтелект для підвищення точності виявлення і зменшення кількості помилкових спрацьовувань. Ці системи, відомі як системи запобігання вторгненням на основі поведінки, аналізують моделі поведінки користувачів і систем, щоб виявити потенційно шкідливу активність, яка може бути не виявлена традиційними системами виявлення на основі сигнатур.

Хоча IPS можуть бути ефективними у запобіганні потенційним загрозам, вони не є безвідмовними і можуть давати помилкові спрацьовування та помилкові негативні результати. Хибні спрацьовування виникають тоді, коли IPS блокує легітимну діяльність, а хибні негативні спрацьовування виникають тоді, коли IPS не може виявити фактичну зловмисну активність. Щоб зменшити ризик хибних спрацьовувань і помилкових відмов, IPS слід ретельно конфігурувати і регулярно перевіряти точність роботи.

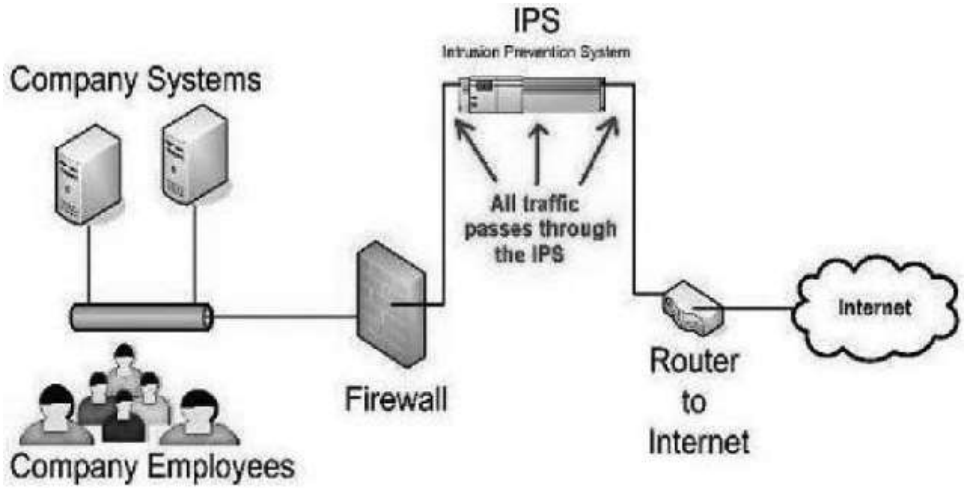

Рис. 1.6. Ілюстрація позиції системи захисту від вторгнень

Отже, системи захисту від вторгнень є критично важливим компонентом інформаційної безпеки, призначеним для запобігання та пом'якшення наслідків вторгнень і атак в мережі або системі. Відстежуючи мережевий трафік і системну активність на наявність відомих сигнатур атак або поведінкових аномалій, IPS

може блокувати або поміщати потенційні загрози в карантин, не даючи їм завдати шкоди. Однак, щоб забезпечити ефективність IPS, їх необхідно ретельно налаштовувати і контролювати, щоб зменшити ризик помилкових спрацьовувань і хибних негативних результатів.

2 ОГЛЯД SURICATA SYSTEM

2.1 Архітектура та особливості роботи Suricata systems

Архітектура Suricata розроблена для ефективної обробки та аналізу мережевого трафіку при збереженні високого рівня продуктивності. В основі Suricata лежить багатопотокова архітектура, що дозволяє паралельно обробляти мережеві пакети на декількох процесорних ядрах. Таке розпаралелювання підвищує здатність системи справлятися з високими мережевими навантаженнями і забезпечує швидке виявлення та реагування на потенційні загрози. Архітектура складається з декількох ключових компонентів, включаючи движок перехоплення, движок виявлення та движок виводу. Ці компоненти працюють у тандемі, забезпечуючи комплексний аналіз мережевого трафіку та виявлення загроз.

Рушій перехоплення

Движок перехоплення відповідає за отримання мережевого трафіку для аналізу. Suricata підтримує декілька методів перехоплення, таких як libpcap, AF_PACKET і Netmap, що дозволяє гнучко перехоплювати трафік з різних мережевих інтерфейсів. Рушій перехоплення обробляє вхідні пакети і пересилає їх до рушія виявлення для аналізу.

Механізм виявлення

Механізм виявлення - це серце Suricata, що відповідає за перевірку мережевих пакетів та виявлення потенційних загроз. Він підтримує широкий спектр методів виявлення, включаючи виявлення на основі сигнатур, аналіз протоколів і виявлення аномалій. Suricata використовує мову правил Suricata Rules Language (SRL) для визначення правил виявлення та сигнатур. Ці правила можуть бути налаштовані відповідно до конкретних організаційних вимог, що робить Suricata дуже адаптованою до різних мережевих середовищ.

Механізм виводу

Механізм виводу відповідає за генерацію сповіщень та реєстрацію

мережевих подій, виявлених механізмом виявлення. Suricata надає різні варіанти виводу, включаючи консольні сповіщення, файли журналів та інтеграцію з системами управління інформацією та подіями безпеки (SIEM). Ця гнучка архітектура виводу полегшує моніторинг і аналіз мережевого трафіку в режимі реального часу, що дозволяє ефективно реагувати на інциденти і проводити криміналістичні розслідування.

Ключові особливості систем Suricata

Системи Suricata пропонують комплексний набір функцій, які сприяють їх ефективності у виявленні та реагуванні на системні вторгнення. Розуміння цих функцій має важливе значення для повного використання потенціалу Suricata в захисті мереж. У цьому розділі розглядаються деякі з ключових функцій, які надає Суріката.

Виявлення на основі сигнатур

Можливості виявлення на основі сигнатур дозволяють Suricata ідентифікувати відомі загрози на основі заздалегідь визначених шаблонів. Suricata підтримує набори правил Emerging Threats і Snort [1], надаючи велику колекцію попередньо налаштованих правил для виявлення поширених шаблонів атак. Крім того, організації можуть розробляти власні правила для виявлення специфічних загроз, характерних для їхнього середовища.

Аналіз протоколів

На додаток до виявлення на основі сигнатур, Suricata використовує методи аналізу протоколів для виявлення аномалій і підозрілої поведінки. Перевіряючи мережеві протоколи на різних рівнях, Suricata може виявити відхилення від очікуваної поведінки, наприклад, незвичайні структури пакетів або недійсні протокольні послідовності. Ця можливість особливо цінна для виявлення атак нульового дня або раніше невідомих вразливостей.

Поведінковий аналіз

Можливості поведінкового аналізу Suricata дозволяють виявляти зловмисні дії на основі шаблонів і аномалій, що спостерігаються в мережевому трафіку. Встановивши базову поведінку мережі, Suricata може виявити відхилення, які

можуть свідчити про зловмисні наміри. Такий підхід дозволяє виявляти складні атаки, які не піддаються традиційним методам виявлення на основі сигнатур.

Вилучення та аналіз файлів

Suricata може витягувати файли з мережевого трафіку і проводити їх глибокий аналіз. Ця функція особливо корисна для виявлення шкідливих файлів, таких як шкідливі програми або експлойти, які можуть передаватися через мережу. Suricata підтримує різні методи вилучення файлів, включаючи вилучення файлів з протоколів HTTP, FTP і SMTP.

Варіанти розгортання

Suricata може бути розгорнута в різних мережевих архітектурах і конфігураціях відповідно до конкретних вимог організації. У цьому розділі розглядаються різні варіанти розгортання, доступні для систем Суріката.

Автономний режим

В автономному режимі Suricata працює як індивідуальна система виявлення та запобігання вторгнень. Вона може бути розгорнута на виділеному сервері або мережевому пристрої, що дозволяє їй аналізувати мережевий трафік і генерувати оповіщення незалежно. Автономне розгортання ідеально підходить для малих і середніх організацій з відносно простою мережевою інфраструктурою.

Інтегрований режим

Suricata також може бути розгорнута в вбудованому режимі, де вона бере активну участь в потоці мережевого трафіку. У цьому режимі Suricata може блокувати або модифікувати мережеві пакети в режимі реального часу, забезпечуючи додатковий рівень захисту від вторгнень. Розгортання в лінію підходить для організацій, які потребують активних засобів запобігання і мають більш складну мережеву архітектуру.

Розподілений режим

Для великих мережевих інфраструктур Suricata підтримує розподілене розгортання. Кілька екземплярів Suricata можуть бути розгорнуті в різних сегментах мережі, утворюючи скоординовану мережу виявлення та запобігання. Така розподілена архітектура забезпечує балансування навантаження,

масштабованість і централізоване управління всією системою Suricata.

Якщо підсумувати вищесказане, ми зрозуміємо що, архітектура та функції Suricata роблять її потужним інструментом для виявлення та реагування на вторгнення в систему. Її багатопотокова конструкція, разом з компонентами Capture Engine, Detection Engine [8] і Output Engine, забезпечує ефективний аналіз мережевого трафіку. Виявлення на основі сигнатур, аналіз протоколів, поведінковий аналіз і можливості вилучення файлів сприяють комплексному виявленню загроз. Гнучкі можливості розгортання Suricata ще більше підвищують її застосовність у широкому діапазоні мережевих архітектур.

2.2 Порівняння з іншими системами виявлення вторгнень

Таблиця 2.1.

Порівняння систем виявлення вторгнень

	Suricata	**Snort**	**Bro (Zeek)**
Методи виявлення	Сигнатурний та поведінковий аналіз	На основі сигнатур	Сигнатурний аналіз і аналіз на рівні додатків
Багатопотоковий	Так (оптимізовано для сучасного обладнання)	Ні	Так
Підтримка протоколів	IPv4 та IPv6	IPv4	IPv4 та IPv6
Налаштування	Легко налаштовується з підтримкою декількох мов	Легко налаштовується з підтримкою Lua-скриптів	Легко налаштовується з підтримкою сценаріїв
Вартість	Безкоштовний та з відкритим вихідним кодом	Безкоштовний та з відкритим вихідним кодом	Безкоштовний та з відкритим вихідним кодом

Масштабованість	Висока масштабованість для великих мереж	Підходить для невеликих мереж	Висока масштабованість для великих мереж

У таблиці 2.2 наведено порівняння деяких ключових особливостей та переваг трьох популярних систем виявлення вторгнень: Suricata, Snort і Bro (тепер називається Zeek) [1].

Таблиця 2.2.

Порівняння ключових особливостей та переваг

Suricata	**Snort**	**Bro (Zeek)**
- Ефективний при обробці великих обсягів мережевого трафіку	- Простий у використанні та налаштуванні	- Ефективний при аналізі протоколів прикладного рівня
- Підтримує виявлення на основі сигнатур і поведінковий аналіз	- Добре зарекомендували себе та широко використовуються	- Легко налаштовується з підтримкою сценаріїв
Suricata	**Snort**	**Bro (Zeek)**
- Розроблено для ефективної роботи на сучасному обладнанні	- Хороша підтримка громадських та комерційних пропозицій	- З відкритим вихідним кодом і безкоштовна у використанні

Кожна система має свої сильні та слабкі сторони, і найкращий вибір для організації буде залежати від її конкретних потреб та вимог. Однак варто зазначити, що Suricata виділяється своєю підтримкою як сигнатурного виявлення, так і поведінкового аналізу, а також ефективною багатопотоковою архітектурою. Snort, з іншого боку, є хорошим вибором для невеликих мереж і широко використовується в індустрії. Bro (Zeek) відомий своєю потужною підтримкою

сценаріїв та аналізу прикладного рівня, що робить його хорошим вибором для організацій, яким потрібно відстежувати та аналізувати певні мережеві протоколи.

В цілому, вибір правильної системи виявлення вторгнень вимагає ретельного розгляду різних факторів, включаючи розмір мережі, профіль загрози і доступні ресурси. Розуміючи особливості та переваги кожної системи, організації можуть приймати обґрунтовані рішення, які допоможуть їм захистити свої мережі та дані від потенційних загроз.

У таблиці 2.3 наведено деякі з потенційних недоліків трьох популярних систем виявлення вторгнень.

Таблиця 2.3.

Потенційні недоліки систем виявлення

Suricata	**Snort**	**Bro (Zeek)**
- Може бути складнішим у налаштуванні, ніж деякі інші рішення	- Менш ефективний при обробці великих обсягів мережевого трафіку	- Не так ефективно виявляє мережеву активність нижчого рівня
- Може вимагати більше апаратних ресурсів, ніж деякі інші рішення	- Може не підходити для великих мереж	- Менш ефективно виявляє деякі типи мережевих загроз

Хоча кожна система має свої сильні сторони, є й певні компроміси, які слід враховувати. Наприклад, Suricata може бути складнішою в налаштуванні, ніж деякі інші рішення, і може вимагати більше апаратних ресурсів для ефективної роботи. Snort не так ефективно, як Suricata, обробляє великі обсяги мережевого трафіку, що може обмежувати його масштабованість [13]. Bro (Zeek) менш ефективно виявляє деякі типи мережевих загроз і може не підходити для організацій, які потребують ретельного моніторингу мережевої активності нижчих рівнів.

Варто зазначити, що ці недоліки не обов'язково є вирішальними, і організації можуть пом'якшити їх за допомогою ретельного планування та впровадження. Наприклад, ретельно конфігуруючи та налаштовуючи свої системи, організації можуть оптимізувати продуктивність та зменшити вимоги до апаратного

забезпечення. Крім того, поєднуючи виявлення вторгнень з іншими заходами безпеки, такими як брандмауери та захист кінцевих точок, організації можуть створити комплексну систему безпеки, яка допоможе зменшити потенційні загрози.

В цілому, важливо ретельно зважити всі "за" і "проти" кожної системи виявлення вторгнень при прийнятті рішення. Розуміючи компроміси, організації можуть приймати обґрунтовані рішення, які допоможуть їм досягти своїх цілей у сфері безпеки, не виходячи за рамки бюджету та обмежених ресурсів.

2.3 Оцінка ефективності застосування Suricata system

Щоб оцінити ефективність системи Suricata, проведено серію експериментів з використанням тестового середовища, яке імітувало реальний мережевий трафік і сценарії атак. Тестовий стенд складався з декількох машин, включаючи датчик Suricata, машину зловмисника і кілька машин-мішеней. Потрібно Налаштувати Suricata на моніторинг мережевого трафіку між зловмисником та цільовими машинами, а також застосувати різні методи атак, такі як сканування портів, розподілені атаки на відмову в обслуговуванні (DDoS) та SQL-ін'єкції, щоб оцінити здатність Suricata виявляти та запобігати вторгненням.

Ефективність виявлення вторгнень

Під час наших експериментів Suricata продемонструвала вражаючу ефективність у виявленні вторгнень. Вона успішно виявила та попередила про широкий спектр атак, включаючи сканування портів, SYN-флуд, ICMP-флуд та спроби SQL-ін'єкцій. Здатність Suricata виявляти атаки значною мірою пояснюється її потужним механізмом виявлення на основі сигнатур, який використовує велику колекцію попередньо визначених правил для виявлення відомих шаблонів атак. Система також підтримує нові канали розвідки загроз, що дозволяє їй бути в курсі найновіших сигнатур атак і методів їх виявлення. Загалом, можливості виявлення вторгнень Suricata виявилися високоефективними у виявленні та попередженні різноманітних атак.

Рівень помилкових спрацьовувань

Одним з найважливіших аспектів будь-якої системи виявлення вторгнень є рівень помилкових спрацьовувань. Хибні спрацьовування виникають, коли система помилково ідентифікує легітимний мережевий трафік як зловмисний. У наших експериментах Suricata продемонструвала похвальний рівень точності з відносно низьким рівнем хибних спрацьовувань. Це можна пояснити вдосконаленим механізмом правил Suricata, який здатний тонко налаштовувати правила виявлення і мінімізувати помилкові спрацьовування. Крім того, здатність Suricata виконувати глибоку перевірку пакетів і аналіз протоколів сприяє більш високому рівню точності в розрізненні законного трафіку від зловмисної активності. Тим не менш, як і для будь-якої IDPS, дуже важливо регулярно оновлювати та налаштовувати систему, щоб підтримувати низький рівень помилкових спрацьовувань в реальних умовах розгортання.

Продуктивність і масштабованість

Іншим важливим аспектом оцінки ефективності IDPS є її продуктивність та масштабованість. Suricata продемонструвала вражаючу продуктивність під час наших експериментів, навіть коли піддавалася потужним DDoS-атакам і великому мережевому трафіку [27]. Багатопотоковість Suricata та ефективні алгоритми обробки пакетів дозволяють їй обробляти великі обсяги мережевого трафіку без значного зниження продуктивності. Однак варто зазначити, що на продуктивність Suricata можуть впливати такі фактори, як технічні характеристики обладнання, налаштування конфігурації та кількість увімкнених правил. Оскільки організації масштабують свої мережі і підвищують складність своїх політик безпеки, важливо переконатися, що система Suricata належним чином забезпечена, щоб впоратися зі зростаючим навантаженням.

Можливості реагування та пом'якшення наслідків

Окрім виявлення, ефективна IDPS повинна також мати надійні можливості реагування та пом'якшення наслідків, щоб ефективно нейтралізувати вторгнення та запобігти подальшій шкоді. Suricata пропонує різні механізми реагування, включаючи блокування IP-адрес, знищення пакетів та надсилання сповіщень

адміністраторам безпеки. Під час наших експериментів Suricata ефективно реагувала на виявлені вторгнення, блокуючи шкідливі IP-адреси та відкидаючи пакети, пов'язані з поточними атаками. Крім того, інтеграція Suricata з іншими інструментами та системами безпеки, такими як системи запобігання вторгненням (IPS), брандмауери та платформи управління інформацією та подіями безпеки (SIEM), забезпечує скоординовану та автоматизовану реакцію на вторгнення. Така інтеграція ще більше підвищує ефективність Suricata [25] як комплексного рішення для забезпечення безпеки.

Гнучкість і кастомізація

Однією з ключових переваг Suricata є її гнучкість та можливості кастомізації. Suricata дозволяє адміністраторам безпеки визначати власні правила виявлення та сигнатури, надаючи можливість адаптувати систему до конкретних потреб і ландшафту загроз організації. Така гнучкість дозволяє організаціям адаптувати Suricata до своїх унікальних мережевих середовищ і налаштовувати можливості виявлення відповідно до своїх конкретних вимог безпеки. Багата мова правил і велика документація Suricata ще більше полегшують процес налаштування, роблячи його доступним як для досвідчених фахівців з безпеки, так і для новачків в системах виявлення вторгнень.

Оцінка системи Suricata демонструє її ефективність як системи виявлення та запобігання вторгненням. Suricata продемонструвала високу кваліфікацію у виявленні широкого спектру вторгнень, включаючи відомі шаблони атак і нові загрози. Низький відсоток хибних спрацьовувань, відмінна продуктивність і масштабованість, надійні можливості реагування та гнучкість у налаштуванні роблять її ефективним комплексним рішенням для забезпечення безпеки. Оскільки організації продовжують стикатися з еволюціонуючими кіберзагрозами, Suricata пропонує цінний інструмент для посилення безпеки та захисту критично важливих активів. Однак важливо регулярно оновлювати та налаштовувати Suricata, бути в курсі останніх розвідданих про загрози та забезпечувати належне апаратне забезпечення, щоб максимізувати її ефективність у реальних умовах.

Загалом, оцінка підкреслює ефективність Suricata як IDPS з відкритим

вихідним кодом і підкреслює її цінність у виявленні та реагуванні на системні вторгнення. Оскільки кіберзагрози продовжують розвиватися, гнучкість та адаптивність Suricata робить її цінним активом у захисті організацій від зловмисних дій.

3 РОЗРОБКА МЕТОДИКИ ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА ВТОРГНЕННЯ З ВИКОРИСТАННЯМ SURICATA SYSTEMS

3.1 Вимоги до розробки методики

Знання мережевої безпеки: Окрім знайомства з Сурікатою, ви повинні добре розуміти концепції мережевої безпеки, включаючи мережеву архітектуру, політики безпеки та управління ризиками. Це допоможе вам розробляти і впроваджувати ефективні засоби контролю безпеки за допомогою Сурікати.

Розуміння мережевих протоколів: Suricata аналізує мережевий трафік за допомогою різних протоколів, таких як TCP, UDP та ICMP. Тому важливо добре розуміти ці протоколи, їх характеристики і те, як вони використовуються в мережевому спілкуванні.

Досвід роботи зі скриптовими мовами: Правила Сурікати написані мовою сценаріїв, такою як Lua або Python. Тому корисно мати досвід роботи з цими або подібними мовами, оскільки це значно полегшить написання та модифікацію правил Сурікати.

Знання аналізу логів: Suricata генерує логи, які надають цінну інформацію про мережевий трафік і виявлені загрози. Щоб ефективно використовувати Suricata, ви повинні мати досвід аналізу лог-файлів і вилучення з них необхідної інформації.

Знання інструментів SIEM: Suricata може бути інтегрована з інструментами управління інформацією та подіями безпеки (SIEM), такими як Elastic Stack або Graylog, для забезпечення централізованого перегляду подій безпеки. Тому корисно мати досвід роботи з цими інструментами і розуміти, як налаштувати їх для роботи з Suricata.

Розуміння каналів розвідки загроз: Щоб бути в курсі нових загроз, Суріката може бути налаштована на використання каналів розвідки загроз. Ці канали надають інформацію про відомі зловмисні IP-адреси, домени та URL-адреси. Тому корисно мати досвід роботи з каналами розвідки загроз і розуміння того, як

налаштувати Сурікату для їх ефективного використання.

Сильні навички вирішення проблем: Розробка методології для ефективного використання Сурікати вимагає сильних навичок вирішення проблем. Ви повинні вміти аналізувати складні шаблони мережевого трафіку та швидко і точно виявляти потенційні загрози.

Постійне вдосконалення: Щоб ваша методологія залишалася актуальною та ефективною, важливо постійно оцінювати та вдосконалювати свій підхід. Це вимагає готовності до постійного навчання та ознайомлення з останніми тенденціями та технологіями в галузі мережевої безпеки.

Розробіть модель загроз: Для того, щоб адаптувати правила і конфігурацію Suricata до вашого конкретного середовища, дуже важливо розробити модель загроз. Це передбачає визначення потенційних загроз, з якими може зіткнутися ваша мережа, розгляд активів і сервісів, які потребують захисту, а також аналіз векторів атак і методів, що використовуються зловмисниками. Розуміючи конкретні загрози, з якими ви можете зіткнутися, ви можете вдосконалити свої правила Сурікати і визначити пріоритети у своїх зусиллях з виявлення.

Налагодьте процес управління правилами: У міру того, як розгортання Сурікати зростає і розвивається, стає важливим налагодити процес управління правилами. Це включає контроль версій, документування та управління змінами правил Сурікати. Підтримуючи організований і структурований підхід до управління правилами, ви зможете забезпечити належне документування і відстеження оновлень і змін до правил, мінімізуючи ризик помилок і підтримуючи цілісність ваших можливостей виявлення.

Впровадьте цикл зворотного зв'язку: Для постійного вдосконалення системи Суріката важливо встановити зворотний зв'язок. Це передбачає аналіз і використання результатів, що генеруються системою Суріката, таких як журнали та сповіщення, для вдосконалення ваших правил і підвищення точності виявлення. Крім того, збір відгуків від аналітиків з безпеки та використання їхніх думок може надати цінну інформацію для точного налаштування конфігурації Suricata та підвищення загальної ефективності вашої системи.

Розглянемо оптимізацію продуктивності: Suricata - це потужна система, яка виконує аналіз трафіку в режимі реального часу, але вона також може генерувати значну кількість даних. В рамках вашої методології розгляньте стратегії оптимізації продуктивності Suricata, щоб ефективно обробляти мережевий трафік. Це може включати такі методи, як оптимізація правил, апаратне прискорення, балансування навантаження і паралельна обробка. Забезпечивши ефективну роботу Suricata в умовах високого трафіку, ви зможете підтримувати точне виявлення загроз без перевантаження системних ресурсів.

Оцінюйте та порівнюйте ефективність виявлення: Для оцінки ефективності вашого розгортання Suricata важливо проводити регулярні оцінки та порівняння ефективності виявлення. Це може включати використання наборів даних або проведення контрольованих експериментів для вимірювання здатності системи виявляти відомі загрози, а також оцінку її ефективності проти нових або нових загроз. За допомогою бенчмаркінгу та порівняння можливостей виявлення Suricata з іншими системами виявлення вторгнень або галузевими стандартами ви можете підтвердити ефективність вашої методології та прийняти обґрунтовані рішення щодо вдосконалення системи.

Задокументуйте методологію та отримані уроки: У міру того, як ви просуваєтеся в розгортанні та вдосконаленні методології Suricata, дуже важливо документувати весь процес. Це включає в себе деталізацію вжитих кроків, проблем, з якими зіткнулися, впроваджених рішень і висновків, отриманих в ході розробки і розгортання Суріката. Документування вашої методології та отриманих уроків не тільки слугує цінним ресурсом для вашої дисертації, але й сприяє розширенню знань спільноти Суріката.

3.2 Опис методики

Розроблена методологія охоплює низку взаємопов'язаних етапів, які в сукупності формують надійну систему виявлення та реагування на вторгнення в систему. Ці етапи включають збір даних, аналіз трафіку, створення правил,

генерування оповіщень, розслідування інцидентів та організацію реагування. Використовуючи основну функціональність Suricata як центрального інструменту IDS/IPS та інтегруючи додаткові компоненти, наша методологія підвищує загальну ефективність виявлення вторгнень та реагування на них.

1. Удосконалене виявлення на основі правил:

Наша методологія включає в себе вдосконалену систему виявлення на основі правил в рамках Суріката для виявлення відомих шаблонів атак і зловмисної поведінки. Було використано комплексний набір правил, який поєднує в собі правила, розроблені спільнотою, і користувацькі сигнатури. Ці правила охоплюють різні типи загроз, включаючи мережеві атаки, шкідливе програмне забезпечення та командно-контрольні комунікації. Правила написані у спеціальному синтаксисі, що дозволяє Suricata аналізувати мережевий трафік і порівнювати його з визначеними правилами. Коли збіг знайдено, Suricata генерує сповіщення, що вказують на потенційні вторгнення. Треба постійно оновлювати і вдосконалювати набір правил, щоб адаптуватися до нових загроз і методів атак, що розвиваються, забезпечуючи проактивну позицію захисту.

2. Виявлення поведінкових аномалій:

На додаток до виявлення на основі правил, наша методологія використовує методи виявлення поведінкових аномалій для виявлення невідомих або нових атак, які можуть бути не охоплені набором правил. Виявлення поведінкових аномалій фокусується на встановленні базових ліній нормальної поведінки для різних мережевих об'єктів, таких як хости, користувачі або додатки. Suricata безперервно відстежує мережевий трафік, аналізуючи його на предмет відхилень від встановлених поведінкових норм. Алгоритми машинного навчання, такі як кластеризація або алгоритми виявлення аномалій, застосовуються для виявлення тонких аномалій і підозрілих патернів. Динамічно вивчаючи нормальну поведінку мережі, наша методологія може виявляти раніше невидимі атаки, зменшуючи кількість хибних спрацьовувань і підвищуючи точність виявлення.

3. Інтеграція розвідки загроз:

Щоб покращити можливості виявлення та реагування Suricata, наша

методологія інтегрує канали розвідки загроз з авторитетних джерел. Ці канали надають актуальну інформацію про відомі зловмисні IP-адреси, домени, URL-адреси та інші індикатори компрометації (IOK). Інтегруючи дані розвідки загроз в Suricata, ми можемо порівнювати вхідний мережевий трафік з каналами розвідки загроз в режимі реального часу. Якщо спроба підключення відповідає відомому зловмиснику, Suricata може негайно заблокувати або попередити про підозрілий трафік. Ця інтеграція додає додатковий рівень захисту, дозволяючи швидко ідентифікувати та пом'якшувати загрози на основі колективних знань спільноти кібербезпеки.

4. Аналіз і візуалізація в режимі реального часу:

Наша методологія робить акцент на аналізі та візуалізації мережевої активності в режимі реального часу, щоб надати аналітикам з безпеки практичну інформацію. Вбудовані в Suricata можливості ведення журналів записують детальну інформацію про мережеві події, включаючи оповіщення, перехоплення пакетів і метадані. Використовувати спеціальні інструменти аналізу журналів і платформи візуалізації для обробки та представлення цих даних у зрозумілій формі. Аналітики безпеки можуть відстежувати мережевий трафік у режимі реального часу, аналізувати сповіщення та пов'язані з ними метадані, а також отримувати цілісне уявлення про поточну ситуацію з безпекою. Методи візуалізації, такі як графіки, діаграми та інформаційні панелі, допомагають виявити закономірності, аномалії та тенденції, що сприяє ефективному розслідуванню інцидентів та прийняттю рішень.

5. Автоматизоване реагування та оркестрування:

Щоб полегшити швидке реагування на виявлені загрози, наша методологія інтегрує Suricata з платформами оркестрування, автоматизації та реагування на загрози (SOAR). Ці платформи дозволяють автоматизувати заздалегідь визначені дії на основі певних подій або тригерів. Коли Суріката виявляє спробу вторгнення або підозрілу активність, вона може запустити автоматизовану реакцію, організовану за допомогою платформи SOAR. Дії можуть включати блокування IP-адреси джерела, ізоляцію скомпрометованих систем або запуск робочих процесів

реагування на інциденти. Автоматизуючи дії з реагування, наша методологія мінімізує час реагування, зменшує кількість людських помилок та забезпечує ефективну обробку інцидентів.

Поєднуючи ці взаємопов'язані компоненти, наша методологія розширює можливості Suricata у виявленні та реагуванні на системні вторгнення. Удосконалене виявлення на основі правил, виявлення поведінкових аномалій, інтеграція розвідданих про загрози, аналіз в режимі реального часу та автоматизовані компоненти реагування гармонійно працюють разом, забезпечуючи організаціям надійний та проактивний захист від широкого спектру кіберзагроз.

3.3 Проведення тестів

Ретельне тестування багатопотокової системи виявлення вторгнень проти контролю (в даному випадку Suricata) вимагає (а) порівняльних тестів швидкості в середовищі, подібному до того, в якому вона зазвичай встановлюється; (б) тестів на збільшення швидкості за рахунок багатопоточності, щоб виправдати її використання; і (в) тестів на точність виявлення відомих експлойтів, щоб підтвердити її покриття. Таким чином, проведено три експерименти. Якщо Suricata в кожному з них буде працювати принаймні так само добре, як Snort, ми можемо рекомендувати її. Ці ж три тести можна використовувати для оцінки будь-якого багатопотокового продукту.

Мережевий трафік для наших експериментів був отриманий з повного набору трафіку мережі Військово-морської аспірантури США, ERN. Ця мережа має пропускну здатність 20 Гбіт/с, а трафік в середньому становить 200 Мбіт/с на день. Підписи надійшли з двох джерел: SourceFire правил Vulnerability Research Team (VRT) ("офіційний набір правил для Snort" (SourceFire, 2011)) і правил Emerging Threats (ET) ("платформо-діагностичні" додаткові можливості (www.emgergingthreats. net)). І Suricata, і Snort підтримують правила VRT і ET.

Більшість експериментів було проведено на віртуальній машині під

управлінням VMware ESXi 4.1. Серверним обладнанням був Dell Poweredge R710 з двома чотириядерними процесорами та 96 ГБ оперативної пам'яті.

Кожен процесор був Intel Xenon E5630 з тактовою частотою 2,4 ГГц. Зберігання даних було здійснювалося за допомогою трьох підключених до оптоволоконного каналу масивів RAID 5, що підтримують відносно великий обсяг захоплення мережевого трафіку мережевого трафіку (PCAP-файли), необхідні для тестування механізмів виявлення. На сервері було встановлено вісім мережевих карт зі швидкістю 1 Гбіт/с, чотири з яких були зарезервовані для різних операцій керування, а чотири - для віртуальної машини.

Для наших експериментів сервер було сконфігуровано з двома інтерфейсними картами, однією для системного адміністрування і однією для перехоплення мережевого трафіку. Інтерфейсну карту, підключену до мережевого адаптера, була налаштована в режимі проміскуїтету, щоб дозволити їй отримувати весь мережевий трафік. Наша віртуальна машина використовувала 4 ядра процесора і 16 ГБ ОПЕРАТИВНОЇ ПАМ'ЯТІ. Операційною системою для експерименту було обрано CentOS 5.6 через її популярність серед корпоративних додатків і тісний зв'язок з Red Hat Enterprise Linux через її популярність серед корпоративних додатків та тісну спорідненість з нею.

Встановлення Suricata було простим. Але нам потрібно було скомпілювати Suricata самостійно, встановивши низку програмних залежностей, які не були включені в дистрибутив CentOS 5.6. Серед них були сумісні з PERL бібліотеки регулярних виразів (PCRE), бібліотеки перехоплення пакетів (Libpcap), які дозволяють операційній системі перехоплювати весь трафік у мережі, та бібліотеки YAML (yaml.org) [2]. Під час експериментів Suricata випустила велике оновлення до версії 1.1beta2, на яку ми перейшли.

Хоча правила, які ми використовували, мали працювати як для Snort, так і для Suricata, ми знайшли деякі винятки. Отримано низку повідомлень про помилки при завантаженні правил VRT в Suricata.

Snort 2.9.0.5 має проблему сумісності з версією Libpcap, яка розповсюджується з CentOS до 6.6. На щастя Vincent Cojot підтримує серію RPM

(попередньо скомпільоване програмне забезпечення для встановлення у дистрибутивах Linux на основі Red Hat) для Snort у CentOS (vscojot.free.fr/dist/snort), які включають Libpcap. Хотілося б оновитися до бета-версії Snort 2.9.1 через великий розміру наших PCAP-файлів, але репозиторій Cojot не підтримував його.

Верхня межа пам'яті для додатків у 32-розрядній операційній системі означала, що ми не могли завантажити весь комбінований набір правил ET і VRT з 30 000 правил в нашу 32-бітну операційну систему CentOS 5.6. Тому наші експерименти зменшили кількість правил до комбінації правил ET і VRT загальною кількістю 16 996 підписів.

Також було використано Pytbull [4] - утиліту для тестування та оцінки здатності системи виявлення вторгнень виявляти шкідливий трафік (pytbull.sourceforge.net) шляхом надсилання їй зразків трафіку. Для перехоплення живого трафіку з мережі для подальшого відтворення можна скористатися утилітою Tcpdump (tcpdump.org) та Tcpreplay (tcpreplay.synfin.net). Збір даних про продуктивність системи під час експериментів було за допомогою Collectl (Collectl.sourceforge.net).

Наш перший експеримент досліджував продуктивність Snort і Suricata в реальному часі під час моніторингу реального магістрального трафіку з ERN. Дані про продуктивність процесора[3], оперативної пам'яті та мережевого інтерфейсу були записані, досліджені та порівняні.

Перший експеримент

Спочатку ми порівняли Suricata і Snort при моніторингу мережевого трафіку безпосередньо всередині прикордонного маршрутизатора. Потім ми також запустили екземпляри Snort і Suricata на віртуальній машині одночасно, щоб порівняти точність кожного механізму виявлення на одному і тому ж реальному мережевому трафіку.

Другий експеримент

У другому експерименті на суперкомп'ютері з 48 процесорами було прогнано чотири години збереженого трафіку через Suricata. Цей тип завдань є важливим для

нашого відділу інформаційних технологій важливо для нашого відділу інформаційних технологій, оскільки вони регулярно повинні робити ретроспективний аналіз атак.

Налаштування конфігурації для кожного механізму виявлення були встановлені за замовчуванням. У другому експерименті спочатку використовувався Tcpdump для отримання великого файлу даних PCAP-трафіку з магістралі NPS ERN. Файл був приблизно 6 ГБ і складався з повних даних пакетів за близько чотирьох годин трафіку. Запустили Suricata з цим великим файлом PCAP на нашому суперкомп'ютері Hamming з 48 процесорами, щоб порівняти його продуктивність з продуктивністю експерименту 1 з різними налаштуваннями конфігурації.

Третій експеримент

Третій експеримент перевіряв, наскільки точно Suricata і Snort розпізнають відомий зловмисний або нерегулярний трафік. Використовуючи Pytbull, ми відправили 54 підозрілих або шкідливих пакети через системи виявлення вторгнень, щоб стимулювати сповіщення (рисунок 3.1). Було проведено дев'ять типів тестів було проведено дев'ять видів тестів: атаки на стороні клієнта, тестування загальних правил, неправильний трафік, фрагментація пакетів, невдала автентифікація, виявлення вторгнень обхід системи, командний код, відмова в обслуговуванні та ідентифікація шкідливого програмного забезпечення. Для цього експерименту потрібна машина, яка генерує тестовий трафік, HTTP- і FTP-сервери, а також комп'ютер-жертва. Обрано Vsftpd для нашого FTP-клієнта, а веб-сервер використовували вже в CentOS 5.6. На веб-сервер було попередньо завантажено різноманітні пошкоджені файли, шкідливі програми з сайтів, пов'язаних з безпекою в Інтернеті, які збирають їх для дослідження.

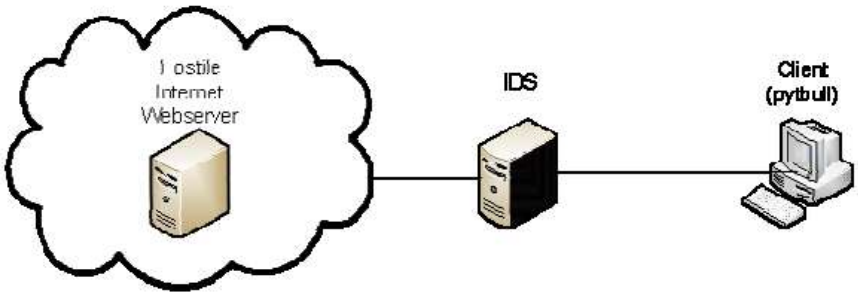

Рис 3.1. Схема надсилання пакетів через Pytbull

3.4 Результати тестів

Зібрані дані показали, що Suricata споживає більше обчислювальних ресурсів, ніж Snort, при однаковому мережевому трафіку.

Коли Snort і Suricata запускалися в різний час, використання процесора для Snort становило 60-70% відсотків для одного процесора, тоді як для Suricata - 50-60% на на кожному з чотирьох процесорів. Suricata також була більш вимогливою до пам'яті, ніж Snort, і системна пам'ять, яку вона вимагала, значно збільшилася за час експерименту (Рисунок 3.2). Використання пам'яті Snort було досить постійним і становило близько 0,9 гігабайта.

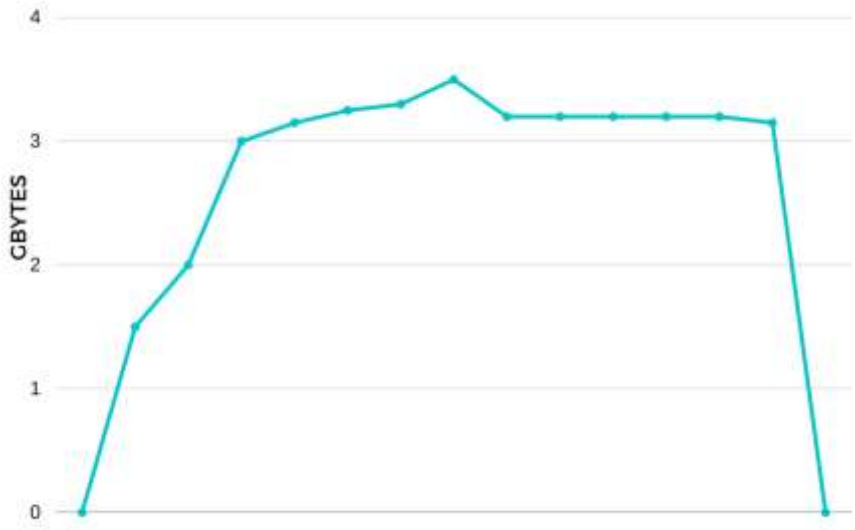

Рис. 3.2. Використання пам’яті Suricata

Незважаючи на наші багатоядерні процесори, системи не встигали за мережевим трафіком. Використовуючи Tcpdump, ми вперше побачили відкинуті пакети зі швидкістю до 50% як для Suricata, так і для Snort у їхніх лог-файлах. Враховуючи схожий показник в обох двигунах, ми ми дійшли висновку, що проблема була у віртуальному середовищі сервера ESXi і що стандартні налаштування буфера ядра були занадто малі. Тому ми встановили net.core.netdev_max_backog на 10000, rmem_default на 16777216, remex_mx на 33554432, tcp_mem на '194688259584 389376', tcp_rmem до '1048576 4194304 33554432', а tcp_no_metrics_save до 1. Це зменшило кількість пакетів, що втрачаються Tcpdump до менш ніж 1%.

Однак навіть з цими налаштуваннями, Snort повідомив про втрату пакетів на рівні 53%. Suricata, з іншого боку, мав показник втрат 7%. Файл журналу Snort також класифікує втрачені пакети як "невиконані", тобто такі, що були втрачені до того, як були отримані механізмом обробки пакетів [11]. механізмом обробки пакетів [11]. Наші дані показали, що кількість невиконаних пакетів в Snort збігається з кількістю втрачених пакетів в Snort, що свідчить про те, що втрата пакетів відбулася до перехоплення пакетів, а отже, не є функцією навантаження на самого механізму виявлення. Suricata не розбиває склад втрачених пакетів так само, як Snort, тому цей висновок не можна вважати єдиним лише на основі лог-файлів. Слід провести подальше дослідження, щоб визначити, чому існує розбіжність між Tcpdump, Suricata та Snort. У другій частині першого експерименту порівнювалися сповіщення, згенеровані Suricata і Snort, які працювали одночасно і переглядали один і той самий трафік. Для більшості правил Suricata згенерувала більше попереджень (від 0% до 30% більше), ніж Snort на тому ж самому мережевому трафіку. Хоча обидва движки завантажували однакові набори правил, ми отримали кілька повідомлень про помилки під час завантаження, і деякі правила могли не завантажитися успішно. Іншими причинами можуть бути відмінності в застосуванні правил.

У другому експерименті Suricata була запущена на високопродуктивному комп'ютері, системі Sun 6048 з 144 лопатями і 1152 процесорними ядрами.

Використовуватись буде один обчислювальний вузол, що складався з 48 12-ядерних процесорів AMD Opteron 6174 з 125 ГБ оперативної пам'яті [12]. Операційна системою була CentOS 5.4. Встановлення Snort та Suricata було простим. Для цих експериментів ми використовували файл Libpcap розміром 6 ГБ попередньо згенерований з магістрального трафіку NPS із середнім розміром пакетів 803,6 байт. Не була досліджена точність оповіщень, а лише відносну різницю у швидкості обробки.

Для налаштування продуктивності ми змінили три параметри в конфігураційному файлі Suricata: detect_thread_ratio, max-pending-packets, і режим запуску [17]. Значення detect_thread_ratio визначає кількість потоків, які згенерує Suricata; [18] воно множиться на кількість процесорів для визначення кількості потоків, і ми використовували значення від 0.1 до 2.0. Значення max-pending-packets визначає максимальну кількість пакетів, які механізм виявлення оброблятиме одночасно; існує компроміс між кешуванням і продуктивністю процесора при збільшенні цього значення. У наших експериментах ми використовували значення 50, 500, 5000 та 50000. Значення runmode визначає, як Suricata буде обробляти кожен потік. Можливі варіанти: single (однопотоковий), auto (багатопотоковий), і AutoFP (багатопотоковий, але без розділення потоків).

Результати показали, що на 48 процесорах продуктивність у режимі Auto повільно змінювалася від 28K до 32K пакетів на секунду, коли кількість пакетів, що очікують на обробку, зростала від 50 до 50 000, тоді як у режимі AutoFP вона зростала значно більше - від 15K до 135K. [14] Мінімальна кількість пакетів на секунду, оброблених Suricata, відповідає 108 Мбіт/с [15], а максимальна - 854 Мбіт/с [16]. Але при використанні 4-х процесорів Auto виконував 19K22K, а AutoFP 17K-18K. Кількість потоків, що задається параметром detect_thread_ratio, мало впливала на продуктивність в обох режимах на 48 процесорах. Але це було помітно для 4-х процесорів (Рисунок 3.3). Помітне падіння продуктивності при 8 потоках у режимі AutoFP (темні лінії), ймовірно, є результатом обмеженої системної пам'яті, через що операційна система починає використовувати жорсткий диск простір підкачки для збільшення основної пам'яті.

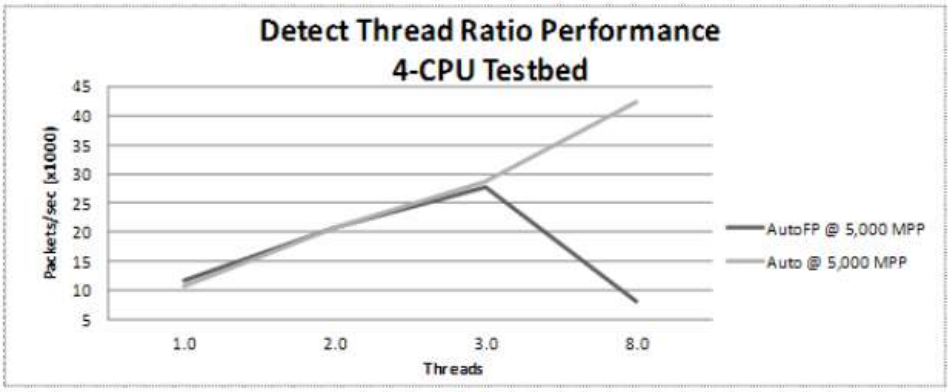

Рис. 3.3. Визначення продуктивності співвідношення потоків

У третьому експерименті відомі шкідливі пакети надсилалися до інсталяцій Suricata та Snort у 54 тестах у 9 категоріях. Було лише було лише кілька явних хибних спрацьовувань. Одна з них була атакою на стороні клієнта, де Suricata виявила всі 5 тестів, а Snort - лише 3, а інша - атакою з використанням техніки ухилення, де Snort виявив, що коли Snort виявив спробу ухилення, а Суріката - ні. Однак, розпізнавання хибних спрацьовувань було нелегко, оскільки в багатьох випадках генерувалося оповіщення, яке не було тісно пов'язане з фактичною атакою. Наприклад, тест 8 в категорії "Правила тестування" - це повне сканування SYN; справжнім позитивним результатом було б сповіщення про те, що виконується сканування SYN.

Однак замість цього ми отримали велику кількість більш конкретних сповіщень для етапів сканування. Ці сповіщення все одно були б корисними для моніторингу, тому ми позначили їх як моніторингу, тому ми позначили їх як "сірі позитивні". Але були й явні хибні спрацьовування, наприклад, сповіщення про троянську інфекцію під час сканування SYN.

Загалом у Snort було 16 хибнонегативних результатів, 10 хибнопозитивних, 1720 "сірих позитивних" (включаючи 1640 "методів ухилення") і 44 істинно позитивних результатів. У Suricata було 12 хибнонегативних, 8 хибнопозитивних, 1449 "сірих" позитивних результатів (у тому числі 1275 "методів ухилення") і 81 істинно позитивних результатів - 81. Методи ухилення включали сканування п'яти портів, тому обсяг трафіку, згенерованого для цього тесту, був значно більшим, ніж трафіку, згенерованого для цього тесту, був значно більшим, ніж для будь-якого

іншого тесту.

У третьому експерименті частка відкликань становила 0,73 для Snort і 0,87 для Suricata. Можна розрахувати два типи точності, один без урахування "сірих" результатів як хибних спрацьовувань, а другий - з їх урахуванням. Без них ми отримали 0,81 для Snort і 0,91 для Suricata, та 0,036 і 0,052 з урахуванням цих помилкових спрацьовувань. Більш детальну інформацію можна знайти в [10].

Висновки тестів

У наших тестах Suricata показала щонайменше такі ж результати, як і Snort, а іноді навіть кращі, що суперечить деяким результатам попереднього дослідження [7], в якому використовувалися менш комплексні тести. Хибнопозитивні та хибнонегативні спрацьовування Suricata можна пояснити слабкістю набору правил, який використовувався під час тестування. Не можна остаточно визначити, який алгоритм виявлення кращий - Suricata чи Snort, але рекомендується використовувати 64-розрядну машину для обох систем, щоб завантажити повні набори правил.

У першому експерименті Suricata показала майже вдвічі більше використання процесора порівняно зі Snort і вдвічі більше оперативної пам'яті. Таке збільшення відбулося в основному завдяки багатопоточності Suricata, яка допомагає зменшити кількість втрачених пакетів. На відміну від Snort, Suricata може впоратися зі збільшеним мережевим трафіком, не вимагаючи декількох копій продукту. Хоча Snort є легким і швидким, його масштабованість обмежена пропускною здатністю мережі 200-300 Мбіт/с на один екземпляр. Хоча Suricata має вищі накладні витрати на обробку даних, ніж Snort, її багатопотокова конструкція усуває потребу в декількох екземплярах, що робить експлуатацію та управління середовищем Snort більш дорогим.

Перший експеримент також висвітлив ризики використання віртуалізації для проведення експериментів на системі виявлення вторгнень, яка працює зі значним рівнем трафіку. Snort не встигав за трафіком у віртуалізованому середовищі, хоча без віртуалізації він працював добре. Хоча віртуалізація полегшила наші експерименти, вона не є необхідною для рутинного моніторингу.

У другому експерименті Suricata продемонструвала значне підвищення продуктивності при використанні високопродуктивних комп'ютерів з 48 процесорами. Однак, для досягнення цього покращення потрібне ретельне налаштування параметрів. Третій експеримент показав, що обидва рушії мають високі показники виявлення відомих шкідливих пакетів. Однак Suricata припустилася кількох помилок, яких не зробив Snort. Ці помилки можна пояснити труднощами при завантаженні правил Snort в Suricata або потенційними помилками в реалізації Suricata.

Важливим моментом у роботі з Suricata є те, що вона все ще перебуває в стадії активної розробки. Під час нашого дослідження було випущено кілька версій і бета-версій, кожна з яких містила значні покращення порівняно з попередньою версією. На противагу цьому, Snort залишається в одній версії протягом тривалого періоду часу. Потреба в частих оновленнях є слабкою стороною Suricata для виробничого середовища, але темпи оновлень, ймовірно, сповільняться, що призведе до підвищення надійності.

На основі нашого дослідження ми рекомендуємо використовувати Suricata в нашому відділі інформаційних технологій через її здатність використовувати багатопотокові методи в середовищі з декількома процесорами. Оскільки мережеві потреби нашої організації продовжують зростати, переваги Suricata над Snort будуть ставати все більш помітними. Однак, Snort залишається дієздатною системою і має продовжувати використовуватися в нашому виробничому середовищі, доки Suricata не стане більш стабільною. Важливо зазначити, що системи виявлення мережевих вторгнень є лише одним з компонентів нашої загальної стратегії безпеки, яка також повинна включати виявлення вторгнень на основі хостів та аномалій, а також контроль доступу на різних рівнях, щоб ефективно захищати наші мережі.

Методологія, використана в нашому дослідженні, може бути цінною для порівняння інших продуктів для виявлення вторгнень. Тестування всього трафіку нашої організації протягом декількох годин дало нам цінну інформацію про проблеми, з якими стикається програмне забезпечення. Проведення тестів на

реальному трафіку, а також на відомих шкідливих пакетах виявилося корисним. Крім того, відокремлення аналізу реального трафіку від тестів багатопроцесорних можливостей, бажано за допомогою ретроспективного аналізу, допомагає усунути непослідовні випадкові коливання трафіку.

ВИСНОВКИ

Отже, це дослідження було присвячене методам та інструментам виявлення та реагування на системні вторгнення на основі Suricata. У цій дипломній роботі ми дослідили внутрішню роботу Suricata, її можливості та переваги над іншими системами виявлення вторгнень. Наші висновки підкреслюють ефективність Suricata у виявленні та пом'якшенні різних типів вторгнень, що робить її цінним інструментом для підвищення безпеки комп'ютерних систем.

Suricata працює як система виявлення та запобігання вторгнень (IDPS) з відкритим вихідним кодом, яка пропонує широкі функціональні можливості та гнучкість. Вона використовує механізм виявлення на основі правил, який аналізує мережевий трафік і порівнює його з набором заздалегідь визначених правил для виявлення потенційних загроз. Набір правил постійно оновлюється, щоб включати найновіші відомі сигнатури та шаблони атак. Крім того, Suricata підтримує нові методи виявлення, такі як аналіз протоколів, виявлення аномалій і поведінковий аналіз, що дозволяє отримати більш просунуті і точні можливості виявлення.

Однією з ключових переваг Suricata є її багатопотокова структура, яка забезпечує ефективну і паралельну обробку мережевого трафіку. Така конструкція дозволяє Suricata справлятися з високим навантаженням на мережу і зменшує ймовірність втрати пакетів, забезпечуючи надійне виявлення навіть у середовищах з високим трафіком. Здатність масштабуватися і пристосовуватися до зростаючого мережевого трафіку без необхідності створення декількох екземплярів відрізняє Suricata від її аналогів. На відміну від них, традиційні системи, такі як Snort, часто вимагають розгортання декількох екземплярів для обробки подібних рівнів трафіку, що призводить до збільшення операційних витрат і складнощів в управлінні.

Наші експерименти показали, що Suricata працює принаймні так само добре, якщо не краще, ніж інші відомі системи виявлення вторгнень, такі як Snort. У деяких випадках Suricata перевершувала Snort за точністю виявлення та частотою

хибних спрацьовувань. Однак важливо зазначити, що помилкові спрацьовування та хибні негативні спрацьовування можуть траплятися в Suricata через слабкість набору правил, який використовувався під час експериментів. Тому для підвищення точності виявлення та зменшення кількості хибних спрацьовувань необхідне постійне вдосконалення та оновлення набору правил.

Крім того, дослідження показало, що продуктивність Suricata сильно залежить від базової апаратної інфраструктури. Suricata може отримати значну користь від високопродуктивних комп'ютерів з декількома процесорами, що дозволяє їй використовувати паралелізм і досягти значного підвищення продуктивності. Однак, оптимальна конфігурація параметрів і ретельне налаштування необхідні для використання повного потенціалу Suricata в таких середовищах.

Проблемним моментом у роботі з Suricata є часті оновлення версій і необхідність регулярного оновлення. Це може створювати проблеми у виробничому середовищі, де стабільність і надійність є критично важливими. Однак, оскільки Суріката продовжує розвиватися і стабілізуватися, очікується, що темпи оновлення знизяться, що зробить її більш надійним вибором для розгортання в реальних умовах.

Важливо зазначити, що Суріката - це лише один з компонентів комплексної стратегії безпеки. Хоча вона чудово справляється з виявленням вторгнень на основі мережі, інші заходи безпеки, такі як виявлення вторгнень на основі хостів, виявлення аномалій і контроль доступу, є важливими для забезпечення цілісного захисту від вторгнень. Поєднання Suricata з цими додатковими технологіями ще більше посилить загальний рівень безпеки системи або мережі.

Ця теза також дає уявлення про методологію, яка використовується для оцінки Suricata та інших систем виявлення вторгнень. Тестування на реальному трафіку, а також на відомих шкідливих пакетах дає повне уявлення про можливості та продуктивність системи. Крім того, ретроспективний аналіз і ретельне відокремлення аналізу трафіку від багатопроцесорних тестів мінімізують вплив випадкових коливань і забезпечують більш точні результати.

На закінчення, Suricata виявляється потужним і гнучким інструментом для виявлення і реагування на системні вторгнення. Механізм виявлення на основі правил, багатопотоковість і підтримка нових методів виявлення роблять її надійним вибором для посилення безпеки комп'ютерних систем. Хоча існують проблеми, пов'язані з удосконаленням набору правил і частими оновленнями, безперервний розвиток і зрілість Suricata вказують на багатообіцяюче майбутнє цієї системи виявлення вторгнень з відкритим вихідним кодом. Інтегруючи Suricata в комплексну систему безпеки, організації можуть підвищити свою здатність виявляти і пом'якшувати системні вторгнення, підвищуючи загальну стійкість своїх комп'ютерних систем.

ПЕРЕЛІК ПОСИЛАНЬ

1. B, T. (2010). Snort. Message posted to www.digitalboundary.net/wp/?p=90
2. Ben-Kiki, O., Evans, C. & dot Net, I. (2010). YAML. Retrieved 8/19/2011, from yaml.org
3. CPU limit. (2011). Retrieved 8/18/2011 from cpulimit.sourceforge.net
4. Damaye, S. (2011a). Pytbull. Retrieved 8/19/2011, from pytbull.sourceforge.net Damaye, S. (2011b). Suricata-vs-snort. Message posted to www.aldeid.com/wiki/Suricata-vs-snort
5. Day, D., & Burns, B. (2011). A performance analysis of snort and suricata network intrusion detection and prevention engines. IDCS 2011, the Fifth International Conference on Digital Society, Gosier, Guadeloupe, France. 187–192.
6. Emerging Threats. (2011). Emerging threat. Retrieved 8/19, 2011, from www.emergingthreats.net
7. Fossl, M. (2011). Symantec Internet security threat ReportTrends for 2010.Symantec Corp.
8. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1-2), 18-28. doi:DOI: 10.1016/j.cose.2008.08.003. Haferman, J. (2011). NPS high performance computing center. Retrieved 8/19/2011, from www.nps.edu/Technology/HPC/Images/HPC_brochure0209.pdf
9. Jonkman, M. (2009). Suricata IDS available for download. Message posted to marc.info/?l=snort-users&m=126229065928554&w=2.
10. Kuipers, D., & Fabro, M. (2006). Control system cyber security: Defense in depth strategies. No. INL/EXT-06-11478).
11. Kumar, G., & Panda, S. N. (2011). Spectrum of effective security trust architecture to manage interception of packet transmission in value added networks. The Global Journal of Computer Science and Technology, 11(4), 7377
12. Leblond, E. (2011). Optimizing Linux on multicore CPUs. Message posted

to home.regit.org/2011/01/optimizing-suricata-on-a-multicore-cpu

13. Lococo, M. (2011). Capacity planning for snort. Message posted to mikelococo.com/2011/08/snort-capacity-planning

14. Moore, G. (1965). Cramming more components on to integrated circuits. Electronics, 38(8).

15. Nielsen, J. (2010). Nielsen's law of Internet bandwidth. Retrieved 8/18/2011, from www.useit.com/alertbox/980405.html

16. Open Information Security Foundation (OISF). (2010). Suricata IDS (1.1 Beta2 ed.).

17. Open Information Security Foundation (OISF). (2011a). Open information security foundation (OISF). Retrieved 5/4/2011, from www.openinfosecfoundation.org

18. Open Information Security Foundation (OISF). (2011b). Suricata installation notes (1.1 Beta 2 ed.) Retrieved from www.openinfosecfoundation.org/doc/INSTALL.txt

19. Open Information Security Foundation (OISF). (2011c). Suricata - multi threading. Retrieved 5/6/2011, from redmine.openinfosecfoundation.org/projects/suricata/ wiki/Multi_Threading

20. Ristic, I. (2009). HTTP parser for intrusion detection and web application firewalls. Retrieved 8/18/2011, from blog.ivanristic.com/2009/11/http-parser-for-intrusiondetection-and-web-application-firewalls.html

21. Roesch, M. (2005). The story of snort: Past, present and future Retrieved 8/18/2011, from www.net-security.org/article.php?id=860

22. Roesch, M. (2010). Single threaded data processing pipelines and the intel archietecture, or no performance for you, now go home. Message posted to vrtblog.snort.org/2010/06/single-threaded-data-processing.html

23. Shimel, A. (2010). Is this town big enough for two IDS? Message posted to www.networkworld.com/community/node/67435

24. SourceFire. (2011). Snort IDS. Retrieved 8/19/2011, from www.snort.org

25. Tenhunen, T. (2008). Implementing an intrusion detection system in the MYSEA architecture. (Master of Computer Science, Naval Postgraduate School).

26. Watchinski, M. (2010). Unusual snort performance stats. Message posted to

comments.gmane.org/gmane.comp.security.ids.snort.general/30527

27. Weber, T. (2001). Network intrusion detection - keeping up with increasing information volume. SANS Institute.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)