Міністерство освіти та науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

Автоматизовані тести систем виявлення вторгнень

Текстова частина до кваліфікаційної роботи за спеціальністю «Інженерія програмного забезпечення» - 121

Керівник кваліфікаційної роботи
доктор технічних наук
Глибовець Андрій Миколайович
_____(Підпис)
«__» _____ 2024 року
Виконав студент
ІПЗ-4 Дейнека Артем Валентинович
«__» _____ 2024 року

Київ 2024

Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра інформатики
Освітній ступінь бакалавр
Спеціальність 121 «Інженерія Програмного Забезпечення»
Освітня програма бакалавр

ЗАТВЕРДЖУЮ
Завідувач кафедри інформатики
Гороховський С. С.

“15” жовтня 2023 року

ЗАВДАННЯ
ДЛЯ КУРСОВОЇ РОБОТИ СТУДЕНТУ

1. Тема роботи «Автоматизовані тести систем виявлення вторгнень», керівник роботи Глибовець Андрій Миколайович, доктор технічних наук
2. Строк подання студентом роботи: 17 травня 2024
3. План роботи

Анотація
1. Дослідження та аналіз предметної області
1.1 Поняття IDS.
1.2 Види IDS
2. Розробка системи
2.1 Спосіб проведення тестування системи виявлення вторгнень
2.2 Вибір цільової системи виявлення вторгнень
2.3 Модуль збору даних про оповіщення
2.4 Модуль проведення атак
2.4.1 Збір даних про ціль
2.4.2 База даних атак
2.4.3 Проведення атаки та перевірка наявності сповіщення
Висновки

№ з/п	ПЕРЕЛІК РОБІТ	Термін виконання	Дата ознайомлення наукового керівника	Підпис наукового керівника	Примітки
1.	Вибір теми, затвердження її на засіданні кафедри та закріплення наукового керівника Узгодження календарного графіка підготовки кваліфікаційної роботи. Ознайомлення студента з критеріями оцінювання кваліфікаційної роботи (п. 8.5).	15 жовтня 2023			
2.	Вивчення джерел літератури, матеріалів архівів, періодичних видань, збір та узагальнення фактів, даних	15 жовтеня 2023– 19 листопада 2023			
3.	Складання плану каліф. роботи та узгодження з науковим керівником	19 листопада 2023			
4.	Написання розділів роботи	19 листопада 2020 – 14 березня 2024			
5.	Проміжний контроль виконання роботи	22 лютого 2024			
6.	Написання кваліфікаційної роботи в цілому, ознайомлення з її першим варіантом наукового керівника	11 січня 2024 – 29 березня 2024			
	Розділ 1 (постановка проблеми, теоретичні основи, огляд літературних джерел)	25 січня 2024			
	Розділ 2 (аналітично-дослідницька частина)	5 березня 2024			
	Розділ 3 (проектно-рекомендаційна частина)	21 березня 2024			
7.	Повне завершення написання кваліфікаційної роботи, оформлення її згідно з вимогами й подання на відгук науковому керівнику	13 квітня 2024 – 06 травня 2024			
8.	Подання кваліфікаційної роботи для перевірки письмових робіт студентів НаУКМА на відповідність вимогам академічної доброчесності,	17 травня 2024			
9.	Публічний захист кваліфікаційної роботи перед екзаменаційною комісією	згідно з розкладом роботи ЕК			

ГРАФІК ПІДГОТОВКИ КУРСОВОЇ РОБОТИ ДО ЗАХИСТУ

Графік узгоджено 10 жовтня 2023 р.
Науковий керівник Глибовець Андрій Миколайович
Виконавець курсової роботи Дейнека Артем

Зміст

Анотація

Захист комп'ютерних систем від кібератак завжди було є і буде важливим завданням, що стоїть перед тими, хто їх розробляє та підримує. Сьогодні існує багато комплексних рішень, які дозволяють відстежувати мережеву активність, виявляти загрози та своєчасно на них реагувати. Одними із таких рішень є система виявлення вторгнень(IDS). Ці системи дозволяють виявляти підозрілу активність у мережі та на окремих пристроях. Налаштування цих програмних рішень є непростим завданням і ціна помилки є надзвичайно високою. Отже, потрібні засоби, що дозволяють здійснювати перевірку належного функціонування цих систем і один із таких засобів ми спробували створити у ході цієї роботи.

1. Дослідження та аналіз предметної області

1.1 Поняття IDS.

Процес виявлення несанкціонованого доступу передбачає моніторинг подій у системі та їх аналіз на предмет *інцидентів*, тобто порушення політик безпеки, допустимої поведінки системи або стандартних безпекових процедур. *Система виявлення вторгнень*(intrusion detection system або IDS) – це програмне або апаратне рішення, що автоматизує вищеописаний процес[1]. Реагування на сповіщення IDS є завданням операційного центру

[1] Scarfone, Karen & Mell, Peter. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS).

безпеки(SOC), і як правило здійснюється або безпосередньо спеціалістами, або ж за допомогою систем протидії вторгненню(intrusion prevention system або IPS). Останні містять не лише засоби для виявлення загроз, а й інструменти для автоматизованого реагування на них. Більшість IDS рішень містять у собі функціонал IPS, а тому нерідко ці поняття вживаються як взаємозамінні, або ж вживається поняття IDPS(intrusion detection and prevention system)[1].

1.2 Види IDS

Системи виявлення вторгнень різняться за обсягом та способом дії. Так основними є два поділи: на host-based IDS(HIDS) та network-based IDS(NIDS) за розміром середовища, яке захищається; на anomaly-based та signature-based за способом виявлення порушень безпеки. Різниця між HIDS та NIDS очевидна із їх назв: перші захищають лише окремо взятий пристрій тоді як другі аналізують трафік всієї мережі. Signature-based системи виявляють порушення базуючись на так званих сигнатурах – зазделегідь відомих і описаних шаблонах поведінки атакуючої сторони. Ці IDS надійно захищають від відомих загроз, але потребують постійного оновлення баз даних сигнатур, оскільки не здатні виявляти нові загрози. З цієї ж причини при появі нового виду загроз такі системи є вразливими до атак, оскільки оновлення бази даних сигнатур займає певний час. Тому для цієї задачі краще підходять anomaly-based системи. Вони виявляють загрозу базуючись на відхиленні тих чи інших показників від норми. Так, наприклад, збільшення мережевого навантаження у 10 разів може свідчити про спробу DoS/DDoS атаки, а тому про таке відхилення від норми буде оповіщення. Системи засновані на аномаліях можуть виявляти загрози незалежно від новизни останньої, якщо та спричиняє аномальну поведінку у мережі. Проте, така система має одну суттєву проблему: необхідно правильно визначити «нормальну» поведінку[2]. Цього можна досягнути двома способами: створивши модель, що аналізує звичайний трафік і на його основі визначає межі норми для тих чи інших показників, або давши користувачу самостійно визначити правила, за якими має вести себе мережа і будь-яке відхилення від них вважати аномалією.

2. Розробка системи

2.1 Спосіб проведення тестування системи виявлення вторгнень

Як було зазначено вище, системи виявлення вторгнень можуть відрізнятись середовищами, які вони захищають та способами захисту. Втім, всі вони мають спільне призначення – виявляти загрози. Саме тому ми будемо перевіряти IDS атакуючи її та спостерігаючи за її реакцією(або її відсутністю). Оскільки кожна система виявлення вторгнень має свій API для доступу до сповіщень ми реалізуємо нашу систему тестування для окремо взятої IDS. Кінцеве рішення буде містити певний набір атак, які здійснюватиме наша програма та набір сповіщень, що мають бути надіслані при проведенні вищевказаних атак. Оскільки таке тестування проводить реальні атаки, ми маємо можливість реально оцінити стан системи захисту та відразу мати інформацію про те, на які саме вразливості варто звернути увагу.

2.2 Вибір цільової системи виявлення вторгнень

Розглягувши різноманітні IDS рішення ми вирішили зупинитися на Wazuh. Основними причинами такого вибору стали:

- Наявність готового SIEM рішення із REST API, інтегрованого з IDS[3]
- Відкритий вихідний код
- Можливість інтеграції інших NIDS

[2] Douligeris, Christos. (2007). Network Security : Current Status and Future Directions.
[3] Wazuh API REST(4.7.4). : URL: https://documentation.wazuh.com/current/user-manual/api/reference.html

Наявність SIEM дозволяє легко отримувати інформацію про наявність чи відсутність реакції на атаку і дозволяє не лише тестувати систему, а й надавати інформацію користувачеві про можливі вразливості у вигляді сповіщень SIEM. Можливість інтеграції NIDS надає можливість тестувати не лише сам Wazuh, а й інші IDS рішення, які інтегровані до нього. Наприклад, такий підхід дозволяє провести тестування IDS Suricata, якщо останню інтегрувати до існуючої Wazuh інфраструктури[4].

2.3 Модуль збору даних про оповіщення

Для збору інформації про сповіщення необхідно спочатку під'єднатись до Wazuh API. Для цього ми напишемо окремий клас WazuhAPI, який відповідатиме за отримання JWT токена та подальшу взаємодію із API.

```
class WazuhAPI:
    """
    This class handles connection and requests to Wazuh API server
    """

    PORT = 55000
    LOGIN_ENDPOINT = "security/user/authenticate"

    def __init__(self, hostname: str, credentials: tuple[str, str] = None,
token: str = None):
        """
        Creates a connection to Wazuh server API

        :param hostname: name of designated Wazuh server.
        :param credentials: pair of credentials (login, password) from Wazuh
server
        :param token: JWT obtained from Wazuh server. If credentials provided,
value of token will be discarded
        """
        self.hostname = hostname
        if credentials is None:
            if token is None:
                raise ValueError("Either credentials or token must be provided")
            self.token = token
        else:
            self.token = WazuhAPI.get_token(hostname, credentials[0],
credentials[1])
```

Конструктор класу негайно робить спробу з'єднатись та отримати токен, якщо користувач надав логін та пароль. Для отримання токена ми використали метод `get_token`, який містить необхідну реалізацію отримання токену. Далі ми реалізували метод `get_alerts`, який збирає статистику оповіщень безпеки. Для реалізації запитів було використано модуль Requests, а тому вищевказаний метод повертає об'єкт типу `Response`. Цей клас містить необхідні методи для отримання інформації у текстовому форматі, або у форматі json, а тому опрацювання цих даних буде виконуватись іншими модулями.

2.4 Модуль проведення атак

2.4.1 Збір даних про ціль

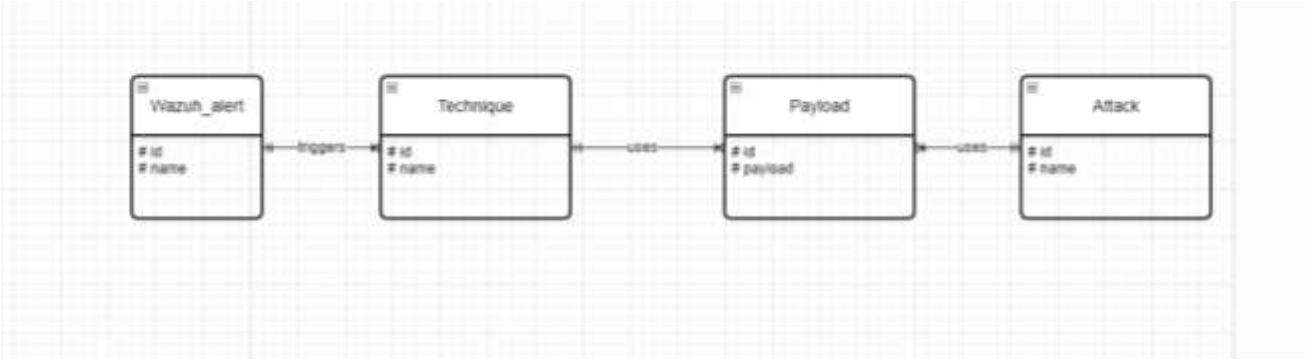
Для спрощення збору даних про можливі поля для вводу наша система використовує Swagger. Для кожної із цілей сервер містить наперед визначений swagger файл, що містить інформацію про ендпоінти та їх API. Зчитування файлів відбувається наступним чином: із файлу config.json(див. Додаток 1) зчитується інформація про цілі, та шляхи до файлів

[4] Network IDS integration - Proof of concept guide. Wazuh documentation : URL: https://documentation.wazuh.com/current/proof-of-concept-guide/integrate-network-ids-suricata.html

swagger, що містять OpenAPI специфікацію. За це відповідає клас TargetInfo(див Додаток 2). Він зчитує yaml файл та зберігає дані у вигляді python об’єкта, а також містить зручний API для отримання потрібної інформації про ендпоінти.

2.4.2 База даних атак

Для проведення атак у рамках тестування система використовує набір зазделегідь відомих технік(SQL ін’єкції, XSS ін’єкції, code-injection), та список даних(payloads), які необхідно ввести користувачу, щоб провести атаку. Кожен із цих вводів повинен викликати реакцію IDS, що відповідає типу атаки. Таким чином наша база даних містить два типи сутностей – payload(ввід) та attack(вид атаки). Структура нашої бази даних може бути описана наступною ER моделлю



Кожна атака може містити декілька різних вводів які використовують різні техніки, а тому спричиняють різні сповіщення. Реалізація виглядає наступним чином:

Класи, що відповідають відповідним сутностям у бд

```
class Technique:
    def __init__(self, uid: int, name: str):
        self.uid = uid,
        self.name = name


class WazuhAlert:
    def __init__(self, uid: int, name: str, triggered_by: Technique):
        self.uid = uid
        self.name = name
        self.triggered_by = triggered_by


class Payload:
    def __init__(self, uid: int, payload: str, technique: list[Technique]):
        self.uid = uid
        self.payload = payload
        self.technique = technique


class Attack:
    def __init__(self, uid: int, name: str, payloads: list[Payload]):
        self.uid = uid
        self.name = name
        self.payloads = payloads
```

Для реалізації з’єднання реалізовано клас AttackDbConnection, який з’єднується із бд та за допомогою sql запитів отримує інформацію про ті, чи інші сутності(див додаток 3).

2.4.3 Проведення атаки та перевірка наявності сповіщення
Використовуючи дані про ціль, зібрані за допомогою модулів, описаних у пункті 2.4.1, ми проводимо атаки, використовуючи шаблони, описані у пункті 2.4.2. Для перевірки IDS ми за допомогою API збираємо дані про сповіщення системи, і якщо вони були відсутні, то ми посилаємо власне сповіщення із інформацією про тип атаки та ввід, який був використаний при атаці. Приклад такого тестування можна побачити у додатку 4

Висновки

Нами було створено систему, що дозволяє автоматично провести тестування IDS. Отримане рішення не здатне повністю замінити повноцінний аудит безпеки, проте є корисним інструментом при його проведенні. Також цей інструмент може бути корисним і при налаштуванні самої системи, оскільки він дозволяє відразу перевірити реакцію системи на спроби атак, хоча через можливі руйнівні наслідки його роботи, варто обмежити його застосування лише системами, які не містять важливих даних.

Список використаних джерел

1. Scarfone, Karen & Mell, Peter. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS).

2. Douligeris, Christos. (2007). Network Security : Current Status and Future Directions.

3. Wazuh API REST(4.7.4). : URL: https://documentation.wazuh.com/current/user-manual/api/reference.html

4 Network IDS integration - Proof of concept guide. Wazuh documentation : URL: https://documentation.wazuh.com/current/proof-of-concept-guide/integrate-network-ids-suricata.html

5. OSSEC documentation : https://www.ossec.net/docs/

6. 5 Open Source Intrusion Detection Systems for SMBs – Towerwall : https://towerwall.com/5-open-source-intrusion-detection-systems-for-smbs/

7. Najafian, Ziaeddin & Aghazarian, Vahe & Hedayati, A.R.. (2015). Signature-Based Method and Stream Data Mining Technique Performance Evaluation for Security and Intrusion Detection in Advanced Metering Infrastructures (AMI). International Journal of Computer and Electrical Engineering. 7. 128-139. 10.17706/IJCEE.2015.V7.879.

8. A Comparison Between Signature Based and Anomaly Based Intrusion Detection Systems By: Brandon Lokesak For: COSC 356 Date: 12/4/2008.

9. Sharkov, George & Todorova, Christina. (2023). Approaching Cyber Situational Awareness Through Digital Services Availability Monitoring and Threat Intelligence: The MonSys Platform Experience. 10.1007/978-3-031-44440-1_4.

10. Intrusion Detection Systems. Rebecca Bace. Peter Mell : NIST Special Publication on Intrusion Detection Systems

11. Intrusion Detection Systems: A Survey and Taxonomy. Stefan Axelsson : Department of Computer Engineering. Chalmers University of Technology. Goteborg, Sweden

12. API Documentation & Design Tools for Teams. URL: https://swagger.io/

13. What Is Extended Detection and Response (XDR)? - Palo Alto Networks. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-extended-detection-response-XDR

Додатки