DOI: https://doi.org/10.36910/6775-2524-0560-2023-52-11
УДК 004.94:656.02
Павленко Андрій Васильович, магістр
Костючко Сергій Миколайович, к.т.н., доцент
https://orcid.org/0000-0002-1262-6268
Луцький національний технічний університет, м. Луцьк, Україна

ВИЯВЛЕННЯ ТА АНАЛІЗ НАЙВРАЗЛИВІШИХ МІСЦЬ ВЕБ-РЕСУРСІВ

Павленко А.В., Костючко С.М. Виявлення та аналіз найвразливіших місць веб-ресурсів. У даній статті виконаний опис найпопулярніших вразливостей веб-ресурсів, вказанні місця на які націлені атаки зловмисників , також пропонується засоби для виявлення та аналізу вразливостей, а саме OpenVAS та ElasticSearch. OpenVAS вибрано, тому що він має ряд істотних переваг, серед існуючих засобів для виявлення вразливостей. Виконано встановлення та показано технологію роботи даних систем.
Ключові слова: вразливість, веб-ресурс, виявлення та моніторинг, загроза, OpenVAS та ElasticSearch.

Pavlenko A.V., Kostiuchko S.M. Detection and analysis of the most vulnerable places of web resources. This article provides a description of the most popular vulnerabilities of web resources, indicates the places targeted by attackers' attacks, and also offers tools for detecting and analyzing vulnerabilities, namely OpenVAS and ElasticSearch. OpenVAS was chosen because it has a number of significant advantages among the existing tools for detecting vulnerabilities. The installation was performed and the technology of these systems was shown.
Keywords: vulnerability, web resource, detection and monitoring, threat, OpenVAS and ElasticSearch.

Постановка проблеми у загальному вигляді та її зв'язок з важливими науковими і практичними завданнями. Веб-ресурси є ключовими елементами в сучасному світі, оскільки багато інформації та послуг доступно через Інтернет. Однак, ці ресурси можуть піддаватися атакам, які використовують вразливості в їхній структурі або програмному забезпеченні [2]. Тому потрібно бути обізнаним про ці загрози, оскільки це сприяє розробці ефективних методів захисту.

Збільшення кількості загроз інформаційній безпеці сприяє впровадженню на підприємствах систем, які здатні виявити та усунути дану проблему. Проблеми необхідно виявити та усунути в найшвидший термін задля усунення загроз, виникнення додаткових капіталовкладень та витоку даних.

Не менш важливим питанням є управління ризиками. Управління ризиками в інформаційних технологіях виявляється ключовою проблемою в сучасному світі, оскільки зростають інциденти безпеки та фінансові збитки, пов'язані з ними.

Це все стосується до сучасних проблем кібербезпеки, оскільки потрібно максимально швидко виявити вразливість, використовуючи при цьому різні методи, а саме алгебраїчні та з використанням нейронних мереж. Завдяки цьому у нас буде можливість зосередитися на ще двох основних проблемах, таких як виявлення вторгнень в реальному часі та виявлення вразливостей у програмних та апаратних системах при цьому проводячи оцінку їх стійкості до вторгнень [5].

Аналіз останніх досліджень, у яких започатковано вирішення проблеми. Важливо встановити та проаналізувати найчастіші місця, на які зловмисники спрямовують свої атаки на веб-сайти. Це можуть бути, наприклад, слабкі паролі, незахищені точки введення даних, незахищені API або розкриті конфіденційні дані [3].

Останні дослідження спрямовані на розробку нових підходів до управління ризиками, які базуються на принципах когнітивних технологій та поведінкових наук. Це може включати в себе застосування штучного інтелекту, машинного навчання, аналізу даних та інших сучасних методів для виявлення, аналізу і мінімізації ризиків в інформаційних технологіях.

Одним з важливих завдань є оцінка ризиків з максимальною точністю, щоб надати пріоритет високотяжким ризикам перед низькотяжкими. Це дозволяє ефективніше спрямовувати ресурси на управління найбільш значущими ризиками [4].

Аналіз атак на системи обробки державних інформаційних ресурсів: Потрібно досліджувати різні типи атак, які можуть бути спрямовані на державні інформаційні ресурси. Проводити аналіз сучасних баз даних, що містять детальний опис вразливостей та атак, а також класифікацію атак та їх параметри. Описувати стратегії здійснення атак та основні фази та особливості проведення атак. Розробку методів виявлення атак та визначення вразливостей: Також потрібно розробити методи виявлення атак та визначення вразливостей державних інформаційних ресурсів, що обробляються засобами інформаційно-телекомунікаційних систем. Вимоги до цих методів враховують вимоги до

методів виявлення атак, параметрів даних та характеристичних особливостей сучасних систем виявлення атак [6].

Опис вразливостей веб-ресурсів

Існує велика кількість вразливостей, кожна з яких має свої характеристики. Вразливість — це слабке місце (недолік) у системі, яким може скористатись зловмисник.

Owasp надає список таких найпопулярніших вразливостей :

Порушення контролю доступу – ця вразливість дозволяє отримати доступ до облікового запису адміністратора [11].

Незахищеність критичних даних – виникає при неналежному зберіганні конфіденційної інформації, такої як номери кредитних карт, медичні дані, паспортні дані, дані майнової приналежності та інші особисті дані. Ці дані є цінними та чутливими і мають велику вартість для власників, а також можуть бути використані зловмисниками для шахрайства, крадіжки особистої інформації, ідентифікаційних злочинів та інших негативних цілей. Незахищеність критичних даних може мати серйозні наслідки для постраждалих осіб, таких як фінансові втрати, втрата репутації, порушення приватності, недостовірність медичних записів, можливість крадіжки особистості та інші негативні наслідки. Крім того, це може також мати вплив на організації та установи, які відповідають за збереження цих даних, шляхом порушення довіри клієнтів та втрати репутації [12].

Внутрішні об'єкти XML. Ця вразливість може бути використана зловмисниками для здійснення атак, таких як атаки на витік даних або впровадження шкідливого коду. Зловмисник може використовувати введення XML, що містить посилання на зовнішні об'єкти, як спосіб отримання несанкціонованого доступу до ресурсів системи або виконання коду, що може завдати шкоди системі. Для вирішення цієї проблеми необхідно використовувати безпечні методи обробки XML. Це може включати належне налаштування аналізаторів XML для обмеження доступу до зовнішніх об'єктів, використання механізмів аутентифікації та авторизації для контролю доступу до ресурсів, а також проведення перевірки та валідації введеного XML для виявлення та блокування потенційно шкідливих або небезпечних елементів [12].

Неправильна конфігурація безпеки – це проблема виявляється тоді, коли система чи програмне забезпечення поставлені в експлуатацію зі стандартними налаштуваннями, які не враховують специфіку середовища чи потенційні загрози, вони стають легким мішенню для зловмисників. Налаштування по замовчуванню часто не забезпечують достатнього рівня безпеки, оскільки вони призначені для загального використання та зручності. Зловмисники знають про це та використовують цю слабкість для здійснення атак. Наприклад, стандартні паролі, відкриті порти, недостатня перевірка автентичності, відсутність захисту від введення даних та інші подібні проблеми можуть виникати через неправильну конфігурацію безпеки. Для попередження цієї проблеми необхідно враховувати специфіку системи та середовища, встановлювати налаштування безпеки, які відповідають найкращим практикам та стандартам безпеки, а також проводити аудит та перевірку системи на предмет виявлення можливих проблем і вразливостей [12].

Використання компонентів з відомими вразливостями – виникає зазвичай, у випадку використання різних не перевірених компонентів в свій код. Компоненти з відомими вразливостями - це програмні бібліотеки, модулі або фреймворки, які мають відомі проблеми безпеки, такі як уразливості у коді, недоліки в реалізації або неправильну обробку даних. Ці вразливості можуть бути відомі публічно, оскільки багато компонентів з відкритим вихідним кодом пройшли процес аудиту безпеки та їх вразливості документовані. Зловмисники використовують ці вразливості для здійснення атак на систему, отримання несанкціонованого доступу до даних, виконання віддаленого коду або впровадження шкідливого програмного забезпечення. Однак, важливо зауважити, що використання компонентів з відкритим вихідним кодом само по собі не є проблемою. Багато з цих компонентів мають активну спільноту розробників, яка виправляє виявлені вразливості та випускає оновлення. Проблема виникає, коли веб-розробники не враховують ці вразливості та не оновлюють використовувані компоненти до останньої безпечної версії [13].

Збої в автентифікації та ідентифікації – це вразливість, що входить до розряду проблем з ідентифікацією, автентифікацією. Виникає зазвичай коли система не здійснює належну перевірку облікових даних, це може дозволити зловмисникам використати неправильні облікові дані для доступу до системи. А також коли механізми автентифікації не налаштовані належним чином, це може відкрити можливості для зламу або обходу механізмів автентифікації.

· Збій криптографії – це вразливість, використовується зловмисниками, коли криптографія налаштована з наявністю помилок або взагалі її відсутності та дозволяє їм отримати доступ до важливих даних в системі. Також часто виникає при помилковому використані криптографічних протоколів, тобто коли вони не використовуються належним чином, це може створити можливості для атаки на криптографію і отримання несанкціонованого доступу до захищених даних.

· Помилки цілісності програмного забезпечення і даних – це вразливість, що зосереджується на проблемах пов'язаних з оновленнями програмного забезпечення, важливими даними та конвеєрами CI/CD (постійна інтеграція/постійна доставка). Виникає під час оновлення програмного забезпечення, а саме коли виконується неправильне оновлення програмного забезпечення, що може призвести до некоректної роботи системи, порушити цілісність даних або відкрити можливості для атак [9].

· Збій в журналі та моніторингу безпеки – це вразливість, що відноситься до проблеми, коли система, відповідальна за збір, реєстрацію та аналіз подій безпеки, не працює належним чином або взагалі перестає функціонувати. Це може статися через різні причини, такі як помилкова конфігурація, технічні проблеми, відмова обладнання або програмного забезпечення. Деякі стандарти безпеки вимагають ведення журналів подій та моніторингу безпеки з метою дотримання вимог відповідності та проведення аудиту. Якщо система моніторингу не працює належним чином, це може призвести до порушення вимог стандартів безпеки та небажаних наслідків у сфері регуляторного відповідності [10].

· Кросс-сайт скриптінг (Cross-Site Scripting, XSS) є типом вразливості, який дозволяє зловмисникам впроваджувати та виконувати зловмисний скрипт на веб-сторінках, які переглядають користувачі. Ця вразливість виникає, коли вхідні дані, які не були належним чином перевірені або екрановані, вставляються безпосередньо на веб-сторінку і виконуються в браузері користувача [7,8].

Найбільш вразливі місця веб-ресурсів

До найбільш вразливих та найбажаніших місць веб–ресурсів до яких зловмисники намагаються отримати доступ, належать дані (інформація) – адже, більшість атак, що проводяться мають за мету отримання конфіденційних даних [1].

Найчастіше причиною появи вразливостей та взломів є написаний програмістами код додатку. Написання безпечного коду вимагає уважності, досвіду та дотримання низки кращих практик безпеки програмування. Деякі з найпоширеніших причин вразливостей, пов'язаних з кодом програмного забезпечення, включають:

Недостатня перевірка введення: Недостатня або неправильна перевірка введення даних може призвести до вразливостей, таких як XSS або SQL-ін'єкції. Програмісти повинні бути обережними та правильно перевіряти, очищувати та валідувати вхідні дані, щоб запобігти вразливостям.

Недостатня аутентифікація та авторизація: Слабкі механізми аутентифікації (перевірка ідентифікації користувача) та авторизації (контроль доступу до ресурсів) можуть дозволити зловмисникам отримати несанкціонований доступ до системи або функцій, які їм не належать. Програмісти повинні ретельно розробляти і реалізовувати механізми аутентифікації та авторизації.

Вразливості у криптографії – застосовується при неправильному використанні алгоритмів шифрування та інших криптографічних функцій може призвести до вразливостей. Наприклад, використання слабких алгоритмів шифрування, недостатня довжина ключів або неналежне зберігання криптографічних матеріалів можуть зробити систему вразливою.

Недостатня обробка винятків та помилок – виникає при неправильній обробці винятків та помилок, що згодом може розкрити конфіденційну інформацію або призвести до вразливостей. Програмісти повинні належним чином обробляти помилки та винятки, щоб уникнути витоку інформації та забезпечити стабільність системи [14].

Найчастіше атаки провадяться на:

незашифровані паролі – зловмисник може отримати доступ до вашого облікового запису, якщо сайт неналежно виконує шифрування паролів. Також потрібно звертати увагу на паролі їхню довжину, використання спеціальних символів;

адміністративний доступ - зловмисники можуть намагатися отримати доступ до адміністративних функцій веб-сайту, таких як панель адміністратора або база даних, для виконання шкідливих дій на веб-сайті або для отримання доступу до конфіденційної інформації;

недостатня обробка введення – це вразливість, яка дозволяє зловмисникам вводити шкідливий код на веб-сайті. Наприклад, зловмисники можуть використовувати вразливості, щоб вбудувати JavaScript-код у веб-сторінки або виконувати інші шкідливі дії;

зловмисники можуть намагатися отримати доступ до сесій користувачів для виконання шкідливих дій на їх ім'я або для викрадення конфіденційної інформації. Не захищені сесії можуть бути скомпрометовані, що дозволяє зловмисникам отримувати доступ до конфіденційної інформації.

можливість завантажувати користувачам файли – веб-сайт дозволяє користувачам завантажувати файли, це може стати вразливістю. Зловмисники можуть завантажувати шкідливі файли, які можуть виконуватися на сервері, що призводить до порушення безпеки веб-сайту. Крім того, якщо файли не захищені належним чином, зловмисники можуть мати доступ до них, що дозволяє їм отримати конфіденційну інформацію, яка міститься в цих файлах;

зловмисники можуть намагатися перехопити дані, що передаються між користувачем та веб-сайтом, такі як логіни, паролі та інші конфіденційні дані. Це може бути здійснено через незахищені з'єднання, такі як HTTP, або за допомогою шкіливих додатків, вірусів або шкідливих програм, які встановлюються на комп'ютери користувачів.

недостатній контроль доступу – це вразливість, яка дозволяє зловмисникам отримувати доступ до чутливої інформації або функцій, до яких вони не мають повинен бути доступ. Це може статися, якщо веб-сайт не використовує належні методи автентифікації та авторизації користувачів.

відкриті директорії системних файлів – атаки використовуються для отримання даних, які зловмисник може здійснення атаки та для одержання доступу до облікових даних сайту. Також відкриті системні файли можуть використовуватися для одержання доступу для цих виконання цих файлів, що можуть взаємодіяти з файловою системою чи базами даних;

Бази даних є дуже важливою складовою веб-сайту. Зловмисники можуть використовувати різні методи атак на базу даних, такі як SQL-ін'єкції, для отримання доступу до конфіденційної інформації, внесення змін у дані, або видалення даних з бази.

Куки – це файли, які зберігаються на комп'ютері користувача та містять інформацію про відвідування веб-сайту. Якщо веб-сайт не захищений належним чином, зловмисники можуть отримати доступ до цієї інформації. За допомогою куків зловмисники можуть отримати доступ до облікових записів користувачів, а також використовувати цю інформацію для атак на інші веб-сайти, які використовують ті ж самі куки.Файли cookie можуть містити конфіденційну інформацію про користувачів, таку як імена користувачів та паролі. Зловмисники можуть використовувати вразливості в реалізації cookie, щоб отримати доступ до цієї інформації.

недостатній захист від переповнення буферу пам'яті – це вразливість, яка дозволяє зловмисникам переповнити буфер, що може призвести до виконання шкідливого коду. Наприклад, зловмисники можуть використовувати цю вразливість, щоб запустити вірус або шкідливу програму на сервері;

програмне забезпечення – може використовуватися зловмисниками у випадку використання ПЗ з не офіційних та неперевірених джерел;

Отримання доступу до журналу подій відбувається для того щоб отримати можливість приховати можливу атаку. Недостатній моніторинг та керування журналами зазвичай призводить до не можливості відстеження моделей поведінки користувачів, що дозволяє зловмисникам скомпрометувати систему.;

SSL сертифікат – наявність даного сертифікату свідчить про безпеку сайту. Тому зловмисники можуть спробувати підробити його, або намагатися отримати доступ до сайту без даного сертифікату;

Форми введення даних – це такі компоненти як форми для входу в систему або форми для заповнення замовлень, часто стають мішенню атак XSS або SQL-ін'єкцій. Зловмисники можуть вставляти шкідливий код в поля введення даних, щоб отримати доступ до захищених даних або виконати шкідливі дії на веб-сайті [15].

Elastic Stack

- ElasticSearch – це один із компонентів, які входять до стеку ELK (складові частини стеку показано на рис.1). ELK Stack – це система, яка складається із таких компонентів: ElasticSearch, Logstash, Kibana, Beats. ElasticSearch є фактично ядром цієї системи, він здійснює поєднання в собі кількох функцій:
- бази даних;

- аналітичної;
- пошукової системи.

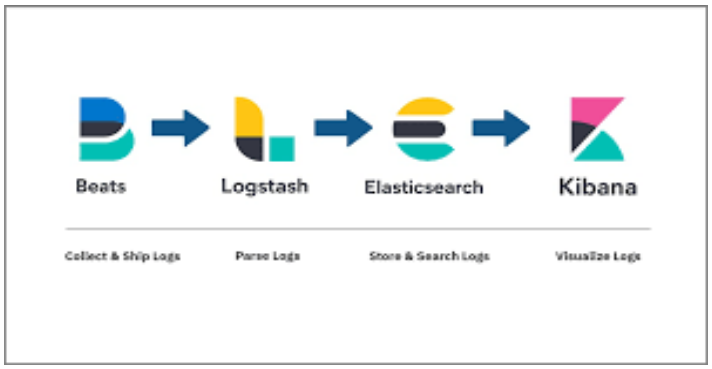

Рисунок 1 – Складові частини ELK Stack

ElasticSearch – це спеціалізоване програмне забезпечення для повнотекстового пошуку. Перевагами даного програмного рішення є легке масштабування і реплікація. Зазвичай використовується в високонавантажених проектах в великими обсягами даних з чим він відмінно справляється. Є не реляційним сховищем (NoSQL) документів в форматі JSON. Вона використовує технологію Lucene. Для роботи та нормального функціонування застосовується Java Virtual Machine, що означає що для роботи потрібно багато ресурсів процесора та оперативної пам'яті. Всі повідомлення, які надходять індексуються як "документ", так само, як і реляційних базах даних. Для роботи з базою даних система використовує JSON запити з допомогою REST API. Це застосовується для видачі індексованих документів та статистика у формі запит – відповідь. Для візуалізації цих даних використовується Kibana.

Logstash – це важлива складова стеку ELK на стороні сервера. Основним застосуванням, якого є здійснення обробки даних. Він здійснює автоматичний збір та обробку даних з кількох джерел одночасно, після цього він надсилає дані в ElasticSearch. Здійснює обробку логів подій із різних джерел. За допомогою цієї утиліти можна виділяти поля та їх значення, можна здійснювати виконання налаштування фільтрації та редагування даних. Налаштування можна виконати через конфігураційні файли. Типова конфігурація складається з кількох вхідних потоків інформації (input) – тут ми вказуємо на, який порт будуть відправлятися дані, містить фільтри для читання цієї інформації (filter), а вихідних даних (output). Приклад конфігураційного файлу показано на рисунку 2.

```
input {
  tcp {
    type => "habr"
    port => "11111"
  }
}
filter {
  mutate {
    type => "habr"
    add_field => [ "habra_field", "Hello Habr" ]
  }
}
output {
  stdout {
    type => "habr"
    message => "%{habra_field}: %{@message}"
  }
}
```

Рисунок 2 –Конфігураційний файл Logstash

Kibana використовується для візуального відображення звітів користувачами у вигляді діаграм. Ще одна із функцій цього елемента є можливість здійснювати через нього адміністрування бази даних. Зазвичай використовується для аналізу журналів та моніторинг додатків та запущених

процесів. Даний інструмент дозволяє легко створювати графіки, кругові діаграми, гістограми та використовується в цілому як інструмент візуалізації ElasticSearch. На рисунку 3 показано звіт в Kibana.

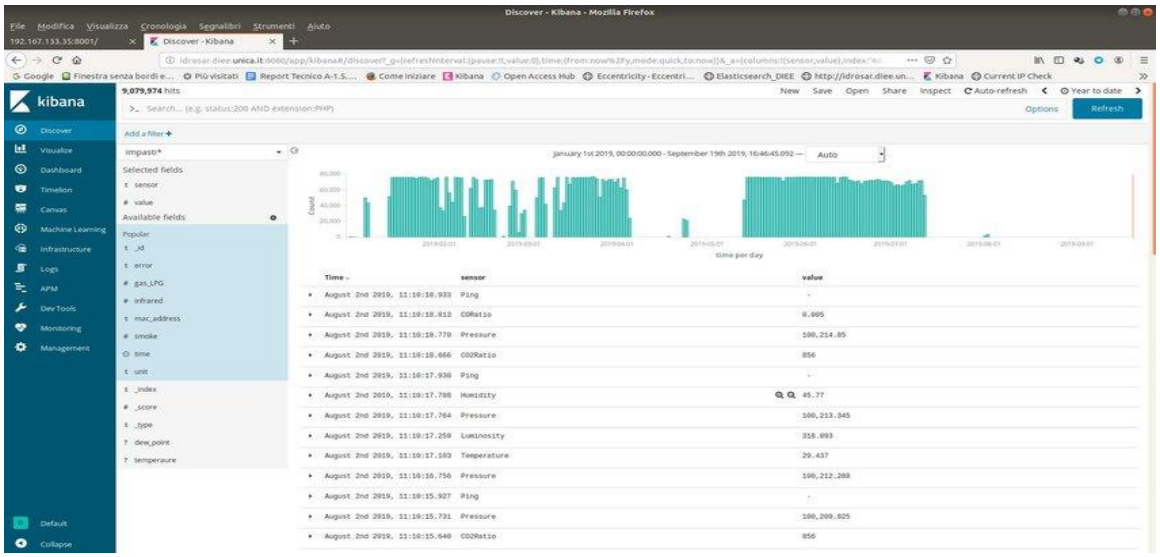

Рисунок 3 –Створений звіт Kibana

Beats – програмне забезпечення з вільним програмним кодом застосовується, як платформа для доставки даних, встановлюється на стороні клієнта. Для роботи використовує бібліотеку libbet, що застосовується для збору даних та налаштування введення, а також надає API для того, щоб мати змогу передати дані від джерела. Встановлюється Beats на пристрої, які розміщуються на некластерних вузлах. Звіт з графіком та діаграмою показаний на рисунку 4 [16].

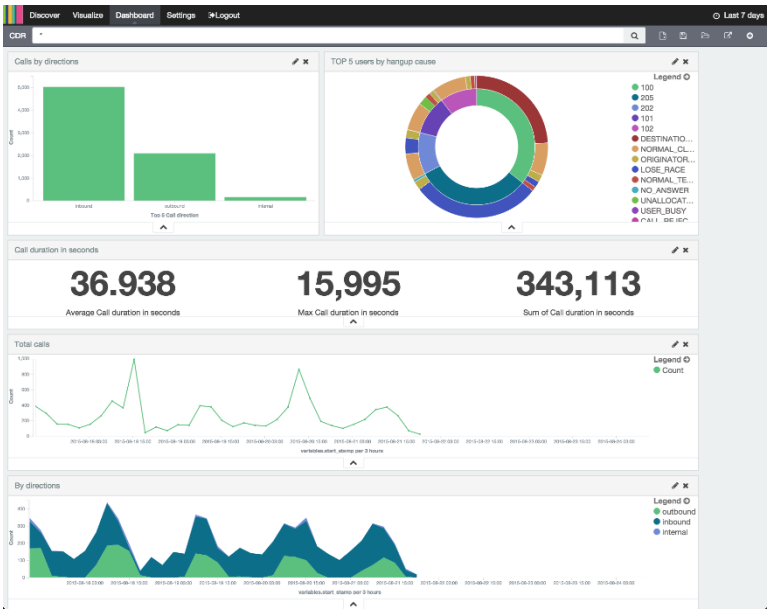

Рисунок 4 –Звіт з графіком та діаграмою

Сканер OpenVAS

OpenVAS – це програма, що складається з декількох сервісів та утиліт, основним завданням, якого є проведення сканування вузлів мережі, веб-сайтів на наявність вразливостей та управління ними.

Сканер проводить пошук IP-адреси, після цього завдяки відкритим портам відбувається пошук незахищених портів та служб. Сканер під час проведення сканування здатний виявити неправильні конфігурації, та 'дірки' в системі безпеки в цілому.

Сканер містить колекцію NVT тестів безпеки приблизна кількість, яких 30000. Для перевірки наявності вразливостей використовуються бази автоматизованого управління CVE та OpenSCAP завдяки цим ресурсам можна переглянути опис проблеми.

OpenVAS побудований за допомогою клієнт-серверної схеми складається він з декількох утиліт та сервісів, кожен з яких має свої функції. Серверна частина встановлюється тільки на Linux-подібні системи. Клієнтська частина може бути встановлена на будь-яку ОС.

Ядро системи встановлюється на порту 9391 ним виступає OpenVAS Scanner. Даний сервіс виконує сканування на предмет наявності вразливостей та неправильних конфігурацій. Для виконання сканування сервіс використовує ресурси: nmap, pnscan, ike-scan та інші [17].

OpenVAS Manager ще одна важлива складова частина. Вона відповідає за збір та аналіз даних після сканування, розміщується на порту 9390. Він використовується для того, щоб здійснити контроль над вразливостями та для виявлення хибних спрацювань, в даному сервісі можна налаштувати сканування за розкладом. Для здійснення управління сканером використовується протокол OTP (OpenVAS Transfer Protocol). Налаштування та інформація, яку зібрали зберігається в базі даних SQLite [18].

OpenVAS Administrator – сервіс, який відповідає за керування обліковими записами та по суті управління всіма керівними функціями системи. Розміщується він на порту 9393.

Консоль управління Greenbone Security Assistant показана на рисунку 5.

Інтерфейс в даного сканера зрозумілий, хоча досить складний в встановлені та налаштуванні. Меню сканера складається з семи пунктів, воно розташовується з самого верху. Поле 'Quick start: Immediately scan an IP address' дозволяє почати сканування вузла або ресурсу одразу не задаючи додаткових параметрів, потрібно буде ввести тільки IP- адресу або ім'я ресурсу.

Рисунок 5 –Консоль управління Greenbone Security Assistant

Для вибору детального сканування потрібно створити нову задачу, де потрібно буде вказати параметри сканування. Створення нової задачі показане на рисунку 6.

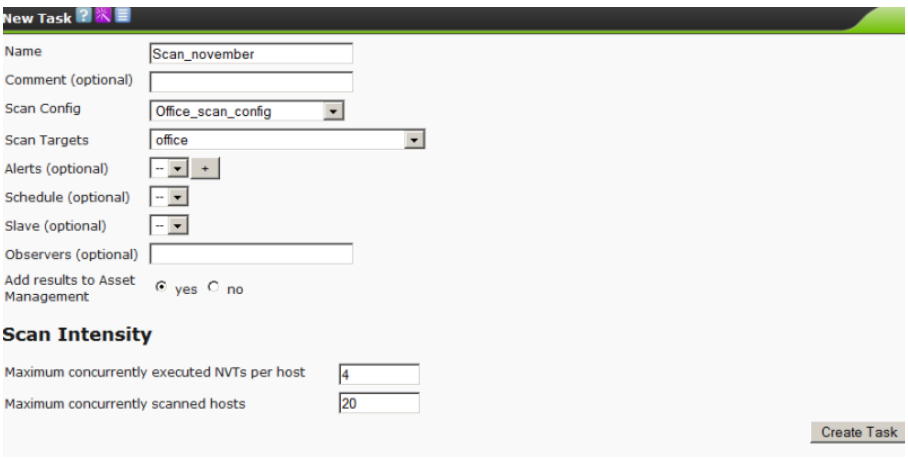

Рисунок 6 – Створення нової задачі

Після цього можна почати процес сканування, зазвичай він дуже тривалий (процес сканування на показаний на рисунку 7).

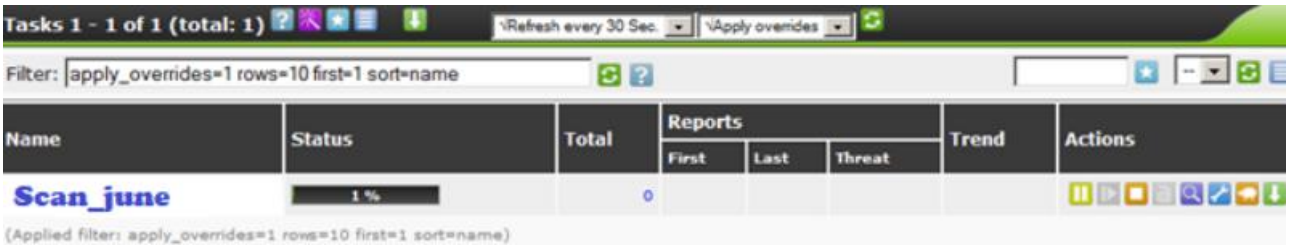

Рисунок 7 –Процес сканування

Після завершення сканування можна натиснути на лупу для детального опису проблеми. На рисунку 8 показані всі знайдені проблеми.

	High	Medium	Low	Log	False Pos	Total	Run Alert	Download
Full report:	43	174	837	8058	0	9112		PDF
All filtered results:	43	174	0	0	0	217		PDF
Filtered results 1 - 100:	29	71	0	0	0	100		PDF

Рисунок 8 –Таблиця зі знайденими проблемами

Висновки

В даній роботі проведено опис найпоширеніших вразливостей веб-ресурсів та виконана ідентифікація місць, на які зловмисники найчастіше спрямовують свої атаки. Аналізуючи ці вразливості та місця атак, стає очевидним, що безпека веб-ресурсів є критично важливою і повинна бути врахована при розробці та підтримці веб-додатків.

Встановлення та застосування сканеру OpenVAS та Elastic search демонструє, яким чином можна використовувати сучасні інструменти для виявлення вразливостей та аналізу безпеки веб-ресурсів. Ці інструменти надають зручність та ефективність при виявленні проблем безпеки та допомагають забезпечити високий рівень захисту веб-додатків.

Результати тестування та демонстрація використання цих засобів підтверджують їхню ефективність та важливість для виявлення вразливостей. При використанні правильно налаштованих інструментів, таких як OpenVAS та Elastic search, можна забезпечити більш повну інформацію про вразливості та прийняти необхідні заходи для запобігання атакам та зловживанням.

Список бібліографічного опису

1. Gaolong W., Yongzhen L. Design and implementation of a web application vulnerability detection system. *2022 International Symposium on Advances in Informatics, Electronics and Education (ISAIEE)*, Frankfurt, Germany, 17–19 December 2022. 2022. URL: https://doi.org/10.1109/isaiee57420.2022.00089 (date of access: 05.06.2023).
2. Source Code Vulnerability Detection Using Vulnerability Dependency Representation Graph / H. Yang et al. *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Wuhan, China, 9–11 December 2022. 2022. URL: https://doi.org/10.1109/trustcom56396.2022.00070 (date of access: 04.06.2023).
3. Deep Analysis of Attacks and Vulnerabilities of Web Security / N. Chillur et al. *Futuristic Trends in Networks and Computing Technologies*. Singapore, 2022. P. 1087–1097. URL: https://doi.org/10.1007/978-981-19-5037-7_78 (date of access: 04.06.2023).
4. Kaur G., Lashkari A. H. Information Technology Risk Management. *Advances in Cybersecurity Management*. Cham, 2021. P. 269–287. URL: https://doi.org/10.1007/978-3-030-71381-2_13 (date of access: 04.06.2023).
5. Летичевський О.О. Сучасні наукові проблеми кібербезпеки. *Вісник НАН України.* 2023. № 2. С. 12—20. https://doi.org/10.15407/visn2023.02.012 (дата звернення: 04.06.2023)
6. Сальник С. В., Сторчак А. С., Крамський А. Є. Аналіз вразливостей та атак на державні інформаційні ресурси, що обробляються в інформаційно-телекомунікаційних системах. *Системи обробки інформації*. 2019. № 2(157). С. 121–128. URL: https://doi.org/10.30748/soi.2019.157.17 (дата звернення: 04.06.2023).
7. OWASP Top Ten | OWASP Foundation. *OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation*. URL: https://owasp.org/www-project-top-ten/ (date of access: 08.06.2023).

8. Cross Site Scripting (XSS) | OWASP Foundation. *OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation*. URL: https://owasp.org/www-community/attacks/xss/ (date of access: 08.06.2023).
9. ПОМИЛКИ В ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ. *Stud*. URL: https://stud.com.ua/128164/informatika/pomilki_programnomu_zabezpechenni (дата звернення: 08.06.2023).
10. A09 Security Logging and Monitoring Failures - OWASP Top 10:2021. *OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation*. URL: https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/ (date of access: 08.06.2023).
11. Чому контроль доступу важливий – приклади та рішення для контролю доступу. *Hideez*. URL: https://hideez.com/uk-ua/blogs/news/access-control (дата звернення: 08.06.2023).
12. Що таке OWASP Top-10?. *UKEY Web application firewall*. URL: https://ukeywaf.com/baza/shho-take-owasp-top-10/ (дата звернення: 08.06.2023).
13. Putting the Sec in DevSecOps: Simplify Application Security | GuardRails. *GuardRails*. URL: https://www.guardrails.io/blog/using-components-with-known-vulnerabilities-a-guide-to-secure-software-development/ (date of access: 08.06.2023).
14. S. Disawal and U. Suman, "An Analysis and Classification of Vulnerabilities in Web-Based Application Development," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2021, pp. 782-785.
15. Як захистити сайт від злому | 10 способів захисту веб-сайту *Fondy. Онлайн платежи для вашего бизнеса*. URL:https://fondy.ua/uk/blog/how-to-protect-website/ (дата звернення: 09.06.2023).
16. What is Elasticsearch?. *Elastic*. URL:https://www.elastic.co/what-is/elasticsearch (date of access: 12.06.2023).
17. Nessus vs OpenVAS: Which is Better? A Head-to-Head Comparison. *Comparitech*. URL: https://www.comparitech.com/net-admin/nessus-vs-openvas/ (date of access: 14.06.2023).
18. OpenVas: framework per l'analisi di vulnerabilità. *HTML.it*. URL:https://www.html.it/pag/71804/openvas-framework-per-lanalisi-di-vulnerabilita/ (date of access: 14.06.2023).
19. V. Satsyk, O. Mekush, N. Lishchyna, N. Khrystynets, L. Gumeniuk and L. Korobchuk, "Soil Analysis Software Tool for Smart Control of Agronomic Data," 2022 12th International Conference on Advanced Computer Information Technologies (ACIT), Ruzomberok, Slovakia, 2022, pp. 364-368, doi: 10.1109/ACIT54803.2022.9913133.
20. P. Savaryn, V. Strekha, M. Brych, L. Brych, V. Kabak and M. Polishchuk, "The Original Method of Controlling a Computer Using Distance Sensors," 2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 2022, pp. 683-688, doi: 10.1109/TCSET55632.2022.9767011.

Rerences

1. Cross Site Scripting (XSS) | OWASP Foundation. *OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation*. URL: https://owasp.org/www-community/attacks/xss/ (date of access: 08.06.2023).
2. Putting the Sec in DevSecOps: Simplify Application Security | GuardRails. *GuardRails*. URL: https://www.guardrails.io/blog/using-components-with-known-vulnerabilities-a-guide-to-secure-software-development/ (date of access: 08.06.2023).
3. S. Disawal and U. Suman, "An Analysis and Classification of Vulnerabilities in Web-Based Application Development," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2021, pp. 782-785.
4. What is Elasticsearch?. *Elastic*. URL:https://www.elastic.co/what-is/elasticsearch (date of access: 12.06.2023).
5. Nessus vs OpenVAS: Which is Better? A Head-to-Head Comparison. *Comparitech*. URL: https://www.comparitech.com/net-admin/nessus-vs-openvas/ (date of access: 14.06.2023).
6. OpenVas: framework per l'analisi di vulnerabilità. *HTML.it*. URL:https://www.html.it/pag/71804/openvas-framework-per-lanalisi-di-vulnerabilita/ (date of access: 14.06.2023).
7. V. Satsyk, O. Mekush, N. Lishchyna, N. Khrystynets, L. Gumeniuk and L. Korobchuk, "Soil Analysis Software Tool for Smart Control of Agronomic Data," 2022 12th International Conference on Advanced Computer Information Technologies (ACIT), Ruzomberok, Slovakia, 2022, pp. 364-368, doi: 10.1109/ACIT54803.2022.9913133.
8. P. Savaryn, V. Strekha, M. Brych, L. Brych, V. Kabak and M. Polishchuk, "The Original Method of Controlling a Computer Using Distance Sensors," 2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 2022, pp. 683-688, doi: 10.1109/TCSET55632.2022.9767011.