ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

Пояснювальна записка

до магістерської роботи
на тему:

«ТЕХНОЛОГІЯ УПРАВЛІННЯ ВРАЗЛИВОСТЯМИ ВЕБ-ДОДАТКУ СУЧАСНОГО ПІДПРИЄМСТВА НА БАЗІ РІШЕННЯ BURP SUITE»

Виконав студент 6 курсу, групи БСДМ-62
спеціальності 125 Кібербезпека
освітньо-професійної програми «Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)
Смолєв Є.С.
(прізвище та ініціали)
Керівник Гахов С.О.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)
Нормоконтролер Чумак Н.С.
(прізвище та ініціали)

КИЇВ – 2022

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІБ – інформаційна безпека

ІТ – інформаційні технології

ОС – Операційна система

ОТ – Операційні технології

API – Application Programming Interface

CD – Continuous Integration

CDN – Contect Delivery Network

CI – Continuous Delivery

CSS – Cascading Style Sheets

CVE – Common Vulnerabilities and Exposures

CVSS – Common Vulnerability Scoring System

CWE – Common Weakness Enumeration

DAST – Dynamic application security testing

DNS – Domain Name System

HTML – Hypertext Markup Language

HTTP – Hypertext Transfer Protocol

IoT – Internet of things

OAST – Out-of-band application security testing

RASP – Runtime Application Self-Protection

SAST – Static application security testing

SE – Social engineering

UI – User Interface

URL – Uniform Resource Locator

XDR – Extended detection and response

ВСТУП

Більшість компаній сьогодні працюють в режимі онлайн і працюють з різними веб-активами, такими як веб-додатки, веб-сайти, веб-сервіси, API та хмарні програмні системи (SAAS). У своєму IT-середовищі вони спілкуються з різними програмними системами, внутрішніми та зовнішніми, таким чином розкриваючи свою функціональність через декілька інтерфейсів.

Дані про атаки зі звіту NTT Global Threat Intelligence Report (GTIR) за 2020 рік показують, що більше половини (55%) усіх атак у 2019 році були комбінацією атак на веб-додатки та конкретні програми, порівняно з 32% роком раніше. Атаки на системи керування вмістом, включаючи WordPress , Joomla, Drupal і NoneCMS, становили близько 20% усіх кібератак . Крім того, понад 28% атак були спрямовані на технологічні платформи, які підтримують веб-сайти, такі як ColdFusion і Apache Struts [1].

Вищесказане визначає актуальність роботи. Саме тому технології управління вразливостями веб-додатків сучасного підприємства такі важливі і можуть полегшити роботу для багатьох спеціалісті з кібербезпеки та додати нові можливості для подальшого виявлення та усунення вразливих місць.

*Об'єкт дослідженн*я: процес забезпечення управління вразливостями веб-додатку сучасного підприємства.

Предмет дослідження: технологія управління вразливостями веб-додатку сучасного підприємства.

Мета роботи: розробити алгоритм для забезпечення процесу управління вразливостями веб-додатку.

Для досягнення цієї мети були поставлені такі наукові завдання:

проаналізувати архітектуру сучасних веб-додатків;

проаналізувати проблему управління вразливостями веб-додатків;

дослідити зміст управління вразливостями веб-додатків;

проаналізувати існуючі методи та засоби управління вразливостями веб-додатка;

проаналізувати компоненти та архітектуру програмного комплексу Burp Suite;

дослідити можливості застосування програмного комплексу Burp Suite з метою забезпечення управління вразливостями веб-додатка сучасного підприємства.

Практичне значення одержаних результатів: розроблено алгоритм управління вразливостями веб-додатка, запропоновано порядок застосування технології управління вразливостями веб-додатка з використанням програмного комплекса Burp Suite, а також розроблено рекомендації фахівцям з кібербезпеки щодо застосування технології управління вразливостями веб-додатків сучасного підприємства.

1 АНАЛІЗ ПРОБЛЕМИ УПРАВЛІННЯ ВРАЗЛИВОСТЯМИ ВЕБ-ДОДАТКУ СУЧАСНОГО ПІДПРИЄМСТВА

1.1. Призначення, структура, функції та умови функціонування веб-додатку сучасного підприємства

Незалежно від того, чи має компанія локальне, національне чи глобальне охоплення, легка та швидка взаємодія з співробітниками та клієнтами стала як ніколи необхідною у світі бізнесу. Компанії все частіше використовують веб-додатки для полегшення спілкування та обміну інформацією.

Веб-додаток – це програма, яка використовує веб-технологію для виконання конкретних завдань. Віддалені веб-сервери розміщують веб-додатки та зберігають відповідну інформацію з численних підключених комп'ютерів Ларрі Волл розробив один з перших веб-додатків у 1987 році. З тих пір веб-додатки зазнали багатьох розробок, і поточні є набагато складнішими за своїми функціями та використанням, ніж попередні спрощені. Вони також набагато більш незамінні для особистого та ділового використання [2].

Веб-додаток – це клієнт-серверна програма. Це означає, що він має клієнтську і серверну сторону. Термін клієнт тут відноситься до програми, яку особа використовує для запуску додатку. Це частина клієнт-серверного середовища, де багато комп'ютерів обмінюються інформацією. Наприклад, у випадку з базою даних клієнтом є програма, за допомогою якої користувач вводить дані. Сервер – це програма, яка зберігає інформацію. Підприємствам необхідно обмінюватися інформацією та укладати угоди зі своїми цільовими клієнтами. Інтернет може бути відмінним і недорогим каналом для цієї мети, якщо є спосіб отримувати та зберігати всі необхідні дані та показувати результати користувачам. Завдяки веб-додаткам користувачі можуть взаємодіяти з бізнесом, використовуючи кошики для покупок або системи керування вмістом [3].

На відміну від веб-сайту (набір статичних сторінок з текстом, зображеннями, аудіо та відео), веб-додаток дозволяє користувачам взаємодіяти з його елементами та маніпулювати вмістом.

Взаємодія користувача з сучасним веб-додатком зазвичай складається з наступних кроків:

1. Клієнт (браузер) зв'язується з DNS-сервером, який надає IP-адресу.

2. Потім браузер надсилає запит для отримання веб-документів (JSON, HTML та інші) на веб-сервер.

3. Веб-сервер пересилає запити в логіку програми (бізнес-рівень програми), яка визначає курс дій залежно від рівня доступу користувача та типу запиту.

4. Якщо запит відповідає критеріям доступу, логіка програми завантажує дані з бази даних і надсилає їх у браузер.

5. Браузер отримує веб-документ з даними, аналізує його та завантажує будь-які статичні файли (JavaScript, CSS, зображення, відео тощо) із CDN. У сучасних веб-додатках логіка програми може мати рівень JavaScript, який потім взаємодіє з бекендом і різними API з браузера.

6. Браузер відтворює інформацію та відображає її як сторінку веб-сайту.

Кожна програма (веб, настільна або мобільна) має дві сторони: інтерфейс і бекенд.

Frontend (на стороні клієнта) – код, який виконується у веб-браузері або інтерфейсі користувача. Це частина програми, яку користувач бачить і з якою взаємодіє.

Backend (на стороні сервера) – сервери, бази даних і логіка програми, які працюють за завісою. Бекенд-компоненти зберігають дані, обробляють запити, передають інформацію, яка відображається в додатку.

Архітектура веб-додатків – це структура взаємовідносин між компонентами веб-додатка на стороні клієнта (фронтенд) і на стороні сервера (бекенд). Простіше кажучи, архітектура визначає, як елементи програми працюватимуть разом [4].

UI Інтерфейс користувача (UI) – це візуальне представлення веб-програми, що відображається у веб-переглядачі та дозволяє користувачам взаємодіяти з ним.

Рідко можна знайти інтерфейс користувача як частину архітектури, але він є компонентом сучасних програм, оскільки частина бізнес-логіки виконується у браузері за допомогою JavaScript.

Вміст веб-сторінки включає:

1. HTML (HyperText Markup Language) – визначає зміст і структуру веб-сторінки.

2. CSS (каскадні таблиці стилів) – визначає стиль (розмір тексту, колір і шрифт) і подання сторінки (розділяє вміст і макет).

3. JavaScript – мова програмування, що відповідає за взаємодію браузера. Рівень JavaScript знаходиться в браузері користувача і взаємодіє з сервером для оновлення веб-сторінки в режимі реального часу. У сучасних моделях архітектури веб-сайтів це дозволяє вмісту сторінки динамічно змінюватися, коли користувачі взаємодіють з елементами інтерфейсу.

Веб-сервери – це комп'ютери, які приймають запити, обробляють їх і надсилають дані в браузер користувача. Термін сервер застосовується до будь-якого комп'ютера, який надає послуги через Інтернет (або приватну мережу). Сервери поділяються на різні категорії залежно від типу служб, які на них працюють. Наприклад, веб-сервери обробляють запити HTTP (порт 80) і HTTPS (порт 443).

Сервери баз даних – це централізовані місця розташування баз даних (організовані колекції інформації). Ці сервери взаємодіють з веб-серверами та надають дані авторизованим користувачам.

Веб-додатки можуть складатися з кількох служб, розміщених на кількох серверах. Проміжне програмне забезпечення обміну повідомленнями допомагає різним програмним модулям, мовам програмування та додаткам взаємодіяти. На стороні клієнта і на стороні сервера веб-програми взаємодіють за допомогою синхронних (HTTPS і TCP) протоколів, тоді як серверні компоненти програми також можуть використовувати асинхронний (AMQP) протокол зв'язку. Крім того, проміжне програмне забезпечення та інтерфейси прикладного програмування (API) об'єднують дані між додатками та серверами.

Кожен сервер має обмежений обсяг оперативної пам’яті та процесора, тому збільшення трафіку користувачів може призвести до проблем із продуктивністю. Балансувальник навантаження обробляє трафік і розподіляє його між веб-серверами на основі доступних обчислювальних ресурсів. Це оптимізує навантаження, дозволяючи веб-програмі працювати безперебійно.

Кеш – це тимчасовий носій, який зберігає дані веб-сторінки та швидше доставляє їх при повторних запитах. В архітектурі сервера веб-додатків кешування зазвичай відбувається у браузері. Наприклад, коли вперше завантажується веб-сторінка, її частини зберігаються у локальному сховищі. Потім, коли сторінка перезавантажується, браузер отримує ці файли безпосередньо з кешу (без зв’язку з веб-сервером).

CDN Мережа доставки вмісту (CDN) – це система серверних вузлів, які зберігають і передають статичний вміст (зображення, аудіо, відео, файли CSS і JavaScript). CDN можуть зменшити затримку для користувачів, які живуть далеко від центрів обробки даних, де розміщено веб-додаток. Уявіть, що користувач з Каліфорнії отримує доступ до веб-програми в центрі обробки даних у Нью-Йорку. Браузер може отримати інформацію з центру обробки даних, але сторінка завантажується довше через відстань. Тепер, припустимо, веб-програма використовує CDN в Каліфорнії. Тоді браузер зможе завантажити важкі файли з сусіднього вузла набагато швидше.

Веб-додатки мають багато переваг, зокрема такі [2]:

1. Їх легко встановити та використовувати на різних комп’ютерах та мобільних пристроях.

2. Для роботи їм потрібен лише сумісний браузер і вони не залежать від жодної конкретної операційної системи чи пристрою.

3. Оскільки вони можуть працювати на кількох платформах, розробникам не потрібно створювати окремі програми на стороні клієнта для різних операційних систем і комп’ютерів.

4. Оскільки веб-програми зберігаються на віддаленому веб-сервері, немає необхідності розміщувати їх локально, і не доведеться турбуватися про обмеження місця на жорсткому диску.

5. Веб-додатки зберігаються на віддаленому веб-сервері, що зменшує потребу в технічній підтримці.

6. Легко вносити зміни у веб-програми та постійно оновлювати їх, оскільки всі оновлення застосовуються централізовано на веб-сервері.

7. Оскільки одна версія веб-програми доступна всім користувачам, це усуває будь-які проблеми з сумісністю.

8. Якщо існує спеціальний веб-додаток для власного бізнесу, є можливість налаштувати та масштабувати його, додати більше функцій відповідно до потреб.

9. Для бізнес-користувачів веб-додатки пропонують краще керування роботою та підвищення ефективності роботи. Існує можливість використовувати їх для інтеграції даних з кількох джерел, обміну електронними таблицями даних з членами команди та співпраці над проектами та звітами.

10. Веб-додатки пропонують більш високий рівень безпеки даних, оскільки дані зберігаються на віддалених виділених веб-серверах. Досвідчені адміністратори серверів відстежують ці сервери на предмет будь-яких потенційних порушень і вживають заходів для їх запобігання.

11. Вони можуть використовувати хмарне сховище даних, що попередить втрату даних в разі пошкодження комп’ютера або мобільного пристрою.

12. Різні веб-програми можуть взаємодіяти між собою, що дозволяє покращити веб-інтеграцію та адаптувати новітні Інтернет-технології.

Користувачі висувають до сучасних веб-додатків максимально високі вимоги. Такі веб-застосунки повинні бути доступні постійно з будь-якої точки світу практично з будь-якого пристрою, незалежно від розміру його екрану. Веб-програми повинні бути безпечними, гнучкими та масштабованими, щоб ефективно справлятися з різкими стрибками навантаження. Повнофункціональні інтерфейси користувача реалізують все більш складні сценарії з використанням клієнтів на основі JavaScript і забезпечують ефективну взаємодію через веб-API [5].

1.2. Аналіз проблеми управління вразливостями веб-додатку

Сучасний бізнес використовує останні технологічні досягнення, щоб перенести більшість своїх операцій в Інтернет. Віддалена робота є більш поширеною, ніж будь-коли раніше, і компанії, розділені великими відстанями, можуть миттєво ділитися конфіденційними даними. Але ці переваги також пов'язані з ризиками. Веб-додатки вразливі до різноманітних кіберзагроз, і багато організацій не мають стратегій або технологій, необхідних для протидії.

Для багатьох компаній 2020 рік був пов’язаний з переходом на віддалену роботу в хмарних корпоративних системах, і командам з безпеки додатків довелося адаптуватися до змін у використанні та зростаючого числа проблем. Згідно зі звітом Verizon Data Breach Investigations Report за 2020 рік, уразливості веб-додатків були причиною 43% зломів та викрадення даних у 2019 році. Дивно, але згідно з дослідженням Enterprise Strategy Group, 79% організацій навмисно запустили вразливий код у виробництво, в той же час вважаючи свою позицію безпеки додатків вище 7 з 10 балів.

Оскільки середня вартість злому даних становить 3,86 мільйона доларів, безпека додатків, безумовно, не є тим, що підприємства можуть ігнорувати. Цифри зростають – за останні п’ять років зросли на 12%. У порівнянні з іншими IT-активами, веб-додатки особливо вразливі до атак, оскільки вони піддаються впливу Інтернету. Багато векторів атаки на веб-додатки зосереджені на маніпулюванні вводами користувачів через веб-форми та машинними вводами через API [6].

У 2020 році NIST опублікував рекорд з понад 18 000 нових уразливостей. Але якість коду не гірша, ніж будь-коли. Існує кілька причин, чому розкривається зростаюча кількість уразливостей:

1. Інновації та прискорена цифровізація бізнесу призводять до збільшення випуску апаратних і програмних продуктів. У 2010 році NIST зафіксував 22 188 нових записів у своєму сховищі CPE, включаючи 1 332 нових продуктів та 406 видавців. У 2020 році в сховищі з’явилося 324 810 записів (+1 460 %), 35 794 нових продуктів (+2 690 %) і 6 060 видавців (+1 490%) [7].

2. Попит на швидший вихід на ринок змушує постачальників скорочувати цикли розробки, щоб швидше випускати та продавати продукти, навіть якщо це означає економію ресурсів, необхідних для забезпечення якості та тестування безпеки.

3. Кіберзлочинність стала прибутковим бізнесом. Зростаюча кількість вразливостей тепер пов’язана з тим, що кіберзлочинці шукають нові інструменти для підтримки своїх атак.

4. Водночас збільшується кількість експертів та незалежних організацій, які займаються дослідженням та розкриттям вразливостей. Демократизація та індустріалізація програм Bug-Bounty не пов’язані з цим.

5. І, нарешті, за рідкісним винятком, таким як GDPR, через відсутність належного законодавства та нормативних актів для захисту прав споживачів у разі вразливості програмного забезпечення, індустрія не має стимулів інвестувати в більш безпечні продукти та брати відповідальність за заподіяну шкоду.

Кібератаки на веб-додатки стали більш інтенсивними та складними, тому кожній організації важливо розробити стратегічний план захисту та реагування на основі найбільш поширених і небезпечних методів веб-вторгнення. Для цього розробники веб-додатків та експерти з кібербезпеки звертаються до топ-10 OWASP, щоб отримати уявлення про найбільш актуальні ризики безпеки [8].

Понад 20 років Фонд Open Web Application Security Project (OWASP) проводить інтенсивні дослідження вразливостей веб-додатків і тенденцій атак, створюючи набір галузевих посібників і стандартів безпеки програмного забезпечення. Серед них OWASP Top 10 є найпопулярнішим і широко використовуваним посібником з підвищення обізнаності щодо безпеки веб-додатків. У списку наведено десять найбільш критичних ризиків веб-безпеки, які актуальні в даний час. Складений з використанням результатів досліджень десятків партнерських організацій, список OWASP Top 10 був вперше опублікований у 2003 році і оновлюється раз на три-чотири роки. У вересні 2021 року список отримав оновлення з 2017 року, що ілюструє деякі з останніх загроз веб-безпеці (рис. 1.1).

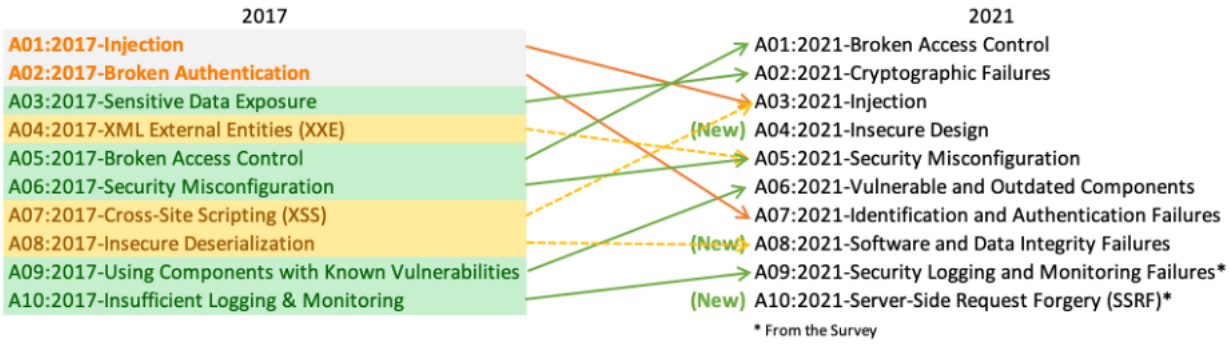

Рис. 1.1 Оновлений OWASP Top 10 [8]

A01:2021-Категорія порушення контролю доступу піднялася з п’ятої позиції до категорії з найсерйознішим ризиком безпеки веб-додатків. Надані дані вказують на те, що в середньому 3,81% протестованих додатків мали одну або кілька вразливостей (CWE) з більш ніж 318 тисячами випадків CWE в цій категорії ризику.

CWE – це розроблений спільнотою список типів недоліків програмного та апаратного забезпечення. Він служить спільною мовою, базовою лінією для виявлення слабких місць, їх усунення та запобігання [9].

34 CWE, зіставлені з порушенням контролю доступу, мали більше випадків у додатках, ніж будь-яка інша категорія. Топ-10 OWASP за 2021 рік додає до даної категорії наступні елементи:

атаки грубою силою;

обхід аутентифікації та авторизації;

видання себе за іншу особу;

скидання пароля (у контексті).

Привабливість цих вразливостей пов’язана з тим, що існує кілька методів використання цих недоліків, більшість з яких не вимагають глибокого розуміння коду. Спільним знаменником, який їх об’єднує, є відсутність захисту для кінцевих користувачів.

A02:2021-Криптографічні збої зміщуються на одну позицію вгору до пункту, раніше відомого як A3:2017-Розкриття чутливих даних, що було скоріше поширеним симптомом, ніж першопричиною. Оновлена назва зосереджена на

збоях, пов’язаних із криптографією. Ця категорія часто призводить до розкриття конфіденційних даних або компрометації системи. Криптографічні збої виникають, коли механізми захисту даних використовуються некоректно: використовується симетричне шифрування з ключем, який використовується кількома компонентами, або приватні ключі поширюються у вигляді простого тексту. Деякі з проблем, які виникають, включають передбачувані генератори випадкових чисел, відсутність ентропії та передбачувані ключі. Ця вразливість може викликати можливість підробки даних у сховищі даних або під час їх передачі по мережі.

A03:2021-Ін’єкції займають третю позицію. 94% додатків були перевірені на певну форму ін’єкції з максимальним рівнем інцидентів 19%, середнім рівнем 3,37%, а 33 CWE, віднесені до цієї категорії, займають друге місце за кількістю випадків у додатках із 274 тисячами випадків. Міжсайтові скрипти тепер є частиною цієї категорії. Топ-10 OWASP на 2021 рік переміщує фокус від ризику безпеки веб-додатків до недоліків бізнес-логіки, коли програму можна зламати через доступ до бази даних. До списку додано такі елементи:

маніпулювання параметрами;

ін'єкції через API.

У деяких випадках веб-програми стають вразливими через залежність від іншої системи, коли компоненти серверної частини можуть бути не налаштовані так безпечно, як вони повинні бути, або коли незахищені компоненти проникають у програму.

A04:2021-Небезпечний дизайн – це нова категорія для 2021 року, зосереджена на ризиках, пов’язаних із недоліками дизайну. Галузь безпеки потребує моделювання загроз, безпечних шаблонів і принципів проектування та еталонних архітектур. Небезпечний дизайн неможливо виправити ідеальною реалізацією, оскільки за визначенням необхідні засоби контролю безпеки ніколи не створювалися для захисту від конкретних атак.

Небезпечні шаблони дизайну були додані до списку, щоб вирішувати проблеми дизайну програми, а не розглядати конкретні вразливості. Топ-10 OWASP за 2021 рік визначає цю вразливість як:

1. Логічні недоліки через неправильні логічні структури.

2. Небезпечні прямі посилання на об'єкти, що включають стани або дії, до яких користувач не повинен мати доступ без належної аутентифікації.

3. Небезпечне використання механізмів аутентифікації та керування сеансами.

У багатьох випадках програми були розроблені із захистом даних позаду. У цих випадках досліднику безпеки дуже легко визначити серйозні недоліки, які можуть призвести до фінансових втрат або крадіжки особистих даних.

A05:2021-Неправильна конфігурація безпеки перемістилася на п’яту позицію. 90% додатків були перевірені на певну форму неправильної конфігурації, із середнім рівнем інцидентів 4,5%, і понад 208 тисяч випадків CWE віднесено до цієї категорії ризику. Колишня категорія A4:2017-XML (XXE) тепер є частиною цієї категорії.

Ця вразливість визначається як:

дані програми в чужих руках;

ненадійне або неналаштоване криптографічне сховище;

слабкі конфігурації безпеки.

Ця вразливість є комбінацією багатьох інших уразливостей, які раніше входили до списку Top 10 OWASP. Він фіксує всі відомі недоліки конфігурації та поєднує їх з деякими новими елементами, такими як використання компонентів з відомими вразливими місцями, недостатній захист від атак і незахищена конфігурація SSL/TLS.

A06:2021-Уразливі та застарілі компоненти раніше називалася «Використання компонентів із відомими вразливими місцями» і займала друге місце в опитуванні спільнот, але також мала достатньо даних, щоб увійти до топ-10 за допомогою аналізу даних. Ця категорія включає небезпечне використання бібліотек, фреймворків, додатків та компонентів сторонніх розробників, що використовуються в програмі. Деякі ключові приклади:

1. Компоненти програми доступні за відомими або загальнодоступними шляхами за замовчуванням.

2. Компоненти використовують застарілі/уразливі компоненти, які забезпечують легкий доступ до конфіденційних даних.

3. Компоненти не використовують стандартні методи безпечного кодування або слабкі стандарти криптографії Якщо будь-яка з цих вразливостей буде виявлена, це може призвести до тих самих руйнівних результатів, для запобігання яких призначена категорія A5:2021-Неправильна конфігурація безпеки.

Ця категорія піднялася з дев'ятого місця у 2017 році і є відомою проблемою, яку важко перевірити та оцінити ризики.

A07:2021-Помилки ідентифікації та автентифікації тепер включають CWE, які більше пов'язані з помилками ідентифікації. Ця категорія все ще є невід'ємною частиною топ 10, але збільшення доступності стандартизованих фреймворків допомагає покращити ситуацію.

Помилки ідентифікації та автентифікації можна визначити як:

недостатні механізми ідентифікації та аутентифікації;

ненадійні механізми ідентифікації та аутентифікації;

ненадійно збережені паролі, маркери скидання пароля або інші небезпечні форми підтвердження особи (наприклад, на основі знань) .

Ця вразливість зазвичай усувається за допомогою багатофакторної аутентифікації, яка є одним із провідних механізмів безпеки додатків.

A08:2021-Збої цілісності програмного забезпечення та даних – це нова категорія для 2021 року, яка зосереджена на створенні припущень, пов'язаних з оновленнями програмного забезпечення, критично важливими даними та конвеєрами CI/CD без перевірки цілісності. A8:2017-Небезпечна десеріалізація тепер є частиною цієї більшої категорії.

Топ-10 OWASP на 2021 рік призначає A8:2021 до вразливостей, пов'язаних із встановленням шкідливого програмного забезпечення, зміною даних або проблемами цілісності даних. Ці проблеми включають такі речі, як:

наявність конфіденційної інформації у вихідному коді або конфігурації;

взаємодія зі сторонніми компонентами/сервісами;

ненадійні вхідні дані, передані як параметр команді ОС.

Неперевірені переспрямування та пересилання в кінцевому підсумку призводять до того, що конфіденційні дані надсилаються третій стороні або веб-переглядачу

Ці вразливості були помічені раніше, але їх помітність неможливо було визначити без використання автоматизованих інструментів. Топ-10 OWASP на 2021 рік сподівається висвітлити ці проблеми, щоб їх можна було належним чином вирішити.

Категорія A09:2021-Security Logging and Monitoring Failures раніше A10:2017-Недостатнє ведення журналу та моніторингу додана з опитування спільнот. Ця категорія розширена, щоб охопити більше типів збоїв, її складно перевірити, і вона погано представлена в даних CVE/CVSS. Однак збої в цій категорії можуть безпосередньо вплинути на видимість, оповіщення про інциденти та криміналістичну експертизу.

Ця категорія охоплює всі проблеми моніторингу безпеки, пов’язані з активністю додатків, керуванням сеансами та доступом до конфіденційних даних. Деякі приклади цієї вразливості включають:

журнали програм, які неправильно налаштовані або не ввімкнені;

неможливість реєстрації достатньої інформації про події аутентифікації;

неможливість реєстрації достатньої інформації, коли користувач отримує доступ до конфіденційних даних;

немає можливості відстежувати або сповіщати про аномальну поведінку законних користувачів.

Ця вразливість включає прогалини, які були помічені в журналах безпеки та моніторингу з часом. Це не лише питання конфігурації, оскільки іноді це пов’язано з тим, що програма не здатна контролювати себе.

A10:2021-Підробка запитів на стороні сервера додана з опитування спільнот. Ця вразливість охоплює всі відомі методи підробки запитів на стороні сервера, тобто атаки, які використовують веб-додаток як вектор атаки.

У топ 10 OWASP за 2021 рік наведено такі приклади цієї вразливості:

недоліки ін'єкції, які дозволяють зловмисникові контролювати або впливати на запит від веб-додатка до серверної бази даних і навпаки;

відсутній контроль автентифікації та керування сеансом, що робить програму вразливою;

відсутність належної обробки помилок у самому веб-додатку

недостатні механізми реєстрації, які ускладнюють розпізнавання або розслідування атаки підроблення запиту.

Ця вразливість часто пов’язана з A5:2021-Неправильна конфігурація безпеки, і зловмисники продовжуватимуть використовувати її, знаходячи шляхи обходу нових механізмів безпеки.

Топ 10 OWASP змінювався багато разів з моменту його заснування в 2003 році, але він все ще служить еталоном, якого слід дотримуватися, коли справа доходить до безпеки веб-додатків.

На основі даних Trend Micro Cloud One у наведеному нижче списку представлено 15 CVE які, як відомо, активно експлуатуються або мають відоме підтвердження концепції (POC). Якщо співвіднести кожну вразливість з її власними категоріями з OWASP Top 10, можна визначити основні недоліки в безпеці веб-додатку [10] (рис 1.2).

Top Vulnerabilities With Known Exploits or Proofs of Concept	CVE	Severity	Related Top 10 2021
Apache Struts2 remote code execution (RCE) vulnerability	CVE-2017-5638	Critical	A03, A06
Apache Struts 2 REST plugin XStream RCE vulnerability	CVE-2017-9805	High	A03, A06
Drupal Core RCE vulnerability	CVE-2018-7600	Critical	A03
Oracle WebLogic server RCE vulnerabilities	CVE-2020-14750	Critical	A03
WordPress file manager plugin RCE vulnerability	CVE-2020-25213	Critical	A03, A06
vBulletin 'subwidgetConfig' unauthenticated RCE vulnerability	CVE-2020-17496	Critical	A03
SaltStack salt authorization weakness vulnerability	CVE-2020-11651	Critical	A01, A02
Apache Struts OGNL expression RCE vulnerability	CVE-2017-12611	Critical	A03, A08
Eclipse Jetty chunk length parsing integer overflow vulnerability	CVE-2017-7657	Critical	A03, A08
Alibaba Nacos AuthFilter authentication bypass vulnerability	CVE-2021-29441	Critical	A07
Atlassian Jira information disclosure vulnerability	CVE-2020-14179	Medium	A05
Nginx crafted URI string handling access restriction bypass vulnerability	CVE-2013-4547	N/A	A01
Apache Struts 2 RCE vulnerability	CVE-2019-0230	Critical	A03, A06
Apache Struts OGNL expression RCE vulnerability	CVE-2018-11776	High	A03, A08
Liferay portal untrusted deserialization vulnerability	CVE-2020-7961	Critical	A08

Рис. 1.2. Відомі вразливості [10]

1.3. Мета та завдання управління вразливостями веб-додатку

Щороку виявляються тисячі нових вразливостей, які вимагають від організацій виправляти операційні системи і додатки, а також змінювати параметри безпеки у всьому своєму мережевому середовищі, щоб завчасно усунути вразливості до того, як вони будуть використані для кібератаки. Організації, які серйозно ставляться до безпеки свого середовища, здійснюють управління вразливими місцями, щоб забезпечити найвищий можливий рівень безпеки.

Управління вразливістю зазвичай визначається як процес виявлення, категоризації, визначення пріоритетів та усунення вразливостей в веб-додатках операційних системах, корпоративних програмах (як у хмарі, так і локально), браузерах і додатках кінцевих користувачів.

Постійний процес управління вразливими місцями спрямований на постійне виявлення вразливостей, які можна усунути за допомогою виправлення та налаштування параметрів безпеки [11].

Більшість організацій мають процес управління вразливими місцями у своїй мережі і все ще не вирішують проблеми з вразливостями. Інститут Ponemon провів опитування 1848 спеціалістів із IT та IT-безпеки в Північній Америці, країнах EMEA, APAC та Латинської Америки. У звіті більшість респондентів самі повідомляють, що їх ефективність у визначенні пріоритетів та виправленні вразливостей низька. Такий же рівень має захисті додатків у хмарі. Це може бути пов’язано з різними причинами, насамперед з неправильним виконанням процесу управління вразливістю [12].

ISO 27005 визначає вразливість як слабкість активу або групи активів, які можуть бути використані однією або кількома кіберзагрозами, де актив – це все, що має цінність для організації, її бізнес-операцій та їх безперервності, включаючи інформаційні ресурси.

Для SANS Institute управління вразливими місцями – це процес, у якому вразливості в IT визначаються та оцінюються ризики цих вразливостей. Ця оцінка призводить до виправлення вразливостей та усунення ризику або офіційного

прийняття ризику керівництвом організації. Згодом управління вразливістю стало фундаментальною практикою в галузі кібербезпеки, і тепер усі професіонали галузі погодяться сказати, що це важливий процес для мінімізації поверхні атак компанії.

Процес управління вразливістю веб-додатку можна розділити на наступні чотири кроки [14]:

виявлення вразливостей;

оцінка вразливостей;

усунення вразливостей;

звітування про вразливості.

Перший і найважливіший крок у будь-якому процесі управління вразливістю, звичайно, полягає в тому, щоб виявити всі вразливості, які можуть існувати у середовищі. Сканер уразливостей робить це, скануючи весь спектр доступних систем, які існують – від ноутбуків, настільних комп'ютерів і серверів до баз даних, брандмауерів, комутаторів, принтерів тощо. Звідти сканер уразливостей визначає будь-які відкриті порти та служби, які запущені в цих системах, входить у ці системи та збирає детальну інформацію, де це можливо, перш ніж співвідносити отриману інформацію з відомими вразливими місцями. Цю статистику можна використовувати для створення звітів, показників та інформаційних панелей для різних цілей.

Тестування безпеки веб-додатків має на меті визначити, чи є веб-додаток уразливим до атак. Він охоплює як автоматизовані, так і ручні методи за допомогою ряду різних методологій:

1. Динамічний тест безпеки додатків (DAST). Цей автоматизований тест безпеки додатків найкраще підходить для внутрішніх програм із низьким рівнем ризику, які мають відповідати нормативним оцінкам безпеки. Для програм середнього ризику та критичних програм, які зазнають незначних змін, найкращим рішенням є поєднання DAST з деякими ручними тестуваннями веб-безпеки на предмет поширених уразливостей. DAST включає тестування розгорнутого або запущеного коду для виявлення вразливостей. Його можна виконувати як вручну, так і автоматично, використовуючи спеціальні інструменти. Ручне тестування

полягає в роботі з API програми за допомогою таких інструментів, як BurpSuite, Fiddler, Postman. Інструменти автоматизації DAST надсилають велику кількість запитів до коду програми, включаючи неочікувані та шкідливі дані, шукаючи вразливості. Вони аналізують результати та визначають слабкі місця безпеки. Після ретельного аналізу безпеки вручну можна використовувати засоби, такі як Owasp Zap, сканер безпеки веб-додатків з відкритим вихідним кодом, щоб прискорити регресійне тестування. Сканери не можуть замінити людину з точки зору творчості, аналізу першопричин або здатності мислити нестандартно, але вони можуть справлятися з рутинними завданнями набагато швидше й об'ємніше.

2. Статичний тест безпеки додатків (SAST). Цей підхід до безпеки програм пропонує автоматизовані та ручні методи тестування. Це найкраще підходить для виявлення помилок без необхідності виконувати програми у виробничому середовищі. Це також дозволяє розробникам сканувати вихідний код і систематично знаходити та усувати вразливості безпеки програмного забезпечення. Багато веб-додатків інтегрують сканування коду на кількох етапах розробки – в основному під час фіксації нового коду в кодовій базі та під час збірки. SAST зазвичай базується на правилах, а результати сканування зазвичай містять помилкові результати, тому потрібно ретельно аналізувати та фільтрувати результати, щоб визначити реальні проблеми безпеки. Можна використовувати інструменти статичного аналізу коду, такі як Sonar Qube для моніторингу проблем безпеки, які можуть виникнути під час розробки. Рекомендується інтегрувати його з конвеєром CI/CD, щоб він сканував кожні зміни. Sonar Qube має якісне візуальне відображення і перевіряє не тільки аспекти безпеки, але також ремонтопридатність і надійність бази коду. Він підтримує понад 20 різних мов програмування, тому працює для більшості інтерфейсних і бекенд фреймворків.

3. Тест на проникнення. Цей ручний тест безпеки програми найкраще підходить для критичних програм, особливо тих, які зазнають серйозних змін. Оцінка включає в себе бізнес-логіку та тестування на основі супротивників, щоб виявити розширені сценарії атак. Тестування на проникнення – це техніка тестування безпеки, яка поєднує інструменти динамічного сканування та досвід у

сфері безпеки людини, щоб знайти прогалини в системі безпеки веб-додатків. Пентестери діють як реальні загрози – експлуатують вразливості, отримують несанкціонований доступ, крадуть дані та порушують роботу служб. Однак вони роблять це за контрактом з власником веб-додатка, згідно з узгодженим обсягом, і не завдаючи реальної шкоди організації. Порівняно з SAST і DAST, ця методика є складнішою у виконанні, але може виявити додаткові ризики, які автоматизовані інструменти можуть упустити.

4. Самозахист додатків під час виконання (RASP). Цей підхід до безпеки додатків, що розвивається, охоплює ряд технологічних прийомів для інструментування програми, щоб можна було відстежувати атаки під час їх виконання та, в ідеалі, блокувати в режимі реального часу.

5. Рішення розширеного виявлення та реагування (XDR) – це нове покоління платформ безпеки, які надають командам безпеки один інтерфейс, який дозволяє їм виявляти загрози та реагувати на них, де б вони не були в IT-середовищі. XDR збирає дані безпеки з усіх рівнів стеку безпеки, включаючи веб-додатки, мережі, приватні та загальнодоступні хмари та кінцеві точки. Він застосовує передову аналітику та автоматизацію для аналізу, сортування та виявлення як відомих, так і невідомих загроз. Найголовніше, він безпосередньо інтегрується з інструментами безпеки і може автоматично реагувати на загрози в режимі реального часу.

Після того, як визначено всі вразливі місця у середовищі, потрібно їх оцінити, щоб належним чином впоратися з ризиками, які вони представляють, відповідно до стратегії управління ризиками організації. Різні рішення для управління вразливими місцями використовують різні рейтинги ризику та оцінки вразливостей, але одна з найпопулярніших платформ для нових програм – це Загальна система оцінки вразливостей (CVSS).

Оцінки вразливості можуть допомогти організаціям визначити, як розставити пріоритети вразливостей, які вони виявили. Важливо також враховувати інші фактори, щоб сформувати повне розуміння справжнього ризику, який створює будь-яка дана вразливість. Також варто зазначити, що сканери вразливостей можуть генерувати помилкові результати в рідкісних випадках, що підкреслює

необхідність включення інших міркувань на додаток до оцінок ризику на цьому етапі процесу.

Управління вразливими місцями – це процес оцінки, класифікації та, в кінцевому рахунку, усунення загроз (уразливостей), з якими стикається організація. Мабуть, неможливо виправити та виправити кожну окрему вразливість у мережі чи комп'ютерній системі, а іноді – це марна трата ресурсів. Зрештою, лише приблизно 2% уразливостей в кінцевому підсумку експлуатуються (Kenna security, 2020). Натомість, це пов'язано з усуненням найнебезпечніших вразливостей та зменшенням ймовірності використання вектора атаки для використання системи. Ось тут і вступає в гру оцінка вразливості.

Після того, визначено пріоритети вразливостей, які було виявлено, важливо якомога швидше усунути їх у співпраці з командою безпеки та зацікавленими сторонами. Залежно від уразливості, про яку йдеться, усунення зазвичай відбувається за одним із трьох шляхів:

1. Виправлення: повне виправлення або виправлення вразливості, щоб нею не можна було скористатися, що зазвичай є найбільш кращим варіантом, коли це можливо.

2. Пом'якшення: коли виправлення не може бути здійснено, організація може вибрати наступний найкращий варіант зменшення ймовірності використання вразливості шляхом впровадження компенсуючих засобів контролю. Це рішення має бути тимчасовим, щоб виграти час для організації, щоб зрештою усунути вразливість.

3. Прийняття: якщо вразливість вважається не критичною або вартість її усунення є набагато більшою, ніж була б, якби вона була використана, організація може вирішити просто не вживати заходів для усунення вразливості.

Визначаючи конкретні стратегії усунення вразливостей, найкраще, щоб команда безпеки організації, власники систем і системні адміністратори об'єдналися і визначили правильний підхід до усунення проблеми – чи то випуск програмного виправлення, чи оновлення парку фізичних серверів. Після того, як виправлення вважається завершеним, доцільно провести ще одне сканування

уразливості, щоб переконатися, що вразливість фактично була ефективно усунена або пом’якшена.

Підвищення швидкості та точності виявлення та усунення вразливостей має важливе значення для управління ризиком, який вони представляють, тому багато організацій постійно оцінюють ефективність своєї програми управління вразливими місцями. Для цієї мети вони можуть скористатися можливостями візуальних звітів, які є в рішеннях для керування вразливими місцями. Озброєні необхідними знаннями, IT-команди можуть визначити, які методи усунення допоможуть їм усунути найбільшу вразливість із найменшими зусиллями. Команди безпеки, зі свого боку, можуть використовувати цю звітність, щоб відстежувати тенденції вразливості з часом і повідомляти керівництву про прогрес щодо зниження ризиків. Ідеальні рішення включатимуть інтеграцію з IT-системами квитків (ticketing system) та інструментами виправлення, щоб прискорити процес обміну інформацією між командами. Це допомагає клієнтам досягти значного прогресу в напрямку зниження ризику. Підприємства також можуть використовувати ці оцінки для виконання своїх вимог щодо відповідності та нормативних вимог.

1.4. Аналіз існуючих технологій управління вразливостями веб-додатку сучасного підприємства

Програмне забезпечення для управління вразливостями може допомогти автоматизувати цей процес. Воно включає в себе сканер уразливостей, а іноді й агенти кінцевих точок, щоб інвентаризувати різноманітні системи в мережі та знайти в них уразливості. Після визначення вразливостей ризик, який вони створюють, необхідно оцінити в різних контекстах, щоб можна було прийняти рішення про те, як найкраще їх усунути. Наприклад, перевірка вразливості може бути ефективним способом контекстуалізації реальної серйозності вразливості.

Багато постачальників програмного забезпечення для оцінки вразливостей вийшли за межі виявлення та оцінки в традиційних локальних IT-середовищах і

тепер здатні надавати інформацію про сучасну поверхню атак, включаючи хмари, контейнери, операційні технології (OT) та Інтернет речей (IoT). Однак деякі пропонують лише частини рішення, а інші розширили свої портфелі, щоб надати більш повну пропозицію. Щоб визначити, яке рішення для оцінки вразливості підходить для організації, важливо зрозуміти варіанти використання та можливості, які підтримує кожен постачальник, і порівняти їх із потребами – і визначити, що допоможе максимально підвищити ефективність програми оцінки вразливості.

Програмне забезпечення для тестування безпеки додатків – це програма для пошуку вразливостей у програмі або середовищі. Тестування безпеки програми слід виконувати, розглядаючи всі кути. Ці інструменти можуть виявляти як відомі, так і невідомі атаки.

Інструменти тестування веб-безпеки можна розділити на дві категорії: засоби автоматизації та інструменти для ручного тестування. Сканери вразливостей, аналізатори коду та аналізатори складу програмного забезпечення є автоматичними інструментами, а такі інструменти, як фреймворки для проведення атак є ручними. Для забезпечення безпеки веб-додатків підприємствам слід виконати деякі практичні кроки. Потрібно інвестувати в якісне програмне забезпечення для тестування безпеки додатків, рішення DAST та інструмент, який може перевіряти веб-активи за заданими критеріями.

Багато компаній вважають безпеку додатковою складовою або розглядають її як окрему проблему. Дуже мало компаній розглядають тестування безпеки як невід’ємну частину циклу розробки. Дослідження Acunetix показують, що 80% підприємств захищають усі свої веб-додатки за допомогою певного тесту безпеки.

Безпеку веб-додатків можна досягти, виявляючи потенційні проблеми на ранніх стадіях та негайно вживаючи правильних дій. Правильний інструмент тестування безпеки програми допоможе досягти даної цілі. Вибираючи інструмент, можна враховувати такі функції, як надання доказів уразливостей, можливості автоматизації та функції звітності інструмента. Докази, надані інструментом, допоможуть прийняти правильні заходи, а також зведуть до мінімуму помилкові

спрацьовування. Останнє, але не менш важливе, це ціна інструменту, на яку варто звернути увагу.

Важко знайти найкращий інструмент тестування безпеки програми. Кожне програмне забезпечення має деякі унікальні функції. Деякі інструменти добре виявляють недоліки безпеки, деякі мають кращі можливості звітування, деякі прості у використанні, а деякі пропонують багатий набір функцій.

Отже, щоб знайти найкращий інструмент, слід провести дослідження та знайти потрібний варіант для конкретного середовища. Інструмент повинен бути зручним у використанні. Невеликі функції також можуть зробити інструмент зручним у використанні. Такі функції, як: дізнатися більше про виявлену вразливість одним клацанням миші, налаштувати електронну пошту у сканері та надіслати сповіщення, будуть мати велике значення та забезпечать зручність. Інструмент повинен мати можливість звітування, і він повинен мати можливість надавати звіти відповідно до дотримуваних правил. Деякі інструменти надають можливість визначити пріоритетність уразливостей. Це допоможе визначитися з наступним курсом дій [15].

Netsparker провів опитування фахівців з безпеки, щоб з’ясувати, як втілити політику та програми безпеки в повсякденну практику. З’ясувалося, що майже 75% керівників довіряють тому, що їхня організація перевіряє всі веб-додатки на наявність уразливостей. З іншого боку, половина співробітників служби безпеки не погоджується з цим фактом. У тому ж дослідженні говориться, що, за даними 60% співробітників DevOps, швидкість виявлення вразливостей безпеки перевищує швидкість, з якою вони були виправлені.

Усі наведені вище результати опитування, статистика та графіки говорять про те, що 20% підприємств не захищають усі веб-додатки і йдуть на прораховані ризики. Це потенційно залишає діри в безпеці. Основні причини, чому не можна сканувати всі веб-програми, включають те, що ризики веб-додатків часто недооцінені, інструменти не можуть сканувати всі веб-програми тощо [16].

Рис.1.3 Пріоритети компаній [16]

Кількість веб-додатків, API та веб-технологій буде зростати. Проблеми можна усунути до того, як вони виникнуть, а процеси можна автоматизувати за допомогою правильних інструментів безпеки.

Список популярних рішень для тестування безпеки веб-додатків включає наступні рішення:

Netsparker;

Acunetix;

BurpSuite;

OWASP ZAP.

Acunetix — це сканер безпеки веб-додатків, який має функції пошуку, виправлення та запобігання вразливостям. Це допоможе захистити веб-сайти, веб-додатки та API. Незважаючи на те, що це сканер уразливостей, він має функції для керування безпекою веб-активів, незалежно від обсягу. За допомогою Acunetix можна запланувати та встановити пріоритет повного сканування, а також

інкрементного сканування. Його можна інтегрувати з системами відстеження, такими як Jira, GitHub тощо.

Acunetix може виявити понад 6500 вразливостей. Він може виявляти такі вразливі місця, як слабкі паролі та відкриті бази даних. Він може виявляти такі вразливості, як ін’єкції SQL, XSS, неправильна конфігурація та out of band вразливості. Це платформа, яка може сканувати всі сторінки та складні веб-додатки. Він може сканувати додатки з однією сторінкою та великою кількістю HTML5 та JavaScript. Acunetix використовує передову технологію запису макросів, яка дозволить сканувати багаторівневі форми та захищені паролем області сайту. Цей сканер веб-додатків може надати повне уявлення про безпеку організації та забезпечить кращі результати за менший час. Це інтуїтивно зрозуміла та проста у використанні платформа.

Acunetix має три тарифні плани: Стандартний, Преміум і Acunetix 360. Ціна платформи буде заснована на багаторічних контрактах.

Netsparker пропонує зручний сканер безпеки веб-додатків, який може використовуватися малими та великими підприємствами. Це платформа з функціями управління вразливими місцями та звітності. Це допоможе визначити пріоритети завдань з усунення проблем, автоматично призначаючи рівень серйозності вразливостям. Netsparker використовує технологію сканування на основі доказів, що дозволяє безпечно використовувати знайдені вразливості та створювати підтвердження концепції. Таким чином він отримає підтвердження про вразливості та зменшить кількість помилкових спрацьовувань.

Netsparker надає вбудовані звіти, а також можливість створювати власні звіти. Він має функції керування командою, такі як створення ролей, призначення проблем тощо. Це дозволяє керувати вразливими місцями за допомогою сторонніх додатків, таких як Azure DevOps, і систем керування вразливістю, таких як Metasploit. Його можна інтегрувати у платформу CI/CD. Netsparker надає всі функції для автоматизації веб-безпеки. Він забезпечує повну видимість веб-активів за допомогою таких звітів, як звіти HIPAA, звіти PCI та звіти OWASP. Служби Netsparker Asset Discovery здійснюють безперервне сканування Інтернету. Він

виявляє активи на основі IP-адрес, інформації про сертифікати SSL тощо. Він підкреслює потенційну шкоду, автоматично призначаючи рівень серйозності вразливостям.

Netsparker пропонує рішення з трьома тарифними планами: Standard, Team і Enterprise.

OWASP Zed Attack Proxy, коротко ZAP – це сканер веб-додатків. Це безкоштовний інструмент з відкритим кодом. ZAP підтримує спеціальна команда міжнародних волонтерів. Для автоматизації безпеки пропонує потужні API. У маркеті ZAP доступні різні доповнення, які розширять функціональність сканеру.

ZAP має функції для активного та пасивного сканування HTTP і пасивного сканування WebSockets. Він надає сповіщення з прапорцем, який вказуватиме на ризик. Він може обробляти різні методи автентифікації для веб-сайтів або веб-програм. ZAP містить багато інших функцій, таких як анти-CSRF-токени, точки зупинки, контексти, сесії HTTP тощо.

OWASP ZAP надає платформу для тестування безпеки. Це гнучка та розширювана платформа для тестування веб-додатків. Існує можливість підключити ZAP до проксі-сервера, який уже використовується. Його можуть використовувати розробники, нові тестувальники безпеки та експерти з тестування безпеки.

Burp Suite – це інструмент для аудиту безпеки веб-додатків. Містить інструменти для складання карти веб-програми, пошуку файлів та папок, модифікації запитів, фазінгу, підбору паролів та багато іншого. Також існує магазин доповнень BApp store, що містить додаткові розширення, що збільшують функціонал.

Burp Suite – це інтегрована платформа, призначена для проведення аудиту веб-програми, як у ручному, так і в автоматичних режимах. Містить інтуїтивно зрозумілий інтерфейс із спеціально спроектованими табами, що дозволяють покращити та прискорити процес атаки. Сам інструмент являє собою механізм, що перехоплює і обробляє всі запити, які надходять від браузера. Є можливість встановлення сертифіката Burp для аналізу https з'єднань. Якщо подивитися

статистику та репорти Bug Bounty програм – практично скрізь можна зустріти використання цього інструменту. Наряду з OWASP ZAP це найпопулярніший набір утиліт для тестування веб-додатків.

Висновки до розділу 1

В цілому вразливості веб-додатків можуть завдавати шкоди та збитків сучасним підприємствам. Згідно зі звітом Verizon Data Breach Investigations Report за 2020 рік, уразливості веб-додатків були причиною 43% зломів та викрадення даних у 2019 році.

При проведенні дослідження було встановлено найпоширеніші типи вразливостей. Провівши аналіз цих вразливостей вдалося встановити, що з кожним роком вони експлуатуються все більше та завдають збитків. Тому процес управління вразливостями грає ключову роль для забезпечення кібербезпеки веб-додатків сучасних підприємств та дозволяє уникнути негативних наслідків.

В якості засобу для підвищення ефективності управління вразливостями були розглянуті інструменти для виявлення та оцінки загроз та визначено їх основні переваги та недоліки. Програмні комплекси даного класу дозволяють домогтися практично повної автоматизації процесу виявлення вразливостей, але при не правильному налаштуванні призводять до нераціональної витрати коштів.

Також було проаналізовано процес управління вразливостями веб-додатка сучасного підприємства і фреймворк OWASP Top-10 – важливу складову для оцінки та усунення відомих вразливостей веб-додатків.

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ УПРАВЛІННЯ ВРАЗЛИВОСТЯМИ ВЕБ-ДОДАТКУ НА БАЗІ РІШЕННЯ BURP SUITE

2.1. Призначення, можливості та функції програмного комплекса BurpSuite

Burp Suite — це фреймворк, написаний на Java, який має на меті забезпечити єдине місце для тестування на проникнення веб-додатків та виявлення вразливостей. Багато в чому ця мета досягається, оскільки Burp Suite є стандартним інструментом для практичних оцінок безпеки веб-додатків. Burp Suite також дуже часто використовується при оцінці мобільних додатків, оскільки ті самі функції, які роблять його настільки привабливим для тестування веб-додатків, майже ідеально підходять для тестування API (інтерфейсів програмного забезпечення), що працюють у більшості мобільних додатків [17].

На найпростішому рівні BurpSuite може захоплювати весь трафік між клієнтом і веб-сервером і керувати ним: це ядро фреймворка. Отримавши запити, можна відправити їх до різних інших частин фреймворку Burp Suite. Ця здатність перехоплювати, переглядати та змінювати веб-запити перед їх відправкою на цільовий сервер (або, в деяких випадках, відповіді до їх отримання браузером), робить BurpSuite ідеальним для будь-якого типу ручного тестування веб-додатків. Доступні різні версії Burp Suite.

Видання Burp Suite Professional і Enterprise вимагають дорогих ліцензій, але мають потужні додаткові функції:

BurpSuite Professional – це необмежена версія BurpSuite Community. Він має такі функції, як: автоматичний сканер уразливостей; fuzzer/bruteforcer, який не обмежений по швидкості; збереження проектів для подальшого використання; формування звіту; вбудований API для інтеграції з іншими інструментами; необмежений доступ для додавання нових розширень для більшої функціональності; доступ до Burp Suite Collaborator (фактично надає унікальний

перехоплювач запитів, розміщений або запущений на сервері, що належить Portswigger). Коротше кажучи, Burp Pro є надзвичайно потужним інструментом, тому він поставляється з ціною 399$ на користувача за річну підписку. З цієї причини Burp Pro зазвичай використовують лише професіонали (ліцензії часто надаються роботодавцями).

Burp Suite Enterprise дещо відрізняється. На відміну від Community та професійних видань, Burp Enterprise використовується для безперервного сканування. Він забезпечує автоматичний сканер, який може періодично сканувати веб-програми на наявність уразливостей майже так само, як Nessus, виконує автоматичне сканування інфраструктури. На відміну від інших випусків BurpSuite, які дозволяють виконувати атаки вручну з власного комп'ютера, Enterprise знаходиться на сервері і постійно сканує цільові веб-програми на наявність вразливостей. Компанії з великими веб-портфелями, які їм потрібно часто сканувати, потребують більше ресурсів для сканування, ніж невеликі компанії, які тільки починають використовувати автоматичне сканування. Саме тому Burp Suite Enterprice Edition поставляється мінімум з 5 агентами сканування за ціною 6995$ на рік з можливістю розширення до більш ніж 50 агентів за 29 450$ на рік.

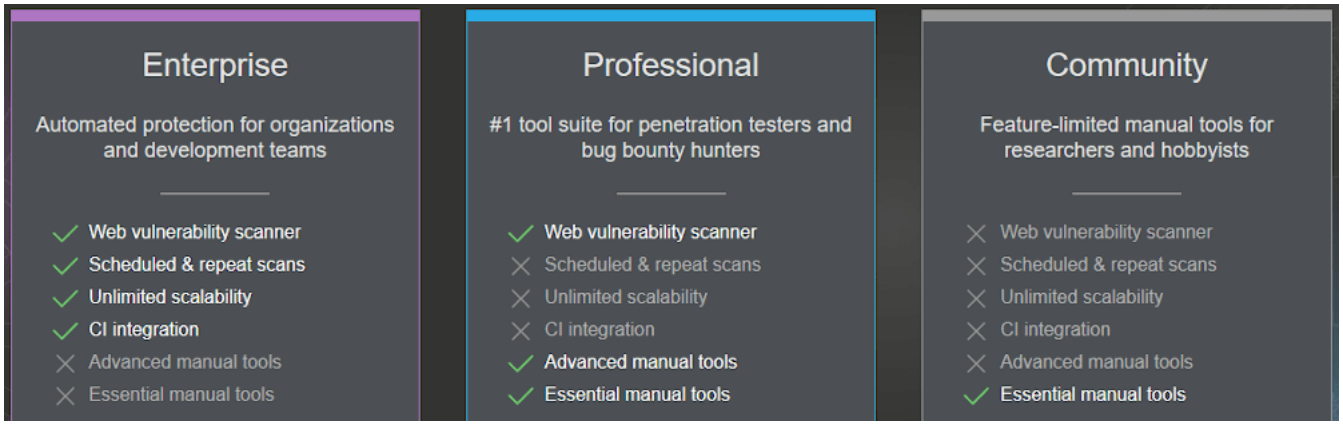

Рис. 2.1 Версії BurpSuite

2.2. Компоненти та архітектура рішення Burp Suite

Community Edition надає кожному доступ до основ Burp. Burp Suite Community Edition – це ручний, обмежений набір інструментів для дослідження веб-безпеки, який включає в себе [18]:

перехоплюючий проксі, який дозволяє інспектувати та модифікувати трафік між браузером та цільовим додатком;

інструмент Intruder для виконання потужних атак та пошуку і експлуатації незвичайних вразливостей;

інструмент Repeater для маніпуляцій та повторного надсилання індивідуальних запитів;

Decoder стандартна утиліта для перетворення текстових рядків з одного формату в інший, зокрема url, html, base64, ascii hex, hex, octal, binary, gzip;

інструмент Sequencer для тестування хаотичних сесійних токенів (маркерів);

Extender, що дозволяє легко писати власні плагіни, для виконання комплексних і високо настроюваних завдань усередині Burp;

Comparer, інструмент для зіставлення рядків і запитів: підсвічуються ділянки, які були змінені, додані або видалені.

Проксі: найвідоміший аспект Burp Suite. Burp Proxy, дозволяє перехоплювати та змінювати запити/відповіді під час взаємодії з веб-додатками. Burp Proxy надає можливість перехопити пакет у той момент, коли він уже сформований барузером, але ще не відправлений на віддалений сервер. Крім того, Burp Proxy дозволяє перехоплювати пакети, а також змінювати заголовки відповіді сервера на основі правил, заданих у форматі регулярних виразів. В цілому, дуже гнучкий та зручний інструмент.

Repeater: друга найвідоміша функція. Burp Repeater дозволяє захоплювати, змінювати, а потім повторно надсилати той самий запит багато разів. Ця функція може бути абсолютно недооціненною, особливо коли потрібно створити корисне навантаження методом проб і помилок (наприклад, у SQLi - Structured Query

Language Injection) або під час тестування функціональності кінцевої точки на наявність недоліків.

Intruder: дозволяє розпорошувати кінцеву точку запитами. Це часто використовується для атак грубої сили або для фазингу кінцевих точок. Він призначений для циклічного повторення запитів з метою збирання інформації про сервер, експлуатацію SQL-ін'єкцій, пошуку XSS-уразливостей. Спочатку у вкладці Target необхідно вказати мету випробувань (або можна відправити, наприклад, з Burp Proxy всю інформацію сюди і всі поля будуть автоматично заповнені). Далі у вкладці Positions необхідно визначити параметри для перевірки або зміни. При цьому необхідно вказати тип атаки – для більшості випадків підходить Sniper, тобто. для кожної зазначеної ділянки проводиться підстановка корисного навантаження зі списку по черзі. Як тільки були визначені всі ділянки, що цікавлять, можна переходити до налаштування атаки і вибору корисного навантаження. Burp Intruder пропонує кілька типів корисного навантаження, які здатні вирішити багато завдань, починаючи від брутфорсу форми авторизації і до автоматичного виявлення XSS.

Декодер: Хоча декодер використовується менше, ніж раніше згадані функції, декодер все ще надає цінну послугу під час перетворення даних – або з точки зору декодування захопленої інформації, або кодування корисного навантаження перед відправленням до цілі. Хоча є й інші сервіси, доступні для виконання такої ж роботи, виконання цього безпосередньо в Burp Suite може бути дуже ефективним.

Comparer: дозволяє порівнювати дві частини даних на рівні слова або байта. Знову ж таки, це не те, що є унікальним для Burp Suite, але можливість надсилати (потенційно дуже великі) фрагменти даних безпосередньо в інструмент порівняння за допомогою однієї комбінації клавіш може значно прискорити роботу.

Sequencer: зазвичай використовується під час оцінки випадковості маркерів, таких як значення cookie сеансу або інші імовірно згенеровані випадково. Якщо алгоритм не генерує безпечні випадкові значення, це може відкрити деякі руйнівні шляхи для атаки.

На додаток до безлічі вбудованих функцій, кодова база Java також дозволяє дуже легко писати розширення для додавання до функціональних можливостей фреймворку Burp. Вони можуть бути написані на Java, Python (за допомогою інтерпретатора Java Jython) або Ruby (за допомогою інтерпретатора Java JRuby).

Модуль Burp Suite Extender може швидко та легко завантажувати розширення в фреймворк, а також надавати маркетплейс для завантаження сторонніх модулів (іменується «BApp Store»). Незважаючи на те, що для завантаження та додавання багатьох із цих розширень потрібна професійна ліцензія, все ще існує чимало таких розширень, які можна інтегрувати з Burp Community. Наприклад, можна розширити вбудовану функціональність журналу Burp Suite за допомогою модуля Logger++.

Однією з основних особливостей BurpSuite Professional Edition є сканер уразливостей, з покриттям понад 100 загальних вразливостей, таких як ін’єкція SQL та міжсайтові сценарії (XSS), з високою продуктивністю проти всіх вразливостей у топ-10 OWASP [19] (рис 2.2).

Найсучасніший сканер веб-додатків у складі Burp Suite точно відображає вміст і функціональні можливості, автоматично обробляючи сеанси, зміни стану, нестабільний вміст і форми аутентифікації додатків.

Burp Scanner включає повний механізм аналізу JavaScript, який використовує комбінацію статичних (SAST) і динамічних (DAST) методів для виявлення вразливостей безпеки в клієнтському JavaScript, наприклад міжсайтових сценаріїв на основі DOM.

Burp став першим у використанні інноваційних позасмугових методів (OAST) для доповнення традиційної моделі сканування. Технологія Burp Collaborator дозволяє Burp виявляти вразливості на стороні сервера, які повністю невидимі в зовнішній поведінці програми, і навіть повідомляти про вразливості, які запускаються асинхронно після завершення сканування (рис. 2.2).

Рис. 2.2 Приклад запиту в Burp Collaborator

Технологію Burp Infiltrator можна використовувати для виконання інтерактивного тестування безпеки додатків (IAST) шляхом інструментування цільових додатків для надання зворотнього зв’язку Burp Scanner в реальному часі, коли його корисні навантаження досягають небезпечних API в програмі.

Логіка сканування Burp постійно оновлюється за допомогою вдосконалень, щоб гарантувати, що він може знаходити останні вразливості та нові крайні випадки наявних уразливостей. За останні роки BurpSuite був першим сканером, який за спиянням дослідницької групи Burp виявляв нові вразливості, зокрема ін’єкції шаблонів та отруєння веб-кешу.

За замовчуванням, якщо ваш комп’ютер підтримує його, Burp використовуватиме вбудований браузер Chromium для всієї навігації цільовими веб-сайтами та додатками. Цей підхід надає кілька основних переваг, дозволяючи Burp Scanner працювати з більшістю клієнтських технологій, які підтримують сучасні браузери (рис. 2.3).

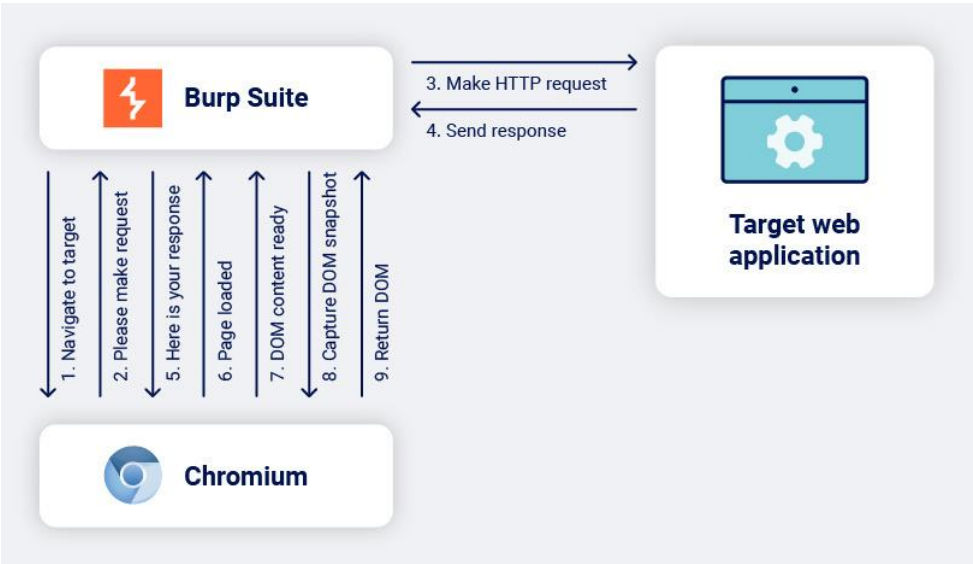

Рис. 2.3 Архітектура сканера

Однією з ключових переваг є можливість ефективно сканувати важкий JavaScript. Деякі веб-сайти мають навігаційний інтерфейс, який динамічно створюється за допомогою JavaScript. Хоча цього вмісту немає в необробленому HTML, Burp Scanner може використовувати вбудований браузер для завантаження сторінки, виконання будь-яких скриптів, необхідних для створення інтерфейсу користувача, а потім продовження сканування в звичайному режимі.

Вбудований браузер також дозволяє Burp Scanner обробляти випадки, коли веб-сайти змінюють запити на льоту за допомогою обробників подій JavaScript. Сканер може ініціювати ці події та виконувати відповідний сценарій, змінюючи будь-які запити за потреби. Наприклад, веб-сайт може використовувати JavaScript для створення нового токена CSRF після події та додавання його до наступного запиту. Burp Suite може взаємодіяти з елементами, на які можна натиснути за допомогою обробників подій JavaScript.

Усі вразливі місця, про які повідомляється, містять детальні спеціальні попередження. Вони включають повний опис проблеми та покрокові поради щодо усунення проблеми (рис 2.4). Консультативні формулювання динамічно генеруються для кожної окремої проблеми з точним описом будь-яких особливостей або моментів усунення.

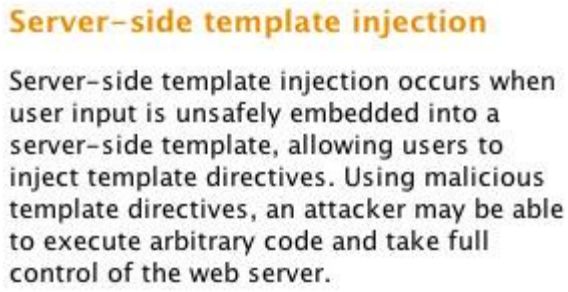

Рис 2.4 Приклад опису вразливості

Архітектура Enterprise Edition більш складна. Burp Suite Enterprise Edition містить такі компоненти [20]:

1. Корпоративний сервер – координує роботу між іншими компонентами, керує плануванням сканування та виконує оновлення програмного забезпечення.

2. Агенти – вони виконують сканування за допомогою вбудованого екземпляра BurpScanner. Агенти можуть бути розподілені між кількома машинами, і пул агентів може зростати нескінченно.

3. Веб-сервер – надає інтерфейс користувачам через веб-інтерфейс і REST API. Веб-сервер встановлено на тій же машині, що й сервер Enterprise.

4. База даних – забезпечує постійне зберігання даних конфігурації та результатів сканування. Існує пакетна база даних, яка підходить для цілей тестування та багатьох виробничих випадків, або існує можливість використовувати власну зовнішню базу даних, якщо потрібно.

На схемі нижче показано різні компоненти програмного забезпечення BurpSuite Enterprise Edition та зв’язки між ними (рис. 2.5).

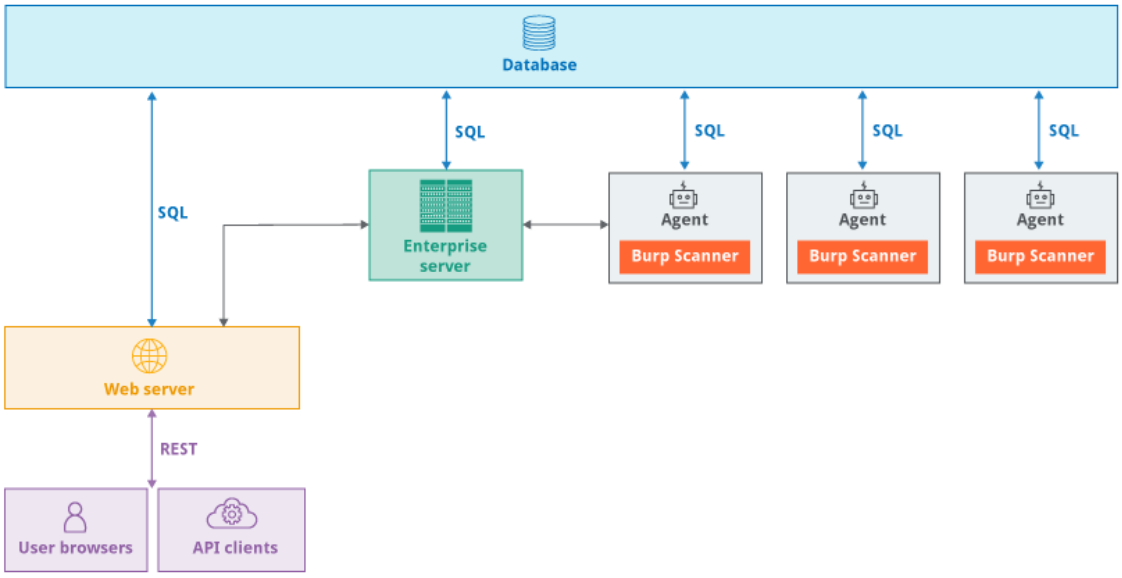

Рис. 2.5 Компоненти BurpSuite Enterprise [20]

Burp Suite Enterprise Edition має надзвичайну масштабованість. Для полегшеного використання можна запускати всі компоненти на одній машині, включно з пакетною базою даних. На машині зі значними ресурсами ця конфігурація повинна підтримувати до 10 одночасних сканувань. На схемі нижче показано розгортання на одній машині (рис. 2.6).

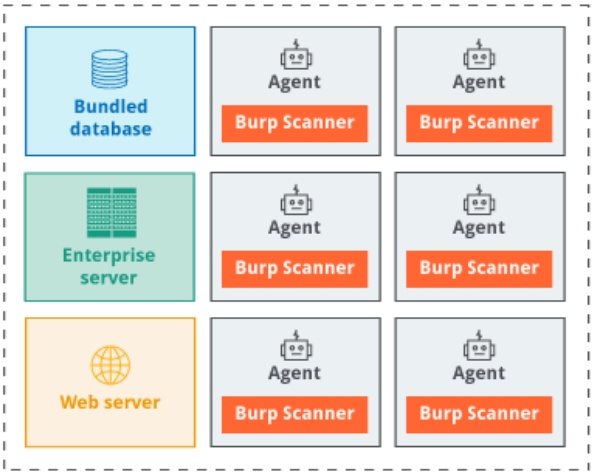

Рис. 2.6 Приклад розгортання на одній машині [20]

З іншого боку, можна запускати агенти на великій кількості машин і використовувати власну зовнішню базу даних для зберігання. Це дає змогу масштабувати кількість одночасних сканувань до необмеженої кількості та використовувати будь-яку наявну інфраструктуру баз даних. На схемі нижче показано розгортання на кількох машинах із зовнішньою базою даних та агентами (рис. 2.7).

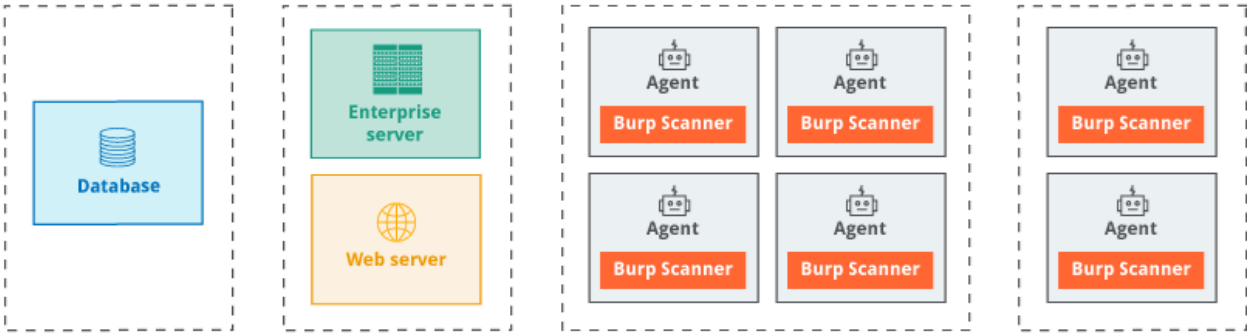

Рис. 2.7 Приклад розгортання із зовнішньою базою даних на кількох машинах [20]

Кожну машину агента та, за бажанням, машину сервера Enterprise, можна налаштувати на запуск кількох логічних агентів. Кожен логічний агент може бути зайнятий виконанням одного сканування в будь-який момент часу.

Архітектура BurpSuite Enterprise Edition дозволяє задовольнити потреби від невеликої організації з кількома веб-сайтами або розробниками, до величезної організації з тисячами веб-сайтів і безліччю команд розробників.

1.3. Вимоги до системи для інсталяції програмного комплекса Burp Suite

Для Burp Suite Professional та Community Edition потрібен комп'ютер із встановленим офіційним середовищем виконання Java (64-розрядна версія). JRE доступні для різних популярних операційних систем, включаючи Windows, Linux і Mac OS X [21].

Для найкращого використання Burp Suite Professional рекомендовано використовувати комп'ютер із принаймні 8 ГБ пам'яті та 2 ядрами центрального процесора. Якщо виконується великий обсяг роботи або тестуються великі чи складні програми, може знадобитися більше пам'яті.

Burp Suite Enterprise Edition потребує 64-розрядної машини під керуванням сучасної операційної системи Windows, Linux або MacOS. Системні вимоги для комп'ютерів із запущеним Burp Suite Enterprise Edition сильно залежать від багатьох факторів. Ці фактори включають [22]:

скільки одночасних сканувань необхідно виконати;

характер та обсяг програми, що сканується;

кількість машин для використання;

чи використовується функція вбудованого браузера для сканування;

кількість згенерованих звітів;

кількість активних користувачів веб-інтерфейсу та API Enterprise Edition.

Як результат, системні вимоги, наведені в цьому розділі, є загальними рекомендаціями, які повинні забезпечити задовільну продуктивність для більшості випадків використання.

Незалежно від бажаного методу розгортання застосовуються такі вимоги:

1. Усі машини, на яких встановлені компоненти Burp Suite Enterprise Edition, повинні мати 64-розрядну архітектуру.

2. Розташування диска (настроєне під час інсталяції) має знаходитися на локально підключеному сховищі, а не в файловій системі мережі. Вільний простір необхідний не тільки для попередньої установки. Дисковий простір використовується для зберігання ефемерних даних під час сканування та оновлення продукту.

3. Щоб отримати максимальну віддачу від Burp Suite Enterprise Edition, потрібно призначити кілька агентів на машину. У багатьох випадках це вимагатиме використання виділеного сервера або віртуальної машини серверного класу, щоб відповідати системним вимогам.

Можна використовувати опцію розгортання на одній машині, запустивши всі компоненти Burp Suite Enterprise Edition на одній машині, або розповсюдити агентів на кількох комп'ютерах.

Якщо необхідно розгорнути Burp Suite Enterprise Edition на комп'ютері на базі Linux, необхідно переконатися, що достатній обсяг пам'яті виділено як простір підкачки. Це може допомогти підвищити стабільність сканування, надаючи запасний варіант у випадках, коли сканування використовує великий обсяг пам'яті.

Рекомендовано виділити такий же обсяг пам'яті, що й доступна оперативна пам'ять, плюс додаткові 2 ГБ. Наприклад, для машини з 32 ГБ оперативної пам'яті слід створити файл підкачки на 34 ГБ.

Для локальних установок, ймовірно, за замовчуванням буде доступний достатній простір підкачки. Однак деякі хмарні служби вимагають створення відповідного файлу підкачки вручну. Для отримання конкретних інструкцій щодо того, як це зробити, необхідно звернутися до документації постачальника хмарних послуг.

Якщо необхідно розгорнути сервер Enterprise та запустити сканування на одній машині, застосовуються наступні вимоги (рис.2.8).

	Вільне місце на диску	RAM (BPS увімкнено)	RAM (BPS вимкнено)	Ядра процесора
Установка основи	10 ГБ	16 Гб	16 Гб	4
За одночасне сканування	+ 20 ГБ	+ 8 ГБ	+ 4 ГБ	+ 4
Всього за 1 сканування	30 ГБ	24 ГБ	20 ГБ	8
Всього для 2 одночасних сканування	50 ГБ	32 ГБ	24 ГБ	12
Всього для 5 одночасних сканувань	110 ГБ	56 ГБ	36 ГБ	24
Всього для 10 одночасних сканувань	210 ГБ	96 ГБ	56 ГБ	44

Рис. 2.8 Вимоги до уствновки на одній машині

Якщо необхідно запустити сканування на окремих машинах агентів, а не на сервері Enterprise, застосовуються наступні вимоги. Комп’ютер, на якому запускається сервер Enterprise, потребує:

вільне місце на диску: 10 ГБ;

оперативна пам'ять: 16 ГБ;

ядра процесора: 4.

Кожній машині агента потрібна певна кількість ресурсів для встановлення та запуску програмного забезпечення, яке керує скануваннями. Тоді потрібні додаткові ресурси залежно від того, скільки одночасних сканувань необхідно запустити (рис. 2.9).

Якщо також потрібно використовувати вбудовану базу даних, потрібно залишити місце для даних, які вона накопичує.

Обсяг даних, який може накопичити Burp Suite Enterprise Edition, залежить від кількості сканувань, які виконуються, і кількості знайдених проблем.

	Вільне місце на диску	RAM (BPS увімкнено)	RAM (BPS вимкнено)	Ядра процесора
Установка основи	10 ГБ	2 ГБ	2 ГБ	2
За одночасне сканування	+ 20 ГБ	+ 8 ГБ	+ 4 ГБ	+ 4
Всього за 1 сканування	30 ГБ	10 ГБ	6 ГБ	6
Всього для 2 одночасних сканування	50 ГБ	18 ГБ	10 ГБ	10
Всього для 5 одночасних сканувань	110 ГБ	42 ГБ	22 ГБ	22
Всього для 10 одночасних сканувань	210 ГБ	82 ГБ	42 ГБ	42

Рис. 2.9 Необхідні ресурси для проведення сканувань

На наступному рисунку наведено приблизну кількість даних, яка, ймовірно, буде накопичена (рис. 2.10).

Кількість сканувань	Зберігання даних
1000	500 МБ
10 000	5 ГБ
100 000	50 ГБ

Рис. 2.10 Вимоги до бази даних

Наступні типи та версії зовнішніх баз даних були перевірені та повністю підтримуються (рис.2.11).

Type	Supported versions
MariaDB	10.2, 10.3, 10.4, 10.5
Microsoft SQL server	2012, 2014, 2016, 2017
MySQL	5.7, 8
Oracle	12.2, 18c, 19c
PostgreSQL	9.4, 9.5, 9.6, 10, 11, 12, 13

Рис. 2.11 Підтримувані бази даних

Висновки до розділу 2

У даному розділі було досліджено програмний комплекс Burp Suite та визначено його основні можливості щодо покращення можливості виявлення вразливостей веб-додатка сучасного підприємства.

Вдалося визначити основні функції даного програмного комплексу. Вони дозволяють проводити пошук та аналіз вразливостей в системі. На основі отриманих даних можна знайти нові вразливості або оцінити та усунути вже виявлені.

Було встановлено, що важливою складовою Burp Suite є сканер вразливостей, що дозволяє відстежувати вразливості веб-додатка для оцінки можливості експлуатації.

Наостанок досліджено архітектуру та можливості для інсталяції програмного комплексу Burp Suite. Можна використовувати опцію розгортання на одній машині, запустивши всі компоненти Burp Suite на одній машині, або розповсюдити агентів Enterprise Edition на кількох комп’ютерах.

3 РОЗРОБЛЕННЯ ВАРІАНТА ТЕХНОЛОГІЇ УПРАВЛІННЯ ВРАЗЛИВОСТЯМИ ВЕБ-ДОДАТКУ СУЧАСНОГО ПІДПРИЄМСТВА НА БАЗІ BURP SUITE

3.1. Розроблення варіанта алгоритма управління вразливостями веб-додатку

Впровадження процесу управління вразливістю дозволяє отримувати об'єктивну оцінку стану захищеності інформаційної системи та веб-додатків, завчасно виявити та усунути вразливості, автоматизувати контроль відповідності вимогам регуляторів, міжнародним та галузевим стандартам.

Зі зростанням кількості загроз веб-додатків багато організацій досі не мають належно організованого процесу керування виправленням та управлінням вразливостями. Це може залишити критичні компоненти веб-додатку невиправленими та вразливими протягом значного періоду часу. Зокрема, наступного циклу виправлення або до застосування патча вручну.

NIST Cybersecurity Framework – це набір рекомендацій щодо пом’якшення організаційних ризиків кібербезпеки, опублікований Національним інститутом стандартів і технологій США (NIST) на основі існуючих стандартів, рекомендацій і практик.

NIST CSF складається з трьох компонентів: основні компоненти фреймворку, рівні реалізації та профілі. Основні компоненти розділені на п’ять сфер кібербезпеки [23]:

ідентифікація;

захист;

виявлення;

відповідь;

відновлення.

Кожна з цих областей включає заходи нижчого рівня для пом'якшення ризику кібербезпеки, а також поділяється на категорії та підкатегорії, які включають описи провідних практик безпеки та плани реагування на інциденти, а також методи, за допомогою яких найкраще отримати успішне відновлення (рис.3.1).

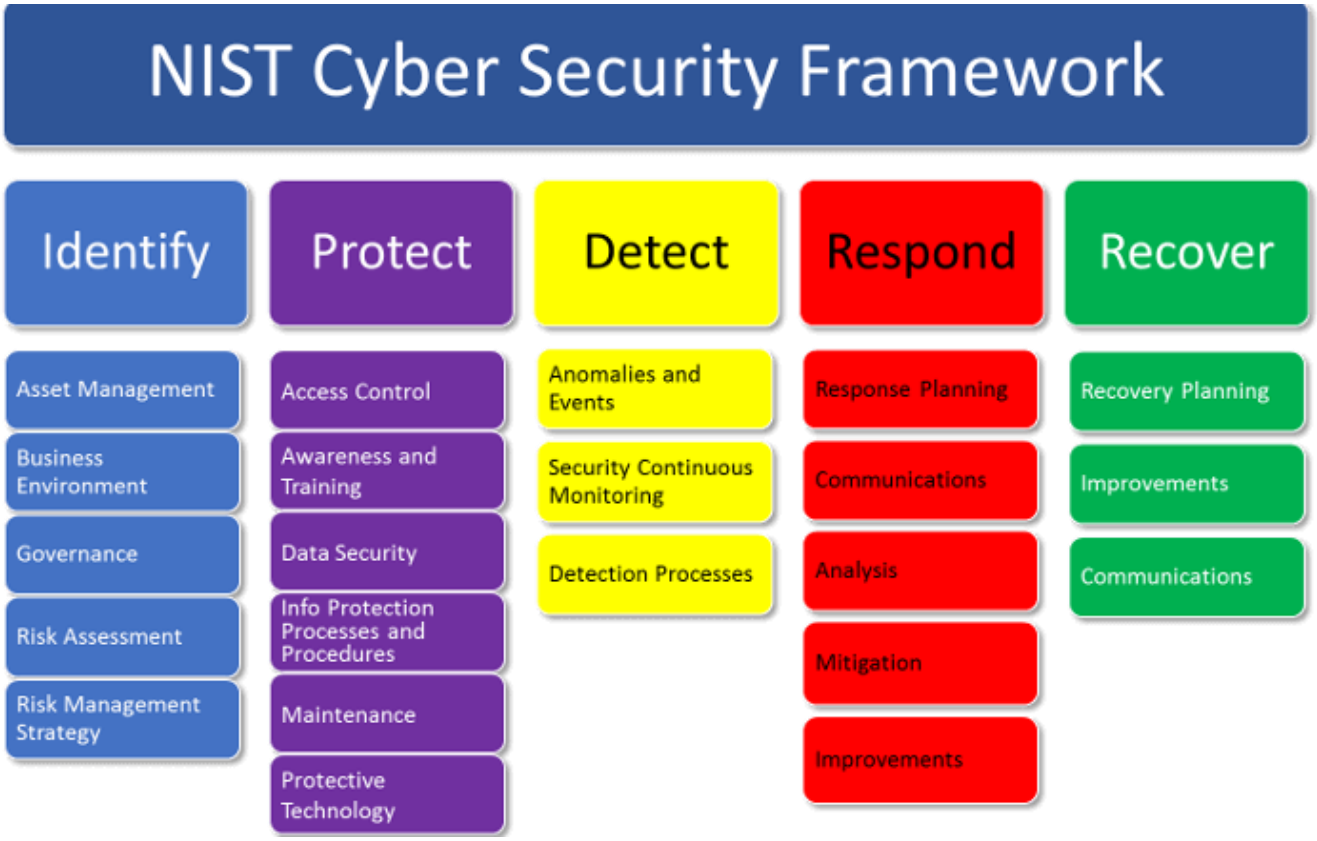

Рис. 3.1 NIST Cybersecurity Framework

На додаток до CSF, NIST випустив понад 200 спеціальних публікацій, які охоплюють багато аспектів управління ризиками кібербезпеки: контроль доступу до ідентифікаційних даних, управління захисними технологіями, реагування на подію або інцидент у сфері кібербезпеки та багато іншого.

Серед найбільш широко використовуваних публікацій NIST є NIST 800-53.

Процес виявлення вразливостей веб-додатку на основі стандарту NIST SP 800-53 надає командам безпеки інформацію про виявлені вразливості, прогрес пом'якшення та можливості зниження ризику.

Щоб забезпечити відповідність нормам, таким як NIST SP 800-53, потрібні інструменти сканування вразливостей веб-додатків і керування ними з можливістю створювати звіти, готові до аудиту, щоб допомогти програмі інформаційної безпеки дотримуватися найкращих практик, постійно тестувати засоби контролю безпеки та усувати критичні вразливості.

Відповідно до NIST SP 800-53 можна визначити три основні цілі для процесу управління вразливостями, зокрема веб-додатку [24]:

1. Своєчасне виявлення вразливостей**.** Чим раніше виявляється вразливість, тим більше буде часу, щоб виправити її або, принаймні, попередити виробника про ситуацію, що зменшить вікно можливостей потенційного зловмисника.

2. Оцінка вразливості веб-додатку. Не всі вразливості однаково впливають на певний веб-додаток. Потрібно провести оцінку ризику, щоб визначити та встановити пріоритети тих вразливостей, які є більш критичними для активів і бізнесу.

3. Проведення відповідних заходів з урахуванням супутніх ризиків. Після того, як визначено найбільш критичні вразливі місця, потрібно подумати про дії та розподіл ресурсів для боротьби з ними. Найбільш обережна форма – це враховувати пов’язаний з вразливостями рівень ризику.

Фреймворк NIST – це глобальний стандарт безпеки, який надає найкращі практичні рішення для підтримки засобів контролю, таких як управління вразливостями. Він встановлює керівні принципи та загальні принципи для створення, впровадження, підтримки та вдосконалення систем управління інформаційною безпекою. Кожен контроль безпеки та ціль, передбачені стандартом, можуть бути пристосовані до конкретних бізнес-цілей та цілей регулювання та допомагають підтримувати загальну відповідність.

Традиційні інструменти сканування вразливостей надають велику кількість інформації, яка може ускладнити оцінку та визначення пріоритету ризику. Сканування уразливостей, оцінка, звітування дають знімок стану безпеки веб-додатку. Тривалий період часу між скануваннями уразливостей може надати зловмисникам можливість отримати доступ до мережі організації. Організація

спочатку повинна мати комплексну програму управління активами, щоб ідентифікувати існуючі компоненти та складові веб-додатку.

3.2. Порядок застосування технології управління вразливостями веб-додатку з програмним комплексом Burp Suite

Під час циклу виявлення виконуються завдання, які істотно підтримують тести на вразливість. Основна діяльність зосереджена на визначенні та уточненні цілей, підготовці інструментів і перевірки їх цілісності, проведенні тестів і перевірки результатів.

Відділ інформаційної безпеки часто не в курсі всіх активів, що є в мережі. Особливо якщо веб-додаток великий, з розгалуженою системою. Сканування лише відомої інфраструктури неминуче дасть неповну інформацію. Неповна інформація означає велику можливість пропустити критичну вразливість і дорого за це заплатити.

Не можна захистити невідомі активи. Якщо не має повної картини того, з чого складається веб-додаток, не буде можливості його якісно захищати. Сучасна інфраструктура складна і постійно змінюється кількісно та якісно.

Тепер IT-інфраструктура, зокрема веб-додатку базується не тільки на стеку класичних технологій (робочі станції, сервери, віртуальні машини), а й на нових – контейнерах, мікросервісах. Служби інформаційної безпеки всіляко уникають останніх, оскільки їм дуже складно працювати з ними за допомогою існуючих наборів інструментів, які складаються переважно зі сканерів. Проблема полягає в тому, що будь-який сканер не може покрити всю інфраструктуру. Щоб сканер достукався до будь-якого вузла в інфраструктурі, потрібно щоб збіглося відразу кілька факторів. Актив має бути всередині периметра організації на момент сканування. Сканери повинні мати мережеві доступи до активів, їх облікових записів, щоб зібрати повну інформацію. Якщо використовуються ресурси хмарних провайдерів, то звідти також потрібно збирати інформацію про активи та конфігурації.

Результати будь-якої програми оцінки безпеки, будь то оцінка вразливості, тест на проникнення, обмежені її обсягом. Не можна усунути ризики, пов’язані з конкретним ресурсом, який не входить в сферу оцінки, оскільки вони просто невідомі. Відповіддю на цю проблему є відстеження активів за допомогою безперервного виявлення активів.

Деякі з активів, які можна знайти у веб-додатку – це домени, субдомени, сторонні служби, публічні та приватні IP-адреси, хмарне сховище тощо.

Процес виявлення активів простий, але нелегкий. Потрібно почати з деякої початкової інформації про організацію, наприклад, назва компанії, домен(и), IP адрес(и) тощо; мати список джерел даних, зіставлених із початковим типом, а потім витягувати з них інформацію. Крім того, цей процес повинен мати рекурсивний характер, щоб перевірити, чи може нова ідентифікована інформація стати початковою інформацією для іншого джерела.

Однією з основних проблем виявлення активів є рівень довіри до автентичності та релевантності ідентифікованої інформації. Будь-яке джерело інформації, яким безпосередньо володіє чи керує організація, має високий рівень довіри, з іншого боку, стороннє джерело може вимагати багаторазових перевірок для перевірки інформації. Одним із таких джерел, на яке можна покладатися, є веб-сайти, які належать цільовій організації. Звичайно, необхідно провести подальшу перевірку активів, визначених із цього джерела, але можна мати значний рівень довіри до них.

Найпростіший спосіб виявляти активи це – шукати конкретні закономірності у відповідях сторінок веб-додатку.

Однак робити це для одного або кількох веб-сайтів було б дуже втомливо. Щоб автоматизувати це, потрібен сканер, який аналізує всі відповіді на регулярні вирази, або доповнення для існуючих інструментів. Оскільки Burp Suite є інструментом тестування для більшості людей, які займаються веб-безпекою, для ного створене розширення Burp Suite «Asset Discover» .

Розширення діє як пасивний сканер, який аналізує відповідь сторінок, які знаходяться в області тестування, і постійно стежить за активом. Ці активи

ідентифікуються та класифікуються на основі шаблонів регулярних виразів для різних видів активів [25] (рис. 3.2).

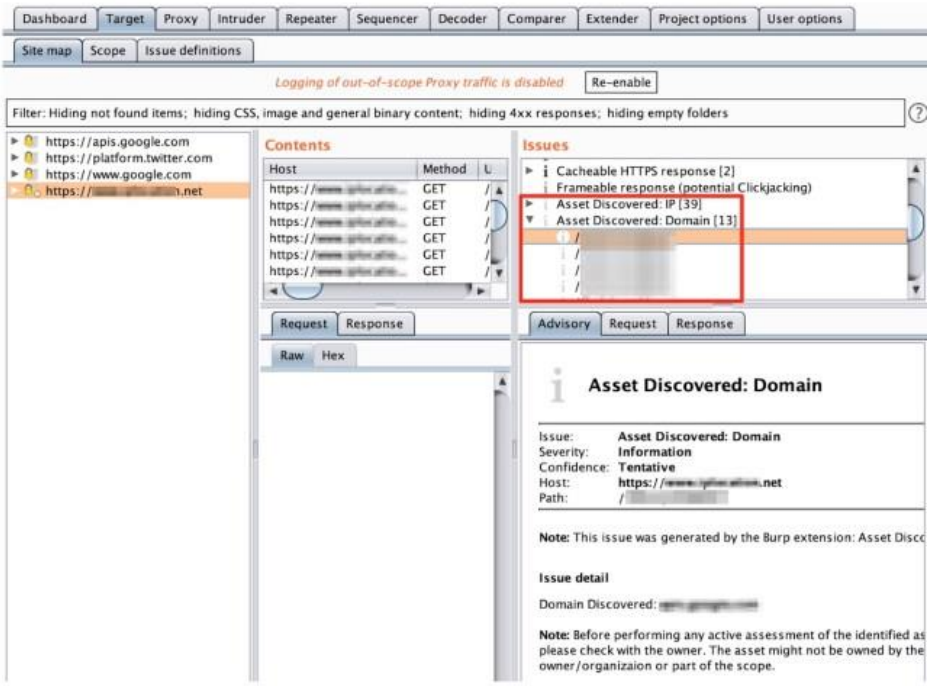

Рис. 3.2 Результат пошуку активів розширенням

Інструмент Target дає огляд вмісту та функціональних можливостей цільового веб-додатку, а також дозволяє керувати ключовими частинами робочого процесу тестування [26].

Спочатку треба вручну відстежити вміст додатку, який тестується. Для цього треба виконати такі дії:

необхідно запустити вбудований браузер Burp або налаштувати зовнішній браузер для роботи з BurpSuite Proxy;

вимкнути перехоплення проксі-сервера та переглянути всю програму вручну;

перейти за кожним посиланням, надсилати кожну форму, проходити кожен багатоетапний процес і ввійти у всі захищені зони.

Цей процес зіставлення вручну заповнить карту цільового сайту всім вмістом, через запити проксі-сервера, а також (через пасивне сканування в реальному часі) будь-яким іншим вмістом, який можна визначити з відповідей програми (за посиланнями, формами тощо). Цей ручний процес зіставлення

дозволить створити досить повний запис на карті сайту всього видимого вмісту додатку, а також повністю ознайомить із програмою.

Для деяких випадків використання автоматизованого сканера BurpSuite є кращим за ручну перевірку, оскільки він фіксує навігаційні шляхи через програму таким чином, що дозволяє Burp Scanner автоматично підтримувати сеанс під час аудиту програми. З іншого боку, відображення вручну дозволяє користувачеві керувати процесом, уникаючи потенційно небезпечних функцій і перевіряючи, що навігаційні дії мають очікувані результати. Вибір ручного та автоматичного відображення дуже залежить від характеру програми та передбачуваного використання результатів.

За замовчуванням сканер BurpSuite переміщається по цільовій програмі за допомогою вбудованого браузера, натискаючи посилання та надаючи дані, де це можливо. Він створює карту вмісту та функціональних можливостей програми у вигляді орієнтованого графіка, що представляє різні місця в програмі та зв’язки між ними (рис. 3.3).

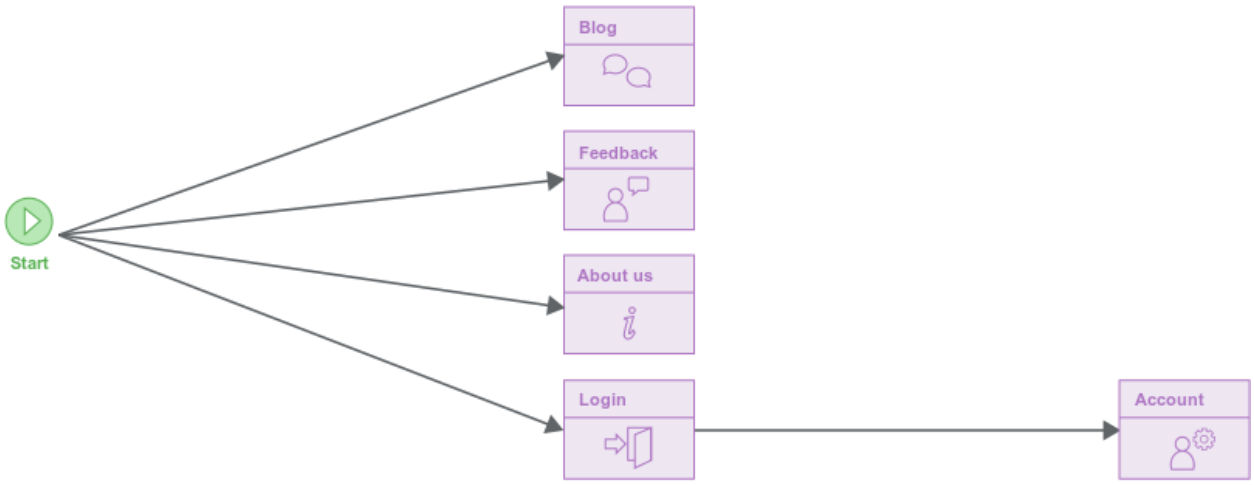

Рис. 3.3 Архітектура компонентів веб-додатка [26]

Сканер не робить припущень щодо структури URL-адрес, яку використовує програма. Розташування ідентифікуються (і пізніше ідентифікуються повторно) на основі їх вмісту, а не URL-адреси, за якою вони були доступні. Це дає змогу сканеру надійно обробляти сучасні програми, які розміщують ефемерні дані, такі

як токени CSRF або розпізнавачі кешу, в URL-адреси. Навіть якщо вся URL-адреса в кожному посиланні змінюється щоразу, сканер все одно створює точну карту.

Цей підхід також дозволяє сканеру обробляти програми, які використовують ту саму URL-адресу, щоб досягти різних місць на основі стану програми або взаємодії користувача з нею.

Поки сканер переміщається та нарощує охоплення цільової програми, він відстежує кінцеві точки в графіку. Вони представляють посилання (або інші навігаційні переходи), які були помічені в програмі, але ще не відвідані. Але сканер ніколи не перескакує до очікуваного посилання і не відвідує його поза контекстом. Замість цього він або здійснює навігацію безпосередньо зі свого поточного розташування, або повертається до початкового розташування та рухається звідти. Це якомога точніше повторює дії звичайного користувача, який переглядає веб-сайт (рис. 3.4).

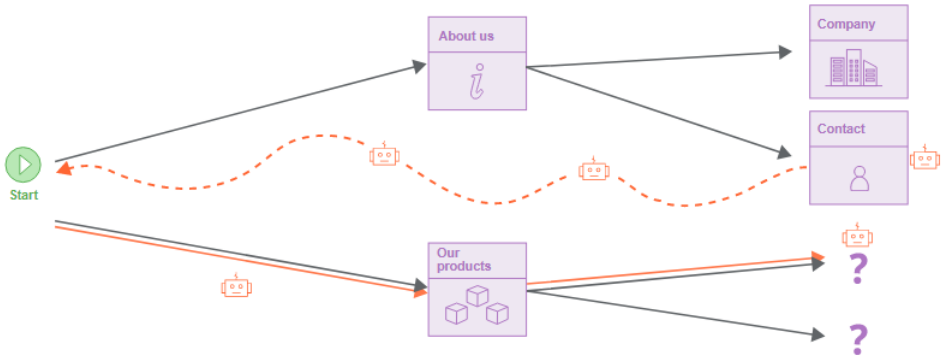

Рис. 3.4 Архітектура сканування [26]

Повністю визначивши вміст веб-додатка та оцінивши поверхню атаки можна керувати процесом детального тестування вразливостей із карти сайту [27]:

1. Вибрати гілки та елементи та використати контекстне меню, щоб надіслати їх іншим інструментам Burp Suite для виконання певних завдань, таких як автоматичне сканування вразливостей за допомогою Burp Scanner, fuzzing за допомогою Burp Intruder або ручного тестування за допомогою Burp Repeater.

2. Можна знову надіслати запити веб-додатку в іншому контексті сеансу та порівняти карти сайту, щоб допомогти визначити вразливості контролю доступу.

3. Можна шукати у гілках карти сайту конкретні вирази, скрипти та коментарі

Периметр сканування визначає цільові активи та визначає тип перевірки безпеки який буде проведено. Поширені варіанти:

мережеве сканування: сканування з обліковими данихми проти сканування без облікових даних;

сканування програм: статичний аналіз коду (SAST) проти динамічного сканування (DAST);

тести безпеки ділової електронної пошти або соціальної інженерії (SE).

Сканування мережі підходять для виявлення відсутніх патчів, неправильних конфігурацій, і облікових даних за замовчуванням на веб-серверах і мережевих пристроях.

Сканування з обліковими даними зазвичай дає точніші результати. Важливо використовувати сканування без облікових даних для веб-додатків доступних в публічній мережі.

Важливо зауважити, коли сканування розгортається вперше (і це може включати перший раз для певної групи активів), необхідно перевірити справність активів до і після.

У той час як SAST аналізує якість коду, DAST моделює реальні атаки. DAST може завдати певної шкоди веб-додатку та серверу. Було б розумно уникати запуску DAST у виробничому середовищі.

Тести безпеки ділової електронної пошти або фішингові тести – це спосіб залучити критичне мислення. Тести соціальної інженерії не дуже поширені, але визнані дуже ефективним способом підвищення самосвідомості співробітників. Необхідно зауважити, що перепідготовці має передувати формальне навчання з інформаційної безпеки.

Після визначення активів та цілей можна перейти безпосередньо до процесу сканування вразливостей. Рекомендовано на цьому етапі приділяти увагу не тільки вразливостям в додатку, але й помилкам у конфігураціях, які теж можуть бути

вразливими. Тут необхідний агентський метод збирання інформації. Сканери можна використовувати для оцінки безпеки периметра.

Burp Scanner автоматизує завдання сканування веб-сайтів на предмет вмісту та вразливостей. Залежно від конфігурації Сканер може сканувати програму, щоб виявити її вміст і функціональність, а також перевіряти програму для виявлення вразливостей. За замовчуванням усі сканування використовуватимуть вбудований браузер, щоб забезпечити максимальне покриття. Імпорт повних послідовностей входу навіть дозволяє Burp Scanner працювати зі складнішими механізмами входу, включаючи єдиний вхід.

Сканування можна запустити різними способами:

1. Сканування з певних URL-адрес виконує сканування шляхом сканування вмісту в межах однієї або кількох наданих URL-адрес і, за бажанням, аудит просканованого вмісту. Для цього необхідно перейти на панель інструментів Burp і натиснути кнопку «Нове сканування». Це відкриє панель запуску сканування, яка дозволить налаштувати деталі сканування.

2. Сканування вибраних елементів дозволяє виконувати лише сканування (без автоматичного пошуку вмісту веб-додатку) конкретних запитів HTTP. Для цього необхідно вибрати один або кілька запитів у будь-якому місці Burp і виберати сканування з контекстного меню. Це відкриє панель запуску сканування, яка дозволить налаштувати деталі.

3. Живе сканування може виконувати сканування в реальному часі для автоматичного сканування запитів, які обробляються іншими інструментами Burp, такими як інструменти проксі або повторювач. Можна точно налаштувати, які запити обробляються, і чи потрібно їх сканувати для виявлення вмісту чи аудиту на наявність уразливостей. Для цього необхідно перейти на панель інструментів Burp і натисніть кнопку «Нове завдання в реальному часі». Це відкриє панель запуску сканування в реальному часі, яка дозволить налаштувати деталі завдання.

4. Миттєве сканування можете запустити миттєве активне або пасивне сканування з контекстного меню. Це означає, що можна швидко перевірити наявність уразливостей, не відкриваючи панель запуску сканування. Можна

отримати доступ до цих параметрів, клацнувши правою кнопкою миші на запит. Крім того, можна налаштувати гарячі клавіші для апуску миттєвого сканування.

Можна контролювати хід і результати сканування різними способами:

1. На панелі інструментів Burp відображаються показники про виконання кожного завдання, а в журналі діяльності відображаються проблеми, про які повідомляють усі запущені сканування.

2. Можна відкрити вікно з деталями завдання для окремого сканування, щоб переглянути журнал діяльності лише для обраного сканування, а також детальний перегляд елементів аудиту для відповідних завдань.

3. Карта цільового сайту показує весь вміст і проблеми, які були виявлені, упорядковані за доменом та URL-адресою.

Після проведення сканування можна переглянути результати, отртмати опис знайдених вразливостей та рекомендацій щодо їх усунення для подальшого аналізу та оцінки (рис. 3.5).

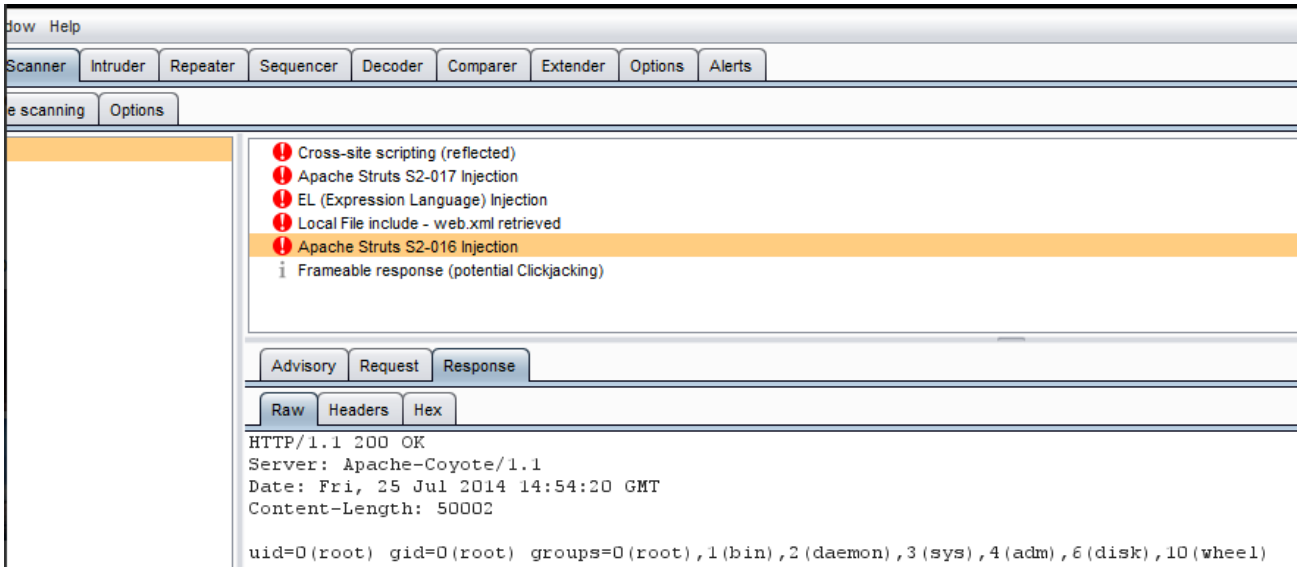

Рис. 3.5. Приклад результату сканування

Оцінка вразливостей відіграє важливу роль в управлінні вразливими місцями і використовується для визначення потенційного ризику та впливу, який вразливість може мати на мережу або комп'ютерну систему. Наприклад, популярна загальна система оцінки вразливості (CVSS) присуджує бали вразливості на основі її складових, доступності та відтворюваності [28].

Загальна система оцінки вразливостей Вперше представлена в 2005 році, Common Vulnerability Scoring System (або CVSS) є дуже популярним фреймворком для оцінки вразливостей і має три основні ітерації. На даний момент поточною версією є CVSSv3.1 (версія 4.0 наразі знаходиться в розробці). Оцінка по суті визначається деякими з наступних факторів (але їх набагато більше):

1. Наскільки легко використати вразливість?

2. Чи існують для цього експлойти?

3. Як ця вразливість впливає конфіденційність, цілісність, доступність? Насправді, існує так багато змінних, що доведеться використовувати калькулятор, щоб визначити оцінку за допомогою цього фреймворку. Уразливості надається класифікація (з п’яти варіантів) залежно від присвоєної оцінки (рис. 3.6).

Rating	Score
None	0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Рис. 3.6. Метрика CVSS

Переваги CVSS:

CVSS існує вже давно.

CVSS популярний в організаціях.

CVSS — це безкоштовний фреймворк, який запроваджують і рекомендують такі організації, як NIST.

Недоліки CVSS:

CVSS ніколи не створювався для того, щоб допомогти визначити пріоритетність уразливостей, натомість просто для призначення значення серйозності.

CVSS ретельно оцінює вразливості залежно від доступного експлойту. Однак лише 20% усіх уразливостей мають доступний експлойт (Tenable., 2020).

Уразливі місця рідко змінюють оцінку після оцінювання, незважаючи на те, що можуть бути знайдені нові розробки, такі як експлойти.

Рейтинг пріоритетності вразливості VPR — це набагато більш сучасний фреймворк в управлінні вразливими місцями, розроблений Tenable, постачальником галузевих рішень для управління вразливими місцями. Цей фреймворк вважається керованим ризиками; це означає, що вразливості оцінюються з великою увагою до ризику, який вразливість представляє для самої організації, а не таких факторів, як вплив (як у CVSS). На відміну від CVSS, оцінка VPR враховує релевантність уразливості. Наприклад, ризик щодо вразливості не розглядається, якщо ця вразливість не стосується організації (тобто вони не використовують програмне забезпечення, яке є вразливим). VPR також є значно динамічним у своїй оцінці, де ризик, який може становити вразливість, може змінюватися майже щодня з віком. VPR використовує подібний діапазон оцінок, що й CVSS, однак дві помітні відмінності полягають у тому, що VPR не має категорії «Немає/інформаційна», а оскільки VPR використовує інший метод оцінки, та сама вразливість матиме інший показник за допомогою VPR, ніж при використанні CVSS (рис. 3.7).

Rating	Score
Low	0.0 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Рис. 3.7 Метрика VPR

Переваги VPR:

При розрахунку ризику VPR враховує понад 150 факторів.

VPR керується ризиками і використовується організаціями, щоб допомогти визначити пріоритети виправлення вразливостей.

Оцінки не є остаточними і дуже динамічні, а це означає, що пріоритет, який слід надати вразливості, може змінюватися в міру старіння вразливості.

Недоліки VPR:

VPR не є open-sourse, як деякі інші фреймворки керування вразливістю.

VPR не враховує тріаду CIA в тій мірі, в якій це робить CVSS; це означає, що ризик для конфіденційності, цілісності та доступності даних не відіграє великого значення для оцінки вразливостей під час використання VPR.

Звітування це один із важливих елементів усередині процесу управління вразливістю.

Ніхто не буде працювати з багатосторінковими звітами з безладним списком вразливостей та описом їх усунення. Потрібно насамперед спілкуватися з колегами та з'ясовувати, що має бути у звіті та як їм зручніше отримувати дані. Наприклад, якомусь адміністратору не потрібний докладний опис вразливості та потрібна лише інформація про патч та посилання на нього. Іншому спеціалісту важливі лише вразливості, знайдені у мережній інфраструктурі.

Під звітністю розуміються не лише паперові звіти. Це застарілий та статичний формат отримання інформації. Людина отримує звіт і ніяк не може вплинути на те, як дані будуть представлені в цьому звіті. Щоб отримати звіт у потрібному вигляді, IT-спеціаліст повинен зв'язатися зі спеціалістом з ІБ та попросити його перебудувати звіт. Час іде, з'являються нові вразливості. Замість того, щоб перекидати звіти з відділу до відділу, фахівці обох напрямків повинні мати можливість спостерігати за даними онлайн та бачити ту саму картину. Тому важливо використовувати динамічні звіти у вигляді налаштованих дашбордів.

Можна експортувати звіт про деякі або всі проблеми, створені Burp Scanner. Майстер звітів дозволяє вибрати різні параметри для звіту, як описано нижче [29].

Формат:

1. HTML – створюється звіт у форматі HTML для друку або перегляду в браузері (рис.3.8).

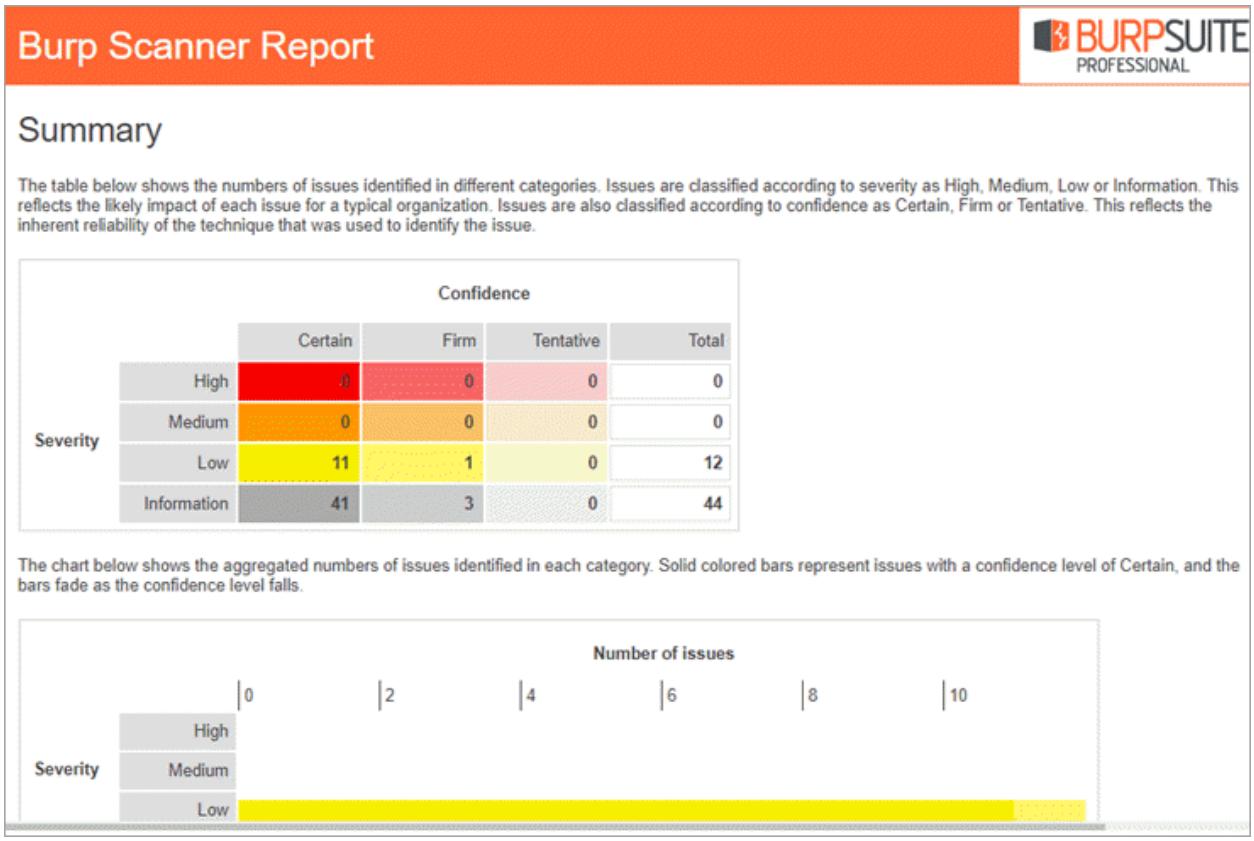

Рис. 3.8 Звіт у форматі HTML

2. XML – створює звіт у форматі XML, придатний для імпорту в інші інструменти або фреймворки для звітності. За бажанням можна кодувати HTTP-запити та відповіді в Base64 у XML виводі. Оскільки HTTP повідомлення можуть містити недруковані символи, які строго заборонені в XML-документах (навіть у блоках CDATA), використання цього параметра є кращим, оскільки він забезпечує сумісність із строгими синтаксичними аналізаторами XML.

Для звітів HTML можна вказати такі деталі:

назва звіту;

як проблеми мають бути організовані у звіті (за типом, серйозністю чи URL-адресою);

кількість рівнів деталізації, які потрібно включити в зміст;

серйозність проблеми, яку слід включити в підсумкову таблицю та гістограму.

Можна вибрати зведення або детальний звіт. Обидва типи звітів містять огляд деталей сканування, таких як включені URL-адреси, використані конфігурації сканування, тривалість сканування тощо. Вони також надають таку статистику:

проблеми за тяжкістю;

відсканованні URL-адреси та URL-адреси з проблемами;

зроблені запити;

розташування;

помилки мережі.

Обидва звіти також містять перелік типів проблем разом із відповідними URL-адресами, на яких було виявлено ці проблеми. Для кожної проблеми вказано впевненість і приблизний рівень серйозності за розсудом Burp Scanner.

Детальний звіт містить ту саму інформацію, що й підсумковий звіт. Однак він містить додатковий розділ, який містить більше інформації про кожне питання. Це включає короткий опис того, що означає тип проблеми, а також довідкову інформацію та деякі поради високого рівня щодо усунення проблеми. Також є посилання на додаткові ресурси, щоб можна було дізнатися більше про тип проблеми.

Нарешті, детальний звіт містить докази того, де було виявлено проблему. Наприклад, це може бути серія HTTP-запитів і відповідей. Для проблем на основі DOM також будуть надані результати динамічного аналізу JavaScript Burp Scanner.

Окрім можливості генерувати звіти про сканування на вимогу, також можна налаштувати Burp Suite Enterprise Edition для автоматичного надсилання підсумкових звітів про сканування. Необхідно додати список адрес електронної пошти, на які буде надсилатися підсумковий звіт, щойно завершиться сканування сайту.

У Burp Suite Enterprise Edition є кілька діаграм, які надають огляд різних показників, пов’язаних із сайтами та скануваннями. Хоча є можливість переглянути їх безпосередньо в програмі, можна завантажити їх у форматі PNG або

JPG. Це може бути корисно, наприклад, для спільного використання діаграми у звіті чи презентації (рис. 3.9)

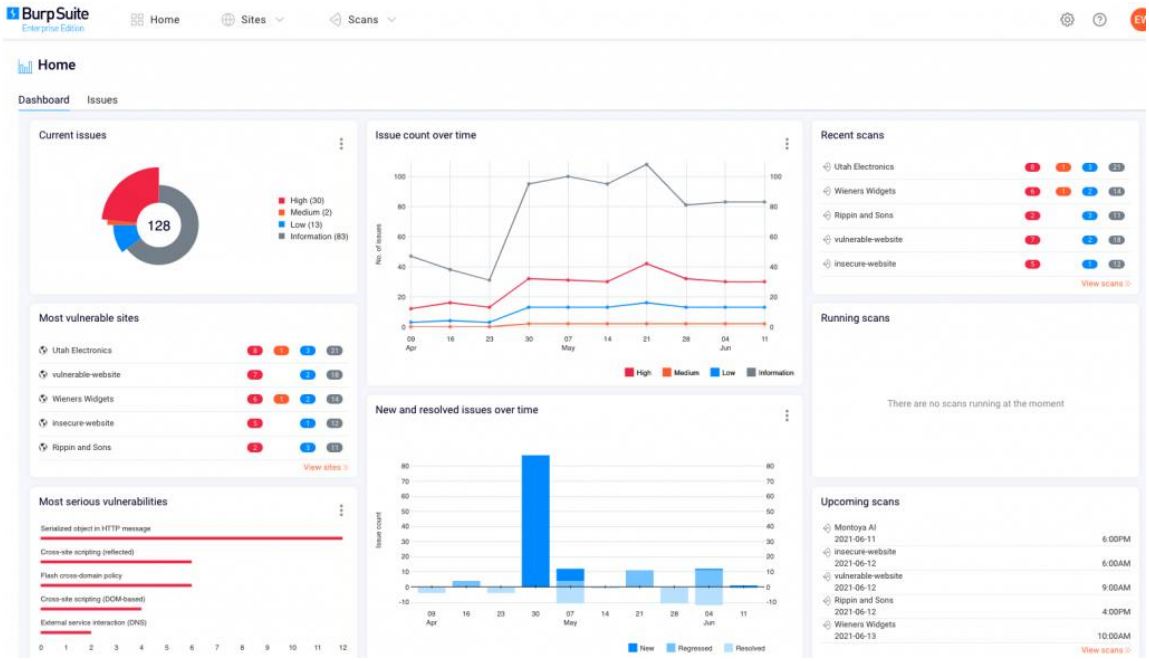

Рис. 3.9 Інструментальна панель та діаграми Burp Suite Enterprise Edition

Для кожного сканування також можна завантажити журнал подій у CSV форматі. Він містить відомості про основні події, які відбулися під час сканування, і може бути корисним для цілей налагодження (рис. 3.10).

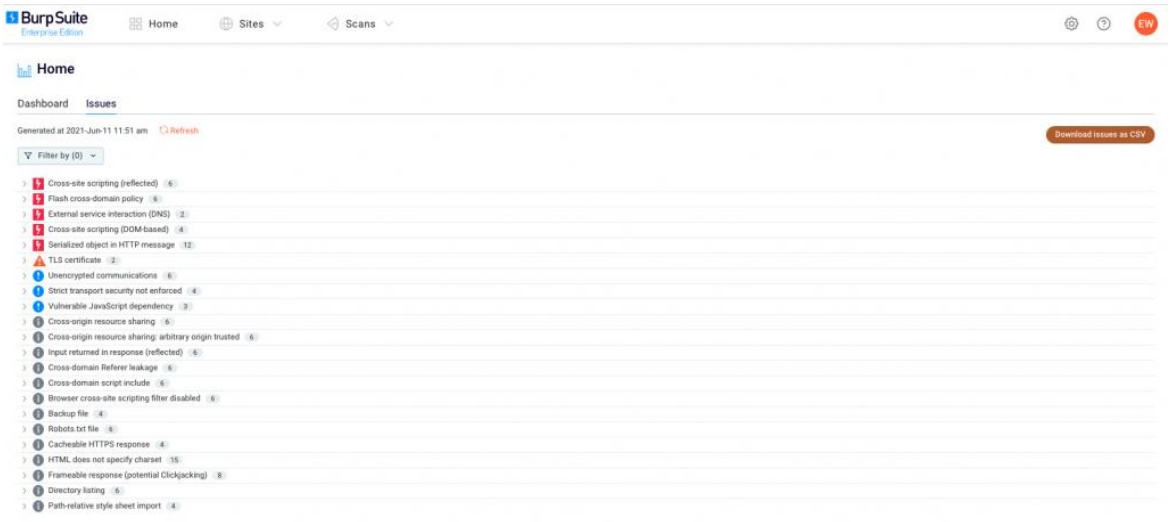

Рис. 3.10 Звіт у форматі CSV

3.3. Розроблення рекомендацій щодо застосування технології управління вразливостями веб-додатку сучасного підприємства

Як допоміжні дії для досягнення поставлених цілей, NIST надає найкращі методи, які слід враховувати під час впровадження засобів контролю безпеки, таких як система управління вразливостями веб-додатку. Стандарт пропонує [24]:

1. Інвентраризувати активи. Ефективне керування вразливими місцями залежить від того, чи організація має відповідну інформацію про інформаційні активи, складові компоненти веб-додатка, як-от виробника програмного забезпечення, версії програмного забезпечення, місця встановлення програмного забезпечення та того, хто несе відповідальність за кожну частину програмного забезпечення.

2. Визначити обов'язки. Управління вразливими місцями вимагає виконання багатьох різних заходів (наприклад, моніторинг, оцінка ризиків, виправлення тощо), тому зручно чітко визначити, хто що робить, щоб забезпечити належне відстеження активів.

3. Визначити довідкові джерела. Сайти виробників, документація, посібники спеціалізовані форуми та групи за особливими інтересами мають бути у списку джерел інформації, з якою можна ознайомитися щодо новин, пов'язаних із вразливими місцями та заходами для виправлення.

4. Справлятися з вразливими місцями за допомогою визначених процедур. Незалежно від терміновості боротьби з уразливістю, важливо обробляти її структуровано. Управління змінами або процедури реагування на інциденти слід розглядати, як важливі складові для усунення вразливостей, оскільки вони можуть підказати, що робити, враховуючи визначення пріоритетів, час реагування, ескалацію реагування тощо.

5. Зробіти записи для аналізу після події (звітування). Ведення записів інцидентів про те, що сталося і які процедури були зроблені, є життєво важливим для того, щоб вивчити інцидент і запобігти подальшим подіям, або принаймні мінімізувати їх вплив, а також для покращення самого процесу управління

вразливістю веб-додатку. Крім того, обов’язково важливо проводити періодичні оцінки, щоб якнайшвидше впровадити покращення або внести виправлення.

Висновки до розділу 3

В рамках даного розділу була проведена розробка системи управління вразливостями веб-додатку на базі рішення Burp Suite з метою покращення можливості виявлення та усунення вразливостей.

В результаті розробки була проведена аналітика, щодо існуючих стандартів та методик для управління вразливостями веб-додатку сучасного підприємства. На основі отриманих даних було реалізовано технологію управління вразливостями веб-додатку на базі рішення Burp Suite.

З метою розробити рекомендації щодо підвищення можливостей виявлення вразливостей веб-додатку сучасного підприємства було визначено основні кроки для проведення аналітики з використанням технології.

ВИСНОВКИ

Метою даної роботи була розробка алгоритму управління вразливостями веб-додатку сучасного підприємства на базі рішення Burp Suite.

В роботі досліджено зміст управління вразливостями та проаналізовані науково-технічні дані. Завдяки даному етапу вдалося визначити основні типи вразливостей веб-додатків та проблеми управління вразливостями у сучасних підприємствах. Була встановлена необхідність управління вразливостями для забезпечення кібербезпеки веб-додатків.

Проаналізовано наявні методи та засоби управління вразливостями та встановлено необхідність своєчасного виявлення вразливостей веб-додатка та автоматизації даного процесу, щоб уникнути збитків і непоправної школи. У зв'язку з цим зростає потреба у високоінтелектуальних сканерах, що дозволяють вирішувати поставлені завдання.

Визначено можливості програмного комплексу Burp Suite та основні функції даного програмного комплексу щодо виявлення вразливостей.

Проаналізовано можливості програмного комплексу Burp Suite щодо виконнаня процесу управління вразливостями веб-додатків у сучасному підприємстві.

Проаналізовано архітектуру програмного комплексу Burp Suite та визначено основні вимоги до інсталяції у сучасному підприємстві.

Проведено дослідження можливостей застосування програмного комплексу Burp Suite та визначено його ефективність для виявлення вразливостей.

В результаті проведеного експерименту з використанням стандарту NIST SP 800-53 та програмного комплексу Burp Suite розроблено алгоритм управіння вразливостями веб-додатка сучасного підприємства, запропоновано порядок застосування технології управління вразливостями. Завдяки можливостям даного програмного комплексу, процес виявлення та усунення вразливостей веб-додатка було вдосконалено.

У даній роботі були наведені приклади та рекомендації того, як можна використовувати фреймворк NIST для, для підтримки засобів контролю.

Встановлено загальні принципи для створення, впровадження, підтримки та вдосконалення системи управління вразливостями веб-додатку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Global Threat Intelligence Report [Електронний ресурс] – Режим доступу: https://services.global.ntt/en-us/insights/2020-global-threat-intelligence-report.

2. What Is a Web Application? How It Works, Benefits and Examples [Електронний ресурс] – Режим доступу: https://www.indeed.com/career-advice/career-development/what-is-web-application.

3. Web application (Web app) [Електронний ресурс] – Режим доступу: https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app.

4. Modern web application architecture to build a high-performance app [Електронний ресурс] – Режим доступу: https://acropolium.com/blog/modern-web-app-architecture.

5. Характеристики современных веб-приложений [Електронний ресурс] – Режим доступу: https://docs.microsoft.com/ru-ru/dotnet/architecture/modern-web-apps-azure/modern-web-applications-characteristics.

6. New Ponemon Report: A Programmatic Approach to Vulnerability Management for Hybrid Multicloud [Електронний ресурс] – Режим доступу: https://securityintelligence.com/posts/security-vulnerability-management-hybrid-multicloud.

7. Hackuity Shake'Up – The future of vulnerability management: threat status and current issues in vulnerability management (1/2) [Електронний ресурс] – Режим доступу: https://www.riskinsight-wavestone.com/en/2021/02/hackuity-shake-up-the-future-of-vulnerability-management-threat-status-and-current-issues-in-vulnerability-management-1-2.

8. OWASP Top 10:2021 [Електронний ресурс] – Режим доступу: https://owasp.org/Top10/A00_2021_Introduction.

9. Common Weakness Enumeration [Електронний ресурс] – Режим доступу: https://cwe.mitre.org/index.html.

10. Overview: OWASP Top 10 2021 [Електронний ресурс] – Режим доступу: https://www.trendmicro.com/en_us/devops/21/k/overview-owasp-top-10-2021.html.

11. Vulnerability management explained [Електронний ресурс] – Режим доступу: https://cybersecurity.att.com/blogs/security-essentials/vulnerability-management-explained.

12. New Ponemon Report: A Programmatic Approach to Vulnerability Management for Hybrid Multicloud [Електронний ресурс] – Режим доступу: https://securityintelligence.com/posts/security-vulnerability-management-hybrid-multicloud.

13. Implementing a Vulnerability Management Process [Електронний ресурс] – Режим доступу: https://www.sans.org/white-papers/34180.

14. Vulnerability Management Process [Електронний ресурс] – Режим доступу: https://www.rapid7.com/fundamentals/vulnerability-management-and-scanning.

15. Application Security Testing Software [Електронний ресурс] – Режим доступу: https://www.softwaretestinghelp.com/application-security-testing-software.

16. Netsparker Survey Reveals Executive Overconfidence in Web Security [Електронний ресурс] – Режим доступу: https://www.netsparker.com/blog/web-security/executive-overconfidence-web-security-netsparker-survey.

17. PortSwigger [Електронний ресурс] – Режим доступу: https://portswigger.net/burp.

18. Burp Suite Community Edition [Електронний ресурс] – Режим доступу: https://portswigger.net/burp/communitydownload.

19. Burp Suite's web vulnerability scanner [Електронний ресурс] – Режим доступу: https://portswigger.net/burp/vulnerability-scanner.

20. Enterprise Edition architecture [Електронний ресурс] – Режим доступу: https://portswigger.net/blog/enterprise-edition-architecture.

21. Burp Suite Software [Електронний ресурс] – Режим доступу: https://portswigger.net/support/burp-suite-software-faqs.

22. BurpSuite Enterprise Edition requirements [Електронний ресурс] – Режим доступу: https://portswigger.net/burp/documentation/enterprise/infrastructure/system-requirements.

23. Complete Guide to NIST: Cybersecurity Framework, 800-53, 800-171 [Електронний ресурс] – Режим доступу: https://reciprocity.com/complete-guide-to-nist-cybersecurity-framework-800-53-800-171.

24. Security and Privacy Controls [Електронний ресурс] – Режим доступу: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final.

25. BurpSuite Extension – Asset Discover [Tool Release] [Електронний ресурс] – Режим доступу: https://redhuntlabs.com/blog/asset-discovery-burp-extension.html.

26. Using the Target tool [Електронний ресурс] – Режим доступу: https://portswigger.net/burp/documentation/desktop/tools/target/using#analyzing-the-attack-surface.

27. Scanning web sites [Електронний ресурс] – Режим доступу: https://portswigger.net/burp/documentation/desktop/scanning#launching-scans.

28. Vulnerabilities 101 [Електронний ресурс] – Режим доступу: https://tryhackme.com/room/vulnerabilities101.

29. Reporting scan results [Електронний ресурс] – Режим доступу: https://portswigger.net/burp/documentation/desktop/scanning/reporting-results.

30. Смолєв Євген Сергійович. Технологія управління вразливостями веб-додатку сучасного підприємства на базі рішення Burp Suite. ВСЕУКРАЇНСЬКА НАУКОВА КОНФЕРЕНЦІЯ «АКТУАЛЬНІ ПРОБЛЕМИ КІБЕРБЕЗПЕКИ» Державний Університет Телекомунікацій. 27 жовтня 2021. Тези доповідей. С. 97 – 98. http://www.dut.edu.ua/uploads/p_2099_79407917.pdf