Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна

Факультет комп'ютерних наук

Спеціальність 125 «Кібербезпека»

Освітня програма «Безпека інформаційних та комунікаційних систем»

«Допущено до захисту»

В.о. зав. кафедрою БІСТ

Ольга МЕЛКОЗЬОРОВА

_____

« » 2023 р.

Пояснювальна записка

до кваліфікаційної роботи магістра

на тему: «Рекомендації забезпечення кібербезпеки веб-застосунків з використанням сучасних сканерів вразливостей»

оцінка « »

Голова ЕК

Олександр ЛЕМЕШКО _____

Керівник: Мелкозьорова О.М. _____

Рецензент: Краснобаєв В.А. _____

Виконавець : студент(ка) групи КБ-61

Бєлецький Денис Романович _____

Харків – 2023

РЕФЕРАТ

Дипломна робота складається зі вступу, чотирьох розділів, висновку, списку використаних джерел і має 57 сторінок основного тексту, 18 рисунків, 3 таблиці. Список використаних джерел містить 8 найменувань. Загальний обсяг роботи 62 сторінки.

Мета роботи: Метою роботи є розробка рекомендацій та стратегій, спрямованих на поліпшення кібербезпеки веб-застосунків за допомогою використання сучасних інструментів сканування вразливостей.

Методи дослідження: Проведення докладного огляду доступних сканерів, їх функціоналу, ефективності, інтеграцій з різними веб-застосунками. Також, використання сканерів вразливостей для тестування різних веб-застосунків з різними рівнями захисту для виявлення і аналізу потенційних уразливостей.

Результати роботи та їх новизна: Результатом цієї роботи є сформульовані рекомендації щодо застосування сканерів вразливостей для покращення кібербезпеки веб-застосунків. Виявлення нових підходів або інноваційних методів, які можуть бути використані для запобігання атакам на веб-застосунки з використанням сканерів вразливостей. Визначення переваг та обмежень різних сканерів вразливостей, враховуючи їхню ефективність та можливості. З кожним днем збільшується кількість кіберзагроз та нові методи атак. Врахування цього вимагає постійного оновлення методів захисту. Помилки в програмному забезпеченні та вразливості постійно з'являються в нових версіях, що вимагає постійного аналізу та усунення.

Рекомендації щодо використання результатів роботи: Рекомендації та результати роботи мають бути детально задокументовані та впроваджені в усі аспекти роботи з безпекою веб-застосунків.

Значущість роботи та висновки: Розроблені рекомендації та застосовані сканери вразливостей можуть значно підвищити рівень захисту веб-застосунків. Це допомагає уникнути атак, захистити дані та запобігти потенційним проникненням. Робота дозволила ідентифікувати слабкі місця веб-застосунків, що можуть стати

точками входу для зловмисників. Це дозволяє вчасно усунути вразливості. Рекомендації, що були розроблені, можуть стати новими стандартами та підходами до кібербезпеки веб-застосунків, поліпшуючи загальні практики захисту.

Припущення про можливі напрямки розвитку або продовження досліджень, що були виконані: У подальшому можливий розвиток методів та стратегій захисту веб-застосунків, враховуючи нові типи атак та вразливостей, які можуть виникнути в майбутньому. Вдосконалення алгоритмів сканування для більш точного виявлення та аналізу різноманітних видів загроз.

Ключові слова: ВЕБ ЗАСТОСУНОК, ІНФОРМАЦІЙНА БЕЗПЕКА, СКАНЕРИ ВРАЗЛИВОСТЕЙ, АТАКА, ВРАЗЛИВОСТІ ВЕБ ЗАСТОСУНКІВ, КІБЕРБЕЗПЕКА ВЕБ ЗАСТОСУНКІВ.

ABSTRACT

The thesis consists of an introduction, four chapters, a conclusion, a list of used sources and has 57 pages of the main text, 18 figures, 3 tables. The list of used sources contains 8 names. The total volume of work is 62 pages.

Purpose of the work: The purpose of the work is to develop recommendations and strategies aimed at improving the cyber security of web applications using modern vulnerability scanning tools.

Methods of review: Conducting a report examining available scanners, its functions, efficiency, integration with various web queries. Also, use opportunity scanners to test different web requests with different levels of support to discover and analyze potential opportunities.

Results of the work and their novelty: The result of this work is formulated recommendations on the use of vulnerability scanners to improve the cyber security of web applications. Identifying new approaches or innovative methods that can be used to prevent attacks against web applications using vulnerability scanners. Determining the advantages and limitations of different vulnerability scanners, considering their effectiveness and capabilities. The number of cyber threats and new attack methods is increasing every day. Taking this into account requires constant updating of protection methods. Bugs in software and vulnerabilities constantly appear in new versions, which requires constant analysis and elimination.

Recommendations for the use of work results: The recommendations and work results should be thoroughly documented and implemented in all aspects of the web application security work.

Significance of work and conclusions: Developed recommendations and applied vulnerability scanners can significantly improve the level of protection of web applications. This helps avoid attacks, protect data and prevent potential intrusions. The work made it possible to identify weak points in web applications that could become entry points for attackers. This allows timely elimination of vulnerabilities. The

recommendations that have been developed can become new standards and approaches to the cyber security of web applications, improving common security practices.

Assumptions about possible directions for development or continuation of research that has been performed: Further development of methods and strategies for protecting web applications is possible, taking into account new types of attacks and vulnerabilities that may appear in the future. Improvement of scanning algorithms for more accurate detection and analysis of various types of threats.

Keywords: WEB APPLICATION, INFORMATION SECURITY, VULNERABILITY SCANNERS, ATTACK, WEB APPLICATION VULNERABILITIES, WEB APPLICATION CYBER SECURITY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІС	-	Інформаційна система
DoS	-	Denial of Service
OWASP	-	Open Web Application Security Project
IPS	-	система запобігання вторгненням
IDS	-	система виявлення вторгнень
WAF	-	Web-Application Firewall
NGFW	-	Next Generation Firewall
HTTPS	-	HyperText Transfer Protocol Secure
XSS	-	Cross Site Scripting
CSRF	-	Cross-Site Request Forgery
URL	-	Uniform Resource Locator

ВСТУП

В сучасному світі, де інтернет є важливою частиною нашого повсякденного життя і бізнес-процесів, забезпечення кібербезпеки стає однією з найважливіших завдань для веб-застосунків. Веб-застосунки, що обробляють конфіденційну інформацію, таку як особисті дані, фінансові операції і комерційні таємниці, є особливо вразливими перед потенційними загрозами безпеки.

Сучасні кібер злочинці намагаються використовувати різноманітні атаки та експлойти для злому веб-застосунків з метою здобуття несанкціонованого доступу до даних, заволодіння контролем над веб-сайтом чи завдання інших шкідливих наслідків. Щоб запобігти цим загрозам, важливо ретельно аналізувати веб-застосунки на предмет вразливостей та вчасно приймати заходи для їх виправлення.

У цьому контексті сучасні сканери вразливостей виявляються незамінними інструментами для забезпечення кібербезпеки веб-застосунків. Сканери вразливостей - це програмні засоби, які автоматично сканують веб-застосунки на предмет слабких місць і потенційних вразливостей, що можуть бути використані зловмисниками. Вони дозволяють ідентифікувати та усувати проблеми безпеки ще до того, як вони можуть бути використані для злому.

Навіть у сучасних сканерах вразливостей іноді можна виявити недоліки або обмеження. Наприклад, певні сканери можуть бути менш ефективними у виявленні певних типів загроз чи вразливостей. Також, деякі можуть мати обмежені можливості інтеграції з певними видами веб-застосунків чи систем. Важливо усвідомлювати ці обмеження і працювати над вдосконаленням сканерів та їх використанням для максимально ефективного захисту веб-застосунків.

У даній роботі ми розглянемо деякі рекомендації щодо забезпечення кібербезпеки веб-застосунків з використанням сучасних сканерів вразливостей.

1 АНАЛІЗ ВРАЗЛИВОСТЕЙ ВЕБ-ЗАСТОСУНКІВ

Веб-застосунки (або веб-додатки) - це програми, які виконуються на веб-серверах і надають можливість користувачам взаємодіяти з ними через веб-браузери на різних пристроях, таких як комп'ютери, смартфони і планшети.

Захист веб-додатків від атак є надзвичайно важливим завданням для забезпечення безпеки і приватності користувачів, а також збереження інформації та функціональності веб-додатків. Сучасні додатки містять у собі багато конфіденційних даних користувачів, такі як особиста інформація, паролі, фінансові дані тощо. Атаки на додатки можуть призвести до витоку цих даних, якщо вони не захищені належним чином. Також, втрата або пошкодження даних користувачів може призвести до серйозних наслідків для бізнесу і його клієнтів. Наприклад, втрата фінансової інформації або інформації про замовлення може завдати значної шкоди репутації та фінансам компанії.

Веб-додатки можуть бути використані для атак на інфраструктуру, такі як видалені сервери або бази даних. Недостатньо захищений додаток може стати вразливим місцем для зловмисників, які намагаються заволодіти контролем над серверами або іншою інфраструктурою.

Зловживання веб-додатками - одна з найпоширеніших причин, яка може призвести до некоректної роботи роботи системи, спаму, та ін. Захист додатків від цих видів атак може забезпечити більш стабільну та надійну роботу.

Саме для того, щоб забезпечити безпеку, стабільну роботу та передбачити витік інформації веб-додатків слід використовувати різноманітні заходи, такі як перевірка введених даних, автентифікація та авторизація, шифрування, виявлення та захист від вразливостей, регулярні оновлення і патчі, моніторинг і логування подій, а також навчання персоналу в питаннях кібербезпеки. Безпека веб-додатків повинна бути невід'ємною частиною розробки та експлуатації веб-проектів

Покращити захист власних додатків прагнуть не тільки компанії та фахівці з кібербезпеки. Після популяризації випадків кібератак почали активно розвиватися мережеві безпекові рішення.

Спочатку, фаєрволи виступали в ролі мережевих фільтрів, розташованих між корпоративною та зовнішньою мережею. Їх завданням було блокування потенційно підозрілих мережевих пакетів на рівнях мережі та каналу, використовуючи аналіз IP-адрес джерела та призначення, міток фрагментації та номерів портів.

IPS/IDS (запобігання вторгнень/виявлення вторгнень), здатні проводити аналіз вмісту мережевих пакетів. Вони порівнюють цей вміст із відомими атаками, що дозволяє бути більш ефективними в розпізнаванні й запобіганні можливим загрозам.

Більшість нападів сьогодні здійснюються шляхом слабкостей самого додатку, а не уразливостей фреймворків та бібліотек. Із збільшенням кількості веб-додатків виникла нова проблема — загальна кількість наявних уразливостей значно перевищує кількість сигнатур, які зберігаються у сучасних системах IPS.

1.1 Конфіденційність даних - чому це так важливо?

Конфіденційність даних - це принцип і практика захисту і збереження інформації від несанкціонованого доступу та розголошення. Це означає, що лише визначені користувачі або системи мають доступ до конфіденційних даних, а інші особи або програми не мають права їх переглядати, змінювати або поширювати. Конфіденційність є однією з основних складових інформаційної безпеки.

У додатках, які працюють з особистими даними користувачів або даними компанії знаходиться багато конфіденційної інформації виток якої може призвести до великих втрат. Ця конфіденційна інформація може включати в себе особисті дані користувачів (такі як ім'я, адреса, електронна пошта, паролі тощо), фінансові дані, інтелектуальну власність (наприклад, програмний код або бізнес-секрети), а також будь-яку іншу інформацію, яку користувачі або компанії вважають конфіденційною.

Забезпечення безпеки та конфіденційності цієї інформації важливо для запобігання несанкціонованому доступу і витоку даних. Веб-розробники та компанії, що розробляють та використовують веб-додатки, зазвичай вживають

різні заходи безпеки, такі як шифрування, аутентифікація та авторизація, для захисту конфіденційної інформації.

Завжди важливо дотримуватися найкращих практик забезпечення безпеки, коли створюєте та використовуєте веб-додатки, щоб захистити конфіденційну інформацію і забезпечити приватність користувачів.

OWASP, або Open Web Application Security Project, це громадська організація, яка зосереджена на безпеці веб-додатків. Їхня мета - розробка та поширення безкоштовних ресурсів, таких як статті, методології, інструменти і технології, для поліпшення безпеки веб-додатків.

Проте, як кіберзлочинці, так і експерти з кібербезпеки, продовжують виявляти вразливості у веб-додатках. Це може мати серйозні наслідки для бізнесу. Наприклад, атаки типу XSS можуть перенаправляти запити користувачів на шкідливі веб-сторінки, а SQL-ін'єкції можуть дозволити доступ до конфіденційних даних, що зберігаються в базах даних веб-сайту.

Звичайно, веб-сервіси принесли безліч переваг, проте з поширенням їх кількості виникає ризик кіберзагроз. У звіті компанії Symantec про глобальні загрози інтернет-безпеці (ISTR) вказується, що зловмисники, які здійснюють атаки на веб-сайти, часто використовують вразливості веб-додатків на серверах або зловживають слабкостями в операційних системах, на яких ці додатки працюють.

Більшість порушень безпеки у веб-додатках виникає внаслідок розробленого командою програмного коду. Це може виникнути внаслідок помилок у кодуванні чи недостатньої уваги до важливості безпечних практик програмування, що може призвести до появи уразливостей у додатках.

Захист веб-застосунків є важливим для будь-якої компанії. Проте серед багатьох доступних рішень, таких як Firewall, IPS / IDS, NGFW, лише Web Application Firewall може надати комплексний захист від відомих і невідомих загроз веб-додатків, а також гарантувати відповідність вимогам регуляторів, наприклад, PCI DSS. Ні класичний Firewall, ні IPS / IDS не можуть забезпечити необхідний рівень безпеки веб-додатків.

1.2 Вразливості сайтів та методи їх усунення

На сьогоднішній день безпека сайтів є однією з найбільш актуальних проблем для їх власників. Це особливо важливо, оскільки у разі доступу шахраїв до значущої інформації, постраждає не лише репутація веб-сайту, а й можливість втрати коштів – основного доходу, ради якого сайт і був створений.

Web-платформа подібна до стійкого замку з високими стінами, обладнаного міцними воротами та оборонними засобами, а навколо нього – рів, заповнений водою. Все це працює ідеально, забезпечуючи надійний захист. Однак з'явився зловмисник, який зумів перелізти через огорожу. Інший спокійно увійшов через ворота, роблячи вигляд, що він простий торговець. А третій виявив секретний прохід і вдало його використав для своїх цілей.

Так само стосовно вразливості сайтів. Ви можете застосувати найсучасніші інструменти для кожного модуля сайту, але іноді система може мати вразливі місця, які стають загрозою для безпеки ресурсу. Хакери використовують саме такі просочення, щоб використовувати помилки в роботі та заволодіти серверами.

Веб-розробники розглядають вразливості як недоліки чи уразливі точки в коді ресурсу або програмах, які використовуються на сервері. Ці місця можуть стати точкою проникнення в систему та спричинити порушення в її функціонуванні.

На етапі розробки сайту можуть виникнути помилки, які призводять до слабкості паролів. Це може викликати атаки або невдалий запуск скриптів та SQL-ін'єкції.

У більшості випадків проблеми виникають через невірне оброблення даних, які надає користувач веб-ресурсу. Це може призвести до вставки некоректних і небезпечних команд у робочий код. Також можуть виникати складні ситуації, наприклад, переповнення буфера обміну, коли в нього вставляють надто великі дані, не переконавшись у тому, що для них там є достатньо місця.

Існують вразливості, які лише теоретично можуть існувати, але більшість їх є реальними проблемами.

1.3 Небезпека вразливості сайту

Перша проблема полягає в втраті контролю над власним ресурсом, що є серйозною загрозою. Розміщений контент може стати причиною включення вашого сайту до чорних списків пошукових систем. Більше того, доступ до прав адміністрування (логінів та паролів) може опинитися у руках хакерів, які можуть звернутися із вимогою викупу, щоб повернути контроль над сайтом.

Друга проблема виникає при доступі зловмисників до баз даних користувачів. Якщо це стосується логінів та паролів або адрес електронної пошти, це ще має відносно невеликі наслідки. Але доступ до платіжних даних створює серйозні ризики.

Третя проблема полягає в тому, що ваш ресурс може стати джерелом розсилки листів із шкідливим кодом, який може призвести до серйозних проблем з роботою комп'ютерів. Шахраї використовують це, щоб викликати зацікавленість у одержувачів, наприклад, вигідними вакансіями чи нагадуваннями про неоплачені штрафи.

Четверта проблема полягає у створенні фішингових (фейкових) сторінок на зламаному ресурсі, які імітують реальні сторінки соціальних мереж, банків чи інтернет-магазинів. Користувачі, вважаючи їх реальними, залишають на них фінансові дані, які потрапляють у руки шахраїв.

Інша небезпека, пов'язана з уразливістю сайтів, полягає у їх можливості заражати інші веб-ресурси за допомогою шкідливих скриптів, які впроваджуються на скомпрометовані ресурси. Ці скрипти можуть використовуватися ботами для організації масштабних DDoS-атак.

Також варто пам'ятати про можливість розміщення на скомпрометованому сайті редиректу, коли відвідувачі автоматично перенаправляються на інші сторінки, де їм пропонується платна підписка. Пошукові системи часто розглядають такі примусові переадресації як ховану рекламу та включають сайти з цими кодами до чорних списків.

Також можливість перенаправлення користувача на сторінки із вірусами. Тут частіше використовуються уразливості, які є характерними для певних

операційних систем або версій програмного забезпечення. Користувач завантажує заражений файл, який потім запускає встановлення вірусних програм, що можуть нашкодити роботі всіх пристроїв у системі.

Крім того, компрометований сайт може потрапити у немилість пошукових систем, знизивши своє рейтинг через встановлені редиректи, надмірну відправку спаму, некоректний контент і т.д. Власник сайту доведеться витратити чимало коштів та зусиль, щоб повернути своєму веб-ресурсу колишню повагу та позиції в пошукових системах.

1.4 Вразливість сайту

Для захисту веб-сайту від злому важливо не тільки вирішувати технічні питання, але і приділяти увагу людському фактору. Проведене тестування на вразливості сайтів показало, які слабкі місця є найбільш небезпечними та можуть спричинити збої у роботі ресурсу будь-якого розміру та спрямування. Ймовірність злому і швидкість відновлення після інциденту залежать від успішності усунення цих вразливостей.

- Використання програмного забезпечення з надійних джерел є ключовим. Власники ресурсів іноді завантажують модулі та плагіни з будь-яких джерел, що може вести до зараження. Щоб уникнути таких проблем, важливо використовувати лише офіційні джерела (або ті, що мають офіційний статус) для оновлення сайту новими елементами дизайну або свіжими версіями систем управління контентом. Наприклад, для WordPress офіційний репозиторій плагінів виступає надійним джерелом.
- Необхідно постійно оновлювати комп'ютери та програмне забезпечення, оскільки веб-спеціалісти прагнуть підтримувати безпеку сайтів, а зловмисники завжди шукають способи проникнути та отримати доступ до важливих даних. Використання нових стабільних версій програмного забезпечення зменшить ризик для зловмисників. Найбільш критичне значення мають компоненти програмного забезпечення, такі як ядро, плагіни, модулі, розширення та теми, через які відбувається управління ресурсами, як от системи управління контентом (CMS). Ці компоненти

обов'язково потрібно систематично оновлювати. Тут важливо пам'ятати, що це може призвести до несумісності між старими та оновленими елементами системи.

- На веб-ресурсах відсутній SSL-сертифікат, що вказує на відсутність спеціального цифрового підпису, який забезпечує безпечний протокол передачі даних через шифрування HTTPS. Якщо наявний сертифікат, користувач побачить замочок у рядку пошуку. Ці сертифікати можна придбати як у платних, так і у безкоштовних центрах сертифікації. У платних сертифікатів термін дії зазвичай довший, і, в разі витоку інформації, гарантується відшкодування фінансових збитків.
- Використання паролів у незашифрованому форматі є необхідністю в шифруванні. Рекомендується застосовувати спеціальні алгоритми хешування, такі як SHA, для захисту паролів. При автентифікації перевіряються лише зашифровані дані користувачів. Запровадження обов'язкових вимог для формування паролів може знизити ризик вразливості. Ці вимоги можуть включати мінімальну кількість символів різних регістрів, використання букв та цифр і т.д. Важливо уникати слабких паролів, таких як "12345". Для надійного захисту рекомендується використовувати комбінації з більш ніж 20 символів, а паролі менше 8 символів вважаються неприйнятними.

1.5 Проблеми процесу безпечної розробки веб-застосунків

Створення безпечних веб-застосунків це в сутності те ж саме завдання, що й створення безпечного програмного забезпечення.

Створення безпечних веб-застосунків є складною задачею, яка охоплює аспекти програмної інженерії, управління та кібербезпеки. Розробка безпечних веб-застосунків вимагає як навичок у програмній інженерії, так і глибокого розуміння технічних аспектів безпеки.

Багато з найпоширеніших методів розробки були адаптовані та модифіковані з метою забезпечення безпеки, оскільки розробка нових методів з нуля вимагає

значних витрат часу та ресурсів. Часто розробники вносять зміни до вже існуючих методів або розширюють їх для вирішення питань безпеки.

Наприклад може бути використання Microsoft SDL в поєднанні з методологією екстремальної розробки програмного забезпечення.

Можливо поєднати підхід "Життєвого циклу безпечної розробки" від Microsoft SDL та методологію "Гнучкої розробки". В цій стратегії вимоги розділяються на спрінти - тобто, на ті, що мають бути виконані протягом кожного ітераційного циклу, і ті, що виконуються лише один раз протягом всього життєвого циклу.

Існує альтернативний підхід для безпечної розробки програмного забезпечення. У цьому підході розробницький колектив адаптує процес розробки так, щоб безпека стала не лише додатковою вимогою, але і необхідним критерієм, що інтегрується у всі етапи, не обмежуючись виключно функціональними потребами.

Деякі важливі критерії розробки безпечного програмного забезпечення враховуються на початкових стадіях процесу розробки. Наприклад, створення моделі загроз, яка визначається на ранніх етапах, інтегрується у подальші фази розробки.

Проте, цей спосіб не завжди використовується у гнучких методологіях розробки. У такому роді розробки, модель загроз часто розглядається як елемент, який може бути вдосконалений протягом наступних ітерацій або спринтів.

Але найбільш популярним способом врахування безпеки в гнучкій розробці є використання "спринтів безпеки" (security sprints).

1.6 Open Web Application Security Project

OWASP (Open Web Application Security Project) є відкритим проектом, спрямованим на забезпечення безпеки веб-застосунків. Ця організація включає в себе представників корпорацій і академічних установ з багатьох країн. Головна мета OWASP полягає в створенні і поширенні відкритої інформації, такої як статті, навчальні матеріали, рекомендації, документація, інструменти і технології, які доступні для всіх зацікавлених користувачів.

OWASP надає рекомендації з безпечної розробки коду, які описують певні методи та прийоми, доступні розробникам для створення веб-застосунків, що забезпечують певну тріаду критеріїв: конфіденційність, цілісність і доступність інформації [1].

- Перевірка введених даних на коректність.
- Кодування вихідних даних у відповідності до контексту.
- Підтвердження особистості та управління паролями.
- Управління сеансами.
- Контроль доступу.
- Захист інформації шляхом криптографічних методів.
- Обробка помилок та ведення журналів.
- Захист конфіденційних даних.
- Забезпечення безпеки комунікації.
- Налаштування системи.
- Безпека бази даних.
- Управління файлами.
- Ефективне використання ресурсів пам'яті.
- Загальні принципи програмування для забезпечення безпеки.

1.7 OWASP TOP 10

Один із найсуттєвіших внесків OWASP був здійснений через проект "Top Ten Vulnerabilities" – перелік десяти найпоширеніших загроз для веб-застосунків, зазвичай оновлюваний що середини трьох років[2].

1.7.1 Injections

Програми використовують SQL запити для отримання, додавання, зміни або видалення даних, наприклад, під час користувацької роботи з особистими відомостями або заповнення форм на сайті. Якщо відсутня або недостатня перевірка даних, зловмисник може вбудувати в веб-інтерфейс додатку спеціальний код, який містить частину SQL-запиту.

1.7.2 Broken Authentication and Session Management

У випадку, коли ваш ідентифікатор стає предметом крадіжки зловмисником, а в системі відсутні перевірки, наприклад, IP-адреси сесії або перевірки встановлення кількох з'єднань у рамках однієї сесії, зловмисник може отримати доступ до системи з правами вашого облікового запису. У випадку, якщо це, наприклад, система Інтернет-банкінгу або кабінет платіжної системи, можливі наслідки несанкціонованого доступу можуть бути очевидні для вас.

1.7.3 Cross Site Scripting

Міжсайтовий скриптінг представляє собою помилку валідації даних, призначених для користувача, яка дозволяє передати JavaScript-код для виконання у браузері користувача. Ці атаки часто також називають HTML-ін'єкціями через подібність механізму їх виконання з SQL-ін'єкціями. Але, на відміну від останніх, код, що впроваджується, виконується безпосередньо у браузері користувача.

1.7.4 Insecure Direct Object References

Ця уразливість також виникає через недостатню перевірку введених користувачем даних. Згадана проблема полягає у тому, що під час виведення конфіденційної інформації, такої як особисті повідомлення чи номери карток клієнтів, для доступу до об'єктів використовується ідентифікатор, що передається відкрито в адресному рядку браузера, але не здійснюється перевірка прав доступу до цих об'єктів.

1.7.5 Security Misconfiguration

Безпека веб-застосунків передбачає належну настройку всіх складових інфраструктури: програмних компонентів, фреймворки, веб-серверів, баз даних та платформи. Настройки серверних компонентів, які задані за замовчуванням, зазвичай не є безпечними і можуть створювати вразливості для атак. Наприклад, можливе вкрадення сесійних куків через JavaScript під час XSS-атаки через відсутність за замовчуванням налаштування.

1.7.6 Sensitive Data Exposure

Багато веб-застосунків недостатньо захищені від витоку конфіденційних даних, таких як дані банківських карток або облікові дані для авторизації. Ця слабкість створює можливість для зловмисників вкрасти або змінити ці дані для своїх особистих цілей.

1.7.7 Missing Function Level Access Control

Ця уразливість, як підказує її назва, проявляється у відсутності перевірки прав доступу до об'єкта, на який робиться запит.

1.7.8 Cross-Site Request Forgery

Механізм атаки CSRF, відомий як XSRF, дає зловмиснику можливість виконувати дії від імені користувача на сервері, якщо там відсутні додаткові перевірки.

1.7.9 Using Components with Known Vulnerabilities

Зазвичай веб-застосунки розробляються з використанням спеціалізованих бібліотек або «фреймворків», які надаються зовнішніми компаніями. Більшість цих компонентів мають відкритий вихідний код, що означає, що вони використовуються не лише вами, а й мільйонами користувачів по всьому світу, які аналізують їхній код, у тому числі й з погляду можливих вразливостей. Важливо зауважити, що це не завжди проходить без наслідків.

1.7.10 Unvalidated Redirects and Forwards

Web-додатки часто переадресовують користувача з однієї сторінки на іншу. Але іноді параметри, які вказують кінцеву сторінку переадресації, можуть бути перевірені недостатньо уважно.

1.8 Брандмауери

Фаєрволи, відомі також як мережеві брандмауери, функціонують як фільтр між внутрішньою корпоративною мережею та зовнішнім Інтернетом. Перші брандмауери обмежувалися блокуванням підозрілих мережевих пакетів, виходячи з IP-адреси джерела, призначення, фрагментації й портів [9].

Сучасні системи вміють аналізувати вміст мережевих пакетів, порівнювати їх із відомими сигнатурами атак та виявляти відхилення в протоколах на рівні застосунків.

Для забезпечення надійного захисту необхіден інноваційний підхід: ретельний аналіз пакетів даних і глибоке вивчення структури веб-додатків, що включає в себе розгляд URL-параметрів, cookies, а також форм введення інформації. У цьому контексті захист може бути забезпечений за допомогою брандмауера додатків Web Application Firewall, який опрацьовує передачу даних через протоколи HTTP і HTTPS [9].

Однак сучасні атаки у більшості випадків експлуатують уразливості саме застосунків, а не мережевої архітектури. Це означає, що захист, який ґрунтується на сигнатурах атак, може бути неефективним, оскільки кількість вразливостей у веб-додатках перевищує кількість сигнатур у базах IPS. Спеціалісти вважають, що саме атаки через веб-додатки стали основними векторами нападу на корпоративні мережі, на які традиційні системи безпеки, як файрволи або антивіруси, не завжди реагують ефективно.

2 ДОСЛІДЖЕННЯ ЗАГРОЗ БЕЗПЕЦІ ВЕБ-ЗАСТОСУНКІВ

Загроза безпеці веб-застосунків (або загрози веб-додатків) вказує на потенційні ризики та вразливості, які можуть виникнути в інтернет-застосунках або веб-додатках, що працюють через браузери користувачів. Ці загрози можуть включати в себе різні види атак, такі як:

- Хакерські атаки
- Дефейсинг
- DOS (Denial of Service)

Для забезпечення безпеки веб-застосунків необхідно приділяти увагу цим загрозам та вживати відповідні заходи безпеки, такі як регулярні оновлення, перевірка на вразливості, встановлення правильних контролів доступу та шифрування даних.

2.1 Хакерські атаки

Напади, спрямовані на веб-застосунки з метою отримання несанкціонованого доступу до даних користувачів, включають SQL-ін'єкції, кросс-сайтові скрипти (XSS), кросс-сайтові запити між сайтами (CSRF) та інші атаки.

2.1.1 SQL ін'єкція (SQL injection)

Це тип атаки на безпеку даних, який використовується для внедрення зловмисного SQL-коду в запити до бази даних через неконтрольовані введення користувача. Ця атака може призвести до розкриття, видалення або модифікації даних в базі даних, а також до виконання дій на рівні бази даних без дозволу.

- Якщо веб-додаток дозволяє користувачам вводити дані, і ці дані безпосередньо включаються в SQL-запити без належної фільтрації або санітарії, зловмисники можуть ввести SQL-код, який виконується в базі даних.
- Якщо URL-параметри безпосередньо використовуються для формування SQL-запитів, то атакуючий може змінити URL таким чином, щоб вставити зловмисний SQL-код.

- Якщо дані, збережені в куках (cookies), використовуються в SQL-запитах, то атака може бути виконана через зміну значень в куках.

Щоб виявити SQL ін'єкції в програмному коді або на веб-сайтах, можна використовувати різні інструменти і сканери безпеки. Ось деякі з них:

- SQLMap: SQLMap - це дуже популярний інструмент для автоматизованого виявлення і вразливостей SQL-ін'єкцій в веб-додатках. Він може виявляти різні типи ін'єкцій і навіть автоматично експлуатувати їх.
- Nessus: Nessus - це комерційний універсальний сканер вразливостей, який може виявляти різні типи вразливостей, включаючи SQL-ін'єкції.
- Acunetix: Acunetix - це інструмент для автоматизованого сканування веб-додатків на наявність вразливостей, включаючи SQL-ін'єкції.
- OpenVAS: OpenVAS - це відкрите програмне забезпечення для сканування вразливостей, яке може виявляти різні типи вразливостей, включаючи SQL-ін'єкції.
- Burp Suite: Burp Suite - це інструмент для тестування на проникнення, який включає в себе сканер вразливостей і може допомогти виявити SQL-ін'єкції в веб-додатках.
- ZAP (OWASP Zed Attack Proxy): ZAP є безкоштовним інструментом для тестування на проникнення, розробленим OWASP. Він має вбудований сканер вразливостей, який може допомогти виявити SQL-ін'єкції.

Для запобігання SQL ін'єкціям важливо використовувати параметризовані запити, що розділяють дані користувача від SQL-коду і не дозволяють їх змішувати. Також слід валідувати та санітувати всі дані, які надходять від користувачів, перш ніж вони використовуються в SQL-запитах. Використовуйте бібліотеки або фреймворки для роботи з базами даних, які надають засоби для запобігання SQL ін'єкціям.

2.1.2 Кросс-сайтовий скриптинг (Cross-Site Scripting або XSS)

Це тип веб-атаки, при якій зловмисник вставляє в веб-сторінку (або веб-додаток) зловмисний JavaScript-код, який буде виконуватися в браузері користувача. Ця атака може мати серйозні наслідки, такі як викрадення сесійних

куків, відправка зловмисних запитів від імені користувача, перенаправлення користувача на інші сайти та інші зловмисні дії.

Існують три основні типи XSS-атак:

- Stored (збережений) XSS: У цьому випадку зловмисник вставляє зловмисний код на сервері або в базі даних, і цей код відображається користувачам, які переглядають відповідну сторінку. Наприклад, коментарі на форумі, де коментарі зберігаються на сервері і показуються всім користувачам, можуть бути потенційно вразливими на Stored XSS.
- Reflected (відбивний) XSS: У цьому випадку зловмисник вставляє зловмисний код в параметр URL або форми, і цей код відображається тільки для користувача, який переходить за посиланням або відправляє запит. Наприклад, якщо сайт показує повідомлення про помилку з параметрами запиту, то зловмиснику може вдалося вставити туди свій XSS-код.
- DOM-based (DOM-основний) XSS: Цей тип XSS відбувається на клієнтській стороні, коли зловмисник використовує JavaScript для зміни DOM-структури сторінки після її завантаження. Зазвичай це відбувається за допомогою JavaScript-функцій, які виконують операції над URL або іншими даними, які потрапляють на сторінку.

Для запобігання XSS-атакам, рекомендується:

- Екранування введених даних: Всі дані, які користувач вводить на вашому сайті або надсилає через форми, повинні бути валідовані та екрановані перед виведенням на сторінку.
- Використовуйте заголовки Content Security Policy (CSP): CSP дозволяє налаштовувати політику безпеки браузера, обмежуючи, з яких джерел може бути виконаний JavaScript на сторінці.
- Використовуйте вбудовані функції безпеки браузера: Багато сучасних браузерів мають вбудовані механізми фільтрації XSS, які можна використовувати разом з CSP.

- Регулярно оновлюйте ваші бібліотеки і фреймворки: XSS-вразливості можуть виникати через вразливості у бібліотеках і фреймворках, які ви використовуєте.

Для виявлення кросс-сайтових скриптів (XSS) в веб-додатках і веб-сайтах можна використовувати різні інструменти та методи. Ось деякі з них:

- Ручна перевірка: Ви можете ручно аналізувати вихідні дані та введені дані на веб-сторінках, шукаючи потенційно небезпечні вставки JavaScript. Використовуйте різні браузерні розширення, такі як "XSS Me" для Firefox або "XSS Validator" для Chrome, які допоможуть виявити XSS-вразливості.
- Автоматизовані інструменти: Існують різні безкоштовні та комерційні інструменти, призначені для виявлення XSS-вразливостей автоматично. Деякі приклади цих інструментів включають в себе:
  - Netsparker
  - Burp Suite
  - OWASP ZAP (Zed Attack Proxy)
  - Acunetix
  - Wapiti
- Ручний аудит коду: Якщо ви маєте доступ до вихідного коду веб-додатка, ви можете ретельно аналізувати його, шукаючи місця, де введені дані не відфільтровуються чи не екрануються перед тим, як вони вставляються у вихідні HTML-сторінки.
- Пентест: Частиною процесу тестування на проникнення (пентесту) є пошук XSS-вразливостей. Досвідчені пентестери можуть використовувати різні методи та інструменти для виявлення цих вразливостей.
- Сервіси веб-перевірки вразливостей: Деякі веб-сервіси, такі як "Google's Web Vulnerability Scanner" або "Mozilla Observatory," можуть виявляти деякі типи XSS-вразливостей на вашому веб-сайті.

2.1.3 Міжсайтова атака з підставленням запитів (Cross-Site Request Forgery або CSRF)

Це тип атаки на безпеку веб-додатків, при якій зловмисник використовує авторизований обліковий запис користувача для виконання непередбачених або зловмисних дій на іншому сайті без його належного дозволу. Ця атака використовує довіру, яку сайт має до користувача, для виконання зловмисних дій в інших контекстах.

Основні характеристики CSRF:

- Псевдо Автентифікація: Атака використовує авторизований сеанс користувача, який вже залогінений на сайті, для виконання дій на іншому сайті.
- Замаскування запитів: Зловмисник формує запити, які маскуються під легітимні запити, що надходять від користувача, і це може запити на виконання зміни стану акаунта чи відправлення фінансових операцій.

Для захисту від атак CSRF рекомендується виконувати наступні дії:

- Використовуйте стандартні методи POST: Використовуйте метод POST для всіх дій, які можуть впливати на стан сайту або користувача. Це зробить складніше підставляти запити, оскільки злоумисникам складніше підміняти POST-запити через HTML-форми.
- Використовуйте токени безпеки: Вставте унікальні токени безпеки в форми і запити, які потребують авторизації. Токени безпеки повинні бути перевірені на сервері для перевірки правомірності запиту.
- Заборонити Cross-Origin Requests: Використовуйте заголовки HTTP, такі як "SameSite" і "Cross-Origin Resource Sharing (CORS)", щоб обмежити доступ до ресурсів на інших сайтах.
- Перевіряйте реферер (Referer): Перевіряйте HTTP-заголовок "Referer", щоб переконатися, що запити приходять від легітимних джерел.
- Робіть перевірки на стороні сервера: Не покладайтесь тільки на перевірки на стороні клієнта, так як їх можна обхідти. Виконуйте всі необхідні перевірки на стороні сервера.

2.2 Дейфсинг

Дефейсинг (англ. "doxxing") - це практика розголошення особистої інформації про людину в Інтернеті з метою її виявлення, ідентифікації, ушкодження, покарання або переслідування. Ця інформація може містити ім'я, адресу, номер телефону, адресу електронної пошти, місце роботи, фотографії та інші особисті дані. Дефейсинг може використовуватися як форма цифрової атаки або кібербулінгу в Інтернеті, і його метою може бути завдання шкоди репутації, створення загрози безпеці або навіть спричинення фізичного насильства проти людини.

Дефейсинг є незаконною і морально неприйнятною практикою. Багато країн мають закони, що забороняють розголошення особистої інформації без згоди особи, яку це стосується, і покарання осіб, які вчиняють такі дії.

2.3 DOS (Denial of Service)

DOS, або "Denial of Service" (відмова в обслуговуванні), є типом кібератаки, яка полягає в спробі зробити комп'ютерну систему, мережу або веб-сайт недоступними для законних користувачів, заважаючи нормальному функціонуванню. Ця атака робиться шляхом надмірного завантаження цільової системи або веб-сайту запитами або трафіком, що перевищує її ресурси і можливості.

Ось деякі ключові риси атаки DOS:

- Перевищення ресурсів: Атакуюча сторона намагається перевищити обсяг обробки запитів або доступних ресурсів на цільовій системі. Це може призвести до тимчасової або тривалої відмови в обслуговуванні для законних користувачів.
- Трафік атаки: Для атаки DOS можуть використовуватися різні методи, включаючи надсилання великої кількості запитів HTTP, UDP або ICMP, а також використання ботнетів (мережі комп'ютерів, підконтрольних атакуючому).

- Ціль: Ціллю атаки DOS може бути веб-сайт, мережевий сервер, послуга в хмарі, DNS-сервер, або будь-яка інша комп'ютерна система, яка може бути піддана надмірному трафіку.
- DDoS (Distributed Denial of Service): В разі DDoS атаки, атакуюча сторона використовує багато комп'ютерів (часто у складі ботнету) для відправки трафіку на цільову систему. Це робить атаку ще більш потужною та складною для виявлення і відсічі.

Сканери безпеки, які використовуються для виявлення вразливостей в веб-додатках та мережах, зазвичай не призначені для виявлення самої атаки на відмову в обслуговуванні (DoS) або розподіленої атаки на відмову в обслуговуванні (DDoS). Зазвичай ці сканери фокусуються на виявленні конкретних вразливостей у веб-додатках або середовищі, а не на виявленні атак на відмову в обслуговуванні, які можуть бути спрямовані на сам сервер або мережу.

2.4 Дослідження сучасних сканерів вразливостей

Сучасні сканери вразливостей веб-додатків - це інструменти, призначені для виявлення та аналізу потенційних вразливостей веб-додатків з метою їхнього подальшого виправлення. Такі сканери допомагають забезпечити безпеку веб-додатків і запобігти можливим атакам, таким як SQL-ін'єкція, перехоплення сесій, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF) та іншим.

Ці сканери можуть лише ініціювати запити до застосунку і перевіряти відповідь, але вони не мають можливості перевірки внутрішнього коду [8]. Іншими словами, сканери веб-застосунків можуть взаємодіяти з застосунком або веб-сайтом лише так, як це робить потенційний зловмисник.

Загалом, сканери вразливостей включають три модулі або етапи: crawler (пошук), attacker (атака) та analysis (аналіз). Починаючи сканування, сканеру передають набір URL-адрес. Модуль кравлера відвідує ці адреси та збирає всі можливі URL на сторінках веб-застосунку. Потім він відвідує всі недавно зібрані адреси, щоб знайти ще більше URL-адрес. Ця дія триватиме, доки модуль сканера не зібере якнайбільше доступних веб-сторінок. Всі вхідні точки до веб-застосунку, такі як поля введення та параметри GET, також збираються під час етапу пошуку.

Після завершення роботи модуля сканера, він передає усю зібрану інформацію (доступні веб-сторінки та вхідні точки) до модуля атаки.

Модуль атаки створює атакові вектори, використовуючи інформацію, отриману від модуля пошуку. Ці вектори містять вхідні дані, які можуть розкрити потенційні вразливості. Атакуючий модуль відправляє запити до веб-застосунку, використовуючи ці атакові вектори. Усі відповіді, одержані від веб-застосунку, передаються модулю аналізу. Для кожної відповіді від застосунку, модуль аналізу намагається визначити наявність слідів, що вказують на потенційні вразливості. Якщо веб-сканер виявляє атакований вектор, який спричинює відповідь, що підозрює наявність вразливості, він позначає веб-сторінки як вразливі і повідомляє, який саме атаковий вектор призвів до вразливості.

Існує широка розмаїтість сканерів веб-застосунків із різними функціональними можливостями. Як приклади можна навести OWASP та The Web Application Security Consortium, які мають списки потужних сканерів [14][18]. З обох цих проектів доступний список сканерів веб-уразливостей, які можна завантажити безкоштовно.

Деякі з популярних сучасних сканерів вразливостей веб-додатків включають:

- OWASP Zap: Це безкоштовний та відкритий джерела проект, створений OWASP (Open Web Application Security Project). Він надає можливості активного сканування та пасивного аналізу вразливостей.
- Burp Suite - це комерційний сканер вразливостей, який надає великий набір інструментів для аналізу безпеки веб-додатків, включаючи Proxy, Scanner, Intruder і інші.
- Nessus: Nessus є одним із найпопулярніших комерційних сканерів вразливостей для мереж та веб-додатків. Він використовується для сканування і аналізу великої кількості вразливостей.
- Nexpose (Rapid7 InsightVM): Це інструмент від компанії Rapid7, який також надає можливості сканування та аналіз.

2.5 Труднощі пошуку вразливостей

Під час аналізу сканерів було виявлено, що у багатьох випадках результати сканування не є достатньо достовірними, особливо коли зловмисні дані спочатку зберігаються в самому веб-застосунку, а потім використовуються для генерації виведення на інших сторінках. Це призводить до ситуації, коли відповідь на запит сканера не містить жодних ознак наявності вразливості, оскільки сама атака спрацьовує на інших сторінках веб-застосунку.

Подібні висновки були зроблені професором з Каліфорнійського інституту Дуп'є, на прикладі атаки, де спочатку дані зберігаються у форматі, що вразливий на перехоплення типу XSS атаки.

Для сканерів вразливостей виявлення CSRF атак є завданням не таким простим, оскільки вони часто дають багато невірних позитивних результатів, тобто вони помічають вразливості, яких насправді немає. Це стається через те, що важко визначити, які конкретні запити повинні бути захищені від CSRF. Один спосіб спробувати знайти цей тип атак полягає у тому, щоб записувати всі запити, а після завершення сканування надсилати окремі запити, використовуючи підозрілі записи як потенційні вразливості. Якщо ці повторні запити виявляються успішними, то є більше підстав вважати, що існує CSRF-уразливість. Проте слід пам'ятати, що результат не може бути абсолютно впевненим, оскільки багато веб-застосунків не вимагають захисту від CSRF на деяких сторінках.

2.6 Порівняння OWASP ZAP та Nessus

OWASP ZAP і Nessus - це два різні інструменти для тестування на проникнення і виявлення вразливостей в програмному забезпеченні та мережах. Ось порівняльний огляд обох інструментів (Таблиця 2.1 – 2.2):

Таблиця 2.1 – Порівняльна характеристика OWASP та Nessus

Name	Scope of application	License	Functionality
OWASP	OWASP ZAP (Zed Attack Proxy) - це відкрите програмне	OWASP ZAP є вільним програмним	OWASP ZAP спеціалізується

Продовження Таблиці 2.1 – Порівняльна характеристика OWASP та Nessus

	забезпечення, розроблене спеціально для тестування на проникнення веб-додатків. Він призначений для виявлення вразливостей на рівні застосунків, таких як SQL-ін'єкції, кросс-сайтовий скриптинг (XSS), ін'єкції команд і т.д.	забезпеченням з відкритим вихідним кодом. Ви можете завантажити його безкоштовно і використовувати його для власних потреб.	на виявленні вразливостей веб-додатків і має багатий набір інструментів для тестування на проникнення, перехоплення та аналізу HTTP-запитів та відповідей, а також автоматизовану і ручну перевірку вразливостей.
NESSUS	Nessus - це комерційний сканер вразливостей, який призначений для сканування мережі та серверів на наявність вразливостей, включаючи вразливості на	Nessus має комерційну ліцензію, і доступ до його повнофункціональної версії може бути платним.	Nessus сканує мережі і сервери для виявлення вразливостей на різних рівнях, включаючи операційні системи, мережеві служби та додатки. Він також надає звіти і рекомендації щодо

Продовження Таблиці 2.1 – Порівняльна характеристика OWASP та Nessus

	рівні операційної системи та додатків.		виправлення вразливостей.

Таблиця 2.2 – Порівняльна характеристика OWASP та Nessus

Name	Interface	Price
OWASP	OWASP ZAP має графічний інтерфейс користувача (GUI) та може бути використаний через командний рядок.	Безкоштовний і відкритий вихідний код.
NESSUS	Nessus також має графічний інтерфейс користувача та може бути налаштований і запущений через веб-інтерфейс.	Nessus має різні варіанти ліцензій, включаючи платні версії.

2.6.1 Статистика сканерів

Порівняльна характеристика з сайту “SourceForge” таких сканерів: Nessus, OWASP ZAP, OpenSCAP. Характеристика проводиться по таким критеріям:\

- Platforms Supported (Підтримка платформ) - є важливою у сфері інформаційних технологій та програмного забезпечення. Сканери регулярно випускають оновлення та патчі, які виправляють виявлені вразливості та забезпечують високий рівень безпеки. Підтримка платформ означає, що користувачі отримують актуальні заходи безпеки, що знижує ризик вразливостей та атак.(Рисунок 2.1)

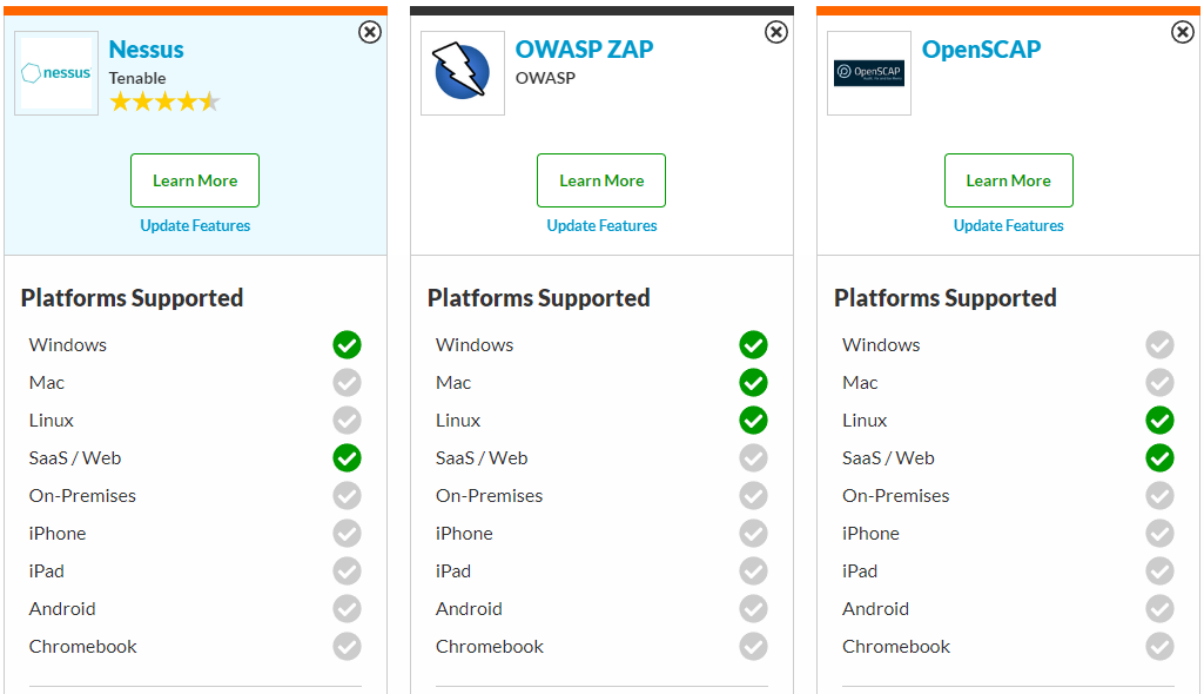


Рисунок 2.1 - Platforms Supported

- Audience (Аудиторія) – У кожного сканера вразливостей є своя аудиторія, та свої сильні сторони, по яким кожна з сфер обирає його.(Рисунок 2.2)

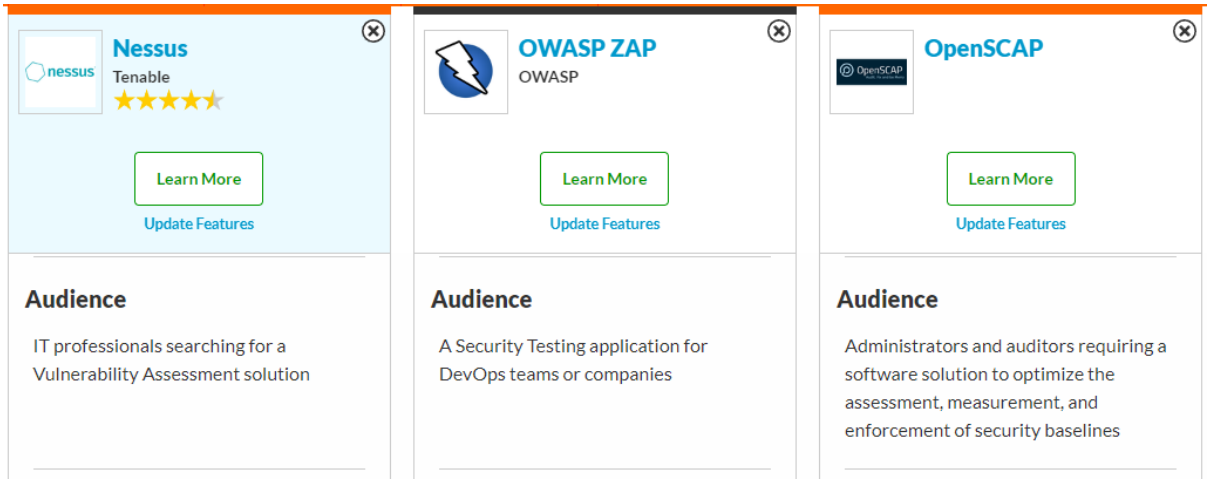


Рисунок 2.2 - Audience

- Support (Підтримка) – Є однією з важливих пунктів, так як підтримка програми дозволяє відстежувати нові загрози, помилки, функціїї та виправляти їх у новій версії. Також, підтримка важлива при виникненні помилки у користувачів.(Рисунок 2.3)

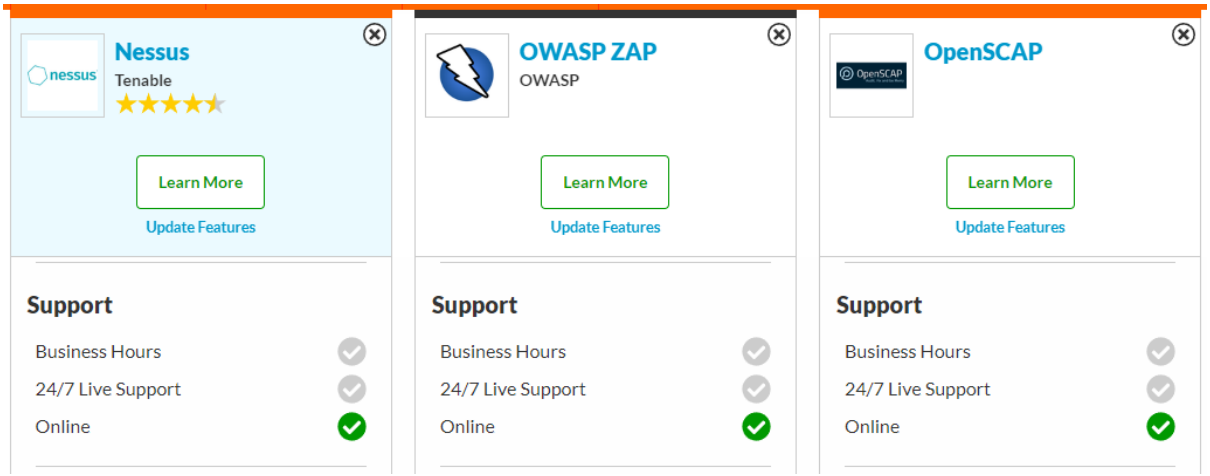


Рисунок 2.3 – Support

- API (Application Programming Interface) - це набір правил та інструкцій, які визначають, як різні програми чи компоненти програмного забезпечення можуть взаємодіяти один з одним. API надає програмам структурований спосіб обміну даними та функціональністю без необхідності знаходження внутрішньої реалізації однієї програми іншою.(Рисунок 2.4)

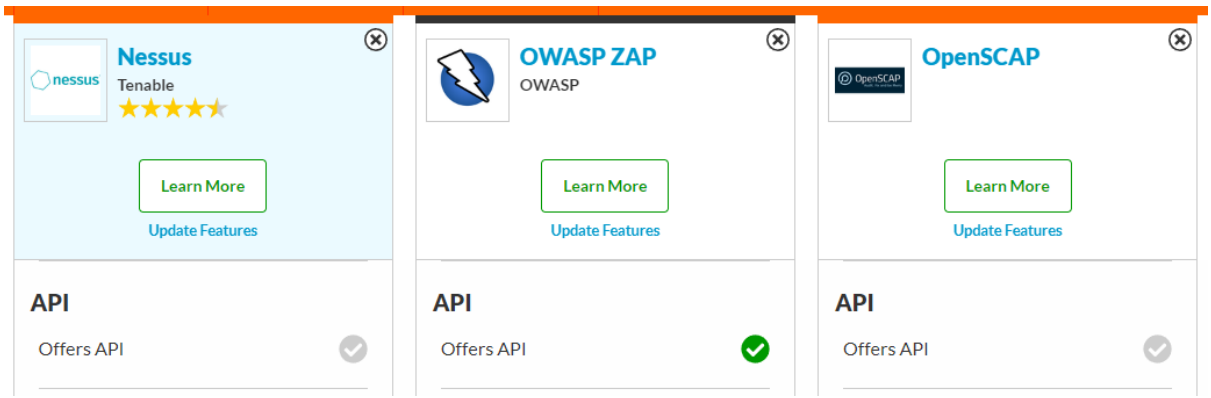


Рисунок 2.4 - API

- Pricing, Reviews, Raitings (Ціна, Відгуки, Рейтинг) – Одне з перших пунктів, куди подивиться користувач при виборі сканера вразливостей, та дізнається усі мінуси та плюси.(Рисунок 2.5)

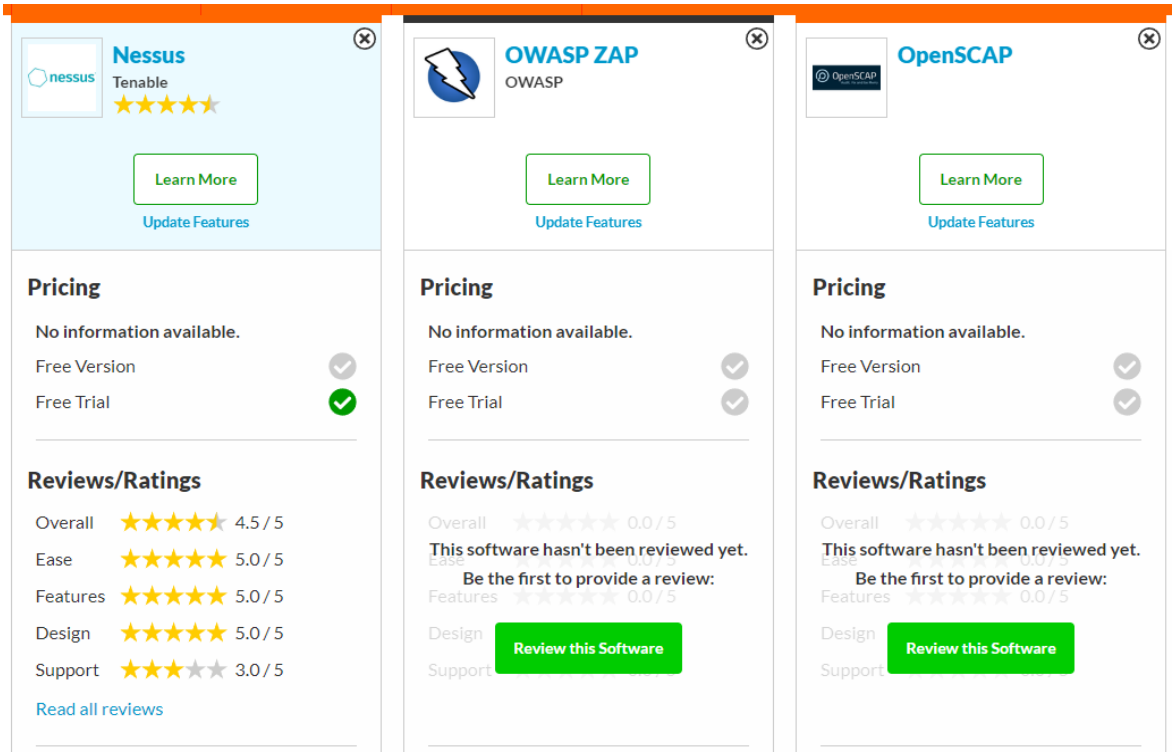


Рисунок 2.5 - Pricing, Reviews, Raitings

- Categories (Категорії) – Деякі аспекти по яким сканер вразливості виграє у інших.(Рисунок 2.6)

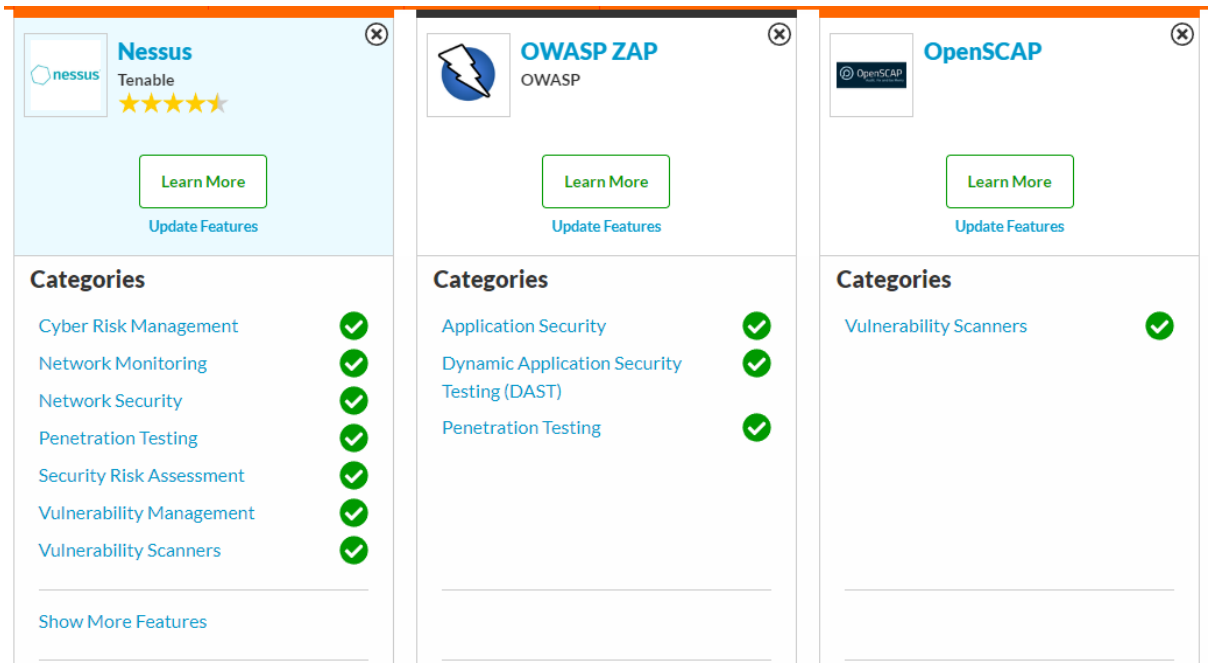


Рисунок 2.6 - Categories

- Integrations (Інтеграції) - процес забезпечення співпраці та взаємодії різних систем, програм або сервісів для обміну даними та виконання спільних функцій. Це може бути реалізовано за допомогою API, плагінів, сервісів, протоколів та інших методів. (Рисунок 2.7)

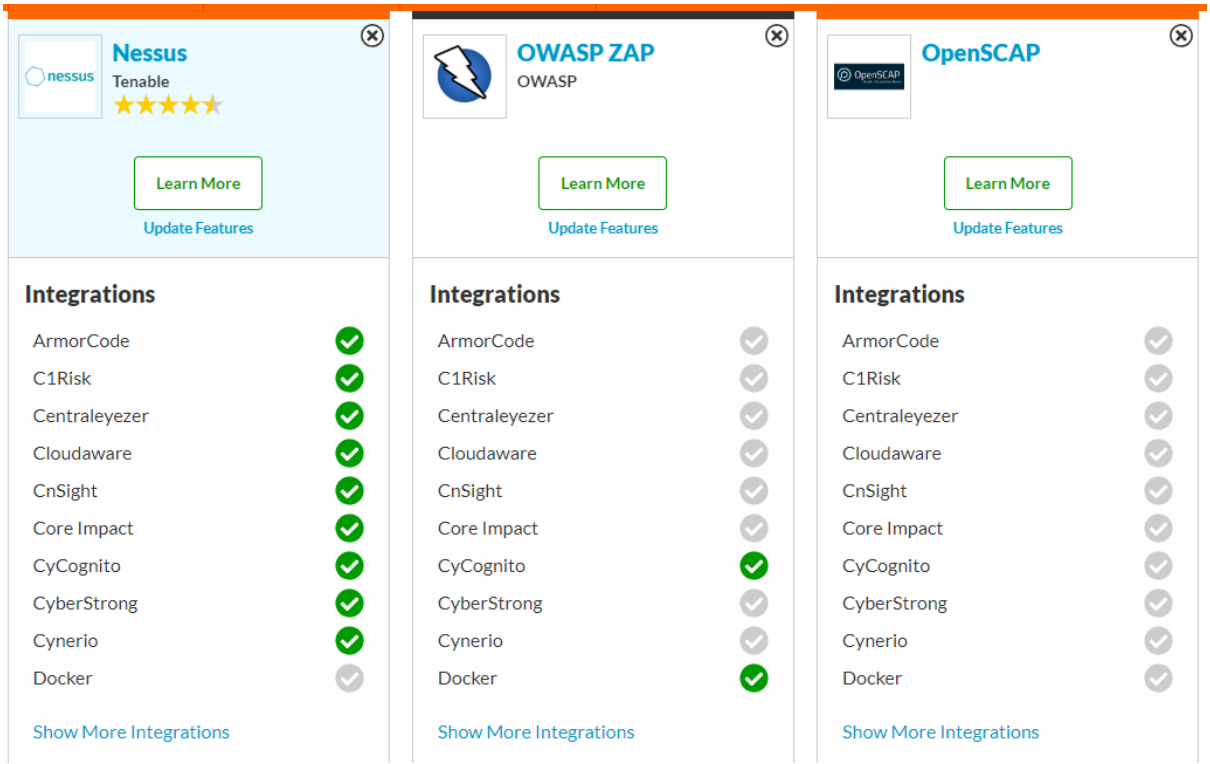


Рисунок 2.7 – Integrations

2.7 OWASP ZAP - тестування сайту

OWASP ZAP - це сканер веб-застосунків, що базується на методиці DAST (Динамічне тестування безпеки додатків). У українській версії цей метод відомий як "метод тестування в чорній скриньці". Методика дозволяє виявляти проблеми безпеки в робочому додатку або веб-сайті, скануючи їх на відомі уразливості. До таких вразливостей можна віднести SQL-ін'єкції, міжсайтовий скриптинг (XSS), Clickjacking та інше.

OWASP ZAP розроблений та підтримується проектом з аналогічною назвою - OWASP (Open Web Application Security Project) - неприбутковою організацією, що спеціалізується на створенні статей, матеріалів, документації, інструментів та технологій для покращення безпеки розробки додатків, а також забезпечення належного рівня інформаційної безпеки вже створених додатків і сайтів.

Вбудовані можливості OWASP ZAP які користуються найбільшою популярністю:

- AJAX CRAWL, пошуковий бот: оглядає сторінки веб-сайту для отримання потрібних даних, які потім представляються у вигляді HTML в кінцевій точці. Кожна пошукова система має свої налаштування, такі як увімкнення або вимкнення JavaScript, CSS, зображень і т.д.
- Проксі-сервер для перехоплення трафіку: програма, яка налаштовує VPN та проксі-сервер для запису HTTP/HTTPS трафіку.
- Автоматичний сканер
- Пасивний сканер
- Інструменти для примусового браузерного доступу
- Фаззер
- Підтримка веб-сокетів
- Підтримка Plag-n-hack

OWASP ZAP використовується для проведення тестів на проникнення та пошуку вразливостей у веб-застосунках. Цей інструмент доступний для досвідчених фахівців у галузі інформаційної безпеки, таких як розробники і професіонали з ручного тестування, а також для початківців у цій галузі, включаючи тих, хто цікавиться безпекою веб-застосунків [1].

Архітектура ZAP базується на плагінах, з онлайн-магазином, де можна додавати нові функції або оновлювати наявні. Інтерфейс користувача зручний та легкий у використанні, має інтуїтивний дизайн.

OWASP ZAP має такі режими:

- Безпечний режим - при використанні цього режиму неможливо виконати небезпечну дію для веб-застосунку.
- Захищений режим – використовуючи цей режим, користувач може здійснювати лише шкідливі дії в межах URL-адрес.
- Стандартний режим –у цьому режимі користувач може робити все, що дозволяє веб-застосунок.

- Режим Атаки – при виявленні нових вразливостей в області дії шпигуна, вразливі вузли автоматично скануються, як тільки їх виявлено.

Розглянемо программу OWASP ZAP (version 2.14.0). та протестуємо сайт з її допомогою.

1) Інтерфейс програми - Інтерфейс програми OWASP ZAP розроблений для зручності користувачів і надає доступ до різних функцій і можливостей, необхідних для сканування веб-додатків на уразливості та виявлення проблем безпеки(Рисунок 2.8).

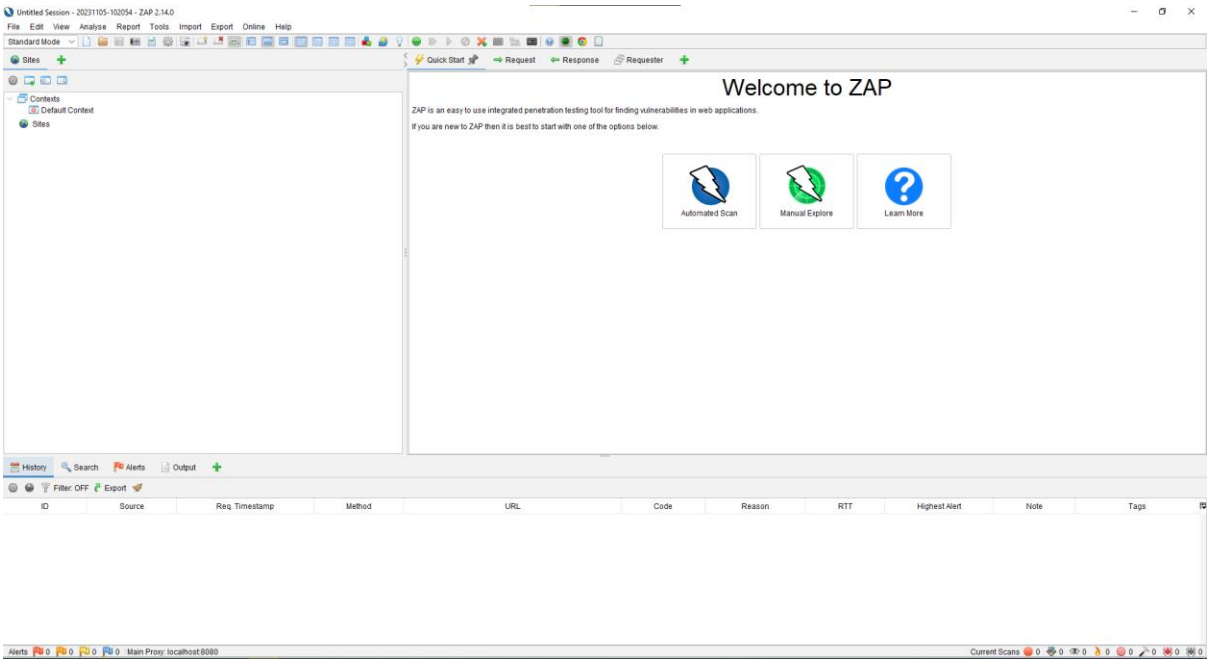


Рисунок 2.8 - Інтерфейс програми - OWASP ZAP

2) Для зручності тестування сайт будемо використовувати мануальне тестування(Рисунок 2.9). Мануальне тестування - це процес, при якому тестери вручну перевіряють функціональність та коректність роботи програмного продукту. Це означає, що тести виконуються без використання автоматизованих інструментів чи сценаріїв. Основна мета мануального тестування - виявлення помилок, аналіз роботи системи та перевірка відповідності програми вимогам. Виберемо браузер для тестування - Google Chrome (Рисунок 2.10):

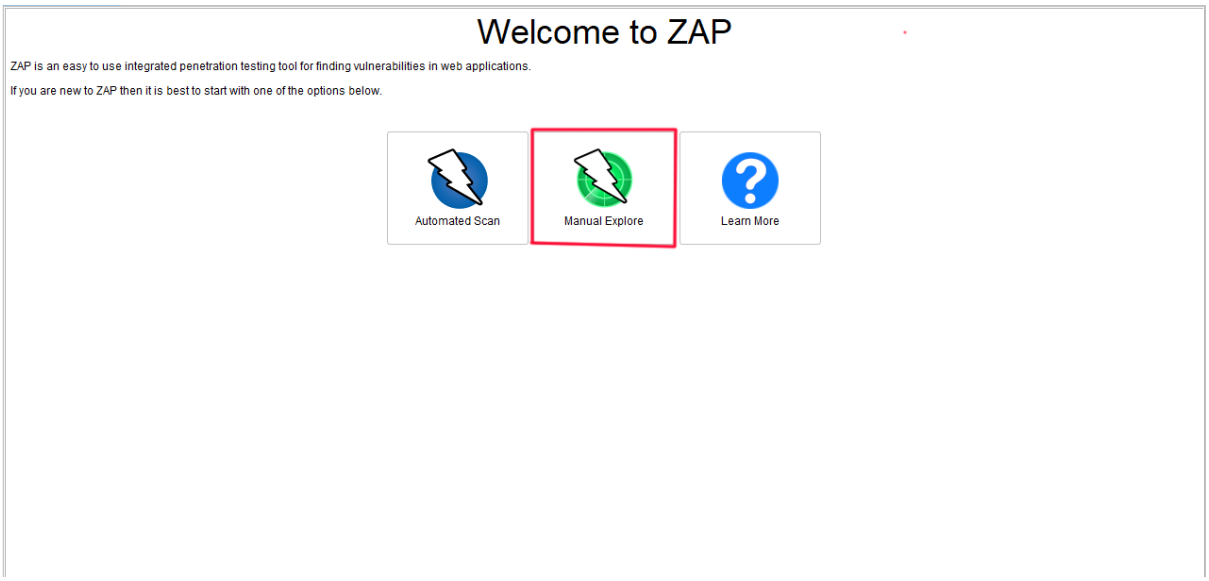


Рисунок 2.9 - Manual Explore OWASP ZAP

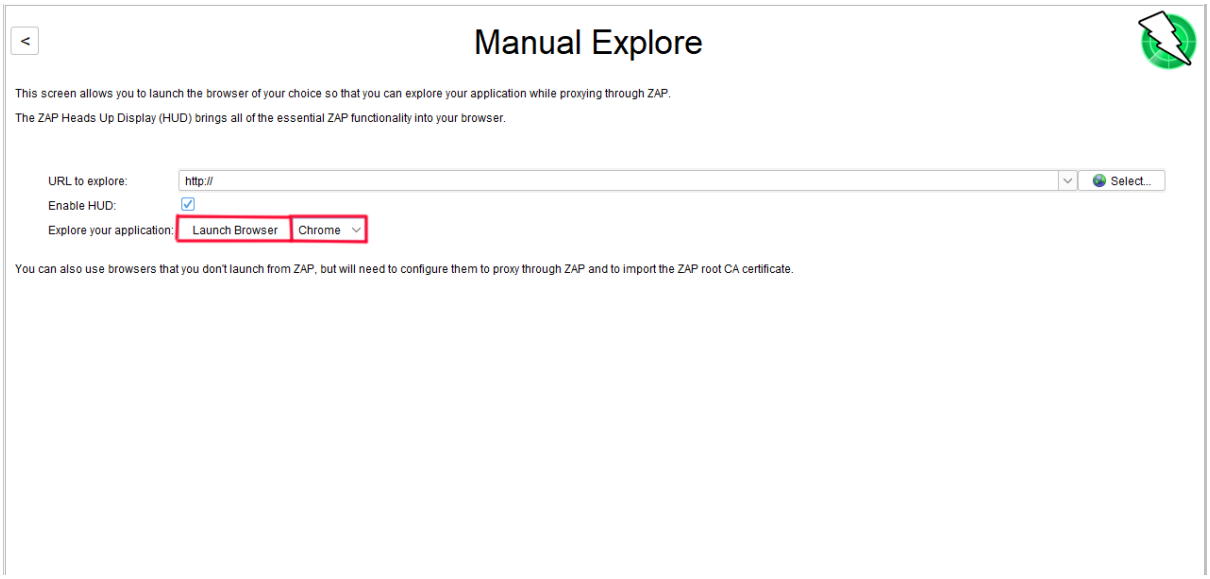


Рисунок 2.10 - Вибрали застосунок - Google Chrome

3) Наступним кроком запустимо інтерфейс програми у браузері та зайдемо на сторінку сайту “prom.ua” для тестування (Рисунок 2.11):

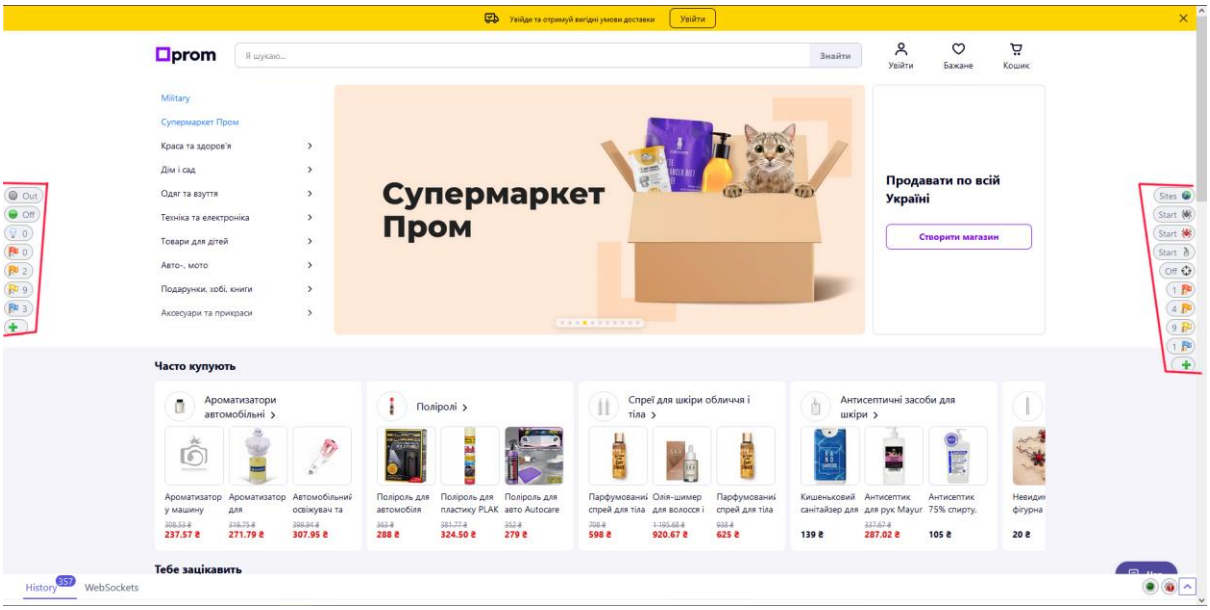


Рисунок 2.11 - Інтерфейс програми у браузері

4) На даний момент, просто при запуску сайту ZAP знайшов лише одну критичну загрозу - це Hash Disclosure - Mac OSX salted SHA-1. На рисунку 2.12 показані усі помилки та вразливості.

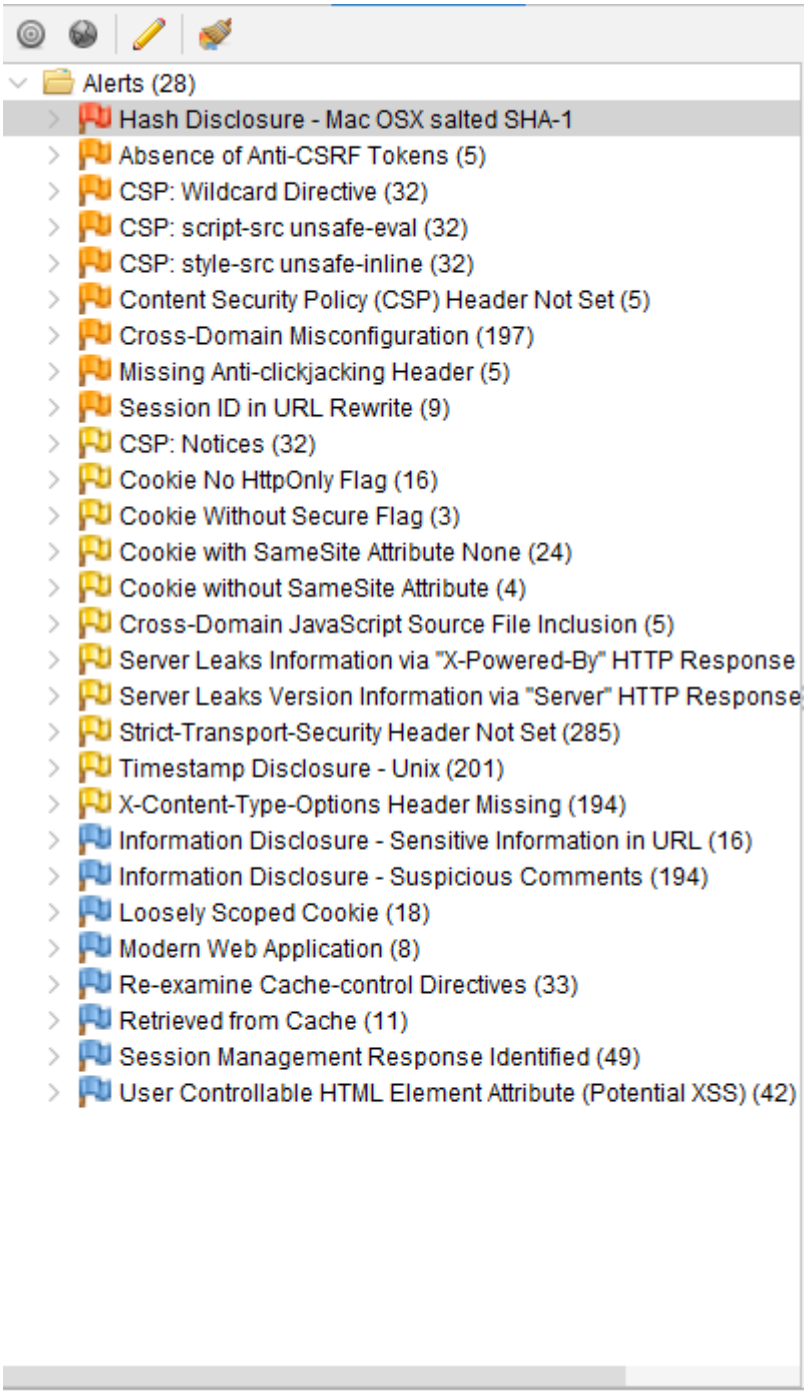


Рисунок 2.12 - Помилки знайдені на сайті до атаки

5) Тепер проведемо активну та спайдер атаку і подивимося знайдені помилки(Рис. 2.13).

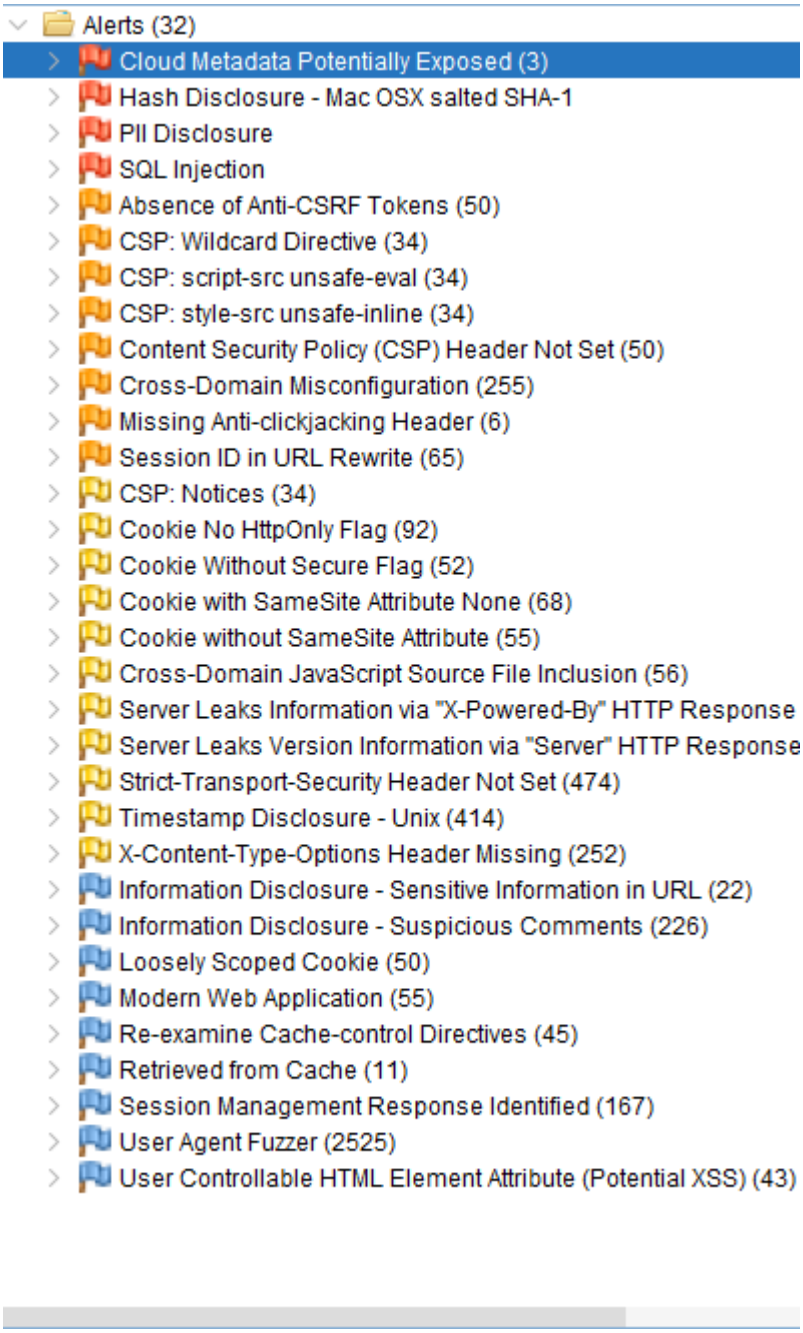


Рисунок 2.13 - Знайдені вразливості після атаки

2.8 Тестування сайту з допомогою сканера вразливості Nessus

1) Інтерфейс Nessus здався мені дещо важчим, ніж у OWASP ZAP. Тим не менш, виберемо тестування веб-застосунку(Рисунок 2.14):

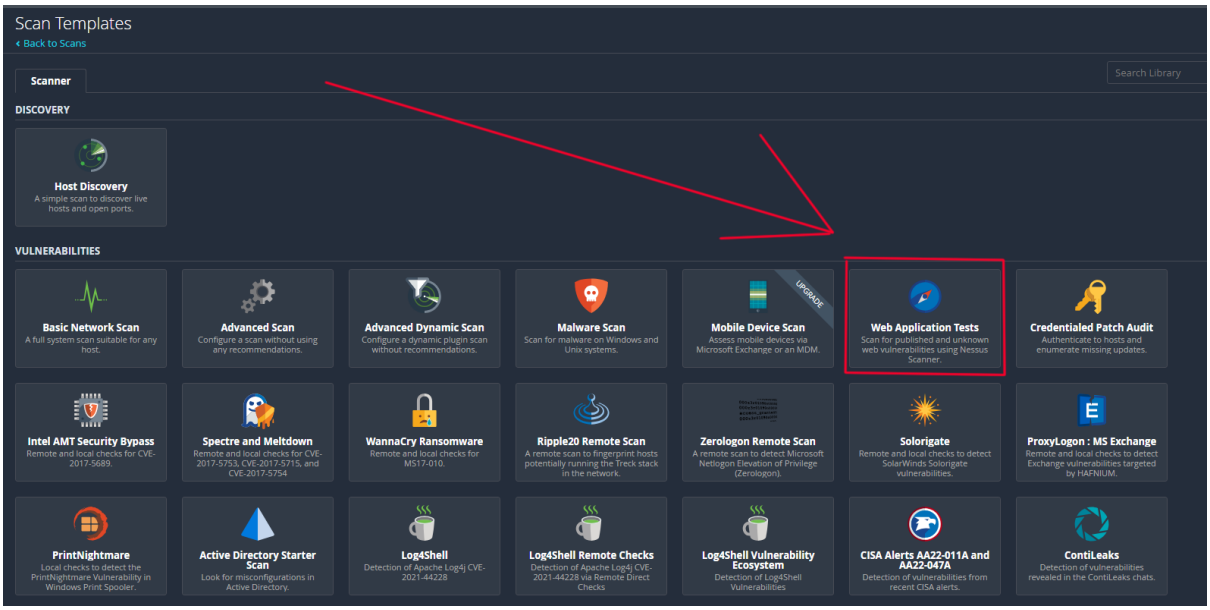


Рисунок 2.14 - Створюємо новий тест

2) Для того, щоб почати тестування сайту треба мати його IP адресу, яку ми знайдемо за допомогою консолі (Рисунок 2.15). На запрос у консолі я отримав адресу, яку ввів у полі IP адреса при створенні нового тесту (Рисунок 2.16).

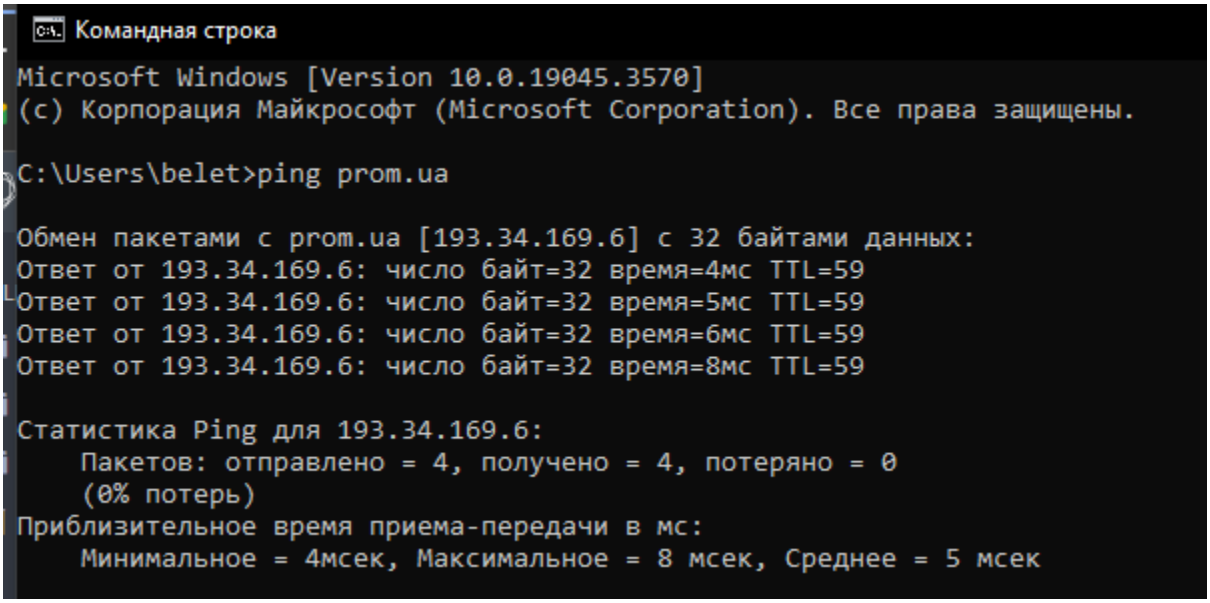


Рисунок 2.15 - Знаходимо IP адресу сайту “Prom”

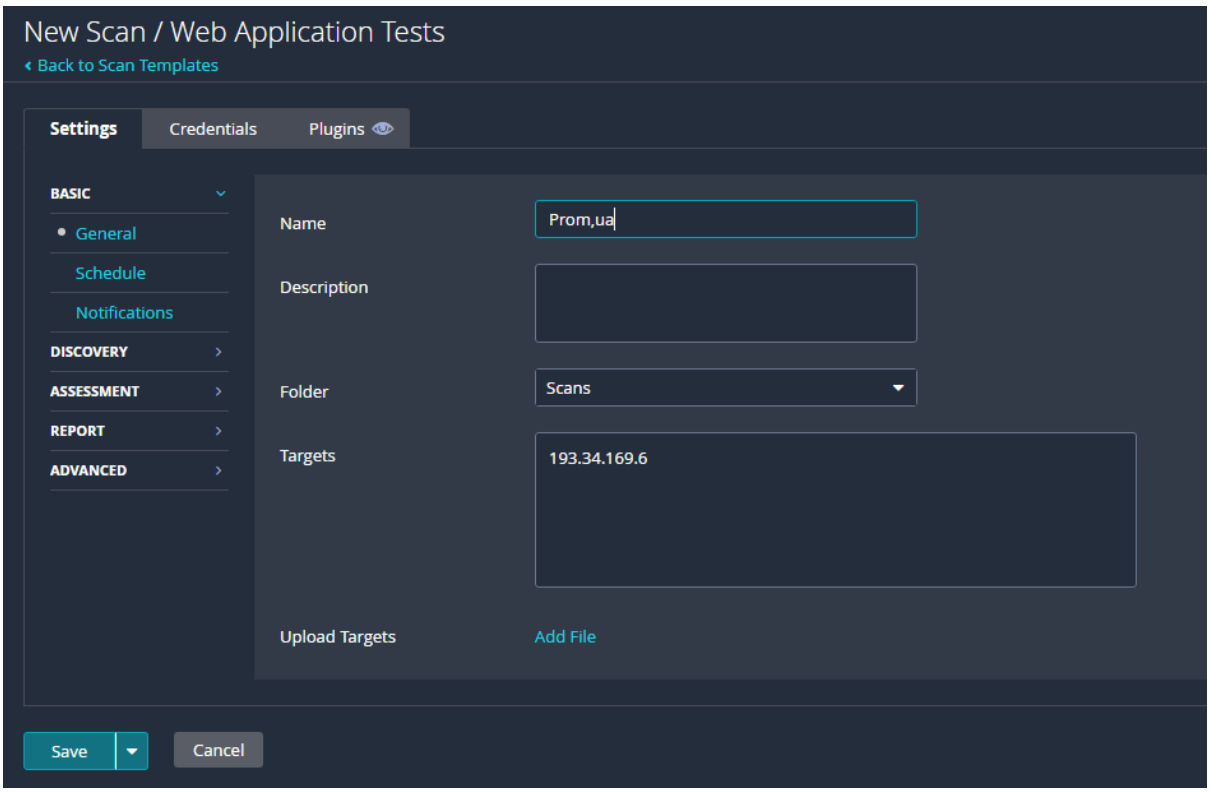


Рисунок 2.16 - Створюємо нове сканування

3) Сканування сайту проходило автоматично, не було можливості зробити мануального тестування, що я вважаю великим мінусом, так як не кожна автоматизація відпрацьоваю 100% правильно (Рисунок 2.17).

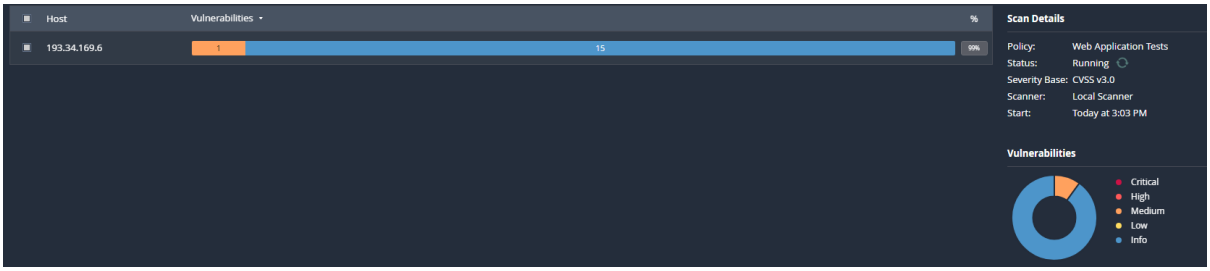
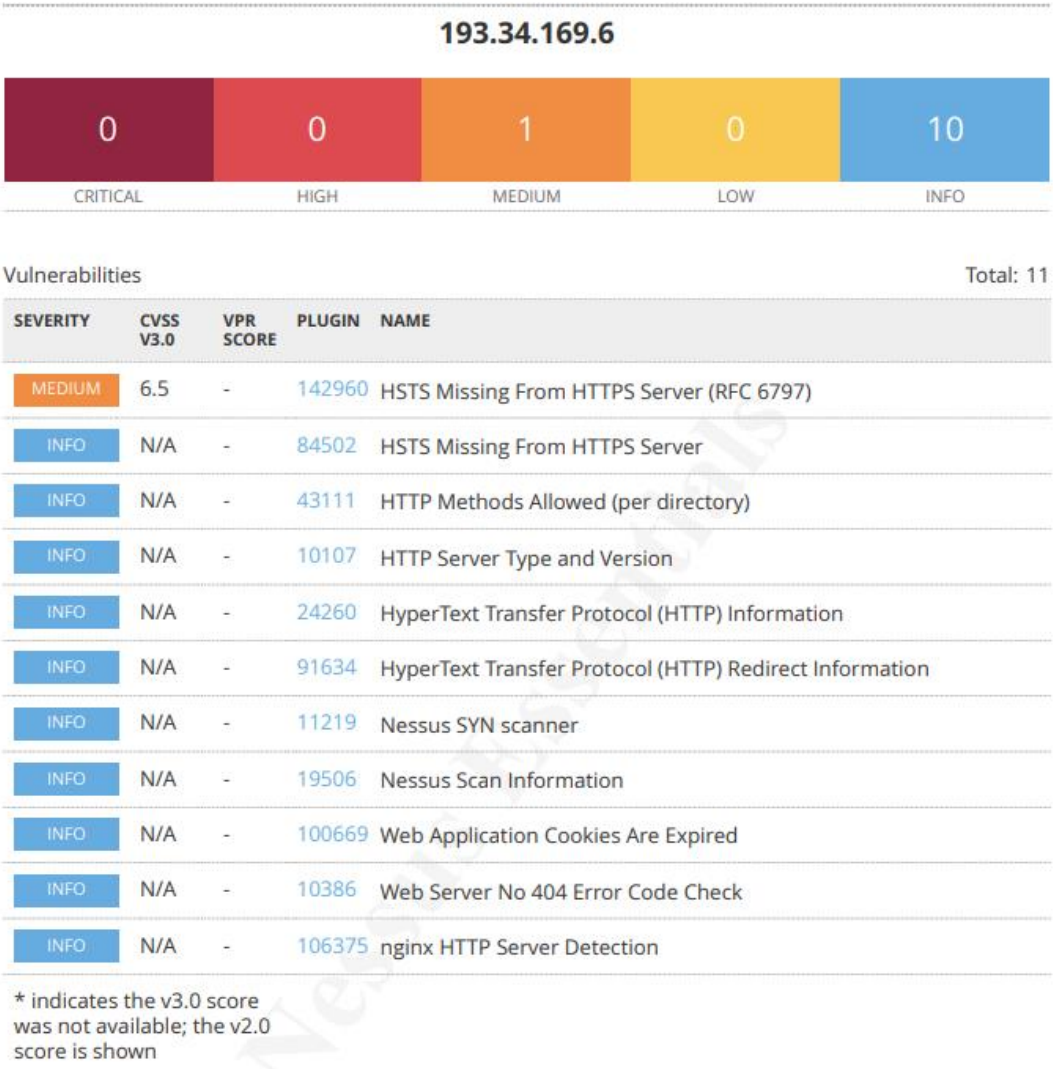


Рисунок 2.17 - Сканування сайту “prom” на вразливості

4) Результати сканування не можна подивитися напряму у Інтерфесі програми, що теж є мінусом, так як для того, щоб продивитися помилки – мені потрібно скачувати PDF файл, що призводить до великої кількості мусора у системі користувача (Рисунок 2.18). Хочу відзначити, що сканер вразливості Nessus знайшов набагато менше помилок та вразливостей, ніж OWASP ZAP:



193.34.169.6

0	0	1	0	10
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 11

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
MEDIUM	6.5	-	142960	HSTS Missing From HTTPS Server (RFC 6797)
INFO	N/A	-	84502	HSTS Missing From HTTPS Server
INFO	N/A	-	43111	HTTP Methods Allowed (per directory)
INFO	N/A	-	10107	HTTP Server Type and Version
INFO	N/A	-	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	-	91634	HyperText Transfer Protocol (HTTP) Redirect Information
INFO	N/A	-	11219	Nessus SYN scanner
INFO	N/A	-	19506	Nessus Scan Information
INFO	N/A	-	100669	Web Application Cookies Are Expired
INFO	N/A	-	10386	Web Server No 404 Error Code Check
INFO	N/A	-	106375	nginx HTTP Server Detection

* indicates the v3.0 score was not available; the v2.0 score is shown

Рисунок 2.18 - Результат сканування

2.9 Система WAF

WAF або Міжмережний екран (він же брандмауер чи фаєрвол) — це компонент безпеки, який контролює мережний трафік, що надходить і виходить із бекенд-системи. Міжмережний екран застосунків (application firewall) спеціально контролює вхідний та вихідний трафік, доступ до програмних застосунків чи сервісів, працює з трафіком (HTTP/S, TCP, UDP тощо), який надходить чи виходить із вебзастосунків.

Отже, WAF відстежує та фільтрує двонаправлений трафік до вебзастосунків. Вхідний трафік складається з легітимних запитів користувачів і запитів від можливих кіберзлочинців. WAF ідентифікує та блокує підозрілі запити, дозволяючи проходити легітимним запитам.

2.9.1 Атаки, які блокує WAF

Сьогодні надійний WAF має виявляти та блокувати широкий спектр атак. Ось кілька прикладів:

- SQL Injection — спроба ввести SQL-команди в керовану даними програму через поле введення.
- OSCI (впровадження команд операційної системи) — спроба виконати команди операційної системи на сервері програми.
- RFI (включення віддаленого файлу) — спроба змусити вебпрограму завантажити та запустити файл дистанційно.
- LFI (включення локального файлу) — спроба змусити вебпрограму виконати код у локальному файлі, який був розміщений або змінений зловмисником.
- XSS (Cross-Site Scripting) — спроба вставити шкідливий скрипт у вебзастосунок на стороні клієнта. Шкідливий код може бути вставлений до сторінки як через вразливість у вебсервері, так і через вразливість на комп'ютері користувача.

Крім того, WAF може забезпечити симетричну фільтрацію шляхом очищення не лише вхідних запитів, але й вихідного трафіку. Зловмисний вихідний трафік може генеруватися, наприклад, якщо в мережі інфіковано один з комп’ютерів. У

цьому разі комп’ютер може почати комунікувати з командним сервером ботнету або брати участь у DDoS-атаці. WAF здатний заблокувати цю діяльність і повідомити адміністраторів про проблему.

2.10 Еволюція WAF

У минулому WAF в основному використовувався для захисту сайтів. Пізніше, з розширенням використання HTTP/S, зросла й потенційна роль WAF. Сьогодні WAF важливий для захисту не лише традиційних бекенд-систем, як-от сайти, а й інших програм і служб, приміром RESTful API на основі HTTP/S, які використовуються для мобільних/власних застосунків.

WAF також розвивалися і в інших напрямках. Спочатку продукти WAF підтримували лише негативну модель безпеки. У цьому випадку набори правил WAF визначають характеристики нелегітимного трафіку. Вхідні запити, які відповідають цим характеристикам, блокуються. Увесь інший трафік дозволено за замовчуванням.

Пізніше було додано позитивну модель безпеки. Це означає, що WAF перевіряє вхідний трафік, і, щоб запит був дозволений, він має відповідати певним характеристикам, які означають дозволений трафік.

Останнім часом WAF (такі як AWS WAF) набувають функцій, які виходять за рамки традиційних можливостей негативних і позитивних моделей безпеки.

Якщо вас цікавить потужне та надійне рішення для веббезпеки, якщо ви хочете захистити свій бізнес від потенційних кіберзагроз, напишіть про це в Мегатрейд: soft@megatrade.ua. Ми надамо консультацію та організуємо можливість ознайомитися з демонстраційною версією nxgen. Нагадаємо, що Мегатрейд є офіційним дистриб’ютором Reblaze в Україні.

2.11 Як працює WAF

Міжмережний екран вебзастосунків (WAF) зазвичай розгортається перед серверною мережею, яку він захищає. Найбільш поширена і, як правило, найефективніша конфігурація — зворотний проксі-сервер, який виконує ретрансляцію запитів клієнтів із зовнішньої мережі на сервери, що логічно

розташовані у внутрішній мережі. WAF слугує посередником між клієнтами та серверною мережею.

Коли WAF розгортається як зворотний проксі-сервер, клієнти не взаємодіють безпосередньо з бекенд-системою. Натомість вони комунікують тільки з WAF. Зазвичай клієнти й не підозрюють, як здійснюється процес комунікацій, адже для них це відбувається невидимо і безшовно.

Вхідні запити клієнта та вихідні відповіді сервера проходять через WAF в обох напрямках. Це дає змогу WAF зупиняти трафік, що порушує його безпекову політику. Він блокує увесь трафік, який вважається ворожим або забороненим з будь-якої причини.

WAF може фільтрувати трафік відповідно до таких стратегій: Негативна модель безпеки

- Позитивна модель безпеки
- Розширені можливості

2.11.1 Негативна модель безпеки

WAF аналізує та очищає трафік, застосовуючи набори правил до запитів. Традиційні WAF були засновані на негативній моделі безпеки: міжмережний екран дозволяє всі вхідні запити, якщо вони не відповідають визначеним сигнатурам загроз або іншим чином не порушують правила безпеки. Фактично йдеться про «чорний список» запитів [9].

Негативна модель безпеки має чимало недоліків, у тому числі:

- Вона не здатна захистити від експлойтів нульового дня або інших атак, які ще не були додані до бази даних загроз.
- Зловмисники можуть обійти фільтрацію WAF, модифікувавши атаку настільки, щоб вона більше не відповідала відомим сигнатурам або уникнути блокування іншими способами.
- Вона не може захистити від всіх типів атак. Так, серед 10 основних ризиків безпеки вебзастосунків, виділених проєктом OWASP*, три не можуть бути ефективно нейтралізовані за допомогою методу «чорних списків», а саме: A2 [порушення автентифікації], A5 [порушення контролю доступу]

та A7 [міжсайтовий скриптинг]. І навіть ті ризики, які можуть бути охоплені правилами негативної моделі безпеки, наприклад A1 [Injection], часто не реалізовані досить глибоко, щоб забезпечити дійсно надійний захист.

2.11.2 Позитивна модель безпеки

Для надійного захисту міжмережний екран вебзастосунків (наприклад, AWS Web Application Firewall) має використовувати позитивну модель безпеки. Запити, які пройшли фільтрацію за «чорними списками», додатково перевіряються, щоб визначити, чи відповідають вони правилам легітимних запитів користувача. Якщо виявлено відхилення, джерело трафіку може бути миттєво заблоковано або (залежно від типу аномалії) все ж таки дозволено, але зазнає більш ретельної перевірки в майбутньому з меншим допустимим відхиленням від заданих правил.

2.11.3 Розширені можливості, крім негативної чи позитивної моделі безпеки

Як згадувалося вище, спочатку робота WAF була заснована на негативних моделях безпеки. Позитивні моделі безпеки були запроваджені пізніше [9].

Сьогодні деякі WAF мають досконаліші можливості. Хоча в цілому це також варіації на базі негативної чи позитивної моделей безпеки, вони досить сильно відрізняються за своїм підходом та ефективністю, тому їх зазвичай відносять до окремої категорії.

2.12 Міжмережеві екрани

Хоча рішення Web Application Firewall (WAF) стали популярними в світі, і компанії, такі як Amazon, використовують захищені сервіси на основі продуктів цього типу, в Україні ще не так великий досвід використання WAF [9].

FortiWeb від Fortinet призначений для середніх і великих компаній, а також сервіс-провайдерів. Цей продукт використовує двосторонній захист від впровадження коду, такого як SQL-ін'єкції та міжсайтовий скриптинг, що дозволяє запобігти крадіжкам особистих даних, фінансовим махінаціям та промисловому шпигунству. FortiWeb надає інструменти для моніторингу і відповідності корпоративній політиці безпеки, а також прискорює роботу веб-додатків шляхом

балансування навантаження на мережу та оптимізації використання мережевих ресурсів.

Fortinet використовує власні технології у своїх рішеннях - спеціалізовані процесори FortiASIC для прискорення обробки і шифрування даних, а також операційну систему FortiOS, оптимізовану під завдання безпеки.

З переваг FortiWeb є фіксована вартість, незалежна від кількості користувачів. Компаніям достатньо придбати пристрій один раз з невеликим запасом по потужності, і їм не потрібно буде витрачати кошти на нове обладнання при збільшенні кількості співробітників.

Також, Fortinet систематично виявляє нові загрози у своєму центрі FortiGuard Security Analytics, що призводить до оновлення сигнатур та чорних списків сайтів кілька разів на день.

Також слід відзначити тісну інтеграцію всіх пристроїв Fortinet, що дозволяє швидко та просто масштабувати систему. Ці системи відрізняються високим рівнем автоматизації операцій і простотою супроводу, що дозволяє уникнути помилок, викликаних людським фактором, і зменшити кількість обслуговуючого персоналу.

Imperva SecureSphere Web Firewall Application є одним з лідерів у захисті від атак на веб-додатки. Його рішення використовують кілька технологій в області безпеки: сигнатурний аналіз, переві

Сімейство продуктів SecureSphere WAF ефективно бореться як з усіма десятьма найбільш відомими атаками OWASP-TOP10, так і з іншими менш розповсюдженими, але більш складними загрозами. Ці пристрої мають можливість перевіряти зашифровані дані, що передаються через SSL (HTTPS) протокол.

Пакет включає в себе такі компоненти [9]:

- SecureSphere Web Firewall Application - захист веб-додатків від кібератак;
- ThreatRadar - база даних з репутацією.

Secure-Sphere WAF вміє глибоко аналізувати внутрішню структуру законної роботи веб-додатків, проводити інтелектуальні дослідження спроб проникнення і атак через HTTP.

Воно вміє протидіяти атакам, таким як переповнення буфера, шкідливі програми та дії зловмисників. Це рішення оснащене механізмом захисту від черв’яків та інших шкідливих атак на веб-сервери та додатки.

Його базу складають механізми на основі сигнатур, взятих з відомої системи Snort, і власні SQL-сигнатури, які розробляються центром досліджень ADC (Application Defense Center) компанії Imperva. Вбудований міжмережевий екран забезпечує надійний захист від несанкціонованих запитів користувачів і атак на рівні мережі.

Система вже напередодні готова звітувати відповідно до норм і стандартів інформаційної безпеки. Це включає можливість створення звітів, спеціально призначених для користувачів, і навіть графіків, а також можливість експорту в різні формати. Також, система пропонує додаткові хмарні сервіси, що спрощують безпеку та дозволяють впоратися з DDoS-атаками. [9]

Одна з ключових переваг у пристроїв SecureSphere WAF — наявність виняткового сервісу ThreatRadar, який забезпечує захист від автоматизованих атак. Завдяки оперативному отриманню достовірної інформації про джерела атак, ThreatRadar надає можливість миттєво блокувати трафік, що надходить від підозрілих джерел, ще до того, як будь-які руйнівні дії будуть здійснені. Рішення від Imperva виділяються своєю чіткою підтримкою та простим впровадженням. [9]

2.13 OpenVAS

OpenVAS представляє собою комплексний інструмент для пошуку слабкостей у системах. Він може виконувати неавтентифіковані та автентифіковані тести, працювати з різними типами Інтернет і промислових протоколів, оптимізувати продуктивність для широкомасштабного сканування та використовувати потужну внутрішню мову програмування для реалізації будь-яких видів тестів на вразливість.

Цей сканер супроводжується широким спектром тестів на вразливість з великою історією та щоденними оновленнями, належить компанії Greenbone Networks і є ключовою складовою частиною їхнього комерційного пакету

управління вразливостями, які об'єднуються під назвою Greenbone Security Manager (GSM).

Цей інструмент спільноти з відкритим кодом під ліцензією GNU GPL має багатофункціональне використання у поєднанні з плагінами, що розширюють його можливості та роблять його складовою частиною більш широкого рішення для управління вразливостями Greenbone. Це рішення GVM (Greenbone Vulnerability Management) включає додаткові функції, керування пристроями та угоди щодо обслуговування, що задовольняють потреби підприємства.

2.14 Аналіз системи CVSS 3.0

Система оцінки вразливостей (CVSS) є відкритим стандартом для передачі характеристик та важкості програмних вразливостей. CVSS складається з трьох груп метрик: базової, тимчасової та екологічної. Базова група описує 16 внутрішніх аспектів вразливості, тимчасова група відображає її характеристики, що змінюються з часом, а екологічна група відтворює аспекти вразливості, унікальні для конкретного користувача чи середовища. Базова оцінка має діапазон від 0 до 10 і може змінюватись під впливом часових та екологічних аспектів. Результат оцінки CVSS також представляється у вигляді векторного рядка - короткого текстового виразу, який відображає ці значення для подальшої інтерпретації.

Основні метрики визначають складність експлуатації вразливості та потенційні наслідки для конфіденційності, цілісності та доступності інформації.

- Вектор атаки – це спосіб вимірювання відстані між потенційним зловмисником та уразливим об'єктом. Ця метрика може включати такі параметри: Network (N), Adjacent Network (A), Local (L), Physical (P).
- Межі експлуатації визначають, чи різняться види компонентів, які можуть бути скомпрометовані, тобто чи порушується цілісність, конфіденційність і доступність інших компонентів системи при використанні вразливості. Метрика може включати такі параметри: Unchanged (U), Changed (C).
- Потреба у взаємодії з користувачем визначає, чи необхідні будь-які дії від користувача системи для успішного виконання атаки на цю систему. Ця метрика може мати такі параметри: Не потрібна (N), Обов’язкова (R).

- Перевірка автентифікації та рівень необхідних привілеїв визначають, чи необхідна процедура автентифікації для здійснення атаки і, у разі необхідності, які саме типи автентифікації. Ці параметри можуть мати такі значення: High (H), Low (L), None (N).
- Метрики впливу визначають рівень важливості для конфіденційності, доступності та цілісності атакованого компонента. Ця метрика може бути визначена наступними значеннями: Medium (M), High (H).
- Складність експлуатації вразливості визначається рівнем технічної складності для здійснення атаки. Впровадження цієї складності напряму залежить від умов, які вимагає сама система. Ця метрика може мати такі параметри: Low (L), High (H) [6].

Відповідно до показника оцінки є таблиця відповідності рейтинга, ці значення наведено в таблиці 2.1.

Таблиця 2.3 – Значення рейтингу вразливості

None	0.0
Low	0.1 – 3.9
Medium	4.0 – 6.9
High	7.0 – 8.9
Critical	9.0 – 10.0

3 РЕКОМЕНДАЦІЇ ЩОДО ЗАПОБІГАННЯ АТАКАМ ВЕБ-ДОДАТКІВ

Атаки на веб-додатки - це різні види вторгнень та зловживань, спрямованих на веб-додатки, щоб отримати несанкціонований доступ, витягти конфіденційну інформацію, змінити поведінку програми або виконати інші шкідливі дії.

Забезпечення кібербезпеки веб-додатків є критично важливим завданням для збереження конфіденційності, цілісності та доступності даних. Використання сучасних сканерів вразливостей може значно полегшити цей процес.

3.1 Встановлення та регулярне сканування

Сканер вразливостей - це інструмент, який автоматизує процес виявлення потенційних слабких місць в програмному забезпеченні, операційних системах, мережевих пристроях та інших компонентах системи. Сканування вразливостей дозволяє виявляти слабкі місця перед тим, як вони можуть бути використані для здійснення атаки.

У багатьох галузях, таких як фінанси, охорона здоров'я та інші, існують вимоги до забезпечення високого рівня безпеки та конфіденційності інформації. Встановлення сканерів вразливостей допомагає відповідати цим вимогам.

Сканування вразливостей дозволяє спрямовувати ресурси на конкретні області, де існують реальні загрози безпеці, замість того, щоб витрачати їх на загальні заходи безпеки.

Проводьте сканування вразливостей регулярно, інтегруйте цей процес у вашій системі CI/CD, щоб автоматизувати тестування безпеки під час розробки і релізу.

Виберіть надійний сканер вразливостей, який підтримує автоматичне виявлення різних типів вразливостей, таких як SQL-ін'єкції, XSS атаки, CSRF, та інших.

3.2 Ручне тестування

Ручне тестування - це процес, при якому тестери вручну перевіряють функціональність програмного продукту, взаємодіючи з ним як з реальним користувачем. Такий підхід включає в себе запуск програми, проведення різних

тестових сценаріїв, введення даних, аналіз реакції програми та перевірку відповідності результатів очікуваним.

Люди можуть виявити проблеми, які не легко автоматизувати або передбачити. Ручне тестування дозволяє тестерам використовувати свій інтуїтивний розум та досвід для пошуку унікальних вад та сценаріїв використання, які можуть бути важко зафіксувати автоматичними тестами.

Ручне тестування може бути ефективним при валідації нефункціональних вимог, таких як продуктивність, масштабованість, надійність та інші. Також, воно може бути корисним при виявленні потенційних проблем безпеки, оскільки тестувальники можуть враховувати нюанси, які не завжди легко автоматизувати. Існують випадки, коли певні типи помилок або неправильне функціонування може бути важко виявити за допомогою автоматизованих тестів. Ручне тестування дозволяє фахівцям з тестування ефективно виявляти такі аспекти.

Хоча автоматизоване тестування може прискорити процес та полегшити деякі завдання, ручне тестування залишається критичним елементом в розробці програмного забезпечення, доповнюючи автоматизовані зусилля та допомагаючи забезпечити високий рівень якості продукції.

3.3 Моніторинг та редагування

Моніторинг та реагування - це важливі аспекти забезпечення безпеки інформаційних систем. Вони допомагають виявляти потенційні загрози та атаки на веб-додатки, а також надають можливість вчасно реагувати на інциденти.

Ведення журналу подій є ключовим для моніторингу безпеки. Лог-файли можуть містити інформацію про намагання несанкціонованого доступу, помилки системи, аномалії в роботі веб-додатків та інші події, які можуть бути індикаторами атак чи проблем безпеки.

Системи виявлення вторгнень (Intrusion Detection Systems - IDS) спостерігають за трафіком мережі та журнальними подіями з метою виявлення аномалій чи знаків можливих атак. Вони можуть бути налаштовані для виявлення специфічних вразливостей веб-додатків. Автоматизовані сканери вразливостей

можуть регулярно сканувати веб-додатки для виявлення потенційних слабких місць та вразливостей, які можуть бути використані зловмисниками.

Розробка та впровадження планів реагування на інциденти, які включають кроки виявлення, реагування, відновлення та аналіз інциденту, допомагає зменшити вплив атак та забезпечити швидке відновлення системи після інциденту. Системи захисту веб-додатків (Web Application Firewalls - WAF) можуть виявляти та блокувати шкідливий веб-трафік, фільтрувати запити та захищати веб-додатки від різних типів атак, таких як SQL-ін'єкції, кросс-сайт-скриптинг та інші.

3.4 Актуальність та оновлення

Кіберзагрози постійно еволюціонують, вимагаючи постійного розвитку технологій захисту. Оновлені сканери вразливостей здатні виявляти нові методи атак, які можуть залишатися невідомими старішим версіям. Це допомагає уникнути витоку даних та атак, що можуть серйозно підірвати безпеку користувачів, компаній та організацій. Оновлення сканерів є ключовим факторам в забезпеченні безпеки веб-застосунків, бо це дозволяє своєчасно виявляти та усувати вразливості, захищаючи дані та веб-сервери від можливих атак.

Актуальність та постійне оновлення сканерів вразливостей стають критичними у сфері кібербезпеки. Оскільки загрози постійно змінюються, атаки еволюціонують, сканери повинні підтримувати свій розвиток, щоб виявляти нові вразливості та потенційні загрози. Актуальність означає, що сканери повинні бути орієнтовані на останні відомі вразливості та методи атак, маючи оновлену базу даних із сховищем інформації про вразливості. Тільки постійне оновлення та вдосконалення дозволяють сканерам ефективно захищати веб-застосунки від останніх загроз.

ВИСНОВОК

Захист веб-додатків у сучасному світі має вирішальне значення, оскільки вони містять особисту інформацію, таку як паролі, фінансові дані та медична інформація. Недостатній захист може призвести до витоку цих даних і порушити конфіденційність осіб. Дослідження, проведені у роботі, допомагають уникнути атак, захистити дані та вдосконалити практики кібербезпеки.

Сканування вразливостей виявляє потенційні проблеми в веб-додатках, такі як помилки в програмному коді чи проблеми з аутентифікацією, які можуть бути використані для атак. Такі сканери також вчасно виявляють загрози і типи атак, наприклад, крос-сайтовий скриптинг або SQL-ін'єкції, що дозволяє приймати відповідні заходи безпеки. Вони також відповідають галузевим стандартам і вимогам законодавства, які передбачають регулярні аудити безпеки веб-додатків.

Сучасні сканери вразливостей веб-застосунків допомагають виявляти та усувати потенційні загрози у програмному коді та інфраструктурі. Однак жоден сканер не є ідеальним, і важливо систематично оновлювати заходи захисту для ефективної кібербезпеки.

Важливо пам'ятати, що кібербезпека - це постійний процес. Хоч сканери вразливостей і є потужним інструментом для забезпечення безпеки, зловмисники постійно шукають нові шляхи атаки. Тому крім використання сканерів, актуальність, оновлення та аналіз систем захисту залишаються ключовими складовими успішної стратегії кібербезпеки. Тільки поєднання новітніх інструментів та постійного вдосконалення заходів безпеки дозволить ефективно захищати веб-застосунки від сучасних загроз.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. OWASP Secure Coding Practices Quick Reference Guide. // OWASP. – 2010.
2. OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks, 2017.
3. Django documentation [Електронний ресурс] // Django Software Foundation. – 2016. – Режим доступу до ресурсу: https://docs.djangoproject.com/en/2.2/. (дата звернення: 15.092023)
4. OWASP ZAP 2.6 Getting Started Guide, 2013. – (OWASP).
5. Middleware [Електронний ресурс] // Django Software Foundation. – 2016. – Режим доступу до ресурсу: https://docs.djangoproject.com/en/2.2/topics/http/middleware/.(дата звернення: 27.10.2023)
6. N. Jovanovic. TxtForum: Script Injection Vulnerability. http://www.seclab. tuwien.ac.at/advisories/TUVSA-0603-004.txt, March 2006.
7. A. Klein. Cross Site Scripting Explained. Technical report, Sanctum. Inc., 2002
8. K. Fu, E. Sit, K. Smith, and N. Feamster. Dos and Don’ts of Client Authentication on the Web. – 2011. - Режим доступа: World Wide Web. — URL: https://pdos.csail.mit.edu/papers/webauth:sec10.pdf. (дата звернення: 19.11.2023)
9. ITBIZ [Електронний ресурс] // Захист веб-додатків: чому це важливо?. – 2023. – Режим доступу до ресурсу: https://itbiz.ua/statti-ta-obzori/zaxist-veb-dodatkiv-chomu-ce-vazhlivo/ .(дата звернення: 27.10.2023)