ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«ТЕХНОЛОГІЯ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ WEB-ДОДАТКІВ НА ОСНОВІ BURP SUITE»

на здобуття освітнього ступеня магістра
зі спеціальності ____125 Кібербезпека_____
(код, найменування спеціальності)
освітньо-професійної програми _Інформаційна та кібернетична безпека_
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____ Микита ВИСОТІН

Виконав: здобувач(ка) вищої освіти групи БСДМ-62
ВИСОТІН Микита
(ПРІЗВИЩЕ, Ім'я)

Керівник: БОРСУКОВСЬКИЙ Юрій
д.т.н, доцент (ПРІЗВИЩЕ, Ім'я)

Рецензент: _____
(ПРІЗВИЩЕ, Ім'я)

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

Кафедра Інформаційної та кібернетичної безпеки
Ступінь вищої освіти Магістр
Спеціальність 125 Кібербезпека
Освітньо-професійна програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКБ
_____ Галина ГАЙДУР
“___”_____ 2023 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Висотіну Микиті Дмитровичу
(прізвище, ім’я, по батькові)

1. Тема кваліфікаційної роботи:
«Технологія виявлення вразливостей web-додатків на основі Burp Suite»
керівник кваліфікаційної роботи: БОРСУКОВСЬКИЙ Юрій, д.т.н., доцент,
(ПРІЗВИЩЕ, Ім’я, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023р. №145.

2. Строк подання студентом кваліфікаційної роботи: 15.12.2023 р.

3. Вихідні дані до кваліфікаційної роботи:
інформаційна система організації;
технологія виявлення вразливостей web-додатків на основі Burp Suite;
наукова та технічна література, експлуатаційна документація, нормативні документи, міжнародні стандарти.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Актуальність виявлення врзливостей у Web-додатків.
2. Зміст процесу виявлення вразливостей у Web-додатків(тестування).
3. Методи та засоби для виявлення вразливостей у Web-додатків.

4. Рекомендації щодо застосування методів та засобів для виявлення вразливостей у Web-додатків під час тестування.

5. Перелік ілюстративного матеріалу:
Презентація PowerPoint

6. Дата видачі завдання 19.10.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Визначення актуальності тестування та виявлення вразливостей Web-додатків.	19.10.2023 р.	
2.	Аналіз наукової та технічної літератури з питань теми бакалаврської роботи.	22.10.2023 р.	
3.	Аналіз методів та засобів виявлення вразливостей у Web-додатків.	27.10. 2023р.	
4.	Рекомендацій щодо методів та засобів виявлення вразливостей у Web-додатків.	03.11.2023 р.	
5.	Розроблення варіантів тестування web-додатків на вразливості.	15.11.2023 р.	
6.	Оформлення результатів дослідження. Проходження плагіату	26.11.2023 р.	
7.	Підготовка доповіді до захисту.	15.12.2023 р.	

Здобувач(ка) вищої освіти _____ Микита ВИСОТІН
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Юрій БОРСУКОВСЬКИЙ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

ПОДАННЯ
ГОЛОВІ ДЕРЖАВНОЇ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Висотін М.Д. до захисту кваліфікаційної роботи
(прізвище та ініціали)

За спеціальністю 125 Кібербезпека
освітньо-професійної програми Інформаційна та кібернетична безпека
(шифр і назва спеціальності)

на тему: «Технологія виявлення вразливостей web-додатків на основі Burp Suite».

Кваліфікаційна робота і рецензія додаються.

Директор інституту _____ Віталій САВЧЕНКО
(*підпис*) (*Ім'я, ПРІЗВИЩЕ*)

Висновок керівника кваліфікаційної роботи

Здобувач ***ВИСОТІН Микита*** обрав тему роботи, метою якої було дослідити технології виявлення вразливостей у web-додатків. Перелік використаних джерел свідчить про вміння здобувача розбиратись в наукових питаннях та застосовувати їх при дослідженнях. Під час виконання кваліфікаційної роботи ***ВИСОТІН Микита*** показав гарну теоретичну та практичну підготовку, вміння самостійно вирішувати питання і робити висновки. Роботу виконував сумлінно, акуратно та вчасно за планом.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача ***ВИСОТІН Микити*** на оцінку **«добре»** та присвоїти йому кваліфікацію магістр з кібербезпеки за спеціалізацією інформаційна та кібернетична безпека.

Керівник кваліфікаційної роботи _____ Юрій БОРСУКОВСЬКИЙ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*
"____" _____ 2023 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач(ка) ***ВИСОТІН Микита.*** допускається до захисту даної роботи в Державній екзаменаційній комісії.

Завідувач кафедри Інформаційної та кібернетичної безпеки
(назва)

_____ Галина ГАЙДУР
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу

здобувача Висотін Микити
на тему: «Технологія виявлення вразливостей web-додатків на основі Burp Suite».

Актуальність:

Автор добре орієнтується в проблематиці та чітко використовує Burp Suite для виявлення потенційних загроз та недоліків веб-додатків. Робота містить важливі практичні приклади та реалістичні сценарії тестування, що підкреслює практичну цінність представленого матеріалу. Також важливо відзначити структурованість та логічний підхід до викладення інформації. Робота заслуговує на високу оцінку за систематизацію знань та якісне їх застосування у сфері виявлення вразливостей веб-додатків.

Позитивні сторони:

1. На основі проведеного аналізу, в роботі встановлено зміст проблем з інформаційною безпекою web-додатків.
2. Досліджено методи та засоби тестування web-додатків на вразивості.
3. Запропоновано варіант технології тестування web-додатків на основі Burp Suite/
4. Текст викладено достатньо грамотно, послідовно. Сформульовано чіткі та змістовні висновки. Графічний матеріал оформлено якісно. Список науково-технічної літератури свідчить про вміння користуватись матеріалами за темою магістерської роботи.

Недоліки:

1. У кваліфікаційній роботі доцільно було б більш детально описати методологію тестування web-додатків.

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи

Висновок: Враховуючи недоліки, кваліфікаційна робота заслуговує оцінку «**добре»**, а здобувач **ВИСОТІН Микита** - присвоєння кваліфікації магістр з кібербезпеки за спеціалізацією інформаційна та кібернетична безпека.

Рецензент:
д.т.н., професор _____ *підпис* _____ *Ім'я, ПРІЗВИЩЕ*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи и на здобуття освітнього ступеня магістра: 76 сторінок, 18 рисунків, 2 таблиці, 14 джерел.

Об'єкт дослідження – виявлення вразливостей веб-додатків.

Предмет дослідження – технологія використання Burp Suite для аналізу та ідентифікації потенційних загроз безпеці веб-додатків.

Мета роботи – головною метою даної магістерської роботи є розробка та вивчення ефективної технології для виявлення вразливостей веб-додатків з використанням інструменту Burp Suite. Робота спрямована на розширення знань в області кібербезпеки та вдосконалення методів захисту веб-додатків від потенційних атак.

Методи дослідження – для досягнення поставленої мети використовуються такі методи дослідження, як аналіз літературних джерел з кібербезпеки, вивчення принципів роботи Burp Suite та його інтеграція у процес виявлення вразливостей. Також використовуються практичні експерименти для перевірки ефективності розробленої технології на реальних веб-додатках. Аналіз отриманих результатів та порівняння з існуючими методами дозволяють зробити висновки щодо ефективності запропонованої технології виявлення вразливостей.

Літературний аналіз: Ретельний огляд наукових статей, книг та онлайн-ресурсів з кібербезпеки та Burp Suite.

Практичні експерименти: Розробка та реалізація експериментальних сценаріїв використання Burp Suite для виявлення вразливостей веб-додатків.

Інтеграція Burp Suite в реальні веб-проекти: Застосування розробленої технології до реальних веб-додатків з метою виявлення та усунення ідентифікованих уразливостей.

Аналіз результатів: Обробка та аналіз отриманих даних, порівняння результатів з існуючими методами виявлення вразливостей.

Отримані результати підтверджують ефективність розробленої технології та надають практичні рекомендації щодо підвищення безпеки веб-додатків.

КОРПОРАТИВНА ІНФОРМАЦІЙНА МЕРЕЖА, КІБЕРБЕЗПЕКА, WEB-ДОДАТКИ, МЕТОДИ ТА ЗАСОБИ, ТЕСТУВАННЯ, АНАЛІЗ, АРХІТЕКТУРА, ТЕХНОЛОГІЇ

ABSTRACT

Text part of the master's qualification work:76 pages, 18 figures, 2 tables, 14 sources.

The purpose of the work is to develop options for network control management technology based on the Cisco Identity Services Engine solution for the organization's information system and recommendations for using the technology.

Object of research - is the detection of web application vulnerabilities.

Subject of research - is the technology of using Burp Suite to analyze and identify potential threats to the security of web applications. The work is aimed at expanding knowledge in the field of cyber security and improving methods of protecting web applications from potential attacks.

Research methods - to achieve the goal, such research methods as the analysis of literary sources on cyber security, the study of the working principles of Burp Suite and its integration into the vulnerability detection process are used. Practical experiments are also used to test the effectiveness of the developed technology on real web applications. Analysis of the obtained results and comparison with existing methods allow us to draw conclusions about the effectiveness of the proposed vulnerability detection technology.

Literature Review: A thorough review of scholarly articles, books, and online resources on cybersecurity and the Burp Suite.

Practical experiments: Development and implementation of experimental scenarios for using Burp Suite to detect vulnerabilities in web applications.

Integration of Burp Suite into real web projects: Applying the developed technology to real web applications in order to identify and eliminate identified vulnerabilities.

Analysis of results: Processing and analysis of received data, comparison of results with existing methods of detecting vulnerabilities.

The obtained results confirm the effectiveness of the developed technology and provide practical recommendations for improving the security of web applications.

CORPORATE INFORMATION NETWORK, CYBER SECURITY, WEB APPLICATIONS, METHODS AND TOOLS, TESTING, ANALYSIS, ARCHITECTURE, TECHNOLOGIES

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

WAF - Веб-протокол захисту (Web Application Firewall)

SQLi - Внедрення коду SQL (SQL Injection)

XSS - Міжсайтовий сценарій (Cross-Site Scripting)

CSRF - Міжсайтова фішингова атака (Cross-Site Request Forgery)

RCE - Виконання коду на віддаленому сервері (Remote Code Execution)

API - Інтерфейс програмування застосунків (Application Programming Interface)

HTTP - Протокол передачі гіпертексту (Hypertext Transfer Protocol)

TLS/SSL - Протоколи захищеної передачі даних (Transport Layer Security/Secure Sockets Layer)

OWASP - Відкритий проект з безпеки веб-додатків (Open Web Application Security Project)

Fuzzer - Інструмент для випадкового або систематичного тестування на вразливості

IDE - Інтегроване середовище розробки (Integrated Development Environment)

HTML - Мова розмітки гіпертексту (Hypertext Markup Language)

Burp Suite - Інструмент для тестування на безпеку веб-додатків та виявлення вразливостей

CORS - Політика обміну ресурсами між різними доменами (Cross-Origin Resource Sharing)

GUI - Графічний інтерфейс користувача (Graphical User Interface)

Вступ

Сучасний розвиток інформаційних технологій визначає нові вимоги до безпеки веб-додатків, які стають предметом постійного атак та загроз. Актуальність проблеми безпеки веб-додатків стає надзвичайно важливою, оглядаючи той факт, що вони є ключовим елементом в електронному взаємодії та обміні інформацією.

Об'єкт дослідження: Об'єктом даної магістерської роботи є процес виявлення та усунення вразливостей веб-додатків з використанням Burp Suite, відомого інструмента для тестування на безпеку веб-додатків.

Мета роботи: Основною метою є розробка та дослідження технології виявлення вразливостей, яка базується на використанні Burp Suite. Робота спрямована на покращення методів захисту веб-додатків та забезпечення їхньої надійності.

Наукові завдання:

1. Вивчення сучасних тенденцій у сфері безпеки веб-додатків.
2. Аналіз можливостей та функціоналу Burp Suite для виявлення вразливостей.
3. Розробка ефективної методології використання Burp Suite у процесі тестування на безпеку.
4. Практична апробація розробленої технології на реальних веб-проектах.

Методи дослідження: Для досягнення поставлених завдань використовуються методи аналізу літературних джерел, експериментальні практичні дослідження, інтеграція Burp Suite в реальні веб-проекти, а також аналіз отриманих результатів.

Практичне значення одержаних результатів: Розробка та впровадження в практику ефективної технології виявлення вразливостей веб-додатків може значно

підвищити рівень їхньої безпеки, сприяючи захисту від потенційних кібератак та забезпечуючи конфіденційність інформації.

Апробація результатів: Отримані результати будуть представлені та обговорені на відповідних конференціях і семінарах у галузі кібербезпеки та інформаційних технологій.

1 ВИЗНАЧЕННЯ ТЕХНОЛОГІЙ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ

1.1. Загальний огляд вразливостей web-додатків.

З кожним роком зростає кількість підприємств, які використовують веб-технології для підвищення продуктивності та залучення нових клієнтів. Хоча Інтернет-сервіси мають багато переваг, кількість кіберзагроз збільшується разом із кількістю додатків. У своєму звіті Global Internet Security Threat Report (ISTR) Symantec стверджує, що кіберзлочинці, які зломлюють веб-сайти, зазвичай використовують вразливості веб-додатків або операційної системи. Наприклад, хакер може використовувати атаки типу XSS, щоб перенаправляти запити користувачів на шкідливі веб-сторінки, а також використовувати SQL-ін’єкції, щоб витягувати різні конфіденційні дані з баз даних сайту.

OWASP — відкритий проект щодо захисту веб-додатків — був створений у відповідь на масові зломи систем безпеки. Тим не менш, зловмисники та експерти з кібербезпеки все ще знаходять вразливості в веб-додатках, які можуть завдати значних втрат компаніям. Більшість взломів веб-додатків є результатом роботи розробників програмного коду. Розробники можуть створювати вразливі програми через помилки при написанні коду або через те, що вони не знають, наскільки важливо використовувати методи безпечного програмування.Безсумнівно, будь-яка компанія повинна захистити свою веб-інфраструктуру.

Тим не менш, із безлічі варіантів захисту, включаючи firewalls, IPS/IDS, NGFW (Next Generation Firewall) і WAF, лише веб-додаткові firewalls здатні забезпечити комплексний захист веб-додатків від відомих і невідомих загроз, а також забезпечити відповідність вимогам регуляторів, таким як PCI DSS. Нестандартні firewallи та IPS/IDS не можуть захистити веб-додатки достатньо.

Еволюція брандмауерів

Файрволи, також відомі як міжмережеві брандмауери екрана, служать фільтром між корпоративною мережею та Інтернетом. Використовуючи IP-адреси джерела та призначення, мітки фрагментації та номери портів, перші брандмауери могли відстежувати підозрілі пакети на мережевому та канальному рівнях. Системи запобігання вторгненню/виявлення вторгнень (IPS/IDS), наприклад, можуть аналізувати пакети мережі та порівнювати їх із сигнатурами атак, які відомі. Крім того, ці системи знаходять і блокують помилки прикладного рівня протоколів.

Тим не менш, сьогодні понад 80% атак є результатом уразливостей додатків, а не архітектури мережі. Таким чином, виявляється, що вищезгадана система захисту не може протистояти сучасним кібератакам. Крім того, зараз існує велика кількість онлайн-додатків, які можна використовувати. Кожен із них може бути слабким.Таким чином, загальна кількість вразливостей набагато більша, ніж кількість сигнатур, присутніх у базах сучасних IPS-систем. Оскільки проникнення через веб-додатки є основним фактором атак на корпоративні мережі, традиційні системи безпеки, такі як файрвол і антивірусна система, не можуть запобігти таким атакам.

Для надійного захисту необхідний значно інший підхід, який включає ретельний аналіз змісту пакетів і глибоке розуміння структур веб-додатків, таких як форми введення даних, печиво та URL-параметри. Програма Web Firewall — це брандмауер, який контролює програми, які передають дані через протоколи HTTP і HTTPS. Він відповідає цим вимогам.

WAF

Важливо пам’ятати, що WAF — це окрема система, яка використовує лише протоколи HTTP/HTTPS. Тим не менш, існує так багато способів обміну даними поверх протоколу HTTP, що використання спеціалізованого інструменту є необхідним. Крім того, більшість веб-файрволів підтримують трафік SSL. Аналітики Gartner стверджують, що здатність перевіряти зашифрований трафік є однією з головних відмінностей WAF від звичайних міжмережевих екранів і IPS.

WAF визначає атаки за допомогою сигнатур і дій. Крім того, другий метод є життєво важливим, оскільки кіберзлочинці можуть використовувати уразливість нульового дня, також відому як уразливість нульового дня, яка знижує ефективність сигнатурного аналізу. Тим часом WAF може визначити модель нормального функціонування програми за допомогою аналізу мережевого трафіку та системних журналів. Він також може використовувати ці дані для визначення невідповідностей у поведінці програмної системи.

WAF може виявити атаки, використовуючи автоматичні інструменти. Класичний файрвол генерує велику кількість помилкових спрацьовувань на кожну підозрілу подію. Щоб оцінити ступінь загрози, яка міститься в цих повідомленнях, їх слід розділити вручну. Тим не менш, WAF може аналізувати тисячі подій і створювати ланцюг розвитку атаки від початку до кінця.Як розпізнати вразливість:

Сучасні цифрові середовища вимагають розуміння вразливостей веб-додатків. Вразливість системи дозволяє несанкціонованим діям, таким як несанкціонований доступ, зміна чи видалення конфіденційної інформації.

Веб-Проникнення Test

При створенні системи захисту веб-додатків тест на проникнення є важливим організаційним моментом. Він стане найкращим підходом до перевірки захищеності інформаційної системи шляхом моделювання спрямованих атак. Тест на проникнення оцінює захист інформаційної системи від несанкціонованого впливу за допомогою кількох моделей вторгнень. Тест на проникнення, який використовується для веб-додатків, обмежений оцінкою ступеня захисту додатків. Активний аналіз додатків є частиною процесу пошуку технічних помилок або проблем. Підсумковий звіт охоплює всі недоліки.

Типи вразливостей:

Велика різноманітність вразливостей може включати такі елементи, як крос-сайтові скрипти (XSS), які дозволяють вбудовувати зловмисний код на веб-сторінки для виконання в браузері користувача, або вбудовані атаки, коли зловмисники впроваджують зловмисний код через вхідні дані. Інші види вразливостей включають витоки запитів через сайт (CSRF), коли зловмисники використовують авторизований сеанс користувача для небажаних дій, і неправильні налаштування системи, які можуть призвести до витоку конфіденційної інформації.

Наслідки вразливостей:

Вразливість має багато негативних наслідків. Це може включати втрату даних, втрату доступу до системи або сервісу, а також пошкодження чи зміну інформації, що може призвести до серйозних порушень безпеки.

Заходи проти вразливостей:

Заходи проти вразливостей є важливим компонентом безпеки. Це включає використання параметризованих запитів для запобігання внесенню атак, валідації та фільтрації вхідних даних для зменшення ризику XSS, а також регулярні аудити безпеки та вчасні патчі для усунення помилок безпеки.

Таблиця 1.1.

Загальний огляд вразливостей web-додатків.

Тип вразливості	Опис	Приклади
Injection Attacks	Атаки на введення зловмисного коду	SQL Injection, Command Injection
Cross-Site Scripting	Впровадження скриптів на веб-сторінку	Stored XSS, Reflected XSS
Cross-Site Request Forgery	Використання авторизованого сеансу	CSRF Token Bypass, Session Riding
Security Misconfigurations	Неправильна системна конфігурація	Default Credentials, Unprotected Resources

Рисунок 1.1.

Атаки зі списку OWASP Top 10 – 2019

Ін’єкція програмного коду та SQL-запитів, також відома як SQL and code injection, передбачає впровадження або вставку SQL-запитів у додаток за допомогою вхідних даних клієнта. Хороший SQL-експлойт може виконувати операції адміністрування базі даних, а також зчитувати та змінювати конфіденційні дані. Через

наявність старих функціональних інтерфейсів він широко використовується в програмах PHP і ASP.

Щоб уникнути втрати важливих даних, які зберігаються в базі даних, а також для запобігання виконання вставленого програмного коду, необхідно реалізувати захист від даної загрози. Недоліки аунтентифікації, також відомі як неправильна аунтентифікація, виникають в основному через неправильне виконання функцій додатків, пов'язаних з аутентифікацією та керуванням сеансом. Ці функції дозволяють зловмисникам компрометувати паролі, ключі або маркери сеансу, навіть використавши інші проблеми впровадження, щоб припустити особистість користувача тимчасово або назавжди. Цей тип атак може бути здійснений хакерами за допомогою різних методів, включаючи атаки на основі словника або навіть заповнення даних, що означає автоматичну ін'єкцію раніше порушених або загальнодоступних пар імені користувача та пароля, яка не завжди пов'язана з поточною метою шахрайського отримання доступу до облікових записів користувачів. Щоб уникнути можливості отримати доступ до чужого облікового запису, необхідно захистити його від цієї загрози. Використання незахищених протоколів (Захист чутливих даних) — багато програм не використовують засоби захисту переданих даних, такі як протокол HTTPS. Наприклад, програмне забезпечення шифрує номери кредитних карт у своїй базі даних за допомогою автоматичного шифрування бази даних. Тим не менш, ці дані розшифровуються автоматично при 84.00%, 53%, 45%, 37%, 29%, 13%, 13%, 5%. Неправильна настройка параметрів безпеки Виконання сценаріїв між сайтами Проблеми з аутентифікацією та проблеми з контролем доступу Ін'єкції програмного коду, а також запити SQL Використання компонентів з відомими вразливостями Розголошення конфіденційних даних Впровадження зовнішніх сутностей XML Рівень загрози становить [] високий [] середній [] низький [].20%.40%.100% 60 % 80 %100% 100.00% отримання, що дозволяє хакерам отримати ці дані у розшифрованому вигляді за допомогою SQL-ін'єкції. Необхідно захистити дані від даних загроз, щоб зловмисники не могли отримати важливі та конфіденційні дані. Впровадження зовнішніх сутностей XML — це тип ін'єкції, який полягає в додаванні до запиту XML атрибутів і сутностей до сервера, що дозволяє несанкціонованому доступу до даних. Наприклад, коли запит XML включає зовнішній файл, розташований на сервері. Захист від даної загрози необхідний, щоб зловмисники не могли отримати серверні файли. Злам контролю доступу — це проблема з методами авторизації, яка дозволяє порушнику отримати додаткові привілеї. Адміністративні інтерфейси, які дозволяють адміністраторам сайтів керувати своїми сайтами в Інтернеті, є одним із типів проблем контролю доступу. Такі функції часто використовуються адміністратором сайтів, щоб ефективно керувати користувачами, даними та вмістом. Необхідність реалазації

захисту від даної загрози полягає у тому, щоб уникнути отримання прав вищого рівня у ВЕБ-додаткові. Некоректне налаштування параметрів безпеки (Security Misconfiguration) - WEB-додаток - це складна система, що складається з багатьох компонентів, таких як WEB-сервер, СУБД та ін. Неправильна конфігурація одного компонента може викликати серйозні проблеми з безпекою всього програми. Щоб уникнути втрати важливих даних, необхідно запровадити захист від даної загрози. Міжсайтове виконання сценаріїв (XSS) означає введення шкідливого коду в HTTP-відповідь клієнта та виконання його на стороні клієнта. Атаки XSS активні та пасивні. Активна атак має більшу небезпеку, з точки зору зловмисника немає необхідності заманити жертву за спеціальним посиланням, йому достатньо вставити шкідливий код в базу даних або у якийсь файл, що знаходиться на сервері. Таким чином, всі, хто відвідує сайт автоматично стають жертвами . Необхідність реалізації захисту від даної загрози полягає у тому, щоб уникнути отримання захищених даних зловмисником. Відсутність валідації даних (Insecure Deserialization) – десеріалізація перетворює послідовність біт в структуровані дані, найчастіше на даному етапі не приділяється достатньо уваги безпеці, наприклад, відсутня валідація типів даних, що призводить до їх підміні.Уникнення отримання неправильних даних і збереження неправильних елементів у базі даних є необхідним заходом для захисту від даної загрози. Розголошення конфіденційних даних дозволяє зловмисникам отримати додаткові дані, такі як логін і пароль адміна, а також інформацію про баги. Необхідні заходи захисту від цієї загрози, щоб запобігти розголошенню конфіденційної та конфіденційної інформації, пов’язаної з розробкою додатку.

1.2. Роль технологій виявлення вразливостей у забезпеченні кібербезпеки.

Технології виявлення вразливостей є важливою частиною кібербезпеки, оскільки вони дозволяють раннє виявлення та усунення потенційно небезпечних слабких місць. З метою захисту конфіденційності, цілісності та доступності інформаційних ресурсів використання цих технологій є важливою частиною комплексної стратегії кіберзахисту.

Вразливості – це слабкі місця в інформаційних системах, які можуть бути використані зловмисниками для незаконного вторгнення, атак або отримання несанкціонованого доступу. Розуміння та виявлення цих вразливостей є першочерговим завданням для забезпечення ефективної кібербезпеки.

Технології виявлення вразливостей включають в себе різноманітні методи та інструменти, спрямовані на пошук, аналіз та усунення слабких місць у програмному та апаратному забезпеченні. Серед найпоширеніших технологій можна виокремити:

Сканування вразливостей: Автоматизовані сканери аналізують системи на предмет вразливостей, використовуючи бази даних відомих уразливостей.

Аналіз коду: Інструменти для статичного та динамічного аналізу програмного коду дозволяють виявити можливі вразливості на етапі розробки.

Виявлення аномалій: Системи виявлення вторгнень та аналізу безпеки дозволяють виявляти незвичайні або підозрілі активності, що можуть свідчити про експлуатацію вразливостей.

Burp Suite забезпечує високий рівень безпеки веб-додатків і є важливим інструментом для тестування вразливостей веб-додатків. Це програмний комплекс, який містить набір інструментів, призначених для пошуку та використання різних типів вразливостей. Burp Suite дозволяє проводити аудит безпеки веб-додатків, перехоплювати та аналізувати трафік між клієнтом і сервером, перевіряти наявність різних типів вразливостей, таких як SQL-ін’єкції, XSS-атаки та CSRF-атаки, а також проводити тестування на проникнення. Щоб максимально ефективно визначити вразливості, можна проводити тестування як вручну за допомогою Burp Suite, так і автоматично. Крім того, інструмент має можливість налаштування спеціальних скриптів для автоматизації процесу тестування та виявлення вразливостей. Завдяки своїм можливостям Burp Suite є незамінним інструментом для команд розробників та тестувальників, які займаються створенням та тестуванням веб-додатків.

Ця установка дозволяє забезпечити високий рівень захисту веб-додатків і зменшити ризик їх вразливості. Burp Suite – це важливий інструмент для тих, хто працює над розробкою та тестуванням веб-додатків. Він гарантує високий рівень безпеки веб-додатків і дозволяє проводити ефективне тестування вразливостей. Сьогодні ми розглянемо основні інструменти, за допомогою яких Burp покращує своє життя. Бурп має багато вкладок, але ми розглянемо деякі з найпопулярніших: Proxy, Intruder, Repeater Decoder, Logger, Comparer. Вивчіть OWASP Top 10, OWASP Juice Shop. За допомогою веб-додатка Juice Shop, який розвиває навички тестування вразливостей, ми дізнаємося про найпоширеніші категорії вразливостей і спробуємо використовувати одну з них на практиці.

1.3. Загальний огляд методів виявлення вразливостей.

Виявлення вразливостей - це процес ідентифікації слабких місць в програмному чи апаратному забезпеченні, які можуть бути використані зловмисниками для несанкціонованого доступу, руйнування, витоку інформації чи інших атак. Існує кілька методів виявлення вразливостей, і їх комбінація забезпечує ефективніше виявлення та усунення проблем безпеки. Ось загальний огляд деяких основних методів:

Сканування вразливостей:

Автоматизовані сканери: Використовуються для виявлення вразливостей шляхом автоматизованого аналізу програмного чи апаратного забезпечення.

Ручне тестування: Включає у себе експертне тестування безпеки, коли фахівець активно аналізує систему на предмет слабких місць.

Аналіз вихідного коду:

Статичний аналіз: Оцінює вихідний код програми без його виконання.

Динамічний аналіз: Аналізує виконання програми в реальному часі для виявлення вразливостей, що можуть виникнути під час виконання.

Моніторинг мережі:

Системи виявлення вторгнень (IDS): Моніторять мережу на виявлення непередбачених або підозрілих активностей.

Системи перехоплення пакетів: Використовуються для аналізу мережевого трафіку та виявлення аномалій. Аналіз безпеки конфігурацій:

Автоматизовані інструменти аудиту конфігурацій: Перевіряють налаштування систем та програм на відповідність безпековим стандартам.

Тестування на злам:

Етичне вторгнення: Спроби взлому системи або мережі за дозволом для виявлення слабкі місця. Аналіз журналів та логів:

Моніторинг та аналіз журналів подій: Слідкування за активністю в системі та виявлення аномалій. Системи виявлення аномалій:

Машинне навчання: Використовує алгоритми машинного навчання для виявлення незвичайних паттернів в активності системи.

Тестування на витік інформації:

Тестування на витік інформації: Аналіз на предмет можливості непередбаченого витоку конфіденційної інформації.

Зазвичай ефективний підхід включає в себе комбінацію цих методів для повного покриття можливих вразливостей в системі чи програмному забезпеченні.

Найкращий спосіб боротьби з вразливістю — починати з мережі. Керування вразливістю мережі стосується всього вашого навколишнього середовища,

включаючи кожен підключений пристрій, операційну систему, апаратне забезпечення, програмне забезпечення, брандмауери та інші компоненти. Зловмисники можуть проникнути в вашу систему через незахищений маршрутизатор Wi-Fi, пристрій IoT із надмірним контролем доступу або неправильний брандмауер.

Щоб переконатися, що перші лінії захисту зміцнені, зменшуючи ймовірність витоку даних, є важливим сканування вразливостей мережі та регулярне встановлення виправлень. Таким чином, у сучасному світі інформаційно-комунікаційні технології (ІКТ) мають значний вплив на соціальний добробут, економічний розвиток і національну безпеку. Комп'ютери, мобільні пристрої та мережі зазвичай входять до ІКТ. ІКТ також охоплюють групу зловмисників зі зловмисними намірами, які також називають мережевими зловмисниками, кіберзлочинцями тощо. Протистояти цій шкідливій кібердіяльності є одним із пріоритетів на міжнародному рівні, а також важливим предметом досліджень. Таким чином, актуальним є питання впровадження та вдосконалення найбільш ефективних методів виявлення вразливостей в ІТК.

Більшість СВА використовують ознаки атаки, як і інші програми сканування вірусів на основі сигнатур, для пошуку атак. СВА намагається заблокувати відому уразливість у базі даних, коли зловмисник намагається її використовувати. одним із прикладів є безкоштовний продукт на основі сигнатур Snort, який працює на операційних системах Windows і Unix. Оскільки Snort є програмою з відкритим вихідним кодом, він має здатність створювати базу даних сигнатур швидше, ніж будь-яка інша система. Підпис Snort використовується всіма продуктами інформаційної безпеки, від комерційних брандмауерів до проміжного програмного забезпечення, такого як Hogwash. Сигнатурні методи базуються на специфічних структурах, які використовуються для дослідження атак. Системний адміністратор створить правила примусового застосування, щоб протистояти таким атакам.

Метод виявлення аномалій – виявлення аномалій, які стосуються основи нормальної роботи системи або поведінки, а потім повідомлення адміністратору. В нормальному режимі роботи системи кількість трафіку в мережі незначно змінюється. Тим не менш, деякі мережі мають незвичайні структури, наприклад військові або 22 розвідувальні мережі, і дії, які відбуваються на сервері, можуть бути неконтрольовані. Слід зазначити, що системний адміністратор хоче відрізнити події СВА від ненормальних подій (на відміну від відомого опису руху) і ненормальних протоколів подій (які відрізняються від протоколів мережі). Хороші моделі виявлення поведінкової активності включають:

• статистичну модель;

• модель, засновану на теорії інформації;

• модель кластера;

• модель класифікації.

1.4. Ключові аспекти технологій виявлення вразливостей.

У відповідь на зростаючий рівень складності та різноманітності загроз у сфері кібербезпеки технології виявлення вразливостей постійно розвиваються. Ось деякі основні елементи, які визначають технології виявлення вразливостей:

Автоматизація:

Сканування вразливостей: За допомогою автоматизованих сканерів виявлення вразливостей можна автоматизовано перевіряти апаратне та програмне забезпечення на наявність слабких місць.

Інтеграція з розробкою:

Інструменти статичного аналізу коду: Інтегровані в процес розробки, щоб допомогти виявити вразливості вихідного коду під час процесу розробки.

Машинне навчання та штучний інтелект:

Аналіз поведінки системи: Машинне навчання використовується для виявлення змін у поведінці системи та аномалій.

Розподілена обробка та хмарні технології:

Сканування у реальному часі: За допомогою хмарних технологій можна проводити сканування та аналіз вразливостей у режимі реального часу, що дозволяє оперативно реагувати на нові небезпеки.

Інтелектуальне сканування:

Аналіз контексту: Інтеллектуальні методи виявлення вразливостей включають аналіз контексту, який враховує конкретні умови та середовища роботи системи.

Тестування на витік інформації:

Моніторинг витоку даних: Використання технологій, які виявляють і відстежують непередбачені витоки конфіденційної інформації.

Інтеграція з системами виявлення вторгнень:

Спільна робота з IDS/IPS: Використання систем виявлення вторгнень і інструментів виявлення вразливостей для швидкого реагування на потенційні атаки.

Аналіз вразливостей IoT-пристроїв:

Спеціалізовані інструменти для IoT: Розробка та використання інструментів, спрямованих на виявлення вразливостей в Інтернеті речей (IoT).

Континуальний моніторинг та тестування:

Системи континуального моніторингу безпеки: Забезпечення постійного виявлення та виправлення вразливостей з урахуванням змін в середовищі.

Ці аспекти допомагають забезпечити більш швидке та ефективне виявлення вразливостей у сучасних інформаційних системах.

Рисунок 1.2.

Алгоритм аналізу мережевого трафіку та виявлення вразливостей/атак.

СВА складають складний процес моніторингу мережевого трафіку, який включає збір, аналіз і вивчення великої кількості даних, які надходять з мережі. У СВА основні етапи моніторингу трафіку мережі:

1. Збір даних: на першому етапі система збирає дані про трафік, що надходить на мережу. Для збору даних можуть використовуватися різні засоби, наприклад, сенсори, вузли збору даних, проксі-сервери та інші.

2. Фільтрація трафіку: Після збору даних вони проходять через фільтр, який видаляє непотрібний трафік. Цей тип трафіку може включати трафік, створений системними процесами. Параметри, такі як порти, IP-адреси та протоколи, можуть бути використані для фільтрації.

3. Аналіз трафіку: після фільтрації дані аналізуються для виявлення вразливостей та атак на мережу. Аналіз може включати в себе різні методи, такі як

статистичний аналіз, аналіз змісту пакетів, виявлення змін у звичайному поведінці трафіку та інші.

4. Виявлення вразливостей та атак: Після завершення аналізу система визначає можливі вразливості та атаки на мережу. Виявлені вразливості та атаки можуть бути класифіковані відповідно до типу та ступеня небезпеки, який вони представляють. Наприклад, система може виявити SQL-ін’єкцію або DDoS-атаку.

5. Попередження про вразливості та атаки: Після виявлення вразливостей і атак на мережу система повинна відстежувати ці події та сповіщати про них. Попередження можна надіслати різними способами, наприклад електронними листами, SMS-повідомленнями або за допомогою спеціального додатку.

6. Реагування на вразливості та атаки: коли мережеві вразливості та атаки виявлено, система повинна мати можливість реагувати на ці події. Наприклад, система може надіслати повідомлення відповідальній особі, яка вживе необхідні заходи щодо усунення вразливостей, або автоматично заблокувати IP-адресу, з якої здійснюється атака.

7. Аудит і звітність: після реагування на вразливість і атаки система повинна проводити аудит і звітність про ці події. Це дозволяє зберігати та використовувати інформацію про події для аналізу та покращення системи виявлення атак. Загалом процес моніторингу мережевого трафіку в системі виявлення атак включає збір, фільтрацію, аналіз, виявлення, попередження, реагування та аудит даних. Кожен із цих етапів має вирішальне значення для забезпечення безпеки мережі та захисту від будь-яких потенційних загроз.

2 ІНСТРУМЕНТИ ТА ТЕХНІКИ BURP SUITE ДЛЯ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ

2.1. Визначення та функціональні можливості Burp Suite.

Burp Suite — це потужний інструмент для тестування вразливостей веб-додатків, який допомагає забезпечити високий рівень безпеки ваших веб-додатків. Це програмний комплекс, який включає в себе набір інструментів, призначених для виявлення та експлуатації різних видів вразливостей. Burp Suite дозволяє проводити аудит безпеки веб-додатків, перехоплювати та аналізувати трафік між клієнтом і сервером, перевіряти наявність різних типів вразливостей, таких як SQL-ін’єкції, XSS-атаки та CSRF-атаки, а також проводити тестування на проникнення.

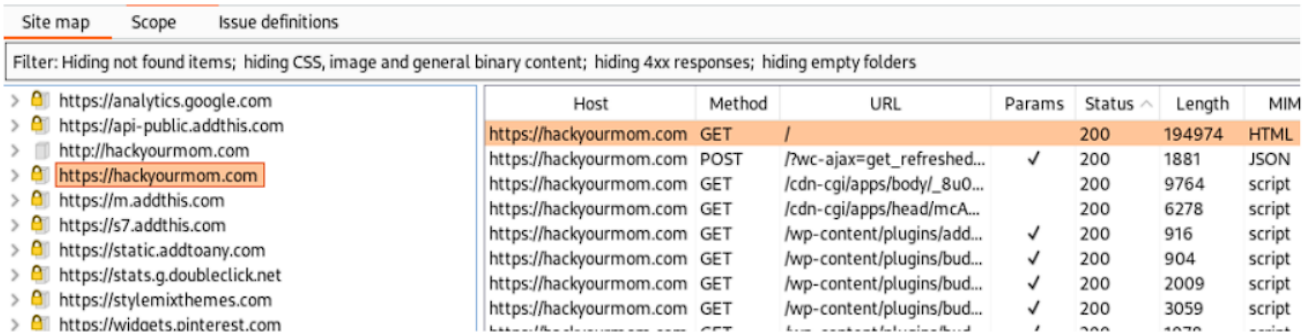
За допомогою Burp Suite можна проводити тестування як вручну, так і автоматично, що дозволяє максимально ефективно виявити вразливості. Крім того, у програмі є можливість налаштувати певні скрипти, щоб автоматизувати процес тестування та виявлення вразливостей. Burp Suite є чудовим інструментом для розробників і тестувальників, які працюють над розробкою та тестуванням веб-додатків.

Ця установка дозволяє забезпечити високий рівень захисту веб-додатків і зменшити ризик їх вразливості. Burp Suite – це важливий інструмент для тих, хто працює над розробкою та тестуванням веб-додатків. Він гарантує високий рівень безпеки веб-додатків і дозволяє проводити ефективне тестування вразливостей.

Сьогодні ми розглянемо основні інструменти, за допомогою яких Burp покращує своє життя. Бурп має багато вкладок, але ми розглянемо деякі з найпопулярніших: Proxy, Intruder, Repeater Decoder, Logger, Comparer. Вивчіть OWASP Top 10, OWASP Juice Shop. За допомогою веб-додатка Juice Shop, який розвиває навички тестування вразливостей, ми дізнаємося про найпоширеніші категорії вразливостей і спробуємо використовувати одну з них на практиці.

Рисунок 2.1.

Site map



У вікні зліва ми бачимо URL ресурсів. У вікні справа є наступні колонки, ось найбільш важливі з них:

– Host. Домен.

– Method. HTTP метод для взаємодії з ресурсом.

– URL. Посилання яке було знайдено Burp’ом.

– Params. Поле відмічено, якщо були додані параметри до запиту.

Рисунок 2.2.

Include in scope

Тут можна додавати цілі, які ви хочете відслідковувати або тестувати на проникнення. Для кожної дії є відповідна кнопка:

– Add. Додати;

– Edit. Редагувати;

– Remove. Видалити;

– Paste URL. Вставити посилання;

– Load …. Завантажити.

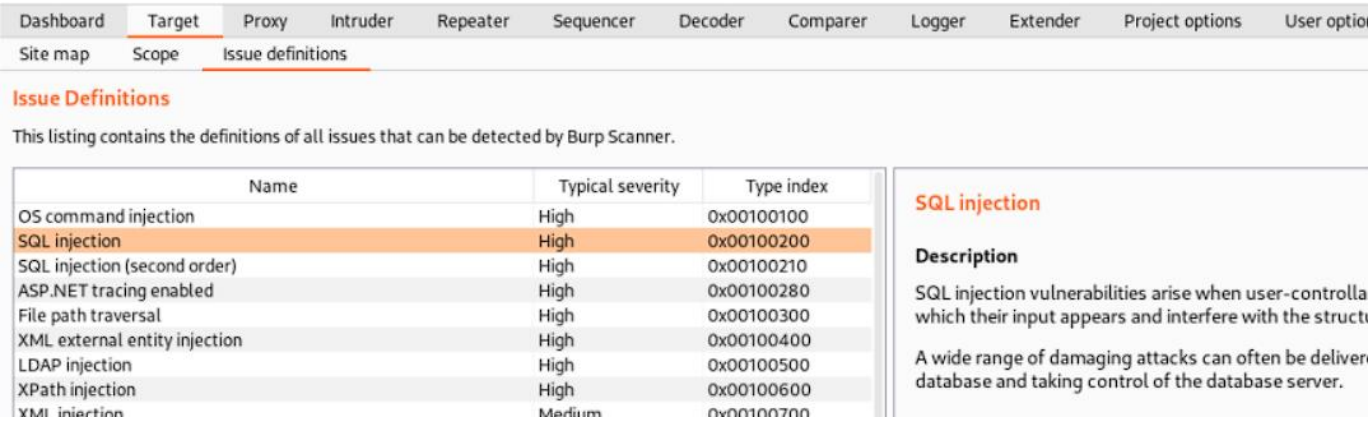
Рисунок 2.3.

Exclude from scope

Тут можна виключати зі скоупу цілі, які наразі вас не цікавлять. Взаємодіяти з цим блоком можна за допомогою відповідних кнопок, опис яких є у блоці Include in scope вище.

Рисунок 2.4.

Issue definitions



Тут ви можете бачити пояснення та опис типів вразливостей. Дана таблиця скаладється з трьох колонок:

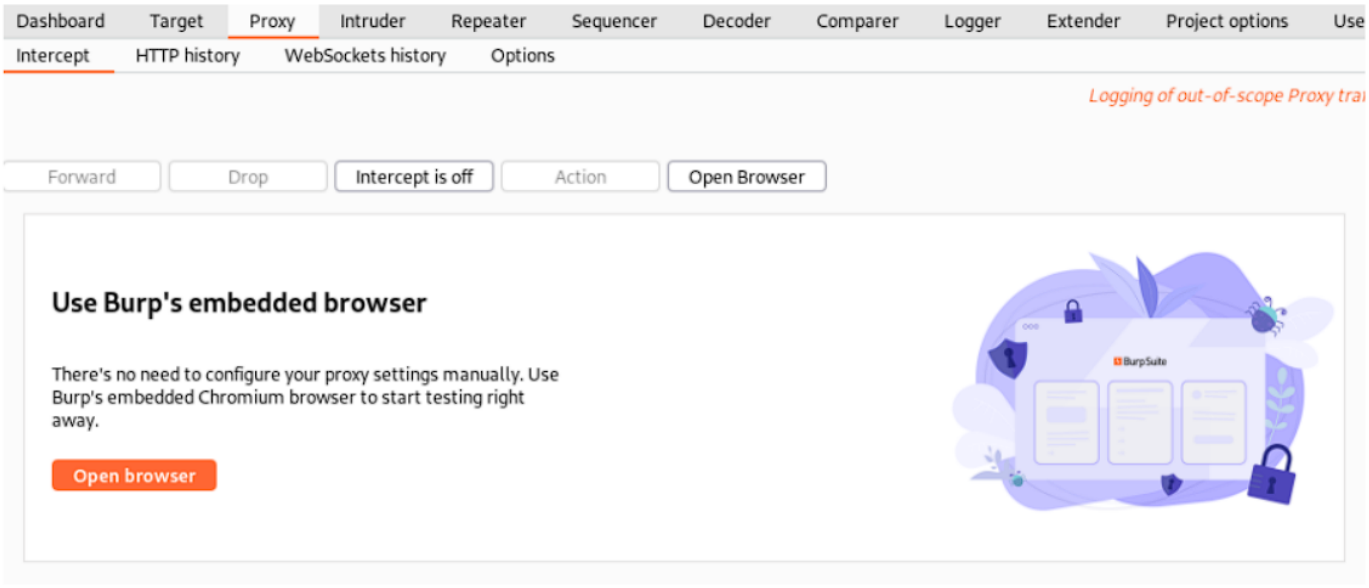
– Name- Назва вразливості.

– Typical severity- Рівень “небезпеки” даної вразливості.

– Type index- Індекс у даній таблиц.

Рисунок 2.5.

Intercept



Найбільш важливі кнопки щоб тицять:

- Forward. У випадку, якщо перехоплення пакетів увімкнено(Intercept is ON), цей пакет можна відредагувати та надіслати далі.

– Drop. Пропустити перехоплений пакет.

– Intercept is on/off. Увімкнути або вимкнути перехоплення пакетів

– Open Browser. Відкрити браузер вже налаштований для роботи з Burp’ом.

У Burp Suite Intrruder використовується для фаззингу. Це дозволяє нам приймати запит, який зазвичай зберігається в проксі-сервері перед надсиланням його Intruder, і використовувати його як шаблон для автоматичного надсилання великої кількості запитів із зміненими значеннями. Наприклад, ми могли б зафіксувати запит, який містить спробу входу, а потім налаштувати Intruder, щоб він замінив поля імені користувача та пароля значеннями зі списку слів. Це дозволило б нам грубо використовувати форму входу.

Фаззинг — це метод автоматизованого тестування програмного забезпечення, який намагається знайти помилки кодування та лазівки в безпеці програмного забезпечення шляхом випадкової подачі недійсних і неочікуваних введених даних і даних у програму.

Attack Type – тип атаки, існує 4 типи:

Sniper. У цьому випадку використовується один набір корисних навантажень. Він одночасно націлює кожну позицію корисного вантажу та розміщує кожне корисне навантаження в цю саму позицію. Цей тип атаки корисний для визначення окремих параметрів запиту для уразливостей, які є поширеними;

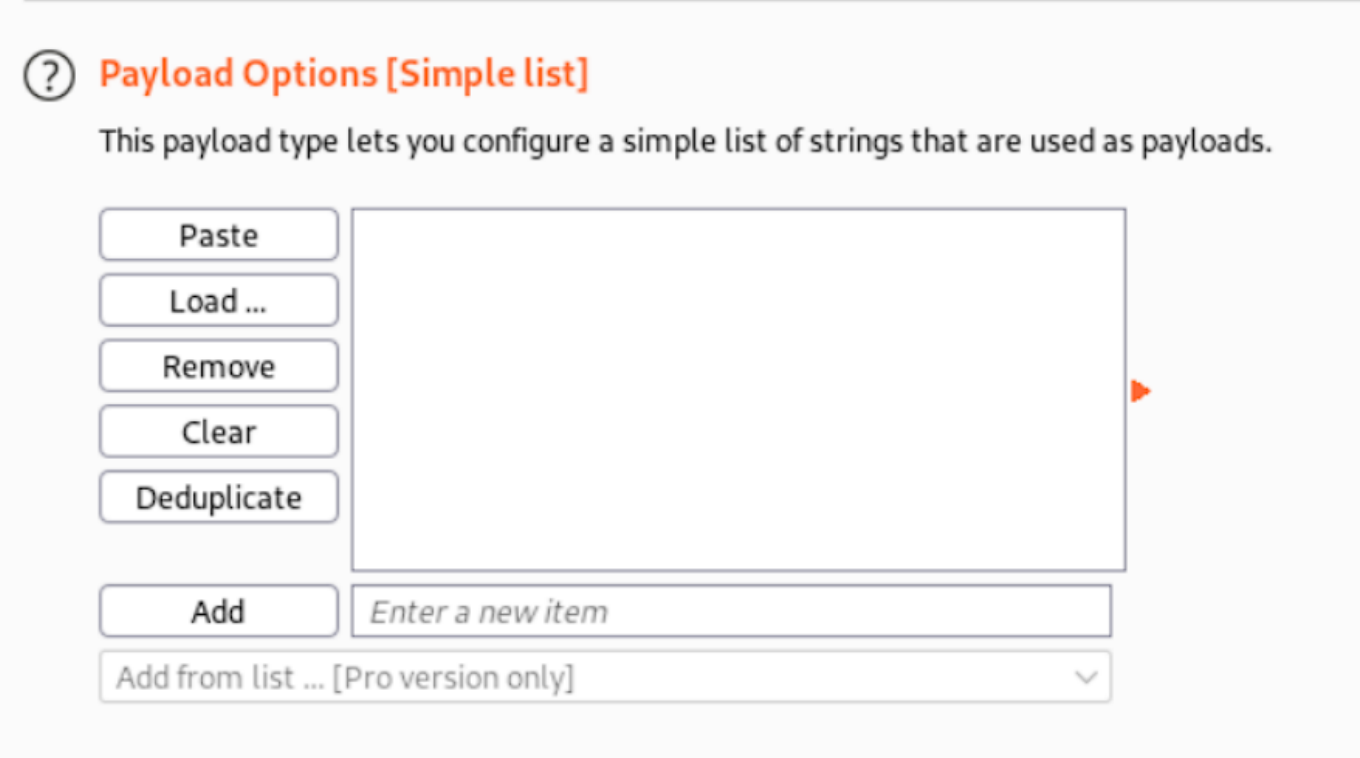
Battering Ram. Щоб досягти цього, використовується одна група корисного навантаження. Він перебирає корисні навантаження та одночасно розміщує їх у всіх визначених позиціях. Коли один і той самий вхід потрібно вставити в кілька місць у запиті, цей тип атаки корисний;

Pitchfork. Використовує різні пейлоади. Кожна позиція має окремий набір пейлоадів, максимум двадцять. Атака одночасно перебирає всі набори пейлоадів і розміщує один пейлоад у кожній певній позиції. Іншими словами, перший запит поміщає перший пейлоад з набору 1 в позицію 1, а другий пейлоад з набору 2 в позицію 2; другий запит поміщає другий пейлоад з набору 1 в позицію 1, а другий пейлоад з набору 2 в позицію 2 і так далі. Коли атака вимагає введення різних, але пов’язаних вхідних даних у різних місцях запиту (наприклад, ім’я користувача в одному параметрі та відомий ідентифікаційний номер, відповідний цьому імені користувача в іншому параметрі), цей тип атаки корисний;

Cluster bomb. Це досягається за допомогою кількох різних пейлоадів. Кожна позиція має окремий набір пейлоадів, максимум двадцять. Атака повторює кожен набір пейлоадів по черзі, щоб перевірити всі перестановки комбінацій пейлоадів. Якщо є дві позиції пейлоадів, атака помістить перший пейлоад з набору пейлоадів 2 в позицію 2 і перебере всі пейлоади в наборі пейлоадів 1 в позиції 1; потім він помістить другий пейлоад з набору пейлоадів 2 в позицію 2 і перебере всі корисні навантаження в набір корисних даних 1 в позиції 1. Коли атака вимагає включення різних непов’язаних або невідомих даних у різні частини запиту (наприклад, під час вгадування облікових даних, імені користувача в одному параметрі та пароля в іншому параметрі), цей тип атаки корисний.

Рисунок 2.6.

Payload Options (Simple list)



Завантаження набору пейлоадів. Тут можна завантажити з файлу або написати власноруч. Застосовується лише до пейлоадів, що відносяться до типу Simple list, але також варто розглянути усі доступні пейлоади, з якимим працює Burp:

-Simple list. Це найпростіший тип пейлоадів, який дозволяє налаштувати простий список рядків, які використовуються як корисні навантаження.

-Runtime file. У цьому типі пейлоадів можна змінити, який файл читатиме рядки пейлоади під час виконання. Щоб уникнути збереження всього списку пейлоадів у пам’яті, це корисно, коли потрібен великий список. Кожен рядок файлу містить один пейлоад, тому пейлоади можуть не містити символів нового рядка;

-Custom iterator. Цей тип пейлоадів дозволяє налаштувати кілька списків елементів і генерувати пейдлажи, використовуючи всі перестановки елементів у списках. Він забезпечує потужний спосіб генерувати власні перестановки символів або інших елементів відповідно до заданого шаблону.

-Character substitution. Цей тип пейлоадів дозволяє налаштувати список рядків і застосувати різні заміни символів до кожного елемента. Це може бути корисно під час атак на вгадування пароля, для створення загальних варіантів словникових слів.

-Numbers. Цей тип пейлоадів генерує числові корисні навантаження в заданому діапазоні та в заданому форматі.

-Dates. Цей тип пейлоадів генерує пейлоади, що містять дату в межах заданого діапазону та у визначеному форматі.

-Username generator. Цей тип пейлоадів дозволяє налаштувати список імен або адрес електронної пошти, а також отримати потенційні імена користувачів з них за допомогою різних поширених схем.

Рисунок 2.7.

Payload processing rule

Add prefix. Додати префікс перед пейлоадом;

Add suffix. Додати суфікс після пейлоаду;

Match/Replace. Заміна будь-яких частин корисного навантаження, які відповідають певному регулярному виразу, на певну строку;

Substring. Витяг частини пейлоадів, починаючи з заданого зміщення (індексовано 0) і до заданої довжини;

Reverse substring. Як правило, він працює за допомогою підрядки, де кінцеве зміщення визначається шляхом відліку від кінця корисного навантаження, а довжина відраховується назад від кінцевого зміщення.

Modify case. Змінює регістр пейлоада, якщо можливо. Доступні наступні параметри:
– No change. Без змін;

– To lower case. Приведення до нижнього регістру;

– To upper case. Приведення до верхнього регістру;

–To Propername. Приведення першої літери до верхнього регістру, інші літери приводяться до нижнього регістру;

– To ProperName. Приведення першої літери до верхнього регістру, інші літери не змінюються.

Encoding. Кодування пейлоадів за допомогою різних схем: URL, HTML, Base64, ASCII шістнадцятковий або інші рядки для різних платформ;

Decoding. Процесс зворотній до Encoding;

Hash. Виконує операцію хешування над пейлоадом;

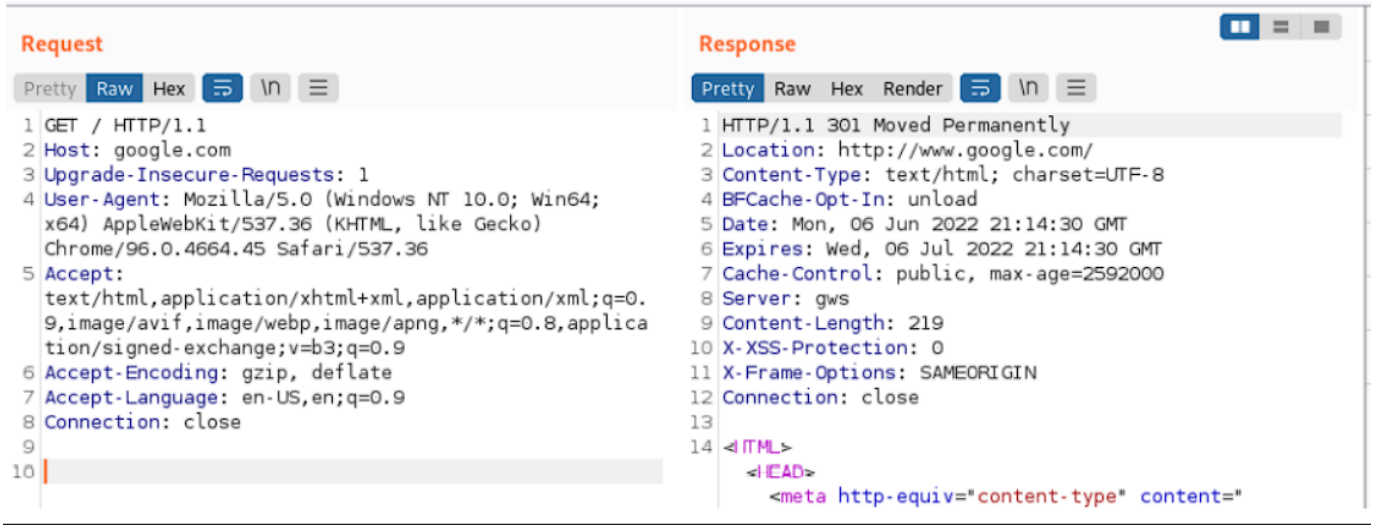
Add raw payload. Це змінює список пейдлоадів і додає необроблене значення пейдлоаду до або після обробленого значення. Іншими словами, це додає додаткове значення, яке буде перевірено пізніше. Наприклад, якщо вам потрібно подати той самий пейлоад як у сирому, так і в хешованому вигляді, це може бути корисним;

Skip if matches regex. Це визначає, чи відповідає поточне значення зазначеному регулярному виразу. Якщо це так, він пропускає пейлоад і переходить до наступного. Це може бути корисним, наприклад, якщо ви знаєте, що значення параметра повинно мати мінімальну довжину і хочете пропустити будь-які значення, коротші за цю довжину.

Invoke Burp Extension. Проблема з розширенням Burp для обробки корисних даних У розширенні має бути Intruder, зареєстрований процесор пейлоадів. Ви можете вибрати процесор зі списку поточних процесорів, зареєстрованих пейлоадами.

Рисунок 2.8.

Repeater



Щоб використовувати Burp Repeater з повідомленнями HTTP, ви можете вибрати повідомлення HTTP в будь-якому місці Burp, а потім вибрати пункт у контекстному меню, щоб перейти до Repeater. Це створить нову вкладку запиту в Repeater, а деталі цілі та редактор повідомлень запиту автоматично будуть заповнені відповідними деталями. Ви також можете вручну відкрити нову вкладку Repeater, вибравши параметр «HTTP». Кожна вкладка Repeater містить ці компоненти для повідомлень HTTP:

Елементи керування для надсилання запитів та навігації в історії запитів.

Показано цільовий сервер, на який буде надіслано запит – ви можете натиснути на деталі цілі, щоб змінити їх.

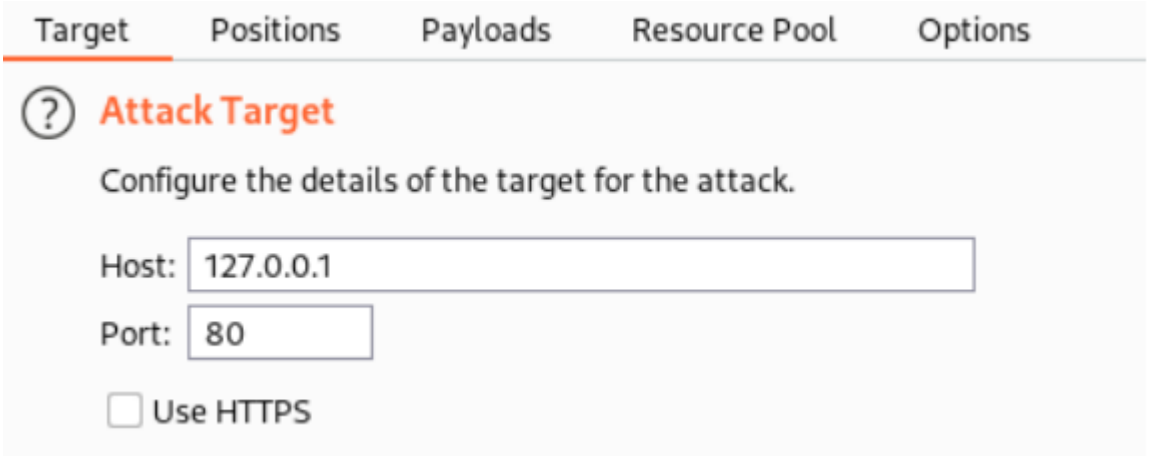
Редактор повідомлень HTTP, що містить запит, який має бути виданий. Ви можете редагувати запит і повторно надсилати його знову і знову.

Редактор повідомлень HTTP, що показує відповідь, отриману від останнього надісланого запиту.

Intruder — це додаток, який використовується для фаззингу в Burp Suite. Це дозволяє нам приймати запит, який зазвичай зберігається в проксі-сервері перед надсиланням його Intruder, і використовувати його як шаблон для автоматичного надсилання великої кількості запитів із зміненими значеннями. Наприклад, ми могли б зафіксувати запит, який містить спробу входу, а потім налаштувати Intruder, щоб він замінив поля імені користувача та пароля значеннями зі списку слів. Це дозволило б нам грубо використовувати форму входу.

Фаззинг – технологія автоматизованого тестування програмного забезпечення, яка використовує випадкову подачу недійсних і неочікуваних введених даних і даних у програму, щоб знайти помилки кодування та лазівки в безпеці. У цьому місці користувач може обрати ціль для проведення атаки, а конфігурація буде застосована, як описано в наступних табах.

Рисунок 2.9.

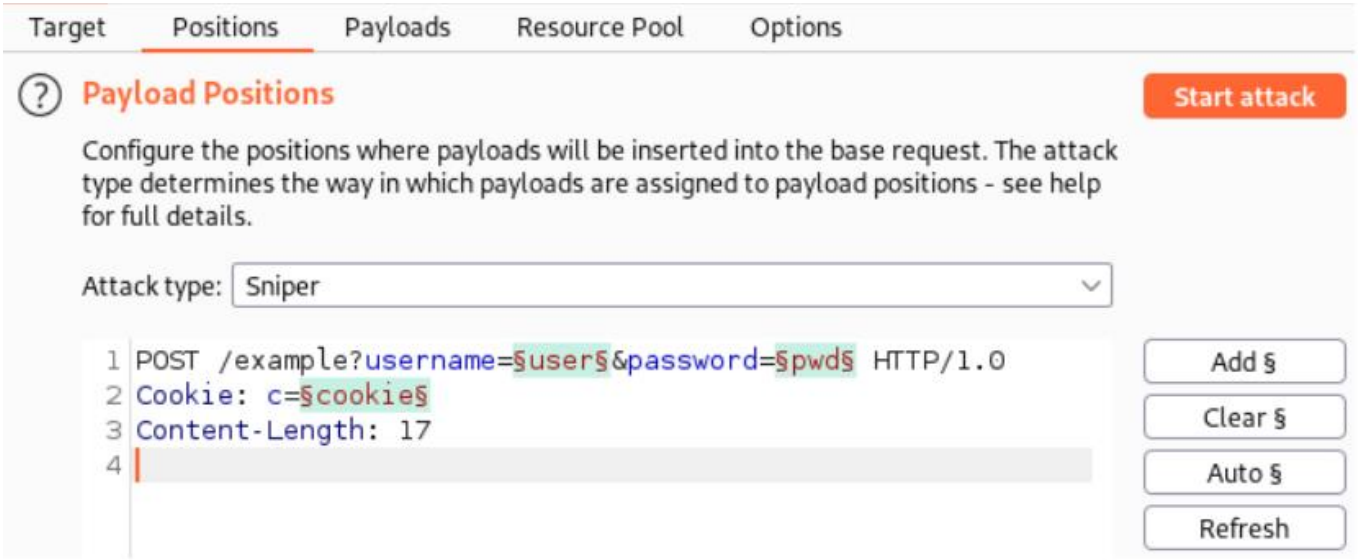


Positions

Ця вкладка використовується для налаштування шаблону запиту для атаки, також маркерів пейлоадів та типу атаки (який визначає спосіб, яким корисні навантаження призначаються позиціям корисного навантаження).

На даному прикладі можна бачити, що у ході атаки будуть змінюватись на певні значення параметри user, pwd та cookie. На що саме буде змінюватись – буде визначено у наступному табі – Payloads.

Рисунок 2.10.



Attack Type – тип атаки, існує 4 типи:

Sniper. У цьому випадку використовується один набір корисних навантажень. Він одночасно націлює кожну позицію корисного вантажу та розміщує кожне корисне навантаження в цю саму позицію. Цей тип атаки корисний для визначення окремих параметрів запиту для уразливостей, які є поширеними;

Battering Ram. Щоб досягти цього, використовується одна група корисного навантаження. Він перебирає корисні навантаження та одночасно розміщує їх у всіх визначених позиціях. Коли один і той самий вхід потрібно вставити в кілька місць у запиті, цей тип атаки корисний;

Pitchfork. використовує різні пейлоади. Кожна позиція має окремий набір пейлоадів, максимум двадцять. Атака одночасно перебирає всі набори пейлоадів і розміщує один пейлоад у кожній певній позиції. Іншими словами, перший запит поміщає перший пейлоад з набору 1 пейлоаду в позицію 1, а другий пейлоад з набору 2 пейлоаду в позицію 2; другий запит поміщає другий пейлоад з набору 1 пейлоаду в позицію 1, а другий пейлоад з набору 2 пейлоаду в позицію 2 і так далі. Коли атака вимагає введення різних, але пов’язаних вхідних даних у різних місцях запиту (наприклад, ім’я користувача в одному параметрі та відомий ідентифікаційний номер, відповідний цьому імені користувача в іншому параметрі), цей тип атаки корисний;

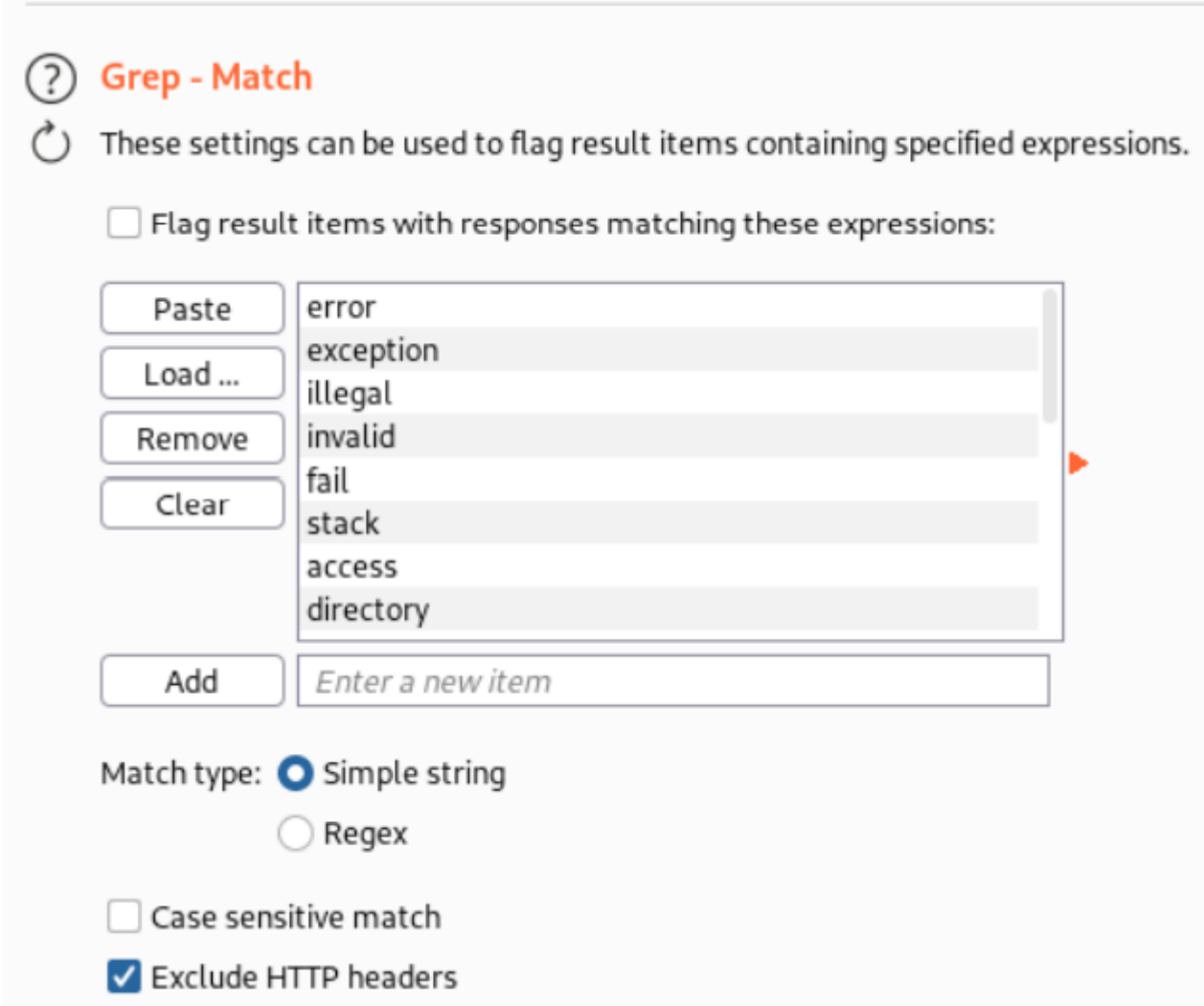
Cluster bomb. Для цього використовується кілька наборів пейлоадів. Для кожної визначеної позиції існує окремий набір пейлоадів (максимум 20). Атака по черзі повторює кожен набір пейлоадів, щоб перевірити всі перестановки комбінацій пейлоадів. Тобто, якщо є дві позиції пейлоадів, атака помістить перше пейлоад з

набору пейлоадів 2 в позицію 2 і перебере всі пейлоади в наборі пейлоадів 1 в позиції 1; потім він помістить друге пейлоад з набору пейлоадів 2 в позицію 2 і перебере всі корисні навантаження в набір корисних даних 1 в позиції 1. Цей тип атаки корисний, коли атака вимагає вставлення різних непов'язаних або невідомих даних у кількох місцях у запиті (наприклад, під час вгадування облікових даних, імені користувача в одному параметрі та пароля в іншому параметрі).

Grep – Match

Елементи результатів, що містять вказані вирази у відповіді, можна описати за допомогою цих параметрів. Burp додасть новий стовпець результатів для кожного елемента, налаштованого в списку. Цей стовпець містить число, яке показує, скільки разів вираз було знайдено в кожній відповіді. Ви можете відсортувати результати за кількістю знайдених виразів, натиснувши на заголовок стовпця.

Рисунок 2.11.



На додаток до списку виразів для відповідності доступні такі параметри:

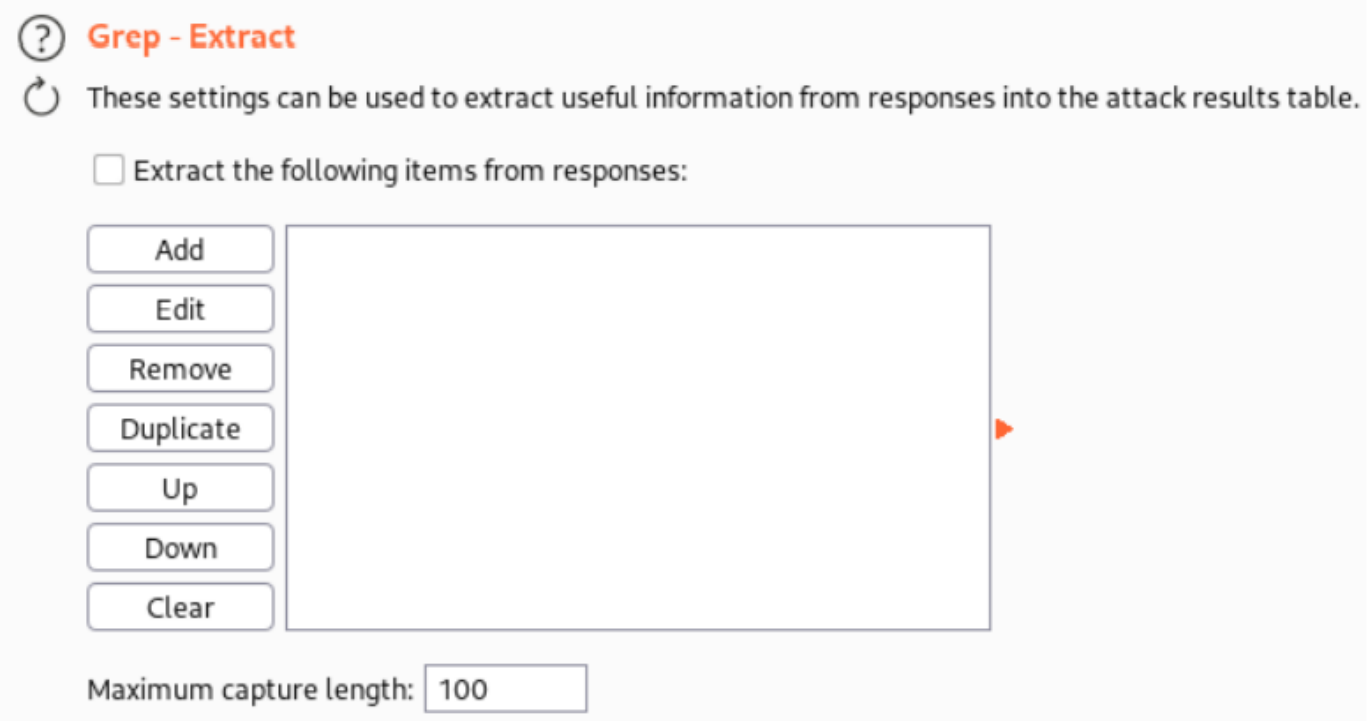
– Match Type. Визначає, чи є вирази простими рядками чи регулярними виразами.

– Case sensitive match. Це визначає, чи має перевірка виразу бути чутливою до регістру. – Exclude HTTP headers. Вказує, чи слід виключати заголовки відповіді HTTP з перевірки.

Grep – Extract

У таблиці результатів атаки можна знайти корисну інформацію з цих налаштувань. Burp додасть новий стовпець результатів для кожного елемента, налаштованого в списку. Цей стовпець містить текст, витягнутий для цього елемента. Клацніть заголовок стовпця, щоб відсортувати витягнуті дані, щоб упорядкувати їх. Цей параметр підтримує широкий спектр атак і корисний для видобутку даних із програми. Наприклад, ви можете знайти заголовок сторінки кожного документа, шукаючи цікаві елементи, перебираючи різні ідентифікатори документів.

Рисунок 2.12.



Використовується для автоматичної заміни даних у реквестах та відповідях клієнта та сервера. Взаємодія з даним блоком здійснюється за допомогою відповідних кнопок. У таблиці є наступні стовпці:

– Enabled. Відображає, чи застосовується даний патерн зараз;

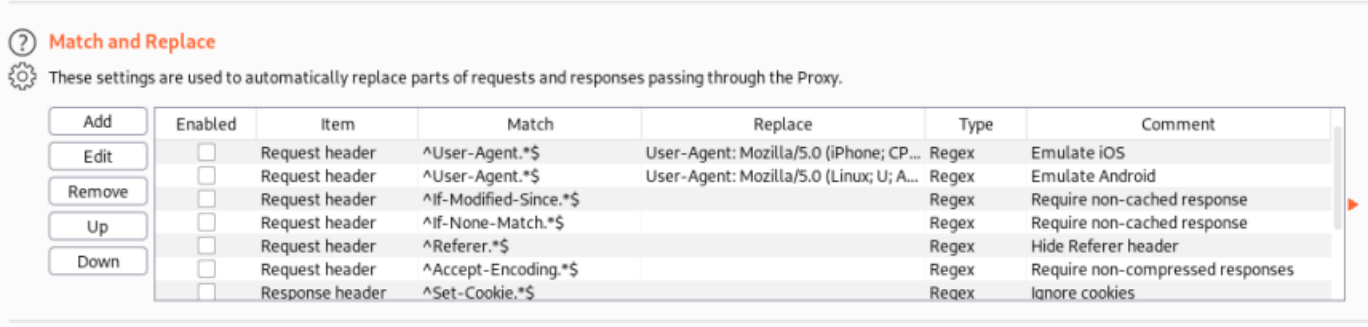
– Item. Тип елементу, що буде замінятися;

– Match. Регулярний вираз того, що хочемо замінити.

– Replace. Дані на які хочемо замінити результат Match.

– Type. Тип даних, що знаходиться у стовпці Match – Comment.

Рисунок 2.12.



TLS Pass Through

Ці налаштування використовуються для визначення цільових веб-серверів, для яких Burp буде безпосередньо проходити через TLS-з'єднання. У перегляді або історії перехоплення Proxy не буде доступних жодних даних про запити чи відповіді, зроблені через ці з'єднання. Передача через TLS-з'єднання може бути корисною в тих випадках, коли непросто усунути помилки TLS на клієнті – наприклад, у мобільних програмах, які виконують закріплення сертифіката TLS. Якщо програма отримує доступ до кількох доменів або використовує комбінацію з'єднань HTTP і HTTPS, то передача з'єднань TLS до певних проблемних хостів все одно дозволяє вам працювати з іншим трафіком за допомогою Burp у звичайний спосіб.

Рисунок 2.13.

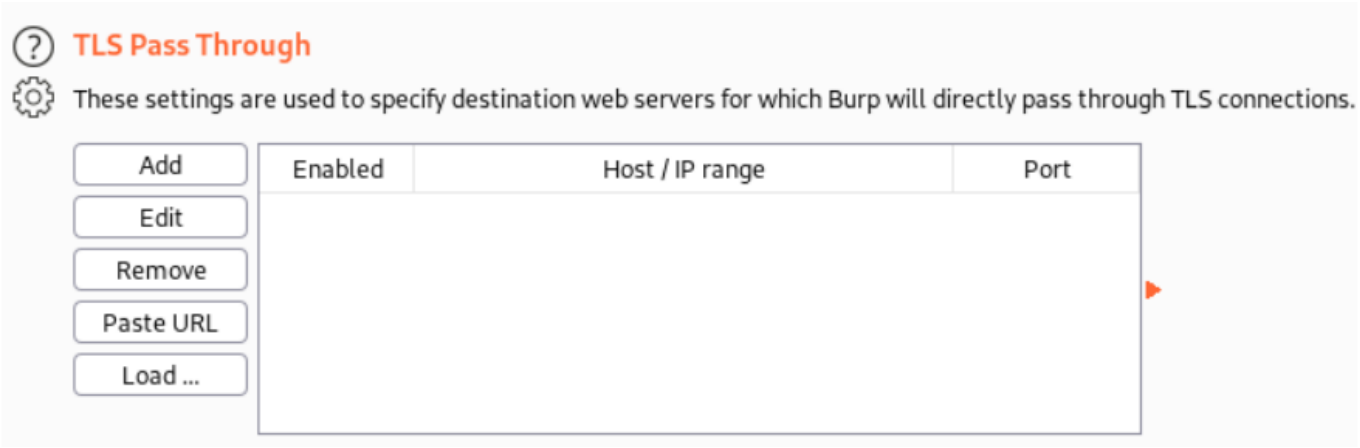
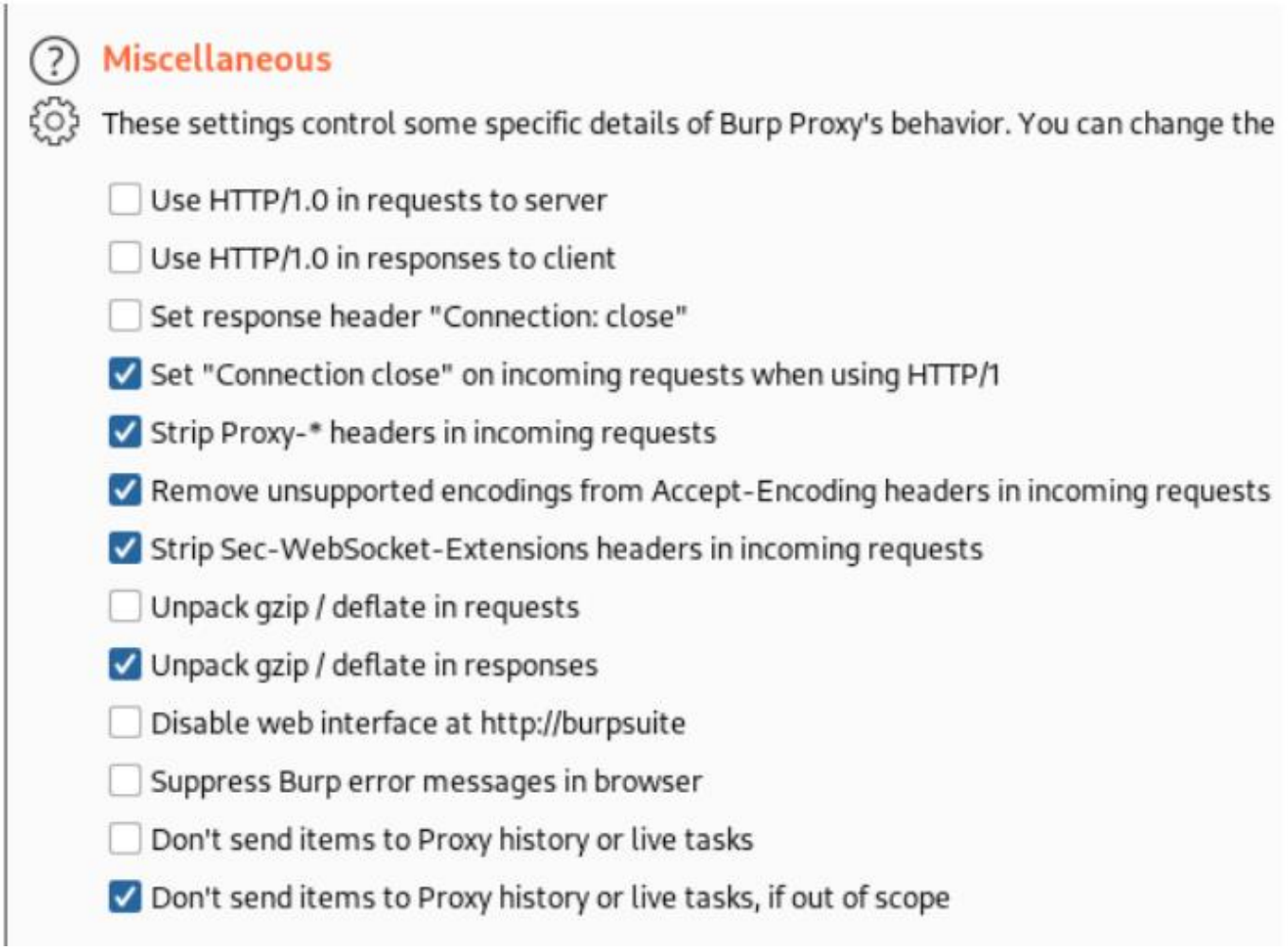

TLS Pass Through
These settings are used to specify destination web servers for which Burp will directly pass through TLS connections.
Add
Edit
Remove
Paste URL
Load ...
Enabled
Host / IP range
Port

Рисунок 2.14.


Miscellaneous
These settings control some specific details of Burp Proxy's behavior. You can change the
Use HTTP/1.0 in requests to server
Use HTTP/1.0 in responses to client
Set response header "Connection: close"
Set "Connection close" on incoming requests when using HTTP/1
Strip Proxy-* headers in incoming requests
Remove unsupported encodings from Accept-Encoding headers in incoming requests
Strip Sec-WebSocket-Extensions headers in incoming requests
Unpack gzip / deflate in requests
Unpack gzip / deflate in responses
Disable web interface at http://burpsuite
Suppress Burp error messages in browser
Don't send items to Proxy history or live tasks
Don't send items to Proxy history or live tasks, if out of scope

Ці параметри контролюють деякі деталі поведінки Burp Proxy. Доступні наступні варіанти: використання HTTP/1.0 у запитах до сервера. Цей параметр визначає, чи використовує Burp Proxy HTTP версії 1.0 у запитах до цільових серверів. Налаштування за замовчуванням включає використання будь-якої версії HTTP браузера. Тим не менш, може знадобитися версія 1.0, щоб деякі застарілі додатки та сервери працювали правильно. — Для відповіді на запити клієнта використовується HTTP/1.0. Усі сучасні браузери підтримують HTTP 1.0 і 1.1 відповідно. Примусове використання версії 1.0 іноді може бути корисним для контролю деяких елементів поведінки браузерів, таких як запобігання спробам виконувати HTTP pipelining. Це тому, що версія 1.0 має менше функцій. — Встановіть відповідний header “Connection: close”. Цей параметр також може бути корисним для запобігання HTTP pipelining в деяких ситуаціях. – Set “Connection: close” on incoming requests. Цей параметр також може бути корисним для запобігання HTTP pipelining в деяких ситуаціях. – Strip Proxy-* headers in incoming requests. Браузери іноді надсилають заголовки запитів, які містять інформацію, призначену для використання проксі-сервера. Зловмисні веб-сайти можуть спробувати змусити браузер включити конфіденційні дані в ці заголовки. Burp Proxy автоматично видаляє ці заголовки з вхідних запитів, щоб запобігти витоку інформації. Burp залишить ці заголовки незмінними після зняття цього параметра. — Виключити неприйнятні encodings з Accept-Encoding headers нових запитів. Зазвичай браузери використовують різні кодування в відповідях, наприклад для стиснення вмісту. Обробка відповідей Burp може відбуватися з деякими кодуваннями. Burp за замовчуванням видаляє непотрібне кодування, щоб зменшити його використання. Якщо сервер потребує підтримки непідтримуваного кодування, можливо, вам доведеться зняти прапорець з цієї опції. — Видаліть headers Sec-WebSocket-Extensions з надісланих запитів. Для розширень, пов’язаних із з’єднаннями WebSocket, браузери можуть підтримувати стиснення вмісту. Обробка відповідей Burp стикається з деякими кодуваннями. Щоб зменшити ймовірність використання розширень, цей заголовок за замовчуванням видаляється Burp. Якщо сервер потребує додаткового розширення, можливо, вам доведеться виключити цю опцію. — Розпакуйте GZIP/знижіть у запитах.

Деякі програми (часто ті, що використовують спеціальні клієнтські компоненти), стискають тіло повідомлення в запитах. Цей параметр контролює, чи Burp Proxy автоматично розпаковує стиснені тіла запиту. Зауважте, що деякі програми можуть зламатися, якщо очікують стиснення тіла, а стиснення було видалено Burp. – Unpack GZIP / deflate in responses. Більшість браузерів приймають у відповідях стиснений GZIP. Цей параметр визначає, чи автоматично Burp Proxy розпакує стислий контент відповідей. Зауважте, що ви часто можете запобігти спробам серверів стиснути відповіді, видаливши заголовок Accept-Encoding із запитів (можливо,

використовуючи функцію відповідності та заміни Burp Proxy). – Disable web interface at http://burp. Ця опція може бути корисною, якщо ви змушені налаштувати прослуховувач на прийом з'єднань із незахищеним інтерфейсом і хочете перешкодити іншим отримати доступ до інтерфейсу Burp у браузері. – Suppress Burp error messages in browser. Burp автоматично подає браузеру повідомлення про помилку, коли виникають певні помилки. Може бути корисно приховати ці повідомлення про помилки, щоб приховати його присутність, якщо ви хочете запустити Burp у прихованому режимі, щоб здійснювати атаки «людина посередині» проти користувача-жертви. — Не відправляйте товари в Live tasks або Proxy history. Цей параметр забороняє Burp записувати будь-які запити в історії проксі-сервера або надсилати їх до активних завдань, таких як пасивне сканування або живий аудит. Це може бути корисно, якщо ви використовуєте проксі-сервер Burp для певних цілей, наприклад, для автентифікації на верхніх серверах або виконання операцій зіставлення та заміни, і ви хочете уникнути накладних витрат на пам'ять і сховище, які тягне за собою ведення журналу. – Don't send items to Proxy history or live tasks, if out of scope. Завдяки цій опції запити, що виходять за межі області, не будуть записані в історію проксі-сервера або надіслані до активних завдань, таких як пасивне сканування або живий аудит. Корисно уникати накопичення даних проекту для елементів, які виходять за скоуп.

2.2. Роль Burp Suite у тестуванні безпеки web-додатків.

Burp Suite забезпечує аудит безпеки веб-додатків, перехоплення та аналіз трафік між клієнтом і сервером, а також тестування на проникнення для різних типів вразливостей, таких як SQL-ін'єкції, XSS-атаки та CSRF-атаки.Яка особливість BurpSuite як проксі?

Він дозволяє абсолютно все, що проходить через певний порт і браузер, «прослухати». Таким чином, багато прихованих багів, шпигунських скриптів, шкідливих редиректів і URL-адрес, витоків, вразливостей, несанкціонованих з'єднань і інших підозрілих дій можна знайти. В режимі реального часу кожен рядок коду, файл і HTTP-заголовок BurpSuite можна переглядати у візуальному Render або сирому RAW/HEX вигляді, перевіряти та тестувати, щоб знайти потенційні вразливості на вкладці Issues. Крім того, усе це подається в надзвичайно вишуканому та організованому вигляді, який можна експортувати. Це найважливіший інструмент для дослідника!Фаззінг можна використовувати як та де завгодно. Ним можна перебирати

логіни і паролі, виявляти XSS/SQL/xPath-ін’єкції, LFI/RFI шелли, приховані директорії, файли, URL-адреси та інші вразливості.

Проведемо енумерацію користувачів на прикладі веб-сайту WordPress. Доступ до неї можна отримати за допомогою таких URL-адрес, як domain.com/?author=n, де кожному автору присвоюється окремий ідентифікатор. Наприклад, введення domain.com/?author=1 відкриє веб-сайт одного автора, автор=2 відкриє веб-сайт іншого автора та так далі. Таким чином, можна визначити конкретні логіни користувачів, які використовуються для входу в систему. Якщо ви хочете уникнути цього, ви можете заборонити доступ до URL-адрес у файлі functions.php.

Це далеко не повний список пентестів, які дозволяють використовувати Burp Suite. Насправді, їх дуже багато. Усі вони привабливі та різноманітні. BurpSuite — це такий-собі конструктор, який дозволяє створювати власні техніки та методики хакерського ремесла, моделювати кіберзагрози та експлуатувати вразливості, поєднувати різні інструменти, щоб тестувати кібербезпеку на професійному рівні.

BurpSuite—це бездоганний інструмент пентестера і аналітика, який розкриває колосальні перспективи і безграничні можливості.

2.3. Застосування засобів Burp Suite для різних типів атак та вразливостей.

Burp Suite - це інструмент для тестування безпеки веб-додатків, який включає в себе низку функцій для виявлення та використання різних вразливостей. Ось кілька прикладів використання Burp Suite для різних типів атак та вразливостей:

Перехоплення та зміна трафіку:

-Використовуйте Proxy в Burp Suite для перехоплення трафіку між браузером та сервером.

-Модифікуйте запити та відповіді для тестування вразливостей.

SQL Injection:

-Використовуйте Intruder для впорядкованого впровадження SQL-запитів з метою знаходження вразливостей.

-Використовуйте Scanner для автоматизованого виявлення SQL-ін'єкцій.

Cross-Site Scripting (XSS):

-Використовуйте Repeater для внесення скриптів та тестування XSS-вразливостей.

-Використовуйте Decoder для кодування та декодування параметрів для обходу фільтрів.

Cross-Site Request Forgery (CSRF):

-Використовуйте Scanner для виявлення CSRF-вразливостей.

-Використовуйте Repeater для вручному відправлення зловмисних запитів.

Запити на вивільнення інформації:

-Використовуйте Intruder для повторного відправлення запитів з різними параметрами для виявлення витоку конфіденційної інформації.

Вразливості управління сесіями:

-Використовуйте Repeater для тестування атак на управління сесіями, такі як злам пароля або видалення сесії.

XML External Entity (XXE) вразливості:

-Використовуйте Intruder для впровадження зловмисних XML-параметрів з метою виявлення XXE-вразливостей.

Зовнішні виклики (Out-of-Band) атаки:

-Використовуйте функції Collaborator у Burp Suite для виявлення зовнішніх викликів, що можуть свідчити про вразливості, такі як Server-Side Request Forgery (SSRF).

Зауважте, що використання Burp Suite має бути дозволено власником веб-додатка та відповідно до моральних норм. Крім того, важливо пам’ятати, що ручне тестування також є важливою частиною тестування безпеки, оскільки автоматичні інструменти можуть не виявити всі потенційні вразливості.

Хоча б раз на місяць, аудит потрібно проводити. Це пов’язано з тим, що електронні ресурси та системи постійно містять дані, які змінюються, оновлюються та кешуються. Усі 24 години на добу відбуваються різноманітні фонові процеси, які можуть зазнавати внутрішнього та зовнішнього впливу. Основні проблеми, які виникають, пов’язані з постійним розвитком технологій, людськими факторами, появою багів і помилок у коді та системах захисту, які хакери намагаються виявити й використовувати.

Чим загрожує відсутність постійного аудиту безпеки? Ви можете не знати, які вразливості є на вашому веб-сайті та наслідки, які вони викликають. Зловмисники можуть зламати або використати ваш веб-сайт без вашого відома. Таким чином, аудит безпеки є однією з найважливіших частин підтримки веб-сайту.

Live scanning:

- Live Active Scanning, у цьому моді сканування Burp приймає індивідуальний запит до додатку, званий «базовим запитом», та змінює його різними способами, призначеними для запуску поведінки, які будуть вказувати на наявність різних уразливостей. Ці змінені запити надсилаються до програми, і отримані відповіді аналізуються. У багатьох випадках подальші запити будуть надсилатися, ґрунтуючись на результатах початкових зондів. Виставляємо значення Use suite scope, що означає брати скоуп завдань, заданих з вкладки Target->Scope.

- Live Passive Scanning, У цьому моді сканування Burp не відправляє жодних нових запитів у додаток - він просто аналізує вміст існуючих запитів та відповідей, знаходячи у них уразливості.

Цей пасивні методи може знайти такі види уразливостей:

- Передача паролів .

- Небезпечні атрибути cookie, такі як відсутність HttpOnly та безпечних прапорів .

- Ліберальний обсяг файлів cookie .

- Міжменний скрипт, включаючи і витік Referer .

- Форми з автозаповненням .

- Кешування вмісту в SSL .

- Подані паролі повертаються у наступних відповідях .

- Небезпечна передача сесійних токенів .

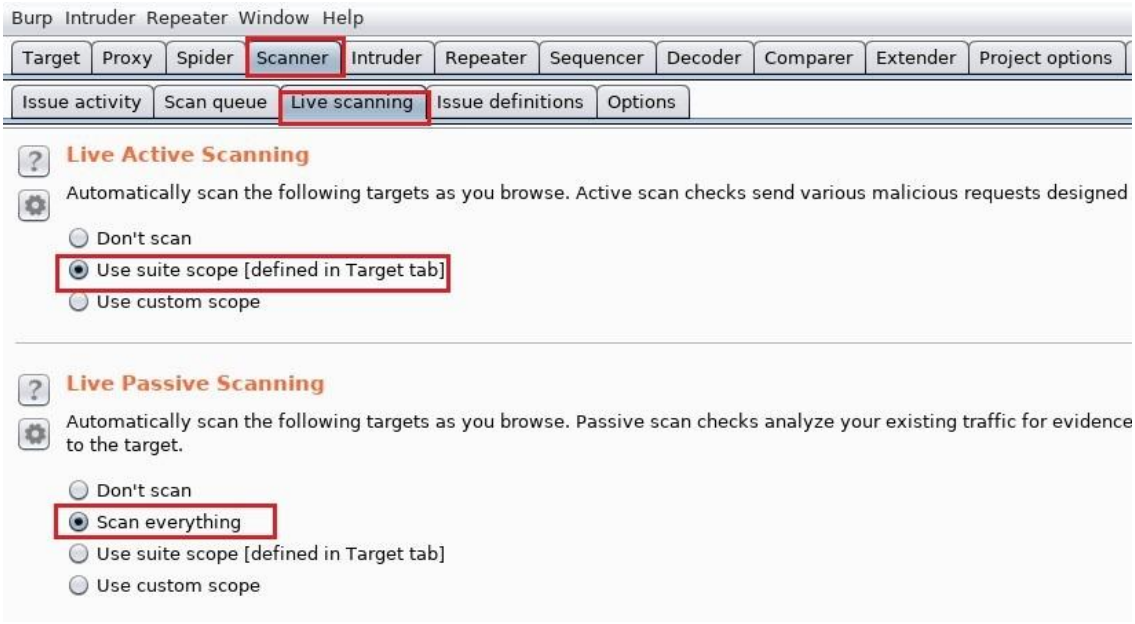
- Витік інформації, такий як внутрішні IP-адреси, адреси електронної пошти, сліди стека і т.д.

- Небезпечна конфігурація ViewState .

- Неоднозначні, неповні, неправильні чи нестандартні директиви Content-type ..

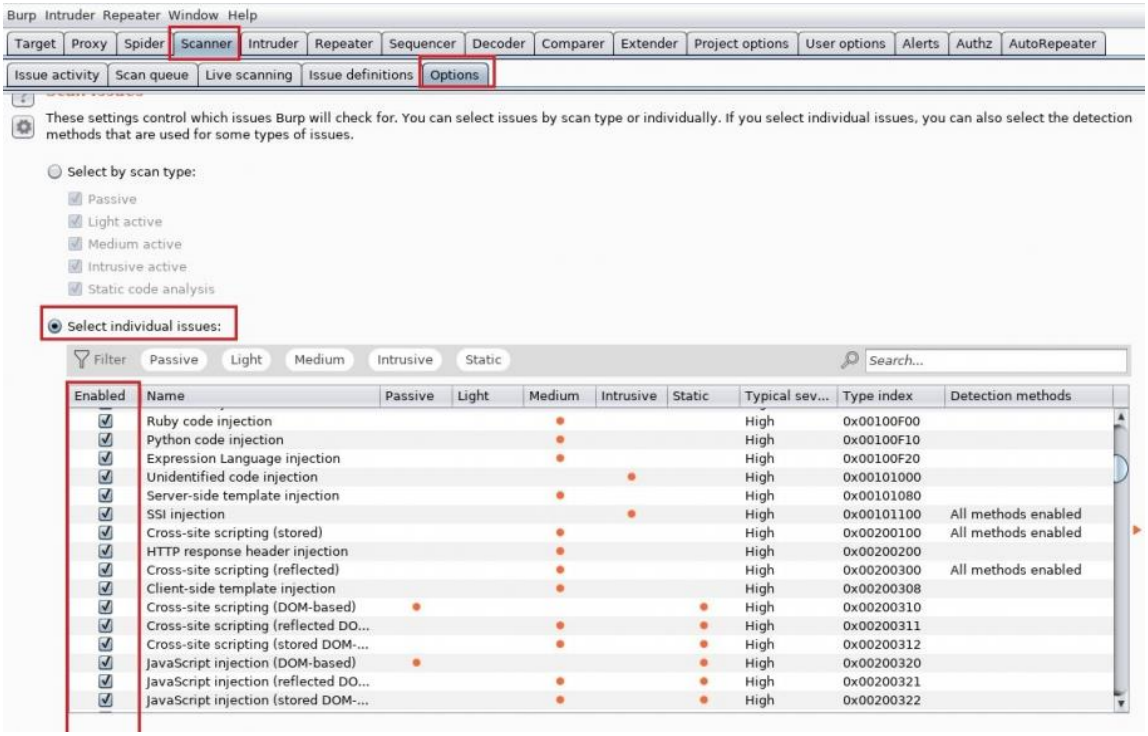
У цьому блоці вибираємо значення Scan everything, що означає сканувати всі ці вразливості.

Рисунок 2.15.



Потім переходимо у вкладку Scanner->Options, де задаватимемо конкретики типів уразливостей, які шукатимемо за допомогою сканера.

Рисунок 2.16.



Загалом для виявлення найбільш поширених вразливостей веб-додатків застосовується глобальне сканування , як за допомогою Burp Suite так і інших програмних засобів для більш точної інформації що до сайту.

У випадку коли сканер показує знайдену вразливість для більшої конкретики застосовується ручне тестування та перевірка на які саме пейлоуди реагує вразливості.

2.4. Переваги та обмеження використання Burp Suite у цьому контексті.

Burp Suite — це набір інструментів, які можна використовувати для аудиту безпеки веб-додатків. У ньому можна знайти інструменти для складання картки веб-програми, пошуку файлів і папок, модифікації запитів, фазінгу, підбору паролів і багато іншого. Крім того, у BApp store є магазин доповнень, який пропонує додаткові розширення, які можуть зробити додаток кращим. Варто відзначити появу мобільного помічника для дослідження безпеки мобільних додатків MobileAssistant для платформи iOS у найновішому релізі.

Burp Suite — це інтегрована платформа, яка дозволяє виконувати ручний або автоматичний аудит веб-додатків. Він має простий інтерфейс із табами, які були спеціально розроблені, щоб покращити та прискорити процес атаки. Сам інструмент працює як проксіруючий механізм, який перехоплює та обробляє всі запити браузера. Є можливість встановити сертифікат burp для аналізу з’єднань https.

Якщо подивитися статистику та репорти Bug Bounty програм – практично скрізь на скріншотах можна зустріти використання цього інструменту. Поруч із OWASP ZAP це найпопулярніший набір утиліт для тестування веб-додатків.

Переваги використання Burp Suite:

1.Зручний інтерфейс: Burp Suite має інтуїтивно зрозумілий інтерфейс, який полегшує використання для професіоналів з тестування на проникнення.

2.Proxy-перехоплення трафіку: Burp Suite Proxy дозволяє перехоплювати, модифікувати та аналізувати HTTP-трафік між браузером та сервером.

3.Automated Scanner: Засіб автоматичного сканування Burp Suite може автоматично виявляти багато загальних вразливостей, таких як SQL-ін'єкції чи Cross-Site Scripting (XSS).

4.Intruder для тестування великої кількості варіантів: Функціональність Intruder дозволяє вам автоматизувати та масштабувати тестування, наприклад, при виявленні SQL-ін'єкцій.

5.Repeater для повторного відправлення та тестування змін: Засіб Repeater дозволяє вам легко повторювати та тестувати змінені запити для атак та експлуатації.

6.Виявлення вразливостей через Collaborator: Функція Collaborator у Burp Suite дозволяє виявляти зовнішні виклики, які можуть вказувати на вразливості, такі як SSRF.

Обмеження використання Burp Suite:

1.Неповна автоматизація: Хоча Burp Suite надає зручні автоматичні інструменти, важливо розуміти, що не всі вразливості можуть бути виявлені автоматично, і ручна перевірка є необхідною.

2.Фальсифікація позитивних результатів: Автоматичні інструменти, включені в Burp Suite, можуть іноді давати фальшиво-позитивні або фальшиво-негативні результати, що вимагає ручного підтвердження.

3.Обмежена підтримка для деяких технологій: Деякі веб-додатки використовують специфічні технології чи відповіді, які можуть бути складні для аналізу автоматичними інструментами.

4.Не завжди ефективний для RESTful додатків: Деякі функції Burp Suite можуть бути менш ефективними при роботі з RESTful веб-додатками порівняно з традиційними додатками.

5.Вимагає досвіду та розуміння принципів безпеки: Ефективне використання Burp Suite передбачає розуміння технік тестування на проникнення та основ безпеки веб-додатків.

3 ПОРІВНЯЛЬНИЙАНАЛІЗ ІНШИХ ІНСТРУМЕНТІВ ТА ПРАКТИЧНИЙ ДОСВІД ВИКОРИСТАННЯ BURP SUITE

3.1. Порівняння Burp Suite із іншими популярними інструментами виявлення вразливостей.

Існує кілька інструментів для виявлення вразливостей в веб-додатках, і кожен з них має свої переваги та обмеження. Ось порівняння Burp Suite з деякими іншими популярними інструментами:

OWASP Zap:

Переваги Burp Suite:

-Burp Suite має більш розгорнутий та потужний інтерфейс. Зручний для використання Repeater і Intruder для тестування та експлуатації вразливостей.

Переваги OWASP Zap:

-OWASP Zap є абсолютно безкоштовним та відкритим інструментом. Запропонована спільнота та підтримка проекту OWASP.

Nessus:

Переваги Burp Suite:

-Burp Suite спеціалізується на тестуванні на проникнення веб-додатків, тоді як Nessus зазвичай використовується для тестування мережевих вразливостей. Burp Suite дозволяє проводити детальний аналіз HTTP-трафіку.

Переваги Nessus:

-Nessus включає сканування мережі та виявлення вразливостей на різних рівнях.

Acunetix:

Переваги Burp Suite:

-Burp Suite є більш універсальним інструментом, що дозволяє вам ручно проводити тестування на проникнення. Розширюваність та гнучкість Burp Suite завдяки великій кількості розширень.

Переваги Acunetix:

-Acunetix має велику базу знань про вразливості та здатний виявляти їх автоматично.

Nexpose:

Переваги Burp Suite:

-Burp Suite забезпечує велику кількість інструментів для ручного та автоматичного тестування. Великий акцент на аналізі та взаємодії з HTTP-трафіком.

Переваги Nexpose:

-Nexpose в першу чергу спеціалізується на скануванні вразливостей в мережі та інфраструктурі.

Qualys Web Application Scanning (WAS):

Переваги Burp Suite:

-Burp Suite дозволяє аналізувати та модифікувати кожен елемент HTTP-запиту, що важливо для тестування на проникнення. Зручний інтерфейс для ручного аналізу вразливостей.

Переваги Qualys WAS:

-Qualys WAS надає хмарне рішення для автоматичного виявлення вразливостей веб-додатків.

Вибір між цими інструментами залежить від конкретних потреб, особливостей веб-додатків та персональних вподобань. Багато фахівців в галузі тестування на проникнення використовують комбінацію різних інструментів для досягнення більш вичерпного покриття та виявлення вразливостей.

3.2. Реальні приклади використання Burp Suite для виявлення та виправлення вразливостей.

Burp Suite легко використовувати, оскільки всі його утиліти та плагіни можуть взаємодіяти один з одним. Ви можете встановити Burp Suite як проксі в налаштуваннях браузера. Для сайтів, які працюють за HTTPS, також потрібно встановити згенерований TLS-сертифікат Burp. У цьому випадку Burp Proxy зберігатиме всі ваші дії в браузері, включаючи надіслані запити та отримані відповіді. Крім браузера, на десктопі можна спробувати перенаправляти HTTP-трафік з мобільних додатків у Burp, а також будь-який HTTP-трафік з десктопних додатків або пристроїв IoT. Інші інструменти можуть отримувати HTTP-запити з історії proxy та працювати з ними.

Наприклад, необхідно перевірити, чи можна влаштувати bruteforce-атаку на підбір OTP-коду. Для цього потрібно перехопити запит на перевірку OTP і передати його інструменту Intruder.

Після цього нам достатньо буде виділити місце в HTTP-запиті, яке потрібно атакувати, і налаштувати значення, які будуть застосовуватися – у цьому випадку у нас перебір чисел від 0000 до 9999.

Рисунок 3.1.

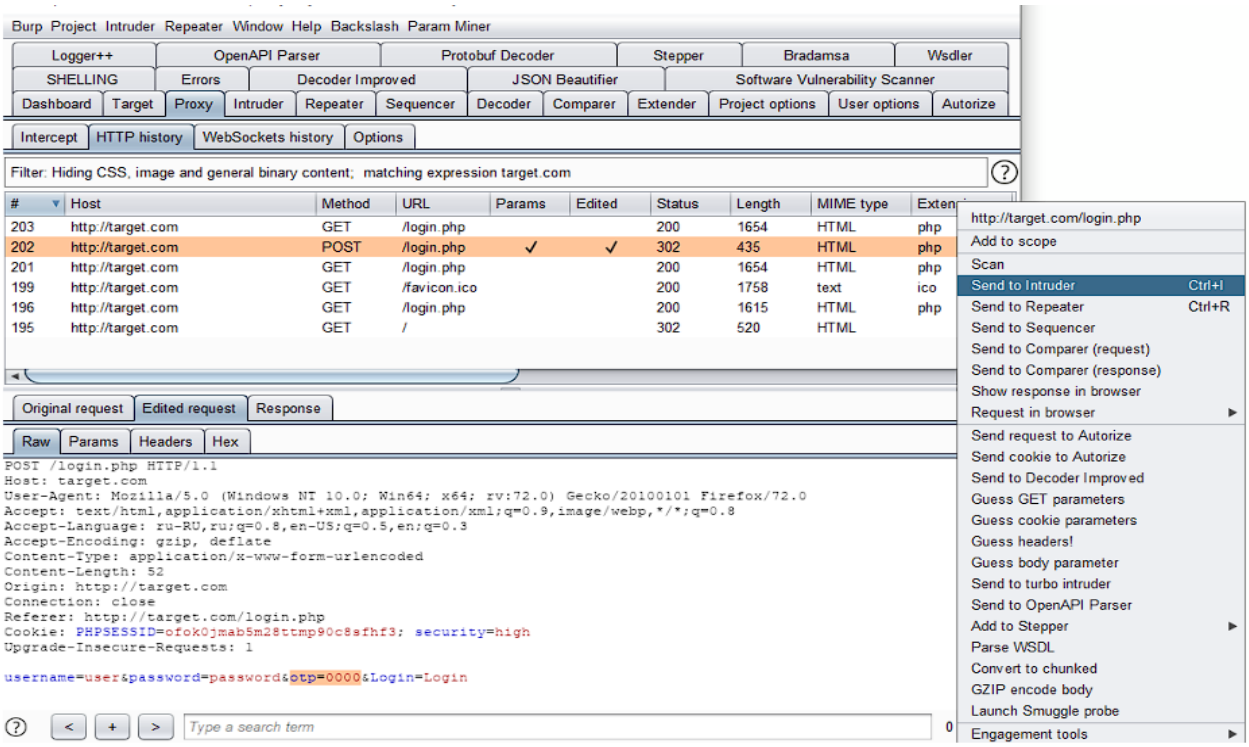
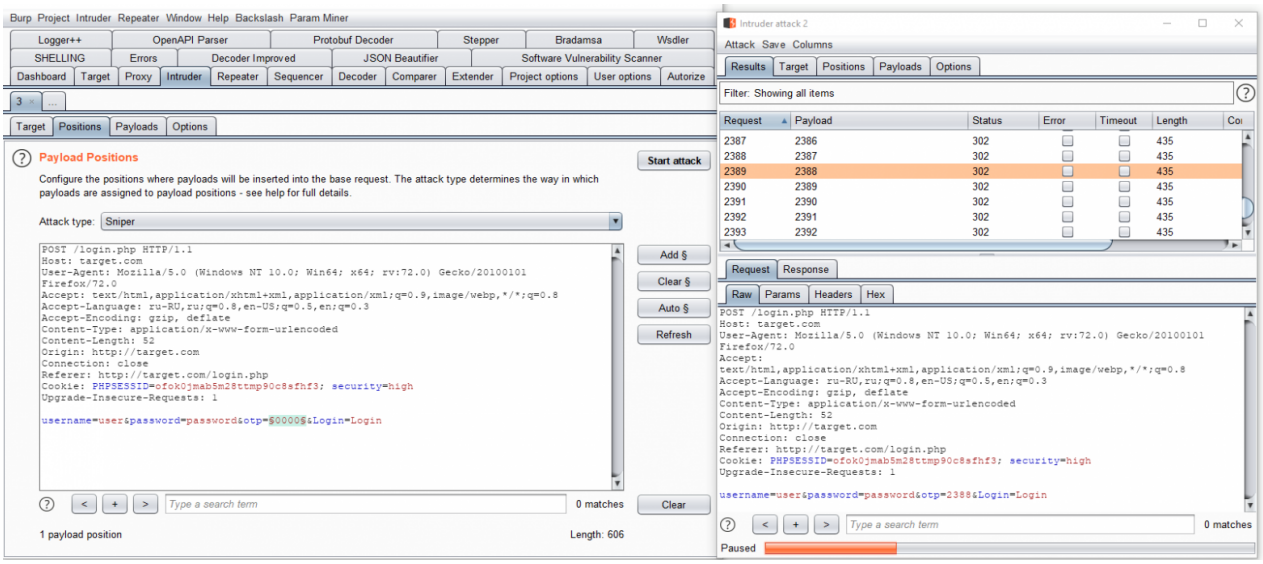
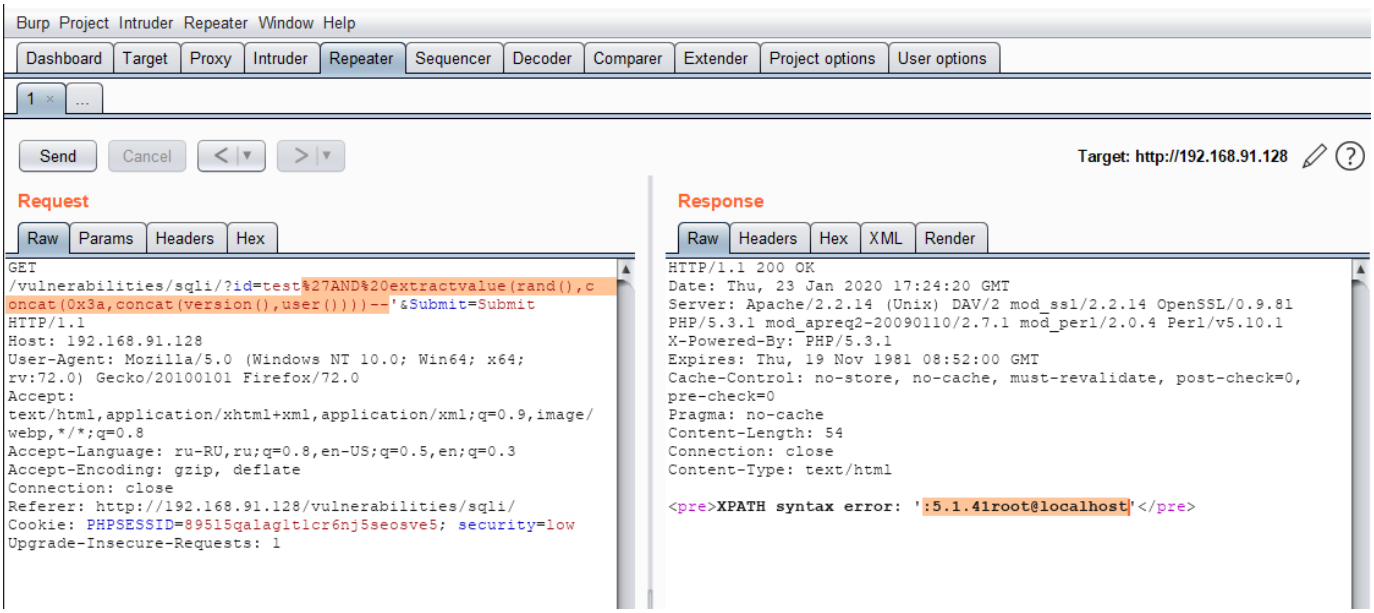


Рисунок 3.2.



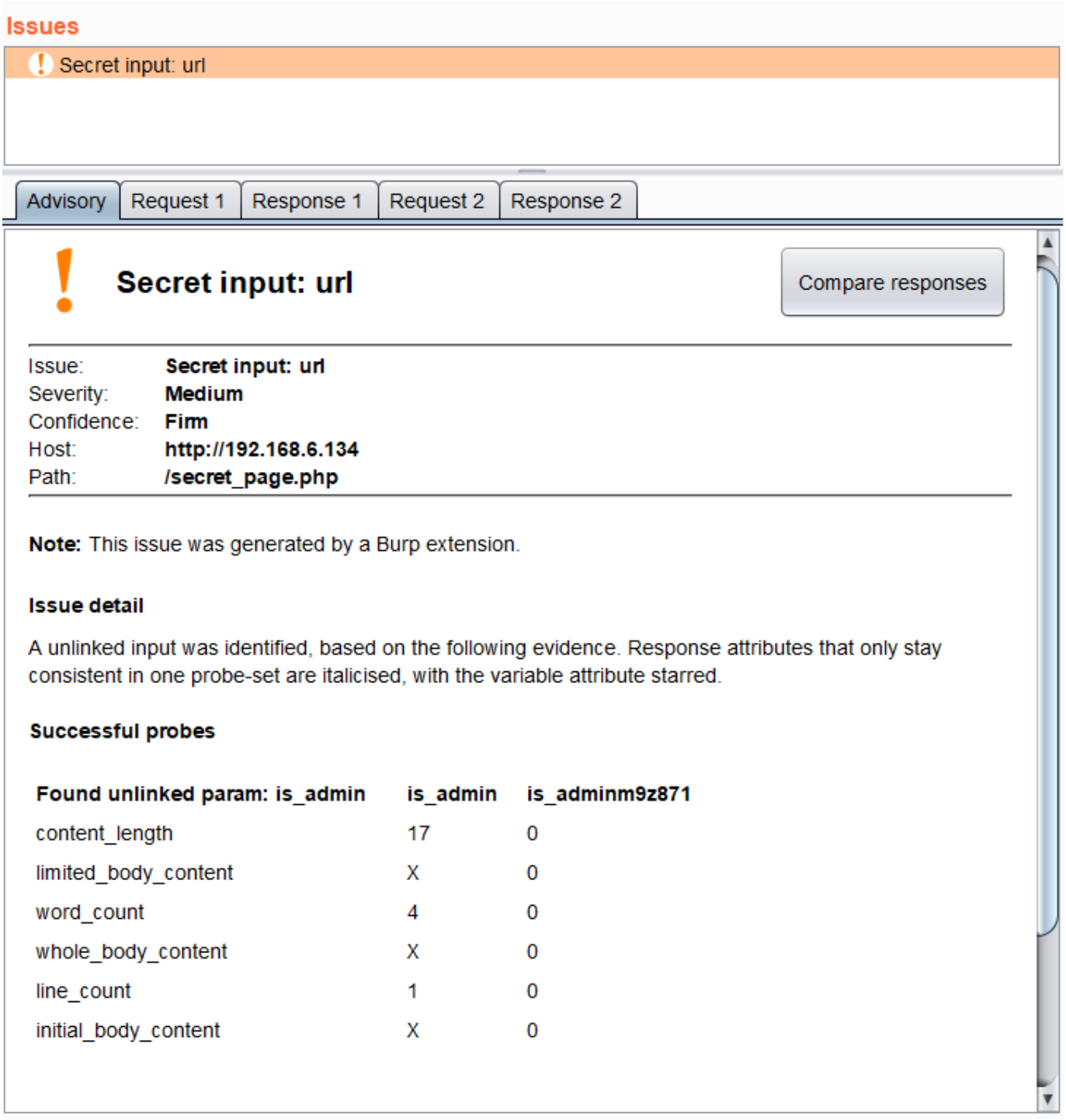
Можна також використовувати інструмент Repeater для вручну виконання будь-яких перевірок, тестів або атак веб-програм. Наприклад, можна легко проексплуатувати SQL-ін'єкцію, якщо інші інструменти, такі як sqlmap, не можуть з нею працювати.

Рисунок 3.3.



Коли ми говорили про Арджуна, ми згадали про парам-мінера. Цей інструмент схожий на Arjun, але працює в Burp і може знаходити приховані заголовки та cookie, крім параметрів. Спочатку він був розроблений як інструмент для пошуку прихованих параметрів, які можуть бути корисними для виявлення вразливостей, схожих на забруднення кешу веб-сайту.

Рисунок 3.4.

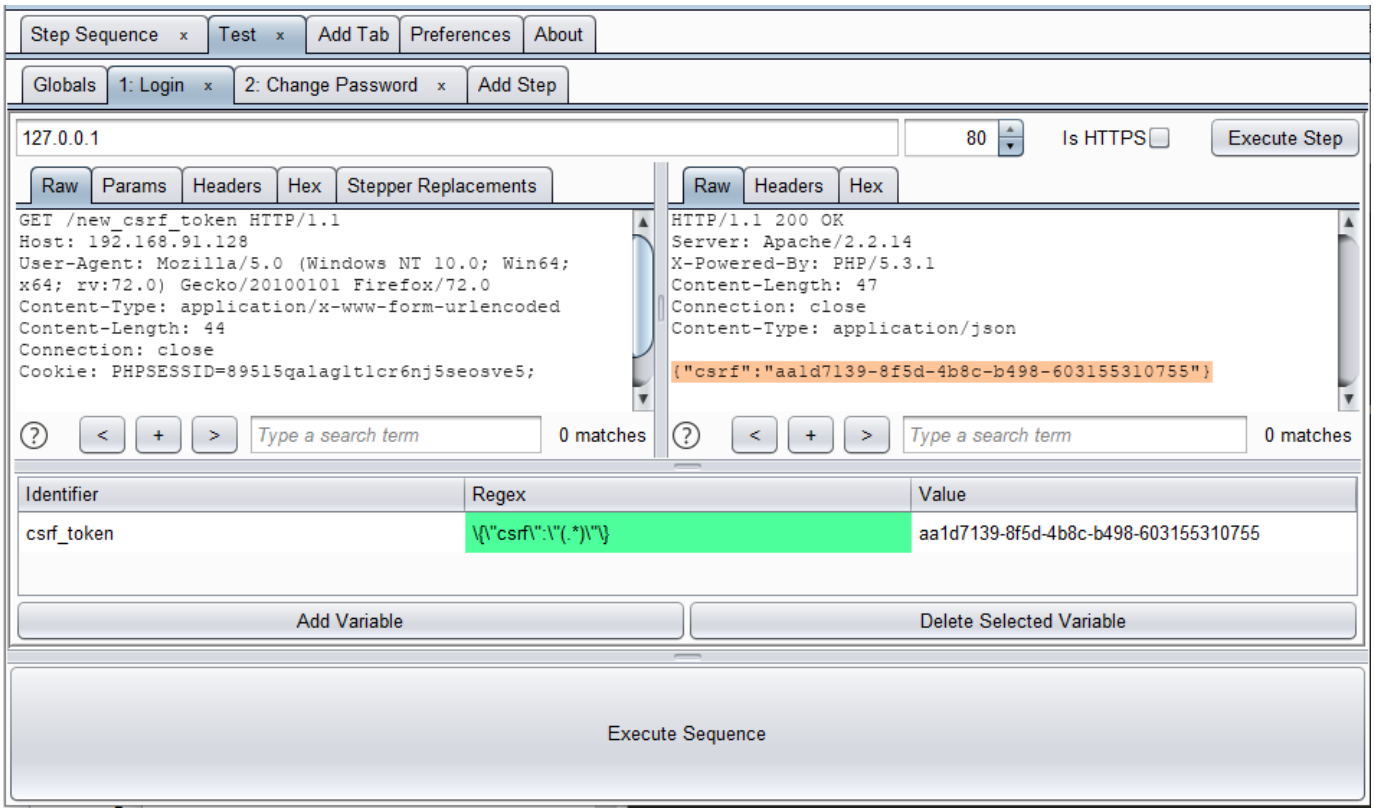


Плагін допомагає знаходити приховані GET/POST-параметри, параметри JSON-запиту, HTTP-заголовки, Cookie.

Дозволяє запускати як аналіз кількох запитів, і всього трафіку.

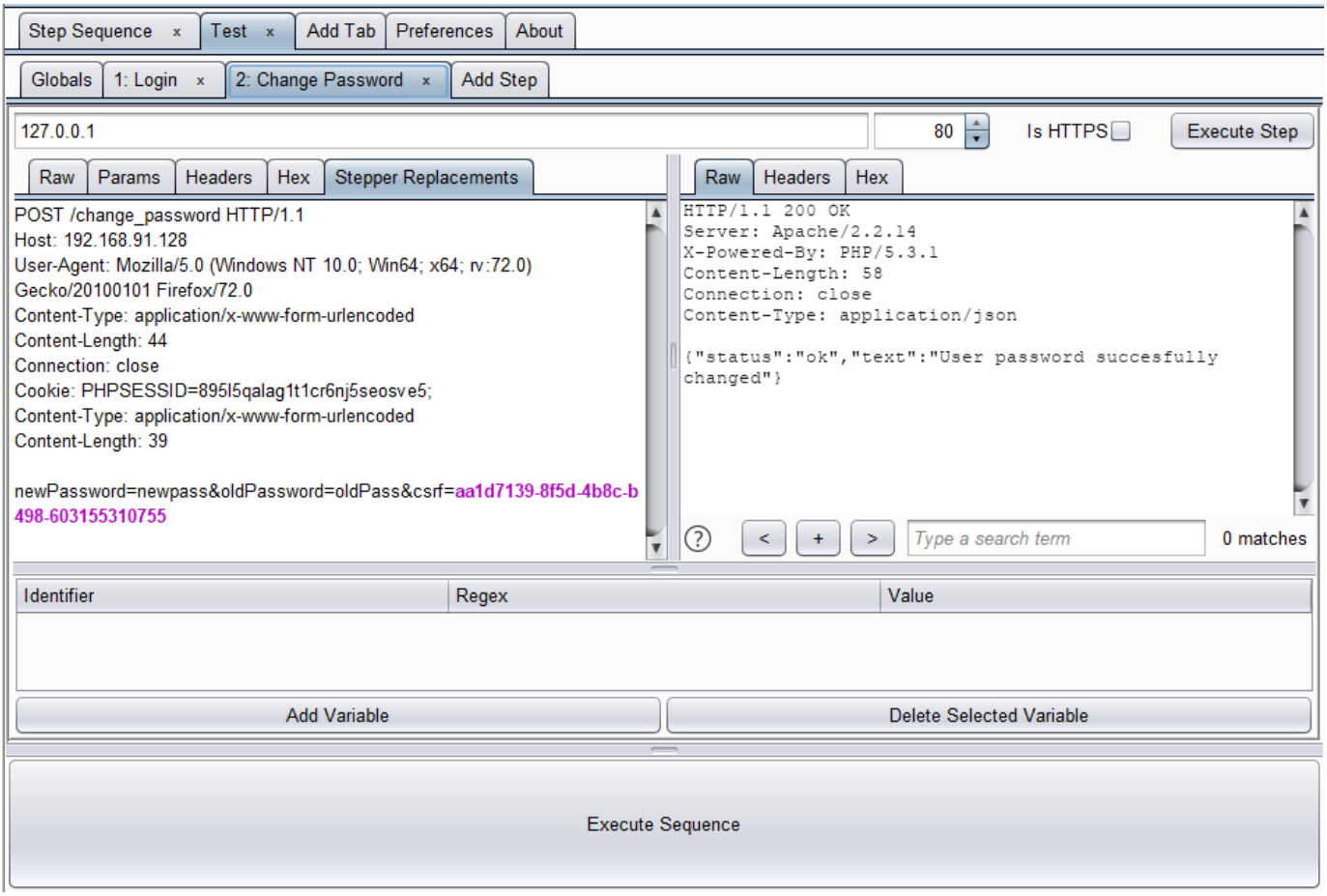
Здається, вам потрібно буде отримати новий CSRF-токен для кожного нового запиту. Це дуже важко зробити вручну. Замість цього ви можете виконати послідовність у Stepper, яка отримає CSRF-токен спочатку, а потім виконає потрібний вам запит. Для цього вам потрібно буде в першому запиті вказати параметр і додати його як змінну, наприклад csrf_token.

Рисунок 3.5.



А потім у наступному запиті вказати, куди цей параметр підставити, використовуючи назву змінної ($VAR:csrf_token$). У результаті Stepper зможе виконати коректну послідовність запитів. Результати підстановки можна побачити у вкладці Stepper Replacements.

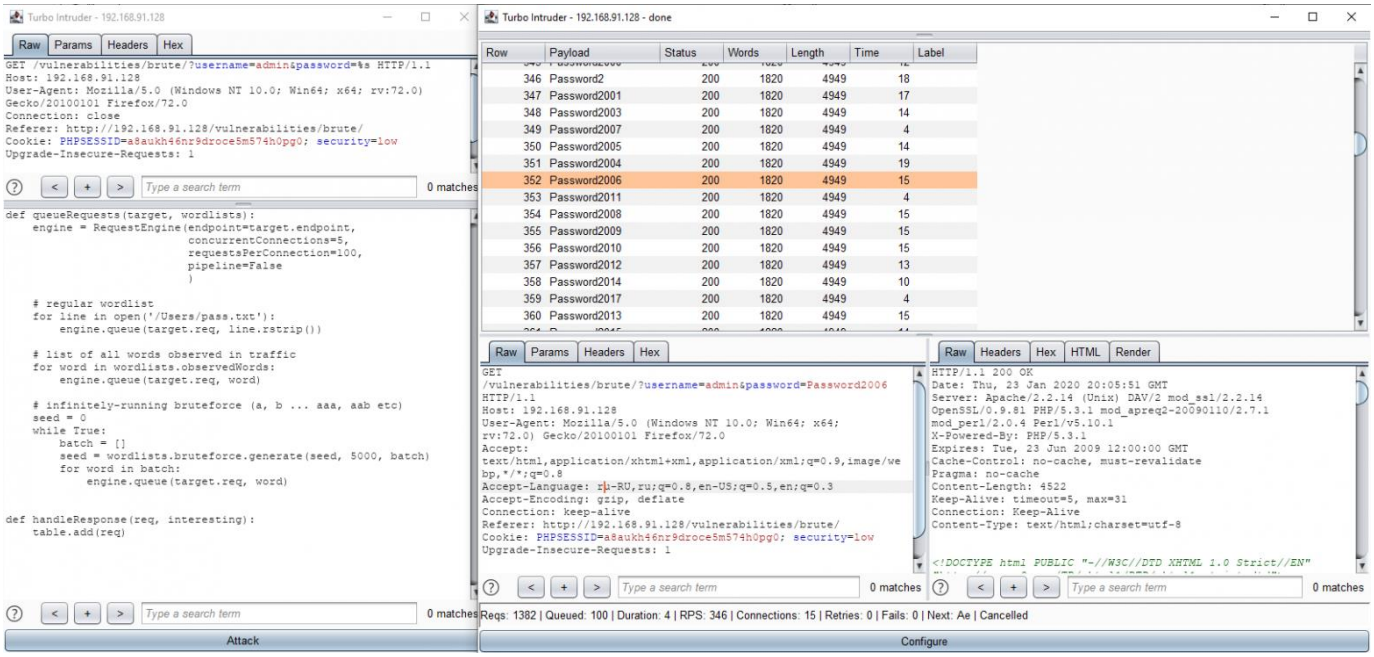
Рисунок 3.6.



Turbo Intruder - найшвидший аналог Intruder, оснащений скриптовим двигуном для відправки та аналізу великої кількості HTTP-запитів. Корисний для тих, хто потребує швидкості. У ньому міститься невеликий скриптовий движок Python, який має функції для тестування стану гонки (наприклад, одномоментне відправлення запитів), що робить його надзвичайно ефективним у пошуку вразливостей, пов’язаних зі станом гонки. Розширення дозволяє заскриптувати різноманітну логіку, таку як багатоступінчасту автентифікацію.

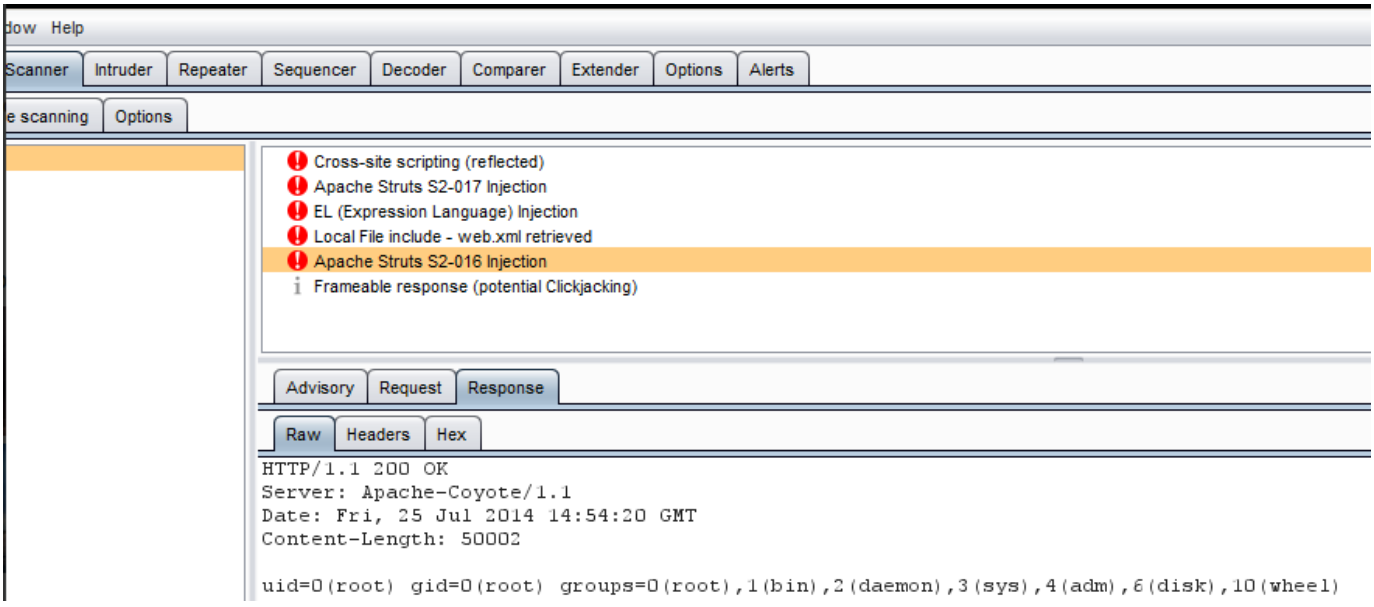
Приклад простий bruteforce-атаки з використанням Turbo Intruder:

Рисунок 3.7.



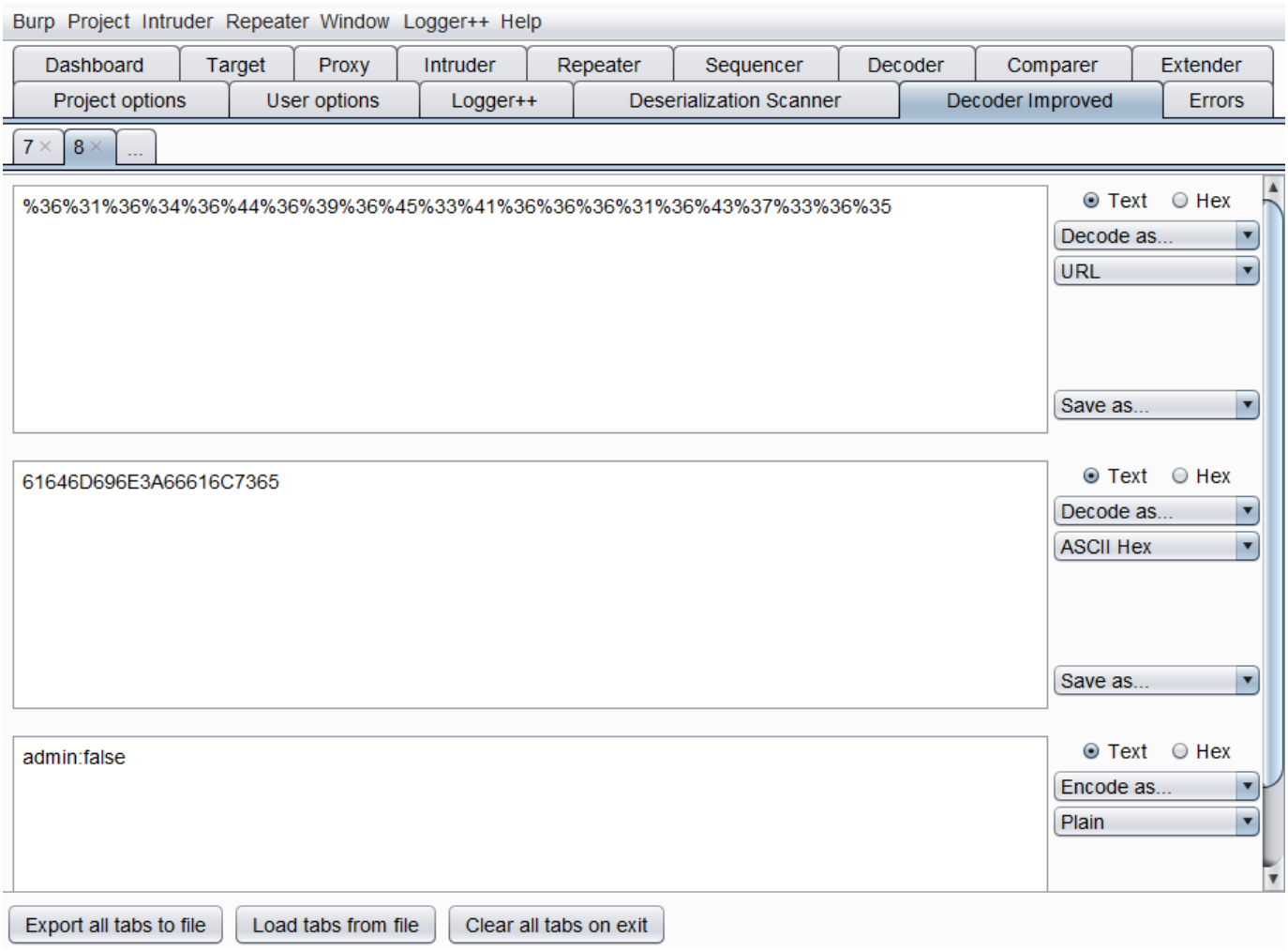
Плагін J2EEScan призначений для атак на програми J2EE (Java 2 Enterprise Edition). Велика кількість перевірок для Java-додатків включена в «під капотом». Це включає перевірки відомих вразливостей, таких як «Apache Struts». Стандартний бурп-сканер також має перевірки. Хоча цей сканер не завжди знаходить щось, його під рукою варто мати.

Рисунок 3.8.



Decoder Improved – по суті, звичайний декодер. Вбудований декодер Burp не такий вже хороший. Його недоліки включають відсутність вкладок, незручність використання Hex-редактора та невелику кількість можливих форматів даних. з цими проблемами та допомагає подолати Декодер покращений.

Рисунок 3.9.



Підтримує все, що є в Decoder, а також дозволяє використовувати алгоритми стиснення (Gzip, Zlib), хешувати дані, а також змінює систему числення (від base 2 base 32).

Підтримує функціональність вкладок.

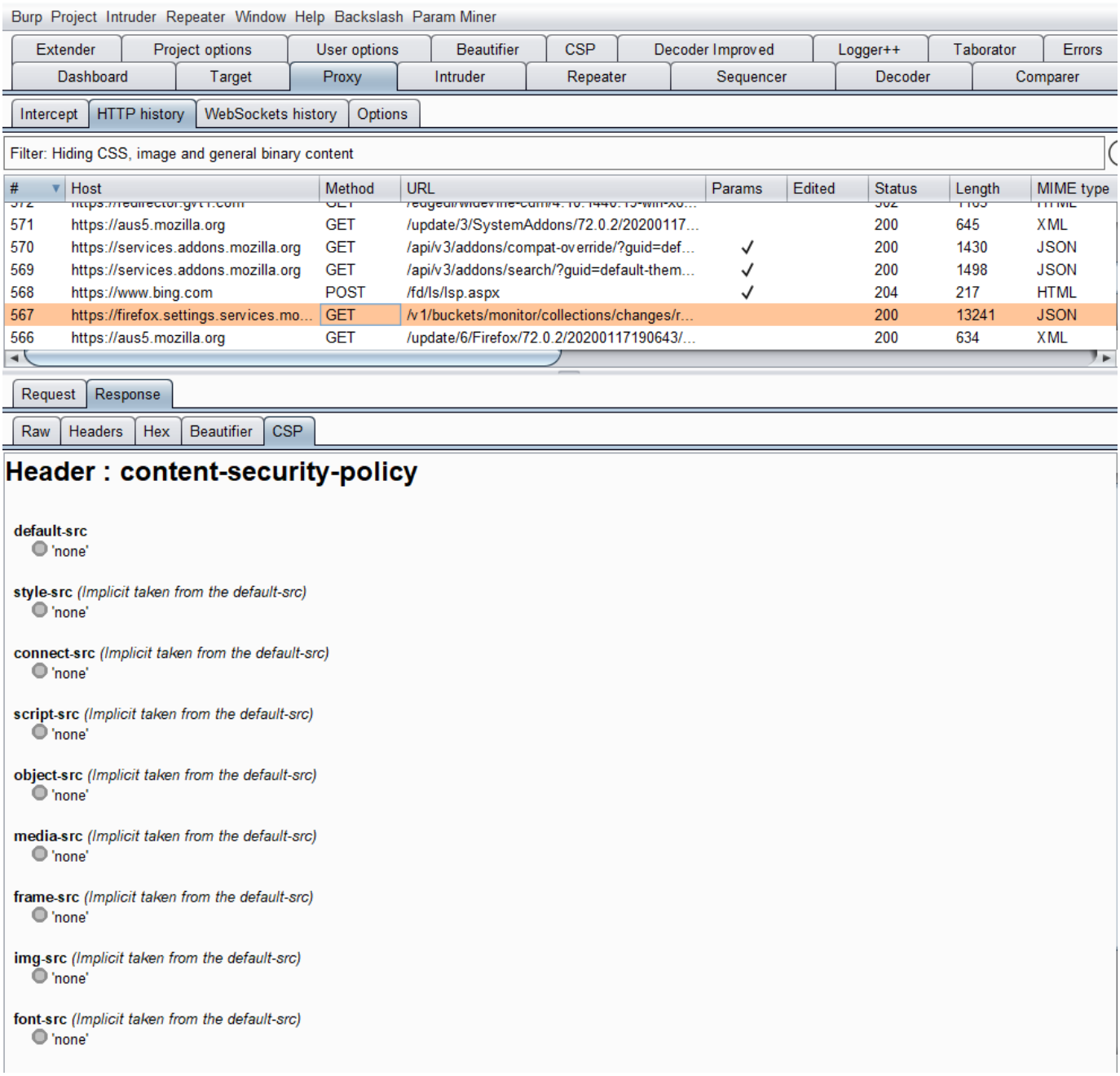
Має покращений Hex-редактор.

Пропонує функціональність регулярних виразів, що дозволяє в процесі кодування/декодування легко підміняти дані.

Підтримує режим заміни лише спецсимволів HTML/URL, тоді як буквено-цифрові символи залишаються без змін.

CSP Auditor - плагін, який допомагає виявити слабкості конфігурації CSP. Крім того, він парсить CSP з відповіді сервера і відображає їх у більш зручному для читання та аналізу вигляді.

Рисунок 3.10.



3.3. Аналіз результатів тестування та їх вплив на безпеку web-додатку.

Аналіз результатів тестування може значно впливати на безпеку веб-додатків і є важливою частиною процесу розробки веб-додатків. Нижче наведено кілька основних етапів і проблем, які можна врахувати під час аналізу результатів тестування та впливу цих результатів на безпеку веб-додатків.:

Запитання безпеки: Чи були знайдені вразливості, такі як SQL-ін'єкції, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery) або інші? Як реагує додаток на невірний ввід або неправомірні запити?

Дослідження вразливостей: Які вразливості були виявлені, і наскільки серйозні вони є? Які можливі наслідки використання цих вразливостей для атакування додатку?

Контроль доступу: Чи правильно реалізований контроль доступу до функціональності додатку? Чи є можливість отримання несанкціонованого доступу до конфіденційної інформації чи функціональності?

Захист від зловживань: Чи є захист від автоматизованих атак, таких як брутфорс або DDoS (розподілене заперечення обслуговування)?

Захист даних: Як взаємодіє додаток з конфіденційною інформацією? Чи забезпечено достатній рівень шифрування для передачі та зберігання даних?

Моніторинг і реагування: Які засоби моніторингу та журналювання використовуються для виявлення подій, пов'язаних з безпекою? Які процедури відновлення та реагування на інциденти налагоджені?

Оновлення та патчі: Чи проводяться регулярні оновлення безпеки для всіх залежностей та компонентів додатку?

Тестування виробничого середовища: Чи було проведено тестування безпеки виробничого середовища? Чи враховані особливості конфігурації виробничого серверу?

Після аналізу результатів тестування рекомендується розробляти план вдосконалення безпеки, виправляти виявлені вразливості та вдосконалювати процеси розробки для запобігання майбутнім проблемам. Також важливо надавати пріоритет заходам безпеки в усіх етапах розробки та підтримки додатку.

Як працює аудит забезпечення якості під час тестування?

Забезпечення якості – це серія послідовних дій для оцінки методології та процесів тестування. Вони визначають потенційні ризики, загрози або проблеми.

Крім того, внутрішні або зовнішні аудитори можуть проводити перевірки. Мета полягає в тому, щоб мінімізувати витрати часу та бюджету.

Аудити проводяться для того, щоб переконатися, що товари відповідають стандартам якості, а також визначити способи покращення методів і процедур тестування. Аудит також дозволяє ефективно тестувати кінцевий продукт. З цієї причини він містить кілька методів тестування, включаючи створення, тестування, виправлення помилок і написання протоколу тестування.

Для чого призначене перевірка якості аудиту?

Виходячи з нашого досвіду, будь-яка організація, яка проводить аудит забезпечення якості, приймає це рішення після багатьох помилок і труднощів. Ця стратегія викликає жаль. Аудит стосується таких питань, як:

Відповідність стандартам: відповідність таким стандартам, як ISO, CMMI та IEEE, необхідна для підтримки якості продукції та задоволення клієнтів.

Ідентифікація та зменшення ризиків: це допомагає організаціям уникнути дорогої переробки перед випуском продукту.

Удосконалення процесів і процедур тестування: аудит визначає аспекти, які потребують особливої уваги. Результати мають спричинити значні зміни в ефективності та результативності тестування, щоб відповідати мінливим потребам ринку та вимогам зацікавлених сторін.

Гарантія задоволеності зацікавлених сторін: аудит гарантує, що кінцевий продукт відповідає вимогам клієнтів, розробників, кінцевих користувачів і регуляторів. Це дозволяє організаціям задовольняти потреби всіх зацікавлених сторін і підтримувати свою репутацію на ринку.

Що отримує клієнт після QA аудиту?

Після перевірки контролю якості клієнт отримує детальний звіт. Документ зазвичай містить наступне:

Опис аудиторського звіту: це резюме аудиту. Він надає повний огляд аудиту, включаючи його обсяг, цілі та методологію.

Результати та висновки: цей розділ документа містить результати аудиту процесу, процедур і кінцевого продукту. У ньому проблеми класифікуються за ступенем важливості, а також за кількістю варіантів для їх вирішення.

Результати аудиту дають конкретні рекомендації щодо способів проведення тестування з більшою ефективністю. Рекомендації ранжуються відповідно до ступеня серйозності проблеми та того, наскільки вони впливають на кінцевий продукт.План дій: містить конкретні заходи щодо вирішення питань аудиту, включаючи часові рамки та відповідальність за кожен вид діяльності.

Висновок: це узагальнює результати аудиту та підкреслює важливість вирішення виявлених проблем, щоб переконатися, що кінцевий продукт відповідає бажаним стандартам якості.

Таким чином, клієнт отримує детальну інформацію про гарантію якості. Він показує сліпі області, які потребують розвитку, і дає корисні поради щодо вирішення проблем. Після цього клієнт може використовувати ці дані для зміни методів перевірки, що зрештою призведе до покращення якості кінцевого продукту та задоволення зацікавлених сторін.

Використовуйте найкращі практики аудиту якості

Деякі з найкращих практик для нашої команди щодо аудитів забезпечення якості:

Компетентність і незалежність: Аудитор повинен мати достатню кваліфікацію та досвід у тестуванні та аналізі. Він також повинен бути незалежним, неупередженим і не мати особистої чи фінансової зацікавленості в результатах аудиту.

Використання методу планування та аудиту, що ґрунтується на оцінці ризику: виявляються потенційні слабкості та проводиться оцінка ризиків. Цей процес полягає в тому, щоб мати глибоке розуміння ризиків, пов’язаних із процесом тестування.

Ведення записів аудиту: будь-які дії, включаючи шаблон плану аудиту забезпечення якості, звіт про аудит і будь-які підтверджуючі докази, мають бути ретельно документовані. Ця інформація залишається в архіві компанії замовника, і вона може бути використана в майбутньому.

Ефективна комунікація: Аудитор повинен залишити відкритими канали комунікації з командою QA, включаючи регулярні оновлення статусу, звіти про хід і відгуки.

Забезпечити зв’язок і можливості для вдосконалення тестування. Аудит слід розглядати як постійний процес покращення якості в тісній співпраці аудиторів і спеціалістів з контролю якості.

ВИСНОВОК

Магістерська робота була присвячена вивченню та аналізу технології виявлення вразливостей web-додатків на основі Burp Suite. Основні висновки та результати дослідження можна сформулювати наступним чином:

Ефективність Burp Suite: Дослідження підтвердило високу ефективність Burp Suite виявлення різноманітних вразливостей, таких як SQL-ін'єкції, Cross-Site Scripting (XSS) та інші. Інструмент надає зручний інтерфейс та потужність для проведення ретельного тестування безпеки web-додатків.
Автоматизація та Ручний аналіз: Burp Suite надає можливості як автоматизованого, так і ручного аналізу вразливостей. Ручний аналіз дозволяє виявляти складні вразливості, які можуть уникати автоматизованим засобам, підвищуючи загальний рівень безпеки додатку.

Рекомендації для Практики: З врахуванням результатів дослідження, рекомендується впроваджувати Burp Suite в процес розробки та тестування web-додатків. Регулярне навчання команди з використання Burp Suite допоможе максимально використовувати його можливості для забезпечення високого рівня безпеки.
Майбутні Перспективи: Під час дослідження були виявлені напрямки для подальшого розвитку та удосконалення Burp Suite. Рекомендується подальше вдосконалення інтеграції з іншими інструментами, а також розширення підтримки сучасних технологій та фреймворків.

У цілому, результати дослідження свідчать про важливість використання Burp Suite для забезпечення високого рівня безпеки web-додатків та надають підстави для подальших досліджень у цьому напрямку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Офіційний сайт Burp Suie: https://portswigger.net/burp (дата звернення: 13.09.2023).
2. Документація Burp Suie: https://portswigger.net/burp/documentation (дата звернення: 20.09.2023).
3. Методи тестування за допомогою Burp Suie: https://hackyourmom.com/servisy/soft/burp-suite-owasp-top-10-owasp-juice-shop/ (дата звернення: 26.09.2023).
4. QUALITY ASSURANCE AUDITS IN TESTING: A GUIDE TO BEST PRACTICE: https://luxequality.com/blog/quality-assurance-audits-in-testing/ (дата звернення: 03.10.2023).
5. Burp Suie: https://svyat.tech/scanning-web-application-with-burp-suite (дата звернення: 14.10.2023).
6. Основи пентестінгу: https://kr-labs.com.ua/blog/penetration-testing-with-burpsuite/ (дата звернення: 25.10.2023).
7. Introduction to Web Application Security Testing with Burp Suite - OWASP: https://owasp.org/www-pdf-archive/Burp_Suite_wiki.pdf (дата звернення: 03.11.2023).
8. Web Application Pentesting with Burp Suite - Medium: https://medium.com/bugbountywriteup/web-application-pentesting-with-burp-suite-4f7f48fc4a4c (дата звернення: 13.11.2023).
9. OWASP Testing Guide v4: https://owasp.org/www-project-web-security-testing-guide/ (дата звернення: 22.11.2023).
10. Web Security Academy - PortSwigger: https://portswigger.net/web-security (дата звернення: 01.12.2023).
11. Stack Overflow - Burp Suite: https://stackoverflow.com/questions/tagged/burp-suite (дата звернення: 03.12.2023).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)