ATTACK DEFENSE

by PentesterAcademy

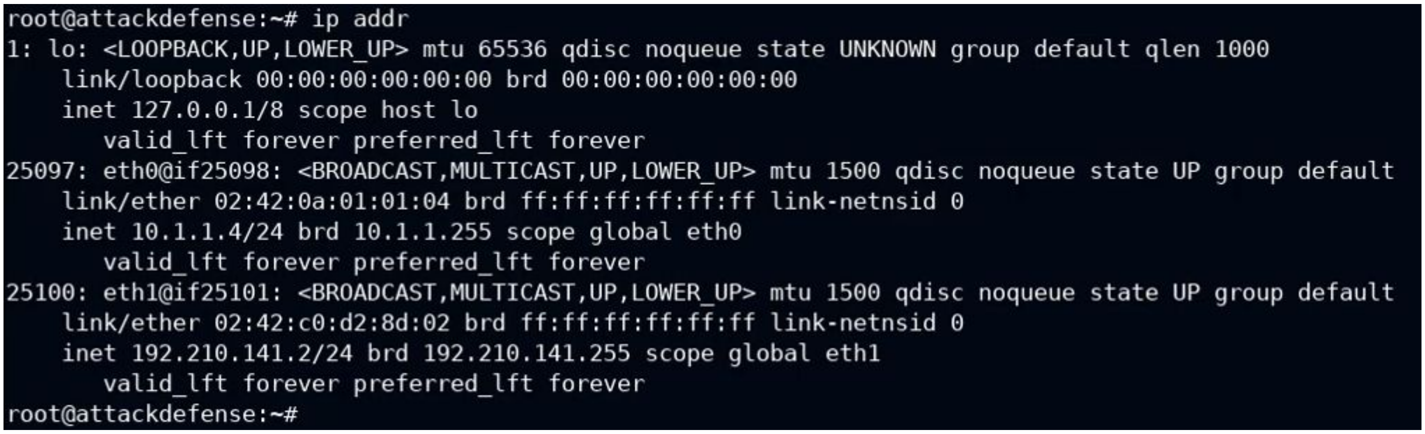
Name	Scanning Web Application with ZAProxy
URL	https://attackdefense.com/challengedetails?cid=1888
Type	Webapp Pentesting Basics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Scan the web application with ZAProxy and identify the possible vulnerabilities.

Step 1: Identifying IP address of the target machine

Command: ip addr

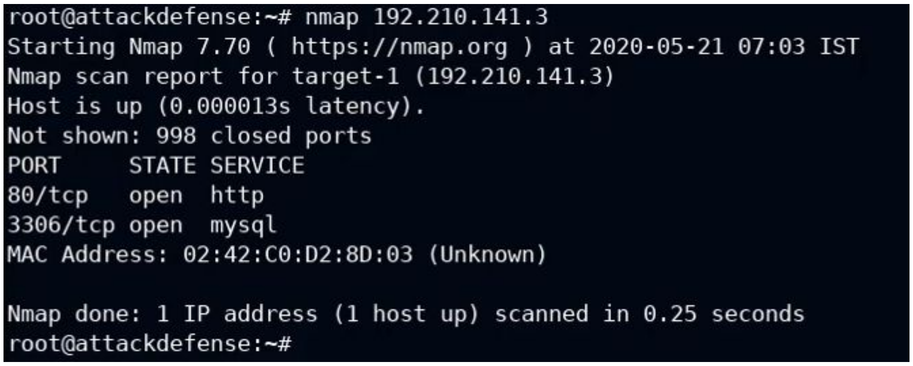


```
root@attackdefense:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
25097: eth0@if25098: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:01:01:04 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.1.4/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
25100: eth1@if25101: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:d2:8d:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.210.141.2/24 brd 192.210.141.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

The IP address of the attacker machine is 192.210.1412. The target machine is located at the IP address 192.210.1413

Step 2: Identifying open ports.

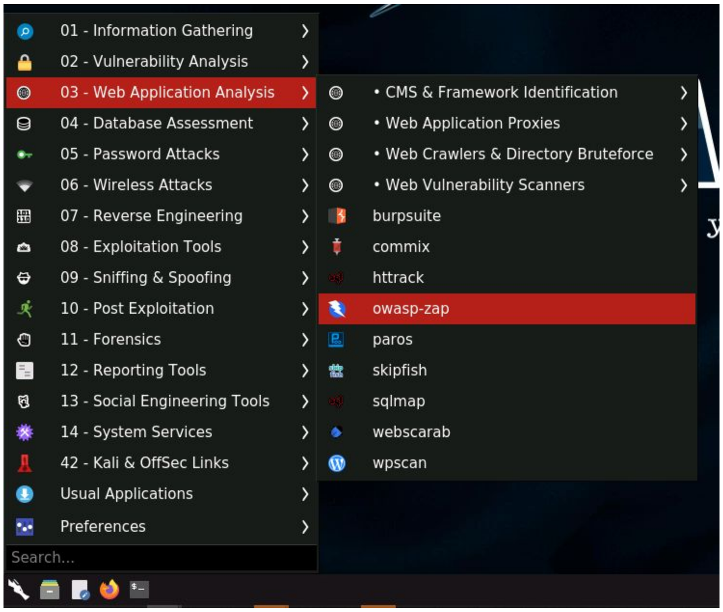
Command: nmap 192.210.1413



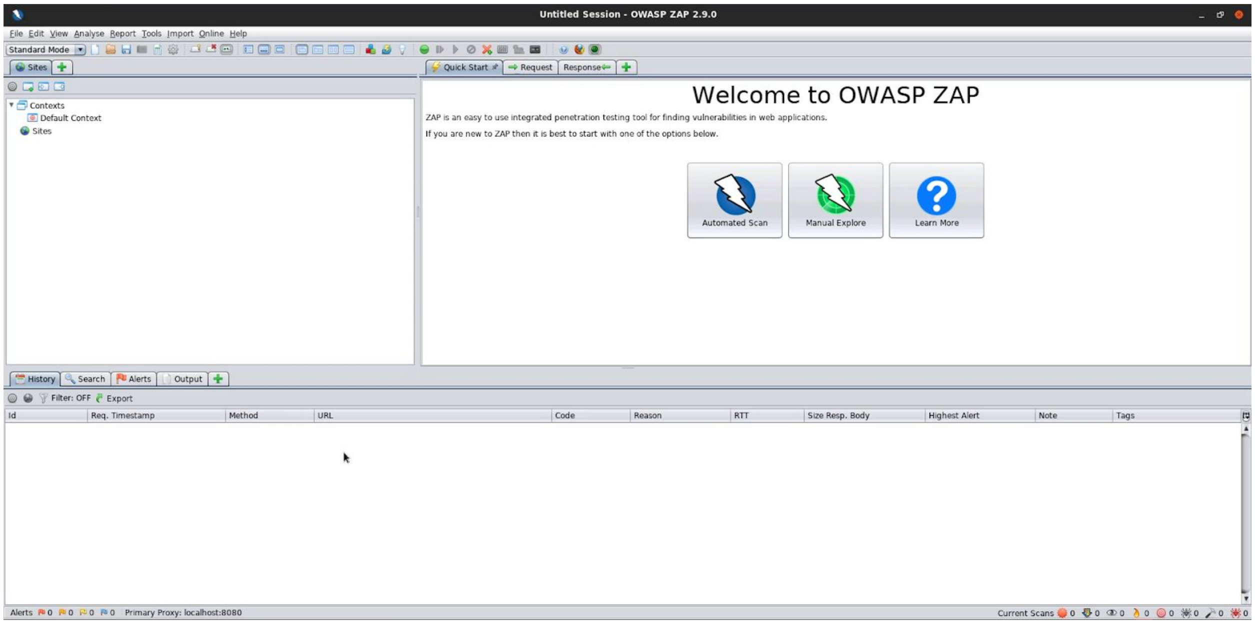
```
root@attackdefense:~# nmap 192.210.141.3
Starting Nmap 7.70 ( https://nmap.org ) at 2020-05-21 07:03 IST
Nmap scan report for target-1 (192.210.141.3)
Host is up (0.000013s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
80/tcp   open  http
3306/tcp open  mysql
MAC Address: 02:42:C0:D2:8D:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
root@attackdefense:~#
```

Port 80 and 3306 are open.

Step 3: Starting Burp Suite. Click on the Menu, Navigate to "Web Application Analysis" and click on "owasp-zap".



ZAP:

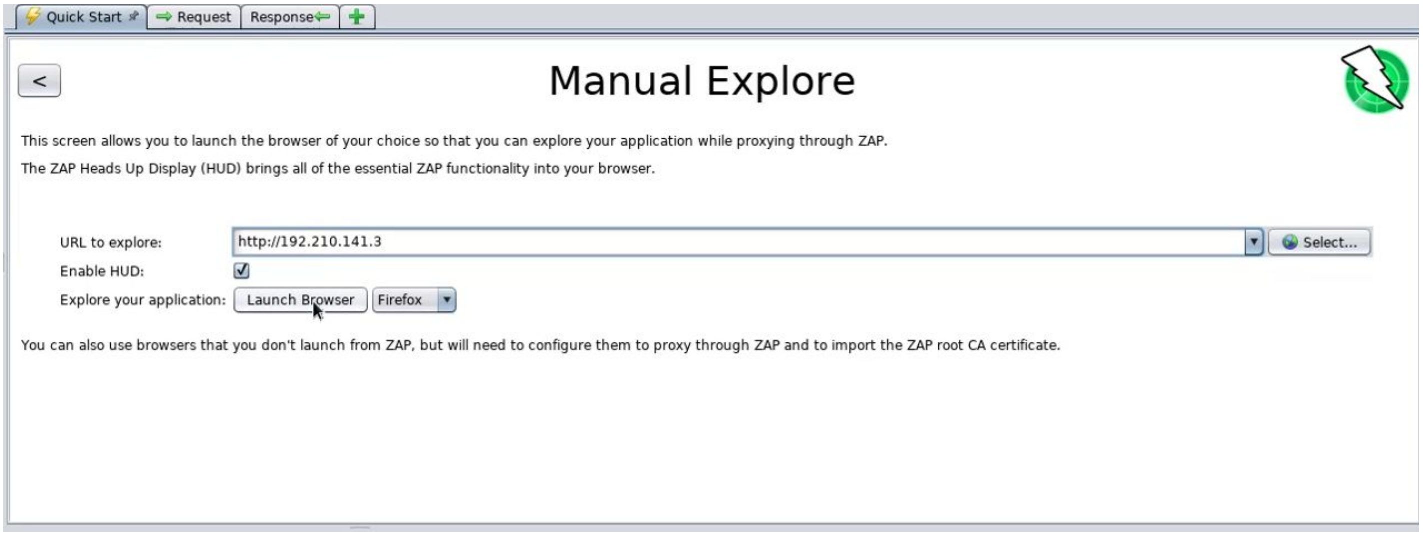


Welcome to OWASP ZAP

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

If you are new to ZAP then it is best to start with one of the options below.

Step 4: Click on "Manual Explore", enter the target IP address in the Input field and click on "Launch Browser".



Manual Explore

This screen allows you to launch the browser of your choice so that you can explore your application while proxying through ZAP.

The ZAP Heads Up Display (HUD) brings all of the essential ZAP functionality into your browser.

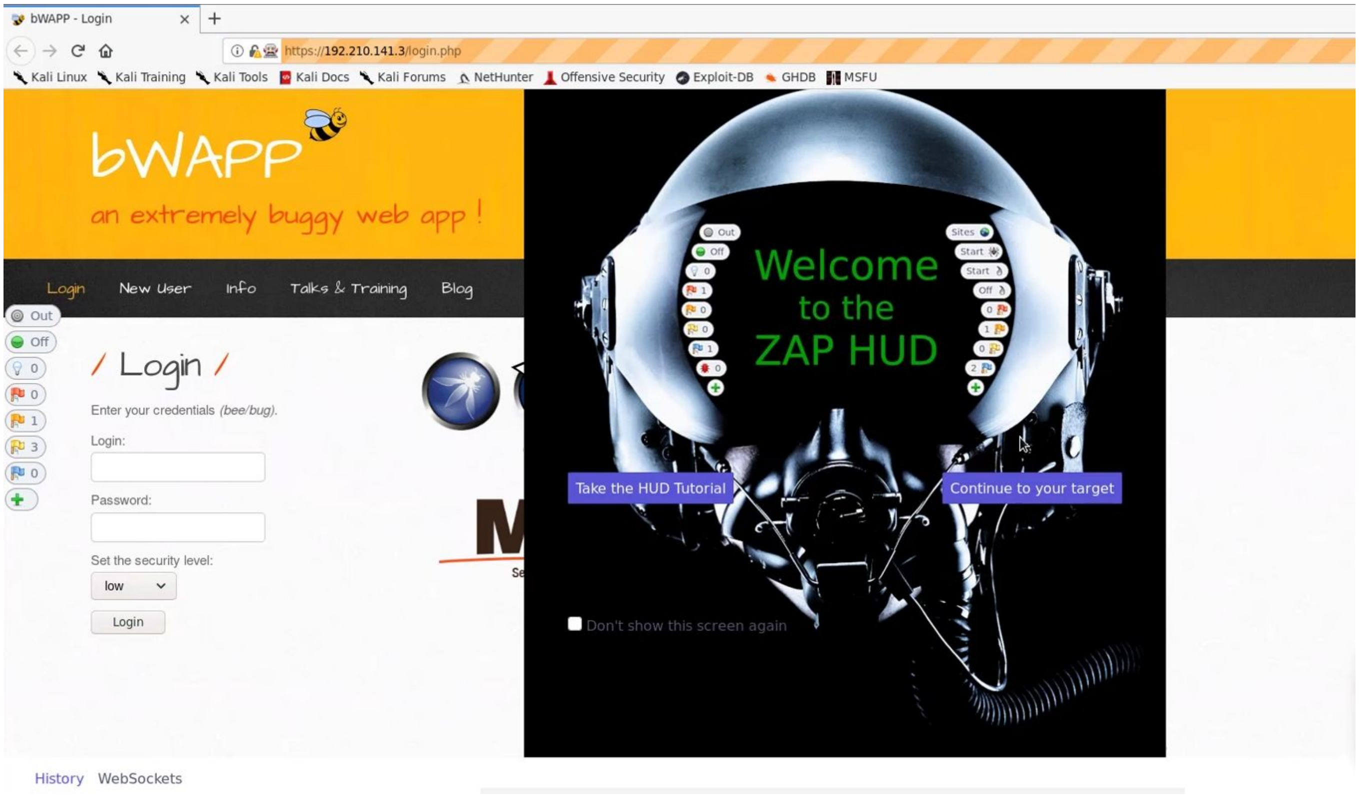
URL to explore: http://192.210.141.3

Enable HUD: ☑

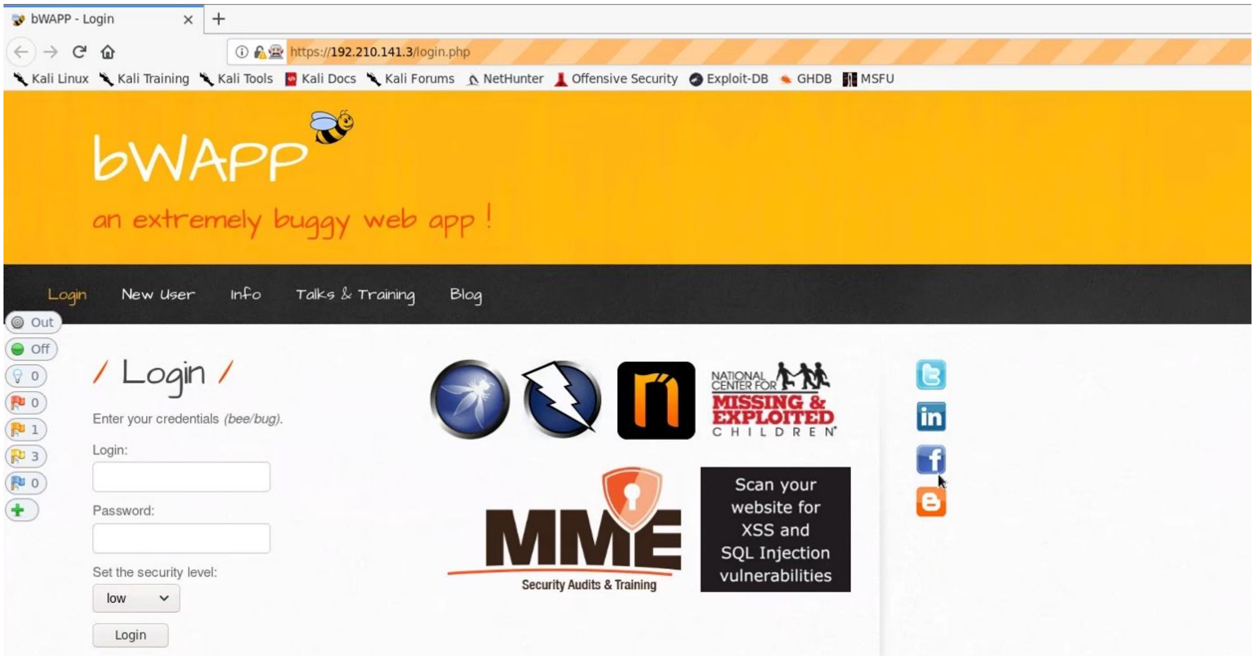
Explore your application: Launch Browser Firefox

You can also use browsers that you don't launch from ZAP, but will need to configure them to proxy through ZAP and to import the ZAP root CA certificate.

A browser session will be started with ZAP HUD.

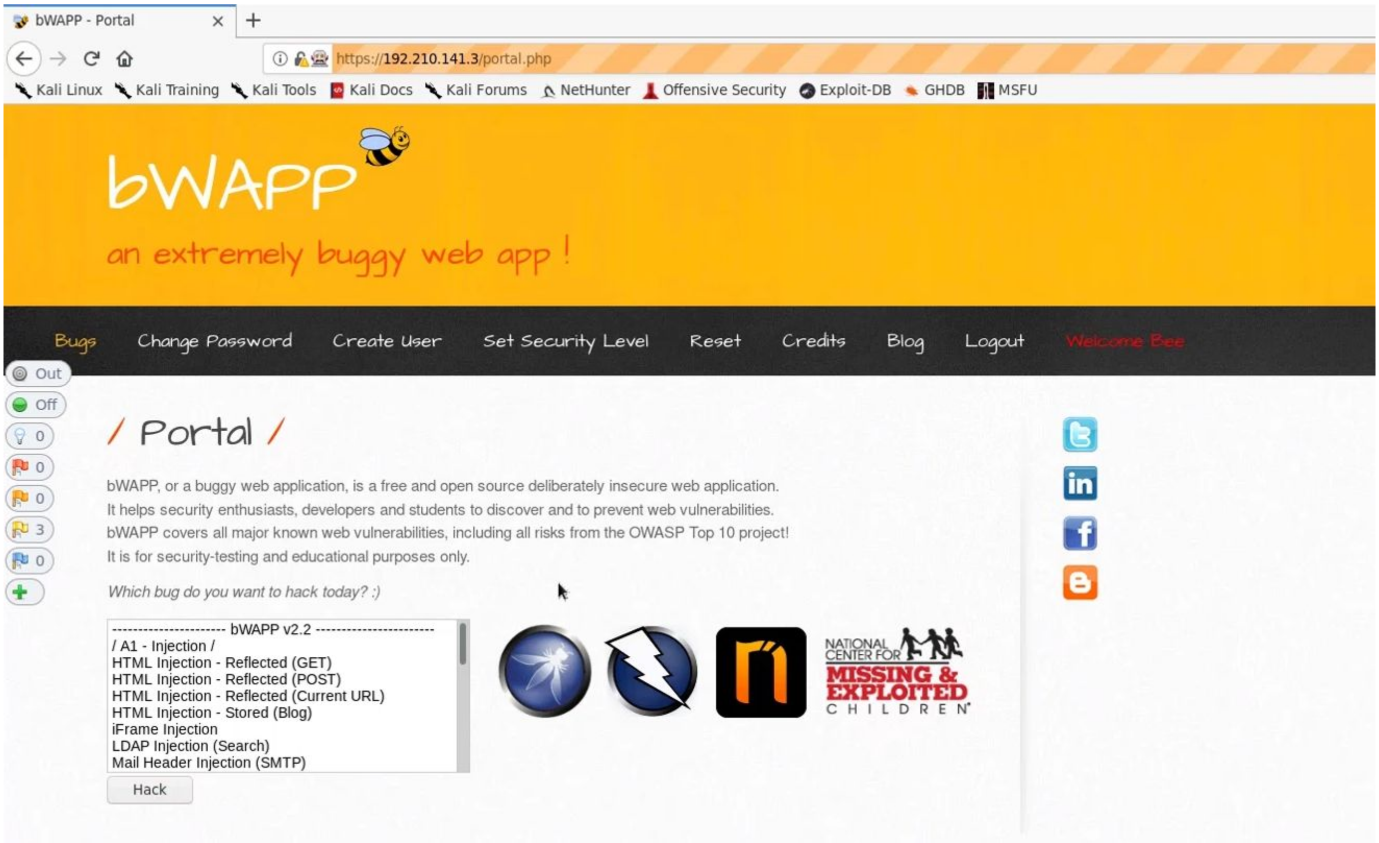


Step 5: Click on "Continue to your target".


bWAPP - Login
https://192.210.141.3/login.php
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHDB
MSFU
bWAPP
an extremely buggy web app !
Login
New User
Info
Talks & Training
Blog
/ Login /
Enter your credentials (bee/bug).
Login:
Password:
Set the security level:
low
Login
NATIONAL CENTER FOR MISSING & EXPLOITED CHILDREN
MME
Security Audits & Training
Scan your website for XSS and SQL Injection vulnerabilities

Step 6: Login to the web application, the login credentials are mentioned on the login page.

Username: bee
Password: bug

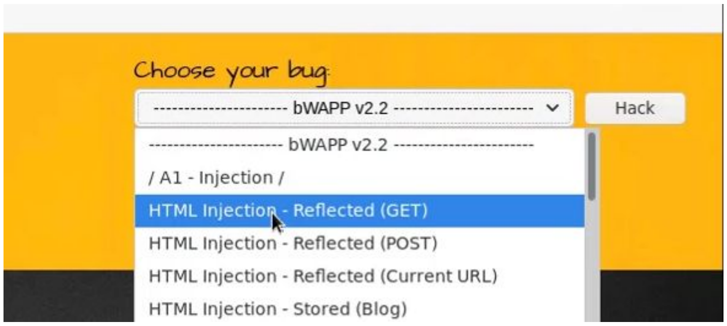


bWAPP, or a buggy web application, is a free and open source deliberately insecure web application.
It helps security enthusiasts, developers and students to discover and to prevent web vulnerabilities.
bWAPP covers all major known web vulnerabilities, including all risks from the OWASP Top 10 project!
It is for security-testing and educational purposes only.

Which bug do you want to hack today? :)

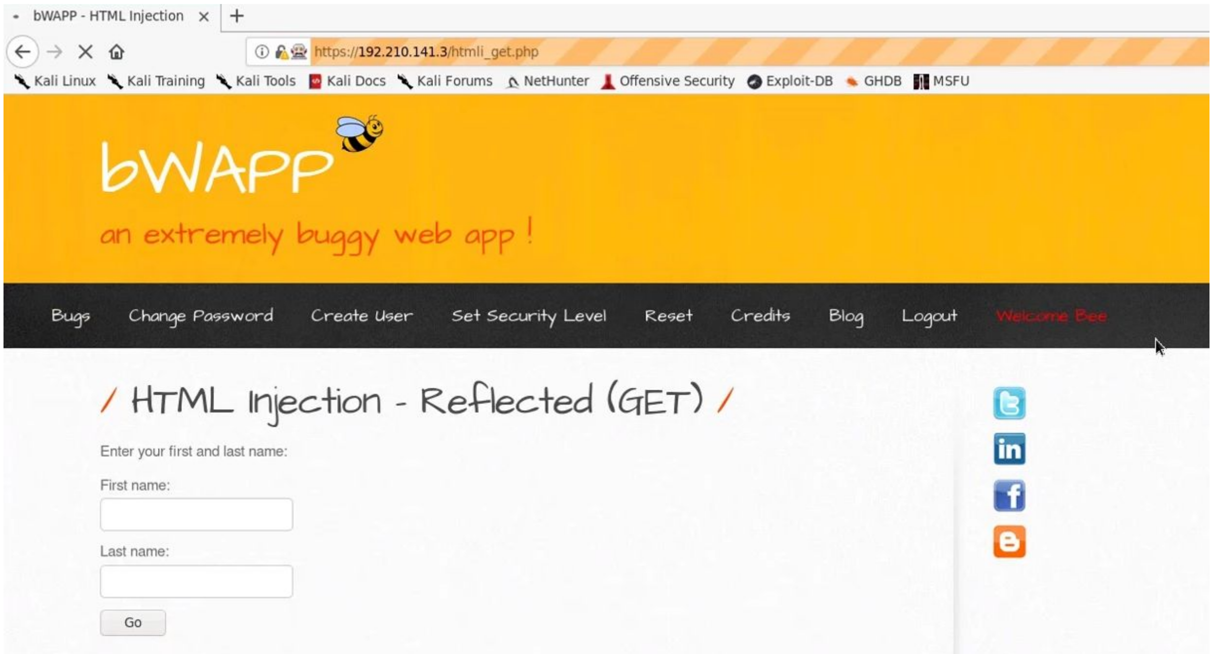
```
---------------------- bWAPP v2.2 ----------------------
/ A1 - Injection /
HTML Injection - Reflected (GET)
HTML Injection - Reflected (POST)
HTML Injection - Reflected (Current URL)
HTML Injection - Stored (Blog)
iFrame Injection
LDAP Injection (Search)
Mail Header Injection (SMTP)
```

Step 7: Access various web pages. From the Choose your bug dropdown, select "HTML Injection - Reflected (GET)" and click on the Hack button.

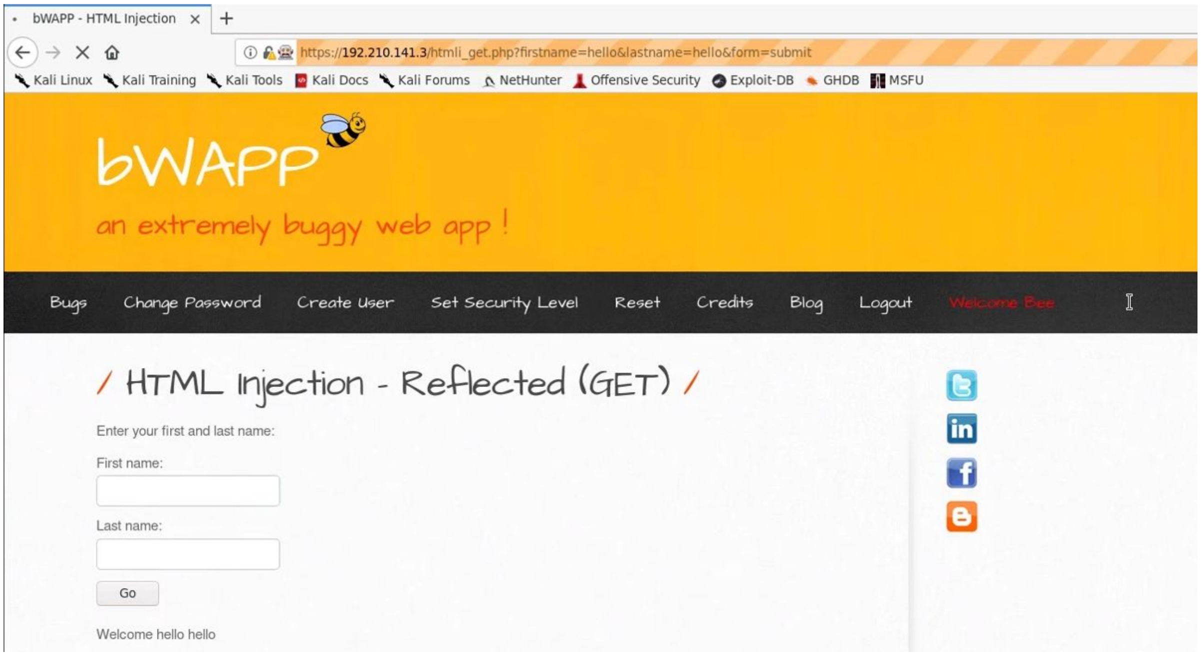


```
---------------------- bWAPP v2.2 ----------------------
/ A1 - Injection /
HTML Injection - Reflected (GET)
HTML Injection - Reflected (POST)
HTML Injection - Reflected (Current URL)
HTML Injection - Stored (Blog)
```

HTML Injection - Reflected (GET):



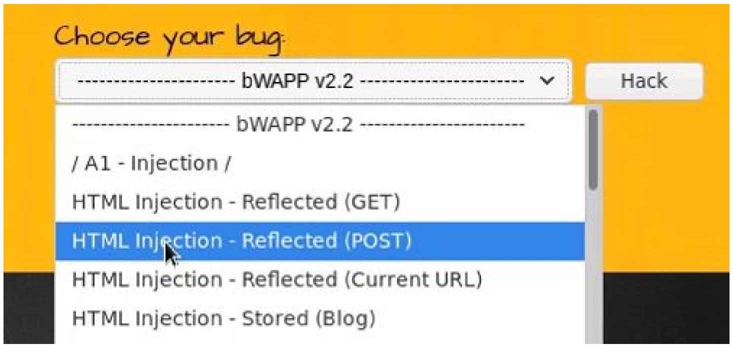
Step 8: Enter any values in the input field and click Go



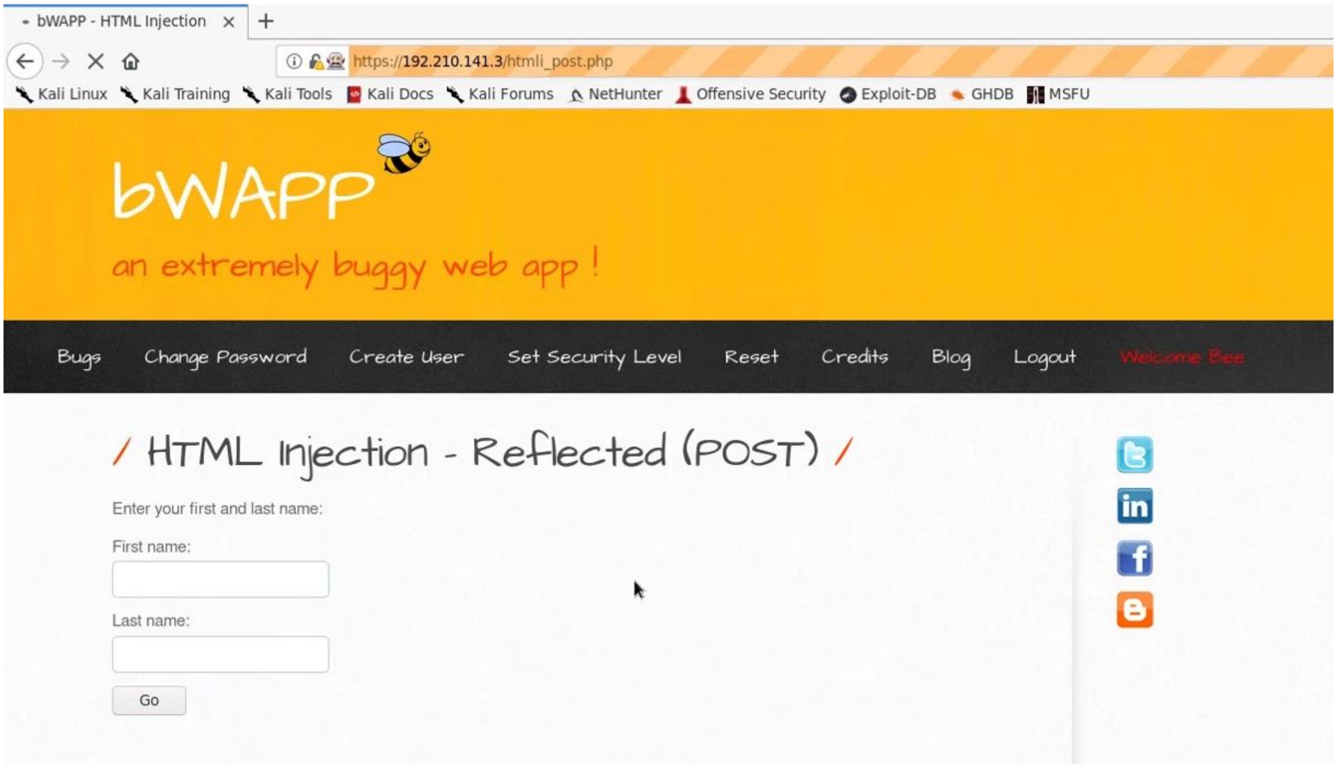
Welcome hello hello

The entered input will appear below the Go button

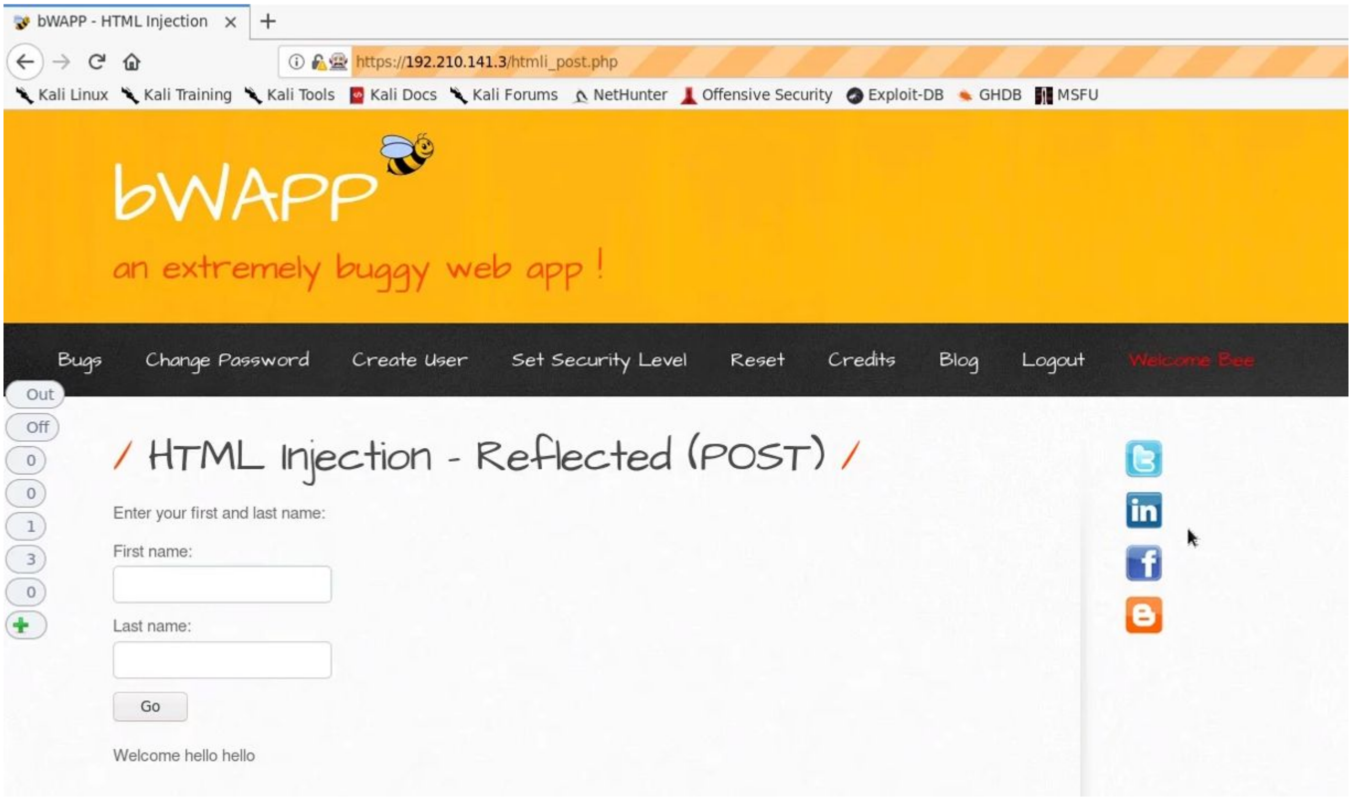
Step 9: From the Choose your bug dropdown, Select "HTML Injection - Reflected (POST)" and click on the Hack button.


Choose your bug:
--------------------- bWAPP v2.2 ----------------------
Hack
--------------------- bWAPP v2.2 ----------------------
/ A1 - Injection /
HTML Injection - Reflected (GET)
HTML Injection - Reflected (POST)
HTML Injection - Reflected (Current URL)
HTML Injection - Stored (Blog)

HTML Injection - Reflected (POST)



Step 10: Enter any values in the input field and click Go



/ HTML Injection - Reflected (POST) /

Enter your first and last name:

First name:

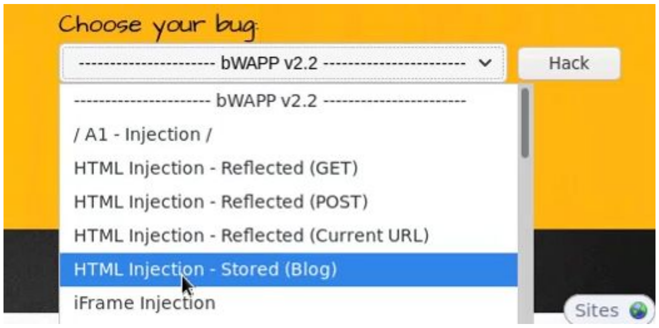
Last name:

Go

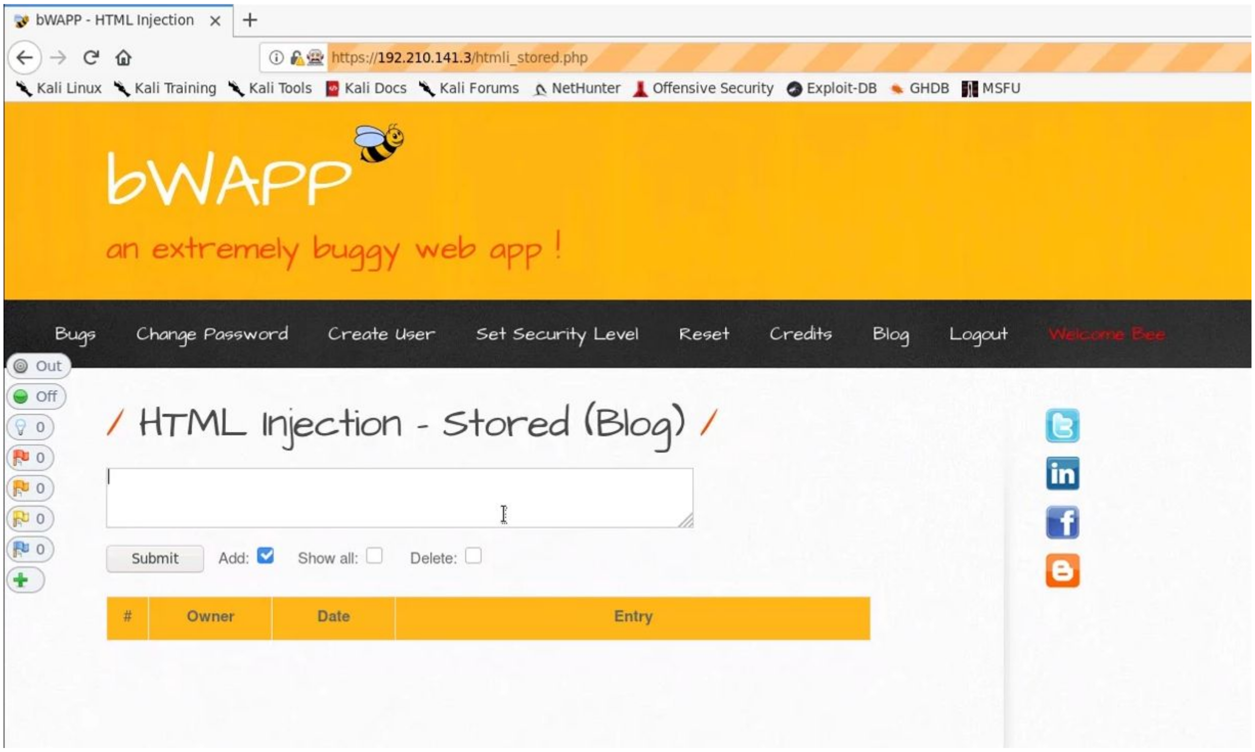
Welcome hello hello

The entered input will appear below the Go button

Step 11: From the Choose your bug dropdown, Select "HTML Injection - Stored (Blog)" and click on the Hack button.


Choose your bug:
---------------------- bWAPP v2.2 ----------------------
Hack
---------------------- bWAPP v2.2 ----------------------
/ A1 - Injection /
HTML Injection - Reflected (GET)
HTML Injection - Reflected (POST)
HTML Injection - Reflected (Current URL)
HTML Injection - Stored (Blog)
iFrame Injection
Sites

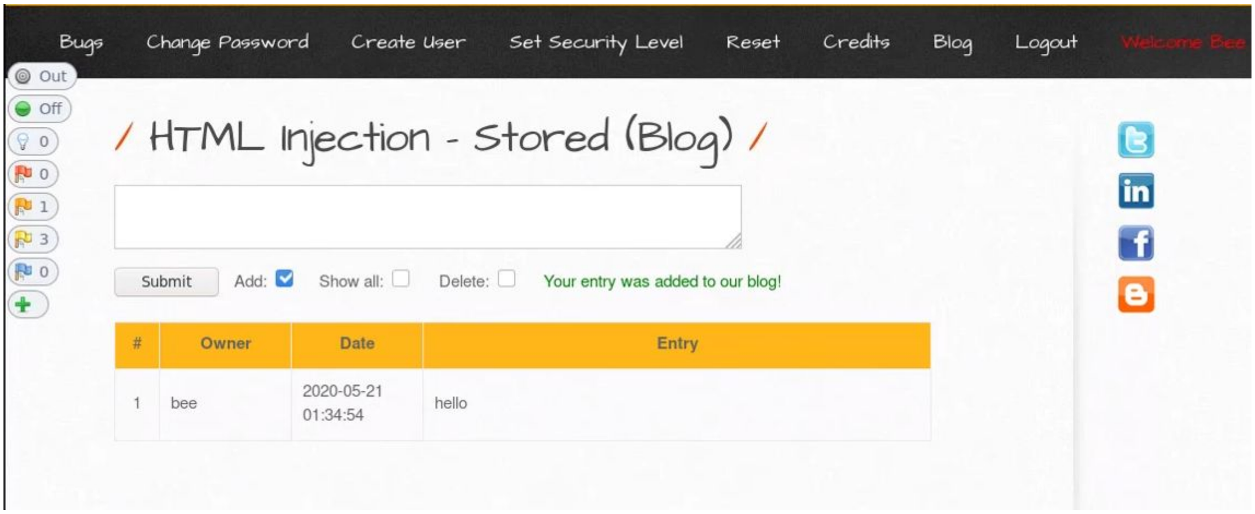
Step 12: Enter any values in the input field and click Submit.



Add: ☑ Show all: □ Delete: □

#	Owner	Date	Entry

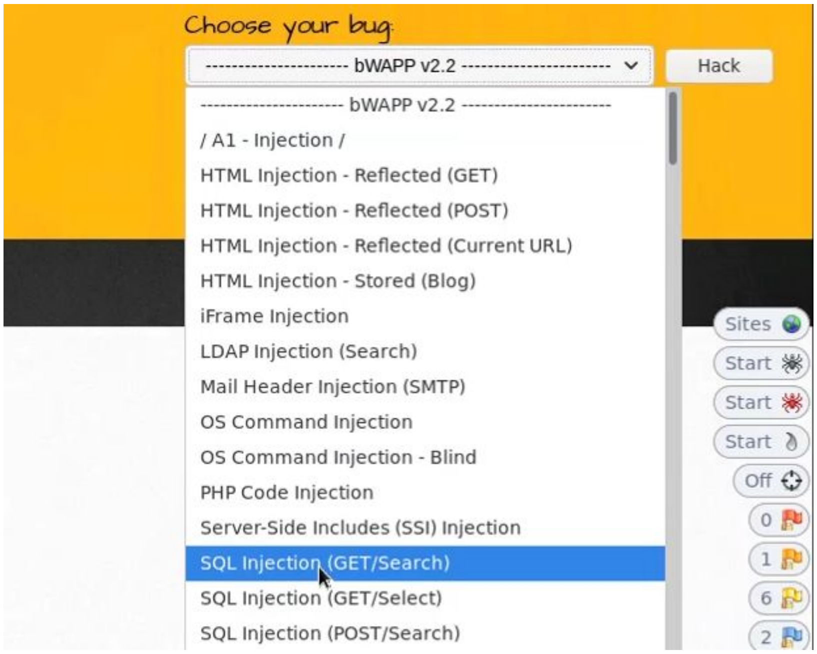
The entered value will appear in the table.



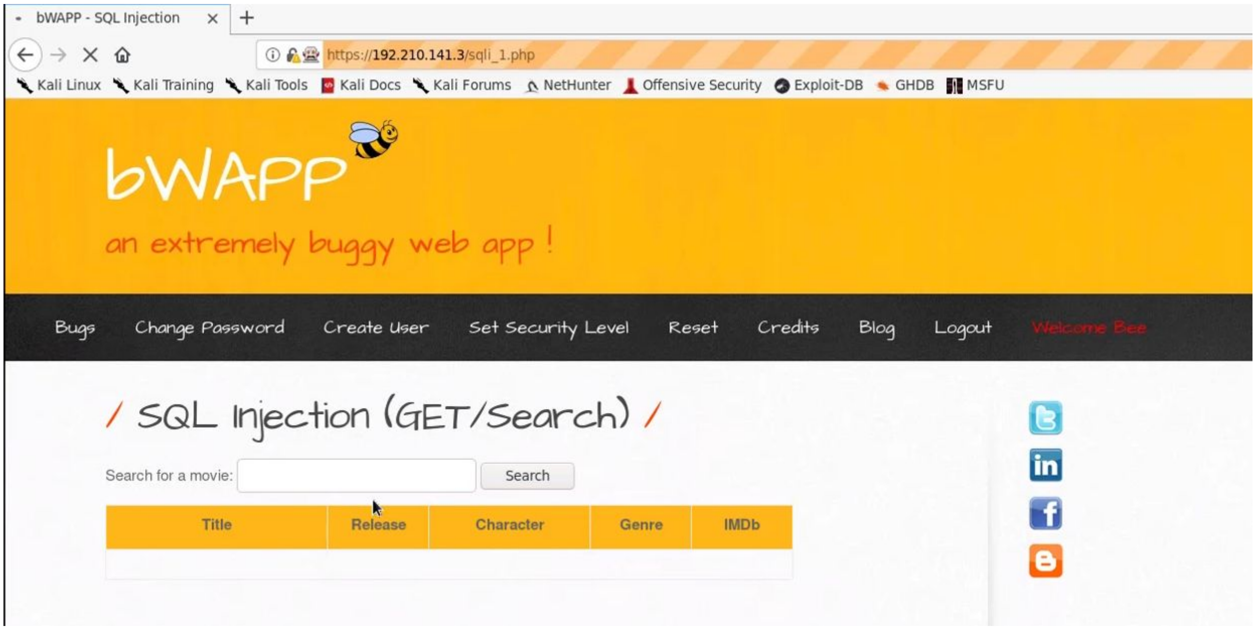
Add: ☑ Show all: □ Delete: □ Your entry was added to our blog!

#	Owner	Date	Entry
1	bee	2020-05-21 01:34:54	hello

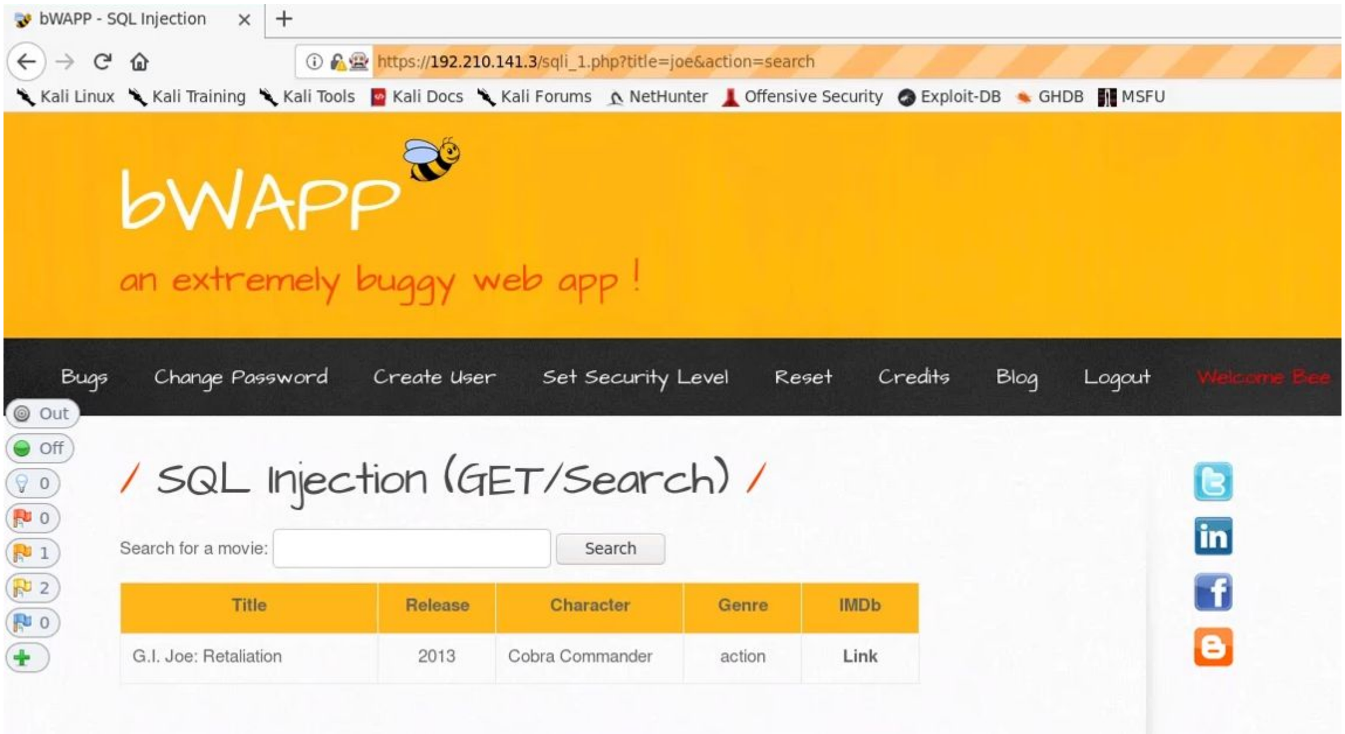
Step 13: From the Choose your bug dropdown, Select "SQL Injection (GET/Search)" and click on the Hack button.


Choose your bug:
-------------------- bWAPP v2.2 ---------------------
Hack
-------------------- bWAPP v2.2 ---------------------
/ A1 - Injection /
HTML Injection - Reflected (GET)
HTML Injection - Reflected (POST)
HTML Injection - Reflected (Current URL)
HTML Injection - Stored (Blog)
iFrame Injection
LDAP Injection (Search)
Mail Header Injection (SMTP)
OS Command Injection
OS Command Injection - Blind
PHP Code Injection
Server-Side Includes (SSI) Injection
SQL Injection (GET/Search)
SQL Injection (GET/Select)
SQL Injection (POST/Search)
Sites
Start
Start
Start
Off
0
1
6
2

SQL Injection (GET/Search)



Step 14: Enter "Joe" and click on the Search button.



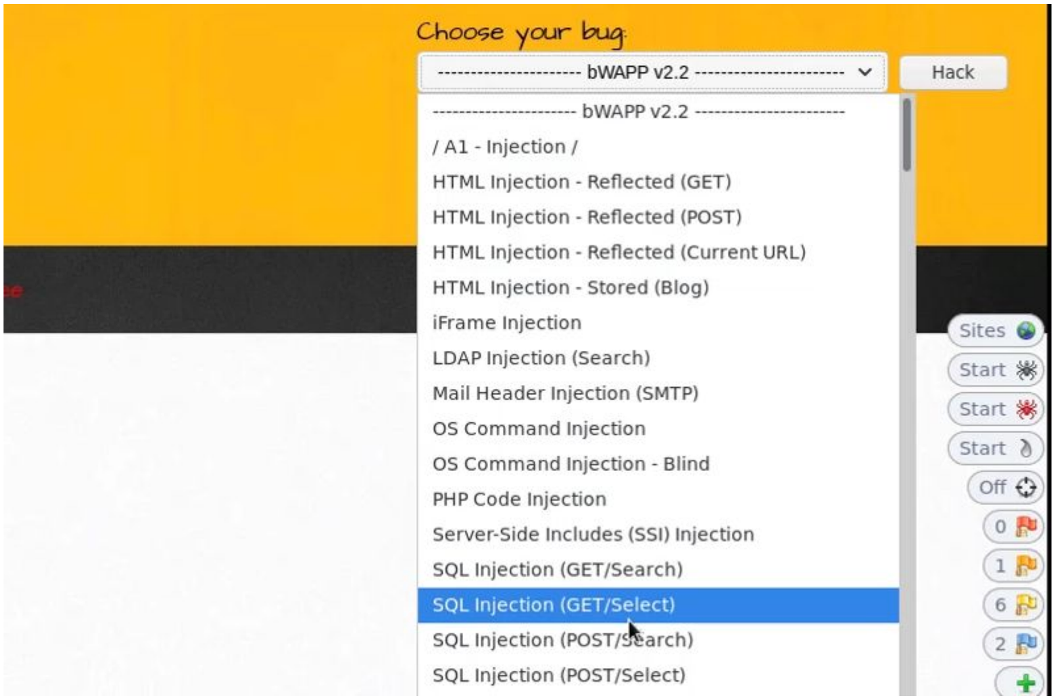
/ SQL Injection (GET/Search) /

Search for a movie: [ ] Search

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link

1 result will appear.

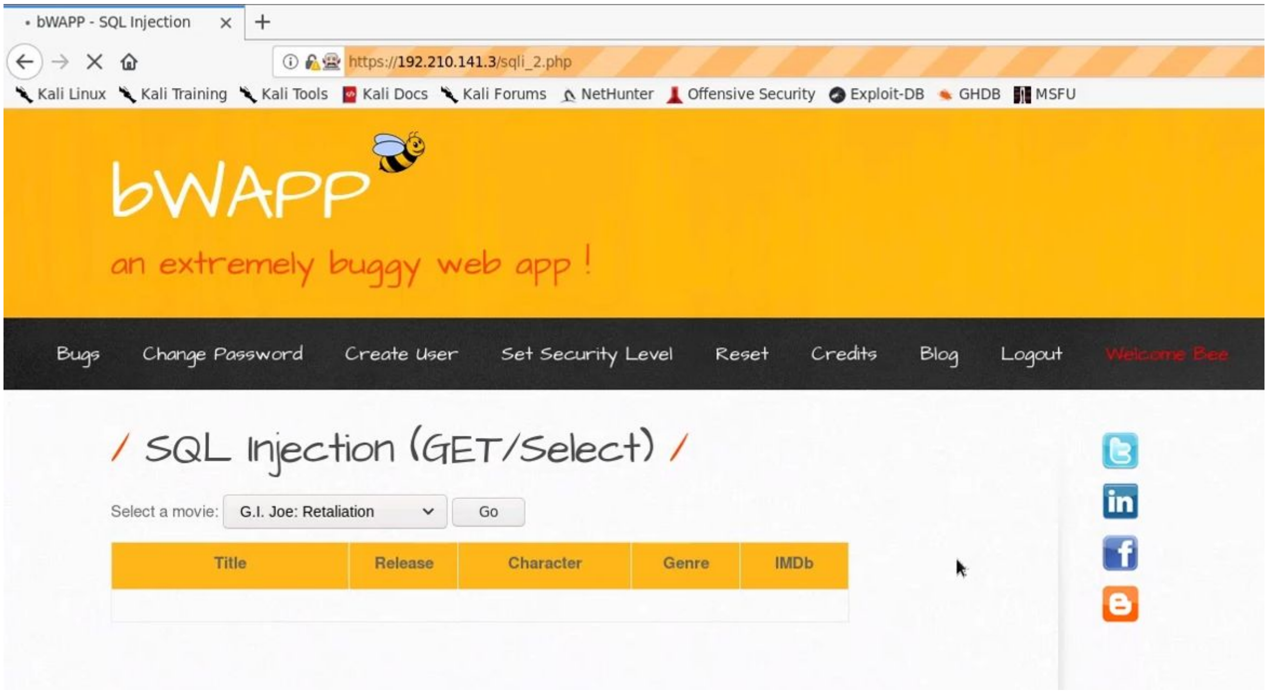
Step 15: From the Choose your bug dropdown, Select "SQL Injection (GET/Select)" and click on the Hack button.


Choose your bug:

-------------------- bWAPP v2.2 --------------------

- / A1 - Injection /
- HTML Injection - Reflected (GET)
- HTML Injection - Reflected (POST)
- HTML Injection - Reflected (Current URL)
- HTML Injection - Stored (Blog)
- iFrame Injection
- LDAP Injection (Search)
- Mail Header Injection (SMTP)
- OS Command Injection
- OS Command Injection - Blind
- PHP Code Injection
- Server-Side Includes (SSI) Injection
- SQL Injection (GET/Search)
- SQL Injection (GET/Select)
- SQL Injection (POST/Search)
- SQL Injection (POST/Select)

SQL Injection (GET/Select)

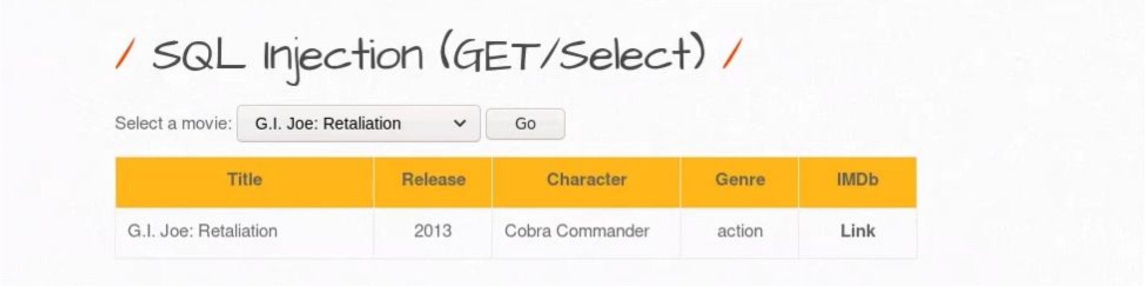


/ SQL Injection (GET/Select) /

Select a movie: G.I. Joe: Retaliation Go

Title	Release	Character	Genre	IMDb

Step 16: "G.I. Joe: Retaliation" is the default selected option. Click on the "Select" button.



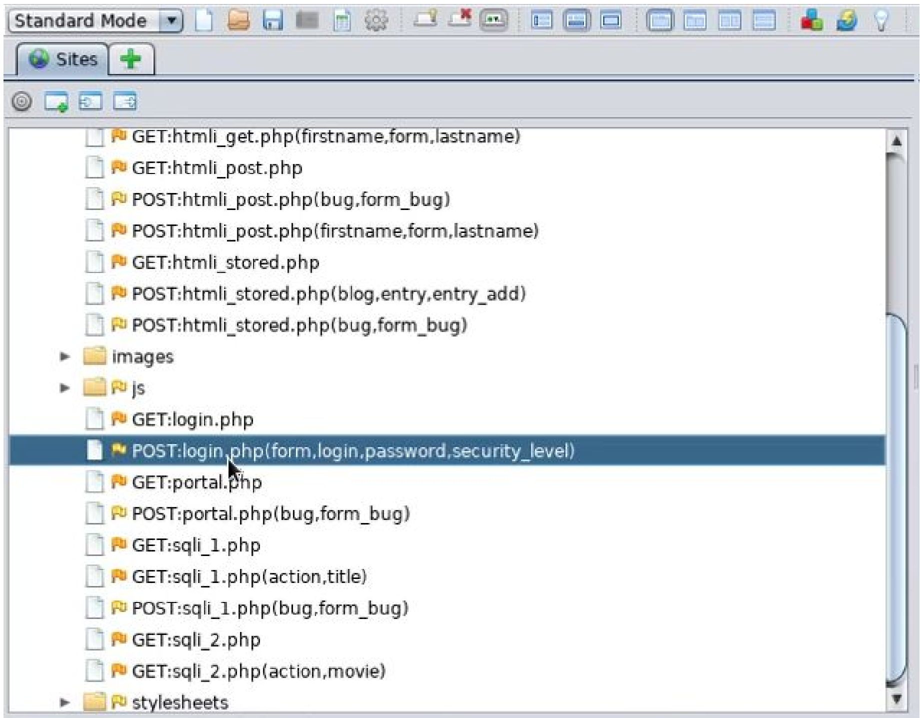
/ SQL Injection (GET/Select) /

Select a movie: G.I. Joe: Retaliation Go

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link

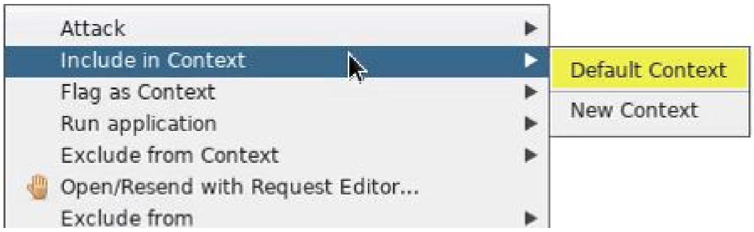
1 result will appear.

Step 17: Configuring ZAProxy to use authenticated session. In ZAProxy, navigate to the sitemap and find the login request.

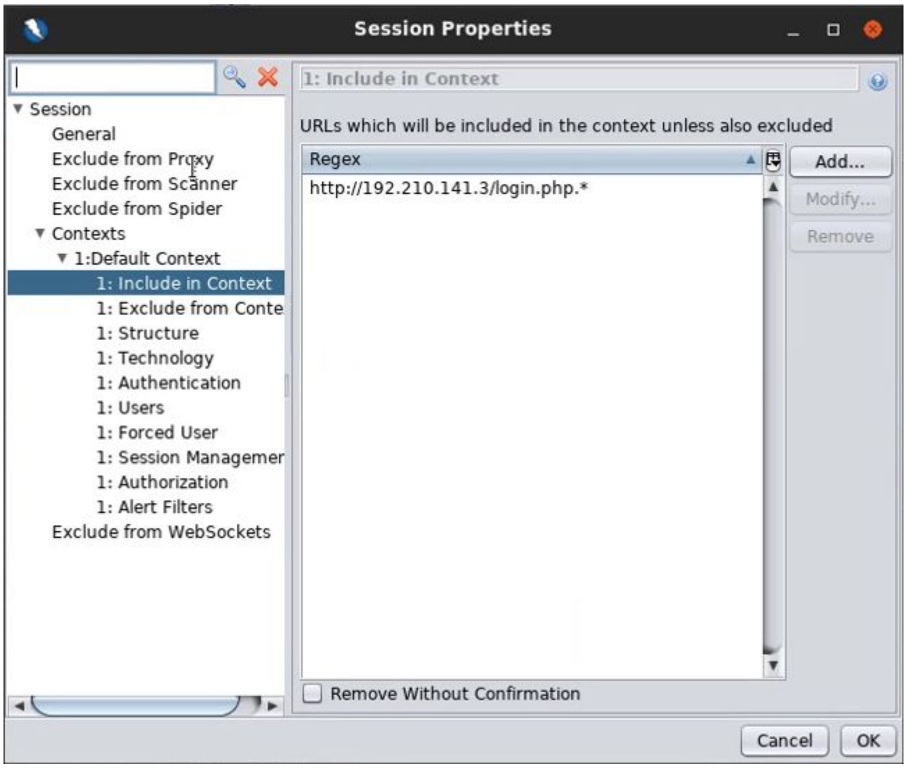


```
GET:htmli_get.php(firstname,form,lastname)
GET:htmli_post.php
POST:htmli_post.php(bug,form_bug)
POST:htmli_post.php(firstname,form,lastname)
GET:htmli_stored.php
POST:htmli_stored.php(blog,entry,entry_add)
POST:htmli_stored.php(bug,form_bug)
images
js
GET:login.php
POST:login.php(form,login,password,security_level)
GET:portal.php
POST:portal.php(bug,form_bug)
GET:sqli_1.php
GET:sqli_1.php(action,title)
POST:sqli_1.php(bug,form_bug)
GET:sqli_2.php
GET:sqli_2.php(action,movie)
stylesheets
```

Step 18: Right click on the POST request, navigate to "Include in Context" and select on "Default Context".



The session Properties window will appear.

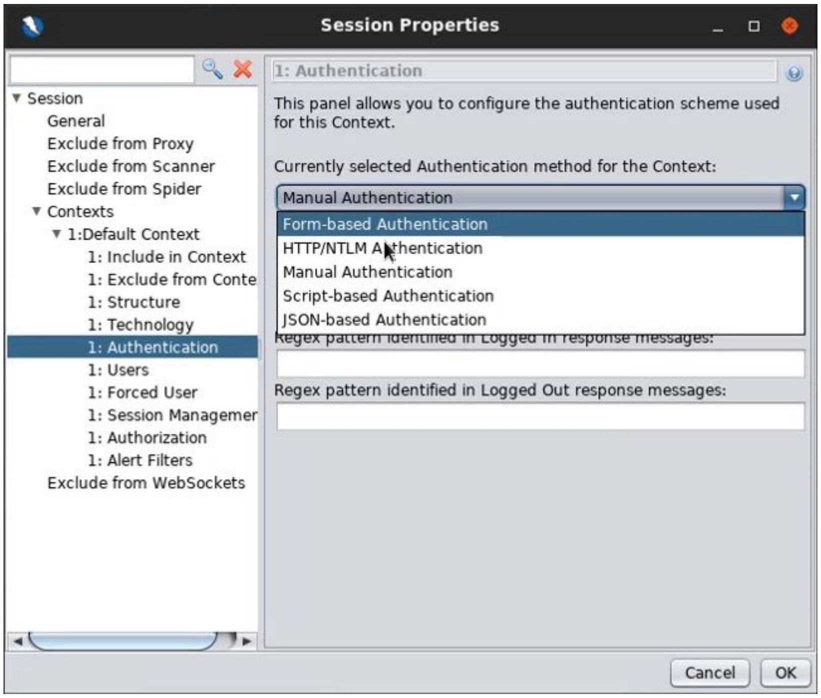


URLs which will be included in the context unless also excluded

```
http://192.210.141.3/login.php.*
```

☐ Remove Without Confirmation

Step 19: Click on the Authentication tab under Default Context menu and select "Form-based Authentication" for the selected method.



1: Authentication

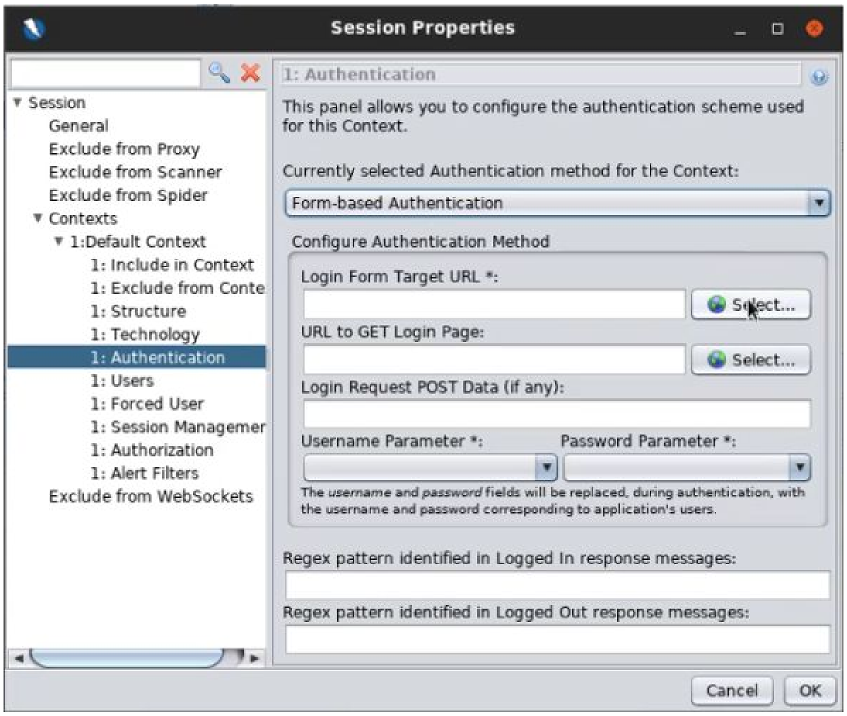
This panel allows you to configure the authentication scheme used for this Context.

Currently selected Authentication method for the Context: Manual Authentication

Regex pattern identified in Logged In response messages:

Regex pattern identified in Logged Out response messages:

Authentication Section:



1: Authentication

This panel allows you to configure the authentication scheme used for this Context.

Currently selected Authentication method for the Context: Form-based Authentication

Configure Authentication Method

Login Form Target URL *:

URL to GET Login Page:

Login Request POST Data (if any):

Username Parameter *:

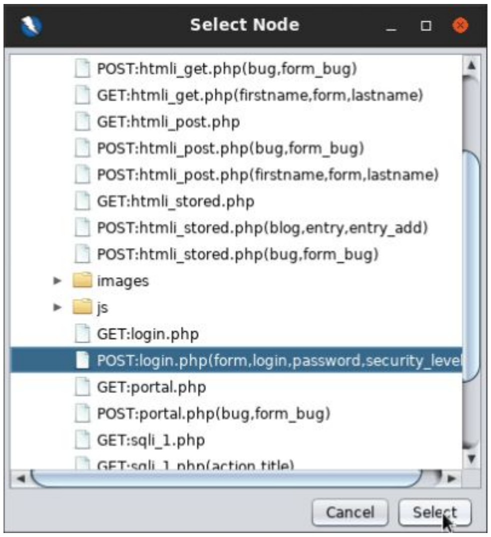
Password Parameter *:

The *username* and *password* fields will be replaced, during authentication, with the username and password corresponding to application's users.

Regex pattern identified in Logged In response messages:

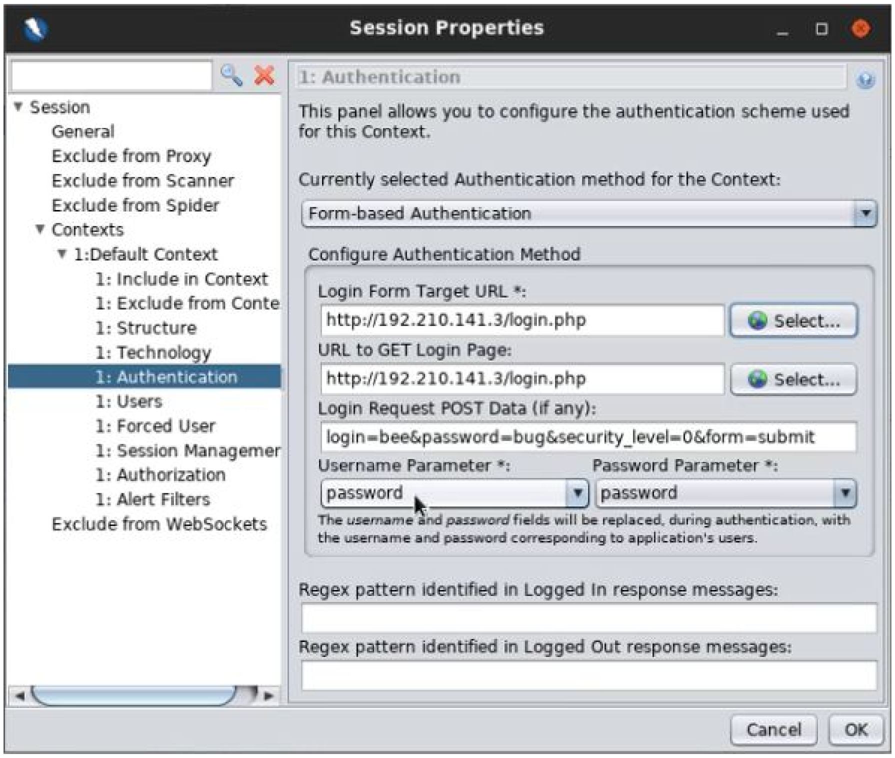
Regex pattern identified in Logged Out response messages:

Step 20: Click on the Select button and select the POST login request.



POST:htmli_get.php(bug,form_bug)
GET:htmli_get.php(firstname,form,lastname)
GET:htmli_post.php
POST:htmli_post.php(bug,form_bug)
POST:htmli_post.php(firstname,form,lastname)
GET:htmli_stored.php
POST:htmli_stored.php(blog,entry,entry_add)
POST:htmli_stored.php(bug,form_bug)
images
js
GET:login.php
POST:login.php(form,login,password,security_level
GET:portal.php
POST:portal.php(bug,form_bug)
GET:sqli_1.php
GET:sqli_1.php(action,title)

The form fields will automatically be filled.



1: Authentication

This panel allows you to configure the authentication scheme used for this Context.

Currently selected Authentication method for the Context: Form-based Authentication

Configure Authentication Method

Login Form Target URL *: http://192.210.141.3/login.php

URL to GET Login Page: http://192.210.141.3/login.php

Login Request POST Data (if any): login=bee&password=bug&security_level=0&form=submit

Username Parameter *: password

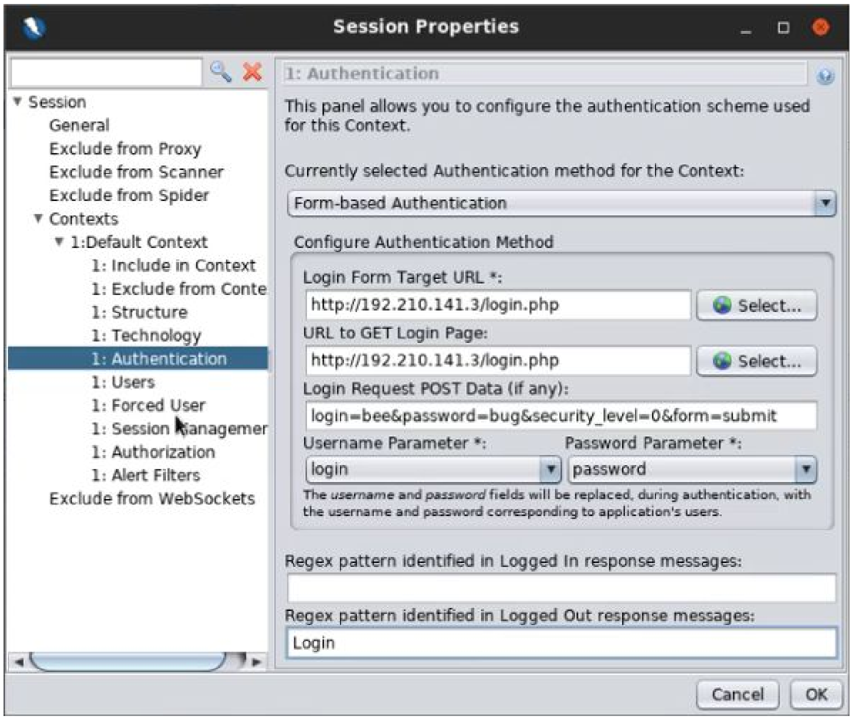
Password Parameter *: password

The *username* and *password* fields will be replaced, during authentication, with the username and password corresponding to application's users.

Regex pattern identified in Logged In response messages:

Regex pattern identified in Logged Out response messages:

Step 21: Set the Username parameter to "login" and Enter "Login" in the "Regex pattern identified in Logged Out response messages".



Session Properties

1: Authentication

This panel allows you to configure the authentication scheme used for this Context.

Currently selected Authentication method for the Context: Form-based Authentication

Configure Authentication Method

Login Form Target URL *: http://192.210.141.3/login.php

URL to GET Login Page: http://192.210.141.3/login.php

Login Request POST Data (if any): login=bee&password=bug&security_level=0&form=submit

Username Parameter *: login

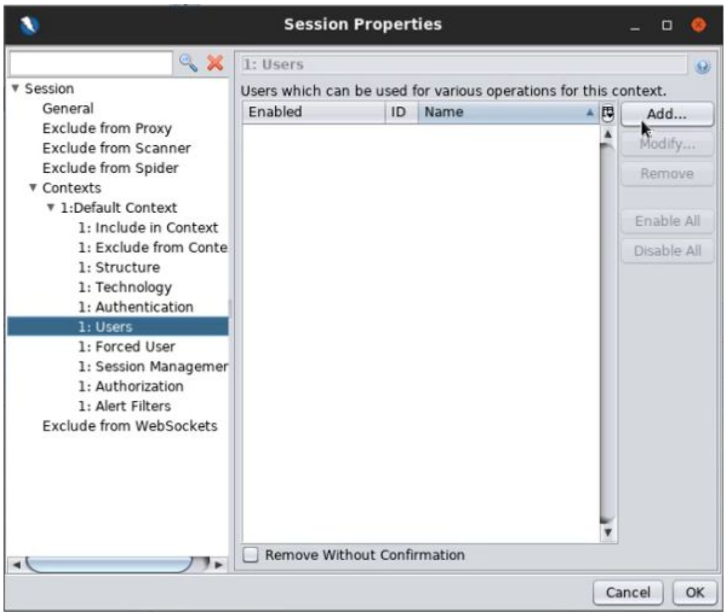
Password Parameter *: password

The *username* and *password* fields will be replaced, during authentication, with the username and password corresponding to application's users.

Regex pattern identified in Logged In response messages:

Regex pattern identified in Logged Out response messages: Login

Step 22: Click on the Users tab.



Session Properties

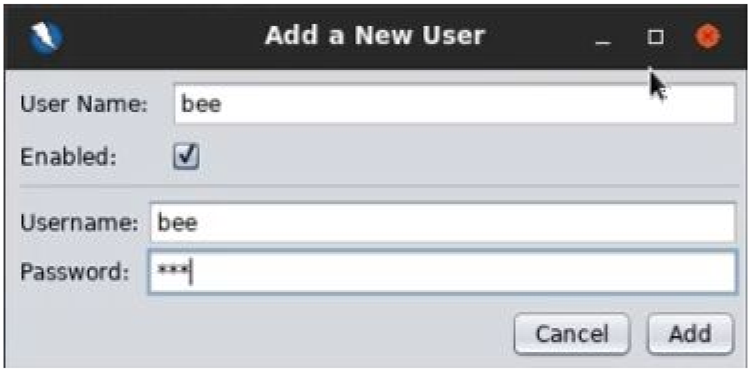
1: Users

Users which can be used for various operations for this context.

Enabled	ID	Name

☐ Remove Without Confirmation

Step 23: Click on the "Add" button and add a new user with username "bee" and password "bug"



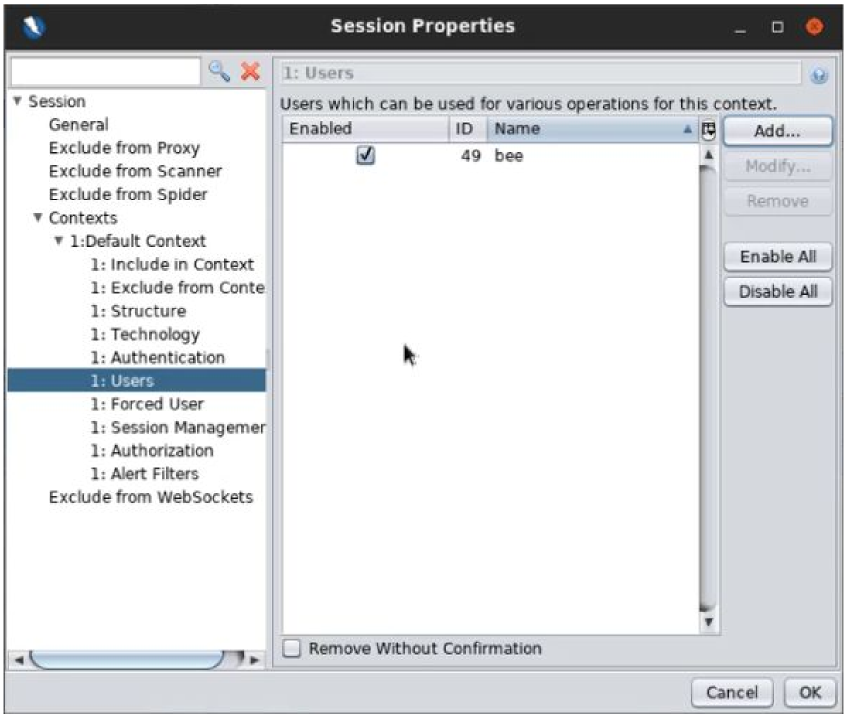
User Name: bee

Enabled: ☑

Username: bee

Password: ***

Step 24: Click on the "OK" button.

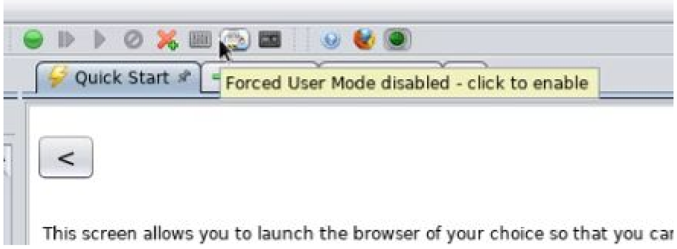


Users which can be used for various operations for this context.

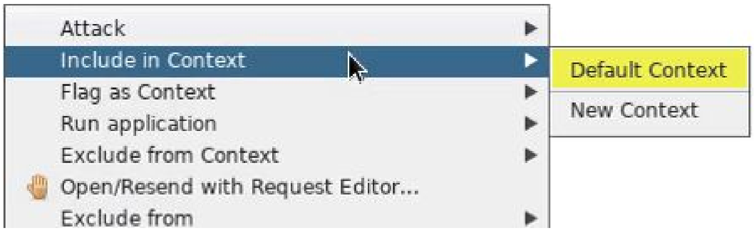
Enabled	ID	Name
☑	49	bee

☐ Remove Without Confirmation

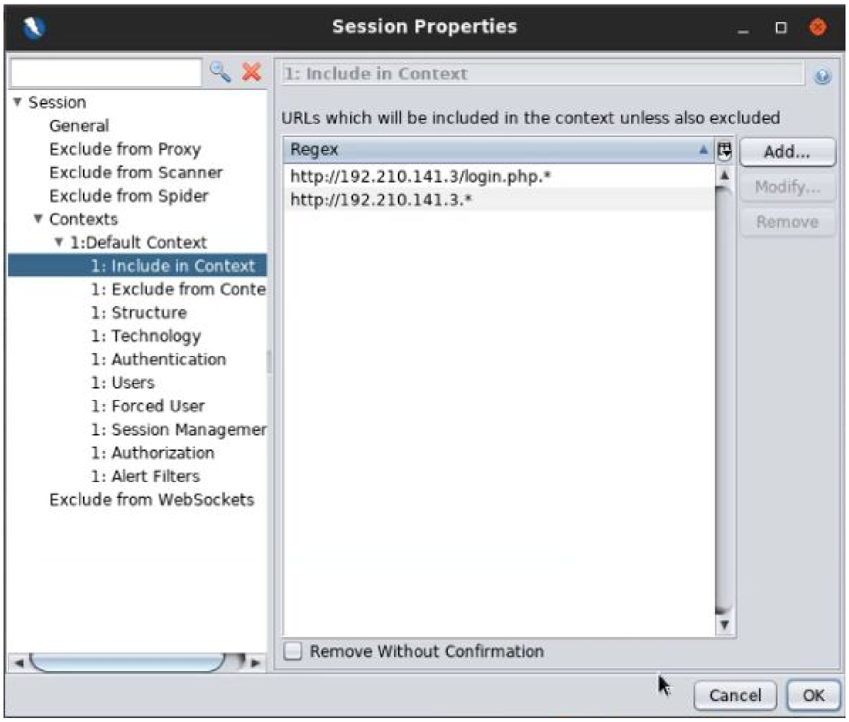
Step 25: Click on the User lock icon.


Quick Start
Forced User Mode disabled - click to enable
<
This screen allows you to launch the browser of your choice so that you car

Step 26: Right click on the Site (http://192.210.141.3), navigate to "Include in Context" and select on "Default Context".



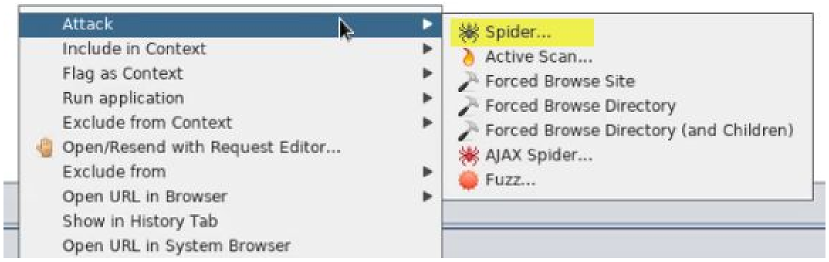
Session Properties window will appear.



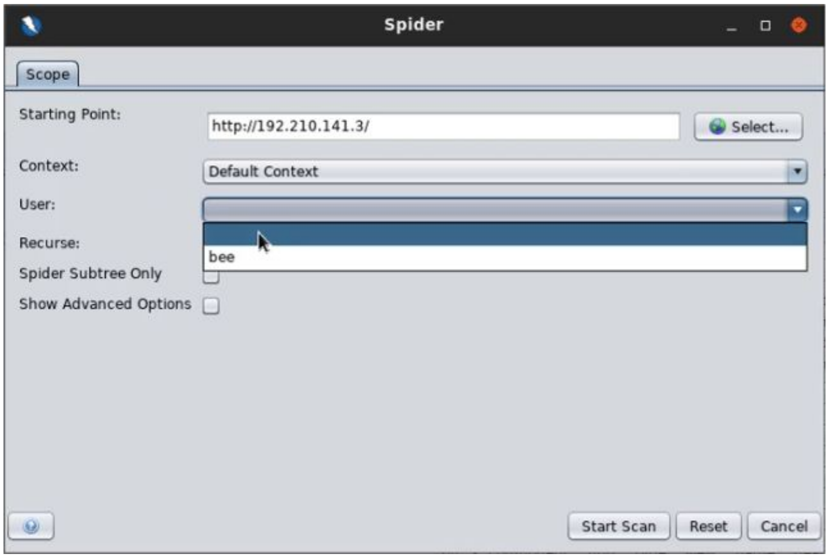
URLs which will be included in the context unless also excluded

Regex
http://192.210.141.3/login.php.*
http://192.210.141.3.*

Step 27: Click on the "OK" button. Right click on the Site (http://192.210.141.3), navigate to Attack and select "Spider".



Step 28: A dialog box will appear, select the "bee" user and click on "Start Scan" button.



Starting Point: http://192.210.141.3/

Context: Default Context

User:

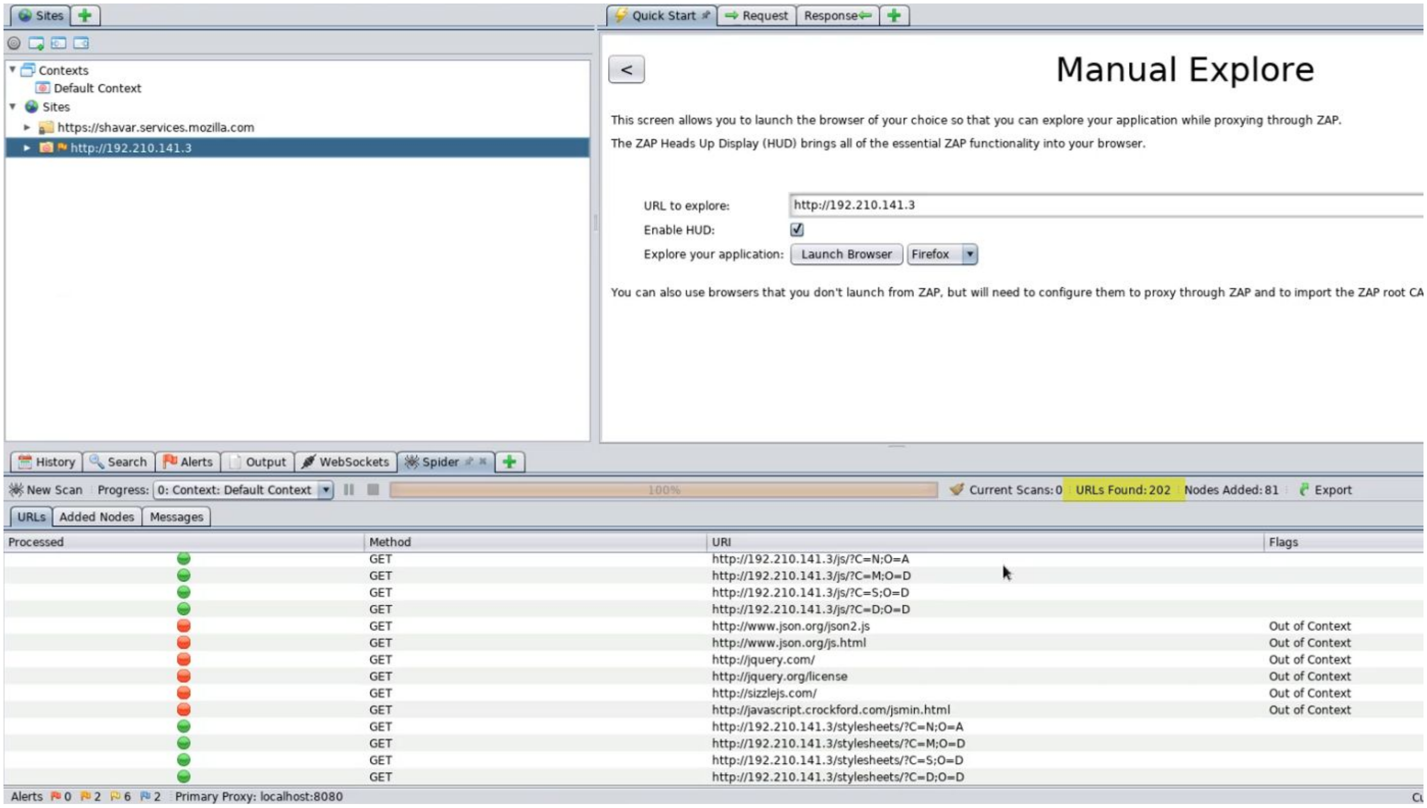
bee

Recurse:

Spider Subtree Only

Show Advanced Options □

Scan Result:



Manual Explore

This screen allows you to launch the browser of your choice so that you can explore your application while proxying through ZAP.
The ZAP Heads Up Display (HUD) brings all of the essential ZAP functionality into your browser.

URL to explore: http://192.210.141.3
Enable HUD: ☑
Explore your application: Launch Browser | Firefox

You can also use browsers that you don't launch from ZAP, but will need to configure them to proxy through ZAP and to import the ZAP root CA

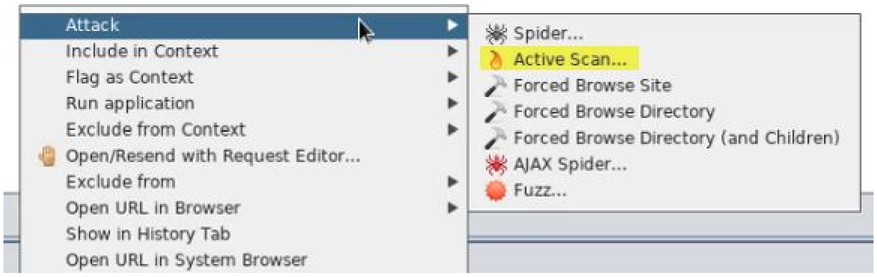
Progress: 0: Context: Default Context 100% Current Scans: 0 URLs Found: 202 Nodes Added: 81

Processed	Method	URI	Flags
🟢	GET	http://192.210.141.3/js/?C=N;O=A	
🟢	GET	http://192.210.141.3/js/?C=M;O=D	
🟢	GET	http://192.210.141.3/js/?C=S;O=D	
🟢	GET	http://192.210.141.3/js/?C=D;O=D	
🔴	GET	http://www.json.org/json2.js	Out of Context
🔴	GET	http://www.json.org/js.html	Out of Context
🔴	GET	http://jquery.com/	Out of Context
🔴	GET	http://jquery.org/license	Out of Context
🔴	GET	http://sizzlejs.com/	Out of Context
🔴	GET	http://javascript.crockford.com/jsmin.html	Out of Context
🟢	GET	http://192.210.141.3/stylesheets/?C=N;O=A	
🟢	GET	http://192.210.141.3/stylesheets/?C=M;O=D	
🟢	GET	http://192.210.141.3/stylesheets/?C=S;O=D	
🟢	GET	http://192.210.141.3/stylesheets/?C=D;O=D	

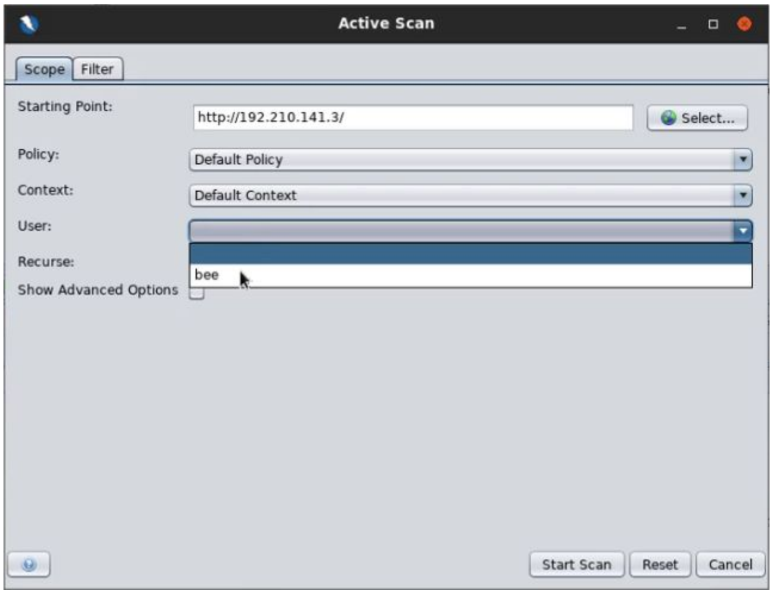
Alerts 0 2 6 2 Primary Proxy: localhost:8080

202 URLs were found.

Step 29: Right click on the Site (http://192.210.141.3), navigate to Attack and select "Active Scan".



Step 30: A dialog box will appear, select the "bee" user and click on "Start Scan" button.



Starting Point: http://192.210.141.3/

Policy: Default Policy

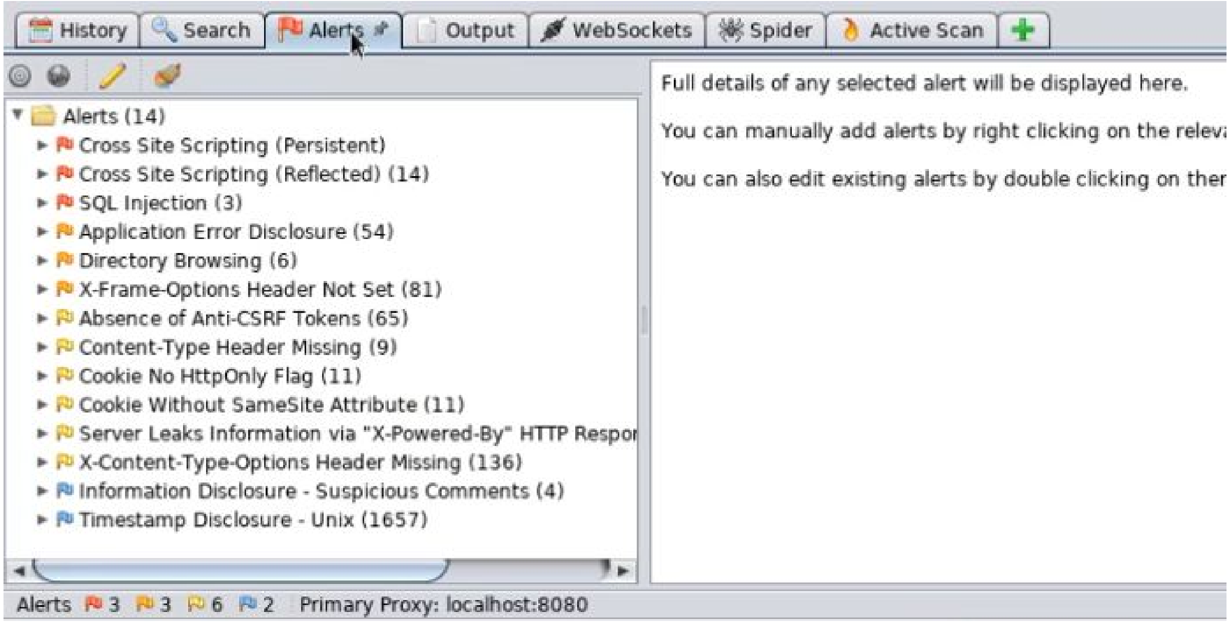
Context: Default Context

User: bee

Recurse:

Show Advanced Options □

Step 31: After the scan completes, click on the "Alerts" tab.



- Alerts (14)
  - Cross Site Scripting (Persistent)
  - Cross Site Scripting (Reflected) (14)
  - SQL Injection (3)
  - Application Error Disclosure (54)
  - Directory Browsing (6)
  - X-Frame-Options Header Not Set (81)
  - Absence of Anti-CSRF Tokens (65)
  - Content-Type Header Missing (9)
  - Cookie No HttpOnly Flag (11)
  - Cookie Without SameSite Attribute (11)
  - Server Leaks Information via "X-Powered-By" HTTP Respor
  - X-Content-Type-Options Header Missing (136)
  - Information Disclosure - Suspicious Comments (4)
  - Timestamp Disclosure - Unix (1657)

Full details of any selected alert will be displayed here.

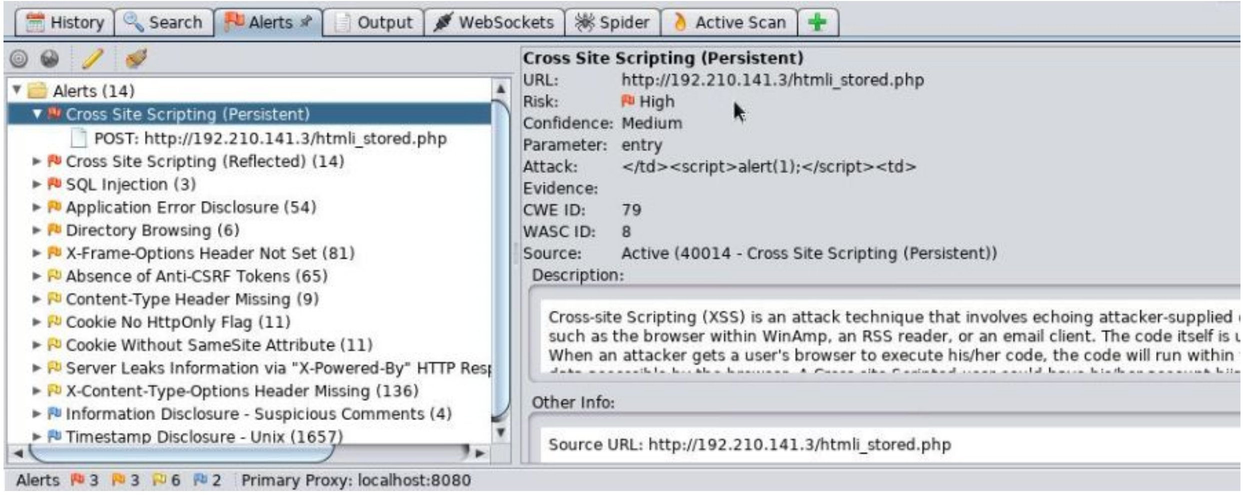
You can manually add alerts by right clicking on the relev:

You can also edit existing alerts by double clicking on ther

Alerts 3 3 6 2 Primary Proxy: localhost:8080

There are 3 critical alerts.

Step 32: Click on Cross Site Scripting (Persistent)



Alerts (14)
- Cross Site Scripting (Persistent)
  - POST: http://192.210.141.3/htmli_stored.php
- Cross Site Scripting (Reflected) (14)
- SQL Injection (3)
- Application Error Disclosure (54)
- Directory Browsing (6)
- X-Frame-Options Header Not Set (81)
- Absence of Anti-CSRF Tokens (65)
- Content-Type Header Missing (9)
- Cookie No HttpOnly Flag (11)
- Cookie Without SameSite Attribute (11)
- Server Leaks Information via "X-Powered-By" HTTP Resp
- X-Content-Type-Options Header Missing (136)
- Information Disclosure - Suspicious Comments (4)
- Timestamp Disclosure - Unix (1657)

```
Cross Site Scripting (Persistent)
URL:         http://192.210.141.3/htmli_stored.php
Risk:        High
Confidence:  Medium
Parameter:   entry
Attack:      </td><script>alert(1);</script><td>
Evidence:
CWE ID:      79
WASC ID:     8
Source:      Active (40014 - Cross Site Scripting (Persistent))
Description:
```

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied
such as the browser within WinAmp, an RSS reader, or an email client. The code itself is
When an attacker gets a user's browser to execute his/her code, the code will run within

Other Info:

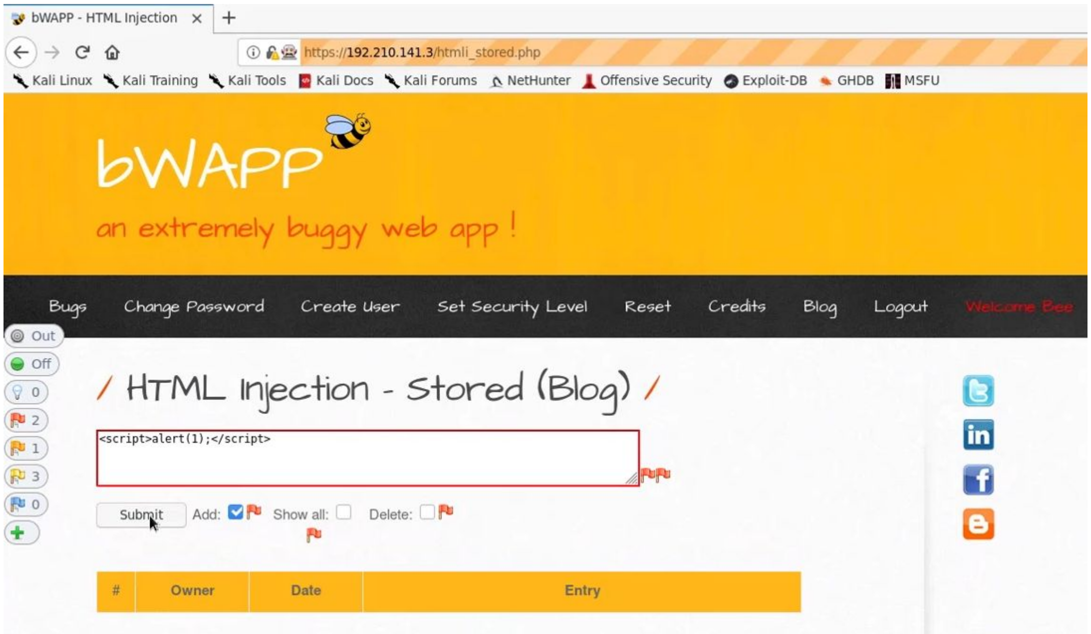
Source URL: http://192.210.141.3/htmli_stored.php

Alerts 3 3 6 2 Primary Proxy: localhost:8080

The information regarding the URL, payload, description about the vulnerability will be displayed.

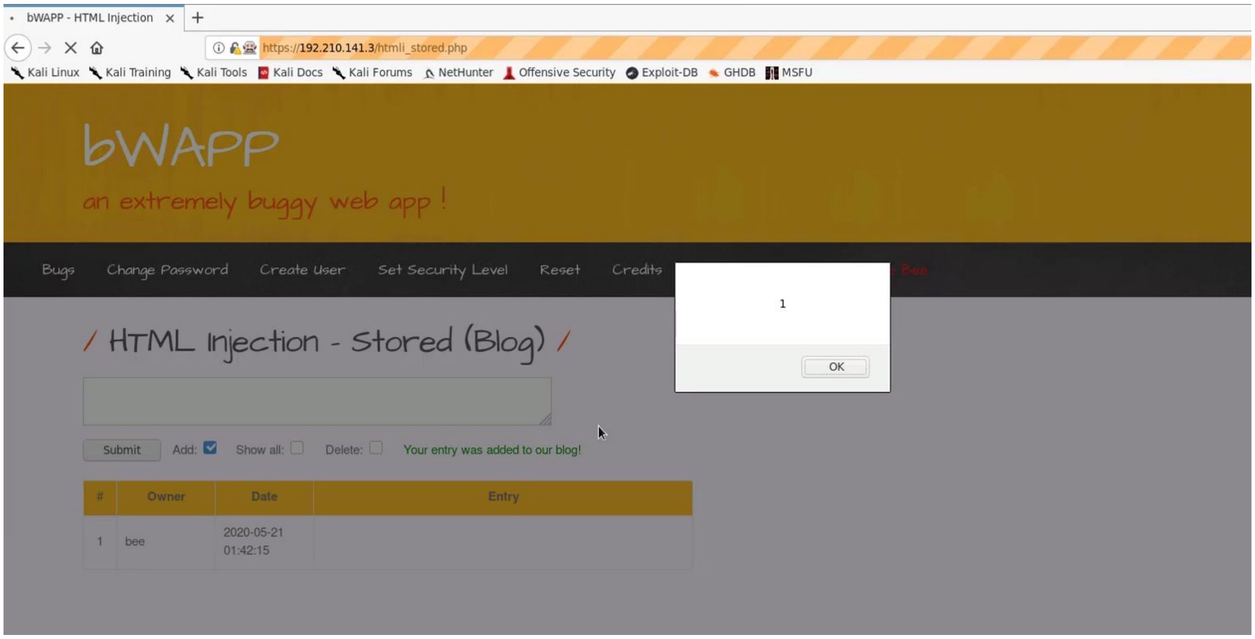
Step 33: Navigate to the URL, Inject the XSS payload and click on Submit button.

URL: http://192.210.141.3/htmli_stored.php

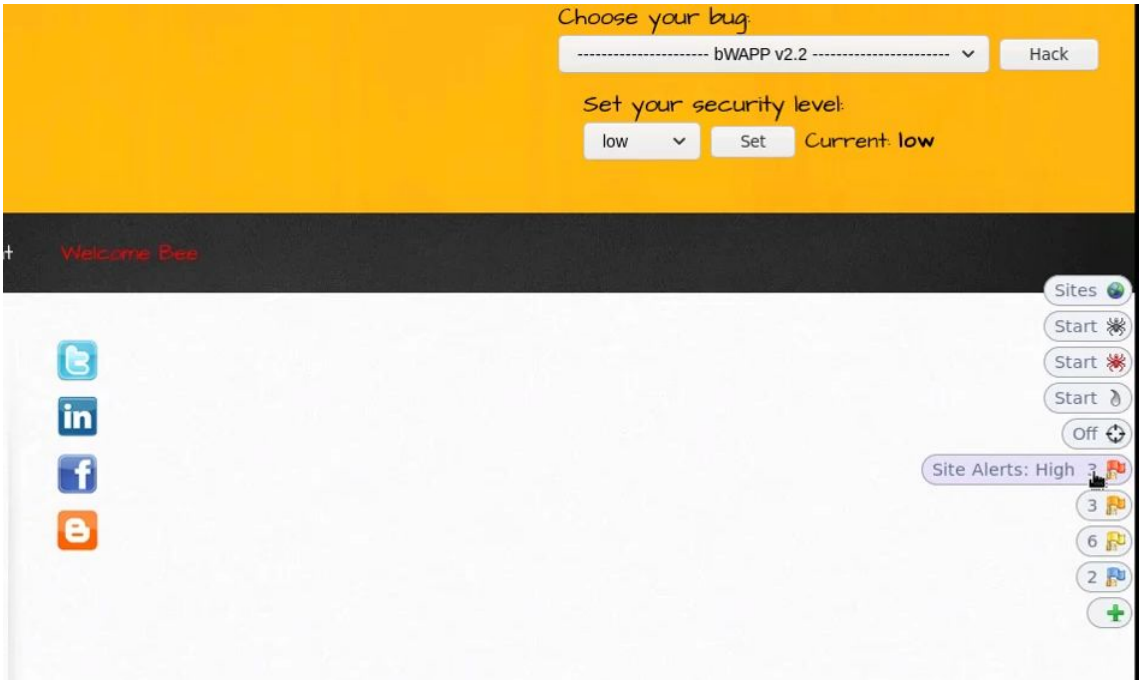


```
<script>alert(1);</script>
```

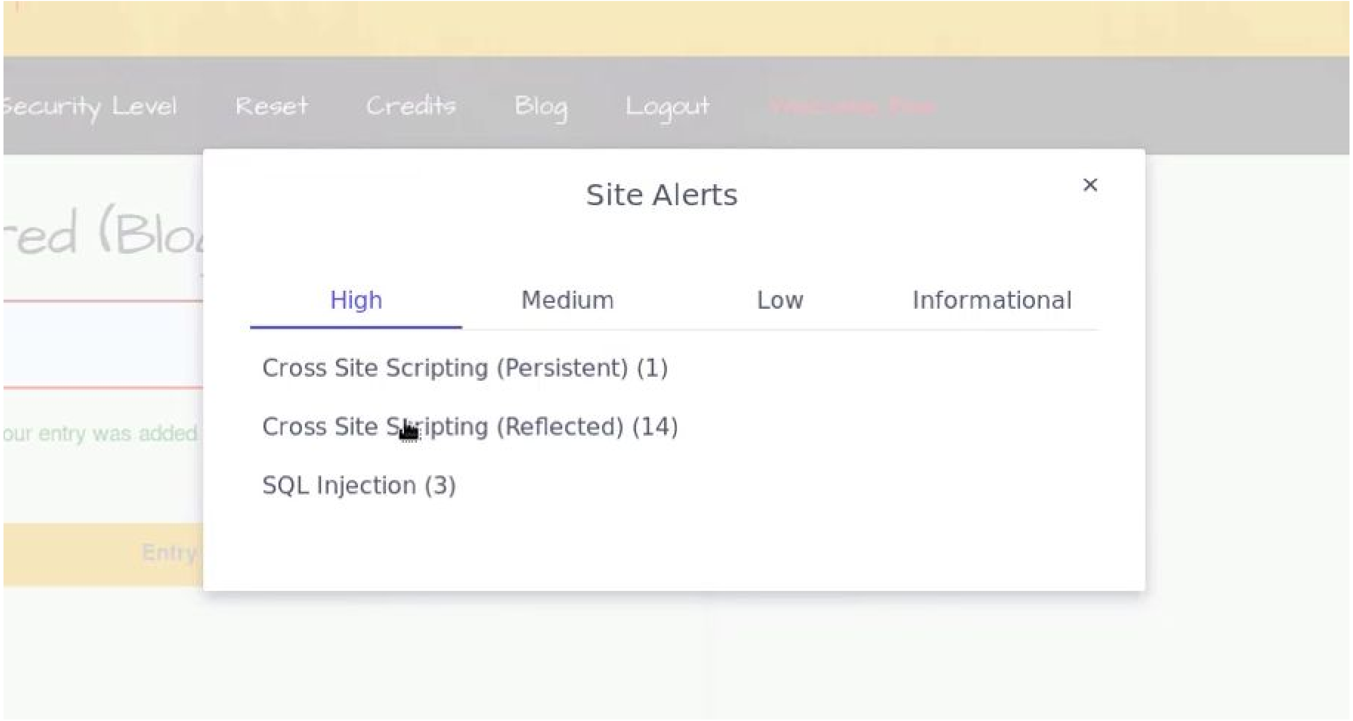
The XSS payload will be triggered.



Step 34: Navigate to the right side and access the Alert section of the ZAP HUD.



ZAP HUD:

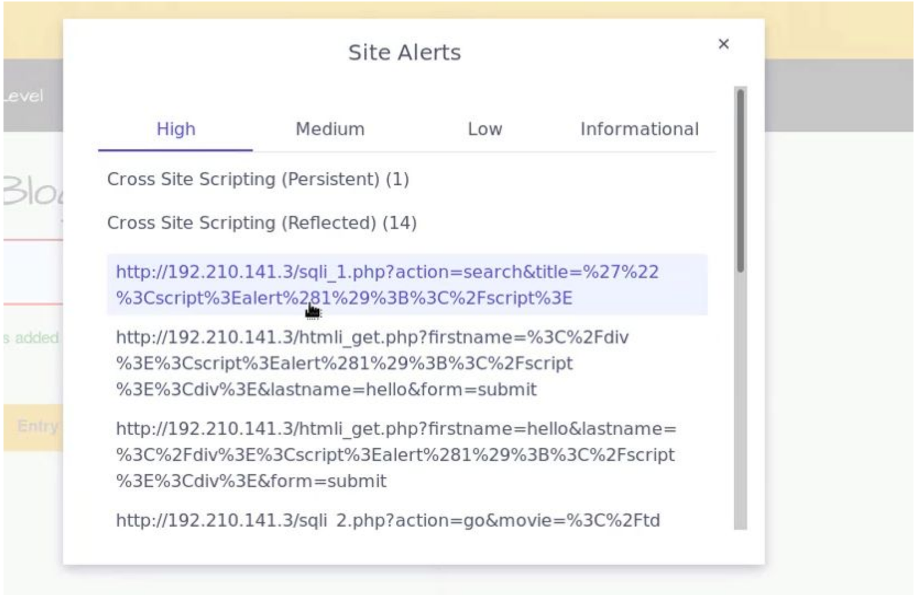


Cross Site Scripting (Persistent) (1)

Cross Site Scripting (Reflected) (14)

SQL Injection (3)

Step 35: Click on the "Cross Site Scripting (Reflected)" and click on the first URL.



Cross Site Scripting (Persistent) (1)

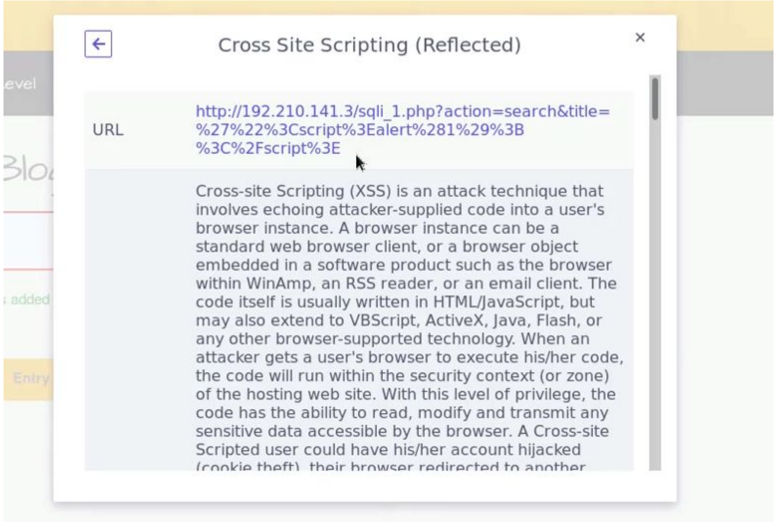
Cross Site Scripting (Reflected) (14)

http://192.210.141.3/sqli_1.php?action=search&title=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E

http://192.210.141.3/htmli_get.php?firstname=%3C%2Fdiv%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E%3Cdiv%3E&lastname=hello&form=submit

http://192.210.141.3/htmli_get.php?firstname=hello&lastname=%3C%2Fdiv%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E%3Cdiv%3E&form=submit

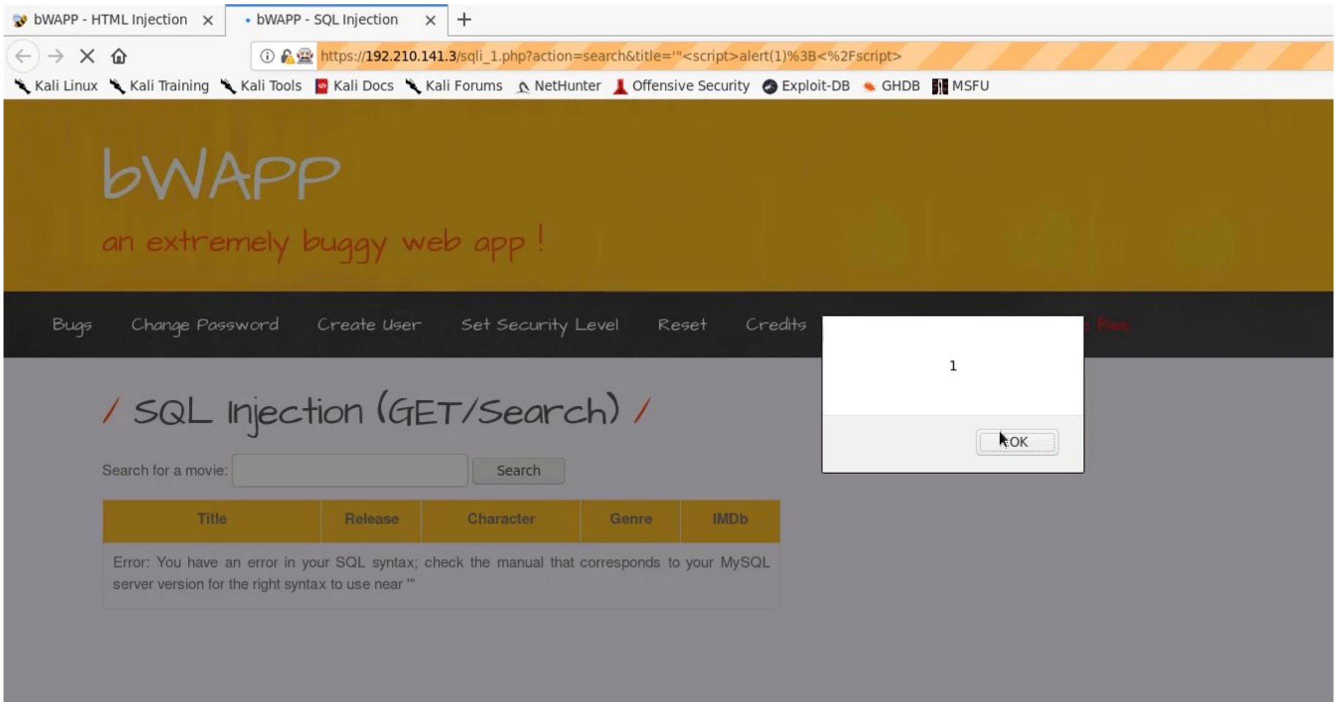
http://192.210.141.3/sqli 2.php?action=go&movie=%3C%2Ftd



URL http://192.210.141.3/sqli_1.php?action=search&title=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology. When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another

Step 36: Click on the URL on the dialog box.



https://192.210.141.3/sqli_1.php?action=search&title='"<script>alert(1)%3B<%2Fscript>

1

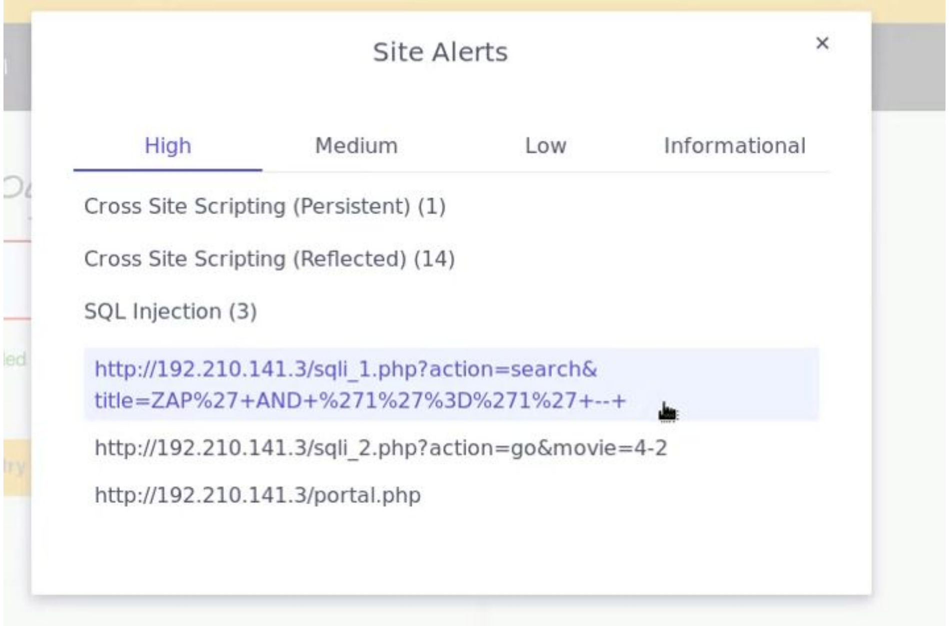
/ SQL Injection (GET/Search) /

Search for a movie: Search

Title	Release	Character	Genre	IMDb
Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '"				

The XSS payload will be triggered.

Step 37: Expand the SQL Injection Section from the Alerts section of ZAP HUD



Site Alerts

High | Medium | Low | Informational

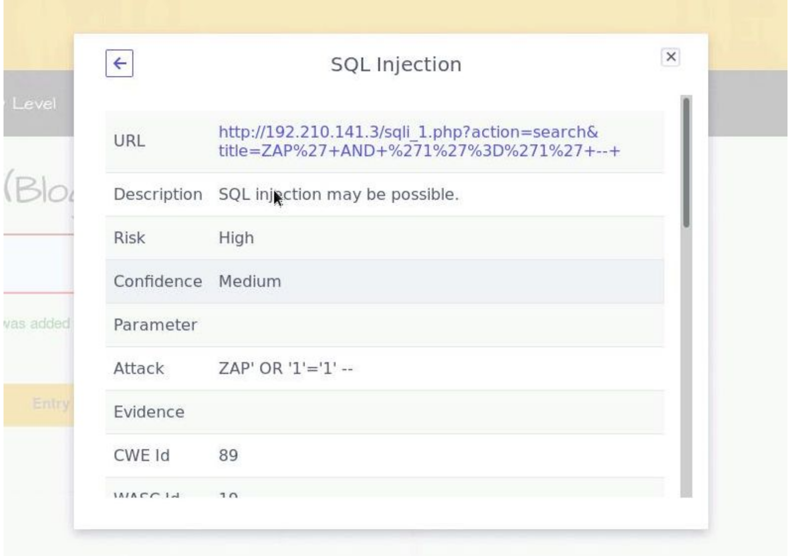
Cross Site Scripting (Persistent) (1)

Cross Site Scripting (Reflected) (14)

SQL Injection (3)

- http://192.210.141.3/sqli_1.php?action=search&title=ZAP%27+AND+%271%27%3D%271%27+--+
- http://192.210.141.3/sqli_2.php?action=go&movie=4-2
- http://192.210.141.3/portal.php

Step 38: Click on the First URL.

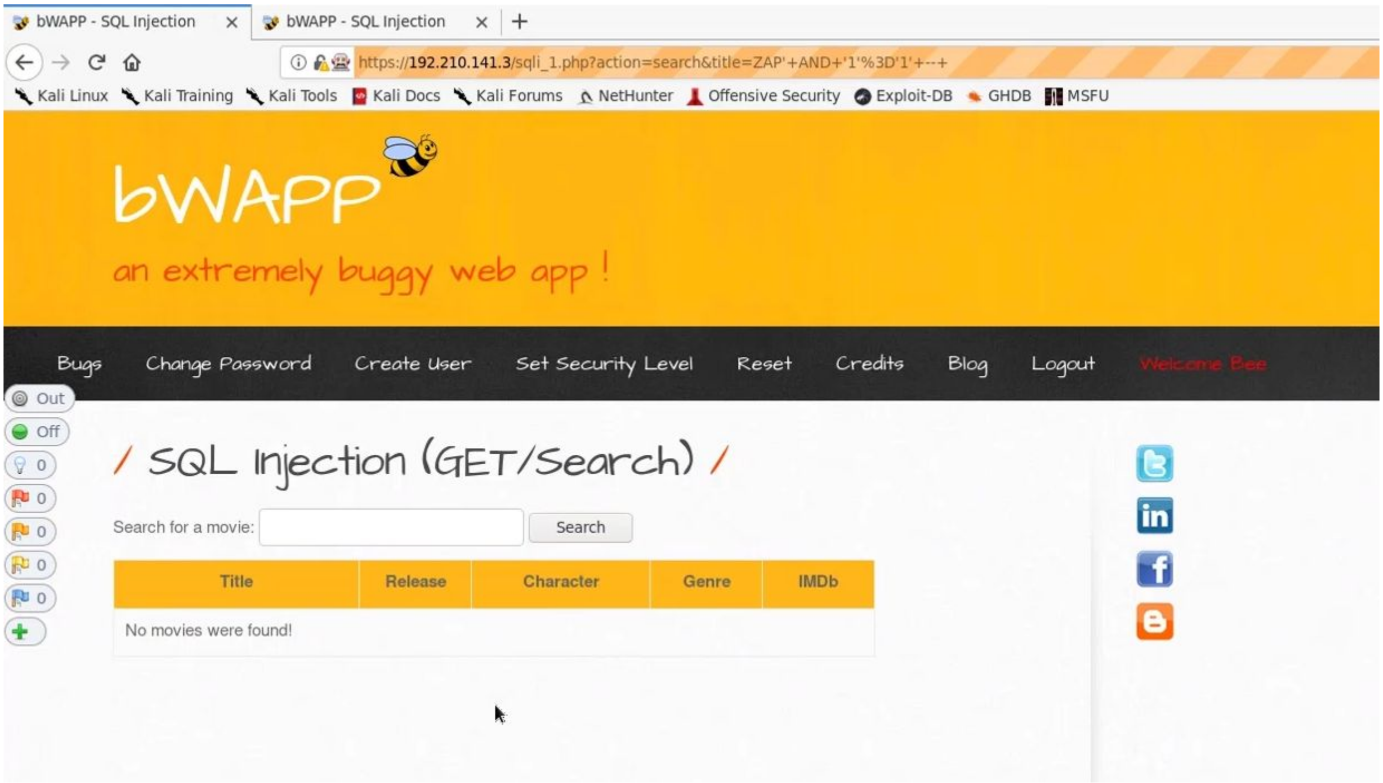


SQL Injection

URL	http://192.210.141.3/sqli_1.php?action=search&title=ZAP%27+AND+%271%27%3D%271%27+--+
Description	SQL injection may be possible.
Risk	High
Confidence	Medium
Parameter	
Attack	ZAP' OR '1'='1' --
Evidence	
CWE Id	89
WASC Id	19

Step 39: Navigate to the URL

URL: http://192.210.141.3/sqli_1.php?action=search&title=ZAP'+AND+'1'%3D'1'+--+



/ SQL Injection (GET/Search) /

Search for a movie:

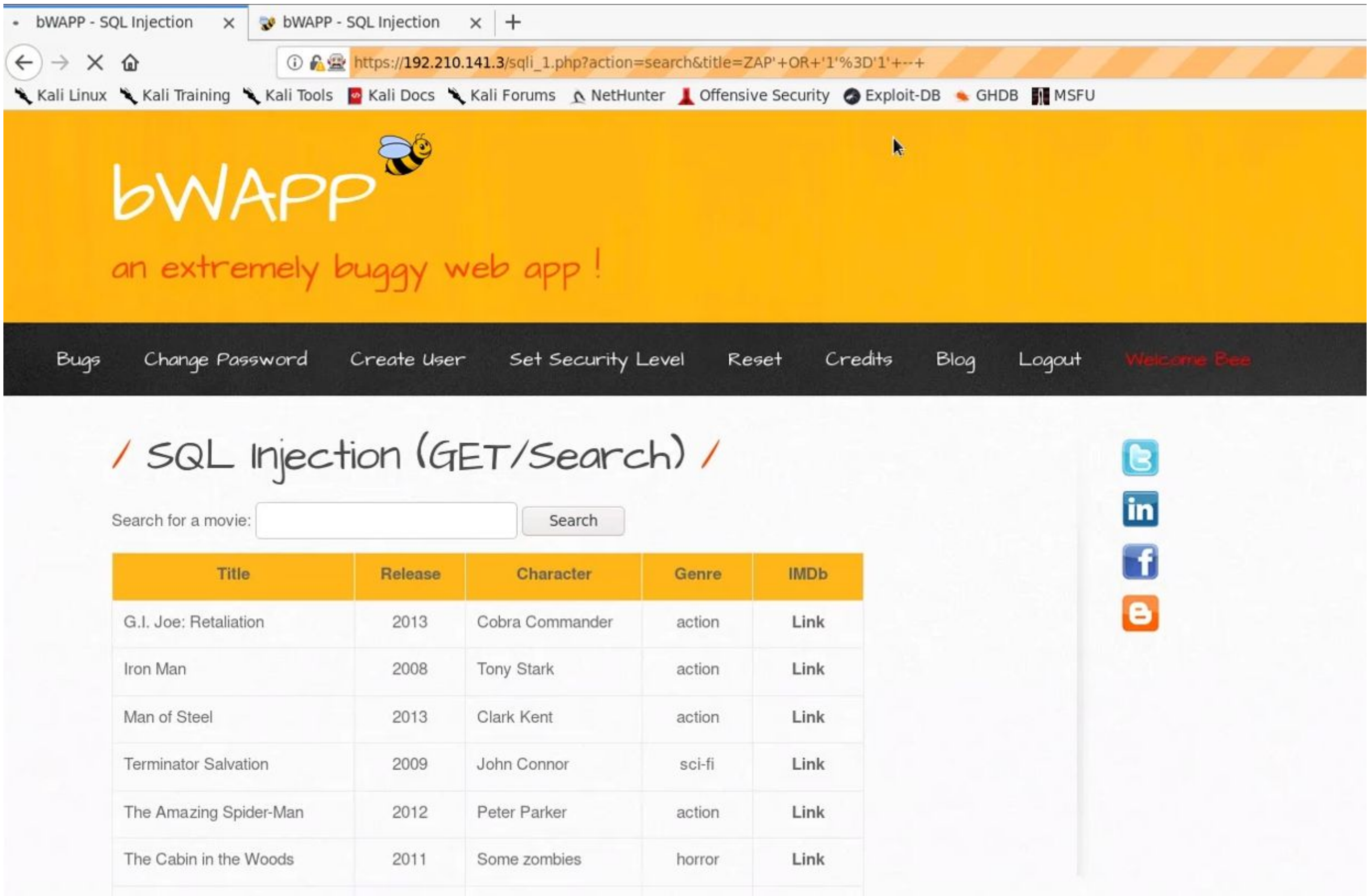
Title	Release	Character	Genre	IMDb
No movies were found!				

No Records will appear.

Step 40: In the URL, change the AND condition into OR.

URL: http://192.210.141.3/sqli_1.php?action=search&title=ZAP'+OR+'1'%3D'1'+--+

The Payload to use is also mentioned on the Vulnerablity information window (step 38).



https://192.210.141.3/sqli_1.php?action=search&title=ZAP'+OR+'1'%3D'1'+--+

/ SQL Injection (GET/Search) /

Search for a movie:

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

All the records present in the table will be dumped on the web page.

References:

1. OWASP Zed Attack Proxy (https://www.zaproxy.org/)
2. bWAPP (http://www.itsecgames.com/)