

## Experiment 01

### Aim:

To install and set up Burp Suite, Mutillidae, and Kali Linux for web application security testing.

---

### Tools:

- **Burp Suite** – Web vulnerability scanner and proxy.
  - **Mutillidae** – A deliberately vulnerable web application.
  - **Kali Linux** – A Linux distribution for penetration testing and security auditing.
- 

### Theory:

- **Burp Suite** is an integrated platform used for testing web application security. It includes features like proxy, spider, scanner, intruder, repeater, and sequencer.
  - **Mutillidae** is a free, open-source, deliberately vulnerable web application for learning and testing security tools.
  - **Kali Linux** is a Debian-based distribution equipped with numerous tools for ethical hacking, penetration testing, and digital forensics.
- 

### Steps for Installation and Setup:

#### 1. Kali Linux Installation

- Download Kali Linux ISO from the official website: <https://www.kali.org/> ● Install using:
  - **VirtualBox/VMware** (recommended for beginners).
  - OR create a bootable USB and install directly on hardware.
- Follow on-screen instructions to complete the installation.

#### 2. Burp Suite Setup

- Burp Suite is pre-installed in Kali Linux. To launch:

Applications → Web Application Analysis → Burp Suite

- Configure the browser (Firefox/Chrome) to use proxy:
  - Proxy IP: 127.0.0.1

- Proxy Port: 8080
- Import Burp's CA certificate into the browser for HTTPS interception.

### 3. Mutillidae Installation

- Prerequisites: Apache, MySQL, PHP (pre-installed in Kali).
- Clone the repository:

git clone https://github.com/webpwnized/mutillidae.git

sudo mv mutillidae /var/www/html/

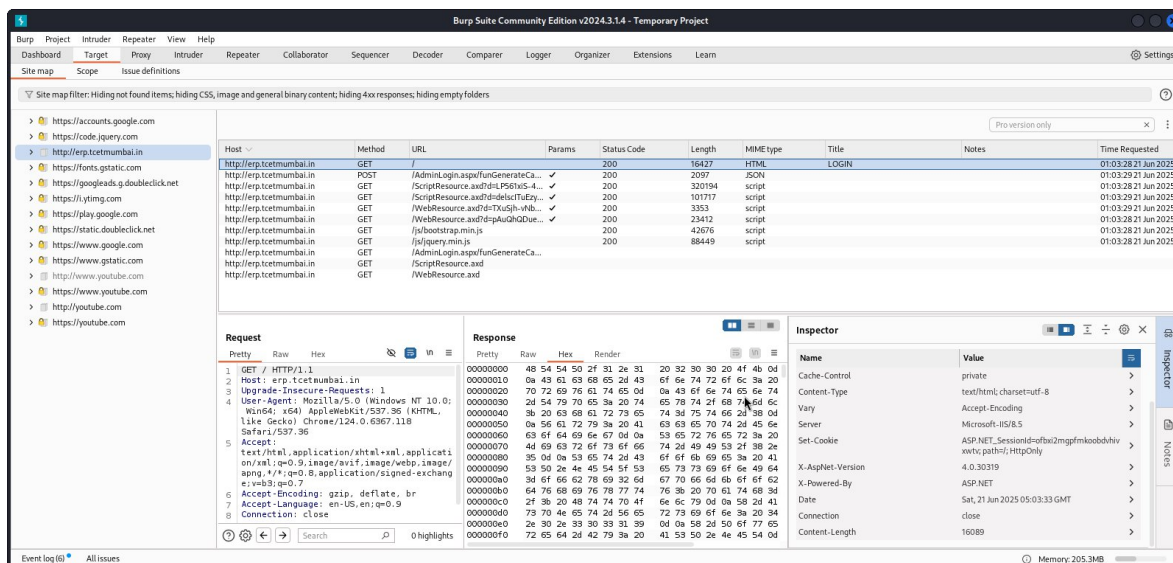
Start services:

sql  
 sudo service apache2 start

sudo service mysql start

- Setup database:
  - Open browser and navigate to http://localhost/mutillidae
  - Click on "Setup/reset the DB" to initialize the database.

Burp Suite:



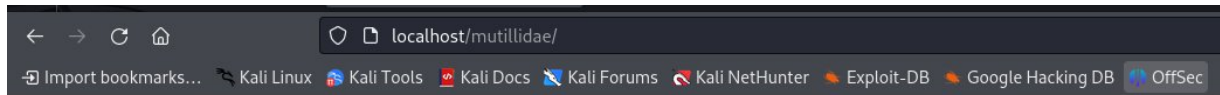
The screenshot shows the Burp Suite interface with the Site map on the left and a list of HTTP requests in the main pane. The Site map shows a directory structure for http://erp.tctcmumbai.in. The list of requests includes a GET request for /, a POST request for /AdminLogin.aspx, and several GET requests for /ScriptResource.axd, /WebResource.axd, and /js/jquery.min.js.

Host	Method	URL	Params	Status Code	Length	MIME type	Title	Notes	Time Requested
http://erp.tctcmumbai.in	GET	/		200	16427	HTML	LOGIN		01:03:28 21 Jun 2025
http://erp.tctcmumbai.in	POST	/AdminLogin.aspx		200	2097	JSON			01:03:29 21 Jun 2025
http://erp.tctcmumbai.in	GET	/ScriptResource.axd?d=PS6H5L...		200	320194	script			01:03:28 21 Jun 2025
http://erp.tctcmumbai.in	GET	/ScriptResource.axd?d=5c5c7UeY...		200	101717	script			01:03:29 21 Jun 2025
http://erp.tctcmumbai.in	GET	/WebResource.axd?d=TXu5h-vh...		200	3353	script			01:03:29 21 Jun 2025
http://erp.tctcmumbai.in	GET	/WebResource.axd?d=pu2hQ2Due...		200	23412	script			01:03:28 21 Jun 2025
http://erp.tctcmumbai.in	GET	/js/bootstrap.min.js		200	42676	script			01:03:28 21 Jun 2025
http://erp.tctcmumbai.in	GET	/js/jquery.min.js		200	88449	script			01:03:28 21 Jun 2025
http://erp.tctcmumbai.in	GET	/AdminLogin.aspx		200	2097	JSON			01:03:29 21 Jun 2025
http://erp.tctcmumbai.in	GET	/ScriptResource.axd		200	320194	script			01:03:28 21 Jun 2025

The Request pane shows the details of the selected GET request for /, including the raw HTTP request and response. The Response pane shows the raw response data, which is a 200 status code with a Content-Type of text/html; charset=utf-8.



Mutillidae:



## Index of /mutillidae

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">CHANGELOG.md</a>	2025-06-21 01:35	1.0K	
<a href="#">CONTRIBUTING.md</a>	2025-06-21 01:35	2.7K	
<a href="#">LICENSE</a>	2025-06-21 01:35	34K	
<a href="#">README-INSTALLATION.md</a>	2025-06-21 01:35	1.5K	
<a href="#">README.md</a>	2025-06-21 01:35	4.7K	
<a href="#">SECURITY.md</a>	2025-06-21 01:35	1.8K	
<a href="#">src/</a>	2025-06-21 01:35	-	
<a href="#">version</a>	2025-06-21 01:35	6	

Apache/2.4.59 (Debian) Server at localhost Port 80

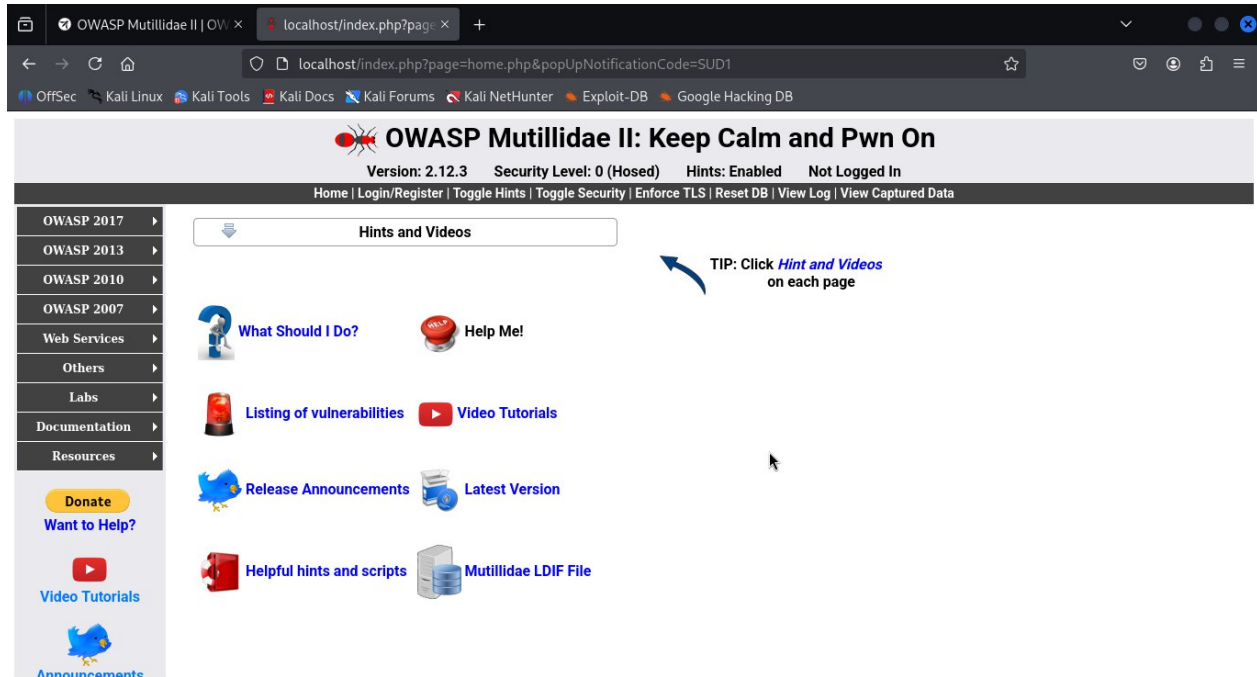
```
(kali㉿kali)-[~]
└─$ git clone https://github.com/webpwnized/mutillidae.git
Cloning into 'mutillidae'...
remote: Enumerating objects: 8099, done.
remote: Counting objects: 100% (1225/1225), done.
remote: Compressing objects: 100% (255/255), done.
remote: Total 8099 (delta 1073), reused 970 (delta 970), pack-reused 6874 (from 3)
Receiving objects: 100% (8099/8099), 10.64 MiB | 10.33 MiB/s, done.
Resolving deltas: 100% (4340/4340), done.

(kali㉿kali)-[~]
└─$ sudo mv mutillidae /var/www/html/
sudo has not been fully tested on this platform and you may experience problems.

(kali㉿kali)-[~]
└─$ sudo service apache2 start

(kali㉿kali)-[~]
└─$ sudo service mysql start

(kali㉿kali)-[~]
└─$
```



## Conclusion:

The experiment provided hands-on experience with setting up a safe and controlled web application security testing environment using Kali Linux, Burp Suite, and Mutillidae.

## Theory Questions:

1. What are the different modules provided by Burp Suite for testing web vulnerabilities?
2. Explain how a browser is configured to work with Burp Suite's proxy.
3. What is the purpose of Mutillidae in web security testing?

## Learning Outcomes:

### Outcomes:

- Demonstrate the ability to successfully install and configure Burp Suite, Mutillidae, and Kali Linux to create a secure testing environment for web application security.



**TCET**  
**BE COMPUTER SCIENCE & ENGINEERING (CYBER SECURITY)**  
Choice Based Credit Grading System (CBCGS)  
Under TCET Autonomy



- Understand and apply the basic functionality of Burp Suite and Mutillidae in conjunction with Kali Linux tools to perform vulnerability assessments and exploit web application security flaws.

**Conclusion:**

**Name :**

**Class: BE CSE**

**Roll no:**

For Faculty Use:

<b>Correction Parameters</b>	<b>Formative Assessment [40%]</b>	<b>Timely completion of Practical [40%]</b>	<b>Attendance / Learning Attitude [20%]</b>	
<b>Marks Obtained</b>				