BRUTE-FORCE ATTACK IMPLEMENTATION USING BURP SUITE & HYDRA ON bWAPP IN KALI LINUX

1. INTRODUCTION

Cybersecurity is essential for modern digital infrastructure. Organizations depend on web applications and online services for their daily operations. One area frequently targeted by attackers is the authentication mechanism. They try to bypass login controls to gain unauthorized access. Among the techniques used, brute-force attacks are straightforward yet effective, especially when systems do not enforce strong password policies, limit login attempts, or implement account lockout measures. In a brute-force attack, an attacker repeatedly submits various username-password combinations until finding the right one.

This project focuses on conducting brute-force attacks in a controlled, ethical testing environment using bWAPP, a purposely vulnerable web application meant for security training and research. By running bWAPP on a local XAMPP server within Kali Linux, the project mimics a real-world login system where authentication weaknesses can be safely examined. Two common penetration testing tools, Burp Suite Intruder and Hydra, demonstrate different methods of brute-force attacks. Burp Suite allows for manual interception and payload-based brute-forcing of web login forms, while Hydra enables quick automated credential testing with predefined wordlists.

By putting these techniques into practice, this project shows how attackers exploit weak authentication systems and what signs indicate successful breaches. The study highlights the risks associated with insecure login mechanisms. It also emphasizes the need for strong security measures like multi-factor authentication, CAPTCHA, account lockout policies, and secure password management. This hands-on experience offers valuable insights for students and security professionals. It helps them create more secure applications and respond effectively to real-world cyber threats.

2. OBJECTIVE OF THE PROJECT

The main goal of this project is to build a realistic and controlled vulnerable environment with bWAPP, where brute-force attacks can be safely carried out and studied. By running bWAPP on XAMPP within Kali Linux, this project offers a practical platform for understanding how flawed authentication methods work. This setup allows for examining weaknesses like poor password security, missing rate-limiting, and inadequate server-side validation.

Another key aim is to study the authentication process by capturing HTTP POST requests with Burp Suite. By intercepting raw login requests, we can see the parameters sent from the client to the server, such as usernames, passwords, and form values. Using this information, the project conducts a Cluster Bomb brute-force attack with Burp Suite Intruder to systematically test various username-password combinations. This approach helps us learn how both manual and semi-automated brute-force methods function and how to recognize successful login attempts by looking at response codes and content length differences.

The last aim is to execute automated brute-force attacks using Hydra, a fast, multi-threaded password-cracking tool. By creating custom wordlists and carrying out HTTP form-based attacks, the project seeks to compare Hydra's performance with that of Burp Suite. Once valid credentials are found, the project reviews the results and suggests important ways to reduce risks, such as enforcing strong passwords, implementing rate-limiting, using CAPTCHA, and adding Multi-Factor Authentication. These aims collectively improve the practical understanding of brute-force attack methods and emphasize the need for secure authentication systems in real-world situations.

3. EXSISTING SYSTEM

In many web applications, authentication mechanisms are poorly implemented. This leaves systems vulnerable to brute-force attacks. Most login pages do not enforce important security measures like rate limiting, account lockout, or CAPTCHA verification. As a result, attackers can repeatedly try many username-password combinations without restrictions. Automated tools like Hydra and Burp Suite Intruder can easily exploit weak credentials. Additionally, many applications still use default or predictable passwords, making it easier for attackers to gain unauthorized access.

Another significant weakness in current systems is how they handle login data and server responses. Some systems store passwords in plain text or use outdated hashing methods. This puts user accounts at risk if there is a breach. Many applications also fail to maintain proper logging or intrusion detection, allowing brute-force attempts to go unnoticed. Because of these issues—unrestricted login attempts, weak password policies, lack of monitoring, and outdated security practices—existing systems are still very vulnerable to brute-force and dictionary attacks.

Key Weaknesses in Existing Systems

✓ No rate-limiting on login attempts

✓ No account lockout after multiple failed attempts

✓ Weak, default, or predictable passwords

✓ No CAPTCHA to block automated bots

✓ Insecure password storage (plain text / weak hashing)

✓ Poor monitoring and logging

✓ Lack of Multi-Factor Authentication (MFA)

✓ Outdated authentication protocols

4. PROPOSED SYSTEM

The proposed system aims to create a completely controlled and secure environment for testing and analyzing brute-force attacks using bWAPP, Burp Suite, and Hydra. By hosting a deliberately vulnerable web application on XAMPP within Kali Linux, the system ensures that all testing remains legal and safe, without impacting real-world servers. This setup allows for a structured observation of how login requests are processed, how attackers capture those requests, and how automated payload-based attacks are carried out. The model also makes it possible to monitor response patterns, such as HTTP status codes, redirect behavior, and differences in content length, which help identify successful credential combinations.

Additionally, the proposed model stresses the importance of understanding and applying security best practices after running the attacks. Once brute-force vulnerabilities are found using Burp Suite Intruder and Hydra, the system serves as a basis for examining how attackers exploit weak authentication methods. It then focuses on suggesting countermeasures like strong password policies, rate-limiting, CAPTCHA integration, server-side validation, account lockout mechanisms, and Multi-Factor Authentication. This model not only shows brute-force attack techniques but also emphasizes effective defense strategies, helping organizations strengthen their authentication systems against real-world threats.

5. SYSTEM REQUIREMENTS

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Hardware Requirements

- **Processor:** Dual-Core or higher
- **RAM:** Minimum 4 GB (8 GB recommended for smooth tool execution)
- **Storage:** At least 10 GB free space for Kali Linux, XAMPP, and wordlists
- **Internet:** Required only for downloading tools and updates
- **System Type:** Laptop or desktop capable of running virtualization or dual boot

Software Requirements

- **Operating System:** Kali Linux (latest version recommended)
- **Web Server:** XAMPP with PHP 5.6.40 (required for bWAPP compatibility)
- **Vulnerable Application:** bWAPP (Buggy Web Application)
- **Attack Tools:**
    - Burp Suite Community Edition
    - Hydra (pre-installed or installed via package manager)
- **Browser:** Firefox (configured to use Burp Proxy)
- **Dependencies:**
    - Apache & MySQL (via XAMPP)
    - Required Linux utilities (wget, unzip, wordlist tools)

6. TOOLS & TECHNOLOGIES USED

1. Burp Suite Community Edition

Burp Suite helps intercept, analyze, and manipulate HTTP requests sent between the browser and the web server. The Intruder module allows for manual and semi-automated brute-force attacks using multiple payload combinations. It is crucial for understanding how attackers capture login requests and spot response differences to find valid credentials.

2. Hydra

Hydra is a powerful, multi-threaded password-cracking tool for automating brute-force attacks. It supports many protocols, including HTTP, SSH, and FTP. In this project, Hydra is used for high-speed brute-force attacks on the bWAPP login form. This demonstrates automated credential-guessing techniques and examines successful login attempts.

3. XAMPP (Apache + MySQL + PHP)

XAMPP hosts the vulnerable web application bWAPP locally. It provides Apache to run the PHP-based application and MySQL to store user credentials. XAMPP creates a safe and controlled test environment for experimenting with brute-force attacks.

4. bWAPP (Buggy Web Application)

bWAPP is an intentionally vulnerable web application made for security training and testing. It includes insecure authentication features that allow brute-force attacks to be executed safely. It acts as the target application for both Burp Suite and Hydra.

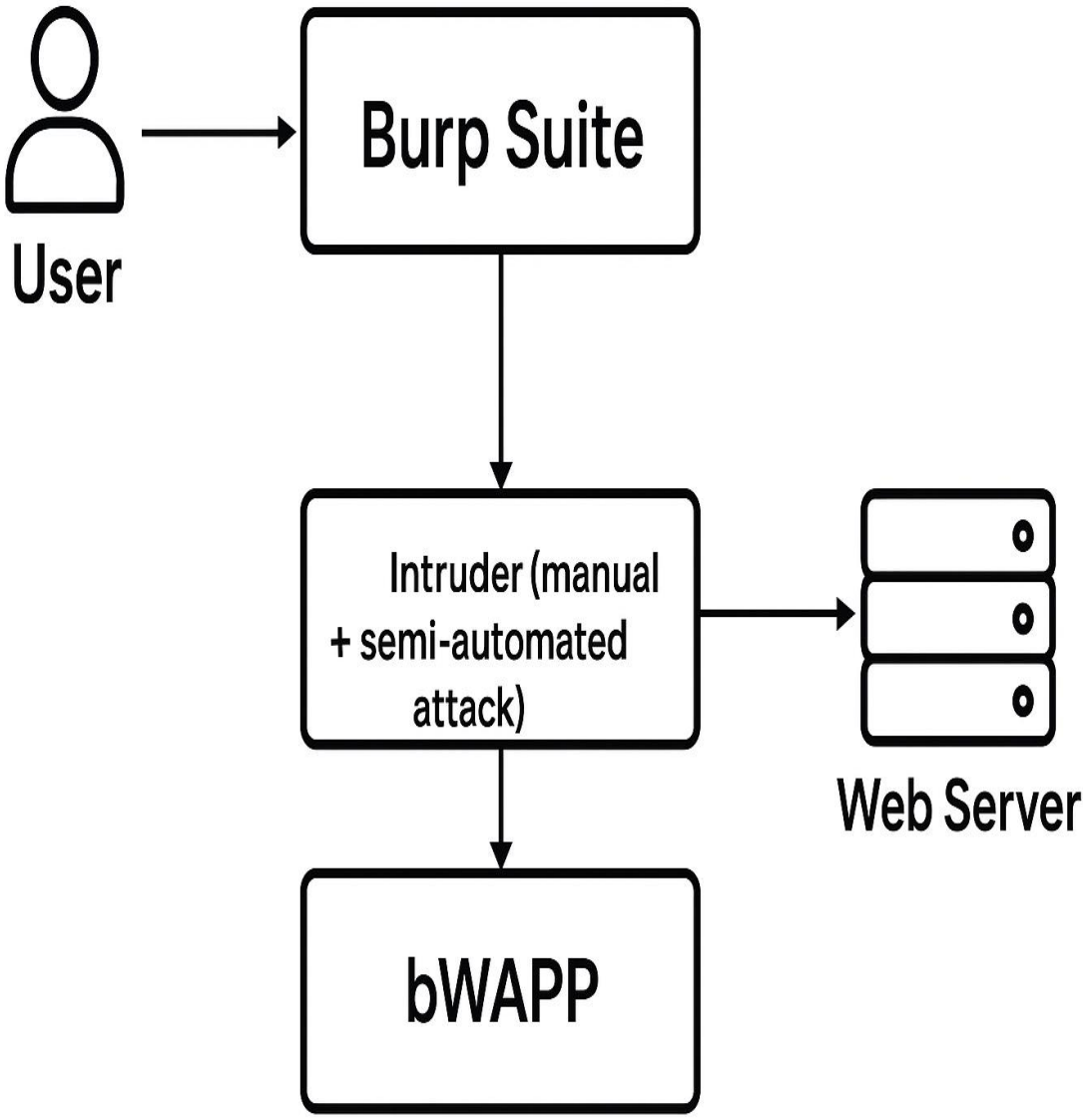
5. Kali Linux

Kali Linux is the main operating system for conducting penetration testing. It comes with pre-installed security tools and offers a suitable environment for running Burp Suite, Hydra, and other necessary utilities.

6. Mozilla Firefox Browser

Firefox is used to access the bWAPP login page and route traffic through the Burp Suite proxy. It helps capture and analyze the login request for brute-force testing.

7. ARCHITECTURE DIAGRAM



8. WORKFLOW OF THE PROJECT

The workflow for this project follows a clear, step-by-step process to carry out brute-force attacks on the bWAPP login system using Burp Suite and Hydra. The approach focuses on capturing the login request, setting attack parameters, conducting manual and automated brute-force attempts, analyzing results, and suggesting security improvements. Each phase is critical for understanding how attackers target weak authentication methods and how to reduce these vulnerabilities with appropriate security measures.

1. Capturing the Login Request

The workflow starts by entering sample credentials into the bWAPP login form. Burp Suite Proxy intercepts this request, which allows us to view the raw HTTP POST data. The intercepted request contains important details such as the username parameter, password parameter, cookies, and validation messages. This captured request serves as the basis for configuring both Burp Suite Intruder and Hydra for the brute-force attack.

2. Configuring Burp Suite Intruder

After capturing the login request, it is sent to the Intruder module in Burp Suite. Here, the username and password fields are chosen as points for injection. Appropriate payload lists (wordlists) are loaded for both fields. The Intruder attack type, usually Cluster Bomb, is selected to test all possible combinations of usernames and passwords. Once configured, the attack is launched. Intruder sends multiple automated login attempts to the target application and logs each response. By comparing response lengths or indicators of success, valid credentials can be identified.

3. Analyzing Burp Suite Results

During the Intruder attack, each login attempt generates a distinct response. The success of brute-force attempts is assessed by analyzing:

- Response length
- HTTP status codes
- Redirect behaviour
- Presence or absence of error messages

Any response that is notably different from the rest (for example, a longer response or a redirect to a dashboard page) suggests a successful login. These variations are recorded as successful credential findings.

4. Preparing for Hydra Attack

Before running Hydra, the necessary parameters from the login request, such as form action path, username/password field names, and failure message, are extracted. A Hydra command is built using this information along with selected username and password wordlists. Hydra is set up to execute many fast, parallel authentication attempts using its multi-threading features.

5. Executing the Hydra Brute-Force Attack

Hydra is run with the prepared command. It quickly attempts all combinations of usernames and passwords against the bWAPP login page using the HTTP POST method. During execution, Hydra shows real-time output displaying attempts, progress, and any successful logins. When correct credentials are found, the tool highlights them clearly, confirming their validity. This automated process is much quicker than manual or semi-automated methods.

6. Verifying Successful Credentials

Any successful username-password combination found by Burp Suite or Hydra is manually checked on the bWAPP login form. This verification step confirms that the identified credentials are valid and can be used. Successful login attempts are documented, and screenshots are taken for inclusion in the project report.

7. Result Compilation and Comparison

The results from both Burp Suite and Hydra are collected and compared. The comparison looks at:

- Number of attempts
- Time taken
- Accuracy
- Speed of detection
- Success rate

Hydra usually completes the process faster due to its multi-threaded design, while Burp Intruder offers deeper request-response analysis. Both tools provide valuable insights into brute-force attack patterns.

8. Reporting and Security Recommendations

The final phase of the workflow involves documenting all findings, including screenshots, tool outputs, and observations. The report highlights the vulnerabilities discovered and shows how easily weak authentication systems can be breached. Security recommendations are included, such as:

- Adding CAPTCHA
- Enforcing account lockout
- Implementing rate limiting
- Using strong password policies
- Monitoring login attempts

These measures help organizations strengthen their authentication systems and lower the risk of brute-force attacks.

9. EXPERIMENTAL SETUP

The experimental setup for this project aims to create a controlled and isolated environment where brute-force attack techniques can be safely carried out without impacting any real-world systems. The setup ensures that all attacks are performed solely on a deliberately vulnerable web application, allowing for ethical analysis of weaknesses in authentication methods. This configuration includes the target system (bWAPP), the attacker system (Kali Linux), necessary software tools, network setup, and components needed for traffic interception and automated password attacks.

1. Target Application Setup

The foundation of the experimental environment is the vulnerable web application bWAPP (Buggy Web Application), specifically designed for security testing and learning. bWAPP is installed on a local machine using a web server stack such as XAMPP that supports Apache, MySQL, and PHP. The application has several security vulnerabilities, including weak login methods, making it a suitable target for brute-force testing. The login page (login.php) serves as the main attack surface for this experiment. The credentials database is stored in the local MySQL server, allowing for controlled credential testing without affecting live systems. Once bWAPP is installed, the database is initialized, and the application can be accessed through the local URL: http://localhost/bWAPP/login.php

This setup ensures that the experiment is conducted in a safe, offline, and private environment.

2. Attacker Machine Setup

A separate system is set up as the attacker environment where all penetration testing tools are run. Kali Linux serves this purpose well because it comes with pre-installed cybersecurity tools and supports detailed network analysis. The key tools needed for this experiment include:

- Burp Suite Community Edition for intercepting and adjusting HTTP requests
- Hydra for carrying out automated brute-force attacks
- Mozilla Firefox for interacting with the target application

The attacker and target systems can run on the same machine or on different systems connected over a local network. Usually, the attacker machine communicates with the target through localhost or a virtual network interface.

3. Network Configuration

The attacker system (Kali Linux) needs to communicate with the target application hosted on XAMPP. The network configuration allows smooth traffic flow between the user's browser, Burp Suite, and the bWAPP application. The browser is set to use Burp Suite's proxy (127.0.0.1:8080) so that all HTTP requests go through Burp before reaching the server. This setup captures the login POST request, allowing analysis of parameters and forwarding it to Burp Suite Intruder for brute-force automation. No internet connection is needed because the entire setup operates locally to ensure security and ethical testing.

4. Wordlist and Payload Preparation

Preparing the username and password wordlists for brute-force attacks is a key part of the experimental setup. Two types of lists are used:

- A username list with possible user IDs like admin, bee, and test
- A password list featuring weak or commonly used passwords from resources like rockyou.txt

These lists are saved in text format and imported into both Burp Suite and Hydra for generating automated credential attempts. The variability in payloads and the size of the wordlists enhance the depth of the attack simulation.

5. Burp Suite Interception Setup

To analyze and manipulate traffic, Burp Suite must be properly configured before beginning the attack:

- Launch Burp Suite on Kali Linux.
- Enable intercept mode in the Proxy tab.
- Set the Firefox browser to use Burp Proxy at 127.0.0.1:8080.
- Attempt a login on bWAPP to let Burp Suite capture the HTTP POST request.

The raw request is saved and then sent to the Intruder module for brute-force testing. This setup allows for an in-depth evaluation of the application's response to repeated login attempts, helping to identify valid credentials based on differences in responses.

6. Hydra Execution Environment

For Hydra to function correctly, specific information from the captured login request needs to be extracted, including:

- Form action path
- Username parameter
- Password parameter
- Failure message returned by the application

Hydra uses this information to create automated brute-force commands that can test many passwords quickly. The setup ensures that the tool targets the correct URL and parameters, enabling a realistic simulation of external brute-force attacks.

7. Verification and Documentation Setup

A dedicated section is prepared to verify and document the results of the experiment. Successful credentials identified using Burp Suite or Hydra are manually tested in the browser to confirm their validity. Screenshots of:

- Captured requests
- Intruder attack results
- Hydra successful login messages
- Verification login results

are collected for inclusion in the project report.

10. IMPLEMENTATION

10.1 Installing Tools

To start brute-force testing, you need to install the right tools on the attacker's system running Kali Linux, which has many penetration-testing utilities already included. The tools you'll use are:

1. Update System Packages

sudo apt update && sudo apt upgrade -y

This makes sure the system is stable and all repositories are up to date.

2. Install Hydra

Hydra is a fast, parallelized brute-force tool that supports multiple protocols.

sudo apt install hydra -y

3. Install Burp Suite

Burp Suite Community Edition helps intercept and manipulate HTTP requests.

sudo apt install burpsuite -y

4. Install Additional Dependencies

sudo apt install wget unzip -y

These tools get Kali Linux ready for both manual and automated brute-force testing.

10.2 Setting Up XAMPP & bWAPP

To create a vulnerable target environment, install bWAPP using XAMPP on your local machine. bWAPP has insecure features, making it perfect for learning about cybersecurity.

Step 1: Download XAMPP (PHP 5.6 Version)

wget https://sourceforge.net/projects/xampp/files/XAMPP%20Linux/5.6.40/xampp-linux-x64-5.6.40-0-installer.run

chmod +x xampp-linux-x64-5.6.40-0-installer.run

sudo ./xampp-linux-x64-5.6.40-0-installer.run

Step 2: Start Apache and MySQL

sudo /opt/lampp/lampp start

Step 3: Extract and Move bWAPP

Download and extract the bWAPP application into the webroot folder:

sudo unzip bwapp.zip -d /opt/lampp/htdocs/

sudo mv /opt/lampp/htdocs/bWAPP /opt/lampp/htdocs/bwapp

sudo chmod -R 777 /opt/lampp/htdocs/bwapp

Step 4: Configure bWAPP Database

Open your browser and go to:

http://localhost/bwapp/install.php

Click on Create Database; the installation is complete.

The vulnerable login page is now available at:

http://localhost/bwapp/login.php

10.3 Burp Suite Cluster Bomb Attack

Burp Suite is used to analyze web requests and perform a manual brute-force attack using the Intruder, Cluster Bomb attack type.

Step 1: Configure Browser Proxy

Browser → Settings → Network

Set proxy:

Address: 127.0.0.1

Port: 8080

This routes all browser traffic through Burp Suite.

Step 2: Capture Login Request

Open Burp → Proxy → Intercept ON

Go to the bWAPP login page

Enter:

Username: test

Password: 12345

Submit → Burp captures:

POST /bwapp/login.php

login=test&password=12345&security_level=0&form=submit

Step 3: Send to Intruder

Right-click → Send to Intruder

Step 4: Mark Payload Positions

Burp automatically highlights parameters.

If not, manually set positions:

login=§test§&password=§12345§&security_level=0&form=submit

Step 5: Configure Cluster Bomb Attack

Payload Set 1 → Usernames

Payload Set 2 → Passwords

Example lists:

Usernames:	**Passwords:**
Admin	123456
Bee	Test
Guest	Admin
Root	Password
Test	Bug

Step 6: Start Attack & Analyze

Start the attack and observe the Status and Length columns.

A valid login shows:

Status code 302 (redirect) OR

Longer or shorter content length

Example discovered credential:

Username: bee

Password: bug

Burp Suite confirms weak authentication in the target web application.

10.4 Hydra Web Form Brute-force Attack

Hydra is used to carry out a fast, automated brute-force attack on the same login form.

Step 1: Create Wordlists

Usernames.txt	**Passwords.txt**
Admin	123456
Bee	Test
Guest	Admin
Root	Password
Test	Bug

Step 2: Identify the Form Parameters

Parameters identified using Burp request:

login=^USER^

password=^PASS^

Failure message:

Invalid credentials or user not activated!

Step 3: Run Hydra Command

```
hydra -L usernames.txt -P passwords.txt 127.0.0.1 http-post-form \
'/bwapp/login.php:login=^USER^&password=^PASS^&security_level=0&form=submit:Invalid credentials or user not activated!'
```

Step 4: Hydra Output Analysis

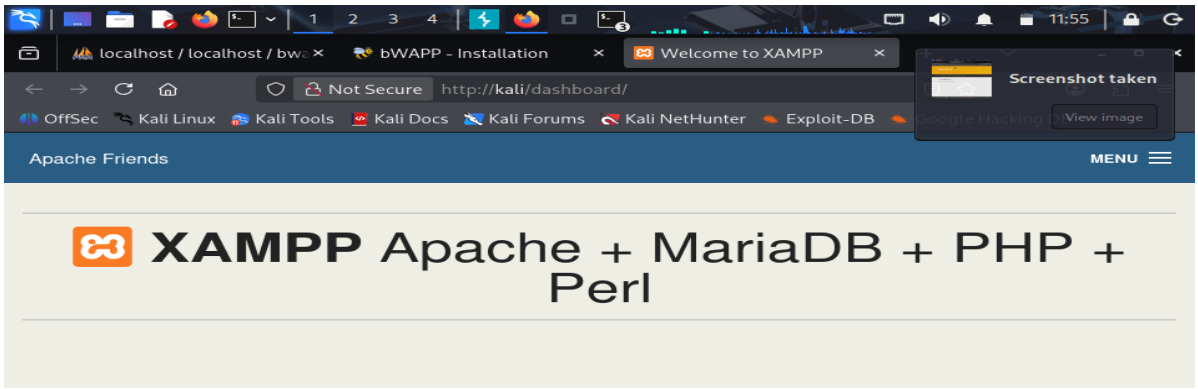
Hydra shows successful credentials like:

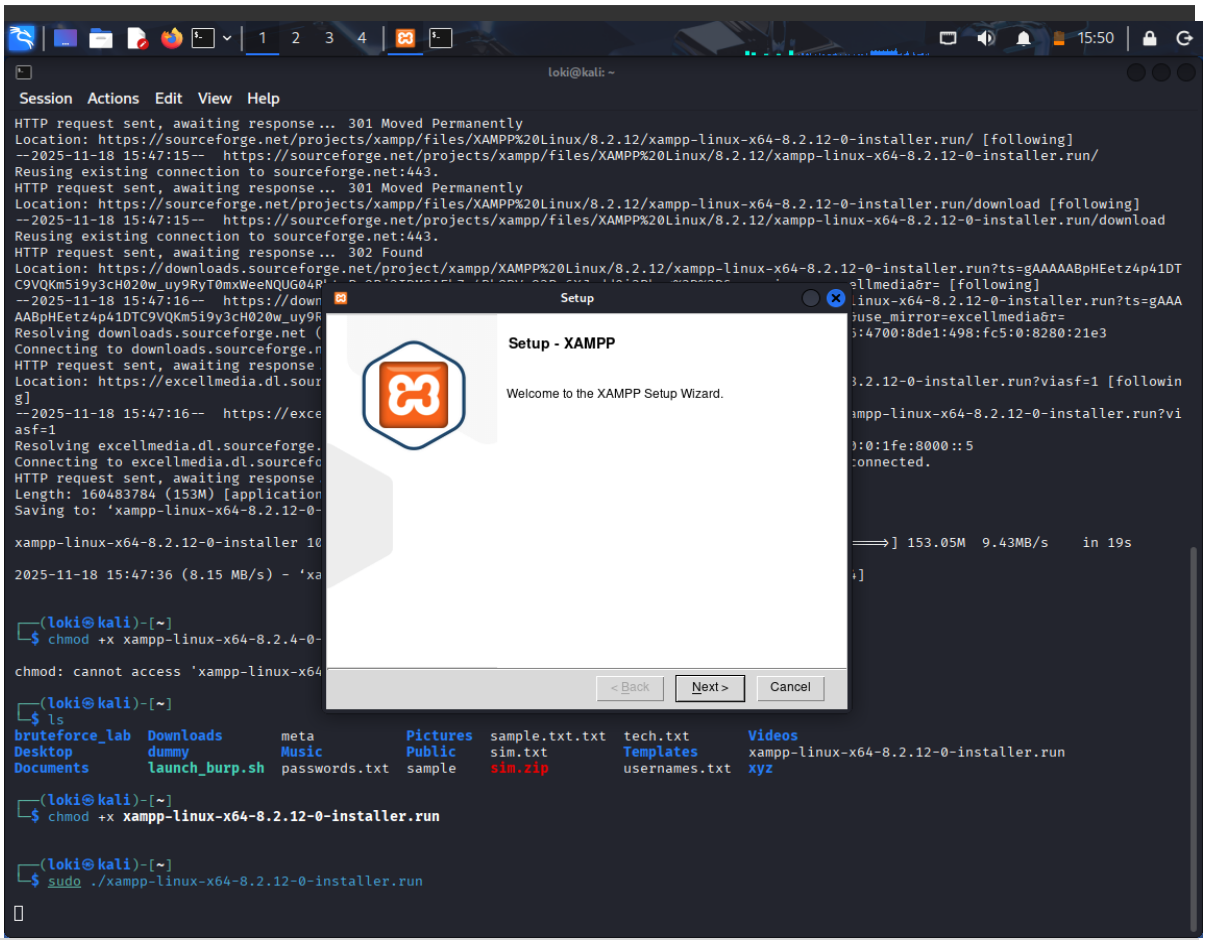
```
[80][http-post-form] host: 127.0.0.1   login: bee   password: bug
1 valid password found
```

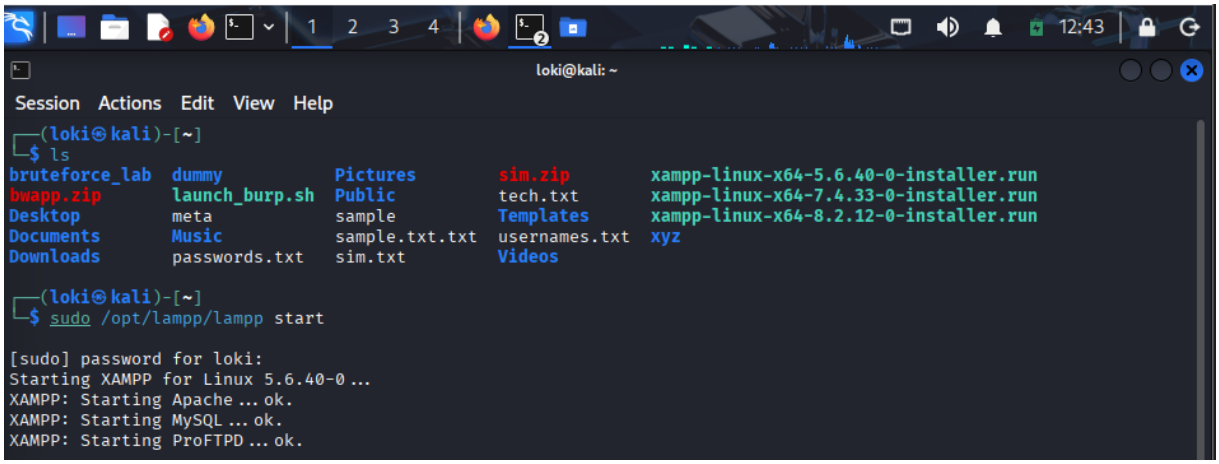
Hydra finishes the attack much faster than Burp Suite because it uses parallel execution threads

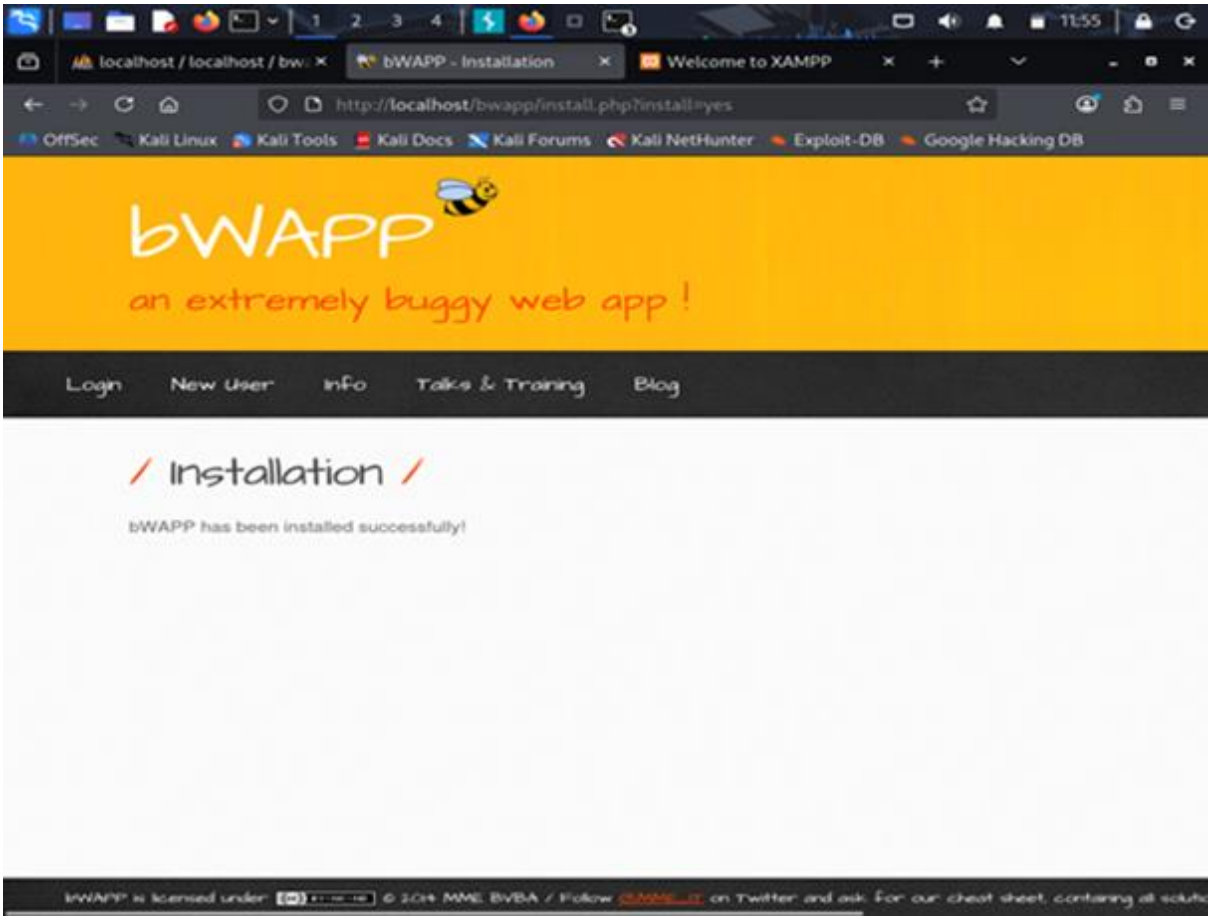
11. OUTPUTS & SCREENSHOTS

1. XAMPP & bWAPP Setup (Environment Setup)

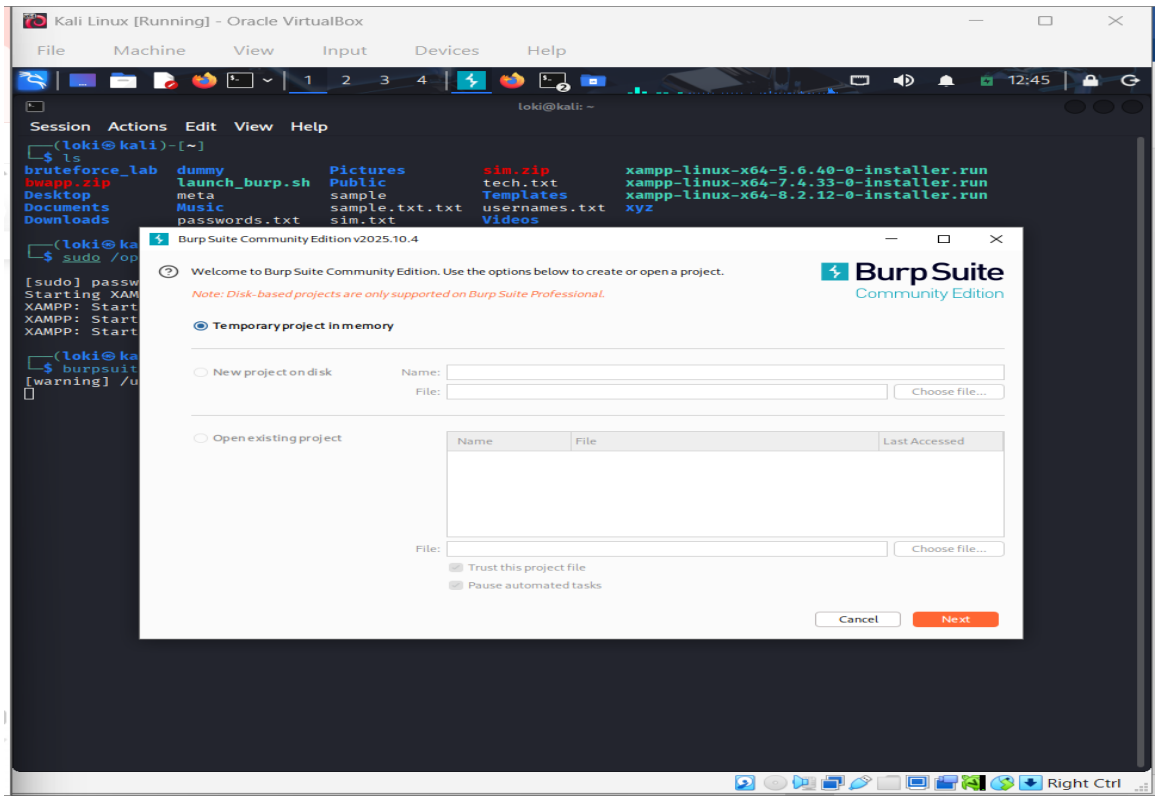


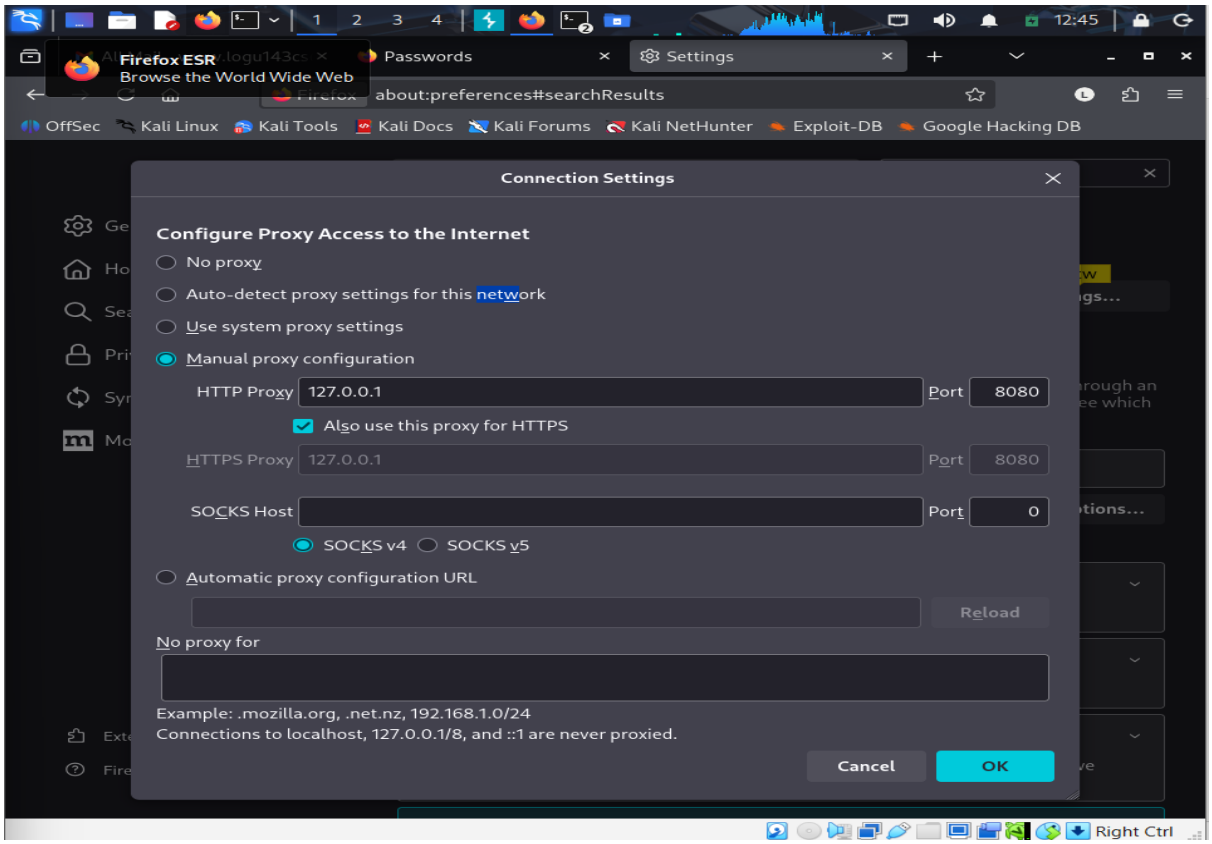


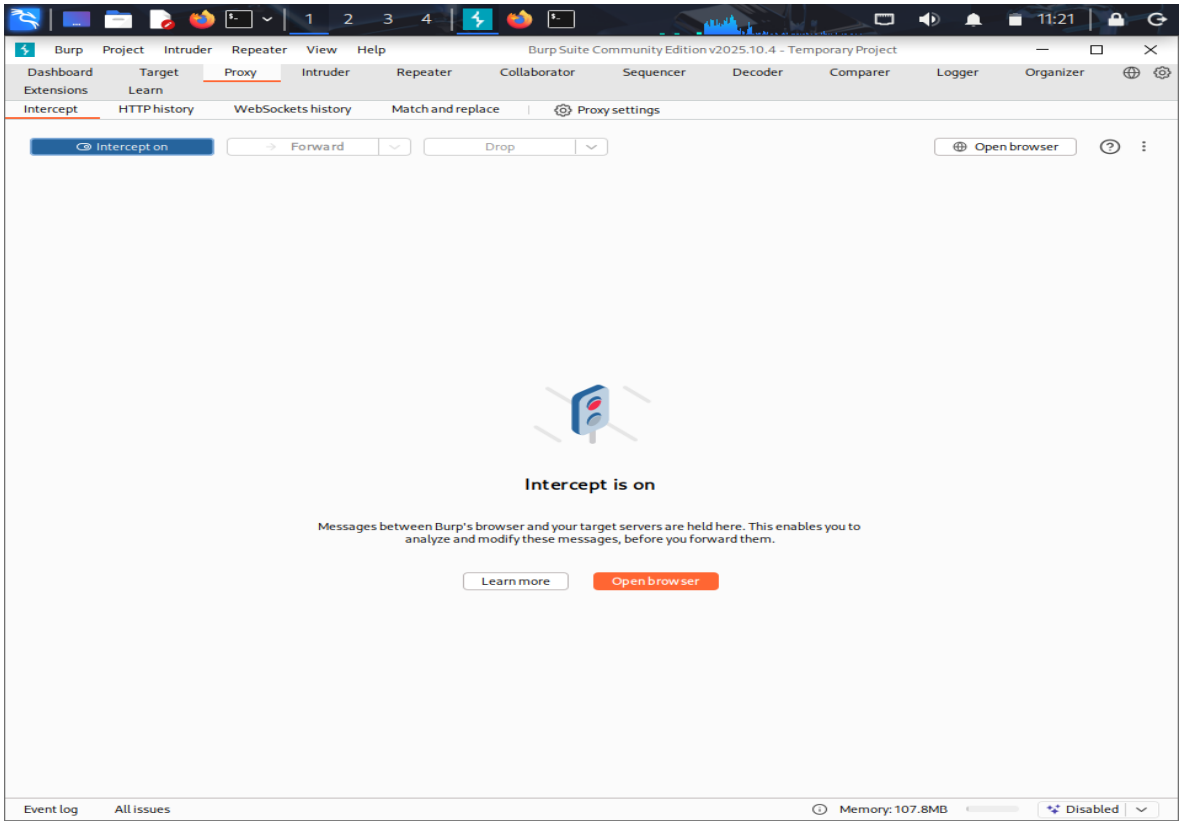

loki@kali: ~
Session Actions Edit View Help
(loki㉿kali)-[~]
$ ls
bruteforce_lab dummy Pictures sim.zip xampp-linux-x64-5.6.40-0-installer.run
bwapp.zip launch_burp.sh Public tech.txt xampp-linux-x64-7.4.33-0-installer.run
Desktop meta sample Templates xampp-linux-x64-8.2.12-0-installer.run
Documents Music sample.txt.txt usernames.txt xyz
Downloads passwords.txt sim.txt Videos
(loki㉿kali)-[~]
$ sudo /opt/lampp/lampp start
[sudo] password for loki:
Starting XAMPP for Linux 5.6.40-0 ...
XAMPP: Starting Apache ... ok.
XAMPP: Starting MySQL ... ok.
XAMPP: Starting ProFTPD ... ok.

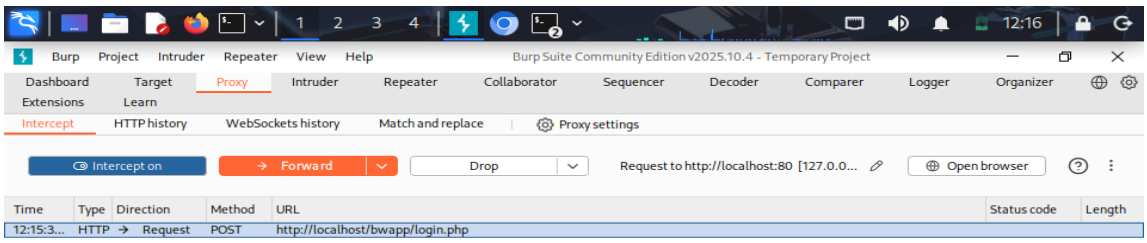

bWAPP - Installation
Welcome to XAMPP
http://localhost/bwapp/install.php?install=yes
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
bWAPP
an extremely buggy web app !
Login New User Info Talks & Training Blog
/ Installation /
bWAPP has been installed successfully!

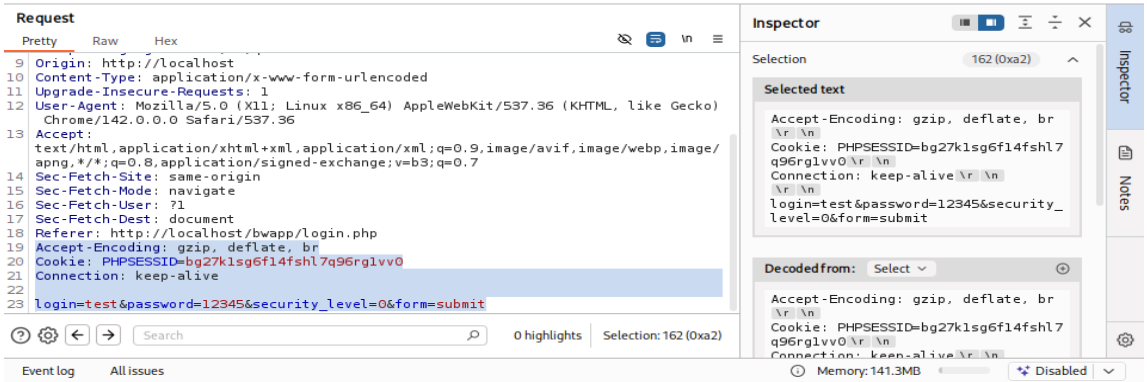
2. Burp Suite Setup



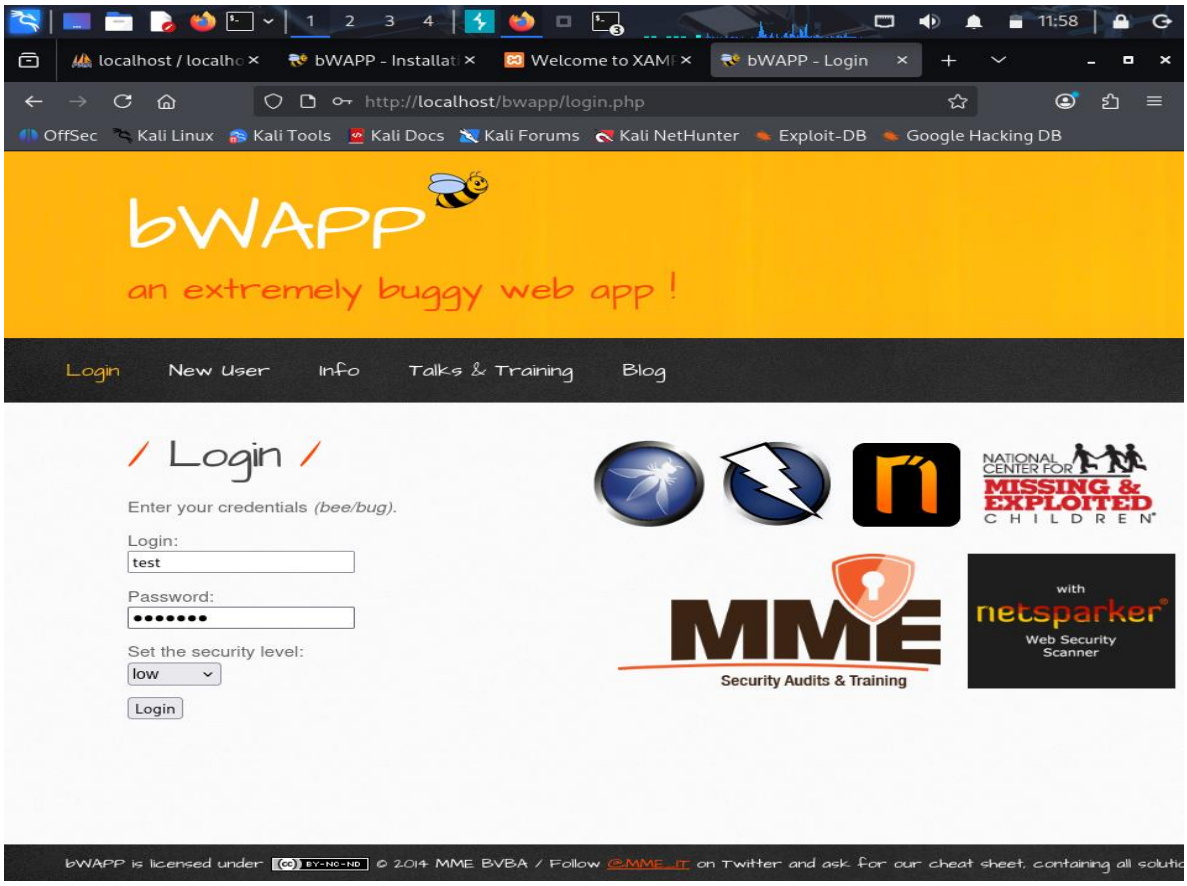


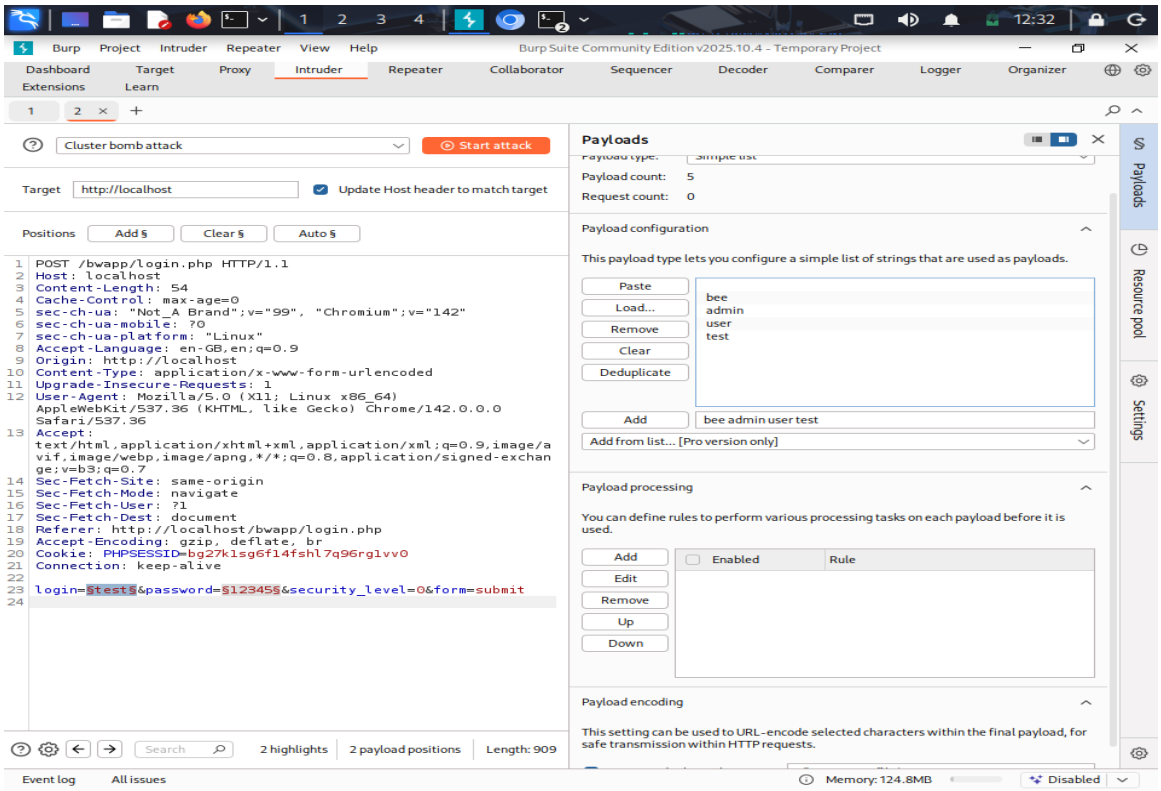

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.4 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer
Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Open browser
Intercept is on
Messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.
Learn more
Open browser
Event log
All issues
Memory: 107.8MB
Disabled

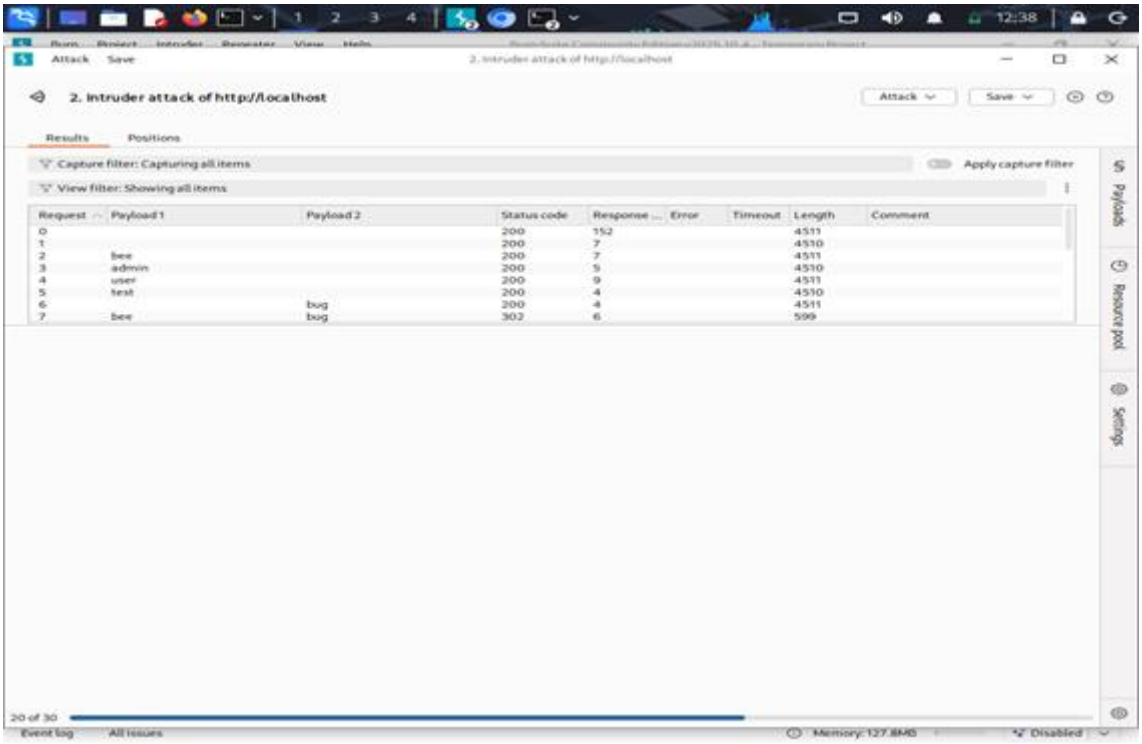

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.4 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer
Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Request to http://localhost:80 [127.0.0...
Open browser
Time Type Direction Method URL Status code Length
12:15:3... HTTP Request POST http://localhost/bwapp/login.php

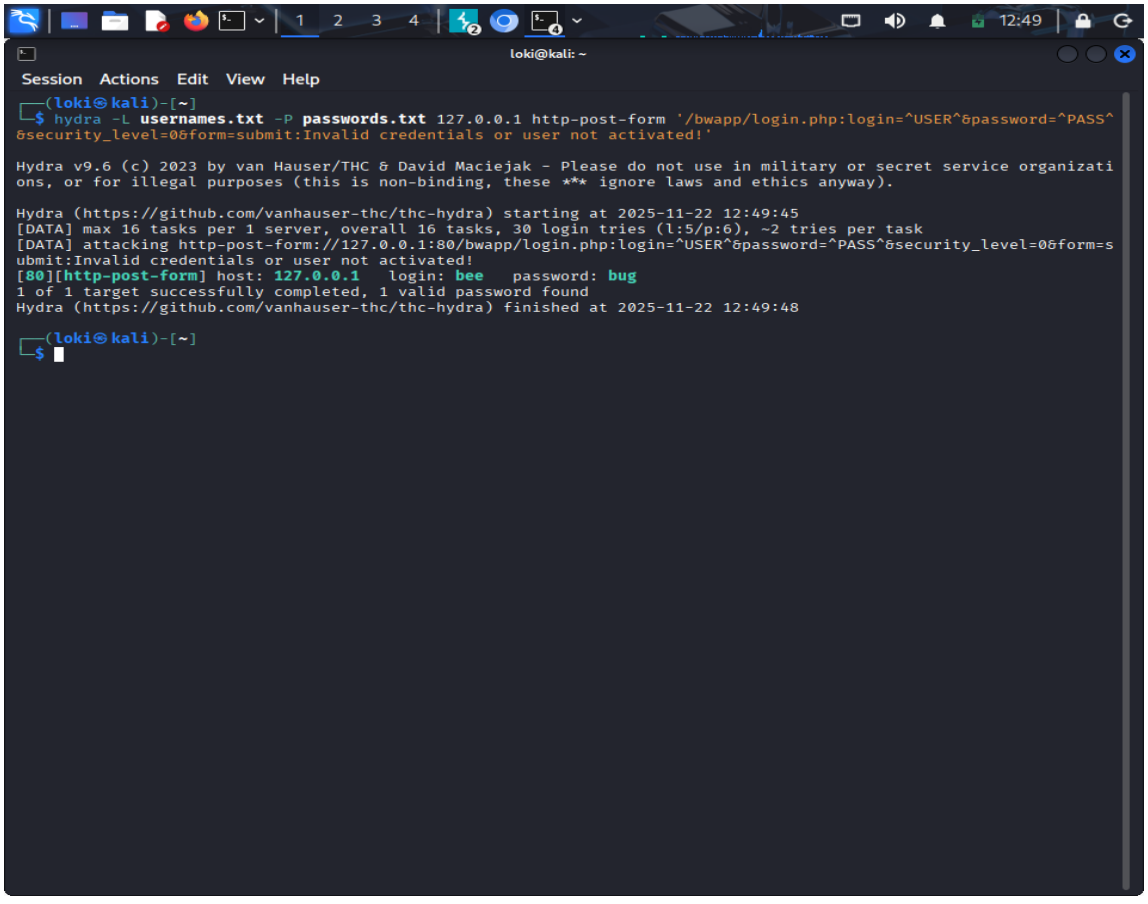

Request
Pretty Raw Hex
9 Origin: http://localhost
10 Content-Type: application/x-www-form-urlencoded
11 Upgrade-Insecure-Requests: 1
12 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: http://localhost/bwapp/login.php
19 Accept-Encoding: gzip, deflate, br
20 Cookie: PHPSESSID=bg27k1sg6f14fshl7q96rg1vv0
21 Connection: keep-alive
22
23 login=test&password=12345&security_level=0&form=submit
Search
0 highlights
Selection: 162 (0xa2)
Inspector
Selection 162 (0xa2)
Selected text
Accept-Encoding: gzip, deflate, br \r \n
Cookie: PHPSESSID=bg27k1sg6f14fshl7q96rg1vv0 \r \n
Connection: keep-alive \r \n \r \n
login=test&password=12345&security_level=0&form=submit
Decoded from: Select
Accept-Encoding: gzip, deflate, br \r \n
Cookie: PHPSESSID=bg27k1sg6f14fshl7q96rg1vv0 \r \n
Inspector
Notes
Event log
All issues
Memory: 141.3MB
Disabled

3. Burp Suite Intruder Attack (Cluster Bomb)

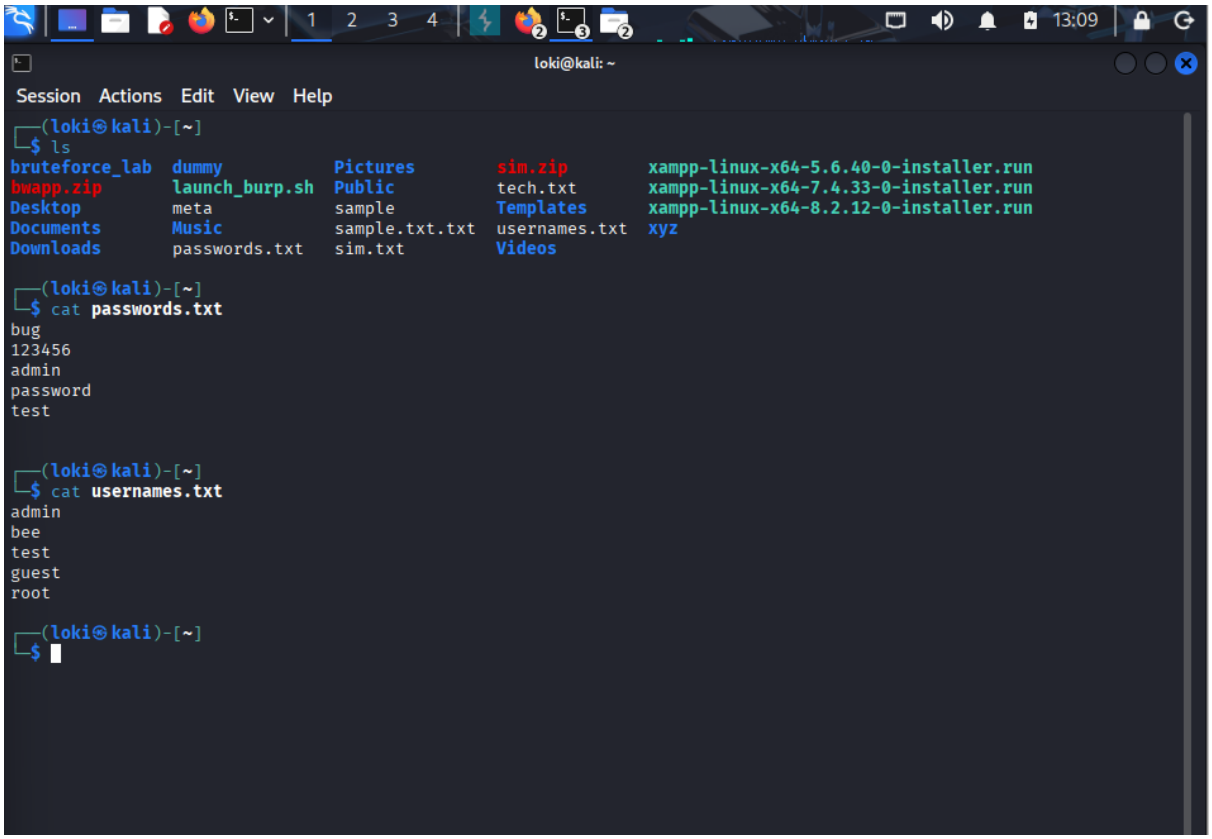


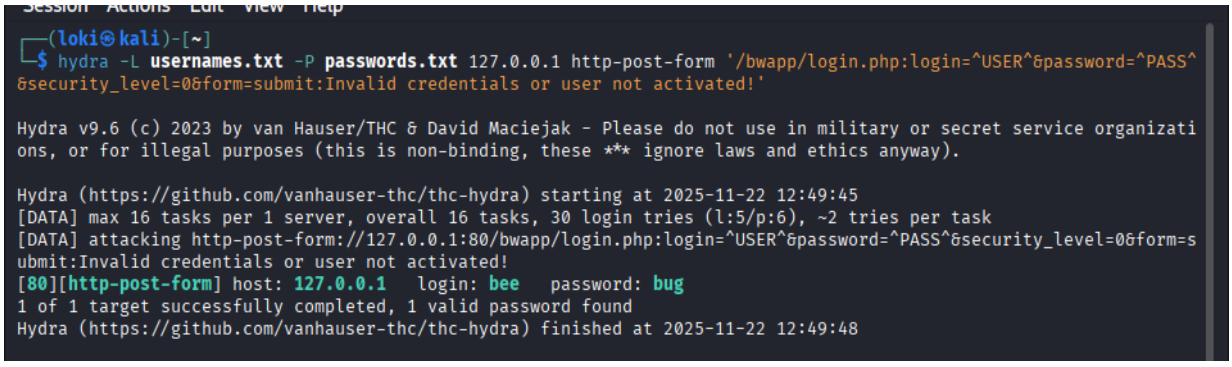



Attack Save
2. Intruder attack of http://localhost
Results Positions
Capture filter: Capturing all items
View filter: Showing all items
Apply capture filter
Request Payload1 Payload2 Status code Response ... Error Timeout Length Comment
0 200 152 4511
1 200 7 4510
2 bee 200 7 4511
3 admin 200 5 4510
4 user 200 9 4511
5 test 200 4 4510
6 bug 200 4 4511
7 bee bug 302 6 599
Payloads
Resource pool
Settings


loki@kali: ~
Session Actions Edit View Help
(loki㉿kali)-[~]
$ hydra -L usernames.txt -P passwords.txt 127.0.0.1 http-post-form '/bwapp/login.php:login=^USER^&password=^PASS^&security_level=0&form=submit:Invalid credentials or user not activated!'
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-22 12:49:45
[DATA] max 16 tasks per 1 server, overall 16 tasks, 30 login tries (l:5/p:6), ~2 tries per task
[DATA] attacking http-post-form://127.0.0.1:80/bwapp/login.php:login=^USER^&password=^PASS^&security_level=0&form=submit:Invalid credentials or user not activated!
[80][http-post-form] host: 127.0.0.1 login: bee password: bug
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-22 12:49:48
(loki㉿kali)-[~]
$

4. Hydra Brute-force Attack





12. MITIGATION TECHNIQUES

Effective security measures are crucial for protecting web applications from brute-force attacks. The following techniques help strengthen authentication systems and lower the risk of unauthorized access:

1. Account Lockout Policy

Lock a user account after a specific number of failed login attempts. This stops attackers from continually guessing passwords.

2. CAPTCHA Integration

Use CAPTCHA or reCAPTCHA during login attempts to block automated bots and brute-force tools like Hydra.

3. Rate Limiting

Limit the number of login requests allowed per IP address in a certain time period. This significantly slows down attackers.

4. Strong Password Policies

Require minimum password length, complexity rules, password expiration, and password history policies to prevent easy guessing.

5. Multi-Factor Authentication (MFA)

Add another layer of security, like an OTP, email verification, or authenticator apps, to ensure that even if the password is compromised, access is denied.

6. IP Blocking and Firewall Rules

Automatically block suspicious IP addresses or use tools like Fail2Ban to detect and stop repeated login failures.

7. Secure Error Messages

Avoid revealing whether the username or password is incorrect. Generic error messages stop attackers from confirming valid usernames.

8. HTTPS Encryption

Make sure login requests are sent through HTTPS/TLS to protect credentials from being intercepted during man-in-the-middle attacks.

9. Intrusion Detection & Monitoring

Use IDS/IPS tools like Snort or Suricata to watch for unusual login activity and alert administrators in real time.

10. Account Activity Logging

Keep detailed logs of login attempts so security teams can spot patterns and detect brute-force attacks early.

13. ADVANTAGES & LIMITATIONS

Advantages

1. Easy and Quick to Set Up

- The entire environment, including Kali Linux, XAMPP, and bWAPP, is very simple to configure.
- No complicated dependencies are needed. This makes it ideal for students and beginners in cybersecurity.
- Tools like Burp Suite and Hydra come pre-installed in most Kali Linux versions, which saves setup time.

2. Demonstrates Real Brute-Force Attack Behavior

- The project gives a practical demonstration of how attackers repeatedly try different username-password combinations.
- Burp Suite helps visualize how login requests are intercepted and modified.
- Response differences, such as status codes, content length, and redirects, clearly show how successful brute-force attempts are identified.

3. Helps Understand Authentication Weaknesses

- The project shows why weak or default passwords are dangerous.
- It highlights the impact of missing protections like CAPTCHA, MFA, lockout policies, and rate-limiting.
- This helps students learn how insecure coding practices can lead to real security breaches.

4. Hydra Provides High Speed and Multi-Threaded Operations

- Hydra can perform hundreds of login attempts per second because of multi-threading.
- It demonstrates how attackers automate large-scale brute-force attacks using efficient tools.
- This makes it clear why modern systems must use strong defensive measures.

5. Completely Safe and Legal Testing Environment

- bWAPP is designed for ethical hacking practice and contains controlled vulnerabilities.
- It ensures there is no damage to external systems or networks.
- This allows students to learn cybersecurity without legal risks.

Limitations

1. Works Only in a Controlled Vulnerable System

- bWAPP behaves differently from real-world secure applications.
- Real applications usually have stronger defenses, such as MFA, device verification, or AI-based login monitoring.
- The success rate of attacks in bWAPP is higher and not always representative of real-world environments.

2. Burp Suite Community Edition Is Slow

- The free version of Burp Suite limits Intruder attack speed significantly.
- Larger username and password lists take much longer to process.
- It does not show the high-speed brute-force capabilities used by attackers with advanced tools.

3. bWAPP Uses Outdated PHP Modules

- bWAPP runs on older PHP (5.x), which is not commonly used today.
- Modern authentication systems rely on advanced methods like token-based login, OAuth, JWT, and encryption, which bWAPP does not support.
- As a result, the training environment lacks modern security features.

4. No Realistic Anti-Brute-Force Protection

- bWAPP does not have real lockout mechanisms or IP blocking.
- There is no CAPTCHA or rate-limiting system.
- This makes brute-force attacks easier compared to real systems.

14. CONCLUSION

This project clearly showed how brute-force attacks can target a weak web application using Burp Suite and Hydra. By setting up bWAPP on XAMPP in a Kali Linux environment, the experiment created a controlled and safe platform to understand the ways attackers can exploit authentication systems. Burp Suite enabled a close look at HTTP POST requests, helped identify login parameters, and allowed the execution of a Cluster Bomb attack. Meanwhile, Hydra sped up brute-force attempts to quickly discover valid credentials. These hands-on exercises made it easier to grasp how brute-force attacks work and highlighted the need to analyze response patterns to recognize successful login attempts.

The results of this project underscore the urgent need for strong security measures in web applications. Weak passwords, unlimited login attempts, absence of multi-factor authentication, and poor error handling were identified as major weaknesses that attackers could take advantage of. By conducting both manual and automated attacks, the project stressed the importance of proactive defense strategies like CAPTCHA, rate limiting, account lockouts, and strong password policies. Overall, this study provided valuable hands-on experience, insights into authentication flaws, and emphasized the importance of applying effective security practices to safeguard web applications against brute-force attacks.

15. REFERENCES

- OWASP Foundation, *Brute Force Attack*, https://owasp.org/www-community/attacks/Brute_force_attack – Accessed 2025.
- PortSwigger, *Burp Suite Documentation*, https://portswigger.net/burp/documentation – Accessed 2025.
- THC Hydra, *Hydra: Fast Network Login Cracker*, https://github.com/vanhauser-thc/thc-hydra – Accessed 2025.
- bWAPP Project, *Buggy Web Application for Security Testing*, http://www.itsecgames.com/ – Accessed 2025.
- XAMPP Official Documentation, *Apache, MySQL, PHP Setup Guide*, https://www.apachefriends.org/ – Accessed 2025.
- Kali Linux Documentation, https://www.kali.org/docs/ – Accessed 2025.
- Rouse, M., "Brute-force Attack," *TechTarget*, https://www.techtarget.com/definition/brute-force-attack – Accessed 2025.
- Gupta, A., *Ethical Hacking and Penetration Testing Guide*, McGraw Hill Education, 2021.
- Goyal, S., *Web Application Security with Burp Suite*, Packt Publishing, 2020.
- Cybersecurity and Infrastructure Security Agency (CISA), *Authentication Best Practices*, https://www.cisa.gov/ – Accessed 2025.
- Singh, R., *Hands-On Penetration Testing with Kali Linux*, BPB Publications, 2022.
- Kumar, V., *Mastering Web Application Security*, Wiley India, 2021.
- Mitnick, K., & Simon, W., *The Art of Intrusion*, Wiley, 2011.
- Hadnagy, C., *Social Engineering: The Science of Human Hacking*, Wiley, 2018.