**CYBERSECURITY INTERNSHIP REPORT**

**TASK 1: Web Application Security Testing**

**Platform Used:** DVWA (Damn Vulnerable Web Application)

**Security Level:** Low

**Tools Used:** Browser (Manual Testing)

**✅ 1. SQL Injection**

**Description**:  
SQL Injection is a vulnerability where malicious SQL queries are injected into input fields to access or manipulate the database.

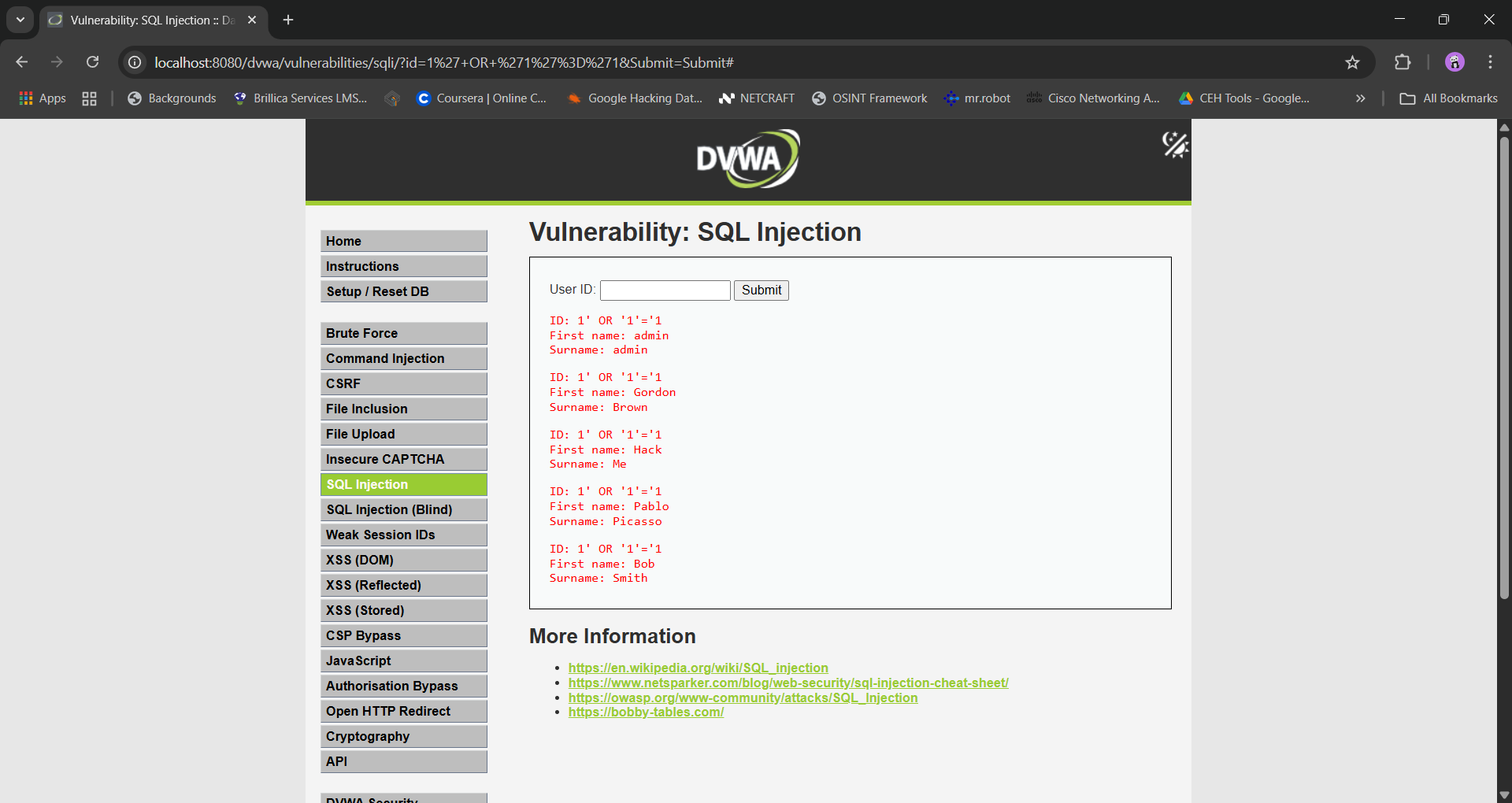
**Steps**:

1. Navigated to DVWA → SQL Injection module.
2. Entered payload in the user ID field:
3. 1’ OR ‘1’=’1
4. Clicked Submit.

**Result**:  
Database returned multiple user details, confirming SQL Injection was successful.

**Mitigation**:

* Use parameterized queries (Prepared Statements).
* Input validation and sanitization.
* Web Application Firewall (WAF).



## ✅ ****2. Command Injection****

**Description**:  
Command injection occurs when system-level commands can be executed via user input.

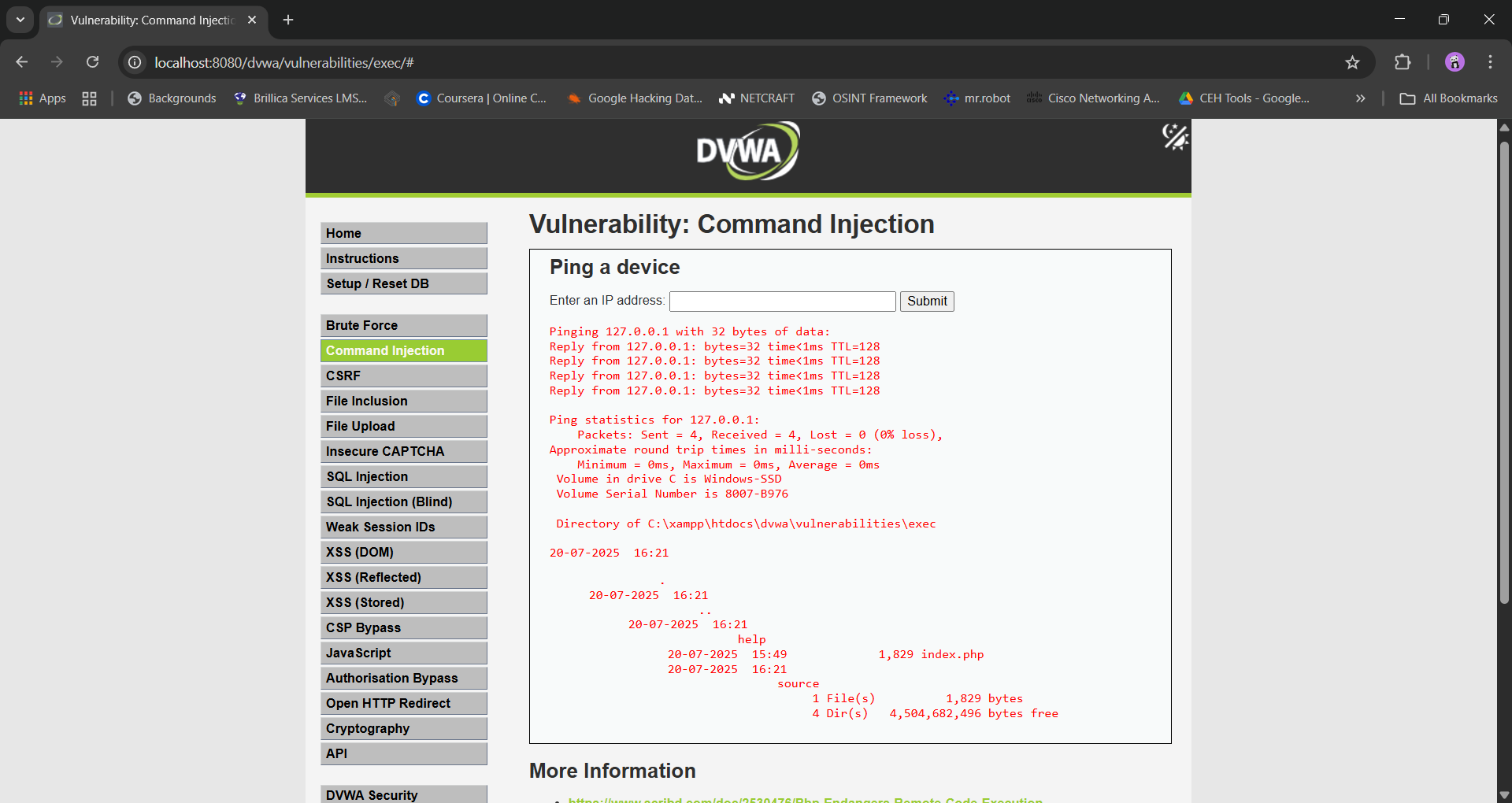
**Steps**:

1. Navigated to DVWA → Command Injection module.
2. Entered IP address with command injection payload:
3. 127.0.0.1 && dir
4. Clicked Submit.

**Result**:  
Output showed current server username, confirming command execution.

**Mitigation**:

* Avoid using system calls with user input.
* Whitelist input and sanitize.
* Use safe APIs or shell wrappers.



## ✅ ****3. Brute Force Attack****

**Description**:  
Brute Force is an attack where multiple username-password combinations are tried to gain unauthorized access.

**Steps**:

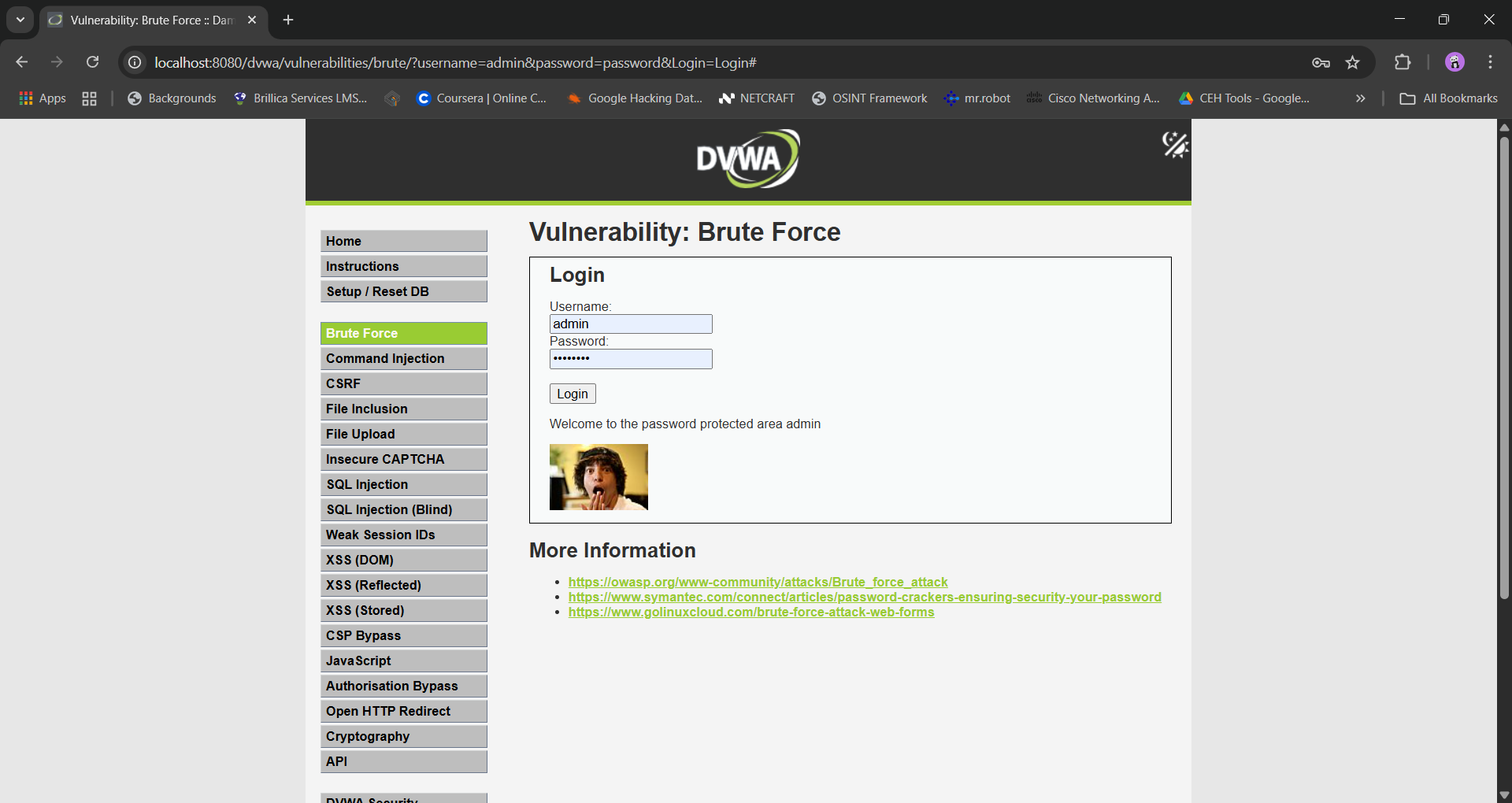
1. Navigated to DVWA → Brute Force module.
2. Entered admin as username.
3. Tried multiple common passwords manually.
4. Login successful with password: <password>

**Result**:  
Message displayed:

**“Welcome to the password protected area admin”**

**Mitigation**:

* Limit login attempts.
* Use CAPTCHA.
* Multi-factor authentication (MFA).
* Account lockout after failed attempts.



## 📝 ****Conclusion****:

DVWA helped identify and exploit common web application vulnerabilities manually. These practical exercises increase understanding of how insecure coding can be exploited and how to defend against such attacks.