VAPT

Vulnerability Assessment and Penetration Testing

PENETRATION TESTING??

An ethical drill to expose and fix hidden weaknesses.

Penetration testing is a controlled and authorised process of simulating real-world cyberattacks on a system, network, or application to identify security weaknesses before attackers can exploit them. It helps organisations find and fix vulnerabilities to improve their overall security

Significance in Ethical Hacking

1. Realistic Security Check – It shows how actual attackers could break into a system, rather than just pointing out theoretical flaws.
2. Risk Identification – Helps in detecting vulnerabilities such as weak passwords, misconfigurations, or unpatched software.
3. Prevention of Attacks – By finding and fixing issues early.
4. Builds Trust – Demonstrates to clients, customers, and stakeholders that the organisation takes security seriously.

STEPS INVOLVED IN PEN TESTING

- **Plan & Authorize** — Define scope, rules, and get written permission.
- **Recon & Scan** — Gather public info and scan targets for live hosts and open services.
- **Find & Verify** — Identify vulnerabilities and safely verify them with non-destructive proofs.
- **Assess Impact & Cleanup** — Determine business impact from successful tests, then remove test artifacts.
- **Report & Re-test** — Deliver prioritized remediation guidance and verify fixes after remediation.

Tools Used in Project

TOOLS	TECHNOLOGY
DVWA (Damn Vulnerable Web App):	Target web application containing pre-built vulnerabilities
XAMPP (Apache + MySQL):	Localhost server to run DVWA
Burp Suite Community Edition:	Proxy tool used for interception, testing, and automation
Kali Linux:	Base operating system used for testing and hosting DVWA

METHODOLOGY:

1. DVWA Setup:
Installed DVWA using XAMPP and configured the database and file permissions.

2. Burp Suite Configuration:
Launched Burp Suite Community Edition and configured the embedded browser to route traffic through Burp Proxy.

3. DVWA Security Level:
Set the DVWA security level to Low to allow unrestricted testing and exploitation.

4. Reconnaissance and Manual Testing:
Manually navigated DVWA modules (SQLi, XSS, File Upload, etc.) to identify vulnerable points.

5. Vulnerability Exploitation:
Performed attacks like SQL Injection, Cross-Site Scripting (XSS), Command Injection, CSRF, etc., using Burp and browser payloads.

6. Documentation:
Recorded each vulnerability with screenshots, payloads, and mitigation steps for the final report.

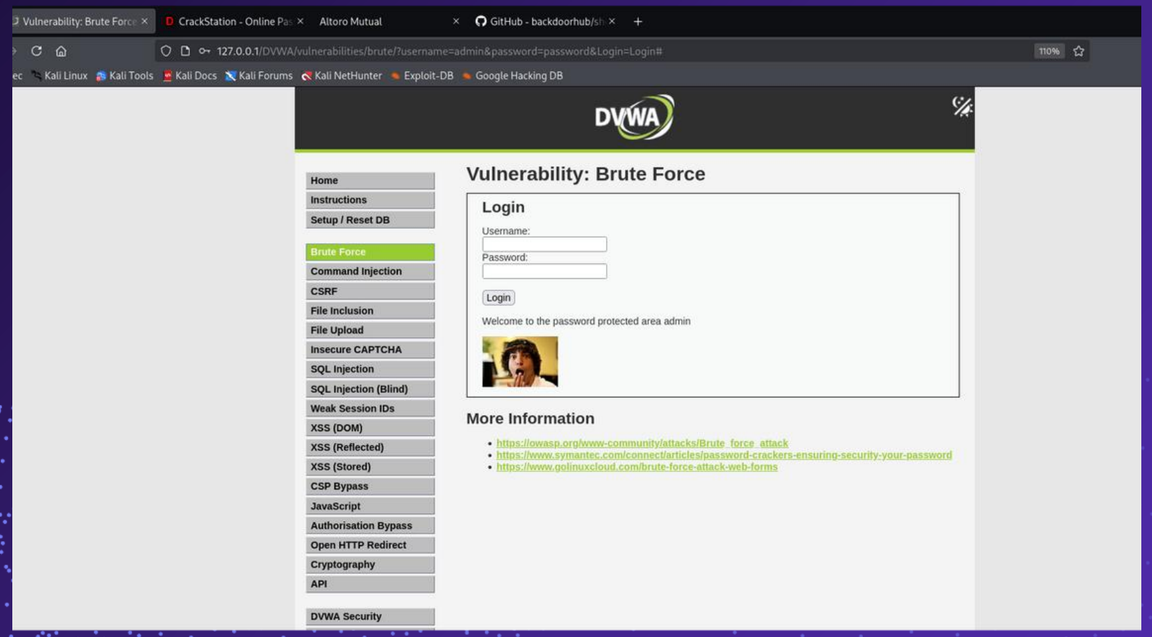
Vulnerabilities in DVWA

#	Vulnerability Type
1	Brute Force
2	Command Injection
3	SQL Injection
4	SQL Injection (Blind)
5	XSS (DOM-Based)
6	XSS (Reflected)
7	XSS (Stored)
8	CSP Bypass
9	Authorization Bypass
10	Open HTTP Redirect
11	Cryptography Flaws
12	JavaScript/Client-side Bypass
13	API Security (if enabled)

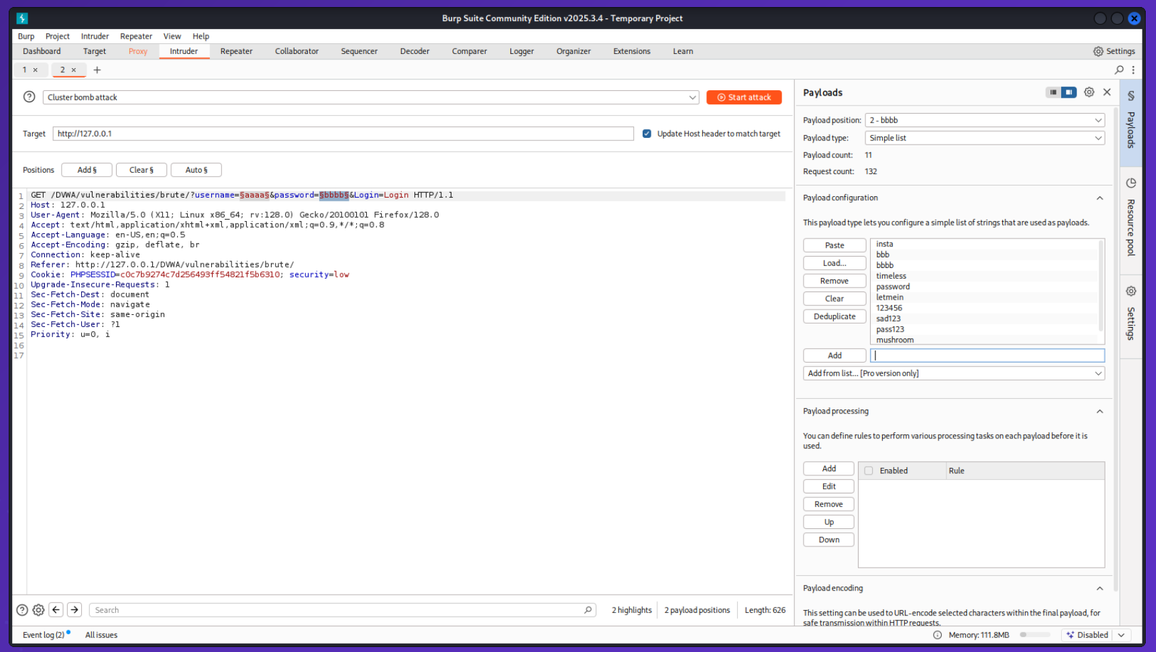
Tested Vulnerability 1: Brute Force

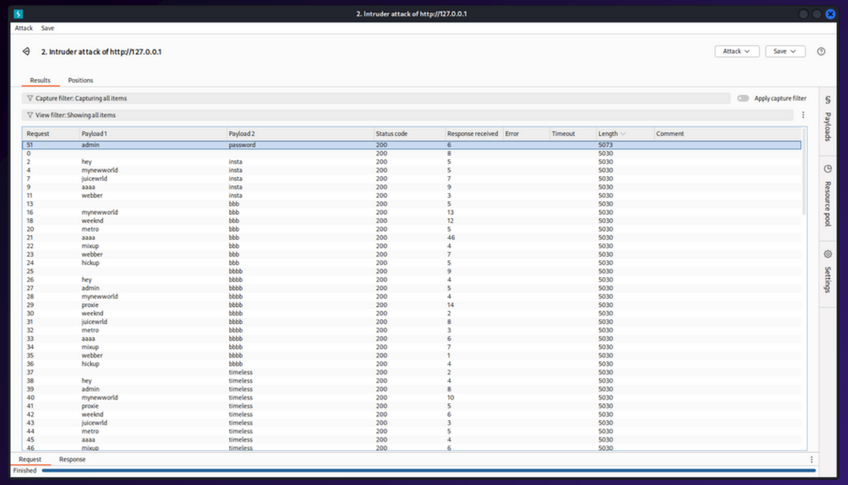
Brute forcing is a method of guessing passwords, PINs, or secret keys by systematically trying all possible combinations until the correct one is found.

Lab prep — Set up an isolated DVWA lab, confirm permissions.



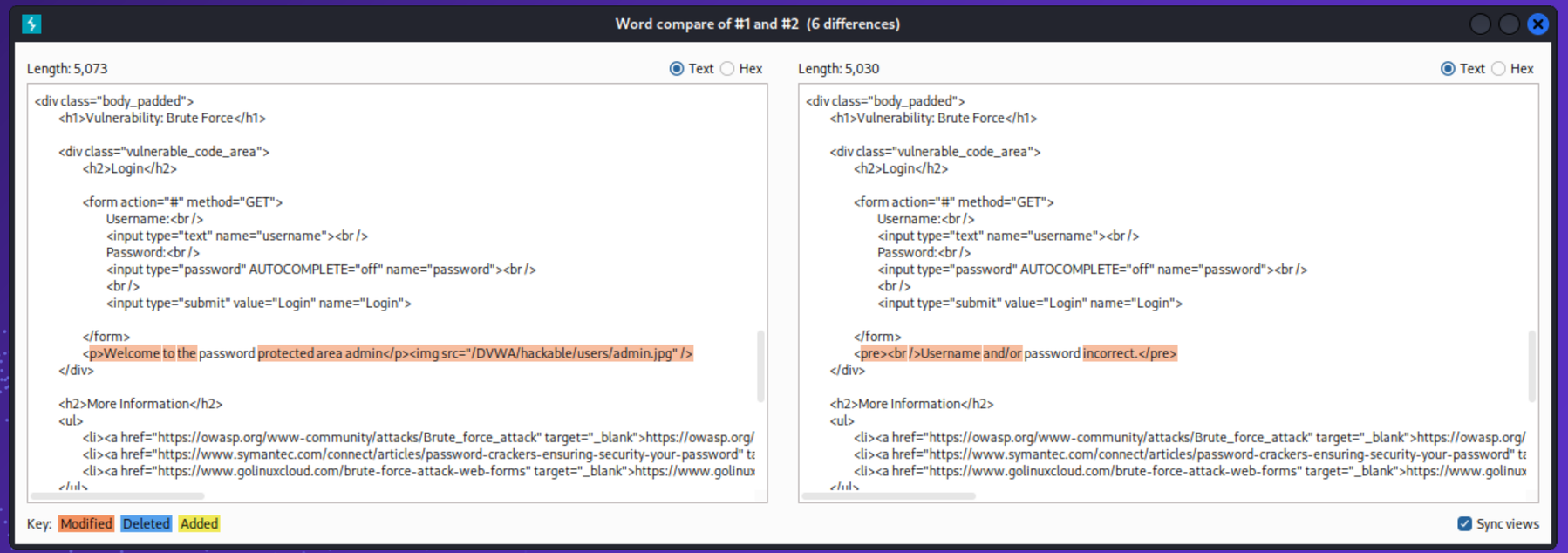
- Intercept login — Route browser traffic through Burp.
- Identify credential parameters — Observe which form fields carry username/password.



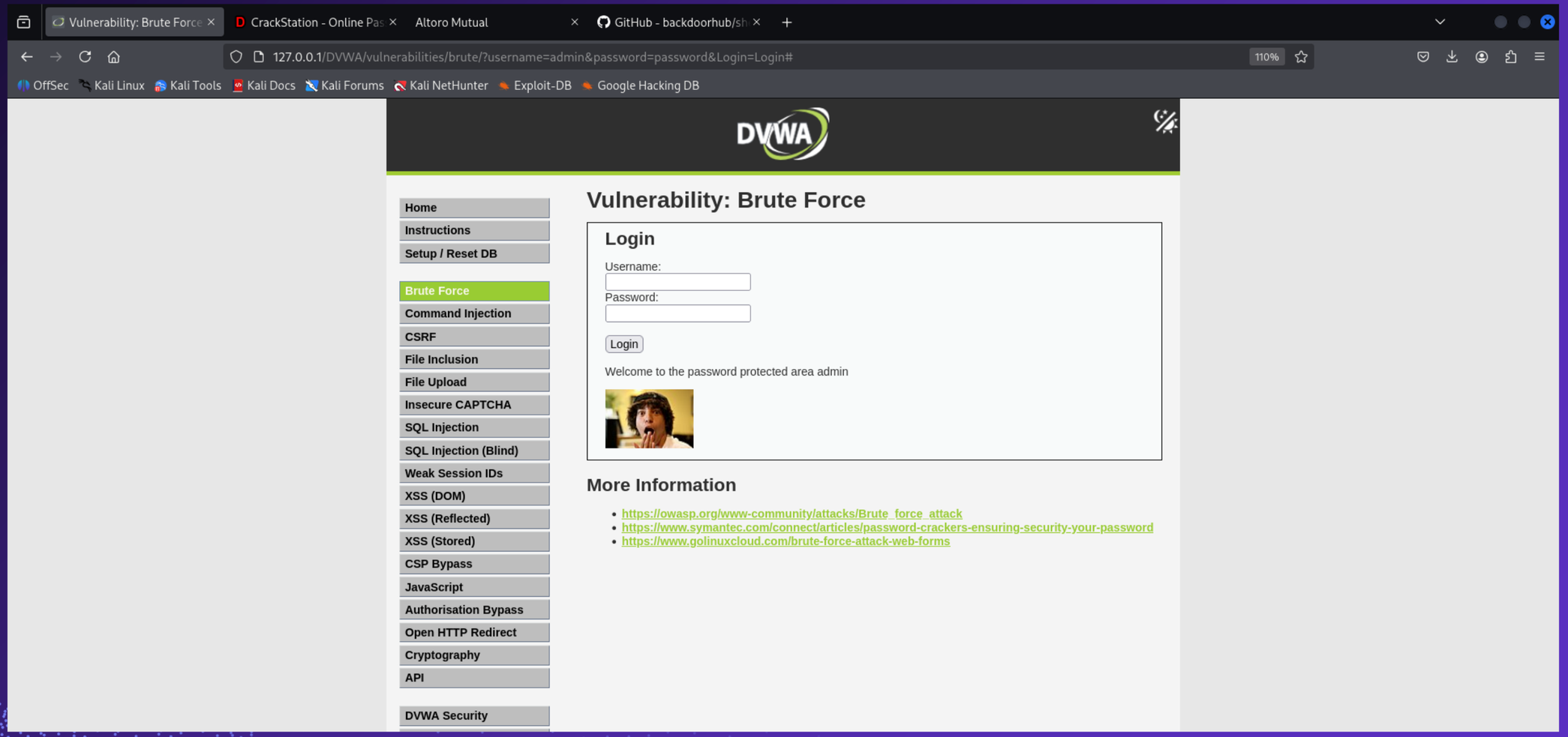


We will check for the parameters with the largest value of length and send it to the compararer

On comparing word to word, we will be able to reach our desired conclusion.



Once we have located the correct credentials from the comparer window, we will attempt to log in using the retrieved username and password.

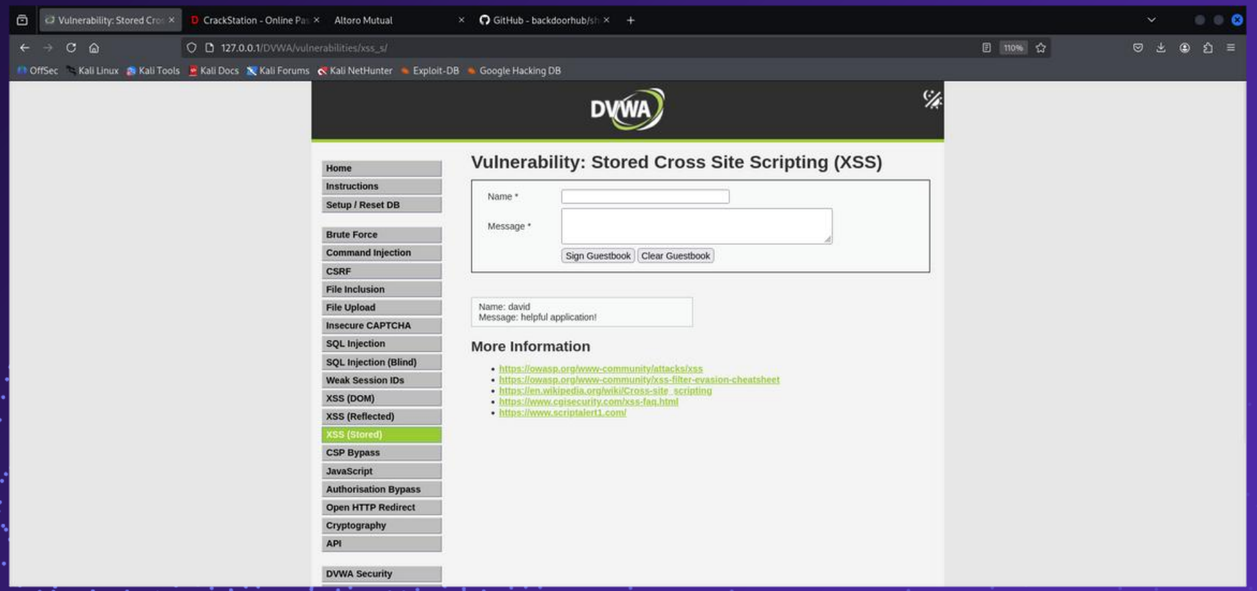


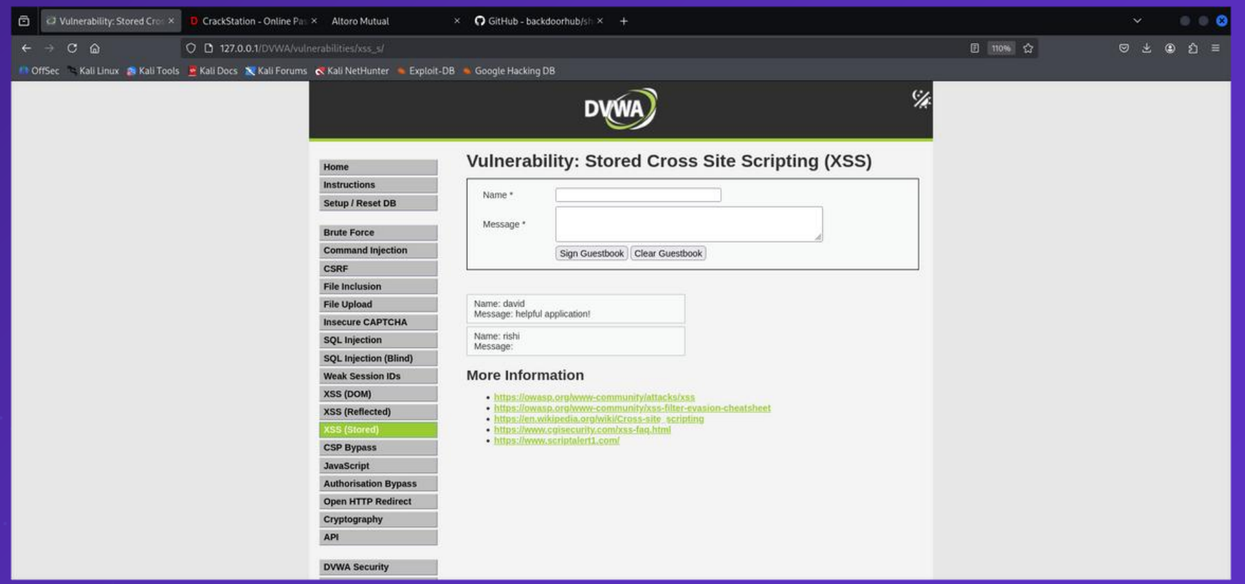
Tested Vulnerability 2: XSS(Stored)

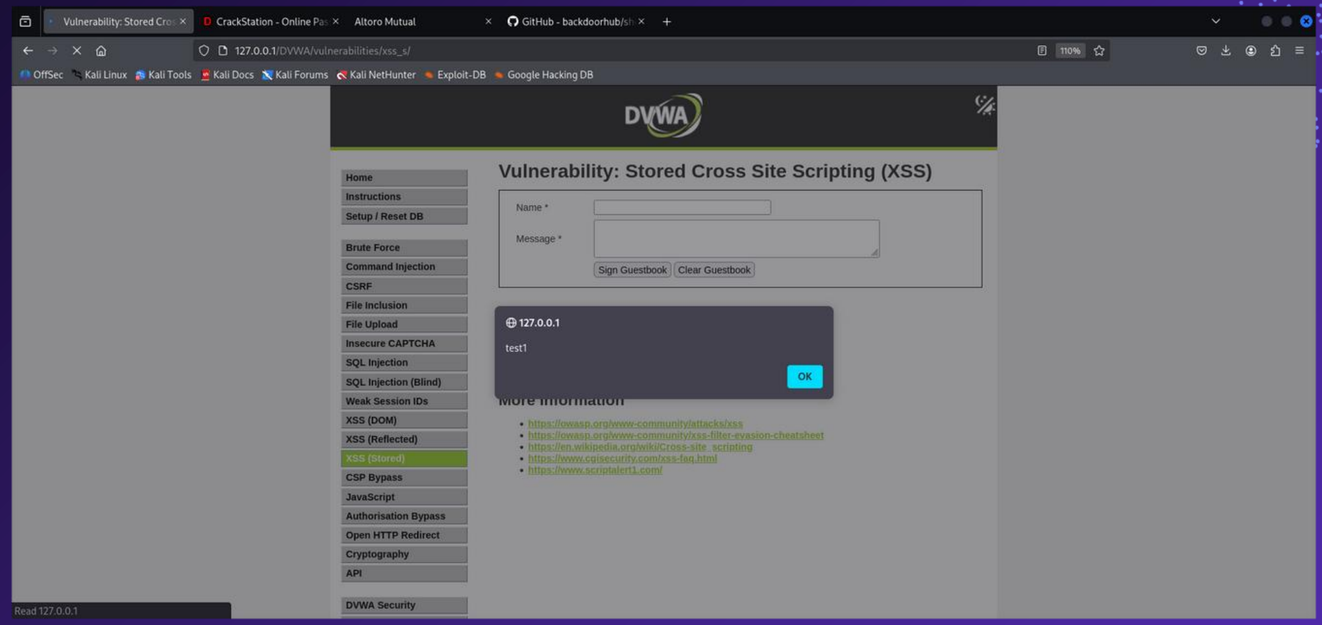
Stored XSS is a vulnerability where malicious scripts submitted by an attacker are saved on the server (usually in a database) and executed every time a user loads the affected page.

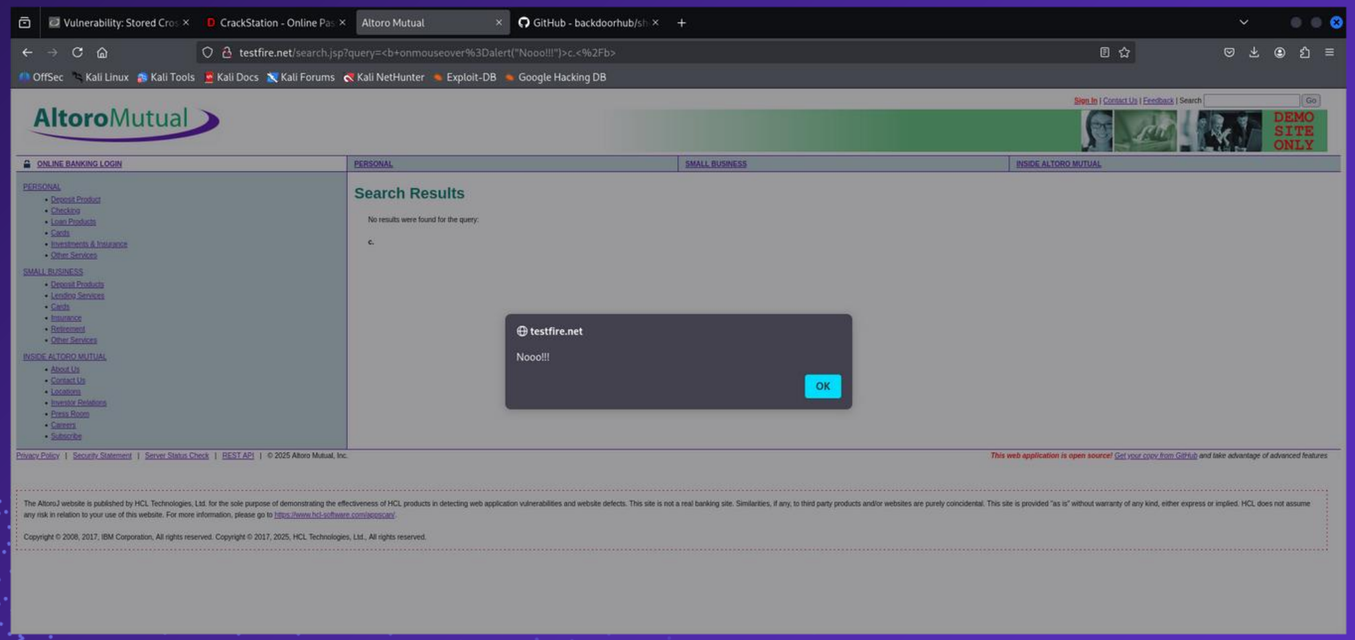
Workflow in DVWA:

1. Attacker submits a script via a vulnerable input form (like comments, messages, or feedback).
2. DVWA stores this script in its database.
3. Whenever a user or admin visits the page, the script runs automatically in their browser.






Vulnerability: Stored Cross Site Scripting (XSS)
Name *
Message *
Sign Guestbook
Clear Guestbook
127.0.0.1
test1
OK
More information
https://owasp.org/www-community/attacks/xss
https://owasp.org/www-community/xss-filter-evasion-cheatsheet
https://en.wikipedia.org/wiki/Cross-site_scripting
https://www.cgisecurity.com/xss-faq.html
https://www.scriptalert1.com/
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Open HTTP Redirect
Cryptography
API
DVWA Security

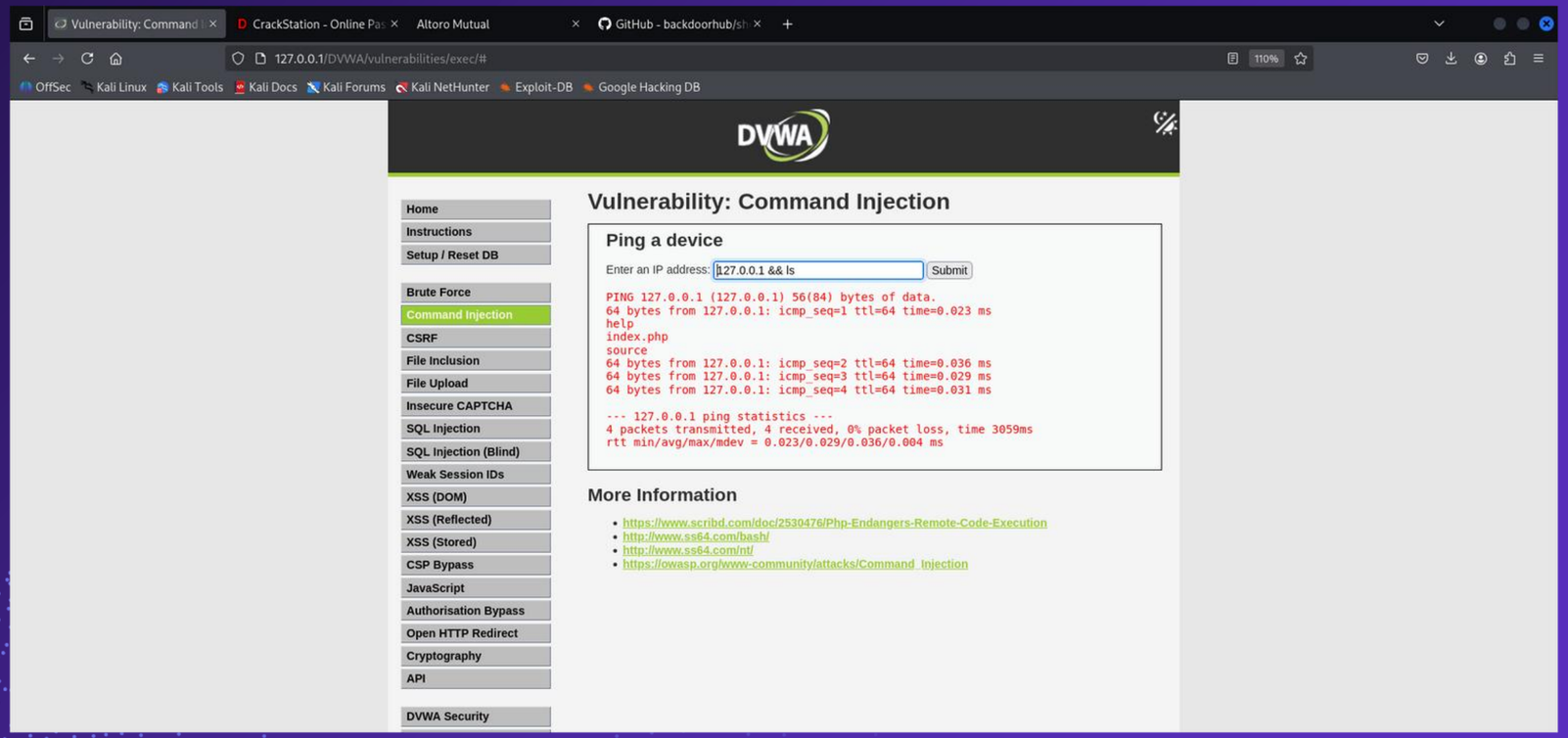

AltoroMutual
Search Results
No results were found for the query:
c.
testfire.net
Nooo!!!
OK

Tested Vulnerability 3: Command Injection

Command Injection is a vulnerability where an attacker can execute arbitrary system commands on the server through a vulnerable application input.

How it works in DVWA:

- DVWA has a vulnerable input form (e.g., ping a host).
- Attacker inputs malicious commands instead of normal input.
- The server executes these commands because it doesn't properly validate or sanitize input.

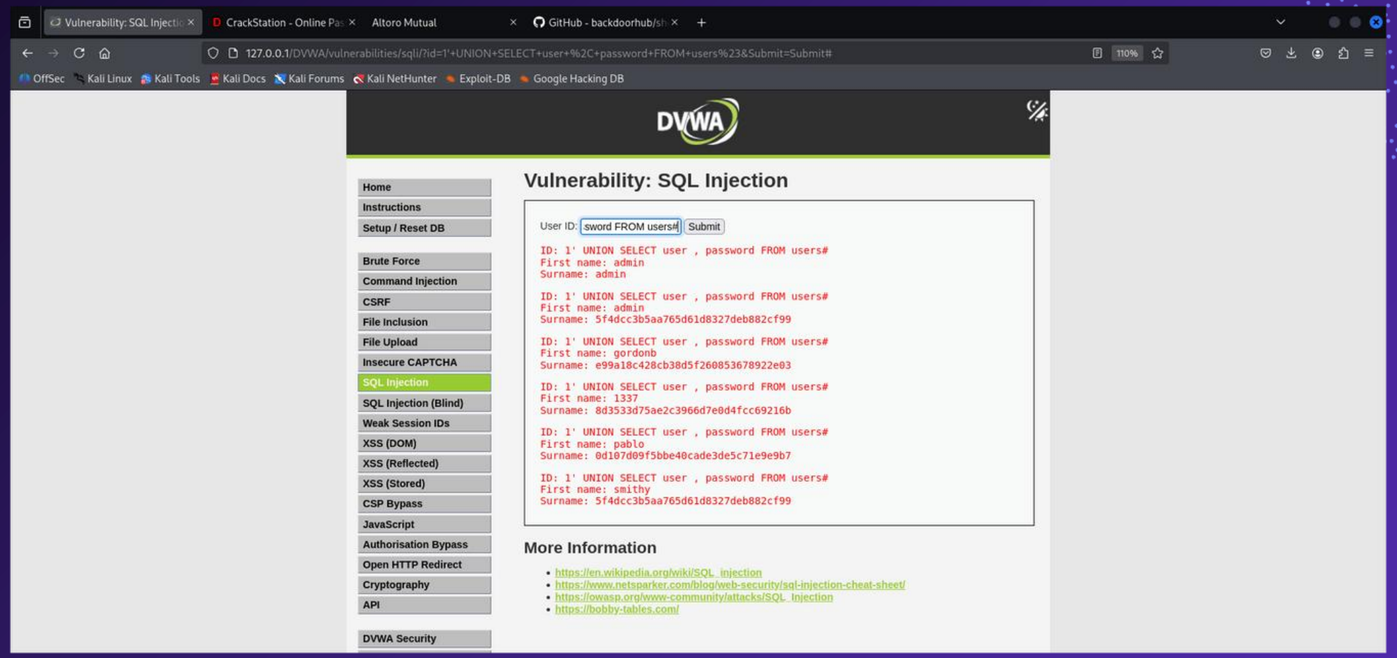


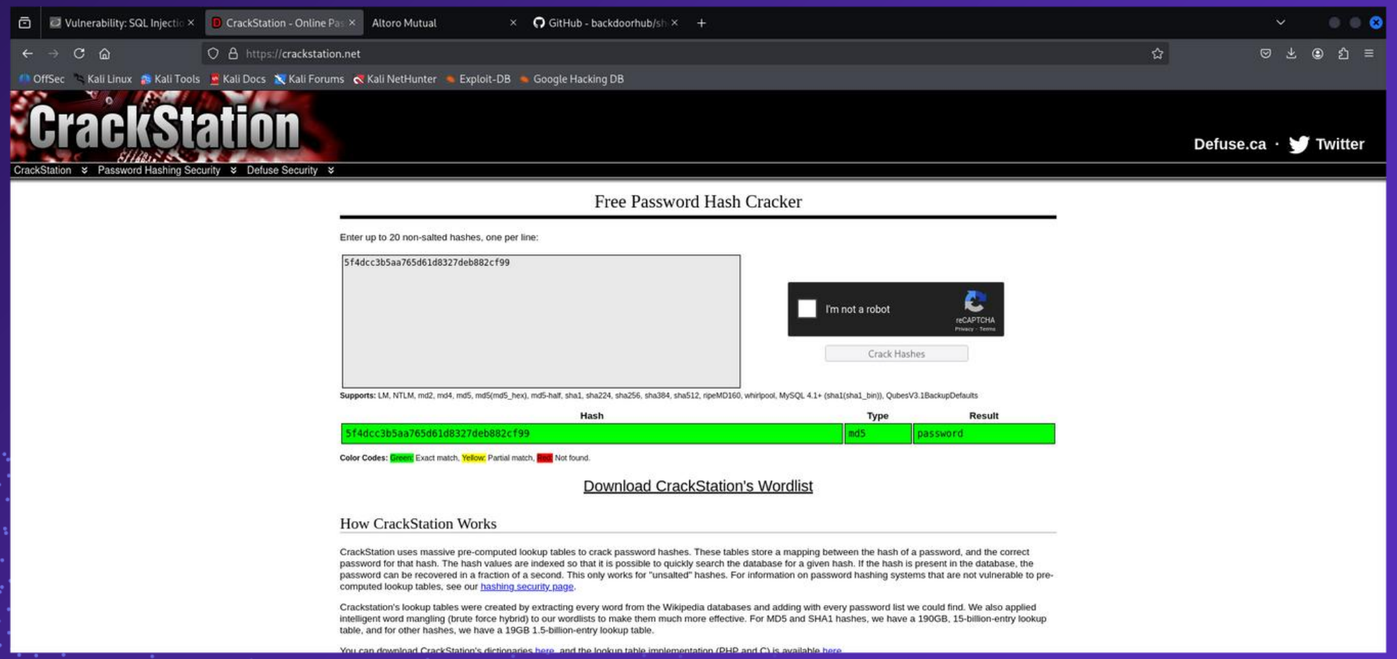
Tested Vulnerability 4: SQL Injection

SQL Injection (SQLi) is a web vulnerability where an attacker can manipulate a database query by injecting malicious SQL code into user input fields.

How it works in DVWA:

- DVWA has a vulnerable input form (e.g., user login or search).
- Attacker enters specially crafted input like ' OR '1'='1 instead of normal data.
- The application executes the SQL query without proper input validation or sanitization.
- The attacker can bypass authentication, retrieve, modify, or delete database data.


Vulnerability: SQL Injection
CrackStation - Online Pas
Altoro Mutual
GitHub - backdoorhub/sh
127.0.0.1/DVWA/vulnerabilities/sqli/?id=1'+UNION+SELECT+user+%2C+password+FROM+users%23&Submit=Submit#
110%
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Open HTTP Redirect
Cryptography
API
DVWA Security
Vulnerability: SQL Injection
User ID: sword FROM users# Submit
ID: 1' UNION SELECT user , password FROM users#
First name: admin
Surname: admin
ID: 1' UNION SELECT user , password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
ID: 1' UNION SELECT user , password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03
ID: 1' UNION SELECT user , password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
ID: 1' UNION SELECT user , password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
ID: 1' UNION SELECT user , password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
More Information
https://en.wikipedia.org/wiki/SQL_injection
https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
https://owasp.org/www-community/attacks/SQL_Injection
https://bobby-tables.com/


Vulnerability: SQL Injection
CrackStation - Online Pas
Altoro Mutual
GitHub - backdoorhub/sh
https://crackstation.net
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
CrackStation
Defuse.ca · Twitter
CrackStation
Password Hashing Security
Defuse Security
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
5f4dcc3b5aa765d61d8327deb882cf99
I'm not a robot
reCAPTCHA
Privacy - Terms
Crack Hashes
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults
Hash
Type
Result
5f4dcc3b5aa765d61d8327deb882cf99
md5
password
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.
Download CrackStation's Wordlist
How CrackStation Works
CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our hashing security page.
Crackstation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list we could find. We also applied intelligent word mangling (brute force hybrid) to our wordlists to make them much more effective. For MD5 and SHA1 hashes, we have a 190GB, 15-billion-entry lookup table, and for other hashes, we have a 19GB 1.5-billion-entry lookup table.
You can download CrackStation's dictionaries here, and the lookup table implementation (PHP and C) is available here.

THANK YOU!!