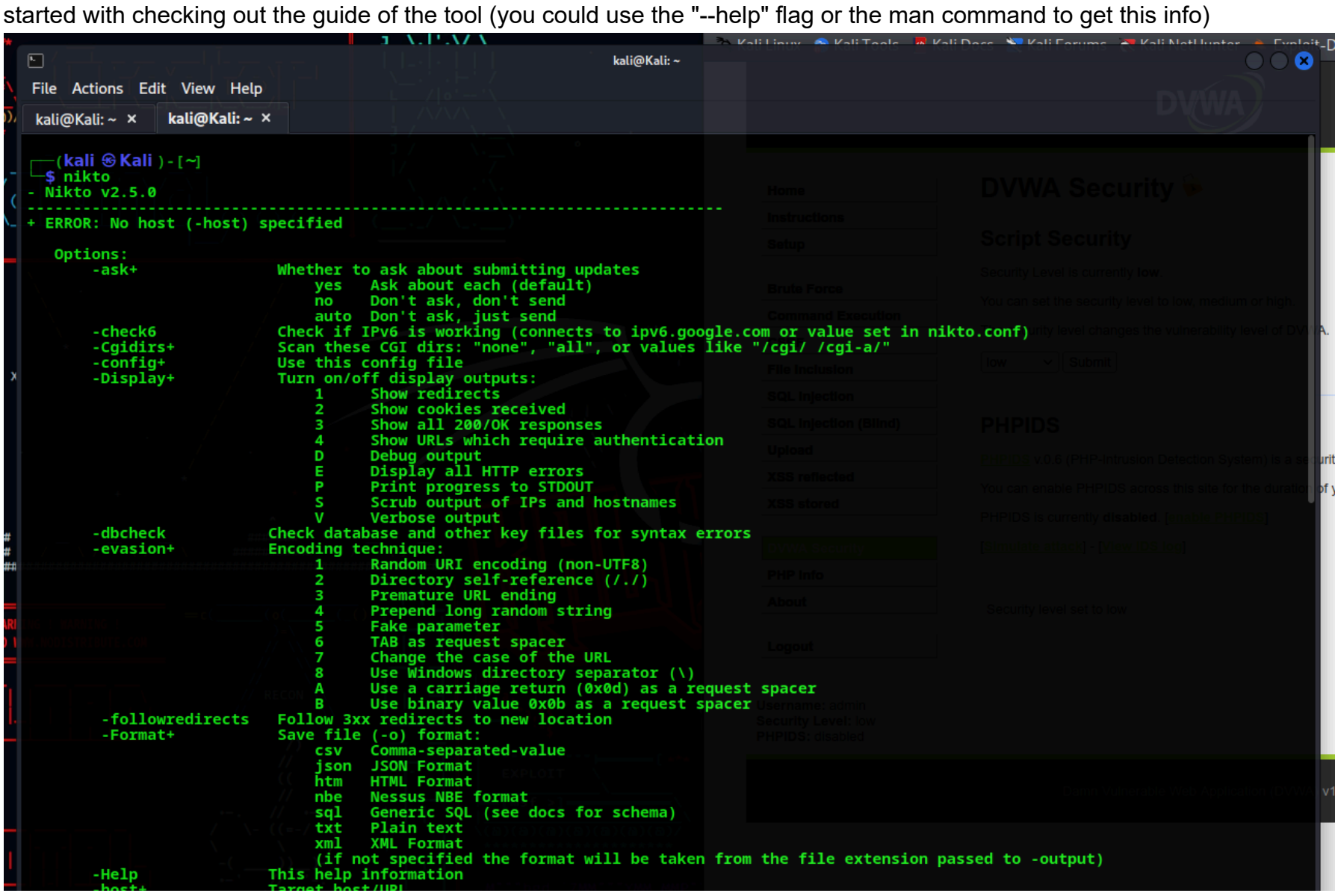
Nikto and SQL injection lab -Maxatefoe

Nikto is an open-source command-line web server scanner that checks for security vulnerabilities and other misconfigurations in a webserver.

This lab was done to display the usage of this tool
Let's get into it.....

- started with checking out the guide of the tool (you could use the "--help" flag or the man command to get this info)



- Created an ip list(using nano) of the addresses of the webserver's we'll be testing out

- used nikto against the targets in the list

```
┌──(kali㉿Kali)-[~]
└─$ nikto -h iplist.txt
- Nikto v2.5.0
---------------------------------------------------------------------------
---------------------------------------------------------------------------
+ Target IP:          10.6.6.23
+ Target Hostname:    10.6.6.23
+ Target Port:        80
+ Start Time:         2026-01-18 12:02:36 (GMT0)
---------------------------------------------------------------------------
+ Server: nginx/1.14.2
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-
Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashi
on to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B^[[B+ /admin/: This might be interesting.
^[[A^[[A^[[A^[[A^[[A^[[A+ /admin/index.html: Admin login page/section found.
+ /wp-admin/: Admin login page/section found.
+ /wp-login/: Admin login page/section found.
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8082 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2026-01-18 12:02:58 (GMT0) (22 seconds)
---------------------------------------------------------------------------
+ Target IP:          10.6.6.13
+ Target Hostname:    10.6.6.13
+ Target Port:        80
+ Start Time:         2026-01-18 12:02:58 (GMT0)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ /: Cookie PHPSESSID created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-
Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashi
on to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: login.php
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /config/: Directory indexing found.
+ /config/: Configuration information may be available remotely.
+ /docs/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /login.php: Admin login page/section found.
+ 16132 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2026-01-18 12:03:21 (GMT0) (23 seconds)
---------------------------------------------------------------------------
+ Target IP:          10.6.6.14
+ Target Hostname:    10.6.6.14
+ Target Port:        80
+ Start Time:         2026-01-18 12:03:21 (GMT0)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ /: Cookie PHPSESSID created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Cookie showhints created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.25.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-
Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashi
```

- as seen it found a lot of vulns like files with login creds and misconfigurations like missing http only flags
- since the results are not easily readable , we used nikto to create and html report to make it more legible (GUI always makes everything visually pleasing lol)
- we parsed this using the "-o" flag to the report name {scanresults.html}

```
+ 4 host(s) tested

┌──(kali㉿Kali)-[~]
└─$ nikto -h 172.17.0.2 -o scanresults.html
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          172.17.0.2
+ Target Hostname:    172.17.0.2
+ Target Port:        80
+ Start Time:         2026-01-18 12:06:20 (GMT0)
---------------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /doc/: Directory indexing found.
+ /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0678
+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/ChangeLog: Server may leak inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 1115138, size: 40540, mtime: Tue Dec  9 17:24:00 2008. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /test/: Directory indexing found.
+ /test/: This might be interesting.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /phpMyAdmin/: phpMyAdmin directory found.
+ /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://typo3.org/
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8910 requests: 0 error(s) and 27 item(s) reported on remote host
+ End Time:           2026-01-18 12:06:56 (GMT0) (36 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested

┌──(kali㉿Kali)-[~]
```

172.17.0.2 / 172.17.0.2 port 80

Target IP	172.17.0.2
Target hostname	172.17.0.2
Target Port	80
HTTP Server	Apache/2.2.8 (Ubuntu) DAV/2
Site Link (Name)	http://172.17.0.2:80/
Site Link (IP)	http://172.17.0.2:80/

URI	/
HTTP Method	GET
Description	/: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
Test Links	http://172.17.0.2:80/ http://172.17.0.2:80/
References	

URI	/
HTTP Method	GET
Description	/: The anti-clickjacking X-Frame-Options header is not present.
Test Links	http://172.17.0.2:80/ http://172.17.0.2:80/
References	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

URI	/
HTTP Method	GET
Description	/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
Test Links	http://172.17.0.2:80/ http://172.17.0.2:80/
References	https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/

URI	/index
HTTP Method	GET
Description	/index: Uncommon header 'tcn' found, with contents: list.
Test Links	http://172.17.0.2:80/index http://172.17.0.2:80/index
References	

URI	/index
HTTP Method	GET
Description	/index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php.
Test Links	http://172.17.0.2:80/index http://172.17.0.2:80/index

a very useful tool, don't you think??

Unto the next one!!!.....

SQL Injection

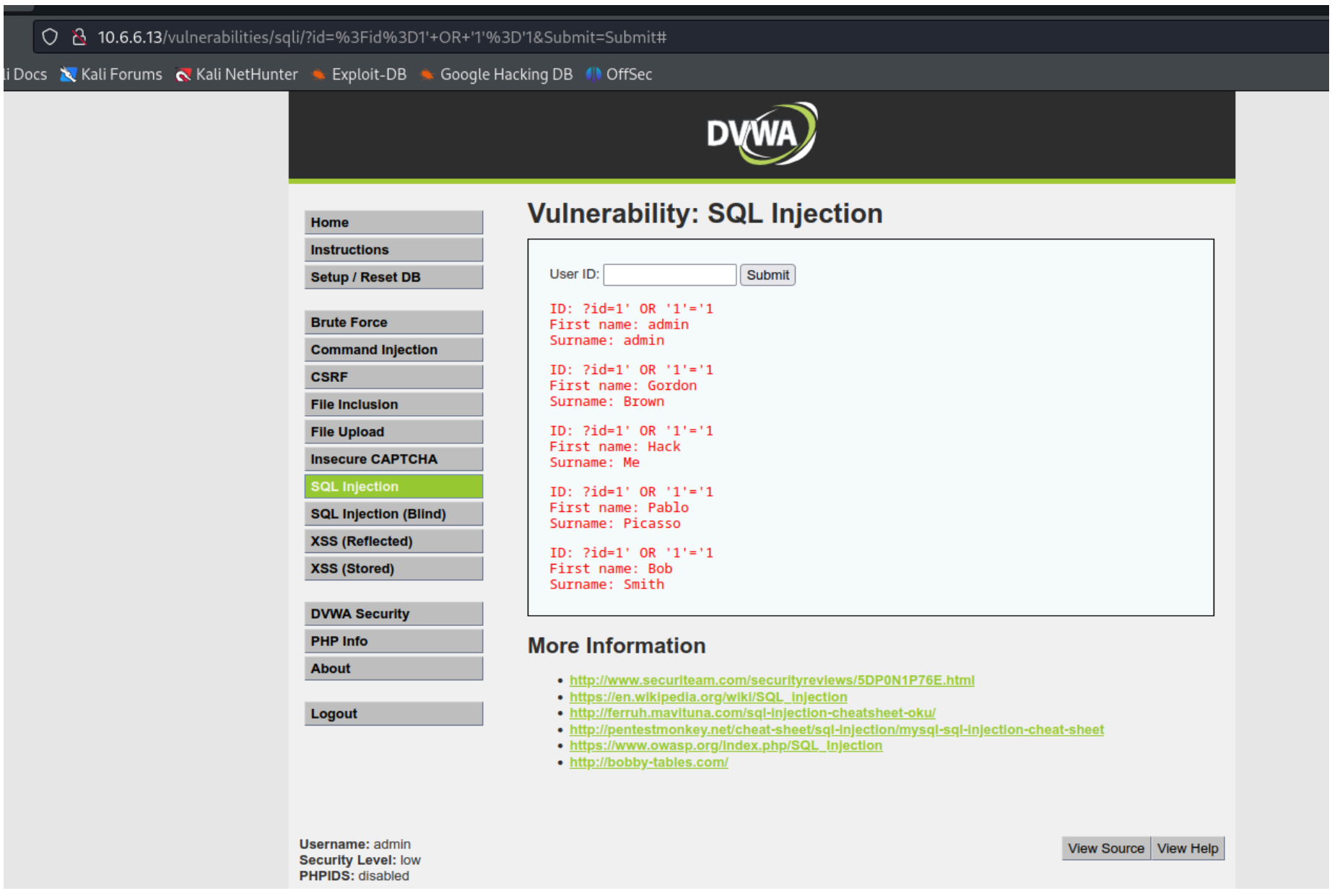
SQL Injection (SQLi) is a cyberattack where malicious SQL code is inserted into data-driven applications via input fields (like login forms, search bars) to trick the database into executing unintended commands, allowing attackers to steal, modify, or delete data, bypass authentication, and even take over the server by exploiting flaws where user input isn't properly validated or sanitized

This lab is done to simulate how SQLi attacks are done

- our target here is 10.6.6.13 a Damn vulnerable web app

Let's begin....

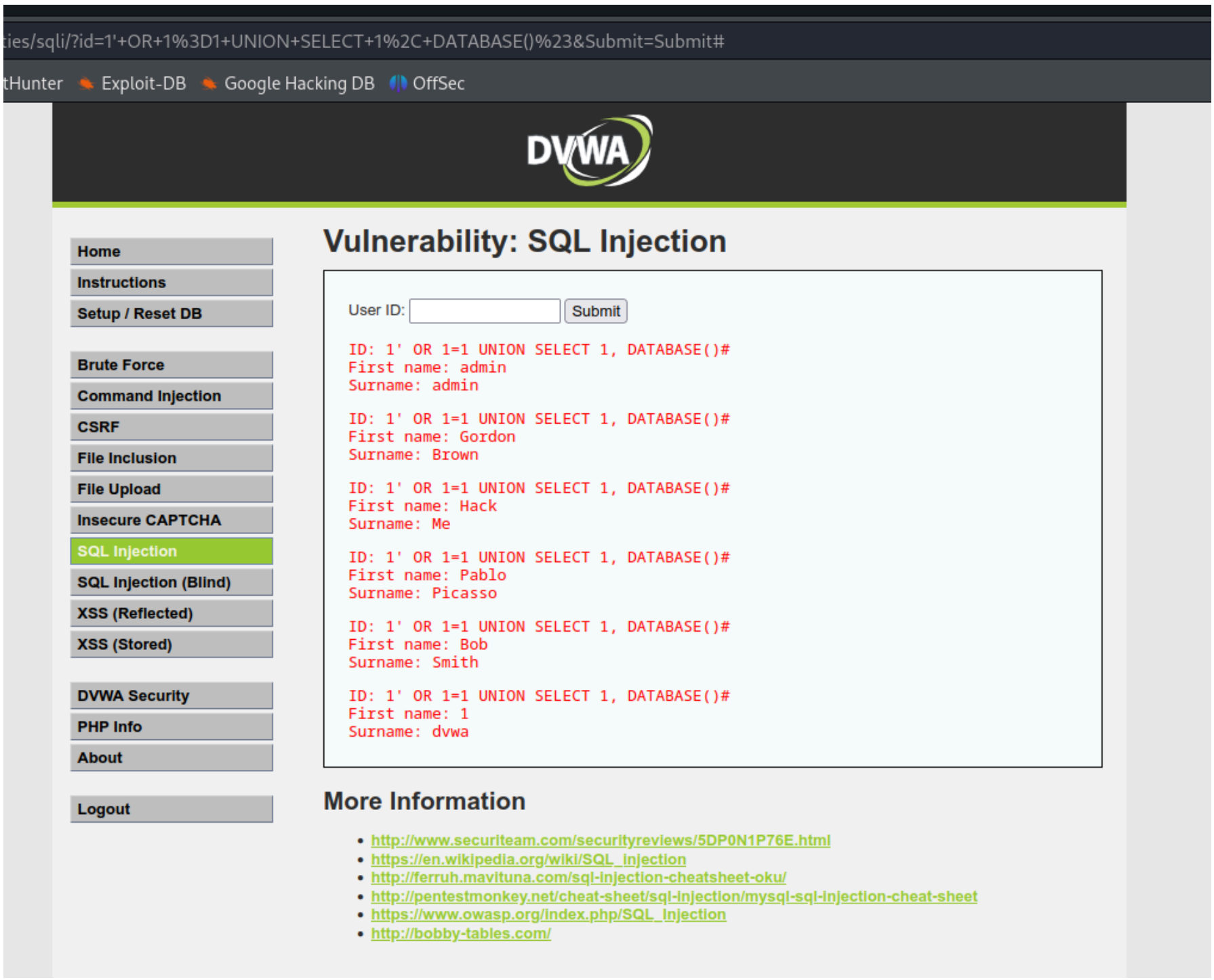
- Input: ?id=1' OR '1'='1 (this is mostly used to test if that input section is vulnerable to SQLi)
- The resulting query becomes: SELECT first_name, last_name FROM users WHERE user_id = '1' OR '1'='1';
- This always evaluates to TRUE, returning all user data.



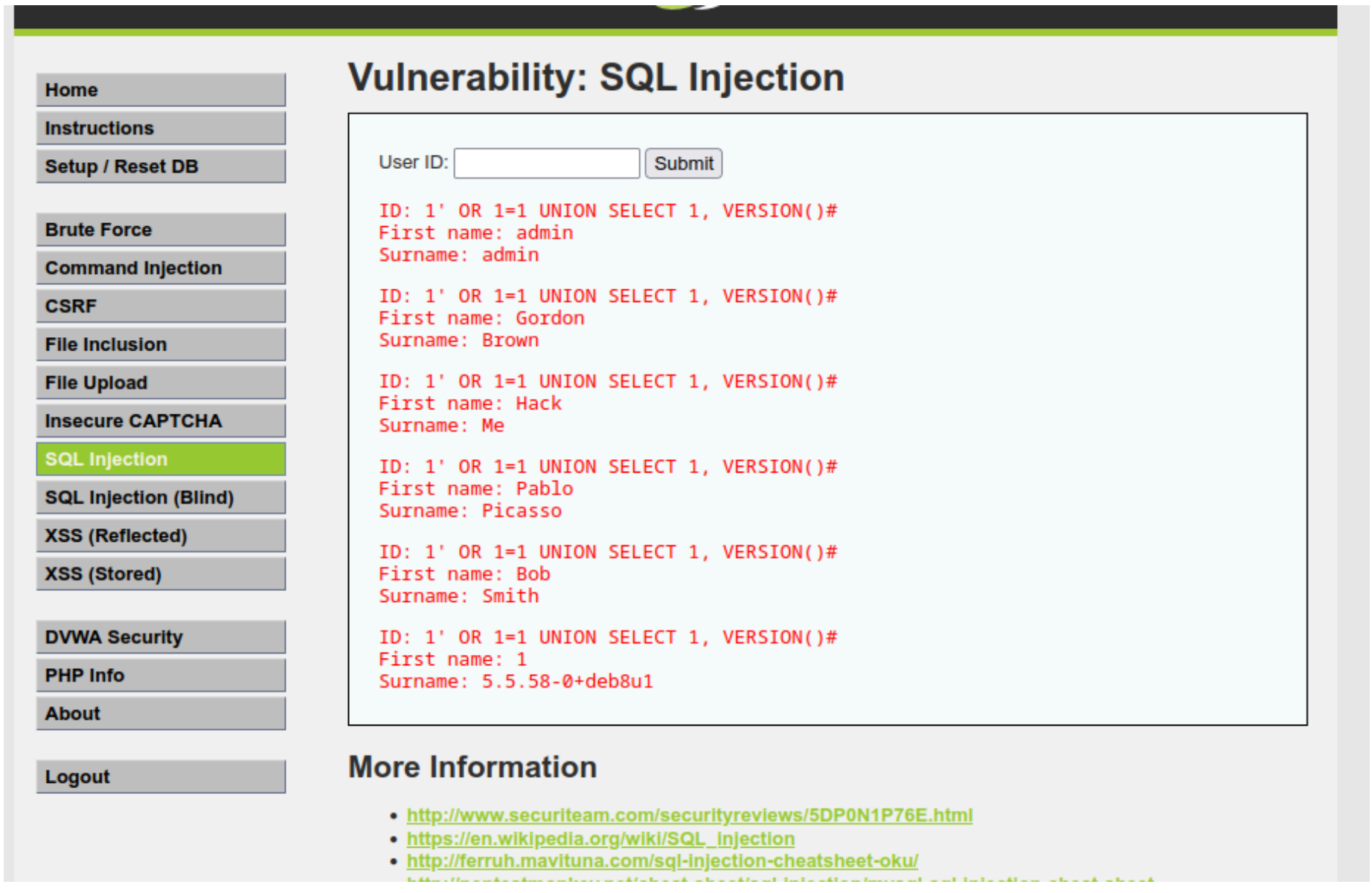
Now that we know its vulnerable we can pull other data using different queries on the database

imma just list some of the data you can get

- the database name using this: 1' OR 1=1 UNION SELECT 1, DATABASE()#

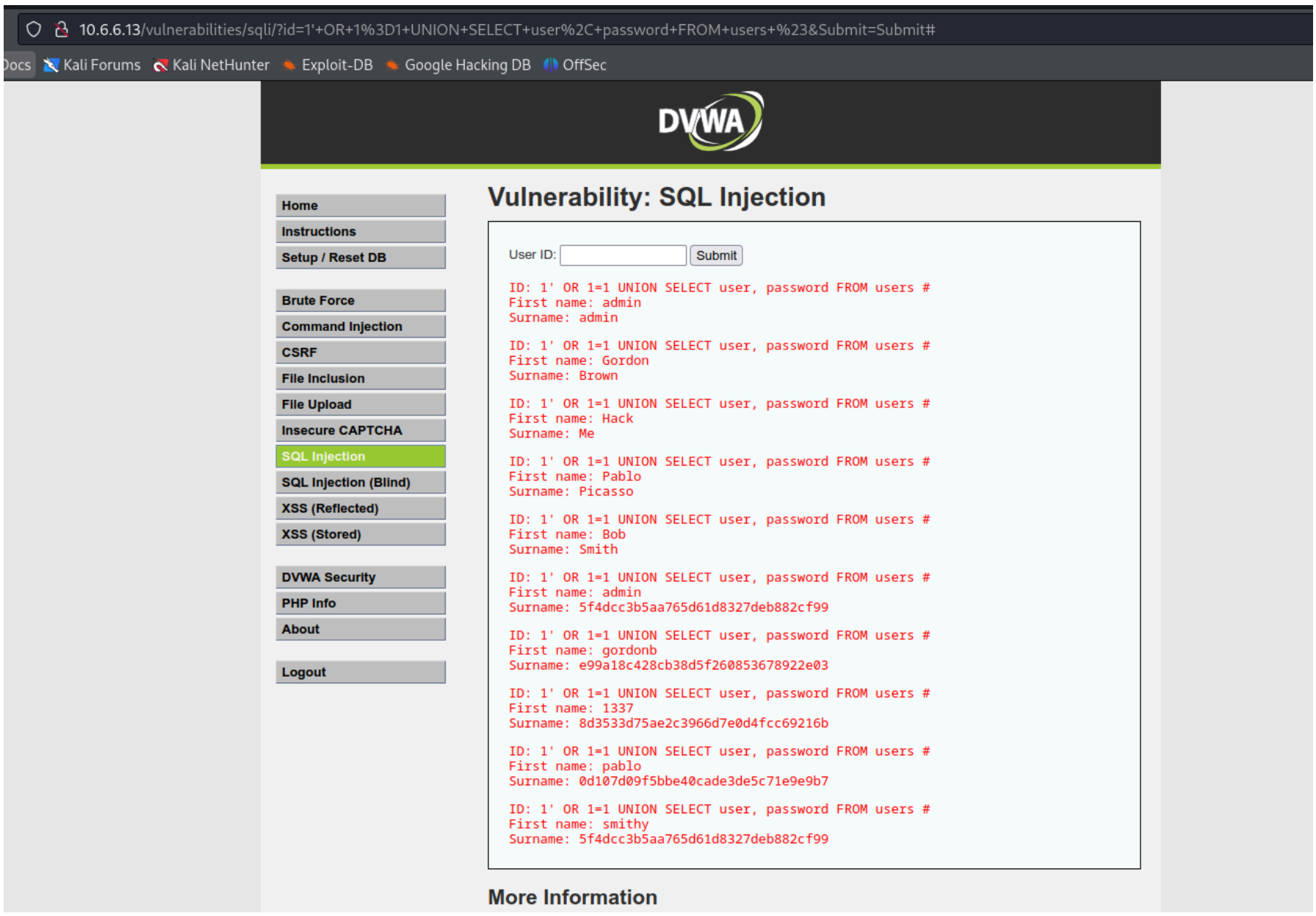


- database name found: dvwa
- The database version using this: 1' OR 1=1 UNION SELECT 1, VERSION()#

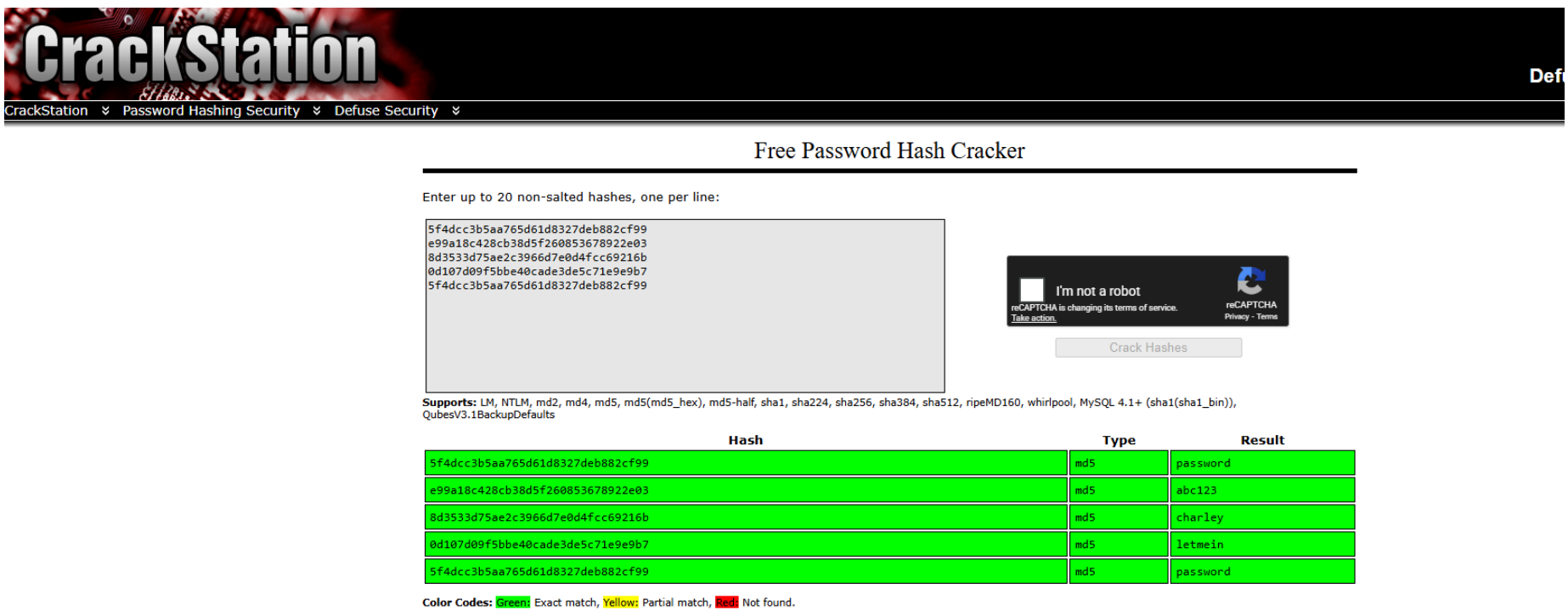


- version found: 5.5.58-0+deb8u1
- we could also get some user names and password from different table

- for example we used: 1' OR 1=1 UNION SELECT user, password FROM users #



- This returns **usernames and passwords** from the database.
- we could crack those password hashes using any cracking tool here we used crackstation



this shows how dangerous of an attack SQLi is