

WEB APPLICATION SECURITY TESTING

Prepared by Mir Sahil Samiruddin
Prepared for TASK 01 Internship at Future Interns
V1.0 May | 25 | 2025

Table Of Contents

Executive Summary	1
Test Scope	1
Results	1
Recommendations	2
Testing Approach	3
Overview	3
Discovery & Reconnaissance	4
Validation & Exploitation	4
Web Application and Internal Network Findings	5
Web Applications Results	6
Web Applications Detailed Findings	7
Details	9

Document Control

Issue Control			
Document Refrence	n/a	Project Number	n/a
Issue	1	Date	25 May 2025
Classification	Open	Author	Mir Sahil
Document Title	Web Application Secuity Report		
Released By	Mir Sahil Samiruddin		

Owner Details	
Name	Mir Sahil Samiruddin
Office/Region	
Contact Number	9152433262
Email Address	Mirsahil050@gmail.com

Revision History			
Issue	Date	Auhor	Comments
1.0	25 May 2025	Mir Sahil	

EXECUTIVE SUMMARY

I conducted a comprehensive security assessment of Sample Vulnerable Web Application named DVWA (Damn Vulnerable Web Application) , in order to determine existing vulnerabilities and establish the current level of security risk associated with the environment and the technologies in use. This Assessment harnessed penetrations testing and social engineering techniques to provide management with an understanding of the risks and security posture of their corporate environment

Test Scope

The Test Scope for this engagement included hosts one of my own vulnerable test lab in my internal network , a web application as well as internally-developed Test Report. Testing was performed May 25-May 2025. Additional days were utilized to produce the report Testing was performed using industry-standard penetration testing tools and frameworks, including OWASP ZAP, Wireshark, Burp Suite,SQLMAP.

RESULTS

The table below includes the scope of the tests performed, as well as the overall results of penetration. testing these environments.

Environment tested	Testing Results
Internal Network	Critical
Web Application	High

To test the security posture of the internal network, we began with a reconnaissance and host discovery phase during which we used port scans, ARP scans, and OSINT tools to fingerprint the operating systems, software, and services running on each target host. After fingerprinting the various targets and determining open ports and services enabled on each host, we executed a vulnerability enumeration phase, in which we listed all potential vulnerabilities affecting each host and developed a list of viable attack vectors. Finally, in order to weed out false positives and validate any remaining vulnerabilities. we attempted to exploit all vulnerabilities affecting the target hosts. After comprehensive testing, only a few vulnerabilities were discovered to be present in the target hosts, and we were ultimately unable to exploit these issues to compromise the confidentiality, integrity, or availability of any of the external hosts in scope.

Multiple Critical-and High-and Medium-severity issues were found affecting hosts on internal network, which require immediate remediation efforts in order to secure the environment against malicious attackers.

RECOMMENDATIONS

The following recommendations provide direction on improving the overall security posture of SampleCorp's networks and business-critical applications:

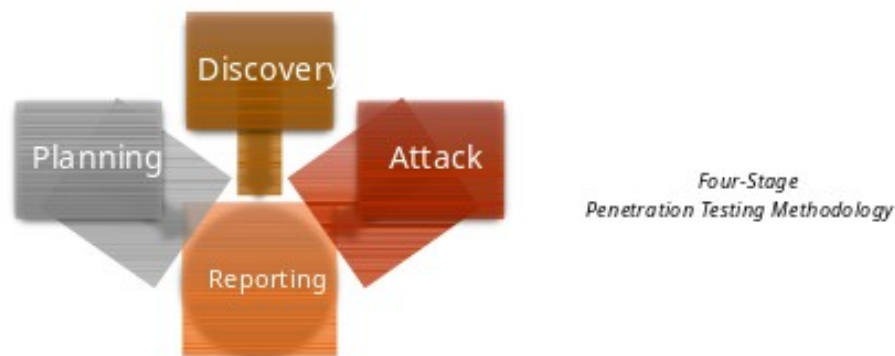
1. Ensure that the credentials protecting the Login instance on host 192.168.56.101 are of suitable complexity to prevent brute force attacks, or disable Secure Admin on the instance to prevent remote access from the Ping .
2. Sanitizing User Input if necessary and Avoid revealing SQL errors in the responses .
3. Enforces Strong password policy and add CAPTCHA to login.
4. Restrict access to the distcd service on host 192.168.56.101 (UDP port 3632).
5. Never concatenate shell commands with user input and touch the host shell 192.168.56.101.
6. Use HTTPS with a valid SSL certificate and every server up to date.
7. Only allow specific MIME types and verify the file extension and file conten on host 192.168.56.101.
8. Update the Ninja Forms plugin to version 2.9.43 or higher on the web app located at <http://192.168.56.101/dvwa/>
9. Increase the strength of the password for the 'password" administrator account on the web app located at <http://192.168.56.101/dvwa/>

TESTING APPROACH

OVERVIEW

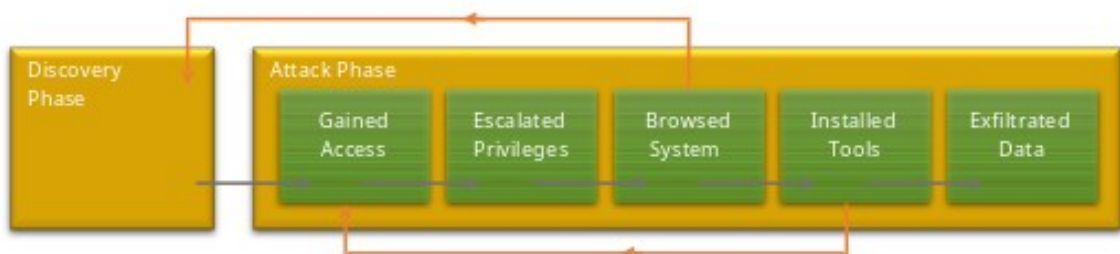
All testing was executed in several related phases.

1. In the planning phase, the rules of engagement were identified, scope of testing and test windows were agreed upon, and testing goals were set.
2. The discovery phase included automated vulnerability scanning along with manual testing to explore and understand the testing target and any vulnerabilities that could be detected by automated tools.
3. The attack phase comprised efforts to exploit any vulnerabilities detected, and to synthesize knowledge gained about the environment, its technology, its users and its function into an escalation of privilege beyond that intended by the customer.
4. The final phase recorded all findings in a manner that supports risk assessment and remediation by the customer. This included the writing of this report.



Additionally, the attack phase comprised several distinct steps, executed iteratively as information was discovered.

1. Gained access to the system or environment in a way that was not intended.
2. Escalated privileges to move from regular or anonymous user to a more privileged position.
3. Browsed to explore the newly accessed environment and identify useful assets and data.
4. Deployed tools to attack further from the newly gained vantage point.
5. Exfiltrated data.



INTERNAL NETWORK FINDINGS

SCOPE

The following externally accessible IP addresses were within the scope of this engagement:

Target IP Address
192.168.56.101

Testing was performed using industry-standard penetration testing tools and frameworks, including NMAP, OWASP ZAP, SQLMAP, Wireshark, and Burp Suite.

NETWORK PENETRATION TESTING RESULTS

Result Classification

Vulnerabilities Found	Yes
Exploited - Elevation of Privileges (EoP)	Yes
Exploited – Remote Code Execution (RCE)	Yes
Sensitive Data Exfiltrated	Yes
Overall Risk	High

There were a significant number of exploited vulnerabilities present on the external network target, including a vulnerability in the Oracle Glassfish server, a vulnerability in the Apache Struts REST Plugin, an unrestricted WebDAV upload vulnerability, misconfigured services, a vulnerability in the DistCC daemon, a Samba RCE vulnerability, and a buffer overflow vulnerability in the SLMail application, all of which led to system compromise of the affected hosts.

Services by Host and by Port

As the first step in the Discovery phase, I conducted network reconnaissance on the provided IP addresses to determine open ports. Each IP address was tested for all TCP and UDP ports by using standard scanning tools like Nmap. The following ports were identified, and ports with exploitable vulnerabilities are highlighted.

IP address	TCP/UDP	Port	Service	Version
192.168.56.101	TCP	21	FTP	Vsftpd 2.3.4
	TCP	22	SSH	OpenSSH 4.71p1 Debian 8ubuntu1 (protocol 2.0)
	TCP	23	Telnet	Linux telnetd
	TCP	25	SMTP	Postfix smtpd
	TCP	53	Domain	ISC BIND 9.4.2
	TCP	80	HTTP	Apache httpd 2.2.8 ((Ubuntu) DAV/2)
	TCP	111	rpcbind	2 (RPC#100000)
	TCP	139	Netbios-ssn	Samba smbd 3.X- 4.X workgroup
	TCP	445	Netbios-ssn	Samba smbd 3.0.20- Debian workgroup
	TCP	512	Exec	Netkit-rsh rexecd
	TCP	513	login	
	TCP	514	shell	Netkit rshd
	TCP	1099	java-rmi	GNU Classpath grmiregistry
	TCP	8180	http	Apache Tomcat/Coyote JSP Engine 1.1
	TCP	2049	nfs	2-3 (RPC#100003)
	TCP	2121	ftp	ProFTPD 1.3.1
	TCP	3306	mysql	MySQL 5.0.51a- 3ubuntu5
	TCP	5432	postgresql	PostgreSQL DB 8.3.0-8.3.7
	TCP	5900	vnc	VNC (protocol 3.3)
	TCP	6000	X11	
	TCP	6667	irc	UnrealIRCd
	TCP	8009	ajp13	Apache Jserv (Protocol v1.3

Web App Sec Report

```
sebastian@Sebastian: ~  
80/tcp open  http      Apache httpd 2.2.8 ((Ubuntu) DAV/2)  
|_http-title: Metasploitable2 - Linux  
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2  
111/tcp open  rpcbind    2 (RPC #100000)  
|_rpcinfo:  
|_  program version  port/proto  service  
|_  100000  2             111/tcp    rpcbind  
|_  100000  2             111/udp    rpcbind  
|_  100003  2.3.4         2049/tcp   nfs  
|_  100003  2.3.4         2049/udp   nfs  
|_  100005  1.2.3         41246/tcp  mountd  
|_  100005  1.2.3         41314/udp  mountd  
|_  100021  1.3.4         33874/tcp  nlockmgr  
|_  100021  1.3.4         53731/udp  nlockmgr  
|_  100024  1             59211/tcp  status  
|_  100024  1             59443/udp  status  
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)  
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)  
512/tcp open  exec       netkit-rsh rxeecd  
513/tcp open  login  
514/tcp open  shell      Netkit rshd  
1099/tcp open  java-rmi   GNU Classpath girmiregistry  
1524/tcp open  bindshell  Metasploitable root shell  
2049/tcp open  nfs        2-4 (RPC #100003)  
2121/tcp open  ftp        ProFTPD 1.3.1  
3306/tcp open  mysql      MySQL 5.0.51a-3ubuntu5  
|_mysql-info:  
|_  Protocol: 10  
|_  Version: 5.0.51a-3ubuntu5  
|_  Thread ID: 64  
|_  Capabilities flags: 43564  
|_  Some Capabilities: ConnectWithDatabase, LongColumnFlag, Support4Auth, SupportsTransactions, SwitchToSlaAfterHandshake, Speaks41ProtocolNew, SupportsCompression  
|_  Status: Autocommit  
|_  Salt: x[=^AhIVS-Cn$BEOIRJZ  
5432/tcp open  postgresql PostgreSQL DB 8.3.0 - 8.3.7  
|_ssl-date: 2025-05-31T13:30:22+00:00; -1h54m28s from scanner time.  
5900/tcp open  vnc        VNC (protocol 3.3)  
|_vnc-info:  
|_  Protocol version: 3.3  
|_  Security types:  
|_  VNC Authentication (2)  
6000/tcp open  X11        (access denied)  
6067/tcp open  irc        UnrealIRCd  
8009/tcp open  ajp13      Apache Jserv (Protocol v1.3)  
|_ajp-methods: Failed to get a valid response for the OPTION request  
8180/tcp open  http       Apache Tomcat/Coyote JSP engine 1.1  
|_http-server-header: Apache-Coyote/1.1  
|_http-favicon: Apache Tomcat  
|_http-title: Apache Tomcat/5.5  
MAC Address: 08:00:27:E3:FF:69 (Oracle VirtualBox virtual NIC)  
Device type: general purpose  
Running: Linux 2.6.X  
OS CPE: cpe:/o:linux:linux_kernel:2.6  
OS details: Linux 2.6.9 - 2.6.33  
Network Distance: 1 hop  
Service Info: Hosts: metasploitable.localdomain, irc.metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
  
Host script results:  
|_ smb-os-discovery:  
|_  OS: Unix (Samba 3.0.20-Debian)  
|_  Computer name: metasploitable  
|_  NetBIOS computer name:  
|_  Domain name: localdomain  
|_  FQDN: metasploitable.localdomain  
|_  System time: 2025-05-31T09:30:06-04:00  
|_  clock-skew: mean: -54m28s, deviation: 1h59m59s, median: -1h54m28s  
|_  smb-security-mode:  
|_  account_used: <blank>  
|_  authentication_level: user  
|_  challenge_response: supported  
|_  message_signing: disabled (dangerous, but default)  
|_  nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: 000000000000 (Xerox)  
|_  smb2-time: Protocol negotiation failed (SMB2)  
  
TRACEROUTE  
HOP RTT ADDRESS  
1 0.54 ms 192.168.56.101  
  
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 28.53 seconds  
sebastian@Sebastian: $  
|_ 100000 2             111/tcp    rpcbind
```

```
sebastian@Sebastian: ~  
5432/tcp open  postgresql PostgreSQL DB 8.3.0 - 8.3.7  
|_ssl-date: 2025-05-31T13:30:22+00:00; -1h54m28s from scanner time.  
5900/tcp open  vnc        VNC (protocol 3.3)  
|_vnc-info:  
|_  Protocol version: 3.3  
|_  Security types:  
|_  VNC Authentication (2)  
6000/tcp open  X11        (access denied)  
6067/tcp open  irc        UnrealIRCd  
8009/tcp open  ajp13      Apache Jserv (Protocol v1.3)  
|_ajp-methods: Failed to get a valid response for the OPTION request  
8180/tcp open  http       Apache Tomcat/Coyote JSP engine 1.1  
|_http-server-header: Apache-Coyote/1.1  
|_http-favicon: Apache Tomcat  
|_http-title: Apache Tomcat/5.5  
MAC Address: 08:00:27:E3:FF:69 (Oracle VirtualBox virtual NIC)  
Device type: general purpose  
Running: Linux 2.6.X  
OS CPE: cpe:/o:linux:linux_kernel:2.6  
OS details: Linux 2.6.9 - 2.6.33  
Network Distance: 1 hop  
Service Info: Hosts: metasploitable.localdomain, irc.metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
  
Host script results:  
|_ smb-os-discovery:  
|_  OS: Unix (Samba 3.0.20-Debian)  
|_  Computer name: metasploitable  
|_  NetBIOS computer name:  
|_  Domain name: localdomain  
|_  FQDN: metasploitable.localdomain  
|_  System time: 2025-05-31T09:30:06-04:00  
|_  clock-skew: mean: -54m28s, deviation: 1h59m59s, median: -1h54m28s  
|_  smb-security-mode:  
|_  account_used: <blank>  
|_  authentication_level: user  
|_  challenge_response: supported  
|_  message_signing: disabled (dangerous, but default)  
|_  nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: 000000000000 (Xerox)  
|_  smb2-time: Protocol negotiation failed (SMB2)  
  
TRACEROUTE  
HOP RTT ADDRESS  
1 0.54 ms 192.168.56.101  
  
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 28.53 seconds  
sebastian@Sebastian: $  
|_ 100000 2             111/tcp    rpcbind
```

Mir Sahil Samiruddin

Email: mirsahil050@gmail.com LinkedIn: www.linkedin.com/in/mir-sahil-606342254

Vulnerability Summary Table

I strongly recommends that the following vulnerabilities be remediated, whether exploited or not, as they represent unnecessary risk.

#	Vulnerability Summary	Risk Level	Recommendation
1	SQL Injection	High	Use prepared statements (PDO/MySQLi), input validation
2	Blind SQL Injection	High	Same as above; parameterized queries
3	Command Injection	High	Use internal PHP functions or sanitize input + escapeshellarg()
4	Reflected XSS	Medium	Use htmlspecialchars () or a templating engine
5	Stored XSS	High	Sanitize input AND output , encode HTML
6	Cross-Site Request Forgery (CSRF)	High	Implement CSRF tokens , set SameSite and HttpOnly cookies
7	Insecure File Upload	High	Validate MIME types, rename files , store outside web root
8	Weak Authentication	High	Enforce strong creds, implement rate-limiting or CAPTCHA
9	Session Management Issues	Medium	Use session_regenerate_id(), set HttpOnly, Secure, SameSite
10	Security Misconfigurations	Medium	Disable verbose errors in production (display_errors = off)
11	JavaScript DOM-Based XSS	Medium	Use encodeURIComponent () or strict JS input handling

Details

1. SQL Injection

RISK : HIGH

Location: 192.168.56.101:8090

Description:

SQL injection is a type of cyber attack where an attacker inserts malicious SQL code into a query through user input fields, allowing them to manipulate or access sensitive data in a database. This vulnerability often arises from improper coding practices in web applications that do not adequately validate user inputs.

Observation:

Vulnerability: SQL Injection

User ID: Submit

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: admin
Surname: admin

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: Gordon
Surname: Brown

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: Hack
Surname: Me

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: Pablo
Surname: Picasso

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: Bob
Surname: Smith

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: admin:5f4dcc3b5aa765d61d8327deb082cf99
Surname: 5

Vulnerability: SQL Injection (Blind)

User ID: Submit

Impact

CVSS Score 9.8

Confidentiality: High

Integrity: High

Availability: High

Attack Vector: Network

No Auth Needed

Impact: DB takeover

Recommendations:

- Use **prepared statements** or ORM
- Sanitize user inputs

Mir Sahil Samiruddin

Email: mirsahil050@gmail.com LinkedIn: www.linkedin.com/in/mir-sahil-606342254

- Disable detailed SQL errors

References:

https://owasp.org/www-community/attacks/SQL_Injection

<https://cwe.mitre.org/data/definitions/89.html>

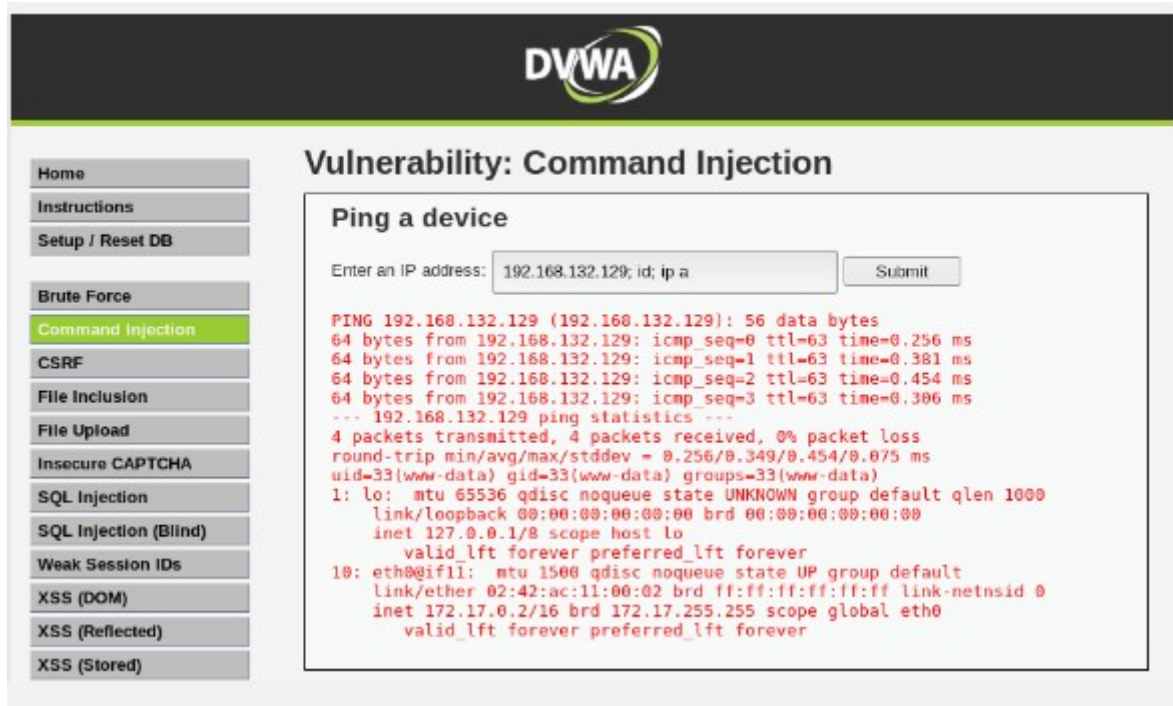
2. Command Injection

RISK : HIGH

Location: 192.168.56.101:8090

Description: Command injection is a type of security vulnerability that allows an attacker to execute arbitrary commands on a host operating system through a vulnerable application. This occurs when an application improperly handles user input, allowing attackers to manipulate commands that the application sends to the system shell.

Observation:



Impact

CVSS Score 10.0

Confidentiality: High

Integrity: High

Availability: High

Attack Vector: Web server

No Auth Needed

Impact: Full System command execution

Recommendations:

- Avoid shell usage
- Use `escapeshellarg()` / `escapeshellcmd()`
- Run app under least-privileged user

References:

https://owasp.org/www-community/attacks/Command_Injection

<https://cwe.mitre.org/data/definitions/78.html>

3. Reflected XSS

RISK : MEDIUM

Location: 192.168.56.101:8090

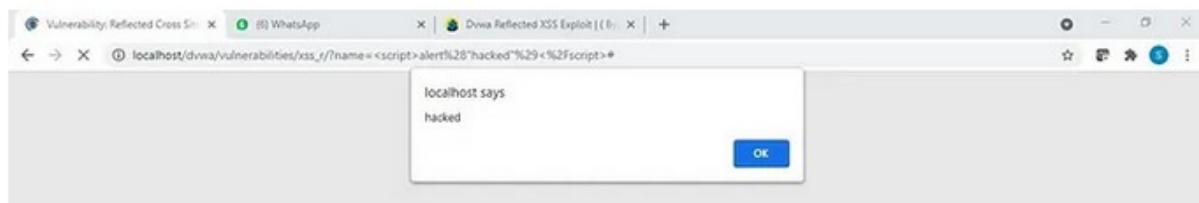
Description:

Reflected XSS (Cross-Site Scripting) is a type of security vulnerability where an attacker injects malicious scripts into a web application, and these scripts are immediately reflected back to the user without proper validation or encoding. This can occur through user input, such as URL parameters, and can lead to unauthorized actions or data theft when the victim interacts with the compromised link.

Observation:



Inject the payload `<script>alert("hacked")</script>`



Impact

CVSS Score: 6.1

Confidentiality Impact: Moderate (Steal cookies, phishing)

Integrity Impact: Low

Availability Impact: None

Access Complexity: Low

Authentication: Not required

Vulnerability Type(s): Cross-Site Scripting (Reflected)

Recommendations:

Mir Sahil Samiruddin

Email: mirsahil050@gmail.com LinkedIn: www.linkedin.com/in/mir-sahil-606342254

- Use `htmlspecialchars()`
- Apply Content Security Policy (CSP)
- Validate/encode user input before output

References:

<https://owasp.org/www-community/attacks/xss/>

<https://cwe.mitre.org/data/definitions/79.html>

4. CSRF

RISK : MEDIUM

Location: 192.168.56.101:8090

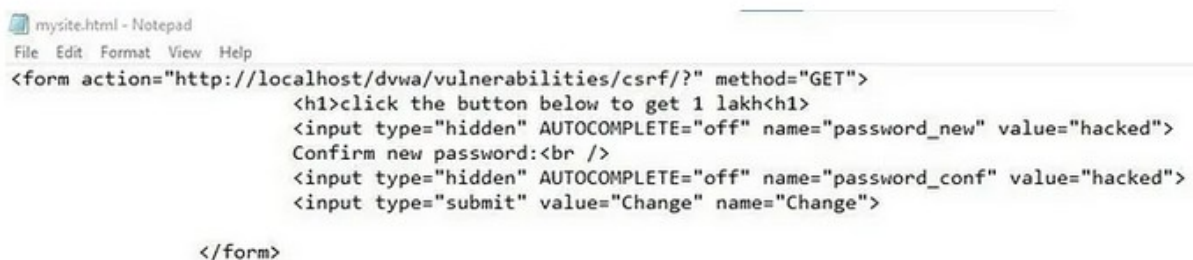
Description:

Cross-site request forgery (CSRF) is a type of attack that tricks a user's web browser into executing unwanted actions on a trusted website where the user is authenticated. This can lead to unauthorized actions like changing account settings or making transactions without the user's knowledge.

Observation:

```
<form action="#" method="GET">
  New password:<br />
  <input type="password" AUTOCOMPLETE="off" name="password_new"><br />
  Confirm new password:<br />
  <input type="password" AUTOCOMPLETE="off" name="password_conf"><br />
  <br />
  <input type="submit" value="Change" name="Change">
</form>
```

3. Change the source code in notepad and save as mysite.html



```
mysite.html - Notepad
File Edit Format View Help
<form action="http://localhost/dvwa/vulnerabilities/csrf/?" method="GET">
  <h1>click the button below to get 1 lakh</h1>
  <input type="hidden" AUTOCOMPLETE="off" name="password_new" value="hacked">
  Confirm new password:<br />
  <input type="hidden" AUTOCOMPLETE="off" name="password_conf" value="hacked">
  <input type="submit" value="Change" name="Change">
</form>
```

Impact

CVSS Score: 7.5

Confidentiality Impact: Medium

Integrity Impact: High (Modify user actions silently)

Availability Impact: Low

Access Complexity: Low

Authentication: Not required (if user is logged in)

Vulnerability Type(s): Cross-Site Request Forgery

Recommendations:

- Add CSRF tokens to all sensitive forms
- Set SameSite cookie attribute

Mir Sahil Samiruddin

Email: mirsahil050@gmail.com LinkedIn: www.linkedin.com/in/mir-sahil-606342254

- Use HttpOnly + Secure flags

References:

<https://owasp.org/www-community/attacks/csrf>

<https://cwe.mitre.org/data/definitions/352.html>

5.XSS (Stored)

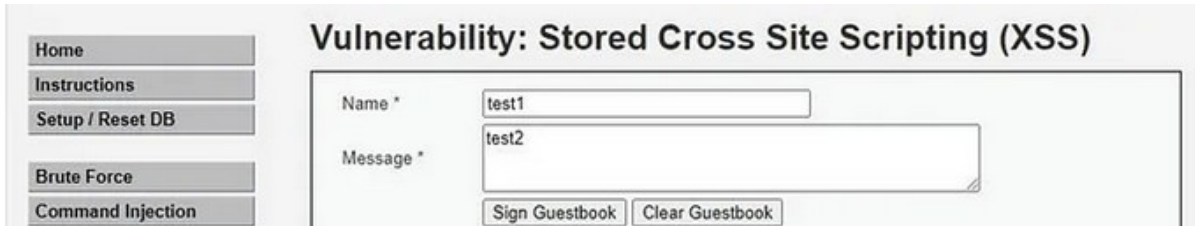
RISK:HIGH

Location:192.168.56.101

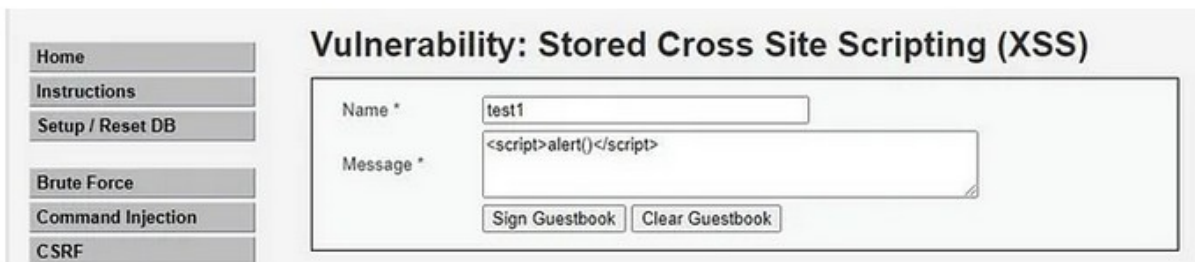
Description:

Stored XSS arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way

Observation:

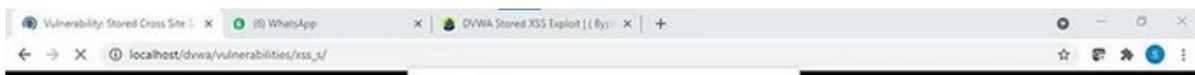


The screenshot shows the DVWA application interface. On the left is a sidebar with navigation links: Home, Instructions, Setup / Reset DB, Brute Force, and Command Injection. The main content area is titled 'Vulnerability: Stored Cross Site Scripting (XSS)'. It contains a form with two input fields: 'Name *' with the value 'test1' and 'Message *' with the value 'test2'. Below the 'Message' field are two buttons: 'Sign Guestbook' and 'Clear Guestbook'. The 'Sign Guestbook' button is highlighted with a red border.



This screenshot shows the same DVWA application interface as the previous one, but with a different payload in the 'Message' field. The 'Message *' field now contains the JavaScript payload '<script>alert()</script>'. The 'Sign Guestbook' button remains highlighted.

Inject the payload `<Script>alert("hacked")</Script>` in the name field and we can enter anything in the message field.



Impact

CVSS Score: 8.0

Confidentiality Impact: High

Integrity Impact: Moderate

Availability Impact: None

Access Complexity: Low

Authentication: Not required

Vulnerability Type(s): Cross-Site Scripting (Stored)

Recommendations:

- Encode all output
- Filter input on entry

Mir Sahil Samiruddin

Email: mirsahil050@gmail.com LinkedIn: www.linkedin.com/in/mir-sahil-606342254

- Use CSP and HttpOnly cookies

References:

<https://portswigger.net/web-security/cross-site-scripting/stored>

6. INSECURE FILE UPLOAD

Risk:HIGH

Location:192.168.56.101

Description:

Whenever the web server accepts a file without validating it or keeping any restriction, it is considered as an unrestricted file upload.

Observation:



The screenshot shows a Notepad window titled '*Untitled - Notepad'. The menu bar includes 'File', 'Edit', 'Format', 'View', and 'Help'. The text area contains the following HTML code:

```
<html>
<body>
<script>alert("You have been Hacked")</script>
</body>
</html>
```

Impact

CVSS Score: 9.9

Confidentiality Impact: High

Integrity Impact: High

Availability Impact: High

Access Complexity: Low

Authentication: Not required

Vulnerability Type(s): Arbitrary File Upload → Remote Code Execution

Recommendations:

- Check MIME, extension, AND file content
- Rename uploaded files
- Store uploads outside web root

References:

https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

<https://cwe.mitre.org/data/definitions/434.html>