Web Application Penetration Testing Report

1. Executive Summary

This security assessment was conducted to evaluate the security posture of a deliberately vulnerable web application using standard penetration testing methodologies. The testing focused on identifying common OWASP Top 10 vulnerabilities that could be exploited by attackers.

The assessment identified critical security weaknesses, including a high-risk SQL Injection vulnerability and weak authentication controls. If exploited, these vulnerabilities could allow unauthorized access to sensitive data and compromise application integrity. Immediate remediation is recommended to reduce overall business risk.

2. Scope and Methodology

2.1 Scope

- Target Application: Damn Vulnerable Web Application (DVWA)
- Target URL: `http://192.168.56.102`
- Testing Type: Web Application Penetration Testing
- Security Level: Low

2.2 Methodology

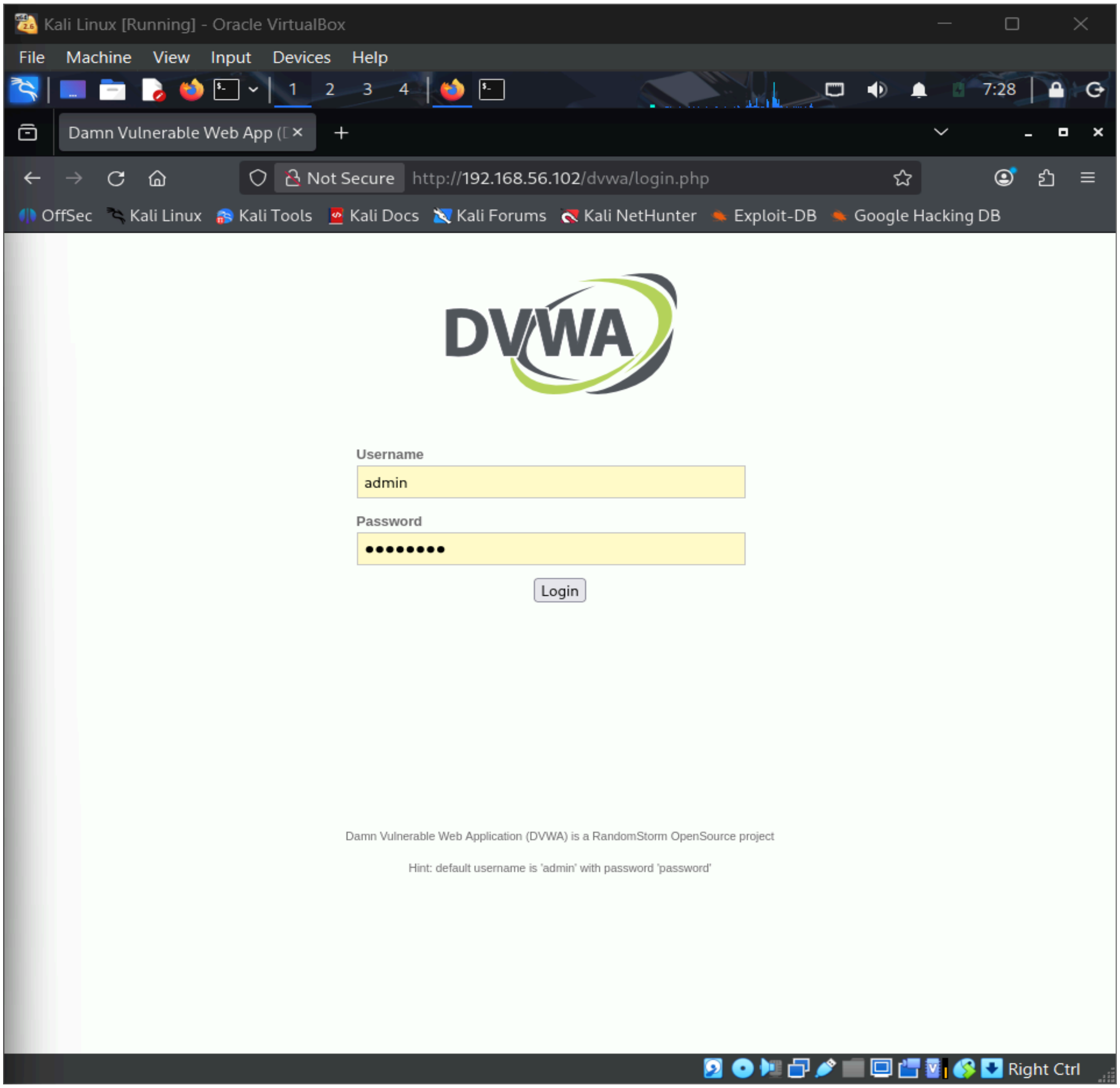
The assessment followed a structured penetration testing approach:

- Manual testing using crafted payloads
- Automated testing using security tools
- Validation of findings through exploitation
- Documentation of risks and remediation steps

3. Technical Findings

During testing, multiple security vulnerabilities were identified due to improper input validation and weak security controls. The most critical finding was a SQL Injection vulnerability that allowed database query manipulation and unauthorized data extraction. Additionally, weak password policies were observed, increasing the risk of credential compromise through brute-force or guessing attacks.

These vulnerabilities indicate insufficient secure coding practices and inadequate authentication enforcement.

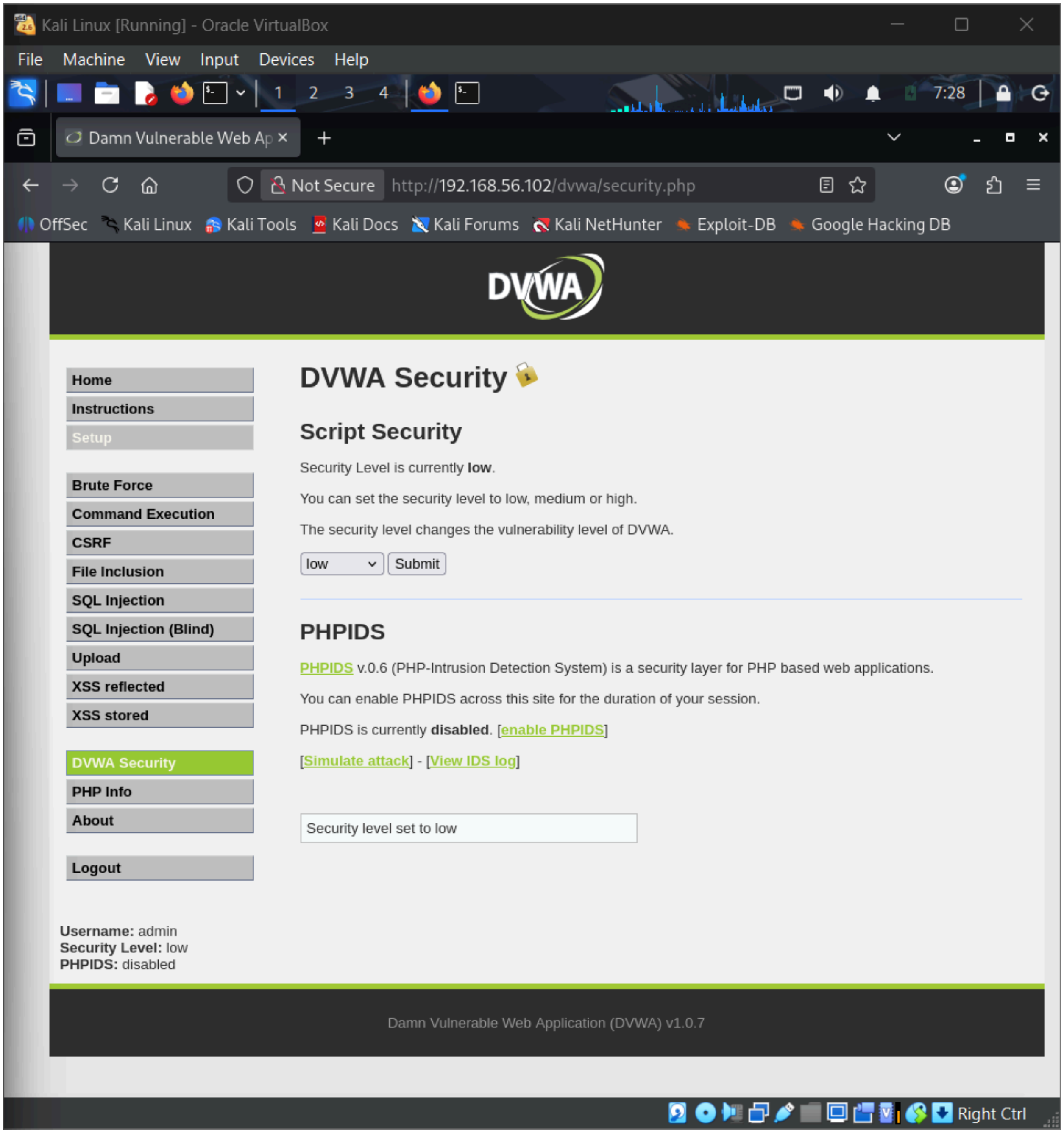

Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
7:28
Damn Vulnerable Web App (I
Not Secure http://192.168.56.102/dvwa/login.php
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
DVWA
Login
Right Ctrl

Username: admin

Password: •••••••••

Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project

Hint: default username is 'admin' with password 'password'



Kali Linux [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Damn Vulnerable Web Ap

Not Secure http://192.168.56.102/dvwa/security.php

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

DVWA Security

Script Security

Security Level is currently **low**.

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.

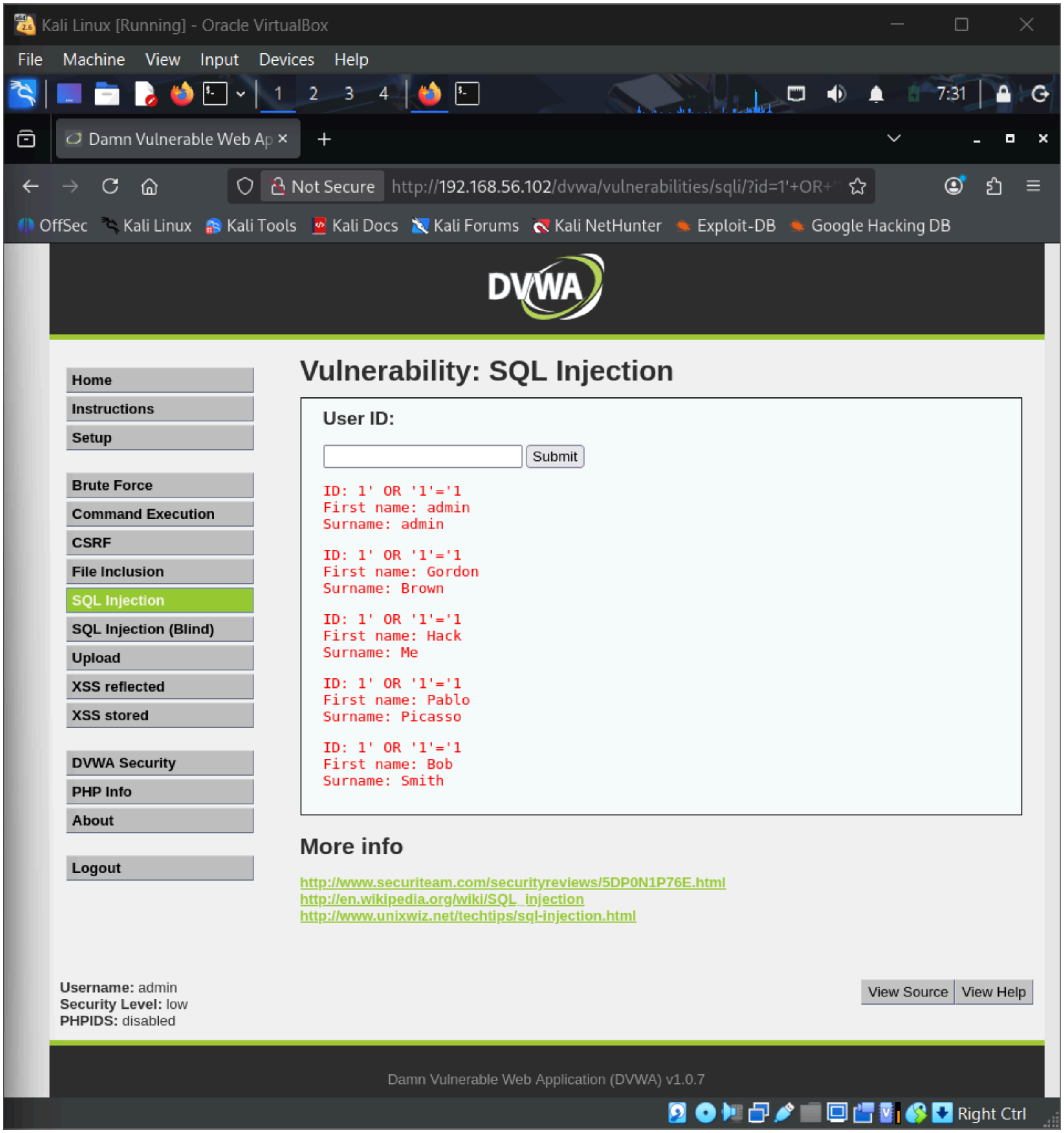
PHPIDS is currently **disabled**. [enable PHPIDS]

[Simulate attack] - [View IDS log]

Security level set to low

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Vulnerability: SQL Injection

User ID:

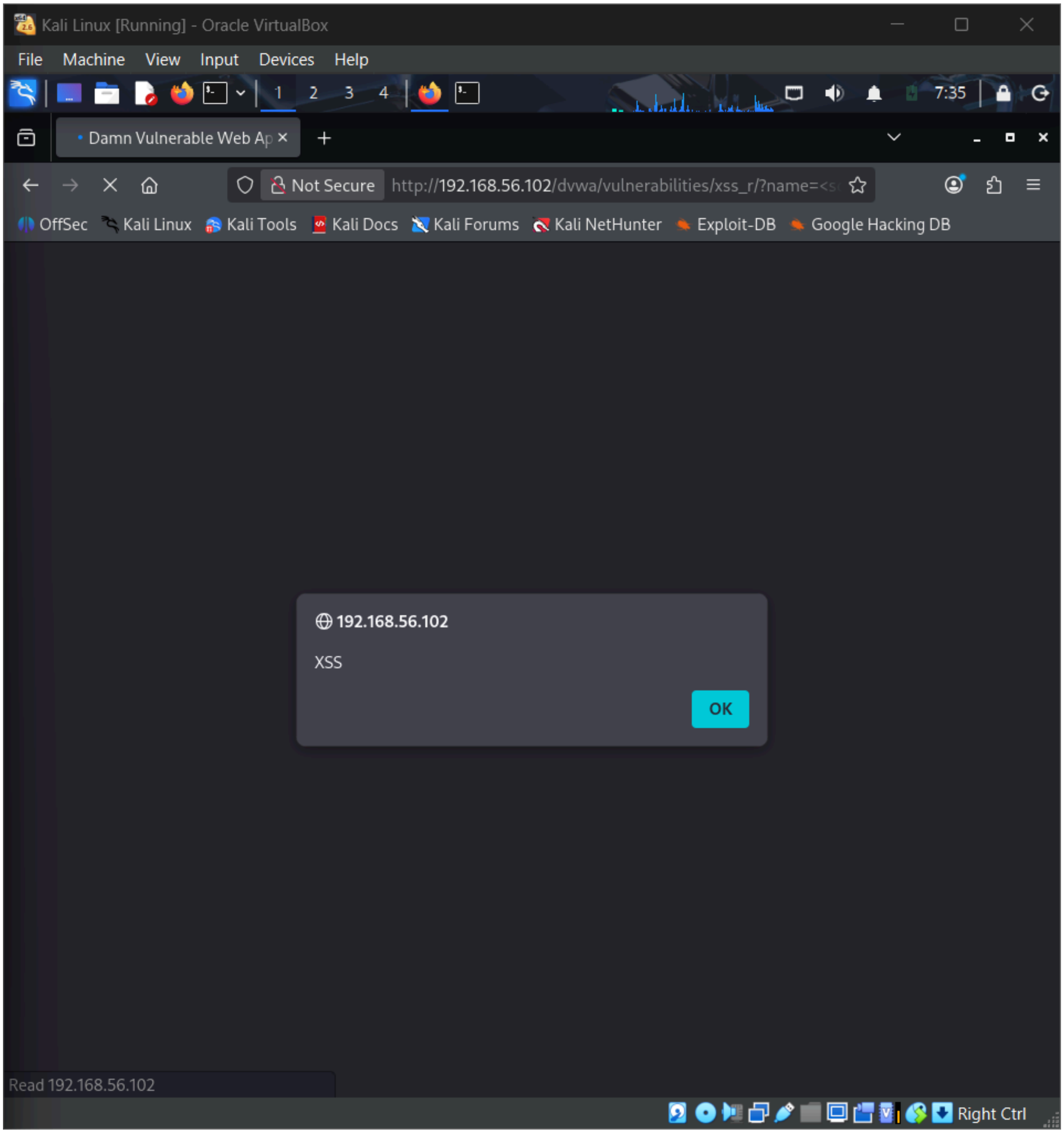
```
ID: 1' OR '1'='1
First name: admin
Surname: admin

ID: 1' OR '1'='1
First name: Gordon
Surname: Brown

ID: 1' OR '1'='1
First name: Hack
Surname: Me

ID: 1' OR '1'='1
First name: Pablo
Surname: Picasso

ID: 1' OR '1'='1
First name: Bob
Surname: Smith
```

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

Damn Vulnerable Web Application (DVWA) v1.0.7



Kali Linux [Running] - Oracle VirtualBox

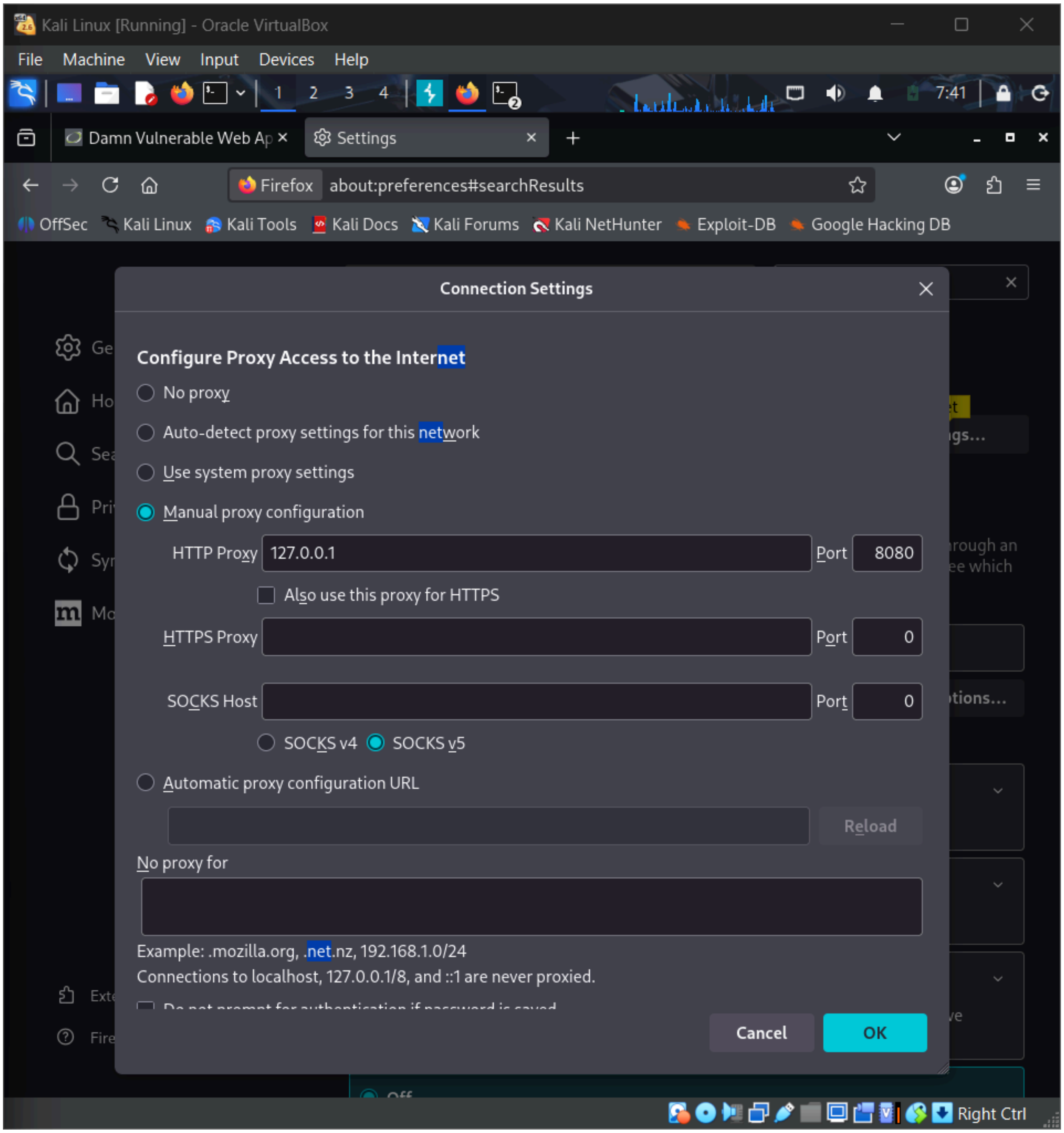
Damn Vulnerable Web Ap

http://192.168.56.102/dvwa/vulnerabilities/xss_r/?name=<s

192.168.56.102

XSS

Read 192.168.56.102



Connection Settings

Configure Proxy Access to the Internet

○ No proxy

○ Auto-detect proxy settings for this network

○ Use system proxy settings

● Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

□ Also use this proxy for HTTPS

HTTPS Proxy Port 0

SOCKS Host Port 0

○ SOCKS v4 ● SOCKS v5

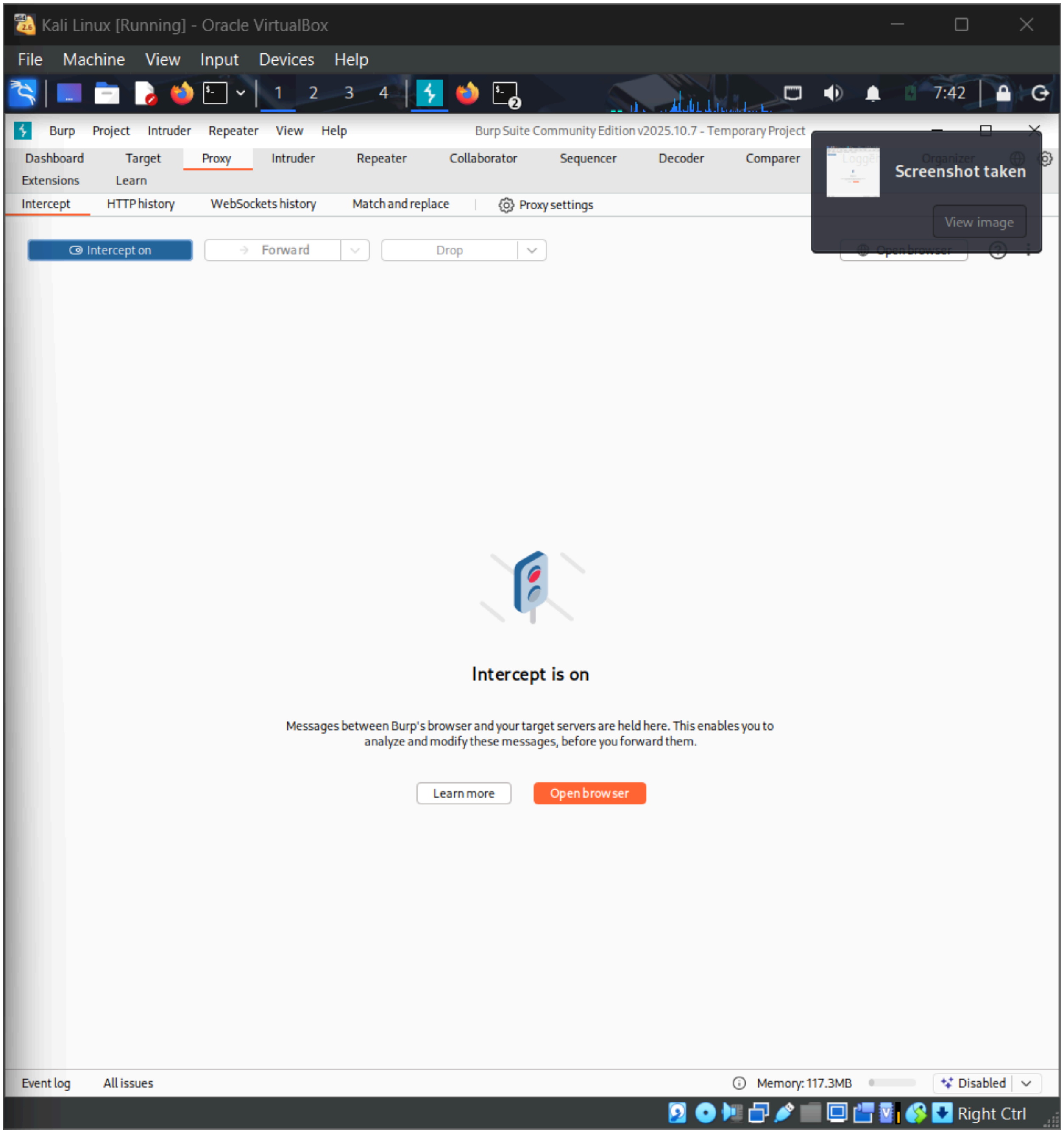
○ Automatic proxy configuration URL

No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

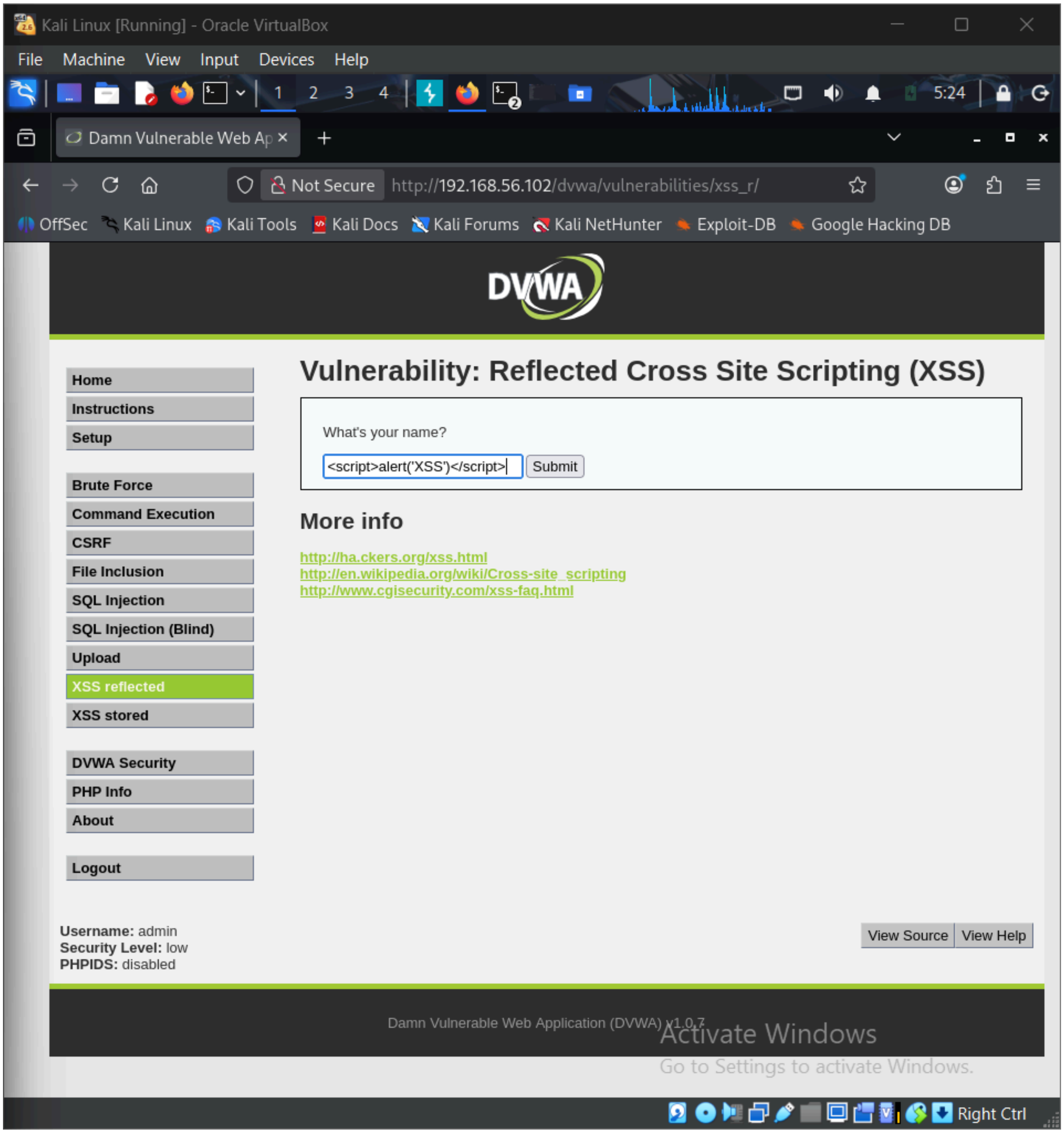
Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.

□ Do not prompt for authentication if password is saved


Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
7:42
Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.7 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer
Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Screenshot taken
View image
Open browser
Learn more
Open browser
Event log All issues
Memory: 117.3MB
Disabled
Right Ctrl

Intercept is on

Messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.



http://192.168.56.102/dvwa/vulnerabilities/xss_r/

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

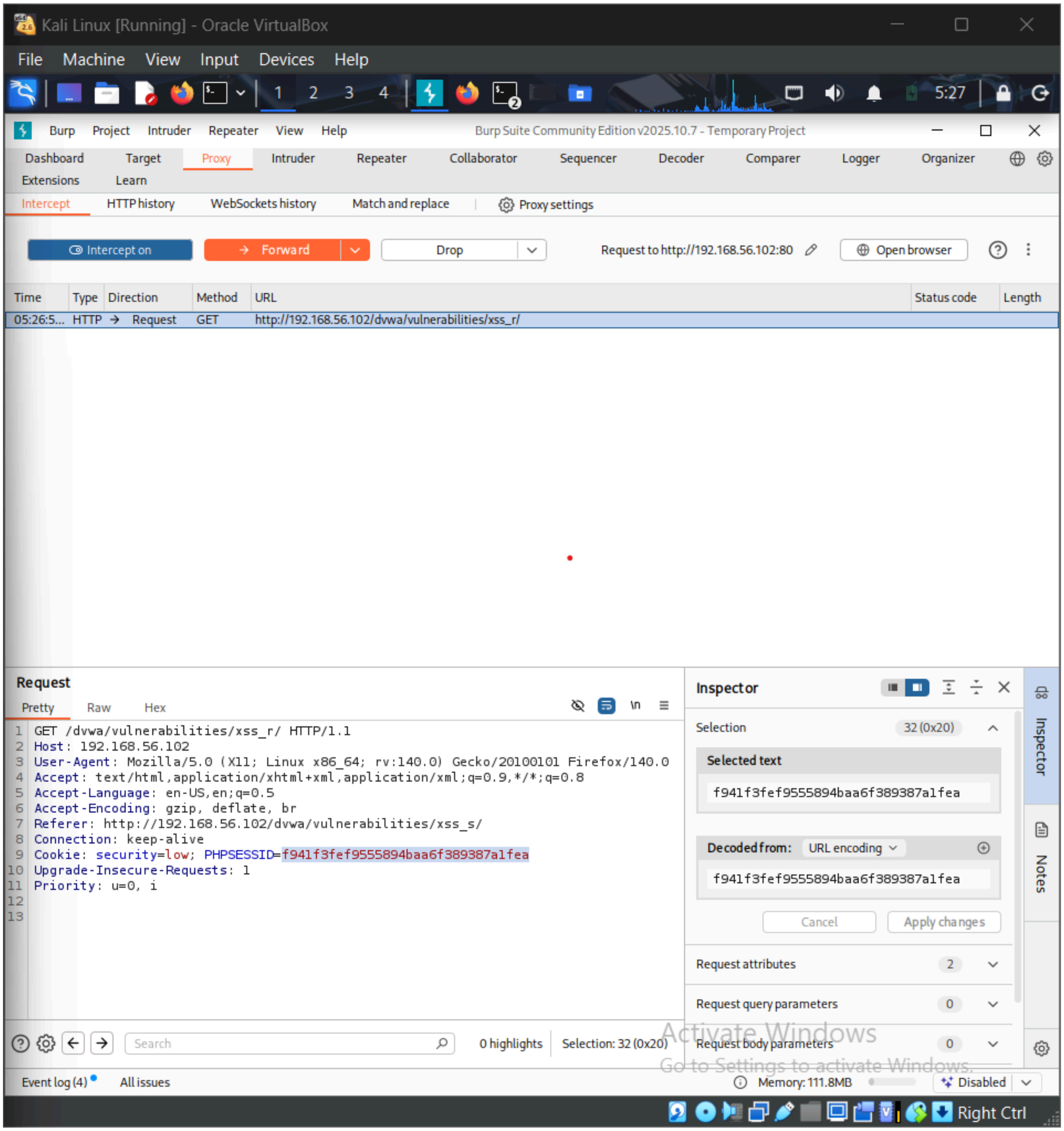
<script>alert('XSS')</script>

More info

http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



Time	Type	Direction	Method	URL	Status code	Length
05:26:5...	HTTP	→ Request	GET	http://192.168.56.102/dvwa/vulnerabilities/xss_r/		

```
GET /dvwa/vulnerabilities/xss_r/ HTTP/1.1
Host: 192.168.56.102
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://192.168.56.102/dvwa/vulnerabilities/xss_s/
Connection: keep-alive
Cookie: security=low; PHPSESSID=f941f3fef9555894baa6f389387a1fea
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

Selection 32 (0x20)

Selected text

f941f3fef9555894baa6f389387a1fea

Decoded from: URL encoding

f941f3fef9555894baa6f389387a1fea

Request attributes 2

Request query parameters 0

Request body parameters 0

4. Findings Table

Finding ID	Vulnerability	CVSS Score	Risk Level	Remediation
F001	SQL Injection	9.1	Critical	Implement input validation and prepared statements
F002	Weak Password Policy	7.5	High	Enforce strong password complexity rules

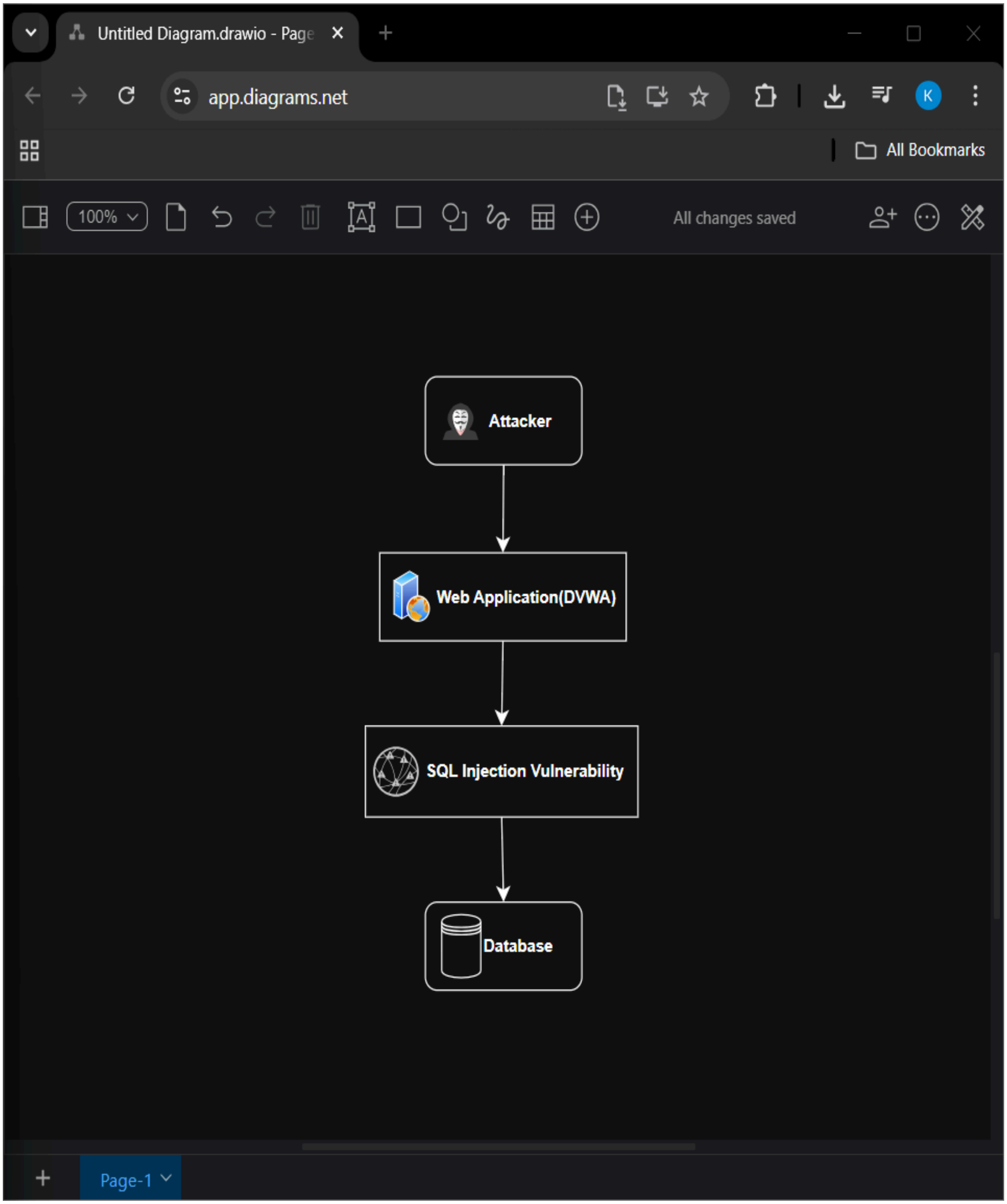
5. Remediation Plan

To mitigate the identified vulnerabilities, the following remediation actions are recommended:

- Implement strict server-side input validation
- Use parameterized database queries
- Enforce strong password policies
- Conduct regular security testing

6. Network Attack Path Visualization

- Network attack path illustrating exploitation flow.



7. Management Brief (Non-Technical Summary – 100 Words)

The security assessment revealed serious weaknesses within the web application that could be exploited to access sensitive information. A critical SQL Injection vulnerability allows attackers to retrieve database records without authorization, while weak password controls increase the risk of account compromise. These issues could lead to data breaches, regulatory penalties, and reputational damage. Management is advised to prioritize remediation by enforcing secure coding practices, strengthening authentication mechanisms, and conducting regular security testing.

8. Conclusion

The penetration testing exercise demonstrated how common web vulnerabilities can be exploited when secure development practices are not followed. Implementing the recommended remediation steps will significantly improve the application’s security posture.