Task 9

Executive Summary

This report presents the findings of a comprehensive vulnerability assessment conducted on the locally hosted vuln-bank web application. The evaluation focused on identifying common web vulnerabilities, including authentication flaws, input validation issues, file upload vulnerabilities, and authorization weaknesses. The assessment identified 2 critical vulnerabilities: 1 classified as High severity and 1 as Medium severity. Detailed remediation recommendations are included to mitigate these risks effectively.

Scope

- **In-Scope:** Local vuln-bank instance operating via Docker.
- **Out of Scope:** External networks, third-party systems, or any non-local infrastructure.

Environment & Tools

- **Target Runtime:** Docker with docker-compose (local environment).
- **OS Used for Testing:** Linux (e.g., Kali Linux).

Methodology

Step 1: Install Required Tools

- Install Git from: https://git-scm.com/downloads (facilitates source code retrieval from GitHub).
- Install Docker Desktop from: https://www.docker.com/products/docker-desktop (enables secure execution of applications in containers).
- Restart the system following installation to ensure optimal configuration.

Step 2: Clone the Vuln-Bank Project

1. Launch the Command Prompt (Windows) or Terminal (Mac/Linux).
2. Execute the following command:
   text
   ```
   git clone https://github.com/Commando-X/vuln-bank.git
   ```
3. This action downloads the project directory to the local machine.
4. Navigate to the project directory:
   text
   ```
   cd vuln-bank
   ```

Step 3: Deploy Vuln-Bank with Docker

1. Initiate the application by running:
   text
   ```
   docker-compose up --build -d
   ```
2. This command builds and launches all application components as outlined in the project documentation.

Step 4: Access the Vuln-Bank Web Application

- Open a web browser (e.g., Chrome or Edge).
- Navigate to: http://localhost:5000 to access the application interface.

Step 5: Identify and Document Vulnerabilities

1. Navigate through all application pages (login, registration, account management, simulated transactions, file uploads, etc.).
2. Detect potential security weaknesses, such as:

- Default credentials (test with common combinations like 'admin/admin').
- SQL Injection vulnerabilities (test inputs such as ' OR '1'='1 in forms).

Step 6: Terminate the Application

- To shut down the application, return to the vuln-bank directory in the terminal and execute:
  text
  ```
  docker-compose down
  ```

- This command safely terminates all active containers.

Summary of Findings

ID	Title	Severity	Location
1	SQL Injection in Login	High	POST /login
2	Weak Password Reset (3-digit PIN)	Medium	/reset

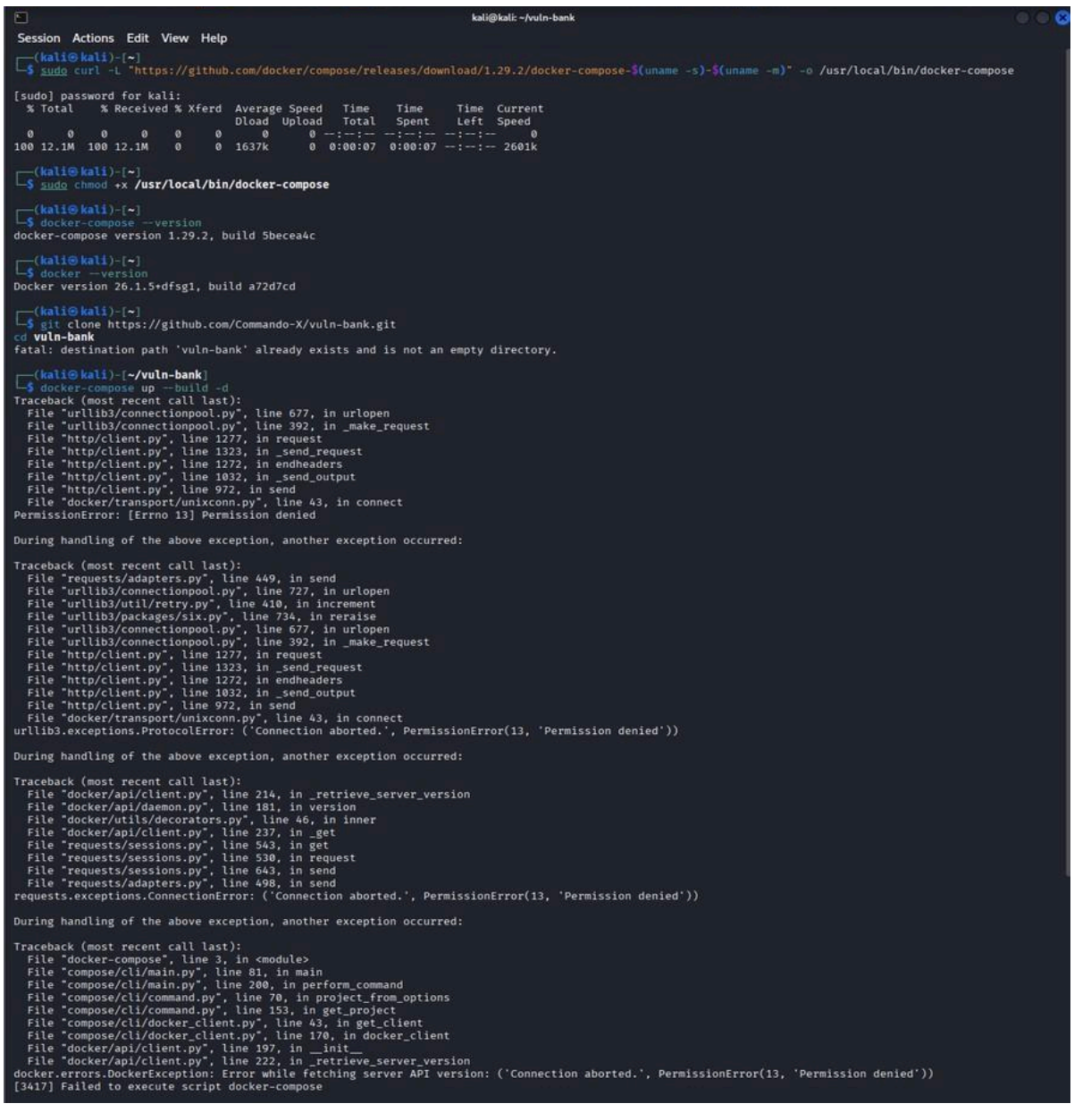
Conclusion

The vulnerability assessment of the vuln-bank web application revealed significant security deficiencies in authentication, data protection, and transaction processing. These vulnerabilities could potentially allow unauthorized access to sensitive data, circumvention of security controls, and manipulation of financial operations. Immediate action is recommended, including the implementation of robust input validation, enhanced authentication mechanisms, secure session management, and vigilant monitoring of AI-driven features. This report underscores the critical need for ongoing security testing and proactive improvements to safeguard against evolving cyber threats.

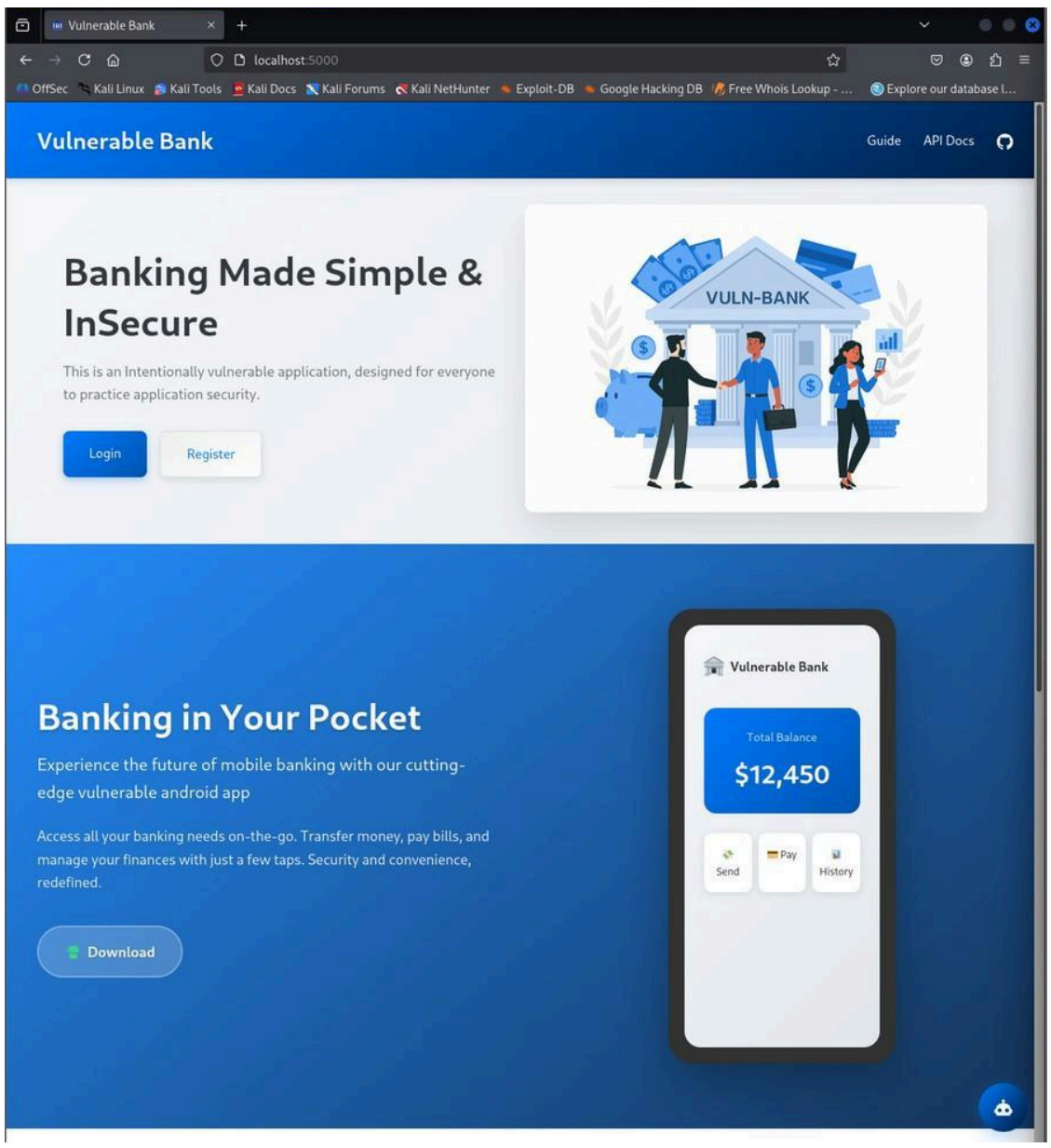
References

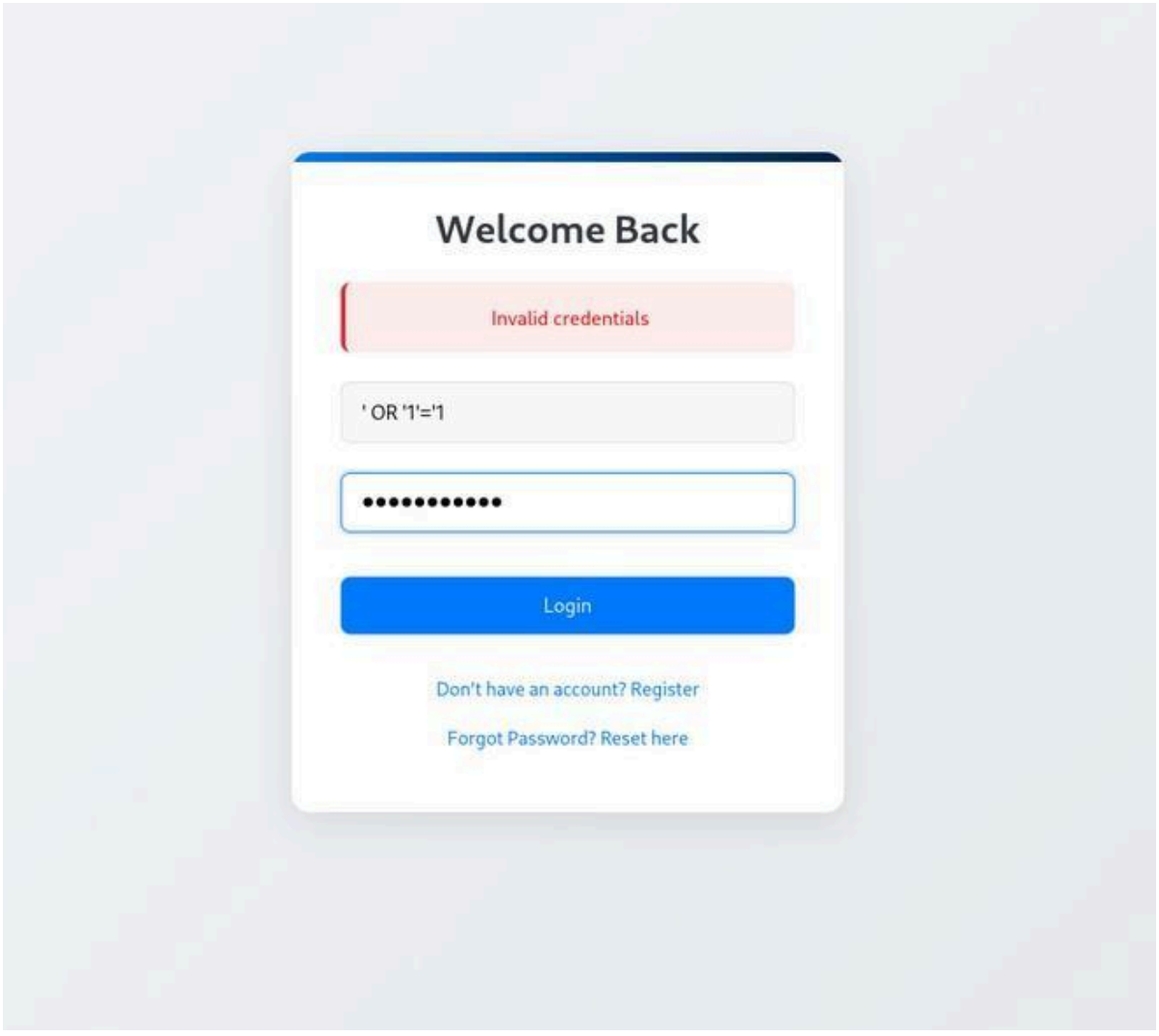
This assessment leveraged the vuln-bank repository README for setup instructions and documentation of intentionally implemented vulnerabilities. Cite the source as: Commando-X vuln-bank README.

Screenshots:



```
┌──(kali㉿kali)-[~/vuln-bank]
└─$ docker-compose down
Traceback (most recent call last):
  File "urllib3/connectionpool.py", line 677, in urlopen
  File "urllib3/connectionpool.py", line 392, in _make_request
  File "http/client.py", line 1277, in request
  File "http/client.py", line 1323, in _send_request
  File "http/client.py", line 1272, in endheaders
  File "http/client.py", line 1032, in _send_output
  File "http/client.py", line 972, in send
  File "docker/transport/unixconn.py", line 43, in connect
PermissionError: [Errno 13] Permission denied

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "requests/adapters.py", line 449, in send
  File "urllib3/connectionpool.py", line 727, in urlopen
  File "urllib3/util/retry.py", line 410, in increment
  File "urllib3/packages/six.py", line 734, in reraise
  File "urllib3/connectionpool.py", line 677, in urlopen
  File "urllib3/connectionpool.py", line 392, in _make_request
  File "http/client.py", line 1277, in request
  File "http/client.py", line 1323, in _send_request
  File "http/client.py", line 1272, in endheaders
  File "http/client.py", line 1032, in _send_output
  File "http/client.py", line 972, in send
  File "docker/transport/unixconn.py", line 43, in connect
urllib3.exceptions.ProtocolError: ('Connection aborted.', PermissionError(13, 'Permission denied'))

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "docker/api/client.py", line 214, in _retrieve_server_version
  File "docker/api/daemon.py", line 181, in version
  File "docker/utils/decorators.py", line 46, in inner
  File "docker/api/client.py", line 237, in _get
  File "requests/sessions.py", line 543, in get
  File "requests/sessions.py", line 530, in request
  File "requests/sessions.py", line 643, in send
  File "requests/adapters.py", line 498, in send
requests.exceptions.ConnectionError: ('Connection aborted.', PermissionError(13, 'Permission denied'))

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "docker-compose", line 3, in <module>
  File "compose/cli/main.py", line 81, in main
  File "compose/cli/main.py", line 200, in perform_command
  File "compose/cli/command.py", line 70, in project_from_options
  File "compose/cli/command.py", line 153, in get_project
  File "compose/cli/docker_client.py", line 43, in get_client
  File "compose/cli/docker_client.py", line 170, in docker_client
  File "docker/api/client.py", line 197, in __init__
  File "docker/api/client.py", line 222, in _retrieve_server_version
docker.errors.DockerException: Error while fetching server API version: ('Connection aborted.', PermissionError(13, 'Permission denied'))
[7496] Failed to execute script docker-compose
```




Welcome Back
Invalid credentials
' OR '1'='1
Login
Don't have an account? Register
Forgot Password? Reset here