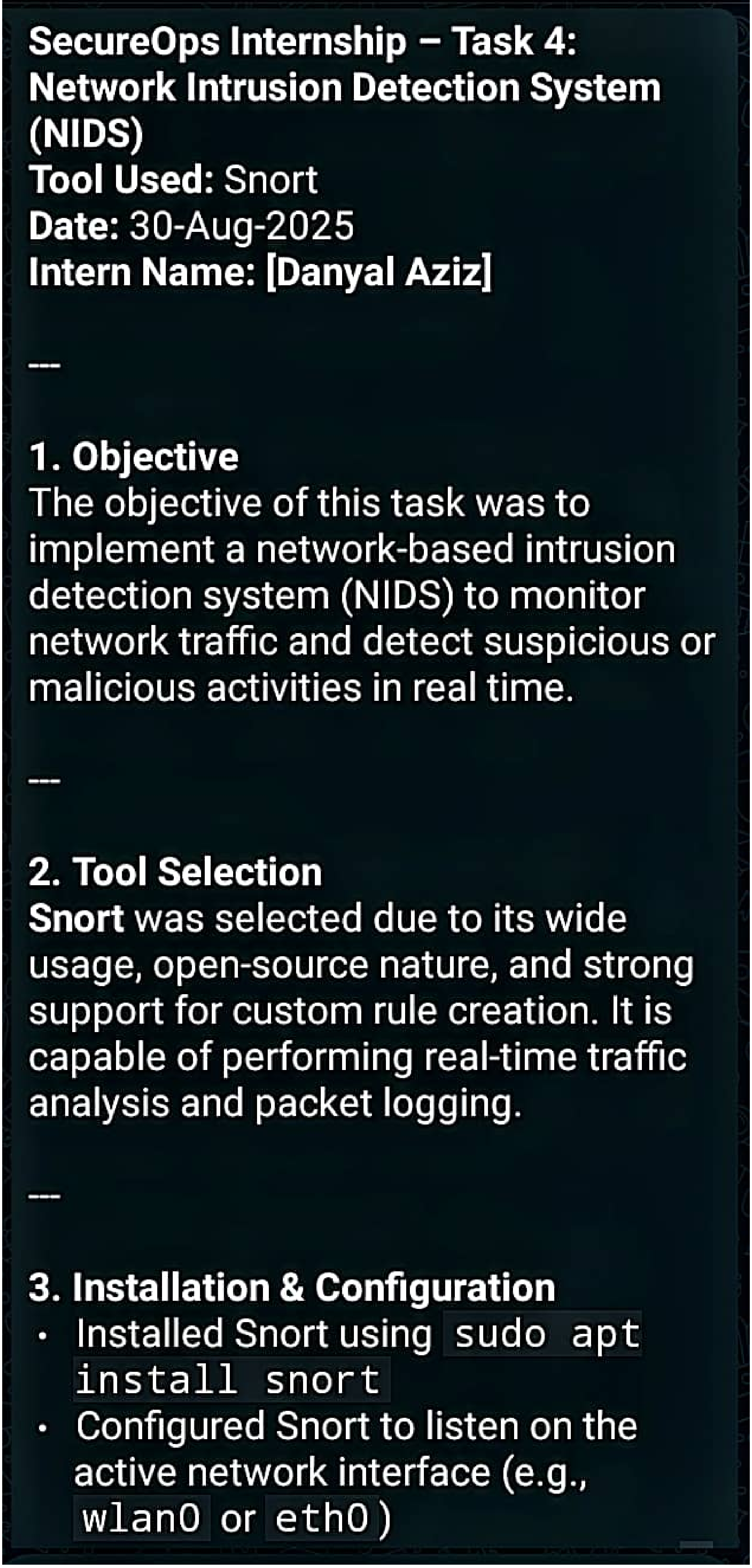


SecureOps Internship – Task 4: Network Intrusion Detection System (NIDS)
Tool Used: Snort
Date: 30-Aug-2025
Intern Name: [Danyal Aziz]

---

1. Objective

The objective of this task was to implement a network-based intrusion detection system (NIDS) to monitor network traffic and detect suspicious or malicious activities in real time.

---

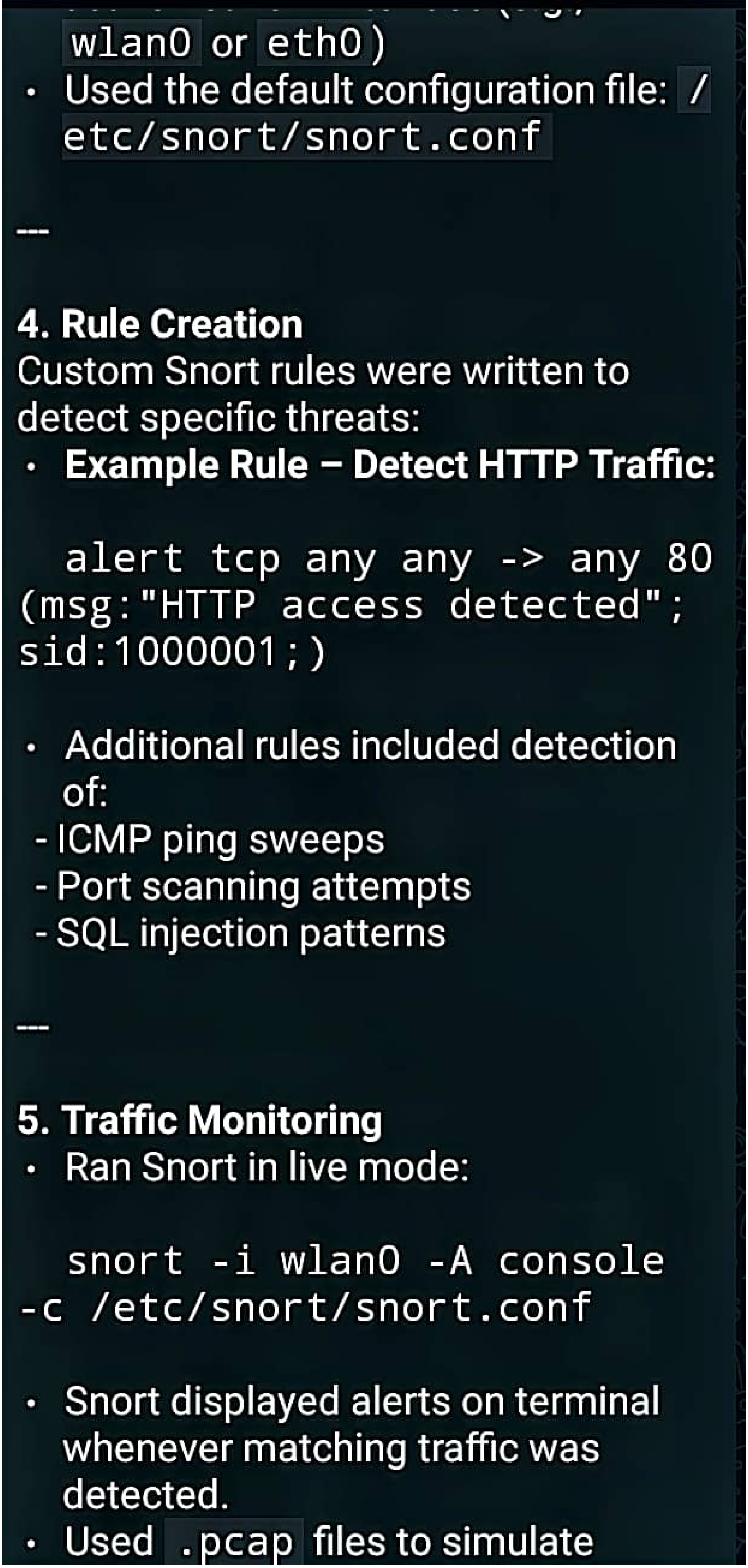
2. Tool Selection

Snort was selected due to its wide usage, open-source nature, and strong support for custom rule creation. It is capable of performing real-time traffic analysis and packet logging.

---

3. Installation & Configuration

- Installed Snort using `sudo apt install snort`
- Configured Snort to listen on the active network interface (e.g., `wlan0` or `eth0`)



wlan0 or eth0)
- Used the default configuration file: /etc/snort/snort.conf

---

4. Rule Creation

Custom Snort rules were written to detect specific threats:

- **Example Rule – Detect HTTP Traffic:**

```
alert tcp any any -> any 80
(msg:"HTTP access detected";
sid:1000001;)
```

- Additional rules included detection of:

 - ICMP ping sweeps
 - Port scanning attempts
 - SQL injection patterns

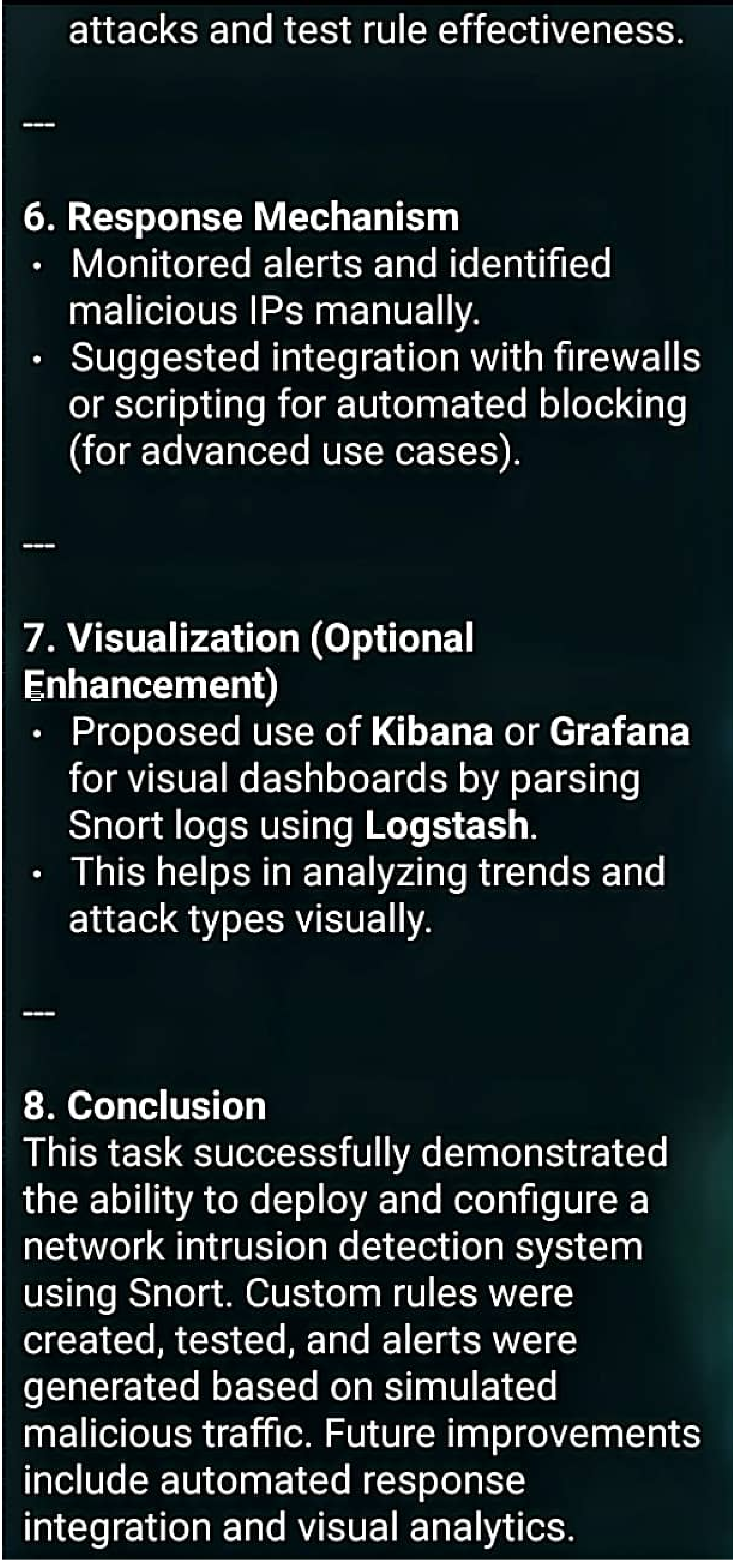
---

5. Traffic Monitoring

- Ran Snort in live mode:

```
snort -i wlan0 -A console
-c /etc/snort/snort.conf
```

- Snort displayed alerts on terminal whenever matching traffic was detected.
- Used .pcap files to simulate



attacks and test rule effectiveness.

---

6. **Response Mechanism**

- Monitored alerts and identified malicious IPs manually.
- Suggested integration with firewalls or scripting for automated blocking (for advanced use cases).

---

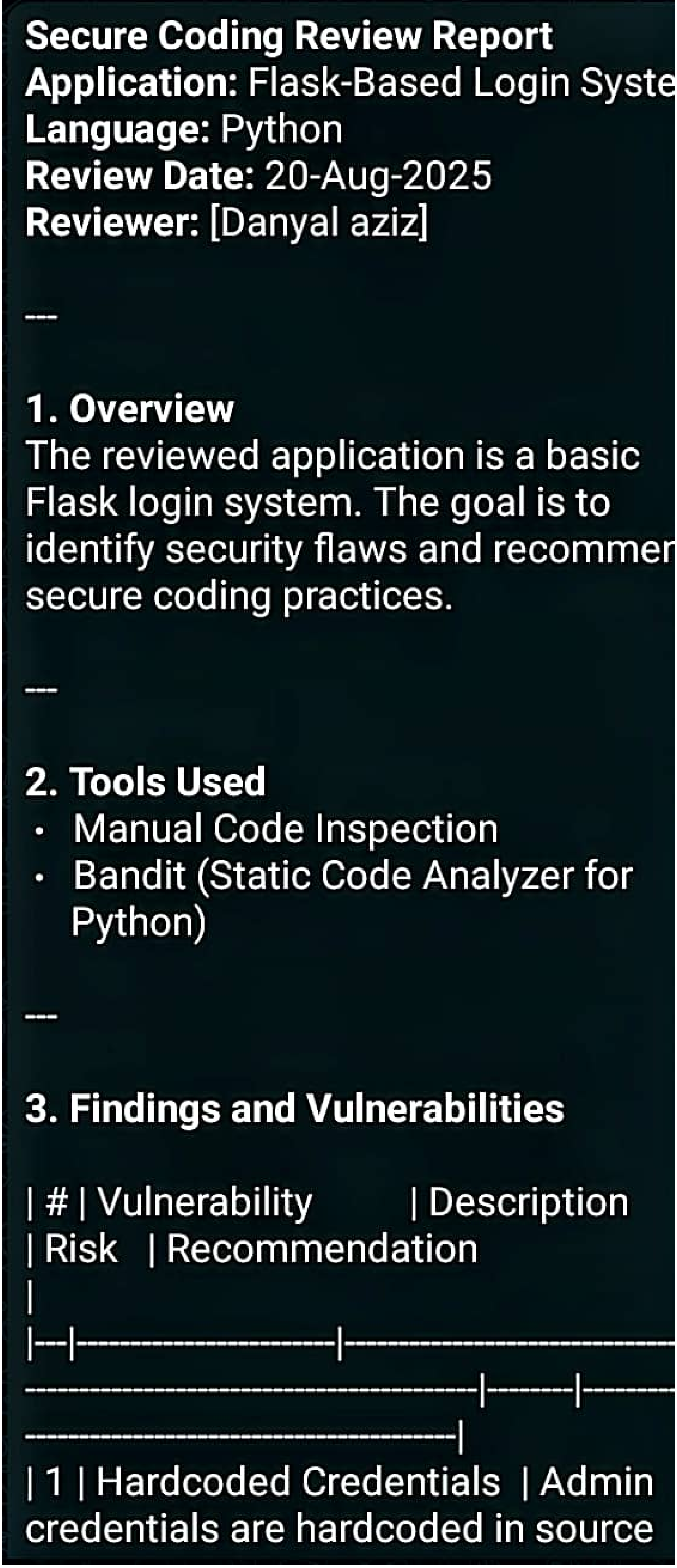
7. **Visualization (Optional Enhancement)**

- Proposed use of **Kibana** or **Grafana** for visual dashboards by parsing Snort logs using **Logstash**.
- This helps in analyzing trends and attack types visually.

---

8. **Conclusion**

This task successfully demonstrated the ability to deploy and configure a network intrusion detection system using Snort. Custom rules were created, tested, and alerts were generated based on simulated malicious traffic. Future improvements include automated response integration and visual analytics.



Secure Coding Review Report
Application: Flask-Based Login Syste
Language: Python
Review Date: 20-Aug-2025
Reviewer: [Danyal aziz]

---

1. Overview

The reviewed application is a basic Flask login system. The goal is to identify security flaws and recommen secure coding practices.

---

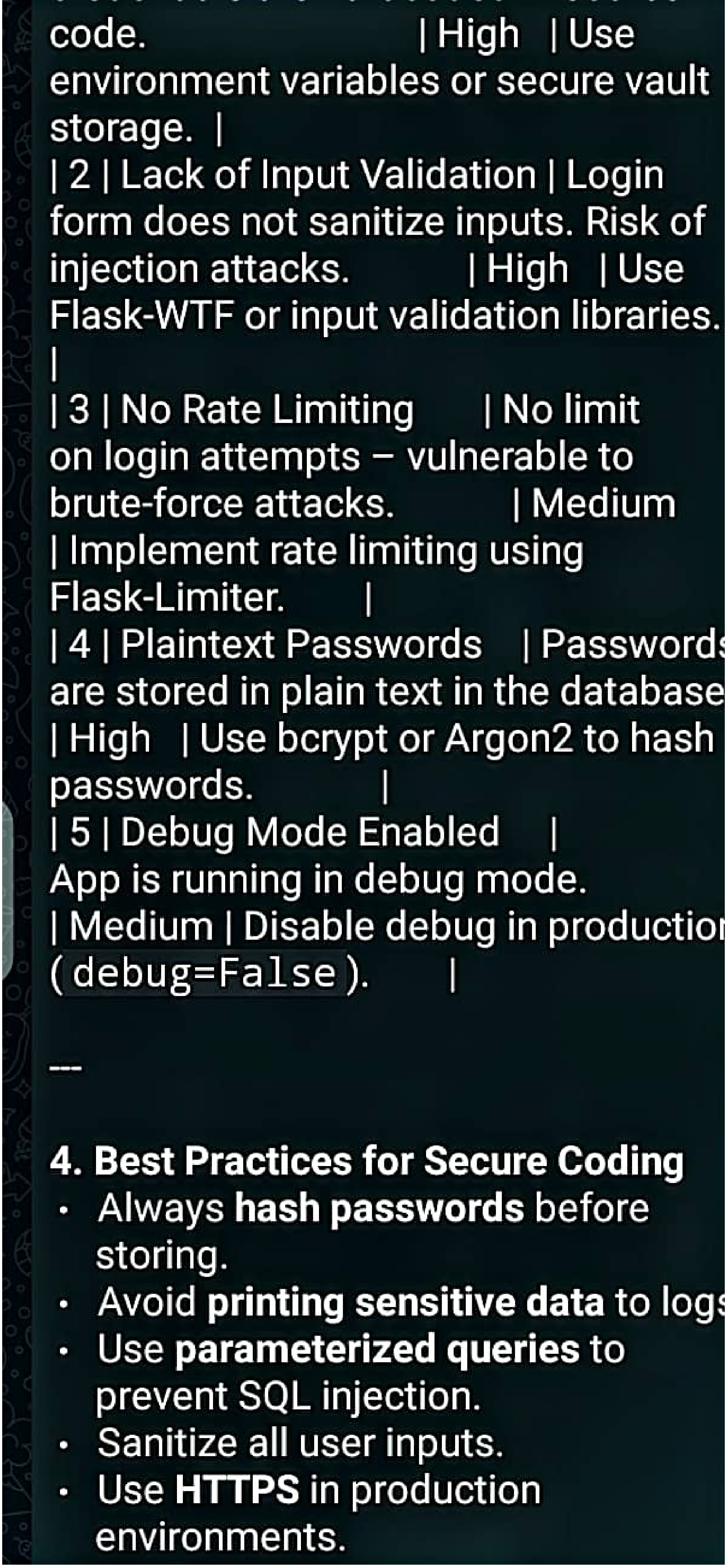
2. Tools Used

- Manual Code Inspection
- Bandit (Static Code Analyzer for Python)

---

3. Findings and Vulnerabilities

```
| # | Vulnerability         | Description
| Risk   | Recommendation
|
|---|---------------------|------------------------------
---------------------------------------------|--------|---------
-------------------------------------------|
| 1 | Hardcoded Credentials  | Admin
credentials are hardcoded in source
```

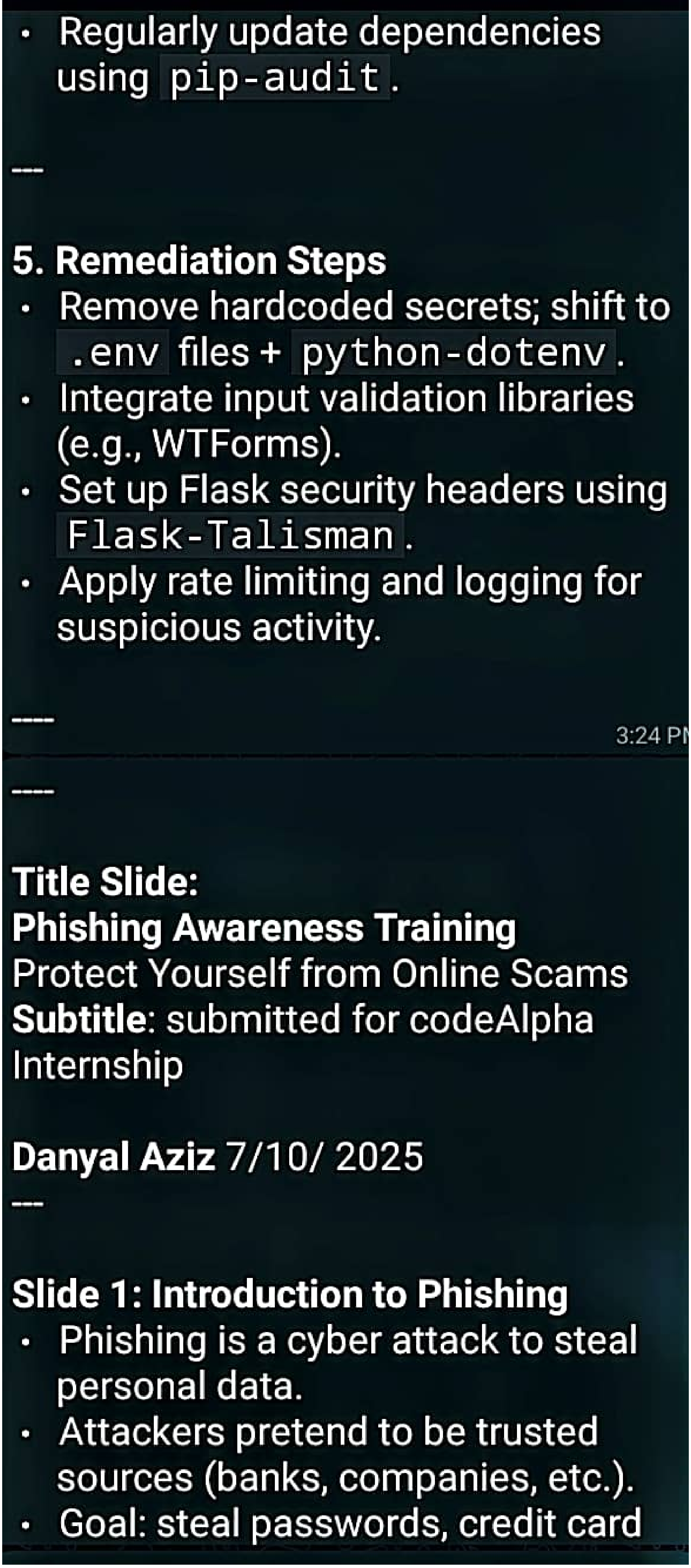


		code.	High	Use environment variables or secure vault storage.
2	Lack of Input Validation	Login form does not sanitize inputs. Risk of injection attacks.	High	Use Flask-WTF or input validation libraries.
3	No Rate Limiting	No limit on login attempts – vulnerable to brute-force attacks.	Medium	Implement rate limiting using Flask-Limiter.
4	Plaintext Passwords	Passwords are stored in plain text in the database	High	Use bcrypt or Argon2 to hash passwords.
5	Debug Mode Enabled	App is running in debug mode.	Medium	Disable debug in production (`debug=False`).

---

4. Best Practices for Secure Coding

- Always **hash passwords** before storing.
- Avoid **printing sensitive data** to logs
- Use **parameterized queries** to prevent SQL injection.
- Sanitize all user inputs.
- Use **HTTPS** in production environments.



- Regularly update dependencies using `pip-audit`.

---

5. Remediation Steps

- Remove hardcoded secrets; shift to `.env` files + `python-dotenv`.
- Integrate input validation libraries (e.g., WTForms).
- Set up Flask security headers using `Flask-Talisman`.
- Apply rate limiting and logging for suspicious activity.

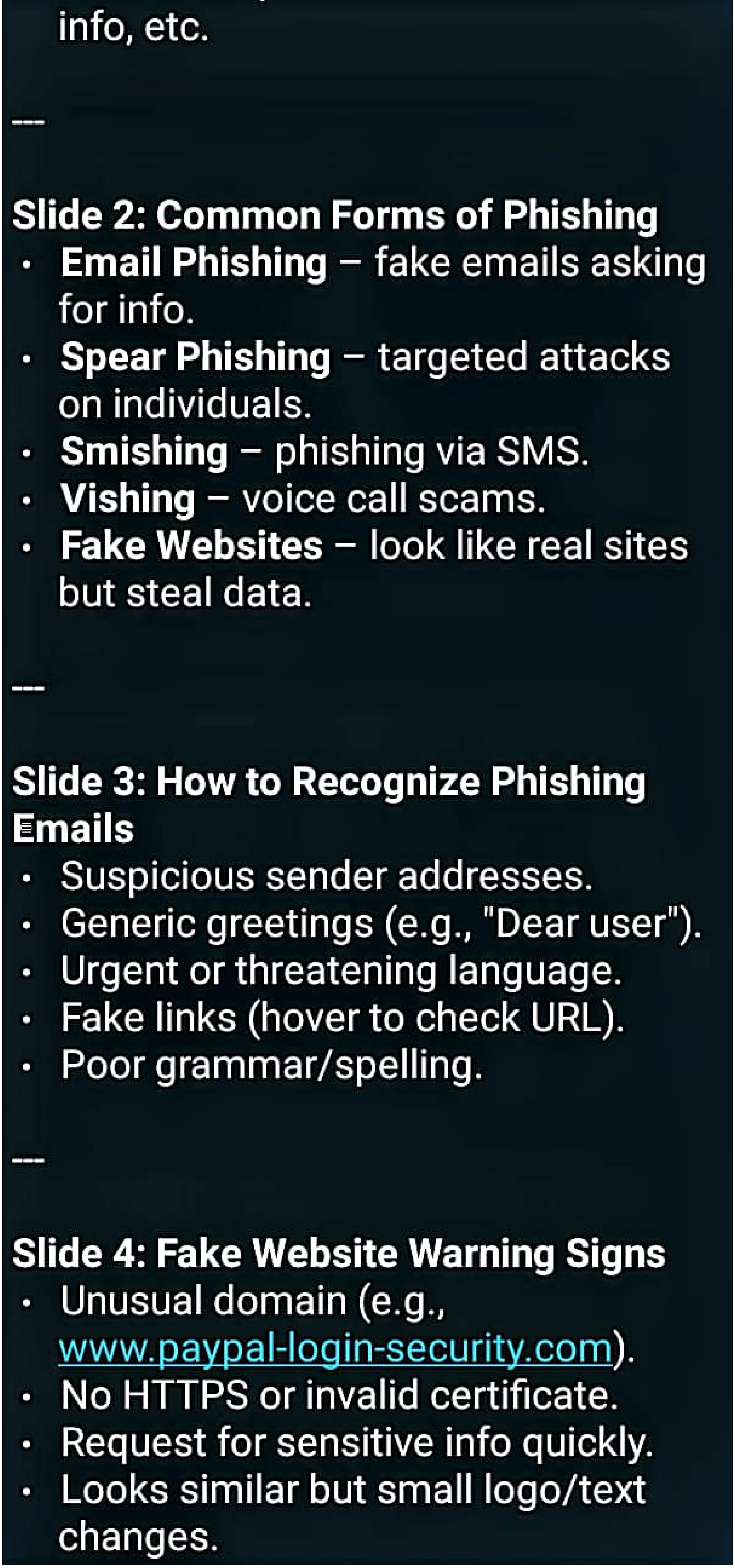
3:24 P

Title Slide:
Phishing Awareness Training
Protect Yourself from Online Scams
Subtitle: submitted for codeAlpha Internship

Danyal Aziz 7/10/ 2025

Slide 1: Introduction to Phishing

- Phishing is a cyber attack to steal personal data.
- Attackers pretend to be trusted sources (banks, companies, etc.).
- Goal: steal passwords, credit card



info, etc.

Slide 2: Common Forms of Phishing

- **Email Phishing** – fake emails asking for info.
- **Spear Phishing** – targeted attacks on individuals.
- **Smishing** – phishing via SMS.
- **Vishing** – voice call scams.
- **Fake Websites** – look like real sites but steal data.

Slide 3: How to Recognize Phishing Emails

- Suspicious sender addresses.
- Generic greetings (e.g., "Dear user").
- Urgent or threatening language.
- Fake links (hover to check URL).
- Poor grammar/spelling.

Slide 4: Fake Website Warning Signs

- Unusual domain (e.g., www.paypal-login-security.com).
- No HTTPS or invalid certificate.
- Request for sensitive info quickly.
- Looks similar but small logo/text changes.

Slide 5: Social Engineering Tactics

- **Fear** – "Your account will be locked!
- **Greed** – "You've won a prize!"
- **Urgency** – "Act now or lose access."
- **Trust** – Impersonating bosses, banks, or government.

Slide 6: How to Protect Yourself

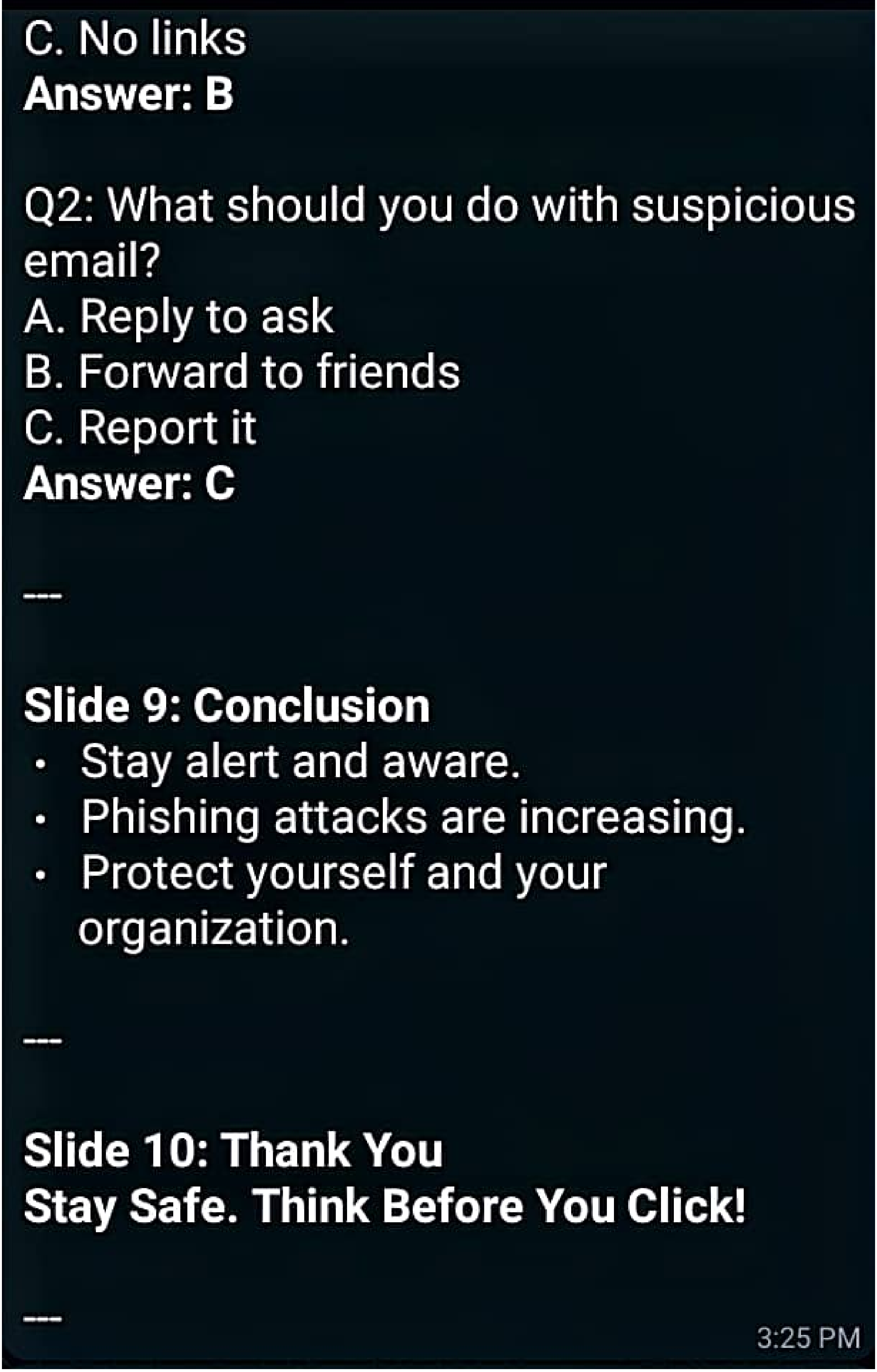
- Never click unknown links.
- Verify sender's identity.
- Use 2-factor authentication.
- Report suspicious messages.
- Use up-to-date antivirus and browse protection.

Slide 7: Real-World Example

A user received an email from "Google Support" asking to reset password.
When clicked, it redirected to a fake page.
Result: Gmail hacked, data stolen.

Slide 8: Quiz (Interactive)

Q1: What is a sign of phishing email?
A. Correct logo
B. Urgent language



C. No links

Answer: B

Q2: What should you do with suspicious email?

A. Reply to ask

B. Forward to friends

C. Report it

Answer: C

Slide 9: Conclusion

- Stay alert and aware.
- Phishing attacks are increasing.
- Protect yourself and your organization.

Slide 10: Thank You

Stay Safe. Think Before You Click!

3:25 PM