Task 11: Phishing Attack Simulation & Detection

1. Understand Phishing Attacks

Phishing is a type of **social engineering attack** in which an attacker impersonates a trusted entity to trick users into revealing sensitive information such as usernames, passwords, or financial details. These attacks usually occur through emails, fake websites, or messages that create a sense of urgency or fear.

Objective of this task:
To understand how phishing attacks work and how they can be detected and prevented through awareness.

Difficulty Faced in Kali Linux

- No direct issue at this stage (conceptual learning).
- Required understanding ethical boundaries before performing simulations.

2. Create Fake Email Template

A fake phishing email template was created using **GoPhish** to simulate a real-world phishing scenario. The email used urgent language such as *"Verify your account immediately"* to psychologically pressure users into clicking a malicious link.

Tools Used

- GoPhish (Email Template module)

Difficulty Faced

- Initial difficulty accessing GoPhish due to **internet connectivity and DNS issues in Kali Linux**.
- GitHub repository could not be cloned until network configuration was fixed.

3. Setup Landing Page

A fake landing page was designed to imitate an account verification page. This page contained input fields for username and password to demonstrate how attackers capture credentials.

Features Enabled

- Capture submitted data
- Capture passwords (for lab simulation only)

Difficulty Faced

- Understanding HTML structure for a simple login page.
- Ensuring the landing page was correctly linked with the email template using GoPhish variables.

4. Send Test Phishing Email

A test phishing email was sent using **MailHog**, a local SMTP testing server, instead of real email services. This ensured that no real users were affected.

SMTP Configuration

- Host: 127.0.0.1
- Port: 1025
- TLS: Disabled

Difficulty Faced

- mailhog package was **not available via apt in Kali Linux**.
- Required manual installation of MailHog binary from GitHub.
- Understanding SMTP testing vs real email sending.

5. Track Responses

GoPhish dashboard was used to monitor:

- Email opened
- Link clicked
- Data submitted

These metrics help measure how users respond to phishing attempts.

Difficulty Faced

- Initial confusion understanding campaign metrics.
- Required multiple test runs to observe response changes.

6. Identify Red Flags

The following phishing red flags were identified during analysis:

- Urgent or threatening language
- Suspicious sender email
- Fake or shortened URLs
- Requests for sensitive information
- Lack of personalization

- Grammar and formatting issues

Difficulty Faced

- Differentiating between legitimate security emails and phishing emails required careful analysis.

7. Learn Prevention Methods

The task emphasized the importance of both **technical and human-based prevention methods**.

Prevention Techniques Learned

- User awareness training
- Email filtering systems
- Multi-factor authentication (MFA)
- URL inspection
- Zero-trust security model

Difficulty Faced

- Understanding how human behavior plays a major role despite technical security controls.

8. Document Simulation

The entire phishing simulation was documented with:

- Step-by-step methodology
- Screenshots
- Results analysis
- Detection techniques
- Prevention strategies

Documentation was prepared for **GitHub submission** and report evaluation.

Difficulty Faced

- Structuring the report professionally
- Selecting relevant screenshots
- Explaining technical errors faced in Kali Linux clearly

Conclusion

This task provided hands-on experience in understanding phishing attacks, simulating them safely using Kali Linux, and learning how to detect and prevent such attacks. The challenges faced during network configuration, tool installation, and setup improved practical troubleshooting and cybersecurity fundamentals.