Lab Manual for
Python for Cyber Security

Co-funded by
the European Union

Introduction

This lab manual is designed to guide students through practical exercises in cybersecurity course using Python. The activities aim to equip students with hands-on experience in threat detection, risk assessment, incident response, and compliance audits.

1. Lab Guidelines

Lab Environment Setup

- **Software**: Python 3.10 or later, PyCrypto, SIEM tools, Kali Linux, Metasploit.
- **Hardware Requirements**: Minimum 8GB RAM, i5 processor, and 100GB storage.
- **Additional Tools**: LOIC, Burp Suite, BeautifulSoup, Python-Nmap.

Code of Conduct

- Adhere to ethical guidelines.
- Avoid using techniques learned for unauthorized activities.

Submission Policy

- Submit lab reports in PDF format within one week of completion.
- Group activities require a joint submission with detailed roles of each participant.

2. List of Lab Sessions

Lab Activities Overview

1. Risk Assessment
2. Bug-Bee Security Project
3. Phishing Awareness
4. DDoS Attack Simulation
5. Threat Intelligence Gathering
6. MITRE ATT&CK Framework Simulation
7. Threat Hunting Exercise
8. Log Collection, Parsing, and Analysis
9. Data Visualization
10. SIEM Implementation
11. Integration of SIEM, SOAR, and EDR
12. Incident Triage

13. Access Control Configuration
14. Compliance Audit
15. Security Policies and Procedures
16. Privacy Impact Assessment
17. Cyber Law Assessment
18. Lab Project Development/Demonstration

3. Detailed Lab Activities

Lab 1: Risk Assessment

- **Objective**: Identify risks to a fictional organization's assets.
- **Tools**: Python scripts, templates for risk matrix.
- **Steps**:
    1. Identify assets, vulnerabilities, and threats.
    2. Calculate risk levels using a formula or Python script.
    3. Create a mitigation strategy report.
- **Deliverables**: Completed risk matrix and report.

Lab 2: Bug-Bee Security Project

- **Objective**: Perform a web application vulnerability assessment.
- **Tools**: Burp Suite, BeautifulSoup.
- **Steps**:
    1. Analyze provided web applications.
    2. Identify vulnerabilities such as SQL injection or XSS.
    3. Propose mitigation strategies.
- **Deliverables**: Documented vulnerabilities and their mitigation.

Lab 3: Phishing Awareness

- **Objective**: Recognize and mitigate phishing threats.
- **Tools**: Python scripts for email analysis.
- **Steps**:
    1. Simulate phishing emails.
    2. Identify key indicators of phishing attempts.

3. Document strategies for avoiding phishing.

- **Deliverables**: Phishing analysis report.

Lab 4: DDoS Attack Simulation

- **Objective**: Experience and mitigate a simulated DDoS attack.
- **Tools**: LOIC, traffic filtering scripts.
- **Steps**:
    1. Simulate a DDoS attack in a controlled environment.
    2. Configure mitigation tools.
    3. Document findings.
- **Deliverables**: DDoS mitigation strategy report.

Lab 5: Threat Intelligence Gathering

- **Objective**: Gather and analyze threat intelligence.
- **Tools**: Open-source tools like Shodan, Python APIs.
- **Steps**:
    1. Collect data related to cybersecurity threats.
    2. Analyze patterns and trends.
    3. Document insights.
- **Deliverables**: Intelligence analysis report.

Lab 6: MITRE ATT&CK Framework

- **Objective**: Understand attack techniques and propose defenses.
- **Tools**: Python, MITRE ATT&CK database.
- **Steps**:
    1. Map a simulated attack to the framework.
    2. Identify and recommend countermeasures.
- **Deliverables**: Attack analysis and defense proposal.

Lab 7: Threat Hunting Exercise

- **Objective**: Simulate a threat hunt in a lab environment.
- **Tools**: Python scripts for anomaly detection.
- **Steps**:
    1. Set up a controlled environment with known threats.

2. Detect anomalies using Python.
3. Respond to identified threats.

- **Deliverables**: Threat hunting report.

Lab 8: Log Collection, Parsing, and Analysis

- **Objective**: Analyze logs for indicators of compromise (IoCs).
- **Tools**: LogParser, Python libraries.
- **Steps**:
    1. Collect logs from different sources.
    2. Parse logs using regex and Python.
    3. Analyze for anomalies.
- **Deliverables**: Log analysis report.

Lab 9: Data Visualization

- **Objective**: Visualize cybersecurity data.
- **Tools**: Matplotlib, Seaborn, Dash.
- **Steps**:
    1. Collect sample data from lab exercises.
    2. Create security dashboards.
    3. Identify patterns and trends.
- **Deliverables**: Visualized security report.

Lab 10: SIEM Implementation

- **Objective**: Deploy a SIEM solution.
- **Tools**: Siemstress, Python integrations.
- **Steps**:
    1. Configure SIEM for log collection.
    2. Set up dashboards and alerting mechanisms.
- **Deliverables**: Functional SIEM dashboard.

Lab 11: Integration of SIEM, SOAR, and EDR

- **Objective**: Simulate integration for advanced threat detection.
- **Tools**: SIEM, SOAR, and EDR solutions.
- **Steps**:

1. Set up a collaboration between tools.
2. Analyze results for real-world scenarios.

- **Deliverables**: Integration demonstration report.

Lab 12: Incident Triage

- **Objective**: Investigate and triage a security incident.
- **Tools**: Incident response playbook, Python.
- **Steps**:
    1. Follow the playbook to identify the incident.
    2. Categorize and propose actions.
- **Deliverables**: Incident response report.

Lab 13: Access Control Configuration

- **Objective**: Configure and test network access controls.
- **Tools**: Python, ACL scripts.
- **Steps**:
    1. Implement ACLs.
    2. Test access permissions.
- **Deliverables**: Access control configuration report.

Lab 14: Compliance Audit

- **Objective**: Conduct a compliance audit.
- **Tools**: Compliance tools, Python scripts.
- **Steps**:
    1. Assess compliance with policies.
    2. Generate a detailed audit report.
- **Deliverables**: Compliance audit report.

Lab 15: Security Policies and Procedures

- **Objective**: Draft security policies.
- **Tools**: Policy templates.
- **Steps**:
    1. Write Acceptable Use and Incident Response Policies.
    2. Present the policies to the class.

- **Deliverables**: Security policies.

Lab 16: Privacy Impact Assessment

- **Objective**: Assess the privacy impact of a system.
- **Tools**: Privacy assessment templates.
- **Steps**:
    1. Conduct a privacy impact analysis.
    2. Propose mitigation strategies.
- **Deliverables**: Privacy impact report.

Lab 17: Cyber Law Assessment

- **Objective**: Examine the completeness of cyber laws.
- **Tools**: Legal documents, Python tools for report generation.
- **Steps**:
    1. Analyze Nepal's cyber laws.
    2. Identify gaps and propose improvements.
- **Deliverables**: Cyber law assessment presentation.

Lab 18: Project Development/Demonstration

- **Objective**: Develop a comprehensive project using Python for cybersecurity.
- **Tools**: Python libraries, cybersecurity tools.
- **Steps**:
    1. Choose a project topic.
    2. Develop, test, and present the project.
- **Deliverables**: Final project presentation and demonstration.

Assessment Criteria

- Participation and Engagement: 30%
- Lab Reports: 34%
- Final Project: 36%

References

- Course books, documentation, and tools as listed in the syllabus.