Lab Manual for IoT Hacking Simulation

# Lab Manual for IoT Hacking Simulation Without Hardware

This lab manual provides a detailed guide for simulating Internet of Things (IoT) hacking techniques in an ethical and controlled environment without requiring physical IoT devices. It focuses on using virtual tools, emulators, and pre-captured datasets to replicate real-world IoT scenarios.

---

## \*\*Lab Objectives\*\*  
1. Understand IoT architecture and communication protocols.  
2. Simulate common IoT vulnerabilities and their exploitation.  
3. Analyze IoT traffic using virtual tools and datasets.  
4. Learn how to identify and secure vulnerabilities in IoT systems.

---

## \*\*Lab Prerequisites\*\*  
1. A computer with the following installed:  
   - \*\*Wireshark\*\*: For IoT traffic analysis.  
   - \*\*Kali Linux\*\*: Includes IoT penetration testing tools such as `Nmap`, `Shodan`, and `Metasploit`.  
   - \*\*Mosquitto MQTT\*\*: For simulating IoT communication using the MQTT protocol.  
   - \*\*Burp Suite\*\*: For intercepting and analyzing API traffic.  
   - \*\*Python\*\*: For scripting IoT-related tasks.  
  
2. Pre-downloaded datasets or IoT simulations:  
   - Example IoT traffic `.pcap` files.  
   - IoT device emulators like [IoTGoat](<https://github.com/OWASP/IoTGoat)> or virtual smart home setups.

3. Knowledge of IoT protocols:  
   - MQTT, CoAP, HTTP/HTTPS, and Zigbee.

---

## \*\*Lab Exercises\*\*

### \*\*Exercise 1: Simulating IoT Communication Using Mosquitto MQTT\*\*

#### \*\*Objective\*\*:  
Simulate IoT communication using the MQTT protocol and identify vulnerabilities.

#### \*\*Steps\*\*:  
1. \*\*Set Up Mosquitto Broker\*\*:  
   - Install Mosquitto on your machine:  
     ```bash  
     sudo apt install mosquitto mosquitto-clients  
     ```  
   - Start the Mosquitto broker:  
     ```bash  
     mosquitto -v  
     ```

2. \*\*Simulate IoT Devices\*\*:  
   - Open a terminal and publish a message:  
     ```bash  
     mosquitto\_pub -h localhost -t "iot/topic" -m "Temperature: 25°C"  
     ```  
   - In another terminal, subscribe to the topic:  
     ```bash  
     mosquitto\_sub -h localhost -t "iot/topic"  
     ```

3. \*\*Analyze Security Flaws\*\*:  
   - Discuss the lack of encryption in MQTT by default.  
   - Highlight the risks of using default credentials.

#### \*\*Expected Outcome\*\*:  
Students will understand how IoT devices communicate using MQTT and the security risks involved.

---

### \*\*Exercise 2: Analyzing IoT Traffic Using Wireshark\*\*

#### \*\*Objective\*\*:  
Analyze pre-captured IoT traffic to identify security issues.

#### \*\*Steps\*\*:  
1. \*\*Open Wireshark\*\*:  
   - Launch Wireshark and load a pre-captured `.pcap` file containing IoT traffic.

2. \*\*Apply Filters\*\*:  
   - Use filters to isolate IoT protocols:  
     - MQTT: `mqtt`  
     - CoAP: `coap`  
     - HTTP: `http`

3. \*\*Analyze Traffic\*\*:  
   - Identify unencrypted messages.  
   - Look for hardcoded credentials or sensitive data in payloads.

#### \*\*Expected Outcome\*\*:  
Students will learn how to identify insecure communication in IoT traffic.

---

### \*\*Exercise 3: Simulating an IoT Device Vulnerability with IoTGoat\*\*

#### \*\*Objective\*\*:  
Understand common IoT vulnerabilities using the IoTGoat framework.

#### \*\*Steps\*\*:  
1. \*\*Set Up IoTGoat\*\*:  
   - Clone the IoTGoat repository:  
     ```bash  
     git clone <https://github.com/OWASP/IoTGoat.git>  
     cd IoTGoat  
     ```  
   - Start the IoTGoat application:  
     ```bash  
     docker-compose up  
     ```

2. \*\*Explore Vulnerabilities\*\*:  
   - Access IoTGoat in a browser at `http://localhost:8080`.  
   - Test common vulnerabilities:  
     - Insecure API endpoints.  
     - Hardcoded credentials.  
     - Insecure firmware update processes.

3. \*\*Document Findings\*\*:  
   - Note each vulnerability and its potential impact.

#### \*\*Expected Outcome\*\*:  
Students will gain hands-on experience with IoT vulnerabilities in a controlled environment.

---

### \*\*Exercise 4: Simulating an IoT API Attack Using Burp Suite\*\*

#### \*\*Objective\*\*:  
Intercept and analyze API traffic to identify security flaws.

#### \*\*Steps\*\*:  
1. \*\*Set Up Burp Suite\*\*:  
   - Launch Burp Suite and configure the proxy listener.

2. \*\*Simulate API Interaction\*\*:  
   - Use a REST API client (e.g., Postman) to send requests to the IoTGoat API.  
   - Capture and inspect the traffic in Burp Suite.

3. \*\*Analyze and Exploit\*\*:  
   - Identify vulnerabilities like insecure endpoints or weak authentication.  
   - Simulate exploiting an insecure API by modifying requests.

#### \*\*Expected Outcome\*\*:  
Students will understand the risks of insecure IoT APIs and how to secure them.

---

### \*\*Exercise 5: IoT Firmware Analysis\*\*

#### \*\*Objective\*\*:  
Analyze IoT firmware for vulnerabilities.

#### \*\*Steps\*\*:  
1. \*\*Obtain Firmware\*\*:  
   - Use pre-downloaded firmware files from public repositories (e.g., [Firmadyne](<https://github.com/firmadyne/firmadyne)).>

2. \*\*Extract and Analyze\*\*:  
   - Extract the firmware contents:  
     ```bash  
     binwalk -e firmware.bin  
     ```  
   - Look for hardcoded credentials or keys.

3. \*\*Document Findings\*\*:  
   - Note insecure practices such as plaintext credentials or outdated libraries.

#### \*\*Expected Outcome\*\*:  
Students will learn how insecure firmware can compromise IoT devices.

---

## \*\*Lab Summary\*\*  
1. \*\*Skills Acquired\*\*:  
   - Simulating IoT communication and vulnerabilities.  
   - Analyzing IoT traffic for security flaws.  
   - Identifying and mitigating IoT-specific risks.

2. \*\*Key Takeaways\*\*:  
   - IoT systems are prone to various vulnerabilities due to poor security practices.  
   - Ethical hacking helps in identifying and addressing these issues proactively.

---

## \*\*Additional Resources\*\*  
1. \*\*Tools and Frameworks\*\*:  
   - IoTGoat: [[https://github.com/OWASP/IoTGoat](https://github.com/OWASP/IoTGoat).](https://github.com/OWASP/IoTGoat%5d(https:/github.com/OWASP/IoTGoat).)  
   - Firmadyne: [[https://github.com/firmadyne/firmadyne](https://github.com/firmadyne/firmadyne).](https://github.com/firmadyne/firmadyne%5d(https:/github.com/firmadyne/firmadyne).)

2. \*\*IoT Security Guidelines\*\*:  
   - OWASP IoT Project: [[https://owasp.org/www-project-internet-of-things/](https://owasp.org/www-project-internet-of-things/).](https://owasp.org/www-project-internet-of-things/%5d(https:/owasp.org/www-project-internet-of-things/).)

3. \*\*Packet Analysis\*\*:  
   - Wireshark Documentation: [[https://www.wireshark.org/docs/](https://www.wireshark.org/docs/).](https://www.wireshark.org/docs/%5d(https:/www.wireshark.org/docs/).)