API Security Testing Lab — Lab Setup Commands (Safe, Lab-only)

Purpose: Quick, copy-paste commands to prepare an isolated lab for API security testing using Burp Suite, Postman, sqlmap, and a vulnerable web/API target (DVWA). All steps are safe and avoid exploit payloads. Run only in isolated, snapshot-able lab VMs.

1) Ubuntu (Target) — quick setup

Goal: Configure static IP, install Docker, run DVWA (lab-only), create evidence folder and baseline artifacts.

Commands (paste on the Ubuntu target VM):

```
# 1. Set static IP (netplan) - replace interface name if needed
sudo tee /etc/netplan/01-lab.yaml > /dev/null <<'EOF'
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: no
      addresses: [192.168.56.100/24]
      gateway4: 192.168.56.1
      nameservers: { addresses: [8.8.8.8,1.1.1.1] }
EOF
sudo netplan apply

# 2. Install Docker (required to run DVWA container)
sudo apt update && sudo apt -y install ca-certificates curl gnupg lsb-release
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com
sudo apt update && sudo apt -y install docker-ce docker-ce-cli containerd.io docker-compose-plugin
sudo usermod -aG docker $USER  # logout/login to apply

# 3. Pull and run a lab DVWA container (lab-only)
sudo docker pull vulnerables/web-dvwa:latest || true
sudo docker run -d --name dvwa -p 80:80 vulnerables/web-dvwa:latest

# 4. Evidence & baseline
mkdir -p ~/lab-evidence/baseline && chmod 700 ~/lab-evidence
uname -a > ~/lab-evidence/baseline/uname.txt
ss -tulnp > ~/lab-evidence/baseline/services.txt
sha256sum ~/lab-evidence/baseline/* > ~/lab-evidence/baseline/baseline.sha256

# 5. Create a test account (lab-only)
sudo useradd -m -s /bin/bash testuser || true
echo 'testuser:Password123' | sudo chpasswd
# 6. Quick check (on attacker): curl http://<target-ip>/ to verify DVWA homepage
```

2) Kali (Attacker) — quick setup

Goal: Ensure attacker VM can reach the target, install sqlmap, and prepare directories for Burp/Postman usage and evidence collection.

Commands (paste on the Kali attacker VM):

```
# 1. Assign a host-only IP if needed (replace eth1 with host-only interface)
sudo ip addr add 192.168.56.10/24 dev eth1 2>/dev/null || true

# 2. Update & install essentials
sudo apt update && sudo apt -y full-upgrade
sudo apt -y install git python3-pip python3-venv net-tools nmap sqlmap

# 3. Burp Suite (Community) - download & run (requires Java)
mkdir -p ~/tools && cd ~/tools
# Download Burp Suite Community from PortSwigger website via browser and save as burpsuite_community.jar
# Example run (after download): java -jar ~/tools/burpsuite_community.jar &

# 4. Postman - GUI (download from https://www.postman.com/) or use 'newman' CLI
sudo apt -y install snapd || true
sudo snap install postman || true

# 5. Prepare evidence folder & venv
mkdir -p ~/lab-evidence && chmod 700 ~/lab-evidence
python3 -m venv ~/venvs/api && source ~/venvs/api/bin/activate
pip install --upgrade pip
pip install requests
deactivate

# 6. Quick recon (save & hash)
nmap -Pn -sC -sV -oN ~/lab-evidence/nmap_192.168.56.100.txt 192.168.56.100
sha256sum ~/lab-evidence/nmap_192.168.56.100.txt > ~/lab-evidence/nmap.sha256

# 7. Start Burp and configure proxy in Postman to point to 127.0.0.1:8080 for manual testing
```

3) Test Plan & Checklist (ready for Google Docs)

Copy this checklist into your Google Doc test plan. Follow ROE: lab-only, snapshots before tests, and keep evidence hashed.

```
Checklist:
1. Enumerate API endpoints (nmap, curl, swagger/openapi discovery)
2. Test for BOLA (Broken Object Level Authorization) using Burp to manipulate resource IDs and tokens
3. Fuzz GraphQL queries (Postman collections + payloads or use graphql-fuzzers)
4. Use sqlmap only on parameterized inputs that appear to be SQL-backed (lab-only)
5. Capture traffic with tcpdump/pcap and forward copies to your logging VM
6. Record every command in lab_log and hash artifacts

Log template (CSV):
Test ID,Vulnerability,Severity,Target Endpoint,Notes
008,BOLA,Critical,/api/users,Tested ID tampering via Bearer token manipulation
009,GraphQL Injection,High,/graphql,Sent mutated queries and observed error responses
```

4) Safe Manual Testing Notes

Use Burp Suite to manipulate API tokens: intercept requests, modify Authorization headers (Bearer tokens), and observe responses. Do not use stolen credentials. For GraphQL fuzzing, create Postman collections with parametrized queries and run them through Burp or directly from Postman with payloads. Use sqlmap only after pinpointing injectable parameters and with explicit lab ROE permission.

5) 50-word API test summary

In a controlled lab, the API security assessment targeted endpoints for Broken Object Level Authorization and GraphQL injection. Using Burp and Postman for token manipulation and query fuzzing, and sqlmap for verifying SQL-injection-prone inputs, the tests validated authorization weaknesses and input-handling gaps. Artifacts and pcaps were hashed and archived.

Generated: 2025-09-21 18:54:40Z — Lab-only commands, no exploit payloads included.