Project ALCHEMY: Cyber Range Setup Guide

Prerequisites

Before beginning the setup process, ensure your system has the following software installed and accessible:

Software	Source	Purpose
VirtualBox	https://www.virtualbox.org/wiki/Downloads	Virtual machine hypervisor
Ubuntu Server 24.04 LTS ISO	https://releases.ubuntu.com/noble/	Base operating system for VMs
Suricata	https://suricata.io/download/	Network intrusion detection system
jq	https://stedolan.github.io/jq/	Command-line JSON processor

Docker will be installed automatically during the VM setup process via terminal script.

Part 1: Network Configuration

Step 1.1 Create a Host-Only Network

The host-only network provides an isolated Layer 2 network for inter-VM communication, completely segregated from your host machine's actual network connectivity.

1. Open **VirtualBox**
2. Navigate to **Tools** → **Network** → **Host-Only Networks**
3. If no host-only network exists, create one with the following parameters:
    - **IPv4 Address:** 192.168.56.1
    - **Netmask:** 255.255.255.0
    - **DHCP Server:** Enabled

This configuration creates a private network segment shared by all virtual machines while maintaining isolation from external networks.

Part 2: Services Virtual Machine (Alchemy-services)

Step 2.1 Create and Configure the VM

1. Create a new virtual machine in VirtualBox with the following specifications:
   System Settings:
   - Chipset: ICH9 (default)
   - EFI: Enabled (optional)
   - CPU: Minimum 2 cores (8 cores or more recommended)
   - RAM: Minimum 4 GB (8 GB recommended)

   Storage Settings:
   - Attach the Ubuntu 24.04 LTS ISO to the virtual optical drive

   Network Settings:
   - Adapter 1: NAT
   - Adapter 2: Host-Only Adapter (select the network created in Step 1.1)
   - Promiscuous Mode: Deny
   - MAC Address: Click "Refresh" to generate a unique MAC address
     - *Example: `08002762FE71`*
     - **Important:** Unique MAC addresses are required for inter-VM communication.

Step 2.2 Install Ubuntu and Base Software

1. Boot the virtual machine and complete the Ubuntu 24.04 installation process
2. After the first boot, log in and execute the following commands:

Update base operating system

sudo apt update && sudo apt -y upgrade

Install Docker

curl -fsSL https://get.docker.com | sh

Add current user to Docker group

sudo usermod -aG docker $USER

Apply group membership

newgrp docker

Step 2.3 Deploy OWASP Juice Shop

The OWASP Juice Shop serves as the vulnerable application running on the services VM. It will be reachable on all network interfaces.

Remove any existing Juice Shop container

docker rm -f juice 2>/dev/null

Deploy Juice Shop with persistent restart policy

docker run -d --name juice --restart unless-stopped -p 0.0.0.0:3000:3000 bkimminich/juice-shop

Step 2.4 Configure Host-Only Network Interface (Persistent)

Configure the secondary network interface (host-only) to obtain an IP address via DHCP on every boot.

Create netplan configuration file

sudo tee /etc/netplan/02-hostonly.yaml >/dev/null <<'EOF'
network:
version: 2
ethernets:
enp0s8:
dhcp4: true
EOF

Note: If your interface is named eth0 instead of enp0s8, substitute accordingly

Set secure permissions on configuration file

sudo chmod 600 /etc/netplan/02-hostonly.yaml

Apply network configuration

sudo netplan apply

Step 2.5 Record the Services VM IP Address

Retrieve and document the IP address assigned to the host-only network interface. This address will be referenced as SVC_IP in later steps.

Method 1: Extract host-only IP directly

ip -4 addr show enp0s8 | awk '/inet /{print $2}' | cut -d/ -f1

Method 2: Display all IPs (last address is typically the host-only IP)

hostname -I

Expected Output Example: 192.168.56.101

Step 2.6 Verify Services VM Functionality

Execute the following commands to confirm proper configuration:

Test Juice Shop on localhost

curl -I http://localhost:3000

Test Juice Shop on host-only interface

curl -I http://$(ip -4 addr show enp0s8 | awk '/inet /{print $2}' | cut -d/ -f1):3000

Verify port 3000 is listening

ss -ltnp | grep 3000

Check firewall status

sudo ufw status

All commands should return successful responses (HTTP 200 or similar).

Part 3: Sensor Virtual Machine (Alchemy-sensor)

Step 3.1 Clone the Services VM

1. In VirtualBox, right-click the **Alchemy-services** VM
2. Select **Clone**
3. Choose **Full Clone**
4. Name the new VM **Alchemy-sensor**

Step 3.2 Configure Sensor VM Network Settings

While the Alchemy-sensor VM is powered down, adjust its network configuration:

1. Open **Settings** for the Alchemy-sensor VM
2. Navigate to **Network**
3. Configure as follows:
    - Adapter 1: NAT
    - Adapter 2: Host-Only Adapter (same network as services VM)
    - Promiscuous Mode: **Allow All** (required for packet capture)
    - MAC Address: Click "Refresh" to generate a unique MAC address
        - *Example:* `080027E71F2B`

Step 3.3 Assign a Unique Host-Only IP

Boot the sensor VM and assign it a different host-only IP address from the services VM.

Install DHCP client

sudo apt -y install isc-dhcp-client

Enable host-only network interface

sudo ip link set enp0s8 up

Request IP address via DHCP

sudo dhclient -v enp0s8

Display assigned IP address

ip -4 addr show enp0s8

If DHCP assigns the same IP as the services VM, assign a static IP instead:

Edit `/etc/netplan/02-hostonly.yaml` to use a static address (e.g., `192.168.56.102`), then run `sudo netplan apply`.

Step 3.4 Verify Inter-VM Connectivity

Test communication between the sensor and services VMs:

Ping the services VM (replace with actual SVC_IP if different)

ping -c 4 192.168.56.101

Test HTTP connectivity to Juice Shop

curl -I http://192.168.56.101:3000

Both commands should succeed without errors.

Step 3.5 Install and Configure Suricata

Suricata is the network intrusion detection system (IDS) that monitors traffic on the sensor VM.

Update package lists and install Suricata and jq

sudo apt update && sudo apt -y install suricata jq

Start Suricata as a background service, monitoring the host-only interface

sudo suricata -i enp0s8 -D

Part 4: Verification and Demonstration

Step 4.1 Live Traffic Capture Demonstration

To verify that the sensor can observe traffic from the services VM, follow these steps:

1. **On the Sensor VM, open two terminal windows** (or use terminal multiplexing)
2. **Terminal A (Monitoring):** Start a live stream of HTTP and flow events captured by Suricata
   sudo tail -f /var/log/suricata/eve.json | jq -r 'select(.event_type=="http" or .event_type=="flow")'
3. **Terminal B (Traffic Generation):** Generate HTTP requests to the Juice Shop on the services VM
   curl -I http://192.168.56.101:3000
   curl -I http://192.168.56.101:3000

4. **Expected Result:** Terminal A should display HTTP and flow events corresponding to the requests made in Terminal B. This confirms that the sensor has network visibility into traffic between the two VMs.

Part 5: Attacker Virtual Machine (Alchemy-attacker)

Step 5.1 Create and Configure the Attacker VM

Create a new virtual machine for MITM attack execution with the following specifications:

Motherboard Settings:

- Base Memory: Minimum 3 GB (3072 MB)
- Boot Order: Hard Disk (first in boot sequence)
- Chipset: ICH9
- Pointing Device: USB Tablet
- I/O APIC: Enabled
- EFI: Disabled
- Hardware Clock in UTC: Enabled

Processor Settings:

- CPU Cores: 2 (minimum; increase if host system allows)
- Execution Cap: 100%
- PAE/NX: Enabled
- Nested VT-x/AMD-V: Enabled

Acceleration Settings:

- VT-x/AMD-V: Enabled
- Nested Paging: Enabled

Display Settings:

- Video Memory: Maximum available

Network Settings:

- Adapter 1 (NAT):
  - Attached to: NAT
  - Adapter Type: Intel PRO/1000 MT Desktop
  - Promiscuous Mode: Allow All
  - Cable Connected: Yes
- Adapter 2 (Host-Only):
  - Attached to: Host-Only Adapter
  - Adapter Type: Intel PRO/1000 MT Desktop
  - Promiscuous Mode: Allow All
  - Cable Connected: Yes

Storage Settings:

- Attach the Ubuntu 24.04 LTS ISO to the virtual optical drive

Step 5.2 Install Ubuntu and Prepare the Environment

1. Boot the VM and complete Ubuntu 24.04 installation
2. After first login, execute the following commands:

Update package repositories and system packages

sudo apt update && sudo apt upgrade -y

Install MITM and packet analysis tools

sudo apt install -y dsniff mitmproxy tcpdump iptables-persistent

Install iptables-legacy for packet redirection

sudo apt install iptables-legacy -y

Step 5.3 Create Logging Directory

Create a dedicated directory for MITM proxy logs and artifacts:

Create the MITM logging directory (replace [user] with your username)

mkdir -p /home/[user]/mitm/logs

Step 5.4 Enable IP Forwarding

Enable IP forwarding to allow the attacker VM to route traffic between network segments:

Enable IP forwarding (temporary)

echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward

To make this permanent, edit /etc/sysctl.conf and set:

net.ipv4.ip_forward=1

Then apply: sudo sysctl -p

Troubleshooting

Issue	Solution
VMs cannot communicate across host-only network	Verify both VMs have unique MAC addresses; check Adapter 2 is set to Host-Only Network on both VMs
Juice Shop returns "Connection Refused"	Ensure Docker container is running: docker ps should list the juice container; verify port 3000 is listening: ss -ltnp | grep 3000
enp0s8 interface not receiving IP address	Confirm DHCP is enabled on host-only network in VirtualBox; run sudo dhclient -v enp0s8 manually
Suricata not logging events	Verify interface name is correct (enp0s8 or eth0); check Suricata is running: ps aux | grep suricata; ensure Promiscuous Mode is set to "Allow All" on sensor VM
Permission denied errors with Docker	Confirm user was added to Docker group: groups $USER should include docker; log out and back in or run newgrp docker
Attacker VM cannot reach services VM	Ensure both Adapter 1 and 2 are properly configured; confirm host-only network connectivity with ping 192.168.56.101 from attacker VM
IP forwarding not persisting after reboot	Edit /etc/sysctl.conf, uncomment and set net.ipv4.ip_forward=1, then run sudo sysctl -p
mitmproxy or dsniff installation fails	Verify Ubuntu is fully updated: sudo apt update && sudo apt upgrade -y; check internet connectivity on attacker VM

Summary

Once all steps are completed, you will have a fully functional cyber range consisting of:

- **Alchemy-services:** A vulnerable application host running OWASP Juice Shop
- **Alchemy-sensor:** A network monitoring host running Suricata IDS

- **Isolated network segment:** Provides controlled communication between VMs without affecting your host machine's network

This infrastructure is now ready for controlled penetration testing, network analysis, and security research activities.