MICS Open Network Security

Part X.2: Practical Intro to Ethical Hacking

Michal Sterzel, Alberto Finardi, Tom Gave, Jan Marxen
Secan-Lab.uni.lu

Winter 2025/2026

Contents

1 Introduction

This short guide documents the "Open Network Security" lab that is included in this repository. It explains how to install the required software on the host, how to start the lab VMs with Vagrant, and how to verify that the DNS/DHCP services are running correctly.

The lab provided here uses two VM roles: a DNS-DHCP server and an Agent client. The participant performs detective tasks on the DNS-DHCP VM by observing logs and DNS traffic.

Updated: October 31, 2025

Contributors: Michal Sterzel, Alberto Finardi, Tom Gave, Jan Marxen

2 Important disclaimer

An Ethical Hacking lab must be isolated and used responsibly. Before running any experiment, ensure you follow these rules:

- Do all exercises only inside an isolated lab environment.
- Do not connect lab network interfaces to production networks.
- Do not run offensive or scanning tools against external hosts.

Failure to follow these rules can have severe legal consequences.

3 Prerequisites

The lab requires the following software on the host machine:

- VirtualBox – the lab uses VirtualBox as provider.
- Vagrant – used to define and run the VMs.
- Ansible – required for provisioning with provided host-based playbooks.

The repository contains installer helpers that attempt to install missing prerequisites automatically:

- Linux/macOS: `./install-prerequisites.sh`
- Windows: `install-prerequisites.bat`

Run the appropriate script for your platform. The Windows script will advise on Ansible installation options (WSL / Python) rather than forcing one method.

4 Lab setup

Starting the lab is intentionally simple. From the repository root run the launcher, which brings up and provisions the VMs in the correct order.

- Linux/macOS: `./launch-lab.sh`
- Windows: `launch-lab.bat`

The launcher starts the DNS-DHCP VM first, then the Agent VM. Ansible playbooks in the repository provision the services: dnsmasq for the server, and client configuration with necessary covert communication abilities. The launcher includes a "clean" option to destroy and recreate VMs (`./launch-lab.sh clean`) and a "noclean" option to keep existing VMs.

After the launcher completes, verify the lab using the helper:

```
chmod +x test-lab.sh
./test-lab.sh
```

The helper prints the DHCP-assigned address for the Agent and performs basic connectivity checks to the DNS server. You can also test manually from the Agent VM after `vagrant ssh`:

```
ip addr show
ip route show
ping -c 4 192.168.10.1    # ping DNS-DHCP server
nslookup example.com 192.168.10.1
```

To shutdown the lab, run the following commands from the repository root:

```
chmod +x shutdown-lab.sh
./shutdown-lab.sh
```

or (windows):

```
./shutdown-lab.bat
```

In summary, the workflow is:

1. Install prerequisites `./install-prerequisites.sh` (if needed).
2. Launch the lab: `./launch-lab.sh`
3. Verify the lab: `./test-lab.sh`
4. Perform exercises inside the lab.
5. Shutdown the lab: `./shutdown-lab.sh`

or (windows):

1. Install prerequisites `install-prerequisites.bat` (if needed).
2. Launch the lab: `launch-lab.bat`
3. Verify the lab: `test-lab.bat`
4. Perform exercises inside the lab.
5. Shutdown the lab: `shutdown-lab.bat`

5 VM roles and lab model

The lab uses two VM roles:

- **DNS-DHCP VM** – runs `dnsmasq` on the static address `192.168.10.1`. It serves DHCP leases (default range `192.168.10.3-192.168.10.100`) and logs incoming DNS queries.
- **Agent VM** – a DHCP client that simulates covert communications by sending encoded DNS queries to the DNS server.

In the exercises the participant acts as the detective by inspecting DNS requests and logs on the DNS-DHCP VM to identify and decode covert messages sent by the Agent.

6 Notes on network isolation and safety

All lab VMs attach to a VirtualBox internal network called `labnet`. This keeps lab traffic inside your host machine. Do not add bridged adapters to lab interfaces; keep them internal or host-only.

If you want to ensure guests have no Internet access, add firewall rules inside the guests. The Ansible playbooks include commented examples for adding restrictive iptables/nftables rules.

7 Troubleshooting

Common checks and commands:

- On the host: run `vagrant status` in each VM folder to verify VM states.
- On the DNS-DHCP VM (via `vagrant ssh`): `sudo systemctl status dnsmasq` and `sudo journalctl -u dnsmasq -n 200`.
- On the Agent VM: check IP and resolution with `ip addr`, `ip route`, `cat /etc/resolv.conf`, and `ping 192.168.10.1`.

Security reminder: perform all exercises only inside this isolated lab and do not direct lab traffic to other networks.

8 Further reading

- Project README and playbooks in this repository (start here).
- dnsmasq man pages and documentation: `http://www.thekelleys.org.uk/dnsmasq/doc.html`
- Ansible documentation: `https://docs.ansible.com/`