Hosting VulnHub Labs on Proxmox — Professional Report

Author: N3R0
Date: 3 November 2025

Executive Summary

This document provides a complete, step-by-step professional guide to hosting VulnHub (vulnerable) labs on a Proxmox server. It covers prerequisites, downloading and extracting OVA/VMDK files, converting disks to QCOW2, creating, and configuring VMs in Proxmox, networking options (NAT, bridged, host-only), recommended tools, security considerations, troubleshooting, and best practices.

Table of Contents

1. Introduction & Objectives
2. System Requirements & Hardware Sizing
3. Proxmox Preparation and Installation (brief)
4. Downloading VulnHub Labs (OVA/VMDK)
5. Extracting OVA and Inspecting VMDK
6. Converting VMDK → QCOW2 (qemu-img)
7. Uploading Disks to Proxmox Storage
8. Creating and Configuring VMs in Proxmox GUI and CLI
9. Accessing & Managing Labs (console, SSH, snapshots)
10. Tools Required (host & guest)
11. Troubleshooting
12. Appendix: Useful Commands & Examples

---

1. Introduction & Objectives

Objective: Host and run vulnerable virtual machines downloaded from VulnHub on a Proxmox VE server for learning, training, and lab exercises. This guide assumes the user has administrative access to a Proxmox server and basic Linux command-line experience.

---

2. System Requirements & Hardware Sizing

- Minimum recommended server for multiple small VulnHub VMs:
    1. CPU: 4 vCPUs (modern CPU with virtualization support — Intel VT-x or AMD-V).
    2. RAM: 16 GB (more if running many VMs; 8 GB minimum for single small VM).
    3. Disk: 250 GB SSD or larger (fast I/O improves lab responsiveness). Use separate storage for backups and ISOs if possible.
    4. Network: 1 Gbps NIC (or better).

Considerations: allocate resources per VM based on the lab's recommended specifications. Use thin provisioning for faster provisioning and space savings.

---

3. Proxmox Preparation and Installation (brief)

If Proxmox VE is not installed, follow Proxmox official documentation to install Proxmox VE 7.x or 8.x on the server. Key steps:

1. **Download Proxmox VE ISO from the official site and create a bootable USB.**
2. **Install Proxmox, configure network and storage during installation.**
3. **Update the system after installation:**
   ssh into the Proxmox host.
   run: apt update && apt full-upgrade -y
4. **Access the Proxmox web UI at https://<proxmox-ip>:8006 and log in as root.**

---

4. Downloading VulnHub Labs (OVA/VMDK)

Option 1

1. Browse VulnHub (https://www.vulnhub.com) and download the VM (OVA/ZIP/VMDK) you want to host. Ensure you have permission to run the lab and you understand any licensing terms.
2. Save the downloaded file to a workstation or directly to your Proxmox host (e.g., /root/downloads).

Option2

1. Download and Extract the vm in the windows then transfer the folder in the server using the Winscp

---

5. Extracting OVA and Inspecting VMDK

OVA files are tar archives containing an .ovf and disk files (commonly .vmdk). To extract and inspect:

1. On Linux (workstation or Proxmox host):
   mkdir /root/vmhack && cd /root/vmhack
   tar -xvf path/to/vm.ova
   ls -l # note the .vmdk and .ovf filenames

Open the .ovf file in a text editor to see recommended CPU, RAM, and network settings.

---

6. Converting VMDK → QCOW2 (qemu-img)

Proxmox prefers QCOW2 for features like snapshots. Use qemu-img to convert VMDK to QCOW2.

Commands (run on the Proxmox host or a Linux workstation with qemu-img installed):

1. Install qemu-utils if needed: apt update && apt install qemu-utils -y
2. Convert: qemu-img convert -O qcow2 source-disk.vmdk target-disk.qcow2

Options and tips:

Use -p to show progress: qemu-img convert -p -O qcow2 source.vmdk target.qcow2

---

7. Uploading Disks to Proxmox Storage

Two common methods: upload via Proxmox GUI (Datacenter → Storage) or use SCP/rsync to transfer and then use qm importdisk.

Option A

On GUI:

In Proxmox web UI, select the VM (or create a placeholder VM), go to Hardware → Add → Hard Disk → select 'Use existing disk image' or upload via ISO/Backup storage browser.

On CLI: Attach the disk to VM as scsi/virtio: qm set 100 --scsihw virtio-scsi-pci --scsi0 local-lvm:vm-100-disk-0

Set boot drive if needed: qm set 100 --boot c --bootdisk scsi0

Option 2

If you have transferred the file directly into the Proxmox using WinSCP then use the command

qm importovf <vm-id> <yourfile.ovf> disk-name

---

8. **Creating and Configuring VMs in Proxmox GUI and CLI**

Create VM with settings matching the lab's .ovf recommendations.

GUI steps:

1. Datacenter → Create VM → set VMID, name, OS type (Linux/other), and uncheck 'Use CD/DVD' if attaching existing disk later.
2. Set CPU cores, memory, and system → Set disk bus type to 'virtio' or 'scsi' if you attached via scsi0.
3. Network: add a network device (virtio) and choose bridge (vmbr0) or isolated vmbr for host-only labs.

CLI example (quick):

1. qm create <VM ID> --name vuln-lab --cores 2 --memory 2048 --net0 virtio,bridge=vmbr0
2. Attach converted disk (from earlier): qm set 101 --scsi0 local-lvm:vm-<VM ID>-disk-0 --scsihw virtio-scsi-pci

Start the VM: qm start 101

Access console via Proxmox GUI (noVNC) or via 'qm terminal 101' on the host.

---

9. **Accessing & Managing Labs (console, SSH, snapshots)**

Access methods: Proxmox noVNC console, SPICE (if configured), SSH (if VM has SSH enabled), web services hosted by the VM.

Snapshots & Backups:

1. 1.Use snapshots for quick rollback when users break the lab: qm snapshot <vmid> snap1
2. Use vzdump backups for full backups: vzdump <vmid> --storage local --compress lzo --mode suspend
3. Schedule backups with Proxmox Backup Server (recommended for production lab infra).

---

10. Tools Required (host & guest)

Host-side tools (Proxmox host / workstation):

1. Proxmox VE (web UI)
2. qemu-utils (qemu-img) for disk conversions
3. wget/curl, tar, unzip (for extracting OVA)
4. ssh/scp/rsync (for file transfers)
5. vim/nano for ovf inspection

Guest-side tools (useful inside lab VMs):

1. Nmap, Netcat (nc), curl, wget
2. Burp Suite / OWASP ZAP (web labs)
3. Metasploit Framework (where allowed)
4. Wireshark / tcpdump
5. Common pentest tooling: hashcat, john, gobuster, dirb, nikto, sqlmap

Optional infrastructure: Proxmox Backup Server, Ansible for provisioning, VLAN-capable switch for network segmentation.

---

11. Troubleshooting

Common issues and fixes:

1. VM won't boot after importing disk: verify disk format and controller type. Try different bus (ideally virtio-scsi).
2. Network unreachable: check bridge assignment, firewall, and whether DHCP is present on that network. Use ip a and ip route within VM.
3. Disk conversion errors: ensure VMDK descriptor references correct extents; if split, assemble using VMware tools or specify the correct vmdk file.
4. Console blank or garbled: try SPICE or noVNC; ensure the correct display/balloon device. Install QEMU guest agents for better integration.

---

12. Appendix: Useful Commands & Examples

Install qemu-utils:

```
apt update && apt install qemu-utils -y
```

Extract OVA:

```
mkdir /root/vmtemp && tar -xvf /path/to/sample.ova -C /root/vmtemp
```

Convert VMDK to QCOW2:

```
qemu-img convert -p -O qcow2 source.vmdk target.qcow2
```

Create VM skeleton:

```
qm create 100 --name vuln1 --cores 2 --memory 2048 --net0 virtio,bridge=vmbr0
```

Import disk to local-lvm:

```
qm importdisk 100 /root/target.qcow2 local-lvm
```

Attach disk:

```
qm set 100 --scsihw virtio-scsi-pci --scsi0 local-lvm:vm-100-disk-0
```

Start VM:

```
qm start 100
```

Snapshot:

```
qm snapshot 100 before-exploit
```

Backup:

```
vzdump 100 --storage local --compress lzo --mode suspend
```
