Phase 1: Lab Design & Setup

Step 1: Lab Goals

1. **Scope of Practice**
   - **Initial Access** (phishing, CVE exploits)
   - **Lateral Movement** (Pass-the-Hash, remote execution)
   - **Privilege Escalation** (local exploits, misconfigurations)
   - **Persistence & Exfiltration**
   - **Reporting**

Step 2: Choose & Prepare Your Hypervisor

You have two main options:

Hypervisor	**Pros**	**Cons**
VMware Workstation	Mature, snapshots, good guest tools	Paid license (Pro)
VirtualBox	Free, cross-platform	Slightly less stable network I/O
Proxmox VE / ESXi	Scalable, clustering, web UI	More complex initial setup

1. **Decide which** you'll use locally. But for this lab we will be using VirtualBox.
2. **Install** it on your host machine:
   - Download from the vendor's site.
   - Follow the installer wizard (default options are fine).
3. **Enable Nested Virtualization** (if you want to run Hyper-V or Docker inside VMs).

Step 3: Design Your Initial Topology

We'll start small, then expand. Create three virtual networks:

1. **Lab-Internal** (isolated, no internet)
2. **Lab-DMZ** (for web servers, simulated internet)
3. **Host-Only** (for management/access from your host)

Core VMs

VM Name	Role	OS	Network
Attacker	Kali Linux (or Parrot)	Linux	Lab-DMZ
DC1	Domain Controller	Windows Server 2019	Lab-Internal
Win10	User Workstation	Windows 10 pro	Lab-Internal
WebApp	Vulnerable web application	Ubuntu + DVWA/ Juice Shop	Lab-DMZ Lab-Internal
FileSrv	Shared storage (SMB/NFS)	Windows/Linux	Lab-Internal

1. **Create Virtual Networks:**
   - In your hypervisor's network editor, define "Lab-Internal" & "Lab-DMZ" as NAT or VLAN segments; "Host-Only" as host-only.
2. **Deploy VMs:**
   - Allocate modest resources: 2 vCPU / 4 GB RAM for each Windows VM; 1 vCPU / 2 GB RAM for Linux VMs initially.
   - Mount ISO images and complete OS installations.
3. **Snapshot Baseline:**
   - Once each VM is up, take a fresh snapshot—this allows you to revert after a destructive test.