Foundation

The first go-around taught me that there is much more involved in establishing a homelab than I initially realized. While the essential requirement is a device capable of running a server, other critical aspects include networking, security, and constructing a supportive infrastructure. By taking the CompTIA Security+ exam and enrolling in several cybersecurity courses during graduate school, I came to understand that a core part of security lies in the correct setup and backbone structure. To develop a strong and secure homelab, I had to start from the foundation. Much like the OSI model, I needed to begin at layer one and build upward.

Choice of Hardware

Choosing the right device can be challenging, but I was pleasantly surprised to discover that even an old, simple PC can be transformed into a fully functioning server. While I couldn't control physical malfunctions, I concentrated on what I could manage. One of my first steps was ensuring there was a backup server running, ready to replace my primary server if needed. It's quite astonishing how effective some used mini PCs can be as powerful servers. They offer low power consumption, compatibility, and ease of upgrades. Buying two mini PCs on eBay with robust specifications and a healthy amount of memory was perfect for starting out. These two PCs are still operational years later. Initially, I didn't focus much on specifications, but it quickly became apparent that memory is consumed far more rapidly than storage.

The best place to start is with your old laptop or PC. It's quite interesting how easily an old laptop can be repurposed. I had a Mac Mini, which Apple discontinued a few years back, making this device no longer usable, at least not with macOS. I attempted to use FlashOS to install the latest macOS on a device that was not supported, and while that worked, there were many ups and downs along the way. However, installing any Linux OS and repurposing an outdated device that had reached end of life, surprisingly, was easier than I thought. Thirteen years later, that Mac Mini runs smoothly and without issues. Additionally, I removed its Wi-Fi module, which provided me with extra space to install a second storage disk inside.

I found a lot of posts online, and many people suggested running a Raspberry Pi. While these devices are efficient, repurposing the device I already had was a much cheaper option and also allowed me to expand my hardware components. This is another significant reason to consider pre-

2020 mini PCs. Newer devices are great, as they are more compact and power-efficient; however, with each new model, more components are being soldered to the motherboard and cannot be replaced. This is especially true for new DDR5 RAM sticks. From my experience, RAM memory is the first thing I run out of and need to upgrade. While acquiring another device is an option, upgrading my current PC from 8GB to 16GB or even 32GB is a much easier and more efficient process.

Networking and Logical Security

The second part of building a solid foundation focused on the next two layers of OSM model: networking. The best way to overcome your fears is by confronting them. I began by educating myself through reading and watching videos, and then I purchased network equipment that allowed for full configuration capabilities. Through my research, I discovered that Ubiquiti equipment was the best choice for my journey. They offer enterprise-level networking hardware and software tailored for home use, all at reasonable prices and with a user-friendly interface. Unlike Cisco, the industry leader, using Ubiquiti didn’t make me feel like I needed years of experience to operate it. A friend also helped with my decision by offering me his UniFi router, requiring only a switch and an access point for complete functionality. Utilizing the Ubiquiti system enabled me to secure every aspect of my network. From configuring VLANs, setting up layer 2 and layer 3 security features, to controlling who connects to my network, I was provided with an enormous amount of data and insight.

While the DNS component wasn’t as comprehensive as Pi-hole, it proved stable and effectively limited application usage. This setup also presented an opportunity for me to learn more about firewalls—specifically how to establish rules, determine which ones to implement, and maintain necessary traffic flow without compromising security beyond standard practices. Early in my network setup, Ubiquiti upgraded their firewall to a zone-based firewall, which opened up a whole new Pandora’s box of options.

The final aspect of building a strong foundation for my homelab involved a proper application setup. A well-configured Proxmox environment, complete with a backup server, seamless migration capabilities, and a firewall installed on Proxmox, was essential. I ensured that my hypervisor management UI was physically and logically separated from the rest of the virtual machines (VMs) on the network. This configuration allowed me to restrict access to the Proxmox management UI to only a few devices on one VLAN, while the VMs running on the server were assigned to their dedicated VLANs. This segregation was particularly vital given my access to the fully configurable zone-based firewall, enabling me to protect specific VMs and provide them with the appropriate level of security while ensuring adequate accessibility for externally-accessed applications, all without compromising the functionality of other apps.

Before I began setting up my servers and network, I was quite a stranger to networking and timid about diving in. However, after completing my setup, I realized how enjoyable networking can be and how eager I was to pursue a career as a network engineer. My journey not only revealed how much I didn’t know but also ignited my passion for network configuration. I learned that a significant portion of security comes from proper network architecture and foundational structure.

Hypervisor Setup

I already had experience with Proxmox software, and their community is fantastic, always quick to answer questions on forums. Additionally, many YouTubers I follow use Proxmox and have produced great setup videos. One channel, in particular, Learn Linux TV, was my main guide in setting up my Proxmox server and ensuring its proper security. The installation of the software is pretty straightforward; once you create a bootable flash drive, you're ready to start. I made sure that this device was on a static IP and outside the DHCP range, which facilitated an easy installation and avoided any future reconfiguration if the IP changed. While this can be configured from the Proxmox installation side, I also set it up in Ubiquiti to ensure that my switch wouldn’t accidentally assign another IP.

Once I logged into Proxmox, the first step I took was to create two additional admin users: one PAM user and one PVE Realm user. The generic account is root, which resides in the PAM realm. I created a secondary PAM user through the command line, granting it admin access in the Proxmox application as well as sudo access through the terminal. After ensuring the setup was correct and that the user worked as intended, I disabled my root account and configured SSH access to only use public key authentication (PK). While the PAM account is sufficient for most operations, I primarily use it for SSH and terminal access. I created a PVE Realm user with admin access for web UI activities. The only limitation of this user is the inability to perform updates, but I can easily SSH into the device to carry those out.

Once my users were configured, I set a few firewall rules in Proxmox at the database level. My machine was already secured with network firewall configurations, but as a good practice, I added additional firewall rules to enhance security. I ensured that only one device could access Proxmox via IP and SSH. Additionally, I made sure that any other service or access point went through a domain with a valid SSL certificate, keeping all communication secure and encrypted. Is this level of security strictly necessary? Not really, as Proxmox uses its own self-signed certificate and protects against threats using HTTPS. However, adding a Let’s Encrypt certificate with my personal domain provides additional security and eliminates that "Potential Risk" message that appears in browsers. All this was achieved without any port forwarding, which I’ll discuss in a later chapter.

The last thing I did was replicate the same configuration steps for the Proxmox Backup Server. When I configured it, it was still in its early stages but was already in a stable version and performed well. Much like the Proxmox Virtual Environment, the Backup Server installation was straightforward. After completing all the same security steps I implemented on the PVE, I connected my server to the Backup Server and scheduled daily backups of every machine created across all clusters. Although I only had one machine initially, this setup allowed me to set it and forget it, making it work automatically—even for future use when I acquire another device and create a cluster. While backups are configured on the Virtual Environment side, prune jobs, verification, and other backup checks and optimizations were set up in the Backup Server.

Once everything was set up and running, it was time to create my first virtual machine and start my homelab applications.

Revision #1
Created 2026-03-04 18:27:32 UTC by lumxux
Updated 2026-03-04 18:29:11 UTC by lumxux