REPORT
Webmin Secure Administration (Dual-Homed)

v1.1.0

Author:

Eldon Gabriel

March 31, 2026

Table of Contents

Disclaimer: This report documents the authors' independent technical work in completing a Mossé Cyber Security Institute (MCSI) laboratory exercise. This reflects the authors'practical implementation of a dual-homed network architecture, static IP configuration, and Webmin service hardening on Ubuntu 24.04 in a controlled environment. No proprietary MCSI videos, lab manuals, or internal instructions were distributed. This documentation is presented as a "Proof of Work" portfolio piece in full compliance with the MCSI's disclosure and academic integrity policies.

Revision History

Version	Date	Author	Description of Changes
1.0.0	01/25/2026	Eldon G.	Initial draft.
1.1.0	03/31/2026	Eldon G.	A dual-homed network architecture was implemented. The report was updated with empirical evidence (screenshots 1-4), including Layer 3/4 verification, host-to-guest connectivity, and PAM-based security hardening.

0.0 Executive Summary

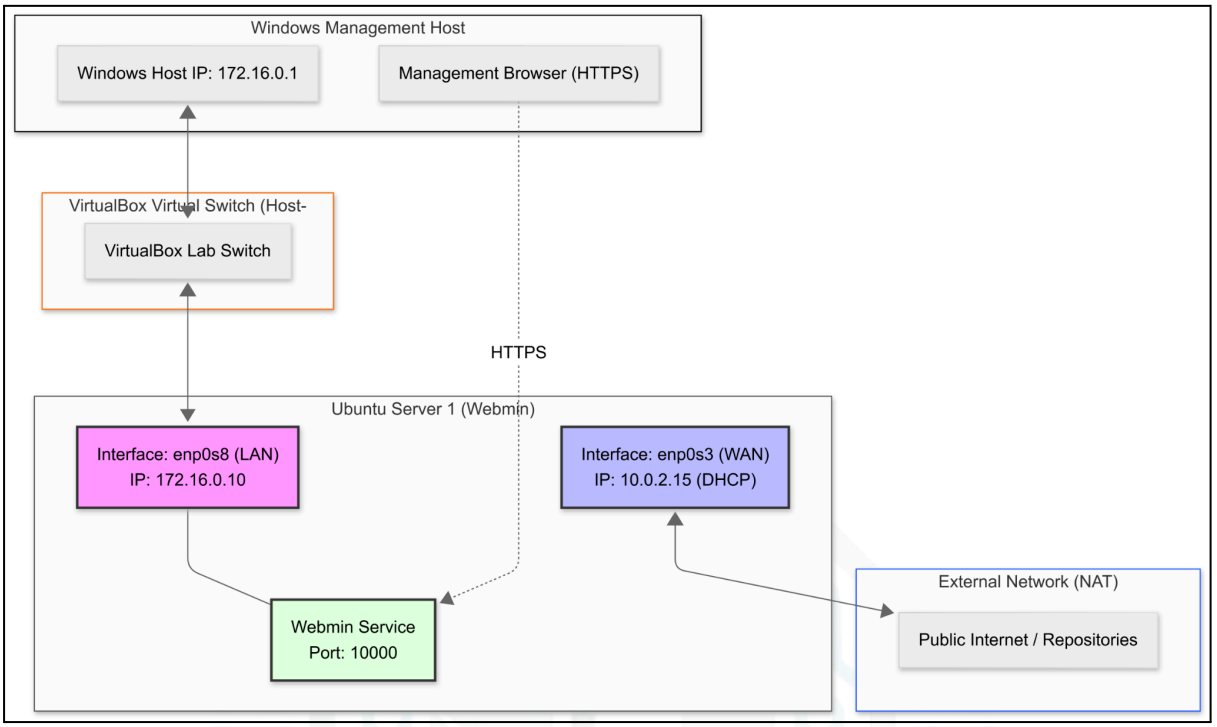

Figure 1: Webmin network architecture. March 31, 2026. Eldon G.

0.1 Project Overview

The objective was to deploy a secure Webmin management instance on an **Ubuntu Server 24.04**. To maintain network security, the server was configured with dual-homed adapters to isolate the management traffic from the external network.

Webmin was selected for this specific deployment to provide deep, module-level control over system configuration (such as Cron jobs and File Management) that extends beyond the standard monitoring capabilities of Cockpit, thereby allowing for more granular administrative control within the lab environment.

This guide is intended for hands-on system administration and security labs, with a focus on validating Webmin functionality with a dual-homed configuration that separates Management (LAN) traffic from update (WAN) traffic, managing users and services, configuring access controls, and recovering from common lockout or misconfiguration scenarios.

Target OS

Ubuntu 24.04 (Debian-based systems noted where applicable)

Scope

Webmin installation, service management, network access verification, Pluggable Authentication Modules (PAM) and internal authentication, firewall configuration, recovery procedures, and lab validation tasks

1.0 Network Configuration

1.1 Interface Mapping and Layer 3 Verification

The network configuration at Layer 3 involves mapping two interfaces: `enp0s3`, which is configured for Network Address Translation (NAT), and `enp0s8`, which is set up as a Host-Only interface. This setup supports distinct network functions, with `enp0s3` providing external network access via the DHCP-assigned IP address `10.0.2.15` (external WAN) and enp0s8 enabling isolated communication within the host environment.

The configuration also involves the assignment and verification of `enp0s8` with a static IP address, specifically `172.16.0.10` for the Host-Only interface, to ensure consistent and reliable network identification and connectivity.

Interface Name	VirtualBox Adapter	Role	IP Address	Subnet Mask
enp0s3	Adapter 1 (NAT)	WAN / Updates	**10.0.2.15**	/24
enp0s8	Adapter 2 (Host-Only)	LAN / Management	**172.16.0.10**	/24

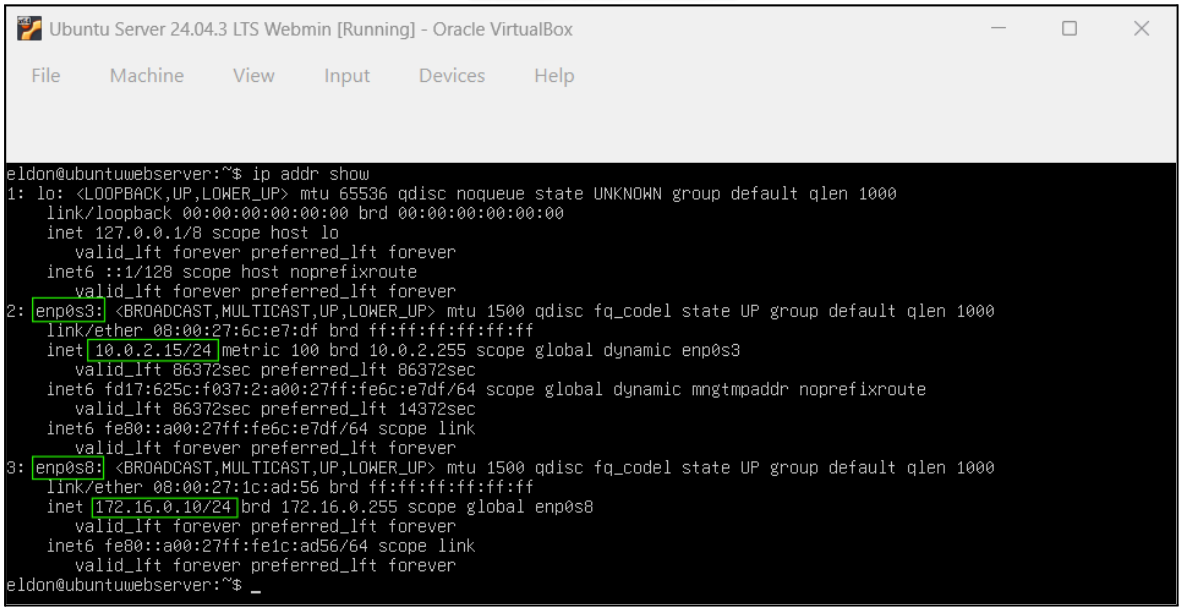

Figure 2: Output of `ip addr` command showing LAN/WAN IPs. March 31, 2026. Eldon G.

2.0 Service & Port Verification

2.1 Webmin Service Status and Layer 4 Verification

The verification process began by confirming that the **Webmin** service was active and running properly. This was achieved by checking the status of `webmin.service` and ensuring that it was in the active state. Following service confirmation, a socket audit was performed to validate that the server was listening on the correct port, specifically port 10000, which is the default listening port for Webmin.

This is typically performed using the command `ss -tulnp | grep :10000`, which filters and displays all TCP and UDP sockets bound to port 10000, along with the associated process details. The output of this command provides proof that the **Webmin** service is correctly bound to the expected IP address and port, thereby confirming its readiness to accept connections.

- **Service Name:** `webmin.service`
- **Listening Port:** `10000`
- **Binding:** `0.0.0.0` (All interfaces)

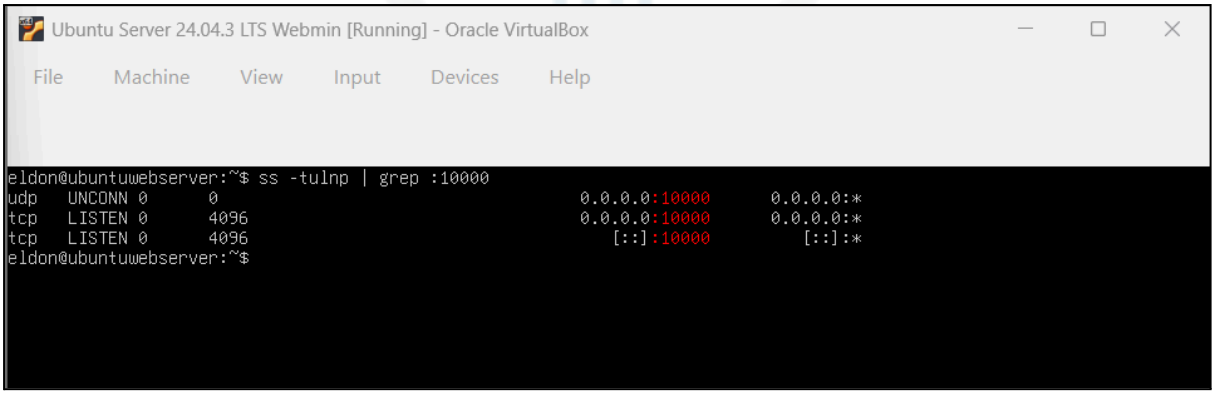

Figure 3: Port 10000 is active and listening to the requests. March 31, 2026 Eldon G.

3.0 Connectivity & Access Control

3.1 Host-to-Guest Connectivity

This demonstrates the successful establishment of communication between the Windows Host and Ubuntu Server within the laboratory environment. This is validated through a series of tests starting with ICMP verification, where a reliable 4 out of 4 ping response confirms network-level connectivity from the Windows Host (IP `172.16.0.1`) to the Ubuntu Server (IP `172.16.0.10`). Further, browser-based access to the Webmin interface at `https://172.16.0.10:10000` is confirmed, indicating that the Webmin service is reachable and responsive over the network.

Authentication to Webmin was performed using PAM-based login credentials, specifically the user `eldon`, to ensure that access control mechanisms were properly enforced and that only authorized users could manage the server via this interface. This connectivity and access control setup verified both network and application layer security and functionality, as outlined in the laboratory documentation.

- **ICMP Verification:** Successful `ping` from `172.16.0.1` to `172.16.0.10`.
- **Browser Access:** Verified login via `https://172.16.0.10:10000`.

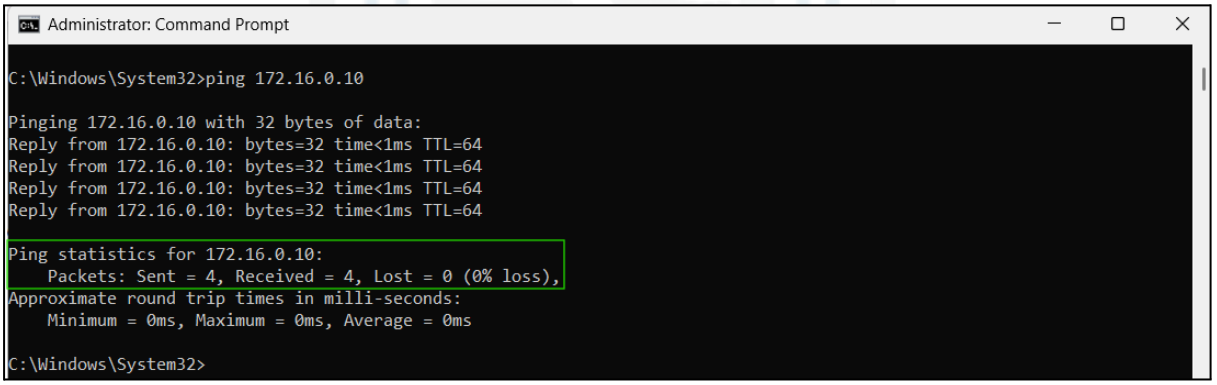

Figure 4: Successful ping to `172.16.0.10`. March 31, 2026 Eldon G.

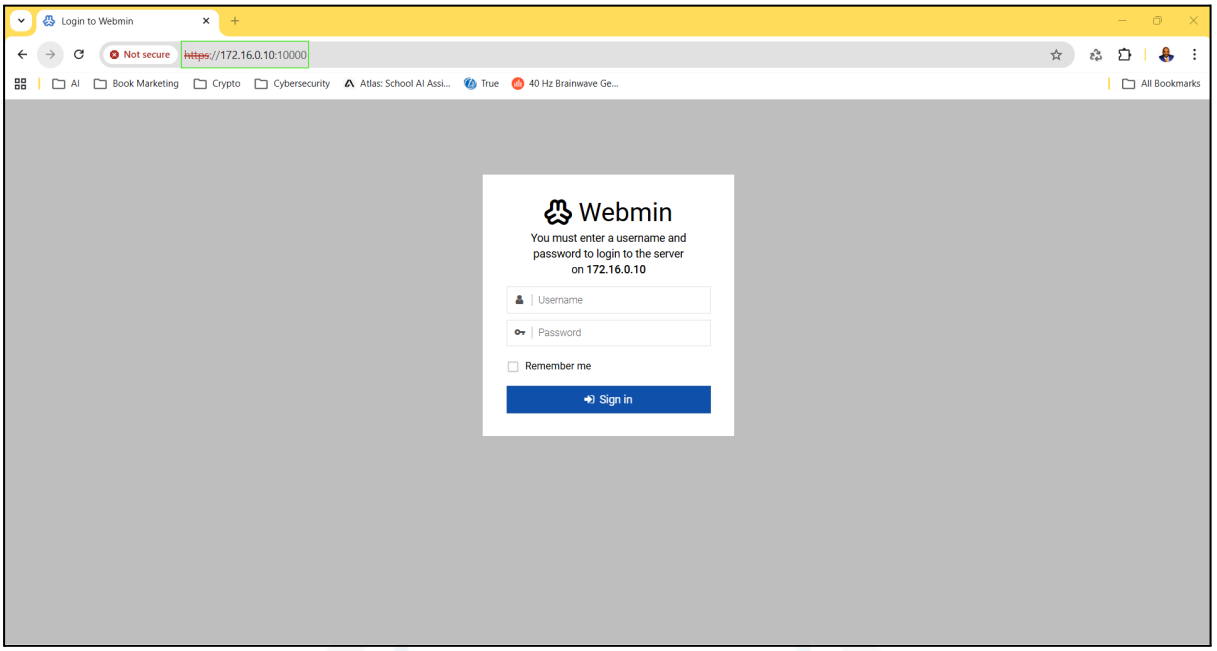

Figure 5: Webmin login page on Windows Host. March 31, 2026. Eldon G.

Technical Note: Virtualization Layer Troubleshooting

During deployment, a synchronization issue was identified between the Windows Host Network Manager and the VirtualBox internal API, preventing the rename of the "VirtualBox Lab Switch."

- **Resolution:** The connection was manually validated via Layer 3 (IP) verification to ensure that the underlying driver was correctly mapped to the 172.16.0.0/24 subnet.
- **Lesson Learned:** Always verify the "Physical" layer (VirtualBox settings) before troubleshooting the "Software" layer (Netplan/OS) when network interfaces fail to initialize.

4.0 Security Hardening

4.1 PAM-Based Authentication

To prevent unauthorized access, the Webmin instance was configured to use **Pluggable Authentication Modules (PAM)**, thereby enabling authentication through standard Linux system users, such as `eldon`, instead of relying on a separate Webmin-specific user database, which is less secure.

This configuration enhances security by leveraging the user management and authentication mechanisms of the existing systems. The setting `pam=1` was verified in the main Webmin configuration file `/etc/webmin/miniserv.conf`, thereby confirming that PAM authentication was enabled. Additionally, the user `eldon` was confirmed to possess `sudo` privileges, granting the necessary administrative rights to perform elevated tasks within the Webmin.

4.2 Protocol Security

Control: Enforcement of encrypted management traffic (encryption-in-transit).

Implementation: > * **Protocol:** All cleartext HTTP access was disabled, and the service was explicitly restricted to **HTTPS (TLS/SSL)**.

- **Socket Binding:** Secure traffic is enforced on **Port 10000**.
- **Security Impact:** This configuration mitigates credential sniffing and man-in-the-middle (MitM) attacks by ensuring the confidentiality and integrity of all administrative sessions.

4.3 Future Hardening Roadmap

Although the current deployment uses encrypted HTTPS and PAM-based authentication, the following security enhancements are scheduled for the next phase of the lab to align with zero-trust principles.

- **Multi-Factor Authentication (MFA):** Implementation of Google Authenticator or Authy modules for Webmin logins.
- **SSH Hardening:** Disabling password-based SSH login on the management interface in favor of SSH Keys.

- **IP Whitelisting:** Restricting Port 10000 access at the OS level of the Uncomplicated Firewall (UFW) to allow traffic from only the Management Host `172.16.0.1`.

5.0 Conclusion

5.1 Key Takeaways

This laboratory demonstrates how to set up a secure Webmin server on an Ubuntu Server 24.04. The system uses two network interfaces to separate the management traffic from the outside traffic.

The setup maintained separate WAN and LAN traffic. This helps control access and maintain network stability. Tests confirmed that Webmin was running using Port 10000 and working over HTTPS. Login security was managed using PAM authentication.

The lab also showed why it is important to check both hardware settings, such as VirtualBox, and software settings in the operating system when troubleshooting.

Overall, the setup proved that Webmin can be used to manage systems safely in an isolated environment.

5.2 Security Implications and Recommendations

This setup still has some security risks, mainly related to login access and network exposure issues. HTTPS and PAM help, but more controls are needed to follow Zero Trust principles and reduce risk.

Technical and Procedural Recommendations

- **Use Multi-Factor Authentication (MFA):** Add MFA using tools such as Google Authenticator or Authy to improve login security.
- **Harden SSH Access:** Disable password logins and use only SSH keys.
- **Restrict Port 10000:** Use a firewall, such as UFW, to allow only trusted IPs, for example, 172.16.0.1.
- **Review User Privileges:** Ensure that users have only the access they need to perform their tasks. Remove extra sudo rights.
- **Maintain Network Separation:** Continue to separate management and update networks to prevent attackers from moving across systems.

Mapping to Security Best Practices and Frameworks

- These steps support the **NIST Cybersecurity Framework**, particularly the Protect function for access control.
- MFA and least privilege follow the **ISO 27001** access control rules.
- Network segmentation and port limits align with **PCI DSS** requirements for secure system design.

These steps improve security, reduce risks, and help meet compliance standards.

Appendix

1.0 Installation

Debian / Ubuntu (Manual Package)

```bash
wget https://prdownloads.sourceforge.net/webadmin/webmin_2.000_all.deb

sudo dpkg -i webmin_2.000_all.deb
sudo apt-get install -f
```

Verify Installation

```
systemctl status webmin
```

2.0 Access & Login

Webmin URL

`https://<VM-IP>:10000`

Authentication

- Default: system `root` account
- Recommended: normal user with `sudo` privileges (PAM-based login)
- Use HTTPS, not HTTP

3.0 Service Management

Check Webmin status

```
sudo systemctl status webmin
```

Start Webmin

```
sudo systemctl start webmin
```

Stop Webmin

```
sudo systemctl stop webmin
```

Restart Webmin

```
sudo systemctl restart webmin
```

4.0 Network & Access Verification

Find the VM IP Address

```
ip a
```

Confirm Webmin is Listening on Port 10000

```
sudo ss -tulpn | grep 10000
```

Access Webmin from a Browser

```
https://<VM-IP>:10000
```

5.0 Logs (Critical for Debugging)

View Webmin Error Log

```
sudo tail -n 50 /var/webmin/miniserv.error
```

View Webmin Access Log

```
sudo tail -n 50 /var/webmin/miniserv.log
```

6.0 Key Files & Directories

Configuration Directory

`/etc/webmin`

Log Directory

`/var/webmin`

Main Configuration File

`/etc/webmin/miniserv.conf`

Authentication Files

```bash
/etc/webmin/miniserv.users
/etc/webmin/webmin.acl
```

7.0 User & PAM Verification

Confirm User Exists

`getent passwd <username>`

Check User Shell (Must Not Be Nologin Or False)

`getent passwd <username> | cut -d: -f7`

Fix The Shell If Blocked

```
sudo usermod -s /bin/bash <username>
```

Test Root Access

```
sudo -i
```

8.0 Webmin Authentication Management

View Webmin Users

```
sudo cat /etc/webmin/miniserv.users
```

View Webmin ACLs

```
sudo cat /etc/webmin/webmin.acl
```

Create or Reset the Webmin Internal User

```
sudo /usr/share/webmin/changepass.pl /etc/webmin <username>
```

Grant Full Webmin Permissions

```
sudo /usr/share/webmin/set-permissions.pl /etc/webmin
<username>
```

9.0 PAM-only Mode for Ubuntu 24.04

Edit Configuration

```
sudo nano /etc/webmin/miniserv.conf
```

Required Settings

```bash
pam=1
passwd_mode=0
```

Force PAM Authentication

```
sudo rm -f /etc/webmin/miniserv.users
```

Restart Webmin After Changes

```
sudo systemctl restart webmin
```

10.0 IP Blocking (Silent Lockout Fix)

Clear the Blocked IPs

```bash
sudo rm -f /etc/webmin/blocked-hosts
sudo rm -f /var/webmin/blocked-hosts
```

Disable Webmin IP Blocking

Edit `/etc/webmin/miniserv.conf` and set

```bash
blockhost_failures=0
blockhost_time=0
```

Note: Temporary bypass for configuration; ensure that `blockhost_failures` is restored to `5` before deployment.

Restart Webmin

```
sudo systemctl restart webmin
```

11.0 Firewall Notes (Lab Context)

- Use **Webmin → Networking → Linux Firewall**
- Implement **deny-all + whitelist** logic
- Restrict Port `10000/TCP`
- Test from:
  - **Whitelisted IP:** Access allowed
  - **Non-whitelisted IP:** Connection blocked
- The browser mode does not change the IP address. Source IP is important.

12.0 Full Webmin Reset (Last Resort)

Stop Webmin

```
sudo systemctl stop webmin
```

Backup Configuration

```
sudo cp -r /etc/webmin /root/webmin-backup
```

Remove Auth State

```bash
sudo rm -f /etc/webmin/miniserv.users
sudo rm -f /etc/webmin/webmin.acl
sudo rm -f /etc/webmin/miniserv.pem
```

Re-Run Setup

```
sudo /usr/share/webmin/setup.sh
```

Start Webmin

```
sudo systemctl start webmin
```

13.0 Common Modules

System

- Users and Groups
- Software Packages
- Running Processes
- Scheduled Cron Jobs

Networking

- Linux Firewall
- Network Configuration

Others

- Command Shell
- File Manager

14.0 Security and Lab Usage Notes

- Default port: `10000`
- Change port in `/etc/webmin/miniserv.conf` if required
- Ensure Webmin version is **> 1.997** (CVE-2022-36446 mitigation)
- Re-enable IP blocking only after firewall rules are correct
- Prefer PAM-only login using a `sudo` user (e.g., `eldon`)
- Use `sudo` inside Webmin for admin tasks
- Restrict access using the Webmin Firewall module (IP whitelist)
- Re-enable IP blocking only after firewall rules are in place
- Keep Webmin & Cockpit on **separate VMs** for operational clarity

This guide covers all the commands required to recover, secure, and demonstrate Webmin access on Ubuntu 24.04.