

# Raspberry Pi Network Hardening Project

Manas Jha

June 2023

## Overview

I'm working on a small personal project to see how much security I can add to my Raspberry Pi setup with just a few tweaks. The main goal isn't to build a professional system — it's to experiment and understand how VPNs, host firewalls, and access controls stack together to make a network tougher to break into. It's basically a mix of curiosity, problem-solving, and learning by doing.

## Project Idea

The idea is pretty straightforward: lock down my Pi so that any admin or SSH access has to go through a private VPN, and then tighten things up with strict firewall rules. I want to see in real time how that extra layer changes what's visible from the outside and how it affects remote access.

## Implementation Summary

- I set up an admin-only VPN so that any management traffic first goes through a secure tunnel.
- I switch the firewall to a default-deny inbound policy so only the VPN port stays open.
- I add protection against repeated login attempts and run a simple log monitor to catch suspicious activity.
- I take full system snapshots and test restoring them to make sure I can recover from mistakes.
- I put together a small Python script that watches for repeated failed login attempts and automatically kicks off an incident workflow when a threshold is hit. My goal here is learning — I don't want to break real systems — so the automated action only targets a set of dummy files I prepare ahead of time. When the trigger fires the system isolates the device from the network, captures detailed forensic logs, and flags the event to my admin channel so I can inspect what happened. After each run I restore from a fresh snapshot and confirm backups are intact, then I remove the experimental code or lock it down so nothing accidental ever runs on production.

## Purpose and Learning

I'm doing this to actually see the effect of adding just one more layer of protection. Even with a setup this simple, I can already see the concept of defense-in-depth in action — one control backs up another. It's helping me understand how these pieces fit together and why layered security matters way more than relying on just one tool.

## Remaining Vulnerabilities

Even though the setup feels tighter, I can still spot some weak spots:

- **MITM Attacks:** If the VPN certs or configs get leaked, someone could still intercept traffic.
- **Social Engineering:** No firewall helps if someone tricks you into giving away credentials.

- **Misconfiguration:** One small rule mistake in the firewall or VPN setup could undo all the security.
- **Zero-day Exploits:** Any unpatched service could still get hit.

## Results and Observations

Right after setting this up, I can see fewer visible ports when I scan from another device. The number of failed SSH attempts drops immediately. Everything feels more under control — every connection now shows up in the logs, so I know exactly what's happening. It's cool to see how much of a difference just one extra layer can make.

## Takeaways

This project shows me that even small changes have a big impact. Security isn't about making things perfect — it's about making attacks slower, louder, and harder to pull off. Doing this hands-on gives me a real sense of how network layers work together, instead of just reading about it.

## Final Thoughts

I'm not doing this for profit or certification — it's just a fun way to learn and test ideas in real time. Even though it's not bulletproof, the setup already feels way more solid. The main lesson here: one extra layer of protection can make all the difference, especially when you're working with small systems like a Raspberry Pi.