Access Control & Identity Management

Security was at the forefront of this project — an idea that sparked a huge part of the build, and the topic I enjoyed most during graduate school. In the Foundation chapter, the focus was around access control at the firewall level, making sure that only specific devices could reach the homelab and that only the necessary ports were open internally for devices to communicate between VLANs. Many of the applications I started deploying had their own basic authentication built in, and for the most part that's more than enough — especially since I'm the only one logging in. However, I wanted to explore Identity Providers and Single Sign-On, and see what it would take to connect all my apps under one umbrella.

I did a lot of research on which solution was the most secure, approachable to learn, and the right fit for my use case. Each option had its own strengths and weaknesses, but I landed on Authentik. Mostly because of its clean UI and the vast library of pre-built integrations it offered. I also follow Jim's Garage on YouTube, and he featured it in several videos. While Authentik seemed more straightforward than some alternatives, it was also very granular. Despite being open source, it's built with enterprise-scale deployments in mind, so the amount of configuration involved was extensive. I appreciated the detailed logs and the depth of data available for troubleshooting. The one downside I experienced was around updates — they could occasionally break things, so I learned to always take a backup before attempting any version upgrade.

Once I had Authentik set up and ready, it was time to learn how to actually use it. Through the available documentation and tutorial videos, I was able to get a solid understanding of the concepts and setup process. I learned the distinction between a *provider* and an *application* in Authentik's model, and the various rules and settings that control authentication behavior. For most applications, I aimed to enable passwordless authentication. Around 70% of the apps I run on my homelab connect via OIDC (OpenID Connect), which is a modern identity layer built on top of OAuth 2.0. The setup is relatively straightforward once you understand the flow. For the remaining 30%, the applications were simpler and had no native SSO support. For those, Authentik offered a Proxy authentication option, where Authentik injects an authentication layer in front of the app through my reverse proxy. Any time someone tries to access one of those applications, they're redirected to Authentik first. However, some applications had no way to disable their own built-in basic authentication, which meant the proxy pass-through didn't work cleanly — at least not in my environment.

Other applications did support disabling their native login entirely and forwarding authentication straight to SSO, which made the experience seamless. Once you're authenticated, you're in, no second prompt. The logs have been useful for both initial setup and ongoing troubleshooting, though I haven't yet wired Authentik into my email for alert notifications or connected it to a

centralized SIEM for log aggregation — both things on my list.

Setting up my own IdP and establishing SSO across my homelab not only made access easier and more consistent, it gave me hands-on experience with identity and access management concepts that I wouldn't have gotten any other way. It turned out to be one of the first homelab skills I directly applied at work. At the time, my organization was standing up one of its first cloud systems, and the security team wanted to use an IdP rather than exposing access directly to Active Directory. They went with Okta, which operates on very similar principles to Authentik. Since no one on the team had much Okta experience, I was able to step in and help with the initial setup — establishing the connections between Okta, our Active Directory, and various applications. While the corporate environment used SAML 2.0 rather than OIDC, the underlying concepts were familiar enough that I could navigate it confidently. Over time, the number of systems connected through Okta continued to grow.

For the most part I still use Authentik today, but I've started evaluating alternatives. Authentik has a strong community and is a genuinely capable tool, but it comes with more configuration surface area than I actually need, and complexity, when not managed carefully, can itself become a security liability. I began exploring other open-source options like Zitadel and Pocket ID, and ultimately landed on Pocket ID for its simplicity and fully passwordless design. It supports only passkeys, which makes it inherently resistant to brute force and credential stuffing attacks. It's still a relatively new project, and I'm still experimenting to see whether it can fully replace Authentik in my environment. It doesn't have the same depth of logging, and it only supports OIDC, but unlike Authentik, it's extremely lightweight. The entire application runs in a single container, whereas Authentik requires at least four, including a dedicated database, largely because it needs to store credential data. Pocket ID, storing no passwords at all, simply doesn't need that overhead.

This is part of a broader practice I've adopted: every tool in my homelab goes through an annual review to evaluate whether there's something better, more stable, or more actively maintained available in the open-source community. Nothing is set in stone, the homelab evolves as the ecosystem does.

Revision #1
Created 2026-05-12 00:35:29 UTC by lumxux
Updated 2026-05-12 00:36:35 UTC by lumxux