NetBird: The Private Mesh VPN That Retires Old VPNs (Integrated with Authentik)

NetBird: The Private Mesh VPN That Retires Old VPNs (Integrated with Authentik)

Let's face it: traditional VPNs (OpenVPN, IPsec, or even WireGuard in a classic hub-and-spoke configuration) suck. They are slow, centralized, and create a massive *single point of failure*. If the main VPN server goes down, the entire company stops communicating. And if the server is up, all traffic must transit through it before reaching its destination, creating an absurd bottleneck.

Then came the era of **Mesh VPNs**, and specifically **NetBird**.

Today we'll look at what NetBird is, why it is a thousand times better than old-school VPNs, and how to install it self-hosted, exploiting our trusted keymaster: **Authentik** (whose infrastructure we already configured here).

And the best news is that following the updated official guide, the installation has become extremely simple. NetBird's automatic installer handles configuring everything (from the Nginx reverse proxy to the Docker Compose, including the default integrated Identity Provider), leaving us only with the task of hooking up Authentik directly from the web panel, without touching a single line of the `.env` file.

Why NetBird Changes Our Lives (Without the Marketing Fluff)

NetBird takes *WireGuard's* ultra-fast encryption protocol and turns it into an intelligent overlay mesh network.

1. **Direct Communication (Peer-to-Peer)**: If machine A needs to talk to machine B, NetBird uses a coordination server only to introduce them. Once the route is established, traffic flows directly

between A and B in encrypted form. The server doesn't see a single byte of our traffic. No bottlenecks, latency reduced to zero.
2. **NAT Traversal**: NetBird integrates STUN and TURN. This means our nodes connect directly even if they are behind double NATs, restrictive ISP routers, or 4G/5G mobile networks. Zero ports to open on your home firewall.
3. **Access Control (Zero Trust)**: Instead of giving the user access to the entire network (like with classic VPNs), on NetBird we can define granular policies: *"The Developers group can only access port 22 on Server X"*.
4. **SSO Integration with Authentik**: We don't have to manage passwords or keys for each client. Anyone wishing to access the network authenticates via Authentik (with mandatory MFA, if desired). If you revoke the user in Authentik, they are instantly kicked out of the mesh network.

Step 1: Preparation in Authentik

First, we need to tell Authentik that NetBird is a trusted application. Log into Authentik's administrative interface.

Create the Provider

1. Go to **Applications** -> **Providers** and click **Create**.
2. Select **OAuth2/OpenID Provider** and click Next.
3. Fill in the fields:
   - **Name**: `NetBird`
   - **Authorization flow**: `default-provider-authorization-explicit-consent` (Authorize Application)
   - **Client type**: `Confidential`
   - **Redirect URIs/Origins**: Leave blank for now (we will add it in Step 4).
   - **Signing Key**: Select Authentik's self-signed certificate.
4. Click **Finish**.
5. Note down the **Client ID** and **Client Secret** that appear on the screen.

Create the Application

1. Go to **Applications** -> **Applications** and click **Create**.
2. Fill in the fields:

- **Name**: `NetBird`
- **Slug**: `netbird`
- **Provider**: Select the `NetBird` provider created in the previous step.

3. Click **Create**.

Authentik's OIDC Issuer URL will be: `https://<YOUR_AUTHENTIK_DOMAIN>/application/o/netbird/`

Step 2: Installing Self-Hosted NetBird

NetBird provides an automatic installation script that configures the entire Docker Compose environment, including Nginx and SSL certificates. We don't have to invent weird configurations.

Log into our target server via SSH and execute:

```
# Download and run the automatic setup script
curl -fsSL https://github.com/netbirdio/netbird/releases/latest/download/getting-started-with-self-hosted-netbird.sh | sh
```

The script will do all the heavy lifting and start the Docker containers. Once the process is complete, we'll have a functioning NetBird dashboard accessible on the domain we configured during the script. In this phase, you will also be asked to create the domain to use and what type of reverse proxy you intend to have (which will then provide the config file).

Step 3: Configuring the OIDC Provider in NetBird

Now we link Authentik directly from NetBird's web interface, without modifying configuration files on the server.

1. Access the web dashboard of our self-hosted NetBird.
2. Go to **Settings** -> **Identity Providers**.
3. Click **Add Identity Provider**.
4. Select **Generic OIDC** and fill in the fields:
    - **Name**: `Authentik` (or whatever name you want to see on the login button)

- **Client ID**: Paste the Client ID copied from Authentik.
- **Client Secret**: Paste the Client Secret copied from Authentik.
- **Issuer**: Enter Authentik's issuer URL (e.g., `https://auth.example.org/application/o/netbird/`).

5. NetBird will show us a **Redirect URL**. **Copy it** before continuing, without clicking Add Provider yet.

Step 4: Configuring the Redirect URI in Authentik

Go back to Authentik to secure the authentication flow.

1. Go to Authentik Admin -> **Applications** -> **Providers** and select the NetBird provider.
2. Click **Edit**.
3. Under **Redirect URIs/Origins**, paste the Redirect URL copied from NetBird's dashboard.
4. **Crucial Option for Logout**: Add NetBird's logout callback as a second address (on a new line): `https://<NETBIRD_DOMAIN>/oauth2/logout/callback` (replacing `<NETBIRD_DOMAIN>` with our NetBird domain). This ensures that when we disconnect from NetBird, we get redirected back cleanly.
5. Make sure the option is set to **Strict** (and not Regex) to match the addresses exactly.
6. Save by clicking **Update**.

Step 5: Complete Setup on NetBird and Test

1. Go back to NetBird's dashboard tab.
2. Click **Add Provider**.
3. Log out of the NetBird dashboard.
4. On the login page, you will now see the **Authentik** button (or the name we chose).
5. Click the button and log in via Authentik: you will be redirected back and authenticated instantly.

Step 6: Synchronizing User Groups (JWT Claims)

If we want the groups configured on Authentik to reflect on NetBird to create automatic network rules, proceed like this:

In Authentik:

1. Go to **Applications** -> **Providers** and edit the NetBird provider.
2. Under **Advanced protocol settings**, check the **Include claims in id_token** box.
3. Make sure the `profile` and `groups` scopes are active for that provider (in Authentik, the `profile` scope includes group information by default).
4. Click **Update**.

In NetBird:

1. Go to **Settings** -> **Groups**.
2. Enable **JWT group sync**.
3. Set the **JWT claim** field to `groups`.
4. Save settings.

From this moment on, NetBird will read group membership directly from the OIDC token signed by Authentik and synchronize users' machines into their respective network groups.

Welcome to modern, clean networking without the headache of configuration files.

Bonus track

There are cases where requesting to open port 51820/udp inbound/outbound to allow peer-to-peer mesh traffic can raise the eyebrows of some old-school sysadmin who might suspect a tunnel.

Netbird allows us to change the port for p2p directly from the client:

```
netbird up --wireguard-port 443
```

At least we can make them believe we are asking for an inbound/outbound opening for normal HTTP/3 traffic.

[May the Talpone be with you]