

Forum:[Hack Forums](#)

Thread: [{Tutorial} Setup your own IRC server over I2p](#)

Pr0m3th3us - June 9, 2026, 6:50 pm

Setup your own IRC server over I2p

Every message you send on Discord, WhatsApp, Telegram is logged, scanned, and stored — even the private ones. If you want a more anonymous way of talking with your friends, you're going to need your own chat server. In this TUT I'm going to show you how to build your own private chat server using IRC, or Internet Relay Chat, over I2p.

Before we get started, let's first explain what IRC and I2p are, especially if you've never used them before. IRC is kind of like the original group chat protocol. It's old — it's really old — but it's also super lightweight. There's no accounts, no login, just servers and channels. It's kind of like texting over the internet, but in a more raw and direct kind of way. I2P (Invisible Internet Project) is an anonymous network layer that runs on top of the regular internet. Instead of routing your traffic directly to a destination, it wraps it in multiple layers of encryption and bounces it through a series of volunteer-run nodes called routers — similar in concept to Tor but with some key architectural differences. This guide walks you through hosting your own private IRC server as an I2P tunnel and connecting to it with HexChat. Like the Tor version, the result is an anonymous group chat with no accounts, no metadata, and no tracking. I2P is particularly well suited for internal darknet services like this — traffic stays within the I2P network and never touches the clearnet. Let's jump in.

Tested on Debian/Ubuntu

What you need: A Linux machine (Debian/Ubuntu Distro)

Sudo access

HexChat installed on any client machines

I2P vs Tor — quick note: I2P uses an HTTP proxy on port 4444 (not SOCKS5/9050 like Tor)

Tunnels are configured through a web UI instead of a config file

Bootstrap takes longer (~5 minutes vs Tor's ~30 seconds)

I2P addresses are base32 strings ending in .b32.i2p

Step 1 — Install packages

Add the I2P repo and install:

```
Code sudo apt-get install -y apt-transport-https curl
curl -o /tmp/i2p.key https://geti2p.net/en/download/debian/dists/stable/Release.gpg
sudo apt-key add /tmp/i2p.key
echo "deb https://deb.i2p2.de/ stable main" | sudo tee /etc/apt/sources.list.d/i2p.list
sudo apt update && sudo apt install -y i2p inspircd hexchat
```

Alternatively if you don't want the official repo, the package in your distro's repos works too but may be older:

```
Code sudo apt update && sudo apt install -y i2p inspircd hexchat
```

Step 2 — Configure InspIRCd

Same as the Tor version — bind InspIRCd to localhost only so it's not exposed on your network interface:

```
Code sudo nano /etc/inspircd/inspircd.conf
```

Set the bind block:

```
Code <bind address="127.0.0.1" port="6667" type="clients">
```

And give your server a name:

```
Code <server name="yourserver.local" description="Private IRC" network="YourNet">
```

Save and close.

Step 3 — Start I2P and wait for bootstrap

Enable and start I2P:

```
Code sudo systemctl enable --now i2p
```

Now open the I2P router console in your browser:

```
Code http://127.0.0.1:7657
```

Wait for the network integration bar to reach "High" or "Good" — this usually takes 3–10 minutes on first run. Don't proceed until it's integrated or your tunnel won't work reliably.

Step 4 — Create a server tunnel

This is the I2P equivalent of configuring a hidden service in torrc.

1. In the router console, go to Hidden Services Manager:

```
Code http://127.0.0.1:7657/i2ptunnelmgr
```

2. Under "I2P Server Tunnels", click "New server tunnel"

3. Fill in the form:

- Type: Standard (TCP)
- Name: anything you like (e.g. MyIRC)
- Host: 127.0.0.1
- Port: 6667
- Auto Start: checked

4. Click Save

5. Back in the tunnel manager, find your new tunnel and click on it. You'll see a Local destination

field — this is your base32 address. It looks like:

Code `abc123def456...xyz.b32.i2p`

Copy and save this. This is the address clients use to connect to your IRC server. Share it with whoever you want to chat with.

Note: The tunnel needs a few minutes after creation before it's reachable by others — I2P has to publish the destination to the network.

Step 5 — Start InspIRCd

Code `sudo systemctl enable --now inspircd`

`sudo systemctl status inspircd`

Should show active (running).

Step 6 — Connect with HexChat

This is done on any machine that wants to connect. Each client machine also needs I2P running with a client tunnel or just the HTTP proxy enabled (it's on by default).

1. Open HexChat -> HexChat menu -> Network List
2. Click Add and name your network (e.g. MyPrivateIRC)
3. Click Edit, remove the default server entry and add:

Code `yourdestination.b32.i2p/6667`

4. Uncheck "Use SSL for all servers on this network"

5. Go to the Advanced tab and configure the I2P proxy: Proxy type: HTTP

Hostname: `127.0.0.1`

Port: `4444`

6. Click Close then Connect

HexChat will route through the I2P HTTP proxy and connect to your server tunnel.

Step 7 — Create a channel

Once connected, in the HexChat input box:

Code `/join #yourchannel`

Anyone else connected to the same destination can join the same way.

Troubleshooting

Connection times out — I2P hasn't fully integrated yet, or the tunnel was just created and hasn't propagated. Wait 5–10 minutes and try again.

"Could not connect to proxy" — I2P isn't running or the HTTP proxy isn't listening on 4444. Check `systemctl status i2p` and verify the proxy is enabled in the router console under Clients.

Tunnel shows "Standby" in tunnel manager — Normal on first creation. It activates once I2P builds the tunnel paths. Give it a few minutes.

inspired won't start — Config syntax error. Run `sudo inspired --runasroot --debug` to see the error.

Clients can't reach the server — Make sure the server tunnel is set to Auto Start and I2P is fully integrated (not just running). Also confirm InspIRCd is actually listening: `ss -tlnp | grep 6667`

Notes

Your I2P destination address is tied to the private key stored in the tunnel config. Don't delete the tunnel or you'll get a new address and have to reshare it.

To back up your destination, export the tunnel's private key from the Hidden Services Manager.

I2P's HTTP proxy on port 4444 handles .i2p addresses — it won't route clearnet traffic by default, which is fine for this use case.

For channel access control, use InspIRCd channel modes: `/mode #channel +k yourpassword` sets a join password.

I2P encryption is end-to-end by design — all traffic between peers is layered-encrypted within the network.

Notice: I made this TUT with help from AI. I put it together quick for user [DeathPenalty666](#) it is not being used as an entry for brotherhood contest. Thanks.