

# Large-scale 1-1 NAT

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2007-09/msg00237.html>

---

- *From:* Christopher Cowart <[ccowart@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:ccowart@xxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 24 Sep 2007 00:25:17 -0700
- 

Hello,

We're working on expanding our wireless network. Unfortunately, we're running out of IP addresses (aren't we all). As much as I'd love to just tell everyone to use IPv6, that isn't gonna fly. The next plan to consider is using an RFC1918 pool and NATing the traffic.

If only it were that simple. The security folks have mandated that anyone who can talk to the internet at large must be individually identifiable. This means having hundreds of users NATing to a single internet-routable IP isn't happening.

I want to try a setup in which we have a big RFC1918 pool of addresses, say 10.8/16. In their initial state, these hosts might NAT to a single public IP and perform some transparent proxying to get them to an authentication page. The firewall on our NAT box would be extremely restrictive for these clients.

When a user authenticates, we will allocate a single public IP for the session. At this time, our code would use ipfw to move the user into a different lookup table and also update the NAT table.

The real question is: what's the best way to dynamically update the NAT table?

It doesn't look like there's any way to have a running natd update its configuration without restarting. That's obviously disruptive.

I also doubt it's a good idea to try to launch a single natd process per authenticated client. We have a /22 and a /23 in our public pools, and we expect to max that out (1500+ clients).

Has anyone attempted a setup like this? Do you have any pointers for designing this to scale well? We are planning on throwing hardware at it, but that only gets us so far.

Thanks for your help,

—

Large-scale 1-1 NAT

Chris Cowart  
Lead Systems Administrator  
Network & Infrastructure Services, RSSP-IT  
UC Berkeley

***Attachment:*** [pgpOOjSH4ZfmD.pgp](#)

*Description:* PGP signature