BIND9 Dynamic DNS (https://blog.jasonantman.com/2010/04/bind9-dynamic-dns/)

I needed a better solution for Dynamic DNS than dyndns.org (http://www.dyndns.org) for something, so I set about setting up DDNS through my BIND9 servers. I found a number of very helpful blog posts, including nsupdate: Painless Dynamic DNS (http://linux.yyz.us/nsupdate/), Painless DDNS part 2: the server (http://linux.yyz.us/dns/ddns-server.html), Secure dynamic DNS howto (http://ops.ietf.org/dns/dynupd/secure-ddns-howto.html) and A DDNS Server Using BIND and Nsupdate (http://www.oceanwave.com/technical-resources/unix-admin/nsupdate.html). Of course, the BIND configuration statement reference (http://www.zytrax.com/books/dns/ch7/statements.html) was also very helpful.

The whole process was relatively simple...

1. Add an RR for the host I want, in the appropriate zone.
2. Generate TSIG keys, distribute them to the client.
3. Make sure you have logging enabled for things of interest in your `logging` section of `named.conf`:

```
category dnssec { syslog_info; };
category update { syslog_info; };
category security { syslog_info; };
```

4. Create a keys.conf file for your keys (you *do* split your configs out into

---

Migrating Kodi to MariaDB, with locked-down per-missions and no im-port/export (https://blog.ja-so-nant-man.com/2021/03/migrating-kodi-to-mariadb-with-locked-down-per-missions-and-no-importex-port/)

OctoPrint Power Outage Ha ndling (https://blo g.ja-so-nant-man.com/2021/03

usable chunks, right?):

```
key foo.example.com. {
    algorithm HMAC-MD5.SIG-ALG.REG.INT;
    secret "this is your secret here (after Key: in th
e .private file)";
};
```

and include it in `named.conf` like:

```
include "/etc/named.d/keys.conf";
```

5. Set an `update-policy` (http://www.zytrax.com/books/dns/ch7 /xfer.html#update-policy) statement in `named.conf`. I just added mine to a specific zone in a specific view (external), as that's the only place I would conceivably want updates right now.

```
update-policy {
    grant * self * A TXT;
};
```

Assuming your TSIG keys are named for specific RRs, this will let any client (with a valid key setup on the server) update its own RR and nothing else.

6. Finally, I created a script for ddns updates on the client. Since I want to be able to fire off this script manually or via cron (if I have to reload BIND, and until I make the needed changes to MultiBINDadmin), I bypassed the usual dhclient stuff and manually grab the current IP from the interface of interest. I symlinked this in `/etc/dhcp3/dhclient-exit-hooks.d` so it will run on DHCP updates.

```
#!/bin/bash
IFACE="eth0"
TTL=3600
SERVER=ns1.example.com
HOSTNAME=foo.example.com
ZONE=example.com
KEYFILE=/root/ddns-keys/Kfoo.example.com.+157+12345.pri
vate

new_ip_address=`ifconfig $IFACE | grep "inet addr:" | a
wk '{print $2}' | awk -F ":" '{print $2}'`
new_ip_address=${new_ip_address/ /}

nsupdate -v -k $KEYFILE << EOF
server $SERVER
zone $ZONE
update delete $HOSTNAME A
update add $HOSTNAME $TTL A $new_ip_address
send
EOF
```

When I finally got things setup, my only problem was with permissions on the zone file directories, which was easily corrected.Once this was straightened out, my `nsupdate` script ran flawlessly, and the update was instantly (thanks to using "notify") propagated out to the slave server.

The only problem that I now have is one of my own creation - I use a small PHP application that I wrote (MultiBINDadmin (http://multibindadmin.jasonantman.com/)) to manage DNS. It's incredibly easy, as it keeps track of internal and external IPs, and everything else, for my zones, and triggers a pull on the master BIND server via the web interface. The only problem I now have is that this messes with DDNS updates. First, if I make changes in the web interface and there's already been a DDNS update that day, the zone serial generated by MultiBINDadmin will match the automatically incremented serial generated by the BIND server. Second, and more troubling, when the BIND server reloads, it loses the dynamic update. So when I push changes to a zone from the web interface, my dynamic updates go away.

For the short-term, I'm just going to check the zone serial before I make any updates and, if need be, manually increment it in the web tool. As to losing the dynamic updates, I'm just going to have cron on the client fire the `nsupdate` script every 30 minutes. I also did a little kludge, setup a vhost on one of my web servers to answer for the dynamic host (as a catch-all page), and set the IP of my web server as the hard-coded RR address in the zone file. If someone tries to use the new (DDNS through my BIND server) address for HTTP and for some reason the current dynamic address disappeared (BIND reloaded), they'll get a little page with a message and the old dyndns.org-based URL.

optimiza-
tion/)

Arch Linux
on a Dell
Precision
5530 / XPS
15
9570 Lapto
p
(https://blo
g.ja-
so-
nant-
man.com
/2019/03
/arch-linux-
on-a-dell-
precision-
5530--xps-
15-9570-
laptop/)

🐦 Latest
Tweets

🐙 GitHub
Repos

awslim-
itchecker
(https://git
hub.com
/jantman
/awslim-
itchecker)
A script and
python pack-
age to check
your AWS
service limits
and usage via
boto3.

When I get around to it (or when this becomes a problem), I'll make two changes to MultiBINDadmin:

1. Before it pushes an update, check the current serial for the zone (ok, this may be a bit interesting, as the internal and external zones could have different serials) and increment from that.
2. Have a "DDNS" flag in the database and GUI for RRs. For all flagged RRs, try to get the (unfortunately external) current address and update the record in the DB before the push.
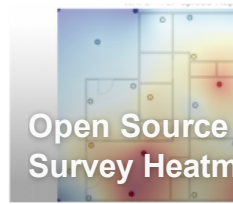
The real question here, which I haven't looked into yet, is how I can interrogate BIND about RRs for the external zone from an internal host.

Comments

Linux Surveillance Camera Software …

Cloud Custodian Architecture, D

Cloud Custodian Architecture, …

Open Source Survey Heatm

misc-scripts (https://github.com/jantman/misc-scripts) A collection of my stand-alone scripts to small/quick for their own repos. All kinds of useful stuff.

biweekly-budget (https://github.com/jantman/biweekly-budget) Responsive Flask/SQLAlchemy personal finance app, specifically for bi-weekly budgeting.

repostatus.org (https://github.com/jantman/repostatus.org) A standard to easily communicate to humans and machines the

Jason Antman's Blog   🔒 Privacy Policy   **1** Login

♡   Sort by Best

Start the discussion…

Name

chines to per-
form WiFi site
surveys and
present the
results as a
heatmap
overlayed on
a floorplan

@jantman
(https://github.c
om/jantman) on
GitHub

↗ Links

Homepage
(http://ww
w.jasonant-
man.com)

Resume
(http://re-
sume.jaso-
nant-
man.com)

↑ Back to top