ENGR3821 Network Engineering

NETENG ASSIGNMENT 3

June 25, 2019

Table of Contents

Disclaimer

This report and its contents are permitted to be redistributed on behalf of the author for the express purpose of peer review and grading only.

Introduction

This report will cover the server selection, setup, configuration, and testing of a master-slave DNS server configuration. The goal is to create a pair of DNS servers where one is hidden from the public and the other will take requests on the other’s behalf. This is accomplished with BIND, a well-known DNS server.

DNS Server Installation

Server Selection

The two main DNS packages that come to mind are Dnsmasq by Simon Kelley and BIND 9. Dnsmasq is a lightweight DNS and DHCP server commonly used within small networks and in embedded devices licenced under the GPLv3[1]. BIND on the other hand is a tried-and-tested DNS server licenced under the MPLv2 that has seen extensive deployment after being introduced as the first DNS server[2].

Unfortunately, Dnsmasq does not seem to support acting as a secondary DNS server, as hinted by the manual page only referencing forwarding to another secondary DNS server, not accepting connections itself[3]. However, BIND supports it out of the box, so it wins on both familiarity and functionality.

Installation

Installing BIND 9 is quite simple for Ubuntu 18.04: `sudo apt install bind9`. It does not require any additional dependencies that are not automatically installed by apt.

To start BIND on boot and start the service now, run `sudo systemctl enable bind9 && sudo systemctl start bind9`

Configuration

Master DNS Configuration

The steps below will configure BIND as the master DNS server for the following domains:

- neoimperialexports.com.aq
- lanthanideproductions.com.nz
- unimaginableuniversity.com.ki
- monopolisticusedvehicles.com.aq
- monopolistictradingcompany.com.nz

Each domain will have ‘www’, ‘ftp’, and ‘mail’ subdomains.

1. Open ‘/etc/bind/named.conf.local’ with a text editor
    a. For each domain, add the following block of text:
    ```
    zone “<domain>” {
        type master;
        file “/etc/bind/db.<domain>”
    };
    ```
    Where `<domain>` is the domain you wish to add
2. For each domain, copy ‘/etc/bind/db.empty’ to ‘/etc/bind/db.<domain>’, where <domain> is the domain
3. Open ‘/etc/bind/named.conf.options’ with a text editor and add `notify  yes;` to send zone update notifications to slave name servers

Now there is a configuration file for each domain with a very basic template. The following steps apply to each domain individually. For simplicity, ‘neoimperialexports.com.aq’ will be used as the domain in this example.

4. Open the domain’s configuration file with a text editor
    a. (Optional) Remove or change the comment block at the top of the file stating not to edit this file
    b. In the SOA record, replace `localhost.` and `root.localhost.` with the desired name for this DNS server and the domain name. Here it will be `ns1.neoimperialexports.com.aq.` and `root.neoimperialexports.com.aq.`
    NOTE 1: Make sure to include the trailing `.` when entering name server names!
    NOTE 2: When changing the domain configuration in future, make sure to increment the serial so that other name servers know to update to the latest zone version

c. In the NS record, replace `localhost.` with the name you set in the previous step
d. Copy the NS record and set it to the desired name for the slave name server. Here it will be `ns2.neoimperialexports.com.aq.`

Now that the template is reconfigured for our domain, we can start adding new records for each subdomain. For this example, we’ll be using A records pointing to 10.1.1.1-2,20-22.

e. Under the NS record, add a new A record that resembles something like: `ns1  IN  A  10.1.1.2`
The components are, from left-to-right: name, record class (INternet), record type, and record data (in this case the IP address)[4]
NOTE: An A record for each nameserver is required to send zone update notifications
f. Repeat the above for the slave name server and each of the ‘www’, ‘ftp’, and ‘mail’ subdomains and their respective IP addresses

The final domain file should look similar to this:

```
$TTL 86400
@    IN   SOA  ns1.neoimperialexports.com.aq.
root.neoimperialexports.com.aq. (
                     1        ; Serial
                604800        ; Refresh
                 86400        ; Retry
               2419200        ; Expire
                 86400 )      ; Negative Cache TTL
;
@    IN   NS   ns1.neoimperialexports.com.aq.
@    IN   NS   ns2.neoimperialexports.com.aq.
ns1  IN   A    10.1.1.2
ns2  IN   A    10.1.1.1

www  IN   A    10.1.1.20
mail IN   A    10.1.1.21
ftp  IN   A    10.1.1.22
```

One that is completed for each domain, test them with `named-checkzone <domain> /etc/bind/db.<domain>` to make sure BIND will parse them correctly. If all the checks succeed, reload the BIND daemon with `sudo systemctl restart bind9` and test name resolution with: `host -l <domain> localhost`.

Slave DNS Configuration

Now that the domains are configured correctly on the master machine, we can copy the domain configuration file over to the slave machine and tweak some values.

1. Copy '/etc/bind/named.conf.local' to the slave machine and replace the existing file
2. Open '/etc/bind/named.conf.local' in a text editor and, for each domain:
    a. Change the type from `master` to `slave`
    b. Add `masters { <ip>; };`, where <ip> is the master server's IP address
    c. In the file path, replace `/etc/` with `/var/cache/` to work with AppArmor restrictions

The slave DNS server is now configured to pull its zone data from the master server. Reload the BIND daemon with `sudo systemctl restart bind9` and test name resolution with: `host -l <domain> localhost`. If all goes well, the slave server has pulled the zone data from the master successfully.

Restricting Master Access

The master and slave server are communicating successfully. Now we need to head back to the master server and tweak it so that it will only respond to queries and zone transfers from the slave.

1. Open '/etc/bind/named.conf.options' with a text editor
    a. Append the following text to the end of the file:

       ```
       acl allowed-hosts {
           <slave_ip>
       };
       ```

       Make sure to replace <slave_ip> the IP address of the slave name server.
    b. Append the following text to the `options` block:

       ```
       allow-query { allowed-hosts; };
       allow-transfer { allowed-hosts; };
       ```

       This will prevent any hosts other than those in the `allowed-hosts` list from making DNS queries to the master server.

The master server is now configured to give access exclusively to the slave server. Reload the BIND daemon with `sudo systemctl restart bind9` and test name resolution with: `host -l <domain> <master_ip>` and `host <subdomain>.<domain> <master_ip>`. You should get a response similar to below:

```
$ host -l lanthanideproductions.com.nz 10.1.1.2
Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

Host lanthanideproductions.com.nz not found: 5(REFUSED)
; Transfer failed.

$ host www.lanthanideproductions.com.nz 10.1.1.2
Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

Host www.lanthanideproductions.com.nz not found: 5(REFUSED)
```

However, if you try the same commands with the slave's IP the queries should return successfully:

```
$ host -l lanthanideproductions.com.nz 10.1.1.1
Using domain server:
Name: 10.1.1.1
Address: 10.1.1.1#53
Aliases:

lanthanideproductions.com.nz name server
ns1.lanthanideproductions.com.nz.
lanthanideproductions.com.nz name server
ns2.lanthanideproductions.com.nz.
ftp.lanthanideproductions.com.nz has address 10.1.2.22
mail.lanthanideproductions.com.nz has address 10.1.2.21
ns1.lanthanideproductions.com.nz has address 10.1.1.2
ns2.lanthanideproductions.com.nz has address 10.1.1.1
www.lanthanideproductions.com.nz has address 10.1.2.20

$ host www.lanthanideproductions.com.nz 10.1.1.1
Using domain server:
Name: 10.1.1.1
Address: 10.1.1.1#53
Aliases:
```

```

www.lanthanideproductions.com.nz has address 10.1.2.20
```

If your results look similar to the above, then congratulations! The master and slave servers are configured correctly.

Testing

The following steps are a capstone test to confirm that the servers are set up correctly. Each section will have an expectation, a series of commands, and expected output. If the output does not look similar, then the configuration for that feature should be checked again.

Restricted Master

First, we will make sure that the master server refuses all queries that are made to it from any IP address other than the slave's. On another machine that is *not* either of the DNS servers:

1. Run `host -l <domain> <master_ip>`
2. Run `host <subdomain>.<domain> <master_ip>`

The output should look similar to the following:

```
$ host -l lanthanideproductions.com.nz 10.1.1.2
Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

Host lanthanideproductions.com.nz not found: 5(REFUSED)
; Transfer failed.

$ host www.lanthanideproductions.com.nz 10.1.1.2
Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

Host www.lanthanideproductions.com.nz not found: 5(REFUSED)
```

If not, refer to step 1 in **Restricting Master Access**

Slave Sourcing from Master

To check whether the slave is correctly sourcing its records from the master server, we will increment the serial number for a domain and perform a zone transfer on the slave to see if it has been updated.

On the master machine:

1. Open ‘/etc/bind/db.<domain>’ with a text editor
    a. Increment the serial number by 1
2. Reload BIND with `sudo systemctl restart bind9`

On another machine that is *not* either of the DNS servers:

1. Run `host -t AXFR <domain> <slave_ip>`

The result of the zone transfer should look like the following:

```
$ host -t AXFR monopolistictradingcompany.com.nz 10.1.1.2
Trying "monopolistictradingcompany.com.nz"
Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53395
;; flags: qr aa ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL:
0

;; QUESTION SECTION:
;monopolistictradingcompany.com.nz. IN  AXFR

;; ANSWER SECTION:
monopolistictradingcompany.com.nz.       86400       IN       SOA
     ns.monopolistictradingcompany.com.nz.
root.monopolistictradingcompany.com.nz.  3  604800  86400  2419200
86400
monopolistictradingcompany.com.nz.       86400       IN       NS
     ns1.monopolistictradingcompany.com.nz.
monopolistictradingcompany.com.nz.       86400       IN       NS
     ns2.monopolistictradingcompany.com.nz.
ftp.monopolistictradingcompany.com.nz. 86400 IN   A 10.1.5.22
mail.monopolistictradingcompany.com.nz. 86400 IN A 10.1.5.21
ns1.monopolistictradingcompany.com.nz. 86400 IN   A 10.1.1.2
ns2.monopolistictradingcompany.com.nz. 86400 IN   A 10.1.1.1
www.monopolistictradingcompany.com.nz. 86400 IN   A 10.1.5.20
monopolistictradingcompany.com.nz.       86400       IN       SOA
     ns.monopolistictradingcompany.com.nz.
root.monopolistictradingcompany.com.nz.  3  604800  86400  2419200
86400

Received 260 bytes from 10.1.1.2#53 in 0 ms
```

Note that the serial number in the SOA record has incremented from 2 to 3, proving that the changes had propagated from the master to slave. If not, follow the steps in **Master DNS Configuration** and **Slave DNS Configuration** and see if there is anything that might have been overlooked.

Correct Domain and Subdomain Configuration

The process of ensuring the correct records were implemented is actually included in the previous test, as a zone transfer contains the entirety of a zone within its response.

On any machine:

1. For each domain:
    a. Run `host -l <domain> <slave_ip>`

The result should look similar to the following (a loop was used in the command for convenience):

```
$ for domain in neoimperialexports.com.aq
lanthanideproductions.com.nz unimaginableuniversity.com.ki
monopolisticusedvehicles.com.aq monopolistictradingcompany.com.nz;
do host -l $domain 10.1.1.2; done

Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

neoimperialexports.com.aq name server
ns1.neoimperialexports.com.aq.
neoimperialexports.com.aq name server
ns2.neoimperialexports.com.aq.
ftp.neoimperialexports.com.aq has address 10.1.1.22
mail.neoimperialexports.com.aq has address 10.1.1.21
ns1.neoimperialexports.com.aq has address 10.1.1.2
ns2.neoimperialexports.com.aq has address 10.1.1.1
www.neoimperialexports.com.aq has address 10.1.1.20
Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

lanthanideproductions.com.nz name server
ns1.lanthanideproductions.com.nz.
lanthanideproductions.com.nz name server
ns2.lanthanideproductions.com.nz.
ftp.lanthanideproductions.com.nz has address 10.1.2.22
mail.lanthanideproductions.com.nz has address 10.1.2.21
```

```
ns1.lanthanideproductions.com.nz has address 10.1.1.2
ns2.lanthanideproductions.com.nz has address 10.1.1.1
www.lanthanideproductions.com.nz has address 10.1.2.20
Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

unimaginableuniversity.com.ki name server
ns1.unimaginableuniversity.com.ki.
unimaginableuniversity.com.ki name server
ns2.unimaginableuniversity.com.ki.
ftp.unimaginableuniversity.com.ki has address 10.1.3.22
mail.unimaginableuniversity.com.ki has address 10.1.3.21
ns1.unimaginableuniversity.com.ki has address 10.1.1.2
ns2.unimaginableuniversity.com.ki has address 10.1.1.1
www.unimaginableuniversity.com.ki has address 10.1.3.20
Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

monopolisticusedvehicles.com.aq name server
ns1.monopolisticusedvehicles.com.aq.
monopolisticusedvehicles.com.aq name server
ns2.monopolisticusedvehicles.com.aq.
ftp.monopolisticusedvehicles.com.aq has address 10.1.4.22
mail.monopolisticusedvehicles.com.aq has address 10.1.4.21
ns1.monopolisticusedvehicles.com.aq has address 10.1.1.2
ns2.monopolisticusedvehicles.com.aq has address 10.1.1.1
www.monopolisticusedvehicles.com.aq has address 10.1.4.20
Using domain server:
Name: 10.1.1.2
Address: 10.1.1.2#53
Aliases:

monopolistictradingcompany.com.nz name server
ns1.monopolistictradingcompany.com.nz.
monopolistictradingcompany.com.nz name server
ns2.monopolistictradingcompany.com.nz.
ftp.monopolistictradingcompany.com.nz has address 10.1.5.22
mail.monopolistictradingcompany.com.nz has address 10.1.5.21
ns1.monopolistictradingcompany.com.nz has address 10.1.1.2
ns2.monopolistictradingcompany.com.nz has address 10.1.1.1
www.monopolistictradingcompany.com.nz has address 10.1.5.20
```

If the zone transfer fails, see the previous tests. If the correct records are not present, see step 1-4 in **Master DNS Configuration**.

Bibliography

[1]
S. Kelley, “Dnsmasq - network services for small networks.” [Online]. Available: http://www.thekelleys.org.uk/dnsmasq/doc.html. [Accessed: 25-Jun-2019].

[2]
Internet Systems Consortium, Inc., “Bind 9,” *Internet Systems Consortium*, 2019. [Online]. Available: https://www.isc.org/bind/. [Accessed: 25-Jun-2019].

[3]
S. Kelley, *DNSMASQ(8) Linux User’s Manual*. 2018.

[4]
P. Mockapetris, *Domain names - implementation and specification*. RFC Editor, 1987.