

NAME:SUHAN B REVANKAR

SEC:G

SRN:PES2UG19CS

412

Implementation of a Local DNS Server

DNS (Domain Name System) is the Internet's phone book; it translates hostnames to IP addresses (and vice versa). This translation is through DNS resolution, which happens behind the scene.

The objectives of this lab are to understand:

- DNS and how it works
- Install and set up a DNS server
- Functionality and operations

Lab Setup

DNS Server: 10.2.22.184

User/Client: 10.2.22.195

Note: Use the default IP address provided by PESU LAN.

First Test:

Ping a computer such as www.flipkart.com. Please use Wireshark to show the DNS query triggered by your ping command and DNS response. Describe your observation. (Take a screenshot).

Part 1: Setting Up a Local DNS Server

Task 1: Configure the User Machine

On the client machine 10.2.22.195, we need to use 10.2.22.184 as the local DNS server. This is achieved by changing the resolver configuration file (`/etc/resolv.conf`) of the user machine, so the server 10.2.22.184 is added as the first nameserver entry in the file, i.e., this server will be used as the primary DNS server. Add the following entry to the `/etc/resolvconf/resolv.conf.d/headfile`.

nameserver 10.2.22.184

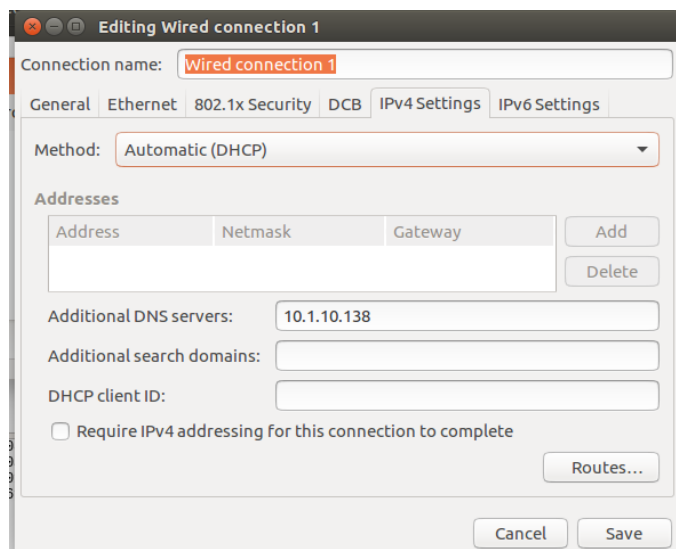
Run the following command for the change to take effect.

sudo resolvconf -u

The following screenshot shows how to set DNS server on the client machine.

```
pesit@SYS-PESU-09:~$ sudo nano /etc/resolvconf/resolv.conf.d/head
pesit@SYS-PESU-09:~$ sudo resolvconf -u
```

Also, add 10.2.22.184 in 'Additional DNS servers' field in IPv4 settings of client



Task 2: Set Up a Local DNS Server

Note: If bind9 server is not already installed, install using the command

```
$ sudo apt-get update
```

```
$ sudo apt-get install bind9
```

```
pesse@SYS-07:~$ sudo apt-get install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
bind9 is already the newest version (1:9.10.3.dfsg.P4-8ubuntu1.17).
0 upgraded, 0 newly installed, 0 to remove and 294 not upgraded.
pesse@SYS-07:~$ sudo nano /etc/bind/named.conf.options
```

Step 1: Configure the BIND9 Server.

BIND9 gets its configuration from a file called `/etc/bind/named.conf`. This file is the primary configuration file, and it usually contains several “include” entries. One of the included files is called `/etc/bind/named.conf.options`. This is where we typically set up the configuration options. Let us first set up an option related to DNS cache by adding a dump-file entry to the options block. The above option specifies where the cache content should be dumped to if BIND is asked to dump its cache.

```
SYS-07: ~
GNU nano 2.5.3 File: /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    dump-file "/var/cache/bind/dump.db";
}
```

The above option specifies where the cache content should be dumped to if BIND is asked to dump its cache. If this option is not specified, BIND dumps the cache to a default file called `/var/cache/bind/named_dump.db`.

Step 2: Start DNS server

```
service bind9 restart
```

```
pesse@SYS-07:~$ sudo service bind9 restart
```

The two commands shown below are related to DNS cache.

The first command dumps the content of the cache to the file specified above, and the second command clears the cache.

```
pesse@SYS-07:~$ sudo rndc dumpdb -cache
pesse@SYS-07:~$ sudo rndc flush
pesse@SYS-07:~$
```

Task 3: Host a Zone in the Local DNS server.

Assume that we own a domain, we will be responsible for providing the definitive answer regarding this domain. We will use our local DNS server as the authoritative nameserver for the domain. In this lab, we will set up an authoritative server for the **example.com** domain.

This domain name is reserved for use in documentation, and is not owned by anybody, so it is safe to use it.

Step 1: Create Zones

We had two zone entries in the DNS server by adding the following contents to **/etc/bind/named.conf** as shown in the below screenshot. The first zone is for forward lookup (from hostname to IP), and the second zone is for reverse lookup (from IP to hostname).

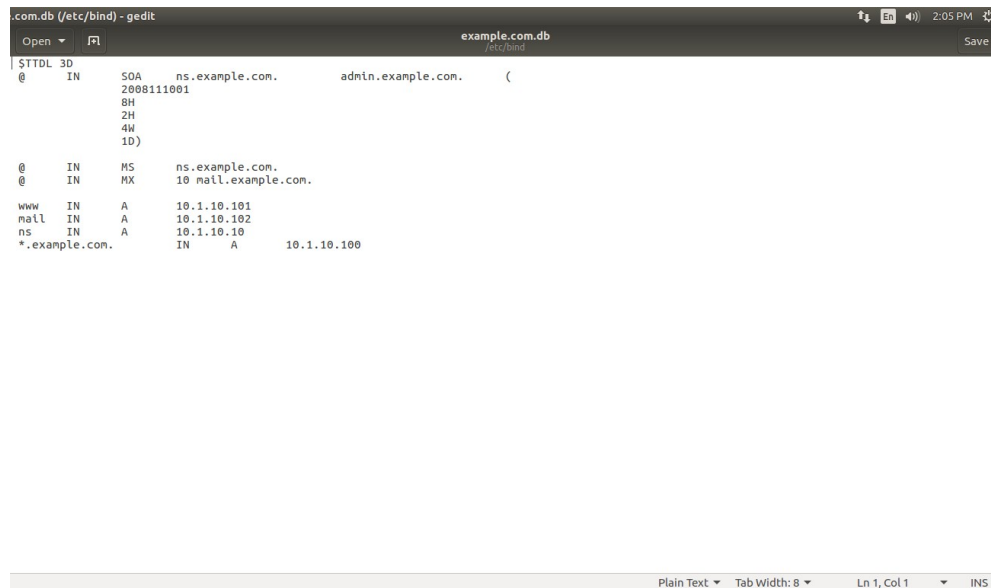
```
pesse@SYS-07:~$ sudo nano /etc/bind/named.conf
pesse@SYS-07:~$ sudo cat /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

zone "example.com" {
    type master;
    file "/etc/bind/example.com.db";
};

zone "10.1.10.in-addr.arpa" {
    type master;
    file "/etc/bind/10.1.10.db";
};
pesse@SYS-07:~$
```

Step 2: Setup the forward lookup zone file

We create **example.com.db** zone file with the following contents in the **/etc/bind/** directory where the actual DNS resolution is stored.



```
.com.db (/etc/bind) - gedit
example.com.db
/etC/bind
Save

$TTL 30
@      IN      SOA     ns.example.com.  admin.example.com.  (
                        2008111001
                        8H
                        2H
                        4W
                        10)

@      IN      NS      ns.example.com.
@      IN      MX      10 mail.example.com.

www    IN      A       10.1.10.101
mail   IN      A       10.1.10.102
ns     IN      A       10.1.10.10
*.example.com.  IN      A       10.1.10.100
```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

The symbol '@' is a special notation representing the origin specified in **named.conf** (the string after "**zone**"). Therefore, '@' here stands for **example.com**. This zone file contains 7 resource records (RRs), including a SOA (Start Of Authority) RR, a NS (Name Server) RR, a MX (Mail eXchanger) RR, and 4 A (host Address) RRs.

Step 3: Setup the reverse lookup zone file

We create a reverse DNS lookup file called **10.2.22.db** for the example.net domain to support DNS reverse lookup, i.e., from IP address to hostname in the **/etc/bind/** directory with the following contents.

```
.db (/etc/bind) - gedit
Open 10.1.10.db
Save

$TTL 3D
@ IN SOA ns.example.com. admin.example.com. (
      2008111001
      8H
      2H
      4W
      1D)

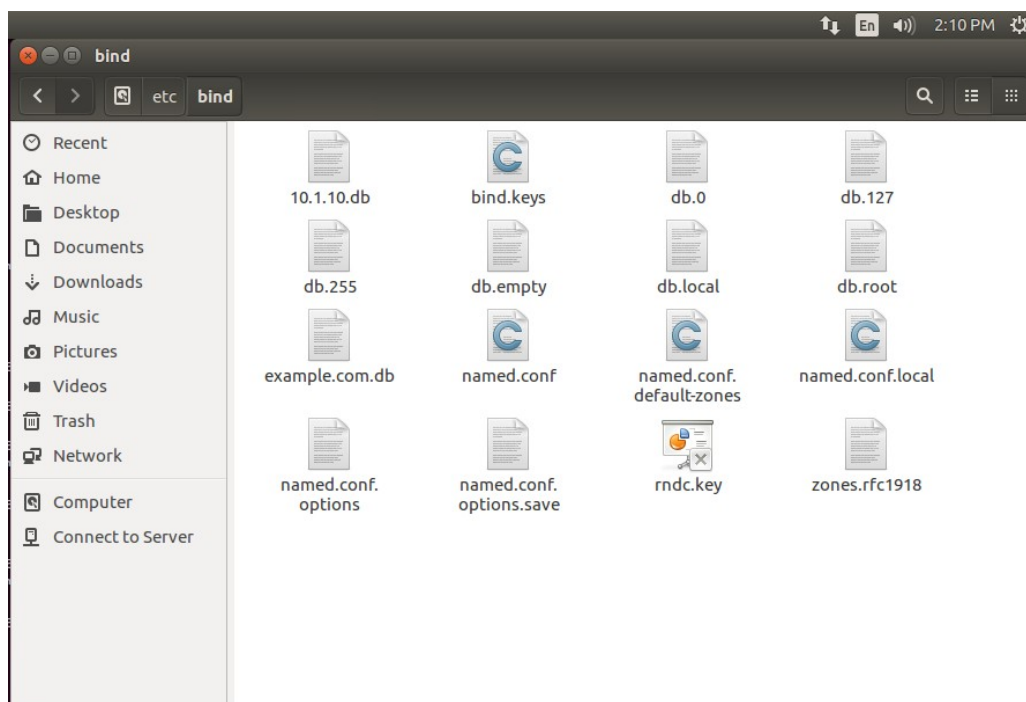
@ IN NS ns.example.com.

101 IN PTR www.example.com.
102 IN PTR mail.example.com.
103 IN PTR ns.example.com.
```

Step 4: Copy the above files into /etc/bind location.

```
pess@SYS-07:/etc/bind$ sudo cp 10.1.10.db /etc/bind
cp: '10.1.10.db' and '/etc/bind/10.1.10.db' are the same file
```

NOTE: The files were already saved in /etc/location.



Files present in left hand side.

Task 4: Restart the BIND server and test

Step 1: When all the changes are made, remember to restart the BIND server. Now we will restart the DNS server using the following command:

\$ sudo service bind9 restart

```
SYS-07: ~  
pesse@SYS-07:~$ sudo service bind9 restart
```

Step 2: Now, go back to the client machine and ask the local DNS server for the IP address of www.example.com using the dig command.

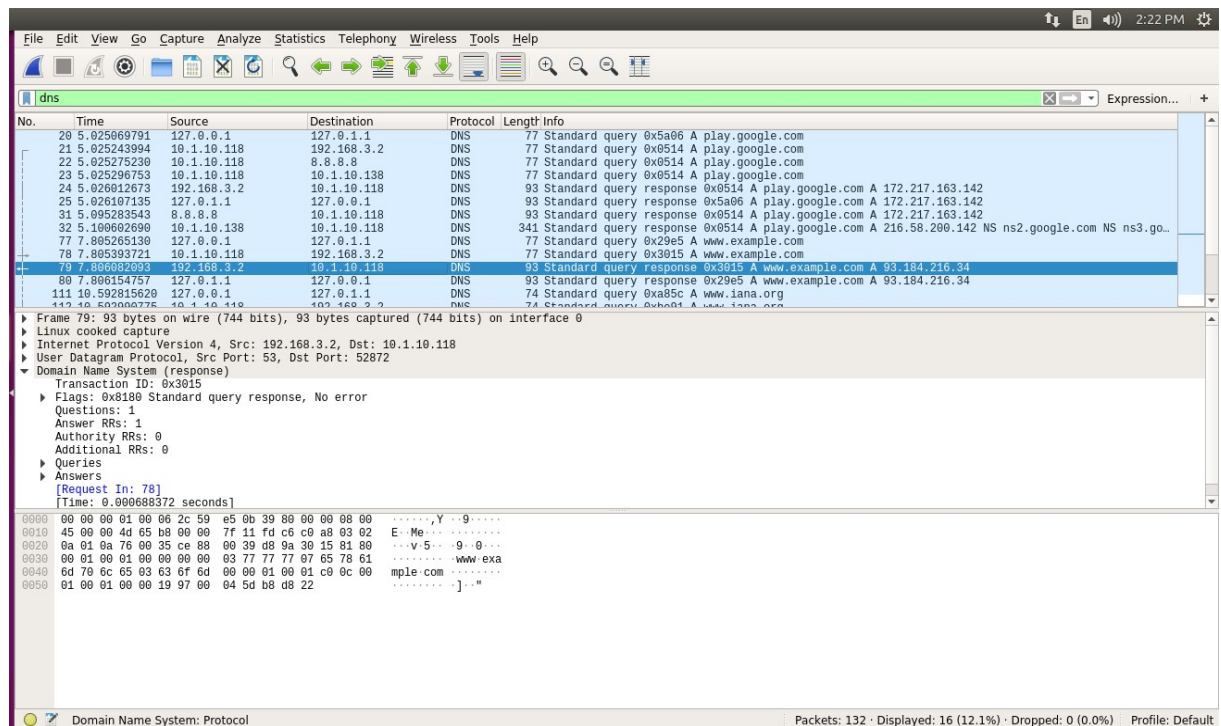
Dig stands for (Domain Information Groper) is a network administration command-line tool for querying DNS name servers. It is useful for verifying and troubleshooting DNS problems

and also to perform DNS lookups and displays the answers that are returned from the name server that were queried. dig is part of the BIND domain name server software suite.

```
pesit@SYS-PESU-09:~$ dig www.example.com  
  
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 16562  
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
;; EDNS: version: 0, flags:; udp: 4096  
;; QUESTION SECTION:  
;www.example.com.          IN      A  
  
;; Query time: 0 msec  
;; SERVER: 10.1.10.138#53(10.1.10.138)  
;; WHEN: Fri Feb 19 14:13:53 IST 2021  
;; MSG SIZE rcvd: 44  
  
pesit@SYS-PESU-09:~$
```

We can see that the ANSWER SECTION contains the DNS mapping. We can see that the IP address of www.example.com is now 10.2.22.101, which is what we have setup in the DNS server.

Step 3: Observe the results in Wireshark capture.



To load and clear DNS cache, use the below commands.

```
pesse@SYS-07:~$ sudo rndc dumpdb -cache
pesse@SYS-07:~$ sudo rndc flush
```

Edmodo Requirements:

- 1) Three Wireshark packet capture screenshots for ping (Packet list pane and Packet details pane) – **ping www.flipkart.com** command
- 2) **dig www.example.com** command (in Terminal)
- 3) Wireshark packet capture – **dig www.example.com** command (Packet list pane and Packet details pane)
- 4) Local DNS cache on server machine