SECTION 1 Configure a DNS Server Using BIND

Computers communicate with each other using their IP addresses. But for humans, it is simpler to address a computer using its name.

DNS (Domain Name System) is a distributed database system that guarantees unique computer names worldwide. It provides computers with IP addresses when a user enters a computer name, and vise versa.

In this section, you learn some advanced techniques of configuring BIND.

Objectives

Objective 1 Retrospection of DNS Basics

In ***SUSE LINUX Advanced Administration*** (Course 3038), you learned basics about configuring BIND on the DNS server.

We want to summarize the basics here:

- Understand How Name Servers Work
- Understand How to Query DNS
- Configure a Caching-Only DNS Server
- Configure a Master Server for Your Domain

Understand How Name Servers Work

Domains are administered locally instead of using a global authority. Each domain has its own administration point (in practice, many domains are administered from one location).

For each domain there is one DNS server (or name server) defined as being "in charge" of its domain. This server is known as the ***master server***, and it is the authority for this domain (providing authoritative answers).

This authoritative information is important because DNS servers also temporarily store information on other domains in a cache and can pass this information on, with the note that it is a non-authoritative answer.

There are other DNS servers called ***slave servers*** for the domain that distribute the load and serve as backups. Slave servers keep a copy of the information on the master server and update this information at regular intervals. This update is called ***zone transfer***.

The following describe the DNS server types available:

Table 1-1

master server	Has the main responsibility for a domain. Gets its data from local files.
slave server	Gets its data from the master server using zone transfer.
caching-only server	Queries data from other DNS servers and stores the information in the cache until its expiration date. All replies are nonauthoritative.
forwarding server	All queries the server cannot answer authoritatively are forwarded to other DNS servers.

Understand How to Query DNS

Various programs are involved in processing a request to the DNS database. The first is the resolver. This is a set of library routines used by various programs.

The resolver makes a request to a DNS server, interprets the answer (real information or error message), and sends back this information to the program that called it up.

If the DNS server receives a request from a resolver, one of 2 things happens:

- If the DNS server is the authority for the requested domain, the DNS server provides the required information to the resolver (the authoritative answer).

  or

- If the DNS server is not the authority for the required domain, the DNS server queries the responsible authority for the request domain and gives the result to the resolver.

  The data is stored in the cache of the DNS server. If there is another request for this data later, the DNS server can provide it immediately (a non-authoritative answer). All data has a timestamp, and information is deleted from the cache after a certain time.

Assume that your DNS server wants to find the IP address of the computer www.suse.de. To do this, the DNS server first makes a request to one of the DNS servers of the root domain.

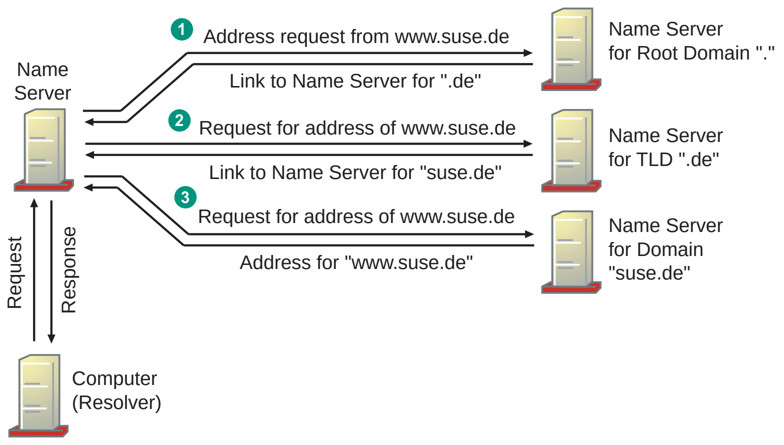
Each DNS server knows the authorities responsible for the TLDs. The address for each authority required is passed onto the requesting DNS server. For www.suse.de, this is a DNS server for the TLD .de, that is, the computer dns2.denic.de.

Our DNS server then asks this for the authority for the domain suse.de and as an answer is given the computer ns.suse.de.

In a third step, this DNS server is queried and (as an answer) gives the IP address of the SUSE web server. This answer is returned by our DNS server to the requesting resolver.

This procedure is illustrated in the following figure:

Figure 1-1


1 Address request from www.suse.de
Link to Name Server for ".de"
Name Server for Root Domain "."
Name Server
2 Request for address of www.suse.de
Link to Name Server for "suse.de"
Name Server for TLD ".de"
3 Request for address of www.suse.de
Address for "www.suse.de"
Name Server for Domain "suse.de"
Request
Response
Computer (Resolver)

The DNS servers for the root domain play a very important role in name resolution. In order to alleviate the server load due to queries, every DNS server stores the information received from other names servers in its cache.

When queries are made, this information is sent without querying the root DNS server anew. However, root DNS servers are very busy despite this caching mechanism. Several thousand queries per second are nothing unusual.

Configure a Caching-Only DNS Server

A caching-only DNS server does not manage its own databases but merely accepts queries and forwards them to other DNS servers. The supplied replies are saved in the cache.

A caching-only DNS server can be used on a workstation or a gateway that has access to an external DNS server.

The DNS server configuration is defined in the file /etc/named.conf. You can use the example file that is installed with the DNS package as a configuration file for a caching-only server.

The following example shows a simple configuration:

```
DA1:~ # ip address show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
#
# /etc/named.conf: Configuration of the name server (BIND9)
#
# Global options
#
options
{
#
# In which directory are the database files?
#
        directory "/var/lib/named";
};
```

The global options are defined in the options block at the beginning of the file. The directory containing the database files (or zone files) is listed. Normally, this is /var/lib/named/.

All filenames that follow the /var/lib/named directory refer to the directory. The directory is created when installing the server package. It contains several preconfigured files. Other options can also be defined in this file.

The Global options are followed by the definition of the database files for the domains managed by the DNS server. Several entries are needed for basic DNS server functions such as those provided by a caching-only server.

Three entries are needed for every DNS server:

- The entry for root DNS servers (not needed for BIND 9 because it has the list of root DNS servers compiled into the software).
- The forward resolution for localhost
- The reverse resolution for the network 127.0.0.0 (localhost)

The following are examples of these entries:

```
## entry for root nameservers#
zone "." in {        type hint;
        file "root.hint";
};

#
# forward resolution for localhost
#
zone "localhost" in {
        type master;
        file "localhost.zone";
};

#
# reverse resolution for localhost
#
zone "0.0.127.in-addr.arpa" in {
        type master;
        file "127.0.0.zone";
};
```

The zone entry for the root DNS servers contains a reference to a file containing the addresses of the root DNS servers. This file (root.hint) is generated in the directory /var/lib/named/ during the installation of the package bind.

The 2 files for the resolution of localhost are also generated during the installation. The structure of these files is explained later.

These entries are used to forward queries to the DNS server directly to the responsible DNS servers. However, this resolution method can be very slow. This problem can be solved by using forwarders.

The DNS server has the addresses of other DNS servers in case it cannot resolve a host name itself. You might be able to use the DNS servers of an Internet provider for this purpose, as they usually have a lot of information in their cache.

You can define these DNS servers in the options block in the file /etc/named.conf, as in the following:

```
options
{
        directory "/var/lib/named";

        forwarders
        {
                10.0.0.254;
        };
};
```

You can enter up to 3 DNS server addresses. Queries that cannot be resolved by the local DNS server are forwarded to one of the specified DNS servers.

If these DNS servers cannot be reached, the queries are sent directly to the root DNS servers.

Configure a Master Server for Your Domain

The following are the tasks you need to do to configure a master DNS server for your domain:

- Adapt the Main Server Configuration File
- Create the Zone Files
- Create Additional Resource Records

Adapt the Main Server Configuration File

You can adapt the configuration for the caching-only DNS server for configuring a DNS server containing its own information files.

This configuration already contains the global entries for the directory and the forwarders (which can be omitted) entries in the options block. The file also contains the mandatory entries for the root servers and the resolution of localhost.

The global options are followed by definitions for the database files (or zone files) for the domains this DNS server serves. At least 2 files are necessary for each domain:

- A file for forward resolution (allocating an IP address to a computer name)
- A file for reverse resolution (allocating a computer name to an IP address)

If several subnets belong to a domain, then one file for each of these networks must be created for reverse resolution.

Each definition begins with the instruction ***zone*** (this is why the database files are also known as zone files), followed by the name of this zone.

For forward resolution, this is always the domain name. For reverse resolution, the network prefix of the IP address must be given in reverse order (10.0.0.0 becomes 0.0.10.) to which the suffix in-addr.arpa is added (0.0.10.in-addr.arpa).

The zone name is always followed by an “in” for Internet. (DNS servers can administer information on different name spaces, not only that of the Internet. Other name spaces are practically never used).

The text in curly brackets defines the type of DNS server this is for the corresponding zone (here it is always the type master; other types are introduced later).

Finally, there is the name of the file in which the entries for this zone are located.

The entries for the Digital Airlines configuration look like the following:

```
#
# forward resolution for the domain digitalairlines.com
#
zone "digitalairlines.com" in
{
        type master;
        file "master/digitalairlines.com.zone";
};
#
# reverse resolution for the network 10.0.0.0
#
zone "0.0.10.in-addr.arpa" in
{
        type master;
        file "master/10.0.0.zone";
};
```

Create the Zone Files

The 2 files for the domain localhost and the file for the root DNS servers are always included in the installation. You do not need to change these files; however, you must create the files required for the actual domain.

The subdirectory /var/lib/named/master/ is used for the database files of a master server.

You need to know the following to manually create the zone files:

- Structure of the Files
- The File /var/lib/named/master/digitalairlines.com.zone
- The File /var/lib/named/master/10.0.0.zone
- The File /var/lib/named/master/localhost.zone
- The File /var/lib/named/master/127.0.0.zone

In these files, the semicolon is used as a comment sign.

Structure of the Files

Each of the database files consists of a series of entries, or resource records. The syntax of these records is always as follows:

reference [TTL] class type value

The following describes each part of a record:

- **reference.** The reference to which the record refers. This can be a domain (or subdomain) or a standalone computer (name or IP address).

- **TTL.** The Time To Live value for the record. If this expires, a default TTL value is used.
- **class.** The class of the record. For TCP/IP networks, this is always IN (internet).
- **type.** The type of the record. The most important types are listed in the table below.
- **value.** The value of the record. The value depends on the type of record as listed in the following:

Table 1-2

Record Type	Meaning	Value
SOA	Start of Authority (term for the authority)	Parameter for the domain
NS	DNS server	Name of one of the DNS servers for this domain
MX	Mail exchanger	Name and priority of a mail server for this domain
A	Address	IP address of a computer
PTR	Pointer	Name of a computer
CNAME	Canonical name	Alias name for a computer

Individual entries must always start in the first column with the reference. If an entry does not start in the first column, the reference is taken from the previous entry.

The File /var/lib/named/master/digitalairlines.com.zone

Unlike earlier versions of BIND, BIND 9 requires you to specify a default TTL for all information at the beginning. This value is used whenever the TTL has not been explicitly given for an entry.

You define the TTL with the following instruction:

```
;
; definition of a standard time to live, here: two days
;
$TTL 172800
```

In this example, the TTL is given in seconds. But it can be given in other units, such as 2D for two days. Other units are M (minutes), H (hours), and W (weeks).

This is followed by the definition of the SOA (Source of Authority) entry, which specifies which DNS server has the authority for this domain:

```
;
; SOA Entry
;
digitalairlines.com. IN SOA da1.digitalairlines.com.
adm.digitalairlines.com. (
                      2004092601 ; serial number
                      1D         ; refresh (one day)
                      2H         ; retry (two hours)
                      1W         ; expiry time (one week)
                      3H         ; "negative" validity (three hours)
                      )
```

The domain to which this entry refers (in the example, **digitalairlines.com**) is listed first. The domain name must end with a dot. If a name does not have a dot at the end, the name of the domain is added on, which could lead to an error here.

After the SOA entry the name of the DNS server is listed (in this example, **da1.digitalairlines.com** with a dot at the end). Alternatively, you could write da1, and the domain name digitalairlines.com would be added after the name.

Next comes the email address of the person who is responsible for the administration of the DNS server. The “@” usually used in email addresses must be replaced by a dot (so the email address in this example is ***hostmaster.example.com***). This is necessary because @ has a special meaning as an abbreviation.

After this information, there is a serial number. Any number can be used, but normally the date and a version number are used here. After any change to the data in this file, the serial number has to be increased.

Slave servers use this number to detect if they need to copy this zone file or not. If the serial number on the master server is greater than that on the slave server, the file is copied.

This is followed by the following time information (the first three entries listed here are only important for slave servers):

- The first entry causes a slave server to query a master server after this length of time, to see if there is a new version of the files (in the example, this is 1D or one day).
- If the slave server cannot reach the master server, the next time entry specifies at what intervals new attempts should be made (in the example, this is 2H or two hours).
- If the master server is not reached for a longer period of time, the first time entry specifies when the slave server should discard its information on this zone (in the example, this is 1W or a week).

  The basic idea here is that it is better not to pass on any information than to pass on outdated information.

- The fourth entry defines for how long negative responses from the DNS server are valid. Each requesting server stores responses in its cache, even if a computer name could not be resolved (in the example, this is 3H or 3 hours).

These time definitions are followed by the name of the computer that is responsible for this domain as the DNS server. In all cases, the master server must be entered here. If slave servers are used, they should also be entered, as in the following:

```
;
; entry for the name server
;
digitalairlines.com.       IN NS    da1.digitalairlines.com.
```

The name of the domain can be omitted at this point. Then the name from the previous entry is taken (the SOA entry).

At the end of this file are the IP addresses that are allocated to computer names. This is done with A (address) entries, as in the following:

```
;
; Allocation of IP addresses to host names
;
da10             IN A     10.0.0.10
da12             IN A     10.0.0.12
da13             IN A     10.0.0.13
```

The File /var/lib/named/master/10.0.0.zone

The file for reverse resolution contains similar entries as the file for forward resolution. At the beginning of the file there is the definition of a default TTL and an SOA entry.

In the SOA and NS entries, the IP address of the network is written in reverse order:

```
; Database file for the domain digitalairlines.com:
; reverse resolution for the network
; 10.0.0.0
;
; Definition  of a default TTL,here: two days
;
$TTL 172800
;
; SOA entry
;
0.0.10.in-addr.arpa. IN SOA da1.digitalairlines.com.
adm.digitalairlines.com. (
                        2004092601 ; serial number
                        1D         ; refresh (one day)
                        2H         ; retry (two hours)
                        1W         ; expiry time (one week)
                        3H         ; "negative" validity(three hours)
                        )
;; Entry for the name server
;
                        IN NS    da1.digitalairlines.com.
```

At the end of this file are the IP addresses that are allocated to computer names, this time with the PTR (Pointer) entry, as in the following:

```
;
; Allocation of host names to IP addresses
;
10                   IN PTR  da10.digitalairlines.com.
12                   IN PTR  da12.digitalairlines.com.
13                   IN PTR  da13.digitalairlines.com.
14                   IN PTR  da14.digitalairlines.com.
```

The following 2 files must exist for the local computer. These are created automatically during installation and should not be modified.

The File /var/lib/named/master/localhost.zone

The following is an example of the file /var/lib/named/master/localhost.zone:

```
$TTL 1W
@        IN SOA       @              root (
                      42             ; serial (d. adams)
                      2D             ; refresh
                      4H             ; retry
                      6W             ; expiry
                      1W )           ; minimum

         IN NS        @
         IN A         127.0.0.1
```

In this example, the “@” character is used as an abbreviation (for this reason, it must be replaced by a dot in the email address in the database files).

Using “@” instead of the domain name causes the file /etc/named.conf to be read to see for which domain this file is responsible.

In this case, it is localhost, which is also used for the name of the DNS server (this is why “@” appears many times in the file).

The File /var/lib/named/master/127.0.0.zone

In this file, the abbreviation “@” is also used. But here the computer name must be given explicitly with localhost (remember the dot at the end):

```
$TTL 1W
@        IN SOA        localhost.     root.localhost. (
                       42        ; serial (d. adams)
                       2D        ; refresh
                       4H        ; retry
                       6W        ; expiry
                       1W )      ; minimum

         IN NS         localhost.
1        IN PTR        localhost.
```

Create Additional Resource Records

Apart from the resource records already discussed (SOA, NS, A, PTR), there are MX and CNAME resource records, which are used to do the following:

- Define Mail Servers for the Domain
- Assign Aliases for Computers

Define Mail Servers for the Domain

To be able to use email addresses in the form geeko@digitalairlines.com, the email server responsible for the domain must be defined (the email cannot be sent directly to the domain, but must be sent to a mail server).

To achieve this, an MX (Mail Exchange) entry must be made in the database file for forward resolution, after the DNS server entry:

```
digitalairlines.com.        IN MX    0 mail
                            IN MX   10 da1
                            IN MX   10 da5
```

If an email is now sent to the address geeko@digitalairlines.com, the computer sending the mail asks the DNS server which computer is the mail server, and is sent the list of the MX entries in return.

Several mail servers can be given. On the basis of their priorities, it is then decided to which computer the email is sent. The priority of mail servers is defined by the number in front of the computer name; the lower this number, the higher the priority.

In this example the computer mail.digitalairlines.com has the highest priority (is therefore the primary mail server).da1.digitalairlines.com and da5.digitalairlines.com both have the same priority.

If the mail server with the highest priority cannot be reached, the mail server with the second highest priority is used. If several mail servers have the same priority, then one of them is chosen at random. An address entry must be made for each mail server.

Assign Aliases for Computers

If you want a computer to be reached by more than one name (such as addressing a computer as da30.digitalairlines.com and www.digitalairlines.com), then corresponding aliases must be given.

These are the CNAME (canonical name) entries in the database file for forward resolution:

```
da30                 IN A       10.0.0.30
www                  IN CNAME   da30
```

The names of the mail servers for the domain (MX entry) cannot be alias names, since some mail servers cannot handle this correctly.

Objective 2 Forward Requests to Other Name Servers

If the name server does not have any information on the computer to be resolved, it tries to fetch this information from another name server.

To do this, the name server has to know how to reach other name servers:

- The Name Servers of the Root Domain
- Forward Requests to Specific Name Servers
- Forward Requests for a Subdomain

The Name Servers of the Root Domain

In the configuration file /etc/named.conf, there should be a zone entry containing a reference to a file where the addresses of the root name servers are located.

This file is created during the installation of the bind package in the directory /var/lib/named/, and is called root.hint.

The zone entry in the file /etc/named.conf looks like this:

```
zone "." in
{
    type hint;
    file "root.hint";
};
```

You must include the **type hint** entry.

Forward Requests to Specific Name Servers

If no additional entries are made, the name server can contact the root name servers directly.

But it is also possible to give the name server the addresses of other name servers.

It will use these first if it cannot resolve a computer name itself.

The definition of such a name server is specified in the **options** section of the file /etc/named.conf:

```
options
{
    directory "/var/lib/named";
    forwarders
    {
         10.0.0.10;
         10.0.0.15;
    };
};
```

In this case, the requests that could not be resolved locally are forwarded to one of the two name servers given.

If neither of these can be reached, the request runs via the root name servers, as described above.

Forward Requests for a Subdomain

If there is another name server used for resolving host names for a subdomain, this name server has to be defined in the corresponding zone entry.

The type is set to forward, the address of the respective name server has to be specified:

```
zone "slc.digitalairlines.com" in
{
    type forward;
    forward only;
    forwarders
    {
         10.0.0.10;
    };
};
```

If you include the entry **forward only**, the name server is instructed never to try to resolve the address itself.

All requests concerning the subdomain slc.digitalairlines.com are forwarded to the name server with the address 10.0.0.10.

A corresponding instruction for reverse resolution must be defined.

Exercise 1-1 DNS Server with Forwarding

In this exercise, you work with a partner to configure a DNS master server and a DNS slave server for the domain digitalairlines.com.

You need to work as a team on all parts of the exercise.

Do the following:

- Part I: Install BIND
- Part II: Configure the DNS Master Server for the Domain digitalairlines.com
- Part III: Configure the DNS Slave Server for the Domain digitalairlines.com
- Part IV: Configure the DNS Master Server for the Domain muc.digitalairlines.com
- Part V: Enable Forwarding

This exercise requires extensive typing to create your DNS files. To save you some time, the files **digitalairlines.com.zone** and **10.0.0.zone** are included on your ***3057 Course CD*** in the directory **exercises/section_1/**.

Part I: Install BIND

To install BIND, do the following on both SUSE LINUX Enterprise Server 9 servers:

1. From the KDE menu, select **System > YaST**.
2. Enter the root password (**novell**) and select **OK**.
3. From the YaST Control Center, select

   Software > Install and Remove Software
4. From the **Filter** drop-down menu, select **Search**.
5. In the **Search** field, enter **bind**; then select **Search**.
6. On the right, select the **bind** package.
7. Select **Accept**; then insert the requested SLES 9 CD.
8. When installation is complete, remove the CD and close the YaST Control Center.

Part II: Configure the DNS Master Server for the Domain digitalairlines.com

Do the following to configure a DNS master server:

1. Open a terminal window and enter **su -** to get root permissions.
2. When prompted, enter the root password **novell**.
3. To rename the file **/etc/named.conf** to **/etc/named.conf.orig**, enter

   mv /etc/named.conf /etc/named.conf.orig

4. Create an new configuration file named /etc/named.conf with the appropriate settings.

   (You can also copy the file from the ***Course*** CD).

5. Configure the **forwarders** line to match the following:

   ```
   forwarders {10.0.0.254;};
   ```

   Make sure that you delete the comment character from the beginning of the forwarders line.

6. Add the following two zone statements after the existing zone statements:

   ```
   zone “digitalairlines.com” in {
           type master;
           file “master/digitalairlines.com.zone”;
   };

   zone “0.0.10.in-addr.arpa” in {
           type master;
           file “master/10.0.0.zone”;
   };
   ```

7. Save and close the file.

8. Create a new file **digitalairlines.com.zone** in the directory **/var/lib/named/master/**.

9. Enter the following zone configuration in the file:

   ```
   $TTL 172800

   digitalairlines.com. IN SOA your_FQDN. hostmaster.digitalairlines.com. (
                                           serial_number
                                           1D
                                           2H
                                           1W
                                           3H
                                           )

   digitalairlines.com. IN NS your_FQDN.
   digitalairlines.com. IN NS slave_FQDN.

   da1             IN A 10.0.0.254
   da2             IN A 10.0.0.2
   da10              IN A 10.0.0.10
   da11              IN A 10.0.0.11
   da12              IN A 10.0.0.12
   ```

   The SOA record (including hostmaster.digitalairlines.com) ***must*** be on a single line.

   Make sure you enter your FQDN (such as **da50.digitalairlines.com**) in the SOA and NS records.

   Use the current date and “01” as the serial number (such as **2005071501**).

Add an A record for your own host, such as
da50 IN A 10.0.0.50

10. Save and close the file.
11. Create a new file **10.0.0.zone** in the directory **/var/lib/named/master/**.
12. Enter the following zone configuration in the file:

```
$TTL 172800
0.0.10.in-addr.arpa. IN SOA your_FQDN.
hostmaster.digitalairlines.com. (
                              serial_number
                              1D
                              2H
                              1W
                              3H
                              )

0.0.10.in-addr.arpa.          IN NS your_FQDN.
0.0.10.in-addr.arpa.          IN NS slave_FQDN.

254         IN PTR da1.digitalairlines.com.
2           IN PTR da2.digitalairlines.com.
10          IN PTR da10.digitalairlines.com.
11          IN PTR da11.digitalairlines.com.
12          IN PTR da12.digitalairlines.com.
```

The SOA record (including hostmaster.digitalairlines.com) ***must*** be on a single line.

Make sure you enter your FQDN (such as da50.digitairlines.com) in the SOA and NS records.

Use the current date and “01” as the serial number (such as **2005071501**).

Add a PTR record for your own host, such as
50 IN PTR da50.digitalairlines.com.

13. Save and close the file.
14. Open a second terminal window and enter **su -** to get root permissions.
15. When prompted, enter the root password **novell**.
16. Enter the command

```
tail -f /var/log/messages
```

17. Switch to the first terminal window and start bind by entering

```
rcnamed start
```

If there are errors in the file /etc/named.conf, they are noted in the output (with specific references and line numbers).
The named daemon will not start until these errors are fixed.

18. From the second terminal window, watch the log output of bind for any messages such as **Unknown RR Type** or **File Not Found**.

    If any errors occur, fix them and restart bind.

19. From the first terminal window, start bind automatically when the system is booted by entering

    insserv named

20. Open the file **/etc/resolv.conf** in a text editor.
21. Delete all existing **nameserver** entries.
22. Add the following entry:

    nameserver *your_ip_address*

23. Save and close the file.
24. Verify that your DNS server works by entering

    host da10.digitalairlines.com

    This should display the IP address of 10.0.0.10.

Part III: Configure the DNS Slave Server for the Domain digitalairlines.com

To configure the DNS slave server, do the following on the DNS slave server:

1. Open a terminal window and and enter **su -** to get root permissions.
2. When prompted, enter the root password **novell**.
3. To rename the file **/etc/named.conf** to **/etc/named.conf.orig**, enter

    mv /etc/named.conf /etc/named.conf.orig

4. Create an new configuration file named /etc/named.conf with the appropriate settings.

    (You can also copy the file from the ***Course*** CD).

5. Configure the **forwarders** line to match the following:

    forwarders {10.0.0.254;};

    Make sure that you delete the comment character from the beginning of the forwarders line.

6. Enter the following two zone statements after the existing statements:

```
zone “digitalairlines.com” in {
    type slave;
    file “slave/digitalairlines.com.zone”;
    masters
    {
        master_server_ip_address;
    };
};
```

```
zone “0.0.10.in-addr.arpa” in {
        type slave;
        file “slave/10.0.0.zone”;
        masters
        {
                master_server_ip_address;
        };
};
```

7. Save the changes and close the editor.
8. Open a second terminal window and enter **su -** to get root permissions.
9. When prompted, enter the root password **novell**..
10. Enter the command

    tail -f /var/log/messages

11. Switch to the first terminal window and start bind by entering

    rcnamed start

> If there are errors in the file /etc/named.conf, they are noted in the output (with specific references and line numbers).
> The named daemon will not start until these errors are fixed.

12. From the second terminal window, watch the log output of bind for any messages such as **Unknown RR Type** or **File Not Found**.
13. If any errors occur, try to fix them and restart bind.
14. Start bind automatically when the system boots by entering

    insserv named

15. From the first terminal window, open the **/etc/resolv.conf** file in a text editor.
16. Delete all existing **nameserver** entries.
17. Add the following entry:

    nameserver *master_server_ip_address*

18. Save and close the file.
19. Verify that your DNS server works by entering

    host da10.digitalairlines.com

Part IV: Configure the DNS Master Server for the Domain muc.digitalairlines.com

To configure the DNS Master Server for the Domain muc.digitalairlines.com, do the following on the slave server:

1. Open a terminal window and enter **su -** to get root permissions.
2. When prompted, enter the root password **novell**.
3. To stop the DNS server, enter

   ```
   rcnamed stop
   ```

4. Open the file **/etc/named.conf** with your favorite editor.
5. Add the following two zone statements after the existing zone statements:

   ```
   zone “muc.digitalairlines.com” in {
           type master;
           file “master/muc.digitalairlines.com.zone”;
   };

   zone “1.0.10.in-addr.arpa” in {
           type master;
           file “master/10.0.1.zone”;
   };
   ```

6. Save and close the file.
7. Create a new file **muc.digitalairlines.com.zone** in the directory **/var/lib/named/master/**.
8. Enter the following zone configuration in the file:

   ```
   $TTL 172800

   muc.digitalairlines.com. IN SOA your_FQDN.
   hostmaster.digitalairlines.com. (
                                       serial_number
                                       1D
                                       2H
                                       1W
                                       3H
                                       )

   muc.digitalairlines.com. IN NS your_FQDN.
   muc.digitalairlines.com. IN MX 1 da1.digitalairlines.com.

   da100          IN A 10.0.1.100
   da101          IN A 10.0.1.101
   da102          IN A 10.0.1.102
   ```

   The SOA record (including hostmaster.digitalairlines.com) ***must*** be on a single line.

   Make sure you enter your FQDN (such as da50.digitalairlines.com) in the SOA and NS records.

Use the current date and "01" as the serial number (such as **2005071501**).

9. Save and close the file.
10. Create a new file **10.0.1.zone** in the directory **/var/lib/named/master/**.
11. Enter the following zone configuration in the file:

```
$TTL 172800
1.0.10.in-addr.arpa. IN SOA your_FQDN.
hostmaster.digitalairlines.com. (
                                        serial_number
                                        1D
                                        2H
                                        1W
                                        3H
                                        )

                IN NS your_FQDN.



100             IN PTR da100.muc.digitalairlines.com.
101             IN PTR da101.muc.digitalairlines.com.
102             IN PTR da102.muc.digitalairlines.com.
```

The SOA record (including hostmaster.digitalairlines.com) ***must*** be on a single line.

Make sure you enter your FQDN (such as da50.digitairlines.com) in the SOA and NS records.

Use the current date and "01" as the serial number (such as **2005071501**).

12. Save and close the file.
13. Open a second terminal window and enter **su -** to get root permissions.
14. When prompted, enter the root password **novell**.
15. Enter the command

    ```
    tail -f /var/log/messages
    ```

16. Switch to the first terminal window and start bind by entering

    ```
    rcnamed start
    ```

17. From the second terminal window, watch the log output of bind for any messages such as **Unknown RR Type** or **File Not Found**.
18. If any errors occur, fix them and restart bind.
19. Open the file **/etc/resolv.conf** in a text editor.
20. Delete all existing **nameserver** entries.
21. Add the following entry:

    ```
    search digitalairlines.com muc.digitalairlines.com
    ```

22. Save and close the file.
23. Verify that your DNS server works by entering

    host da100.muc.digitalairlines.com

    This should display the IP address of 10.0.1.100.

Part V: Enable Forwarding

To forward requests concerning a subdomain from the master server to the slave server, do the following on the master server:

1. Open a terminal window and enter **su -** to get root permissions.
2. When prompted, enter the root password **novell**.
3. To stop the DNS server, enter

    rcnamed stop

4. Open the **/etc/named.conf** file with a text editor.
5. Add the following zone after the other zone definitions:

```
zone “muc.digitalairlines.com” in
{
        type forward;
        forward only;
        forwarders
        {
                IP_address_of_the_slave_server;
        };
};

zone “1.0.10.in-addr.arpa” in
{
        type forward;
        forward only;
        forwarders
        {
                IP_address_of_the_slave_server;
        };
};
```

6. Save and close the file.
7. Open the file /var/lib/named/master/digitalairlines.com.zone with your text editor.
8. Add after the list of name servers and before the address records the following line:

    muc.digitalairlines.com. IN NS da2.digitalairlines.com.

9. Save and close the file.
10. Open a second terminal window and enter **su -** to get root permissions.

11. When prompted, enter the root password **novell**.

12. Enter the command

    tail -f /var/log/messages

13. Switch to the first terminal window and start bind by entering

    rcnamed start

14. From the second terminal window, watch the log output of bind for any messages such as **Unknown RR Type** or **File Not Found**.

15. If any errors occur, fix them and restart bind.

16. Verify that your DNS server works by entering

    host da100.muc.digitalairlines.com

    This should display the IP address of 10.0.1.100.

(End of Exercise)

Objective 3 Restrict Access to the Name Server

To prevent a name server from becoming overloaded, it can be useful to allow access only from specific computers or networks.

To do this, use the instruction **allow-query**. This instruction is defined in the **options** section of the /etc/named.conf file:

```
options
{

    ...

    allow-query
    {
        10.0.0.0/24;
        10.0.1.0/24;
    };
};
```

If you include this instruction, only computers from the two subnets defined can access this name server.

All other computers will receive an error message.

Do not use this instruction for the name server that provides information about your network to hosts on the Internet.

Objective 4 Configure Logging Options

The configuration statement logging can be used to define the logging behavior of BIND.

Log messages are written by using channels.

In addition to some predefined channels, an unlimited number of channels can be defined.

Every channel has to contain a destination that determines what to do with the messages.

Possible destinations are

- **syslog**. Messages are passed to the syslog daemon.
- **null**. Messages are dumped.
- **file**. Specifies the file (inside the chroot environment) to which the messages are written.

Every message has a priority level.

You can use the entry **severity** to determine from which priority level messages are written to this channel.

Possible levels of **severity** are

- **critical**
- **error**
- **warning**
- **notice**
- **info**
- **debug** [***level***]
- **dynamic**

The top five severities (**critical**, **error**, **warning**, **notice**, and **info**) are the familiar severity levels used by syslog.

debug is name server debugging for which you can specify a debug level. If you omit the debug level, then the level is assumed to be one. If you specify a debug level, you will see messages of that level when name server debugging is turned on.

If you specify **dynamic** severity, then the name server will log messages that match its debug level.

The default severity is **info**, which means you won't see debug messages unless you specify the severity.

The following is an example for a channel definition:

```
logging
{
    channel sec_file
    {
          file "/var/log/bind_security.log" versions 3 size 20m;
          severity debug 3;
          print-time yes;
          print-category yes;
          print-severity yes;
    };
};
```

All messages with debug level 3 sent to this channel are written to the /var/log/bind_security.log file with a time stamp (**print-time yes**), category (**print-category yes**) and severity level (**print-severity yes**). As soon as the file reaches a size of 20 MB (**size 20m**), a new file is created. Three old versions of the file are stored (**version 3**).

The following is an examples for predefined channels:

```
channel default_syslog {
    syslog daemon;
    severity info;
};

channel null {
    file "/dev/null";
};
```

Here, two predefined channels are listed: syslog daemon and null device.

The parameter **category** determines where messages are written to.

There are various categories, like **default** and **config**.

The following is an example for a **category** entry:

```
channel "my_security_channel"
{
        file "my_security_file";    # use an absolute path
        severity info;
};

category "security"
{
        "my_security_channel";
        "default_syslog";
        "default_debug";
};
```

This logs security events to the file my_security_file but also keeps the default logging behavior.

If no special rules are specified for a specific category, the rules of the **default** category apply.

If no specific logging options are specified in named.conf, a number of default values apply.

The following is an example:

```
category default { default_syslog; default_debug; };
```

If you use this default setting, all messages are sent to the syslog daemon.

In the debug status, messages are written to the log file named.run.

Objective 5 Security Aspects for Zone Transfer

In this objective, you learn about:

- Security Aspects of the Zone Transfer
- Zone Transfers from Slave Servers

Security Aspects of the Zone Transfer

It is important to ensure that only authorized hosts can perform a zone transfer.

Restricting the zone transfer is done using the **allow-transfer** statement. This statement takes either an IP address or the name of an encryption key as a parameter.

We strongly recommend that you use an encryption key, since the IP address can be spoofed quite easily.

The key is used to sign requests from the slave server. Only signed requests for zone transfers are accepted by the master server.

First, a key must be generated and made known to the master server and slave servers.

To do this, you do the following steps:

- Create a Key
- Configure the Master Server
- Configure the Slave Server

Create a Key

To create a key, use the **dnssec-keygen** command. The name of the key is printed on the screen:

```
da2:/var/lib/named # dnssec-keygen -a HMAC-MD5 -b 128 -n ZONE zonetransfer
Kzonetransfer.+157+01389
```

The options are explained in the following table:

Table 1-3

Option	Description
-A HMAC-MD5	The encryption procedure used
-b 128	The length of the key (in the example, 128 bits)
-n ZONE	The type of key

If you use this command, two files are created in the current directory:

```
da2:/var/lib/named # ls -l K*
-rw------- 1 root root 56 Feb 21 10:39 Kzonetransfer.+157+01389.key
-rw------- 1 root root 81 Feb 21 10:39 Kzonetransfer.+157+01389.private
```

The numbers at the end of the filename will be slightly different when you run this command.

Both files contain the same key:

```
da2:/var/lib/named # cat Kzonetransfer.+157+01389.key
zonetransfer. IN KEY 512 3 157 KhDVspoqFonWKv58rFXOWw==
da2:/var/lib/named # cat Kzonetransfer.+157+01389.private
Private-key-format: v1.2 Algorithm: 157 (HMAC_MD5)
Key: KhDVspoqFonWKv58rFXOWw==
```

This key has to be included in the configuration file /etc/named.conf on both the master and the slave server

Configure the Master Server

The following entries must be made in the file /etc/named.conf on the master server:

```
options {
    ...
};

key zonetransfer
{
     algorithm HMAC-MD5;
     secret "KhDVspoqFonWKv58rFXOWw==";
};

zone "digitalairlines.com" in
{
     type master;
     file "master/digitalairlines.com.zone";
     allow-transfer
         {
              key zonetransfer;
         };
};
```

A corresponding entry must be made for reverse resolution.

```
zone "0.0.10.in-addr.arpa" in
{
     type master;
     file "master/10.0.0.zone";
     allow-transfer
     {
         key zonetransfer;
     };
};
```

The key (the file /etc/named.conf) should not be readable for anybody except root and the group named.

This is ensured by the predefined permissions of /etc/named.conf:

```
da:~ # ls -l /etc/named.conf
-rw-r-----   1 root named 572 Jun  3 14:30 /etc/named.conf
```

Configure the Slave Server

On the slave server, define the following in the /etc/named.conf file:

- The key.
- The master server will handle all requests signed with this key (by using the **server** instruction)

The following is an example:

```
options
{
    ...
};

key zonetransfer
{
    algorithm HMAC-MD5;
    secret "KhDVspoqFonWKv58rFXOWw==";
};

server 10.0.0.2
{
    keys
    {
        zonetransfer;
    };
};
```

It is also important that only the user root and the group named be able to read the configuration file containing the key.

If a computer now tries to carry out a zone transfer without possessing the key, the zone transfer is refused.

On the master server, corresponding messages are listed in the file /var/log/messages.

On the master server, a successful zone transfer generates a message in /var/log/messages similar to the following:

```
Apr 21 10:37:18 da2 named[3976]: client 10.0.0.20#32769: transfer of
'digitalairlines.com/IN': AXFR started
```

A failed zone transfer generates a message similar to this in /var/log/messages on the master server:

```
Apr 21 10:38:59 da2 named[3976]: client 10.0.0.20#32775: request has
invalid signature: tsig verify failure
```

In /var/log/messages on the slave server, a message similar to this is generated:

```
Apr 21 10:40:01 da20 named[4099]: zone 0.0.10.in-addr.arpa/IN: refresh:
failure trying master 10.0.0.2#53: tsig indicates error"
```

Zone Transfers from Slave Servers

You should prohibit zone transfers from slave servers although it is possible to perform them.

If you only have one server (the master) that allows zone transfers to slave servers, you make sure that all slave servers get exactly the same information.

To prohibit the zone transfer from a server, use the following statement:

```
options
{
    allow-transfer {none;};
};
```

Exercise 1-2 Configure Zone Transfers from the Master Server to Slave Servers

To configure zone transfers from a master to a slave server, do the following:

- Part I: Generate a Key
- Part II: Configure the Master Server
- Part III: Configure the Slave Server

Part I: Generate a Key

1. To stop the DNS server, enter

   rcnamed stop

2. Change the directory by entering

   cd /var/lib/named

3. To generate a key, enter (one one line)

   dnssec-keygen -a HMAC-MD5 -b 128 -n ZONE zonetransfer

4. Record the name of the key in the space below:

5. Enter **cat *key_name*.private** and record the key (written in the last line of the output) in the space below:

Part II: Configure the Master Server

Do the following:

1. On the master server, open the **/etc/named.conf** file with a text editor.
2. Add the following lines ***after*** the **option** section:

3.
   ```
   key zonetransfer
   {
           algorithm HMAC-MD5;
           secret "your_key";
   };
   ```

   In the **secret** option, enter the key you recorded in Part I, Step 4.

4. Change the content of the zone description of digitalairlines.com as follows:

```
zone "digitalairlines.com" in
{
        type master;
        file "master/digitalairlines.com.zone";
        allow-transfer
        {
                key zonetransfer;
        };
};
```

5. Change the content of the zone description of 0.0.10.in-addr.arpa as follows:

```
zone “1.0.10.in-addr.arpa” in {
       type master;
       file “master/10.0.1.zone”;
        allow-transfer
        {
                key zonetransfer;
        };
};
```

6. For reverse resolution, add the following lines:

```
zone "0.0.10.in-addr.arpa" in
{
        type master;
        file "master/10.0.0.zone";
        allow-transfer
        {
                key zonetransfer;
        };
};
```

7. Save the file and exit the text editor.
8. Open a second terminal window and enter **su -** to get root permissions.
9. When prompted, enter the root password **novell**.
10. Enter the command

    ```
    tail -f /var/log/messages
    ```

11. Switch to the first terminal window and start bind by entering

    ```
    rcnamed start
    ```

12. From the second terminal window, watch the log output of bind when the slave server is started.

Part III: Configure the Slave Server

Do the following:

1. On the slave server, open the **/etc/named.conf** file with a text editor.

2. Add the following lines ***after*** the **option** section

   ```
   key zonetransfer
   {
           algorithm HMAC-MD5;
           secret "key_of_the_master_server";
   };

   server IP_address_of_the_master_server
   {
           keys
           {
                   zonetransfer;
           };
   };
   ```

   In the **secret** option, enter the key of the master server.

3. Save the file and exit the text editor.

4. Remove the files in the directory /var/lib/named/slave/ by entering:

   ```
   rm /var/lib/named/slave/*
   ```

5. Open a second terminal window and enter **su -** to get root permissions.

6. When prompted, enter the root password **novell**.

7. Enter the command

   ```
   tail -f /var/log/messages
   ```

8. Switch to the first terminal window and start bind by entering

   ```
   rcnamed start
   ```

9. From the second terminal window, watch the log output of bind if the zone transfer will be done.

(End of Exercise)

Objective 6 Configure Dynamic DNS

If the number of zones and hosts increases, it is inconvenient to edit the zone files manually.

You can modify the resource record sources of the name server dynamically without editing and reloading files. This is called dynamic DNS. Dynamic DNS can also be used by external services like DHCP.

To allow dynamic changes, add the following line in the zone definition.

```
allow-update { 127.0.0.1; };
```

In this example, only the loopback address is used, but it is also possible to add IP addresses of other workstations.

To edit the DNS records dynamically, use the command **nsupdate**:

```
da2:~ # nsupdate
>
```

Before using nsupdate, make sure that all zone files have the correct access permissions. If they not set properly, use the following commands:
chgrp -R named /var/lib/named
chmod R g+w /var/lib/named

Dynamic updates are stored in a journal file for the zone. This file is automatically generated by the server when the first dynamic update is performed.

The name of the journal file is created by appending the extension .jnl to the name of the corresponding zone file. The journal file is in a binary format and should not be edited manually.

The contents of the journal file is written to the zone file every 15 minutes. When the name server is shut down, the contents of the journal file are written to the zone file, too.

Dynamic updates will change the layout of your zone files when the data is written to them. You should either use an editor or **nsupdate** to modify your zone information instead of using both tools.

The most important nsupdate optione are

- **server *server* [*port*]**. Sends all dynamic update requests to the name server ***server***.

  If no ***port*** number is specified, the default DNS port number 53 is used.

- **update delete *domain* [*ttl*] [*class*] [*type* [*value...*]]**. Deletes any resource records named ***domain***.

If the record type and value are provided, only matching resource records will be removed.

The internet class (IN) is assumed if ***class*** is not supplied.

ttl is ignored. It is only allowed for compatibility.

- **update add *domain ttl* [*class*] *type value*...**. Adds a new resource record with the specified ***ttl*** (in seconds), ***class*** and ***value***.
- **send**. Sends the current message. This is necessary to actually execute the command.

  This is equivalent to entering a blank line.

All commands are described in the man page of nsupdate: **man 8 nsupdate**.

The following is an example of using nsupdate:

```
da2:~ # nsupdate
> server 127.0.0.1
> update add da13.digitalairlines.com 86400 A 10.0.0.13
>
> update delete da13.digitalairlines.com A
>
> update add 13.0.0.10.in-addr.arpa 86400 PTR da13.digitalairlines.com
>
```

This will generate messages like the following in /var/log/messages:

```
May 20 11:13:58 da2 named[5161]: client 127.0.0.1#32781: updating zone
'digitalairlines.com/IN': adding an RR
May 20 11:13:58 da2 named[5161]: journal file
master/digitalairlines.com.zone.jnl does not exist, creating it
May 20 11:13:58 da2 named[5161]: zone digitalairlines.com/IN: sending
notifies (serial 2005051903)
May 20 11:21:50 da2 named[5161]: client 127.0.0.1#32783: updating zone
'0.0.10.in-addr.arpa/IN': adding an RR
May 20 11:21:50 da2 named[5161]: journal file master/10.0.0.zone.jnl does
not exist, creating it
May 20 11:21:50 da2 named[5161]: zone 0.0.10.in-addr.arpa/IN: sending
notifies (serial 2005051902)
```

The journal files will be created automatically:

```
da2:/var/lib/named # dir master/
total 16
drwxrwxr-x  2 root  named 200 May 20 11:21 .
drwxrwxr-x  9 root  named 408 May 20 10:40 ..
-rw-rw-r--  1 root  named 463 May 19 12:06 10.0.0.zone
-rw-r--r--  1 named named 814 May 20 11:21 10.0.0.zone.jnl
-rw-rw-r--  1 root  named 440 May 19 14:51 digitalairlines.com.zone
-rw-r--r--  1 named named 794 May 20 11:13 digitalairlines.com.zone.jnl
```

Press **Ctrl + d** or enter **quit** to quit nsupdate.

In the following table, the most important record types are listed:

Table 1-4

Record Type	Meaning	Value
SOA	Start of Authority (term for the authority)	Parameter for the domain
NS	DNS server	Name of a DNS server for this domain
MX	Mail exchanger	Name and priority of a mail server for this domain
A	Address	IP address of the computer
PTR	Pointer	Name of the computer
CNAME	Canonical name	Alias name for the computer

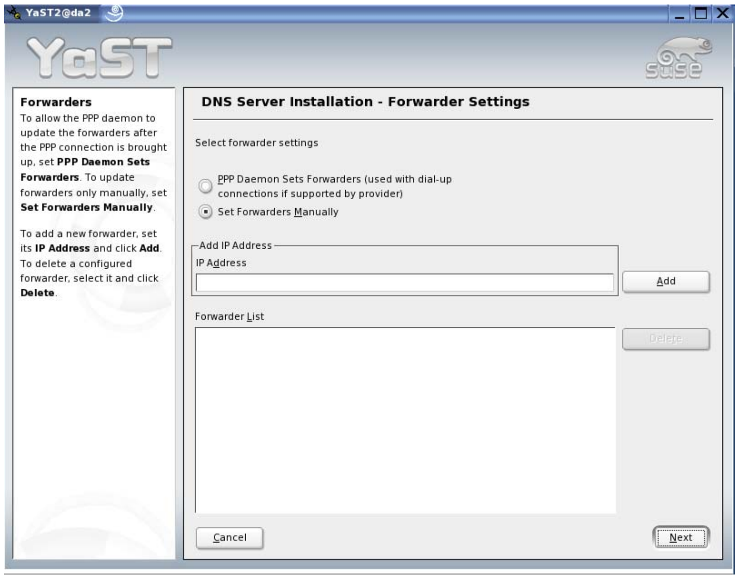
In Section 2 Use DHCP to Manage Networks, you will learn how to use a DHCP server to provide dynamic DNS updates.

Objective 7 Special Aspects of SUSE LINUX Enterprise Server 9 Configuration

One special aspect of SUSE LINUX Enterprise Server 9 is, that you can also use YaST to configure a BIND name server. To start the module, select

YaST2 > Network Services > DNS Server

Figure 1-2



Forwarders
To allow the PPP daemon to update the forwarders after the PPP connection is brought up, set **PPP Daemon Sets Forwarders**. To update forwarders only manually, set **Set Forwarders Manually**.

To add a new forwarder, set its **IP Address** and click **Add**. To delete a configured forwarder, select it and click **Delete**.

DNS Server Installation - Forwarder Settings

Select forwarder settings

PPP Daemon Sets Forwarders (used with dial-up connections if supported by provider) [✓ Set Forwarders Manually]

Add IP Address
IP Address

YaST is not able to decode manually created configuration and zone files.

See the installation and administration guide for details. We want to have a closer look to another special aspect of SUSE LINUX Enterprise Server 9 here.

The last entry of the /etc/named.conf file (installed with the bind package) is the following line:

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

If you include of this entry, the /etc/named.conf.include file is interpreted.

This meta file contains all files to be interpreted in addition to the /etc/named.conf file.

These files must be entered in the variable

NAMED_CONF_INCLUDE_FILES

in the file /etc/sysconfig/named.

This variable is interpreted by the named start script.

Files to be included should be located in the /etc/named.d/ directory.

By default, the name server runs in a chroot environment (**NAMED_RUN_CHROOTED="yes"** in /etc/sysconfig/named).

For security reasons, chroot should always be used.

All relevant files are copied to the respective subdirectories in /var/lib/named/. For example, /etc/named.conf is copied to /var/lib/named/etc/named.conf.

To disable the chroot environment, set the variable to **no**.

Additional options for starting the name server can be specified in /etc/sysconfig/named.

Objective 8 Use rndc to Control the Name Server

You can use the tool **rndc** (remote name daemon control) to manage the name server from a local or a remote host.

To use this tool, the requests have to be authenticated. This is done using a key similar to the key for the zone transfer. By default, a key is provided in the file /etc/rndc.key.

As this file comes with a default key from the bind package, do not use this key on a production system. You should always create your own key.

The key for rndc is generated using the command **rndc-confgen**. This generates a default configuration that is sent to standard output.

```
da2:~ # rndc-confgen > /etc/rndc.conf
```

The configuration file /etc/rndc.conf on the client used to manage the name server looks like this:

```
key rndc_key {
    algorithm "HMAC-MD5";
    secret "d3YWEw9jxo5UZ6N/EcIbFg==";
};

options {
    default-key “rndc-key”
    default-server 127.0.0.1;
    default-port 953;
};
```

The first statement defines the algorithm used to generate the key and the secret key. The second statement defines the default server to manage the request (the IP address of the name server) and the default key to use for the requests.

Using rndc, you can always provide other parameters.

As the file /etc/rndc.conf contains the key, read access needs to be limited:

```
da20:~ # dir /etc/rndc.conf
-rw-r-----  1 root named 141 Apr  7 16:16 /etc/rndc.conf
```

Instead of creating /etc/rndc.conf and adding information about the key to /etc/named.conf, the file /etc/rndc.key can be used for automatic rndc configuration. The file /etc/rndc.key is created using the command **rndc-confgen -a**.

On the name server, access via rndc has to be permitted.

This is done in the file /etc/named.conf by using the **controls** statement.

You also have to provide information about the key.

```
key rndc_key
{
        algorithm HMAC-MD5;
        secret "d3YWEw9jxo5UZ6N/EcIbFg==";
};

controls
{
        inet 127.0.0.1
                allow {localhost;} keys {rndc_key;};
        inet 10.0.0.2
                allow {localhost; 10.0.0.20;} keys {rndc_key;};
};
```

The **inet** statement defines which computers get access using rndc.

The first parameter defines the IP address on which named will listen for rndc requests.

In this example, the first statement instructs the computer to listen on the loopback interface with the IP address 127.0.0.1, and the second statement to listen on the interface with the IP address 10.0.0.2, too.

After the **allow** statement, a list of hosts that are allowed to access is provided.

For the loopback interface, access is allowed only from localhost.

For the IP address 10.0.0.2, access is allowed from local host and computer 10.0.0.20.

In both cases, the key named rndc_key has to be used to get access.

When you add the controls statement in /etc/named.conf, the name server listens on port 953 after reloading:

```
da20:~ # nmap da2

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2005-04-21 12:34
CEST
Interesting ports on da2.digitalairlines.com (10.0.0.2):
(The 1653 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
111/tcp open  rpcbind
427/tcp open  svrloc
631/tcp open  ipp
953/tcp open  rndc
```

If you enter rndc without any options, you see a list of available parameters:

```
da20:~ # rndc
Usage: rndc [-c config] [-s server] [-p port] [-k key-file ] [-y key] [-V]
command

command is one of the following:

  reload        Reload configuration file and zones.
  reload zone [class [view]]
                Reload a single zone.
  refresh zone [class [view]]
                Schedule immediate maintenance for a zone.
  reconfig      Reload configuration file and new zones only.
  stats         Write server statistics to the statistics file.
  querylog      Toggle query logging.
  dumpdb        Dump cache(s) to the dump file (named_dump.db).
  stop          Save pending updates to master files and stop the server.
  halt          Stop the server without saving pending updates.
  trace         Increment debugging level by one.
  trace level   Change the debugging level.
  notrace       Set debugging level to 0.
  flush         Flushes all of the server's caches.
  flush [view]  Flushes the server's cache for a view.
  status        Display status of the server.
  *restart      Restart the server.

* == not yet implemented
Version: 9.2.3
```

Using the options, you can always specify another server or another key to use for the request.

The following is an example for using rndc:

```
da20:~ # rndc status
number of zones: 4
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running

da20:~ # rndc dumpdb
```

The last line creates a dump of the cache of the name server.

The dump file is /var/lib/named/named_dump.db.

This file needs to be created with the correct permissions before cache data can be written to it:

- **touch /var/lib/named/named_dump.db**
- **chgrp named /var/lib/named_dump.db**
- **chmod g+w /var/lib/named_dump.db**

You can view the contents of the dump file using less or cat.

If an rndc request is tried from a host without access permissions (either not listed in the controls statement or without the correct key), an error message similar to the following will appear in /var/log/messages on the name server:

```
Apr 21 11:29:28 da2 named[4210]: rejected command channel message from
10.0.0.21#32770
```

On the denied client, the following message displays:

```
da21:~ # rndc dumpdb
rndc: recv failed: connection reset
```

Objective 9 Check Configuration Files

If you changed the configuration of your name server, check the syntax of the configuration files before you enable them.

To change the configuration, the BIND package provides tools:

- named-checkconf
- named-checkzone

named-checkconf

named-checkconf checks the syntax of the named configuration file (/etc/named.conf by default).

If the tool cannot detect an error, there is no output.

In case of an error, you get a message including the line number and a hint:

```
da2:~ # named-checkconf
/etc/named.conf:52: expected IP address near 'listen-on-v6'
```

If you want to check another file instead of /etc/named.conf**,** you can append the filename to the named-checkconf command.

named-checkzone

The command **named-checkzone** checks the syntax of a zone file.

named-checkzone has two parameters:

named-checkzone *zone-name file-name*

In contrast to named-checkconf, you see a message if no errors are found:

```
da2:~ # named-checkzone digitalairlines.com
/var/lib/named/master/digitalairlines.com.zone
zone digitalairlines.com/IN: loaded serial 2005071501
OK
```

Summary

Objective	Summary
1. Retrospection of DNS Basics	DNS translates host names into IP addresses. DNS is a distributed database. On SUSE LINUX Enterprise Server 9, you can use the BIND software to set up your own DNS server. A caching-only DNS server is not responsible for its own domain, it just forwards requests to other name servers and caches the result for later requests. A master server is responsible for its domain. It also provides resource information to host entries like the IP address of the mail server. DNS server information is stored in zone files. A slave DNS server receives copies of the domain zone files from the master server. Using slave servers enhances the reliability of the DNS.
2. Forward Requests to Other Name Servers	If the name server does not have any information on the computer to be resolved, it tries to fetch this information from another name server. In the configuration file /etc/named.conf, there should be a zone entry containing a reference to a file that contains the IP addresses of the root name servers. In this case the requests that could not be resolved locally are forwarded to one of the root servers. If requests for a subdomain are forwarded to another name server. That server must be defined in the corresponding zone entry. Using the entry **forward only**, the name server is instructed never to try to resolve the address itself.
3. Restrict Access to the Name Server	To prevent a name server from becoming overloaded, allow access only from specific computers or networks. This instruction is defined in the **options** section of the /etc/named.conf file.
4. Configure Logging Options	The logging block in the file /etc/named.conf contains the configuration of the BIND logging options. Log messages are written by means of channels. In addition to some predefined channels, an unlimited number of channels can be defined. Each channel has to contain a destination that determines what to do with the messages.

Objective	Summary
5. Security Aspects for Zone Transfer	Ensure that only authorized hosts can perform a zone transfer. This can be done by using an encryption key. Generate the key by using the **dnssec-keygen** command.
6. Configure Dynamic DNS	You can to modify the resource records of BIND dynamically and without editing and reloading files. Dynamic DNS can also be used by external services like DHCP. To manipulate the DNS records dynamically, use the **nsupdate** command.
7. Special Aspects of SUSE LINUX Enterprise Server 9 Configuration	A BIND 9 name server can also be configured by selecting **YaST2 > Network Services > DNS Server** The last entry of the /etc/named.conf file (installed with the bind package) interprets the file /etc/named.conf.include. This meta file contains all files to be interpreted in addition to the /etc/named.conf file.
8. Use rndc to Control the Name Server	The tool **rndc** (remote name daemon control) can be used to manage the name server from a local or a remote host. To use this tool, the requests have to be authenticated. You can create your key by using the command **rndc-confgen**. The configuration file /etc/rndc.conf on the client needs the following information: ■ The algorithm used for the generation of the key ■ The IP address of the name server ■ The default key to use for the requests On the name server, access via rndc has to be permitted. This is done in the file /etc/named.conf with the controls statement. You also have to provide information about the key.
9. Check Configuration Files	**named-checkconf** checks the syntax of the named configuration file (/etc/named.conf by default). **named-checkzone** checks the syntax of a zone file.