CST8246 – DNS, Part 1

Objectives

- Install and configure a **DNS server**.
- Automate the installation and configuration of a **caching and authoritative DNS server**.
- Verify running **services and their listening ports**.
- Explore newly installed packages using rpm or other package management utilities.
- Utilize **log files** to troubleshoot DNS-related issues.

Lab Outcomes

- Successfully **set up a caching DNS server**.
- Configure an **authoritative DNS server** for a specified domain.
- Automate DNS setup using a script to reduce errors and streamline deployment.
- Identify and verify active **DNS services and their corresponding ports**.
- Use system logs to diagnose and resolve **DNS configuration issues**.

Lab Deliverables

- **DNS local database files** configured to allow name resolution for supplied DNS names.
- **Automated script** that installs, configures, and validates a **DNS server** setup.
- A working **DNS server responding to client queries**.
- Full demonstration requirements are listed on **Brightspace**, where this lab was downloaded.

Section A - Initial Setup

Testing Name Resolution Using Your Existing DNS Server

Before configuring your own **DNS server**, test your system's **DNS client (stub resolver)**.

1. **Verify Your Current DNS Configuration:**

- Ensure that at least one **name server** is listed in `/etc/resolv.conf`
- Open a web browser and access any website to confirm that name resolution is working.

2. **Test Name Resolution Using dig:**
    - `dig` is the recommended tool for querying DNS records.
    - Syntax:

      ```
      dig [@nameserver] fqdn
      ```

    - **Note:** The nslookup utility was deprecated in the early 2000s but was later reintroduced. However, **dig** and **host** are preferred tools on RHEL 8.

Forward Lookup Test

- Use dig to query a domain name (e.g., Google's web server):
- dig google.com
- Verify the output:
    - You receive at least one answer (ANSWER: 1 in the header).
    - The flags include:
        - qr (query response)
        - rd (recursion desired)
        - ra (recursion available)
    - Identify the **name server** that responded. Does it match the one in `/etc/resolv.conf`?

Reverse Lookup Test

- Perform a **reverse DNS lookup** (resolve an IP to a domain name):

  ```
  dig -x <IP_address>
  ```

Querying a Specific DNS Record Type

- Look up specific **DNS resource records** (e.g., MX, SOA, NS):

  ```
  dig <record_type> <domain_name>
  ```

- Example: Retrieve Google's **name servers**:

  ```
  dig NS google.ca
  ```

Querying a Specific Name Server

- To query a specific name server:

  ```
  dig @<nameserver> <fqdn>
  ```

Tracing the DNS Delegation Path

- To trace the full **DNS resolution process**, use:

  ```
  dig <fqdn> +trace
  ```

Installing the DNS Server on RHEL 8

Installation & Setup

1. **Install or update BIND (named)**:
   - Install BIND and its utilities:

     ```
     dnf install bind bind-utils -y
     ```

   - Verify installation:

     ```
     rpm -q bind bind-utils
     ```

   - The **DNS service** runs as named.
2. **Understanding the DNS Client (Stub Resolver) in RHEL 8**
   - The **stub resolver** is built into **glibc** and handles name resolution.
   - Common system routines:
     - getaddrinfo → Resolves **domain names to IP addresses**.
     - getnameinfo → Resolves **IP addresses to domain names**.

Monitoring & Logging in RHEL 8

- **Monitor DNS logs in real-time:**

  ```
  journalctl -f -u named
  ```

- **Alternatively, use the traditional log file (if enabled):**

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

Name Server Configuration

Overview

A name server serves two primary roles:

1. **Caching Name Server**
   - Resolves domain names for local clients, typically within an internal network.
   - Handles **recursive queries**, storing responses temporarily to speed up future lookups.
2. **Authoritative Name Server**
   - Provides official DNS records for domains it manages.
   - Responds to **iterative queries** from other name servers.

Configuration Files

- The **main configuration file** for BIND (`/etc/named.conf`) defines:
  - Global server settings.
  - Locations of **zone files** (which store DNS records).
- Some distributions may not include a default named.conf file. To check, use:

  ```
  rpm -ql bind | grep named.conf
  ```
- A typical BIND setup consists of:
  - A **primary configuration file** (named.conf).
  - A **hints file** (listing root servers).
  - A **zone file** for each authoritative domain.

BIND Configuration Directives

Global Server Options (options Block)

- When setting up a new DNS service, start with a **basic configuration**, test it, and then modify options to match your environment.
- If a directive is missing, **default values** apply.
- The **options block** defines global service settings, including:
  - directory → Specifies the parent directory for DNS files (typically `/var/named`).

Zone Configuration (zone Block)

Each zone block specifies a domain for which the server is authoritative. At a minimum, this includes:

- **Local zone** (for localhost resolution).
- **Reverse local zone** (for reverse lookups).
- **Root hints zone** (used for recursive queries).

Why is the Root Hints Zone Important?

- A caching name server **first queries root servers** if an answer is not found in its cache.
- The **hints zone** provides a list of root servers for such lookups.

Example configuration for the **hints zone**:

```
zone "." IN {
  type hint;
  file "named.ca";
};
```

Localhost Zones

These prevent unnecessary queries for **localhost** from reaching root servers.

- **Forward lookup for localhost:**

```
zone "localhost" IN {
  type master;
  file "localhost.zone";
};
```

- **Reverse lookup for 127.0.0.1:**

```
zone "1.0.0.127.IN-ADDR.ARPA" IN {
  type master;
  file "named.loopback";
};
```

Including External Configuration Files

- BIND allows configurations to be split across multiple files.
- The include directive in `/etc/named.conf` **references additional files** containing service settings or zone definitions.

Example:

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

Configuring the DNS Client

Once BIND is installed, you need to configure both your **server** and **client** machines to use the DNS server exclusively for name resolution. This ensures that every resolver on the network—including the one on the name server itself—consistently references the same DNS server.

Steps to Configure the DNS Resolver:

1. **Modify the Resolver Configuration (/etc/resolv.conf)**
   - Before testing BIND, **comment out any existing nameserver entries** in /etc/resolv.conf.
   - This prevents misleading test results—otherwise, it may appear that BIND is working correctly when it is not.
2. **On the DNS Server:**
   - Add the loopback address (127.0.0.1) to the resolver configuration.
   - This ensures that the server **resolves DNS queries locally** before forwarding them elsewhere.
   - Example entry in `/etc/resolv.conf`:

     ```
     nameserver 127.0.0.1
     ```
   - (Optional) Specify the domain for which the server is authoritative using the search directive.
     - This allows **short, unqualified hostnames** to be automatically appended with the domain name.
     - Example: If the domain is exampleMN.lab, the entry would be:

       ```
       search exampleMN.lab
       ```
     - **Note:** This setting is used by the resolver library and some utilities, but **not by dig**.

3. **On Client Machines:**
   - Configure each client to use the **DNS server's IP address** in `/etc/resolv.conf`.
   - Example: If the DNS server's IP is 192.168.1.1, the entry would be:

     ```
     nameserver 192.168.1.1
     ```

Preventing DHCP from Overwriting DNS Settings

If your system obtains an IP address via DHCP, you may find that `/etc/resolv.conf` gets overwritten on each network restart.

To prevent this:

- Edit the network interface configuration file (e.g., `/etc/sysconfig/network-scripts/ifcfg-<interface>`).
- Add the following line:

```
PEERDNS=no
```

This prevents DHCP from modifying your DNS settings automatically.

Section B - BIND Configuration: Setting Up a Caching-Only Name Server

A **caching-only name server** is not authoritative for any specific zone. Instead, it is primarily used for **resolving domain names for internal clients** within an organization. Over time, it builds a **cache** of resolved queries, improving lookup performance for all local clients.

Note: If a configuration file already exists, it is usually set up as a caching-only name server by default.

Configuring a Caching-Only Name Server

Create or Modify the /etc/named.conf File

- It’s a good idea to **back up the existing configuration** before making changes.
- **You can use the existing file**, but be prepared for any consequences! 😆

Essential Configuration Steps:

- Add the **directory directive** to specify the working directory.
- Include the **"hints" zone** to enable resolution starting from the root servers.
- Include **zone files for the localhost zone** in a separate configuration file.
  - (You can use the default named.rfc1912.zones file provided by the BIND package—just reference it correctly in your config.)

Starting the Name Server

Open Four Terminal Sessions for monitoring and testing:

- **One for the BIND service**
- **One for viewing logs** (journalctl -f -u named)
- **One for client queries**
- **One for checking active network connections** (ss)

Key Actions:

- Start or restart the BIND service.
- Check the log output for any errors or important messages.
- Verify that the service is running and listening on the correct interfaces and ports using: `ss -tulnp | grep named`

Section C - BIND Configuration: Setting Up an Authoritative Name Server

Global Server Configuration Overview

An **authoritative name server** manages and provides authoritative responses for one or more DNS zones. It is primarily used for name resolution by external **clients** (other name servers) outside the organization.

The **BIND configuration** defines the behavior of the name server and specifies the location of the **zone files**, which contain DNS records for the zones the server is responsible for.

To configure your name server to be authoritative for the **exampleMN.lab** zone, follow these steps:

Step 1: Create the Zone Files

You need to define the necessary DNS records for your domain.

Step 2: Update the BIND Configuration

Modify the configuration file to include the **zone directives** for the domain(s) for which the server will be authoritative.

Required Zone Directives

At a minimum, your BIND configuration should include the following **zone directives**:

Hints Zone – Identifies the root servers (optional but recommended).
Localhost Forward Zone – Prevents queries for localhost from being forwarded to the root servers.
Localhost Reverse Zone – Provides reverse lookup for localhost.
Forward Zone(s) – Specifies the domain(s) for which your server is authoritative.
Reverse Zone(s) – Provides reverse DNS resolution for the authoritative domains.

Part 1: Setting Up a Forward Zone File

A **forward zone file** defines DNS records for a domain. This file typically includes:

- **Default TTL (Time-to-Live)** – Controls how long records are cached.
- **Comment Section** – Describes the zone configuration.

- **SOA (Start of Authority) Record** – Defines the primary name server and administrative contact.
- **NS (Name Server) Records** – Lists authoritative name servers for the domain.
- **A (Address) Records** – Maps hostnames to IP addresses.

Example Zone File Entries

- **Default TTL:**

  ```
  $TTL 86400  # 24 hours
  ```

- **Origin Directive:**

  ```
  $ORIGIN example.net.
  ```

- **SOA Record Example:**

  ```
  @ IN SOA ns1.example.net. dnsadm.example.net. (
    2000122401  ; Serial Number (use date + revision)
    28800       ; Refresh (8h)
    14400       ; Retry (4h)
    604800      ; Expire (1w)
    10800       ; Minimum TTL (3h)
  )
  ```

- **Name Server Record:**

  ```
  example.net. IN NS ns1.example.net.
  ```

- **A Record for the Name Server:**

  ```
  ns1 IN A 192.168.1.1
  ```

- **A Record for an FTP Server:**

  ```
  ftp IN A 192.168.1.2
  ```

Configuring the Forward Zone File for exampleMN.lab

- Create a forward zone file named fwd.exampleMN.lab and place it in the BIND configuration directory.
  Include the following records:

- Default TTL value

- Comment section identifying the zone
- SOA (Start of Authority) record
- NS (Name Server) record for ns1.exampleMN.lab
- A record for ns1.exampleMN.lab
- A record for ftp.exampleMN.lab, using the assigned IP address

BIND Configuration Update

In your BIND configuration file `(/etc/named.conf`), add a **zone block directive** for the **forward zone**.

Since this is the **master DNS server**, the server type should be set to master.
The **zone records** are stored in the zone file, which must be correctly referenced.

Example Configuration for exampleMN.lab

```
zone "exampleMN.lab" IN {

  type master;

  file "/etc/named/fwd.exampleMN.lab";

};
```

Testing Your Name Server

Once the configuration is complete:

- **Start the BIND service**
  Verify the service status using:

  ```
  ss -tulnp | grep named
  ```

- **Check logs** to confirm that zones loaded successfully:

  ```
  journalctl -f -u named
  ```

- **Run BIND utilities to check for syntax errors:**

  ```
  named-checkconf
  ```

```
named-checkzone exampleMN.lab /etc/named/fwd.exampleMN.lab
```

Testing with the dig Utility

Use dig to verify that the authoritative name server is working correctly. Record the results for each test.

- **Lookup ns1.exampleMN.lab**

  ```
  dig ns1.exampleMN.lab
  ```

Expected: The query should succeed and include the **"aa" (authoritative answer) flag**.

- **Lookup the FTP server (ftp.exampleMN.lab)**

  ```
  dig ftp.exampleMN.lab
  ```

Expected: The query should return the correct IP address.

- **Lookup the NS record for your zone**

  ```
  dig NS exampleMN.lab
  ```

Expected: The output should list ns1.exampleMN.lab as the authoritative name server.

- **Lookup the SOA record for your zone**

  ```
  dig SOA exampleMN.lab
  ```

Expected: The query should return the SOA record, including the primary name server and admin contact.

Procedure

1. **Ensure Firewall Configuration**
   - Disable any firewall rules that may interfere with DNS operation **or** automate the necessary firewall rules using firewalld or iptables.
2. **Automating DNS Installation and Configuration**
   - Develop a **setup script** that:
     - Installs the **BIND DNS package** (bind and bind-utils).
     - Configures the **caching DNS server** (/etc/named.conf).
     - Sets up **zone files** for the authoritative DNS server.
     - Restarts and enables the **DNS service (named)**.
     - Updates firewall rules to allow DNS traffic on **UDP/TCP port 53**.
3. **Testing & Verification**
   - Use dig and nslookup to **query the DNS server** and verify responses.
   - Run ss -tulnp or netstat -tulnp to check if **DNS is listening on port 53**.
   - Check journalctl -u named and /var/log/messages for potential errors.
4. **Troubleshooting**
   - **If the DNS server is not responding**, review the named.conf file for syntax errors.
   - **Validate zone files** using named-checkconf and named-checkzone.