Raw Notes

These are the raw notes from building this project.

- Internal DNS Configuration for isp1.net in a Private Network Environment
- Building the ISP Mail System with Roundcube
- Kea DHCP Configuration Documentation

Internal DNS Configuration for isp1.net in a Private Network Environment

Objective

This guide describes the steps to configure Core DNS to forward all requests for **isp1.net** to **ISP DNS 1** within an isolated network. The goal is to set up an authoritative, internal-only DNS for **isp1.net**, ensuring local queries are resolved internally without reaching external DNS servers.

Requirements

- Core DNS as the main DNS resolver for internal clients
- ISP DNS 1 as authoritative for **isp1.net**
- Internal domain **isp1.net** that only resolves within the private network, avoiding external DNS lookups

Setup Steps

1. Define the Zone for isp1.net on ISP DNS 1

First, configure ISP DNS 1 to serve as the authoritative DNS for **isp1.net**.

1. Edit the ISP DNS 1 configuration file (typically located at `/etc/bind/named.conf.local`) to add the zone for **isp1.net**. Define the zone as a master and specify `allow-query` to any IP and `allow-transfer` permissions for Core DNS (10.1.0.10).

```
zone "isp1.net" {
    type master;
    file "/etc/bind/zones/db.isp1.net";
    allow-query { any; };
    allow-transfer { 10.1.0.10; };
};
```

Create the zone file, typically at `/etc/bind/zones/db.isp1.net`. This file should include the SOA record, NS record, and A records for all devices within **isp1.net**.

```
$TTL 86400
@   IN  SOA isp-dns1.isp1.net. admin.isp1.net. (
        2023102701 ; Serial
        3600       ; Refresh
        1800       ; Retry
        1209600    ; Expire
        86400      ; Minimum TTL
    )
; Nameserver
    IN  NS  isp-dns1.isp1.net.
; A records
isp-router1     IN  A   10.10.0.2
isp-dns1        IN  A   10.10.1.10
isp-gateway     IN  A   10.10.2.1
isp-business    IN  A   10.10.3.1
```

Restart BIND on ISP DNS 1 to apply the changes:

```
sudo systemctl restart bind9
```

2. Configure Core DNS to Use ISP DNS 1 for isp1.net

On Core DNS, define a **stub zone** for **isp1.net** that points to **ISP DNS 1** as the authoritative DNS server for this domain.

1. Add a stub zone entry for **isp1.net** to the Core DNS configuration file, typically located at `/etc/bind/named.conf.local`. In this entry, specify the type as `stub`, set the `masters` to ISP DNS 1's IP (10.10.1.10), and add a `forwarders` directive with empty braces to prevent

forwarding to external servers.

```
zone "isp1.net" {
    type stub;
    masters { 10.10.1.10; };
    forwarders {};  # Prevents external forwarding for isp1.net
};
```

- Explanation of the `forwarders {};` Directive: By setting `forwarders {};`, we stop Core DNS from forwarding requests for **isp1.net** to any external DNS servers. This directive is crucial to ensure Core DNS exclusively queries ISP DNS 1 for this internal-only domain.
- Restart BIND on Core DNS to load the new configuration:

```
sudo systemctl restart bind9
```

3. Verifying the Configuration

Use the following steps to confirm that the configuration is working correctly.

1. Run a direct query to ISP DNS 1 from Core DNS to confirm that ISP DNS 1 is serving the **isp1.net** records correctly:

```
dig @10.10.1.10 isp-router1.isp1.net
```

Test forwarding from Core DNS by querying **isp1.net** records without specifying ISP DNS 1, confirming that Core DNS is forwarding queries correctly to ISP DNS 1:

```
dig isp-router1.isp1.net @10.1.0.10
```

Use tcpdump or a similar tool to verify that DNS requests for **isp1.net** are reaching ISP DNS 1 and returning the expected responses:

```
sudo tcpdump -i eth0 host 10.10.1.10 and port 53
```

Troubleshooting and Common Issues

1. **REFUSED Errors**: If Core DNS receives REFUSED responses, ensure that ISP DNS 1 has `allow-query` and `allow-transfer` settings configured to allow access from Core DNS (10.1.0.10).
2. **allow-query-cache Denials**: If cache queries are denied, add `allow-query-cache { 10.1.0.10; localhost; };` to ISP DNS 1 to permit Core DNS to access cached entries for faster responses.

3. **No Matching ‘Forwarders’ Statement**: The `forwarders {};` directive is necessary in this configuration to prevent Core DNS from forwarding **isp1.net** queries to global DNS servers. Adding this directive in the stub zone settings ensures exclusive forwarding to ISP DNS 1.

Building the ISP Mail System with Roundcube

For a setup emulating around 100 users accessing a Roundcube webmail server, you'll want a balance of resources to handle concurrent webmail sessions, email delivery, and storage for messages. Here’s a guideline for hardware resources to support this experimental environment effectively:

Recommended Hardware Resources

1. **CPU**:
   - **4-6 cores**: This will allow you to handle concurrent connections, especially as users access the webmail interface and the server handles SMTP/IMAP transactions.
   - Emulated users tend to put moderate stress on the CPU due to encryption (TLS/SSL for IMAP, SMTP) and web traffic, so having multiple cores will ensure responsiveness.
2. **Memory (RAM)**:
   - **8-12 GB RAM**: Roundcube itself is lightweight, but Dovecot and Postfix will need memory to handle simultaneous IMAP connections. Each user session can consume a small portion of RAM, especially with multiple concurrent mail retrievals.
   - Consider **16 GB** if you want extra room for scalability or plan to run additional monitoring tools.
3. **Storage**:
   - **200-500 GB SSD**: Storage needs can vary, but this range should be adequate for emails and attachments. SSDs are preferable to HDDs for faster I/O performance, which will help with message retrieval and database access.
   - **Database Storage**: Consider at least **50 GB** allocated for the MariaDB database, especially if you plan to retain a large amount of mail data.
4. **Network Bandwidth**:
   - **1 Gbps Network Interface**: Internal experiments typically don’t need heavy outbound bandwidth, but a reliable internal network interface with 1 Gbps is sufficient for handling mail traffic and simultaneous IMAP sessions.
5. **Additional Considerations**:
   - **Load Testing and Monitoring Tools**: Include tools like **Grafana, Prometheus**, or simple **Linux system metrics** to monitor CPU, memory, and network usage as you emulate load. This can help you identify bottlenecks and tune the configuration as needed.
   - **Virtual Machine Sizing**: If running in a VM, allocate resources as close to physical hardware specs as possible, avoiding overcommitted resources.

This setup will support your experiment with enough headroom to simulate user activity realistically. Let me know if you want further details on specific configurations for any of these components!

Kea DHCP Configuration Documentation

Overview

This configuration file is set up to provide DHCP services on a **10.10.1.0/24** network, specifically for an internal ISP user services network. It includes specific IP reservations, DNS settings, and comprehensive logging configurations to track DHCP activity at a detailed level.

Configuration Details


```
{
  "Dhcp4": {
    "interfaces-config": {
      "interfaces": [ "ens18" ]
    },
    "match-client-id": false,
    "subnet4": [
      {
        "subnet": "10.10.1.0/24",
        "pools": [],
        "option-data": [
          {
            "name": "domain-name-servers",
            "data": "10.10.1.10"
          },
          {
            "name": "domain-name",
            "data": "isp1.net"
          }
        ],
        "reservations": [
          {
```

```
        "hw-address": "BC:24:11:07:88:16",
        "ip-address": "10.10.1.100"
       }
     ]
    }
  ],
  "loggers": [
    {
     "name": "kea-dhcp4",
     "severity": "DEBUG",
     "output_options": [
       {
        "output": "/var/log/kea/dhcp4.log",
        "maxver": 10
       }
     ]
    },
    {
     "name": "kea-dhcp4.dhcpsrv",
     "severity": "DEBUG",
     "output_options": [
       {
        "output": "/var/log/kea/dhcp4-dhcpsrv.log",
        "maxver": 10
       }
     ]
    },
    {
     "name": "kea-dhcp4.leases",
     "severity": "DEBUG",
     "output_options": [
       {
        "output": "/var/log/kea/dhcp4-leases.log",
        "maxver": 10
       }
     ]
    }
  ]
}
```

```
}
```

Explanation of Each Section

1. **"interfaces-config"**:
   - **Purpose**: Defines the network interfaces Kea should listen on for DHCP requests.
   - **Setting**: `"interfaces": ["ens18"]` binds Kea to `ens18`, the network interface for the **10.10.1.0/24** subnet.
2. **"match-client-id": false**:
   - **Purpose**: Instructs Kea to ignore the DHCP Client ID and rely on the hardware (MAC) address for client identification. This can be helpful if clients do not consistently provide a Client ID or if MAC-based reservations are used.
3. **"subnet4"**:
   - **Purpose**: Defines the subnet settings for the **10.10.1.0/24** network.
   - **Details**:
     - `"subnet": "10.10.1.0/24"` specifies the address range.
     - `"pools": []` indicates no dynamic IP pool is configured, meaning only reservations will be assigned.
     - `"option-data"` includes DHCP options:
       - `"domain-name-servers": "10.10.1.10"` specifies the DNS server.
       - `"domain-name": "isp1.net"` sets the DNS search domain for the network.
     - `"reservations"` contains static IP reservations:
       - **Example Reservation**: A device with MAC `BC:24:11:07:88:16` receives the reserved IP address `10.10.1.100`.
4. **"loggers"**:
   - **Purpose**: Enables detailed logging at the **DEBUG** level, splitting logs across different categories.
   - **Details**:
     - `"kea-dhcp4"`: General DHCP activity log file at `/var/log/kea/dhcp4.log`.
     - `"kea-dhcp4.dhcpsrv"`: DHCP server-specific events logged at `/var/log/kea/dhcp4-dhcpsrv.log`.
     - `"kea-dhcp4.leases"`: Lease assignment and release events logged at `/var/log/kea/dhcp4-leases.log`.
   - **Log Rotation**: Each log file has `maxver: 10`, allowing up to 10 rotated log files.

Usage Notes

- **Reservations**: This configuration only serves devices with reservations due to the empty pool configuration. Ensure all required devices have reservations.
- **Logging**: The **DEBUG** level will produce detailed logs useful for troubleshooting and monitoring DHCP activity.

This setup should provide reliable, reservation-based IP assignment with comprehensive logging for monitoring DHCP activity on the **10.10.1.0/24** network. Let me know if there are additional

features or details you'd like to incorporate!