Set up and run a BIND validating recursive resolver.

```
Current version: 2024110700
```


Introduction

In this lab, you will set up, test and use the **resolv1** virtual machine in your network topology as your first recursive resolver. You will use BIND as DNS software. It is the first, oldest, and has been for a long time, the most widely used DNS software. It is developed by ISC and can operate as an authoritative name server or a recursive resolver[1].

> ⚠ **Warning**
>
> For deployment in a real-life environment, it is crucial to follow and apply industry operational and security best practices. For the purpose of this lab, we are just considering the basics.

1. Install and configure your server.

You will connect to the **resolv 1** container in your network topology and follow the below steps to install and configure your BIND validating recursive resolver.

Start by updating the system as follow:

```
$ sudo apt update -y
$ sudo apt upgrade -y
```

Install BIND packages that you need to run the DNS recursive resolver.

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

Take note of your server's IP addresses (IPv4 and IPv6) as you will need them later in the configuration.

```
$ sudo ip addr | grep "scope global"
    inet 100.100.X.67/26 brd 100.100.X.127 scope global eth0
    inet6 fd--:----:X:64::67/64 scope global
$
```

Then, use your prefered text editor (vi, vim, nano, etc.) to edit one of the configuration files of BIND located at `/etc/bind/named.conf.options` :

```
$ sudo nano /etc/bind/named.conf.options
```

This configuration file already has some content that you will now update to look like the below. Important reminder is to replace the IP addresses (resolver IP and network IP) to the correct ones.

```
options {
 // ...
 // #### EXISTING CONFIGURATION OMMITED ... ####
 //
  dnssec-validation auto;
  deny-answer-addresses { my_interfaces_IP-addresses; };
  listen-on { my_IPv4; };
  listen-on-v6 { my_IPv6; };
  allow-query { my_allowed_sources; };
  recursion yes;
};

// ######## Access-control lists ######
acl my_interfaces_IP-addresses {
  127.0.0.1/32;
  100.100.X.67/32;
  ::1/128;
  fd--:----:X:64::67/128;
};
```

```
acl my_allowed_sources {
  localhost;
  100.100.0.0/16;
  fd--:----::/32;
};

acl my_IPv4 {
  localhost;
  100.100.X.67;
};

acl my_IPv6 {
  localhost;
  fd--:----:X:64::67;
};
```

> **Note**
>
> - "**deny-answer-addresses**": tells the resolver to reject address (A or AAAA) records if the corresponding IPv4 or IPv6 addresses match with any of the resolver local IP addresses. This filtering is intended to prevent “DNS rebinding attacks” by telling named to ignore referrals to loopback and local addresses - including any VIPs on load balancers, if used. The reason for this is that an attacker could return bogus referrals, which contain NS names that resolve to loopback addresses or (if they have managed to discover them somehow) real interface addresses on your server. You don't want your server trying to query itself as part of the recursion process as this is pointless and wastes resources, which is the object of the attack [2].
> - "**dnssec-validation auto**": the default option that enables DNSSEC validation in your BIND resolver with a default DNS root zone trust anchor. Other options are "yes" and "no".
> - "**listen-on**": the interface(s) and port(s) on which the server listens for incoming queries sent using IPv4. By default, port 53 is used. Else, use listen-on port PORT_NUMBER.
> - "**listen-on-v6**": the interface(s) and port(s) on which the server listens for incoming queries sent using IPv6. By default, port 53 is used. Else, use listen-on port PORT_NUMBER.
> - "**allow-query**": specifies the hosts (clients) that are allowed to ask ordinary DNS questions to this server.
> - "**recursion**": enables (yes) or disables (no) the recursion function on this server.

Once you finish editing the file, verify the server configuration with the follwing command:

```
$ sudo named-checkconf
```

If there is any error in your *named* configuration, you will receive some outputs and you will need to fix the errors returned by the tool. Else, restart the server so it applies the new configuration:

```
$ sudo systemctl restart bind9
```

Check the status of the BIND service:

```
$ sudo systemctl status bind9
```

You should get an output similar to the below:

```
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
    Drop-In: /run/systemd/system/service.d
             └─zzz-lxc-service.conf
     Active: active (running) since Thu 2024-11-07 10:52:39 UTC; 3s ago
       Docs: man:named(8)
   Main PID: 5649 (named)
      Tasks: 26 (limit: 38066)
     Memory: 53.3M
     CGroup: /system.slice/named.service
             └─5649 /usr/sbin/named -f -u bind

Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: zone 127.in-
addr.arpa/IN: loaded serial 1
Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: zone localhost/IN:
loaded serial 2
Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: all zones loaded
Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: running
Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: managed-keys-zone:
Key 20326 for zone . is now trusted (acceptance timer complete)
Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: resolver priming
query complete
Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: checkhints:
b.root-servers.net/A (170.247.170.2) missing from hints
Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: checkhints:
b.root-servers.net/A (199.9.14.201) extra record in hints
Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: checkhints:
b.root-servers.net/AAAA (2801:1b8:10::b) missing from hints
Nov 07 10:52:40 resolv1.grpX.<lab_domain>.te-labs.training named[5649]: checkhints:
b.root-servers.net/AAAA (2001:500:200::b) extra record in hints
```

If you got the output "active (running)" without any particular error, the named service has properly started and next will be to test it by resolving DNS queries.

2. Test your BIND validating resolver

You will run the below queries and analyze their outputs in order to confirm your resolver is operational and validating DNSSEC ("**ad**" flag in the output for responses from domains that are DNSSEC signed).

1. dig SOA com. @100.100.X.67

2. dig SOA com. @100.100.X.67 +dnssec
3. dig DS com. @100.100.X.67
4. dig A www.icann.org @100.100.X.67
5. dig A www.icann.org @100.100.X.67 +dnssec
6. dig DS icann.org @100.100.X.67
7. dig DNSKEY icann.org @100.100.X.67 +dnssec +multi
8. dig SOA ispcp.info @100.100.X.67 +dnssec

Time to answer a few questions:

- Did you receive the "ad" flag in the response for all your DNS requests ?
- Why didn't you receive the "ad" flag nor an RRSIG record in the last response ?

Now try the following queries as well:

1. dig A www.dnssec-failed.org @100.100.X.67
2. dig www.dnssec-failed.org @100.100.X.67 +dnssec
3. dig www.dnssec-failed.org @100.100.X.67 +dnssec +cd

What did you notice ? Why ?

> **Note**
>
> The "+cd" means "checking disabled". The resolver server will not perform DNSSEC validation of responses if this bit is set in the dig query.

3. Update the resolver used by the Operating System.

Now that you have a working validating resolver, you can use it as the default resolver for your OS.

Still in **resolv1** server, edit the **/etc/resolv.conf** config file and update it as bellow.

```
nameserver 100.100.X.67
nameserver 9.9.9.9
```

Save and exit.

Then, try again all the previous DNS queries without specifying the server IP address.

Time to answer a few questions:

- Did you get all the responses as expected ?
- Which server responded and why ?

4. Temporary disable DNSSEC validation for broken domains

Various DNSSEC configuration faults on a signed zone can lead to a "bogus" or "SERVFAIL" result during validation at the resolver. This is actually what happens to the *dnssec-failed.org* zone.

While it is generally the responsibility of the zone administrator to fix the issue, the recursive resolver admin may need to temporarily disable DNSSEC validation for such "broken" domain to allow their users continue accessing it. The following configuration in the resolver will help you achieve that.

Edit the config file */etc/bind/named.conf.options* and add the following **inside** *options { ... };* statement :

```
options {
 // #### EXISTING CONFIGURATION OMMITED ... ####
 validate-except {
       "YOUR_BROKEN_DOMAIN.TLD";
       "dnssec-failed.org";
       };
};
```

Apply the configuration, restart named service and verify its status.

Then, confirm that the exception is working as expected by running the previous dig queries for dnssec-failed.org zone and verify that you received the resource records instead of SERVFAIL.

Once you have confirmed this, you can now remove the exception from your configuration.

Conclusion

Well done if you have completed this lab and have a validating resolver which is functioning. However, as you may suspect, what you did is the basic and a deployment in production environment requires additional configuration efforts. We do recommend you to have a look into BIND 9 Administrator Reference Manual and its configuration reference to learn more about this software.

Finally, security aspects are also important to consider. ISC has developped the BIND resolver best practices that you can follow. You should also refer to KINDNS and its guidelines, or other sources in the industry to learn more about best practices to operate recursive resolvers.

---

1. About BIND9: https://www.isc.org/bind/ ↩

2. BIND Best Practices - Recursive: https://kb.isc.org/docs/bind-best-practices-recursive ↩