Set up and run an Unbound validating recursive resolver

```
Current version: 2024110700
```

Introduction

In this lab, you will set up, test and use the **resolv2** virtual machine in your network topology as your second recursive resolver. You will use Unbound software, a validating, recursive, caching DNS resolver designed to be fast and lean. It incorporates modern features based on open standards and is developped by NlnetLabs[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. Set up Unbound validating recursive server (resolv2).

You will connect to the **resolv 2** container in your network topology and follow the below steps to install and set up your Unbound validating recursive resolver.

Start by updating the system as follow:

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

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

```
$ sudo apt-get install unbound -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.68/26 brd 100.100.X.127 scope global eth0
    inet6 fd--:----:X:64::68/64 scope global
 $
```

Then, use your prefered text editor (vi, vim, nano, etc.) to edit the Unbound configuration file located at `/etc/unbound/unbound.conf` :

```
$ sudo nano /etc/unbound/unbound.conf
```

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 to the correct ones.

Add the options to:

- listen to all interfaces (0.0.0.0 for IPv4 and ::0 for IPv6)
- allow queries from specific source IP addresses: in the current case these source addresses will be 127.0.0.1/32, 100.100.0.0/16, and fd--:----::/32
- allow UDP and TCP, IPv4 and IPv6 DNS queries.

  The configuration file should look like the below.

```
# Unbound configuration file for Debian.
#
# See the unbound.conf(5) man page.
#
# See /usr/share/doc/unbound/examples/unbound.conf for a commented
# reference config file.
#
# The following line includes additional configuration files from the
# /etc/unbound/unbound.conf.d directory.
```

```
server:
        interface: 0.0.0.0
        interface: ::0

        access-control: 127.0.0.0/8 allow
        access-control: 100.100.0.0/16 allow
        access-control: fd89:59e0::/32 allow

        port: 53

        do-udp: yes
        do-tcp: yes
        do-ip4: yes
        do-ip6: yes

include: "/etc/unbound/unbound.conf.d/*.conf"
```

It is recommended to checks the configuration file for syntax and other errors :

```
$ sudo unbound-checkconf
```

If all looks good, you should get something similar to the following:

```
unbound-checkconf: no errors in /etc/unbound/unbound.conf
```

Then restart the server to apply the changes:

```
$ sudo systemctl restart unbound
```

Check the status of the UNBOUND process:

```
$ sudo systemctl status unbound
```

You should obtain an output similar to the following:

```
● unbound.service - Unbound DNS server
     Loaded: loaded (/lib/systemd/system/unbound.service; enabled; vendor preset: enabled)
    Drop-In: /run/systemd/system/service.d
             └─zzz-lxc-service.conf
     Active: active (running) since Thu 2024-11-07 20:52:56 UTC; 5s ago
       Docs: man:unbound(8)
    Process: 5296 ExecStartPre=/usr/lib/unbound/package-helper chroot_setup (code=exited, 
status=0/SUCCESS)
    Process: 5299 ExecStartPre=/usr/lib/unbound/package-helper root_trust_anchor_update 
(code=exited, status=0/SUCCESS)
   Main PID: 5303 (unbound)
      Tasks: 1 (limit: 38066)
     Memory: 7.6M
     CGroup: /system.slice/unbound.service
             └─5303 /usr/sbin/unbound -d

Nov 07 20:52:56 resolv2.grpX.<lab_domain>.te-labs.training systemd[1]: Starting Unbound
DNS server...
Nov 07 20:52:56 resolv2.grpX.<lab_domain>.te-labs.training package-helper[5302]:
/var/lib/unbound/root.key has content
Nov 07 20:52:56 resolv2.grpX.<lab_domain>.te-labs.training package-helper[5302]: success:
the anchor is ok
Nov 07 20:52:56 resolv2.grpX.<lab_domain>.te-labs.training unbound[5303]: [5303:0] notice:
init module 0: subnet
Nov 07 20:52:56 resolv2.grpX.<lab_domain>.te-labs.training unbound[5303]: [5303:0] notice:
init module 1: validator
Nov 07 20:52:56 resolv2.grpX.<lab_domain>.te-labs.training unbound[5303]: [5303:0] notice:
init module 2: iterator
Nov 07 20:52:56 resolv2.grpX.<lab_domain>.te-labs.training unbound[5303]: [5303:0] info:
start of service (unbound 1.9.4).
Nov 07 20:52:56 resolv2.grpX.<lab_domain>.te-labs.training systemd[1]: Started Unbound DNS
server.
```

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

2. Test your Unbound 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.68
2. dig SOA com. @100.100.X.68 +dnssec
3. dig DS com. @100.100.X.68
4. dig A www.icann.org @100.100.X.68
5. dig A www.icann.org @100.100.X.68 +dnssec
6. dig DS icann.org @100.100.X.68
7. dig DNSKEY icann.org @100.100.X.68 +dnssec +multi
8. dig SOA ispcp.info @100.100.X.68 +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.68
2. dig www.dnssec-failed.org @100.100.X.68 +dnssec
3. dig www.dnssec-failed.org @100.100.X.68 +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 **resolv2** server, edit the **/etc/resolv.conf** config file and update it as bellow.

```
nameserver 100.100.X.68
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.

Add the following at the end of the **/etc/unbound/unbound.conf** file:

```
# Disable DNSSEC validation for the following zones
domain-insecure: "YOUR_BROKEN_DOMAIN.TLD"
domain-insecure: "dnssec-failed.org"
```

Apply the configuration, restart Unbound 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 Unbound documentation to learn more about this software.

Finally, security aspects are also important to consider. You should refer to KINDNS and its guidelines, or other sources in the industry to learn more about best practices to operate recursive resolvers.

---

1. About Unbound recursive resolver : https://www.nlnetlabs.nl/projects/unbound/about/ ↩