Innopolis University

SYSTEM AND NETWORKING ENGINEERING

Classical Internet Applications

LABORATORY REPORT 3 corrected

Domain Name System 1

Student Name	Student ID
Sergey Grebennikov	47611

Lecturer:

Rasheed Hussain

Submission Date : September 28, 2017

Contents

1 Introduction

The Domain Name System (DNS) is a computer distributed system for obtaining domain information. Most often it is used to obtain an IP address by the host name and get information about mail routing. The basis of the DNS is the representation of the hierarchical structure of the domain name and zones.

There are *BIND*[1] and *Unbound+NSD*[2] DNS implementation. *BIND* can serve both as a caching and as an authoritative nameserver. The *Unbound* nameserver is a caching nameserver and the *Name Server Daemon (NSD)* is an authoritative name server.

Assignments:

1. Downloading and Installing a Caching Nameserver
2. Configuring and Testing
3. Running and Improving the Name Server
4. (Installing and) Configuring an Authoritative Nameserver
5. Additional tasks

Initial Settings:

- IP-address: **188.130.155.46/27**
- DNS implementation: **Unbound+NSD**
- Domain: **os3.su**
- Subdomain: **st13.os3.su**

[1] Berkeley Internet Name Domain

[2] Name Server Daemon

2 Downloading and Installing a Caching Nameserver

The sources for the latest version of *Unbound* can be download from http://unbound.net.

2.1 Validating the Download

The Unbound website provides signature files in addition to the Unbound tarball. These can be used to check if you have downloaded the version they intended to distribute.

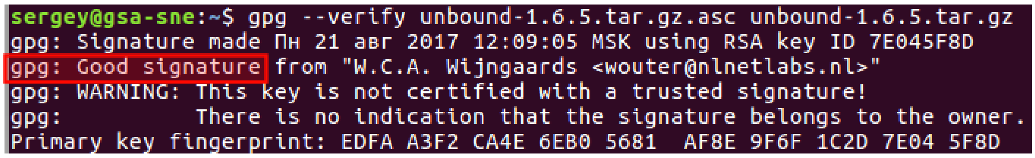
1. Why is it wise to use a signature to check your download?

   Answer: A signature allows to verify integrity and authentication of a downloaded file.

Ok. But Why is it wise to use a signature to check your download?

So we can make sure that the file has not been changed by an unauthorized person to install backdoors

Download the Unbound tarball and check its validity using one of the signatures (Figure 1).
Result:



```
sergey@gsa-sne:~$ gpg --verify unbound-1.6.5.tar.gz.asc unbound-1.6.5.tar.gz
gpg: Signature made Пн 21 авг 2017 12:09:05 MSK using RSA key ID 7E045F8D
gpg: Good signature from "W.C.A. Wijngaards <wouter@nlnetlabs.nl>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: EDFA A3F2 CA4E 6EB0 5681  AF8E 9F6F 1C2D 7E04 5F8D
```

Figure 1: Verifying the signature

2. Which kind of signature is the best one to use? Why?

   Answer: There is no unambiguous answer to this question, but it is probably GnuPG, because it uses a combination of conventional cryptography with a symmetric key for speed and public key cryptography for the convenience of secure key exchange.

Ok, PGP is more safe but the reasons are ...

Because PGP uses keys at least 1024 bits

2.2 Installation Documentation

Apart from the source code, the distributions contain documentation about the servers and DNS. For Unbound, the doc/ directory contains all information, including a README.

Most things about DNS are described and standardized in so-called “Request For Comments (RFCs)”, created and published by the Internet Engineering Task Force (IETF). A good DNS RFC to start with is RFC 1034: “Domain Names - Concepts and Facilities.”

2.3 Compiling

Configure, compile and then install the servers in the directory **/usr/local/** (Figure 1). Let the server write its state information, such as the **named.pid** file, in **/var/run**.

```
$ sudo apt-get install libssl-dev
$ sudo apt install libexpat1-dev
$ cd ~/unbound-1.6.5/
$ ./configure --with-pidfile=/var/run/unbound.pid
$ make
$ make check
$ sudo make install
```

3 Configuring and Testing

Compiling and installing a server is relatively simple, but configuring a DNS server is not trivial. To keep things simple, we will start with Unbound running as a caching-only name server. This type of name server does not control any zone data.

Question

3. Why are caching-only name servers still useful?

   Answer:

   - They do not participate in zone transfer, and therefore there is no zone transfer traffic
   - They can be placed on the far side of a slow WAN link and provide host name resolution for remote offices that do not require a high level of host name resolution support
   - They can be implemented to provide secure host name resolution when configured as Forwarders

> Not only caching DNS servers can't transfer zones, zone transfer usually allowed only to the secondary servers.

> ok

3.1 Main Configuration

For Unbound the main configuration file is **/usr/local/etc/unbound/unbound.conf**. It contains general options for the name server as well as references to other configuration files. The Unbound distribution already provides an example file in the desired location.

3.2 Root Servers

Our server needs a file containing references to the DNS root servers, the root hint file. The root hint file contains a list of root servers that our server uses to retrieve a more recent list of root servers. This file can be downloaded from ftp://ftp.rs.internic.net/domain.

```
$ wget ftp://ftp.rs.internic.net/domain/root.zone
$ sudo mv root.zone /usr/local/etc/unbound/
```

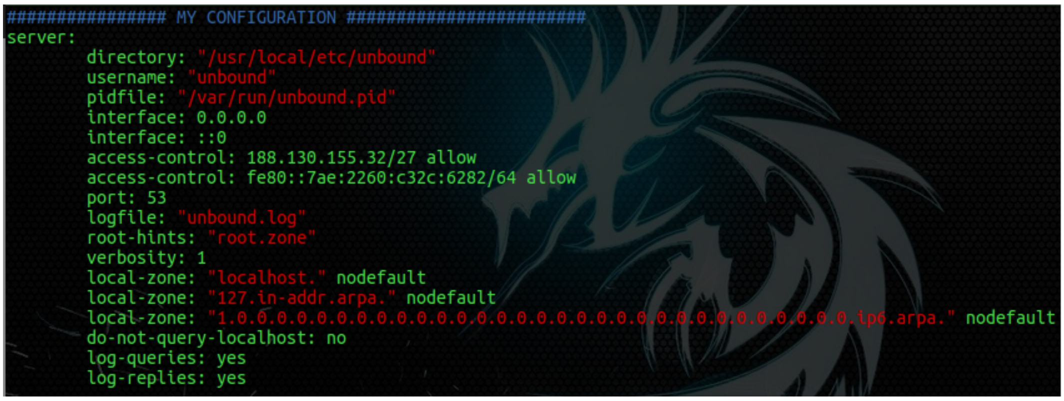
3.3 Resolving localhost

To resolve the loopback address 127.0.0.1 to the name localhost for Unbound, this is done automatically (the local-zone feature in the unbound.conf file).

Question:

4. Now that you know all the elements of the main configuration, configure a **unbound.conf** file for a caching-only name server.

   Show the configuration file in your report. (Figure 2)

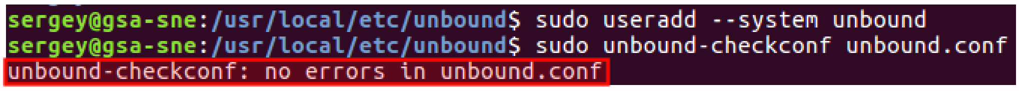


```
################ MY CONFIGURATION ########################
server:
        directory: "/usr/local/etc/unbound"
        username: "unbound"
        pidfile: "/var/run/unbound.pid"
        interface: 0.0.0.0
        interface: ::0
        access-control: 188.130.155.32/27 allow
        access-control: fe80::7ae:2260:c32c:6282/64 allow
        port: 53
        logfile: "unbound.log"
        root-hints: "root.zone"
        verbosity: 1
        local-zone: "localhost." nodefault
        local-zone: "127.in-addr.arpa." nodefault
        local-zone: "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa." nodefault
        do-not-query-localhost: no
        log-queries: yes
        log-replies: yes
```

Figure 2: The configuration file **unbound.conf**

3.4 Testing

The **unbound-checkconf** program uses to check the syntax of configuration file. It prints a line and returns a result value. (Figure 3)



```
sergey@gsa-sne:/usr/local/etc/unbound$ sudo useradd --system unbound
sergey@gsa-sne:/usr/local/etc/unbound$ sudo unbound-checkconf unbound.conf
unbound-checkconf: no errors in unbound.conf
```

Figure 3: The result of unbound-checkconf command

Question:

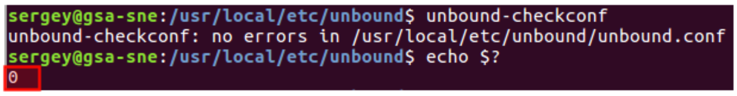
5. Why do the programs return a result value?

   Answer:

   Return values allow a process to monitor the exit state of another process often in a parent-child relationship. This helps to determine how this process terminated and take any appropriate steps necessary, depends on success or failure. A return value **0** indicates success (Figure 4) and any other one - failure.

ok but please add usage of values with description

To check a result value of the last process you should run the following command: **echo $?**



```
sergey@gsa-sne:/usr/local/etc/unbound$ unbound-checkconf
unbound-checkconf: no errors in /usr/local/etc/unbound/unbound.conf
sergey@gsa-sne:/usr/local/etc/unbound$ echo $?
0
```

Figure 4: The command result

4 Running and Improving the Name Server

To start the DNS server by hand is necessary to use **unbound -d -vv**. It will start the daemon with debug level 2.

It would be better to configure the name server to write debug information to a log file. To configure DNS server to write debug information to a log file it is necessary to write the following line in the **unbound.conf** file :

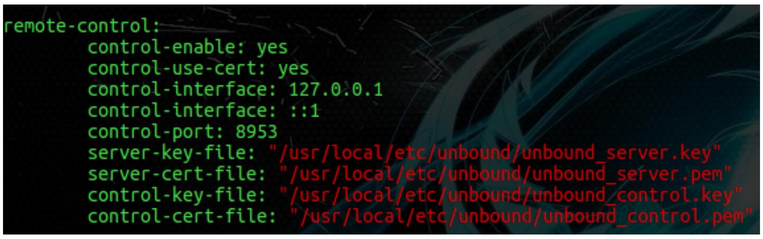
```
logfile: "unbound.log"
```

It is also better to use a remote server control utility to start and stop the server. For Unbound it is called **unbound-control**.

Question

Configure the server to use remote control:

6. Show the changes you made to your configuration to allow remote control. (Figure 5)



```
remote-control:
        control-enable: yes
        control-use-cert: yes
        control-interface: 127.0.0.1
        control-interface: ::1
        control-port: 8953
        server-key-file: "/usr/local/etc/unbound/unbound_server.key"
        server-cert-file: "/usr/local/etc/unbound/unbound_server.pem"
        control-key-file: "/usr/local/etc/unbound/unbound_control.key"
        control-cert-file: "/usr/local/etc/unbound/unbound_control.pem"
```

Figure 5: Remote Control Section

7. What other commands/functions does **unbound-control** provide?

   Answer:

   The main unbound-control commands:

   - **start** Start the server
   - **stop** Stop the server
   - **reload** Reload the server
   - **stats** Print statistics
   - **status** Display server status
   - **dump_cache** Display the cache
   - **list_stubs** List the stub zones in use
   - **list_forwards** List the forward zones in use
   - ...

8. What do you need to put in **resolv.conf** (and/or other files) to use your own name server?

   Answer:

   The **localhost** address (**127.0.0.1** through **127.255.255.254** for IPv4 and **::1** for IPv6)

ok Does this record will be saved after reboot ? Do you have another options to configure your PC with specific DNS server ?

This record will not be saved after reboot. Network Manager configuration can be used to configure PC with specific DNS server.

```
$ sudo vim /etc/NetworkManager/system-connections/Wired\ connection\ 1
...
[ipv4]
address1=188.130.155.46/27,188.130.155.33
dns=127.0.0.1
...
$ sudo systemctl stop NetworkManager.service
$ sudo systemctl start NetworkManager.service
```

5 (Installing and) Configuring an Authoritative Nameserver

Install and test the NSD server.

Installing:

```
$ sudo useradd --system nsd
$ ./configure --with-configdir=/usr/local/etc/nsd --with-nsd_conf_file=/usr/local/etc/nsd/nsd.conf
--with-dbfile=/usr/local/etc/nsd/nsd.db --with-zonesdir=/usr/local/etc/nsd/ --with-user=nsd
$ make
$ sudo make install
```

NSD systemd service file

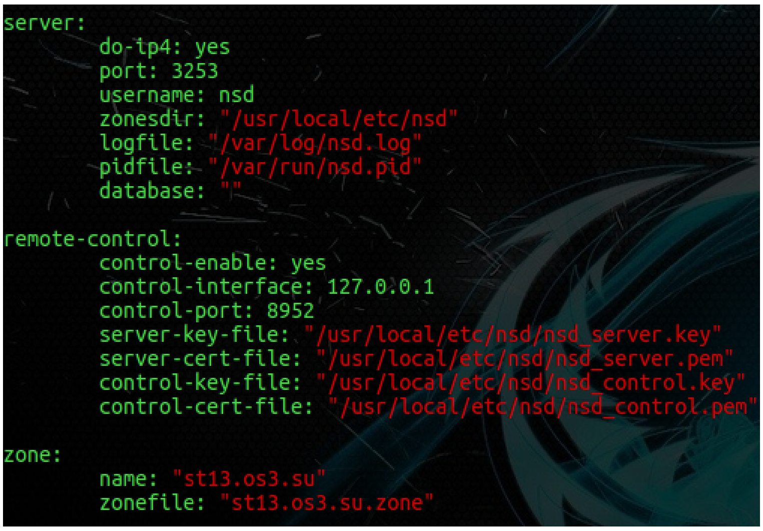
```
$ sudo vim /lib/systemd/system/nsd.service
```

```
[Unit]
Description=DNS server NSD
Wants=nss-lookup.target
After=syslog.target network.target remote-fs.target nss-lookup.target

[Service]
PIDFile=/var/run/nsd.pid
ExecStart=/usr/local/sbin/nsd
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s QUIT $MAINPID
PrivateTmp=true
```

```
[Install]
WantedBy=multi-user.target
```

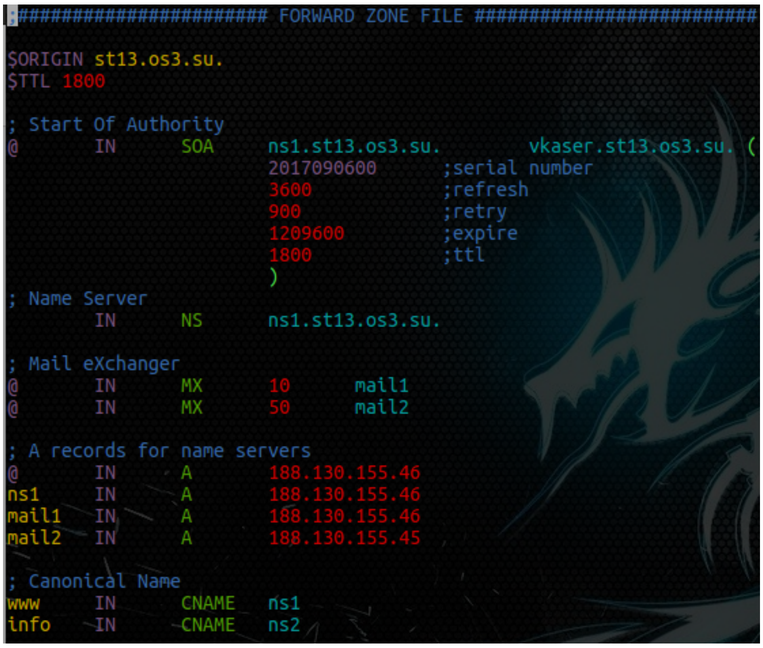
```
$ sudo ln -s /lib/systemd/system/nsd.service
↪  /etc/systemd/system/multi-user.target.wants/
$ sudo systemctl daemon-reload
```

Configuring: Figure 6, Figure 7



```
server:
        do-ip4: yes
        port: 3253
        username: nsd
        zonesdir: "/usr/local/etc/nsd"
        logfile: "/var/log/nsd.log"
        pidfile: "/var/run/nsd.pid"
        database: ""

remote-control:
        control-enable: yes
        control-interface: 127.0.0.1
        control-port: 8952
        server-key-file: "/usr/local/etc/nsd/nsd_server.key"
        server-cert-file: "/usr/local/etc/nsd/nsd_server.pem"
        control-key-file: "/usr/local/etc/nsd/nsd_control.key"
        control-cert-file: "/usr/local/etc/nsd/nsd_control.pem"

zone:
        name: "st13.os3.su"
        zonefile: "st13.os3.su.zone"
```

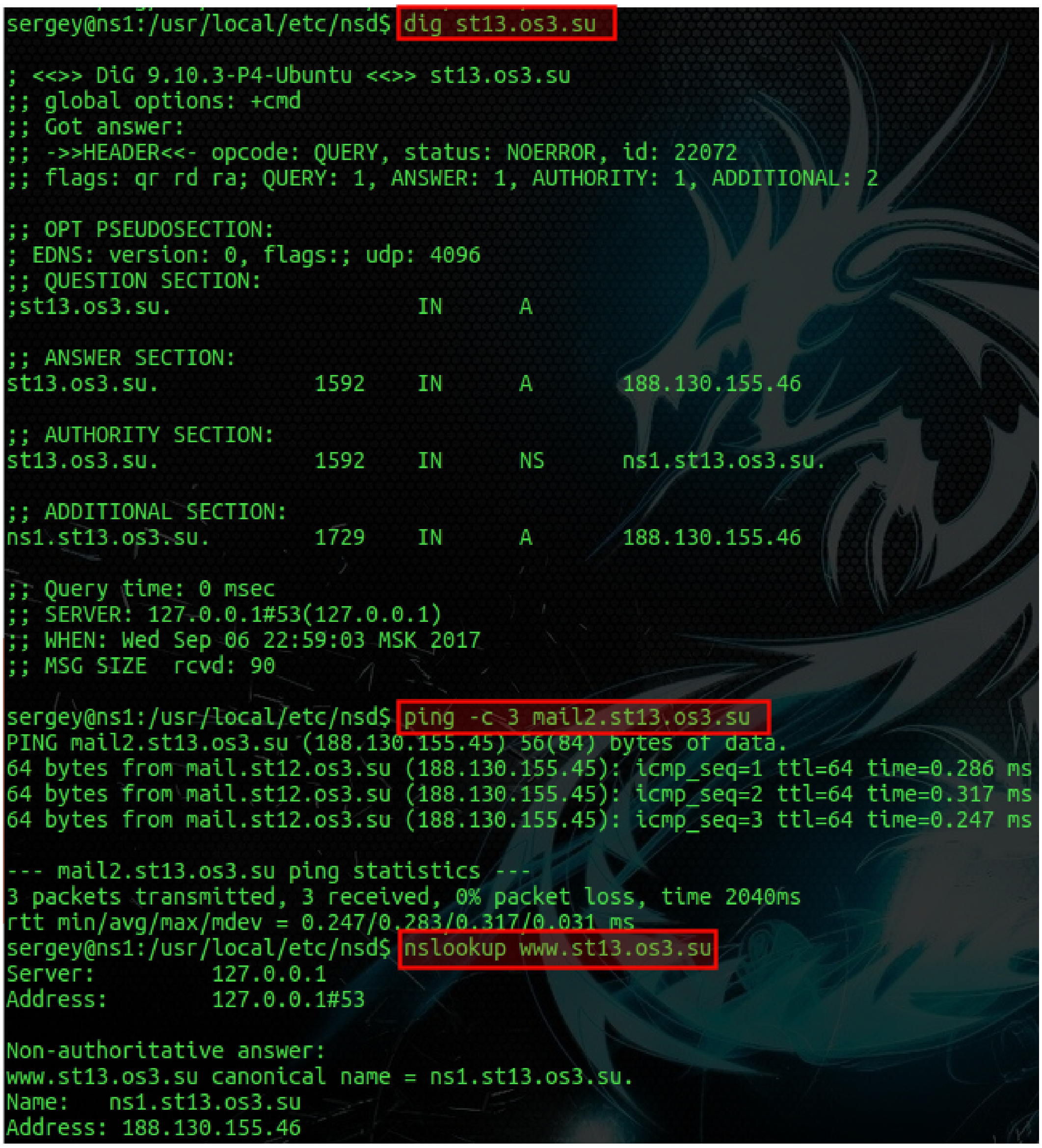
Figure 6: The NSD configuration file



```
;######################## FORWARD ZONE FILE ###########################

$ORIGIN st13.os3.su.
$TTL 1800

; Start Of Authority
@       IN      SOA     ns1.st13.os3.su.        vkaser.st13.os3.su. (
                        2017090600      ;serial number
                        3600            ;refresh
                        900             ;retry
                        1209600         ;expire
                        1800            ;ttl
                        )
; Name Server
        IN      NS      ns1.st13.os3.su.

; Mail eXchanger
@       IN      MX      10      mail1
@       IN      MX      50      mail2

; A records for name servers
@       IN      A       188.130.155.46
ns1     IN      A       188.130.155.46
mail1   IN      A       188.130.155.46
mail2   IN      A       188.130.155.45

; Canonical Name
www     IN      CNAME   ns1
info    IN      CNAME   ns2
```

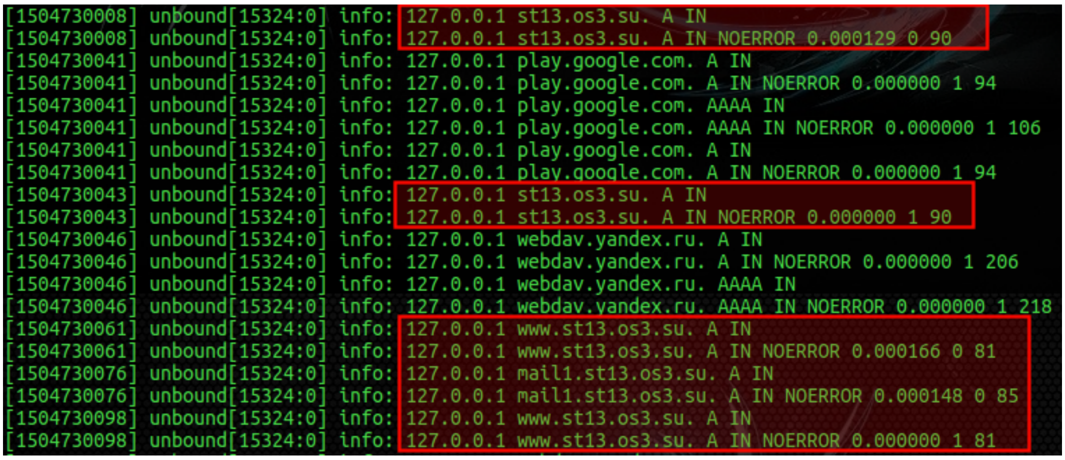
Figure 7: The forward mapping zone file

Result: Figure 8
Questions:



```
sergey@ns1:/usr/local/etc/nsd$ dig st13.os3.su

; <<>> DiG 9.10.3-P4-Ubuntu <<>> st13.os3.su
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22072
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;st13.os3.su.                   IN      A

;; ANSWER SECTION:
st13.os3.su.            1592    IN      A       188.130.155.46

;; AUTHORITY SECTION:
st13.os3.su.            1592    IN      NS      ns1.st13.os3.su.

;; ADDITIONAL SECTION:
ns1.st13.os3.su.        1729    IN      A       188.130.155.46

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Sep 06 22:59:03 MSK 2017
;; MSG SIZE  rcvd: 90

sergey@ns1:/usr/local/etc/nsd$ ping -c 3 mail2.st13.os3.su
PING mail2.st13.os3.su (188.130.155.45) 56(84) bytes of data.
64 bytes from mail.st12.os3.su (188.130.155.45): icmp_seq=1 ttl=64 time=0.286 ms
64 bytes from mail.st12.os3.su (188.130.155.45): icmp_seq=2 ttl=64 time=0.317 ms
64 bytes from mail.st12.os3.su (188.130.155.45): icmp_seq=3 ttl=64 time=0.247 ms

--- mail2.st13.os3.su ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.247/0.283/0.317/0.031 ms
sergey@ns1:/usr/local/etc/nsd$ nslookup www.st13.os3.su
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
www.st13.os3.su canonical name = ns1.st13.os3.su.
Name:   ns1.st13.os3.su
Address: 188.130.155.46
```

Figure 8: Answers



```
[1504730008] unbound[15324:0] info: 127.0.0.1 st13.os3.su. A IN
[1504730008] unbound[15324:0] info: 127.0.0.1 st13.os3.su. A IN NOERROR 0.000129 0 90
[1504730041] unbound[15324:0] info: 127.0.0.1 play.google.com. A IN
[1504730041] unbound[15324:0] info: 127.0.0.1 play.google.com. A IN NOERROR 0.000000 1 94
[1504730041] unbound[15324:0] info: 127.0.0.1 play.google.com. AAAA IN
[1504730041] unbound[15324:0] info: 127.0.0.1 play.google.com. AAAA IN NOERROR 0.000000 1 106
[1504730041] unbound[15324:0] info: 127.0.0.1 play.google.com. A IN
[1504730041] unbound[15324:0] info: 127.0.0.1 play.google.com. A IN NOERROR 0.000000 1 94
[1504730043] unbound[15324:0] info: 127.0.0.1 st13.os3.su. A IN
[1504730043] unbound[15324:0] info: 127.0.0.1 st13.os3.su. A IN NOERROR 0.000000 1 90
[1504730046] unbound[15324:0] info: 127.0.0.1 webdav.yandex.ru. A IN
[1504730046] unbound[15324:0] info: 127.0.0.1 webdav.yandex.ru. A IN NOERROR 0.000000 1 206
[1504730046] unbound[15324:0] info: 127.0.0.1 webdav.yandex.ru. AAAA IN
[1504730046] unbound[15324:0] info: 127.0.0.1 webdav.yandex.ru. AAAA IN NOERROR 0.000000 1 218
[1504730061] unbound[15324:0] info: 127.0.0.1 www.st13.os3.su. A IN
[1504730061] unbound[15324:0] info: 127.0.0.1 www.st13.os3.su. A IN NOERROR 0.000166 0 81
[1504730076] unbound[15324:0] info: 127.0.0.1 mail1.st13.os3.su. A IN
[1504730076] unbound[15324:0] info: 127.0.0.1 mail1.st13.os3.su. A IN NOERROR 0.000148 0 85
[1504730098] unbound[15324:0] info: 127.0.0.1 www.st13.os3.su. A IN
[1504730098] unbound[15324:0] info: 127.0.0.1 www.st13.os3.su. A IN NOERROR 0.000000 1 81
```

Figure 9: Log-file

9. Show the forward mapping zone file in your log.

   Answer: Figure 9

10. If Azat had not yet implemented the delegation, what information would you need to give him so that he can implement it?

    Answer: The server name that will be the authoritative server for the delegated subdomain and its IP address.

What else?

The port number of the NSD to configure zone transfer

11. What important requirement is not yet met for your subdomain?

    Answer:

    The secondary (or slave) name server has not been created.

OK Please specify the condition too

A secondary name server is required to improve the reliability and fault tolerance

6 Conclusion

The DNS plays a key role in modern information systems. Its task is to transfer the domain name to the IP address, and also serves the e-mail routing. The DNS has a decentralized structure with the ability to iteratively process requests. The DNS main concepts are Domain Name Space, Resource Records, Name Servers, Resolvers.

7 References

[1] Man page: unbound

[2] Man page: unbound.conf

[3] Man page: nsd

[4] Man page: nsd.conf

[5] Man page: nsd-control

[6] Man page: unbound-control

[7] How To Use NSD, an Authoritative-Only DNS Server, on Ubuntu 14.04 [https://www.digitalocean.com/community/tutorials/how-to-use-nsd-an-authoritative-only-dns-server-on-ubuntu-14-04]