Classical internet applications
Lab session: DNS-2

Anatoly Tykushin Security and Network Engineering, a.tykushin@innopolis.ru

August 29, 2016

Contents

1 Why is reverse zone is still useful?

With reverse DNS, your Internet connection provider (ISP) must point (or "sub-delegate") the zone ("....in-addr.arpa") to your DNS server. Reverse DNS is mostly used by humans for such things as tracking where a web-site visitor came from, or where an e-mail message originated etc.Many e-mail servers on the Internet are configured to reject incoming e-mails from any IP address which does not have reverse DNS.

2 Setting up reverse zone

2.1 Set up your own reverse zone for your IPv4 subnet. Use 10.192.5.0/24 subnet

To set up a reverse zone we need to change configuration for 3 files:

- **unbound.conf (added stub-zone)**
- **nsd.conf (added zone)**
- **5.192.10.zone with**
  - **Name servers**
    IN NS sne1.st5.os3.su
    IN NS sne2.st5.os3.su

- **PTR Records**
  1 IN PTR sne1.st5.os3.su
  58 IN PTR sne2.st5.os3.su
  77 IN PTR sne3.st5.os3.su

2.2 Show that a reverse lookup works

```
tyvision@tyvision95:/usr/local/etc/unbound$ dig -x 10.192.5.58

; <<>> DiG 9.10.3-P4-Ubuntu <<>> -x 10.192.5.58
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59502
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;58.5.192.10.in-addr.arpa. IN PTR

;; ANSWER SECTION:
58.5.192.10.in-addr.arpa. 86400 IN PTR sne2.st5.os3.su.5.192.10.in-addr.arpa.

;; AUTHORITY SECTION:
5.192.10.in-addr.arpa. 86340 IN NS sne1.st5.os3.su.5.192.10.in-addr.arpa.
5.192.10.in-addr.arpa. 86340 IN NS sne2.st5.os3.su.5.192.10.in-addr.arpa.

;; Query time: 0 msec
;; SERVER: 188.130.155.38#53(188.130.155.38)
;; WHEN: Wed Aug 24 22:26:42 MSK 2016
;; MSG SIZE  rcvd: 116
```

3 Delegating Your Own Zone

3.1 How did you set up the subdomain in your own zone file?

By adding 1 NS-record and an A-record. They are listed below:

```
                IN  NS  st6.os3.su.
;st6.st5.os3.su. IN A  188.130.155.39
```

3.2 What changes did you perform in nsd.conf file?

A couple of options was written to the zone definition file: **Notify** parameter in our pattern to reference our slave server's IP address. Set up the **provide-xfr** parameter exactly the same way.

```
notify: 188.130.155.39 NOKEY
provide-xfr: 188.130.155.39/32 NOKEY
```

3.3 Results of the performed delegation test

```
tyvision@tyvision95:~$ drill sne.st6.st5.os3.su @188.130.155.39
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 58265
;; flags: qr aa rd ra ; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
;; QUESTION SECTION:
;; sne.st6.st5.os3.su. IN A

;; ANSWER SECTION:
sne.st6.st5.os3.su. 86400 IN A 188.130.155.113

;; AUTHORITY SECTION:
```

```
st6.st5.os3.su. 86400 IN NS sne.st6.st5.os3.su.

;; ADDITIONAL SECTION:

;; Query time: 0 msec
;; SERVER: 188.130.155.39
;; WHEN: Thu Aug 25 18:31:34 2016
;; MSG SIZE  rcvd: 66
```

3.4 Setting up a slave server

3.4.1 How did you set up the slave nameserver?

Slave nameserver was set up by using **allow-notify** parameter in our pattern to reference our master server's IP address. Set up the **request-xfr** parameter exactly the same way.

3.4.2 Show the changes in configuration files

```
zone:
    name: "st6.st5.os3.su"
    zonefile: "st6.st5.zone"
    allow-notify: 188.130.155.39 NOKEY
    request-xfr: 188.130.155.39 NOKEY
```

3.5 What happens if the primary nameserver for the subdomain fails?

As we have only one DNS server authoritative for subdomain - it would be a single point of failure that could effectively isolate the entire hosts. We'll get field QUERYSTATUS = SERVFAIL in the answer packet

If we have a slave DNS server - they will catch the queries.

3.6 Considering that the slave nameserver is also the delegating nameserver, explain why this is essentially a bad setup?

I except that this will be a recursion which can be occasionally an infiinite loop.

4 Zone transfer

4.1 Output of the DNS tool

```
tyvision@tyvision95:~$ dig axfr st6.os3.su @188.130.155.39

; <<>> DiG 9.10.3-P4-Ubuntu <<>> axfr st6.os3.su @188.130.155.39
;; global options: +cmd
st6.os3.su. 86400 IN SOA st6.os3.su. st6.os3.su. 2016082502 360000 3600 3600000 3600
st6.os3.su. 86400 IN NS st5.os3.su.
st6.os3.su. 86400 IN NS st6.os3.su.
st6.os3.su. 86400 IN A 188.130.155.39
sne.st6.os3.su. 86400 IN A 188.130.155.39
sne1.st6.os3.su. 86400 IN A 188.130.155.39
sne2.st6.os3.su. 86400 IN A 188.130.155.39
st5.st6.os3.su. 86400 IN A 188.130.155.39
st6.os3.su. 86400 IN SOA st6.os3.su. st6.os3.su. 2016082502 360000 3600 3600000 3600
;; Query time: 0 msec
;; SERVER: 188.130.155.39#53(188.130.155.39)
;; WHEN: Fri Aug 26 17:29:18 MSK 2016
;; XFR size: 9 records (messages 1, bytes 241)
```

4.2 Steps description in the transfer process

1. The secondary DNS server requests the SOA record from the primary DNS server - SERVER - for Zone DOMAIN.COM. Note DNS Question Type.

2. The primary DNS server responds with the contents of the SOA record in the Answer Section.

3. Having compared the version number (serial number) and found it to be different than its current version number, the secondary DNS server now requests a Zone Transfer. Note the Question Type in the DNS Question Section.

4. The primary DNS server complies with the request for a Zone Transfer. The entire contents of the Zone file are transferred in the DNS Answer section.

To proof order of this stages

```
16:18:01.263307 IP 188.130.155.39.58250 > 188.130.155.38.domain: Flags [S], seq 3524492933, win 29200,
options [mss 1460,sackOK,TS val 18630976 ecr 0,nop,wscale 7], length 0
16:18:01.263330 IP 188.130.155.38.domain > 188.130.155.39.58250: Flags [S.], seq 2124635083,
ack 3524492934, win 28960, options [mss 1460,sackOK,TS val 18917106 ecr 18630976,nop,wscale 7], length 0
16:18:01.263585 IP 188.130.155.39.58250 > 188.130.155.38.domain: Flags [.], ack 1, win 229,
options [nop,nop,TS val 18630977 ecr 18917106], length 0
16:18:01.263647 IP 188.130.155.39.58250 > 188.130.155.38.domain: Flags [P.], seq 1:46, ack 1, win 229,
options [nop,nop,TS val 18630977 ecr 18917106], length 4524916 [1au] AXFR? st5.st6.os3.su. (43)
16:18:01.263655 IP 188.130.155.38.domain > 188.130.155.39.58250: Flags [.], ack 46, win 227,
options [nop,nop,TS val 18917106 ecr 18630977], length 0
16:18:01.263707 IP 188.130.155.38.domain > 188.130.155.39.58250: Flags [P.], seq 1:208, ack 46, win 227,
options [nop,nop,TS val 18917106 ecr 18630977], length 20724916*- 7/0/1 SOA,
NS st5.st6.os3.su., NS sne.st5.st6.os3.su., A 188.130.155.38, A 188.130.155.34, A 188.130.155.34,
SOA (205)
16:18:01.263937 IP 188.130.155.39.58250 > 188.130.155.38.domain: Flags [.], ack 208, win 237,
options [nop,nop,TS val 18630977 ecr 18917106], length 0
16:18:01.264315 IP 188.130.155.39.58250 > 188.130.155.38.domain: Flags [F.], seq 46, ack 208, win 237,
options [nop,nop,TS val 18630977 ecr 18917106], length 0
16:18:01.264369 IP 188.130.155.38.domain > 188.130.155.39.58250: Flags [F.], seq 208, ack 47, win 227,
options [nop,nop,TS val 18917107 ecr 18630977], length 0
16:18:01.264582 IP 188.130.155.39.58250 > 188.130.155.38.domain: Flags [.], ack 209, win 237,
options [nop,nop,TS val 18630977 ecr 18917107], length 0
```

4.3 What information did the slave server receive?

The answer was given in the previous question. Slave server gets Zone information from the primary server:

- SOA record, almostly it checks serial number
- all data from the zone file which stored on primary server.

4.4 Changes of configuration files

Changes was made for **nsd.conf** file. Unbound caching-only server was switched off and NSD was moved to port 53.

```
zone:
    name: "st6.st5.os3.su"
    zonefile: "st6.st5.zone"
    allow-notify: 188.130.155.39 NOKEY
    request-xfr: 188.130.155.39 NOKEY

zone:
    name: "st5.os3.su"
    zonefile: "st5.os3.su.zone"
    #notify: 188.130.155.39 NOKEY
    provide-xfr: 0.0.0.0/0 NOKEY
```

Note: string (notify ...) commented because of ACL check failure. This will be fixed in a shorted period of time.

4.5 Show how to make BIND/NSD run in a chroot environment

You have to set option in file **nsd.conf**

chroot: *directory*
NSD will chroot on startup to the specified directory. Note that if elsewhere in the configuration you specify an absolute path-name to a file inside the chroot, you have to prepend the chroot path...

And then use command **nsd-control reload** (also you can previously check **nsd-checkconfig** command)

4.6 What do all those parameters in the SOA record do, and what use could fiddling with them have?

As it is mentioned in RFC1033,

```
SOA  (Start Of Authority)


          <name>  [<ttl>]  [<class>]  SOA  <origin>  <person>  (
                           <serial>
                           <refresh>
                           <retry>
                           <expire>
                           <minimum> )

   The Start Of Authority record designates the start of a zone.  The
   zone ends at the next SOA record.

   <name> is the name of the zone.

   <origin> is the name of the host on which the master zone file
   resides.

   <person> is a mailbox for the person responsible for the zone.  It is
   formatted like a mailing address but the at-sign that normally
   separates the user from the host name is replaced with a dot.

   <serial> is the version number of the zone file.  It should be
   incremented anytime a change is made to data in the zone.

   <refresh> is how long, in seconds, a secondary name server is to
   check with the primary name server to see if an update is needed.  A
   good value here would be one hour (3600).

   <retry> is how long, in seconds, a secondary name server is to retry
   after a failure to check for a refresh.  A good value here would be
   10 minutes (600).

   <expire> is the upper limit, in seconds, that a secondary name server
   is to use the data before it expires for lack of getting a refresh.
   You want this to be rather large, and a nice value is 3600000, about
   42 days.

   <minimum> is the minimum number of seconds to be used for TTL values
   in RRs.  A minimum of at least a day is a good value here (86400).
```

Most of these fields are pertinent only for name server maintenance operations. However, MINIMUM is used in all query operations that retrieve RRs from a zone. Whenever a RR is sent in a response to a query, the TTL field is set to the maximum of the TTL field from the RR and the MINIMUM field in the appropriate SOA. Thus MINIMUM is a lower bound on the TTL field for all RRs in a zone. Note that this use of MINIMUM should occur when the RRs are copied into the response and not when the zone is loaded from a master file or via a zone transfer. The reason for this provison is to allow future dynamic update facilities to change the SOA RR with known semantics.