FreeIPA Training Series

DNS zone transfers from FreeIPA to non-FreeIPA slave servers

FreeIPA 3.0 and bind-dyndb-ldap 2.3

Petr Špaček <pspacek@redhat.com>

01-03-2013

Text file based (traditional) zones

- Zone = database used by DNS server.
- Text file, could be edited and distributed by hand.
- @ is a shorthand for zone origin, e.g. “example.com.”
- Zone origin will be appended to any **name** without period at the end.
  - e.g. “ns1” will be expanded to “ns1.example.com.”

```
@   IN    SOA   ns1 mail ( ; Start of Authority record
    78   ; serial number
    1D   ; refresh
    1H   ; retry
    1W   ; expire
    3H ); minimum           ; SOA record ends with “)”

;name    [TTL]    class type  record_content
@                   IN    NS    ns1                 ; NS record
;line above will be expanded to:
;example.com.      IN    NS    ns1.example.com.
ns1                 IN    A     192.0.2.1           ; glue record
```

Zone transfer

- DNS protocol allows copying of the zone (database) from one host to another.
- Traditional “one master-multiple slaves” schema.
- Slave servers have **read-only** copy of the database, all updates have to be made on the master server – single point of failure for DNS dynamic updates.
- Slave servers provide redundancy and allow load-balancing for **read only queries**.
- Slave servers have to be specified in NS records otherwise clients will ignore them.
- Zone is periodically transferred from master to slaves.

Zone transfer

- Parameters of the zone transfer are specified in SOA (Start of Authority) record. It will be described later.
- SOA serial number (“database version”) on master and slave is compared before each zone transfer:

No transfer is done when
`(master_serial <= slave_serial)`

- DNS protocol has two standard types of zone transfer:
  - AXFR – full transfer, “classical” way, RFC 5936
  - IXFR – incremental transfer, only changes made between old and new serial number are transferred, RFC 1995

Start of Authority (SOA) record

```
(       ;  SOA starts as each other record, ( ) allow splitting over lines
@       ;  record name = zone origin (example.com.)
1W      ;  optional, Time-To-Live for caches (1 week, i.e. 604800 seconds)
IN      ;  optional, class Internet, other classes (CHaos) are usually unused
SOA     ;  record type = Start of Authority

;  SOA record specifics starts here, as defined by RFC 1035 section 3.3.13.
ns1     ;  master server name, all dynamic updates should go here
hostmaster ;   e-mail of DNS server admin
78 ;  SOA serial, "database version number", RFC 1982
1D ;  refresh, time interval between two successive zone transfers
1H ;  retry, time interval before a failed refresh should be retried
1W ;  expire, upper time limit on the cache before the zone is no longer
   ;  authoritative. Slave will stop serving this zone a week after the last
   ;  successful zone transfer.
3H ;  minimum, TTL for negative cache, RFC 2308
)
```

A note about SOA serial numbers

- SOA serial serial is unsigned 32 bit integer.
- Value 0 can trigger bugs in some software, please start with 1.
- Range: 1 – 2^{32}-1 (i.e. 1 – 4 294 967 295)
- **It can safely "overflow"** (modulo arithmetic):
  - Maximal defined addition is 2^{31}-1, i.e. 2 147 483 647
  - Bigger addition will result in wrong comparison results.
  - 1 **is smaller than** 2 000 000 000
  - 2 000 000 000 **is smaller than** 4 000 000 000
  - 4 000 000 000 **is smaller than** 1
- RFC 1982 contain all the gory details.

DNS Notify (RFC 1996)

- Disadvantages of periodical zone pooling (quotation from RFC 1996):
  - “Longer refresh times are beneficial in that they reduce load on the master servers, but that benefit comes at the cost of long intervals of incoherence among authority servers whenever the zone is updated.”
- “The DNS NOTIFY transaction **allows master servers to inform slave servers when the zone has changed** – an interrupt as opposed to poll model – which it is hoped will reduce propagation delay while not unduly increasing the masters' load.”
- Notification is sent to all servers listed in NS records.

DNS Notify (RFC 1996)

- **Slave** after receiving a NOTIFY message have to:
  - Do normal DNS query for SOA record from affected zone.
  - Compare SOA serial from local copy with SOA serial on master server.
  - Start zone transfer only if (slave_serial **<** master_serial).

- As a result, DNS NOTIFY mechanism and zone transfers will not work without proper SOA serial maintenance (incrementation after each change in the zone).

Configuring zone transfers without FreeIPA

- Master side – /etc/named.conf

```
zone "example.com." IN {
  type master;
  file "master/example.com";
  allow-transfer { 203.0.113.0/29; };
};
```

- Slave side – /etc/named.conf

```
zone "example.com." IN {
  type slave;
  file "slave/example.com";
  masters { 192.0.2.1; };
};
```

Zone transfer with NOTIFY in logs

- Master side – /var/log/messages

```
zone example.com/IN:
  sending notifies (serial 101)
client 203.0.113.1#43793:
  transfer of 'example.com/IN': AXFR-style IXFR started
client 203.0.113.1#43793:
  transfer of 'example.com/IN': AXFR-style IXFR ended
```

- Slave side – /var/log/messages

```
client 192.0.2.1#41290:
  received notify for zone 'example.com'
zone example.com/IN: Transfer started.
transfer of 'example.com/IN' from 192.0.2.1#53:
  connected using 203.0.113.1#43793
zone example.com/IN: transferred serial 101
transfer of 'example.com/IN' from 192.0.2.1#53:
  Transfer completed: 1 messages, 237 records,
  5673 bytes, 0.002 secs (2836500 bytes/sec)
zone example.com/IN: sending notifies (serial 101)
```

FreeIPA specifics – DNS data in LDAP

- FreeIPA replaced traditional text file with tree of objects stored in LDAP database. DNS server BIND 9 uses bind-dyndb-ldap plugin for accessing the database.
- Zone transfer mechanism between FreeIPA servers is not necessary because whole LDAP database is replicated between all FreeIPA servers.
- Replication is done by 389 DS. DNS server doesn't have to care about data synchronization (in ideal case).
- FreeIPA supports traditional zone transfer method for non-FreeIPA slaves.
  - It didn't work well with bind-dyndb-ldap < 2.0.

FreeIPA specifics – problems with LDAP database

- Single database shared and replicated between all DNS servers brings some new problems:
  - Administration tools can change DNS data on any server at any time.
  - DNS server has to notice the change.
  - SOA serial has to be incremented after each change.
  - Replication between FreeIPA servers takes some time. Different DNS servers will see changes in different time and potentially in different order.
  - Replication of serial numbers could create race conditions. Perfect global synchronization of SOA serial could be very complex and expensive.

Automatic SOA serial number incrementation

- **FreeIPA hack**: Each DNS server maintains own serial number independently, each server can return different SOA serial value.
- DNS server watches LDAP DB with persistent search. (ancient RFC draft-ietf-ldapext-psearch-03)
- 389 DS sends “Entry Change Notification” to DNS server after each change.
- Each change in LDAP database triggers SOA serial incrementation in DNS server according to following algorithm:

```
If (old_serial < current_unix_timestamp)
    new_serial = current_unix_timestamp
else if (old_serial >= current_unix_timestamp)
    new_serial = old_serial + 1
```

Configuration: allowing zone transfer from FreeIPA

- Set `allow-transfer` attribute in specific DNS zone.
- Syntax is exactly same as in `named.conf`:
  - Bare IP address and network/mask are accepted.
  - It is possible to mix IPv4 and IPv6 addresses.
  - BIND keywords “any” and “none” (default).
- `$ ipa dnszone-mod example.com '--allow-transfer=192.0.2.8;203.0.113.0/29;2001:DB8:AA::1;2001:DB8:BB::/64;'`
- (Semicolon has to be handled specially in BASH.)
- This configuration allows zone transfer but doesn't solve SOA serial incrementation.

Configuration: SOA serial auto-incrementation

- **FreeIPA 3.0 should configure SOA serial auto-incrementation by default!**
- LDAP attribute `idnsSOAserial` must not be replicated between FreeIPA servers.
- E.g. replication agreement with server `ipa2.example.com` is stored in LDAP under DN: `cn=meToipa2.example.com,cn=replica,cn=dc\3Dexample\,dc\3Dcom,cn=mapping tree,cn=config`
- Attribute `nsDS5ReplicatedAttributeList` has to contain `idnsSOAserial` (among others).

Configuration: SOA serial auto-incrementation

- **FreeIPA 3.0 configures SOA serial auto-incrementation by default!**
- This is what is added to /etc/named.conf by FreeIPA installer:

```
dynamic-db "ipa" {
  arg "psearch yes";
  arg "zone_refresh 0";
  arg "serial_autoincrement yes";
}
```

- Persistent search is required for serial_autoincrement.
- Zone refresh has to be disabled (0 or option not present).
  - Persistent search and zone refresh are mutually exclusive.

Configuration: SOA serial initialization: new zones

- **Recommendation**: Set initial SOA serial number to some small number (e.g. 1) when adding a new zone to FreeIPA.
- SOA serial will be immediately replaced with current UNIX timestamp if serial auto-incrementation is already enabled.
- This will allow to serial to follow real UNIX timestamp, as described on previous slide *Automatic SOA serial number incrementation*
  - Theoretically, all FreeIPA servers should present SOA serial close to timestamp of last update.
  - This could help non-FreeIPA slaves with failover between masters.

Configuration: SOA serial initialization: existing zones

- Before the very first zone transfer please reset SOA serial in a zone to value smaller than current UNIX timestamp (to e.g. 1).
- This prevents situation where serial > current UNIX timestamp, i.e. serial will not contain timestamp from the future.
- Do the reset **only once**, before you start with zone transfer to non-FreeIPA slaves first time.
- This reset could help non-FreeIPA slaves with failover between masters.
  - Serial incrementation algorithm is on previous slide *Automatic SOA serial number incrementation*

Testing zone transfers

- Use `dig` from `bind-utils` package
- `$ dig @DNS_server_IP -t AXFR +multiline example.com`
  should list all records in the zone (some parts cut):

```
; <<>> DiG 9.8.2rc1 <<>> -t AXFR example.com @192.0.2.1 +multiline
;; global options: +cmd
example.com.      86400 IN SOA ipa.example.com. hostmaster.example.com. (
                  1357314370 ; serial
                  3600       ; refresh (1 hour)
                  900        ; retry (15 minutes)
                  1209600    ; expire (2 weeks)
                  3600       ; minimum (1 hour) ; Usage according to RFC 2308
                  )
example.com.      86400 IN NS  ipa.example.com.
ipa.example.com.  86400 IN A   192.0.2.1
example.com.      86400 IN SOA ipa.example.com. hostmaster.example.com. (
    <snip>
                  )
    <snip>
;; XFR size: 4 records (messages 1, bytes 149)
```

- SOA record should appear twice
  (as the first and last transferred record)

Debugging zone transfers

- Zone transfer with `dig` failed:

```
; <<>> DiG 9.8.2rc1 <<>>
;; global options: +cmd
; Transfer failed.
```

- Check the logs on DNS (FreeIPA) server:
  - `/var/log/messages`
  - `/var/named/data/named.run`
  - Location is defined by `directory` and `logging` & `channel` & `file` directives in `named.conf`.

Debugging zone transfers: incorrect allow-transfer ACL

- BIND log should say:

```
named[4241]: client 192.0.2.9#44656:
zone transfer 'example.com/AXFR/IN' denied
```

  - Add `192.0.2.9` to `allow-transfer` ACL.

- You used @IPv4 as server's address but log say:

```
named[4241]: client ::ffff:192.0.2.9#51276:
zone transfer 'example.com/AXFR/IN' denied
```

  - => Your kernel is a a bit quirky ... Try one of following:

a) Add `match-mapped-addresses yes;` to `options` section in `/etc/named.conf`

b) Add `::ffff:192.0.2.9` addresses to allow-transfer ACL

Debugging zone transfers: zone is not active

- BIND log should say:

```
named[4241]: client 192.0.2.9#34772:
bad zone transfer request: 'example.com/IN':
non-authoritative zone (NOTAUTH)
```

- Zone `example.com` is not an active zone on the DNS server. Is the zone enabled?

```
$ ipa dnszone-show example.com.
  Zone name: example.com
<snip>
  Active zone: FALSE
```

Debugging zone transfers: zone is invalid

- Zone is invalid when some requirements for DNS zones are not satisfied. E.g. NS records don't have corresponding A/AAAA (so-called glue) records.
- BIND doesn't log anything about the zone transfer in this case, but **zone inconsistency is logged**:

```
named[2310]: zone example.com/IN:
NS 'ns1.example.com' has no address records
(A or AAAA)
named[2310]: zone example.com/IN:
not loaded due to errors.
```

- Query for SOA record if nothing was logged:

```
$ dig -t SOA @DNS_server_IP example.com
<snip>
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL
```

Debugging zone transfers: connection failed

- Firewall or DNS server is improperly configured or is not running.

```
$ dig -t AXFR @DNS_server_IP example.com
;; Connection ... failed: connection refused.
;; connection timed out; no servers could be reached
```

- UDP **and TCP** port 53 have to be opened in firewall (even for regular DNS clients): RFC 5966 section 1
- Query for SOA record usually uses UDP:

```
$ dig -t SOA @DNS_server_IP example.com
<snip>
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
```

Debugging zone transfers: connection failed

- Big answers can't be transferred without TCP (some TXT records were added to the zone for this test):

```
$ dig -t TXT @DNS_server_IP example.com
;; Truncated, retrying in TCP mode.
;; Connection ... failed: connection refused.
```

- Check `listen` directives in `/etc/named.conf` if firewall is opened:

```
options {
        // turns on IPv6 for port 53
        // IPv4 is on by default for all ifaces
        listen-on-v6 {any;};
}
```

Limitations

- Bind-dyndb-ldap supports only full zone transfers (AXFR). Whole zone is transferred each time.
- Generally, slave servers can't fail over between different FreeIPA masters because each FreeIPA server maintains own independent SOA serial number.
  - Fail-over could work in environments with update rate smaller than 1 change per second. In that case SOA serial number should be nearly same on all hosts.
  - `multi-master` option in BIND 9.9.2 affects **only logging** but not the zone transfer logic. It will not help with fail-over!