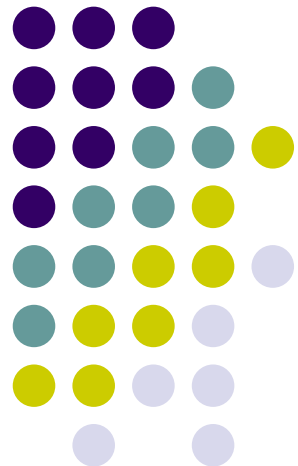
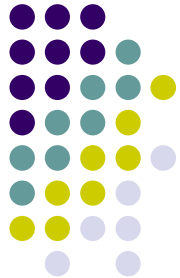
LDAP back-end for BIND9 – schema and tools

Stig Venaas

venaas@uninett.no



Introduction

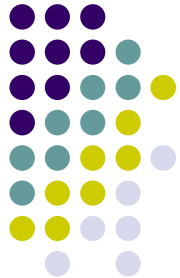


- Started working on LDAP back-end for BIND9 in 2001 to test sdb interface
  - Just an experiment, nothing serious
  - Soon realised that no good LDAP schema existed for storing zones, so created one
  - A bit more since then serious

- Will first talk about the LDAP schema for storing zones
- And next some tools using this

- This is my own private work and not in any way related to my employer
- For more information, source code etc, see http://www.venaas.no/ldap/bind-sdb/ or contact me

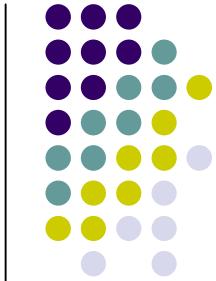
LDAP dnszone schema 1/4



- Cosine dnsdomain defines some RR types. We need more, and also some other stuff
- Example dnszone LDAP entry for SOA

```
dn: relativeDomainName=@, dc=my-domain, dc=com
objectClass: dNSZone
relativeDomainName: @
zoneName: my-domain.com
dNSTTL: 3600
dNSClass: IN
sOARecord: ns.my-domain.com. hostmaster.my-
  domain.com. 2001030201 3600 1800 604800 86400
nSRecord: ns.my-domain.com.
nSRecord: ns.other-domain.com.
mXRecord: 10 mail.my-domain.com.
mXRecord: 20 mail.other-domain.com.
```

LDAP dnszone schema 2/4

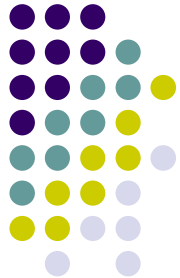


- Example dnszone LDAP entry for a host

```
dn: relativeDomainName=my-host, dc=my-domain,
   dc=com
objectClass: dNSZone
relativeDomainName: my-host
zoneName: my-domain.com
dNSTTL: 3600
dNSClass: IN
aRecord: 10.10.10.10
mXRecord: 10 mail.my-domain.com.
mXRecord: 20 mail.other-domain.com.
```

- You can choose whatever directory structure and naming you like. Different branches for different zones or entries for all zones together is up to you

LDAP dnszone schema 3/4



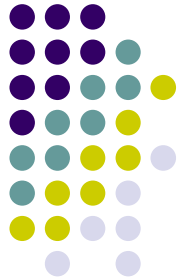
- RRs for a node might span multiple entries and one entry might contain RRs for multiple nodes
  - As an example, what if you want the same MX records for several hosts

```
dn: relativeDomainName=my-hosta, dc=my-domain, dc=com
objectClass: dNSZone
relativeDomainName: my-hosta
zoneName: my-domain.com
dNSTTL: 3600
dNSClass: IN
aRecord: 10.10.10.10

dn: mXRecord=10 mail.my-domain.com, dc=my-domain, dc=com
objectClass: dNSZone
relativeDomainName: my-hosta
relativeDomainName: my-hostb
zoneName: my-domain.com
mXRecord: 10 mail.my-domain.com.
mXRecord: 20 mail.other-domain.com.
```

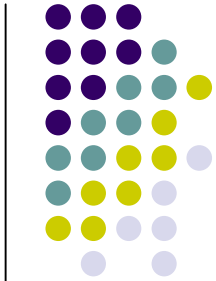
- Here the MX records are for both my-hosta and my-hostb, and will be merged with the other RR data for my-hosta and my-hostb
- If multiple nodes have exactly the same RR data, then just one entry is needed for them all

LDAP dnszone schema 4/4



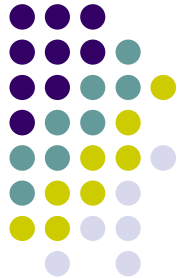
- Schema gives very close mapping to zone files but is also flexible
- Future proof in that OIDs and LDAP attribute names are derived from IANAs current and future RR type and value definitions
- It’s useful to use strings and human readable LDAP data
- Using LDAP for storing zones can be useful when you want applications/scripts to manipulate zones
  - One example is web interface for maintaining zones
    - Can be useful when people with no DNS clue are to maintain some zone info
  - Someone has written script to update LDAP zone based on DHCP leases
  - Other applications? Tunnel brokers, ENUM, SPAM related rules?
- LDAP offers fine grained access control. Some may have access to modify RRs for only certain nodes, or certain RR types for all

LDAP back-end for BIND9



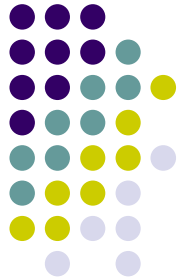
- This makes use of BIND9's sdb interface
- Master for a zone can be configured to use the LDAP back-end instead of a zone file.
  - In named.conf you use the database directive instead of file
- Whenever someone queries the master, an LDAP query is made. No caching
- This may be useful in some special cases. Some applications/scripts might update the zone and a query always gives up to date info
- Additional delay, but for some uses, LDAP is quick enough. Can talk to LDAP using UNIX domain socket, and LDAP server may do indexing and caching
- If you want BIND to cache, you can use a stealth master, and only list slaves. Slaves still do AXFR and are not concerned with how data is stored at master
- Can also have multiple masters for a zone. By using LDAP replication, each master can have a local LDAP replica of zone

ldap2zone tool



- There are advantages to storing zones in LDAP
- If you want unmodified BIND and want caching etc as usual, you might still store zones in LDAP
- ldap2zone tool creates zone files from LDAP
- It can return different exit codes depending on whether serial number has changed
- This means you can query LDAP very frequently. Whenever serial number changes, you then create a new zone file and tell server to reload the zone
- http://www.venaas.no/dns/ldap2zone/

Questions/comments?



- Some people find storing zones in LDAP useful. Quite a few are using the BIND9 back-end or tools like ldap2zone.
- Is it useful to any of you? Ideas for uses I haven’t mentioned?
- Please contact me venaas@uninett.no or see http://www.venaas.no/ldap/bind-sdb/ for more info