Managing DNS zones using Git

Ondřej Caletka
Ondrej.Caletka@cesnet.cz

October 2018

Contents

cesnet

Why ISPs still manage DNS zones manually

cesnet

webhosting
- core business
- typically own solution integrated in the control panel

enterprise
- forward and reverse host records
- integrated with DHCP/directory server

ISP
- mostly reverse records
- not many tools deal properly with (classless-) subdelegations

Classless subdeledations according to BCP 20 / RFC 2317

```
128/25  IN NS server.example.com.
        IN NS secondary.example.com.
128     IN CNAME 129.128/25
129     IN CNAME 129.128/25
…
```

```
…
252     IN CNAME 252.128/25
253     IN CNAME 253.128/25
254     IN CNAME 254.128/25
255     IN CNAME 255.128/25
```

cesnet

DNS management evolution in CESNET

- direct editting of the zone file on the server
  - forgotten serial update means inconsistency
  - syntax error means outage
- move to hidden master server, public slave servers
  - editting zone on the hidden master server
  - no outage due to typos
  - DNSSEC signatures by OpenDNSSEC
  - local Git repository for version tracking
- **hidden master controlled by Git repository**
  - syntax errors are avoided by Git hooks
  - servers are reloaded after change in the repository
  - DNSSEC signatures in independent component
  - allows splitting management to different teams

cesnet

Motivation to change

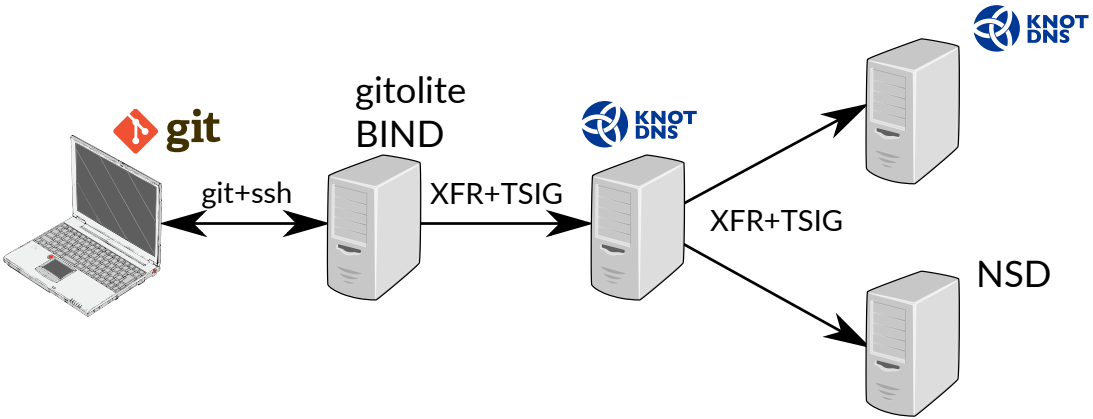
- many steps, error-prone
  1. do the change
  2. increase the serial
  3. resign the zone
  4. reload DNS server
  5. commit to the Git repository
- some issues with OpenDNSSEC
  - occasional deadlocks of SQLite database – recommended to switch to MySQL
  - painful upgrade from 1.3 to 1.4, non-trivial upgrade to 2.0
  - KASP inconsistency due to sharing keys between zones
  - no support for algorithm rollover (until 2.0)
  - no support for CDS/CDNSKEY KSK rollover

cesnet

Managing zone files using Git

cesnet

Desired state

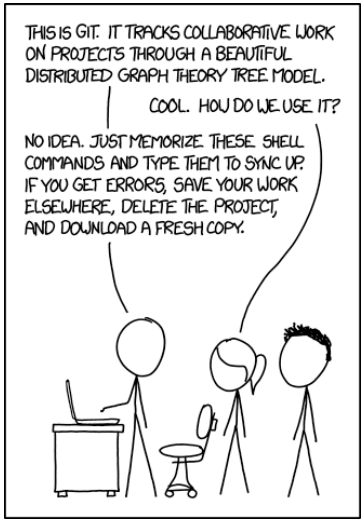


operator hidden master signer public slaves cesnet

Managing zone files using Git

- functionality can be extended by *hooks*
- *hooks* are always local, cannot be enforced
- RIPE NCC uses shell scripts
- similar project GitZone for managing DNS records
  - implemented in Perl
  - combination of DNS zone and Git repository management
  - discovered too late; NIH syndrome ☺

If that doesn't fix it, git.txt contains the phone number of a friend of mine who understands git. Just wait through a few minutes of 'It's really pretty simple, just think of branches as…' and eventually you'll learn the commands that will fix everything.



Source: xkcd 1597, CC-BY-NC

Checking zone file errors

Tool `/usr/sbin/named-compilezone`

- part of BIND server
- reads zone file in the same way BIND reads it
- outputs cannonical zone data
- compilation errors or serial number on stderr

1. compile zone during `pre-commit` hook
2. if it fails, refuse commit
3. otherwise compile the HEAD version
4. if the canonical zones differ and serial hasn't been updated, refuse commit
   - you can freely change comments/whitespace/ordering

cesnet

Getting the zone name

- `named-compilezone` requires zone name
- zone name = file name is a good idea, unless you strictly follow BCP 20
  - slash in classless reverse zone name
  - actually not required or recommended by BCP 20
  - we've been already doing it like this
- you can put `$ORIGIN` to the beginning of zone file

BCP 20 - Classless IN-ADDR.ARPA delegation

The examples here use "/" because it was felt to be more visible and pedantic reviewers felt that the 'these are not hostnames' argument needed to be repeated. We advise you not to be so pedantic, and to not precisely copy the above examples, e.g. substitute a more conservative character, such as hyphen, for "/".

Google shell style guide

If you are writing a script that is more than 100 lines long, you should probably be writing it in Python instead. Bear in mind that scripts grow. Rewrite your script in another language early to avoid a time-consuming rewrite at a later date.

My own tool dzonegit

- Python 3.5+, 520 sloc + 365 sloc of tests
- no Python dependecies, can be run as a script
- MIT licence, universal

cesnet

User's guide

Simple installation

```
$ wget -O .git/hooks/pre-commit https://…/dzonegit.py
$ chmod +x .git/hooks/pre-commit
```

Full installation (preferred for server deployments)

```
$ python3 -m venv .venv
$ source .venv/bin/activate
(.venv)$ pip install dzonegit
(.venv)$ deactivate
$ ln -s ../../.venv/bin/dzonegit-pre-commit \
        .git/hooks/pre-commit
```

Requirements: Python 3.5+, Git and `named-compilezone`

cesnet

Server-side setup

- *bare* repository with external working tree
  - `git push` to checked-out branch of non-bare repository is not recommended
  - you don't want to have `.git` in your working tree[1]
- Gitolite can be used as the Git repository manager with custom *hooks*

  `dzonegit-pre-receive` refuses commits that fail the tests

  `dzonegit-post-receive` checks out the external working copy, generate DNS server snippets, reload/reconfigure DNS server on demand
- configurable parameters are stored in the Git repository configuration
  - where to put external working tree
  - where is configuration snippet template
  - what to call for reload/reconfig

[1] no problem in this case but watch out for web deployments.

cesnet

dzonegit tool conclusion

- set of Git *hooks*, run during Git operations
- refuses commits with broken zone files
- published zone files and config snippets after each change
- calls *reload* command for every modified zone
- calls *reconfig* command if zone is introduced or deleted
- supports multiple repositories – *blacklists* a *whitelist* of zone names
  - to avoid duplicate zone definition in DNS server config

cesnet

DNSSEC signing

cesnet

Basic requirements

- use TSIG-protected zone transfer for input/output
- no need for HSM (similar to SSH and TLS keys)
- primary data needs **similar protection as the keys**
- sharing keys between zones should be rather avoided
  - not a standard and well tested feature
  - high number of keys is not a problem
  - no operational issues with registry-side CDS/CDNSKEY support

cesnet

Connecting the signer

```
repo dns-masters
    …
    config dzonegit.conffiletemplate2 = /etc/dzonegit/knot.json
    config dzonegit.conffilepath2 = /var/lib/dzonegit/masters-knot.conf
    config dzonegit.reconfigcmd2 = "/usr/sbin/knotc reload"
```

Config template

```
{
  "header": "# Autogenerated by dzonegit on $datetime.\n\nzone:\n",
  "item": " - domain: \"$zonename\"\n   $zonevar\n",
  "zonevars": {
    "ces.net": "template: signed\n   dnssec-policy: rsa",
    "rpz.cesnet.cz": "template: unsigned",
    "*.cz": "template: signed\n   dnssec-policy: ecdsa_cz",
    "*.ip6.arpa": "template: signed\n   dnssec-policy: ecdsa"
  }
}
```

Generated snippet

```
# Autogenerated by dzonegit on Fri Oct  5 14:43:58 2018. Do not edit.
zone:
 - domain: "8.b.d.0.1.0.0.2.ip6.arpa"
   template: signed
   dnssec-policy: ecdsa
 - domain: "rpz.cesnet.cz"
   template: unsigned
 - domain: "ces.net"
   template: signed
   dnssec-policy: rsa
```

cesnet

Future work

cesnet

Secure delegation automation

- works well for `.cz`[2]
- can be implemented on our side for other registries
  - reverse zones delegated from RIPE NCC
  - other TLDs via registrar API (*which registrar provides such?*)
- dealing with **subdomains of our own zones**
  - earlier this year, first customer requested secure reverse zone delegation
  - automating DS updates leads to automated editting of the zone files

[2] `.ch`, `.li` getting support now

cesnet

RIPE-DB DS updater prototype

- dedicated `mntner` with API access
- could implement `CSYNC` as well
- no need to care for insecure → secure bootstrapping
  - you have to add `mntner` and first `DS` rdata

Workflow

1. REST query for maintained `domain` objects
2. DNSSEC-aware resolver query for CDS record of each domain
3. check if DNSSEC signature inception is later that `last-modified` attribute of the `domain` object (replay-attack protection)
4. REST update of the `domain` object

Conclusion

- new signer and Git-based management deployed in September 2018
- already migrated `.cz` zones from shared RSA keys to separate ECDSA keys
- issues and pull requests are welcome

`https://github.com/oskar456/dzonegit`

cesnet

Any questions?

Ondřej Caletka
Ondrej.Caletka@cesnet.cz
https://**Ondřej.Caletka.cz**

Backup slides

cesnet

Gitolite setup example

```
repo dns-masters
    RW+     =   @masters
    option hook.pre-receive = dzonegit-pre-receive
    option hook.post-receive = dzonegit-post-receive
    config dzonegit.checkoutpath = /var/lib/dzonegit/dns-masters/
    config dzonegit.conffiletemplate = /etc/dzonegit/template-bind.json
    config dzonegit.conffilepath = /var/lib/dzonegit/masters-bind.conf
    config dzonegit.reconfigcmd = "/usr/sbin/rndc reconfig"
    config dzonegit.zonereloadcmd = "/usr/sbin/rndc reload"
```

Config snippet template

```
{
  "header": "# Autogenerated by dzonegit on $datetime. Do not edit.\n",
  "item": "zone \"$zonename\" { type master; file \"$zonefile\"; };"
}
```

Gitolite integration of dzonegit

$UNIXTIME support

- no need to increase serial manually
- implemented by Git clean/smudge filters
  - smudge filter files during checkout
  - clean filter files during commit
- you just need to setup *smudge* filter on the server side

Setting up *smudge* filter

```
$ cat repositories/dns-masters.git/info/attributes
*.zone filter=dzonegit
$ git config --global -l
filter.dzonegit.smudge=/usr/local/bin/dzonegit-smudge-serial
```

cesnet

Knot DNS as on-slave signer config

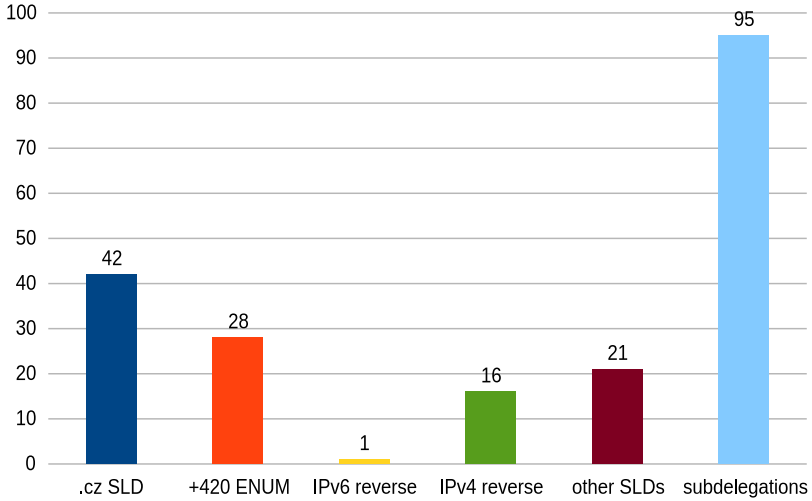
```
template:
  - id: default
    storage: "/var/lib/knot"
    zonefile-load: none
    zonefile-sync: -1
    journal-content: all
    master: master
    acl: acl_master  # NOTIFY from master
    acl: acl_slave   # AXFR from slave
    notify: slave
    dnssec-signing: on
    dnssec-policy: ecdsa_cz
```

```
policy:
  - id: ecdsa_cz
    algorithm: ecdsap256sha256
    zsk-lifetime: 30d
    ksk-lifetime: 90d
    rrsig-lifetime: 30d
    rrsig-refresh: 15d
    nsec3: on
    ksk-submission: resolvers

submission:
  - id: resolvers
    parent: ns.cesnet.cz
    parent: google_dns
    parent: cloudflare_dns
    check-interval: 61m
```

cesnet

Zones hosted at CESNET



cesnet

Support for `$INCLUDE`

- not supported so far
- can be part of the solution for keeping the DS records
- requires major refactoring
  - *checkout* the repository to a temporary directory
  - detect, which zone was changed with change of any file

Avoid pre-commit checks on the working tree

Version to be commited is created **during `git add`**. At the time `git commit` is called, the working tree can contain different data.

cesnet