Extreme DNS

Fundamental Internet Applications

Pierre-Philipp Braun <pbraun@nethence.com>

Table of contents

Name Resolution

```
 (\(~)/)
  )@_@(  #
 ((q_p))'
 /\|U|/\
/  `='  \
```

Static vs. dynamic

- static with `/etc/hosts`
- vs. dynamic (NIS+, DNS, NetBIOS-NS, ...)
- define which one(s) to use in `/etc/nsswitch.conf`

DNS

yet another L7 protocol

Everybody clear on what it does?...

==> The principle should be clear already for 3rd year bachelors

It binds / maps IPs with names so you can call e.g.

```
http://domain.tld/
```

instead of

```
http://1.2.3.4/
```

DNS client setup

- directly with `/etc/resolv.conf`
- –or– by means of a stub-resolver with caching

Stub-resolver products

- resolvconf + dnsmasq
- systemd-resolved

Full-blown DNS server products

Popular ones

- ISC BIND - can do everything
- NLnet Labs NSD - authoritative only
- NLnet Labs Unbound - forwarding only & cache
- Knot DNS - authoritative only

Authoritative (server conf points to zone-file)

```
vi /var/chroot/nsd/etc/nsd/nsd.conf

zone:
    name: ”os3.su”
    zonefile: ”%s.db”
```

DNS records (zone-file format)

```
vi /var/chroot/nsd/etc/os3.su.db

$ORIGIN os3.su.
$TTL 1800

@                IN NS          ns
@                IN NS          ns2
ns               IN A           62.210.110.7
ns2              IN A           62.210.16.8

@                IN A           62.210.110.7
*                IN A           62.210.110.7

@                IN MX 5        mx
mx               IN A           188.130.155.139

some-host        IN A           x.x.x.x
npf              IN CNAME       some-host
```

DNAME example

redhat got bought by IBM, right? now imagine they want to get rid of the name

```
$ORIGIN redhat.com.

@       IN DNAME redhat.ibm.com.
```

now `anything.redhat.com` goes and resolves `anything.redhat.ibm.com`.

Authoritative features

- Delegations
- master-slave with XFR & notify
- DNSSEC island vs full chain of trust
  - Unbound possibly validating
  - still optional…
- alternatives to DNSSEC
  - DNS over HTTPS (DoH)
  - DNS over TLS (DoT)

How a forwarder works

non-authoritative

It does iterative queries (so you can do recursive queries on him)

```
cat /usr/share/dns/root.hints
```

```
.
net.
online.net.
```

DNS queries

iterative vs. recursive

```
host nethence.com
host nethence.com 8.8.8.8
host -r # non-recursive query

dig nethence.com +short
dig nethence.com +short @8.8.8.8
# +[no]recurse
# +[no]trace
```

Root servers

The 13 root name servers are operated by 12 independent organisations

Are some in Russia (not counting Belarus & friends)?...

==> Yes, those are spread everywhere now. As of Feb 2021 in Russia we've got

```
E – NASA Ames Research Center – 1 in Moscow
F – Internet Systems Consortium, Inc. ==> 2 in Moscow + 1 St-
Peter
J – Verisign, Inc. ==> 1 in Moscow + 1 St-Peter
K – RIPE NCC ==> 1 in Moscow + 1 St-Peter + 1 Novosibirsk
L – ICANN ==> 3 in Moscow + 1 St-Peter
I – Netnod ==> 1 St-Peter
```

Super-duper server for Siberia

```
Novosibirsk, RU
Operator    RIPE NCC
IPv4    193.0.14.129
IPv6    2001:7fd::1
ASN 25152
```

Recursive queries

Old-school client setup

```
vi /etc/resolv.conf

    nameserver ...
```

–or– new-school stub-resolvers

```
...
```

–or– validating-resolver on localhost!

```
vi /etc/unbound/unbound.conf

    forward-zone:
    name: ”.”
    forward-addr: x.x.x.x@53
```

Why a caching forwarder is a good thing to have

- saves some traffic (if not bandwidth)
- safer / internal
- possibly also a DNSSEC **validating resolver**

```
  (\(~)/)
   )@_@(   #
  ((q_p))'
  /\|U|/\
/   `='   \
```

// Questions on name services?…

Quick checkup

What kind of server is NSD?…

What kind of server is Unbound?…

What kind of server is BIND?…

==> NSD is an authoritative-only server

==> Unbound is a forwarding & caching server

- does iterative queries by itself
- and delivers a service for recursive queries to happen

==> BIND can do anything

Clear on DNS records?

- NS
- A
- CNAME & DNAME
- MX & priority – which (preferably) needs an A record

BONUS // deep-dive RFCs and try and evaluate use-cases where an MX with a CNAME record would be a problem.

Short form is nicer

zone file starts e.g.

```
$ORIGIN example.net.
$TTL 1800
```

you then can omit the domain!

```
host                IN A x.x.x.x
```

even in the reverse situation e.g. pointing to `mx.example.net`!

```
@                   IN MX 5 mx
```

Extreme DNS

NS record

```
@    IN NS   xc
@    IN NS   nssec.online.net. ;62.210.16.8

xc   IN A    62.210.110.7
```

now how to handle delegation?

==> just define an NS record for the sub-domain

```
lab IN NS   <NAME-SERVER-ADDRESS>
lab IN NS   <BACKUP-NAME-SERVER-ADDRESS>
```

What about the IP, do we need to define it here?...

==> Two delegation use-cases

- we’re delegating a subdomain that will be handled by another hostmaster@
- we’re delegating a subdomain to an NS that belongs to itself (e.g. let your research lab handle its own name resolution service)

Another hostmaster@

We want to delegate `lab.example.net` to foreign-company's NS

```
lab IN NS   ns1.foreign-company.net.
lab IN NS   ns2.foreign-company.net.
```

…problem solved already!

NS belongs to itself

HERE COMES THE GLUE

```
$ORIGIN example.net.

lab IN NS   ns1.lab.example.net.
lab IN NS   ns2.lab.example.net.

ns1.lab IN A    x.x.x.x
ns2.lab IN A    x.x.x.x
```

Otherwise the client wouldn’t know where to find the targetted domain’s NS

Wildcards on steroids

The usual stuff e.g.

```
@                IN A 62.210.110.7
*                IN A 62.210.110.7
```

Wildcard is one level only e.g.

```
meet             IN CNAME france1
*.meet           IN CNAME france1
*.auth.meet      IN CNAME france1
```

SPF

```
@                IN TXT ”v=spf1 mx a –all”
*                IN TXT ”v=spf1 mx a –all”
```

LAB // anything there?…

CAA

```
@                IN CAA 128 issue ”letsencrypt.org”
@                IN CAA 128 iodef ”mailto:pbraun@nethence.com”
*                IN CAA 128 issue ”letsencrypt.org”
*                IN CAA 128 iodef ”mailto:pbraun@nethence.com”
```

Even more exotic food, really?

TLSA usage 0 (PKIX-TA) allowing LE here

```
_25._tcp.slackmx      IN TLSA 0 1 1 60b87575447dcba2a36b7d11ac09fb24a9
_993._tcp.slackmx     IN TLSA 0 1 1 60b87575447dcba2a36b7d11ac09fb24a9
_443._tcp           IN TLSA 0 1 1 60b87575447dcba2a36b7d11ac09fb24a9db4
_443._tcp.pub         IN TLSA 0 1 1 60b87575447dcba2a36b7d11ac09fb24a9d
```

DKIM

```
sep2020._domainkey IN TXT ”v=DKIM1; k=rsa; ” ”p=MIGfMA0GCSqGSIb3DQEBAQ
```

DMARC

```
_dmarc          IN TXT ”v=DMARC1; p=none”
```

Aren’t you sick already?

```
_mta-sts      IN TXT ”STARTTLS is enforced anyhow (https://nethence.cor
_smtp._tls     IN TXT ”v=TLSRPTv1; rua=mailto:abuse@nethence.com”
```

PoCs & projects - don’t worry be happy

you’re in a Master program

What happens in case Russia decides to cut-off the rest of the world?…

LAB // what happens in case of a split? PoC a root-server farm and simulate a fork

DNS-based CDN

- tried BIND with GeoIP
- starting to make the PoC with a friend
- got a connection from `8.8.8.8` in the logs

Any idea what's the problem here?...

And so guess what happens?...

==> Yeah, that's a well known open DNS proxy

==> There's no way of knowing what country the client is from just by means of DNS

LAB // discuss and try to solve the issue – maybe by mixing both the end-protocol for GeoIP when DNS-GeoIP fails?

More products

- (major authoritative ones where BIND / NSD / Knot)
- djbdns / tinydns
- *other full-blown and/or tiny DNS daemons out there?*

LAB // benchmark DNS servers and emphasis on possible advantages / exclusive features

Rare vuln with djbdns

LAB // dig in, study and describe the vuln djbdns had

Many vulns with dnsmasq

LAB // PoC the latest dnsmasq vulns

- check-out available PoCs and tools
- try to understand what the hell in going-on
- attempt to exploit it yourself

DHCP - DNS updates

- it's hard to compete with MS AD on this front
- even the workstation's hostname can resolve on the network
- need to enable DynDNS updates

LAB // setup DHCP and a DNS server so discovered hosts start to resolve already

// Questions on extreme DNS and project ideas?

PoCs & Research

LAB // Successfully PoC Kaminsky dns poisoning attack