Unbound

a caching, validating DNSSEC resolver

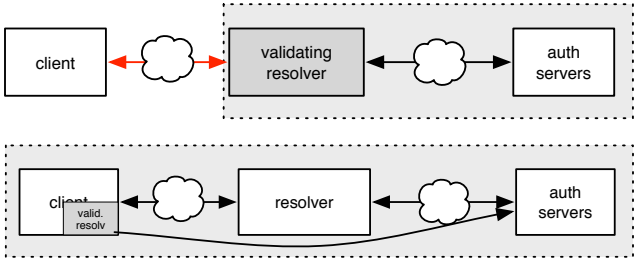
UKUUG Spring 2011 Conference
Leeds, UK
March 2011

Jan-Piet Mens

```
$ dig 1.1.0.3.3.0.8.1.7.1.9.4.e164.arpa naptr
```

Do you trust your name server?

- DNS clients typically trust the name servers they use
- But they need not be trustworthy
  - Rogue DHCP server hands out resolv.conf pointing to pirates
  - Attackers can take over networks (think WiFi in hotels)
  - Viruses/trojans can alter local configuration
- In all cases:
  - We loose control over DNS replies
- Install a validating DNS resolver "close" to applications



Unbound as a DNS cache (SEC-less)

- Unbound is a secure, caching-only, portable DNS server
- Maintained by NLNetlabs under BSD license
- Designed with DNSSEC and IPv6 from the ground up
- Trusts nothing
- Good security
- Many "distros" have packages
  - I recommend newest version

  `http://unbound.net/`
- Lightweight, fast, and easy to configure
- No split-personalities
- (And I was first to write about Unbound :-)

Configuration

- One file (but we'll add more later on)

```
$ cat /etc/unbound/unbound.conf
server:
    access-control: 127.0.0.1/8 allow
    verbosity: 1
```

- Launch unbound (and watch your syslog)

```
# unbound
```

- Query it

```
$ dig +short @127.0.0.1 ukuug.jpmens.org txt
"Leeds"
```

- That's it!

Configure unbound-control

- Configure unbound-control

```
unbound-control-setup
```

  - Generates certificates

- Enable in unbound.conf

```
remote-control:
  control-enable: yes
```

- Restart unbound and test

```
$ unbound-control status
version: 1.4.8
verbosity: 1
threads: 1
modules: 2 [ validator iterator ]
uptime: 252 seconds
unbound (pid 9331) is running...
```

Your workstation

- Ensure unbound is running
- Configure your workstation to use it!

```
$ cat /etc/resolv.conf
nameserver 127.0.0.1
```

- DHCP on Linux

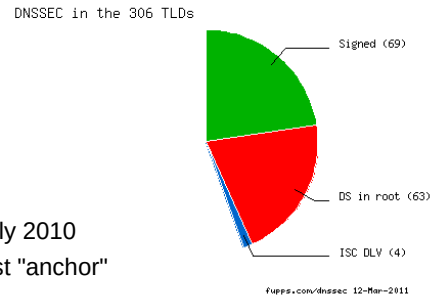
```
$ cat dhclient.conf
...
prepend domain-name-servers 127.0.0.1;
```

- Mac?

```
scutil
```

The Chain of Trust

- The root zone's DNSKEY record is well known
- Establish a chain of trust from the root to any signed zone
- Each link validates the next
- Parent's DS record validates child zone's DNSKEY
- A child's DS record in parent is signed by private key of parent



- Chain of trust
  - root zone signed in July 2010
  - validation starts at trust "anchor"

Enable DNSSEC validation

- Needs root DNSSEC trust anchor
- unbound-anchor utility retrieves root zone's DNSSEC key securely

```
$ unbound-anchor -a /etc/unbound/root.key
```

- Configure trust anchor in unbound.conf

```
auto-trust-anchor-file: "/etc/unbound/root.key"
```

- Ensure unbound-anchor in start-up scripts
- Reload

```
$ unbound-control reload
```

- Did that work?

```
$ dig +dnssec @127.0.0.1 ukuug.jpmens.org txt
```

dig

- dig +dnssec and watch for AD flag indicating successful validation

```
$ dig +dnssec @127.0.0.1 ukuug.jpmens.org txt
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, [...]
```

- Invalid or bogus DNSSEC data will not be returned

```
$ dig +dnssec @127.0.0.1 www.dnssec-failed.org
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 3
```

dig (cont'd)

- CD flag indicates application wants to validate itself

```
$ dig +cd +dnssec @127.0.0.1 www.dnssec-failed.org
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY:

;; ANSWER SECTION:
www.dnssec-failed.org. 7200 IN A 68.87.64.48
```

Unbound logging

- Enable more logging in Unbound

```
val-log-level: 2
```

  - [9331:0] info: validation failure <www.dnssec-failed.org. A IN>: signature expired from 68.87.72.244 for key dnssec-failed.org. while building chain of trust

- Is that true?

```
$ dig +multiline +cd +dnssec @127.0.0.1 \
        www.dnssec-failed.org rrsig
;; ANSWER SECTION:
www.dnssec-failed.org. 7019 IN RRSIG \
    A 5 3 7200 20090201000000 (
    20090101000000 48621 dnssec-failed.org.
    gM8IbzE3N4xx4DQog+W2UvY+BwnLIJojFmuQUdUb7FAm
    wtD3k673q+005FDCW8xf88b+9QtvslrpNyi5ZLUq4v9k
    Xdya9Je0O2ByYjfrgjYqk4Qu37lfPe+iGvl9aSSMyGeu
    UHv9NWWY10nXjCp2rTdCSpXc7xt3CSMW7pFNFg0= )
```

Configure DLV

- DNS Look-aside Validation
- We need the DLV DNSKEY record

```
$ dig +dnssec @127.0.0.1 dlv.isc.org dnskey
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 4, ...
;; ANSWER SECTION:
dlv.isc.org.  7200  IN  DNSKEY  257 3 5 BEAAAAPHMu/5...
dlv.isc.org.  7200  IN  DNSKEY  256 3 5 BEAAAAOlYGw5...
dlv.isc.org.  7200  IN  RRSIG  DNSKEY 5 3 7200 20110...
dlv.isc.org.  7200  IN  RRSIG  DNSKEY 5 3 7200 20110...
```

- Copy the SEP DNSKEY (KSK) into a file dlv.key

```
| grep -P 'DNSKEY\s+257'
```

- Add it to Unbound

```
dlv-anchor-file: "dlv.key"
```

DLV (cont'd)

- Unbound will then
  - Check for "manual" trust anchors in the configuration
  - Get trust from delegation tree (DS in parent)
  - Try to look up trust in DLV zone
- DNSSEC validation via parent (i.e. root) has priority over DLV

Browser tests

- DNSSEC Validator for Firefox

```
http://www.dnssec-validator.cz/
```

- DNSSEC or not?

```
http://dnssec-or-not.net
http://dnssectest.sidn.nl/
```

DNSSEC testbed (or your own)

- Set up your own trust anchors in a file "my.keys", which contains DS or DNSKEY records

```
p0000.aa IN DS 47534 8 1 74526d3f57...
p0000.aa IN DS 47534 8 2 82512fb4ad...
de.  86400 IN DNSKEY 257 3 8 AwEAAZ1FqQED8QBrk3Jk4q96lg
example.com IN DS 47534 8 3 296fc89ee0...
```

- Configure keys into Unbound

```
trust-anchor-file: "/etc/unbound/my.keys"
```

- Alternatively use trust-anchor configuration statements
- Configure the zone

```
stub-zone:
   name: "de"
   stub-addr: 81.91.161.228 # auth-fra.dnssec.denic.de
   stub-addr: 87.223.175.25 # auth-ams.dnssec.denic.de
```

Serve local data

- Unbound can serve "local" data to its clients
- For example, a static "zone":

```
local-zone: "ukuug." static
local-data: "beamer.ukuug. IN A 192.168.1.12"
local-data: 'paul.ukuug. TXT "Hi Paul!"'
local-data-ptr: "192.168.1.12 beamer.ukuug"
```

- Will it work?

```
$ dig +short @127.0.0.1 paul.ukuug  txt
"Hi Paul!"
```

- No DNSSEC
- But local data can be added on-the-fly with unbound-control

More local data

- Override a single name (all others resolved normally)

```
local-data: "foo.jpmens.org A 127.0.0.1"
```

- Redirect a whole domain to an IP

```
local-zone: "example.aa" redirect
local-data: "example.aa A 127.0.0.9"
```

Forwarding

- Unbound on workstation behind corporate DNS?

```
forward-zone:
   name: "."
   forward-addr: 192.168.1.20
```

- Upstream server must be DNSSEC-enabled
  - Unbound is by default
  - For BIND you need to configure dnssec-enable

Advanced topics

- There's an optional Python module built into Unbound
  - Full control over DNS queries sent out by Unbound
  - Full Control over DNS replies returned to Unbound clients
  - Prototyping
- Gather Unbound statistics

```
http://unbound.net/documentation/howto_statistics.html
```

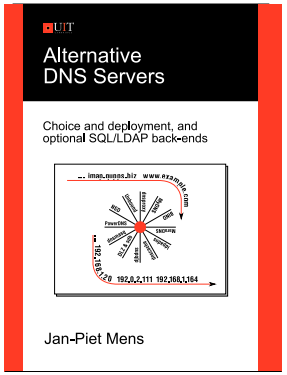
- Wrap a resolver into your own application with libunbound

```
http://unbound.net/documentation/libunbound.html
```

Further reading

- Alternative DNS Servers, UIT, 2009, Jan-Piet Mens

```
http://mens.de/:/altdns
```



Thank you

Questions?

Whoami

```
$ dig 1.1.0.3.3.0.8.1.7.1.9.4.e164.arpa naptr

;; ANSWER SECTION:
1.1.0.3.3.0.8.1.7.1.9.4.e164.arpa. 3575  IN NAPTR 3 10 "u" "E2U+http" "!^.*$!http:mens.de!" .
1.1.0.3.3.0.8.1.7.1.9.4.e164.arpa. 3575  IN NAPTR 3 20 "u" "E2U+http" "!^.*$!http:blog.fupps.com!" .
1.1.0.3.3.0.8.1.7.1.9.4.e164.arpa. 3575  IN NAPTR 4 10 "u" "E2U+mailto" "!^.*$!mailto:jp@mens.de!" .
1.1.0.3.3.0.8.1.7.1.9.4.e164.arpa. 3575  IN NAPTR 1 10 "u" "E2U+sip" "!^.*$!sip:5552064@sipgate.de!" .
1.1.0.3.3.0.8.1.7.1.9.4.e164.arpa. 3575  IN NAPTR 2 10 "u" "E2U+tel" "!^.*$!tel:+491718033011!" .
```