Logiciel

Unbound : un serveur DNS local

Unbound est un serveur de noms DNS, c'est à dire qu'il traduit un nom de domaine en adresse IP. [1)]

Il fait cette traduction :

- soit d'après sa propre config,
- soit en faisant appel à d'autres serveurs DNS.

Dans un souci d'efficacité, il peut mettre les résultats en cache.

Nous utiliserons Unbound en tant que serveur DNS sur le réseau local pour définir des domaines factices (comme on le ferait avec un fichiers hosts, mais en plus poussé et automatisé).

Le serveur **Unbound** sera installé sur un serveur du réseau local (un Raspberry Pi dans cet exemple).

Pré-requis

- un réseau local.
- et une connexion à Internet active pour les tests.

Installation sur la machine du serveur

Sous Linux : installez les paquets **unbound,dnsutils,ldnsutils** ou en ligne de commande (comme sur un Raspberry Pi via SSH) :

```
...@...:~$ sudo apt install unbound dnsutils ldnsutils
```

Sous Windows : Téléchargez l'installateur **unbound_setup_x.x.x.exe** sur la page http://unbound.net/download.html et lancez-le.

Configuration

Sous Linux

Les fichiers de configuration (nommés **<xxx>.conf**) sont désormais placés dans le répertoire **/etc/unbound/unbound.d/**.

Ainsi, le fichier **/etc/unbound/unbound.conf** reste

inchangé.

Voici un exemple de fichier **local.conf** pour un serveur DNS avec déclaration automatique des sous-domaines :

/etc/unbound/unbound.conf.d/local.conf

```
server:
    #verbosity: 1

    # Répondre aux requêtes DNS sur
toutes les interfaces réseau.
    interface: 0.0.0.0
    port: 53

    # ouverture à tout le monde
    access-control: 0.0.0.0/0 allow
    chroot: ""

    # emplacement du fichier de log
    logfile: "/var/log/unbound.log"
    # je ne souhaite pas "pourrir"
le syslog
    use-syslog: no

    # zone *.chateau
    local-zone: "chateau." redirect
    local-data: "chateau. IN A
127.0.0.1"

    # zone *.framboise
    local-zone: "framboise."
redirect
    local-data: "framboise. IN A
127.0.0.1"

# Utilisation du DNS "normal" (ici,
celui de la box) pour tout le reste
forward-zone:
    name: "."
    forward-addr: 192.168.0.254
```

Pour les options, voir la page unbound.conf(5) - page de man.

Fichier exemple fourni avec unbound

Voici le fichier exemple fourni avec unbound :

/usr/share/doc/unbound/examples/unbound.conf

```
#
# Example configuration file.
#
# See unbound.conf(5) man page,
version 1.4.22.
#
# this is a comment.

#Use this to include other text
into the file.
#include: "otherfile.conf"

# The server clause sets the main
parameters.
server:
    # whitespace is not necessary,
but looks cleaner.

    # verbosity number, 0 is least
verbose. 1 is default.
    verbosity: 1

    # print statistics to the log
(for every thread) every N seconds.
    # Set to "" or 0 to disable.
Default is disabled.
    # statistics-interval: 0

    # enable cumulative statistics,
without clearing them after
printing.
    # statistics-cumulative: no

    # enable extended statistics
(query types, answer codes, status)
    # printed from unbound-control.
default off, because of speed.
    # extended-statistics: no

    # number of threads to create.
1 disables threading.
    # num-threads: 1

    # specify the interfaces to
answer queries from by ip-address.
```

```
    # The default is to listen to
localhost (127.0.0.1 and ::1).
    # specify 0.0.0.0 and ::0 to
bind to all available interfaces.
    # specify every
interface[@port] on a new
'interface:' labelled line.
    # The listen interfaces are not
changed on reload, only on restart.
    # interface: 192.0.2.153
    # interface: 192.0.2.154
    # interface: 192.0.2.154@5003
    # interface: 2001:DB8::5

    # enable this feature to copy
the source address of queries to
reply.
    # Socket options are not
supported on all platforms.
experimental.
    # interface-automatic: no

    # port to answer queries from
    # port: 53

    # specify the interfaces to
send outgoing queries to
authoritative
    # server from by ip-address. If
none, the default (all) interface
    # is used. Specify every
interface on a 'outgoing-
interface:' line.
    # outgoing-interface:
192.0.2.153
    # outgoing-interface:
2001:DB8::5
    # outgoing-interface:
2001:DB8::6

    # number of ports to allocate
per thread, determines the size of
the
    # port range that can be open
simultaneously.  About double the
    # num-queries-per-thread, or,
use as many as the OS will allow
you.
    # outgoing-range: 4096

    # permit unbound to use this
```

```
port number or port range for
    # making outgoing queries,
using an outgoing interface.
    # outgoing-port-permit: 32768

    # deny unbound the use this of
port number or port range for
    # making outgoing queries,
using an outgoing interface.
    # Use this to make sure unbound
does not grab a UDP port that some
    # other server on this computer
needs. The default is to avoid
    # IANA-assigned port numbers.
    # If multiple outgoing-port-
permit and outgoing-port-avoid
options
    # are present, they are
processed in order.
    # outgoing-port-avoid:
"3200-3208"

    # number of outgoing
simultaneous tcp buffers to hold
per thread.
    # outgoing-num-tcp: 10

    # number of incoming
simultaneous tcp buffers to hold
per thread.
    # incoming-num-tcp: 10

    # buffer size for UDP port 53
incoming (SO_RCVBUF socket option).
    # 0 is system default.  Use 4m
to catch query spikes for busy
servers.
    # so-rcvbuf: 0

    # buffer size for UDP port 53
outgoing (SO_SNDBUF socket option).
    # 0 is system default.  Use 4m
to handle spikes on very busy
servers.
    # so-sndbuf: 0
    # on Linux(3.9+) use
SO_REUSEPORT to distribute queries
over threads.
    # so-reuseport: no

    # EDNS reassembly buffer to
```

```
advertise to UDP peers (the actual
buffer
    # is set with msg-buffer-size).
1480 can solve fragmentation
(timeouts).
    # edns-buffer-size: 4096

    # Maximum UDP response size
(not applied to TCP response).
    # Suggested values are 512 to
4096. Default is 4096. 65536
disables it.
    # max-udp-size: 4096

    # buffer size for handling DNS
data. No messages larger than this
    # size can be sent or received,
by UDP or TCP. In bytes.
    # msg-buffer-size: 65552

    # the amount of memory to use
for the message cache.
    # plain value in bytes or you
can append k, m or G. default is
"4Mb".
    # msg-cache-size: 4m

    # the number of slabs to use
for the message cache.
    # the number of slabs must be a
power of 2.
    # more slabs reduce lock
contention, but fragment memory
usage.
    # msg-cache-slabs: 4

    # the number of queries that a
thread gets to service.
    # num-queries-per-thread: 1024

    # if very busy, 50% queries run
to completion, 50% get timeout in
msec
    # jostle-timeout: 200
    # msec to wait before close of
port on timeout UDP. 0 disables.
    # delay-close: 0

    # the amount of memory to use
for the RRset cache.
    # plain value in bytes or you
```

```
can append k, m or G. default is
"4Mb".
    # rrset-cache-size: 4m

    # the number of slabs to use
for the RRset cache.
    # the number of slabs must be a
power of 2.
    # more slabs reduce lock
contention, but fragment memory
usage.
    # rrset-cache-slabs: 4

    # the time to live (TTL) value
lower bound, in seconds. Default 0.
    # If more than an hour could
easily give trouble due to stale
data.
    # cache-min-ttl: 0

    # the time to live (TTL) value
cap for RRsets and messages in the
    # cache. Items are not cached
for longer. In seconds.
    # cache-max-ttl: 86400

    # the time to live (TTL) value
for cached roundtrip times,
lameness and
    # EDNS version information for
hosts. In seconds.
    # infra-host-ttl: 900

    # the number of slabs to use
for the Infrastructure cache.
    # the number of slabs must be a
power of 2.
    # more slabs reduce lock
contention, but fragment memory
usage.
    # infra-cache-slabs: 4

    # the maximum number of hosts
that are cached (roundtrip, EDNS,
lame).
    # infra-cache-numhosts: 10000

    # Enable IPv4, "yes" or "no".
    # do-ip4: yes

    # Enable IPv6, "yes" or "no".
```

```
    # do-ip6: yes

    # Enable UDP, "yes" or "no".
    # do-udp: yes

    # Enable TCP, "yes" or "no".
    # do-tcp: yes

    # upstream connections use TCP
only (and no UDP), "yes" or "no"
    # useful for tunneling
scenarios, default no.
    # tcp-upstream: no

    # Detach from the terminal, run
in background, "yes" or "no".
    # do-daemonize: yes

    # control which clients are
allowed to make (recursive) queries
    # to this server. Specify
classless netblocks with /size and
action.
    # By default everything is
refused, except for localhost.
    # Choose deny (drop message),
refuse (polite error reply),
    # allow (recursive ok),
allow_snoop (recursive and
nonrecursive ok)
    # deny_non_local (drop queries
unless can be answered from local-
data)
    # refuse_non_local (like
deny_non_local but polite error
reply).
    # access-control: 0.0.0.0/0
refuse
    # access-control: 127.0.0.0/8
allow
    # access-control: ::0/0 refuse
    # access-control: ::1 allow
    # access-control:
::ffff:127.0.0.1 allow

    # if given, a chroot(2) is done
to the given directory.
    # i.e. you can chroot to the
working directory, for example,
    # for extra security, but make
sure all files are in that
```

```
directory.
    #
    # If chroot is enabled, you
should pass the configfile (from
the
    # commandline) as a full path
from the original root. After the
    # chroot has been performed the
now defunct portion of the config
    # file path is removed to be
able to reread the config after a
reload.
    #
    # All other file paths (working
dir, logfile, roothints, and
    # key files) can be specified
in several ways:
    #     o as an absolute path
relative to the new root.
    #     o as a relative path to
the working directory.
    #     o as an absolute path
relative to the original root.
    # In the last case the path is
adjusted to remove the unused
portion.
    #
    # The pid file can be absolute
and outside of the chroot, it is
    # written just prior to
performing the chroot and dropping
permissions.
    #
    # Additionally, unbound may
need to access /dev/random (for
entropy).
    # How to do this is specific to
your OS.
    #
    # If you give "" no chroot is
performed. The path must not end in
a /.
    # chroot: "/etc/unbound"

    # if given, user privileges are
dropped (after binding port),
    # and the given username is
assumed. Default is user "unbound".
    # If you give "" no privileges
are dropped.
    # username: "unbound"
```

```
    # the working directory. The
relative files in this config are
    # relative to this directory.
If you give "" the working
directory
    # is not changed.
    # directory: "/etc/unbound"

    # the log file, "" means log to
stderr.
    # Use of this option sets use-
syslog to "no".
    # logfile: ""

    # Log to syslog(3) if yes. The
log facility LOG_DAEMON is used to
    # log to, with identity
"unbound". If yes, it overrides the
logfile.
    # use-syslog: yes

    # print UTC timestamp in ascii
to logfile, default is epoch in
seconds.
    # log-time-ascii: no
    # print one line with time, IP,
name, type, class for every query.
    # log-queries: no

    # the pid file. Can be an
absolute path outside of
chroot/work dir.
    # pidfile:
"/etc/unbound/unbound.pid"

    # file to read root hints from.
    # get one from
ftp://FTP.INTERNIC.NET/domain/named
.cache
    # root-hints: ""

    # enable to not answer
id.server and hostname.bind
queries.
    # hide-identity: no

    # enable to not answer
version.server and version.bind
queries.
    # hide-version: no
```

```
    # the identity to report. Leave
"" or default to return hostname.
    # identity: ""

    # the version to report. Leave
"" or default to return package
version.
    # version: ""

    # the target fetch policy.
    # series of integers describing
the policy per dependency depth.
    # The number of values in the
list determines the maximum
dependency
    # depth the recursor will
pursue before giving up. Each
integer means:
    #     -1 : fetch all targets
opportunistically,
    #     0: fetch on demand,
    #    positive value: fetch that
many targets opportunistically.
    # Enclose the list of numbers
between quotes ("").
    # target-fetch-policy: "3 2 1 0
0"

    # Harden against very small
EDNS buffer sizes.
    # harden-short-bufsize: no

    # Harden against unseemly large
queries.
    # harden-large-queries: no

    # Harden against out of zone
rrsets, to avoid spoofing attempts.
    # harden-glue: yes

    # Harden against receiving
dnssec-stripped data. If you turn
it
    # off, failing to validate
dnskey data for a trustanchor will
    # trigger insecure mode for
that zone (like without a
trustanchor).
    # Default on, which insists on
dnssec data for trust-anchored
zones.
```

```
    # harden-dnssec-stripped: yes

    # Harden against queries that
fall under dnssec-signed nxdomain
names.
    # harden-below-nxdomain: no

        # Harden the referral path
by performing additional queries
for
    # infrastructure data.
Validates the replies (if
possible).
    # Default off, because the
lookups burden the server.
Experimental
    # implementation of draft-
wijngaards-dnsext-resolver-side-
mitigation.
    # harden-referral-path: no

    # Use 0x20-encoded random bits
in the query to foil spoof
attempts.
    # This feature is an
experimental implementation of
draft dns-0x20.
    # use-caps-for-id: no

    # Enforce privacy of these
addresses. Strips them away from
answers.
    # It may cause DNSSEC
validation to additionally mark it
as bogus.
    # Protects against 'DNS
Rebinding' (uses browser as network
proxy).
    # Only 'private-domain' and
'local-data' names are allowed to
have
    # these private addresses. No
default.
    # private-address: 10.0.0.0/8
    # private-address:
172.16.0.0/12
    # private-address:
192.168.0.0/16
    # private-address:
169.254.0.0/16
    # private-address: fd00::/8
```

```
    # private-address: fe80::/10

    # Allow the domain (and its
subdomains) to contain private
addresses.
    # local-data statements are
allowed to contain private
addresses too.
    # private-domain: "example.com"

    # If nonzero, unwanted replies
are not only reported in
statistics,
    # but also a running total is
kept per thread. If it reaches the
    # threshold, a warning is
printed and a defensive action is
taken,
    # the cache is cleared to flush
potential poison out of it.
    # A suggested value is
10000000, the default is 0 (turned
off).
    # unwanted-reply-threshold: 0

    # Do not query the following
addresses. No DNS queries are sent
there.
    # List one address per entry.
List classless netblocks with
/size,
    # do-not-query-address:
127.0.0.1/8
    # do-not-query-address: ::1

    # if yes, the above default do-
not-query-address entries are
present.
    # if no, localhost can be
queried (for testing and
debugging).
    # do-not-query-localhost: yes

    # if yes, perform prefetching
of almost expired message cache
entries.
    # prefetch: no

    # if yes, perform key lookups
adjacent to normal lookups.
    # prefetch-key: no
```

```
    # if yes, Unbound rotates RRSet
order in response.
    # rrset-roundrobin: no

    # if yes, Unbound doesn't
insert authority/additional
sections
    # into response messages when
those sections are not required.
    # minimal-responses: no

    # module configuration of the
server. A string with identifiers
    # separated by spaces.
"iterator" or "validator iterator"
    # module-config: "validator
iterator"

    # File with trusted keys, kept
uptodate using RFC5011 probes,
    # initial file like trust-
anchor-file, then it stores
metadata.
    # Use several entries, one per
domain name, to track multiple
zones.
    #
    # If you want to perform DNSSEC
validation, run unbound-anchor
before
    # you start unbound (i.e. in
the system boot scripts).  And
enable:
    # Please note usage of unbound-
anchor root anchor is at your own
risk
    # and under the terms of our
LICENSE (see that file in the
source).
    # auto-trust-anchor-file:
"/etc/unbound/root.key"

    # File with DLV trusted keys.
Same format as trust-anchor-file.
    # There can be only one DLV
configured, it is trusted from root
down.
    # Download
http://ftp.isc.org/www/dlv/dlv.isc.
org.key
    # dlv-anchor-file:
```

```
"dlv.isc.org.key"

    # File with trusted keys for
validation. Specify more than one
file
    # with several entries, one
file per entry.
    # Zone file format, with DS and
DNSKEY entries.
    # Note this gets out of date,
use auto-trust-anchor-file please.
    # trust-anchor-file: ""
    # Trusted key for validation.
DS or DNSKEY. specify the RR on a
    # single line, surrounded by
"". TTL is ignored. class is IN
default.
    # Note this gets out of date,
use auto-trust-anchor-file please.
    # (These examples are from
August 2007 and may not be valid
anymore).
    # trust-anchor: "nlnetlabs.nl.
DNSKEY 257 3 5
AQPzzTWMz8qSWIQlfRnPckx2BiVmkVN6LPu
pO3mbz7FhLSnm26n6iG9N
Lby97Ji453aWZY3M5/xJBSOS2vWtco2t8C0
+xeO1bc/d6ZTy32DHchpW
6rDH1vp86Ll+ha0tmwyy9QP7y2bVw5zSbFC
refk8qCUBgfHm9bHzMG1U BYtEIQ=="
    # trust-anchor:
"jelte.nlnetlabs.nl. DS 42860 5 1
14D739EB566D2B1A5E216A0BA4D17FA9B03
8BE4A"

    # File with trusted keys for
validation. Specify more than one
file
    # with several entries, one
file per entry. Like trust-anchor-
file
    # but has a different file
format. Format is BIND-9 style
format,
    # the trusted-keys { name flag
proto algo "key"; }; clauses are
read.
    # you need external update
procedures to track changes in
keys.
    # trusted-keys-file: ""
```

```
    # Ignore chain of trust. Domain
is treated as insecure.
    # domain-insecure:
"example.com"

    # Override the date for
validation with a specific fixed
date.
    # Do not set this unless you
are debugging signature inception
    # and expiration. "" or "0"
turns the feature off. -1 ignores
date.
    # val-override-date: ""

    # The time to live for bogus
data, rrsets and messages. This
avoids
    # some of the revalidation,
until the time interval expires. in
secs.
    # val-bogus-ttl: 60

    # The signature inception and
expiration dates are allowed to be
off
    # by 10% of the signature
lifetime (expir-incep) from our
local clock.
    # This leeway is capped with a
minimum and a maximum.  In seconds.
    # val-sig-skew-min: 3600
    # val-sig-skew-max: 86400

    # Should additional section of
secure message also be kept clean
of
    # unsecure data. Useful to
shield the users of this validator
from
    # potential bogus data in the
additional section. All unsigned
data
    # in the additional section is
removed from secure messages.
    # val-clean-additional: yes

    # Turn permissive mode on to
permit bogus messages. Thus,
messages
    # for which security checks
```

```
failed will be returned to clients,
    # instead of SERVFAIL. It still
performs the security checks, which
    # result in interesting log
files and possibly the AD bit in
    # replies if the message is
found secure. The default is off.
    # val-permissive-mode: no

    # Ignore the CD flag in
incoming queries and refuse them
bogus data.
    # Enable it if the only clients
of unbound are legacy servers
(w2008)
    # that set CD but cannot
validate themselves.
    # ignore-cd-flag: no

    # Have the validator log failed
validations for your diagnosis.
    # 0: off. 1: A line per failed
user query. 2: With reason and bad
IP.
    # val-log-level: 0

    # It is possible to configure
NSEC3 maximum iteration counts per
    # keysize. Keep this table very
short, as linear search is done.
    # A message with an NSEC3 with
larger count is marked insecure.
    # List in ascending order the
keysize and count values.
    # val-nsec3-keysize-iterations:
"1024 150 2048 500 4096 2500"
    # instruct the auto-trust-
anchor-file probing to add anchors
after ttl.
    # add-holddown: 2592000 # 30
days

    # instruct the auto-trust-
anchor-file probing to del anchors
after ttl.
    # del-holddown: 2592000 # 30
days

    # auto-trust-anchor-file
probing removes missing anchors
after ttl.
```

```
    # If the value 0 is given,
missing anchors are not removed.
    # keep-missing: 31622400 # 366
days

    # the amount of memory to use
for the key cache.
    # plain value in bytes or you
can append k, m or G. default is
"4Mb".
    # key-cache-size: 4m

    # the number of slabs to use
for the key cache.
    # the number of slabs must be a
power of 2.
    # more slabs reduce lock
contention, but fragment memory
usage.
    # key-cache-slabs: 4

    # the amount of memory to use
for the negative cache (used for
DLV).
    # plain value in bytes or you
can append k, m or G. default is
"1Mb".
    # neg-cache-size: 1m

    # By default, for a number of
zones a small default 'nothing
here'
    # reply is built-in.  Query
traffic is thus blocked.  If you
    # wish to serve such zone you
can unblock them by uncommenting
one
    # of the nodefault statements
below.
    # You may also have to use
domain-insecure: zone to make
DNSSEC work,
    # unless you have your own
trust anchors for this zone.
    # local-zone: "localhost."
nodefault
    # local-zone: "127.in-
addr.arpa." nodefault
    # local-zone:
"1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.a
```

```
rpa." nodefault
    # local-zone: "10.in-
addr.arpa." nodefault
    # local-zone: "16.172.in-
addr.arpa." nodefault
    # local-zone: "17.172.in-
addr.arpa." nodefault
    # local-zone: "18.172.in-
addr.arpa." nodefault
    # local-zone: "19.172.in-
addr.arpa." nodefault
    # local-zone: "20.172.in-
addr.arpa." nodefault
    # local-zone: "21.172.in-
addr.arpa." nodefault
    # local-zone: "22.172.in-
addr.arpa." nodefault
    # local-zone: "23.172.in-
addr.arpa." nodefault
    # local-zone: "24.172.in-
addr.arpa." nodefault
    # local-zone: "25.172.in-
addr.arpa." nodefault
    # local-zone: "26.172.in-
addr.arpa." nodefault
    # local-zone: "27.172.in-
addr.arpa." nodefault
    # local-zone: "28.172.in-
addr.arpa." nodefault
    # local-zone: "29.172.in-
addr.arpa." nodefault
    # local-zone: "30.172.in-
addr.arpa." nodefault
    # local-zone: "31.172.in-
addr.arpa." nodefault
    # local-zone: "168.192.in-
addr.arpa." nodefault
    # local-zone: "0.in-addr.arpa."
nodefault
    # local-zone: "254.169.in-
addr.arpa." nodefault
    # local-zone: "2.0.192.in-
addr.arpa." nodefault
    # local-zone: "100.51.198.in-
addr.arpa." nodefault
    # local-zone: "113.0.203.in-
addr.arpa." nodefault
    # local-zone:
"255.255.255.255.in-addr.arpa."
nodefault
    # local-zone:
```

```
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa." nodefault
    # local-zone: "d.f.ip6.arpa." nodefault
    # local-zone: "8.e.f.ip6.arpa." nodefault
    # local-zone: "9.e.f.ip6.arpa." nodefault
    # local-zone: "a.e.f.ip6.arpa." nodefault
    # local-zone: "b.e.f.ip6.arpa." nodefault
    # local-zone: "8.b.d.0.1.0.0.2.ip6.arpa." nodefault

    # a number of locally served zones can be configured.
    #     local-zone: <zone> <type>
    #     local-data: "<resource record string>"
    # o deny serves local data (if any), else, drops queries.
    # o refuse serves local data (if any), else, replies with error.
    # o static serves local data, else, nxdomain or nodata answer.
    # o transparent gives local data, but resolves normally for other names
    # o redirect serves the zone data for any subdomain in the zone.
    # o nodefault can be used to normally resolve AS112 zones.
    # o typetransparent resolves normally for other types and other names
    #
    # defaults are localhost address, reverse for 127.0.0.1 and ::1
    # and nxdomain for AS112 zones. If you configure one of these zones
    # the default content is omitted, or you can omit it with 'nodefault'.
    #
    # If you configure local-data without specifying local-zone, by
    # default a transparent local-
```

```
zone is created for the data.
    #
    # You can add locally served
data with
    # local-zone: "local." static
    # local-data:
"mycomputer.local. IN A 192.0.2.51"
    # local-data: 'mytext.local TXT
"content of text record"'
    #
    # You can override certain
queries with
    # local-data:
"adserver.example.com A 127.0.0.1"
    #
    # You can redirect a domain to
a fixed address with
    # (this makes example.com,
www.example.com, etc, all go to
192.0.2.3)
    # local-zone: "example.com"
redirect
    # local-data: "example.com A
192.0.2.3"
    #
    # Shorthand to make PTR
records, "IPv4 name" or "IPv6
name".
    # You can also add PTR records
using local-data directly, but then
    # you need to do the reverse
notation yourself.
    # local-data-ptr: "192.0.2.3
www.example.com"

    # service clients over SSL (on
the TCP sockets), with plain DNS
inside
    # the SSL stream.  Give the
certificate to use and private key.
    # default is "" (disabled).
requires restart to take effect.
    # ssl-service-key:
"path/to/privatekeyfile.key"
    # ssl-service-pem:
"path/to/publiccertfile.pem"
    # ssl-port: 443

    # request upstream over SSL
(with plain DNS inside the SSL
stream).
```

```
    # Default is no.  Can be turned
on and off with unbound-control.
    # ssl-upstream: no

# Python config section. To enable:
# o use --with-pythonmodule to
configure before compiling.
# o list python in the module-
config string (above) to enable.
# o and give a python-script to
run.
python:
    # Script file to load
    # python-script:
"/etc/unbound/ubmodule-tst.py"

# Remote control config section.
remote-control:
    # Enable remote control with
unbound-control(8) here.
    # set up the keys and
certificates with unbound-control-
setup.
    # control-enable: no

    # what interfaces are listened
to for remote control.
    # give 0.0.0.0 and ::0 to
listen to all interfaces.
    # control-interface: 127.0.0.1
    # control-interface: ::1

    # port number for remote
control operations.
    # control-port: 8953

    # unbound server key file.
    # server-key-file:
"/etc/unbound/unbound_server.key"

    # unbound server certificate
file.
    # server-cert-file:
"/etc/unbound/unbound_server.pem"

    # unbound-control key file.
    # control-key-file:
"/etc/unbound/unbound_control.key"

    # unbound-control certificate
file.
```

```
    # control-cert-file:
"/etc/unbound/unbound_control.pem"

# Stub zones.
# Create entries like below, to
make all queries for 'example.com'
and
# 'example.org' go to the given
list of nameservers. list zero or
more
# nameservers by hostname or by
ipaddress. If you set stub-prime to
yes,
# the list is treated as priming
hints (default is no).
# With stub-first yes, it attempts
without the stub if it fails.
# stub-zone:
#   name: "example.com"
#   stub-addr: 192.0.2.68
#   stub-prime: no
#   stub-first: no
# stub-zone:
#   name: "example.org"
#   stub-host: ns.example.com.

# Forward zones
# Create entries like below, to
make all queries for 'example.com'
and
# 'example.org' go to the given
list of servers. These servers have
to handle
# recursion to other nameservers.
List zero or more nameservers by
hostname
# or by ipaddress. Use an entry
with name "." to forward all
queries.
# If you enable forward-first, it
attempts without the forward if it
fails.
# forward-zone:
#   name: "example.com"
#   forward-addr: 192.0.2.68
#   forward-addr: 192.0.2.73@5355
# forward to port 5355.
#   forward-first: no
# forward-zone:
#   name: "example.org"
```

```
#   forward-host: fwd.example.com
```

Liste des serveurs DNS racines

Téléchargez le fichier **named.cache** (liste des serveurs DNS racines sur lequel s'appuiera **unbound** pour répondre aux requêtes et enregistrez-le dans le répertoire **/var/lib/unbound/** en le renommant **root.hints** :

```
$ wget ftp://FTP.INTERNIC.NET/domain/named.cache -O
/var/lib/unbound/root.hints
```

Voici l'exemple fourni avec le paquet : Exemple de fichier Unbound.conf

verbosity: <chiffre>
Niveau de détail des messages.

- 0 ⇒ pas de message, que les erreurs.
- **1 ⇒ informations opérationnelles.** (par défaut)
- 2 ⇒ informations opérationnelles détaillées.
- 3 ⇒ informations au niveau requête, classées par requête.
- 4 ⇒ informations au niveau de l'algorithme.
- 5 ⇒ enregistre l'identification des clients non mis en cache.

interface: <ip address[@port]>
Interface à utiliser pour se connecter au réseau.
Par défaut : localhost, port par défaut (fourni par le paramètre port, 53 sinon).
Cette interface sert à l’écoute des requêtes des clients et au renvoi des réponses.
Peut être fourni plusieurs fois pour travailler sur de multiples interfaces.
Les interfaces ne sont pas modifiées par un reload (kill -HUP), mais seulement au redémarrage.

port: <port number>
Numéro de port sur lequel le serveur répond aux requêtes.
par défaut : 53

do-ip4: <yes or no>
Active ou désactive les réponses aux requêtes IP4.

Par défaut : yes.

do-ip6: <yes or no>

Active ou désactive les réponses aux requêtes IP6.

Par défaut : yes.

do-udp: <yes or no>

Active ou désactive les réponses aux requêtes UDP.

Par défaut : yes.

do-tcp: <yes or no>

Active ou désactive les réponses aux requêtes TCP.

Par défaut : yes.

do-daemonize: <yes or no>

Active ou désactive le fonctionnement en arrière-plan (comme un démon).

Par défaut : yes.

access-control: <IP netblock> <action>

netblock : plage d'adresses IP4 ou IP6 suivie de /size pour un bloc de réseau sans classes.
Actions possibles : deny, refuse, allow, allow_snoop, deny_non_local ou refuse_non_local.

- **deny** : bloque les requêtes des hôtes de cette plage d'adresses
- **refuse** : bloque aussi les requêtes mais renvoie un message d'erreur DNS rcode REFUSED.
- **allow** autorise les requêtes des hôtes de cette plage d'adresses. N'autorise que les clients récursifs, les requêtes non recursives sont refusées.
- **allow_snoop** autorise aussi l'accès non recursif. Cela donne un accès à la fois récursif et non récursif.
- **deny_non_local** : les messages non autorisés sont sautés
- **refuse_non_local** ils reçoivent le code d'erreur REFUSED.

Si aucun **deny** ne correspond, la correspondance de plage la plus spécifique est utilisée.

Par défaut, seul localhost est autorisé, le reste est bloqué.

root-hints: <filename>

Lire les indications de racine dans ce fichier.

Par défaut : rien, en utilisant des builtin pour la classe IN.

Le fichier a le format des fichiers de zone, avec seulement root names et addresses.

hide-identity: <yes or no>

hide-version: <yes or no>

Cacher les infos sur le serveur DNS.

harden-glue: <yes or no>

limite l'usurpation de DNS.

Par défaut : yes.

harden-dnssec-stripped: <yes or no>

Requérir les infos DNSSEC pour les zones de confiance.

Par défaut : on.

use-caps-for-id: <yes or no>

Ne pas tenir compte de la casse dans la requête : MonSite.com équivaut à monsite.com.

no par défaut

cache-min-ttl: <seconds>

valeur mini de la TTL en secondes. Ne pas dépasser 1h

Par défaut : 0.

prefetch: <yes or no>

activation du prefetch. Si un requête est faite lorsque la tll expire dans moins de 10% du temps qu'il lui est imparti, le cache se mettra à jour aussitôt après avoir répondu à la requête.

Par défaut : no.

num-threads: <number>

Le nombre de threads à créer pour servir les clients. Utilisez 1 pour pas de threading.

msg-cache-slabs: <number>

rrset-cache-slabs: <number>

infra-cache-slabs: <number>

key-cache-slabs: <number>

Nombre de slabs à utiliser . Doit être une puissance de 2 du num-threads.

rrset-cache-size: <number>

msg-cache-size: <number>

Taille du cache. A plain number is in bytes, append 'k', 'm' or 'g' for kilobytes, megabytes or gigabytes (1024*1024 bytes in a megabyte).

Par défaut : 4 mégaoctets.

so-rcvbuf: <number>

Taille du buffer pour le port UPD en entrée. Évite la perte de message lors des requêtes

Par défaut : 0 (utiliser la valeur système)

private-address: <IP address or subnet>

Give IPv4 of IPv6 addresses or classless subnets. These are addresses on your private network, and are not allowed to be returned for public

internet names. Any occurrence of such addresses are removed from DNS answers. Additionally, the DNSSEC validator may mark the answers bogus. This protects against so-called DNS Rebinding, where a user browser is turned into a network proxy, allowing remote access through the browser to other parts of your private network. Some names can be allowed to contain your private addresses,
By default all the local-data that you configured is allowed to, and you can specify additional names using private-domain. No private addresses are enabled by default.
We consider to enable this for the RFC1918 private IP address space by default in later releases. That would enable private addresses for 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 169.254.0.0/16 fd00::/8 and fe80::/10, since the RFC standards say these addresses should not be visible on the public internet. Turning on 127.0.0.0/8 would hinder many spam-blocklists as they use that. Adding ::ffff:0:0/96 stops IPv4-mapped IPv6 addresses from bypassing the filter.

unwanted-reply-threshold: <number>

Si non nulles, les réponses indésirables ne sont pas seulement signalés dans les statistiques, mais aussi ajoutées à un total cumulé maintenu par thread. Quand le seuil est atteint, un avertissement est affiché et une action défensive est prise, le cache est vidé pour éviter l'empoissonnement DNS. Une valeur de 10000 est suggérée, la valeur par défaut est de 0 (service désactivé).
Par défaut : 0 (désactivé).

do-not-query-localhost: <yes or no>

Interdire de répondre aux requêtes du localhost ?
Par défaut : yes (localhost est interdit)
Si no, localhost est utilisable.

val-clean-additional: <yes or no>

Est-ce que cette section supplémentaire, doit être conservée intacte pour les données non-sécurisées ? Utile pour protéger les utilisateurs d'une validation de données potentiellement boguées. Toutes les données non signés dans la section supplémentaire seront retirés des messages sécurisés
Par défaut : yes.

chroot: <directory>

If chroot is enabled, you should pass the

configfile (from the commandline) as a full path from the original root. After the chroot has been performed the now defunct portion of the config file path is removed to be able to reread the config after a reload.
All other file paths (working dir, logfile, roothints, and key files) can be specified in several ways: as an absolute path relative to the new root, as a relative path to the working directory, or as an absolute path relative to the original root. In the last case the path is adjusted to remove the unused portion.
The pidfile can be either a relative path to the working directory, or an absolute path relative to the original root. It is written just prior to chroot and dropping permissions. This allows the pidfile to be /var/run/unbound.pid and the chroot to be /var/unbound, for example.
Additionally, unbound may need to access /dev/random (for entropy) from inside the chroot.
If given a chroot is done to the given directory. The default is “/usr/local/etc/unbound”.
If you give “” no chroot is performed.

username: <name>
If given, after binding the port the user privileges are dropped. Default is “unbound”. If you give username: “” no user change is performed.
If this user is not capable of binding the port, reloads (by signal HUP) will still retain the opened ports. If you change the port number in the config file, and that new port number requires privileges, then a reload will fail; a restart is needed.

directory: <directory>
Sets the working directory for the program. Default is “/usr/local/etc/unbound”. On Windows the string “%EXECUTABLE%” tries to change to the directory that unbound.exe resides in.

use-syslog: <yes or no>
Sets unbound to send log messages to the syslogd, using syslog(3). The log facility LOG_DAEMON is used, with identity “unbound”. The logfile setting is overridden when use-syslog is turned on.
The default is to log to syslog.

pidfile: <filename>
The process id is written to the file. Default is “/usr/local/etc/unbound/unbound.pid”. So, kill -HUP `cat

/usr/local/etc/unbound/unbound.pid` triggers a reload, kill -TERM `cat /usr/local/etc/unbound/unbound.pid` gracefully terminates.

root-hints: <filename>

Read the root hints from this file.
Default is nothing, using builtin hints for the IN class. The file has the format of zone files, with root nameserver names and addresses only. The default may become outdated, when servers change, therefore it is good practice to use a root-hints file.

Voici un exemple commenté

/etc/unbound/unbound.d/mondns.com

```
server:
  # Répondre aux requêtes DNS sur toutes les interfaces réseau.
  interface: 0.0.0.0
  # plage adresses ip autorisées à consulter le serveur dns :
  # uniquement le réseau local
  access-control: 192.168.0.0/24 allow
  # Ignorer la casse
  # HomeServer-DIY.net sera traduit en homeserver-diy.net et le serveur et communiquera la bonne IP
  use-caps-for-id: yes
  # Renforcer la vie privée des adresses du LAN. Ne mettre que des adresses locales
  private-address: 192.168.1.0/24
```

Pour créer des DNS avec tous les sous-domaines, utiliser la directive **redirect**.

Exemple pour localhost :

```
server:
    # zone *.localhost
    local-zone: "localhost" redirect
    local-data: "localhost A 127.0.0.1"
```

Il génère tous les sous-domaines xxx.localhost, a.localhost, etc.

Serveur DNS avec déclaration automatique des sous-domaines

Ouvrez avec les droits d'administration le fichier **/etc/unbound/unbound.conf** pour le modifier comme voulu :

Voici un exemple de fichier unbound.conf :

Nous créons ici des noms *.dev.com

/etc/unbound/unbound.conf

```
server:
    verbosity: 1

    interface: 192.168.0.31

    access-control: 0.0.0.0/0 allow
    chroot: ""

    logfile: "/var/log/unbound.log"
    use-syslog: no

    local-zone: "dev.com." redirect
    local-data: "dev.com. IN A 192.168.0.1"

# Utilisation du DNS "normal" pour tout le reste (ici, celui de la box)
forward-zone:
    name: "."
    forward-addr: 192.168.0.254
```

verbosity
degré de précision des messages
interface
adresse réseau du serveur (il peut y avoir plusieurs lignes pour plusieurs adresses
access-control:

0.0.0.0/0 allow
ouverture à tout le monde
logfile:
emplacement du fichier de log
use-syslog: no
pour ne pas “pourrir” le syslog
local-zone:

```
                la zone *.dev.com

                local-zone: “dev.com.” redirect
                            crée les redirections
                local-data: “dev.com. IN A 192.168.0.1”
                            IP de la machine qui sert ce
                            domaine et ses sous-domaines
forward-zone:
                Utilisation du DNS “normal” pour tout le reste

                name: “.”
                            pour tout le reste
                forward-addr: 192.168.0.254
                            DNS de la box
```

Test

Lancez :

```
...@...:~ $ unbound-checkconf
```

Exemples

Exemple fourni avec le paquet

/etc/unbound/unbound.d/monserveurdns.com

```
server:
  # Les lignes suivantes concernent la configuration de unbound
pour les
  # performance crypto DNSSEC en utilisant la clé des serveurs
root
  auto-trust-anchor-file: "/var/lib/unbound/root.key"
  # Activer les logs
  # verbosity: 1 (par défaut)
  #Répondre aux requêtes DNS sur toutes les interfaces réseau.
  interface: 0.0.0.0
  #Port sur lequel sont fait les requêtes DNS
  #port: 53 (par défaut)
  #support de l'IPv4
  #do-ip4: yes (par défaut)
  #support de  l'IPv6
  do-ip6: no
  #support udp
  #do-udp: yes (par défaut)
  #support tcp
```

```
  #do-tcp: yes (par défaut)
  #plages adresse ip autorisée à consulter le serveur dns
  #access-control: 127.0.0.0/8 allow
  #access-control: 192.168.1.0/24 allow
  #access-control: 192.168.1.26
  #plage ip comprise entre 192.168.0.0 et 192.168.255.255
  access-control: 192.168.1.0/16 allow
  #emplacement du fichier indiquant les infos pour consulter les
serveurs DNS root
  #fichier à télécharger là à cette adresse:
ftp://FTP.INTERNIC.NET/domain/named.cache
  root-hints: "/var/lib/unbound/root.hints"
  #Cacher les infos sur le serveur DNS
  hide-identity: yes
  hide-version: yes
  #paramètre limitant l'usurpation de DNS
  harden-glue: yes
  #Requérir les infos DNSSEC pour les zones de confiance
  harden-dnssec-stripped: yes
  #Options permettant de ne pas prendre la casse en compte lors
des requêtes d'url.
  #HomeServer-DIY.net sera traduit en homeserver-diy.net par le
serveur et il communiquera la bonne IP
  use-caps-for-id: yes
  #valeur mini de la TTL en secondes. Ne pas dépasser 1h
  cache-min-ttl: 3600
  #valeur max de la TTL en secondes.
  cache-max-ttl: 86400
  #activation du prefetch. Si un requête est faite lorsque la tll
expire dans moins de 10% du temps qu'il lui est imparti
  #le cache se mettra à jour aussitôt après avoir répondu à la
requête.
  prefetch: yes
  #nombre de core du serveur dns
  num-threads: 2
  ## Tweaks et optimisations du cache
  #Nombre de slabs à utiliser . Doit être une puissance de 2 du
num-threads.
  msg-cache-slabs: 8
  rrset-cache-slabs: 8
  infra-cache-slabs: 8
  key-cache-slabs: 8
  #Réglage de la taille du cache en Mo:
  rrset-cache-size: 51m
  msg-cache-size: 25m
  #Taille du buffer pour le port UPD en entrée. Evite la perte de
message lors des requêtes
  so-rcvbuf: 1m
  #Renforcer la vie privée des adresses du LAN. Ne mettre que des
adresses locales
  private-address: 192.168.1.0/24
```

```
  #Si non nulles, les réponses indésirables ne sont pas seulement
signalés dans les statistiques,
  #mais aussi ajoutées à un total cumulé maintenu par thread.
  #Si elle atteint le seuil, un avertissement est affiché et une
action défensive est prise, le cache est vidé pour éviter
l'empoissonnement DNS.
  #Une valeur de 10000 est suggérée, la valeur par défaut est de 0
(service désactivé).
  unwanted-reply-threshold: 10000
  #Autorisé à répondre aux requêtes du localhost
  do-not-query-localhost: no
  #Emplacement du fichier root.key pour utilisation de DNSSEC
  #auto-trust-anchor-file: "/var/lib/unbound/root.key"
  # Est-ce que cette section supplémentaire, doit être conservée
intacte pour les données non-sécurisées
  # Utile pour protéger les utilisateurs d'une validation de
données potentiellement boguées
  # Toutes les données non signés dans la section supplémentaire
seront retirés des messages sécurisés
  val-clean-additional: yes
```

Conf de Korben, Voir : http://korben.info/installer-unbound-serveur-dns-sous-linux.html

```
server:
  #verbosity: 1
  interface: 0.0.0.0
  #port: 53
  #do-ip4: yes
  #do-ip6: yes
  #do-udp: yes
  #do-tcp: yes
  do-daemonize: yes
  access-control: 0.0.0.0/0 allow
  #access-control: 0.0.0.0/0 refuse
  #access-control: 127.0.0.0/8 allow
  chroot: "/var/unbound"
  username: "unbound"
  directory: "/var/unbound"
  use-syslog: yes
  pidfile: "/var/run/unbound.pid"
  root-hints: "/var/unbound/named.cache"
```

avec un utilisateur dédié créé par :

- ```
  sudo groupadd unbound
  sudo useradd -d /var/unbound -m -g unbound -s /bin/false unbound
  ```

Le répertoire /var/unbound sera utilisé par Unbound et contiendra les fichiers de config.
  ```

Bloquer les pubs sur internet

De nombreuses pages web contiennent du code qui affiche des pubs en s'appuyant sur les noms de domaines des régies publicitaires de google, yahoo etc. Lors du chargement de la page, votre ordinateur fera donc une requête DNS pour résoudre les domaines de ces régies.

Pour bloquer ces pubs, il suffit de configurer **unbound** pour qu'il retourne une adresse IP non attribuée du réseau local. Mais celle du localhost de l'exemple fonctionne aussi très bien. Voici ce qu'il suffit d'ajouter à la fin du fichier unbound.conf:

```
  local-zone: "doubleclick.net" redirect
  local-data: "doubleclick.net A 127.0.0.1"
  local-zone: "googlesyndication.com" redirect
  local-data: "googlesyndication.com A 127.0.0.1"
  local-zone: "googleadservices.com" redirect
  local-data: "googleadservices.com A 127.0.0.1"
  local-zone: "google-analytics.com" redirect
  local-data: "google-analytics.com A 127.0.0.1"
  local-zone: "ads.youtube.com" redirect
  local-data: "ads.youtube.com A 127.0.0.1"
  local-zone: "adserver.yahoo.com" redirect
  local-data: "adserver.yahoo.com A 127.0.0.1"
```

Cette liste n'est pas exhaustive ; le fichier suivant contient un grand nombre de domaines utilisés par un grande nombre de régies publicitaires : Regies pub.odt

Sous Windows

La configuration se fait en éditant le fichier **C:\Program Files\Unbound\unbound.conf**.

A ce même endroit se trouve un fichier exemple **C:\Program Files\Unbound\example.conf** que l'on peut recopier en le renommant **unbound.conf** pour partir de cette base.

Utilisation

Sous ubuntu

Contrôle

Relancer le service :

```
...@...:~ $ sudo service unbound restart
```

Démarrer **unbound** :

```
...@...:~ $ sudo systemctl start unbound
```

Arrêter **unbound** :

```
...@...:~ $ sudo systemctl stop unbound
```

Redémarrer **unbound** :

```
...@...:~ $ sudo systemctl restart unbound
```

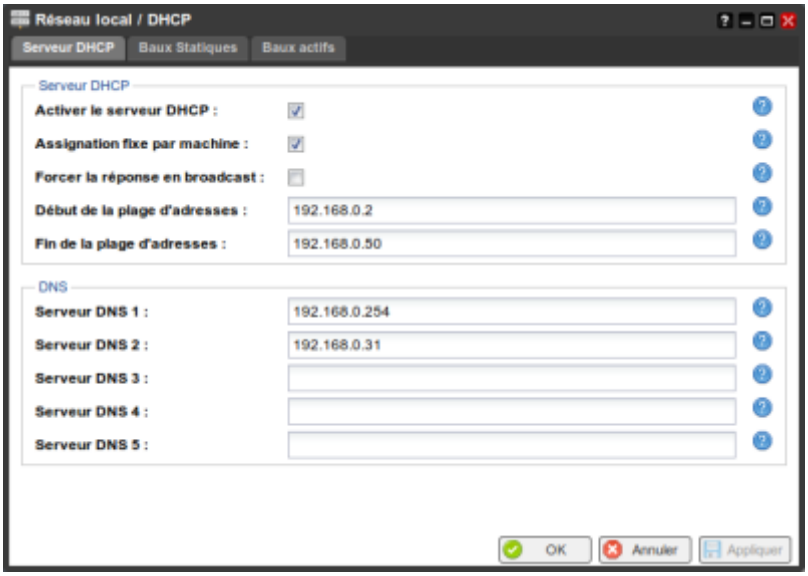
Connaître l'état d'**unbound** :

```
...@...:~ $ sudo systemctl status unbound
```

Sous Windows

Utilisation depuis les autres machines

Le plus simple est d'utiliser le protocole DHCP : déclarez sur votre box l'adresse de votre serveur hébergeant unbound pour que toutes vos machines utilisent ce nouveau service auto-hébergé:



Désinstallation

- Pour supprimer cette application, il suffit de supprimer son paquet :

```
...@...:~ $ sudo apt-get remove unbound
```

- ou pour effacer tout :

```
...@...:~ $ sudo apt-get purge unbound
```

Voir aussi

- **(en)** manuel (pdf)
- **(fr)** doc ubuntu : https://doc.ubuntu-fr.org/unbound
- **(fr)** https://techarea.fr/creer-resolveur-dns-unbound-debian/
- **(fr)** https://memo-linux.com/debian-installer-le-serveur-dns-unbound/
- **(en)** http://blog.loicg.net/developpement-web/dns-local-virtualdocumentroot/
- **(en)** site officiel : http://unbound.net/
- **(en)** page de man unbound : https://www.unbound.net/documentation/unbound.html
- **(en)** page de man unbound.conf : https://www.unbound.net/documentation/unbound.conf.html
- **(fr)** unbound.conf(5) - page de man
- **(en)** tutoriel https://calomel.org/unbound_dns.html

Basé sur « manuel » par unbound.

[1]
C'est une alternative plus simple a configurer que **bind9** pour un LAN de petite taille.