logiciel, BROUILLON

Exemple de fichier unbound.conf

Réglages

clause server:

La clause **server:** définit les paramètres généraux.

clause python:

clause remote-control:

clause stub-zone:

clause forward-zone:

clause auth-zone:

clause view:

DNSCrypt

CacheDB

IPSet

Essai de traduction du fichier

Fichier de configuration exemple.

Voir la page de manuel unbound.conf (5), version 1.4.22.

```
# ceci est un commentaire.
```

Les lignes blanches sont facultatives, elles clarifient le texte.

Utilisez cette option pour inclure un autre texte dans le fichier.

```
# include: "otherfile.conf"
```

Clause server:

La clause server: définit les paramètres principaux.

local-zone:

Configure une zone locale.

Utilisez local-data: pour entrer des données dans la zone locale.

Les réponses pour les zones locales sont des réponses de DNS autorisées.

Par défaut les zones sont de classe IN.

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

Syntaxe

```
local-zone: <zone> <type>
```

<zone>

Les zones par défaut sont localhost, reverse 127.0.0.1 et :: 1 et les zones AS112. Les zones d'AS112 sont des zones reverse DNS pour l'utilisation privée et auxquelles les serveurs sur Internet ne peuvent pas fournir des réponses correctes.

Elles sont configurées par défaut pour donner nxdomain (aucune information reverse) les réponses. Les valeurs par défaut peuvent être annulées en spécifiant votre propre zone locale de ce nom, ou en utilisant le type 'nodefault'.

Ci-dessous une liste des zones implicites.

? localhost :: The IP4 and IP6 localhost information is given. NS and SOA records are provided for completeness and to satisfy some DNS update tools. :: Default content:

```
local-zone: "localhost." static
local-data: "localhost. 10800 IN NS localhost."
local-data: "localhost. 10800 IN
    SOA localhost. nobody.invalid. 1 3600 1200 604800 10800"
```

```
local-data: "localhost. 10800 IN A 127.0.0.1"
local-data: "localhost. 10800 IN AAAA ::1"
```

!!

? reverse IPv4 loopback :: Default content:

```
local-zone: "127.in-addr.arpa." static
local-data: "127.in-addr.arpa. 10800 IN NS localhost."
local-data: "127.in-addr.arpa. 10800 IN
    SOA localhost. nobody.invalid. 1 3600 1200 604800 10800"
local-data: "1.0.0.127.in-addr.arpa. 10800 IN
    PTR localhost."
```

!!

? reverse IPv6 loopback :: Default content:

```
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.
                    0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa." static
local-data: "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.
    0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa. 10800 IN
    NS localhost."
local-data: "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.
    0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa. 10800 IN
    SOA localhost. nobody.invalid. 1 3600 1200 604800 10800"
local-data: "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.
    0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa. 10800 IN
    PTR localhost."
```

!!

? reverse RFC1918 local use zones :: Reverse data for zones 10.in-addr.arpa, 16.172.in-addr.arpa to 31.172.in-addr.arpa, 168.192.in-addr.arpa. The local-zone: is set static and as local-data: SOA and NS records are provided. !!
? reverse RFC3330 IP4 this, link-local, testnet and broadcast :: Reverse data for zones 0.in-addr.arpa, 254.169.in-addr.arpa, 2.0.192.in-addr.arpa (TEST NET 1), 100.51.198.in-addr.arpa (TEST NET 2), 113.0.203.in-addr.arpa (TEST NET 3), 255.255.255.255.in-addr.arpa. !!
? reverse RFC4291 IP6 unspecified :: Reverse data for 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.
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa.
```

? reverse RFC4193 IPv6 Locally Assigned Local Addresses :: Reverse data for zone D.F.ip6.arpa. !!
? reverse RFC4291 IPv6 Link Local Addresses :: Reverse data for zones 8.E.F.ip6.arpa to B.E.F.ip6.arpa. !!
? reverse IPv6 Example Prefix :: Reverse data for zone 8.B.D.0.1.0.0.2.ip6.arpa. This zone is used for tutorials and examples. You can remove the block on this zone with:

```
local-zone: 8.B.D.0.1.0.0.2.ip6.arpa. nodefault
```

You can also selectively unblock a part of the zone by making that part transparent with a local-zone

statement. This also works with the other default zones. !!

? local-data: “<resource record string>” :: Configure local data, which is served in reply to queries for it. :: The query has to match exactly unless you configure the local-zone as redirect. If not matched exactly, the local-zone type determines further processing. If local-data is configured that is nota subdomain of a local-zone, a transparent local-zone is configured. For record types such as TXT, use single quotes, as inlocal-data: 'example. TXT “text”'. !!
? local-data-ptr: “IPaddr name” :: Configure local data shorthand for a PTR record with the reversed IPv4 or IPv6 address and the host name. For example “192.0.2.4 www.example.com”. TTL can be inserted like this:

```
"2001:DB8::4 7200 www.example.com"
```

!!

<type>

Le type détermine la réponse à donner s'il n'y a pas de correspondance avec local-data.

Les types sont deny, refuse, static, transparent, redirect, nodefault, typetransparent et sont expliqués ci-dessous. Après cela les valeurs par défaut sont énumérées.

? deny :: Ne pas retourner de réponse, abandonner la requête. !!
? refuse :: Envoie en réponse un message d'erreur, avec le rcode REFUSED. S'il y a une concordance avec des données locales, une réponse à la requête est retournée. !!
? static :: Sinon, on répond à la requête par nodata ou nxdomain.\ Pour une réponse négative un SOA est inclus dans la réponse si le présent comme local-data pour le domaine zone apex. !!
? transparent :: S'il y a une concordance avec des données locales, une réponse à la requête est retournée. :: Sinon, si la requête a un nom différent, la requête est résolue normalement. :: Si la requête est pour un nom donné dans localdata mais qu'aucun type de données n'est donné dans localdata, alors une réponse est retournée. :: Si aucune local-zone n'est donnée, local-data crée une zone transparente par défaut. !!
? typetransparent :: S'il y a une concordance avec des données locales, une réponse à la requête est retournée. :: Si la requête est pour un nom différent ou pour le même nom, mais pour un type différent, la requête est résolue normalement. :: C'est comme transparent mais les types qui ne sont pas énumérés dans les données locales sont résolus normalement, ainsi si un enregistrement A est dans les données locales, il ne provoque pas de réponse de nodata pour les requêtes d'AAAA. !!
? redirect :: pour le nom de zone, la réponse à la requête est fournie à partir des données locales. :: Cela répond aux requêtes pour la zone et tous les sous-domaines de la zone avec les données locales pour la zone. :: Cela permet de rediriger un domaine pour retourner un autre enregistrement d'adresse à l'utilisateur final. Par exemple, avec

```
local-zone:   "example.com."   redirect
local-data:  "example.com. A 127.0.0.1"
```

les requêtes pour www.example.com et www.foo.example.com sont redirigées, pour que les utilisateurs avec les navigateurs de web ne puissent pas accéder aux sites avec le suffixe example.com. !!

? nodefault :: annule les valeurs par défaut !!

Paramètres les plus courants

? verbosity: <chiffre> :: (défaut **1**) :: Niveau de détail des messages, de 0 (pas de détail) à 5 (très détaillé). !!
? interface: <adresse IP[@port]> :: Interface à utiliser pour se connecter au réseau. Cette interface est écoutée pour les requêtes des clients, et les réponses aux clients sont envoyées par elle. Peut être donné plusieurs fois pour travailler sur plusieurs interfaces. Si aucune n'est fournie, la valeur par défaut est d'écouter localhost. Les interfaces ne sont pas modifiées par un reload (kill -HUP) mais seulement au redémarrage. Un numéro de port peut être spécifié avec @port (sans espace entre l'interface et le numéro de port), si non spécifié le port par défaut est utilisé (celui de port). :: 0.0.0.0 and \:\:0 pour lier à toutes les interfaces disponibles. :: chaque interface[@port] doit être sur une nouvelle ligne **interface:**.
? Exemples : :: interface: 192.0.2.153 :: interface: 192.0.2.154 :: interface: 192.0.2.154@5003 :: interface: 2001:DB8\:\:5 :: !!
? port: <numéro de port> :: Numéro de port, par défaut 53, sur lequel le serveur répond aux requêtes.
? Exemple: :: port: 53 !!
? access-control: <plage IP> <action> :: 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. :: La plage IP est donnée sous forme IP4 ou IP6 avec /size pour une plage sans classes. L'action peut être deny, refuse, allow, allow_snoop, deny_non_local ou refuse_non_local. :: * **deny** bloque les requêtes provenant d'hôtes à partir de cette plage. :: * **refuse** bloque aussi les requêtes mais retourne un message d'erreur DNS rcode REFUSED. :: * **allow** donne accès aux clients à partir de cete plage. Il ne donne accès qu'aux clients récurrents (ce qui est le cas de presque tous les clients). Les requêtes non récursives sont rejetées. L'action allow permet aux requêtes non récursives d'accéder au local-data qui est configuré. La raison en est que cela ne concerne pas l'algorithme de recherche récursive du serveur unbound et les données statiques sont servies dans la réponse. Cela supporte les opérations normales où les requêtes non récursives sont faites pour les données authoritative. Pour les requêtes non récursives, les réponses du cache dynamique sont refusées. :: * **allow_snoop** donne aussi un accès non recursif. Cela donne à la fois l'accès récursif et non récursif. Le nom allow_snoop se réfère à l'espionnage de cache, une technique qui utilise des requêtes non récursives pour examiner le contenu du cache (pour des actes de malveillance). Toutefois, les requêtes non récursives peuvent aussi être un bon outil de débogage (quand vous voulez pour examiner le contenu de la mémoire cache). Dans ce cas utilisez allow_snoop pour votre hôte d'administration. :: Par défaut, seul localhost est autorisé, le reste est refusé. La valeur par défaut est refusée, parce que c'est protocol-friendly. Le protocole DNS n'est pas conçu pour gérer les paquets rejetés en raison de la politique, et le rejet peut entraîner des réessais (éventuellement excessifs) de requêtes. :: * **deny_non_local** et **refuse_non_local** sont pour les hôtes qui ne sont autorisés à interroger que le authoritative local-data, ils ne sont pas autorisés à une pleine récursivité mais seulement aux données statiques. Avec deny_non_local, les messages qui ne sont pas admis sont abandonnés, avec refuse_non_local, ils reçoivent le code d'erreur REFUSED. !!
? Exemples : :: 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 !!
? chroot: <directory> :: Si chroot est activé, vous devez passer en ligne de commande le chemin d'accès complet (à partir de la racine d'origine). Une fois le chroot effectué, la partie inactive du chemin du fichier de configuration est retirée pour pouvoir relire la config après un rechargement. :: Tous les autres chemins de fichiers (répertoire de travail, fichier journal, roothints, et les fichiers clés) peuvent être spécifiés dans plusieurs façons

un chemin absolu par rapport à la nouvelle racine, un chemin relatif au répertoire de travail, ou un chemin absolu par rapport à la racine d'origine. Dans ce dernier cas, le chemin est ajusté pour éliminer la portion inutilisée. :: Le pidfile peut être soit un chemin relatif au répertoire de travail, ou un chemin absolu par rapport à la racine d'origine. Il est écrit juste avant les permissions de chroot et d'abandon. Cela permet au pidfile d'être /var/run/unbound.pid et le chroot /var/unbound, par exemple. :: En outre, unbound peut avoir besoin d'accéder à /dev/random (pour l'entropie) depuis l'intérieur du chroot. :: S'il est renseigné, le chroot est fait pour le répertoire donné. La valeur par défaut est /etc/unbound Si vous donnez `aucun chroot n'est effectué. ;? Exemple : :: chroot: “/etc/unbound” !! ;? logfile: <filename> :: Si on met` , le journal est envoyé à stderr, ou nulle part une fois transforné en daemon. Le fichier journal y est ajouté dans le format suivant: [secondes depuis 1970] unbound[pid:tid]: type: message. Si cette option est donnée, l'option use-syslog est réglée sur 'no'. Le fichier journal est rouvert (pour append) lorsque le fichier de configuration est relu, par SIGHUP. !!

? use-syslog: <yes or no> :: Fait que unbound envoie les messages de log au syslogd, via sys-log. Le dispositif de journal LOG_DAEMON est utilisé, avec l'identité “unbound”. Le réglage du fichier journal est écrasé quand use-syslog est activé. défaut

envoyer le journal à syslog. !!

?local-zone: <zone> <type> :: Configure une zone locale. Le type détermine la réponse à donner s'il n'y a pas de correspondance avec local-data. Les types sont deny, refuse, static, transparent, redirect, nodefault, typetransparent, et sont expliqués ci-dessous. Après cela, les paramètres par défaut sont listés. Utilisez local-data: pour entrer des données dans la zone locale.Les réponses pour les zones locales sont des réponses DNS autoritaires. Par défaut, les zones sont de classe IN. :: Si vous avez besoin de données faisant autorité plus compliquées, avec referrals, wildcards, CNAME/DNAME, ou un service autoritaire DNSSEC, authoritative service, installez un stub-zone pour elle comme détaillé dans la section de zone de stub ci-dessous. !!

? deny :: Ne pas envoyer une réponse, ignorer la requête. Si il y a une correspondance à partir de données locales, une réponse est fournie à la requête.

? refuse :: Envoyer un message d'erreur avec rcode REFUSED. S'il y a une correspondance à partir de données locales, une réponse est faite à la requête.

? static :: S'il y a une correspondance à partir de données locales, une réponse est faite à la requête. Sinon, une réponse est donnée avec nodata ou nxdomain. Pour une réponse négative, un SOA est inclus dans la réponse si il se présente comme local-data pour la zone apex domain.

? transparent :: S'il y a une correspondance à partir de données locales, une réponse est faite à la requête. Sinon, si la requête est pour un nom différent, la requête est résolue normalement. Si la requête est pour un nom donné dans localdata mais qu'iln'y ait pas ce type de données dans localdata, une réponse noerror nodata est retournée. Si aucune local-zone n'est donnée,local-data crée une zone transparente par défaut.

? typetransparent :: S'il y a une correspondance à partir de données locales, une réponse est faite à la requête. Sinon, si la requête est pour un nom différent ou pour le même nom mais pour un type différent, la requête est résolue normalement. Donc, comme pour transparent mais les types ne sont pas répertoriés dans les données locales, sont résolus normalement, si un enregistrement A est dans les données locales qui ne provoque pas une réponse nodata pour les requêtes AAAA.

? redirect :: La réponse à la requête est fournie à partir des données locales pour le nom de la zone. Il peut n'y avoir aucune donnée locale sous le nom de zone. Cela répond à des requêtes pour la zone, et tous les sous-domaines de la zone avec les données locales pour la zone. Cela peut servir à pour rediriger un domaine pour renvoyer à l'utilisateur final un enregistrement d'adresse différent, avec

```
local-zone:   "example.com."   redirect
```

et

```
local-data:  "example.com. A 127.0.0.1"
```

les requêtes pour www.example.com et www.foo.example.com sont redirigées, de telle sorte que les utilisateurs avec des navigateurs Web ne peuvent pas accéder à des sites avec le suffixe example.com.

? nodefault :: Utilisé pour désactiver le contenu par défaut pour les zones de AS112. Les autres types désactivent aussi le contenu par défaut pour la zone. L'option 'nodefault' n'a pas d'autre effet que de désactiver le contenu par défaut pour la zone donnée. :: Les zones par défaut sont localhost, reverse 127.0.0.1 and ::1, et les zones AS112. Les zones AS112 sont des zones DNS inverses pour un usage privé et des adresses IP réservées pour lesquels les serveurs sur l'Internet ne peuvent pas fournir des réponses correctes. Ils sont configurés par défaut pour donner des réponses nxdomain (aucune information inverse). Les valeurs par défaut peuvent être désactivées en spécifiant votre propre zone locale de ce nom, ou en utilisant le type 'nodefault'. Voici une liste du contenu de la zone par défaut.
? localhost :: Les informations IP4 et IP6 localhost est donnée. Les enregistrements NS et SOA sont prévus pour l'exhaustivité et à satisfaire certains des outils update tools. Contenu par défaut :

```
    local-zone: "localhost." static
    local-data: "localhost. 10800 IN NS localhost."
    local-data: "localhost. 10800 IN
        SOA localhost. nobody.invalid. 1 3600 1200 604800 10800"
    local-data: "localhost. 10800 IN A 127.0.0.1"
    local-data: "localhost. 10800 IN AAAA ::1"
```

!!

? reverse IPv6 loopback :: Default content:

```
                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.ip6.arpa." static
                local-data: "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.ip6.arpa. 10800 IN
                    NS localhost."
                local-data: "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.ip6.arpa. 10800 IN
                    SOA localhost. nobody.invalid. 1 3600 1200 604800
10800"
                local-data: "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.ip6.arpa. 10800 IN
                    PTR localhost."
```

!!

? reverse RFC1918 local use zones :: Reverse data for zones 10.in-addr.arpa, 16.172.in-addr.arpa to 31.172.in-addr.arpa, 168.192.in-addr.arpa. The local-zone: is set static and as local-data: SOA and NS records are provided. !!
? reverse RFC3330 IP4 this, link-local, testnet and broadcast :: Reverse data for zones 0.in-addr.arpa, 254.169.in-addr.arpa,

```
                2.0.192.in-addr.arpa  (TEST  NET  1), 100.51.198.in-
```

```
addr.arpa
                (TEST  NET   2),   113.0.203.in-addr.arpa   (TEST   NET
3),
                255.255.255.255.in-addr.arpa.
```

!!

? reverse RFC4291 IP6 unspecified :: Reverse data for 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.ip6.arpa.
```

!!

? reverse RFC4193 IPv6 Locally Assigned Local Addresses :: Reverse data for zone D.F.ip6.arpa. !!
? reverse RFC4291 IPv6 Link Local Addresses :: Reverse data for zones 8.E.F.ip6.arpa to B.E.F.ip6.arpa. !!
? reverse IPv6 Example Prefix :: Reverse data for zone 8.B.D.0.1.0.0.2.ip6.arpa. This zone is used for tutorials and examples. You can remove the block on this zone with:

```
local-zone: 8.B.D.0.1.0.0.2.ip6.arpa. nodefault
```

You can also selectively unblock a part of the zone by making that part transparent with a local-zone statement. This also works with the other default zones. !!

? local-data: “<resource record string>” :: Configure local data, which is served in reply to queries for it. The query has to match exactly unless you configure the local-zone as redirect. If not matched exactly, the local-zone type determines further processing. If local-data is configured that is not a subdomain of a local-zone, a transparent local-zone is configured. For record types such as TXT, use single quotes, as in local-data: 'example. TXT “text”'. :: If you need more complicated authoritative data, with referrals, wildcards, CNAME/DNAME support, or DNSSEC authoritative service, setup a stub-zone for it as detailed in the stub zone section below. !!
? local-data-ptr: “IPaddr name” :: Configure local data shorthand for a PTR record with the reversed

```
          IPv4  or  IPv6  address and the host name.  For example "192.0.2.4
          www.example.com".  TTL can be  inserted  like  this:  "2001:DB8::4
          7200 www.example.com" !!
```

? 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.ip6.arpa.” 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.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 :: 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. :: 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'. !!

Autres paramètres

? statistics-interval: 0 :: print statistics to the log (for every thread) every N seconds. Set to “” or 0 to disable. Default is disabled. !!
? statistics-cumulative: no :: enable cumulative statistics, without clearing them after printing. !!
? extended-statistics: no :: enable extended statistics (query types, answer codes, status) printed from unbound-control. default off, because of speed. !!
? num-threads: 1 :: number of threads to create. 1 disables threading. !!
? interface-automatic: no :: enable this feature to copy the source address of queries to reply. Socket options are not supported on all platforms. experimental. !!
? outgoing-interface: 192.0.2.153
? outgoing-interface: 2001:DB8\:\:5
? outgoing-interface: 2001:DB8\:\:6 :: 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-range: 4096 :: 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-port-permit: 32768 :: permit unbound to use this port number or port range for making outgoing queries, using an outgoing interface. !!
? outgoing-port-avoid: “3200-3208” :: 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-num-tcp: 10 :: number of outgoing simultaneous tcp buffers to hold per thread. !!
? incoming-num-tcp: 10 :: number of incoming simultaneous tcp buffers to hold per thread. !!
? so-rcvbuf: 0 :: 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-sndbuf: 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-reuseport: no :: on Linux(3.9+) use SO_REUSEPORT to distribute queries over threads. !!
? edns-buffer-size: 4096 :: EDNS reassembly buffer to advertise to UDP peers (the actual buffer is set with msg-buffer-size). 1480 can solve fragmentation (timeouts). !!
? max-udp-size: 4096 :: Maximum UDP response size (not applied to TCP response). Suggested values are 512 to 4096. Default is 4096. 65536 disables it. !!
? msg-buffer-size: 65552 :: buffer size for handling DNS data. No messages larger than this size can be sent or received, by UDP or TCP. In bytes. !!
? msg-cache-size: 4m :: 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-slabs: 4 :: 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. !!
? num-queries-per-thread: 1024 :: the number of queries that a thread gets to service. !!
? jostle-timeout: 200 :: if very busy, 50% queries run to completion, 50% get timeout in msec !!
? delay-close: 0 :: msec to wait before close of port on timeout UDP. 0 disables. !!
? rrset-cache-size: 4m :: 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-slabs: 4 :: 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. !!
? cache-min-ttl: 0 :: 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-max-ttl: 86400 :: the time to live (TTL) value cap for RRsets and messages in the cache. Items are not cached for longer. In seconds. !!
? infra-host-ttl: 900 :: the time to live (TTL) value for cached roundtrip times, lameness and EDNS version information for hosts. In seconds. !!
? infra-cache-slabs: 4 :: 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-numhosts: 10000 :: the maximum number of hosts that are cached (roundtrip, EDNS, lame). !!
? do-ip4: yes :: Enable IPv4, “yes” or “no”. !!
? do-ip6: yes :: Enable IPv6, “yes” or “no”. !!
? do-udp: yes :: Enable UDP, “yes” or “no”. !!
? do-tcp: yes :: Enable TCP, “yes” or “no”. !!
? tcp-upstream: no :: upstream connections use TCP only (and no UDP), “yes” or “no” useful for tunneling scenarios, default no. !!
? do-daemonize: yes :: Detach from the terminal, run in background, “yes” or “no”. !!

? username: “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. !!
? directory: “/etc/unbound” :: the working directory. The relative files in this config are relative to this directory. If you give “” the working directory is not changed. !!
? log-time-ascii: no :: print UTC timestamp in ascii to logfile, default is epoch in seconds. !!
? log-queries: no :: print one line with time, IP, name, type, class for every query. !!
? pidfile: “/etc/unbound/unbound.pid” :: the pid file. Can be an absolute path outside of chroot/work dir. !!
? root-hints: “” :: file to read root hints from. get one from ftp://FTP.INTERNIC.NET/domain/named.cache !!
? hide-identity: no :: enable to not answer id.server and hostname.bind queries. !!
? hide-version: no :: enable to not answer version.server and version.bind queries. !!
? identity: “” :: the identity to report. Leave “” or default to return hostname. !!
? version: “” :: the version to report. Leave “” or default to return package version. !!
? target-fetch-policy: “3 2 1 0 0” :: 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 (“”). !!
? harden-short-bufsize: no :: Harden against very small EDNS buffer sizes. !!
? harden-large-queries: no :: Harden against unseemly large queries. !!
? harden-glue: yes :: Harden against out of zone rrsets, to avoid spoofing attempts. !!
? harden-dnssec-stripped: 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-below-nxdomain: no :: Harden against queries that fall under dnssec-signed nxdomain names. !!
? harden-referral-path: 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.
? use-caps-for-id: no :; Use 0x20-encoded random bits in the query to foil spoof attempts. This feature is an experimental implementation of draft dns-0x20. !!
? 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 :: 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-domain: “example.com” :: Allow the domain (and its subdomains) to contain private addresses. local-data statements are allowed to contain private addresses too. !!
? unwanted-reply-threshold: 0 :: 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). !!
? do-not-query-address: 127.0.0.1/8

? do-not-query-address: \:\:1 :: 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-localhost: yes :: if yes, the above default do-not-query-address entries are present. if no, localhost can be queried (for testing and debugging). !!
? prefetch: no :: if yes, perform prefetching of almost expired message cache entries. !!
? prefetch-key: no :: if yes, perform key lookups adjacent to normal lookups. !!
? rrset-roundrobin: no :: if yes, Unbound rotates RRSet order in response. !!
? minimal-responses: no :: if yes, Unbound doesn't insert authority/additional sections into response messages when those sections are not required. !!
? module-config: “validator iterator” :: module configuration of the server. A string with identifiers separated by spaces. “iterator” or “validator iterator” !!
? auto-trust-anchor-file: “/etc/unbound/root.key” :: 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). !!
? dlv-anchor-file: “dlv.isc.org.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 !!
? trust-anchor-file: “” :: 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: “jelte.nlnetlabs.nl. DS 42860 5 1 14D739EB566D2B1A5E216A0BA4D17FA9B038BE4A”
? trust-anchor: “nlnetlabs.nl. DNSKEY 257 3 5
AQPzzTWMz8qSWIQlfRnPckx2BiVmkVN6LPupO3mbz7FhLSnm26n6iG9N
Lby97Ji453aWZY3M5/xJBSOS2vWtco2t8C0+xeO1bc/d6ZTy32DHchpW
6rDH1vp86Ll+ha0tmwyy9QP7y2bVw5zSbFCrefk8qCUBgfHm9bHzMG1U BYtEIQ==” :: 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). !!
? trusted-keys-file: “” :: 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. !!
? domain-insecure: “example.com” ::Ignore chain of trust. Domain is treated as insecure. !!
? val-override-date: “” :: 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-bogus-ttl: 60 :: The time to live for bogus data, rrsets and messages. This avoids some of the revalidation, until the time interval expires. in secs. !!
? val-sig-skew-min: 3600
? val-sig-skew-max: 86400 :: 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-clean-additional: yes :: 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-permissive-mode: no :: 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. !!
? ignore-cd-flag: 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. !!
? val-log-level: 0 :: 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-nsec3-keysize-iterations: “1024 150 2048 500 4096 2500” :: 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. !!
? add-holddown: 2592000 # 30 days :: instruct the auto-trust-anchor-file probing to add anchors after ttl. !!
? del-holddown: 2592000 # 30 days :: instruct the auto-trust-anchor-file probing to del anchors after ttl. !!
? keep-missing: 31622400 # 366 days :: auto-trust-anchor-file probing removes missing anchors after ttl. If the value 0 is given, missing anchors are not removed. !!
? key-cache-size: 4m :: 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-slabs: 4 :: 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. !!
? neg-cache-size: 1m :: 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”. !!
? local-data-ptr: “192.0.2.3 www.example.com” :: 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. !!
? ssl-service-key: “path/to/privatekeyfile.key”
? ssl-service-pem: “path/to/publiccertfile.pem”
? ssl-port: 443 :: 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-upstream: no :: request upstream over SSL (with plain DNS inside the SSL stream). Default is no. Can be turned on and off with unbound-control. !!

Clause python:

Python config section. To enable:

- use * –with-pythonmodule to configure before compiling.
- list python in the module-config string (above) to enable.
- and give a python-script to run.

? python-script: “/etc/unbound/ubmodule-tst.py” :: Script file to load !!

Clause remote-control:

Remote control config section.

? control-enable: no :: Enable remote control with unbound-control(8) here. set up the keys and certificates with unbound-control-setup. !!
? control-interface: 127.0.0.1
? control-interface: \:\:1 :: what interfaces are listened to for remote control. give 0.0.0.0 and \:\:0 to listen to all interfaces. !!
? control-port: 8953 :: port number for remote control operations. !!
? server-key-file: "/etc/unbound/unbound_server.key" :: unbound server key file. !!
? server-cert-file: "/etc/unbound/unbound_server.pem" :: unbound server certificate file. !!
? control-key-file: "/etc/unbound/unbound_control.key" :: unbound-control key file. !!
? control-cert-file: "/etc/unbound/unbound_control.pem" :: unbound-control certificate file. !!

Clause stub-zone:

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.
```

Clause forward-zone:

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.

```
    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
```

Commentaires

Les lignes blanches ne sont pas nécessaires, mais donnent un aspect plus propre.

Voir aussi

Contributeurs principaux : Jamaique.

Basé sur <Titre original de l'article> par <Auteur Original>.