BIND 9.10

Service de nom de domaine

Introduction

BIND (Berkeley Internet Name Domain) implémente un serveur de nom de domaine. Il peut agir en tant que serveur de nom Autoritatif (AA), Serveur Primaire ou esclave, serveur cache, stealth, et avoir plusieurs de ces rôles en même temps.

Ressources requises

Les besoins matériel pour DNS sont généralement modeste, cependant, l'utilisation de DNSSEC peut éprouver les CPU. L'option max-cache-size permet de limiter la quantité de mémoire pour le cache. Additionnellement, l'options max-acache-size peut être utilisé pour limiter la quantité de mémoire utilisée par le mécanisme. C'est une bonne pratique d'avoir suffisamment de mémoire pour charger toutes les zones et données en cache. Cependant, la meilleur manière de déterminer la quantité de mémoire est de regarder le serveur en opérations. Après quelques semaines de traitement, le serveur atteind un niveau relativement stable où les entrées expirent dans le cache aussi vite qu'elles sont insérées.

Opérations du serveur de nom

Cette section décrit de nombreux outils indispensable de diagnostique, d'administration et de supervision disponible pour contrôler et debugger le service.

dig (Domain Information Groper) est un outil complet de recherche. Il a 2 modes : interactif simple, et batch.

host Convertis le noms d'hôte en adresses Internet et inversement, et d'autres fonctionnalités

nslookup nslookup a 2 modes : interactif simple, et non-interactif. Il permet de récupérer de requêter des serveurs de nom.

named-checkconf Vérifie la syntaxe d'un fichier named.conf

named-checkzone Vérifie un fichier maître

named-compilezone Similaire à named-checkzone, mais dump le contenu dans un fichier spécifié

rndc (Remote Name Daemon Control) permet de contrôler les opérations de named

Signaux

SIGHUP Force le serveur à relire named.conf et recharger la base

SIGTERM Force le serveur à se terminer proprement

SIGINT Force le serveur à se terminer proprement

Résolveur léger

traditionnellement, les applications sont liées avec une librairie résolveur qui envoient des requêtes DNS récursives à un serveur de nom cache. IPv6 introduit une nouvelle complexité dans le processus de résolution, tel que les chaînes A6 et les enregistrement DNAME, et la rechercher simultanné IPv4 et IPv6. BIND9 peut cependant fournir des services de résolution aux clients locaux en utilisant une combinaison d'une librairie de résolution légère et un processus de résolution dans l'hôte local. Ils communiquent en utilisant un protocole basé sur UDP qui est distinct et plus léger que le protocole DNS complet.

Pour utiliser l'interface de résolution légère, le système doit lancer les service lwresd ou un serveur de nom local configuré avec la déclaration lwres.

Par défaut, les applications utilisant la libraire de résolution légère vont faire des requêtes sur UDP sur la loopback sur le port 921. Le service ne fait que des recherches DNS, mais dans le future, pourra regarder dans hosts, NIS, etc.

lwresd est essentiellement un serveur de nom cache uniquement. Parce qu'il doit fonctionner sur tous les hôtes, il est conçu pour fonctionner sans configuration ou avec une configuration minimale.

Éléments du fichier de configuration

La liste suivante décris les éléments utilisé dans ce document :

acl_name Le nom d'une address_match_list comme définis par la déclaration acl

address_match_list Une liste d'un ou plusieurs ip_addr, ip_prefix, key_id ou acl_name

masters_list Une liste nommé d'un ou plusieurs ip_addr avec optionnellement key_id et/ou ip_port

domain_name Une chaîne qui est utilisée comme nom DNS

namelist Une liste d'un ou plusieurs éléments domain_name

doted_decimal Un à 4 entiers de 0 à 255 séparés par des '.'

ip4_addr Une adresse IPv4

ip6_addr Une adresse IPv6

ip_addr une ip4_addr ou ip6_addr

ip_dscp Un nombre entre 0 et 63, utilisé pour sélectionner un point de code de services différentiés (DSCP) à utiliser pour le trafique sortant dans les OS qui supportent DSCP.

ip_port Un numéro de port IP (0 à 65535)

ip_prefix Un masque de sous-réseau

key_id Une domain_name représentant le nom d'une clé partagée

key_list Une liste d'une ou plusieurs key_id

number Un nombre entien non-négatif 32bits

path_name Chemin de fichier

port_list Une liste d'un ip_port ou une plage de port

size_spec Entien non-signé 64bits

yes_or_no soit yes soit no

dialup_option yes, no, notify, notify-passive, refresh, ou passive.

Liste de correspondance d'adresses

Syntaxe :

```
address_match_list = address_match_list_element ; [ address_match_list_element; ... ]
address_match_list_element = [ ! ] (ip_address [/length] | key key_id | acl_name | { address_match_list } )
```

Commentaires

Syntaxe :

```
/* ceci est un commentaire */
// ceci est un commentaire
# ceci est un commenentaire
```

Déclaration

acl Définis une liste d'adresse ip nommée

controls Déclare des canaux de contrôle utilisé par rndc

include Inclure un fichier

key Spécifie les information de clé pour l'authentification et l'autorisation

logging Spécifie ce que le serveur log, et où

lwres Configure named pour agir également comme résolveur léger

masters Définis une liste de serveurs maîtres.

options Contrôle les options de configuration globale au serveur

server Définis certaines options par serveur

statistics-channels Déclare des canaux de communication pour avoir accès aux statistiques de named

trusted-keys Définis les clés DNSSEC de confiance

managed-keys Liste les clé DNSSEC à conserver à jour avec rfc5011

view Définis une vue

zone définis une zone

ACL

Les mots clés intégrés sont :

any Matche tous les hôtes

none Ne matche aucun hôte

localhost Matche les IPv4 et IPv6 de toutes les interfaces réseaux dans le système. Cette liste est dynamique

localnets Matche tous les hôtes dans un réseau IPv4 ou IPv6 pour lequel le système à une interface.

Quand BIND 9 est construit avec GeoIP, les acl peuvent également être utilisées pour des restrictions d'accès géographique, avec un élément sous la forme **geoip [db database] field value**. field indique quel champ correspondre (country, region, city, continent, postal, metro, area, tz, isp, org, asnum, domain, et netspeed).

Controls

```
controls {
 [ inet ( ip_addr | * ) [ port ip_port ] allow { address_match_list } keys { key_list }; ] [ inet ...; ]
 [ unix path perm number owner number group number keys { key_list }; ] [ unix ...; ] };
```

La déclaration controls déclare les canaux de contrôle à utiliser par les administrateurs système pour contrôler les opérations du serveur. Ces canaux sont utilisés par rndc. Un canal inet est un socket TCP. Le port par défaut est 953. La capacité à utiliser des commandes dans le canal est contrôlé par les clauses allow et keys. Un canal de contrôle unix est un socket unix écoutant dans le chemin spécifié.

Si aucune déclaration controls n'est spécifiée, named en définis un sur l'interface de bouclage à l'adresse 127.0.0.1 et tente de charger une clé dans rndc.key dans $SYSCONFDIR (spécifié à la compilation). pour désactiver l'utilisation des canaux, créer une déclaration vide : controls { } ;

include

Permet d'inclure d'autres fichiers de configuration

key

La déclaration key définis une clé secrète partagée à utiliser avec TSIG ou un canal de commande. La déclaration key peut être définis globalement ou dans une déclaration view. L'algorithm spécifie l'algorithme utilisé. named supporte hmac-md5, hmac-sha1, hmac-sha224, hmac-sha384 et hmac-sha512.

logging

La déclaration logging configure une variété d'option de logs pour le serveur. Ses canaux associent des méthodes de sortie, options de format de niveaux de sévérité avec un nom qui peut ensuite être utilisé avec la catégorie.

Les catégories sont :

default définis les options de log pour les catégories qui n'ont pas de configuration définie.
general Tout ce qui n'est pas classifié dans des catégories sont définis ici
database Les messages liés aux bases de données utilisé en interne pour stocker les zones et données du cache
security Approbation et refus des requêtes
config traitement de fichier de configuration
resolver Résolution DNS, tel que les recherche récursives
xfer-in Transfer de zone reçus par le serveur
xfer-out Transfer de zone envoyés par le serveur
notify Le protocole NOTIFY
client traitement des demandes client
unmatched Messages que named n'est pas capable de déterminer la classe ou pour lesquels il n'y pas de vue.
network Opérations réseaux
update Mises à jour dynamique
update-security Approbation et refus de demandes de mise à jours
queries reporte les IP des client et numéro de port.
query-erros Informations sur les requêtes résultant de certaines erreurs
dispatch dispatching des packet entrants aux modules
dnssec traitement DNSSEC et TSIG
lame-servers Mauvaises configuration dans le serveurs distants, découvert par bind
delegation-only requêtes qui ont été forcés au NXDOMAIN en résultat d'une zone delegation-only
edns-disabled Requêtes qui ont été forcés à utiliser plain DNS dû à un timeoute.
RPZ Informations sur les erreurs en réponse à des stratégies de fichier de zone

query-errors

la catégorie query-errors est prévue pour un but de débuggage. au niveau de débug 1 ou +, chaque réponse avec le rcode SERVFAIL est loggé comme suit :

```
client 127.0.0.1#61502: query failed (SERVFAIL) for www.example.com/IN/AAAA at query.c:3880
```

Qui signifie une erreur détecté à la ligne 3880 du fichier source query.c. Au debug niveau 2 et +, des informations détaillées de résolution récursive sont loggés :

```
fetch completed at resolver.c:2970 for www.example.com/A in 30.000183: timed out/success [domain:example.com,
referral:2,restart:7,qrysent:8,timeout:5,lame:0,neterr:0, badresp:1,adberr:0,findfail:0,valfail:0]
```

La première partie avent le ' :' montre qu'une résolution récursive pour des enregistrements AAA de www.example.com complétés en 30.000183 secondes et le résultat final a été déterminé à la ligne 2970 du fichier source resolver.c.

La partie suivante montre le résultat final détecté et le dernier résultat de la validation DNSSEC. Dans cet exemple, la requête a échoué parce que tous les serveurs sont down ou inateignable. La dernière partie affiche des informations de statistiques collectés pour cette tentative de résolution :

referral Le nombre de référants que le résolveur a reçu durant le processus de résolution.

restart Le nombre de cycles que le serveur a tenter les serveurs distants du domaine.

qrysent Le nombre de requêtes que le résolveur en envoyés au domaine

timeout Le nombre de timeout depuis que le résolveur a reçu la dernière réponse

lame Le nombre de serveurs lames que le résolveur a détecté soit par une réponse invalide, ou en résultat d'une recherche dans la base d'adresse de BIND9 (ADB), où les serveurs lames sont en cache.

neterr Nombre de résultats erronés que le résolveur a rencontré en envoyant des requêtes au domaine. Peut être dû au serveur inateignable et le résolveur a reçus un ICMP unreachable.

badresp Le nombre de réponses attendus (autre que lame) aux requêtes envoyées par le résolveur au domaine

adberr Erreurs en trouvant des adresses de serveur distant du domaine dans la ADB. Un cas commun est que le nom du serveur distant n'a pas d'adresse

findfail Érreurs en résolvant les adresses de serveur distant. C'est un nombre total d'erreur via le processus de résolution.

valfail Erreurs de validation DNSSEC, dans le processus de résolution.

Au débug niveau 3 et +, les mêmes messages que le niveau 1 sont loggés pour d'autres erreurs que SERVFAIL. Noter que des réponses négatives telles que NXDOMAIN ne sont pas vus comme erreurs ici.

Au débug niveau 4 et +, les même message que le niveau 2 sont loggés pour d'autres erreurs que SERVFAIL.

lwres

```
lwres {
 [ listen-on { ip_addr [port ip_port] [dscp ip_dscp] ;
 [ ip_addr [port ip_port] [dscp ip_dscp] ; ... ] }; ]
 [ view view_name; ]
 [ search { domain_name ; [ domain_name ; ... ] }; ]
 [ ndots number; ]
};
```

Plusieurs déclaration lwres peuvent être configurés avec différentes propriétés. view spécifie la vue ou placer le résolveur. search est équivalent à la déclaration dans /etc/resolv.conf. Elle fournis une liste de domaine qui sont ajoutés aux noms relatifs dans les requêtes. ndots est équivalent à la déclaration dans /etc/resolv.conf, et indique le nombre de '.' minimum dans un nom de domaine relatif qui devrait résulter en un match exact avant que les éléments de search soient ajoutés

masters

```
masters name [port ip_port] [dscp ip_dscp] { ( masters_list |
ip_addr [port ip_port] [key key] ) ; [...] };
```

Les déclarations masters permettent à un jeu commun de masters d'être facilement utilisés par plusieurs zones stub et slaves.

options

La déclaration options définis les options globales utilisées par BIND. cette déclaration peut apparaître seulement une fois dans la fichier de configuration.

```
options {
 [ attach-cache cache_name; ]
 [ version version_string; ]
 [ hostname hostname_string; ]
 [ server-id server_id_string; ]
 [ directory path_name; ]
 [ key-directory path_name; ]
 [ managed-keys-directory path_name; ]
 [ named-xfer path_name; ] // obsolete
 [ tkey-gssapi-keytab path_name; ]
 [ tkey-gssapi-credential principal; ]
 [ tkey-domain domainname; ]
 [ tkey-dhkey key_name key_tag; ]
 [ cache-file path_name; ] //not-used
 [ dump-file path_name; ]
 [ bindkeys-file path_name; ]
 [ secroots-file path_name; ]
 [ session-keyfile path_name; ]
 [ session-keyname key_name; ]
 [ session-keyalg algorithm_id; ]
 [ memstatistics yes_or_no; ]
 [ memstatistics-file path_name; ]
 [ pid-file path_name; ]
 [ recursing-file path_name; ]
 [ statistics-file path_name; ]
 [ zone-statistics full | terse | none; ]
 [ auth-nxdomain yes_or_no; ]
 [ deallocate-on-exit yes_or_no; ] //obsolete
 [ dialup dialup_option; ]
 [ fake-iquery yes_or_no; ]//obsolete
 [ fetch-glue yes_or_no; ] //obsolete
 [ flush-zones-on-shutdown yes_or_no; ]
 [ has-old-clients yes_or_no; ]//obsolete
 [ host-statistics yes_or_no; ]//obsolete
 [ host-statistics-max number; ]//obsolete
 [ minimal-responses yes_or_no; ]
 [ multiple-cnames yes_or_no; ] //obsolete
 [ notify yes_or_no | explicit | master-only; ]
 [ recursion yes_or_no; ]
 [ request-sit yes_or_no; ]
 [ request-nsid yes_or_no; ]
 [ rfc2308-type1 yes_or_no; ]
 [ use-id-pool yes_or_no; ] //obsolete
 [ maintain-ixfr-base yes_or_no; ] //obsolete
```

```
 [ ixfr-from-differences (yes_or_no | master | slave); ]
 [ dnssec-enable yes_or_no; ]
 [ dnssec-validation (yes_or_no | auto); ]
 [ dnssec-lookaside ( auto | no | domain trust-anchor domain ); ]
 [ dnssec-must-be-secure domain yes_or_no; ]
 [ dnssec-accept-expired yes_or_no; ]
 [ forward ( only | first ); ]
 [ forwarders { [ ip_addr [port ip_port] [dscp ip_dscp] ; ... ] }; ]
 [ dual-stack-servers [port ip_port] [dscp ip_dscp] { ( domain_name [port ip_port] [dscp ip_dscp] | ip_addr
[port ip_port] [dscp ip_dscp]) ; ... }; ]
 [ check-names ( master | slave | response ) ( warn | fail | ignore ); ]
 [ check-dup-records ( warn | fail | ignore ); ]
 [ check-mx ( warn | fail | ignore ); ]
 [ check-wildcard yes_or_no; ]
 [ check-integrity yes_or_no; ]
 [ check-mx-cname ( warn | fail | ignore ); ]
 [ check-srv-cname ( warn | fail | ignore ); ]
 [ check-sibling yes_or_no; ]
 [ check-spf ( warn | fail | ignore ); ]
 [ allow-new-zones { yes_or_no }; ]
 [ allow-notify { address_match_list }; ]
 [ allow-query { address_match_list }; ]
 [ allow-query-on { address_match_list }; ]
 [ allow-query-cache { address_match_list }; ]
 [ allow-query-cache-on { address_match_list }; ]
 [ allow-transfer { address_match_list }; ]
 [ allow-recursion { address_match_list }; ]
 [ allow-recursion-on { address_match_list }; ]
 [ allow-update { address_match_list }; ]
 [ allow-update-forwarding { address_match_list }; ]
 [ update-check-ksk yes_or_no; ]
 [ dnssec-update-mode ( maintain | no-resign ); ]
 [ dnssec-dnskey-kskonly yes_or_no; ]
 [ dnssec-loadkeys-interval number; ]
 [ dnssec-secure-to-insecure yes_or_no ;]
 [ try-tcp-refresh yes_or_no; ] // obsolete
 [ allow-v6-synthesis { address_match_list }; ] // obsolete
 [ blackhole { address_match_list }; ]
 [ no-case-compress { address_match_list }; ]
 [ use-v4-udp-ports { port_list }; ]
 [ avoid-v4-udp-ports { port_list }; ]
 [ use-v6-udp-ports { port_list }; ]
 [ avoid-v6-udp-ports { port_list }; ]
 [ listen-on [ port ip_port ] [dscp ip_dscp] { address_match_list }; ]
 [ listen-on-v6 [ port ip_port] [dscp ip_dscp] { address_match_list }; ]
 [ query-source ( ( ip4_addr | * )
  [ port ( ip_port | * ) ]
  [ dscp ip_dscp] |
  [ address ( ip4_addr | * ) ]
  [ port ( ip_port | * ) ] )
  [ dscp ip_dscp] ; ]
 [ query-source-v6 ( ( ip6_addr | * )
  [ port ( ip_port | * ) ]
  [ dscp ip_dscp] |
  [ address ( ip6_addr | * ) ]
  [ port ( ip_port | * ) ] )
  [ dscp ip_dscp] ; ]
 [ use-queryport-pool yes_or_no; ] //obsolete
 [ queryport-pool-ports number; ] //obsolete
```

```
 [ queryport-pool-updateinterval number; ] //obsolete
 [ max-transfer-time-in number; ]
 [ max-transfer-time-out number; ]
 [ max-transfer-idle-in number; ]
 [ max-transfer-idle-out number; ]
 [ tcp-clients number; ]
 [ reserved-sockets number; ]
 [ recursive-clients number; ]
 [ serial-query-rate number; ]
 [ serial-queries number; ] //obsolete
 [ tcp-listen-queue number; ]
 [ transfer-format ( one-answer | many-answers ); ]
 [ transfers-in number; ]
 [ transfers-out number; ]
 [ transfers-per-ns number; ]
 [ transfer-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ transfer-source-v6 (ip6_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ alt-transfer-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ transfer-source-v6 (ip6_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ alt-transfer-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ alt-transfer-source-v6 (ip6_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ use-alt-transfer-source yes_or_no; ]
 [ notify-delay seconds ; ]
 [ notify-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ notify-source-v6 (ip6_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ notify-to-soa yes_or_no ; ]
 [ also-notify { ip_addr [port ip_port] [dscp ip_dscp] [key keyname] ; [ ip_addr [port ip_port] [dscp
ip_dscp] [key keyname] ; ... ] };[ max-ixfr-log-size number; ]
 [ max-journal-size size_spec; ]
 [ coresize size_spec ; ]
 [ datasize size_spec ; ]
 [ files size_spec ; ]
 [ stacksize size_spec ; ]
 [ cleaning-interval number; ] //obsolete
 [ heartbeat-interval number; ]
 [ interface-interval number; ]
 [ statistics-interval number; ]
 [ topology { address_match_list }];
 [ sortlist { address_match_list }];
 [ rrset-order { order_spec ; [ order_spec ; ... ] ] };
 [ lame-ttl number; ]
 [ max-ncache-ttl number; ]
 [ max-cache-ttl number; ]
 [ max-zone-ttl number ; ]
 [ sig-validity-interval number [number] ; ]
 [ sig-signing-nodes number ; ]
 [ sig-signing-signatures number ; ]
 [ sig-signing-type number ; ]
 [ min-roots number; ]
 [ use-ixfr yes_or_no ; ] // obsolete
 [ provide-ixfr yes_or_no; ]
 [ request-ixfr yes_or_no; ]
 [ treat-cr-as-space yes_or_no ; ] // obsolete
 [ min-refresh-time number ; ]
 [ max-refresh-time number ; ]
 [ min-retry-time number ; ]
 [ max-retry-time number ; ]
 [ port ip_port; ]
 [ dscp ip_dscp] ;
```

```
[ additional-from-auth yes_or_no ; ]
[ additional-from-cache yes_or_no ; ]
[ random-device path_name ; ]
[ max-cache-size size_spec ; ]
[ match-mapped-addresses yes_or_no; ] // obsolete
[ filter-aaaa-on-v4 ( yes_or_no | break-dnssec ); ]
[ filter-aaaa-on-v6 ( yes_or_no | break-dnssec ); ]
[ filter-aaaa { address_match_list }; ]
[ dns64 ipv6-prefix {
 [ clients { address_match_list }; ]
 [ mapped { address_match_list }; ]
 [ exclude { address_match_list }; ]
 [ suffix IPv6-address; ]
 [ recursive-only yes_or_no; ]
 [ break-dnssec yes_or_no; ]
}; ];
[ dns64-server name ]
[ dns64-contact name ]
[ preferred-glue ( A | AAAA | NONE ); ]
[ edns-udp-size number; ]
[ max-udp-size number; ]
[ max-rsa-exponent-size number; ]
[ root-delegation-only [ exclude { namelist } ] ; ]
[ querylog yes_or_no ; ]
[ disable-algorithms domain { algorithm;
 [ algorithm; ] }; ]
[ disable-ds-digests domain { digest_type;
 [ digest_type; ] }; ]
[ acache-enable yes_or_no ; ]
[ acache-cleaning-interval number; ]
[ max-acache-size size_spec ; ]
[ clients-per-query number ; ]
[ max-clients-per-query number ; ]
[ masterfile-format (text|raw|map) ; ]
[ empty-server name ; ]
[ empty-contact name ; ]
[ empty-zones-enable yes_or_no ; ]
[ disable-empty-zone zone_name ; ]
[ zero-no-soa-ttl yes_or_no ; ]
[ zero-no-soa-ttl-cache yes_or_no ; ]
[ resolver-query-timeout number ; ]
[ deny-answer-addresses { address_match_list } [ except-from { namelist } ];]
[ deny-answer-aliases { namelist } [ except-from { namelist } ];]
[ prefetch number [number] ; ]
[ rate-limit {
 [ domain domain ; ]
 [ responses-per-second [size number] [ratio fixedpoint] number ; ]
 [ referrals-per-second number ; ]
 [ nodata-per-second number ; ]
 [ nxdomains-per-second number ; ]
 [ errors-per-second number ; ]
 [ all-per-second number ; ]
 [ window number ; ]
 [ log-only yes_or_no ; ]
 [ qps-scale number ; ]
 [ ipv4-prefix-length number ; ]
 [ ipv6-prefix-length number ; ]
 [ slip number ; ]
 [ exempt-clients { address_match_list } ; ]
```

```
  [ max-table-size number ; ]
  [ min-table-size number ; ]
 } ; ]
 [ response-policy {
  zone zone_name ;
  [ policy given | disabled | passthru | drop | nxdomain | nodata | cname domain
  [ recursive-only yes_or_no ; ]
  [ max-policy-ttl number ; ] ;
  [ recursive-only yes_or_no ; ]
  [ max-policy-ttl number ; ]
  [ break-dnssec yes_or_no ; ]
  [ min-ns-dots number ; ]
  [ qname-wait-recurse yes_or_no ; ]
 } ; ]
};
```

attach-cache <string> (options,view)autorise plusieurs vues à partager une seule base de cache. Chaque vue a sa propre base de cache, mais si plusieurs vues ont la même stratégie opérationnelle pour la résolution de nom et de cacheng, ces vues peuvent partager le même cache et sauver de la mémoire et possiblement améliorer l'efficacité de résolution en utilisant cette option. L'implémentation actuelle exige que les vues partageant le même cache soient consistant avec les options suivantes : check-names, cleaning-interval, dnssec-accept-expired, dnssec-validation, max-cache-ttl, max-ncache-ttl, max-cache-size, et zero-no-soa-ttl.

directory <path_name> Le répertoire de travail du serveur. Tout chemins non-absolus dans le fichier de configuration sera pris comme relatif à ce répertoire.

key-directory <path_name> Lors de mise à jours dynamique de zone sécurisés, le répertoire où sont les fichiers de clé DNSSEC et publique/privé (ne concerne pas bind.keys, rndc.key ni session.key)

managed-keys-directory <path_name> Spécifie le répertoire dans lequel stocker les fichiers qui suivent les clés DNSSEC.

tkey-gssapi-keytab <path_name> Le fichier keytab à utiliser pour les mises à jours GSS-TSIG. Si cette option est définie et pas gssapi-credential, les mises à jours seront autorisés avec toute clé matchant un principal dans le keytab

tkey-gssapi-credential <principal> L'accréditif de sécurité avec lequel le serveur devrait authentifier les clés demandées par le protocole GSS-TSIG. Actuellement seul kerberos 5 est supporté, et l'accréditif est un principal kerberos que le serveur peut obtenir via le fichier de clé système définis par tkey-gssapi-keytab. Normalement ce principal est sous la forme 'DNS/server.domain'. tkey-domain doit également être définis si un keytab spécifique n'est pas définis dans tkey-gssapi-keytab

tkey-domain <domainname> Le domaine ajouté à tous les noms de toutes les clés partagées générées avec TKEY.

tkey-dhkey key_name key_tag ; Clé Diffie-Hellman utilisée par le serveur pour générer des clés partagées avec les clients en utilisant le mode dh. Le serveur doit être capable de charger les clés publique et privée depuis les fichiers dans le répertoire de travail courant. Dans la plupart des cas, le nom de clé devrait être le nom d'hôte du serveur.

dump-file path_name Chemin du fichier où le serveur dump la base, invoqué par rndc dumpdb.

memstatistics-file path_name chemin du fichier où le serveur écrit les statistiques d'utilisation mémoire.

pid-file path_name Chemin du fichier pid où le serveur écrit sont pid.

recursing-file path_name chemin du fichier où le serveur dump les requêtes recursives, invoqué par rndc recursing.

statistics-file path_name chemin du fichier où le serveur ajoute des statistiques, invoqué par rndc stats.

bindkeys-file path_name chemin du fichier pour remplacer les clés de confiance intégrée par named.

secroots-file path_name Le chemin du fichier où le serveur dumps les security root, invoqué par rndc secroots.

session-keyfile path_name chemin du fichier dans lequel écrire un clé de session TSIG générée par named à utiliser par nsupdate -l.

session-keyname key_name Le nom de la clé à utiliser pour la clé de session TSIG. défaut : local.ddns.

session-keyalg L'algorithme à utiliser pour la clé de session TSIG. (hmac-sha1, hmac-sha224, hmac-sha256, hmac-sha384, hmac-sha512 et hmac-md5)

port ip_port port UDP/TCP d'écoute du serveur. Défaut : 53

random-device path_name Source d'entropy à utiliser par le serveur

preffered-glue (A | AAAA | NONE) Si spécifié, le type listé sera émis avant d'autre glue dans la section additionnelle d'une réponse.

root-delegation-only [exclude { namelist }] ; Active la délégation-only dans les TLD et les zones root avec une liste d'exclusion optionnelle. Si une zone delegation-only déssert également une zone enfant, il n'est pas toujours possible de déterminer si une réponse vient de la zone delegation-only ou de la zone enfant. Les enregistrements SOA NS et DNSKEY sont des enregistrements apex uniquement et une réponse correspondante qui contient ces enregistrement ou DS est traitée comme venant d'une zone enfant. les enregistrements RRSIG sont également examinés pour voir s'il y a évidence que la réponse vient de la zone enfant. Les réponse déterminée comme venant de la zone enfant ne sont pas convertis en réponse NXDOMAIN. Noter que certains TLD ne sont pas délégation-only.

disable-algorithms domain { algorithm ; [algorithm ;] } ; Désactive les algorithmes DNSSEC spécifiés. Peut être spécifié plusieurs fois. Seul la déclaration match le mieux sera utilisé.

disable-ds-digests domain { digest_type ; [digest_type ;] } ; Désactive le types digests DS/DLV spécifiés. Peut être spécifié plusieurs fois. Seul la déclaration match le mieux sera utilisé.

dnssec-lookaside (auto | no | domain trust-anchor domain) ; Fournis le validateur avec une méthode alternative pour valider les enregistrements DNSKEY en haut de la zone.

dnssec-must-be-secure domain yes_or_no Spécifie les hiérarchies qui doivent être ou non sécurisés (signé et validé). à no, la validation DNSSEC permet des réponses non-sûres.

dns64 ipv6-prefix { [clients { address_match_list } ;] [mapped { address_match_list } ;] [exclude { address_match_list } ;] [suff
Cette directive instruit named de retourner les adresses IPv4 mappées en requêtes AAAA quand il n'y a pas de records AAAA. Prévue pour être utilisé en conjonction avec NAT64. Chaque dns64 définis un préfix DNS64.

exclude définis une liste d'IPv6 qui seront ignorées s'il elles apparaissent dans des enregistrement AAAA du nom de domaine.

suffix définis les bits restants des bits d'adresse IPv4.

recursive-only à yes la synthèse dns64 ne se produit que pour les requêtes récursives.

break-dnssec à yes, la synthèse dns64 se produit même si le résultat, si validé, crée une erreur de validation DNSSEC.

dnssec-update-mode (maintain | no-resign) ; à maintain dans une zone master qui est signée avec DNSSEC et configurée pour les mises à jours dynamiques, et si named a accès à la clé de signature privée de la zone, named signe automatiquement toutes nouveaux enregistrements ou changement et maintient les signature pour la zone en régénérant les records RRSIG quand ils approchent de la date d'expiration. À no-resign, named signe les records mais la maintenance des signatures est désactivées.

max-zone-ttl number Spécifie une valeur maximale de TTL permise. Une zone avec un TTL supérieur est rejeté. C'est utile pour DNSSEC parce qu'en régénérant une nouvelle DNSKEY, l'ancienne clé doit rester disponible jusqu'à ce que les records RRSIG aient expirés des caches. Cette option garantie que le plus grand TTL dans le zone ne sera pas supérieur.

zone-statistics full | terse | none ;] Le serveur collecte des données statistiques dans toutes les zones (full), ou des statistiques minimales (terse). les statistiques sont disponible via les canaux de statistiques, ou rndc stats

automatic-interface-scan yes_or_no Si supporté par l'OS, rescanne automatiquement les interfaces réseaux quand les adresses sont ajoutées et supprimées

allow-new-zones yes_or_no à Yes, les zones peuvent être ajoutées en temps réel, via rndc addzone ou supprimées, via rndc delzone.

auth-nxdomain yes_or_no à yes, le bit AA est toujours mis dans les réponses NXDOMAIN, même si le serveur n'est pas autoritatif. Utile pour de très vieux serveur DNS.

memstatistics yes_or_no Écris les statistiques mémoire dans le fichier spécifié par memstatistics-file en quittant.

dialup dialup_option à yes, le serveur traite toutes les zones comme si elle faisaient du transfert de zone via un lien dialup. cette option peut également être spécifiée dans une vue ou une zone. Si la zone est master, alors le serveur envoie une demande NOTIFY à tous les esclaves. Cela déclenche la vérification du numéro de série de la zone dans l'esclave. Le jeu de serveur qui reçoivent le NOTIFY peut être contrôlé par notify et also-notify. Si la zone est esclave ou stub, le serveur supprime les requêtes de refresh régulières, et les effectue seulement quand heartbeat-interval expire en plus d'envoyer des requêtes NOTIFY. à 'notify', permet d'envoyer seulement des messages NOTIFY, 'notify-passive' envoie des message NOTIFY et supprime les requêtes de refresh normales quand heartbeat-interval expire, et passive qui désactive simplement le traitement refresh normal.

dialup mode	normal refresh	heart-beat refresh	heart-beat notify
no (default)	yes	no	no
yes	no	yes	yes
notify	yes	no	yes
refresh	no	yes	no
passive	no	no	no
notify-passive	no	no	yes

flush-zones-on-shutdown yes_or_no Quand un serveur de nom quitte du à un SIGTERM, vide ou non les écritures de zone en attente.

minimal-responses yes_or_no à yes, en générant les réponses le serveur ajoute seulement les enregistrements de l'autorité, et les sections additionnelles quand elle sont requises. Améliore les performances du serveur.

notify yes_or_no | explicit | master-only à yes, les messages DNS NOTIFY sont envoyés quand une zone est changée. Les messages sont envoyés aux serveurs listés dans les enregistrements NS de la zone, excepté le serveur maître identifié dans le champ SOA MNAME), et à tous serveurs listés dans l'option also-notify. à 'master-only', notifie sont envoyés seulement pour les zones maître. à 'explicit', notifie seulement les serveurs explicitement listés dans also-notify. Peut également être spécifié dans les déclarations de zone.

notify-to-soa yes-on-no à yes, ne vérifie pas les serveurs de nom dans le RRset NS avec les SOA MNAME. Parfois, un slave est listé dans le SOA MNAME, cette option permet de lui envoyer les messages NOTIFY

recursion yes_or_no à yes, le serveur fait le travail nécessaire pour les requêtes récursives DNS et répondre au client. Noter que à no, cela n'empêche pas les clients d'avoir les réponses dans le cache. Le caching peut se produire à cause d'opérations interne du serveur, tel que les recherche d'adresses NOTIFY.

request-nsid yes_or_no à yes, une option EDNS(0) NSID (Name Server Identifier) vide est envoyée avec toutes les requêtes aux serveurs de noms autoritatifs durant la résolution itérative. Si le serveur autoritatif retourne une options NSID dans sa réponse, son contenu est loggé dans la catégorie resolver au niveau info.

request-sit yes_or_no à yes, une option EDNS SIT (Source Identity Token) est envoyée avec la requête. Si le résolveur à précédemment échoué à parler au serveur, le SIT retourné dans la précédente transaction est envoyée. C'est utilisé par le serveur pour déterminer si le résolveur lui a parlé avant.

sit-secret yes_or_no à yes, c'est une clé secrète partagée utilisée pour générer et vérifier les options EDNS SIT dans un cluster anycast. à no, génère un secret aléatoirement au démarrage.

rfc2308-type1 yes_or_no à yes, le serveur envoie des records NS avec le SOA pour les réponses négatives.

provide-ixfr yes_or_no détermine si le serveur local, agissant comme maître, répond avec un transfert de zone incrémental. à yes, le transfert incrémental est fournis quand c'est possible.

request-ixfr yes_or_no Détermine si le serveur local, agissant comme slave, demande des transferts de zone incrémental.

additional-from-auth, additional-from-cache yes_or_no Contrôle le comportement d'un serveur autoritatif en répondant aux requêtes qui ont des données additionnelles, ou en suivant les chaînes CNAME et DNAME. Quand ces 2 options sont a yes, et qu'une requête est répondue depuis une donnée autoritative, la section data additionnelle de la réponse sera remplie en utilisant les données d'autres zones autoritatives et depuis le cache. Éviter la recherche depuis ces données additionnelles accélèrent les opérations du serveur. Ces options sont prévues pour être utilisée uniquement dans les serveurs ou vues authoritative-only, Tenter de les mettre à no sans spécifier recursion no cause le serveur à ignorer les options et logger un message d'erreur.

filter-aaaa-on-v4 yes_or_no | break-dnssec Cette option est disponible si bind9 est compilé avec –enable-filter-aaaa. Aide la transition ipv4 vers ipv6 en ne donnant pas d'adresses IPv6 aux clients DNS sauf s'ils ont des connections IPv6 Internet.

filter-aaaa-on-v6 Identique, excepté qu'il filtre les réponses AAAA aux requêtes depuis les clients ipv6 au lieu des clients IPv4. Pour filtrer toutes les réponses, définir les 2 options à yes.

ixfr-from-differences yes_or_no | master | slave À yes, si le serveur charge une nouvelle version d'une zone maître depuis son fichier de zone ou reçois une nouvelle version d'un fichier slave via un transfert de zone, il compare la nouvelle version à la précédente et calcule un jeu de différences. La différence est ainsi loggées dans le fichier journal de la zone pour que les changements puissent être transmis aux slaves comme transfert de zone incrémental. accepte également 'master' et 'slave' aux niveaux vue et option qui permet d'activer pour toutes les zones master ou slaves respectivement.

multi-master yes_or_no Devrait être activé quand il y'a plusieurs serveurs maîtres pour une zone. À yes, named ne log rien quand le numéro de série dans le maître est inférieur à ce que named à.

dnssec-enable yes_or_no Active le support DNSSEC dans named.

dnssec-validation yes_or_no Active la validation DNSSEC dans named.

dnssec-accept-expired yes_or_no Accepte les signatures expirées en vérifiant les signature DNSSEC. Définir à yes laisse named vulnérable aux attaques replay.

querylog yes_or_no Spécifie si le query logging devrait être démarré quand named démarre. Si querylog n'est pas spécifié, alors le query logging est déterminé par la présence de la catégorie de logging queries.

check-names (master | slave | response) (warn | fail | ignore) ; Cette option est utilisée pour restreindre le jeu de caractère et syntaxe de certains nom de domaine dans les fichiers master et/ou dans les réponse DNS reçues du réseaux. Le défaut varie en accord avec l'utilisation. Pour les zones maître, le défaut est 'fail'. Pour les zones slave, le défaut est 'warn'. Pour les réponses reçues, le défaut est 'ignore'.

check-dup-records (warn | fail | ignore) ; Vérifie les zones maître pour les enregistrements qui sont traités différemment par DNSSEC mais sémantiquement égaux en DNS plain.

check-mx (warn | fail | ignore) ; vérifie si l'enregistrement MX apparaît pour référrer à une IP.

check-wildcard yes_or_no Cette option est utilisée pour vérifier les wildcard non terminals, qui sont généralement le résultat d'une erreur de compréhension de l'algorithme de matching wildcard. Cette option affectes les zones master.

check-integrity yes_or_no Effectue des vérifications d'intégrité avant de charger une zone. Cela vérifie que les records MX et SRV réfèrent à une adresse. Pour les records MX et SRV, seuls les noms d'hôte dans la zone sont vérifiés. Pour les records NS, seuls les noms sous le top of zone sont vérifiés.

check-mx-cname (warn | fail | ignore) ; définis le comportement de check-integrity en vérifiant les enregistrement MX qui réfèrent à des CNAME.

check-srv-cname (warn | fail | ignore) ; définis le comportement de check-integrity en vérifiant les enregistrement SRV qui réfèrent à des CNAME.

check-sibling yes_or_no vérifie également si des sibling glue existent.

check-spf (warn | fail | ignore) ; définis le comportement de check-integrity en vérifiant que 2 formes de record Sender Policy Framework (records TXT commençant avec 'v=spf1' et SPF) existent ou non.

zero-no-soa-ttl yes_or_no En retournant des réponses autoritatives négatives au demandes SOA, définis le TTL de l'enregistrement SOA retourné dans la section autorité à 0.

zero-no-soa-ttl-cache yes_or_no En cachant une réponse négative d'une requête SOA définis le TTL à zero.

update-check-ksk yes_or_no à yes, vérifie le bit ksk dans chaque clé pour déterminer comment la clé devrait être utilisée en générant les RRSIGs pour une zone sécurisée. Normalement, les clés de signature de zone (c'est à dire les clés avec le bit ksk mis) sont utilisés pour signer toute la zone, alors que les clé de signature de clé (les clés avec le bit ksk mis) sont seulement utilisé pour signer le RRset DNSKEY dans la zone apex. Cependant, si cette option est à no, le bit ksk est ignoré, les ksk sont traités comme s'ils étaient des ZSK et sont utilisé pour signer toute la zone.

dnssec-dnskey-kskonly yes_or_no a yes et update-check-ksk à yes, seul les clés de signature de clé seront utilisées pour signer les RRset DNSKEY dans la zone apex. Les clés de signature de zone seront utilisées pour signer le reste de la zone, mais par le RRset DNSKEY.

dnssec-loadkeys-interval Quand une zone est configurée avec auto-dnssec maintain, sont dépôt de clé doit être vérifié périodiquement pour voir si une nouvelle clé a été ajoutée ou la métadonnée de timing d'une clé existant a été mise à jours. cette option définis la fréquence de vérification automatique, en minute. défaut : 60 (de 1 à 1440)

try-tcp-refresh yes_or_no Tente de rafraîchir la zone en utilisant TCP si les requêtes UDP échouent. pour compatibilité avec BIND8.

dnssec-secure-to-insecure yes_or_no Permet à une zone dynamique de transiter d'une zone sécure à une zone insécure en supprimant tous les records DNSKEY.

Forwarding

Les options suivantes peuvent être utilisées pour créer un grand cache sur quelques serveurs, réduisant le trafique sur les liens vers les serveurs de nom externe. Le forwarding peut également être utilisé pour autoriser les requêtes par les serveur qui n'ont pas d'accès directe à Internet, mais souhaitent rechercher des noms exterieurs. Le forwarding se produit seulement dans le requêtes pour lequels le serveur n'est pas autoritatif et n'a pas de réponse dans son cache.

forward only | first Cette option est seulement significative si une liste de forwarders n'est pas vide. À 'first', le serveur requête les forwarders en premier, puis recherche la réponse par lui-même. À 'only', le serveur ne requête que les forwarders.

forwaders { [ip_addr [port ip_port] [dscp ip_dscp] ; ...] }; Spécifie les adresses IP à utiliser pour le forwarding. Peut également être configuré par domaine.

Dual-stack

Les serveurs dual-stack sont utilisés comme serveurs de secours pour fonctionner lors de problèmes d'accessibilité du à un manque de support pour IPv4 ou IPv6 sur la machine hôte.

dual-stack-servers Spécifie les noms d'hôte ou adresses des machine avec un accès aux transport IPv4 et IPv6. Si un nom d'hôte est utilisé, le serveur doit être capable de résoudre le nom en utilisant seulement le transport qu'il a. Si la machine est dual-stackée, alors cette option n'a pas d'effet sauf si un accès à un transport a été désactivé.

Contrôle d'accès

allow-notify Spécifie quels hôtes sont autorisés à notifier ce serveur, un slave, ou les changements de zone en plus des zones maître. Peut également être spécifié dans une déclaration zone. Est seulement significatif pour une zone slave. Si non spécifié, traite les messages de notification seulement pour les zones maître.

allow-query Spécifie quels hôtes sont autorisés à requêter le serveur. Peut également être spécifié dans une zone.

allow-query-on Spécifie quelles adresses locales peuvent accepter les requêtes DNS. Uniquement vérifié pour les requêtes qui sont permises par allow-query. Peut être spécifié dans les déclarations de zone

allow-query-cache Spécifie quels hôtes peuvent obtenir les réponses depuis les caches. défaut : localnets ; localhosts ;

allow-recursion spécifie quels hôtes sont autorisés à faire de requêtes récursives sur ce serveur. défaut : localnets ; localhosts ;

allow-recursion-on Spécifie quelles adresses peuvent accepter des requêtes récursives

allow-update Spécifie quels hôtes sont autorisés à envoyer des mises à jours Dynamic Updates pour les zones maître.

allow-update-forwarding Spécifie quels hôtes sont autorisés à envoyer des mises à jours Dynamic Uptates aux zones esclaves à forwarder au master. défaut none, peut être any, d'autres valeurs n'ont pas de sens, vu que c'est de la responsabilité du contrôle d'accès sur serveur maître de gérer ça.

allow-transfer Spécifie quels hôtes sont autorisés à recevoir les transferts de zone du serveur. Peut également être spécifié dans les zones.

blackhole Spécifie une liste d'adresses auxquelles le serveur refusera de répondre ou d'utiliser pour résoudre une requête.

filter-aaaa Spécifie une liste d'adresses auxquelles filter-aaaa-on-v4 s'applique

no-case-compress Spécifie une liste d'adresses qui nécessite que les réponses utilisent la compression sensible à la casse. Cet ACL peut être utilisée quand named doit travailler avec des client qui ne sont pas compliant rfc1034.

resolver-query-timeout La quantité de temps que le resolver passe à tenter de répondre à une requête récursive avant d'échouer, en seconde.

Interfaces

listen-on [port port_num] { acl_name ; } Les interfaces et ports que le serveur utilise pour répondre aux requêtes peut être spécifié ici. Il s'agit d'adresses IPv4. Plusieurs déclarations peuvent être définis.

listen-on-v6 [port port_num] { acl_name ; } Idem pour les adresse IPv6

adresse de requête

query-source address * port * ; Si le serveur ne connaît pas la réponse à une question, il va requêter d'autres serveurs de nom. Cette option spécifie l'adresse et le port utilisé pour de telles requêtes.

query-source-v6 address * port * ; Idem pour ipv6

Transfert de zone

Bind a des mécanismes en place pour faciliter les transferts de zone et définir des limites sur la quantité de charge de ce transfert dans le système.

also-notify Définis une liste d'adresses de serveurs de nom à qui envoyer également les messages NOTIFY. Cela permet de s'assurer que les copies de zone sont rapidement convergés sur les serveurs Stealth. Optionnellement un port peut être spécifié avec chaque adresse. Une clé TSIG optionnelle peut aussi être spécifié avec chaque adresse. Les listes masters peuvent être utilisées pour cela.

max-transfer-time-in termine les transferts de zone entrants durant plus longtemps que ce délai en minute. défaut 120, max 40320

max-transfer-idle-in termine les transferts de zone entrants qui ne progressent plus depuis ce délai en minute. défaut 60

max-transfer-time-out termine les transferts de zone sortants durant plus longtemps que ce délai en minute. défaut 120, max 40320

max-transfer-idle-out termine les transferts de zone sortants qui ne progressent plus depuis ce délai en minute. défaut 60

serial-query-rate Les serveurs slave vont périodiquement requêter le master pour trouver si des numéros de zone ont changés. Chacune de ces requêtes utilise une quantité de bande passante du serveur slave. Cette options définis le nombre maximum de requêtes envoyées pas secondes. défaut : 20. contrôle également le taux d'émission des messages NOTIFY pour les zone slave et master.

transfer-format Les transferts de zone peuvent être envoyés en 2 format : one-answer et many-answers. Ce dernier est plus efficace mais seulement supporté par les serveur DNS récents.

transfers-in Nombre maximum de transferts de zone entrants qui peuvent être lancés simultanément. défaut : 10. Des valeurs plus grandes peuvent accelerer la convergence des zones slaves, mais augmente la charge du système.

transfer-out Le nombre maximum de transferts de zone sortants qui peuvent être lancés simultanément. défaut : 10. Les demandes de transfert de zone au delà de cette limite seront refusés.

transfer-per-ns Le nombre maximum de transferts de zone qui peuvent être lancés simultanément depuis un serveur de nom. défaut : 2. Des valeurs plus grandes peuvent accelerer la convergence des zones slaves, mais augmente la charge du système.

transfer-source Détermine quelles adresses locales seront utilisée pour les connections IPv4 TCP utilisées pour le transfert de zone entrant sur le serveur. Détermine également l'adresse IPv4 et optionnellement le port UDP, utilisé pour les requêtes de rafraîchissement et le forward de mises à jours dynamique.

transfer-source-v6 Idem, sur ipv6

notify-source Détermine quelles adresses locale source, et optionnellement le port UDP sera utilisé pour envoyer des messages NOTIFY. Cette adresse doit apparaître dans les masters de zone des serveurs slave ou dans une clause allow-notify.

notify-source-v6 idem, sur IPv6

Liste de ports UDP

use-v4-udp-ports { range 1024 65535 ; } ; Spécifie les port udp à utiliser pour récupérer la plage de ports éphémère.

use-v6-udp-ports { range 1024 65535 ; } ; idem sur ipv6

avoid-v4-udp-ports {} ; Permet d'excluse une plage de port

avoid-v6-udp-ports {} ; Idem sur ipv6

Limites de ressource système

Les ressources du système utilisées par le serveur peuvent être limités. la valeur 'unlimited' désactive la limite, ou utilise la valeur maximale.

coresize Taille maximum d'un core dump.

datasize quantité maximum de données mémoire que le serveur peut utiliser. C'est une limite hard.

files Nombre maximum de fichiers que le serveur peut ouvrir simultanément.

stacksize Quantité de mémoire de pile que le serveur peut utiliser.

Limites de ressources serveur

Les options suivantes définissent les limites de la consommation de ressource du serveur forcés en interne par le serveur au lieu de l'OS

max-journal-size Définis une taille maximum pour chaque fichier journal. À l'approche de cette taille, les anciens enregistrements sont supprimés. max 2Go.

recursive-clients nombre maximum de recherches récursives simultanés que le serveur effectue à la demande des client. défaut : 1000. Chaque client récursif utilise environ 20Ko de mémoire.

tcp-clients Nombre maximum de connexions TCP simultanés que le serveur accepte. défaut : 100

reserved-sockets Nombre de descripteur de fichier réservés pour TCP, stdio, etc. Doit être suffisant pour couvrir le nombre d'interfaces utilisées par named. Défaut 512, minimum 128, et maximum = maxsockets -128

max-cache-size Quantité de mémoire à utiliser pour le cache du serveur, en octets. Quand la limite est atteinte, les enregistrement expirent prématurément.

tcp-listen-queue Profondeur de file d'écoute. défaut et minimum : 10. Si le kernel support le filtre d'acceptation 'dataready', cela contrôle également combien de connexions TCP seront mis en file dans l'espace kernel en attendant que les données soient acceptés.

Intervals de tâches périodique

heartbeat-interval Le serveur effectue des tâche de maintenance de zone pour toutes les zones marquées dialup. défaut : 60minutes.

interface-interval Le serveur scanne la liste des interfaces réseaux. défaut 60minutes

statistics-interval les statistiques du serveur sont loggés à cet interval. défaut : 60minutes

topology

Quand le serveur choisi un serveur de nom à requêter depuis une liste de serveurs de noms, il préfère celui qui est topologiquement plus proche de lui. La déclaration topologie prend une liste d'adresse, et l'interprète. La position d'une adresse dans la liste indique sa distance. Le premier élément de la liste est le plus proche.

```
topology {
 10/8;
 !1.2.3/24;
 { 1.2/16; 3/8; };
};
```

sortlist

La réponse à une requête DNS peut consister de plusieurs RR formant un RRset. Le serveur de nom va normalement retourner les RR dans le RRset dans un ordre indéterminé. Le résolveur client devrait réarranger les RR de manière approprié, en utilisant les adresses dans le réseau local de préférence. Cependant, tous les résolveurs ne peuvent pas le faire correctement. Quand un client utilise un serveur local, le trie peut être effectué dans le serveur, basé sur l'adresse du client. Cela nécessite seulement de configurer les serveurs de nom, pas les clients.

La déclaration sortlist prend une address_match_list et l'interprète même plus spécifiquement que la déclaration topology. Chaque déclaration doit être un address_match_list avec 1 ou 2 éléments. Le premier élément (qui peut être uni adresse IP, un préfixe IP, un nom d'ACL ou une address_match_list imbriquée) de chaque liste est vérifiée avec l'adresse sources de la requête jusqu'à ce qu'un match est trouvé.

Une fois l'adresse source de la requête est matchée, si la déclaration contient seulement un élément, l'élément est placé en premier dans la réponse. Si la déclaration est une liste de 2 éléments, le second élément est traité de la même manière que dans la déclaration topology. Chaque liste a une distance assignée et l'adresse dans la réponse avec la distance minimum est placée au début de la réponse.

Dans l'exemple suivant, les requêtes reçues depuis une des adresses de l'hôte lui-même aura une réponse d'adresse préférés sur le réseau local. Ensuite, les adresses préférées sont les adresses dans le réseaux 192.168.1/24, puis soit 192.168.2/24 soit 192.168.3/24. Les requêtes

reçues depuis un hôte dans le réseaux 192.168.4/24 ou 192.168.5/24 auront d'autres adresse sur ces réseaux. Les requêtes reçues depuis un hôte dans le réseau 192.168.1/24 va préférer d'autres adresses dans le réseaux 192.168.2/24 et 192.168.3/24. Les requêtes reçues depuis un hôte dans le réseau 192.168.4/24 ou 192.168.5/24 vont seulement préférer d'autres adresse dans leur réseau respectif.

```
sortlist {
 // IF the local host THEN first fit on the following nets
 { localhost;
  { localnets;
   192.168.1/24;
   { 192.168.2/24; 192.168.3/24; }; }; };
 // IF on class C 192.168.1 THEN use .1, or .2 or .3
  { 192.168.1/24;
   { 192.168.1/24;
    { 192.168.2/24; 192.168.3/24; }; }; };
 // IF on class C 192.168.2 THEN use .2, or .1 or .3
 { 192.168.2/24;
  { 192.168.2/24;
   { 192.168.1/24; 192.168.3/24; }; }; };
 // IF on class C 192.168.3 THEN use .3, or .1 or .2
 { 192.168.3/24;
  { 192.168.3/24;
   { 192.168.1/24; 192.168.2/24; }; }; };
 // IF .4 or .5 THEN prefer that net
 { { 192.168.4/24; 192.168.5/24; };
 };
};
```

rrset-order

Quand plusieurs records sont retournés dans une réponse, il peut être utile de configurer l'ordre des records placés dans la réponse. La déclaration rrset-order permet la configuration de l'ordre des records dans une réponse à plusieurs records. Un orders_spce est définis comme suit :

```
[class class name] [type type name] [name "domain name"] order ordering
```

Si aucune classe n'est spécifiée, le défaut est ANY. Si aucun type n'est spécifié, le défaut est ANY. Si aucun nom n'est spécifié, le défaut est '*'. Les valeurs légales pour ordering sont :

fixed Les records sont retournés dans l'ordre qui ont été définis dans le fichier de zone.

random Les records sont retourné aléatoirement

cyclic Les records sont retourné en cycle round-robin.

Exemple :

```
rrset-order {
 class IN type A name "host.example.com" order random;
 order cyclic;
};
```

Force les réponse pour les enregistrements de type A dans la classe IN ayant host.example.com comme suffixe, à toujours être retourné aléatoirement. Les autres enregistrements sont retournés de manière cyclique. Si plusieurs rrest-order sont spécifiés, ils ne sont pas combinés, le dernier s'applique.

tuning

lame-ttl Définis le ttl de mise en cache des serveur lame. 0 désactive la mise en cache. défaut 600, max 1800. Contrôle également la quantité de temps d'erreurs de validation DNSSEC qui sont mis en cache.

max-ncache-ttl Pour réduire le trafic réseau et augmenter les performances, le serveur stocke les réponses négative. Cette option définis la durée de rétention pour ces réponses dans le serveur en secondes. défaut : 10800, max 7 jours.

min-roots Nombre minimum de serveurs root requis pour une requête pour que les serveurs root soient acceptés. défaut : 2

sig-validity-interval Spécifie le nombre de jours dans le future d'expiration des signatures DNSSEC générées automatiquement en résultat à des mises à jours dynamique. Il y a un second champs optionnel qui spécifie le délai de régénération des signatures avant expiration. Le second champ est spécifié en jours si l'interval de base est supérieur à 7 jours, sinon il est spécifié en heures. l'interval de base est de 30 jours, donnant une re-signature de 7 jours et demi. Les valeurs maximum sont 10ans. 3660 jours.

sig-signing-nodes Spécifie le nombre maximum de nœuds à examiner dans chaque quantum en signant une zone avec une nouvelle DNSKEY. défaut : 100

sig-signing-signatures Spécifie un seuil de signatures qui vont terminer le traitement d'un quantum en signant une zone avec une nouvelle DNSKEY. défaut : 10.

sig-signing-type Spécifie un type RDATA privé à utiliser en génèrent des records d'état de signature. défaut : 65534. Il est prévu que ce paramètre soit supprimé une fois qu'il y aura un type standard.

min-refresh-time, max-refresh-time, min-retry-time, max-retry-time Ces options contrôlent le comportement du serveur en rafraîchissant une zone ou en retentant des transferts échoués. Généralement, les valeurs SOA pour la zone sont utilisés, mais ces valeurs sont définis par le master, donnant au serveur administrateurs de serveurs slaves peut de contrôle sur leur contenu. Ces options permettent à l'administrateur de définir un temps et tentative de refresh minimum et maximum soit par zone, par vue, ou globalement. Ces options sont valides pour les zone slaves et stub. Défaut : 300, 2419200, 500, 1209600

edns-udp-size Définis la taille de tampon UDP EDNS initial, pour contrôler la taille des paquets reçus depuis les serveurs autoritatifs en réponse aux requêtes récursives. Les valeurs valides sont 512 à 4096 (défaut). changer cette valeur est d'avoir des réponses udp à passer via des firewalls qui bloquent les paquets fragmentés et/ou bloquent les paquets DNS UDP supérieurs à 512 octets.

max-udp-size Définis la taille maximale de message UDP EDNS à envoyer en octets. Les valeurs valides sont 512 à 4096 (défaut). baisser cette valeur encourage l'utilisation de TCP.

masterfile-format Spécifie le format de fichier des fichiers de zone. Défaut : text, qui est la représentation standard, excepté pour les zone slaves, dans lequel le défaut est raw. Les autres formats sont à générer avec named-compilezone, ou dumpé par named.

clients-per-query, max-clients-per-query Définis la valeur initiale (minimal) et le nombre maximum de clients récursifs simultanément pour une requête données (<qname,qtype,qclass>) que le serveur va accepter avant de de supprimer les clients additionnels. défaut : 10 et 100.

notify-delay Délai en secondes entre l'envoie d'un jeu de messages notify pour une zone. défaut 5

max-rsa-exponent-size Taille de l'exposant RSA maximum, en bits, qui sera accepté lors de la validation. de 35 à 4096bits.

prefetch Quand une requête est reçue pour une données cachée qui va expirer sous peut, named peut rafraîchir la donnée depuis le serveur autoritatif immédiatement, s'assurant que le cache a toujours la réponse disponible. Cette option spécifie le TTL déclencheur : quand une donnée en cache avec un TTL inférieur est rencontré durant le traitement d'une requête, elle sera rafraîchie. de 1 à 10secondes. Un second argument optionnel spécifie le TTL eligible : la plus petite valeur TTL original qui sera accepté pour un record pour être éligible au prefetching. Ce TTL doit être d'au moins 6 secondes de plus que le TTL déclencheur. défaut : 2 9.

informations de zones

Le serveur fournis des informations de diagnostique utile au travers de zone intégrées sous le pseudo domain bind, dans la classe CHAOS. Ces zones font partie d'une vue intégrée de classe CHAOS qui est séparée de la vue par défaut de classe IN. La plupart des options de configurations globales s'appliquent à cette vue, mais certaines sont remplacées en interne : notify-recursion et allow-new-zones sont toujours à no, et rate-limit est mis pour autoriser 3 réponses par seconde. Pour désactiver ces zones, utiliser les options ci-dessous, ou cacher la vue CHAOS en définissant une vue explicite de classe CHAOS qui matche tous les clients.

version La version du serveur à reporter via une requête du nom version.bind avec le type TXT de classe CHAOS. Le défaut est le vrai numéro de version est utilisé. spécifier version none désactive le traitement des requêtes.

hostname Le nom d'hôte que le serveur devrait reporter via une requête du nom hostname.bind avec le type TXT, classe CHAOS. hostname none désactive le traitement des requêtes

server-id L'id que le serveur devrait reporter en reçevant une requête NSID (Name Server IDentifier), ou une requête de nom ID.SERVER avec le type TXT, classe CHAOS. server-id none le désactive.

Zones vide embarquées

named a des zones embarquées vides. Ces zones devraient normalement être répondues localement et ne devraient pas être envoyées vers les serveurs root. En particulier, ces zones couvrent les espaces de nom pour les adresses des rfc1918, rfc5193, rfc5737 et rfc6598. Elles incluent l'espace de nom inversé pour les adresse locales IPv6, et les adresse de lien local, l'adresse de bouclage IPv6 et les adresses IPv6 inconnues. La liste actuelle est :

10.IN-ADDR.ARPA
16.172.IN-ADDR.ARPA
17.172.IN-ADDR.ARPA
18.172.IN-ADDR.ARPA
19.172.IN-ADDR.ARPA
20.172.IN-ADDR.ARPA
21.172.IN-ADDR.ARPA
22.172.IN-ADDR.ARPA
23.172.IN-ADDR.ARPA
24.172.IN-ADDR.ARPA
25.172.IN-ADDR.ARPA
26.172.IN-ADDR.ARPA
27.172.IN-ADDR.ARPA
28.172.IN-ADDR.ARPA
29.172.IN-ADDR.ARPA
30.172.IN-ADDR.ARPA
31.172.IN-ADDR.ARPA
168.192.IN-ADDR.ARPA
64.100.IN-ADDR.ARPA
65.100.IN-ADDR.ARPA
66.100.IN-ADDR.ARPA
67.100.IN-ADDR.ARPA
68.100.IN-ADDR.ARPA
69.100.IN-ADDR.ARPA
70.100.IN-ADDR.ARPA
71.100.IN-ADDR.ARPA
72.100.IN-ADDR.ARPA
73.100.IN-ADDR.ARPA
74.100.IN-ADDR.ARPA
75.100.IN-ADDR.ARPA
76.100.IN-ADDR.ARPA
77.100.IN-ADDR.ARPA
78.100.IN-ADDR.ARPA
79.100.IN-ADDR.ARPA
80.100.IN-ADDR.ARPA
81.100.IN-ADDR.ARPA

82.100.IN-ADDR.ARPA
83.100.IN-ADDR.ARPA
84.100.IN-ADDR.ARPA
85.100.IN-ADDR.ARPA
86.100.IN-ADDR.ARPA
87.100.IN-ADDR.ARPA
88.100.IN-ADDR.ARPA
89.100.IN-ADDR.ARPA
90.100.IN-ADDR.ARPA
91.100.IN-ADDR.ARPA
92.100.IN-ADDR.ARPA
93.100.IN-ADDR.ARPA
94.100.IN-ADDR.ARPA
95.100.IN-ADDR.ARPA
96.100.IN-ADDR.ARPA
97.100.IN-ADDR.ARPA
98.100.IN-ADDR.ARPA
99.100.IN-ADDR.ARPA
100.100.IN-ADDR.ARPA
101.100.IN-ADDR.ARPA
102.100.IN-ADDR.ARPA
103.100.IN-ADDR.ARPA
104.100.IN-ADDR.ARPA
105.100.IN-ADDR.ARPA
106.100.IN-ADDR.ARPA
107.100.IN-ADDR.ARPA
108.100.IN-ADDR.ARPA
109.100.IN-ADDR.ARPA
110.100.IN-ADDR.ARPA
111.100.IN-ADDR.ARPA
112.100.IN-ADDR.ARPA
113.100.IN-ADDR.ARPA
114.100.IN-ADDR.ARPA
115.100.IN-ADDR.ARPA
116.100.IN-ADDR.ARPA
117.100.IN-ADDR.ARPA
118.100.IN-ADDR.ARPA
119.100.IN-ADDR.ARPA
120.100.IN-ADDR.ARPA
121.100.IN-ADDR.ARPA
122.100.IN-ADDR.ARPA
123.100.IN-ADDR.ARPA
124.100.IN-ADDR.ARPA
125.100.IN-ADDR.ARPA
126.100.IN-ADDR.ARPA
127.100.IN-ADDR.ARPA

0.IN-ADDR.ARPA

127.IN-ADDR.ARPA

254.169.IN-ADDR.ARPA

2.0.192.IN-ADDR.ARPA

100.51.198.IN-ADDR.ARPA

113.0.203.IN-ADDR.ARPA

255.255.255.255.IN-ADDR.ARPA

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

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

8.B.D.0.1.0.0.2.IP6.ARPA

D.F.IP6.ARPA

8.E.F.IP6.ARPA

9.E.F.IP6.ARPA

A.E.F.IP6.ARPA

B.E.F.IP6.ARPA

Les zones vides sont définissables au niveau de la vue et s'appliquent seulement aux vues de classe IN. Désactiver les zones vides sont seulement hérités depuis les options s'il n'y a pas de zones vide désactivée spécifiée au niveau de la vue. Pour remplacer la liste des zones désactivées, on peut désactiver la zone root au niveau de la vue, par exemple : disable-empty-zone "." ;

empty-server spécifie que le nom de serveur va apparaître dans le record SOA retourné pour les zones vide. Si rien n'est spécifié, le nom de zone est utilisé

empty-contact Spécifie quel nom de contact appparaît dans le record SOA retourné pour les zones vides. Si rien n'est spécifié, '.' est utilisé

empty-zones-enables Active ou désactive les zones vide. Par défaut, elles sont activés.

disable-empty-zone Désactive des zones vide individuellement. Par défaut aucune n'est désactivée. Peut être spécifié plusieurs fois.

cache de section additionnel

Le cache additionnel, également appelée acache, est un cache interne pour améliorer les performances de réponse de bind9. Quand une section de cache additionnel est activé, bind9 cache un raccourci interne vers la section additionnelle pour chaque réponse RR. Noter que acache est un mécanisme interne à bind9.

Le cache additionnel ne change par le contenu de la réponse excepté l'ordre RRset de la section additionnelle, mais peut améliorer les performances de réponse significativement. C'est particulière efficace pour les serveurs autoritatifs pour une zone qui a de nombreuses délégations avec de nombreux glue RR.

Pour obtenir un maximum de performances, définir additional-from-cache à no est recommandé, vu que l'implémentation actuelle de acache ne crée pas de raccourcis depuis le cache de données DNS. Un désavantage de acache est qu'il nécessite plus de mémoire. La cache de section additionnel a un effet mineur sur l'ordre RRset dans la section additionnelle. sans acache, l'ordre cyclic est effectif pour la section additionnelle aussi bien que les sections réponse et autorité. Cependant, le acache fixe l'ordre quand il est caché la première fois, et le même ordre est concervé ensuite sans regarder le rrset-order. L'effet devrait être minime vu qu'un RRset dans la section additionnelle ne contient généralement que peu de RR dans lequel l'ordre n'est pas important.

acache-enable active le acache. défaut : no.

acache-cleaning-interval en minute. Vide les entrées gelées du cache. défaut : 60

max-acache-size Taille maximale en Mo à utiliser pour le acache.

filtrage de contenu

bind9 fournis la capacité de filtrer les réponses DNS des serveurs DNS externes contenant certains types de données dans le section réponse. Spécifiquement, il peut rejeter les adresses A ou AAAA si les adresses correspondantes matchent un liste donnée de l'option deny-answer-addresses. Il peut également rejeter les CNAME ou DNAME si le nom alias matche la liste de deny-answer-aliases. Si l'option namelist est spécifiée avec except-from, les records dont le nom requêté matche la liste seront acceptés sans regarder le filtre. De même, si le nom alias est un sous-domaine de la zone correspondancte, le filtre deny-answer-aliases ne s'applique pas ; par exemple, même si example.com est spécifié pour deny-answer-aliases, www.example.com. CNAME xxx.example.com. retourné par un serveur example.com sera accepté.

deny-answer-addresses { address_match_list } [except-from { namelist }] ;] Définis les adresses ignorées et les exceptions

deny-answer-aliases { namelist } [except-from { namelist }] ; Rejète les records CNAME ou DNAME si le nom alias matche la liste

Ré-écriture de zone de stratégie de réponse (RPZ)

Bind9 inclus un mécanisme limité pour modifier les réponses DNS pour les demandes analogues aux blacklists DNS des anti-spams. Les réponses peuvent être changées pour refuser l'existance des domaines (NXDOMAIN), refuser l'existance d'adresses IP pour les domaines (NODATA), ou contenir d'autres adresses IP ou données.

Les zones de stratégie de réponse sont nommés dans l'option response-policy pour la vue ou dans les options globales. Les zones de stratégie de réponse sont ordinairement des zones contenant des RRsets qui peuvent être requêtés normalement si autorisés. Il est généralement mieux de restreindre ces requête avec quelque-chose comme allow-query { localhost ; } ;

Une options response-policy peut supporter plusieurs zones de stratégie. Pour maximiser les performances, un arbre radix est utilisé pour rapidement identifier les zones de stratégie de réponse qui déclenchent le match de la requête actuelle. Cela impose une limite de 32 dans le nombre de zones de stratégie dans une simple response-policy. 5 déclencheurs de stratégie peuvent être encodés dans les records RPZ.

RPZ-CLIENT-IP Les records IP sont pilotés par l'adresse IP du client DNS. Ces déclencheurs sont encodés en records qui ont leur propre owner name qui sont des sous-domaines de rpz-client-ip relativisés au nom d'origine de la zone de stratégie et encode une adresse ou un block d'adresse. Les adresses IPv4 sont représentées sous la forme **prefixlength.B4.B3.B2.B1.rpz-ip**. Les préfixes IPv4 doivent être entre 1 et 32. Tous les 4 octets doivent être présents. B4 est la représentation décimale d'au moins un octet significatif de l'adresse IPv4 dans in-addr.arpa.

Les adresses IPv6 sont encodés dans un format similaire : **prefixlength.W8.W7.W6.W5.W4.W3.W2.W1.rpz-ip**. Chaque Wx est un chiffre hexadécimal représentant 16bits de l'adresse IPv6. Tous les mots doivent être présents, excepté les 0 consécutifs, remplacés par **.zz**, analogue à ' : :'.

QNAME Les enregistrements de stratégie QNAME sont déclenchés par les recherches de nom des requêtes et cible des enregistrement CNAME résolu pour générer la réponse. Le nom d'un QNAME est le nom de recherche relativisé à la zone de stratégie.

RPZ-IP Les déclencheurs sont des adresses IP dans un enregistrement A ou AAA dans la section ANSWER d'une réponse. Ils sont encodé comme les déclencheurs client-IP excpté comme sous-domaine rpz-ip

RPZ-NSDNAME Déclenche les correspondance de nom des serveurs autoritatfs pour les demande de nom, un parent du nom de la recherche, un CNAME pour le nom recherché, ou un parent du CNAME. Ils sont encodés comme sous-domaine de rpz-nsdname relativisé au nom d'origine RPZ. NSIP déclenche la correspondance des adresses IP dans les RRsets A et AAAA pour les domaines qui peuvent être vérifiés avec les records de stratégie NSDNAME.

RPZ-NSIP Déclencheurs encodés comme déclencheurs IP excepté comme sous-domaine rpz-nsip. Les déclencheurs NSDNAME et NSIP sont vérifiés seulement avec les noms avec au moins min-ns-dots points. La valeur par défaut est 1 pour exclure les TLD.

Les réponses sont vérifiés avec toutes les zones de stratégie de réponse, donc 2 ou plusieurs enregistrements de stratégie peuvent être déclenchés par une réponse. Parce que les réponse DNS sont ré-écrites en accord avec au moins un record de stratégie, un simple record encodant une action (autre que les actions DISABLED) doit être choisi. Les déclencheurs ou les records qui les encodent sont choisis pour la ré-écriture dans l'ordre suivant :

- Choisis le record déclenché dans la zone qui apparaît en premier dans l'option response-policy
- Préfère CLIENT-IP à QNAME à IP à NSDNAME à NSIP dans une simple zone
- Avec les déclencheurs NSDNAME, préférer le déclencheur qui matche le nom le plus petit dans l'ordre DNSSEC
- Avec IP ou NSIP, préfère le déclencheur avec le préfixe le plus long
- Avec les déclencheurs avec la même longueur de préfixe, préfère le déclencheur IP ou NSIP qui matche la plus petite adresse IP.

Quand le traitement d'une réponse est redémarré pour résoudre des records DNAME ou CNAME et qu'un record de stratégie n'a pas été déclenché, tous les zones de stratégie de réponse sont consultées de nouveau pour les noms DNAME ou CNAME et les adresses.

Les jeux de record RPZ sont tout type de record DNS excepté DNAME ou DNSSEC qui encode les actions ou réponses aux recherches individuelles. N'importe quelle de ces stratégies peuvent être utilisées avec n'importe quel de ces déclencheurs. Par exemple, bien que TCP-only est communément utilisé avec les déclencheurs client-IP, il peut être utilisé avec n'importe quel déclencheur pour forcer l'utilisation de TCP pour les réponses avec les owner names dans une zone.

PASSTHRU Stratégie de liste blanche spécifiée par un CNAME dont la cible est rpz-passthru. Il impose de ne pas ré-écrire la réponse.

DROP Startégie de blacklist spécifiée par un CNAME dont la cible est rpz-drop. Détruit la réponse. Rien n'est envoyée au client DNS.

TCP-Only Stratégie spécifiée par un CNAME dont la cible est rpz-tcp-only. Il change les réponses UDP en réponses DNS tronquées qui nécessite que le client DNS retente avec TCP. Utilisé pour limiter les attaque DNS reflection.

NXDOMAIN Le domaine indéfini est encodé par un CNAME dont la cible est '.'

NODATA Le jeu vide de RR est spécifié par un CNAME dont la cible est '*.'. Il ré-écrit la réponse à NODATA ou ANCOUNT=1

Local Data Un jeu de records DNS ordinaire peut être utilisé pour répondre aux requêtes. Les recherches pour les types de records qui ne sont pas dans le jeu sont répondus avec NODATA. Une forme spécifiale est un wildcard "*.example.com".

Toutes les action spécifiées dans les records individuels dans un zone de stratégie peuvent être remplacés avec une clause policy dans l'option response-policy. Une organisation utilisant une zone de stratégie fournie par une autre organisation peut utiliser ce mécanisme pour rediriger les domaines vers sont propre walled garden.

GIVEN Indique 'ne pas remplacer mais effectuer l'action spécifiée dans la zone.

DISABLED La stratégie ne fait rien mais log ce qui aurait été fait.

PASSTHRU, DROP, TCP-Only, NXDOMAIN, NODATA Remplace avec la stratégie par record

CNAME domain Force tous les records RPZ à agir comme s'ils étaient des records 'cname domain'

Par défaut, les actions encodés dans une zone de stratégie de réponse sont appliqués seulement aux requêtes qui demande une récursion. Ce défaut peut être changé pour une simple zone de stratégie ou pour toutes les zones de stratégie dans une vue avec une clause recursive-only no ;.Cette fonctionnalité est utile pour servir les même fichiers de zone en et hors d'un cloud rfc1918 et utilisant RPZ pour supprimer les réponses qui contiendrait des valeurs rfc1918 dans la vue.

Également, par défaut les action RPZ sont appliquées seulement aux demandes DNS qui ne demandent pas de métadonnées DNSSEC ou quand aucun record DNSSEC n'est disponible pour le nom demandé. Cela peut être changé avec break-dnssec yes.

Aucun enregistrement DNS n'est nécessaire pour un déclencheur QNAME ou Client-IP. Le nom ou l'adresse IP elle-même est suffisante, donc en principe le nom recherché ne doit pas être recherché récursivement. Cependant, ne pas résoudre le nom demandé peut fuiter le fait que la ré-écriture est utilisé et que ce nom est listé dans une zone de stratégie. Pour empêcher cette fuite d'information, par défaut toute récursion nécessaire pour une recherche est faite avant tout déclencheur. Vu que les domaines listés ont souvent des serveurs autoritatifs lents, ce mode peut être couteux en temps. l'option qname-wait-recurse no ; change ce mode. L'option n'affecte pas les déclencheurs QNAME ou client-ip dans les zones de stratégie listées après d'autres zones contenant des déclencheurs IP, NSIP et NSDNAME, parce que la réponse serait dépendant des records RRSIG trouvés durant la résolution. Utiliser cette option peut causer des erreurs de réponse tels que SERVFAIL.

Le TTL d'un record modifié par les stratégie RPZ sont définis depuis le TTL de l'enregistrement dans le zone de stratégie. Il est limité à la valeur maximum. max-policy-ttl change ce maximum.

Par exemple, on peu utiliser cette déclaration :

response-policy { zone "badlist"; };
et cette déclaration de zone :
zone "badlist" {type master; file "master/badlist"; allow-query {none;}; };
Avec ce fichier de zone :

```
$TTL 1H
@ SOA LOCALHOST. named-mgr.example.com (1 1h 15m 30d 2h)
NS LOCALHOST.
; QNAME policy records. There are no periods (.) after the owner names.
nxdomain.domain.com CNAME . ; NXDOMAIN policy
.nxdomain.domain.com CNAME . ; NXDOMAIN policy
nodata.domain.com CNAME *. ; NODATA policy
.nodata.domain.com CNAME *. ; NODATA policy
bad.domain.com A 10.0.0.1 ; redirect to a walled garden
AAAA 2001:2::1
bzone.domain.com CNAME garden.example.com.
; do not rewrite (PASSTHRU) OK.DOMAIN.COM
ok.domain.com CNAME rpz-passthru.
; redirect x.bzone.domain.com to x.bzone.domain.com.garden.example.com
.bzone.domain.com CNAME *.garden.example.com.
; IP policy records that rewrite all responses containing A records in 127/8
; except 127.0.0.1
8.0.0.0.127.rpz-ip CNAME .
32.1.0.0.127.rpz-ip CNAME rpz-passthru.
; NSDNAME and NSIP policy records
ns.domain.com.rpz-nsdname CNAME .
48.zz.2.2001.rpz-nsip CNAME .
; blacklist and whitelist some DNS clients
112.zz.2001.rpz-client-ip CNAME rpz-drop.
8.0.0.0.127.rpz-client-ip CNAME rpz-drop.
; force some DNS clients and responses in the example.com zone to TCP
16.0.0.1.10.rpz-client-ip CNAME rpz-tcp-only.
example.com CNAME rpz-tcp-only.
.example.com CNAME rpz-tcp-only.
```

Limiter le taux de réponses

Les réponses UDP excessivement identique peuvent être contrôlés en configurant un rate-limit dans une déclaration options ou view. Ce mécanisme réduit l'utilisation des serveurs autoritatifs pour amplifier les attaques DOS. Les réponses Short truncated (TC=1) peuvent être envoyées pour fournir des réponse limitées aux clients légitimes dans une plage d'adresses IP forgés. Les clients légitimes réagissent aux réponses tronquée ou supprimées en retentant avec UDP ou TCP, respectivement.

Ce mécanisme est prévu pour les serveurs autoritatifs. Il peut être utilisé sur les serveurs récursifs mais peut ralentir les applications telles que les serveurs SMTP et les client HTTP qui recherche souvent le même domaine.

rate-limite utilise un schéma de crédit ou un jeton. Chaque combinaison de réponse et client identique a un compte conceptuel qui apprend un nombre spécifié de crédits chaque seconde. Les réponses sont supprimées ou tronquées quand le compte est négatif. Les réponses sont suivies dans une fenêtre de temps qui est de 15 secondes par défaut, mais peut être changé avec l'option window (1 à 3600). Le compte ne peut plus être positif que sur la limite par seconde ou plus que la limite window. Quand le nombre spécifié de crédits pour une classe de réponse est 0, ces réponses ne sont plus limitées.

Les notions de réponse identique et de client DNS pour le rate-limit ne sont pas simple. Tous les réponses à un block d'adresse sont comptés comme si c'était un seul client. Les longueurs de préfixe des blocks d'adresse sont spécifiées dans ipv4-prefix-lengh (défaut 24) et ipv6-prefix-lengh (défaut : 56).

Tout réponse non-vide pour un nom de domaine valide (qname) et type d'enregistrement (qtype) sont identique et on une limite spécifiée par l'option responses-per-second (c à d, réponses par seconde avec seulement un simple argument et aucun modifieurs additionnels.) Le défaut est 0, qui indique qu'il ne devrait pas y avoir de limite. Les réponse vide (NODATA) sont limitées par nodata-per-second. Les demandes pour un ou tous les sous-domains indéfinis d'un domaine valide résultent en erreurs NXDOMAIN, et sont identiques sans regarder le type de recherche. Elles sont limitées par nxdomains-per-second. Cela contrôle certaines attaques en utilisant des noms aléatoire, mais peuvent être relaxés et désactivé dans les serveurs qui attendent de nombreuses réponses NXDOMAIN, tels que depuis les blacklists anti-spam. Les référants et délégations au serveur d'un domaine donné sont identique et sont limités par referrals-per-second.

Les réponses générées depuis les wildcards locaux sont comptés et limités comme s'ils étaient pour le nom de domain parent. Cela contrôle le flooding en utilisant les noms aléatoires.

Toutes les requêtes qui résultent en erreurs DNS autre que NXDOMAIN, tel que SERVFAIL et FORMERR, sont identique sans regarder le nom recherché (qname) ou le type de records (qtype). Cela contrôle les attaques utilisant les requêtes invalides ou distantes, sur les serveurs autoritatif cassés. Par défaut la limite sur les erreurs est la même que responses-per-second, mais peut être changé avec errors-per-seconds.

De plus, jusqu'à 4 options responses-per-seconde (en plus de la valeur de base) peuvent être configurés, avec des paramètres additionnels pour indiquer qu'elles s'appliquent aux réponses plus grande qu'une taille donnée, ou avec un facteur d'amplification plus grand qu'une valeur donnée. Le paramètre size définis la taille de réponse DNS minimum qui va déclencher l'utilisation de cette option respones-per-second. Le paramètre ratio définis la taille de réponse / taille de requêtes DNS minimum qui sont dans la plage, à 2 décimales. Ces limites sélective sont appliquées après que d'autres limite aient été appliquées et ne s'appliquent qu'aux réponses positives.

Par exemple :

```
rate-limit {
 responses-per-second 10;
 responses-per-second size 1100 5;
};
```

Indique que les réponse devraient être limitées à 10 par seconde pour les réponses jusqu'à 1099 octets, mais seulement 5 par seconde pour les réponses plus grandes. Cette configuration :

```
rate-limit {
 responses-per-second 10;
 responses-per-second ratio 7.25 5;
 responses-per-second ratio 15.00 2;
};
```

Indiquent que les réponses devraient être limitées à 10 par seconde si le facteur d'amplification est inférieur à 7,25, 5 par seconde s'il est supérieur à 7,25, mais inférieur à 15, 2 par seconde au-delà de 15. Les tailles et ratios peuvent être utilisées ensemble, par exemple :

```
rate-limit {
 responses-per-second 10;
 responses-per-second size 1000 ratio 5.00 5;
 responses-per-second ratio 10.00 2;
};
```

Cette configuration va limiter à 5 par seconde si le ration est au-dessus de 5 ou la taille supérieur à 1000.

De nombreuses attaques utilisant DNS impliquent les requêtes UDP avec des adresses sources forgées. Limiter le taux empêche l'utilisation de BIND9 pour flooder un réseaux avec des réponses à des requêtes avec les adresses source forgées, mais pourrait laisser un tier bloquer les réponses aux requêtes légitimes. Il y a un mécanisme qui peut répondre à certaines requêtes légitimes d'un client dont l'adresse a été forgée dans un flood. Définir slip à 2 (son défaut) cause toute autre requête UDP à être répondue avec une réponse tronquée. La petite taille et la fréquence réduite, et le manque d'amplification des réponses les rendent moins attractive pour les attaques DOS. slip doit être entre 0 et 10. Une valeur de 0 ne tronque pas les réponses, elles sont supprimée. Une valeur de 1 cause chaque réponse à être tronquées. Certaines réponses d'erreurs telles que REFUSED ou SERVFAIL ne peuvent être remplacée avec des réponses tronquées et sont fuitées au taux slip.

Quand le taux de recherche excède qps-scale, les valeurs responses-per-second, errors-per-second, nxdomains-per-second et all-per-second sont réduite par le ratio du taux courant à la valeur qps-scale. Cette fonctionnalité peut resserer les défense durant les ataques. Par exemple, avec qps-scale 250 ; responses-per-second 20 ; et un taux de recherche total de 1000 recherches par secondes pour toutes les recherches de tous les clients DNS incluant via TCP, alors la limite de réponses/seconde effective change à (250/1000)*20 ou 5. Les réponses envoyées via TCP ne sont pas limitées mais sont comptées pour calculer le taux de recherche par seconde.

La clause optionnelle domain spécifie l'espace de nom auquel les limites s'appliquent. Il est possible d'utiliser différentes limites pour différents noms en spécifiant plusieurs blocks rate-limit.

Les limiteurs de taux pour différents espaces de nom maintiennent des compteurs séparés : si, par exemple, il y a une déclaration rate-limit pour com et un autre pour example.com, les recherche correspondant à example.com ne seront pas débitée avec le limiteur de taux pour com. Si une déclaration rate-limit ne spécifie pas un domain, elle s'applique au domaine root et donc tout l'espace de nom DNS.

Les communautés de clients DNS peuvent avoir leur propre limites en plaçant les déclaration rate-limit dans le vues. un rate-limit dans une vue remplace et ne complète pas les déclarations globales. Les clients DNS dans une vue peuvent être exemptés de limites avec la clause exempt-clients.

Les réponses UDP de tous type peuvent être limitées avec la phrase all-per-second. Cette limite devrait être au moins 4 fois plus grand que les autres limites. La taille maximum de la table utilisée pour tracker les recherches et limiter le taux de réponses est définis avec max-table-size. Chaque entrée dans la table est entre 40 et 80 octets. La table a besoin approximativement d'autant d'entrée que le nombre de requêtes reçues par secondes. Défaut : 20000. Pour réduire le coût du démarrage à froid, min-table-size (défaut : 500) peut définir la taille minimum. Utiliser log-only yes ; pour tester les paramètre de limitation sans supprimer une seul recherche.

server

```
server ip_addr[/prefixlen] {
 [ bogus yes_or_no ; ]
 [ provide-ixfr yes_or_no ; ]
 [ request-ixfr yes_or_no ; ]
 [ request-nsid yes_or_no ; ]
 [ request-sit yes_or_no ; ]
 [ edns yes_or_no ; ]
 [ edns-udp-size number ; ]
 [ nosit-udp-size number ; ]
 [ max-udp-size number ; ]
 [ transfers number ; ]
 [ transfer-format ( one-answer | many-answers ) ; ]]
 [ keys { string ; [ string ; [...]] } ; ]
 [ transfer-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ transfer-source-v6 (ip6_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ notify-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ notify-source-v6 (ip6_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ query-source [ address ( ip_addr | * ) ]
  [ port ( ip_port | * ) ] [dscp ip_dscp] ; ]
 [ query-source-v6 [ address ( ip_addr | * ) ]
  [ port ( ip_port | * ) ] [dscp ip_dscp] ; ]
 [ use-queryport-pool yes_or_no; ] //obsolete
 [ queryport-pool-ports number; ] //obsolete
 [ queryport-pool-updateinterval number; ] //obsolete
};
```

La déclaration server définis les caractéristiques à associer avec un serveur de noms distant. Si un préfixlen est spécifié, cela permet de couvrir plusieurs serveurs. La déclaration server peut se produire en haut de la configuration ou dans une déclaration view. Si une vue contient des déclarations server, les déclarations server dans la configuration globale sont ignorés pour la vue.

bogus yes_or_no Si un serveur distant donne de mauvaises données, empêche d'autres requêtes sur lui. défaut : no

provide-ixfr yes_or_no détermine si le serveur local, agissant comme maître, répond avec un transfert de zone incrémental. à yes, le transfert incrémental est fournis quand c'est possible.

request-ixfr yes_or_no Détermine si le serveur local, agissant comme slave, demande des transferts de zone incrémental.

request-nsid yes_or_no à yes, une option EDNS(0) NSID (Name Server Identifier) vide est envoyée avec toutes les requêtes aux serveurs de noms autoritatifs durant la résolution itérative. Si le serveur autoritatif retourne une options NSID dans sa réponse, son contenu est loggé dans la catégorie resolver au niveau info.

request-sit yes_or_no à yes, une option EDNS SIT (Source Identity Token) est envoyée avec le requête. Si le résolveur à précédemment échoué à parler au serveur, le SIT retourné dans la précédente transaction est envoyée. C'est utilisé par le serveur pour déterminer si le résolveur lui a parlé avant.

edns yes_or_no Détermine si le serveur local tente d'utiliser EDNS en communiquant avec le serveur distant. défaut : yes

edns-udp-size Définis la taille de tampon UDP EDNS initial, pour contrôler la taille des paquets reçus depuis les serveurs autoritatifs en réponse aux requêtes récursives. Les valeurs valides sont 512 à 4096 (défaut). changer cette valeur est d'avoir des réponses udp à passer via des firewalls qui bloquent les paquets fragmentés et/ou bloquent les paquets DNS UDP supérieurs à 512 octets.

max-udp-size Définis la taille maximale de message UDP EDNS à envoyer en octets. Les valeurs valides sont 512 à 4096 (défaut). baisser cette valeur encourage l'utilisation de TCP.

nosit-udp-size number Définis la taille maximum de réponses UDP qui seront envoyées pour les requêtes sans un token d'identité valide.

transfers number Limite le nombre de transfert de zones entrantes simultané du serveur spécifié. Non spécifié, la limite est définis en accord avec transfers-per-ns

transfer-format Les transferts de zone peuvent être envoyés en 2 format : one-answer et many-answers. Ce dernier est plus efficace mais seulement supporté par les serveur DNS récents.

keys { string ; [string ; [...]] } Identifie une clé définie par la déclaration key, à utiliser pour une transaction TSIG en dialoguant avec le serveur distant.

transfer-source Détermine quelles adresses locales seront utilisée pour les connections IPv4 TCP utilisées pour le transfert de zone entrant sur le serveur. Détermine également l'adresse IPv4 et optionnellement le port UDP, utilisé pour les requêtes de rafraîchissement et le forward de mises à jours dynamique.

transfer-source-v6 Idem, sur ipv6

query-source address * port * ; Si le serveur ne connaît pas la réponse à une question, il va requêter d'autres serveurs de nom. Cette option spécifie l'adresse et le port utilisé pour de telles requêtes.

query-source-v6 address * port * ; Idem pour ipv6

[SECTION] name="statistics-channels" table="codes" imbrication="0"

statistics-channels {

[inet (ip_addr | *) [port ip_port] [allow { address_match_list }] ;]

[inet ... ;] } ;

Les déclaration statistics-channels déclarent des canaux de communication à utiliser par les administrateurs système pour obtenir un accès à des informations de statistiques du serveur de nom. Cette déclaration est prévue pour être flexible pour supporter plusieurs protocoles de communication dans le future, mais actuellement seul l'accès HTTP est supporté. Il nécessite que bind9 soit compilé avec libxml2 et/ou json-c. Ces déclarations sont acceptées même s'il est construit sans la librairie, mais les accès HTTP vont échouer dans erreur.

Un canal de contrôle inet est un socket TCP en écoute au port spécifié sur l'ip spécifiée qui peut être une adresse IPv4 ou IPv6. La tentative d'ouvrir un canal de statistique est restreinds par la clause allow. Les statistiques sont disponible dans divers formats et vues en fonction de l'URI utilisée pour y accéder. Le format xml est accéssible via http ://127.0.0.1 :8888/ ou <http ://127.0.0.1 :8888/xml. Un fichier CSS est inclus qui peut formater les statistiques XML en tables quand il est lus dans un navigateur, et en graphique avec Google Charts API quand il est utilisé avec un navigateur javascript.

http ://127.0.0.1 :8888/xml/v2> schéma xml v2

http ://127.0.0.1 :8888/xml/v3> schéma xml v3

http ://127.0.0.1 :8888/xml/v3/status Sous-jeu de statistiques (uptime et dernières reconfigurations)

http ://127.0.0.1 :8888/xml/v3/server Statistiques serveur et résolveur

http ://127.0.0.1 :8888/xml/v3/zones Statistiques de zone

http ://127.0.0.1 :8888/xml/v3/net status réseau et socket

http ://127.0.0.1 :8888/xml/v3/mem statistiques mémoire

http ://127.0.0.1 :8888/xml/v3/tasks statistiques des tâches

http ://127.0.0.1 :8888/json Jeu complet de statistiques au format JSON

http ://127.0.0.1 :8888/json/v1/status Sous-jeu de statistiques (uptime et dernières reconfigurations) JSON

http ://127.0.0.1 :8888/json/v1/server Statistiques serveur et résolveur au format JSON

<http ://127.0.0.1 :8888/json/v1/zones Statistiques de zone au format JSON

<http ://127.0.0.1 :8888/json/v1/net status réseau et socket au format JSON

<http ://127.0.0.1 :8888/json/v1/mem statistiques mémoire au format JSON

<http ://127.0.0.1 :8888/json/v1/tasks statistiques des tâches au format JSON

trusted-keys

```
trusted-keys {
 string number number number string ;
 [ string number number number string ; [...]]
};
```

La déclaration trusted-keys définis les racines de sécurité DNSSEC. Une racide de sécurité est définie quand la clé publique pour une zone non-authoritative est connue, mais ne peut pas être obtenue de manière sécurisée via DNS, soit parce que c'est une zone root DNS ou parce que sa zone parent n'est pas signée. Une fois qu'une clé a été configurée comme clé de confiance, elle est traitée comme si elle avait été validée. Le résolveur tente la validation DNSSEC sur toutes les données dans les sous-domaines d'un security root.

Toutes les clés (et zones correspondantes) listées dans trusted-keys sont réputés exister sans regarder de quelle zone parent il s'agit. Similairement pour toutes les clés listées dans trusted-keys, elles sont utilisées pour valider le RRsed DNSKEY. Le RRset DS du parent ne sera pas utilisé.

Cette déclaration peut contenir plusieurs clés, chacune consistant du nom de domaine de la clé, les flags, protocoles, algorithmes, et la représentation base64 de la clé. La déclaration trusted-keys peut être définis globalement, ou dans une vue. Si les 2 sont en place, elles sont additives.

managed-keys

```
managed-keys {
 name initial-key flags protocol algorithm key-data ;
 [ name initial-key flags protocol algorithm key-data ; [...]]
};
```

la déclaration managed-keys, tout comme trusted-keys, définis les DNSSEC security roots. La différence est que managed-keys peut être conservé à jour automatiquement, sans l'intervention de l'opérateur. Cette déclaration possède un mot clé supplémentaire, initial-key, contenant la clé d'initialisation.

view

```
view view_name
 [class] {
 match-clients { address_match_list };
```

```
 match-destinations { address_match_list };
 match-recursive-only yes_or_no ;
 [ view_option; ...]
 [ zone_statement; ...]
};
```

La déclaration view implémente le split-dns. Elle possède ses propres zones. Les zones disponible dans une vue ne sont accessible que pour cette vue. De nombreuses options de la déclaration options peuvent également être utilisées dans une déclaration vue.

zone

zone master :

```
zone zone_name [class] {
 type master;
 [ allow-query { address_match_list }; ]
 [ allow-query-on { address_match_list }; ]
 [ allow-transfer { address_match_list }; ]
 [ allow-update { address_match_list }; ]
 [ update-check-ksk yes_or_no; ]
 [ dnssec-dnskey-kskonly yes_or_no; ]
 [ dnssec-loadkeys-interval number; ]
 [ update-policy local | { update_policy_rule [...] }; ]
 [ also-notify { ip_addr [port ip_port] [dscp ip_dscp] ;
  [ ip_addr [port ip_port] [dscp ip_dscp] ; ... ] }; ]
 [ check-names (warn|fail|ignore) ; ]
 [ check-mx (warn|fail|ignore) ; ]
 [ check-wildcard yes_or_no; ]
 [ check-spf ( warn | fail | ignore ); ]
 [ check-integrity yes_or_no ; ]
 [ dialup dialup_option ; ]
 [ file string ; ]
 [ masterfile-format (text|raw|map) ; ]
 [ journal string ; ]
 [ max-journal-size size_spec; ]
 [ forward (only|first) ; ]
 [ forwarders { [ ip_addr [port ip_port] [dscp ip_dscp] ; ... ] }; ]
 [ ixfr-base string ; ]
 [ ixfr-from-differences yes_or_no; ]
 [ ixfr-tmp-file string ; ]
 [ request-ixfr yes_or_no ; ]
 [ maintain-ixfr-base yes_or_no ; ]
 [ max-ixfr-log-size number ; ]
 [ max-transfer-idle-out number ; ]
 [ max-transfer-time-out number ; ]
 [ notify yes_or_no | explicit | master-only ; ]
 [ notify-delay seconds ; ]
 [ notify-to-soa yes_or_no; ]
 [ pubkey number number number string ; ]
 [ notify-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ notify-source-v6 (ip6_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ zone-statistics full | terse | none; ]
 [ sig-validity-interval number [number] ; ]
 [ sig-signing-nodes number ; ]
 [ sig-signing-signatures number ; ]
 [ sig-signing-type number ; ]
```

```
 [ database string ; ]
 [ min-refresh-time number ; ]
 [ max-refresh-time number ; ]
 [ min-retry-time number ; ]
 [ max-retry-time number ; ]
 [ key-directory path_name; ]
 [ auto-dnssec allow|maintain|off; ]
 [ inline-signing yes_or_no; ]
 [ zero-no-soa-ttl yes_or_no ; ]
 [ serial-update-method increment|unixtime; ]
 [ max-zone-ttl number ; ]
};
```

zone slave :

```
zone zone_name [class] {
 type slave;
 [ allow-notify { address_match_list }; ]
 [ allow-query { address_match_list }; ]
 [ allow-query-on { address_match_list }; ]
 [ allow-transfer { address_match_list }; ]
 [ allow-update-forwarding { address_match_list }; ]
 [ dnssec-update-mode ( maintain | no-resign ); ]
 [ update-check-ksk yes_or_no; ]
 [ dnssec-dnskey-kskonly yes_or_no; ]
 [ dnssec-loadkeys-interval number; ]
 [ dnssec-secure-to-insecure yes_or_no ; ]
 [ try-tcp-refresh yes_or_no; ]
 [ also-notify [port ip_port] [dscp ip_dscp] { ( masters_list | ip_addr
  [port ip_port]
  [dscp ip_dscp]
  [key key] ) ; [...] }; ]
 [ check-names (warn|fail|ignore) ; ]
 [ dialup dialup_option ; ]
 [ file string ; ]
 [ masterfile-format (text|raw|map) ; ]
 [ journal string ; ]
 [ max-journal-size size_spec; ]
 [ forward (only|first) ; ]
 [ forwarders { [ ip_addr [port ip_port] [dscp ip_dscp] ; ... ] }; ]
 [ ixfr-base string ; ]
 [ ixfr-from-differences yes_or_no; ]
 [ ixfr-tmp-file string ; ]
 [ maintain-ixfr-base yes_or_no ; ]
 [ masters [port ip_port] [dscp ip_dscp] { ( masters_list | ip_addr
  [port ip_port]
  [dscp ip_dscp]
  [key key] ) ; [...] }; ]
 [ max-ixfr-log-size number ; ]
 [ max-transfer-idle-in number ; ]
 [ max-transfer-idle-out number ; ]
 [ max-transfer-time-in number ; ]
 [ max-transfer-time-out number ; ]
 [ notify yes_or_no | explicit | master-only ; ]
 [ notify-delay seconds ; ]
 [ notify-to-soa yes_or_no; ]
 [ pubkey number number number string ; ]
 [ transfer-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ transfer-source-v6 (ip6_addr | *) [port ip_port] [dscp ip_dscp] ; ]
```

```
 [ alt-transfer-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ alt-transfer-source-v6 (ip6_addr | *)
  [port ip_port]
  [dscp ip_dscp] ; ]
 [ use-alt-transfer-source yes_or_no; ]
 [ notify-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ notify-source-v6 (ip6_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ zone-statistics full | terse | none; ]
 [ sig-validity-interval number [number] ; ]
 [ sig-signing-nodes number ; ]
 [ sig-signing-signatures number ; ]
 [ sig-signing-type number ; ]
 [ database string ; ]
 [ min-refresh-time number ; ]
 [ max-refresh-time number ; ]
 [ min-retry-time number ; ]
 [ max-retry-time number ; ]
 [ key-directory path_name; ]
 [ auto-dnssec allow|maintain|off; ]
 [ inline-signing yes_or_no; ]
 [ multi-master yes_or_no ; ]
 [ zero-no-soa-ttl yes_or_no ; ]
};
```

zone hint

```
zone zone_name [class] {
 type hint;
 file string ;
 [ delegation-only yes_or_no ; ]
 [ check-names (warn|fail|ignore) ; ] // Not Implemented.
};
```

zone stub

```
zone zone_name [class] {
 type stub;
 [ allow-query { address_match_list }; ]
 [ allow-query-on { address_match_list }; ]
 [ check-names (warn|fail|ignore) ; ]
 [ dialup dialup_option ; ]
 [ delegation-only yes_or_no ; ]
 [ file string ; ]
 [ masterfile-format (text|raw|map) ; ]
 [ forward (only|first) ; ]
 [ forwarders { [ ip_addr [port ip_port] [dscp ip_dscp] ; ... ] }; ]
 [ masters [port ip_port] [dscp ip_dscp] { ( masters_list | ip_addr
  [port ip_port]
  [dscp ip_dscp]
  [key key] ) ; [...] }; ]
 [ max-transfer-idle-in number ; ]
 [ max-transfer-time-in number ; ]
 [ pubkey number number number string ; ]
 [ transfer-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ transfer-source-v6 (ip6_addr | *)
  [port ip_port] [dscp ip_dscp] ; ]
 [ alt-transfer-source (ip4_addr | *) [port ip_port] [dscp ip_dscp] ; ]
 [ alt-transfer-source-v6 (ip6_addr | *)
  [port ip_port] [dscp ip_dscp] ; ]
```

```
 [ use-alt-transfer-source yes_or_no; ]
 [ zone-statistics yes_or_no ; ]
 [ database string ; ]
 [ min-refresh-time number ; ]
 [ max-refresh-time number ; ]
 [ min-retry-time number ; ]
 [ max-retry-time number ; ]
 [ multi-master yes_or_no ; ]
};
```

zone static-stub :

```
zone zone_name [class] {
 type static-stub;
 [ allow-query { address_match_list }; ]
 [ server-addresses { [ ip_addr ; ... ] }; ]
 [ server-names { [ namelist ] }; ]
 [ zone-statistics yes_or_no ; ]
};
```

zone forward :

```
zone zone_name [class] {
 type forward;
 [ forward (only|first) ; ]
 [ forwarders { [ ip_addr [port ip_port] [dscp ip_dscp] ; ... ] }; ]
 [ delegation-only yes_or_no ; ]
};
```

zone redirect :

```
zone "." [class] {
 type redirect;
 file string ;
 [ masterfile-format (text|raw|map) ; ]
 [ allow-query { address_match_list }; ]
 [ max-zone-ttl number ; ]
};
```

zone delegation-only :

```
zone zone_name [class] {
 type delegation-only;
};
```

zone référencée :

```
zone zone_name [class] {
 [ in-view string ; ]
};
```

types de zone

master Le serveur a une copie maître des données de la zone et est capable de fournir des réponse autoritatives pour lui.

slave une zone esclave est un réplica de la zone maître.

stub Une zone stub est similaire à une zone esclave, excepté qu'elle ne réplique que les records NS de la zone maître et non la zone entière. Les zones stub ne sont pas standard. Non recommandé.

static-stub Similaire à une zone stub, mais les données de zone sont configurés statiquement.

forward Une zone forward est une manière de configurer un forwarding par domaine. Une zone forward peut contenir des déclations forward et/ou forwarders.

hint Le jeu de serveurs de nom racine est spécifié en utilisant ce type de zone.

redirect Les zones redirect sont utilisées pour fournir des réponses aux requêtes quand la résolution normale retourne NXDOMAIN. Seul une zone redirect est supportée par vue.

delegation-only C'est utilisé pour forcer le status de delegation-only des zones d'infrastructure (COM, NET, ORG). Toute réponse reçue sans délégation implicite ou explicite dans la section authority seront traité comme NXDOMAIN. Ne s'applique pas dans l'apex de zone.

Classes

Le nom de la zone peut optionnellement être suivi par une classe. Si une classe n'est pas spécifiée, la classe IN est assumée. La classe hesiod est nommée pour un service d'information du projet Athena du MIT. Elle est utilisée pour partager des informations sur diverses bases de données, tels que les utilisateurs, groupes, imprimantes, etc. HS est synonyme de hesiod. Un autre développement du MIT est chaosnet, un protocole LAN créé dans les années 1970.

options de zone

update-policy local | { update_policy_rule [...] }; voir plus bas

also-notify n'a de sens que si notify est actif pour cette zone. Le jeu de machines qui vont recevoir un messages DNS NOTIFY pour cette zone.

check-names (fail | warn | ignore) Cette option est utilisée pour restreindre le jeu de caractères et la syntaxe de certains noms de domaine dans les fichiers master et/ou les réponses DNS reçues du réseaux

database Spécifie le type de base à utiliser pour stocker les données de zone. La chaîne est interprétée comme une listed de mots séparés par un espace blank. Le premier mot identifie le type de base, et les autres mots sont passés en argument à la base à interpréter. (défaut : rbt), la base red-black-tree in-memory natif de bind9. Cette base n'a pas d'argument.

delegation-only Ce flag ne s'applique qu'aux zones forward, hint, et stub. À yes, toutes les zones sont également traitées comme si elle étaient également de type delegation-only.

forward (only | first) N'a de sens que si la zone a une liste forwarders. only échoue si le forwarding échoue, first, tente dans ce cas une résolution normale.

forwarders non spécifié dans une zone forward, aucun forwarding n'est fait.

journal permet d'écraser le nom de fichier du journal par défaut.

zone-statistics yes_or_no à yes, le serveur conserve des informations de statistique pour cette zone, qui peuvent être dumpé dans le statistics-file définis dans les options du serveur.

server-address dans les zone static-stub, liste d'adresses IP pour lesquelles les requêtes devraient être envoyés en résolution récursive pour la zone.

server-names dans les zone static-stub, liste les noms de domaine des serveurs de nom qui agissent comme serveurs autoritatifs.

ixfr-from-differences Noter que les choix master et slave ne sont pas disponible au niveaux des zones.

auto-dnssec (allow | maintain | off) Les zones configurées pour le DNS dynamique peuvent également utiliser cette option pour permettre différents niveaux de gestion de clé DNSSEC automatique. 'allow' permet aux clés d'être mises à jours et de resigner la zone quand l'utilisateur émet la commande rndc sign zonename. 'maintain' est identique, mais ajuste également automatiquement les clés DNSSEC de la zone en accord avec les métadonnées de timing de clé. 'off' désactive la fonctionnalité

serial-update-method (increment | unixtime) Les zones configurées pour le DNS dynamique peuvent utiliser cette option pour définir la méthode de mise à jours qui sera utilisée pour le numéro de série de la zone dans le SOA.

inline-signing à yes, active la signature "bump in the wire" d'une zone, où une zone non-signée y est transférée et chargée depuis le disque et une version signée de la zone est servie, avec possiblement, un numéro de série différent. désactivé par défaut.

Stratégie de mise à jours dynamique

BIND9 supporte 2 méthodes alternatives d'acceptation de clients pour effectuer des mises à jours dynamique dans une zone, configuré par les options allow-update et update-policy, respectivement.

allow-update fonctionne de la même manière que dans les versions précédentes de bind. Il donne aux clients la permission de mettre à jours les enregistrements de n'importe quel nom dans la zone.

update-policy permet un contrôle plus fin sur les mises à jours permises. Un jeu de règles est spécifié, où chaque règle autorise ou refuse l'accès pour un ou plusieurs noms à mettre à jours. Si la mise à jours dynamique nécessite que le message soit signé, l'identité du signataire peut être déterminé.

Les règles sont spécifiées dans l'option de zone update-policy, et sont seulement significatifs pour les zones maître. Quand la déclaration update-policy est présente, allow-update ne doit pas être présent. La déclaration update-policy examine seulement le signataire du message ; l'adresse sources n'est pas utilisée.

Il y a une règle update-policy prédéfinie qui peut être activée avec la commande update-policy local. Activer cette règle génère une clé de session TSIG et la place dans un fichier (par défaut : /var/run/named/session.key, le nom de la clé local-ddns, et l'algorithm HMAC-SHA256, peut être changé avec les options session-<xxx>), et autorise cette clé à mettre à jours la zone.

Un client fonctionnant sur le système local, avec les permissions appropriées, peut lire ce fichier et l'utiliser pour signer les demandes de mise à jours. Cette règle est équivalente à update-policy { grant local-ddns zonesub any ; } ;

La commande nsupdate -l envoie des demandes de mise à jours à l'hôte local, et les signes en utilisant a clé de session.

D'autres définitions ressemblent à :

```
( grant | deny ) identity nametype [ name ] [ types ]
```

Chaque règle autorise ou refuse les privilèges. Une fois qu'un message matche une règle, l'opération est immédiatement autorisé ou refusé et aucune autre règle n'est examinée. Une règle est matchée quand le signataire matche le champs identity, le nom matche le champ name avec le champs nametype, et le type matche les types spécifiés dans le champs type.

Aucun signataire n'est requis pour tcp-self et 6to4-self cependant la conversion du mappage inverse / préfixe doit matcher le champ identity.

Le champ identity spécifie un nom ou un nom wildcard. Normalement, c'est le nom d'une clé TSIG utilisée pour signer le demande de mise à jour. Quand un échange TKEY a été utilisé pour créer un secret partagé, l'identité de la clé partagée est la même que l'identité de la clé utilisée pour authentifier l'échange TKEY. TKEY est également la méthode de négociation utilisée par GSS-TSIG, qui établis une identité que est le principal Kerberos du client, tel que 'user@host.domain'. Quand le champ identity spécifie un nom wildcard, il est sujet à l'expansion DNS, donc la règle s'applique à plusieurs identités. Le champs identity doit contenir un nom fqdn.

Pour les types de nom krb5-self, ms-self, krb5-subdomain, et ms-subdomain, le champ identity spécifie le royaume Windows ou Kerberos auquel la machine appartient.

Le champs nametype a 13 valeurs :

name Exact match. Cette réglè est identique au contenu du champ name.

subdomain Cette règle matche quand le nom à mettre à jours est un sous-domaine au, ou identique au, contenu du champ name.

zonesub similaire, excepté qu'elle matche quand le nom à mettre à jours est un sou-domaine de la zone dans laquelle update-policy apparaît.

wildcard Le champs nom est sujet à expansion DNS, et cette règle matche quand le nom à mettre à jours est une expansion valide

self Cette règle matche quand le nom à mettre à jours matche le contenu de identity. Le champ name est ignoré mais devrait être identique à identity.

selfsub Similaire à self excepté que les sous-domaines de self sont également mis à jours.

selfwild Similaire à self excepté que seul les sous-domaines de self sont mis à jours.

ms-self Cette règle prend un principal de machine Windows (machine@REALM) et le convertit en machine.realm.

ms-subdomain Cette règle prend un principal de machine Windows (machine@REALM) et le convertit en machine.realm pour mettre à jours les sous-domaines de machine.realm.

krb5-self Cette règle prend un principal de machine Kerberos (host/machine@REALM) et le convertit en machine.realm.

krb5-subdomain Cette règle prend un principal de machine Kerberos (host/machine@REALM) et le convertit en machine.realm pour mettre à jours les sous-domaines de machine.realm.

tcp-self Autorise les mises à jours qui ont été envoyés via TCP et pour lequelles le mappage standard pour initier l'adresse IP dans in-adddr.arpa et ip6.arpa match le nom à mettre à jours. Noter qu'il est théoriquement possible de spoofer ces sessions TCP.

6to4-self Autorise le préfixe 6to4 à être mis à jours par une connexion TCP depuis le réseaux 6to4 ou depuis l'adresse IPv4 correspondante. C'est prévu pour pouvoir ajouter des RRsets NS ou DNAME.

external Cette règle autorise named à déférer la décision de l'accès à un service externe. La méthode de communication avec le service est spécifié dans le champs identity, le format est "local :path" où path est l'emplacement d'un socket unix. Actuellement local est le seul mécanisme supporté. Les requêtes au service externe sont envoyés via le socket unix en datagrammes. Le service répond avec une valeur 4 octets contenant soit 0 soit 1.

Dans tous les cas le champ name doit spécifier un nom fqdn. Si aucun type n'est spécifié explicitement, cette règle matche tous les types excepté RRSIG, NS, SOA, NSEC, NSEC3. Les types peuvent être spécifiés par nom, incluant "ANY" qui matche tous les types sauf NSEC et NSEC3, qui ne peuvent jamais être mis à jours.

Plusieurs vues

Quand les vues sont utilisées, une zone peut être référencée par plus d'une d'entre-elles. Souvent, les vues contiennent différentes zones avec le même nom, permettant à différents client de reçevoir des réponses différentes pour la même requête. Parfois, plusieurs vues contiennent des zones identiques. La zone in-view fournis une manière efficace de le faire : elle permet à une vue de référencer une zone qui a été définie dans le vue précédemment définie.

Fichier de zone

Un nom de domaine identifie un nœud. Chaque nœud a un jeu d'informations de ressources, qui peut être vide. Ce jeu de ressources associé avec un nom particulier est composé de RR séparés. L'ordre RR dans un jeu n'est pas significatif et n'a pas besoin d'être préservé par les serveurs de nom, résolveurs, ou autres parties du DNS. Cependant, trier plusieurs RR est permis dans un but d'optimisation. Les composants d'un Resource Records sont :

owner name Le nom de domaine où le RR est trouvé

type Une valeur 16bits qui spécifie le type d'enregistrement

TTL le TTL du RR. Ce champs est un entier 32bits en secondes, utilisé par les résolveurs quand ils cachent les RR.

class Une valeur 16bits qui identifie une famille de protocole ou une instance de protocole.

RDATA La donnée de ressource. Le format des données est spécifique au type.

Les types suivants sont des RR valides :

A Une adresse IPv4 d'hôte décrite dans la rfc1035

AAAA adresse IPv6, décrite dans la rfc1886

A6 Adresse IPv6. Peut être une adresse partielle (un suffixe) et une indirection vers le nom où le reste de l'adresse peut être trouvée. Expérimental. Décrit dans la rfc2874.

AFSDB Emplacement des serveurs de base de données AFS. Expérimental. décris dans la rfc1183

APL Liste de préfixe d'adresse. Expérimental. Décrit dans la rfc3123

CERT Maintient un certificat numérique. Décris dans la rfc2538

CNAME Identifie le nom canonique d'un alias. décris dans la rfc1035

DHCID Uilisé pour identifier quel client DHCP est associé avec ce nom. Décris dans la rfc4701

DNAME Remplace le nom de domaine spécifié avec un autre nom de domaine à recherche. Alias effectivement toute une arborescence de l'espace de nom de domaine au lieu d'un simple enregistrement comme dans le cas d'un CNAME. Décris dans la rfc2672

DNSKEY Stocke une clé publique associée avec une zone DNS signée. Décrit dans la rfc 4034

DS Stocke le hash d'une clé publique associée avec une zone DNS signée. Décris dans la rfc4034

GPOS Spécifie la position globale. Remplacée par LOC

HINFO Identifie le CPU et l'OS utilisé par un hôte. Décris dans la rfc1035

IPSECKEY Fournis une méthode pour stocker un clé IPsec dans DNS. Décris dans la rfc4025

ISDN Resprésentation des adresses ISDN. Expérimental. Décris dans la rfc1183.

KEY Stocke une clé publique associée avec un nom DNS. Utilisé dans DNSSEC original, remplacé par DNSKEY dans DNSSECbis, mais reste utilisé avec SIG(0). Décris dans la rfc2535 et 2931

KX Identifie un échangeur de clé pour ce nom DNS. Décris dans la rc2230

LOC Pour stocker les informations GPS. Décris dans la rfc1876. Expérimental.

MX Identifie un échange de mail pour le domaine avec un préfixe 16-bits suivis par le nom d'hôte de l'échange de mail. Décris dans les rfc974 et 1035

NAPTR Name authority Pointer. Décris dans la rfc1706

NSAP Un point d'accès de service réseaux. Décris dans la rfc1706

NS Serveur de nom autoritatif pour le domaine. Décris dans la rfc1035

NSEC Utilisé dans DNSSECbis pour indiquer que les RR avec un owner name dans un certain interval de nom n'existent pas dans une zone et indique quels types de RR sont présent pour un nom existant. Décris dans la rfc4034

NSEC3 Identique à NSEC, mais est plus couteux pour le serveur et le client. Décris dans la rfc5155

NSEC3PARAM Utilisé dans DNSSECbis pour dire au serveur autoritatif quelles chaînes NSEC3 sont disponibles. Décris dans la rfc5155.

NXT remplacé par NSEC dans DNSSECbis. Dcéris dans la rfc2535

PTR Un pointer vers une autre partie de l'espace de nom de domaine. Décris dans la rfc1035

PX Fournis le mappage entre la rfc822 et les adresse X.400. Décris dans la rfc2163

RP Information des personnes responsable du domaine. expérimental. Décris dans la rfc1183

RRSIG Contients les données de signature DNSSECbis. Décris dans la rfrc4034

RT liaison route-through pour les hôtes qui n'ont pas leur propre aire d'adresse réseaux. Expérimental. Décris dans la rfc1183

SIG Contient les données de signature DNSSEC. Remplacé par RRSIG dans DNSSECbis. Décris dans la rfc2535 et 2931

SOA Identifie le départ de la zone d'autorité. Décris dans la rfc1035

SPF Contient le Sender Policy Framework pour un domaine de messagerie donné. Décris dans la rfc4408

SRV Information sur des services connus (remplace WKS). Décris dans la rfc2782.

SSHFP Fournis une manière sécurisée de publier des empreintes de clés ssh. Décris dans la rfc4255.

TXT enregistrement text. Décris dans la rfc1035

WKS Remplacé par SRV

X25 Représentation des adresse réseaux X.25. Expérimental. Décris dans la rfc1183

Les classes suivante sont valides :

IN l'Internet

CH Chaosnet.

HS Hesiod

Définir les TTL

Le TTL du champ RR est une entier 32 bits représenté en unités de secondes, et est principalement utilisés par les résolveurs quand ils cachent leur RR. Le TTL décris combien de temps un RR peut être maintenu en cache avant d'être supprimé. 3 types de TTL sont actuellement utilisés dans une zone :

SOA Le dernier champs dans le SOA est la TTL de cache négatif. Il contrôle le temps que d'autres serveurs vont maintenir en cache le no-such-domain (NXDOMAIN) pour vous. Le temps maximum est 3 heures.

$TTL La directive TTL en haut de la zone (avant le SOA) donne un TTL par défaut pour tous les RR sans jeu TTL spécifique.

RR TTLs Chaque RR peut avoir un TTL comme second champs dans le RR, qui va contrôler la durée en cache.

Mappage inverse dans IPv4

La résolution de nom inverse est accomplie au moyen d'un domaine in-addr.arpa et d'enregistrements PTR. Les entrées dans le domaine in-addr.arpa sont créés dans l'ordre inverse. La ligne $ORIGIN sert à fournir le contexte.

Autres directives de fichier de zone

Le format de fichier maître a été initialement définis dans la rfc1035 et a ensuite été étendu. Bien que le format de fichier maître lui-même est indépendant de la classe, tous les enregistrements dans un fichier maître doivent être de même classe. Les directives de fichier maître incluent $ORIGIN, $INCLUDE, et $TTL.

@ Utilisé dans un label ou nom, ce symbole représente l'origine courante. au début de la zone, c'est le nom de la zone.

$ORIGIN domain-name [comment] Jeux de nom de qui sont ajoutés à tout enregistrement non qualifié.

$INCLUDE filename [origin] [comment] Lit et traite le fichier comme s'il étair dans le fichier.

$TTL default-ttl [comment] Définis le TTL pour les enregistrements suivants.

Extension de fichier maître BIND

```
$GENERATE range lhs [ttl] [class] type rhs [comment]
```

$GENERATE est utilisé pour créer une série d'enregistrement qui diffèrent uniquement des autres par un itérateur. $GENERATE peut être utilisé pour faciliter la génération de jeux d'enregistrements requis pour supporter les sous-délégations inversées /24 décrites dans la rfc2317 : la délégation sans classe IN-ADDR.ARPA.

```
$ORIGIN 0.0.192.IN-ADDR.ARPA.
$GENERATE 1-2 @ NS SERVER$.EXAMPLE.
$GENERATE 1-127 $ CNAME $.0
```

est équivalent à :

```
0.0.0.192.IN-ADDR.ARPA. NS SERVER1.EXAMPLE.
0.0.0.192.IN-ADDR.ARPA. NS SERVER2.EXAMPLE.
1.0.0.192.IN-ADDR.ARPA. CNAME 1.0.0.0.192.IN-ADDR.ARPA.
2.0.0.192.IN-ADDR.ARPA. CNAME 2.0.0.0.192.IN-ADDR.ARPA.
...
127.0.0.192.IN-ADDR.ARPA. CNAME 127.0.0.0.192.IN-ADDR.ARPA.
```

Génère un jeu d'enregistrements A et MX. Noter que MX est un chaîne entre guillemets, qui sont enlevés au traitement :

```
$ORIGIN EXAMPLE.
$GENERATE 1-127 HOST-$ A 1.2.3.$
$GENERATE 1-127 HOST-$ MX "0 ."
```

est équivalent à :

```
HOST-1.EXAMPLE. A 1.2.3.1
HOST-1.EXAMPLE. MX "0 ."
HOST-2.EXAMPLE. A 1.2.3.2
HOST-2.EXAMPLE. MX 0 .
HOST-3.EXAMPLE. A 1.2.3.3
HOST-3.EXAMPLE. MX 0 .
...
HOST-127.EXAMPLE. A 1.2.3.127
HOST-127.EXAMPLE. MX 0 .
```

Statistiques

BIND9 maintient beaucoup de statistiques et fournis de nombreuses interfaces pour que les utilisateurs aient accès à ces statistiques. Les statistiques disponible incluent tous les compteurs qui étaient disponible dans BIND8 et ont du sens dans BIND9, plus d'autres informations. Les informations de statistiques sont catégorisés dans les sections suivantes :

Incoming Requests Le nombre de requêtes DNS entrantes pour chaque OPCODE

Incoming Queries Le nombre de demandes entrantes pour chaque type de RR

Outgoing Queries Le nombre de demandes sortantes pour chaque types de RR envoyé depuis le résolveur interne. maintenu par vue.

Name Server Statistics Compteurs de statistiques sans regarder les opérations de maintenance de zone telles que les transferts de zone.

Zone Maintenance Statistics Compteurs d'opérations de maintenances des zones telles que les tranfers de zone

Resolver Statistics Compteurs sur la résolution de noms dans le résolveur interne. Maintenu par vue.

Cache DB RRsets Nombre de RRsets par type de RR et noms non-existants stockés dans la base de cache. un ! indique un type de RRset étant non-existant (NXRRSET). Maintenu par vue.

Tocket I/O statistics Compteur de statistiques sur les événements réseaux

Un sous-jeu de Name Server Statistics est collecté et affiché par zone pour lequel le serveur a l'autorité quand zone-statistics est à yes. Ces compteurs sont affichés avec leur noms de zone et vue. Dans certains cas le nom de la vue est omis pour la vue par défaut.

Il y a actuellement 2 interfaces pour accéder aux statistiques. Une est en texte claire stocké dans le fichier statistics-file. L'autre est accessible à distance via un canal de statistiques.

Fichier de statistiques

Le format texte commence avec une ligne comme :

```
+++ Statistics Dump +++ (973798949)
```

Le nombre entre parenthèses est un horodatage Unix, mesuré en secondes depuis l'epoch. La suite est un jeu d'informations, qui sont catégorisés comme décris plus haut. Chaque section commence avec une ligne comme :

```
++ Name Server Statistics ++
```

Chaque section consiste de lignes, chacune contenant la valeur de compteurs suivie par sa description contextuelle. Pour simplifier, les compteurs avec une valeur de 0 ne sont pas affichés.

Le fichier se termine avec une ligne similaire à la première (même horodatage) :

```
-- Statistics Dump -- (973798949)
```

Compteurs

Les tables suivantes résument les compteurs statistique que BIND9 fournis.

Name Server Statistics Counters

Requestv4 Requêtes IPv4 reçues. Note : compte également les requêtes non répondues.

Requestv6 Requêtes IPv6 reçues. Note : compte également les requêtes non répondues.

ReqEdns0 Requêtes EDNS(0) reçues

ReqBadEDNSVer Requêtes avec une version EDNS non-supportée reçues

ReqTSIG Requêtes avec TSIG reçues

ReqSIG0 Requêtes avec SIG(0) reçues

ReqBadSIG Requêtes avec une signature TSIG ou SIG(0) invalide reçues

ReqTCP Requêtes TCP reçues

AuthQryRej Requêtes autoritatives rejetées

RecQryRej Requêtes récursives rejetées

XfrRej Requêtes de tranfers de zone rejetées

UpdateRej Requêtes de mise à jour dynamique rejetées

Response Réponses envoyées

RespTruncated Réponses tronquées envoyées

RespEDNS0 Réponses avec EDNS(0) envoyés

RespTSIG Réponses avec TSIG envoyés

RespSIG0 Réponses avec SIG(0) envoyés

QrySuccess Demande résultant une réponses réussie. (réponse NOERROR avec au moins un RR)

QryAuthAns Requêtes résultant une réponse autoritative

QryNoauthAns Requêtes résultant une réponse non-autoritative

QryReferral Requêtes résultant une réponse référantes.

QryNxrrset Requêtes résultant une réponse NOERROR sans données.

QrySERVFAIL Demandes résultant un SERVAIL

QryFORMERR Demandes résultant un FORMERR

QryNXDOMAIN Demandes résultant un NXDOMAIN

QryRecursion Demande qui ont impliqué une récursion pour trouver la réponse finale

QryDuplicate Demande que le serveur à tenter en récursion mais découvert qu'une demande avec la même IP, port, queryID, nom, type et classe à déjà été traité.

QryDropped Recherches récursive pour lequelles le serveur a découvert un nombre excessif de recherches pour le même nom, classe et type. C'est le nombre de recherches supprimées dues aux options clients-per-query et max-clients-per-query

QryFailure Autres erreurs de recherche.

XfrReqDone Demande de transfert de zone complétés

UpdateReqFwd Demandes de mise à jours forwardés

UpdateRespFwd Réponses de mise à jours forwardés

UpdateFwdFail Mise à jours dynamique forwardés échoués

UpdateDone Mise à jours dynamique forwardés complétées

UpdateFail Mise à jours dynamique échouées

UpdateBadPrereq Mise à jours dynamique rejetées à cause de prérequis non respectés

RateDropped Réponses supprimées par les limites

RateSlipped Réponses tronquées par les limites

RPZRewrites Réponses de ré-écriture de policy zone.

Zone Maintenance Statistics Counters

NotifyOutv4 notifications IPv4 envoyées

NotifyOutv6 notifications IPv6 envoyées

NotifyInv4 notifications IPv4 reçues

NotifyInv6 notifications IPv6 reçues

NotifyRej Notifications entrantes rejetées

SOAOutv4 Recherches de SOA IPv4 envoyées

SOAOutv6 Recherches de SOA IPv6 envoyées

AXFRReqv4 AXFR IPv4 demandées

AXFRReqv6 AXFR IPv6 demandées

IXFRReqv4 IXFR IPv4 demandées

IXFRReqv6 IXFR IPv6 demandées

XfrSuccess Demandes de transferts de zone réussies

XfrFail Demandes de transferts de zone échouées

Resolver Statistics Counters

Queryv4 Recherches IPv4 envoyées

Queryv6 Recherches IPv6 envoyées

Responsev4 Réponses IPv4 reçues

Responsev6 Réponses IPv6 reçues

NXDOMAIN NXDOMAIN reçues

SERVFAIL SERVFAIL reçues

FORMERR FORMERR reçues

OtherError Autres erreurs reçues

EDNS0Fail Recherches EDNS(0) échouées

Mismatch Réponses non-correspondantes reçues. le DNS ID l'adresse source et/ou le port sources de la réponse ne correspond pas à ce qui était attendu. Peut indiquer une tentative de cache poisoning

Truncated Réponses tronquées reçues

Lame Lame délégations reçues

Retry Retentatives de recherche effectuées

QueryAbort Recherche annulées du à un contrôle de quotas

QuerySockFail Erreur en ouvrant des sockets. Généralement due à un limitation des descripteurs de fichier.

QueryTimeout Timeouts de recherche

GlueFetchv4 recherches d'adresse NS IPv4 invoquées

GlueFetchv6 recherches d'adresse NS IPv6 invoquées

GlueFetchv4Fail recherches d'adresse NS IPv4 échouées

GlueFetchv6Fail recherches d'adresse NS IPv4 échouées

ValAttempt Validation DNSSEC tentées

ValOk Validation DNSSEC réussie

ValNegOk Validation DNSSEC sur des informations négatives réussies

ValFail validations DNSSEC échouées

QryRTTnn Table de fréquence des RTTs de recherche.

Socket I/O Statistics Counters

<TYPE>Open Sockets ouverts avec succès

<TYPE>OpenFail Erreur d'ouverture de sockets

<TYPE>Close Sockets fermés

<TYPE>BindFail Bind de sockets échoués

<TYPE>ConnFail Erreurs de connexions aux sockets

<TYPE>Conn Connections établies avec succès

<TYPE>AcceptFail Erreur d'acceptation de demandes de connexion entrantes. Non applicable à UDP

<TYPE>Accept Connexions entrant acceptées. Non applicable à UDP

<TYPE>SendErr Erreurs dans les opération d'envois sur socket

<TYPE>RecvErr Erreurs d'opérations de reception sur socket

chroot and setuid

BIND9 peut être lancé dans un environnement chroot. Il n'est pas nécessaire de compiler named statiquement, mais en fonction de l'OS, il peut être nécessaire de définir /dev/zero, /dev/random, /dev/log, et /etc/localtime.