Liste non exhaustive des questions potentielles pour l'examen de réseau avancé pour chacun des protocoles, connaître les ports utilisés (client et serveur) et le(s) protocole(s) de la couche transport utilisé.

**1. Questions sur la compréhension du protocole DNS:**

* **Champs du record SOA. Utilité de chacun des champs. Certains champs serviront de base pour le descriptif des transferts de zone.**

IN SOA <hôte source> <adresse mail de contact> <num de série> <tmp de rafraichissement> <intervalle d’essai> <tmp d’expiration> <TTL>

*Hôte source* : l’hôte sur lequel le fichier est maintenu.

*Adresse mail de contact* : mail internet pour la personne responsable du fichier de zone pour le domaine

*Num de série* : numéro de version du fichier de zone, incrémenté à chaque modification, sert pour déterminer si un transfert de zone est nécessaire entre le serveur maitre et le ou les serveurs de sauvegarde.

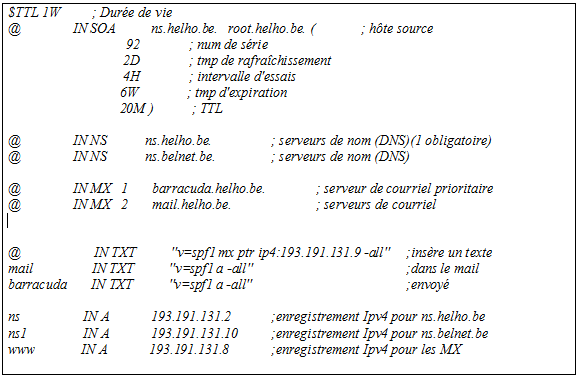
*Tmp de rafraichissement* : écart de temps en seconde qu’un serveur de nom secondaire doit attendre en 2 vérifications auprès du serveur principal pour vérifier si le fichier de zone a été modifier et qu’un transfert soit demandé.

*Intervalle d’essai* : intervalle de temps en seconde qu’un serveur secondaire doit attendre avant de réessayer un transfert de zone qui a raté.

*Tmp d’expiration* : intervalle de temps en seconde qu’un serveur secondaire doit continuer à essayer de changer de zone. Après expiration, les infos de vieille zone seront écartées.

*TTL* : temps de vie, la valeur TTL reprise dans l’enregistrement SOA est utilisée comme valeur de cache négative.

* **A** : enregistrement d’adresse IPV4 qui fait correspondre un nom d'hôte à une adresse IPv4 de 32 bits. Le nom d’hôte peut être sous la forme d’un FQDN en étant terminé par le « . » associé au TLD.
* **AAAA** : enregistrement d’adresse IPV6 qui fait correspondre un nom d'hôte à une adresse IPv6 de 128 bits.
* **CNAME** : enregistrement de nom canonique qui permet de créer un alias vers un nom d’hôte. Ces alias sont utiles lorsque des noms de domaines multiples sont résolus vers une même adresse IP ;
* **MX**: enregistrement d’échange de courriel (Mail Exchange) qui définit les serveurs de courriel pour ce domaine. Cet enregistrement est associé à une préférence dans le cas où plusieurs serveurs sont définis pour le même domaine.
* **NS**: enregistrement de serveur de nom qui définit les serveurs DNS de ce domaine. Il est important d’y renseigner le serveur primaire est les serveurs de sauvegardes pour que les avis de notifications puissent être envoyés lorsque le numéro de série du fichier de zone change et que le serveur est redémarré.
* **TXT**: enregistrement de type texte et qui permet à un administrateur d'insérer un texte quelconque. Nous retrouvons notamment dans cet enregistrement les informations de type SPF (Sender Policy Framework) permettant de valider les serveurs de messagerie utilisés pour un domaine donné. Ce mécanisme est utilisé dans les solutions de lutte contre le spam.



* **Décrire les différentes zones que l'on peut créer dans un serveur DNS et leur utilité.**

Maitre : Le serveur a une copie maitre des données et autorité sur la zone.

Esclave : réplique d’une zone maitre, contient la liste appelé « master » et contient une ou plusieurs adresses IP que l’esclave peut contacter pour mettre à jour la copie de sa zone.

Stub : semblable à une zone esclave sauf qu’elle ne contient que les enregistrements NS de la zone maitre.

Forward : utilisée pour rediriger toutes les demandes vers un autre serveur. Si aucune close « forwarders » n’est présente dans la zone ou qu’une liste vide est renseignée, aucun transfert ne sera effectué pour les demandes vers un autre serveur.

Hint : comprend l’ensemble des serveurs de noms racines.

* **Utilisation des requêtes récursives et itératives. Structure hiérarchisée du DNS**

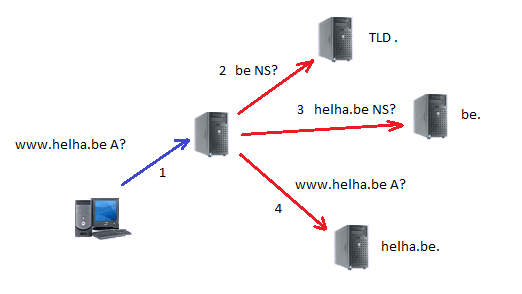
Il y a 2 types de demande vers un serveur DNS : récursives, itératives et inverse. Il faut garder à l’esprit qu’un serveur DNS peut être client d’un autre serveur DNS.

Le serveur DNS recevant la demande est responsable de la gestion du processus de résolution. Ce système doit envoyer la requête à un serveur différent pour chaque domaine.

La charge sur le serveur original provient d’une requête récursive : celle-ci place le moins possible de charges sur le client. Le serveur DNS recevant la requête peut retourner une réponse contenant l’adresse IP associée au nom ou un message d’erreur expliquant son incapacité à résoudre le nom. Le serveur est responsable de tous les appels vers les secteurs de plus haut niveau jusqu’à résolution du nom.

Une requête itérative nécessite un plus grand effort de la part du client. Lorsqu’un serveur reçoit une requête itérative, il ne retourne que l’information en sa possession sans effectuer de requête supplémentaire. S’il ne possède pas d’information, il peu fournir l’adresse de l’un des serveurs de noms racine.

Ces types de requêtes ne sont pas réservés aux clients. Quand un serveur DNS envoie une requête à un autre serveur DNS, il fonctionne alors comme un client et peut être configuré pour envoyer l’un des types de requêtes.



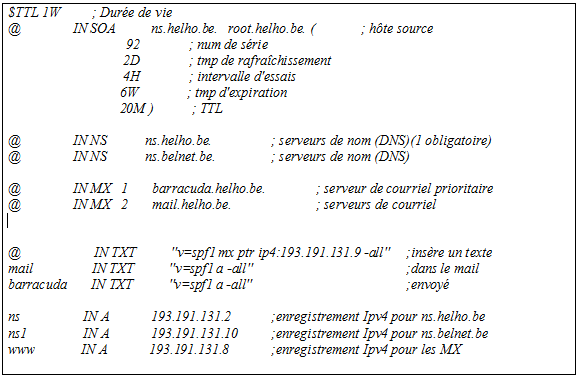
*Le résolveur envoie une requête récursive vers le serveur DNS paramétré dans sa pile TCPIP. S’il ne peut résoudre ce nom, le serveur envoie différentes requête itératives pour obtenir la réponse. Les serveurs DNS sont en général configurés pour accepter les requêtes itératives pour leurs zones mais uniquement les requêtes récursives pour leur client.*

Pour améliorer les performances lors des demandes de résolution, les résolveurs et les serveurs récursifs possèdent une cache interne. Dans un réseau local, on peut utiliser un serveur qui n’a autorité sur aucune zone mais dont le but est de servir de cache en relayant les demandes récursives vers un autre serveur externe.

Un serveur TLD doit fournir les informations au serveur de noms capable de répondre aux requêtes itératives. Un serveur récursif n’a pas la possibilité de déterminer de façon dynamique la liste des adresses de serveur TLD. Ces informations doivent être mise à jour lors de l’installation du serveur DNS.

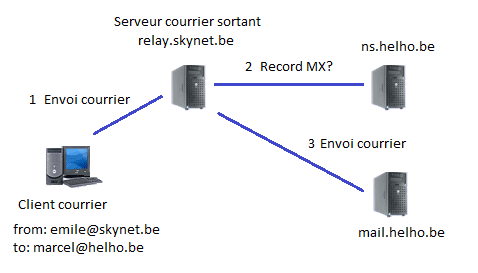
Les serveurs racines sont au nombre de 13 et appartiennent au même domaine root-servers.net. Ces serveurs sont gérés par 12 organisations : 2 européennes, 1 japonaise et 9 américaines. Ces serveurs sont répartit à travers le monde.

* **Sur base d'un exemple de topologie réseau comprenant des serveurs dont les caractéristiques FQDN, adresse IP et fonction (ex serveur mail) sont indiquées, donner le ou les fichiers de zone correspondants ainsi que le fichier de configuration avec les éventuelles options.**



* **Pouvoir expliquer l'utilité de champs particulier tels que les champs MX, champs TXT (de type SPF)**

MX = définit les serveurs de courriel pour le domaine. Record associé à une préférence dans le cas où plusieurs serveurs sont définis pour le même domaine.



TXT (SPF) = Permet de valider les serveurs de messagerie utilisés pour un domaine. Mécanisme utilisé dans les solutions contre le spam.

**2. Questions sur la compréhension du protocole DHCP:**

* Expliquer les différents modes d'attribution des adresses IP qu'un serveur DHCP peut fournir.
* Expliquer le mécanisme par lequel un client DHCP peut obtenir une adresse IP à partir d'un serveur DHCP. Nous partons du principe que le serveur se trouve sur le même réseau que le serveur DHCP. Reprendre notamment les 4 étapes et les expliciter (dhcpdiscover, dhcpoffer, dhcprequest, dhcpack)
* Expliquer le mécanisme par lequel un client DHCP peut obtenir une adresse IP à partir d'un serveur DHCP présent sur un autre sous réseau (utilité de l'IP Helper des routeurs)
* Expliquer le mécanisme par lequel un client DHCP peut demander une adresse IP précédemment obtenir.
* Expliquer les mécanismes de location d'adresses en faisant notamment référence aux différents temps utilisés et les actions prises par le client pour renouveler son adresse.

**3. Questions sur la compréhension du protocole FTP et TFTP:**

* Expliquer la gestion des canaux de commandes et de données dans le cadre d'un mode de connexion actif ou passif. Expliquer concrètement ces mécanismes par les différentes commandes utilisées dans le protocole. Expliquer les possibles problèmes au niveau de la configuration des firewall à traverser et les différentes solutions à apporter.
* Expliquer les deux mécanismes de cryptage et les problèmes éventuels rencontrés au niveau des firewall à traverser ainsi que les différentes solutions.
* Connaître les commandes de base permettant de s'authentifier et de pouvoir gérer les dossiers ainsi que les uploads et downloads.
* Effectuer une comparaison entre le protocole TFTP et FTP (au niveau des protocoles de la couche transport, des mécanismes d'authentification, gestion des dossiers...) Expliquer l'utilité du protocole TFTP.

**4. Questions sur la compréhension du protocole SMTP:**

* Dans les mécanismes d'échange de courriers, décrire les différents éléments constitutifs tels que: MUA, MSA, MTA, MDA... Décrire par un exemple simple l'envoi d'un mail d'une adresse A vers une adresse B.
* En vous basant notamment sur les différents champs dans l'entête d'un mail, expliquer comment on peut lutter contre le mécanisme de SPAM.
* Expliquer comment, par un exemple simple, tester l'envoi d'un mail vers un serveur SMTP en utilisant une connexion de type Telnet (en mode commande).
* Expliquer ce qu'est MIME ainsi que par un exemple comment on peut envisager l'envoi de fichiers attachés dans un mail. Donner un petit exemple commenté. Expliquer les différentes techniques de codage.
* Dans le cadre des possibilités d'utilisation de caractères accentués (ou étrangers) dans les sujets et parties de corps du document, expliquer les techniques d'encodage utilisées pour rester compatible avec le protocole SMTP de type ASCII

**5. Questions sur la compréhension du protocole IPV6:**

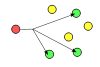
* **Décrire les différents types d'adresses IPV6 unicast et multicast.**

En IPv6, il n’y a plus de broadcast, on part donc sur de l’unicast et multicast.

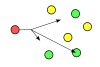
**Unicast:** Cette adresse identifie une seule interface. Un paquet envoyé à une adresse Unicast est reçu uniquement par l’interface dépositaire de cette adresse.\*



**Multicast:** Une telle adresse identifie un groupe défini de plusieures interfaces de plusieurs noeuds différents. Un paquet émis vers une adresse multicast est reçu par tous les membres du groupe multicast. Les adresses multicast commence par le préfixe ***FF00:/8***



**Anycast:** Ce type d’adresse est nouveau par rapport à IPv4. Une adresse anycast identifie un groupe d’interfaces de plusieurs noeuds (Comme le multicast). En revanche, lorsqu'un paquet est émis vers une adresse Anycast, il est reçu par l’un des membre du groupe anycast (N’importe lequel, mais a priori le plus proche en terme de routage réseau).



* **Expliquer les différentes adresses qu'un client peut posséder dans un environnement réseau classique.**

**Adresse de lien local (link-local): fe80::/10**

* Adresses attribuées aux cartes d’interface de façon dynamique.
* Valides uniquement dans un espace local ne traversant pas de routeur. Le préfixe d’une adresse de lien local est fe80::/10.
* Remplace l’ARP de l’IPv4
* Struture:

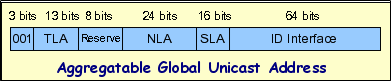


**Adresse de site local (Unique Local Addresse(ULA)): fec0::/10**

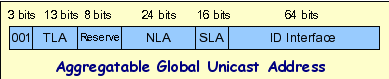
* Elle peut être comparée aux adresses privées de l’IPv4.
* Elles peuvent être routées mais ne peuvent être utilisées sur internet.

**Adresse utilisable sur Internet ( Global Unicast): 2001:db8:acaf:fd00::/64**

* Equivalent de nos adresses IPv4 publiques. On les définira plus précisément comme des destinations publiées sur l'Internet.
* Les routeurs transfèrent le trafic vers ces destinations. Elles sont donc "globalement routables".
* On les identifie par une valeur comprise entre 0x2000 à 0x3FFF sur les 4 premiers hexas de l'adresse.
* Structure:



* **Expliquer comment est construite la partie adresse réseau d'une adresse IPV6 unicast globale. Citez les différents intervenant dans les différentes parties de cette adresse**

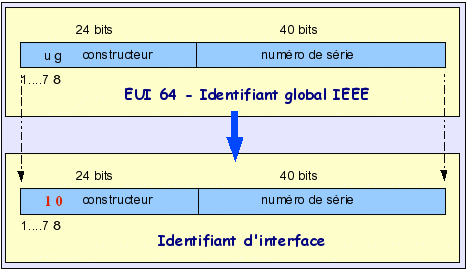


1. **Préfixe (3 bits):** Actuellement la valeur est de 001 et correspond à une adresse unicast publique.
2. **Top Level Agregator(TLA)(13bits):** correspond aux différents blocs libérés par **IANA** et donc le premier porte le numéro 1. Les 16 bits correspondants seront alors 001:0000000000001 et en hexadécimal, nous aurons 2001.
3. **Bits de réserve (8 bits) :**  Pour l'évolution de l'adressage, ces bits pourront être ré-attribués aux TLA ou NLA dans l'avenir car pour l'instant, ces besoins sont difficilement quantifiables.
4. **Next Level Aggregator (NLA) (24 bits)** **:** Représente l'unité d'agrégation basse (Belnet).

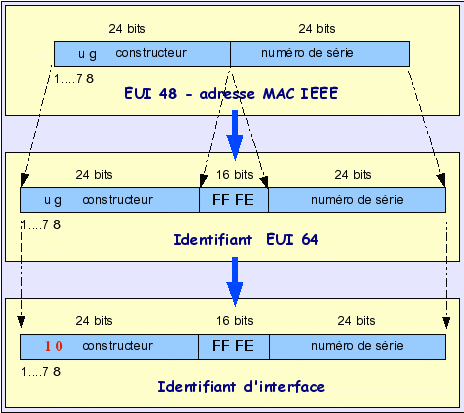
**Site Level Aggregator (SLA) (16 bits):** Ces bits sont sous la responsabilité du gestionnaire de site. Cette partie peut être hiérarchiser par le gestionnaire et définir ses propres sous réseaux dans cette plage (Proximus).  
  
***Pour résumer, les 48 premiers bits + les 16 bits suivants identifient la partie réseau de l'adresse IPv6, c'est-à-dire 64 bits (la première moitié haute de l'adresse IPv6 qui en comporte 128).***

* **Expliquer comment est construite la partie d'adresse IPV6 identifiant l'hôte.**

1. **Première solution :** Adresse physique qui serait dans un format **EUI-64** de l'**IEEE**. Il suffira alors de simplement insérer cette adresse physique dans les 64 bits de l'adresse IPv6



1. **Deuxième solution** : Adresses physiques représentées sur 48 bits comme le sont les adresses MAC de nos cartes réseau Ethernet. Il s'agit du format **EUI-48** de l'**IEEE**. Pour combler les 64 bits de l'adresse IPv6, l'opération suivante sera alors effectuée



* **Expliquer les différents modes d'attributions d'adresse IPV6 pour un hôte. Faire intervenir dans votre explication les notions de Router Sollicitation et Router Advertissement.**

On peut soit attribuer les adresses IPv6 de façon statique, indépendantes de l'EUI-64, ainsi que de désactiver l'auto configuration des adresses et de la route par défaut. Soit, on peut laisser l’attribution dynamique (auto-configuration) et là, on retrouve 2 modes d’attributions:

1. **Stateless (sans état):**

**Particularités:**

* Peut fonctionner **sans routeur.**
* Utilise le protocol ICMPv6.

**Principe:**

Le principe de base de l'auto-configuration sans état est qu'une machine génère son adresse IPv6 à partir d'informations locales et d'informations fournies par un routeur. Le routeur fournit à la machine les informations sur le sous-réseau associé au lien, il donne le préfixe.

**Fonctionnement:**

Le mécanisme d’auto configuration des IP en mode stateless prend aussi en charge les adresses globales grâce à des messages de type Router Solicitation et Router Advertisement avec les routeur voisins.

En stateless, on peut configurer les IP de façon automatique au travers de la découverte des préfixes. Pour chaque préfixe renseigné sur le routeur pour une interface donnée, on retrouve deux flags :

* + - Flag Autonomous : Instruit l’host qui reçoit la donnée en générant une adresse automatique à-p-d du préfixe. Si le flag = false, pas d’adresse générée sur base du préfixe.
    - Flag OnLink : Instruit l’host qui reçoit la donnée que le préfixe est OnLink et qu’il doit être utilisé dans la config automatique, e.g. : dans la table de routage de l’host.

Dans cette solution, aucune config n’est envisagée pour les mécanismes de résolution de nom. On doit donc soit configurer les informations manuellement soit utiliser un serveur DHCPv6 qui se limite à attribuer des IPv6 stateless et d’attribuer des options comme : serveur DNS, serveur de temps, etc…

Dans la config des messages Router Advetisement sur un routeur, on peut retrouver deux flags Managed Configuration et Other Parameters Configuration -> permet de définir la façon dont les IP et paramètres sont obtenus automatiquement par un serveur DHCPv6. Les deux flags sont **M et O flag**

1. **Statefull (avec état):**

**Particularités:**

* offre une information de configuration plus riche et un contrôle sur l'affectation des paramètres de configuration.
* bâti sur le modèle du client-serveur et repose sur l'utilisation du protocole DHCPv6.
* **Expliquer les mécanismes par lesquels les informations de routage sont fournies à un hôte de façon dynamique par les routeurs. Expliquer l'utilité des différents flags.**

Dans les serveurs DHCPV4, il était possible de pouvoir définir des routes et une passerelle par défaut comme options à fournir sur les clients. Dans les premières normes de la version DHCPV6, ces options n'étaient pas présentes. A l'heure actuelle, ces options ont été ajoutées mais pas encore implémentées dans tous les serveurs et/ou clients DHCP.

Comme nous l'avons vu précédemment, la passerelle par défaut est ajoutée automatiquement lors de la découverte des routeurs. Pour ce qui est des tables de routage, elles seront générées sur base des préfixes définis sur les routeurs. Il suffira pour chacun des préfixes, de mettre le flag 'autonomous' à 0 et le flag 'OnLink' à 1. -> via les RA

Si plusieurs routeurs sont présents sur le même segment, nous pourrons définir un ordre de préférence pour les différentes routes fournies si des routes pouvaient être redondantes. L'hôte pourra donc voir un routeur comme routeur primaire ou routeur secondaire. (RFC 4191). Nous pouvons donner trois valeurs: High, Medium, Low

Pour la passerelle par défaut, il est possible si plusieurs routeurs existent de pouvoir supprimer la passerelle de l'hôte pour un routeur déterminé. Il suffira pour le routeur en question de placer le default lifetime à 0. Un hôte recevant cette propriété écartera l'adresse de lien du routeur si le lifetime vaut 0.

* **Expliquer comment IPV6 supprime le brodcast et le remplace notamment dans les mécanismes de résolution ARP qui était propre à IPV4 et utilisait du broadcast. Expliquer les différents groupes multicast auxquels un hôte adhère.**

**all nodes multicast group** ( ff02::1) - reçu et traité par tous les noeuds sur le lien local

**all routers multicast grou**p ( ff02::2 ) - reçu et traité par tous les routeurs sur le lien local

**Solicited node multicast address** ( ff02::1:ff<24-dernier-bits> )

*address unicast = 2001:db8:acad::1fc8:12a1* -> snma = ff02::1:ffc8:12a1

*Intérêt de l'utilisation de la solicited node address*: Par rapport au bon vieux broadcast, cette méthode (Voir Question 5) fait appel à de multiples groupes multicast dont peu de nodes font partie en même temps. Il est bien entendu que le multicast sur un segment Ethernet de base se traduit par du broadcast au niveau 2. Cependant, dans le cadre d'équipements de niveau 2 et 3 gérant effectivement le multicast (chaque adhésion multicast IPv6 donne lieu à une adhésion au niveau 3, par ND, mais également au niveau 2 par du broadcast Ethernet), l'utilisation du ND et donc du multicast permettrait de réduire de façon spectaculaire le nombre de broadcasts sur le réseau, et de limiter le trafic de résolution des adresses MAC de façon très considérable, en tous cas de le convertir de broadcast à quasi-unicast.

ARP est remplacé par le protocole **NDP** (Neighbor Discovery Protocol). Il permet donc à un hôte de “découvrir” l’adresse de couche 2 (link-layer addr., *e.g.* MAC) en connaissant son adresse de couche 3 (*e.g.* IP). L’interface source (A) envoi un paquet ICMPv6 de type 135 (**Neighbor solicitation message**) avec comme adresse de destination la *solicited node multicast address* de l’hôte dont il connaît l’adresse ip. Cela permet d’éviter une surcharge du trafic broadcast en couche 2, car en effet, la trame n’aura plus comme adresse de destination une adresse de diffusion (FFFF.FFFF.FFFF) mais une adresse multicast (3333.FF<24-dernier-bits-addr.-unicast>). La conséquence est donc que seul les hôtes concernés recevront *et* traiteront le paquet NS. Ces derniers répondront par un paquet ICMPv6 de type 136 (**Neighbor Advertisement Message**), contenant dans le champ de données l’adresse de couche 2.

**6. Questions sur la compréhension du protocole DNSSEC:**

* **Expliquer comment sont assurés l'authenticité et l'intégrité des réponses fournies par un serveur DNSSEC? Faire intervenir dans vos réponses l'utilisation des enregistrements DNSKEY, RRSIG, ZSK et KSK.**

Afin d’assurer l’authenticité et l’intégrité des réponses, DNSSEC est basé sur un modèle de cryptographie à clé publique.

L’enregistrement DNSKEY est utilisé pour transmettre une clé publique entre le résolveur et le serveur de nom. Cette clé publique est associée à la clé privée que va utiliser le serveur d’autorité pour signer les hash des enregistrements RRSET. Les RRSET (ressources record set) sont un regroupement d’enregistrement en fonction du nom et du type de ces enregistrements.

Une clé publique est associée à une zone non à un serveur. Le résolveur utilisera cette clé publique écrite dans l’enregistrement DNSKEY pour vérifier la signature du serveur d’autorité.

On retrouve 2 types de clé : KSK et ZSK. La clé KSK signera seulement les DNSKEY RRSET tandis que la clé ZSK signera les autres RRSET dans la zone.

**ZSK (zone signing key)** : correspond à la clé privée utilisée pour signer une zone, c’est-à-dire les enregistrements demandés par les serveurs récursifs. Une clé n’expire jamais, seules les signatures RRSIG ont une période de validité. Mais pour des questions de sécurité, il est conseillé de renouveler la clé tous les mois. Les serveurs DNSSEC intègre un mécanisme de rotation de clé de sorte que chaque mois, la clé est changée.

Une 1ère technique consiste à générer une nouvelle clé et à signer la zone avec cette clé en laissant l’ancienne clé présente dans le fichier de zone le temps de l’expiration des données dans la cache mais cette technique est pénalisante du fait d’une double signature pour chaque enregistrement.

Une 2ème technique consiste à pré publier la nouvelle clé. On retrouve une clé active et une clé passive dans le sens où une seule des 2 est utilisée pour signer les enregistrements de la zone. On peut ajouter la nouvelle clé juste avant la valeur TTL de la clé ou les 2 clés en même temps au début du de la période du mois. En fin de période, la zone est signée avec la clé passive, elle-même signée avec la clé KSK.

**KSK (key signing key)** : clé privée destinée à signer une clé de zone (ZSK), n’est utilisée que pour signer un enregistrement DNSKEY. Les signatures des clés ZSK ont, en général, une durée de validité de 1 an. Le remplacement de cette clé est plus complexe du fait que l’on doit également intervenir dans la zone parent.

1. On ne peut controler quand le parent changera ou publiera le nouvel enregistrement DS.
2. On doit maintenir l’ancienne KSK le temps que le TTL de l’enregistrement DS arrive à expiration pour éviter des problèmes de caches.

Il faut aussi tenir compte du fait que le parent puisse accepter ou pas plusieurs enregistrements DS pour une même zone enfant.

Pour les parents n’acceptant qu’un seul enregistrement DS :

1. Introduire une nouvelle KSK dans la zone enfant.
2. Signer les ZSK avec la nouvelle KSK en maintenant les anciennes signatures.
3. Attendre que les caches aient la nouvelle version de la zone.
4. Publier le nouvel enregistrement DS chez le parent.
5. Attendre que le parent publie ce nouvel enregistrement.
6. Attendre le plus long temps entre la TTL de l’ancien et du nouveau DS.
7. Déprécier d’ancienne KSK et supprimer les anciennes signatures ZSK.

Pour les parents acceptant plusieurs enregistrements DS :

1. Introduire une nouvelle KSK dans la zone enfant
2. Signer les ZSK avec la nouvelle KSK en maintenant les anciennes signatures
3. Publier le nouvel enregistrement DS en maintenant l’ancien
4. Attendre que le parent publie ce nouvel enregistrement DS
5. Attendre le plus long temps entre la TTL de l’ancien et du nouveau DS
6. Déprécier l’ancien KSK et supprimer les anciennes signatures ZSK
7. Supprimer l’ancien enregistrement DS du parent

* **Expliquer comment assurer l' authenticité et l'intégrité dans le cas de réponses négatives ( un enregistrement demandé n'existe pas). Faire intervenir dans vos réponses l'utilisation des enregistrements NSEC et NSEC3**

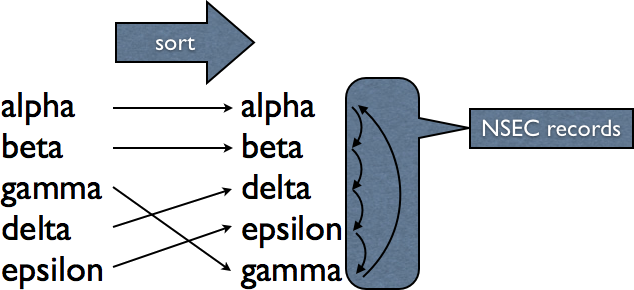
NSEC : un enregistrement NSEC contient les informations nécessaires à l’identification des enregistrements existants pour un nom donné, ainsi que le prochain nom existant dans la zone. Ces 2 informations suffisent pour prouver qu’un enregistrement ou qu’un domaine n’existe pas.

Exemple :

Ns1.esit.be. 3600 NSEC [www.esit.be](http://www.esit.be). A AAAA RRSIG NSEC

[www.esit.be](http://www.esit.be). Correspond au nom de domaine suivant.

A AAAA RRSIG NSEC correspond à une liste de RSET correspondant au domaine suivant. Le NSEC du dernier nom de domaine pointe vers le 1er nom de domaine de la liste.



Le problème de ce type d’enregistrement est la possibilité sans que le transfert ne soit validé pour une zone, de récupérer l’ensemble des différent noms de domaine. C’est pour éviter ce genre de problème que 2 autres types d’enregistrement sont apparus : NSEC3PARAM et NSEC3.

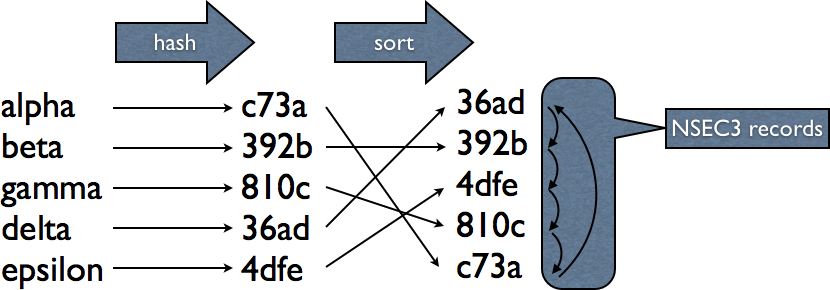
NSEC3 : permet la gestion des réponses négatives tout en luttant contre des attaques de type « énumération de zone ». L’enregistrement NSEC3 utilise une fonction de hachage appliquée aux noms renvoyés. Ainsi, sera envoyé le hash du prochain nom de la zone, ne permettant pas de reconstruire la zone entière.

NSEC3PARAM : utilisé par les serveurs DNS pour calculer et déterminer quels sont les enregistrements NSEC3 à inclure pour les réponses de non existence.

Il comprendra les enregistrements suivants :

* Algorithme de hashage
* Flag « opt-out » indiquant si les délégations sont signées ou pas
* Itérations : combien de fois l’algo de hashage doit-il être appliqué
* Salt : valeur de salt pour le calcul de hashage

La hashage fonctionne comme ceci : lorsque le résolveur voudra vérifier la réponse négative, il va générer le hash du nom demandé, exemple : h2 et vérifier que h2 est compris entre h1 et h3 (indiqué dans l’enregistrement NSEC3). Il est conseillé de remplacer régulièrement le salt.



* **Expliquer le mécanisme par lequel un chaînage de confiance est établi entre un serveur DNSSEC et son parent. Faire intervenir dans vos réponses l'utilité de la clef KSK et l'utilisation de l'enregistrement DS.**

L’enregistrement DS (delegation signer) assure la chaine de confiance dans les liaisons entre les serveurs DNS parents et enfants. Il sera placé dans la zone parent et sera lié à l’enregistrement DNSKEY de la zone enfant. Il contiendra également un hash de la clé KSK de la zone enfant, stockée avec l’enregistrement NS indiquant la délégation vers la zone enfant. Cet enregistrement DS est signé avec le reste des données de la zone parente.

Cet enregistrement est demandé par le résolveur au serveur autoritaire dès qu’il a besoin de vérifier une signature reçue par la zone fille. La zone mère lui indique qu’il peut faire confiance à ce serveur en lui renvoyant un hash de la clé signé par elle-même. Le résolveur sait ainsi qu’il peut lui faire confiance.

A chaque maillon de la chaine, on retrouve le même schéma : un enregistrement DS permet d’avoir la confiance en une KSK, la KSK permet d’avoir confiance en l’ensemble des clés de la zone et les ZSK permettent alors d’avoir confiance dans les autres enregistrements.

Exemple graphique DNSSEK avec esit.be :

ROOT « . »

fr

…

be

…

helha

esit



esit.be

* **Expliquer les raisons nécessitant la mise en place d'un mécanisme de rotation de clefs. Expliquer ce mécanisme pour les clefs de type ZSK et KSK**

On ne peut controler quand le parent changera ou publiera le nouvel enregistrement DS.

On doit maintenir l’ancienne KSK le temps que le TTL de l’enregistrement DS arrive à expiration pour éviter des problèmes de caches.

Il faut aussi tenir compte du fait que le parent puisse accepter ou pas plusieurs enregistrements DS pour une même zone enfant.

Pour les parents n’acceptant qu’un seul enregistrement DS :

1. Introduire une nouvelle KSK dans la zone enfant.
2. Signer les ZSK avec la nouvelle KSK en maintenant les anciennes signatures.
3. Attendre que les caches aient la nouvelle version de la zone.
4. Publier le nouvel enregistrement DS chez le parent.
5. Attendre que le parent publie ce nouvel enregistrement.
6. Attendre le plus long temps entre la TTL de l’ancien et du nouveau DS.
7. Déprécier d’ancienne KSK et supprimer les anciennes signatures ZSK.

Pour les parents acceptant plusieurs enregistrements DS :

1. Introduire une nouvelle KSK dans la zone enfant
2. Signer les ZSK avec la nouvelle KSK en maintenant les anciennes signatures
3. Publier le nouvel enregistrement DS en maintenant l’ancien
4. Attendre que le parent publie ce nouvel enregistrement DS
5. Attendre le plus long temps entre la TTL de l’ancien et du nouveau DS
6. Déprécier l’ancien KSK et supprimer les anciennes signatures ZSK
7. Supprimer l’ancien enregistrement DS du parent

Une 1ère technique consiste à générer une nouvelle clé et à signer la zone avec cette clé en laissant l’ancienne clé présente dans le fichier de zone le temps de l’expiration des données dans la cache mais cette technique est pénalisante du fait d’une double signature pour chaque enregistrement.

Une 2ème technique consiste à pré publier la nouvelle clé. On retrouve une clé active et une clé passive dans le sens où une seule des 2 est utilisée pour signer les enregistrements de la zone. On peut ajouter la nouvelle clé juste avant la valeur TTL de la clé ou les 2 clés en même temps au début du de la période du mois. En fin de période, la zone est signée avec la clé passive, elle-même signée avec la clé KSK.

7. Questions sur la compréhension du protocole HTTP:

* Etre à même de pouvoir décrire les différents champs de l'entête pour gérer les éléments suivants: les cookies (dépôt d'un cookie sur le serveur, récupération d'un cookie envoyé par le serveur, validité d'un cookie) gestion de la cache dans les différentes normes du protocole et sur tous les niveaux (client, proxy...), gestion des sites virtuels.
* Citer les différentes grandes familles de codes que le serveur peut retourner en réponse à une requête.
* Décrire les différents champs présents dans les requêtes et les réponses pour pouvoir gérer l'encodage et la fonctionnalité multilingue possible des applications web.