Les faiblesses du DNS

Gilles Guette et Bernard Cousin

IRISA, Campus de beaulieu, 35042 Rennes Cedex

L'accès aux services sur l'Internet repose sur l'utilisation massive de l'infrastructure DNS (Domain Name System). À l'origine, aucune mesure pour la sécurisation du DNS n'a été prise. Le DNS qui est pourtant une ressource critique de l'Internet demeure vulnérable. Pour remédier à cette situation, DNSsec fut créé par l'IETF afin de garantir l'intégrité des données du DNS ainsi que l'authentification de la source de ces données. DNSsec se base sur la cryptographie à clé publique pour fournir différents services de sécurité. Nous allons voir dans ce document les différentes vulnérabilités du DNS ainsi que les solutions apportées par DNSsec puis nous terminerons par une analyse des faiblesses du DNSsec.

Keywords: résolution de nom, DNSsec, cryptographie, authentification, intégrité.

1 Introduction

L'accès aux services sur l'Internet repose sur l'utilisation massive de l'infrastructure DNS (Domain Name System) [Moc87a, Moc87b, AL02]. Le but du DNS est de fournir un service de traduction appelé résolution de noms. À l'origine, le système de résolution de noms était basé sur le fichier `HOSTS.TXT` qui contenait les informations nécessaires à la traduction nom-adresse et inversement. Le problème d'une telle méthode (un unique fichier) est qu'elle est devenue inadaptée face à la croissance du nombre de machines connectées à l'Internet. C'est pourquoi le choix d'une structure arborescente a été fait. La structure arborescente permet une délégation aisée de la gestion des zones formées par des parties de l'arbre.

Le DNS est une ressource critique et nécessaire au bon fonctionnement des applications Internet, mais il est aussi très vulnérable comme nous pouvons le voir sur la figure 1 où est représentée une architecture simplifiée du DNS d'une zone. Une description des éléments composant l'architecture se trouve dans la deuxième section. Les différents échanges de données possibles sont représentés par des flèches épaisses (requête/réponse et transfert de zone).

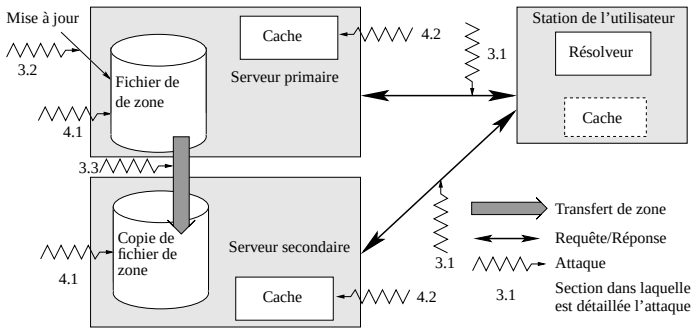

FIG. 1: Les vulnérabilités du DNS.

Les flèches brisées indiquent différents points vulnérables du système. Certaines de ces vulnérabilités sont présentées dans [AA02] et [LMMM00].

Dans la deuxième section de ce document, nous présentons le fonctionnement du DNS. Dans les troisième et quatrième parties les attaques visant le DNS sont décrites d'abord au niveau des transactions puis

au niveau des données stockées sur les serveurs de noms. Dans la cinquième section, nous détaillons une solution pour la sécurisation du DNS : DNSsec [Eas99, Wel00, Gun02]. Enfin nous montrons dans la dernière section quelques faiblesses du DNSsec.

2 Le fonctionnement du DNS

Pour pouvoir communiquer au travers d'un réseau, deux machines doivent tout d'abord connaître leurs adresses respectives. Le Domain Name System (DNS) est chargé, entre autres, d'effectuer la traduction d'un nom (plus facile à mémoriser) vers une adresse IP et inversement.

Deux entités du DNS servent à la résolution de noms : le *serveur de noms* qui maintient les informations sur la zone et le *résolveur* qui extrait les données du serveur de noms pour répondre à la requête d'un client.

Le serveur de noms a la charge de stocker les informations d'une zone dans le fichier de zone. Sa fonction essentielle est de répondre aux requêtes en utilisant les données de son fichier de zone, ou s'il ne possède pas les informations nécessaires, de faire suivre la requête à un serveur de noms plus à même de répondre. Deux critères sont essentiels pour un serveur de noms : la rapidité de réponse et la fiabilité du service. Pour assurer la rapidité de réponse les serveurs de noms et/ou les résolveurs peuvent posséder un cache dans lequel ils mémorisent les enregistrements auxquels ils ont eu accès récemment. Afin d'assurer la fiabilité du service DNS, la politique d'implantation des serveurs requiert généralement que toutes les zones soient supportées par plusieurs serveurs de noms, les serveurs secondaires assurant la redondance du serveur primaire.

Le résolveur se situe entre l'application qui demande la résolution du nom et le serveur de noms. Il est situé sur la même machine que l'application. C'est lui qui reçoit la requête de l'application, interroge le serveur de noms et renvoie la réponse à l'application. Il est généralement composé d'un ensemble minimal de routines et délègue la totalité du travail aux serveurs de noms. Ce type de résolveur est appelé résolveur basique ou *stub resolver*. Le résolveur peut disposer de son propre cache pour stocker les données auxquelles il a déjà accédées car l'un des critères important d'un résolveur est de minimiser (voire d'éliminer) les délais du réseau et de résolution des requêtes.

2.1 Services nécessaires en sécurité

En sécurité, on distingue 4 services qui peuvent être fournis par la cryptographie [Sch96] :

- *L'authentification* : le destinataire d'un message doit pouvoir s'assurer de son origine. Un intrus ne doit pas être capable de se faire passer pour quelqu'un d'autre.
- *L'intégrité* : le destinataire d'un message doit pouvoir vérifier que celui-ci n'a pas été modifié durant son acheminement. Un intrus doit être incapable de faire passer un message modifié pour légitime.
- *La confidentialité* : un intrus récupérant des données ne doit pas être en mesure de les interpréter.
- *La non-répudiation* : l'expéditeur d'un message ne doit pas pouvoir, par la suite, nier avoir envoyé le message.

L'authentification et l'intégrité des données sont deux services absolument nécessaires au DNS pour fournir un service sûr. En effet, il faut être en mesure de vérifier que les données reçues d'un serveur de noms n'ont pas été modifiées et que c'est bien le serveur interrogé qui les a envoyées. Dans la conception initiale du DNS, aucune mesure n'a été prise pour assurer la sécurité de ce service et aucun des services de sécurité présentés ci-dessus n'est fourni.

Dans la suite de ce document nous ne nous intéressons qu'à l'authentification et à l'intégrité des données issues du DNS. En effet, la confidentialité est contraire à la philosophie du DNS qui veut que toutes les données soient publiques.

La section suivante présente des attaques contre le DNS, montrant ainsi la faiblesse de ce système.

3 Attaques visant à corrompre les transactions DNS

Il existe deux types de transactions dans le DNS. Celles qui servent à la résolution de noms (requêtes et réponses) et les transactions de maintenance de la base de données (messages de mise à jour et de transfert de zone).

Nous présentons dans un premier temps des attaques ciblant les messages de résolution, puis dans un second temps nous présentons des attaques ciblant les messages de mise à jour et de transfert de zone.

3.1 Attaques visant à corrompre les requêtes DNS

Les requêtes DNS circulent sans protection ni controle d'intégrité sur le réseau. Il est donc possible, pour quelqu'un capable d'intercepter un message DNS, de le modifier.

En effet, il suffit d'analyser le trafic sur le réseau à l'écoute d'une requête DNS (ce qui est particulièrement aisé sur les réseaux locaux où un support commun est partagé par plusieurs stations). Lorsqu'une requête DNS passe, l'attaquant la duplique, l'analyse et envoie une réponse erronée à cette requête avant que le serveur de noms n'ait eu le temps de répondre.

Il existe des variantes à cette attaque sur les requêtes DNS s'appuyant sur des équipements spécifiques. Supposons que l'attaquant ait pris le contrôle d'un routeur ou qu'il ait modifié les tables de routage de celui-ci pour faire passer les paquets par une machine sous son contrôle. Le principe est le même que lors de la première attaque présentée.

Un résolveur envoie une requête DNS pour une résolution de noms par exemple, cette requête passe par le routeur piraté qui la fait suivre normalement vers son destinataire. Celui-ci répond et, lorsque la réponse arrive au routeur dont l'attaquant a pris le contrôle, il bloque la réponse et modifie les champs adéquats (c'est-à-dire insère les données falsifiées qu'il veut transmettre et modifie certaines sections de l'en-tête). Une fois la réponse modifiée, le routeur la transmet à son véritable destinataire.

Les deux attaques que nous venons de présenter permettent de montrer la vulnérabilité des messages DNS et la simplicité avec laquelle ces attaques peuvent être mise en place. Il existe en effet des logiciels appelés *sniffer* qui permettent d'écouter tout ce qui passe sur le réseau sur lequel ils sont présents. Il sont facilement utilisables pour récupérer des messages DNS.

L'impact des attaques par interception de messages est gênant puisqu'elles permettent de rediriger une demande de connexion (à un site bancaire par exemple) vers n'importe quelle machine (notamment celles sous le contrôle du pirate) ouvrant ainsi la possibilité de récupérer des information confidentielles.

3.2 Attaques visant à corrompre les messages de mise à jour DNS

Lorsque des modifications de topologie ont lieu dans la zone (nouvelle adresse attribuée, présence d'une nouvelle machine, etc.), le serveur primaire en est informé par un message de mise à jour. Tout comme les requêtes et les réponses DNS, les messages de mise à jour sont vulnérables. Il y a deux manières d'exploiter ces vulnérabilités : en interceptant un message de mise à jour ou en créant un message de mise à jour complet. Les méthodes d'attaque par interception de messages sont similaires à celles présentées dans la section précédente. Si l'attaque se fait par la création d'un message DNS complet, deux cas peuvent se présenter : le serveur qui reçoit le message n'effectue pas l'authentification de la source émettrice des données ou il effectue une authentification de la source grâce à son adresse IP, par exemple.

Dans le premier cas, l'attaquant n'a aucune contrainte supplémentaire à la création de son message. Dans le second cas, il devra masquer son adresse IP et se faire passer pour une machine qui a le droit d'effectuer une mise à jour, comme le serveur DHCP [Dro97] par exemple (le serveur DHCP a la charge d'attribuer des adresses IP à des machines nouvellement connectées).

Sans authentification sûre de la source émettrice des données et sans contrôle des droits de cette source à effectuer cette opération, le serveur primaire met à jour sa base de données en fonction des informations contenues dans le message. Contrairement aux attaques visant les requêtes et les réponses DNS qui permettent de tromper un résolveur, les attaques visant à falsifier les messages de mise à jour DNS ont un impact plus étendu. En modifiant les données contenues dans le fichier de zone du serveur primaire, les données corrompues se propagent grâce au fonctionnement normal des mécanismes DNS et atteignent ainsi les serveurs secondaires, les caches des serveurs de noms locaux, puis les résolveurs des machines connectées à ces serveurs.

Il est possible d'intervenir de la même manière sur les messages de transfert de zone.

3.3 Attaques visant à corrompre les messages de transfert de zone

Le serveur primaire est le seul à posséder le fichier de zone original. Les serveurs secondaires doivent récupérer ces informations, c'est-à-dire faire une copie du fichier de zone soit périodiquement, pour éviter

que la copie dont ils disposent soit trop vieille (*refresh*), soit parce qu'ils ont été avertis que le fichier original a été modifié (*notify*). Cette transaction est appelée transfert de zone. Il est important de noter que ce mécanisme est le seul qui permette aux serveurs secondaires d'obtenir le fichier de zone.

Les méthodes d'attaques sont les mêmes que celles présentées ci-dessus pour les messages de mise à jour : l'interception de message et la création d'un message de transfert de zone. Dans les deux cas le but est de fournir de fausses informations à un serveur secondaire qui les stockera comme copie du fichier de zone et s'en servira pour répondre aux requêtes DNS qu'il recevra.

Les messages de mise à jour et de transfert de zone sont très vulnérables car ils ne bénéficient d'aucune protection. En effet, aucun mécanisme ne garantit l'intégrité des messages et l'authentification de la source. Ces deux services permettraient pourtant de fournir une parade efficace aux attaques présentées précédemment, comme nous le verrons ultérieurement. Le paragraphe suivant étudie les attaques possibles contre les données DNS stockées sur les serveurs de noms.

4 Attaques visant les enregistrements stockés sur les serveurs

Nous venons de voir que les transactions du DNS sont vulnérables, tant les messages de résolution (requêtes ou réponses) que ceux de mise à jour. La présence de plusieurs serveurs secondaires implique l'existence de copies du fichier de zone.La section suivante s'intéresse aux données stockées sur les serveurs.

4.1 Corruption du fichier de zone

Rappelons que l'infrastructure du DNS est arborescente et partitionnée en plusieurs domaines eux-mêmes découpés en différentes zones. Chaque zone est gérée individuellement.

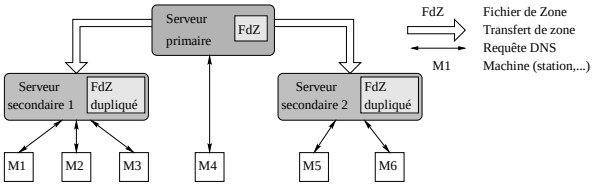

FIG. 2: Organisation des serveurs d'une zone.

Comme le montre la figure 2, une zone comporte au moins un serveur primaire qui possède les données de la zone dans un fichier local. La zone peut aussi posséder un ou plusieurs serveurs secondaires qui récupèrent périodiquement une copie du fichier de zone en la demandant au serveur primaire. Chaque serveur secondaire se sert de sa copie du fichier de zone pour répondre aux requêtes DNS qu'il reçoit.

L'attaquant peut prendre le contrôle d'un serveur de noms (primaire ou secondaire) et en modifier le fichier de zone. S'il s'agit d'un serveur secondaire d'une zone, il transmettra des informations fausses à toutes les machines qui l'interrogeront. S'il s'agit d'un serveur primaire, la zone est entièrement corrompue car c'est à partir de lui que s'effectue le transfert de zone, c'est-à-dire l'envoi d'une copie du fichier de zone aux serveurs secondaires.

Les serveurs de noms hébergeant le fichier de zone ou une copie de celui-ci sont des cibles privilégiées. Un autre type d'attaque peut être mené lorsque l'attaquant possède les droits nécessaires sur un serveur de noms primaire, soit parce qu'il les a usurpés ou soit tout simplement parce qu'il en est l'administrateur. Ce type d'attaque a été décrit par Steve M. Bellovin [Bel95] et repris par Christoph L. Schuba [Sch93]. Dans son article, Bellovin montre la faiblesse des "r-commandes" de Berkeley du type `rlogin`, `rsh`, etc. Avant que Berkeley ne produise un correctif pour combler une déficience du démon `rlogin`, celui-ci se servait de l'adresse IP de la machine qui voulait se connecter, pour retrouver le nom de cette machine (associé à l'adresse IP). Il vérifiait ensuite si la machine ayant ce nom avait le droit de se connecter. Il était alors facile pour quelqu'un disposant d'un serveur DNS de modifier le nom correspondant à l'adresse IP d'une machine en sa possession afin de s'introduire dans d'autres systèmes.

4.2 Pollution de cache

Pour améliorer les performances du DNS, c'est-à-dire diminuer les temps de réponse lors d'une résolution de noms, les serveurs de noms utilisent un cache dans lequel ils placent les enregistrements du DNS auxquels ils ont accédés récemment. L'utilisation d'un cache permet, si la réponse s'y trouve, d'éviter d'envoyer une requête sur le réseau et ainsi de diminuer la charge réseau et le temps de réponse. La confiance accordée par le serveur aux données qu'il a placées en cache font de celui-ci une cible particulièrement intéressante pour les attaques. Ce type d'attaque est appelé pollution de cache (ou *cache poisoning*). Une telle attaque est détaillée ci-dessous.

Chaque message DNS est caractérisé par un identifiant de 16 bits qui est le premier champ de l'en-tête de ce message. Cet identifiant est repris dans la réponse, permettant ainsi l'association avec la requête correspondante. Le principe de l'attaque que nous allons présenter maintenant (figure 3) est de trouver la valeur de cet identifiant pour générer une réponse en apparence correcte, forcer un serveur de noms à mettre en cache une réponse fausse et polluer ainsi son cache.

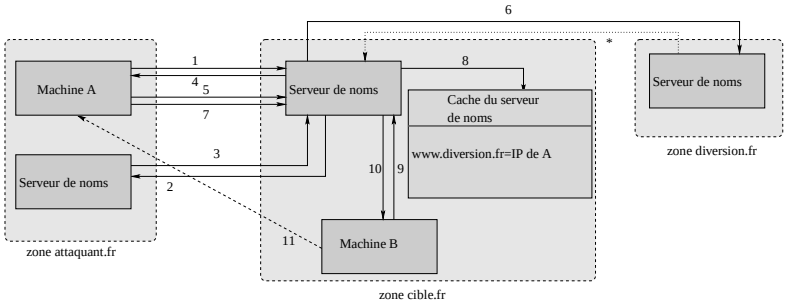

FIG. 3: Pollution de cache.

Le déroulement de l'attaque est le suivant : la machine attaquante A interroge le serveur de noms de la zone `cible.fr` (étape 1) en demandant l'adresse IP d'une machine se trouvant dans la zone `attaquant.fr`. Le serveur de noms de la zone `cible.fr` va interroger le serveur de noms de la zone `attaquant.fr` (étape 2) qui va lui donner l'adresse IP demandée (étape 3) et le serveur de noms de la zone `cible.fr` fera suivre cette adresse IP à la machine attaquante A (étape 4). À ce moment la machine A récupère le champ ID qui contient l'identifiant du message DNS.

L'attaque proprement dite commence à partir de la cinquième étape, A envoie une requête pour avoir l'adresse de `www.diversion.fr` (étape 5). Le serveur de noms de la zone `cible.fr` va interroger le serveur de noms de la zone `diversion.fr` (étape 6). Pendant ce temps la machine A inonde (étape 7) le serveur de noms de la zone cible avec des réponses à sa propre demande en indiquant que `www.diversion.fr` est accessible par l'adresse IP de A. Ces requêtes successives sont identiques, à l'exception du champ ID qui est incrémenté à chaque réponse envoyée, l'ID de départ étant celui récupéré lors de l'étape 4. Une fois le bon identifiant atteint, le serveur de noms accepte ce message comme étant la réponse correcte et la place dans son cache (étape 8). Á partir de cet instant, la réponse du serveur de noms de la zone `diversion.fr` sera ignorée.

Dès qu'une machine B de la zone `cible.fr` demande l'adresse de `www.diversion.fr` à son serveur de noms, le serveur de noms se servant de son cache pour répondre lui indiquera l'adresse IP de la machine A et B dialoguera avec A pensant s'adresser à `www.diversion.fr`.

L'étape $*$ indique que le serveur de noms de la zone `diversion.fr` répond normalement à la requête qui lui a été envoyée mais que cette réponse est ignorée, une réponse étant déjà parvenue au serveur de noms. Le but de cette attaque est de placer de fausses informations dans le cache d'un serveur de noms, afin que celui-ci se serve en priorité des fausses informations.

Les enregistrements stockés par les serveurs de noms ne sont pas plus sécurisés que les messages DNS et souffrent aussi de l'absence d'un mécanisme garantissant leur intégrité. La difficulté pour corrompre les enregistrements stockés sur les serveurs repose sur la difficulté à s'introduire sur la machine hébergeant

le fichier de zone. Une fois qu'une intrusion sur la machine hôte a été réalisée, les données DNS sont vulnérables. De plus les mécanismes de cache sont aussi vulnérables si l'on ne peut pas contrôler l'origine des données que l'on veut mettre en cache. La section suivante présente une évolution du DNS en vue de sa sécurisation, le DNSsec.

5 La solution DNSsec

Le DNS sécurisé s'appuie essentiellement sur une cryptographie à clé publique et sur des signatures numériques. Il a pour objectif de garantir l'intégrité des données et l'authentification. Pour utiliser ces mécanismes et générer les signatures numériques, chaque zone possède une (ou plusieurs) paire de clés (publique/privée) appelées clés de zone. Ces clés servent à signer les enregistrements contenus dans le fichier de zone. Chaque serveur doit être configuré de manière à supporter l'existence de deux paires de clés simultanément pour assurer le recouvrement temporel lors du changement d'une paire de clés arrivant à expiration. Il ne faut pas oublier que le DNS est un service très sollicité, nécessitant la recherche du bon compromis entre sécurité et performance. Ne rien faire pour garder de bonnes performances sans aucune sécurité est très dangereux mais utiliser systématiquement la cryptographie provoquerait un effondrement des performances, ce qui est aussi inacceptable de la part d'un tel service.

Pour stocker les clés, les signatures et des données nécessaires au DNSsec, trois types d'enregistrement ont été ajoutés au DNS, il s'agit des enregistrements de types `KEY`, `SIG` et `NXT`. Détaillons chacun de ces types d'enregistrement ainsi que leur utilité.

5.1 Les enregistrements spécifiques au DNS sécurisé

DNSsec s'appuie sur une cryptographie à clé publique et la clé publique de zone doit être rendue disponible. C'est le rôle de l'enregistrement KEY. Il est chargé de contenir une clé publique de zone, les informations concernant les différentes utilisations possibles de cette clé, ainsi que les protocoles et les algorithmes avec lesquels elle peut être utilisée. La clé privée servant à générer les signatures des enregistrements est conservée dans un endroit sûr. Les signatures seront stockées dans un enregistrement spécifique au DNSsec, l'enregistrement de type SIG.

Le processus de signature fonctionne de la manière suivante : chaque enregistrement du fichier de zone est signé avec la clé privée de zone. La signature ainsi produite est placée dans un enregistrement de type SIG. Chaque enregistrement de type SIG est lié au type d'enregistrement dont il contient la signature.

Si un tiers modifie les enregistrements contenus dans le fichier de zone, la signature associée ne sera plus correcte, et la vérification de cette signature indiquera que les enregistrements ont été modifiés. Ceci garantit l'intégrité des enregistrements. De plus, seul le serveur interrogé possède la clé capable de générer les signatures correctes. Si la vérification des signatures est positive, il est ainsi possible d'assurer que c'est le propriétaire de la clé privée qui a envoyé les enregistrements, la source des données étant ainsi authentifiée.

Le problème qui se pose est d'envoyer une réponse sécurisée pour une question portant sur un nom ou des enregistrements qui n'existent pas. Si un résolveur demande l'adresse d'une machine qui n'existe pas, il recevra en réponse un message contenant un code d'erreur dans l'en-tête du message DNS, sans signature vérifiable. Pour éviter cette solution, DNSsec possède un nouveau type d'enregistrement, l'enregistrement NXT (non-existant). Cet enregistrement contient un vecteur de bits indiquant tous les types d'enregistrements existant pour le nom associé à l'enregistrement NXT, ainsi que le prochain nom se trouvant dans le fichier de zone selon l'ordre établi (ordre lexicographique sur les labels et du label le plus à droite vers celui le plus à gauche). Un enregistrement NXT possède aussi son enregistrement SIG associé, ainsi il est possible de déduire de manière sécurisée, qu'un nom n'existe pas, s'il se situe entre le nom du domaine associé à l'enregistrement NXT et le nom contenu dans l'enregistrement NXT, ou qu'un type d'enregistrement n'existe pas, si le bit associé à ce type est à 0 dans le vecteur de bits de l'enregistrement NXT reçu en réponse.

Nous avons maintenant les enregistrements nécessaires à la sécurisation du DNS, mais il est possible de générer une paire de clés (publique/privée), de signer des enregistrements illicites et de répondre à une requête DNS interceptées, avec ceux-ci. Pour éviter ce cas de figure, il faut donc pouvoir avoir confiance dans la clé publique qui est envoyée par le serveur interrogé. Il existe deux moyens :

– Soit cette clé est configurée statiquement par l'administrateur, dans ce cas on lui fait confiance sans vérification, c'est une *Trusted Key* (ou clé de confiance).
– Soit on est capable de valider un chemin sécurisé à partir d'une clé de confiance jusqu'à la clé fournie, c'est ce qu'on appelle une chaîne de confiance.

Une des *Trusted Keys* est généralement la clé publique de la zone racine soutenue par 13 serveurs de noms. La zone racine est la zone la plus haute dans l'arbre DNS, celle où commence toutes les résolutions de noms.

5.2 Les méthodes RFC 2535 et Delegation Signer

Le point névralgique est donc l'enregistrement KEY, car il faut être en mesure de lui faire confiance. Deux propositions existent pour construire une chaîne de confiance. Celle introduit par DNSsec, défini dans le RFC 2535 [Eas99] se base sur les enregistrements existants. La seconde définit un nouvel enregistrement, l'enregistrement DS (*Delegation Signer* [Gun02]), qui permet à la zone parente de valider l'enregistrement KEY de sa zone fille.

Le RFC 2535 propose une variante dans le processus de signature des enregistrements, ceci uniquement pour l'enregistrement de type KEY d'une zone. Cette variante est utilisée lorsque la zone parente de la zone fille est elle-même sécurisée, c'est-à-dire utilise aussi DNSsec. C'est alors la clé privée de la zone parente qui va être utilisée pour signer l'enregistrement KEY de la zone fille, établissant ainsi une relation de confiance entre ces deux zones. Le processus est le suivant : l'administrateur de la zone fille envoie son enregistrement KEY à l'administrateur de la zone parente, celui-ci signe l'enregistrement KEY et renvoie la signature. L'enregistrement KEY de la zone fille est maintenant signé par la clé de zone de sa zone parente. Ainsi, créer la signature de l'enregistrement KEY de la zone fille nécessite deux messages. Cette procédure doit être reproduite à chaque fois qu'une modification est faite sur l'enregistrement KEY de la zone fille ou lorsque la date de validité de la signature est dépassée.

La méthode *Delegation Signer*, utilise un enregistrement DS stocké dans la zone parente, qui contient entre autres, l'identifiant de la clé publique de la zone fille et un hachage du nom de la zone fille concaténé avec la valeur de la clé de la zone fille. Pour obtenir l'enregistrement DS on procède de la manière suivante : l'administrateur de la zone fille signe son *Keyset* avec sa clé privée, envoie son enregistrement KEY contenant la clé publique correspondante à l'administrateur de la zone parente. Celui-ci crée l'enregistrement DS correspondant qui est conservé dans la zone parente. Il y a un seul message envoyé pour générer l'enregistrement DS et un nouveau message sera envoyé pour modifier l'enregistrement DS uniquement lorsque la clé de zone de la zone fille, qui a signé son enregistrement KEY, change.

5.3 Le processus de validation de la chaîne de confiance

Lorsqu'un résolveur envoie une requête DNS, il contacte un des serveurs de noms de sa zone. Si celui-ci n'a pas la réponse à la requête et qu'aucune des zones intermédiaires ne se trouve dans son cache, la recherche de la réponse commence par l'interrogation d'un serveur racine qui enverra l'adresse du serveur de noms de la zone intermédaire le plus à même de répondre.

La requête sera alors envoyée à ce serveur de noms qui retournera la réponse ou le nom d'un autre serveur de noms. Le processus sera réitéré jusqu'à l'obtention de la réponse ou d'un message d'erreur. Lors de cette recherche, le serveur qui envoie une réponse envoie aussi les enregistrements de type KEY et SIG(KEY) de sa zone, ainsi que l'enregistrement DS adéquat si la zone le possède. Le RFC 2535 précise que tout enregistrement placé dans la section *Answer* d'un message DNS doit y être accompagné de l'enregistrement SIG associé. À la fin de la résolution nous disposons donc de la liste des serveurs de noms ayant fourni une réponse partielle ou la réponse demandée, ainsi que leurs clés publiques respectives et la signature de ces clés. Il reste à présent à vérifier ces informations.

Les signatures sont vérifiées grâce aux clés fournies par l'enregistrements KEY de la zone parente si la zone est conforme au RFC 2535, ou grâce à l'enregistrement KEY de la zone fille et à l'enregistrement DS correspondant stocké dans la zone parente.

5.4 La protection des messages de transfert de zone

Grâce aux mécanismes présentés précédemment, DNSsec est capable de garantir l'intégrité des données stockées sur les différents serveurs de noms ainsi que l'authentification de la source de ces données. Il reste

néammoins un problème à résoudre : les signatures numériques protègent les enregistrements auxquels elles sont associées, mais ne fournissent aucune garantie sur l'intégrité des messages envoyés. En effet, l'en-tête du message n'est pas signé, il peut donc être modifié, notamment les champs précisant le nombre d'enregistrements dans les différentes sections du message. Il est ainsi possible d'augmenter le paramètre indiquant le nombre d'enregistrements présents dans le message et d'insérer des enregistrements illégaux (cependant leur signature et leur date de validité doivent être correctes, il s'agit d'un rejeu).

Pour éviter de telles manipulations, il existe deux mécanismes fournissant l'intégrité des messages envoyés. Il s'agit de TSIG [VGEW00] et de SIG(0) [Eas00].

L'enregistrement TSIG contient la signature de l'intégralité du message et des variables TSIG. Il garantit l'authentification et l'intégrité du message entier et utilise des mécanismes de cryptographie à clé symétrique. L'utilisation d'algorithmes à clé symétrique a l'avantage de rendre le traitement cryptographique plus rapide mais il y a tout de même un inconvénient : ce mécanisme ne peut pas passer à l'échelle à cause du nombre de clés à gérer (une clé pour chaque couple de machines). TSIG peut donc être utilisé entre un serveur primaire et ses serveurs secondaires pour sécuriser le transfert de zone, en effet le nombre de serveurs de noms restant raisonnable pour la plupart des zones, la gestion des clés pose moins de difficultés.

Un autre mécanisme existe pour signer les transactions, il s'agit de type SIG(0) [Eas00]. Il fonctionne exactement comme l'enregistrement SIG, c'est-à-dire qu'il contient une signature de tout le message y compris l'en-tête. Une attention particulière doit être portée sur la protection de la clé privée qui doit alors être gardée disponible en permanence sur le seveur de noms pour permettre la signature des messages.

Si TSIG ou SIG(0) sont utilisés, un tiers ne peut modifier ou ajouter des informations dans le message sans en modifier la signature. Si la signature est modifiée le message DNS est rejeté. Malheureusement ces mécanismes ne sont pas utilisables systématiquement. En effet, l'utilisation de la cryptographie pour signer chaque message puis vérifier les signatures prend du temps et impliquerait une forte dégradation des temps de réponse et des performances du DNS.

Les mécanismes fournis par DNSsec permettent d'apporter une parade aux attaques présentées précédemment en garantissant l'intégrité des données et l'authentification de la source de ces données, au prix d'une augmentation des coûts. Un tiers ne peut plus modifier les messages DNS à sa guise, ni en envoyer en se faisant passer pour quelqu'un d'autre. Le DNSsec n'est cependant pas parfait et possède aussi des faiblesses comme le montre la section suivante.

6 Les faiblesses de DNSsec

DNSsec sécurise les transactions DNS, que ce soit pour les mises à jour des fichiers de zone ou pour les requêtes de résolution de noms, ainsi que les données stockées sur les différents serveurs de noms. Mais DNSsec reste vulnérable, notamment à cause de l'enregistrement de type NXT, de l'augmentation de la taille des messages DNS et de l'existence de plusieurs versions de DNSsec incompatibles entre elles.

6.1 NXT Walk

Comme nous l'avons dit dans la section précédente, un enregistrement de type NXT est associé à chaque nom. Cet enregistrement possède deux champs. Le premier contient le prochain nom se trouvant dans la zone et le second un vecteur de bits représentant tous les types d'enregistrements possibles. Pour tous les types d'enregistrement présents dans cette zone, les bits correspondants sont positionnés à 1. Ainsi, lorsqu'un résolveur effectue une demande concernant un nom ou un type n'existant pas, il reçoit en réponse l'enregistrement NXT qui lui permet de prouver la non-existence. Il est possible de détourner l'utilisation de cet enregistrement en envoyant des requêtes successives sur des ressources inexistantes afin de récupérer tous les enregistrements NXT présents dans le fichier de zone. Avec cet ensemble d'enregistrements il est possible de cartographier totalement la zone interrogée.

Cette utilisation du champ NXT n'est pas considérée comme une faiblesse par tout le monde. En effet, toutes les données du DNS étant publiques, il est possible par des requêtes licites d'obtenir les enregistrements désirés. Ce que l'on peut reprocher à l'enregistrement NXT c'est qu'il fournit beaucoup d'informations et facilite cette méthode de collecte d'informations. Une tentative de réduction de la divulgation d'informations est présentée dans [Jos00, Jos01] mais cette solution ne semble pas avoir été retenue par le groupe de travail de l'IETF. L'idée de Josefsson est de fournir un hash du nom suivant, à la place du nom

complet présent dans l'enregistrement NXT, afin de diminuer les données dévoilées par l'enregistrement NXT.

6.2 Déni de service

Pour garantir la sécurisation des transactions, DNSsec a ajouté de nouveaux enregistrements au DNS, notamment les enregistrements de type SIG. Tout ceci a un coût en terme de place et de temps de traitement, la taille d'un fichier de zone DNSsec étant en moyenne sept fois plus grande que la taille du même fichier de zone pour le DNS. Le DNS est conçu pour utiliser en premier lieu le protocole UDP, avec une taille maximum de message égale à 512 octets pour minimiser le temps de traitement induit par la segmentation. Avec DNSsec, la limite fixée pour le DNS est dépassée beaucoup plus facilement. Lorsque DNSsec ne peut communiquer par UDP à cause du dépassement de cette limite il utilise le protocole TCP avec une négociation de connexion et donc un accroissement de la charge du réseau et un délai supplémentaire.

Néammoins, il est possible de détecter les machines effectuant un nombre important de requête sur un court laps de temps, grâce à leurs adresses IP. On peut ainsi développer une politique de détection d'attaque par déni de service, en fixant un seuil de tolérance.

Aucun système n'est vraiment à l'abri d'une attaque par déni de service et le DNS ne fait pas exception à la règle. L'accroissement de la taille des messages, l'augmentation de la charge du réseau et l'augmentation de la charge de travail pour un serveur de noms ou un résolveur lorsqu'il doit vérifier les signatures, sont autant d'éléments qui rendent DNSsec encore plus sensible à ce type d'attaque.

6.3 Cohabitation 2535/DS

Nous avons vu qu'il existe deux techniques d'implémentation de DNSsec, le RFC 2535 et le draft IETF DS. Ces deux techniques diffèrent sur un point essentiel de DNSsec, la signature de l'enregistrement KEY. Le processus de signature étant différent, les enregistrements SIG associés aux enregistrements KEY ne contiennent pas la même valeur, c'est-à-dire que selon la méthode employée, l'enregistrement KEY n'est pas signé avec la même clé (il est signé avec la clé de la zone pour la méthode *Delegation Signer* et avec la clé de la zone parente pour le RFC 2535). De plus, dans le draft IETF DS un nouvel enregistrement est défini, l'enregistrement DS.

Ces deux méthodes sont incompatibles. Lorsque deux serveurs appartennant à deux zones complétement sécurisées et utilisant chacune une technique différente veulent communiquer, le demandeur sera incapable de valider la chaîne de confiance parce qu'il ne saura pas vérifier les signatures. Dans ce cas, si le serveur de noms envoyant la requête possède une politique de sécurité *lâche*, il échangera des messages DNS classiques avec le serveur de noms interrogé et les messages seront ainsi vulnérables aux attaques précédemment énoncées, bien que les deux machines se trouvent dans des zones sécurisées. Si le résolveur ou le serveur de noms envoyant la requête possède une politique de sécurité *stricte* il refusera la réponse et les deux machines ne pourront pas communiquer.

Il n'existe pas de solutions miracles à ce problème de compatibilité. La seule éventualité raisonable est d'abandonner 2535 afin de mettre en place au plus vite une infrastructure basée sur DS, qui est une solution plus mature.

6.4 Cohabitation DNS/DNSsec

De la même manière, la cohabitation entre les deux systèmes DNS et DNSsec est un obstacle à une sécurité optimale. En effet, le modèle topologique qui semble se distinguer est celui des *îlots de sécurité*, c'est-à-dire d'une suite dans l'arbre DNS de zones sécurisées et de zone non-sécurisées. Ceci est dû au compromis entre l'augmentation de travail nécessaire à la sécurisation de la zone et le besoin réel de sécurité de cette zone. Certaines zones peuvent se considérer comme peu sujettes aux attaques et peuvent ne pas implémenter de tels mécanismes de sécurité.

Une telle topologie pose un problème lors de la résolution de noms si un serveur de noms fournissant une réponse totale ou partielle à la requête appartient à une zone non-sécurisée. Les enregistrements contenus dans le message ne sont plus protégés par des signatures numériques, ils sont donc vulnérables et peuvent être modifiés afin d'envoyer au résolveur une réponse falsifiée. Le seul moyen d'obtenir une réponse sécurisée, est que la totalité des serveurs ayant fournis une réponse appartiennent à des zones sécurisées par le DNSsec, que tous les messages échangés contiennent les enregistrements et leurs signatures associées

et que la chaîne de confiance puisse être validée. En cas de manquement à l'une de ces conditions, une politique de sécurité *stricte* refusera la réponse que lui fournit le serveur de noms.

7 Conclusions

DNSsec est un apport essentiel en terme de sécurité, il garantit (grâce à différents mécanismes) l'intégrité des données ainsi que l'authentification de l'origine des données échangées, permettant ainsi d'apporter une parade à la plupart des attaques contre le DNS classique, telles que la modification de messages DNS.

DNSsec est prévu pour atteindre une sécurité optimale si son développement est total, c'est-à-dire s'il remplace entièrement le DNS et qu'il n'existe pas de cohabitation DNS/DNSsec. Le modèle d'*îlots de sécurité*, c'est-à-dire une succession de zones sécurisées et de zones non sécurisées, ainsi que différentes implémentations DNSsec incompatibles entre elles, diminuent les capacités du DNSsec. En effet, dès qu'une zone non sécurisée est traversée par une résolution, on ne peut plus faire confiance aux données reçues car elles ont pu être modifiées. Il faut donc tendre vers une acceptation d'un standard unique, en l'occurrence DS, afin d'aboutir à une sécurité optimale.

Les besoins de sécurité toujours croissants tendent à l'implémentation totale de DNSsec pour assurer une sécurité maximale. La charge de travail supplémentaire introduite reste raisonnable pour des ordinateurs toujours plus puissants.

Références

[AA02] D. Atkins and R. Austein. Threat Analysis Of The Domain Name System. Draft IETF, Nov 2002.

[AL02] P. Albitz and C. Liu. *DNS and BIND*. O'Reilly & Associates, Inc., Sebastopol, CA., fourth edition, Jan 2002.

[Bel95] S. M. Bellovin. Using the Domain Name System for System Break-Ins. In *Proceedings of the fifth Usenix UNIX Security Symposium*, pages 199–208, Salt Lake City, UT, Jun 1995.

[Dro97] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, Mar 1997.

[Eas99] D. Eastlake. Domain Name System Security Extensions. RFC 2535, Mar 1999.

[Eas00] D. Eastlake. DNS Request and Transaction Signatures (SIG(0)s). RFC 2931, Sep 2000.

[Gun02] O. Gundmundsson. Delegation Signer Resource Record. Draft IETF, Dec 2002.

[Jos00] S. Josefsson. Authenticating denial of existence in DNS with minimum disclosure. Draft IETF, Nov 2000.

[Jos01] S. Josefsson. Network Application Security Using The Domain Name System. Master's Thesis, Royal Institute of Technology, Department of Numerical Analysis and Computer Science, 2001.

[LMMM00] A. Lioy, F. Maino, M. Marian, and D. Mazzocchi. DNS Security. In *Terena Networking Conference*, May 2000.

[Moc87a] P. Mockapetris. Domain Names - Concept and Facilities. RFC 1034, Nov 1987.

[Moc87b] P. Mockapetris. Domain Names - Implementation and Specification. RFC 1035, Nov 1987.

[Sch93] C. L. Schuba. Addressing Weaknesses in the Domain Name System. Master's Thesis, Purdue University, Department of Computer Sciences, Aug 1993.

[Sch96] B. Schneier. *Applied cryptography*. John Wiley & Sons, Inc., New York, N.Y., second edition, 1996.

[VGEW00] P. Vixie, O. Gudmunsson, D. Eastlake, and B. Wellington. Secret Key Transaction Authentication for DNS (TSIG). RFC 2845, May 2000.

[Wel00] B. Wellington. Secure Domain Name System (DNS) Dynamic Update. RFC 3007, Nov 2000.