Étude des conséquences de clés compromises dans DNSSEC

Gilles Guette

IRISA, Campus de Beaulieu, 35042 Rennes CEDEX, FRANCE
gilles.guette@irisa.fr

Le système de noms de domaine est une base de donnée hiérarchique et distribuée basée sur un modèle arborescent. Le protocole DNS est largement utilisé pour effectuer essentiellement la correspondance entre un nom de machine et son adresse IP. Les extensions de sécurité du DNS (DNSSEC) ont été conçues pour protéger ce protocole. Pour cela, DNSSEC utilise la cryptographie à clé publique ainsi que des signatures numériques. Une zone DNSSEC possède au moins une paire de clés (publique/privée) pour signer ses données DNS et fournir ainsi deux services de sécurité essentiels : l'intégrité et l'authenticité des données. Pour faire confiance à des données DNS, un client DNSSEC doit en vérifier les signatures numériques avec la clé de zone appropriée. Cette vérification est basée sur l'établissement d'une chaîne de confiance entre des zones sécurisées. Pour construire cette chaîne, le client a besoin d'un point d'entrée sécurisé : une clé de zone configurée dans le client comme clé de confiance. Puis, le client doit trouver un chemin sécurisé partant de ce point jusqu'aux données DNS demandées. Les clés de zones sont essentielles au fonctionnement de DNSSEC et sont utilisées à toutes les étapes d'une résolution de nom. Dans ce papier, nous présentons une étude des conséquences d'une clé compromise sur le protocole DNSSEC. Nous décrivons les attaques pouvant alors être menees grâce à une clé compromise et nous présentons les actions possibles en cas de compromission.

Mots-clés: DNSSEC, sécurité réseau, clés compromises, révocation

1 Introduction

Le *Domain Name System* (DNS) [Moc87a, Moc87b, AL02] est une base de données distribuée et hiérarchique utilisée le plus souvent pour effectuer la correspondance entre un nom de machine et son adresse IP. Dans sa conception originelle, le DNS n'inclut aucun service de sécurité tels que l'intégrité ou l'authentification, ce qui laisse ce protocole vulnérable [Bel95, Sch93, AA04, GC03]. Pour pallier ces vulnérabilités, l'*Internet Engineering Task Force* (IETF) a standardisé les extensions de sécurité DNS (DNSSEC).

DNSSEC [Eas99, AAL$^+$05a, AAL$^+$05c, AAL$^+$05b] repose sur l'utilisation de la cryptographie à clé publique pour fournir l'intégrité et l'authenticité des données DNS. Chaque nœud de l'arbre DNS, appelé *zone*, possède au moins une paire de clés publique/privée utilisée pour générer les signatures numériques des informations de zone. L'unité de base de ces informations est l'enregistrement de ressource (RR). Chaque RR possède un type particulier qui indique le type des données qu'il contient. Par exemple, un enregistrement DNSKEY contient une clé publique de zone, un enregistrement RRSIG contient une signature et un enregistrement A contient une adresse IPv4.

Pour faire confiance à des données DNS, un résolveur (le client DNS) construit une chaîne de confiance [Gie01] en partant d'un point d'entrée sécurisé dans l'arbre DNS [KSL04] (c'est-à-dire d'une clé de confiance configurée statiquement dans le résolveur), jusqu'à l'enregistrement de ressource demandé. Un résolveur est capable de construire une chaîne de confiance s'il possède un point d'entrée sécurisé et s'il ne traverse que des zones sécurisées jusqu'à la ressource demandée.

Dans ce papier, nous présentons dans la section 2 le fonctionnement de DNSSEC, les différents types de clés et le processus de vérification. Puis, dans la section 3 nous étudions les conséquences d'une clé compromise dans DNSSEC. Enfin, dans la section 4 nous présentons les actions possibles en cas de compromission d'une clé de zone.

2 DNSSEC : principes et fonctionnement

Le système de noms de domaine est organisé selon un modèle arborescent. L'arbre DNS est divisé en domaines et en zones.

2.1 Domaines et zones

Un domaine est un sous-arbre complet de l'arbre DNS. Le nom d'un domaine est obtenu par la concaténation des étiquettes de chaque nœud de l'arbre en commençant à la racine du sous-arbre et en remontant jusqu'à la racine de l'arbre DNS. Comme deux nœuds frères ne peuvent pas avoir la même étiquette, l'unicité des noms est garantie. Un domaine peut être inclus dans un autre domaine, par exemple le domaine `irisa.fr.` est inclus dans le domaine `fr.`, comme le montre la figure 1.

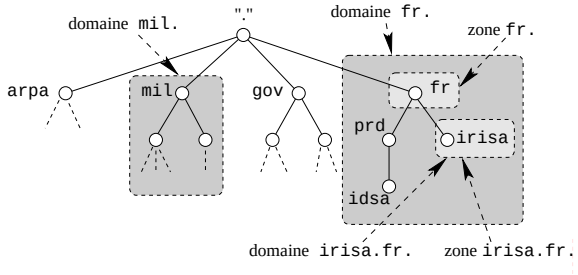


FIG. 1: Domaines et zones DNS.

Chaque domaine est constitué d'une ou plusieurs zones. La zone est l'unité administrative du DNS. Elle est représentée par un nœud dans l'arbre. Une zone est gérée par un ou plusieurs serveurs de noms qui conservent les informations de la zone dans un *fichier de zone*.

2.2 Les entités du DNS

Trois entités avec des rôles distincts interviennent dans l'architecture DNS : le serveur de noms autoritaires, le serveur cache récursif et le résolveur (voir [Moc87a, AL02]).

Le serveur de noms autoritaire. Un serveur de noms fait autorité sur une zone. Il conserve les enregistrements de ressource dans son fichier de zone. Chaque enregistrement est associé à un nom. Le serveur de noms reçoit des requêtes DNS portant sur un nom de domaine et répond avec les enregistrements contenus dans son fichier de zone.

Le serveur cache. Les serveurs caches ne font autorité sur aucune zone, ils ne possèdent pas de fichier de zone. Un serveur cache est généralement placé sur un réseau afin de recevoir les requêtes des résolveurs locaux. Le serveur cache a pour rôle de répondre à ces requêtes en utilisant les informations précédemment reçues qu'il conserve en mémoire durant un certain temps. S'il ne possède pas la réponse, le serveur cache fait suivre ces requêtes au serveur autoritaire qu'il pense le plus à même de posséder la réponse, met en cache la réponse reçue et la fait suivre au résolveur. L'utilisation de serveurs caches permet de diminuer la charge sur les serveurs autoritaires en mutualisant les informations [Sit00, JSBM01, CK01].

Le résolveur. Le résolveur est l'entité cliente qui est sollicitée par les applications et qui envoie la requête DNS appropriée à un serveur cache ou à un serveur de noms. Après avoir effectué la résolution de nom, le résolveur retourne la réponse à l'application.

2.3 La chaîne de confiance DNSSEC

Les extensions de sécurité DNSSEC utilisent la cryptographie à clé publique et définissent de nouveaux enregistrements de ressource pour conserver les clés et les signatures. Chaque zone ayant déployée DNSSEC possède une ou plusieurs clés de zone. La partie publique de chaque clé de zone est conservée dans un

enregistrement DNSKEY. La partie privée d'une clé de zone est gardée secrète et devrait être placée dans un endroit sûr. Cette partie privée est utilisée pour générer les signatures numériques de chaque enregistrement de ressource. Puis, ces signatures sont placées dans des enregistrements RRSIG. Pour faire confiance à un enregistrement, un résolveur doit vérifier au moins une signature de cet enregistrement et faire confiance à la clé qui a généré cette signature. Bien que chaque enregistrement soit signé par chaque clé, la vérification d'une seule de ses signatures permet de faire confiance à un enregistrement.

2.3.1 Les deux types de clés dans DNSSEC

Il y a deux manières pour un résolveur de faire confiance à une clé de zone. Soit cette clé est configurée dans le résolveur, il s'agit d'une *clé de confiance* (aussi appelé *point d'entrée sécurisé* [KSL04]). Soit le résolveur fait confiance à un enregistrement *Delegation Signer* (DS) [Gun03] qui authentifie cette clé. Un enregistrement DS est un enregistrement conservé dans la zone parente et qui identifie une clé d'une zone fille, l'enregistrement DS est signé par les clés de la zone parente. Un enregistrement DS contient un haché d'une clé de la zone fille et l'identifiant de cette clé.

Ainsi, un enregistrement DS crée un lien sécurisé entre une zone parente et une zone fille, mais crée aussi une dépendance entre deux enregistrements (DS et DNSKEY). En effet, lorsqu'une clé est renouvelée dans la zone fille, l'enregistrement DS correspondant doit être mis à jour dans la zone parente. Cela implique une communication entre la zone fille et la zone parente. Pour minimiser le trafic entre les deux zones et le travail de la zone parente dû à la mise à jour du DS, certaines clés n'ont pas d'enregistrement DS associé. Le modèle *Delegation Signer* introduit une distinction entre deux types de clés, les *Key Signing Key* (KSK) et les *Zone Signing Key* (ZSK). Une KSK possède un enregistrement DS associé dans la zone parente et signe uniquement les enregistrements DNSKEY. Une ZSK ne possède pas d'enregistrement DS associé et signe tous les enregistrements de ressource contenu dans le fichier zone.

Même si le modèle *Delegation Signer* identifie deux rôles distincts pour les clés, KSK et ZSK, il n'y a pas d'obligation pour l'administrateur d'une zone d'utiliser ces deux différents rôles. Il est possible que les petites zones utilisent uniquement une clé (ayant les deux rôles) tandis que les zones plus grandes utiliseront plusieurs clés (au moins une pour chaque rôle).

2.3.2 La vérification de signatures dans DNSSEC

Pour diminuer la taille du fichier de zone, les enregistrements associés au même nom et possédant le même type sont regroupés et signés ensemble. Par exemple, si une zone possède trois enregistrements DNSKEY, ces trois enregistrements sont regroupés en un DNSKEY RRset (ou keyset). Puis, les trois clés génèrent chacune une signature pour le DNSKEY RRset obtenu. Ainsi, nous obtenons trois signatures du DNSKEY RRset et la vérification d'une seule de ces trois signatures permet de faire confiance à tous le DNSKEY RRset et donc au trois clés qu'il contient.

Lors de la construction d'une chaîne de confiance, le résolveur part d'une de ses clés de confiance et suit le chemin jusqu'à la ressource demandée en vérifiant à chaque étape les signatures et les liens sécurisés représentés par les couples DS-DNSKEY. La figure 2 présente les différentes étapes suivies par un résolveur possédant la clé de la zone racine comme clé de confiance, lorsque celui-ci veut faire confiance aux enregistrements de la zone `irisa.fr.`.

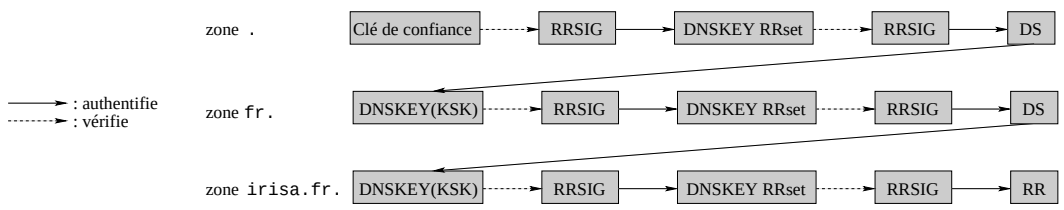


FIG. 2: Établissement d'une chaîne de confiance.

À chaque maillon de la chaîne, nous retrouvons le même schéma : un enregistrement DS permet d'avoir confiance en une KSK, une KSK permet d'avoir confiance en l'ensemble des clés de la zone (KSK et ZSK) et les ZSK permettent alors d'avoir confiance dans les autres enregistrements.

Après avoir présenté le fonctionnement des extensions de sécurité DNS, nous analysons dans la section suivante les conséquences de la présence d'une clé compromise dans DNSSEC, ainsi que les menaces induites par le modèle arborescent du DNS.

3 Conséquences d'une clé compromise dans DNSSEC

La révocation de clés est un service très coûteux dont au moins une des parties en cause dans le protocole, le client ou le serveur, doit s'acquitter. Dans certaines circonstances, il est préférable d'accepter les risques liés à l'absence de système de révocation de clés et de laisser les détenteurs des clés prendre les mesures nécessaires à leur protection. Il semble que c'est cette voie que les concepteurs de DNSSEC ont décidé de suivre [Cha03]. Aucun service de révocation spécifique n'est intégré à l'infrastructure DNSSEC.

Il existe plusieurs manières de compromettre une clé. La première est la cryptanalyse. Une personne malveillante obtient la partie privée d'une clé grâce à des connaissances mathématiques et du matériel cryptographique généré par la clé ou parfois en utilisant des failles dans les protocoles générant les clés ou les signatures numériques. Ce type d'attaques sur les clés est des plus difficiles, les algorithmes et protocoles cryptographiques étant conçus pour que la cryptanalyse soit impossible à réaliser sur une échelle de temps humaine. À titre indicatif, le temps de calcul théorique pour la cryptanalyse d'une clé RSA (couramment utilisée dans DNSSEC) est donné dans le tableau 1 [Odl95]. Nous pouvons aussi citer qu'une équipe de chercheurs allemands a cassé le RSA-576 en décembre 2003.

Taille de clé RSA en bits	MIPS.an
512	3.10^4
768	2.10^8
1024	3.10^{11}
1280	1.10^{14}
1536	3.10^{16}
2048	3.10^{20}

TAB. 1: Temps de cryptanalyse d'une clé RSA en MIPS.an selon sa taille.

Nous avons vu qu'il est conseillé de conserver les clés privées dans un endroit sûr (sur une machine ou un support déconnecté de tout réseau). La seconde manière de compromettre une clé est, si cette consigne n'est pas respectée, d'accéder à ces clés par le réseau et de les copier en déjouant toutes les sécurités mises en place. La troisième possibilité est lorsque l'attaquant a un accès physique aux clés privées : il pourrait s'agir par exemple d'un administrateur malhonnête ou mécontent.

À cause de la structure arborescente de DNSSEC, dès lors qu'une clé de zone est compromise, tout le sous-domaine de la zone est menacé. Une clé compromise permet à l'attaquant de créer de fausses délégations et aussi de faire passer de fausses réponses ou de faux enregistrements comme tout à fait licites et corrects cryptographiquement.

Dans la suite, nous supposons qu'il existe dans la zone étudiée les deux types de clés. Cette distinction permet de faciliter l'étude sans perte de généralité. Si une clé est utilisée comme KSK et comme ZSK, alors toutes les propriétés énoncées s'appliquent.

3.1 Une ZSK est compromise

Lorsqu'une ZSK est compromise, l'attaquant qui possède la partie privée de cette clé est capable de signer n'importe quel enregistrement qu'il aura préalablement créé. Nous pouvons noter que les signatures générées par l'attaquant peuvent être vérifiées grâce à la ZSK compromise. Cette ZSK est elle même authentifiée par les KSK en place dans la zone car il existe une chaîne de confiance menant à la ZSK compromise. L'attaquant n'a donc pas besoin de compromettre aussi une KSK de la zone (figure 3).

3.2 Une KSK est compromise

Lorsqu'une KSK est compromise, l'attaquant est capable de signer des enregistrements DNSKEY. Néanmoins, comme il n'est pas obligatoire de limiter le rôle des clés, l'attaquant peut utiliser cette KSK pour

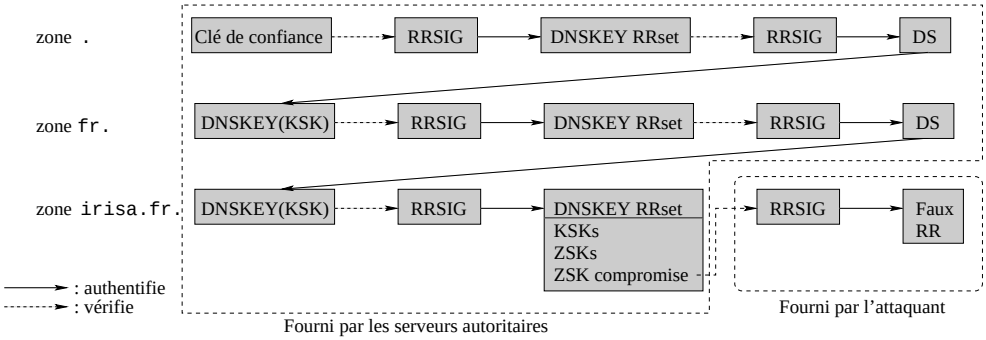


FIG. 3: Authentification d'un RR grâce à une ZSK compromise.

signer tous les types d'enregistrements. Cette KSK ayant un enregistrement DS associé il existe une chaîne de confiance l'authentifiant (figure 4).

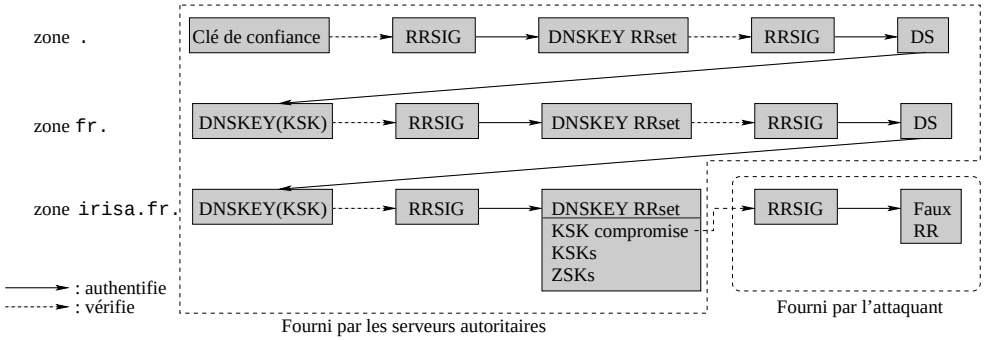


FIG. 4: Authentification d'un RR grâce à une KSK compromise.

3.3 Exemple d'attaque

Un attaquant ayant compromis une clé de zone n'a généralement pas accès en écriture sur le fichier de zone. L'attaquant ne pouvant pas atteindre la source des données elle même, il va cibler les serveurs caches pour tenter d'y placer de faux enregistrements. Ces fausses informations seront ainsi relayées jusqu'aux résolveurs par ces serveurs caches. Cette pollution de cache DNSSEC devient aussi facile qu'une pollution de cache DNS, car tout le matériel cryptographique fourni par l'attaquant est correct.

En effet, un serveur cache DNSSEC n'accepte des enregistrements que s'il peut construire une chaîne de confiance et en vérifier les signatures. Or, la chaîne de confiance existe (voir les figures 3 et 4 et pour les faux enregistrements qu'il a créés, l'attaquant peut en générer des signatures correctes grâce à la clé compromise.

Les messages DNS sont caractérisés par un identifiant sur 16 bits contenu dans l'en-tête du message. Cette identifiant est placé dans la requête puis repris dans la réponse, les résolveurs ayant souvent plusieurs requêtes en attente, cela permet de faire l'association question-réponse. La seule difficulté pour l'attaquant est donc de forcer le serveur cache à envoyer une requête déterminée et à lui fournir la réponse avec le bon identifiant. Pour forcer le serveur cache à envoyer la requête voulue, il suffit à l'attaquant de poser la question au serveur cache. Celui-ci, n'ayant pas la réponse, fera suivre au serveur de noms autoritaire. La récupération du bon identifiant, quant à elle n'est pas un obstacle majeur. Il existe différentes techniques [Gue05] qui facilitent cette récupération, comme notamment la possession d'un serveur de noms ou encore l'existence d'un paradoxe de l'anniversaire qui donne la probabilité de collision suivante (formule 1) :

$$\mathcal{P}(ID_x = ID_y) = 1 - (1 - \frac{1}{65535})^{\frac{n\times(n-1)}{2}} \quad (1)$$

Avec 302 messages, un attaquant a 50% de chance de trouver le bon identifiant. L'attaque est présentée sur la figure 5.

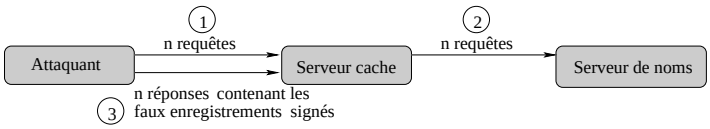


FIG. 5: Mise en œuvre du paradoxe de l'anniversaire.

Avec ce type d'attaque, l'attaquant peut placer dans les serveurs caches les informations qu'il désire concernant la zone compromise et par exemple détourner tout le trafic destiné à une zone donnée (site de consultation bancaire). L'attaquant pouvant renouveler périodiquement son attaque, tant que l'administrateur de la zone compromise n'a pas pris les mesures nécessaires, la zone est menacée et l'attaque reste efficace.

Dans la section suivante, nous présentons les moyens qu'un administrateur de zone peut mettre en œuvre pour répondre à la compromission d'une de ces clés.

4 Actions en cas clé compromise

Comme il n'existe pas de système de révocation dans DNSSEC, les moyens de défenses sont toujours réactifs. C'est-à-dire, qu'il faut que l'administrateur de la zone compromise s'aperçoive qu'un attaquant possède une de ses clés privées. Cela n'est pas forcément évident étant donné que les cibles des attaques sont surtout les caches voire les résolveurs et non pas les serveurs de noms gérés par l'administrateur concerné. De plus, les seules mesures que l'administrateur de la zone peut prendre concernent ses propres serveurs, pas les caches ou les clients pollués.

4.1 Actions possibles en cas de KSK compromise

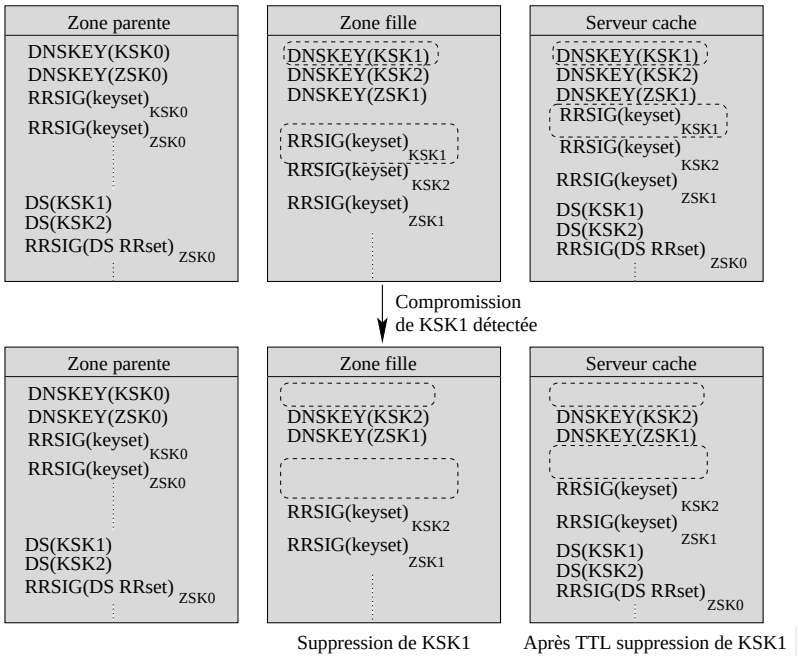
Lorsque l'administrateur d'une zone s'aperçoit de la compromission d'une de ses clés, il retire immédiatement la clé concernée de son fichier de zone et le re-signe (la signature du fichier est obligatoire car des changements sont intervenus, le DNSKEY RRset a été modifié). Néanmoins, cela n'est pas suffisant pour stopper immédiatement l'attaque à cause du lien, existant entre la zone compromise et sa zone parente, représenté par les couples (DS-DNSKEY) comme le montre la figure 6. Ce lien permet de construire une chaîne de confiance authentifiant la clé compromise.

La partie (a) de la figure 6 montre l'état du fichier de zone de la zone compromise, l'état du fichier de zone de la zone parente et l'état de la mémoire d'un serveur cache. Ce serveur cache n'est pas pollué. Pour protéger sa zone, l'administrateur a détruit la clé compromise KSK1 puis a re-signé son fichier de zone. Un résolveur envoyant une requête au serveur cache pour obtenir les enregistrements DNSKEY recevra un DNSKEY RRset contenant KSK2 et ZSK1 (les clés effectivement déployées dans la zone).

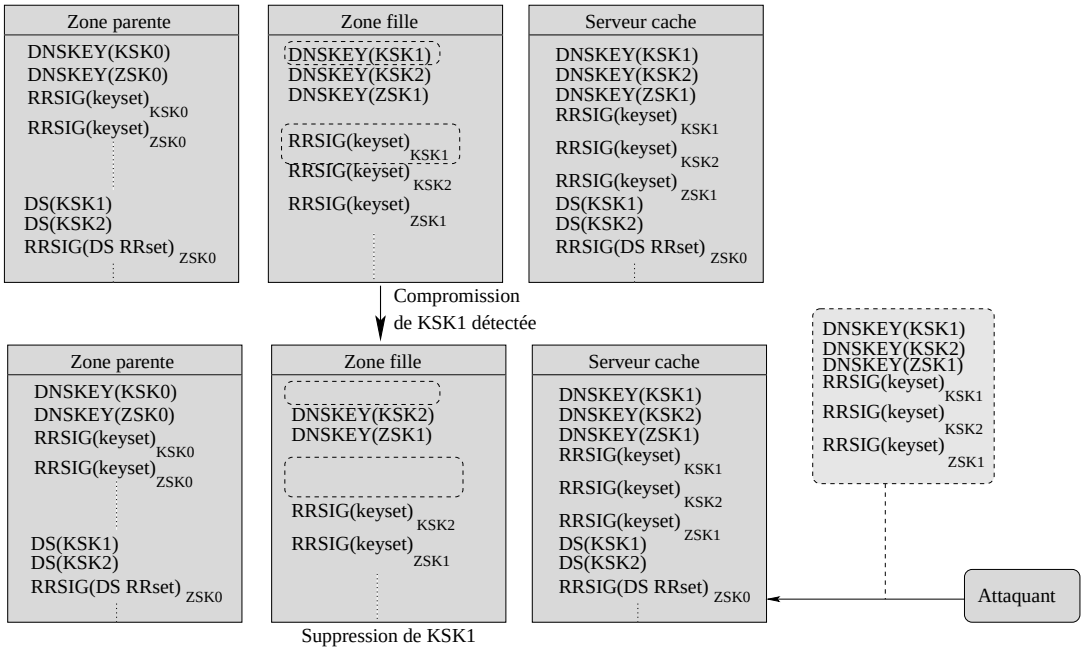
La partie (b) de la figure 6 montre l'état du serveur cache lorsque l'attaque de pollution menée par l'attaquant a réussi. Nous remarquons que même si la clé compromise est retirée du fichier de zone, cette clé est toujours en cache. Ainsi, même si tout est dans un état cohérent au niveau du fichier de zone, un résolveur envoyant une requête au serveur cache pour obtenir les enregistrements DNSKEY recevra la clé compromise KSK1 et les signatures que cette clé a générées. Ce résolveur est alors en mesure de construire une chaîne de confiance incluant DS(KSK1) et KSK1 qui authentifie la clé compromise et les faux enregistrements qu'elle a signés. Cela est possible même si la clé compromise ne se trouve plus dans le fichier de zone. Détruire la clé compromise n'est donc pas suffisant pour stopper l'attaque car il existe un enregistrement DS authentifiant la clé compromise et donc une chaîne de confiance.

Pour que la chaîne de confiance authentifiant la clé compromise n'existe plus, il faut détruire de la même manière l'enregistrement DS concerné dès que possible. La figure 7 montre toutefois que cela n'est pas suffisant pour stopper immédiatement l'attaque.

Lorsque l'attaquant effectue la pollution de cache, pour contourner la suppression de l'enregistrement DS, il ajoute cet enregistrement DS aux données qu'il tente de placer dans le serveur cache. Le rejeu de



(a) Sans pollution de cache



(b) Avec pollution de cache

FIG. 6: État des fichiers des zones parente et fille et de la mémoire d'un serveur cache (compromission d'une KSK).

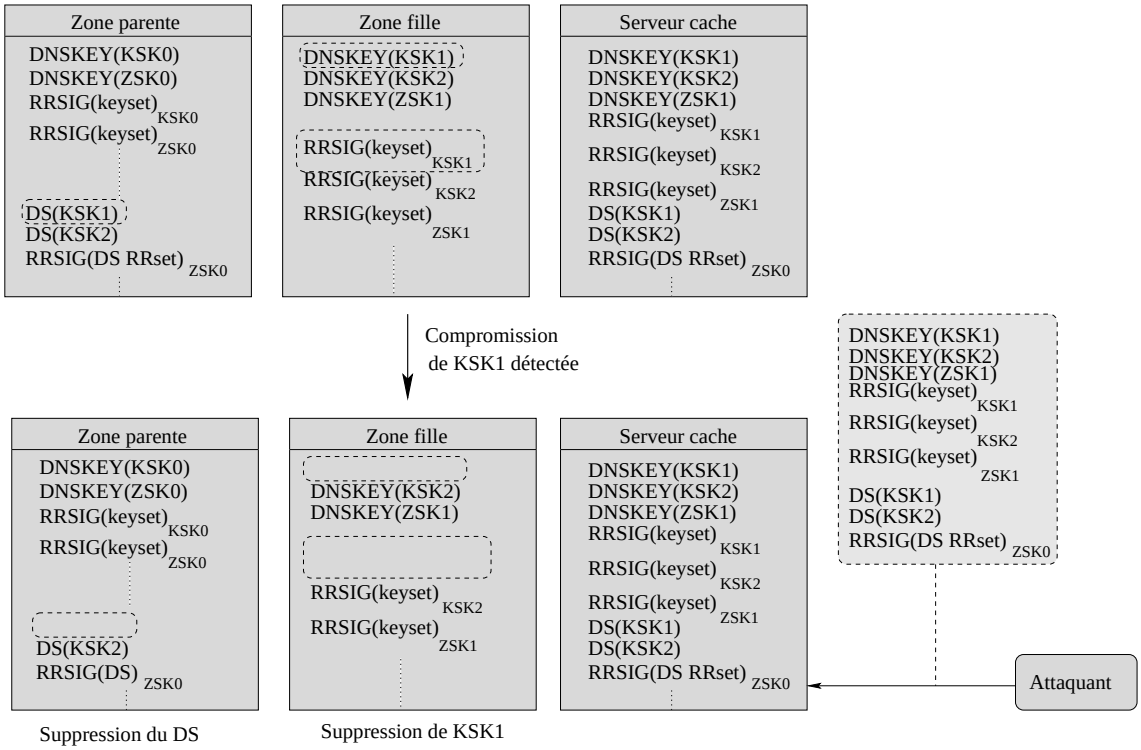


FIG. 7: Pollution d'un serveur cache lorsque clés et DS ont été détruits (compromission d'une KSK).

l'enregistrement DS fait qu'un résolveur interrogeant le serveur cache pollué est toujours en mesure de construire une chaîne de confiance authentifiant la clé compromise. Ce problème est dû au caractère statique de DNSSEC et à l'existence de multiples manières de constituer une chaîne de confiance, chaque couple DS-DNSKEY étant une possibilité. Donc, même si l'enregistrement DS correspondant à la clé compromise est enlevé du fichier de zone de la zone parente, la chaîne de confiance peut être construite avec les enregistrements DS restants. Ces enregistrements peuvent perdurer encore un certain temps dans le fichier de zone. L'attaque par pollution de cache est donc valide jusqu'à :

$$\max_{k\in K, p\in P}\left(expiration(RRSIG(DNSKEY\ RRset)_k), expiration(RRSIG(DS_c)_p)\right),$$

où K est l'ensemble des KSK non compromises de la zone, P est l'ensemble des ZSK de la zone parente et DS_c l'enregistrement DS référençant la clé compromise et expiration la date incluse dans le champ *expiration time* de l'enregistrement RRSIG.

Cette formule correspond à l'expiration de tous les couples DS-DNSKEY matérialisant une chaîne de confiance[†]. Le premier terme de la formule correspond à l'expiration des signatures des KSK non compromises de la zone. Après ce temps, une seule chaîne de confiance peut encore exister, celle représentée par la clé compromise et son enregistrement DS. Comme l'attaquant peut signer le DNSKEY RRset avec sa clé compromise, pour détruire cette dernière chaîne de confiance, il faut attendre l'expiration de la signature du DS associé. Le second terme de la formule correspond à la destruction de cette dernière chaîne de confiance. Une fois la signature de l'enregistrement DS associé à la clé compromise expirée, il n'existe plus de chaîne de confiance authentifiant la clé compromise : elle est alors inutilisable.

[†] Rappelons que les enregistrements sont signés en RRset, ici en DNSKEY RRset, donc une signature correcte authentifie toutes les clés.

4.2 Actions possibles en cas de ZSK compromise

Lorsqu'une ZSK est compromise, toutes les KSK de la zone valident une signature du DNSKEY RR compromis. Même si la clé compromise est enlevée du fichier de zone, un résolveur utilisant un serveur cache pollué (c'est-à-dire contenant l'ancien DNSKEY RRset avec la clé compromise) sera capable d'authentifier la clé compromise tant que les signatures générées par les KSK n'auront pas expiré. Nous pouvons noter qu'il n'est pas nécessaire d'ôter des fichiers de zones les KSK ou les enregistrements DS. Une fois que la ZSK est enlevée du fichier de zone, les signatures du DNSKEY RRset changent. Ainsi, l'attaque ne sera encore valide que jusqu'à l'expiration des vieilles signatures du DNSKEY RRset, c'est à dire jusqu'à :

$$\max_{k \in K} (expiration(RRSIG(DNSKEY\ RRset)_k))$$

où K est l'ensemble des KSK et expiration la date incluse dans le champ *expiration time* de l'enregistrement RRSIG.

Cette formule correspond à l'expiration des signatures des KSK et donc à la destruction de la chaîne de confiance authentifiant la ZSK compromise. L'attaquant ne pouvant pas générer une nouvelle signature du DNSKEY RRset avec une clé possédant un enregistrement DS associé, la clé compromise est inutilisable.

4.3 Discussion et travaux futurs

La durée de validité d'une signature est généralement de l'ordre de la semaine ou du mois selon le type de clé. La durée d'utilisation conseillée [KG06] est d'un an pour une KSK et d'un mois pour une ZSK, il apparaît que la fenêtre d'action (le temps d'expiration des signatures) dont dispose un attaquant ayant une clé compromise est assez grande.

Aux vues des possibilités offertes par ce type d'attaques, nous pensons que la gestion locale des clés compromises, comme cela est préconisé actuellement, est insuffisante et ne permet pas une défense efficace. Nous travaillons actuellement sur la définition d'un système de révocation pour DNSSEC. Plusieurs points peuvent être soulignés, tout d'abord ce système de révocation doit être, selon nous, intégré dans DNSSEC. L'ajout d'un mécanisme extérieur introduirait de nouveaux problèmes de dépendances, de gestion et de sécurité de ce système.

Ensuite, si ce système est intégré à DNSSEC et qu'il revêt la forme de nouveaux enregistrements, ceux-ci vont être signés. L'enregistrement de révocation, devant annoncer les clés révoquées et potentiellement compromises, sera signé (protégé) par ces mêmes clés. Un attaquant serait donc en mesure de créer une fausse liste de révocation et d'en générer une signature correcte avec la clé compromise. Il est donc clair que l'ajout d'un seul enregistrement de révocation dans une zone est insuffisant. Une coopération entre la zone parente et la zone fille doit être mise en place. Ce qui pose comme contrainte de définir un système minimisant le travail de la zone parente tout en garantissant la sécurité des échanges entre ces deux zones. De plus, si des informations nécessaires à la révocation sont localisées en différents endroits, dans la zone fille et dans la zone parente par exemple, le système doit garantir la récupération de toutes les informations nécessaires et cela dans tous les cas. Là encore, les serveurs caches peuvent poser problèmes.

5 Conclusions

Dans ce papier, nous avons présenté succinctement les extensions de sécurité DNS et leur fonctionnement. Nous avons décrit comment DNSSEC fondait sa confiance sur les signatures des enregistrements de ressource et l'utilisation de la cryptographie à clé publique. Nous avons aussi décrit ce qu'un attaquant pouvait faire avec une clé compromise en fonction du type de clé et nous avons décrit une des attaques possibles : l'attaque par pollution de cache. Dans la section 4, nous avons présenté les mesures que doit prendre un administrateur pour répondre à une compromission de clé. Néanmoins, nous avons montré qu'il n'existe pas de moyen immédiat de stopper les attaques par clés compromises même si celles-ci sont détectées car DNSSEC possède un caractère statique. Ce qui est vrai à un instant t le reste jusqu'à expiration des signatures et ce même si les enregistrements sont retirés des fichiers de zone. Il suffit donc, lors d'une attaque, de rejouer les anciens enregistrements tant que leurs signatures sont encore valides. Nous avons fourni, dans la section 4, les formules permettant de calculer la durée d'effectivité de ces attaques.

La compromission de clés étant critique pour DNSSEC, cela étant amplifié par la structure arborescente du DNS, nous sommes actuellement en cours de définition d'un système de révocation pour DNSSEC dont nous avons donné quelques éléments dans la section 4.3.

Références

[AA04] D. Atkins and R. Austein. Threat Analysis of the Domain Name System. RFC 3833, août 2004.

[AAL+05a] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, mars 2005.

[AAL+05b] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, mars 2005.

[AAL+05c] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, mars 2005.

[AL02] P. Albitz and C. Liu. *DNS and BIND*. O'Reilly & Associates, Inc., Sebastopol, Californie, 4th edition, janvier 2002.

[Bel95] S. M. Bellovin. Using the Domain Name System for System Break-Ins. In *Proceedings of the 5th Usenix UNIX Security Symposium*, pages 199–208, juin 1995.

[Cha03] B. Chan. Identity-Based PKI for DNSSEC. Thèse de master, Royal Holloway University of London, 2003.

[CK01] E. Cohen and H. Kaplan. Proactive Caching of DNS Records: Addressing a Performance Bottleneck. In *Symposium on Applications and the Internet*, pages 85–94, janvier 2001.

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

[GC03] G. Guette and B. Cousin. Les faiblesses du DNS. In *2ème rencontre francophone sur la Sécurité et Architecture Réseaux (SAR)*, pages 235–244, juillet 2003.

[Gie01] R. Gieben. Chain of trust: The parent-child and keyholder-keysigner relations and their communication in dnssec. Master's thesis, University of Nijmegen, 2001.

[Gue05] G. Guette. *Gestion de clés dans les extensions de sécurité DNS*. Thèse de doctorat, Université de Rennes 1, 2005.

[Gun03] O. Gundmundsson. Delegation Signer Resource Record. RFC 3658, décembre 2003.

[JSBM01] J. Jung, E. Sit, H. Balakrishan, and R. Morris. DNS Performance and the Effectiveness of Caching. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop '01*, pages 153–167, novembre 2001.

[KG06] O. Kolkman and R. Gieben. DNSSEC operational practices. Draft IETF, travail en cours, février 2006.

[KSL04] O. Kolkman, J. Schlyter, and E. Lewis. Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag. RFC 3757, avril 2004.

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

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

[Odl95] A. M. Odlyzko. The future of integer factorization. *CryptoBytes (The technical newsletter of RSA Laboratories)*, 1(2):5–12, 1995.

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

[Sit00] E. Sit. A Study of Caching in the Internet Domain Name System. Thèse de master, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Sciences, mai 2000.