SOMMES-NOUS PRÊTS À PASSER SOUS DNSSEC POUR UNE NAVIGATION PLUS SÉCURISÉE ?

Sandoche Balakrichenan – Afnic – sandoche.balakrichenan@afnic.fr

mots-clés : ?????????????????????

Depuis de nombreuses années, l'attestation de l'authenticité des noms de domaines était effectuée par les Autorités de certification (AC). Une solution alternative a été recherchée suite à plusieurs attaques mettant en exergue la vulnérabilité de l'infrastructure AC.

Le protocole DANE développé par l'IETF permet à un domaine d'attester lui-même les entités autorisées à le représenter, en utilisant un PKI alternatif – DNSSEC basée sur le DNS.

Le début de cet article présente le problème, puis introduit brièvement DNSSEC, explique comment DANE pourrait être implémenté, et enfin conclut sur les défis à relever pour passer du modèle d'authentification web actuel à DNSSEC.

1 Les événements qui ont conduit à repenser le mécanisme d'authentification sur Internet

On peut dire que c'est à partir de mars 2011 que les remparts de l'authentification sur Internet s'écroulaient les uns après les autres. Le 15 mars 2011, Comodo, un fournisseur leader sur le marché des certificats X.509 **[1] [2]** a découvert que l'un de ses affiliés avait été compromis par un attaquant ayant créé un compte utilisateur chez eux. Avec ce compte, l'attaquant a créé des demandes de certificats pour plusieurs sites web importants comme login.live.com, mail.google.com, login.yahoo.com, etc., et il est sûr qu'il a obtenu au moins un certificat X.509 pour ces sites.

Alors que l'on pensait **[3]** que l'attaque Comodo était un cas isolé dans la vie de l'industrie de l'Autorité de Certification (AC), quatre mois plus tard, une autre AC, DigiNotar, subit une attaque. L'attaquant qui avait agi en mars contre Comodo a revendiqué l'attaque sur DigiNotar. Même si rien ne prouve que les deux attaques viennent de la même personne, dans les deux cas, l'attaquant a réussi à trouver un chemin dans l'infrastructure de l'AC et à délivrer des certificats X.509 valides pour des domaines hors de leur contrôle.

Ces attaques très médiatisées ont suscité un certain nombre de questions. En particulier celle-ci : le modèle d'authentification existant fourni par l'infrastructure AC est-il vulnérable ? Et si oui, comment précisément atténuer ces vulnérabilités? Il y a un consensus assez large au sein des parties prenantes pour dire que quelque chose doit être fait, mais peu d'entente sur comment cela devrait être techniquement fait.

2 PKI X.509 (PKIX) – la PKI pour la navigation sécurisée

Pour initier une navigation web sécurisée, un utilisateur peut entrer l'*Uniform Resource Identifier* (URI) https://example.com dans un navigateur, dans laquelle

les caractères précédant le « :// » sont l'identifiant du protocole, et les caractères suivant la « :// » indiquent le nom de domaine du serveur. « *HyperText Transfer Protocol Secure* (HTTPS) », l'identifiant de protocole, indique que la communication entre l'utilisateur et le serveur web du domaine doit être effectuée en toute sécurité, via le protocole TLS.

En utilisant le nom de domaine, le navigateur commence par obtenir l'adresse IP du serveur web hébergeant le site en effectuant une résolution DNS du nom de domaine. Ensuite, il se connecte en *Transmission Control Protocol* (TCP) au serveur web du domaine et peut ainsi envoyer et recevoir des données.

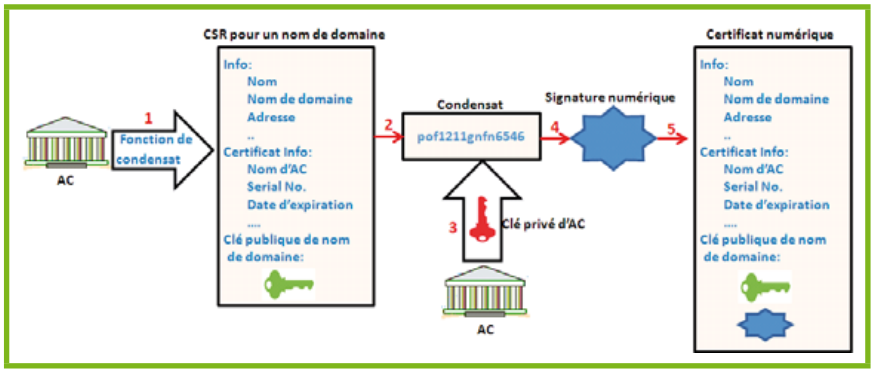

Fig. 1 : Processus de création d'un certificat numérique.

Afin d'établir une connexion TLS, le navigateur demande au serveur web d'envoyer sa clé publique. Le serveur web envoie cette clé dans un certificat X.509.

2.1 Processus de création d'un certificat X.509

Un certificat X.509 est l'attestation par une AC, d'une liaison entre une clé publique du site et son nom de domaine. L'AC fournit le certificat à un titulaire de domaine après la validation de ses références. Le certificat qui est créé par l'AC contient des informations telles que la clé publique du domaine, le nom de domaine, le nom de l'AC qui a créé le certificat, la période de validité du certificat, etc. Ensuite, l'AC crée un condensat (hash) du certificat et le chiffre avec sa propre clé privée. Le condensat chiffré est la « signature numérique ». Le certificat doté de cette signature est désormais appelé « Certificat numérique » (cf. figure 1).

Note

À partir d'ici, nous appelons le certificat X.509 « le certificat ».

2.2 Comment un navigateur vérifie le certificat

À des fins d'authentification, le navigateur doit vérifier que le certificat reçu a bien été envoyé par le serveur correspondant au nom de domaine. Afin d'en vérifier l'authenticité, tout d'abord, le navigateur vérifie si le certificat est fourni par une « AC reconnue ».

Les éditeurs de navigateurs autorisent une organisation d'être une AC reconnue, seulement si elle est jugée « digne de confiance », c'est-à-dire à la condition qu'elle suit un ensemble des principes et des procédures, et qu'elle fournit des certificats uniquement pour les titulaires corrects des domaines. Une fois que les éditeurs de navigateurs acceptent une organisation comme une AC reconnue, son certificat est ajouté à une liste (AC accréditées) dans son magasin.

Comme le certificat reçu (à partir d'un domaine) contient les informations de l'AC, le navigateur vérifie d'abord si l'AC est dans sa liste d'AC reconnues. Si tel est le cas, le navigateur vérifie la signature du certificat à l'aide de la clé publique (sauvegardée au préalable dans son magasin) de l'AC.

Cette clé publique est alors utilisée pour une communication chiffrée entre le navigateur et le serveur web en utilisant le protocole TLS.

2.3 Lacune dans le modèle PKIX

En bref, la taille de la liste des AC reconnues des navigateurs populaires tels que Chrome, Firefox, Internet Explorer, etc., varie, mais est de l'ordre de plusieurs centaines d'AC. Par exemple, un navigateur tel que Firefox fait confiance à 1482 certificats **[4]**. Ceci augmente la vulnérabilité du modèle.

Même si une seule AC, parmi toutes celles reconnues par le navigateur, est compromise, elle peut être utilisée pour générer un certificat pour tout nom de domaine qui pourrait être alors authentifié par un navigateur tel que Firefox. La faille dans le modèle PKIX actuel est que le propriétaire d'un domaine n'avait jusqu'à présent, aucun moyen de dire au navigateur, quelle AC ou quel certificat devait être utilisé pour vérifier l'accès à son serveur web.

Les attaques de Comodo et DigiNotar ont démontré comment un attaquant peut compromettre une AC vulnérable ou ses affiliés et créer des faux certificats pour des domaines tels que Google, Yahoo, Microsoft, etc.

3 Vers une PKI basée sur Domain Name Security Extensions (DNSSEC)

Le problème avec le modèle PKIX n'est pas la sécurité de la technologie PKIX, mais plutôt que la grande taille de la liste des AC reconnues par défaut par les navigateurs entraîne une grande probabilité de risques. Différentes techniques ont été proposées pour réduire la probabilité d'attaques en utilisant le modèle PKIX existant, mais il y a aussi des propositions pour utiliser un autre type de PKI. Une PKI alternative proposée pour réduire la probabilité d'attaques est également (comme le PKIX) définie par l'IETF – i.e. DNSSEC **[5]** basé sur DNS.

Même si cet article ne détaille pas les avantages et les inconvénients des différentes techniques proposées, la vue **[6]** parmi de nombreux experts en matière de sécurité est que la PKI basée sur DNSSEC est, à long terme la meilleure option pour sécuriser la navigation sur Internet.

DNSSEC désigne un ensemble défini d'extensions de sécurité du protocole DNS. Ces extensions utilisent les mécanismes de signature cryptographique asymétrique pour authentifier les enregistrements DNS. Les signatures et les clés publiques se présentent sous la forme de nouveaux enregistrements DNS qui permettent d'assurer l'authentification.

Avec DNSSEC, l'origine et l'intégrité des données reçues peuvent être vérifiées, en utilisant une ou plusieurs paires de clés associées a la zone DNS.

Les sous-sections suivantes commencent par expliquer comment DNSSEC protège la résolution DNS, et enfin, comment introduire le besoin du protocole « *DNS authentication of Named Entities* » (DANE) **[7]** qui, conjointement à DNSSEC pourrait être utilisé pour sécuriser les communications HTTPS.

3.1 Un exemple de comment une zone DNS est signée en utilisant DNSSEC

Un exemple fictif d'une zone DNS est comme suit :

```
; Fichier de zone pour  www.example.com
$TTL 1h              ; Time To Live
example.com. IN SOA ns.example.com. mail.example.com. (
             2013100304; Serial number
             3h          ; Refresh
             1h          ; Retry
             1h          ; expire
             1h          ; Negative cache )
example.com.  IN  NS     dns1.examplehost.com.
example.com.  IN  NS     dns2.examplehost.com.
example.com.  IN  A      192.0.2.1
```

Pour activer DNSSEC, l'administrateur de zone (comme example.com) crée une paire de clés publique et privée. La clé privée doit être stockée dans un endroit sûr et la clé publique est publiée dans la zone.

Dans DNSSEC, il est recommandé aux administrateurs de zone de créer deux paires de clés publiques et privées, où l'une est appelée « *Zone Signing Key* (ZSK) » et l'autre paire « *Key Signing Key* (KSK) ». Les deux clés publiques sont publiées dans la zone DNS du domaine comme enregistrement DNS (*Resource Record*) du type « DNSKEY ». Un exemple de DNSKEY dans une zone DNS non signée est :

```
; ZSK public key
example.com.  IN   DNSKEY 256 3 5
AwEAAda013Wp4CQaUBrExCIRZCYpT5K93FIPvOXfTkgsrgfsgfdgfdgfddfdfg

; KSK Public key
example.com.  IN   DNSKEY 257 3 5   A9Vze/B+hmwDJ+83cZ1JWW2G9geiboe
MrWA1SOWrDIdEWiEXCrqgHfHqfgljkgh
                                    a6/qcliyz2BwktpNwqorj62zT44iyEI
IfKQZLWYBj9BuspyIEXeoyrlBDWMmn+fv
```

Dans une zone DNS signée avec DNSSEC, chaque *Resource Record Set* (RRset) **[8]** (« ensemble d'enregistrements de données ») a un enregistrement DNS du type *RRset Signature* (RRSIG) **[9]**. Le RRSIG contient la signature du condensat du RRset.

La clé privée de la ZSK est utilisée pour produire le RRSIG pour tout le contenu de la zone DNS à l'exception des deux DNSKEY. Les DNSKEY sont signées par la KSK.

Il est important de noter que la zone DNSSEC signée contient non seulement le RRSIG, mais également les données non signées. L'exemple ci-dessous montre le contenu du type d'enregistrement « A », pour une zone « example.com » signée avec DNSSEC.

```
; l'enregistrement du type "A"
example.com.  1   A   192.0.2.1
; RRSIG de l'enregistrement "A"
example.com. 1 RRSIG A 5 5 1  (20130930064057 20130403064057 3960
example.com.
                                s8dMOWQjoTKEo1bsK+EYUY+32dsqdsfdsfdsf
sdfdssssdfsfsfdsssdfshqt0AaiD= )
```

La clé privée de la KSK est utilisée pour signer les deux DNSKEY (qui constituent une seule RRset) et le résultat sera un seul RRSIG comme suit :

```
example.com. 1   RRSIG   DNSKEY 5 4 1 (20130930064057
20130403064057 21001 example.com.

ufPzKty1MSGWtSRELZbHjL24RM5a5D43FSqK2FEhi/OUnWVwfeE+t615HosXv

+sAP9Ic8p93Pv6qkMVuD5ffL0wnXx3QFcGgoxCaEQ3w7VlibOddjhdUodM= )
```

3.2 Construction de la chaîne de confiance

La sous-section précédente expliquait comment les enregistrements d'une zone DNS sont signés à l'aide des extensions DNSSEC.

À ce stade, DNSSEC ajoute un niveau de sécurité dans lequel une réponse fausse à une requête DNS sera identifiée par un résolveur validant DNSSEC

Mais, si l'attaquant a réussi à envoyer une réponse fausse avec une fausse RRSIG, signée par sa propre paire de clés privée/publique générée, et également la fausse DNSKEY avant la réponse correcte, un résolveur validant ne sera pas en mesure de détecter que les données ont été altérées.

Dans le modèle PKIX, afin de s'opposer à une telle faille, le certificat est authentifié par un tiers de confiance, l'AC. Afin d'authentifier la clé publique dans DNSSEC, la validation d'un tiers se fait sur la base d'une « chaîne de confiance » cryptographique placée dans le DNS.

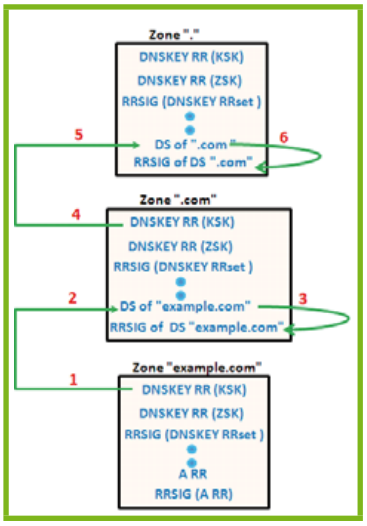

Fig. 2 : Un exemple de comment une chaîne de confiance est établit.

Comme indiqué (cf. figure 2), la chaîne de confiance est construite comme suit :

- De la zone « example.com », un enregistrement type « *Delegation Signer* (DS) » **[9]** est créé par l'administrateur de zone. L'enregistrement DS est le condensat du KSK DNSKEY.

  Cette 'DS' est publiée (l'administrateur utilise l'interface web de son bureau d'enregistrement afin de publier son DS. Le bureau d'enregistrement quant à lui, possède un canal sécurisé avec le *Top Level Domain* (TLD) '.com' pour publier les DS) dans la zone parente de « example.com », qui est « .com ».
- Le 'DS' doit être signé par la clé privée ZSK de la zone parent « .com ».

  De la même façon, le 'DS' pour la zone « .com » est publié par sa zone parente, qui est la racine DNS, et signé comme indiqué dans les étapes 4, 5 et 6 (cf. figure 2). Les opérations à chaque zone se font par l'administrateur de la zone concernée.

C'est ainsi que la chaîne de confiance est établie.

3.3 Vérification des données en utilisant la chaîne de confiance

Cette sous-section explique comment une réponse à requête DNS signée par DNSSEC est validée par un résolveur validant.

Note

Le résolveur validant est un résolveur DNS qui effectue la validation DNSSEC des données qu'il reçoit.

Afin de résoudre en toute sécurité dans le DNSSEC, la « clé publique » d'une zone DNS doit être configurée avec le résolveur validant (Fig. 3).

1. Le logiciel de résolution DNS intègre une « Trust anchor » qui correspond à la clé publique (KSK) de la racine du DNS. Pour l'authentification, le résolveur validant vérifie si le « Trust anchor » et le KSK dans la zone racine du DNS sont les mêmes.
2. Il prend ensuite la RRSIG de DNSKEY dans la zone de racine pour vérifier l'intégrité des clés (KSK et ZSK) présentes dans la racine.
3. En utilisant la clé ZSK vérifiée, l'intégrité des données dans la zone racine est vérifiée. Il convient de noter que la zone contient également l'enregistrement DS de `.com`.
4. Le DS du `.com` obtenu à partir de la racine est utilisé pour authentifier le KSK dans la zone '.com '.
5. Ensuite, une vérification des deux clés (KSK et ZSK) est effectuée comme au point (2).
6. La vérification de toutes les données dans la zone `. com` (qui comprend les DS « example.com ») se fait comme au point (3).
7. Les enregistrements DS de 'example.com' obtenus à partir du serveur du `.com` sont utilisés pour authentifier le KSK dans la zone « example.com ».
8. Ensuite, la vérification des deux clés (KSK et ZSK) est effectuée conformément au point (2).
9. La vérification de toutes les données dans la zone « example.com » se fait comme au point (3).

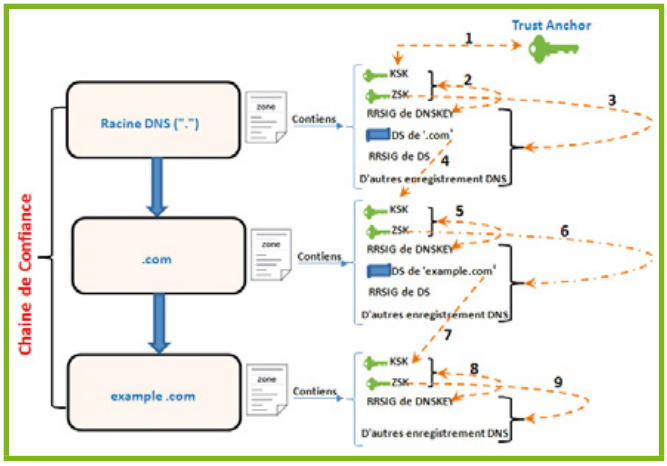

Fig. 3 : Un exemple de comment la vérification est faite en utilisant la chaîne de confiance établie.

L'avantage que présente la « chaîne de confiance » est que le résolveur validant n'a pas à faire confiance de manière explicite aux clés publiques de tous les noms de domaines. Il doit seulement faire confiance à la clé située en tête de la chaîne de confiance. Lorsqu'il doit valider une réponse DNS, tout ce qu'il doit faire est de survoler la chaîne de confiance depuis la racine jusqu'au sous-domaine concerné.

3.4 La nécessité pour le protocole DANE

Comme indiqué précédemment, il est important de rappeler que DNSSEC n'est pas une solution pour la confidentialité des données, mais une solution pour l'authentification de l'origine des données et assurer leur intégrité.

Mais, pour une navigation sécurisée, la confidentialité est obligatoire puisque les données telles que les coordonnées bancaires, mots de passe de connexion, etc. envoyées sur Internet sont destinées à n'être vues que par l'expéditeur et le récepteur. Dans le modèle PKIX, la confidentialité est fournie par TLS. Mais pour un échange de clé sécurisé dans TLS, il est nécessaire de vérifier le certificat qui contient la liaison de la clé publique du serveur web du domaine et de son identité.

Le problème de la PKI basée sur DNSSEC est : comment un navigateur sera-t-il capable d'authentifier le certificat reçu sans la disponibilité d'un tiers de confiance tel que l'AC ? Pour résoudre ce problème, DANE a été proposé par l'IETF.

4 DANE

DANE offre la possibilité de stocker les informations du certificat d'un domaine dans sa zone DNS. Pour cela, le protocole DANE introduit un nouvel enregistrement DNS, appelé TLSA.

Un enregistrement TLSA comprend quatre champs : **Certificate usage**, **Selector**, **Matching type** et **Certificate for Association** (cf. Figure 4).

1. **Certificate usage** : **0** ou **1** indique que le navigateur doit valider le certificat cible en utilisant l'infrastructure PKIX. Dans ce cas, DANE renforce le modèle PKIX existant (cf. la sous-section 4.1). Si les valeurs sont **2** ou **3**, alors la validation se fera uniquement en utilisant l'infrastructure DNSSEC (cf. la sous-section 4.2).
2. **Selector** : **0** indique que la valeur dans le champ **Certificate for association** est le Certificate complet et **1** indique, que ce champ contient uniquement la clé publique du certificat.
3. **Matching type** : **0** indique que la valeur dans le champ **Certificate for association** est le Certificate complet, la valeur **1** est le condensat SHA-256 et **2** est le condensat SHA-512.
4. **Certificate for Association** - La valeur du champ **Certificate for association** de TLSA pourrait être - le certificat complet du domaine, ou son condensat ou encore sa clé publique.

Fig. 4 : Différents champs de TLSA.

DANE permet de limiter la probabilité d'attaque (Section 2.3) parce que pour valider, une application DANE (e.g. Navigateur) cherche si le champ **Certificate for association** dans l'enregistrement TLSA correspond au certificat cible (i.e le certificat obtenu en se connectant au serveur web) basé sur d'autres champs dans l'enregistrement TLSA .

4.1 Comment DANE renforce le modèle PKIX existant?

Prenons l'exemple d'un domaine existant **https://fedoraproject.org**, ayant un certificat TLS valide signé par une AC. Sans DANE, la probabilité d'une attaque est plus grande, comme expliqué dans la sous-section 2.3. Pour limiter la surface d'attaque, l'administrateur de domaine a publié le TLSA original et le RRSIG de TLSA comme suit :

```
_443._tcp.fedoraproject.org.  IN   TLSA  0  0  1
D4C4C99819F3A5F2C6261C9444C62A8B263B39D 5037FB2B
                                                CCE35CDCABE272C6A


_443._tcp.fedoraproject.org.  IN RRSIG TLSA  5 4 300 20131207140552
20131107140552 7725
                                          fedoraproject.org pC7BPs1jp41G2x
VHziqsCwf7c1J15Kc+p
                                          VLVmS4Rc1DPAHIA+EdXdiISGLJIbgJ75L
oeZLO1QJIhw7PplaEa5Pow
                                          +nYj
m086dbjVzPIySnAPT1K2ebcA7N8NapiY19wa08oSirgqOZ8NDy
                                          6CVcofV/oAC+xMW/XbAwAkD+75 1+s=
```

Lorsque le navigateur valide un domaine en suivant le protocole DANE, il obtient tout d'abord le TLSA lors de la résolution DNS, puis il demande le certificat pour initier une connexion TLS. La valeur **0** dans le champ **Certificate Usage** de TLSA indique au navigateur qu'il doit effectuer la validation en utilisant l'infrastructure PKIX, et uniquement le certificat de l'AC identifié par le champ **Certificate for association** de TLSA. Aucun autre certificat AC pour le domaine ne sera validé par le navigateur, limitant ainsi la surface d'attaque.

4.2 Comment DANE peut être utilisé pour la navigation sécurisée utilisant uniquement DNSSEC?

Prenons l'exemple d'un autre domaine existant https://dane.rd.nic.fr. Le certificat créé pour ce domaine n'est pas signé par une AC, mais est auto-signé. L'enregistrement TLSA dans la zone du domaine est publié comme suit :

```
_443._tcp.dane.rd.nic.fr.   IN  TLSA  3   0   1
e781577c0eabb6701a0caf287e48ceebd3ba64b81792ef4970
                                                5f0f5e1070331b

_443._tcp.dane.rd.nic.fr.   IN  RRSIG   TLSA 5 6 1  20130930064057
20130403064057 3960
                                          dane.rd.nic.fr.
R4a3RdZ7kwXVjdzp/tpxHn1iBoE5DG/qP03rfW
                                          I6mZJlsnhXfKnMg2XDhl+Y
kaZ34nFrKrAoKKf41e5bFdcwr04vSspK
                                          8X+ebdb90l0U8rp6dA5ZaR
y6mybhi6HkE6t AXCKoApphjCeOB1c=
```

La valeur **3** dans le champ **Certificate Usage** indique au navigateur qu'il doit effectuer la validation uniquement en utilisant l'infrastructure DNSSEC. Basé sur les valeurs dans le champ **Selector** (**0**), et le **Matching type** (**1**) , le navigateur doit faire correspondre le champ **Certificate for association** avec le certificat. Le navigateur poursuit la procédure TLS seulement lorsqu'il existe une correspondance.

Ainsi DANE renforce non seulement la sécurité de la navigation en utilisant le modèle existant PKIX, mais fournit également une autre option en utilisant seulement l'infrastructure DNSSEC.

4.3 Mise en place DANE pour un nom de domaine

Pour un administrateur de domaine, la première étape, lors d'une mise en œuvre de DANE pour son nom de domaine consiste à créer le TLSA. Pour créer ce TLSA, le serveur Web du domaine doit soit avoir un certificat fourni par une AC, soit un certificat auto-signé.

Des outils tels que SWEDE **[10]** peuvent être utilisés pour créer le TLSA en fonction des différentes options. L'exemple ci-dessous montre comment le TLSA pour un certificat auto-signé est créé en utilisant SWEDE:

```
./swede create --usage 3 --output rfc dane.rd.nic.fr
The result is as follows:
Attempting to get certificate from 192.134.7.155
Got a certificate with Subject: /C=FR/ST=SQY/L=Montigny/O=EXAMPLE/
OU=ReD/CN=dane.rd.nic.fr.
_443._tcp.dane.rd.nic.fr.  IN  TLSA  3  0  1
e781577c0eabb6701a0caf287e48ceebd3ba64b81792ef49705f0f5e1070331b
```

SWEDE se connecte au nom de domaine spécifié et vérifie qu'il s'agit bien d'un domaine valide. Puis, il récupère le certificat TLS. Selon les paramètres fournis (par exemple la valeur du **Certificate Usage**, **3** dans l'exemple ci-dessus), il crée le TLSA. Le TLSA obtenu est publié dans la zone DNS du domaine, et la zone est ensuite signée avec DNSSEC par l'administrateur de domaine.

4.4 La nécessité d'une application client pour valider selon DANE

Dans le modèle PKIX, le navigateur possède une liste d'AC de confiance préinstallée dans son magasin afin de valider TLS. Mais, à ce jour aucun navigateur n'est capable d'effectuer la validation DNSSEC et DANE nativement.

Note

Dans cet article, le navigateur a été considéré comme une application client , mais pour DANE, une application client ne se limite pas seulement à un navigateur

Pour une preuve de concept pour DANE (cf. figure 5 page suivante), nous avons utilisé de plugin « Extended DNSSEC validator » **[11]** pour Firefox. Ce plugin vérifie le TLSA dans la zone DNS, et le valide avec le certificat. Le plugin est livré avec le résolveur unbound **[12]** validant DNSSEC. En ayant un résolveur intégré avec le navigateur, l'utilisateur souhaitant mettre en œuvre DANE a l'avantage de ne pas à avoir à s'inquiéter de l'activation DNSSEC sur son poste ou auprès de son FAI.

Note

Pour tester DANE à ce stade (sans la disponibilité de DANE intégré dans le navigateur), assurez-vous que le plugin (Extended DNSSEC Validator) est installé. Le plugin fonctionne actuellement que dans Debian et uniquement sur les versions Firefox antérieures à 20.0

5 Défis en passant d'un modèle PKIX à une PKI basée sur DNSSEC pour HTTPS

Le manque de déploiement DNSSEC constitue le défi le plus important pour une utilisation de DANE de façon transparente pour une navigation sécurisée.

Du côté des serveurs, le problème semble se résoudre, vu que les TLDs de grande envergure et la racine DNS sont signés avec DNSSEC.

Comme mentionné précédemment, il y a une nécessité de validation de DNSSEC et de TLSA du côté client, or

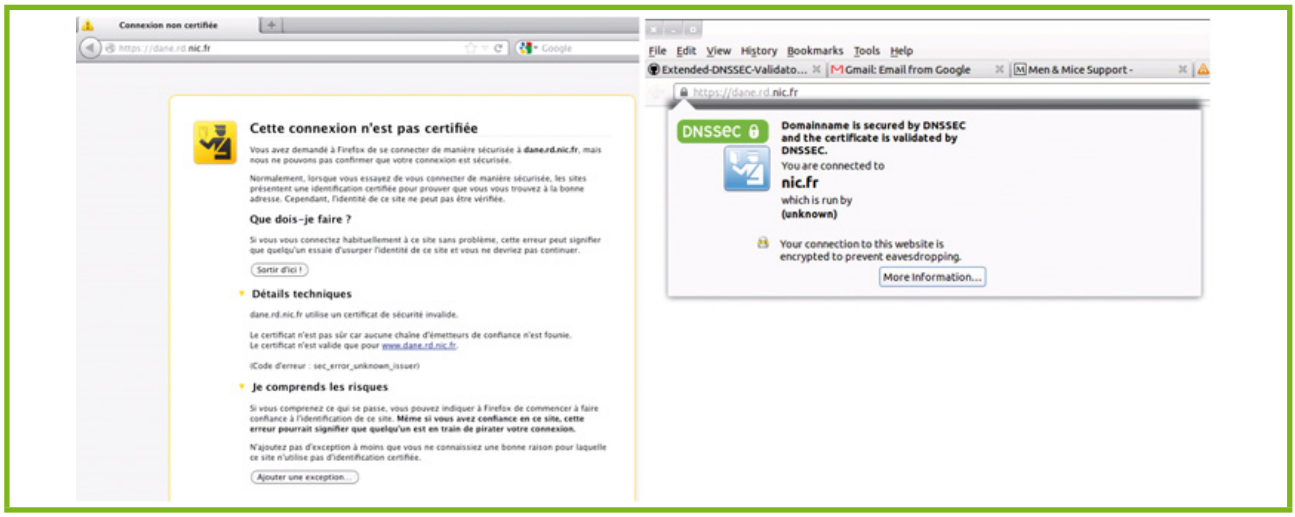

Fig. 5 : (Gauche) Capture d'écran du nom de domaine 'https://dane.rd.nic.fr' sans le plugin « Extended DNSSEC Validator » installé. (Droite) Capture d'écran du même nom de domaine avec le plugin installé.

il n'existe aucun navigateur pour le moment capable de supporter DNSSEC et DANE nativement.

Des plugins comme « Extended DNSSEC validator » sont une option, mais cela nécessite de la part de l'utilisateur une installation afin d'accéder aux domaines utilisant DANE. C'est un inconvénient, les administrateurs de domaine n'activeront pas DANE avec DNSSEC PKI sans être assurés que tous les utilisateurs accéderont à leur domaine sans problèmes.

D'un point de vue de performance, ajouter DNSSEC au processus de connexion TLS génère des ralentissements significatifs. En plus de l'utilisation des protocoles classiques de TLS et la validation de certificats, le client doit attendre plusieurs allers-retours de résolutions DNS pour enfin valider la chaîne de signatures DNSSEC. À tous ces retards combinés peuvent venir s'ajouter quelques secondes de latence de l'établissement de la connexion, ce qui entraîne une mauvaise expérience utilisateur.

D'un point de vue sécurité, DNSSEC est une PKI qui suit le modèle « top-down » où, si la racine du DNS est compromise, l'ensemble de la communication Internet pourrait être compromise. La racine semble être fortement protégée et il n'y a pas eu d'incidents jusqu'à présent. Un scénario possible est celui dans lequel l'un des TLD est compromis. Dans cette situation, tous les domaines sous cette TLD seraient compromis.

Conclusion

Jusqu'à présent, les navigateurs web se sont basés sur la certification des AC pour s'assurer de la validité des serveurs web avec un nom de domaine. La promesse de DANE, sécurisé par DNSSEC est une interaction plus directe (sans l'intervention des AC) entre les clients et les domaines avec lesquels ils échangent. À court terme, DANE peut être déployé pour renforcer la PKIX existante. À long terme, DANE avec un DNSSEC entièrement déployé ainsi que les défis de la transition relevés permettra aux opérateurs de domaine (avec des certificats auto-signés plutôt que de payer un CA) de se porter garants de leur propre nom de domaine. ■

RÉFÉRENCES

[1] Peter Saint-Andre and Jeff Hodges, « Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS) », RFC 6125, March 2011.

[2] David Cooper, Stefan Santesson, Stephen Farrell, Sharon Boeyen, Russell Housley, and Tim Polk, « Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile », RFC 5280, May 2008.

[3] http://news.cnet.com/8301-27080_3-20048831-245.html

[4] https://www.eff..org/observatory

[5] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose, « DNS Security Introduction and Requirements », RFC 4033, March 2005.

[6] Ivan Ristic and Wolfgang Kandek, « SSL and Browsers : The Pillars of Broken Security » RSA Conference 2012

[7] P. Hoffman, J. Schlyter, « The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol : TLSA », RFC 6698, August 2012

[8] http://www.bortzmeyer.org/2181.html

[9] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose, « Resource Records for DNS Security extensions », RFC 4034, March 2005.

[10] SWEDE - https://github.com/pieterlexis/swede

[11] http://people.redhat.com/pwouters/ - [fichier mozilla-extval-0.7.xpi]

[12] http://www.bortzmeyer.org/unbound.html