RFC 6394 : Use Cases and Requirements for DNS-based Authentication of Named Entities (DANE)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 octobre 2011

Date de publication du RFC : Octobre 2011

http://www.bortzmeyer.org/6394.html

Le projet DANE ("*DNS-based Authentication of Named Entities*", anciennement KIDNS - "*Keys In DNS*"), vise à améliorer et/ou remplacer les certificats X.509 utilisés dans des protocoles comme TLS (RFC 5246 [1]). Ces certificats souffrent de plusieurs problèmes, le principal étant qu'une Autorité de Certification (AC) malhonnête ou piratée peut créer des certificats pour **n'importe quelle entité**, même si celle-ci n'est pas cliente de l'AC tricheuse. C'est ce qui s'est produit dans les affaires Comodo et DigiNotar et ces deux affaires ont sérieusement poussé aux progrès de DANE. Ce premier RFC du groupe de travail DANE <http://tools.ietf.org/wg/dane> établit les scénarios d'usage : quels problèmes veut résoudre DANE et selon quels principes ?

Revenons d'abord sur le fond du problème (section 1 du RFC) : lors de l'établissement d'une session TLS (par exemple en HTTPS), le client se connecte à un serveur dont il connait le nom (par exemple impots.gouv.fr, cf. RFC 6125), le serveur (et, plus rarement, le client) présente un certificat numérique à la norme X.509 (bien qu'un certificat PGP soit aussi possible, cf. RFC 6091). Ce certificat suit le profil PKIX, normalisé dans le RFC 5280, un sous-ensemble de X.509. Comme ce certificat peut avoir été présenté par un homme du milieu qui a intercepté la session (par exemple en jouant avec BGP ou OSPF), il faut l'authentifier, sinon TLS ne protégerait plus que contre les attaques passives, laissant ses utilisateurs à la merci des attaques actives. Cette authentification se fait typiquement aujourd'hui en vérifiant que le certificat a été signé par une des plusieurs centaines d'AC qui se trouvent dans le magasin de certificats de n'importe quel navigateur Web. Le point important, et qui est l'une des deux principales faiblesses de X.509 (l'autre étant son extrême complexité), est que n'importe laquelle de ces centaines d'AC peut signer un certificat. Ainsi, dans l'affaire DigiNotar, une AC néerlandaise était piratée, le pirate lui faisait générer des certificats pour *.google.com ou *.gmail.com (alors que Google n'est pas du tout client de DigiNotar) et ces certificats sont acceptés par tous les navigateurs. Le pirate a vendu ou donné les clés privées de ces certificats au gouvernement iranien, qui n'a plus eu qu'à monter

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5246.txt

une attaque de l'homme du milieu (facile lorsqu'on est un état policier et qu'on contrôle tous les FAI du pays) pour détourner les utilisateurs vers un faux Gmail, qui présentait un certificat acceptable. La dictature intégriste pouvait alors lire les messages et emprisonner ou tuer les opposants. (Le fait que l'attaque venait d'Iran est établi par les requêtes OCSP des victimes. Cf. le rapport sur l'opération Tulipe Noire <http://www.rijksoverheid.nl/ministeries/bzk/documenten-en-publicaties/rapporten/2011/09/05/diginotar-public-report-version-1.html>.)

Identifiée depuis longtemps (mais niée par le lobby des AC ou par l'UIT, à l'origine de la norme X.509), cette vulnérabilité a mené au projet DANE. L'idée de base est qu'introduire un intermédiaire (l'AC) est en général une mauvaise idée pour la sécurité. Pour reprendre l'exemple ci-dessus, il vaudrait mieux permettre à Google de prouver sa propre identité. Comme quasiment toutes les transactions sur l'Internet commençent par une requête DNS, pourquoi ne pas utiliser la zone DNS `google.com` pour y mettre le certificat ? L'idée est ancienne, mais elle ne suscitait guère d'intérêt tant que le DNS lui-même n'était pas un tant soit peu sécurisé. Le déploiement rapide de DNSSEC, à partir de 2009, a changé la donne : mettre des clés cryptographiques ou des certificats dans le DNS devient réaliste. D'où la création, en février 2011, du groupe de travail DANE <http://tools.ietf.org/wg/dane> de l'IETF, chargé de trouver une technique de sécurité qui marche.

Ce RFC ne donne pas encore de solution (au moment de sa parution, celle-ci est très avancée et a des chances sérieuses d'être publiée en 2012). Il explique les scénarios d'usage, pour aider à évaluer la future solution. Pour décrire ces scénarios, on recourt évidemment aux personae traditionnels :

— Alice gère le serveur `alice.example.com`,
— Bob, le client, se connecte à ce serveur,
— Charlie est l'AC,
— Trent est un émetteur de certificats mais qui n'est pas dans les magasins de certificats d'AC typiques.

Trois scénarios sont ensuite décrits dans la section 3, le œur de ce RFC :

— « Contrainte sur l'AC » (dit aussi « type 0 »),
— « Contrainte sur les certificats » (« type 1 »),
— « Certificats locaux » (« type 2 »),

Dans les deux premiers cas, DANE ne fait qu'ajouter de la sécurité à X.509 et on a toujours besoin de l'IGC actuelle. Le troisième est le plus novateur, permettant de remplacer complètement l'IGC, et ne gardant de X.509 que son format de certificats. (Notez que ces trois scénarios ne se retrouvent pas exactement dans le RFC 6698 sur le protocole qui a quatre types et non pas trois.)

Voyons d'abord le premier scénario (section 3.1) , les contraintes sur l'AC. Dans un cas comme celui des piratages de Comodo ou de DigiNotar, Alice, qui est cliente d'une autre AC, Charlie, est inquiète du risque d'un faux certificat pour `alice.example.com`, émis par l'AC voyoute. Elle voudrait dire aux visiteurs de `https://alice.example.com/` que les seuls certificats légitimes sont ceux émis par Charlie. En d'autres termes, elle voudrait que la vérification du certificat de son serveur parte du certificat de Charlie, pas de n'importe quel certificat d'AC du magasin.

Notez que, si Alice dispose d'un tel système, elle peut aussi s'en servir vis-à-vis de Charlie. Lorsqu'elle demande un certificat à ce dernier, il pourrait vérifier qu'Alice l'a bien listé comme AC.

Donc, Alice va mettre dans le DNS un enregistrement qui dit « je suis cliente de Charlie ». Cet enregistrement doit-il être sécurisé par DNSSEC ? La question est toujours très discutée au sein du groupe DANE. A priori, DNSSEC n'est pas impératif puisque cet enregistrement DNS n'est qu'une sécurité supplémentaire : le X.509 traditionnel fonctionne toujours en dessous. D'un autre côté, cette sécurité supplémentaire est très faible si on ne met pas DNSSEC : un attaquant qui peut monter une attaque de l'Homme du Milieu pour détourner vers son propre serveur peut probablement aussi supprimer les enregistrements DANE au passage. Notons aussi que le fait d'exiger la validation X.50 comme avant

limite les possibilités d'attaque par un registre DNS. Avec le type 0 (contrainte sur l'AC), un attaquant devrait contrôler à la fois l'enregistrement DNS et l'Autorité de Cértification.

Second scénario, la contrainte sur le certificat (section 3.2), alias type 1. Alice est cliente de Charlie mais se demande si Charlie ne va pas émettre d'autres certificats pour `alice.example.com`, par exemple parce que les machines de Charlie ont été piratées. Elle voudrait donc indiquer aux visiteurs de `https://alice.example.com/>` que seul tel certificat est valable. Alice va donc mettre le certificat (ou un condensat cryptographique de celui-ci) dans le DNS. Comme avec le type 0, toute la validation classique de X.509 est ensuite appliquée. Dans ce mode, DANE ne fait qu'ajouter une vérification supplémentaire.

Bien plus « disruptive » est la possibilité de se passer complètement des AC et de dire dans le DNS « voici le certificat à utiliser », sans qu'il doit validé par d'autres mécanismes que DNSSEC. Cette possibilité (section 3.3), dite type 2, est l'équivalent des certificats auto-signés actuels, mais avec possibilité de les vérifier (grâce à DNSSEC). Alice se passe alors complètement de Charlie. (Notez que le cas où Alice est cliente de Trent est équivalent, le certificat de Trent n'étant quasiment dans aucun magasin. Alice va alors publier dans le DNS le certificat de Trent et non le sien.)

Notez que ce mécanisme n'empêche pas Bob d'avoir sa propre politique de validation. On pourrait imaginer un Bob suspicieux vis-à-vis de DNSSEC qui n'accepterait que les enregistrements DANE de type 0 ou 1. Ou un Bob qui aurait une liste noire des certificats à qui il ne fait pas confiance, DANE ou pas DANE.

Cette fois, avec le type 2, DNSSEC est absolument indispensable. Sans lui, il n'y a plus aucune validation du certificat. Ne publiez donc pas d'enregistrements de type 2 avant d'avoir vérifié que DNSSEC était correctement déployé et que vous le maîtrisez parfaitement. Pour la même raison, le type 2 est celui qui met le plus de responsabilités chez les opérateurs DNS. Si ceux-ci trahissent ou sont piratés, tout le système est en danger. Mais rappelez-vous que, contrairement à ce qui se passe pour X.509, le risque est uniquement dans les prestataires que vous choisissez. Si vous choisissez d'utiliser un nom de domaine en `.org`, vous savez qu'Afilias peut tout faire rater, mais que ni l'AFNIC, ni CNNIC n'ont ce pouvoir. Alors qu'avec X.509, même en choisissant soigneusement son AC, on peut être victime de n'importe quelle autre AC, qu'on n'a pas choisi. Donc, conseil pratique, choisissez bien vos prestataires (registre DNS, bureau d'enregistrement, hébergeur DNS, etc).

Mais il est clair que DANE donne plus de responsabilité (on peut même dire de pouvoir) aux acteurs du DNS. Ce point est souvent à la base de la plupart des critiques de DANE `<http://blog.thoughtcrime.org/ssl-and-the-future-of-authenticity>`.

Autre point à garder en tête lorsqu'on évalue la sécurité de DANE : aujourd'hui, avec X.509, le contrôle de la zone DNS donne déjà souvent accès à des certificats (bien des AC ne font pas d'autres vérifications que l'envoi d'un courrier avec un "*cookie*", ou la publication d'un certain enregistrement).

Voilà, vous connaissez l'essentiel sur DANE. la section 4 couvre toutefois d'autres points, moins centraux, sous forme d'une liste de souhaits pour le futur protocole :

- Capacité à gérer des certificats différents pour une même machine, pour des services tournant sur des ports différents (par exemple un serveur HTTP et un serveur IMAP).
- Pas de possibilité d'attaque par repli.
- Possibilité de combiner plusieurs assertions (par exemple de type 1 et 2, la première servant pour les clients qui refusent de croire au type 2 et qui tiennent à faire une validation X.509 classique).
- Le moins de dépendances possibles, à part bien sûr celle sur DNSSEC.
- Le moins d'options possibles, pour faciliter l'analyse de sécurité (IPsec avait été beaucoup critiqué pour le trop grand nombre de choix qu'il offrait).

Le protocole qui met en œuvre ce cahier des charges a été normalisé en août 2012 dans le RFC 6698.

Pour ceux qui souhaitent approfondir le sujet, j'ai fait un exposé sur la question (avec article détaillé `<http://www.bortzmeyer.org/jres-dane-2011.html>`) en novembre 2011 aux JRES à Toulouse.