RFC 7960 : Interoperability Issues between Domain-based Message Authentication, Reporting, and Conformance (DMARC) and Indirect Email Flows

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 11 octobre 2016

Date de publication du RFC : Septembre 2016

https://www.bortzmeyer.org/7960.html

Le mécanisme DMARC permet d'indiquer dans le DNS la politique d'un domaine concernant l'authentification du courrier. Si je reçois un message prétendant venir de ma-banque.example, et qu'il n'est pas authentifié (ni SPF, ni DKIM, ni autre chose), comment savoir si c'est parce que ma banque est nulle en sécurité du courrier, ou bien parce que le message est un faux ? DMARC (normalisé dans le RFC 7489 [1]) permet de répondre à cette question en publiant un enregistrement qui indique si le courrier est censé être authentifié ou pas. Comme toutes les techniques de sécurité, ce mécanisme est imparfait et il pose notamment des problèmes avec les messages **indirects**. Par exemple, si vous avez une adresse à votre ancienne université, alice@univ.example et que le courrier qui lui est adressé est automatiquement transmis à votre adresse professionnelle, alice@evilcorp.example, comment DMARC va-t-il réagir avec cette indirection ? C'est ce qu'explore ce RFC.

Voici la politique DMARC de Gmail. Elle est tolérante (p=none, accepter les messages non authentifiés) :

```
% dig TXT _dmarc.gmail.com
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59294
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
_dmarc.gmail.com. 600 IN TXT "v=DMARC1; p=none; rua=mailto:mailauth-reports@google.com"
...
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7489.txt

Paypal est plus violent, demandant qu'un récepteur rejette les messages prétendant venir de Paypal mais pas authentifiés :

```
_dmarc.paypal.com. 300 IN TXT "v=DMARC1; p=reject; rua=mailto:d@rua.agari.com; ruf=mailto:dk@bounce.paypal.c
```

La question de départ de l'administrateur système est « si je mets une politique DMARC restrictive (genre `p=reject`), vais-je perdre des courriers légitimes, à cause d'indirections comme les listes de diffusion ? » (section 1 du RFC). Il est d'autant plus difficile de répondre à cette question que l'architecture du courrier électronique est complexe et mal spécifiée (RFC 5598). Bien des logiciels ne suivent pas les règles, d'autant plus que beaucoup de ces règles n'ont pas été explicites dès le début. (Un bon exemple : avec un `.forward`, le serveur de courrier doit-il garder l'expéditeur indiqué dans l'**enveloppe** du message ? Essayez de trouver un RFC qui spécifie cela !)

La section 2 de notre RFC décrit les causes des problèmes DMARC. Si un message est légitime (le destinataire veut le recevoir), et qu'il est en outre techniquement correct, un problème, au sens de ce RFC, est quand la politique de DMARC (qui peut être de rejet) est appliquée à ce message, parce qu'il a été transmis indirectement. C'est injuste. Évidemment, si la politique DMARC est `p=none` (ne rien faire), ce n'est pas un vrai problème. Mais un `p=reject` peut être très ennuyeux.

Première cause de problèmes, des différences entre les identificateurs utilisés. DMARC n'authentifie pas directement, il dépend de SPF (RFC 7208) et DKIM (RFC 6376) pour cela. Ce qui intéresse l'utilisateur, évidemment, c'est le nom dans le champ `From:` (RFC 5322) du message. Pour lui, c'est ça l'expéditeur. Mais DKIM, et surtout SPF, n'ont pas la même conception : ils peuvent utiliser d'autres identificateurs. DMARC considère que l'accord ("*alignment*", cf. RFC 7489, section 3.1) entre les identificateurs authentifiés par SPF et DKIM, et le champ `From:` du message peut être strict ou laxiste. « Strict » indique une correspondance parfaite, laxiste se limite à vérifier que le nom de domaine est le même.

Le principal identificateur utilisé par SPF est celui donné par la commande `MAIL FROM` dans la session SMTP (RFC 5321). C'est la principale cause de désaccord : si SPF authentifie cet identificateur et qu'il est différent de l'adresse utilisée dans le champ `From:` de l'en-tête du message, que faire ?

Deuxième cause de problème, le relayage d'un message. Si Bob `bob@isp.example` écrit à `alice@univ.example` et qu'elle (ou son administrateur système) a demandé un relayage automatique vers `alice@evilcorp.example`, les serveurs de courrier de `evilcorp.example` verront un message prétendant être de `isp.example`, mais transmis par les serveurs de `univ.example`... SPF sera content ou pas, selon la façon exacte dont a été fait le relayage (en préservant le `MAIL FROM` ou pas). DMARC ne sera jamais content car, si le `MAIL FROM` a été changé (reflétant le relais), SPF authentifiera mais il n'y aura plus d'accord entre les identificateurs.

Évidemment, si le message est modifié en cours de route, c'est encore pire. SPF ne protège pas l'intégrité du message, mais DKIM le fait. Mais qui diantre se permet de modifier les messages ? Hélas, pas mal de gestionnaires de listes de diffusion le font. DKIM a une option (déconseillée... voir la section 8.2 du RFC 6376) pour ne signer que le début du message, évitant ainsi que la signature soit invalidée par l'ajout, par exemple, d'un paragraphe final. Cela ne couvre que le cas des messages simples, sans MIME, où la modification est un simple ajout à la fin. Autre possibilité de DKIM pour éviter d'invalider les signatures en cas de modification du message, le mode "*relaxed*" de canonicalisation du contenu, qui permet de supporter des modifications triviales comme la transformation de N espaces consécutifs en un seul.

Reprenant le vocabulaire du RFC 5598 (relisez-le d'abord !), la section 3 de notre RFC liste les différents composants de la messagerie qui peuvent jouer un rôle dans les transmissions indirectes, et les problèmes qu'elles posent. D'abord, le MSA ("*Message Submission Agent*".) C'est la première ligne de vérification : il fait respecter les règles d'une organisation (ADMD, "*ADministrative Management Domain*"). S'il accepte un message où le champ `From:` du RFC 5322 n'est pas dans un domaine contrôlé par l'ADMD, il y a des chances que DMARC râle par la suite. Le RFC cite plusieurs cas d'usage où cela se produit : la fonction « envoyer cet article à un ami » de certains sites Web, par exemple, puisque le message va partir avec le domaine du lecteur de l'article, pas avec celui du site Web. On peut trouver de nombreux autres exemples, comme un service de gestion d'agenda qui envoie des courriers de rappel, en utilisant comme expéditeur l'adresse de l'utilisateur, ce qui est plutôt une bonne chose (pour des messages du genre « l'heure de la réunion a changé ») mais peut gêner DMARC. (Mon exemple préféré est le cas où on a une adresse de courrier mais pas de moyen de soumettre du courrier via cette organisation, ce qui est fréquent avec les adresses de fonction. Par exemple, on est membre d'une organisation qui fournit des adresses à ses membres et/ou responsables, ce qui permet de recevoir du courrier, mais on n'a pas de MSA pour en envoyer, on doit donc utiliser celui d'une autre organisation.)

Et les MTA, eux, quel est leur rôle dans les problèmes DKIM ? S'il change l'encodage (par exemple en passant du « 8 bits » à l'abominable "*Quoted-Printable*"), il va invalider les signatures DKIM (la canonicalisation de DKIM ne prévoit pas des transformations aussi radicales, même si elles ne modifient pas le message final). Idem si le MTA corrige les en-têtes du message pour les rendre conformes (une tâche qui relève plutôt du MSA, le MTA devant lui, transmettre fidèlement les messages qu'il a choisi d'accepter) : cela sort également du champ de la canonicalisation DKIM et cela invalide donc les éventuelles signatures. Enfin, le changement ou la suppression de certaines parties MIME (par exemple l'élision d'un document ZIP attaché, pour des raisons de protection contre les logiciels malveillants transmis par courrier) va évidemment également rendre les signatures invalides.

Et le MDA ? Peut-il casser des choses, également ? Oui, s'il passe les messages par Sieve (RFC 5228), qui a la possibilité d'ajouter ou de retirer des en-têtes, voire de modifier le corps (extension Sieve du RFC 5703). Si les tests DMARC sont faits après le passage de Sieve, ou bien si le message est ensuite réinjecté dans le système de courrier, des problèmes peuvent se produire.

Reste le cas des intermédiaires ("*mediators*"). Ce sont les entités qui prennent un message, puis le réexpédient (parfois après modification). Un exemple est l'"*alias*". Via une entrée dans `/etc/aliases` ou bien via un `.forward`, ou bien via le `redirect` de Sieve, ou encore via encore une autre méthode, un message initialement destiné à une adresse est finalement transmis à une autre. C'est par exemple courant pour les adresses « ancien élève », que fournissent certaines universités, et qui permettent de garder à vie une adresse dans le domaine de l'établissement où on a fait ses études. Un certain nombre d'associations professionnelles fournissent un service équivalent. En général, ces intermédiaires ne cassent pas DKIM (ils ne modifient pas le message) mais, selon la façon dont ils redirigent, peuvent invalider l'autorisation SPF.

Un autre exemple d'intermédiaire classique est le gestionnaire de listes de diffusion. En plus de rediriger un message (ce qui fait que le message écrit par `alice@univ.example` n'est **pas** émis par les serveurs de courrier de l'université), ces logiciels changent souvent le message, par exemple en ajoutant une inutile étiquette `[Ma jolie liste]` aux sujets des messages, en ajoutant un texte à la fin (instructions de désabonnement, pourtant déjà possibles avec l'en-tête `List-Unsubscribe:`), en retirant des pièces jointes, ou bien (surtout dans les théocraties comme les États-Unis) en remplaçant des gros mots par des termes plus acceptables.

Toutes ces modifications vont probablement invalider les signatures DKIM (cf. RFC 6377) et faire que les messages envoyés par certains participants à la liste (ceux qui ont une politique DMARC `p=reject`) ne seront pas reçus par les destinataires qui testent cette politique. (Si un avis de non-remise est transmis,

le logiciel de gestion de la liste peut en déduire que l'adresse n'existe pas, et désabonner d'autorité le destinataire.)

Et les filtres ? Certaines organisations insèrent dans le réseau des dispositifs qui vont analyser le courrier, par exemple à la recherche de logiciel malveillant. Souvent, ils vont modifier les messages, afin de supprimer ces contenus indésirables. Ces modifications vont évidemment invalider les signatures. Idem si on change ou supprime des URL contenus dans le message et considérés « dangereux ». Même chose avec un système anti-spam qui ajouterait un `[SPAM]` dans le sujet.

En revanche, le courrier reçu d'un serveur secondaire (MX de secours), qui a pris le relais pendant une panne du primaire, puis expédié le courrier quand le primaire remarche, ne pose pas de problèmes. Bien sûr, les tests SPF échoueront mais, normalement, on ne fait pas ces tests sur le courrier qui vient de son propre serveur secondaire.

Bon, voici le tour d'horizon complet de tout ce qui peut marcher mal. Mais que faire ? La section 4 du RFC s'attaque aux solutions. Elles sont nombreuses et très différentes. Mais attention : DMARC est là pour rejeter des messages considérés comme invalides. On peut arranger les choses pour que certains de ces messages « passent » mais cela va contre le but de DMARC. Si les messages sont de grande valeur (transactions financières, par exemple), il vaut mieux ne pas chercher de solutions, et simplement se contenter de messages transmis directement, ne subissant pas de traitements qui vont invalider SPF ou DKIM.

C'est d'autant plus vrai que l'écosystème du courrier électronique est très complexe. On trouve un zillion de logiciels différents, plus ou moins bien écrits. Par exemple, des gens utilisent encore Qmail, qui n'a plus eu une seule mise à jour depuis 1998. Certaines des mesures ou contre-mesures utilisées pour la sécurité du courrier sont parfaitement légales, mais vont casser tel ou tel logiciel qui est utilisé à certains endroits.

Assez d'avertissements, les solutions. D'abord, du côté de l'expéditeur. Celui-ci (ou son premier MTA) peut faire des efforts pour améliorer l'accord entre les identificateurs. Un logiciel sur `info.example` qui envoie du courrier pour le compte de `bob@univ.example` peut ainsi décider d'utiliser un en-tête `From:` qui ne posera pas de problème, celui du vrai envoyeur, et de mettre l'adresse de Bob dans un `Reply-To:`. Comme la plupart des solutions présentées dans cette section 4, elle est imparfaite (le destinataire peut se demander qui est cet envoyeur qu'il ne connait pas). Le RFC fournit de nombreux autres exemples de désaccord entre identités, qui peuvent être réparés en changeant un peu le processus d'envoi du message. Comme le disait ma grand-mère, « il y a toujours une solution, pour peu que chacun y mette du sien ».

Les envoyeurs peuvent aussi limiter le risque de modifications invalidantes, en ne signant pas trop d'en-têtes avec DKIM, ou en envoyant des messages parfaitement formés (pour éviter aux serveurs ultérieurs la tentation de les « réparer »).

Les receveurs peuvent aussi agir mais leurs possibilités sont plus limitées, note le RFC.

Entre les expéditeurs et les receveurs, il y a tous les intermédiaires qui prennent un message et le ré-expédient. Ce sont souvent eux qui causent le problème, et ils sont donc souvent en position de le réparer. Par exemple, ils peuvent changer le `From:` du message pour mettre le leur, ce qui permettrait à peu près n'importe quelle modification, et serait plus « franc » (puisque le message n'est plus tout à fait l'original, autant changer l'auteur...) Évidemment, dans ce cas, assez violent, il faut au minimum garder l'information sur l'émetteur originel, avec l'en-tête `Original-From:` (RFC 5703). Le problème est que

le récepteur humain sera sans doute déconcerté par cet expéditeur (d'autant plus qu'`Original-From:` est peu ou pas affiché).

Comme les modifications invalident les signatures, les ré-expéditeurs pourraient les éviter, par exemple en ajoutant des en-têtes au lieu de modifier les existants, lorsqu'ils veulent ajouter un contenu (du genre « ceci est un spam »). Il serait peut-être préférable, dans certains cas, de rejeter les messages plutôt que de les modifier, ce qui cassera la vérification de signatures plus loin.

Et, en parlant des ré-expéditeurs, les listes de diffusion, pas vraiment prévues par DKIM, que faire pour elles ? Le RFC 6377 a déjà traité leur cas. Une technique courante est de modifier le champ `From:` pour mettre l'adresse de la liste, réduisant l'auteur original à un commentaire dans cet en-tête (avis personnel : je déteste ça). Comme cela rend difficile de répondre en privé au vrai auteur d'un message, l'ajout d'un `Reply-To:` peut aider. Une autre solution est d'emballer le message original dans une partie MIME `message/rfc822`. Cette partie resterait intact et le message emballant aurait comme expéditeur la liste. Mais peu de MUA savent afficher proprement ce genre de messages (spécialement dans le monde des mobiles).

Encore plus fasciste, le gestionnaire de liste pourrait interdire l'abonnement des gens utilisant une adresse où il y a une politique DMARC autre que `p=none`. (Le RFC oublie de parler du cas où une politique `p=reject` n'existait pas au moment de l'abonnement mais a été rajoutée après.)

Enfin, il existe aussi des solutions qui sont encore en cours de discussion à l'IETF, et dont le RFC décourage l'usage dans un environnement de production. Ce sont entre autres des extensions au modèle du RFC 8601 pour créer une chaîne d'authentification où chaque acteur important signerait le message en route. Ou, plus radical, des mécanismes stockant l'état initial d'un message avant transformation, pour pouvoir retrouver cet état original et vérifier la signature.

Bref, le problème n'est pas résolu...