RFC 7601 : Message Header Field for Indicating Message Authentication Status

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 septembre 2015

Date de publication du RFC : Août 2015

https://www.bortzmeyer.org/7601.html

Il existe désormais plusieurs techniques pour authentifier les courriers électroniques. Certaines peuvent nécessiter des calculs un peu compliqués et on voudrait souvent les centraliser sur une machine de puissance raisonnable, dotée de tous les logiciels nécessaires. Dans cette hypothèse, le MUA ne recevra qu'une synthèse (« Ce message vient bien de `example.com` ») et pourra alors prendre une décision, basée sur cette synthèse. C'est le but de l'en-tête `Authentication-Results:`, normalisé originellement dans le RFC 5451 [1] six ans plus tôt, auquel a succédé le RFC 7001, que ce nouveau RFC met légèrement à jour (il y a peu de changements, le principal étant la correction de l'erreur #4201 <http://www.rfc-editor.org/errata_search.php?eid=4201>). Depuis, notre RFC a lui-même été remplacé par le RFC 8601.

Avec des techniques d'authentification comme DKIM (RFC 6376) ou SPF (RFC 7208), les calculs à faire pour déterminer si un message est authentique peuvent être complexes (DKIM utilise la cryptographie) et nécessiter la présence de bibliothèques non-standard. Les installer et les maintenir à jour sur chaque machine, surtout en présence d'éventuelles failles de sécurité qu'il faudra boucher en urgence, peut être trop pénible pour l'administrateur système. L'idée de ce RFC est donc de séparer l'opération en deux : l'authentification est faite sur un serveur, typiquement le premier MTA du site (cf. annexe C pour une discussion de ce choix), celui-ci ajoute au message un en-tête indiquant le résultat de ladite authentification et le MUA (ou bien le MDA, voir la section 1.5.3 pour un bon rappel sur ces concepts) peut ensuite, par exemple par un langage de filtrage comme procmail ou Sieve, agir sur la base de ce résultat. L'idée n'est donc pas de montrer la valeur de cet en-tête à M. Michu (voir la section 4.1 pour quelques risques que cela poserait), mais d'en faire une donnée pour un programme. Cet en-tête marche pour tous les protocoles d'authentification et surpasse donc les en-têtes spécifiques comme le `Received-SPF:` de SPF (section 1 du RFC). Le filtrage des messages non authentifiés n'est **pas** obligatoire (section 1.4) : agir - ou pas - sur la base de l'en-tête `Authentication-Results:` est une décision politique locale.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5451.txt`

J'ai utilisé le terme de « site » pour désigner un ensemble de machines gérées par la même organisation mais le RFC a un terme plus rigoureux, ADMD ("*ADministrative Management Domain*"). La frontière d'un ADMD est la « frontière de confiance » ("*trust boundary*"), définie en section 1.2. Un domaine administratif de gestion est un groupe de machines entre lesquelles il existe une relation de confiance, notamment du fait que, à l'intérieur de l'ADMD, l'en-tête `Authentication-Results:` ne sera pas modifié ou ajouté à tort (section 1.6 : l'en-tête n'est pas protégé, notamment il n'est pas signé). Il existe de nombreuses variantes organisationnelles du concept d'ADMD. Un ADMD inclus typiquement une organisation (ou un département de celle-ci) et d'éventuels sous-traitants. Il a un nom, l'`authserv-id`, défini en section 2.2.

L'en-tête `Authentication-Results:` lui-même est formellement défini en section 2. Il appartient à la catégorie des en-têtes de « trace » (RFC 5322, section 3.6.7 et RFC 5321, section 4.4) comme `Received:` qui doivent être ajoutés en haut des en-têtes et jamais modifiés. La syntaxe de `Authentication-Result` est en section 2.2. L'en-tête est composé du `authserv-id`, le nom de l'ADMD et d'une série de doublets (méthode, résultat), chacun indiquant une méthode d'authentification et le résultat obtenu. L'annexe B fournit une série d'exemples. Elle commence (annexe B.1) par un message sans `Authentication-Results:` (eh oui, il n'est pas obligatoire). Puis (tiré de l'annexe B.3), une authentification SPF réussie, au sein de l'ADMD `example.com`, donnera :

```
Authentication-Results: example.com;
          spf=pass smtp.mailfrom=example.net
Received: from dialup-1-2-3-4.example.net
              (dialup-1-2-3-4.example.net [192.0.2.200])
          by mail-router.example.com (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Wed, Mar 14 2009 17:19:07 -0800
From: sender@example.net
Date: Wed, Mar 14 2009 16:54:30 -0800
To: receiver@example.com
```

Rappelez-vous qu'il peut y avoir plusieurs authentifications. Voici un cas (annexe B.4) avec SPF et l'authentification SMTP du RFC 4954 :

```
Authentication-Results: example.com;
          auth=pass (cram-md5) smtp.auth=sender@example.net;
          spf=pass smtp.mailfrom=example.net
Received: from dialup-1-2-3-4.example.net (8.11.6/8.11.6)
              (dialup-1-2-3-4.example.net [192.0.2.200])
          by mail-router.example.com (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Fri, Feb 15 2002 17:19:07 -0800
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.com
From: sender@example.net
```

L'une des authentifications peut réussir et l'autre échouer. Un exemple (annexe B.6) avec deux signatures DKIM, une bonne et une qui était correcte au départ (regardez le premier `Authentication-Results:`) mais plus à l'arrivée, peut-être parce qu'un gestionnaire de liste de diffusion a modifié le message :

```
Authentication-Results: example.com;
       dkim=pass reason="good signature"
         header.i=@mail-router.example.net;
       dkim=fail reason="bad signature"
         header.i=@newyork.example.com
Received: from mail-router.example.net
          (mail-router.example.net [192.0.2.250])
```

```
      by chicago.example.com (8.11.6/8.11.6)
          for <recipient@chicago.example.com>
          with ESMTP id i7PK0sH7021929;
      Fri, Feb 15 2002 17:19:22 -0800
DKIM-Signature: v=1; a=rsa-sha256; s=furble;
      d=mail-router.example.net; t=1188964198; c=relaxed/simple;
      h=From:Date:To:Message-Id:Subject:Authentication-Results;
      bh=ftA9J6GtX8OpwUECzHnCkRzKw1uk6FNiLfJl5Nmv49E=;
      b=oINEO8hgn/gnunsg ... 9n9ODSNFSDij3=
Authentication-Results: example.net;
      dkim=pass (good signature) header.i=@newyork.example.com
Received: from smtp.newyork.example.com
          (smtp.newyork.example.com [192.0.2.220])
      by mail-router.example.net (8.11.6/8.11.6)
          with ESMTP id g1G0r1kA003489;
      Fri, Feb 15 2002 17:19:07 -0800
DKIM-Signature: v=1; a=rsa-sha256; s=gatsby;
      d=newyork.example.com;
      t=1188964191; c=simple/simple;
      h=From:Date:To:Message-Id:Subject;
      bh=sEu28nfs9fuZGD/pSr7ANysbY3jtdaQ3Xv9xPQtS0m7=;
      b=EToRSuvUfQVP3Bkz ... rTB0t0gYnBVCM=
From: sender@newyork.example.com
Date: Fri, Feb 15 2002 16:54:30 -0800
To: meetings@example.net
```

La liste complète des méthodes figure dans un registre IANA <https://www.iana.org/assignments/email-auth/email-auth.xhtml> (section 6). De nouvelles méthodes peuvent être enregistrées en utilisant la procédure « Examen par un expert » du RFC 5226.

La section 2.3 détaille l'authserv-id. C'est un texte qui identifie le domaine, l'ADMD. Il doit donc être unique dans tout l'Internet. En général, c'est un nom de domaine comme laposte.net. (Il est possible d'être plus spécifique et d'indiquer le nom d'une machine particulière mais cette même section du RFC explique pourquoi c'est en général une mauvaise idée : comme les MUA du domaine n'agissent que sur les Authentication-Results: dont ils reconnaissent l'authserv-id, avoir un tel identificateur qui soit lié au nom d'une machine, et qui change donc trop souvent, complique l'administration système.)

La section 2.7 explique les résultats possibles pour les méthodes d'authentification (en rappelant que la liste à jour des méthodes et des résultats est dans le registre IANA <https://www.iana.org/assignments/email-auth/email-auth.xhtml>). Ainsi, DKIM (section 2.7.1) permet des résultats comme pass (authentification réussie) ou temperror (erreur temporaire au cours de l'authentification, par exemple liée au DNS). Des résultats similaires sont possibles pour SPF (section 2.7.3).

Notons la normalisation d'une méthode traditionnelle d'authentification faible, le test DNS du chemin « adresse IP du serveur -¿ nom » et retour. Baptisée iprev, cette méthode, bien que bâtie sur la pure superstition (cf. section 7.11) est utilisée couramment. Très injuste (car les arbres des résolutions inverses du DNS, in-addr.arpa et ip6.arpa, ne sont pas sous le contrôle du domaine qui envoie le courrier), cette méthode discrimine les petits FAI, ce qui est sans doute un avantage pour les gros, comme AOL qui l'utilisent. Attention aux implémenteurs : aussi bien la résolution inverse d'adresse IP en nom que la résolution droite de nom en adresse IP peuvent renvoyer plusieurs résultats et il faut donc comparer des ensembles. (Cette méthode qui, contrairement aux autres, n'avait jamais été exposée dans un RFC avant le RFC 5451, est décrite en détail dans la section 3, avec ses sérieuses limites.)

Autre méthode mentionnée, auth (section 2.7.4) qui repose sur l'authentification SMTP du RFC 4954. Si un MTA (ou plutôt MSA) a authentifié un utilisateur, il peut le noter ici.

Une fois le code d'authentification exécuté, où mettre le `Authentication-Results:` ? La section 4 fournit tous les détails, indiquant notamment que le MTA doit placer l'en-tête en haut du message, ce qui facilite le repérage des `Authentication-Results:` à qui on peut faire confiance (en examinant les en-têtes `Received:` ; en l'absence de signature, un `Authentication-Results:` très ancien, situé au début du trajet, donc en bas des en-têtes, ne signifie pas grand'chose). On se fie a priori aux en-têtes mis par les MTA de l'ADMD, du domaine de confiance. L'ordre est donc important. (La section 7 revient en détail sur les en-têtes `Authentication-Results:` usurpés.)

Ce n'est pas tout de mettre un `Authentication-Results:`, encore faut-il l'utiliser. La section 4.1 s'attaque à ce problème. Principe essentiel pour le MUA : ne pas agir sur la base d'un `Authentication-Results:`, même si ce n'est que pour l'afficher, sans l'avoir validé un minimum. Comme le `Authentication-Results:` n'est pas signé, n'importe qui a pu en insérer un sur le trajet. Le RFC précise donc que les MUA doivent, par défaut, ne rien faire. Et qu'ils doivent ne regarder les `Authentication-Results:` qu'après que cela ait été activé par l'administrateur de la machine, qui indiquera quel `authserv-id` est acceptable.

Naturellement, le MTA d'entrée du domaine devrait supprimer les `Authentication-Results:` portant son propre `authserv-id` qu'il trouve dans les messages entrants : ils sont forcément frauduleux (section 5). (Le RFC accepte aussi une solution plus simpliste, qui est de supprimer tous les `Authentication-Results:` des messages entrants, quel que soit leur `authserv-id`.)

Arrivé à ce stade de cet article, le lecteur doit normalement se poser bien des questions sur la valeur du `Authentication-Results:`. Quel poids lui accorder alors que n'importe quel méchant sur le trajet a pu ajouter des `Authentication-Results:` bidons ? La section 7, consacrée à l'analyse générale de la sécurité, répond à ces inquiétudes. 7.1 détaille le cas des en-têtes usurpés. Les principales lignes de défense ici sont le fait que le MUA ne doit faire confiance aux `Authentication-Results:` que s'ils portent le `authserv-id` de son ADMD **et** le fait que le MTA entrant doit filtrer les `Authentication-Results:` avec son `authserv-id`. Comme l'intérieur de l'ADMD, par définition, est sûr, cela garantit en théorie contre les `Authentication-Results:` usurpés. Le RFC liste néanmoins d'autres méthodes possibles comme le fait de ne faire confiance qu'au **premier** `Authentication-Results:` (le plus récent), si on sait que le MTA en ajoute systématiquement un (les éventuels `Authentication-Results:` usurpés apparaîtront après ; mais certains serveurs les réordonnent, cf. section 7.3). Pour l'instant, il n'y a pas de méthode unique et universelle de vérification du `Authentication-Results:`, le RFC propose des pistes mais ne tranche pas.

Comme toujours en sécurité, il faut bien faire la différence entre authentification et autorisation <https://www.bortzmeyer.org/authentifier-et-autoriser.html>. Un spammeur a pu insérer un `Authentication-Results:` légitime pour **son** `authserv-id`. Même authentifié, il ne doit pas être considéré comme une autorisation (section 7.2).

Plusieurs mises en œuvre de ce système existent déjà comme dans MDaemon, sendmail (via sid-milter <http://sourceforge.net/projects/sid-milter>), Courier, OpenDKIM <http://www.opendkim.org/>, etc. Si on veut analyser les en-têtes `Authentication-Results:` en Python, on a le module authres <https://pypi.python.org/pypi/authres/>. Parmi les grosses usines à courrier centralisées, Gmail met systématiquement cet en-tête, par exemple :

```
Authentication-Results: mx.google.com; spf=pass \
          (google.com: domain of stephane@sources.org designates 217.70.190.232 \
              as permitted sender) smtp.mail=stephane@sources.org
```

Outre Gmail, à la date de publication du RFC, Yahoo et Hotmail ajoutaient cet en-tête.

Les changements depuis le RFC 7001 sont peu nombreux (annexe D pour une liste complète). Le RFC 7410, qui créait le registre des types d'information possibles sur l'authentification <https://www.iana.org/assignments/email-auth/email-auth.xhtml#property-types> a été intégré et est donc remplacé par notre nouveau RFC. Autrement, l'un des principaux changements concerne la bogue #4201 <http://www.rfc-editor.org/errata_search.php?eid=4201>. Le texte du précédent RFC disait que la source de l'authentification devait être un en-tête du message alors que cela peut être un champ particulier d'un en-tête (le cas du champ `i` dans les signatures DKIM d'exemple dans cet article, cf. 2.3 et 2.7.1). Autrement, les changements par rapport au RFC 7001 sont surtout des détails et les mises en œuvre actuelles devraient continuer sans trop d'histoires. Je rappelle que ce RFC 7601 n'est plus d'actualité, ayant été remplacé par le RFC 8601.