RFC 8807 : Login Security Extension for the Extensible Provisioning Protocol (EPP)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 août 2020

Date de publication du RFC : Août 2020

https://www.bortzmeyer.org/8807.html

Le protocole EPP d'avitaillement de ressources (par exemple de noms de domaine) a un mécanisme de sécurité très simple, à base de mots de passe indiqués lors de la connexion. Ce mécanisme avait plein de limitations ennuyeuses, dans le RFC original, et ce nouveau RFC vise à les réduire.

L'authentification dans EPP est décrite dans le RFC 5730[1], section 2.9.1.1 (voir aussi sa section 7). Le client EPP envoie un mot de passe, qui doit faire entre 6 et 16 caractères (cf. le type `pwType` dans le schéma XSD du RFC 5730, section 4.1). Le client peut changer son mot de passe en indiquant un nouveau mot via le protocole EPP, sans avoir à passer par un autre service. En outre, la session EPP est typiquement portée sur TLS, ce qui assure la confidentialité du mot de passe, et la possibilité d'ajouter une authentification par le biais d'un certificat client. Mais c'est tout. Le RFC 5730 ne permet pas de mot de passe plus long, ne permet pas au client de savoir combien de tentatives erronées ont eu lieu, ne permet pas d'indiquer l'expiration d'un mot de passe (si ceux-ci ont une durée de vie limitée) et ne permet pas au serveur EPP d'indiquer, s'il refuse un nouveau mot de passe, pourquoi il le juge inacceptable.

Cette extension à EPP figure désormais dans le registre des extensions <https://www.iana.org/assignments/epp-extensions/epp-extensions.xml>, créé par le RFC 7451.

Notre RFC ajoute donc plusieurs nouveaux éléments XML qui peuvent apparaitre en EPP (section 3). D'abord, la notion d'évènement. Un évènement permet d'indiquer, dans un élément `<event>` :

— l'expiration d'un mot de passe,
— l'expiration du certificat client,

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5730.txt`

— la faiblesse d'un algorithme cryptographique,
— la version de TLS, par exemple pour rejeter une version trop basse,
— les raisons du rejet d'un nouveau mot de passe (« le mot de passe doit comporter au moins une majuscule, une minuscule, un chiffre arabe, un chiffre romain, une rune et un hiéroglyphe », et autres règles absurdes mais fréquentes),
— des statistiques de sécurité, comme le nombre de connexions refusées suite à un mot de passe incorrect.

En utilisant `loginSec` comme abréviation pour l'espace de noms de l'extension EPP de ce RFC, `urn:ietf:params:` voici un exemple d'évènement, indiquant qu'il faudra bientôt changer de mot de passe :

```
 <loginSec:event
     type="password"
     level="warning"
     exDate="2020-04-01T22:00:00.0Z"
     lang="fr">
     Le mot de passe va bientôt expirer.
</loginSec:event>
```

On pourrait voir aussi cet évènement indiquant le nombre de connexions ratées, ce qui peut alerter sur une tentative de découverte du mot de passe par force brute :

```
<loginSec:event
     type="stat"
     name="failedLogins"
     level="warning"
     value="100"
     duration="P1D"
     lang="fr">
     Trop de connexions ratées.
</loginSec:event>
```

Si on utilise des mots de passe qui suivent les nouvelles règles, il faut l'indiquer en mettant dans l'ancien élément `<pw>` la valeur `[LOGIN-SECURITY]`. Si elle est présente, c'est qu'il faut aller chercher le mot de passe dans le nouvel élément `<loginSec:pw>` (idem pour un changement de mot de passe).

La section 4 du RFC donne les changements pour les différentes commandes EPP. Seule `<login>` est concernée. Ainsi, `<login>` gagne un élément `<loginSec:userAgent>` qui permet d'indiquer le type de logiciel utilisé côté client. Mais le principal ajout est évidemment `<loginSec:pw>`, qui permet d'utiliser les mots de passe aux nouvelles règles (mots de passe plus longs, par exemple). Il y a aussi un `<loginSec:newPw>` pour indiquer le nouveau mot de passe à utiliser. Notez que, si le mot de passe comprend des caractères Unicode, il est recommandé qu'ils se conforment au RFC 8265 (profil `OpaqueString`).

Voici un exemple d'une commande `<login>` nouveau style :

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <login>
      <clID>ClientX</clID>
      <pw>[LOGIN-SECURITY]</pw>
      ...
    </login>
    <extension>
      <loginSec:loginSec
        xmlns:loginSec=
          "urn:ietf:params:xml:ns:epp:loginSec-1.0">
        <loginSec:userAgent>
          <loginSec:app>EPP SDK 1.0.0</loginSec:app>
          <loginSec:tech>Vendor Java 11.0.6</loginSec:tech>
          <loginSec:os>x86_64 Mac OS X 10.15.2</loginSec:os>
        </loginSec:userAgent>
        <loginSec:pw>this is a long password not accepted before</loginSec:pw>
      </loginSec:loginSec>
    </extension>
    ...
  </command>
</epp>
```

Et une réponse positive du serveur EPP à cette connexion, mais qui avertit que le mot de passe va expirer le 1er juillet :

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <extension>
      <loginSec:loginSecData
        xmlns:loginSec=
          "urn:ietf:params:xml:ns:epp:loginSec-1.0">
        <loginSec:event
          type="password"
          level="warning"
          exDate="2020-07-01T22:00:00.0Z"
          lang="en">
          Password expiring in a week
        </loginSec:event>
      </loginSec:loginSecData>
    </extension>
    ...
  </response>
</epp>
```

Et, ici, lorsqu'un client a voulu changer son mot de passe expiré, avec `<loginSec:newPw>`, mais que le nouveau mot de passe était trop simple (cf. les recommandations de l'ANSSI `<https://www.ssi.gouv.fr/entreprise/guide/mot-de-passe/>`):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
```

```
    <result code="2200">
      <msg>Authentication error</msg>
    </result>
    <extension>
      <loginSec:loginSecData
        xmlns:loginSec=
          "urn:ietf:params:xml:ns:epp:loginSec-1.0">
        <loginSec:event
          type="password"
          level="error"
          exDate="2020-03-24T22:00:00.0Z">
          Password has expired
        </loginSec:event>
        <loginSec:event
          type="newPW"
          level="error"
          lang="fr">
          Mot de passe vraiment trop trivial.
        </loginSec:event>
      </loginSec:loginSecData>
    </extension>
    ...
  </response>
</epp>
```

Le schéma complet figure dans la section 5 du RFC.

Un changement plus radical aurait été de passer à un cadre d'authentification plus général comme SASL (RFC 4422) mais l'IETF a choisi une évolution plus en douceur.

À l'heure actuelle, je ne connais que deux mises en œuvre de ce RFC, dans le SDK de Verisign, en et dans le logiciel libre Net : :DRI `<https://metacpan.org/release/Net-DRI>`. Apparemment, aucun serveur EPP de « grand » registre n'annonce l'extension, à part Verisign.