RFC 8908 : Captive Portal API

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 septembre 2020

Date de publication du RFC : Septembre 2020

https://www.bortzmeyer.org/8908.html

Un des nombreux problèmes posés par les portails captifs est l'interaction avec le portail, par exemple pour accepter les CGU et continuer. Ces portails ne sont en général prévus que pour une interaction avec un humain. Ce RFC décrit une API ultra-simple qui permet à des programmes, au moins de savoir s'il y a un portail, quelles sont ses caractéristiques et comment sortir de captivité.

L'API suit les principes du RFC 8952 [1]. Elle permet donc de récupérer l'état de la captivité (est-ce que j'ai un accès à l'Internet ou pas), et l'URI de la page Web avec laquelle l'humain devra interagir.

Comment est-ce que la machine qui tente de se connecter a appris l'existence de l'API et son URI d'entrée ? Typiquement via les options de DHCP ou de RA décrites dans le RFC 8910. On accède ensuite à l'API avec HTTPS (RFC 2818). Il faudra naturellement authentifier le serveur, ce qui peut poser des problèmes tant qu'on n'a pas un accès à l'Internet complet (par exemple à OCSP, RFC 6960, et à NTP, RFC 5905, pour mettre l'horloge à l'heure et ainsi vérifier que le certificat n'a pas encore expiré). De même, des certificats intermédiaires qu'il faut récupérer sur l'Internet via "*Authority Information Access*" (AIA, section 5.2.7 du RFC 5280) peuvent poser des problèmes et il vaut mieux les éviter.

L'API elle-même est présentée en section 5 du RFC. Le contenu est évidement en JSON (RFC 8259), et servi avec le type `application/captive+json`. Dans les réponses du serveur, l'objet JSON a obligatoirement un membre `captive` dont la valeur est un booléen et qui indique si on on est en captivité ou pas. Les autres membres possibles sont :

— `user-portal-uri` indique une page Web pour humains, avec laquelle l'utilisateur peut interagir,
— `venue-info-url` est une page Web d'information,

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc8952.txt`

— `can-extend-session`, un booléen qui indique si on peut prolonger la session, ce qui veut dire que cela peut être une bonne idée de ramener l'humain vers la page Web `user-portal-uri` lorsque la session va expirer,
— `seconds-remaining`, le nombre de secondes restant pour cette session, après quoi il faudra se reconnecter (si `can-extend-session` est à Vrai),
— `bytes-remaining`, la même chose mais pour des sessions limitées en quantité de données et plus en temps.

Ces réponses de l'API peuvent contenir des données spécifiques au client, et donc privées. Auquel cas, le serveur doit penser à utiliser `Cache-control: private` (RFC 7234) ou un mécanisme équivalent, pour éviter que ces données se retrouvent dans des caches.

Un exemple complet figure en section 6 du RFC. On suppose que le client a découvert l'URL `https://example.c` via un des mécanismes du RFC 8910. Il envoie alors une requête HTTP :

```
GET /captive-portal/api/X54PD39JV HTTP/1.1
Host: example.org
Accept: application/captive+json
```

Et reçoit une réponse :

```
HTTP/1.1 200 OK
Cache-Control: private
Date: Mon, 02 Mar 2020 05:07:35 GMT
Content-Type: application/captive+json

{
      "captive": true,
      "user-portal-url": "https://example.org/portal.html"
}
```

Il sait alors qu'il est en captivité, et que l'utilisateur doit aller en `https://example.org/portal.html` pour accepter des CGU léonines, s'authentifier, etc. Une fois que c'est fait, il peut continuer à faire des requêtes à l'API et avoir, par exemple :

```
{
      "captive": false,
      "user-portal-url": "https://example.org/portal.html",
      "venue-info-url": "https://flight.example.com/entertainment",
      "seconds-remaining": 326,
      "can-extend-session": true
}
```

D'autres membres de l'objet JSON pourront apparaitre, selon la procédure « Spécification nécessaire » (RFC 8126), un registre IANA <`https://www.iana.org/assignments/captive-portals/captive-portals.xml`> a été créé pour les stocker.

Un peu de sécurité pour finir (section 7 du RFC). Le protocole de notre RFC impose l'utilisation de HTTPS (donc de TLS) ce qui règle pas mal de problèmes, notamment de confidentialité et d'authentification. À noter donc (mais cela en vaut la peine) que cela complique un peu les choses pour l'administrateur du portail captif, qui ne peut pas se contenter de HTTP en clair, et qui doit avoir un certificat valide.

(Cet argument de la complexité avait été mentionné lors des discussions à l'IETF, où certains trouvaient notre RFC trop exigeant.) Combien de fois ai-je vu des portails captifs avec un certificat auto-signé et/ou expiré !

Mais attention, TLS va seulement sécuriser le fait qu'on se connecte bien au serveur indiqué via les méthodes du RFC 8910. Or, comme ces méthodes ne sont pas elle-mêmes très sûres, la sécurité du portail captif ne doit jamais être surestimée.

Le protocole décrit dans le RFC 8910 et dans ce RFC a été testé lors de réunions IETF, sur le grand réseau de ces réunions. En dehors de cela, il n'y a pas encore de déploiement. Je crains que, vu la nullité technique de la plupart des points d'accès WiFi, vite installés, mal configurés et plus maintenus après, il ne faille attendre longtemps un déploiement significatif.