RFC 7593 : The eduroam architecture for network roaming

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 6 octobre 2015

Date de publication du RFC : Septembre 2015

https://www.bortzmeyer.org/7593.html

Le service eduroam permet aux étudiants et employés des universités et centres de recherche européens de se connecter en Wi-Fi même quand ils ne sont pas dans leur université habituelle. Ce RFC décrit le fonctionnement d'eduroam et les choix effectués lors de sa conception (notamment l'absence de ces abominables portails captifs). La taille d'eduroam et ses années de fonctionnement indiquent que ces choix étaient bons et qu'eduroam peut être un modèle pour bien d'autres déploiements de Wi-Fi.

Le projet d'un accès pour le monde académique européen a démarré en 2002 (cf. la proposition initiale <http://www.terena.org/activities/tf-mobility/start-of-eduroam.pdf>, par un des auteurs du RFC). Aujourd'hui, eduroam couvre 10 000 sites (dont un en Papouasie Nouvelle-Guinée <https://www.eduroam.org/index.php?p=media&id=13>) et des millions d'utilisateurs.

Les buts d'eduroam étaient (section 1 du RFC) :

— Identification unique des utilisateurs, gérée par leur établissement d'origine. Si un chercheur de l'Université Copernic visite l'Université de Pise, pas question que les Italiens soient obligés de lui créer un compte, de lui attribuer encore un nouveau mot de passe, etc. (Point de terminologie au passage : dans ce scénario, l'université d'origine du visiteur est l'IdP - "*Identity Provider*" - et celle qu'il visite est le SP - "*Service Provider*", qui fournit l'accès.)
— Accès au réseau de l'établissement visité et par là à tout l'Internet. (Ce qui n'interdit pas de mettre les visiteurs dans un VLAN à part, ce qui se fait souvent.)
— Minimum de procédures administratives. Il faut éviter, par exemple, que chacun des 10 000 établissements membres ait à faire quelque chose (signer un contrat, échanger des certificats) avec chacun des 9 999 autres. Cela ne passerait pas à l'échelle.
— Facile à configurer et utiliser : tous les académiques ne sont pas des experts en informatique.
— Sûr. Pas question que n'importe quel craqueur de passage puisse avoir un accès gratuit et irresponsable via une université.
— Raisonnablement respectueux de la vie privée. Par exemple, le SP, l'établissement d'accueil, ne devrait pas être en mesure de tout savoir de ses visiteurs.

— Évidemment fondé sur des normes ouvertes, de manière à ne pas imposer l'utilisation d'un vendeur particulier.

Avec un tel cahier des charges (section 1), plutôt ambitieux, les solutions qui avaient été envisagées et testées étaient :

— Utilisation de VPN. Sûre mais ne passant pas du tout à l'échelle.
— Portail captif. Sécurité bien trop mauvaise (et, je rajoute, très pénible pour les utilisateurs). Les portails captifs ont tous les défauts (cf. l'excellente FAQ d'eduroam <https://www.eduroam.org/index.php?p=faq#captive>) : reposant sur un détournement du trafic, ils sont indistinguables d'une attaque de l'Homme du Milieu et ils contribuent donc à mal éduquer les utilisateurs (par exemple à ignorer les avertissements HTTPS), reposant sur du trafic Wi-Fi en clair, ils permettent tout un tas d'attaques comme la surveillance des communications du voisin, nécessitant une action manuelle d'interaction avec un site Web, ils ne fonctionnent pas avec des programmes automatiques comme Windows Update, ils ne font pas d'authentification de bout en bout (il faut donner son mot de passe au SP), etc. (Note personnelle : il est anormal qu'ils existent encore en 2015.)
— 802.1X, la solution finalement choisie.

Désormais, eduroam repose sur trois piliers :

— 802.1X pour l'authentification locale, avec le point d'accès Wi-Fi,
— EAP (RFC 3748 [1]) pour transporter l'authentification, de manière sûre et confidentielle,
— RADIUS (RFC 2865), pour le routage des requêtes, jusqu'à l'établissement d'origine (l'IdP).

L'utilisation d'EAP dans RADIUS est décrite dans le RFC 3579.

L'architecture d'eduroam est décrite en section 2 de notre RFC. Qui dit authentification dit confiance : d'où vient la confiance dans eduroam ? Il y a une relation de confiance entre l'utilisateur et l'IdP, son établissement d'origine, vérifiée par une authentification mutuelle. Et il y a une relation de confiance entre l'IdP et le SP (l'établissement d'accueil), fondée sur RADIUS.

Si une université, ou établissement analogue, participe au service eduroam, ses bornes Wi-Fi vont publier le SSID `eduroam`. C'est celui que choisit l'utilisateur la première fois, la connexion sera typiquement faite automatiquement les fois suivantes. Notez qu'il est toujours possible qu'un point d'accès méchant diffuse un faux `eduroam` : l'utilisateur ne doit pas considérer que tout SSID `eduroam` est sûr ! Notez aussi, mais c'est moins grave, qu'un utilisateur qui est dans son université habituelle peut se connecter via eduroam, avec les éventuelles limitations que cela implique, plutôt qu'au réseau local de son université. Pour s'accrocher au point d'accès Wi-Fi du SP, la machine de l'utilisateur utilise, on l'a vu, 802.1X. Les transmissions radio sont sécurisées par WPA2 avec AES. Il n'y a pas d'accès anonyme à eduroam, tout utilisateur doit avoir un compte sur un des IdP.

Et EAP, il sert à quoi ? Il permet une protection de bout en bout des informations d'authentification. Même si l'utilisateur s'est connecté à un faux réseau prétendant être eduroam, celui-ci ne pourra pas intercepter, par exemple les mots de passe. Seul l'IdP, l'institution d'origine de l'utilisateur, verra son mot de passe. EAP n'est pas directement un mécanisme d'authentification mais un protocole qui permet de négocier et transporter plusieurs mécanismes d'authentification (du « simple » mot de passe - RFC 4746 - aux certificats - RFC 5216). EAP est de bout en bout, entre l'utilisateur et son IdP, son fournisseur d'identité. Si l'université de rattachement de l'utilisateur déploie une technique d'authentification nouvelle et très rare, pas de problème : les SP, et les serveurs RADIUS n'ont pas à la gérer, seul EAP sera au courant. L'utilisateur s'identifie avec un NAI (RFC 7542), comportant un nom local à l'IdP, un @ et un domaine (ou royaume, "*realm*") unique. Pour des raisons de protection de la vie privée, on peut utiliser un NAI ne comportant que le domaine (par exemple `@ma-fac.example`), le NAI complet n'étant transporté que dans EAP (pensez à un tunnel), et donc chiffré et inaccessible aux intermédiaires.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc3748.txt`

Le point d'accès Wi-Fi va ensuite être un client RADIUS pour transmettre les informations permettant l'authentification. (Du point de vue sécurité, notez que cette communication entre le client RADIUS et le premier serveur RADIUS est entièrement chez le SP, et sécurisée par lui, typiquement via les secrets partagés de RADIUS.)

Une fois qu'on est arrivé au serveur RADIUS du SP, que se passe-t-il ? Il faut relayer jusqu'au serveur RADIUS de l'IdP. Les serveurs RADIUS de eduroam sont organisés de manière arborescente, en suivant à peu près l'arbre du DNS. Si un professeur à `soton.ac.uk` visite l'Université de l'Utah, sa requête va d'abord passer par le serveur RADIUS de cette université. Ce serveur de l'université `utah.edu` ne connait que le serveur RADIUS et `edu`. Il lui transmet donc la requête, le serveur de `edu` passe à la racine (notez que, contrairement au DNS, on ne part pas de la racine) et la requête redescend ensuite vers le serveur de l'IdP (serveurs de `ac.uk` puis de `soton.ac.uk`).

Donc, pour détailler le processus, les étapes étaient :

— Le professeur (enfin, son logiciel, ce qui se nomme dans WPA le "*supplicant*") envoie une requête EAP au point d'accès Wi-Fi de l'Utah, avec une identité `anonymous@soton.ac.uk`,
— Le point d'accès états-unien fait suivre au serveur RADIUS de l'Université de l'Utah,
— Ce serveur voit que la requête n'est pas pour lui (le domaine `soton.ac.uk` n'est pas le sien), il fait suivre au serveur RADIUS de `.edu`,
— Ce serveur voit que ce n'est pas pour lui (ce n'est pas un `.edu`), il transmet donc à la racine (au passage, notez que la racine est évidemment composée de plusieurs serveurs, pour des raisons de redondance),
— La racine peut donc transmettre au serveur RADIUS de `ac.uk` (notez que la racine est généralement configurée avec des TLD mais `ac.uk` est une exception),
— Et ce serveur va transmettre à l'université, puisqu'il connait `soton.ac.uk`,
— Le serveur RADIUS de l'université britannique va alors décapsuler, trouver la requête EAP et la traiter (c'est-à-dire authentifier l'utilisateur),
— La réponse (message RADIUS `Access-Accept` ou `Access-Reject`) est renvoyée dans l'Utah en suivant la hiérarchie RADIUS d'eduroam en sens inverse,
— Le serveur RADIUS de l'Université de l'Utah peut alors informer le point d'accès Wi-Fi qu'il doit laisser passer (ou pas) l'utilisateur.

On voit qu'il faut en faire bouger des électrons, et très vite et de manière très fiable. eduroam a logiquement connu quelques malheurs (section 3 du RFC). Par exemple, les serveurs RADIUS à utiliser pour router la requête sont statiquement configurés (une grosse différence avec le DNS). Le RFC 2865 ne dit pas ce qu'il faut faire en cas de panne. RADIUS fonctionne sur UDP et la panne n'est donc pas indiquée explicitement, on a juste une absence de réponse. Pour l'utilisateur final, cela peut se traduire par un long délai, suivi d'un refus d'accès inexplicable. En prime, comme il y a plusieurs relais RADIUS successifs, l'administrateur d'une université, en cas de non-réponse, ne sait pas quel relais a défailli. Son RADIUS ne peut pas distinguer une panne du premier relais d'une panne de la racine, sauf en constatant que les éventuelles requêtes pour des domaines du même TLD marchent. Un processus de déboguage pénible (cf. l'article « "*Dead-realm marking feature for Radiator RADIUS servers*" `<http://www.eduroam.cz/dead-realm/docs/dead-realm.html>` »). Et il n'y avait pas non plus dans RADIUS de message de retour indiquant la fin de la panne (ce point a toutefois été résolu avec le RFC 5997).

RADIUS est nettement faiblard en terme de signalisation des erreurs mais le protocole 802.1X ne permet pas non plus de transmettre à l'utilisateur final un message très détaillé. Un serveur RADIUS qui dépend d'un serveur dorsal extérieur (LDAP ou SGBD par exemple) a donc un choix cornélien si ce serveur extérieur est défaillant : le serveur RADIUS peut ne pas répondre du tout, mais cela va être interprété comme une panne par les relais RADIUS, qui risquent de ne plus envoyer de requêtes. Et le logiciel de l'utilisateur ne va pas transmettre de message utile, voire il donnera des avis erronés (« vérifiez votre mot de passe »), en raison du manque d'informations. Soit, autre solution, le serveur RADIUS répond par un refus (RFC 2865, section 4.3), ce qui serait mieux pour les autres relais RADIUS

(ils verront que leur camarade répond toujours) mais très déroutant pour l'utilisateur final, qui verra des messages trompeurs et effrayants, par exemple « votre mot de passe est incorrect ou votre compte est fermé ». Ainsi, Windows, dans un tel cas, efface les informations d'authentification qui étaient enregistrées et oblige l'utilisateur à tout recommencer. Ce problème a suscité de nombreux appels à l'assistance utilisateur, car les gens avaient évidemment oublié le mot de passe qu'ils avaient configuré des semaines ou des mois auparavant...

Les innombrables discussions dans la communauté eduroam ou dans le groupe de travail RADEXT `<https://tools.ietf.org/wg/radext>` de l'IETF n'ont pas permis d'arriver à un consensus. Si RADIUS sur TCP, introduit depuis, résout une partie des problèmes, il ne solutionne pas celui du dorsal (LDAP ou SGBP) planté. Il faudrait un mécanisme analogue au code d'erreur 500 de HTTP mais RADIUS n'en a pas.

Autre problème, la complexité du routage entre les relais RADIUS. Le faire sur la base du TLD marche bien pour les ccTLD (le serveur RADIUS national est administré par l'association eduroam du pays) mais pas du tout pour `.com`. Il a fallu mettre des exceptions manuelles comme `kit.edu` routé vers le serveur allemand car le KIT est en Allemagne, ce qui n'était pas évident vu son TLD... Cela pose un problème de passage à l'échelle : s'il n'y a qu'environ 200 ccTLD (dont 50 sont aujourd'hui dans eduroam), il peut y avoir potentiellement des milliers d'établissements dans les gTLD, chacun avec son exception à mettre dans les tables de routage, et à propager vers tous les serveurs (pour des raisons de fiabilité, il y a plusieurs serveurs racines).

Je l'ai dit, RADIUS fonctionne sur UDP. Ce choix était raisonnable dans les configurations simples du début (un client RADIUS, le point d'accès, et un serveur RADIUS, sans relais), mais il a des conséquences désagréables pour la configuration bien plus complexe d'eduroam. EAP est un protocole bavard et il n'est pas rare d'avoir 8 à 10 aller-retours entre le serveur RADIUS du SP et celui de l'IdP. La perte d'un seul paquet nécessitera de tout recommencer, UDP n'ayant pas de retransmission. Dans certains cas, les données EAP sont de taille importante (c'est surtout le cas avec EAP-TLS, où il faut transmettre de gros certificats) et on peut voir des paquets RADIUS dépasser la MTU et être fragmentés. Si, en théorie, la fragmentation n'est pas un problème, en pratique, on sait qu'elle marche mal sur l'Internet, notamment en raison des nombreux pare-feux bogués ou mal configurés. Ces problèmes sont très difficiles à déboguer car, si Alice et Bob constatent que les paquets entre eux ne passent pas, la cause peut être un réseau tiers, situé sur le chemin, et qu'ils ne contrôlent pas. (RADIUS sur TCP, RFC 6613, résout ce problème. Le RFC 6613 discute plus en détail ce problème.)

Autre limite de RADIUS tel qu'il était au début d'eduroam : la protection de la vie privée. Le protocole permettait de chiffrer les mots de passe mais pas le reste du trafic. C'est donc à EAP de contourner ce problème, par exemple avec son authentification TLS. Mais cela ne suffit pas car il y a toujours des attributs « privés » transmis en clair par RADIUS comme `Calling-Station-ID` qui peut permettre de trouver le terminal de l'utilisateur, ou comme `NAS-IP-Address` qui permet de trouver le point d'accès. (Le problème se pose aussi dans la réponse : si le message d'acceptation est trop bavard, par exemple en indiquant le département de l'utilisateur à l'université, cela laisse fuiter des informations en clair.) Cela permet donc à un attaquant puissant, en surveillant tous les flux RADIUS (ce qui est largement dans les moyens de la NSA), de suivre à la trace bien des gens. À noter que une telle récolte d'informations n'a pas encore été documentée (on ne trouve pas RADIUS ou eduroam dans les documents Snowden) mais cela ne veut pas dire qu'elle n'a pas eu lieu. La seule solution est de chiffrer toute la session RADIUS, ce qui n'était pas possible dans le RADIUS original (mais le devient avec le RFC 6614).

EAP protège également en permettant l'utilisation d'identificateurs « anonymes », le vrai identificateur n'étant transmis que dans la session EAP chiffrée. Encore faut-il que l'utilisateur configure son logiciel pour cela, et tous les logiciels ne le permettent pas.

Ces sécurités fournies par EAP ne marchent que si on se connecte au bon serveur EAP. Si on se connecte à la place au terrible Homme du Milieu, plus aucune sécurité ne joue. Est-ce que les utilisateurs ont tous configuré leur logiciel pour tester la validité du certificat du serveur EAP, et la tester contre une liste d'AC sérieuses ? On peut en douter et il est donc sans doute possible de capturer des sessions Wi-Fi en présentant un faux serveur EAP.

Ces problèmes de configuration d'EAP sont d'autant plus pénibles que beaucoup des mises en œuvres d'EAP ne permettent pas aux administrateurs système de diffuser facilement des configurations pré-remplies (et, lorsqu'il y en a, elles sont spécifiques à un vendeur), où tous les paramètres de sécurité ont été fixés aux bonnes valeurs, et où l'utilisateur n'a plus qu'à taper nom et mot de passe (ou bien indiquer le certificat client).

Les difficultés pratiques rencontrées avec l'architecture d'eduroam ont mené à certains changements, décrits dans la section 4 du RFC. Certains changements ont nécessité une action normalisatrice à l'IETF. Ces changements peuvent cohabiter avec l'infrastructure actuelle (pas question de tout raser pour recommencer de zéro) et sont donc déployés progressivement. Les deux grands changements sont le remplacement progressif d'UDP par TCP, et l'utilisation de TLS pour sécuriser la session RADIUS (l'ancien système était un secret partagé entre les deux pairs RADIUS).

Au passage, pourquoi ne pas avoir remplacé RADIUS par Diameter (RFC 6733) ? Cela avait été envisagé puisque, sur le papier, le protocole Diameter, bien plus complexe et « "*enterprise-grade*" » (c'est-à-dire apprécié des DSI, et n'ayant pas de mise en œuvre en logiciel libre), disposait déjà des services voulus. Mais l'examen de l'offre Diameter existante, par rapport aux exigences d'eduroam (logiciels gratuits ou bon marché, gestion des authentifications EAP les plus courantes, accès à des dorsaux courants comme MySQL, etc) a été décevante. En prime, les points d'accès Wi-Fi existants n'avaient pas de client Diameter (et n'en ont toujours pas).

D'où le choix de travailler à l'IETF pour développer les nouveaux services (RFC 6613, RFC 6614), et avec les développeurs de logiciel libre pour les mettre en œuvre.

Déployer TCP et TLS dans l'infrastructure de serveurs RADIUS a permis de détecter les serveurs en panne (avec les "*keepalive*" de TCP, cf. RFC 6613, section 2.6) et de protéger le contenu échangé contre les écoutes. Les certificats X.509 d'eduroam, qui servent à authentifier les pairs RADIUS, sont fournis par un ensemble d'AC accréditées.

TCP et TLS laissent un problème, celui du routage statique des requêtes, dans la hiérarchie des serveurs eduroam. La solution se nomme « découverte dynamique ». Il y a deux problèmes à résoudre : trouver le serveur RADIUS responsable d'un domaine donné, et s'assurer de son authenticité. Pour le premier problème, eduroam utilise des enregistrements S-NAPTR (RFC 3958) avec le service privé `x-eduroam:radius.tls`, donnant accès à des enregistrements SRV indiquant le serveur à contacter. Par exemple, un cas réel chez RESTENA :

```
%  dig NAPTR restena.lu
...
restena.lu.  21600 IN NAPTR 100 10 "s" "x-eduroam:radius.tls" "" _radsec._tcp.eduroam.lu.
...

% dig SRV _radsec._tcp.eduroam.lu
...
_radsec._tcp.eduroam.lu. 43198 IN SRV 0 0 2083 tld1.eduroam.lu.
_radsec._tcp.eduroam.lu. 43198 IN SRV 10 0 2083 tld2.eduroam.lu.
```

Ces deux enregistrements nous disent que le domaine `restena.lu` a un serveur RADIUS joignable via le nom `_radsec._tcp.eduroam.lu` et que ce nom correspond à deux serveurs, `tld1.eduroam.lu` et `tld2.eduroam.lu`, tous les deux opérant sur le port 2083. (Ce système est décrit dans le RFC 7585.) Le RFC note aussi que le nouveau système de découverte dynamique rendra le système plus souple mais peut-être aussi plus fragile, par suite de la complexité ajoutée.

Le second problème est la vérification de l'authenticité d'un serveur. Après tout, n'importe qui peut mettre des enregistrements NAPTR et SRV dans son domaine et prétendre avoir un serveur RADIUS d'eduroam. La solution a été de réutiliser les AC qui émettent les certificats nécessaires pour RADIUS-sur-TLS. Le serveur RADIUS doit donc vérifier que le pair en face :

— A un certificat émis par une des AC de la PKI d'eduroam (comme eduPKI `<https://www.edupki.org/>`, le nom dans le certificat étant alors `/DC=org/DC=edupki/CN=eduPKI`),
— Que ce certificat contient un OID désignant la politique eduroam eduroam (cf. « *"Delivery of Advanced Network Technology to Europe, "eduPKI""* `<https://www.edupki.org/edupki-pma/edupki-trust-profiles/>` »). Par exemple, aujourd'hui, un `openssl s_client -connect $RADIUS_SERVER:$RADIUS_PORT |openssl x509 -text` sur le serveur RADIUS doit montrer « Policy : 1.3.6.1.4.1.27262.1.13.1.1.1.3 » (je trouve des serveurs qui ont plutôt une vieille version de la politique, comme 1.3.6.1.4.1.27262.1.13.1.1.1.0).

Une dizaine de pays et une centaine de domaines utilisent ce nouveau système de découverte dynamique. Les principaux problèmes sont les problèmes classiques de X.509, gestion des CRL, par exemple.

Le progrès ne va évidemment pas s'arrêter là. eduroam considère aussi deux alternatives :

— DANE (RFC 6698), qui a le gros avantage de permettre de se dispenser complètement de la PKI, en mettant les certificats dans le DNS. La principale limite de DANE, pour le cas d'eduroam, est que DANE permet au client d'authentifier le serveur mais pas (encore) le contraire, alors qu'eduroam a besoin de deux.
— ABFAB (décrit dans l'"*Internet-Draft*" `draft-ietf-abfab-arch`), prometteur mais encore loin d'être complètement spécifié.

Vu le très grand nombre d'utilisateurs d'eduroam, il ne faut pas s'étonner qu'il y ait des incidents, des abus, etc. La section 5 du RFC décrit les mécanismes techniques permettant de gérer ces contrariétés. D'abord, la comptabilité : dans eduroam, c'est une affaire à gérer localement dans le SP (l'université d'accueil). L'IdP (l'université d'origine) n'est pas informée (cf. le « *"eduroam Compliance Statement"* `<http://www.eduroam.org/downloads/docs/eduroam_Compliance_Statement_v1_0.pdf>` »). Mesurer l'activité d'un utilisateur est un pré-requis pour la lutte contre certains abus. D'un autre côté, le SP ne voit pas forcément l'identité du visiteur, juste son domaine (si le visiteur a configuré les identificateurs « anonymes ») ce qui rend difficile la comptabilité. Le RFC 4372 fournit un mécanisme de comptabilité par utilisateur, "*chargeable user identity*" (qui envoie un pseudonyme, pas le « vrai » nom), mais il est très peu implémenté.

En dix ans de fonctionnement, eduroam n'a pas eu d'incident de sécurité sérieux. Cela peut indiquer que l'architecture de sécurité est bonne : il n'y a pas d'accès anonyme, chaque utilisateur est identifié, ce qui peut expliquer leur retenue. Le RFC note que cela veut peut-être dire plutôt que les réseaux utilisés par eduroam ne sont pas assez supervisés et que certains abus passent peut-être inaperçus.

Quoiqu'il en soit, la partie politique de la sécurité est traitée dans « *"eduroam Policy Service Definition"* `<https://www.eduroam.org/downloads/docs/GN3-12-192_eduroam-policy-service-definition_ver28_26072012.pdf>` », notre RFC ne décrivant que les moyens techniques. D'abord, le SP est évidemment libre de bloquer un utilisateur qui abuse. Comment ? Si l'utilisateur a activé les identifiants « anonymes », le SP ne possède que le nom de domaine et l'adresse MAC (sauf coopération avec l'IdP, forcément lente et complexe et donc pas utilisable en urgence). Le SP peut donc, si ses équipements le permettent, bloquer la requête EAP sur la base de ces deux éléments. Mais changer l'adresse MAC est trop facile, et un attaquant déterminé peut donc contourner cette protection. Autre solution, le SP peut attendre la réponse du serveur RADIUS de l'IdP et regarder si elle contient l'attribut

`Chargeable-User-Identity` du RFC 4372, qui contient un identifiant unique de l'utilisateur (unique par SP : le SP envoie son `Operator-Name` - RFC 5580, et l'IdP calcule un identifiant qui dépend du SP, pour les raisons décrites en section 6.1). C'est la meilleure solution car le SP peut alors bloquer cet utilisateur, et uniquement lui, soit en ne répondant pas à ses requêtes, soit en fabriquant des `Access-Reject` RADIUS. S'il n'y a pas de `Chargeable-User-Identity` dans la réponse ? La seule solution restant, et elle est violente, est de bloquer le domaine entier, ce qui aura l'effet de bord de bloquer les visiteurs qui ont le malheur d'avoir un compte au même endroit que l'abuseur.

Et bloquer au niveau de l'IdP, alors ? Lui peut facilement fermer un compte, empêchant ainsi l'utilisateur d'accèder à tout site eduroam. Si le SP envoie son nom dans la requête (avec l'attribut RADIUS `Operator-Name` du RFC 5580), on peut même ne bloquer un compte que pour certains SP, par exemple ceux qui se sont plaints de cet utilisateur. Cela permet des politiques plus fines. Par exemple, imaginons un SP qui interdise le partage d'œuvres culturelles. Un visiteur utilise BitTorrent et le SP demande à l'IdP de le bloquer, alors même que l'utilisation de BitTorrent n'est pas interdite sur le réseau de l'IdP. Bloquer l'utilisateur seulement s'il est sur le réseau du SP râleur permet de satisfaire les politiques du SP et de l'IdP. Sans cette finesse, que permet l'attribut `Operator-Name`, l'IdP risquerait de bloquer complètement un utilisateur, qui n'était peut-être même pas au courant des politiques restrictives du SP.

Passons maintenant à la protection de la vie privée, un des gros morceaux d'eduroam (section 6 du RFC). Si eduroam avait été conçu par Cazeneuve ou Facebook, tout aurait été enregistré tout le temps mais eduroam a au contraire été prévu avec des fortes préoccupations de vie privée. D'abord, on l'a vu, la possibilité de NAI « anonymes » n'indiquant que le domaine ("*outer identities*"). Ce système empêche, par exemple, deux SP de déterminer si deux visiteurs de leurs campus, ayant le même IdP, sont une seule et même personne. Excellent pour la vie privée, mais cela rend compliqué l'attribution d'une éventuelle mauvaise action. D'où l'ajout de l'attribut `Chargeable-User-Identity` dans eduroam. Étant calculé par l'IdP et différent pour chaque SP, il empêche les collusions de SP entre eux, mais permet de remonter les traces d'un éventuel abuseur.

À noter que le domaine, lui, est visible du SP, des différents serveurs RADIUS qui relaient (ils en ont besoin pour le routage) et, si on n'utilise pas TLS, de tous les indiscrets sur le trajet. Si l'IdP est une très grosse université, cela peut frustrer l'observateur mais, dans le cas de petits établissements, cela peut l'aider. Il serait difficile de résoudre ce problème sans refaire tout eduroam.

Par défaut, l'IdP ne sait pas où se trouvent ses utilisateurs, puisque son serveur RADIUS ne voit que les requêtes du serveur RADIUS supérieur (typiquement, celui du pays). Cet enchaînement de serveurs RADIUS, peut sembler lourd, et difficile à déboguer (le point discuté en section 3). Mais il est avantageux, question vie privée.

Par contre, cette protection de l'utilisateur contre son propre IdP peut être affaiblie si on utilise la découverte dynamique (l'IdP voit les requêtes DNS, cf. RFC 7626), et évidemment si le SP envoie son nom dans la requête RADIUS (attribut `Operator-Name`).

Et la sécurité, à part cette question de vie privée ? Il faut bien sûr relire les sections Sécurité des normes RADIUS, EAP et 802.1X, plus la section 7 de notre RFC, qui discute les problèmes plus spécifiques à eduroam. D'abord, la sécurité de bout en bout repose complètement sur EAP. Si un Homme du Milieu réussit à se faire passer pour le serveur EAP de l'IdP, il a table ouverte. Le client EAP doit donc exiger une authentification du serveur, et le serveur doit faire ce qu'il faut pour être authentifié (par exemple, publier le nom qui doit apparaitre dans le certificat).

En théorie, cette authentification du serveur résout complètement le problème. En pratique, il reste des pièges. Par exemple, certains clients EAP ne permettent de vérifier que l'AC, pas le nom (le sujet

du certificat). Tout attaquant qui peut obtenir un certificat quelconque de la même AC peut donc se faire passer pour le serveur EAP. Il vaut donc mieux ne se fier qu'à des AC spécifiques à eduroam. D'autres clients EAP pratiquent le TOFU : ils se fient au premier certificat et le mémorisent pour la suite. Si un utilisateur était chez un point d'accès pirate la première fois, c'est le certificat de l'attaquant qui sera cru. Les utilisateurs doivent donc veiller à configurer leurs clients eduroam proprement (un outil existe pour cela <https://cat.eduroam.org>). Pire, certains logiciels permettent à l'utilisateur de complètement sauter la vérification (« "*No CA certificate is required*" » sur NetworkManager, par exemple). L'utilisateur qui sélectionnerait cette option pour se faciliter la vie se rendrait très vulnérable aux attaques de l'Homme du Milieu. Dernier piège d'authentification : en utilisant les identités « anonymes » (NAI ne comportant que le nom du domaine), rien n'empêche un utilisateur malveillant de placer dans la session EAP un NAI indiquant un autre domaine. Si le serveur RADIUS de l'IdP indiqué dans le NAI « anonyme » accepte de relayer les requêtes, l'utilisateur pourra se faire authentifier, et le SP ne connaitra alors pas le vrai domaine d'origine. Cela diminue sérieusement la responsabilité de l'utilisateur et lui permet de brouiller ses traces. Voilà pourquoi les serveurs RADIUS d'eduroam ne doivent pas relayer ces requêtes.

Comme tout service sur l'Internet, eduroam est évidemment susceptible de recevoir des attaques par déni de service. Sur les points d'accès Wi-Fi, on peut voir des serveurs DHCP pirates (cf. RFC 7610), des émetteurs de RAcailles (cf. RFC 6105), etc. eduroam est moins vulnérable que le point d'accès Wi-Fi typique avec portail captif, car l'authentification préalable est nécessaire (ce qui limite les attaques par un client tentant d'épuiser la réserve d'adresses IP en faisant des requêtes DHCP répétées) et le Wi-Fi est chiffré (par WPA2), ce qui limite l'usurpation ARP.

Certaines attaques par déni de service spécifiques à eduroam existent. Par exemple, un attaquant situé physiquement à portée d'un point d'accès Wi-Fi eduroam peut envoyer de manière répétée des demandes d'accès en indiquant un domaine situé dans un TLD différent : cela obligera plusieurs serveurs RADIUS, y compris ceux de la racine, à traiter sa demande, et une négociation EAP à se lancer. Heureusement pour eduroam, EAP est synchrone, donc un attaquant donné ne peut lancer qu'une requête à la fois, pour un domaine donné. eduroam reçoit aujourd'hui des centaines de milliers de requêtes d'authentification réussies par jour (et bien plus qui échouent) et un attaquant qui voudrait tuer eduroam avec un afflux de requêtes devrait donc y consacrer de gros moyens. En tout cas, de telles attaques n'ont pas encore été vues.

À noter qu'il existe aussi des attaques involontaires. La plupart des demandes d'authentification qui échouent concernent des comptes qui étaient valides mais ne le sont plus (étudiant qui a terminé ses études, par exemple). Il est donc probable que les gens dont les comptes sont expirés aient oublié de déconfigurer leur logiciel et que celui-ci envoie toujours des demandes dès qu'il passe près d'un point d'accès eduroam. Il y a donc un « "*botnet*" de fait » composé de toutes ces machines d'ex-utilisateurs. Le changement de machine par l'utilisateur, lorsque l'ancienne est trop vieille, ne résout pas le problème : aujourd'hui, surtout sur les ordiphones, les données d'authentification sont sauvegardées dans le "*cloud*" (pour que Google et la NSA y aient plus facilement accès) et sont remises sur le nouveau jouet lorsque l'ancien est remplacé. Le "*botnet*" des anciens d'eduroam ne fait donc que grossir. Il n'y a pas encore de solution propre à ce problème, peut-être améliorer les logiciels client pour qu'ils arrêtent d'envoyer automatiquement des requêtes après N échecs sur une période de X jours.

Un point sur lequel le RFC passe assez rapidement est le problème d'interopérabilité qui peut se poser quand deux auteurs de logiciel n'ont pas compris de la même façon ces normes complexes qu'il faut programmer correctement. Un exemple d'alerte est « "*RADIUS Attribute Issues regarding RFC5580 (Operator-Name and others) with several RADIUS servers (including Microsoft IAS and NPS)*" <https://www.eduroam.org/downloads/docs/advisory/eduroamOT-admin-advisory-004.pdf> », un amusant problème de normalisation avec le RFC 5580, RFC qui a été souvent cité précédemment.

Si vous êtes intéressé par ce système, je vous recommande la lecture du site Web d'eduroam <https://www.eduroam.org/>.