RFC 7465 : Prohibiting RC4 Cipher Suites

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 19 février 2015

Date de publication du RFC : Février 2015

https://www.bortzmeyer.org/7465.html

Fini, RC4, au moins dans TLS ! Cet algorithme de cryptographie, qui était secret à l'origine et prétendait être ainsi plus sûr, a fait l'objet de tellement d'attaques cryptanalytiques réussies qu'il ne doit **plus** être utilisé dans le protocole TLS.

Normalisé dans le RFC 5246 [1], TLS sécurise un grand nombre de protocoles TCP/IP, le plus connu étant HTTP. Comme tous les protocoles IETF utilisant la cryptographie, il n'est pas lié à **un** algorithme de cryptographie particulier. En effet, les progrès de la cryptanalyse font que les algorithmes qui semblaient sûrs à une époque ne durent pas forcément éternellement. Il est donc crucial de permettre l'arrivée de nouveaux algorithmes, et le retrait des vieux (comme cela a été fait pour MD5, cf. RFC 6151). C'est ce qu'on nomme l'**agilité cryptographique**. Dans TLS, la liste des algorithmes acceptés est enregistrée à l'IANA <https://www.iana.org/assignments/tls-parameters/tls-parameters.xml#tls-parameters-4> et ceux acceptés par le client TLS sont transmis au début de la négociation, dans le message `ClientHello` (RFC 5246, section 7.4.1.2). Parmi ceux proposés, le serveur en choisira un et l'indiquera dans le message `ServerHello` (RFC 5246, section 7.4.1.3). (En fait, c'est un peu plus compliqué, le client transmet des **suites**, chaque suite contenant plusieurs algorithmes, notamment un asymétrique, et un symétrique comme l'est RC4. Par exemple, `TLS_ECDH_ECDSA_WITH_RC4_128_SHA` est l'algorithme asymétrique ECDSA avec RC4) Depuis ce RFC 7465, les suites cryptographiques utilisant RC4 ne doivent **plus** apparaître dans ces messages.

RC4 n'a pas été normalisé (il était secret au début, un exemple illustrant bien la vanité de la sécurité par l'obscurité) mais a été documenté dans le livre de Schneier, « *"Applied Cryptography"* <https://www.schneier.com/book-applied.html> » (à partir de la deuxième édition). Parmi les attaques réussies contre RC4 :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5246.txt

— Paul, G. et S. Maitra, « "*Permutation after RC4 Key Scheduling Reveals the Secret Key*" <https://eprint.iacr.org/2007/208.pdf> »
— Mantin, I. et A. Shamir, « "*A Practical Attack on Broadcast RC4*" <http://saluc.engr.uconn.edu/refs/stream_cipher/mantin01attackRC4.pdf> »
— Fluhrer, S., Mantin, I., et A. Shamir, « "*Weaknesses in the Key Scheduling Algorithm of RC4*" <http://www.crypto.com/papers/others/rc4_ksaproc.pdf> »
— AlFardan, N., Bernstein, D., Paterson, K., Poettering, B., et J. Schuldt, « "*On the security of RC4 in TLS and WPA*" <https://www.usenix.org/conference/usenixsecurity13/security-rc4-tls> » Usenix en 2013

Toutes ces attaques ne sont pas forcément exploitables de manière réaliste. Mais la cryptanalyse progresse tous les jours. Si l'attaque est **un peu** trop dure aujourd'hui, elle sera possible demain et triviale après-demain (sans compter le fait qu'en cryptanalyse, tout n'est pas publié, certaines organisations ne disent pas ce qu'elles font). Notre RFC estime qu'aujourd'hui, ces attaques sont presque utilisables en pratique. Il est donc temps de virer RC4.

La section 2 est donc simple et courte : le client TLS ne doit plus indiquer de suite cryptographique utilisant RC4 et, s'il le fait, le serveur ne doit plus les sélectionner. Si la totalité des suites proposées par le client utilise RC4, le serveur doit rejeter la connexion, en utilisant l'alerte `insufficient_security(71)` (cf. RFC 5246, section 7.2). Ce dernier point avait fait l'objet d'un débat dans le groupe de travail, car il revient à rejeter complètement certaines mises en œuvre de TLS (rappelons que certains sont toujours incapables de parler les versions 1.1 et 1.2 de TLS, malgré les failles de sécurité que cela cause). Toutefois, ce cas de logiciels ne gérant que RC4 semble rare dans la nature.

Par contre, le registre IANA <https://www.iana.org/assignments/tls-parameters/tls-parameters.xml#tls-parameters-4> ne dispose pas d'un moyen pour indiquer cette obsolescence de RC4 donc des programmeurs / administrateurs système distraits ne feront peut-être pas attention à ce RFC. Diffusez-le largement !

Un exemple de faille de sécurité récente (publiée un mois après le RFC) concernant RC4 : la faille Invariance Weakness <http://www.imperva.com/docs/HII_Attacking_SSL_when_using_RC4.pdf>. Un autre exemple est présenté sur Numerous Occurrence MOnitoring & Recovery Exploit <http://www.rc4nomore.com/>.