RFC 9116 : A File Format to Aid in Security Vulnerability Disclosure

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 avril 2022

Date de publication du RFC : Avril 2022

https://www.bortzmeyer.org/9116.html

Lorsqu'on repère un problème de sécurité sur un site Web ou, plus généralement, dans l'informatique d'une organisation, de deux choses l'une : soit on est un méchant et on exploite l'information à son profit, soit on fait partie des bons et on va essayer de faire en sorte que le problème soit résolu. Une solution évidente est de signaler le problème à l'organisation qui gère le site Web. Mais comment ? Entre le formulaire de contact cassé qui refuse votre adresse de courrier en prétendant qu'elle est illégale, et le message envoyé à un humain sous-payé et sous-qualifié qui cliquera tout de suite sur « Problème résolu » pour faire grimper ses chiffres de résolution, prévenir quelqu'un de compétent et de responsable est souvent un parcours du combattant ! D'où le choix de ce RFC de proposer encore une nouvelle méthode : un fichier à un endroit bien connu sur le site Web, au format structuré, et contenant les informations essentielles, le `security.txt`.

Comme beaucoup de techniques utilisées sur l'Internet, ce format a été déployé bien avant d'être officiellement décrit dans un RFC. On trouve aujourd'hui de tels fichiers sur plusieurs sites, y compris sur ce blog `</.well-known/security.txt>` (regardez-le tout de suite si vous voulez avoir une idée de ce qu'on y met).

La section 1 du RFC décrit plus en détail le problème. Toute personne qui a déjà essayé de signaler un problème de sécurité à une organisation reconnaitra ses propres mésaventures. Le RFC commence par rappeler que trouver un contact, quoique très difficile, n'est pas tout : il faut aussi s'informer sur la politique de traitement des signalements de cette organisation, car plus d'un citoyen ayant voulu signaler une vulnérabilité s'est retrouvé accusé d'être un vilain pirate, et a parfois réellement été poursuivi en justice. Et enfin il faut s'informer sur les moyens de communications sécurisés puisque, par définition, on va transmettre des informations délicates. Est-ce qu'on peut utiliser PGP (RFC 4880 [1]), par exemple ? Pour ce qui est des contacts, il existe en théorie plusieurs solutions :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc4880.txt`

— Le RFC 2142 exige un certain nombre d'adresses de courrier pour chaque domaine, comme `security@LE-DOMAINE` pour les signalements de sécurité. Il y a peu d'organisations où cette adresse fonctionne techniquement et, même quand c'est le cas, elle n'est pas forcément lue.
— Les RFC 3013 (section 2), RFC 2350 (section 3.2) et RFC 2196 (section 5.2) prévoient des adresses de contact pour différents types d'acteurs. (Même remarque que sur le point précédent.)
— Les bases des registres (de noms de domaine et d'adresses IP) contiennent également des adresses de contacts, en général accessibles via whois (cf. RFC 7485) ou RDAP. (Même remarque que sur le point précédent, en ajoutant que ces bases sont purement déclaratives, que leur exactitude est faible et baisse avec le temps.)
— Aucune de ces solutions ne donne facilement accès à la politique de divulgation de l'organisation. Et, surtout, même si le RFC garde un silence poli sur la question, toute personne qui a essayé de contacter une organisation sait bien que ces adresses de contact sont souvent erronées, dépassées, ou bien arrivent chez des gens qui s'en f...ent.

Le dernier point est à la fois une motivation important pour créer une nouvelle solution et une des principales critiques qui avaient été formulées à l'encontre du projet `security.txt` lors de la discussion à l'IETF : pourquoi est-ce que ça serait mieux avec un nouveau format ? Si une organisation est assez négligente pour ne pas tenir à jour les informations présentes chez le registre de son nom de domaine, comment espérer qu'elle tienne à jour son `security.txt` ? Il n'y a évidemment pas de certitude à ce sujet, juste l'espoir qu'un autre format, et surtout un autre mécanisme de mise à jour augmentera les chances que l'information soit correcte.

Le `security.txt` est un fichier analysable par un programme (mais quand même lisible par un humain), et qui permet d'indiquer tout ce qui est nécessaire, par exemple au chercheur de vulnérabilités qui a trouvé quelque chose. Il vise à signaler les **vulnérabilités** (pas encore exploitées), pas les **incidents** (car un attaquant qui a réussi a peut-être modifié le fichier `security.txt`).

La spécification, maintenant (section 2 du RFC). Le `security.txt` est un simple fichier texte qui doit être déposé à un endroit précis du site Web, `/.well-known` (RFC 8615, et `security.txt` a été enregistré dans le registre des URL bien connus <`https://www.iana.org/assignments/well-known-uris/well-known-uris.xml#well-known-uris-1`>). Son type doit être `text/plain` et il doit être accessible en HTTPS, pour d'évidentes raisons de sécurité. Il doit être en Unicode, utilisant le profil du RFC 5198. Analysable par un programme, il doit se conformer à la grammaire spécifiée dans le RFC. Il comporte plusieurs champs, chacun sur une ligne, et ayant un nom et une valeur. La plupart des champs sont optionnels. Le champ le plus connu est `Contact`, qui est obligatoire, et voici un exemple :

```
Contact: mailto:stephane%2Bsecurity@bortzmeyer.org
```

(Vous avez reconnu l'encodage de l'URI spécifié dans la section 2.1 du RFC 3986. `%2B` est le signe plus.) Les lignes commençant par un croisillon sont des commentaires.

Le RFC recommande que les `security.txt` soient signés avec OpenPGP (RFC 4880). Naturellement, comme avec n'importe quelle signature, sa valeur dépend de si le vérificateur connait de manière certaine la clé publique qui a été utilisée.

Un certain nombre de champs sont définis aujourd'hui, la plupart optionnels. Parmi eux :
— `Acknowledgments` : un lien vers une page des remerciements, pour que la personne ayant l'intention de signaler un problème sache qu'elle pourra être remerciée publiquement. Question carotte pour les chercheurs de vulnérabilités, il y a aussi un champ `Hiring` [Caractère Unicode non montré [2]].

2. Car trop difficile à faire afficher par LaTeX

— `Canonical` : l'URL officiel de ce `security.txt`. Cela peut permettre de détecter certaines erreurs de configuration (`security.txt` copié sur le mauvais site).
— `Contact` : sans doute le champ le plus important, et un des rares qui soit obligatoire. Il contient un URI indiquant comment contacter quelqu'un de responsable, à qui confier le problème de sécurité. L'intérêt de définir la valeur de ce champ comme un URI est que cela permet de ne pas se limiter à une méthode de contact particulière. Puisque c'est un URI, les adresses de courrier électronique doivent être mises sous forme d'URI `mailto:` (RFC 6068). Pareil pour les numéros de téléphone (RFC 3966). Le champ `Contact` peut être répété, afin d'avoir plusieurs mécanismes de signalement (dans ce cas, l'ordre est important, la méthode recommandée est la première).
— `Encryption` : indique la clé de chiffrement à utiliser pour contacter. C'est encore un URI (qui peut être un URI de plan `dns:`, pour les clés stockées dans le DNS du RFC 7929).
— `Expires` : ce champ est obligatoire et indique (au format du RFC 3339) la date limite d'utilisation de ce `security.txt` (pour éviter que de vieux fichiers pas maintenus restent utilisés).
— `Policy` : un lien vers la page décrivant la politique de l'organisation en matière de signalement de vulnérabilités. Rappelez-vous qu'un problème de beaucoup de personnes qui ont détecté une vulnérabilité est le risque que le signalement de celle-ci leur vaille des ennuis juridiques, en raison de la politique de beaucoup d'organisations qui est de nier les problèmes et de poursuivre devant les tribunaux ceux qui les signalent.
— `Preferred-Languages` : des étiquettes de langue (RFC 5646) indiquant dans quelles langues on préfère recevoir les rapports (et là, je pense à cet informaticien étatsunien qui s'était étonné publiquement qu'un webmestre chinois ne répondait pas à ses signalements de vulnérabilité faits en anglais…).

Les champs figurent dans un registre IANA <`https://www.iana.org/assignments/security-txt-fields/security-txt-fields.xml#security-txt-fields`>. D'autres champs pourront être rajoutés à ce registre dans le futur, en suivant la politique « Examen par un expert » (RFC 8126). Pour faciliter ces futurs ajouts, il faut ignorer les champs qu'on ne connait pas. Un exemple réel de `security.txt` ? Regardez celui de ce blog <`/.well-known/security.txt`>.

Le `security.txt` est publié via le site Web mais ne s'applique pas forcément qu'au Web, il peut aussi servir pour l'organisation en général. Par contre, il ne s'applique pas aux sous-domaines : un `security.txt` sur `eu.org` n'est pas valable pour `exodus-privacy.eu.org`.

Vu le but de ce `security.txt`, il n'est pas étonnant que la section 5 du RFC, consacrée à la sécurité, soit si détaillée, d'autant plus qu'elle a fait l'objet de vigoureuses discussions à l'IETF. Je n'en cite ici qu'une partie, n'hésitez pas à lire toute cette section 5. D'abord, l'objection la plus évidente : si le site Web a été piraté, le `security.txt` ne peut plus être considéré comme digne de confiance. Les signalements risquent d'être perdus ou même envoyés à l'attaquant qui aura mis son adresse dans le `security.txt`. C'est vrai, mais c'est le cas de toutes les informations de contact. Par exemple, si le compte de l'organisation auprès du BE a été piraté (comme dans l'attaque moyen-orientale de 2018 <`https://www.bortzmeyer.org/attaques-noms-domaine-explications.html`>), les informations attachées au nom de domaine, et récupérées par whois ou RDAP, sont également suspectes. Le RFC recommande que les organisations qui ont un `security.txt` le supervisent automatiquement et vérifient notamment le champ `Canonical`. Et, bien sûr, que le `security.txt` soit signé. Les personnes qui signalent une vulnérabilité ont tout intérêt à vérifier le `security.txt`. Et puis surtout, `security.txt` est conçu pour la **réponse à vulnérabilité**, pas la **réponse à incident**. L'utiliser pour signaler « vous avez une faille » est raisonnable, s'en servir pour signaler « vous êtes piraté » l'est moins.

Comme toutes les informations mises en ligne (cf. l'exemple des informations sociales sur un nom de domaine…), le `security.txt` peut être faux, soit dès le début, soit parce qu'il n'a pas été maintenu correctement. En tapant cet article, je regardais le `security.txt` d'une entreprise française de sécurité informatique et le champ `Encryption` contenait un URL qui pointait…vers un 404. Dans une très grosse entreprise française travaillant sur de la haute technologie, notamment en sécurité, un URL dans

le `security.txt` donne...403! Au même endroit, la signature du `security.txt` est incorrecte...Le champ `Expires` permet de détecter les `security.txt` trop vieux et pas maintenus mais le problème est vaste et on peut s'attendre à se casser souvent le nez sur des informations incorrectes. Comme pour toutes les informations en ligne, les organisations qui publient un `security.txt` devraient s'assurer, dans leurs procédures, qu'il est maintenu à jour.

Certains croient qu'ils ont le droit de se livrer à des tests d'attaques sur tout site Web trouvé sur l'Internet. C'est évidemment faux, de même qu'on n'a pas le droit dans le rue de tenter de crocheter toutes les serrures pour voir lesquelles sont vulnérables. Même l'existence d'un `security.txt` ne vaut pas autorisation de tester le site. Le champ `Policy` peut indiquer si des tests d'attaque sont autorisés et dans quelles conditions. Sinon, on doit se limiter aux failles découvertes dans le cours de l'utilisation normale du site (ceci n'est pas un conseil juridique : la loi est compliquée et dépend du pays).

Naturellement, comme toujours lorsqu'on publie une adresse de courrier, elle va recevoir du spam. Plus gênant, le fait que le `security.txt` soit analysable par un programme pourrait amener certains bots qui font des soi-disant tests de sécurité à envoyer des messages automatiques de faible valeur <https://news.ycombinator.com/item?id=19152145>. (Par exemple, j'ai vu un bot qui regardait la version d'Apache et envoyait automatiquement un courrier si la version lui semblait trop vieille. Ce genre d'avertissements mécaniques n'a aucune valeur, notamment parce que certains systèmes d'exploitation bouchent les failles de sécurité sans changer la version et, surtout, parce qu'il y a peu de chance qu'un logiciel puisse faire avec succès quelque chose d'aussi délicat qu'une analyse de sécurité ; les logiciels sont utiles pour une première reconnaissance, mais il ne faut pas envoyer de message d'avertissement avant qu'un humain compétent n'ait vérifié.)

Voilà, nous avons fait le tour du RFC. Si vous êtes responsable de la sécurité d'une organisation, à vous de vous mettre au travail pour concevoir et documenter une politique de signalement des failles de sécurité (c'est le gros du travail), de rédiger un `security.txt` (c'est trivial) et de faire en sorte qu'il soit maintenu (ce n'est pas évident). Si vous avez trouvé une faille de sécurité dans un site Web ou ailleurs chez une organisation, pensez à regarder si elle a un `security.txt` mais ne l'utilisez pas aveuglément, comparez avec d'autres sources d'information. Pour vous instruire, vous pouvez regarder le site Web du projet <https://securitytxt.org/>. Il comprend notamment un bon formulaire pour aider à fabriquer son `security.txt` (mais rappelez-vous que ce qui est difficile, c'est l'élaboration de la politique). Ce site contient également une liste de logiciels qui peuvent aider <https://securitytxt.org/projects>. Le moteur de recherche Shodan lit les `security.txt` et les affiche proprement mais, avec le "*virtual hosting*", comme Shodan travaille par adresse IP, ça ne marche pas souvent. Sinon, YesWeHack <https://www.yeswehack.com/> fournit une extension aux navigateurs Web <https://blog.yeswehack.com/vulnerability-coordination/showcasing-your-vulnerability-disclosure> pour afficher le `security.txt` (pas très utile, je trouve, elle se contente d'afficher le fichier tel quel, sans vraie valeur ajoutée).

Le `security.txt` est-il obligatoire ? Cela dépend de votre environnement. Le DHS étatsunien le recommande <https://cyber.dhs.gov/bod/20-01/#can-we-use-a-securitytxt-file> pour les organismes qui dépendent de lui. En France, l'arrêté du 18 septembre 2018 portant approbation du cahier des clauses simplifiées de cybersécurité <https://www.legifrance.gouv.fr/eli/arrete/2018/9/18/ECOP1825228A/jo/texte>, dans son article 6.4, dit que « Afin que ces signalements soient effectifs et efficaces, les conventions d'usage en cybersécurité sont respectées (security.txt, abuse@). Dans tous les cas, il faut moins d'une minute pour trouver le point d'entrée approprié du signalement. ».

Terminons par un tour des `security.txt` existants (mes commentaires concernent l'état de ce fichier en août 2021 ; il a pu changer depuis). Commençons par un exemple très simple, mais correct, celui de la société Cyberzen <https://www.cyberzen.com/> :

```
% curl https://www.cyberzen.com/.well-known/security.txt
Contact: contact@cyberzen.com
Expires: Sun, 31 Dec 2023 23:59 +0200
Preferred-Languages: fr, en
```

Ensuite le registre de `.be` a un très bon `security.txt`, très complet, avec commentaires :

```
%  curl https://www.dnsbelgium.be/.well-known/security.txt
# Our security address
Contact: mailto:csirt@dnsbelgium.be?subject=rdp_dnsbelgium.be
# Our OpenPGP key
Encryption: https://www.dnsbelgium.be/.well-known/pgp-key.txt
# Canonical URL
Canonical: https://www.dnsbelgium.be/.well-known/security.txt
# Our security policy
Policy: https://www.dnsbelgium.be/responsible-disclosure-policy
Preferred-Languages: en, nl, fr
Expires: Mon, 1 Nov 2021 00:00:00 +0100
```

Autre exemple, Google a un `security.txt` :

```
Contact: https://g.co/vulnz
Contact: mailto:security@google.com
Encryption: https://services.google.com/corporate/publickey.txt
Acknowledgements: https://bughunter.withgoogle.com/
Policy: https://g.co/vrp
Hiring: https://g.co/SecurityPrivacyEngJobs
```

Le champ `Expires`, pourtant obligatoire, manque encore (il a été ajouté dans les dernières révisions du projet de spécification).

En France, la Sécu en a un et, contrairement à celui de Google, il est signé avec OpenPGP :

```
% curl https://www.ameli.fr/.well-known/security.txt
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Contact: mailto:abuse@assurance-maladie.fr
Encryption: https://raw.githubusercontent.com/AssuranceMaladieSec/abuse/master/abuse-gpg-public-key.txt
Policy: https://assurancemaladiesec.github.io/abuse/reporting/
Acknowledgments: https://assurancemaladiesec.github.io/abuse/thanks/
Preferred-Languages: fr,en
-----BEGIN PGP SIGNATURE-----

iQIzBAEBCgAdFiEEDSx9bqnSlmkiIRXbSDqGYCymPFIFAl4V9/cACgkQSDqGYCym
PFIB7Q/9EI7fNOfyoCqnEH4chiTIW8fx32ldnlaE6WTgdMeQJmkyJrhd2osPAV/j
...
```

Ah, profitons-en pour vérifier la signature (la clé publique, curieusement, est hébergé chez un tiers, GitHub) :

```
% wget https://www.ameli.fr/.well-known/security.txt
...
2020-04-08 18:41:10 (6.87 MB/s) - 'security.txt' saved [1189]

% wget https://raw.githubusercontent.com/AssuranceMaladieSec/abuse/master/abuse-gpg-public-key.txt
...
2021-08-12 09:03:52 (11,1 MB/s) - 'abuse-gpg-public-key.txt' saved [3159/3159]

% gpg --import abuse-gpg-public-key.txt
gpg: key 483A86602CA63C52: public key "Abuse Assurance Maladie <abuse@assurance-maladie.fr>" imported
gpg: Total number processed: 1
gpg:               imported: 1

% gpg --verify security.txt
gpg: Signature made Wed Jan  8 16:40:39 2020 CET
gpg:                using RSA key 0D2C7D6EA9D29669222115DB483A86602CA63C52
gpg: Good signature from "Abuse Assurance Maladie <abuse@assurance-maladie.fr>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 0D2C 7D6E A9D2 9669 2221  15DB 483A 8660 2CA6 3C52
```

C'est parfait, tout fonctionne (et c'est documenté <https://assurancemaladiesec.github.io/security-txt-implementation-at-assurance-maladie.html>). Autre exemple en France, le service Flus <https://flus.fr/>, dont le `security.txt` est décrit dans un bon article <https://flus.fr/securite>. Voici le fichier :

```
% curl https://flus.fr/.well-known/security.txt
Contact: https://flus.fr/contact
Contact: security@flus.io
Expires: Fri, 01 Apr 2022 00:00 +0000
Policy: https://flus.fr/securite
Acknowledgments: https://flus.fr/securite
Canonical: https://flus.fr/.well-known/security.txt
Preferred-Languages: fr, en
```

OpenSSL a bien sûr un `security.txt`. À une époque, la signature PGP était incorrecte…

```
% gpg --verify security.txt.asc  security.txt
gpg: Signature made Thu Jan  4 04:22:26 2018 CET
gpg:                using RSA key EFC0A467D613CB83C7ED6D30D894E2CE8B3D79F5
gpg: BAD signature from "OpenSSL OMC <openssl-omc@openssl.org>" [unknown]
```

Cela illustre un problème commun à **tous** les mécanismes de publication d'information de contact : l'information n'est pas facile à maintenir et tend à se dégrader avec le temps.

Quelques lectures supplémentaires qui peuvent être intéressantes :

— Si vous êtes prêt à payer, la norme ISO ISO.29147.2018 parle de « "*Security techniques - Vulnerability disclosure*" ».
— CMU a un « "*CERT Guide to Coordinated Vulnerability Disclosure*" <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=503330> ».
— Le projet a plein de ressources si vous travaillez dans la détection et signalement de vulnérabilités.
— Autres articles en français sur le `security.txt` : celui de Bruno Kerouanton <https://bruno.kerouanton.net/blog/2020/08/11/le-fichier-security-txt/> et celui de y0no <https://y0no.fr/posts/decouverte-security-txt/>.
— Et un article de l'ANSSI sur le signalement de vulnérabilités <https://www.ssi.gouv.fr/en-cas-dincident/vous-souhaitez-declarer-une-faille-de-securite-ou-une-vulnerabi
>.