Groupe de travail réseau
Request for Comments : 2943
Catégorie : Encours de normalisation
Traduction Claude Brière de L'Isle

R. Housley, T. Horting, P. Yee
SPYRUS
septembre 2000

Authentification Telnet avec DSA

Résumé
Le présent document définit un mécanisme d'authentification Telnet qui utilise l'agorithme de signatures numérique (DSA, *Digital Signature Algorithm*) [FIPS186]. Il s'appuie sur l'option d'authentification Telnet [RFC2941].

1. Noms et codes des commandes

AUTHENTICATION : 37

Commandes d'authentification :

IS	0
SEND	1
REPLY	2
NAME	3

Types d'authentification :

DSS	14

Modificateurs :

AUTH_WHO_MASK	1
AUTH_CLIENT_TO_SERVER	0
AUTH_SERVER_TO CLIENT	1
AUTH_HOW_MASK	2
AUTH_HOW_ONE_WAY	0
AUTH_HOW_MUTUAL	2
ENCRYPT_MASK	20
ENCRYPT_OFF	0
ENCRYPT_USING_TELOPT	4
ENCRYPT_AFTER_EXCHANGE	16
ENCRYPT_RESERVED	20
INI_CRED_FWD_MASK	8
INI_CRED_FWD_OFF	0
INI_CRED_FWD_ON	8

Commandes de sous option :

DSS_INITIALIZE	1
DSS_TOKENBA	2
DSS_CERTA_TOKENAB	3
DSS_CERTB_TOKENBA2	4

2. Extensions Telnet de sécurité

Telnet, en tant que protocole, n'a pas de concept de sécurité. Sans option négociée, il passe simplement les caractères entre les terminaux réseau virtuels (NVT, *network virtual terminal*) représentés par les deux processus Telnet. Dans son utilisation la plus courante comme protocole pour l'accès à des terminaux distants (accès TCP 23) Telnet se connecte à un serveur qui exige une authentification au niveau de l'utilisateur par un nom d'utilisateur et un mot de passe en clair ; le serveur ne s'authentifie pas lui-même envers l'usager.

L'option d'authentification Telnet assure l'authentification de l'utilisateur et celle du serveur. L'authentification de l'utilisateur remplace ou augmente le mécanisme de mot de passe de l'hôte. L'authentification du serveur est normalement faite en conjonction avec l'authentification de l'utilisateur.

Afin de prendre en charge ces services de sécurité, les deux entités Telnet doivent d'abord négocier leur volonté de prendre en charge l'option d'authentification Telnet. Si il y a accord pour prendre en charge cette option, les parties sont alors capables d'effectuer les négociations de sous option sur le protocole d'authentification à utiliser, et éventuellement sur le nom d'utilisateur distant à utiliser pour les vérifications d'autorisation.

La négociation de l'authentification et des paramètres survient au sein d'une série d'échanges non limitée. Le serveur propose une liste ordonnée par préférences des types (mécanismes) d'authentification qu'il prend en charge. En plus de la liste des mécanismes qu'il accepte, le serveur qualifie chaque mécanisme avec un modificateur qui spécifie si l'authentification est unilatérale ou mutuelle, et dans quelle direction l'authentification est à effectuer. Le client choisit un mécanisme de cette liste et répond au serveur en indiquant son choix et le premier ensemble de données d'authentification nécessaire pour le type d'authentification choisi. Le serveur et le client procèdent alors au nombre d'itérations nécessaires pour arriver à l'authentification demandée.

3. Utilisation de l'algorithme de signature numérique (DSA)

DSA *(Digital Signature Algorithm)* est aussi appelé norme de signature numérique (DSS, *Digital Signature Standard*) et les deux noms sont utilisés de façon interchangeable. Le présent document spécifie une méthode dans laquelle DSA peut être utilisé pour réaliser certains services de sécurité lorsque ils sont utilisés en conjonction avec l'option d'authentification Telnet. SHA-1 [FIPS180-1] est utilisé avec DSA [FIPS186].

DSA peut fournir l'authentification aussi bien unilatérale que mutuelle. Du fait de la nature caractère par caractère de Telnet, il ne convient pas très bien pour l'application de services de seule protection de l'intégrité, et donc l'utilisation du profil de DSA fournit l'authentification mais n'assure pas l'intégrité de session. La présente spécification suit les jetons et échanges définis dans la PUB 196 FIPS du NIST [FIPS196], Norme pour les mécanismes d'authentification d'entités par clés de chiffrement publiques incluant l'appendice A sur le codage ASN.1 des messages et des jetons. Toutes les données qui sont couvertes par une signature numérique doivent être codées en utilisant les règles de codage distinctif (DER, *Distinguished Encoding Rules*). Cependant, d'autres données peuvent utiliser les règles de codage de base (BER, *Basic Encoding Rules*) ou les règles de codage distinctif (DER) [X.208].

3.1 Authentification unilatérale avec DSA

L'authentification unilatérale doit être faite de client à serveur. On donne ci-après les étapes nécessaires du protocole pour effectuer l'authentification DSA comme spécifiée dans FIPS PUB 196 dans le cadre de l'option d'authentification Telnet. En cas de défaillances, les codes de retour suivent ceux spécifiés dans l'option d'authentification Telnet. On n'en fait pas la liste ici car ils ne varient pas selon le mécanisme utilisé. La norme FIPS PUB 196 emploie un ensemble d'échanges qui sont effectués pour assurer l'authentification. Chaque échange emploie divers champs et jetons, dont certains sont facultatifs. De plus, chaque jeton a plusieurs sous champs qui sont facultatifs. Un sous-ensemble des champs et sous champs a été retenu pour définir la conformité à la présente norme. Les jetons sont codés en ASN.1, comme défini dans l'Appendice A de FIPS PUB 196, et chaque jeton est désigné pour indiquer la direction dans laquelle il s'écoule (par exemple, le JetonBA s'écoule de la partie B à la partie A). Toutes les données qui sont couvertes par une signature numérique doivent être codées en utilisant les règles de codage distinctives (DER, *Distinguished Encoding Rules*). Les données qui ne sont pas couvertes par une signature numérique peuvent utiliser les règles de codage de base (BER, *Basic Encoding Rules*) ou de DER [X.208]. La Figure 1 illustre les échanges de l'authentification unilatérale.

Durant l'authentification, le client peut fournir au serveur le nom d'utilisateur en utilisant la sous option Nom d'authentification. Si la sous option Nom n'est pas utilisée, le serveur va généralement inviter à fournir un nom et mot de passe en clair. La sous option Nom doit être envoyée après que le serveur a envoyé une liste des types d'authentification

pris en charge et avant que le client finisse l'échange d'authentification ; cela assure que le serveur ne va pas inviter à un nom et mot de passe d'utilisateur. Dans la figure 1, la sous option Nom est envoyée immédiatement après que le serveur a présenté la liste des types d'authentification pris en charge.

Pour l'authentification DSS unilatérale, la paire de type d'authentification de deux octets est DSS AUTH_CLIENT_TO_SERVER | AUTH_HOW_ONE_WAY | ENCRYPT_OFF | INI_CRED_FWD_OFF. Cela indique que le mécanisme d'authentification DSS sera utilisé pour authentifier le client auprès du serveur et qu'aucun chiffrement ne sera effectué.

CertA est le certificat de client. Les deux certificats sont des certificats X.509 qui contiennent des clés publiques DSS [RFC2459]. Le client doit valider le certificat du serveur avant d'utiliser la clé publique DSA qu'il contient.

Au sein de l'échange d'authentification sans limite, la mise en œuvre est grandement simplifiée si chaque portion de l'échange porte un identifiant univoque. Pour cette raison, un identifiant de sous option d'un seul octet est porté immédiatement après la paire de type d'authentification de deux octets.

Les échanges détaillés à la Figure 1 ci-dessous supposent la connaissance de FIPS PUB 196 et de l'option d'authentification Telnet. Le client est la partie A, tandis que le serveur est la partie B. À la fin des échanges, le client est authentifié auprès du serveur.

Client (partie A) **Serveur (Partie B)**

<----- IAC DO AUTHENTICATION

IAC WILL AUTHENTICATION ------->

<---- IAC SB AUTHENTICATION SEND

<liste des options d'authentification> IAC SE

IAC SB AUTHENTICATION NAME <nom d'usager> ------->
IAC SB AUTHENTICATION IS DSS
AUTH_CLIENT_TO_SERVER | AUTH_HOW_ONE_WAY |
ENCRYPT_OFF | INI_CRED_FWD_OFF DSS_INITIALIZE
IAC SE ------->

<---- IAC SB AUTHENTICATION REPLY DSS
AUTH_CLIENT_TO_SERVER |
AUTH_HOW_ONE_WAY |
ENCRYPT_OFF | INI_CRED_FWD_OFF

DSS_TOKENBA Séquence(JetonID, JetonBA)

IAC SE

IAC SB AUTHENTICATION IS DSS
AUTH_CLIENT_TO_SERVER | AUTH_HOW_ONE_WAY |
ENCRYPT_OFF | INI_CRED_FWD_OFF DSS_CERTA_TOKENAB
Séquence(JetonID, CertA, JetonAB) IAC SE ---------->

Figure 1

3.2 Authentification mutuelle avec DSA

L'authentification mutuelle est légèrement plus complexe. La Figure 2 illustre les échanges.

Pour l'authentification DSS mutuelle, la paire de type d'authentification de deux octets est DSS AUTH_CLIENT_TO_SERVER | AUTH_HOW_MUTUAL | ENCRYPT_OFF | INI_CRED_FWD_OFF. Cela indique que le mécanisme d'authentification DSS va être utilisé pour authentifier mutuellement le client et le serveur et qu'aucun chiffrement ne sera effectué.

Client (Partie A) **Serveur (Partie B)**

IAC WILL AUTHENTICATION ------->

<-- IAC DO AUTHENTICATION

<-- IAC SB AUTHENTICATION SEND

<liste des options d'authentification> IAC SE

IAC SB AUTHENTICATION NAME <nom d'usager> ---->
IAC SB AUTHENTICATION IS DSS
AUTH_CLIENT_TO_SERVER | AUTH_HOW_MUTUAL |
ENCRYPT_OFF | INI_CRED_FWD_OFF DSS_INITIALIZE IAC SE -->

<-- IAC SB AUTHENTICATION REPLY DSS
AUTH_CLIENT_TO_SERVER |
AUTH_HOW_MUTUAL | ENCRYPT_OFF |
INI_CRED_FWD_OFF DSS_TOKENBA
Séquence(JetonID, JetonBA) IAC SE

Client (Partie A) **Serveur (Partie B)**

IAC SB AUTHENTICATION IS DSS
AUTH_CLIENT_TO_SERVER | AUTH_HOW_MUTUAL |
ENCRYPT_OFF | INI_CRED_FWD_OFF DSS_CERTA_TOKENAB
Séquence(JetonID, CertA, JetonAB) IAC SE ----->

<-- IAC SB AUTHENTICATION REPLY DSS
AUTH_CLIENT_TO_SERVER | AUTH_HOW_MUTUAL | ENCRYPT_OFF |
INI_CRED_FWD_OFF DSS_CERTB_TOKENBA2
Séquence(JetonID, CertB, JetonBA2) IAC SE

Figure 2

4. Syntaxe ASN.1

Comme on l'a dit précédemment, un sous ensemble conforme des champs et sous champs définis à partir de FIPS PUB 196 a été choisi. La présente section donne la syntaxe ASN.1 pour ce sous ensemble conforme.

La Figure 1 et la Figure 2 comportent des représentations des structures définies dans cette section. Les mises en œuvre devraient se référer au tableau suivant pour déterminer les définitions ASN.1 qui correspondent aux références des figures :

Figure 1 Séquence(JetonID, JetonBA) MessageBA
Séquence(JetonID, CertA, JetonAB) MessageAB

Figure 2 Séquence(JetonID, JetonBA) MessageBA
Séquence(JetonID, CertA, JetonAB) MessageAB
Séquence(JetonID, CertB, JetonBA2) MessageBA2

Les définitions ASN.1 suivantes spécifient le sous ensemble conforme de FIPS 196. Pour rester simple, aucun champ ou sous champ facultatif n'est inclus. La définition ASN.1 pour CertificationPath est importée de la Recommandation UIT-T X.509, et la définition ASN.1 pour Nom est importée de la Recommandation UIT-T X.501. Ces définitions ASN.1 ne sont pas répétées ici. Toutes les valeurs de signature DSA sont codées comme une séquence de deux entiers, employant les mêmes conventions que spécifiées dans la RFC 2459, paragraphe 7.2.2.

```
MessageBA ::= SEQUENCE {
        jetonId             [0] JetonId,
          jetonBA JetonBA  }

JetonBA ::= SEQUENCE {
        ranB                NombreAléatoire,
          horodatageB       HoroDatage  }

MessageAB ::= SEQUENCE {
        jetonId             [0] JetonId,
          certA     [1] CertData,
          jetonAB JetonAB  }

JetonAB ::= SEQUENCE {
        ranA                NombreAléatoire,
          ranB      NombreAléatoire,
          entitéB   NomEntité,
```

```
        horodatageB     HoroDatage,
        absigValue      CHAINE D'OCTETS  }

MessageBA2  ::=  SEQUENCE  {
      idJeton           [0] IdJeton,
        certB    [1] CertData,
        jetonBA2        JetonBA2  }

TokenBA2  ::=  SEQUENCE  {
      ranB              [0] NombreAléatoire,
        ranA     [1] NombreAléatoire,
        entityA  NomD'Entité,
        horodatageB2    HoroDatage,
        ba2sigValue     CHAINE D'OCTETS  }

CertData  ::=  SEQUENCE  {
      certPath          [0] CertificationPath  }                    -- voir X.509

NomEntité  ::=  SEQUENCE DE CHOIX  {                                -- seul un est permis !
      directoryName     [4] Nom  }                                  -- voir X.501

NombreAléatoire  ::=  ENTIER                                        -- 20 octets

JetonId  ::=  SEQUENCE  {
      typeJeton         ENTIER,                                     -- voir le tableau ci-dessous
        protoVerNo      ENTIER  }       -- toujours 0x0001

HoroDatage  ::=  Heure universelle
```

Le Type de jeton TokenId. est utilisé pour distinguer le type de message et le type d'authentification (unilatérale ou mutuelle). Le tableau suivant donne les valeurs nécessaires pour mettre en œuvre la spécification :

Type de message	**Type d'authentification**	**Type de jeton TokenId.**
MessageBA	Unilatérale	0x0001
	Mutuelle	0x0011
MessageAB	Unilatérale	0x0002
	Mutuelle	0x0012
MessageBA	Mutuelle	0x0013

5. Considérations pour la sécurité

La totalité du présent mémoire est consacrée aux mécanismes de sécurité. Pour que DSA fournisse l'authentification exposée, la mise en œuvre doit protéger la clé privée de toute divulgation.

Les mises en œuvre doivent générer de façon aléatoire des clés privées DSS, les valeurs 'k' utilisées dans les signatures DSS, et les noms occasionnels. L'utilisation de générateurs de nombres pseudo aléatoires ((PRNG, *pseudo-random number generator*) inadéquats pour générer des valeurs cryptographiques peut résulter en une sécurité faible ou inexistante. Un attaquant peut trouver beaucoup plus facile de reproduire l'environnement de PRNG qui a produit les valeurs et chercher dans le petit ensemble de possibilités résultant, plutôt que d'utiliser une recherche en force brute. La génération de nombres aléatoires de qualité est difficile. La [RFC1750] offre des conseils importants dans ce domaine, et l'Appendice 3 de FIPS PUB 186 [FIPS186] donne une technique de PRNG de qualité.

6. Remerciements

Nous tenons à remercier William Nace pour son soutien durant la mise en œuvre de cette spécification.

7. Considérations relatives à l'IANA

Le type d'authentification DSS et ses valeurs de sous option associées sont enregistrés auprès de l'IANA. Toutes valeurs de sous option utilisées pour étendre le protocole décrit dans le présent document doivent être enregistrées auprès de l'IANA avant utilisation. L'IANA a reçu pour instruction de ne pas allouer de nouvelles valeurs de sous option sans que soit soumise la documentation de leur usage.

8. Références

FIPS180-1 "Secure Hash Standard". FIPS Pub 180-1. 17 avril 1995. <http://csrc.nist.gov/fips/fips180-1.pdf>
FIPS186 "Digital Signature Standard (DSS)". FIPS Pub 186. 19 mai 1994. <http://csrc.nist.gov/fips/fips186.pdf>

FIPS196 "Standard for Entity Authentication Using Public Key Cryptography". FIPS Pub 196. 18 février 1997. <http://csrc.nist.gov/fips/fips196.pdf>

[RFC1750] D. Eastlake 3rd, et autres, "Recommandations d'aléa pour la sécurité", décembre 1994. *(Info., remplacée par la RFC4086)*

[RFC2459] R. Housley, W. Ford, W. Polk et D. Solo, "Profil de certificat d'infrastructure de clé publique X.509 et de CRL pour l'Internet", janvier 1999. *(Obsolète, voir la RFC 3280) (P.S.)*

[RFC2941] T. Ts'o, éd., J. Altman, "Option d'authentification Telnet", septembre 2000. *(P.S.)*

X.208 Recommandation UIT-T X.208, "Spécification de la notation de syntaxe abstraite n° 1 (ASN.1)". 1988.
X.501 Recommandation UIT-T X.501, "L'annuaire – Modèles". 1988.
X.509 Recommandation UIT-T X.509, 'L'annuaire – Cadres d'authentification". 1988.

9. Adresse des auteurs

Russell Housley
SPYRUS
381 Elden Street, Suite 1120
Herndon, VA 20172
USA
mél : housley@spyrus.com

Todd Horting
SPYRUS
381 Elden Street, Suite 1120
Herndon, VA 20172
USA
mél : thorting@spyrus.com

Peter Yee
SPYRUS
5303 Betsy Ross Drive
Santa Clara, CA 95054
USA
mél : yee@spyrus.com

10. Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'éditeur des RFC est actuellement assuré par la Internet Society.