Request for Comments : 501
Groupe de travail Réseau
NIC: 15718

K. Pogran, MIT-Multics
11 mai 1973
Traduction Claude Brière de L'Isle

Débroussaillage du "transfert libre de fichier"

Alors que le réseau ARPA devient plus mûr, on se trouve confronté à es problèmes et concepts délibérément laissés de côté et non touchés à des stades antérieurs du développement du réseau. Parmi les problèmes qui viennent maintenant sur le devant de la scène sont le contrôle d'accès, l'authentification de l'utilisateur, et la comptabilité. Ces questions découlent immédiatement des efforts de développement de méthodes uniformes pour fournit un accès limité "libre" aux serveurs de transfert de fichiers des systèmes hôtes, pour satisfaire aux besoins des utilisateurs en transmission de messagerie et de services similaires.

Plusieurs propositions ont été faites, décrites par des phrases telles que "messagerie avec moins de connexion", "comptes 'libres'", "transfert de fichier libre", etc. Ces propositions ont inévitablement incorporé en elles une notion particulière de la façon dont des choses comme le contrôle d'accès et l'authentification d'utilisateur sont accomplies et ces propositions, qui font sciemment ou sans le savoir des présupposés sur la façon de mettre en œuvre de tels mécanismes, rencontrent inévitablement de violentes critiques de la part de ceux dont les systèmes mettent en œuvre des mécanismes assez différents.

Dans la RFC 467, Bob Bressler propose des façons d'aider les utilisateurs qui souhaitent transférer des fichiers de ou vers des "systèmes qui ont un petit parfum de sécurité, mais sur lesquels l'utilisateur n'a pas de privilèges d'accès". Malheureusement, en commençant au premier paragraphe de la RFC, les notions de contrôle d'accès sur les fichiers (exemples de mécanismes de protection), et de contrôle d'accès au système (authentification de l'utilisateur) sont profondément embrouillées. De plus, il fait des hypothèses glissantes sur la nature et l'utilisation des mécanismes de comptabilité et des comptes sur les sites de serveur. La RFC 487 s'est aussi profondément enlisée dans ses hypothèses sur la nature du contrôle d'accès et sur les aspects de l'authentification de l'utilisateur des mises en œuvre de serveur de transfert de fichiers.

Bien sûr, ce qui est nécessaire dans cette conjoncture c'est une discussion lucide des concepts généraux impliqués dans les mécanismes de protection, et en particulier dans les contrôles d'accès de système de fichiers. Et bien, vous ne trouverez rien de cela dans le reste de cette RFC. Vous trouverez peut-être un exposé suffisant pour débrouiller ce que la RFC 487 a embrouille ; le reste devra venir plus tard.

Dans de nombreux systèmes, les mécanismes qui contrôlent l'accès au système, les mécanismes qui contrôlent l'accès aux fichiers, et les mécanismes de comptabilité sont tous reliés au moment où un utilisateur potentiel du système est authentifié : le système a vérifié son nom d'utilisateur, son mot de passe, son compte, ou quoi que ce soit, et a décidé qu'il est, bien sûr, un utilisateur valide du système. Cet utilisateur, qui aimerait que soit effectuée une sorte de traitement des informations en son nom, est appelé un principal dans le vocabulaire de la "confidentialité et de la protection". Un certain nombre de processus sont établis initialement pour ce principal, et un identifiant de principal (en principe infalsifiable) est associé à ce ou ces processus, de sorte que ses demandes d'accès aux fichiers et autres ressources système puissent être correctement validées. De plus, l'identité du compte auquel imputer les charges des ressources consommées par ces processus est associée aux processus à ce moment [1], bien que dans certains systèmes, un processus puisse changer son identifiant de compte à tout moment.

La première question est : quel identifiant de principal un processus de serveur de transfert de fichier va t-il utiliser ? Il y a au moins deux possibilités : 1) le serveur de transfert de fichier peut faire fonctionner un processus de "système démon", avec (généralement) un identifiant de principal hautement privilégié. Lorsqu'il agit au nom d'un utilisateur, il doit, lui-même, évaluer de façon interprétative l'accès de cet utilisateur au fichier. Il doit aussi être capable d'imputer au compte de cet utilisateur les ressources qu'il utilise. 2) Un processus de serveur de transfert de fichier peut recevoir le propre identifiant de principal de cet utilisateur. Avec cette mise en œuvre, la validation de l'accès de l'utilisateur aux fichiers est effectuée automatiquement par les mécanismes usuels de système de fichier.

Le paragraphe 4 de la RFC 487 suppose clairement la mise en œuvre 1) : "Si un utilisateur se connecte à un serveur FTP et fait une demande de fichier sans fournir un nom d'utilisateur – mot de passe, le serveur devrait alors examiner les paramètres d'accès au fichier ..." Les systèmes qui se préoccupent vraiment de la protection peuvent préférer la mise en œuvre 2), et pour de bonnes raisons – elle suit le "principe du moindre privilège", qui déclare qu'un processus devrait exécuter le plus petit privilège d'accès qu'il requiert pour effectuer sa tâche correctement. Faire fonctionner un processus de serveur de transfert de fichier avec l'identifiant de principal d'un utilisateur plutôt qu'avec un système démon laisse le système beaucoup moins exposé aux dommages causés par des actions incorrectes du serveur de transfert de fichier [2].

La question suivante est : À qui impute t-on les transferts de fichiers ? Bressler essaye d'établir quelques lignes directrices pour déterminer à qui imputer l'utilisation de transfert de fichier "sans connexion" (lire : "libre") : "En clair, la mémorisation d'un fichier dans le répertoire d'un utilisateur peut être imputée à cet utilisateur." Comment le mot "mémorisation" est il utilisé ici ? Assurément, cet utilisateur" peut être facturé pour le disque ou autre support de mémorisation des charges encourues par ce fichier qui occupe maintenant de l'espace, mais est-il légitime de taxer "cet utilisateur" pour les ressources d'entrée/sortie et/ou de CPU utilisées par quelqu'un d'autre pour transférer un fichier sur la Toile, et le placer dans le répertoire de cet utilisateur ? Par exemple, le receveur d'un message devrait il être débité du coût des ressources consommées par celui qui l'a envoyé ? (Souhaitez vous payer les coûts d'envoi de tous les pourriels qui arrivent sur votre boîte aux lettres ?)

Au téléphone, Bob m'a expliqué qu'il souhaitait un mécanisme qui pourrait, par exemple, me rendre capable, à sa demande, de transférer un fichier de mon système à son répertoire sur son système, sans exiger que je connaisse son mot de passe. Très bien. Dans cette situation, il paraîtrait sensé de débiter le compte de Bressler pour son transfert de fichier. Mais comment un utilisateur non authentifié peut-il dire à un serveur "Imputez ceci au compte de Bressler parce qu'il a dit qu'il était d'accord " ? Les pièges abondent. Le processus de serveur de transfert de fichier a besoin d'être capable de débiter le compte d'un utilisateur arbitraire ; cela présuppose encore une fois la mise en œuvre 1) du serveur de transfert de fichier décrit plus haut (ou autrement tout processus d'utilisateur dans le système aurait la capacité de taxer le compte de n'importe quel utilisateur ; cela ne semble pas souhaitable). Une approche plus raisonnable serait d'imputer cette instance de processus de serveur de transfert de fichier à un compte général de "services réseau". Les mécanismes pour accomplir cela sont présentés dans la RFC 491 [3].

La RFC 487 suggère bien à propos que la restitution de fichiers dans un répertoire "système" devrait être imputée en "frais généraux". Ici aussi, quelques hypothèses larges sont faites sur la nature des mécanismes de comptabilité et des comptes sur les sites de serveur. De plus, une perte indésirable de généralité est imposée au serveur de transfert de fichier : il faut maintenant qu'il ait la capacité de distinguer les noms de chemin des fichiers "système" des fichiers "d'utilisateur". Dans un certain nombre de systèmes, il n'y a pas de distinction syntaxique entre les deux, et le même mécanisme général peut être utilisé pour manipuler les deux sortes de fichiers (si on peut faire une distinction entre les deux). L'ajout de code au serveur de transfert de fichier qui examine le nom de chemin donné dans chaque demande, pour déterminer de quelle sorte il est, semble être antithétique de l'objectif d'uniformité et de généralité que beaucoup des systèmes d'aujourd'hui ont réalisé.

L'affirmation qu'une activité de transfert de fichier d'un utilisateur du réseau peut être imputée à un compte de "frais généraux" à l'échelle du système contient deux hypothèses : l'existence d'un tel compte ne peut pas être supposée sur tous les systèmes ; de plus, si il existe bien, dans certains cas, ce n'est pas le bon compte à imputer. Certainement, on peut toujours dire que le coût des communications inter utilisateurs serait supporté au sein de la communauté du réseau ARPA (ce à quoi revient le transfert "libre" de fichier), ce qui signifierait que de telles activités devraient être imputées à des comptes financés par l'ARPA. Si une opération d'un système hôte est entièrement financée par l'ARPA (ou si sa direction ne se soucie pas de qui paye pour cette activité) cela a alors un sens d'imputer l'activité de transfert "libre" de fichier à un compte de "frais généraux du système". D'un autre côté, ce n'est pas le cours des choses correct pour un système hôte dont le fonctionnement n'est pas financé par l'ARPA, car l'imputation des transferts "libres" de fichiers à des "frais généraux du système" aurait pour résultat de passer les coûts aux consommateurs locaux qui n'ont aucun intérêt dans le réseau ARPA.

Finalement, Bressler suggère que pour la restitution de fichier, les charges de CPU "peuvent être suffisamment faibles pour ne pas causer de problèmes majeurs". Croire cela est de la naïveté. Cela peut se trouver vrai pour un système qui n'impute pas aux utilisateurs le temps passé à exécuter le code de supervision, ou les sous-programmes d'entrée/sortie, et où les frais généraux du logiciel réseau n'apparaissent pas sur la facture de l'utilisateur. Dans ce cas, les frais généraux du logiciel réseau doivent contribuer aux "frais généraux du système", dont le coût doit être supporté par tous les utilisateurs. Je ne pense pas que beaucoup de gens dans la communauté de l'Internet considèreraient le temps réel (par opposition au temps passé) de CPU à transférer un fichier soit négligeable. Certainement, si un système est très populaire ou très occupé du point de vue de l'Internet, le temps de CPU cumulé passé à des transferts "libres" de fichiers, cumulé à la fin d'une période comptable (une semaine ? un mois ? un an ?) ne sera pas négligeable !

Dans la présente RFC, j'ai pris à parti la RFC 487 de Bob Bressler, principalement à cause de sa confusion sur plusieurs questions distinctes (bien qu'en rapport les unes avec les autres), et les hypothèses de mise en œuvre qu'elle contient et qui sont en conflit avec (ou vont largement à l'encontre des) les mécanismes et les philosophies sur la conception qui existent sur les autres systèmes (en particulier, le système qui m'est le plus familier, Multics) [4]. La portée des discussions de la présente RFC va au delà: Nous devons reconnaître qu'il est difficile de proposer des mécanismes valables sur tout l'Internet pour fournir les services désirables sans construire des hypothèses sur la façon dont ils seront mis en œuvre. Nous en sommes à un point où nous réclamons des services très sophistiqués, et nous proposons des mécanismes d'une sophistication correspondante. Il est temps de commencer à parler de la façon dont divers systèmes accomplissent des choses telles que l'authentification de l'utilisateur, le contrôle d'accès, et ainsi de suite, de sorte qu'on puisse tous gagner une compréhension plus claire de telles questions, et d'être capables de proposer des mécanismes incorporant moins d'hypothèses de mise en œuvre.

Notes finales :

[1] Dans certains systèmes, il y a une correspondance biunivoque entre principal et compte.

[2] On devrait noter que les systèmes qui choisissent la mise en œuvre 2) peuvent exiger une séquence d'authentification de l'utilisateur (commandes USER, PASS, et éventuellement ACCT) avant de permettre aucun transfert de fichier, comme déclaré explicitement à la page 17 de la RFC 354 (NIC 10596), et à la page 20 de la RFC 454 (NIC 14333). Cette séquence d'authentification serait exigée pour s'assurer que l'identifiant de principal est associé au processus de serveur de transfert de fichier nouvellement créé ; le processus n'est pas admis à fonctionner tant que son identifiant de principal n'a pas été établi.

[3] Noter qu'il y a au moins deux scénarios pour accomplir le transfert que Bressler désire : soit on "pousse" le fichier, en utilisant "FTP d'utilisateur" et le "serveur FTP" de son système, soit on "pousse" le fichier, en utilisant son "FTP d'utilisateur" et le "serveur FTP" de mon système. Bob choisit le premier scénario ; on peut objecter que, comme c'est Bob qui veut que le fichier soit transféré, le second scénario est le plus approprié. Une RFC à venir de Mike Padlipsky s'étend sur ces points, ainsi que sur une solution de remplacement entièrement différente.

[4] Padlipsky insiste sur le fait que j'ai aussi montré la supériorité de la mise en œuvre 2) de serveur de transfert de fichier (décrite ci-dessus), mais je réfute cette conclusion. Ceux qui sont intéressés peuvent vouloir examiner sa spécification de protocole unifié de niveau utilisateur, qui se fonde sur des prémices similaires.

[La présente RFC a été mise en forme pour entrée dans les archives en ligne des RFC par Via Genie]