BlindFTP

Documentation - 03/11/2008

Sommaire

Qu'est-ce que BlindFTP ?

BlindFTP est un logiciel qui permet de transférer des fichiers par réseau IP entre 2 machines à travers une liaison unidirectionnelle, sans besoin d'acquittements. L'absence d'acquittement est compensée par une redondance des données transmises par UDP. C'est un script Python simple et portable (Windows, Linux, MacOSX, ...).

Il est principalement conçu pour bâtir une interconnexion de type « diode réseau », par exemple à partir d'équipements Ethernet optiques et d'une fibre.

Des explications plus détaillées sur la constitution et l'utilisation d'une diode réseau sont disponibles à cette adresse: http://www.decalage.info/sstic06

Site officiel

Le site officiel du projet est http://adullact.net/projects/blindftp/. Quelques informations complémentaires sont aussi disponibles sur le site http://www.decalage.info/fr/python/blindftp.

Les personnes intéressées pour contribuer au développement peuvent y trouver les informations nécessaires. Il est nécessaire de s'inscrire sur le site Adullact pour demander à participer au projet.

Auteurs

- Philippe Lagadec (PL) - philippe.lagadec(a)laposte.net

- Laurent Villemin (LV) - ysetur(a)gmail.com

Licence

Ce logiciel est régi par la licence CeCILL soumise au droit français et respectant les principes de diffusion des logiciels libres. Vous pouvez utiliser, modifier et/ou redistribuer ce programme sous les conditions de la licence CeCILL telle que diffusée par le CEA, le CNRS et l'INRIA sur le site "http://www.cecill.info". Une copie de cette licence est également fournie avec le logiciel.

En contrepartie de l'accessibilité au code source et des droits de copie, de modification et de redistribution accordés par cette licence, il n'est offert aux utilisateurs qu'une garantie limitée. Pour les mêmes raisons, seule une responsabilité restreinte pèse sur l'auteur du programme, le titulaire des droits patrimoniaux et les concédants successifs.

A cet égard l'attention de l'utilisateur est attirée sur les risques associés au chargement, à l'utilisation, à la modification et/ou au développement et à la reproduction du logiciel par l'utilisateur étant donné sa spécificité de logiciel libre, qui peut le rendre complexe à manipuler et qui le réserve donc à des développeurs et des professionnels avertis possédant des connaissances informatiques approfondies. Les utilisateurs sont donc invités à charger et tester l'adéquation du logiciel à leurs besoins dans des conditions permettant d'assurer la sécurité de leurs systèmes et ou de leurs données et, plus généralement, à l'utiliser et l'exploiter dans les mêmes conditions de sécurité.

Le fait que vous puissiez accéder à cette documentation signifie que vous avez pris connaissance de la licence CeCILL, et que vous en avez accepté les termes.

Versions fournies

Actuellement BlindFTP est toujours en cours de développement, et il est fourni en 2 versions pour faciliter la compréhension du code source:

- **bftp_simple.py**: version simple à comprendre, non optimisée.
- **bftp.py**: version améliorée, ordonnancement des transferts plus optimisé.

Installation

BlindFTP est écrit en langage Python, ce qui permet d'obtenir un code relativement simple et portable. Il a été testé avec succès sur Windows XP, MacOSX 10.3.9 et Linux. Il devrait également fonctionner sur d'autres systèmes d'exploitation supportés par Python, peut-être après quelques modifications mineures.

Il faut cependant installer un interpréteur Python pour le faire fonctionner, si possible la dernière version disponible sur le site http://www.python.org. Attention BlindFTP a été testé avec Python 2.5, mais la compatibilité n'est pas garantie avec Python 2.6 et 3.0.

Sous Windows, les extensions Win32 sont également nécessaires. Une solution simple est d'installer ActivePython (http://www.activestate.com/Products/ActivePython), qui comprend l'interpréteur Python et les extensions PyWin32, mais cette version de Python n'est pas toujours la plus récente. On peut trouver séparément les extensions win32 à cette adresse: http://sourceforge.net/projects/pywin32

BlindFTP s'appuie également sur quelques modules tiers fournis avec le code source, qu'il peut être utile de mettre à jour:

- path: http://pypi.python.org/pypi/path.py
- xfl: http://www.decalage.info/en/python/xfl
- plx: http://www.decalage.info/en/python/plx

L'installation de BlindFTP consiste simplement à recopier les fichiers fournis dans un répertoire quelconque. Il est ensuite possible de créer des scripts pour lancer BlindFTP avec les options voulues, voire de programmer le lancement de ces scripts en fonction des besoins.

Sitecustomize.py – éviter les erreurs Unicode

Depuis Python 2.4, la conversion des chaînes Unicode est devenue plus stricte, et il n'est pas rare de voir un script déclencher une exception du style « UnicodeDecodeError », généralement lorsqu'on tente d'afficher une chaîne contenant des accents. En effet par défaut le codec de base de Python est « Ascii », et il n'autorise que des caractères de 7 bits sans accents.

BlindFTP contient quelques fonctions d'affichage et de conversion pour remédier à ce problème, cependant il est toujours possible que ce type d'exception soit déclenché dans des cas particuliers.

Il existe toutefois une solution, qui nécessite de configurer le codec de base de Python en « Latin 1 ». Pour cela, il est nécessaire de créer un fichier « sitecustomize.py » dans le répertoire lib/site-packages de votre répertoire Python (par exemple C:\Python25), avec le contenu suivant:

```
# sitecustomize.py
# this file can be anywhere in your Python path,
# but it usually goes in ${pythondir}/lib/site-packages/
import sys
sys.setdefaultencoding('latin_1')
```

Utilisation

Test rapide

Le script **demo.bat** est fourni pour tester rapidement BlindFTP sur une machine Windows sans avoir besoin d'une diode réseau matérielle. Ce script crée simplement deux répertoires emission et reception, et lance deux processus BlindFTP pour recopier synchroniser automatiquement les deux répertoires.

Un script équivalent pourrait facilement être créé pour UNIX.

Utilisation avec une diode réseau matérielle

L'utilisation normale de BlindFTP consiste à transférer des fichiers à travers une liaison unidirectionnelle de type « diode réseau »: Pour cela on recopie le contenu d'un répertoire d'une machine source vers une machine cible, et on synchronise les changements si le répertoire source est modifié au cours du temps.

Une application très utile est la recopie de signatures antivirus ou de mises à jour de logiciels (Windows, Linux, ...) depuis Internet vers un réseau sécurisé.

Par exemple, la machine 192.168.1.2 émet le répertoire "source" vers la machine 192.168.1.3, dont l'adresse MAC est 01:23:45:67:89:0A, dans le répertoire "destination". L'option -b permet d'émettre les fichiers en boucle.

La commande "arp -s" est nécessaire pour enregistrer manuellement l'adresse MAC de la machine de réception dans la table ARP, puisque la liaison diode ne permet pas les requêtes ARP.

1) Sur la machine qui reçoit, sous Windows:

```
bftp.py -r reception -a 192.168.1.3
```

Sous UNIX:

```
python bftp.py -r reception -a 192.168.1.3
```

On peut noter que BlindFTP nécessite d'indiquer l'adresse IP de l'interface destinée à recevoir les données.

2) Sur la machine qui émet, sous Windows:

```
arp -s 192.168.1.3 01-23-45-67-89-0A
bftp.py -s source -a 192.168.1.3 -b
```

Sous UNIX:

```
arp -s 192.168.1.3 01:23:45:67:89:0A
python bftp.py -s source -a 192.168.1.3 -b
```

Options

Pour afficher les autres options: lancer **bftp.py -h**

```
Usage: bftp.py [options] <fichier ou repertoire>
```

```
Options:
  -h, --help        show this help message and exit
  -e, --envoi       Envoyer le fichier
  -s, --synchro     Synchroniser l'arborescence
  -r, --reception   Recevoir des fichiers dans le repertoire indique
  -a ADRESSE        Adresse destination: Adresse IP ou nom de machine
  -p PORT_UDP       Port UDP
  -l DEBIT          Limite du debit (Kbps)
  -d, --debug       Mode Debug
  -b, --boucle      Envoi des fichiers en boucle
  -P PAUSE          Pause entre 2 boucles (en secondes)
  -c, --continue    Fichier de reprise a chaud
```

Compensation des pertes de données

Comme la liaison est unidirectionnelle, l'émetteur ne reçoit aucun acquittement et il ne peut donc pas être averti en cas de perte de données côté récepteur. En utilisation normale, on constate que ce dernier ne reçoit pas toujours tous les datagrammes UDP, car les tampons de sa pile IP peuvent être rapidement saturés si le processeur est occupé à une autre tâche (interface graphique, écriture disque, serveur, ...). Cela peut se produire même avec une machine très performante, en donnant la priorité maximum au processus BlindFTP.

Dans ce cas on constate simplement que 1 ou plusieurs paquets successifs ne sont pas reçus.

Voici les diverses solutions employées pour compenser ces pertes de données:

Redondance

Les données sont transmises plusieurs fois, au cas où elles seraient perdues. Dans BlindFTP, cette redondance se fait au niveau de chaque fichier: les fichiers sont simplement réémis indéfiniment (version 0.11) ou bien avec une limite de N renvois (version 0.14).

Un projet précédent effectuait cette redondance au niveau des datagrammes IP sans tenir compte des fichiers. Il est apparu que cette solution nécessite un protocole plus complexe que celui de BlindFTP et n'apporte pas beaucoup d'avantages par rapport a la redondance au niveau fichier.

Il est important de noter que l'utilisation de la redondance est une sécurité, mais qu'il vaut mieux éviter d'y avoir recours dans le cas général. En effet dès qu'on doit retransmettre un paquet d'un fichier pour le recevoir complètement, le débit de réception est divisé par 2 voire plus.

Limitation du débit

Le débit d'émission est volontairement limité à une valeur qui permet de ne pas avoir de perte de données lors du premier envoi, dans un cas d'utilisation normale de la machine réceptrice. La valeur de cette limite dépend donc fortement du matériel employé, du système d'exploitation, des applications, et de la taille des fichiers transmis.

Une valeur typique avec un matériel PC « bas de gamme » et une liaison Ethernet optique est de l'ordre de 8 à 12 Mbits/s.

Pour limiter le débit d'émission on emploie l'option -l :

```
python bftp.py -s source -a 192.168.1.3 -b -l 12000
```

Augmentation des tampons de la pile IP

Cette solution permet a priori de réduire les risques de perte de données, et donc d'employer une limite de débit plus élevée. Elle est cependant dépendante du système d'exploitation, et doit être configurée en dehors de BlindFTP.

Toute contribution sur la mise en oeuvre de cette solution pour les divers systèmes d'exploitation serait la bienvenue, afin de compléter cette documentation. :-)

Principe de fonctionnement et applications

Le principe de fonctionnement est décrit en détail dans l'article et la présentation de SSTIC06:

http://actes.sstic.org/SSTIC06/Diode_ExeFilter/

Cette documentation sera mise à jour ultérieurement.

Erreurs connues

- Lorsque l'émetteur et le récepteur fonctionnent sur des systèmes d'exploitation différents, le format de date des fichiers (retourné par la fonction getmtime de Python) et le format binaire des paquets BlindFTP peuvent être incompatibles. Il est donc recommandé d'employer le même système d'exploitation.

Évolutions envisagées

Bien qu'il soit utilisé en semi-production depuis mi-2005, le fonctionnement de BlindFTP est encore très spartiate, et de nombreux points restent à améliorer:

- gestion plus complète des exceptions
- gestion des fichiers temporaires
- arrêt « propre » en cas d'erreur
- journalisation des transferts et des erreurs
- optimisation du débit
- calcul de statistiques sur les transferts (débit utile, pertes, ...)
- optimisation des envois
- surveillance d'un répertoire, pour détecter les fichiers modifiés sans tout relire
- interface graphique
- reprise d'un transfert après erreur ou arrêt volontaire
- transfert d'autres données que des fichiers: courriels, syslog, SNMP, datagrammes UDP quelconques
- sous Unix, abandon des privilèges root dès que possible
- etc...

Une liste plus précise de ces améliorations pourra être fournie aux développeurs souhaitant contribuer au projet.