IPSWITCH

WS_FTP

Professional

Guide de la sécurité

IPSWITCH

Ipswitch Inc.
10 Maguire Rd
Suite 200
Lexington, MA 02421
États-Unis

Web : http://www.ipswitch.com
Téléphone : 781.676.5700
Fax : 781.676.5710

Droits d'auteur

Historique des versions

Version 9.0 Mise en circulation en juin 2004

Version 2006 Mise en circulation en juin 2005

Chapitre 1 Transfert de fichiers sécurisé

Sélection d'une méthode de transfert sécurisé 1

Chapitre 2 SSL (Secure Sockets Layer)

Aperçu 3
Pourquoi utiliser SSL ? 5
Comment établir une connexion SSL ? 5
Génération d'un certificat 6
Importation d'un certificat 7
Sélection d'un certificat 8
Autorités approuvées 8
Certificat non approuvé 10
Utilisation d'un pare-feu NAT 11

Chapitre 3 SSH (Secure Shell)

Aperçu 13
Pourquoi utiliser SSH ? 13
Comment établir une connexion SSH ? 14
Génération d'une paire de clés SSH 14
Exportation d'une clé publique SSH 15

Chapitre 4 OpenPGP

Aperçu 17
Comment activer le mode OpenPGP ? 18
Comment activer le mode OpenPGP pour un site par défaut ? 18
Génération d'une paire de clés 19
Importation d'une clé 19
Exportation d'une paire de clés 19
Scénario 20

Chapitre 5 Utilisation de pare-feux

Multiples pare-feux 23
Types de pare-feux 24
Configuration d'un pare-feu 24
Utilisation d'un pare-feu configuré 25
Utilisation de UPnP 26

Annexe A Éditeur FireScript

Rôle d'un script FireScript 27
Composants de script FireScript 27
Séquence de connexion 30
Langage FireScript 31
Variables FireScript 31
Expansion de chaîne 33
Expressions de fonction 34
Instructions de script FireScript 34
Instructions de commutation 35
Instructions case 35
Continue 37
Sauts et étiquettes 37
Return 38
Autodetect 38
Instructions SSL 39
Mots clés FireScript 39

Transfert de fichiers sécurisé

Chapitre 1

Ce guide décrit les protocoles de sécurité utilisés dans Ipswitch WS_FTP Professional : SSL, SSH et OpenPGP. Il explique aussi comment configurer Ipswitch WS_FTP Professional pour ces protocoles afin de sécuriser les connexions.

Ce chapitre fournit un aperçu et une comparaison de chaque protocole pour vous aider à déterminer le mieux adapté à vos besoins.

Sélection d'une méthode de transfert sécurisé

La méthode utilisée pour sécuriser des transferts de fichiers dépend de vos objectifs sur le plan sécurité. Le tableau suivant vous aidera à choisir la méthode de sécurité la mieux adaptée à vos besoins :

	Configuration du client ?	Configuration du serveur ?	Cryptage de l'accès ?	Crypter les canaux de commande ?	Crypter le transfert de fichiers ?	Crypter le fichier lui-même ?
SSL	O	O	O	O	O	N
SSH	O	O	O	O	O	N
OpenPGP	O	N	N	N	N	O

Dans ce chapitre :

Sélection d'une méthode de transfert sécurisé

À propos de SSL

À propos de SSH

À propos de OpenPGP

REMARQUE : Avec SSL et SSH, l'administrateur du serveur auquel vous tentez de vous connecter devra vous indiquer le type de serveur utilisé à cette adresse. Si vous ne connaissez pas le type et que vous tentez d'établir une connexion SSL ou SSH à un serveur qui ne supporte pas les protocoles nécessaires, la connexion échoue.

À propos de SSL

SSL (Secure Socket Layer) est un protocole permettant de crypter et décrypter les données transmises par le biais de connexions Internet directes. Quand un client établit une connexion SSL avec un serveur, toutes les données échangées avec ce serveur sont codées à l'aide d'un algorithme mathématique complexe qui rend difficile le décodage des données interceptées.

À propos de SSH

SSH (Secure Shell) est un protocole de sécurité qui vous permet d'établir une connexion sécurisée à un serveur sur lequel les protocoles SSH et SFTP (Secure File Transfer Protocol) sont installés.

SSH crypte toutes les communications entre client et serveur. Quand une connexion SSH est établie, SFTP est le protocole utilisé pour toutes les tâches réalisées par le biais de cette connexion sécurisée.

À propos de OpenPGP

OpenPGP est une méthode de cryptage de fichiers à base de clés. Grâce à cette méthode, seul le destinataire peut recevoir et décrypter les fichiers. OpenPGP permet de sécuriser les communications par messagerie électronique, mais sa technologie est également applicable à FTP.

OpenPGP utilise deux clés cryptographiques pour sécuriser les fichiers. Une clé publique sert à crypter le fichier et seule la clé privée correspondante peut le décrypter.

REMARQUE :à la différence de SSL et SSH, OpenPGP n'est pas un type de connexion, mais une méthode de cryptage de fichier avant sa télétransmission. Par conséquent, le mode OpenPGP peut être utilisé conjointement avec les connexions FTP, SSL ou SSH standard.

SSL (Secure Sockets Layer)

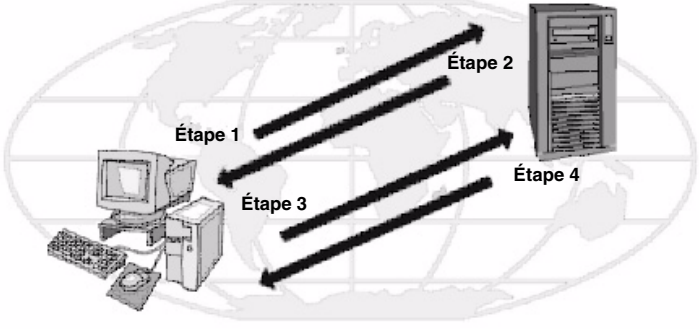
Chapitre 2

SSL (Secure Sockets Layer) peut être utilisé conjointement avec FTP pour assurer une meilleure sécurité. Ce chapitre présente un aperçu du protocole SSL et décrit son fonctionnement dans Ipswitch WS_FTP Professional.

Aperçu

SSL est un protocole de cryptage/décryptage des données transmises par connexion Internet directe. Quand un client établit une connexion SSL avec un serveur, toutes les données échangées avec ce serveur sont codées à l'aide d'un algorithme mathématique complexe qui rend difficile le décodage des données interceptées.

Voici une illustration étape par étape du fonctionnement de SSL.



Étape 1 : Le client établit la connexion initiale avec le serveur et demande l'établissement d'une connexion SSL. Si SSL implicite est utilisé, la connexion initiale est cryptée. Si SSL explicite est utilisé, le contact initial n'est pas crypté.

Étape 2 : Si sa configuration est correcte, le serveur envoie au client son certificat et sa clé publique.

Dans ce chapitre :

Aperçu

Pourquoi utiliser SSL ?

Comment établir une connexion SSL ?

Génération d'un certificat

Importation d'un certificat

Sélection d'un certificat

Autorités approuvées

Certificat non approuvé

Utilisation d'un pare-feu NAT

Étape 3 : Le client compare le certificat du serveur à une base de données d'autorités approuvées. Si le certificat figure dans cette liste, le client fait confiance au serveur et passe à l'étape 4. Si le certificat est absent de cette liste, l'utilisateur doit l'ajouter à la base de données d'autorités approuvées avant de passer à l'étape 4.

Étape 4 : Le client utilise cette clé publique pour crypter une clé de session qu'il envoie au serveur. Si le serveur demande le certificat du client à l'étape 2, celui-ci doit alors l'envoyer.

Étape 5 : Si le serveur est configuré pour la réception de certificats, il compare celui reçu à ceux figurant dans la liste de sa base de données d'autorités approuvées et accepte ou rejette la connexion.

Si la connexion est rejetée, un message d'erreur est envoyé au client. Si la connexion est acceptée, ou si le serveur n'est pas configuré pour la réception de certificats, il décode la clé de session du client avec sa propre clé privée et renvoie un message de réussite au client, en ouvrant par la même occasion un canal sécurisé pour les données.

Pour comprendre le protocole SSL, il faut comprendre les éléments qui le constituent. Voici une liste de ces éléments avec le rôle de chaque.

Client. Dans ce cas, le client est Ipswitch WS_FTP Professional.

Certificat. Le fichier de certificat contient les informations d'identification du client ou du serveur. Ce fichier est utilisé lors des négociations de connexion pour identifier les parties intéressées. Dans certains cas, le certificat du client doit être signé par le certificat du serveur pour rendre possible l'ouverture d'une connexion SSL. Les fichiers de certificat ont l'extension .crt.

Clé de session. La clé de session est utilisée par le client et le serveur pour crypter les données. Elle est créée par le client.

Clé publique. La clé publique est le dispositif à l'aide duquel le client crypte une clé de session. Elle n'existe pas en tant que fichier, mais dérive de la création d'un certificat et d'une clé privée. Les données cryptées avec une clé publique ne peuvent être décryptées qu'avec la clé privée utilisée pour sa création.

Clé privée. La clé privée décrypte la clé de session du client qui est cryptée par une clé publique. Le fichier de clé privée a l'extension .key. Les clés privées ne doivent JAMAIS être distribuées à quiconque.

Demande de signature de certificat. Une demande de signature de certificat est générée chaque fois qu'un certificat est créé. Ce fichier est utilisé quand vous devez faire signer votre certificat. Une fois le fichier Demande de signature de certificat signé, un nouveau certificat est créé et peut être utilisé pour remplacer le certificat non signé.

Pourquoi utiliser SSL ?

SSL améliore la sécurité FTP en cryptant et sécurisant la plupart des aspects de la connexion.

REMARQUE : Vous ne pouvez utiliser SSL que si le serveur FTP a été configuré pour accepter les connexions SSL. Si vous désirez utiliser SSL mais que votre serveur ne supporte pas ce protocole, contactez l'administrateur de votre serveur.

Comment établir une connexion SSL ?

Pour établir une connexion SSL avec un serveur configuré pour SSL :

1. Créez un profil de site et sélectionnez **FTP/SSL implicite** ou **FTP/SSL (AUTH SSL)** lorsque le type de serveur vous est demandé.
2. Quand vous cliquez sur **Connecter**, Ipswitch WS_FTP Professional informe le serveur que vous souhaitez établir une connexion SSL. Le serveur transmet alors au client un certificat par lequel il s'identifie. Si ce certificat est déjà présent dans la base de données des Autorités approuvées, la connexion est établie.
3. Dans le cas contraire, la boîte de dialogue Certificat non approuvé s'affiche.
4. Sélectionnez l'option nécessaire, puis cliquez sur **OK**. Si le serveur ne requiert pas le renvoi d'un certificat, la connexion sécurisée est établie. Toutes les données échangées entre le client et le serveur seront cryptées.

Si le serveur requiert le renvoi d'un certificat par WS_FTP Professional, suivez les instructions de Vérification du certificat client.

Vérification du certificat client

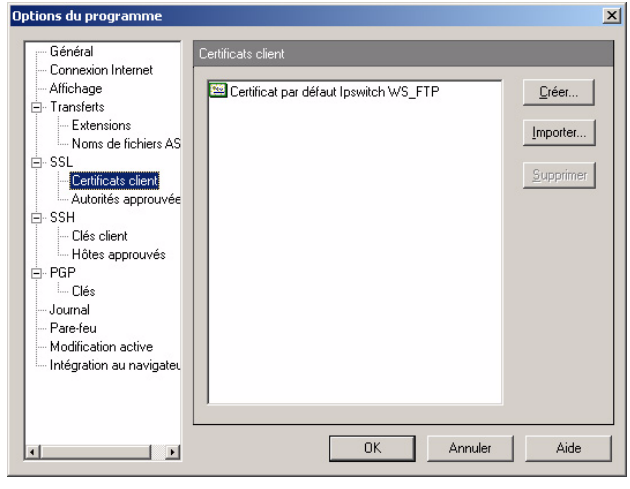
Si le serveur auquel vous tentez de vous connecter exige du client qu'il lui renvoie un certificat d'identification, vous devez :

1. Configurer le site et sélectionner **FTP/SLL implicite** ou **FTP/SSL (AUTH SSL)** lorsque le type de serveur vous est demandé.
2. Créer un certificat. Reportez-vous à la section « Génération d'un certificat » à la page 6 pour plus de détails.
3. Envoyer le fichier Demande de signature de certificat à l'administrateur de votre serveur.
4. Une fois que l'administrateur du serveur a signé la demande de signature du certificat, celle-ci vous est renvoyée.
5. Après avoir reçu le fichier, suivez les instructions de « Sélection d'un certificat » à la page 8 et sélectionnez le nouveau certificat dans la zone **Certificat**.
6. Connectez-vous au serveur.

Génération d'un certificat

Pour créer un certificat SSL :

1. Dans la fenêtre principale, sélectionnez **Outils > Options**. La boîte de dialogue Options du programme s'affiche.
2. Sélectionnez **Certificats client**.



3. Cliquez sur **Créer**. L'assistant **Création d'un certificat client SSL** s'affiche.
4. Entrez un nom dans la zone **Certificat**. Ce nom sera celui du certificat généré par Ipswitch WS_FTP Professional.
5. Sélectionnez la date d'expiration du certificat.
6. Entrez, puis entrez à nouveau une phrase de passe pour ce certificat. La phrase de passe sert à crypter la clé privée.

> **REMARQUE :** Il est important de retenir cette phrase de passe. Elle peut être formée d'une combinaison quelconque de mots, symboles, espaces ou chiffres.

7. Cliquez sur **Suivant** pour continuer.
8. Complétez toutes les zones sous Informations du certificat :

   Localité : nom de la ville où vous vous trouvez. (Ex. : Brest)

   Départ./Province : nom de l'état, du département ou de la province où vous vous trouvez. (Ex. **:** Finistère)

Organisation : nom de société ou de particulier.

Nom commun : nom de la personne qui crée le certificat ou nom de domaine complet du serveur associé à l'hôte.

E-mail : adresse de courrier électronique de la personne titulaire du certificat.

Service : nom du service dans l'organisation. (Ex. **:** Recherche et Développement)

Pays : pays où vous résidez, exprimé sous forme de code à deux lettres. (Ex. **:** US)

9 Quand toutes les zones sont remplies correctement, cliquez sur **Suivant** pour continuer. Si des zones sont vides, vous ne pouvez pas continuer.

10 Passez en revue les informations de la dernière boîte de dialogue et cliquez sur **Terminer** pour créer le certificat.

Si vous créez un certificat pour Ipswitch WS_FTP Professional, envoyez la demande de signature du certificat (par courrier électronique) à l'administrateur de votre serveur. S'il exige ce certificat, il le signera et vous le renverra. Quand vous recevez le certificat, vous devez l'importer dans votre base de données de certificats.

Importation d'un certificat

Pour utiliser un certificat qui vous a été envoyé, ou un certificat généré par le serveur Ipswitch WS_FTP, vous devez l'importer dans votre base de données de certificats.

Pour importer un certificat :

1 Dans la fenêtre principale, sélectionnez **Outils > Options**. La boîte de dialogue Options du programme s'affiche.

1 Sélectionnez **Certificats client**, puis cliquez sur le bouton **Importer**. L'assistant Importation d'un certificat s'affiche.

2 Sélectionnez un certificat, puis cliquez sur **Suivant**.

3 Sélectionnez le fichier de clé privée pour ce certificat, puis cliquez sur **Suivant**.

4 Entrez la phrase de passe utilisée pour créer ce certificat, puis cliquez sur **Suivant**.

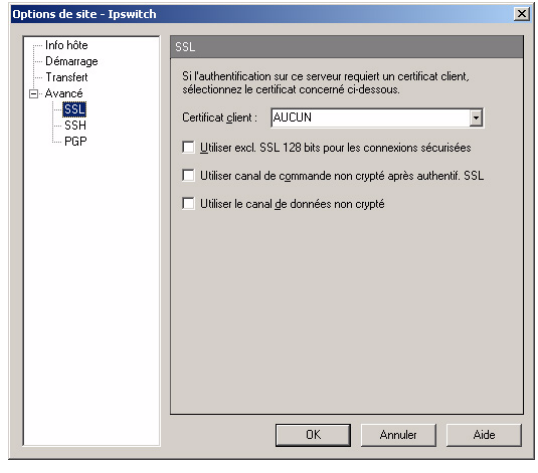
5 Entrez le nom voulu pour identifier le certificat dans votre liste de certificats, puis cliquez sur **Suivant**.

6 Passez en revue les informations de la dernière boîte de dialogue et cliquez sur **Terminer** pour ajouter le certificat à la liste.

Sélection d'un certificat

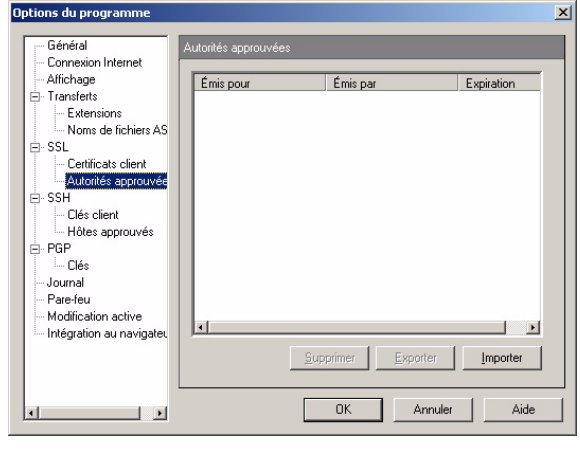
Les certificats sont utilisés au niveau du site, ce qui signifie que vous devez sélectionner un certificat pour chaque profil de site que vous créez (vous pouvez utiliser le même certificat pour tous vos sites si vous le désirez).

Les certificats sont sélectionnés dans la boîte de dialogue Options de site : SSL à l'aide de la zone de liste **Certificat client**. Celle-ci affiche tous les certificats dans la boîte de dialogue Options du programme : Certificat client.



Autorités approuvées

La boîte de dialogue **Autorités approuvées** stocke la liste de noms de certificat approuvés par l'utilisateur.



Affichage du certificat

Émis pour : identification du destinataire du certificat.

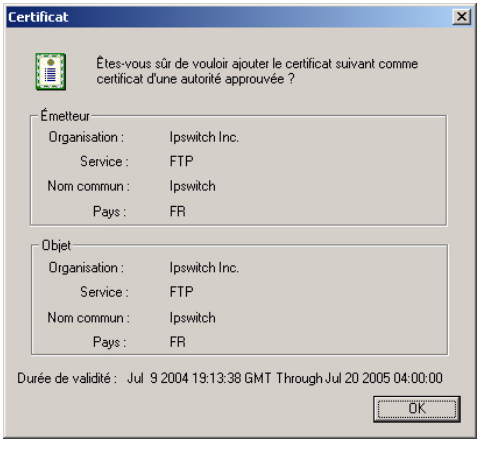
Émis par : identification du signataire du certificat.

Expiration : date d'expiration du certificat.

Ajout d'un certificat

Pour ajouter un certificat à la base de données :

1 Cliquez sur le bouton **Importer** et sélectionnez le chemin et le nom de fichier pour le certificat. La boîte de dialogue **Ajouter un certificat** s'affiche.



2 Passez en revue les informations dans cette boîte de dialogue, puis cliquez sur **Oui** pour ajouter le certificat à la base de données.

Exportation d'un certificat

Pour exporter un certificat depuis la base de données Autorités approuvées :

1 Sélectionnez le certificat à exporter depuis la base de données.

2 Cliquez sur le bouton **Exporter**.

Sélectionnez le dossier dans lequel le certificat doit être copié et tapez le nom souhaité pour le fichier de certificat.

3 Cliquez sur **OK**.

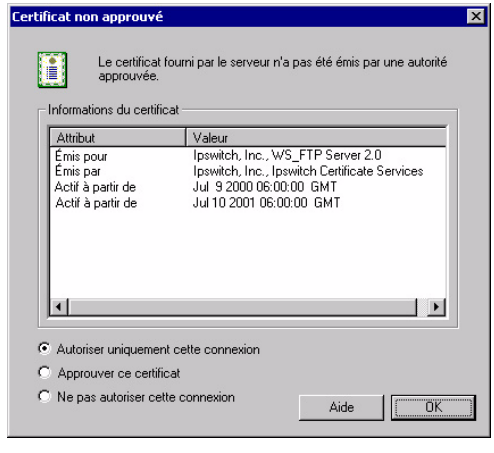
Suppression d'un certificat

Pour supprimer un certificat :

1. Sélectionnez le certificat à supprimer.
2. Cliquez sur **Supprimer**.
3. Un avertissement vous conseille alors d'exporter le certificat avant de le supprimer. La suppression du certificat entraîne celle du fichier de certificat.
4. Cliquez sur **OK** pour supprimer le certificat.

Certificat non approuvé

Quand vous établissez une connexion SSL à un serveur, celui-ci vous envoie un certificat. Si ce certificat ne figure pas dans la liste de l'onglet Autorité approuvée ou s'il n'a pas été signé par un certificat présent dans cette liste, cette boîte de dialogue s'affiche.



Informations du certificat

Émis pour : nom de la personne ou de la société propriétaire du certificat.

Émis par : nom de la personne ou de la société signataire du certificat.

Actif à partir de : date à laquelle ce certificat a été activé.

Expiration : date à laquelle le certificat affiché cesse d'être valide. Options de certificat

Options de certificat

Autoriser uniquement cette connexion : Si cette option est sélectionnée, la connexion est établie, mais Ipswitch WS_FTP Professional ne reconnaît quand même pas le certificat comme autorité approuvée. Lors de la tentative suivante de connexion à ce serveur, cette boîte de dialogue s'affiche à nouveau.

Approuver ce certificat : Si cette option est sélectionnée, la connexion est établie et le certificat est ajouté à la base de données des Autorités approuvées sur l'onglet Autorité approuvée. Des connexions peuvent ainsi être établies sans invite par la suite.

Ne pas autoriser cette connexion : Si cette option est sélectionnée, la connexion prend fin.

Utilisation d'un pare-feu NAT

Lors de l'utilisation d'un pare-feu NAT (Network Address Translation), des problèmes se présenteront peut-être si vous tentez d'utiliser le cryptage SSL. Pour y remédier, vous devez configurer Ipswitch WS_FTP Professional et le pare-feu pour permettre les connexions entrantes à votre PC. Ipswitch WS_FTP Professional doit indiquer au serveur que celui-ci doit se connecter à l'adresse IP externe, et le pare-feu doit rediriger vers votre PC les connexions en provenance du serveur. Vous devez aussi limiter le nombre de ports que le pare-feu ouvre pour ces connexions. Ces modifications permettent souvent l'emploi de SSL avec un pare-feu NAT.

Configuration de SSL à travers un pare-feu NAT

1. Dans la fenêtre principale, sélectionnez **Outils > Options**. La boîte de dialogue Options du programme s'affiche.
2. Sélectionnez **SSL**. Les options SSL apparaissent.
3. Sélectionnez l'option **Forcer l'adresse IP PORT**.
4. Entrez l'**Adresse IP** du pare-feu NAT du client.
5. Sélectionnez l'option **Limiter la plage des ports locaux**.
6. Sélectionnez les numéros de port **Minimum** et **Maximum**.
7. Cliquez sur **OK**.

SSH (Secure Shell)

Chapitre 3

Le protocole SSH (Secure Shell) permet d'améliorer la sécurité FTP standard. Ce chapitre décrit l'emploi du protocole SSH dans Ipswitch WS_FTP Professional.

Aperçu

SSH est un protocole de sécurité qui permet d'établir une connexion sécurisée à un serveur sur lequel les protocoles SSH et SFTP (Secure File Transfer Protocol) sont installés.

SSH crypte toutes les communications entre client et serveur. Quand une connexion SSH est établie, SFTP est le protocole utilisé pour toutes les tâches réalisées par le biais de cette connexion sécurisée.

REMARQUE : Ipswitch WS_FTP Professional supporte uniquement SFTP/SSH2.

Pourquoi utiliser SSH ?

SSH améliore la sécurité standard de FTP en cryptant et sécurisant tous les aspects de la connexion et du transfert.

REMARQUE : vous ne pouvez utiliser SSH que si le serveur FTP supporte et a été configuré pour accepter les connexions SSH. Si vous désirez utiliser SSH mais que votre serveur ne supporte pas ce protocole, contactez l'administrateur de votre serveur.

Dans ce chapitre :

Aperçu

Pourquoi utiliser SSH ?

Comment établir une connexion SSH ?

Génération d'une paire de clés SSH

Exportation d'une clé publique SSH

Comment établir une connexion SSH ?

L'établissement d'une connexion SSH requiert un minimum de configuration supplémentaire pour les profils de sites existants ou nouveaux.

Quand vous créez un nouveau profil de site à l'aide de l'**Assistant Connexion**, changez simplement le **Type de serveur** à **SFTP/SSH** à l'invite de l'assistant.

Pour modifier un profil de site existant :

1. Sélectionnez le site dans la liste des **sites configurés**.
2. Cliquez sur le bouton **Modifier**.
3. Cliquez sur l'onglet **Avancé**.
4. Dans la liste déroulante **Type de serveur**, sélectionnez **SFTP/SSH**. Cliquez sur **OK**.
5. Sélectionnez un mode d'authentification :
   - **Mot de passe** : Si votre serveur utilise l'authentification par mot de passe, la configuration est terminée. Lors de la prochaine connexion à ce site, SSH est utilisé pour sécuriser la connexion.
   - **Clé publique** : Si votre serveur utilise l'authentification par clé publique, sélectionnez **Avancé > SSH**. Sélectionnez la paire de clés correcte dans **Paire de clés SSH**. Si aucune paire de clés n'est disponible, vous pouvez en créer ou en importer une.
6. Cliquez sur **OK** pour fermer la boîte de dialogue Options de site.
7. Cliquez sur **Fermer** pour fermer la boîte de dialogue Gestionnaire de sites.

Quand vous utilisez ce profil pour vous établir une connexion, le client tente automatiquement d'établir une connexion SSH au port 22.

Génération d'une paire de clés SSH

1. Dans la fenêtre principale, sélectionnez **Outils > Options**. La boîte de dialogue Options du programme s'affiche.
2. Sélectionnez **SSH > Clés client**.
3. Cliquez sur **Créer**. L'Assistant Génération de paire de clés de client SSH s'affiche.
4. Suivez les invites à l'écran pour mettre fin à l'exécution de l'assistant.

Exportation d'une clé publique SSH

1. Dans la fenêtre principale, sélectionnez **Outils > Options**. La boîte de dialogue Options du programme s'affiche.
2. Sélectionnez **SSH > Clés client**.
3. Cliquez sur **Exporter**. La boîte de dialogue Enregistrer sous... s'affiche.
4. Entrez un nom de fichier, puis cliquez sur **Enregistrer**.

OpenPGP

Chapitre 4

Le protocole OpenPGP peut être utilisé avec FTP pour améliorer la sécurité. Ce chapitre décrit le fonctionnement de OpenPGP dans Ipswitch WS_FTP Professional, présente les étapes du transfert de fichier avec le cryptage OpenPGP, et propose un scénario illustrant comment OpenPGP peut aider à résoudre un problème commercial standard.

REMARQUE : Ipswitch WS_FTP Professional contient un logiciel basé sur les normes définies dans « Proposed Standard RFC 2440 » du OpenPGP Working Group de la Internet Engineering Task Force (IETF). Ipswitch WS_FTP Professional est compatible avec les clés OpenPGP, PGP ou GPGP.

Aperçu

OpenPGP est une méthode de cryptage de fichiers à base de clés. Grâce à cette méthode, seul le destinataire peut recevoir et décrypter les fichiers. OpenPGP permet de sécuriser les communications par messagerie électronique, mais sa technologie est également applicable à FTP.

OpenPGP utilise deux clés cryptographiques pour sécuriser les fichiers. Une clé publique sert à crypter le fichier et seule la clé privée correspondante peut le décrypter.

REMARQUE : à la différence de SSL et SSH, OpenPGP n'est pas un type de connexion, mais une méthode de cryptage de fichier avant sa télétransmission. Par conséquent, le mode OpenPGP peut être utilisé conjointement avec les connexions FTP, SSL ou SSH standard.

Voici une illustration étape par étape du fonctionnement de OpenPGP avec FTP.

Étape 1 : Le fichier à télétransmettre est crypté avec une clé publique préalablement fournie par le destinataire.

Étape 2 : Le fichier crypté est télétransmis au serveur FTP.

Étape 3 : Le destinataire récupère le fichier sur le serveur FTP.

Dans ce chapitre :

Aperçu

Comment activer le mode OpenPGP ?

Comment activer le mode OpenPGP pour un site par défaut ?

Génération de paire de clés

Importation d'une clé

Exportation d'une paire de clés

Scénario

Étape 4 : Avec la clé privée (qui, avec la clé publique utilisée pour crypter le fichier, fait partie initialement de la paire de clés), le destinataire décrypte le fichier et accède à son contenu.

Comment activer le mode OpenPGP ?

Le mode PGP est activé une fois la connexion établie avec le serveur.

1. Dans Ipswitch WS_FTP Professional, sélectionnez l'onglet du serveur distant.
2. Sélectionnez **Outils > Mode OpenPGP** ou cliquez sur **Mode OpenPGP** dans la barre d'outils. La boîte de dialogue Mode OpenPGP s'affiche.
3. Sélectionnez la méthode de transfert OpenPGP préférée dans la liste d'options :
   - **Crypter** les fichiers avec une clé de votre porte-clés. Sélectionnez les **clés de cryptage** à utiliser pour crypter les fichiers.
   - **Signer** les fichiers avec votre clé privée comme signature numérique. Sélectionnez la **clé de signature** à utiliser pour signer les fichiers. Entrez la **Phrase de passe** de clé de signature.
   - Sélectionnez **Crypter et Signer** pour utiliser les deux options en même temps.
4. Cliquez sur **OK** pour fermer la boîte de dialogue. Le mode OpenPGP est alors activé pour la durée de la connexion ou jusqu'à sa désactivation.

Comment activer le mode OpenPGP pour un site par défaut ?

REMARQUE : il vous faut au moins une clé OpenPGP dans votre porte-clés pour pouvoir configurer un site pour l'activation automatique du mode OpenPGP à chaque connexion.

1. Dans la fenêtre principale, sélectionnez **Connecter > Gérer les sites**. La boîte de dialogue Gérer les sites s'affiche.
2. Dans la liste, retrouvez et sélectionnez le site pour lequel vous voulez activer automatiquement le mode OpenPGP lors de la connexion. Cliquez sur **Modifier**. La boîte de dialogue Options de site s'affiche.
3. Développez **Avancé** et cliquez sur **OpenPGP**. Les options OpenPGP s'affichent.
4. Sélectionnez **Utiliser le Mode de transfert OpenPGP après la connexion**.

5 Sélectionnez la méthode de transfert OpenPGP préférée dans la liste d'options :

- Cryptez les fichiers avec une clé de votre porte-clés. Sélectionnez les clés de cryptage à utiliser pour crypter les fichiers.
- Signez les fichiers avec votre clé privée comme signature numérique. Sélectionnez la clé de signature à utiliser pour signer les fichiers. Entrez la phrase de passe de clé de signature.
- Sélectionnez **Crypter et Signer** pour utiliser les deux options en même temps.

6 Cliquez sur **OK** pour enregistrer les valeurs et fermez la boîte de dialogue.

Génération d'une paire de clés

1 Dans la fenêtre principale, sélectionnez **Outils > Options**. La boîte de dialogue Options du programme s'affiche.

2 Sélectionnez **OpenPGP > Clés**.

3 Cliquez sur **Créer**. L'Assistant Génération de clé OpenPGP s'affiche.

4 Suivez les invites à l'écran pour exécuter le processus de création de clés.

Importation d'une clé

1 Dans la fenêtre principale, sélectionnez **Outils > Options**. La boîte de dialogue Options du programme s'affiche.

2 Sélectionnez **OpenPGP > Clés**.

3 Cliquez sur **Importer**. L'Assistant Importation de clé OpenPGP s'affiche.

4 Suivez les invites à l'écran pour exécuter le processus d'importation de clés.

Exportation d'une paire de clés

1 Dans la fenêtre principale, sélectionnez **Outils > Options**. La boîte de dialogue Options du programme s'affiche.

2 Sélectionnez **OpenPGP > Clés**.

3 Sélectionnez la clé à exporter, puis cliquez sur **Exporter**. L'Assistant Exportation de clés OpenPGP s'affiche.

4 Suivez les instructions à l'écran pour exporter vos clés.

Scénario

L'exemple réel suivant démontre l'emploi de OpenPGP pour une tâche courante.

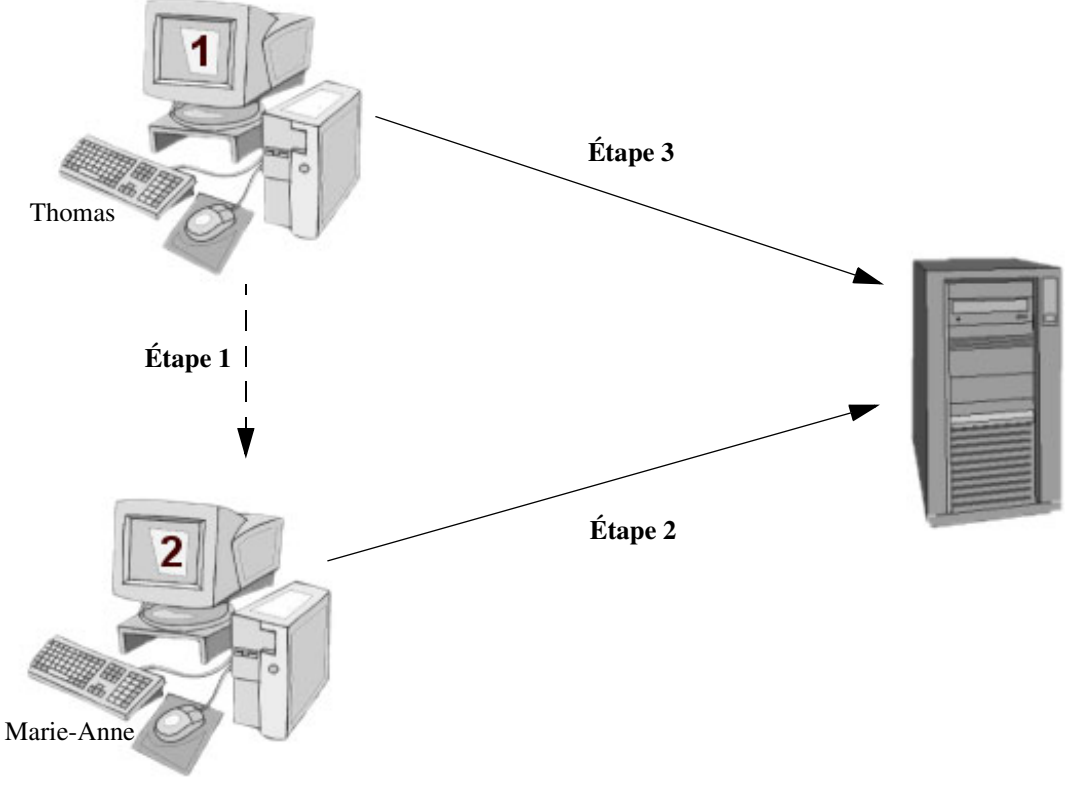
Tâche

Marie-Anne travaille dans un bureau régional de sa société. Elle doit envoyer régulièrement des enregistrements confidentiels concernant les employés à Thomas qui se trouve au siège social. Par le passé, elle gravait les fichiers sur des CD qu'elle envoyait par la poste. Elle doit trouver une solution qui lui permette de réduire les frais et de remettre les fichiers à Thomas dans de meilleurs délais.

Problèmes

Les informations compilées sont trop volumineuses pour la messagerie électronique. Le transfert doit aussi être sécurisé pour que seul Thomas ait accès aux informations.

Solution



1. Thomas envoie sa clé publique à Marie-Anne par courrier électronique. Il peut utiliser Ipswitch WS_FTP Professional pour générer ou exporter une clé.
2. Marie-Anne crypte et télétransmet le fichier en utilisant la clé publique de Thomas.
   a. Elle importe la clé dans Ipswitch WS_FTP Professional.
   b. Elle se connecte au serveur FTP de la société.
   c. Elle active le mode OpenPGP en utilisant la clé de Thomas pour le cryptage.
   d. Elle télétransmet le fichier.
3. Thomas télécharge et décrypte le fichier avec sa clé privée.

Utilisation de pare-feux

Chapitre 5

Certaines organisations installent un pare-feu (ou passerelle) pour isoler leur réseau local du reste de l'Internet. Un pare-feu consiste en matériel ou logiciel configuré pour bloquer des types précis d'accès ou d'informations, et empêcher leurs entrées dans le réseau. La plupart des pare-feux bloquent l'accès au réseau local depuis l'extérieur, tout en permettant aux individus dans le réseau d'accéder à la plupart des ressources extérieures.

Ipswitch WS_FTP Professional vous permet de paramétrer et de créer une configuration de pare-feux que vous pouvez ensuite utiliser lors de la connexion à un site FTP lorsque ce pare-feu est actif. Vous configurez une seule fois le pare-feu et vous pouvez ensuite associer cette configuration à tout site pour lequel elle s'avère nécessaire.

L'éditeur FireScript vous permet d'éditer des scripts de pare-feux pour les adapter à vos besoins. Pour plus de détails, reportez-vous à Annexe A : « Éditeur FireScript » à la page 27.

Multiples pare-feux

La création de multiples configurations de pare-feux est nécessaire dans certains cas. Ainsi, si vous utilisez un ordinateur portable à différents sites équipés de pare-feux différents, vous pouvez créer une configuration différente pour chaque, ce qui facilite votre travail en cas de déplacement.

Vous voudrez peut-être aussi définir plusieurs configurations de pare-feux si votre réseau local comporte plusieurs routeurs configurés comme pare-feux. Dans ce cas, vous associerez une configuration de pare-feux différente à un site FTP selon l'emplacement de votre lieu de travail dans le réseau.

En outre, vous aurez peut-être un certain nombre de sites sécurisés (par exemple, des sites FTP de votre société) exigeant chacun un pare-feu différent (ou aucun pare-feu).

Dans ce chapitre :

Multiples pare-feux

Types de pare-feux

Configuration d'un pare-feu

Utilisation d'un pare-feu configuré

Utiliser UPnP

Types de pare-feux

Le tableau suivant présente tous les types conventionnels de pare-feux et les informations que vous devrez entrer pour chacun dans Ipswitch WS_FTP Professional.

Type de pare-feu	Informations à entrer dans Ipswitch WS_FTP Professional
Proxy OPEN	Nom d'hôte (ou adresse IP)
SITE hostname	Nom d'hôte (ou adresse IP), Nom d'utilisateur (ID)
Transparent	Nom d'utilisateur (ID), Mot de passe
USER after logon	Nom d'hôte (ou adresse IP), Nom d'utilisateur (ID), Mot de passe
USER fireID@remoteHost	Nom d'hôte (ou adresse IP), Nom d'utilisateur (ID), Mot de passe
USER remoteID@fireID @remoteHost	Nom d'hôte (ou adresse IP), Nom d'utilisateur (ID), Mot de passe
USER remoteID @remoteHost fireID	Nom d'hôte (ou adresse IP), Nom d'utilisateur (ID), Mot de passe
USER with no logon	Nom d'hôte (ou adresse IP)
SOCKS4 et SOCKS5	Nom d'hôte (ou adresse IP), Nom d'utilisateur (ID), Mot de passe

Configuration d'un pare-feu

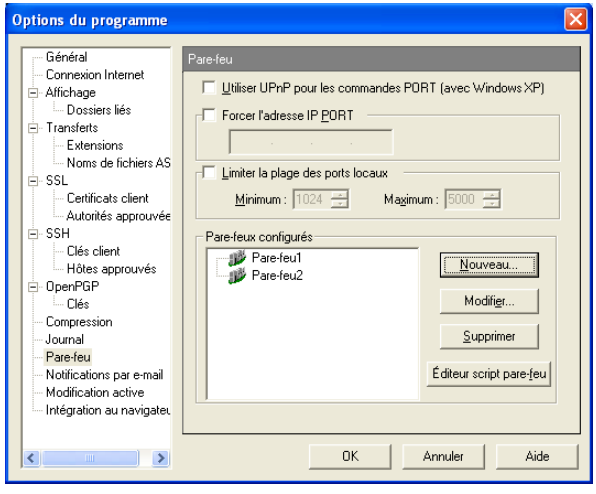
Les informations de configuration de pare-feu sont fournies par l'administrateur réseau. Pour plus de détails, reportez-vous à **Types de pare-feux** ci-dessus.

REMARQUE : pour certains pare-feux basés sur routeur, vous voudrez sans doute utiliser le mode passif. Dans ce mode, les connexions pour données sont établies par le client FTP (Ipswitch WS_FTP Professional) au lieu du site FTP.

Pour configurer un pare-feu :

1. Sélectionnez **Outils > Options**.
2. Sélectionnez le dialogue **Pare-feu**.
3. Cliquez sur **Nouveau**.
4. Suivez les instructions de l'assistant **Nouveau pare-feu**.

5 Quand vous cliquez sur **Terminer**, le pare-feu est ajouté à la liste **Pare-feux configurés**.



Vous pouvez alors associer la configuration de pare-feux au site, comme décrit dans **Utilisation d'un pare-feu configuré** ci-dessous.

Utilisation d'un pare-feu configuré

Après avoir configuré un pare-feu, vous pouvez appliquer la configuration de pare-feux à un site FTP.

Dans le Gestionnaire de sites :

1 Sélectionnez un site.

2 Cliquez sur le bouton **Modifier**.

3 Sélectionnez l'onglet **Avancé**.

4 Sélectionnez le pare-feu configuré dans la liste déroulante **Pare-feu**.

Utilisation de UPnP

Si vous utilisez Windows XP, vous pourrez peut-être configurer automatiquement votre pare-feu pour l'ouverture des ports nécessaires et obtenir l'adresse IP externe à l'aide de UPnP.

Pour activer UPnP :

1. Cliquez sur **Options** dans la barre d'outils ou sélectionnez **Outils > Options** dans le menu. La boîte de dialogue Options du programme s'affiche.
2. Sélectionnez **Pare-feu**.
3. Sélectionnez **Utiliser UPnP pour les commandes PORT**.

Éditeur FireScript

Annexe A

Cette annexe décrit le rôle et la syntaxe du langage FireScript et son emploi pour établir une connexion FTP à travers un pare-feu.

Dans ce chapitre :

Rôle d'un script FireScript

Composants de script FireScript

Séquence de connexion

Variables FireScript

Expansion de chaîne

Expressions de fonction

Instructions de script FireScript

Instructions de commutation

Instructions case

Continue

Sauts et étiquettes

Return

Autodetect

Instructions SSL

Mots clés FireScript

Rôle d'un script FireScript

Un script FireScript permet de personnaliser la séquence de commandes et réponses utilisées pour la connexion à un serveur FTP. Cette personnalisation peut s'avérer nécessaire si votre serveur FTP requiert l'émission de commandes non standards avant ou après la connexion, ou si certains types de pare-feux se trouvent entre le client et le serveur.

Les scripts FireScript sont créés dans le langage FireScript, adapté et développé spécifiquement pour Ipswitch WS_FTP Professional. Les scripts FireScript peuvent exécuter les fonctions que Ipswitch WS_FTP Professional utilise en interne pour connecter à un type d'hôte ou pare-feu. Ils vous permettent toutefois de déterminer si ces fonctions sont utilisées, et à quel moment. Plus précisément, ils déterminent quand procéder à la détection automatique du type d'hôte, et quand sécuriser une connexion SSL. Ils déterminent s'il faut tenter d'émettre la commande XAUTH, et s'il est nécessaire de se connecter à un compte utilisateur après l'envoi de l'ID utilisateur et du mot de passe.

Composants de script FireScript

Un script FireScript est divisé en trois sections : **fwsc**, **commentaire** et **script**. Comme dans un fichier ini de Windows, le nom de la section figure seul sur une ligne, entre crochets, suivi du reste de la section.

La section **fwsc** a une structure interne consistant en paires nom=valeur comme une section ini dans Windows. Elle contient des informations d'identification sur le script et indique les variables requises par le script.

La section **comment** consiste en texte libre à l'intention du personnel technique. Elle n'est pas prise en compte par l'exécutable de script.

La section **script** contient la portion exécutable des scripts et est conforme à la syntaxe FireScript.

Voici un exemple FireScript démontrant cette structure.

[fwsc]

```
author=Ipswitch
connectto=firewall
```

… les autres valeurs possible incluraient en général 'required=' et 'version='

[comment]

Voici un exemple de script pour se connecter à un serveur proxy FTP. Il est incomplet car plusieurs commandes indispensables à la connexion ont été enlevées à des fins de clarification. Le but principal est de démontrer l'organisation du script FireScript en trois sections.

[script]
send ("OPEN %HostAddress") {}
tryssl;
send ("USER %HostUserId")
{
case (300.0,399) :

```
continue ;

case any :

return (false) ;
}
```

… la majeure partie du script n'est pas montrée en raison de sa taille.

```
label success;
gossl;
return (true);
```

Section fwsc

La section **fwsc** permet de spécifier des informations sur le script comme dans un fichier ini de Windows. La plupart des paramètres sont présents pour votre information. Ceci inclut les champs **author** et **version**. Quelques paramètres sont utilisés par l'exécutif de script pour déterminer s'il faut afficher ou non la boîte de dialogue de connexion, et quelle adresse IP doit être utilisée.

L'analyseur reconnaît et stocke les valeurs pour les paramètres suivants :

Paramètres de fwsc	
Paramètre	Signification et valeurs
author	Pour information uniquement. Auteur du script FireScript.
version	Pour information uniquement. Numéro de version du fichier de script.
verdate	Pour information uniquement. Date à laquelle cette version a été mise à jour.
required	Une liste de champs séparés par une virgule qui doivent être présents pour l'exécution du script FireScript. La boîte de dialogue de connexion s'affiche si tous les champs requis ne sont pas présents, et le bouton **Connecter** est désactivé jusqu'à ce que tous les champs requis aient été remplis.
preask	Une liste de champs facultatifs séparés par une virgule. En son absence, la boîte de dialogue de connexion s'affiche.
connectto	'firewall' ou 'host'. Ce paramètre indique à Ipswitch WS_FTP Professional, l'adresse IP à utiliser pour établir la connexion.

Les paramètres non reconnus sont ignorés.

Section comment

Utilisez la section **comment** pour décrire les actions du script FireScript. Le code du script FireScript doit être bien décrit, pour faciliter sa compréhension et les mises à jour ultérieures. L'exécutif FireScript ignore la section comment.

Vous pouvez aussi insérer des commentaires dans la section script en utilisant le délimiteur de commentaire '//' comme dans C++ et Java. Le texte présent sur une ligne qui suit la séquence '//' est ignorée par l'analyseur.

Section script

La section **script** consiste en une séquence d'instructions qui envoie des commandes au pare-feu ou au serveur FTP. Certaines de ces instructions ont un résultat ou déclenchent une réponse du pare-feu ou du serveur FTP. Il existe une structure de contrôle simple qui permet au script d'utiliser différents chemins d'exécution, d'après ce résultat ou cette réponse.

Séquence de connexion

Une demande de connexion à un site FTP provient d'actions de l'utilisateur dans l'interface Classic ou Explorer, ou d'un des utilitaires de Ipswitch WS_FTP Professional (par exemple, Recherche ou Synchronisation). Parfois, des connexions supplémentaires sont demandées par le Gestionnaire de transferts pour reprendre ou accélérer des transferts. Toutes les connexions sont créées par la fonction CreateConnection dans l'interface de programmation d'applications (API) de Ipswitch WS_FTP Professional.

La séquence de connexion consiste en deux phases.

- Phase 1 : établir la connexion au pare-feu ou au serveur FTP.
- Phase 2 : envoyer des commandes pour se connecter et autoriser l'utilisateur connecté. C'est lors de cette phase que les commandes FireScript sont exécutées.

La première phase fonctionne de la même manière, que Ipswitch WS_FTP Professional utilise un script FireScript ou un de ses types de pare-feu interne. Avant d'exécuter le script, Ipswitch WS_FTP Professional vérifie la section **fwsc** pour retrouver la liste de champs marqués **required** et **preask**. Si un seul manque, la boîte de dialogue d'accès s'affiche. Si l'utilisateur spécifie toutes les informations requises et appuie sur **Connecter**, Ipswitch WS_FTP Professional vérifie le champ **connectto**. En fonction de ce champ, Ipswitch WS_FTP Pro établit une connexion à l'adresse IP et au port du pare-feu, ou à l'adresse IP et au port du serveur FTP. Si ce champ est absent, Ipswitch WS_FTP Professional utilise par défaut l'adresse IP du pare-feu, si celui-ci est présent.

Une fois la connexion établie avec succès et un socket valide ouvert, Ipswitch WS_FTP Professional appelle l'exécutif FireScript pour exécuter le script FireScript. Si le script FireScript se connecte correctement et renvoie un code de succès, la fonction CreateConnection renvoie la connexion autorisée à l'appelant.

Langage FireScript

Le langage FireScript contient une version limitée d'éléments qui vous est peut-être déjà familière si vous avez créé des scripts ou programmes dans d'autres langages. Il utilise des variables, déclarations et instructions pour réaliser des actions et diriger le flux du programme. Chacun de ces éléments est décrit dans les sections qui suivent.

Sur le plan de la syntaxe, les instructions FireScript se terminent par un point-virgule. Elles peuvent donc occuper plusieurs lignes, et plusieurs instructions peuvent être présentes sur une ligne. Une chaîne, par contre, ne peut recouper plusieurs lignes. Le guillemet de fermeture final doit figure sur la même ligne de code source que le guillemet d'ouverture. Ainsi, le code ci-après est valide :

```
contains
(

lastreply,
"Bienvenue à mon site FTP"
)
;
```

mais le code qui suit n'est pas valide :

```
contains ( lastreply, "Bienvenue à
mon site FTP" );
```

Variables FireScript

Les scripts Firescript opèrent avec les informations d'accès fournies par Ipswitch WS_FTP Professional. Ceci inclut aux moins les ID d'utilisateur et les mots de passe, l'adresse IP et le port du serveur FTP, et parfois l'adresse IP et le port du pare-feu. Ces champs sont souvent lus depuis un profil de site, une URL FTP ou depuis la ligne de commande. Comme décrit précédemment, si certaines des informations requises manquent, la séquence de connexion présente la boîte de dialogue de connexion pour que l'utilisateur puisse les entrer de manière interactive. L'exécutif de script stocke ces informations dans un ensemble de variables intrinsèques avant d'entamer l'exécution. Il existe en outre des variables intrinsèques qui contiennent le résultat de la dernière commande émise. Celles-ci sont définies par l'exécutif de script après l'exécution de telles instructions.

La syntaxe d'utilisation d'une variable dépend de l'instruction ou de l'expression où elle est utilisée. Voici une liste de toutes les variables intrinsèques :

Variables intrinsèques de script FireScript	
Variable	Signification et usage
FwUserId	L'ID de l'utilisateur sur le pare-feu. Certains pare-feux exigent que les utilisateurs se connectent au pare-feu pour que d'autres connexions puissent être établies à travers celui-ci.
FwPassword	Le mot de passe de l'utilisateur sur le pare-feu. Requis si l'utilisateur doit se connecter au pare-feu.
FwAccount	Compte sur le pare-feu. Requis si l'utilisateur doit spécifier un compte sur le pare-feu. Rarement utilisé, mais inclus si besoin est.
FwAddress	L'adresse IP du pare-feu. Requise si l'utilisateur doit se connecter au pare-feu, pour que celui-ci se connecte à son tour au serveur FTP et serve donc de proxy (mandataire).
HostUserId	L'ID de l'utilisateur sur le serveur FTP. Presque toujours requis. Spécifiez 'anonymous' si l'utilisateur n'a pas d'ID d'utilisateur sur le serveur.
HostPassword	Le mot de passe de l'utilisateur sur le serveur FTP. Presque toujours requis conjointement avec un ID d'utilisateur. Utilisez votre adresse de messagerie électronique comme mot de passe lors de l'emploi de 'anonymous' comme ID d'utilisateur.
HostAccount	Le compte de l'utilisateur sur le serveur FTP. Pour accéder à certaines informations sous certains systèmes d'exploitation, les serveurs FTP sur ces systèmes requièrent l'envoi d'un compte après une connexion réussie avec ID d'utilisateur et mot de passe.
HostAddress	L'adresse IP de l'hôte. L'exécutif de script peut se connecter directement à cette adresse, ou envoie l'adresse à un pare-feu qui agit alors en tant que proxy (mandataire).
LastFtpCode	Le code numérique à 3 chiffres de la dernière réponse reçue du serveur FTP ou du pare-feu. Par exemple, après une connexion réussie, LastFtpCode est égal à 230.
LastReply	Le texte de la dernière réponse du serveur. Par exemple, « 230 user logged in » (230 utilisateur connecté)

Les scripts FireScript ne nécessitent pas et n'utilisent pas les variables définies par l'utilisateur. Il n'y a donc pas de déclarations de variables. En outre, comme le script FireScript ne peut définir directement la valeur d'une des variables intrinsèques, les instructions d'affectation sont inutiles.

Expansion de chaîne

Certaines commandes et fonctions du langage FireScript acceptent des chaînes comme arguments. Vous pouvez alors leur passer une variable chaîne ou un littéral chaîne entre doubles guillemets, par exemple « Voici une chaîne ». Pour placer un double guillemet à l'intérieur d'une chaîne, faites-le précéder du signe de pourcentage '%'. Le signe de pourcentage '%' est un caractère d'échappement qui permet d'inclure des variables et des guillemets dans les chaînes.

La séquence %% est remplacée par un seul %.
La séquence %" est remplacée par ".
% suivi du nom d'une variable est remplacé par la valeur de la variable.

Par exemple, l'instruction de script suivante :

```
send ("OPEN %HostAddress")
```

Si HostAddress est égal à « ftp.ipswitch.com » lors de l'invocation de ce script, la commande est étendue pour donner :

```
send ("OPEN ftp.ipswitch.com")
```

l'expression,

```
contains (lastreply, "%% full")
```

est étendue lors de l'exécution pour donner :

```
contains(lastreply "% full")
```

et l'instruction

```
send ("SITE SETLOG %"f:\log files\access.log%" -clear")
```

la chaîne étendue envoyée est :

```
SITE SETLOG "f:\log files\access.log" -clear
```

Passer une variable chaîne équivaut (mais en plus rapide) à passer un littéral de chaîne qui étend la variable.

Exemple :

```
isempty(FwPassword)
```

équivaut (mais en plus rapide) à

```
isempty("%FwPassword")
```

Expressions de fonction

Actuellement, le langage FireScript ne permet pas d'expressions entièrement développées. Il inclut deux expressions de fonction avec quelques opérateurs booléens pour évaluer l'état des variables. Il s'agit de **contains** et **isempty**. Les opérateurs booléens supportés sont **not** et **and**.

La fonction **contains** accepte deux chaînes et renvoie **true** si la seconde est trouvée dans la première. La recherche est sensible à la casse. Les deux chaînes sont d'abord étendues.

La fonction **isempty** accepte une chaîne et renvoie **true** s'il existe des caractères dans la chaîne. Vous pouvez l'utiliser pour tester si une valeur a été spécifiée pour une des variables intrinsèques.

L'opérateur booléen **not** inverse la valeur renvoyée par l'expression de fonction.

Exemple :

Si la variable HostAccount contient la valeur 'usr987i'
isempty (HostAccount) renvoie false mais
not isempty(HostAccount) est évalué comme true.

L'opérateur booléen **and** requiert que toutes les conditions spécifiées soient true.

Par exemple, si la variable HostAccount contient une valeur telle que 'usr987I'
La dernière réponse du serveur est "230 User logged in, please send account"

l'expression suivante est alors évaluée comme true :

```
case (200..299) and not isempty(HostAccount) and
contains(lastreply, "ACCOUNT") :
```

Instructions de script FireScript

Le langage FireScript inclut plusieurs types d'instruction. Les instructions entraînent l'exécution d'actions, ou dirigent le flux d'exécution du script. Les sections qui suivent décrivent les types d'instruction.

Instructions de commutation

L'instruction **send** et l'instruction **xauth**, appelées instruction de commutation, impliquent une instruction de commutation immédiate d'après la réponse du serveur. L'instruction de commutation contient des instructions **case** similaires aux instructions case de Java et C++, sauf que les conditions ne sont pas des constantes vérifiées par rapport à une expression unique.

Une instruction de commutation telle que **send** et **xauth** est toujours suivie immédiatement d'un ensemble d'instructions case entre accolades { <instructions case> }. L'ensemble d'instructions case peut être vide, auquel cas il n'y a rien entre les accolades, mais les accolades doivent être présentes.

Exemple d'instruction de commutation :

```
send ("USER %FwUserId") {}
```

L'instruction **send** accepte un argument unique, la chaîne à envoyer au serveur. La chaîne est étendue avant son envoi. La longueur maximum autorisée pour la chaîne étendue est d'environ 512 octets, la longueur maximum d'une ligne FTP. La commande send attend alors une réponse du serveur et évalue la réponse par rapport aux conditions dans chacune des instructions case incluses.

L'instruction **xauth** n'accepte aucun argument. Elle examine la bannière de bienvenue pour détecter une invitation xauth fournie par le serveur Ipswitch WS_FTP. Si elle n'est pas connectée au serveur Ipswitch WS_FTP ou ne trouve pas l'invitation, **xauth** reste sans effet, et les instructions case ne sont pas évaluées. Si elle trouve l'invitation, elle code l'ID utilisateur et le mot de passe et envoie la commande xauth au serveur. Elle attend ensuite la réponse et l'évalue par rapport aux instructions case, comme la commande **send**.

Instructions case

Les instructions Case sont incluses dans des instructions de commutation. Une instruction case contient une liste de l'ensemble de conditions auxquelles la réponse du serveur doit satisfaire pour que le cas considéré soit activé.

La liste de conditions est suivie d'un deux-points ':'.

Les instructions case sont traitées dans l'ordre où elles se présentent jusqu'à ce que la première correspondance soit trouvée.

Dès qu'une correspondance est trouvée pour les conditions dans une instruction case, les instructions emboîtées sont exécutées.

Une condition case peut être une liste de codes et plages de codes FTP, une expression de fonction, ou un des cas spéciaux, **any** et **timeout**.

Si un cas inclut une liste de codes/plages de codes ftp, la liste doit apparaître en premier, suivie d'une expression de fonction. La liste avec virgules de séparation est entre parenthèses. Chaque élément dans la liste doit être un code unique à 3 chiffres, ou une plage spécifiée par deux codes à 3 chiffres, séparés par deux points de suspension '..'. La plage est inclusive et il est recommandé de spécifier la limite inférieure en premier.

Les cas spéciaux **any** et **timeout** doivent apparaître seuls.

Exemples d'instruction case

La condition case qui suit correspondra si le code ftp renvoyé est 226 ou 231.

```
case (226, 231) :
```

Les conditions case qui suivent correspondront si le code ftp renvoyé est 226 ou 231, ou entre 250 et 299 (inclus). Ainsi 250 correspondra, de même que 251, 252 etc. jusqu'à 299

```
case (226, 231, 250..299) :
```

Les conditions case qui suivent correspondent si le code ftp renvoyé est dans les 300 et que la chaîne renvoyée contient le texte « adresse e-mail ».

```
case (300..399) and contains(lastreply, "adresse e-mail") :
```

Les conditions case qui suivent correspondent si le code ftp renvoyé est 500 ou supérieur et que la chaîne renvoyée contient le message d'erreur spécifié.

```
case (500..999) and contains(lastreply, "user %HostUserId
cannot login.") :
```

Si une instruction case contient plusieurs conditions, celles-ci doivent être séparées par **and**. L'opérateur **and** spécifie que toutes les conditions dans la liste doivent être satisfaites. Ainsi, dans l'exemple précédent, le code ftpcode doit être entre 500 et 599 ET la dernière réponse doit aussi contenir la chaîne spécifiée. Les deux doivent être true. Si une condition est false, l'instruction case n'a pas de correspondance.

L'opérateur **not** inverse le résultat d'une fonction. Ainsi, vous voudrez peut-être vous assurer que la dernière réponse ne contient pas une chaîne précise. Par exemple :

```
case (500..599) and not contains(lastreply, "server is busy") :
```

Il n'y a pas d'opérateur **or**. La même logique peut être appliquée à l'aide de plusieurs instructions case.

La condition case qui suit correspondra si le délai de la commande send vient à expiration.

```
case timeout :
```

Si elle est présente, la condition case **any**, couvrant tous les cas, doit figurer en dernier dans la liste incluse. Si elle est suivie d'autres instructions case, celles-ci ne seront jamais évaluées.

Par exemple, la condition case suivante aura toujours une correspondance.

```
case any:
```

Si des instructions case se chevauchent et que deux instructions case correspondent à la réponse, la première rencontrée est exécutée.

Exemple :

```
case (200..299) and contains(lastreply, "please send user
account") :

...
case (200..299) :
...
```

Si la condition case avec la fonction contains apparaît après celle sans la condition, elle n'est jamais évaluée.

Continue

À la différence de C et C++, l'exécution à l'intérieur d'une instruction case ne passe pas automatiquement à l'instruction case suivante. Seules les instructions figurant sous l'instruction case activée sont exécutées. L'exécution se poursuit alors à l'instruction suivante après l'instruction de commutation qui la renferme. L'instruction **continue** saute à l'instruction qui suit l'instruction de commutation qui la renferme. Elle opère comme une instruction break, dans une instruction switch C/C++, sauf qu'elle n'est pas absolument nécessaire.

Les instructions de commutation ne peuvent être emboîtées. Par conséquent, les instructions **send** et **xauth** ne peuvent figurer à l'intérieur d'une instruction **case**.

Sauts et étiquettes

Une instruction de saut transfère l'exécution à une partie différente du script. La destination du saut doit être définie par une étiquette qui figure aussi dans le script. Les exemples de script FireScript Ipswitch utilisent des sauts à différentes séquences de code à partir d'instructions case. Par conséquent, le code exécuté dépend de l'instruction case qui est activée.

Une déclaration d'étiquette comprend le mot label, suivi du nom de l'étiquette et d'un point-virgule.

Une instruction de saut comprend le mot jump, suivi du nom de la destination de saut et d'un point-virgule.

Une étiquette ne peut figurer à l'intérieur d'une instruction case. Vous ne pouvez pas sauter à l'intérieur d'une instruction case.

Return

L'instruction return opère comme une fonction car elle accepte un seul paramètre, true ou false pour indiquer le succès ou l'échec. Elle met fin à l'exécution du script et retourne à l'appelant. Si elle renvoie true, la connexion suppose la réussite de l'accès et une autorisation accordée. Si elle renvoie false, l'appelant peut essayer à nouveau ou abandonner la connexion.

Autodetect

L'instruction **autodetect** examine la dernière réponse du serveur pour aider à déterminer le type d'hôte du serveur FTP auquel il est connecté. Autodetect examine normalement la bannière de bienvenue. Par conséquent, l'instruction doit être placée immédiatement après le renvoi de la bannière de bienvenue. Voici deux exemples de bannière renvoyée depuis deux serveurs FTP connus. Autodetect détecterait le premier comme étant un serveur Microsoft NT, et le second comme étant un serveur Ipswitch WS_FTP.

```
220 tstsrvnt Microsoft FTP Service (Version 3.0).

220 tstsrvws X2 WS_FTP Server 1.0.5 (1737223651)
```

Si la connexion a été établie directement au serveur FTP hôte et que la bannière de bienvenue a déjà été renvoyée avant le début d'exécution du script, **autodetect** doit être la première instruction dans le script. Si la connexion a été établie avec le pare-feu et que la bannière de bienvenue du serveur ftp hôte est disponible plus loin dans le script, l'instruction **autodetect** doit être placée à cet endroit. Si le pare-feu avale ou remplace la bannière de bienvenue depuis l'hôte ftp ou que, pour une autre raison, le client ftp ne voit jamais la bannière de bienvenue, omettez l'instruction **autodetect**. Ipswitch WS_FTP Professional tente alors de déterminer le type d'hôte après l'exécution du script.

Autodetect est sans effet si le type d'hôte dans le profil de site est défini à une valeur autre que 'Autodétection'. L'instruction **autodetect** ne renvoie pas de valeur et ne change pas le flux du script.

Instructions SSL

Les commandes **tryssl** et **gossl** tentent d'ouvrir un canal sécurisé avec le serveur via SSL. La différence entre les deux commandes est la suivante : si **gossl** échoue, l'exécution du script se termine et celui-ci renvoie false, alors que si **tryssl** échoue, l'exécution du script continue. Les commandes peuvent apparaître plus d'une fois dans le programme. Si une connexion sécurisée n'a pas été demandée, ou a déjà été établie, les commandes sont sans effet. Si les commandes ne sont pas sécurisées, elle affichent une boîte de message demandant à l'utilisateur s'il désire continuer sans protection, essayer à nouveau d'obtenir SSL plus loin dans la séquence, ou abandonner la connexion. Si l'utilisateur choisit de continuer sans protection, les appels futurs à **tryssl** ou **gossl** sont sans effet.

Quand l'exécution du script se termine, l'exécutif de script vérifie le statut SSL de la connexion pour s'assurer qu'une demande de connexion sécurisée a été honorée. Dans le profil de site, si l'utilisateur a sélectionné **Utiliser SSL**, l'exécutif de script émet un avertissement à son intention si la connexion n'est pas sécurisée. Il peut alors abandonner la connexion. Cet avertissement est émis si une connexion sécurisée a été tentée et que l'utilisateur a choisi de continuer sans protection.

Le placement des tentatives d'ouverture de canal SSL peut s'avérer très important, selon le type de pare-feu à travers lequel le script établit la connexion.

Mots clés FireScript

Voici une liste complète de tous les mots clés utilisés et interprétés par le langage. Vous ne pouvez pas les utiliser comme étiquettes.

gossl	tryssl	autodetect
send	xauth	case
continue	and	not
any	timeout	return
jump	label	true
false		

Mots réservés FireScript

Les mots suivants sont réservés pour les versions futures du langage et de l'analyseur. Vous ne devez pas les utiliser comme étiquettes.

switch	if	for
next	while	loop
break	function	int
bool	string	var
password	ou	

Instructions FireScript

gossl	tryssl	autodetect
send	xauth	jump
return	continue	

Fonctions intrinsèques FireScript

contains	isempty

Variables intrinsèques FireScript

FwUserId	FwPassword	FwAccount
FwAddress	HostUserId	HostPassword
HostAccount	HostAddress	LastFtpCode
LastReply		

F

FireScript 27

P

pare-feu 23
passerelles 23
PGP
- activation du mode PGP 18
- activation du mode PGP pour un site par défaut 18
- aperçu 17
- génération d'une paire de clés 19
- importation d'une clé 19

Proxy OPEN (pare-feu) 24

S

SITE hostname (pare-feu) 24
SSH
- génération d'une paire de clés SSH 14

SSL
- autorités approuvées 8
  - ajout d'un certificat 9
  - exportation d'un certificat 9
  - suppression d'un certificat 10
- génération d'un certificat 6
- sélection d'un certificat 8

SSL (définition)
- certificat 4
- clé de session 4
- clé privée 4
- clé publique 4
- client 4
- demande de signature de certificat 4

T

Transparent (pare-feu) 24
types de pare-feu 24

U

UPnP 26
USER fireID@remoteHost (pare-feu) 24
USER remoteID @remoteHost fireID (pare-feu) 24
USER remoteID@fireID @remoteHost (pare-feu) 24
UTILISATEUR sans accès (pare-feu) 24