IPSWITCH
FILE TRANSFER

WS_FTP Professional 12

Sicherheitshandbuch

KAPITEL 1

Sichere Dateiübertragung

In diesem Kapitel

Ein sicheres Übertragungsverfahren auswählen

Welches Verfahren Sie für sichere Dateiübertragungen verwenden, hängt von Ihren Sicherheitsanforderungen ab. Die folgende Tabelle kann Ihnen bei der Auswahl des besten Verfahrens für Ihre Anforderungen helfen:

	Client-Konfiguration?	Server-Konfiguration?	Anmeldung verschlüsselt?	Befehlskanal verschlüsselt?	Dateiübertragung verschlüsselt?	Eigentlich Datei verschlüss
SSL	Ja	Ja	Ja	Ja	Ja	Nein
SSH	Ja	Ja	Ja	Ja	Ja	Nein
OpenPGP	Ja	Nein	Nein	Nein	Nein	Ja

Hinweis Sowohl bei SSL als auch bei SSH kann der Administrator des FTP-Servers Ihnen mitteilen, welcher Servertyp unter der Adresse, an die Sie Dateien senden möchten, eingerichtet ist. Wenn Sie den Server-Typ nicht kennen und versuchen, eine SSL- oder eine SSH-Verbindung zu einem Server herzustellen, der die erforderlichen Protokolle nicht unterstützt, kann die Verbindung nicht aufgebaut werden.

Info über SSL

SSL (Secure Socket Layer) ist ein Protokoll zum Verschlüsseln und Entschlüsseln von Daten, die über direkte Internetverbindungen übertragen werden. Wenn ein Client eine SSL-Verbindung mit einem Server aufbaut, werden die an diesen Server gesendeten und die von diesem Server empfangenen Daten mit einem komplexen mathematischen

Algorithmus verschlüsselt; ggf. abgefangene Daten können nur schwer entschlüsselt werden. Weitere Informationen finden Sie in Secure Sockets Layer (SSL) (siehe Seite 5).

SSH

SSH (Secure Shell) ist ein Sicherheitsprotokoll, mit dem sichere Verbindungen zu Servern aufgebaut werden können, auf denen die Protokolle SSH und SFTP (Secure File Transfer Protocol) eingerichtet wurden.

SSH verschlüsselt die gesamte Kommunikation zwischen Client und Server. Bei SSH-Verbindungen werden sämtliche Funktionen mit SFTP ausgeführt. Weitere Informationen finden Sie in Secure Shell (SSH) (siehe Seite 19).

Info über OpenPGP

OpenPGP ist ein auf Schlüsseln beruhendes Verschlüsselungsverfahren, mit dem Dateien so verschlüsselt werden, dass nur der vorgesehene Empfänger die Daten erhalten und entschlüsseln kann. OpenPGP ist besonders im E-Mail-Verkehr verbreitet, kann aber auch für FTP-Übertragungen genutzt werden.

OpenPGP schützt Dateien mithilfe von zwei Kryptografieschlüsseln: Dateien werden mit einem öffentlichen Schlüssel verschlüsselt. Die verschlüsselten Dateien können nur mit dem jeweils passenden privaten Schlüssel entschlüsselt werden. Weitere Informationen finden Sie unter OpenPGP (siehe Seite 23).

Hinweis Anders als SSL und SSH ist OpenPGP kein Verbindungstyp, sondern ein Verfahren zur Verschlüsselung hochzuladender Dateien. In dieser Funktion kann der OpenPGP-Modus in Verbindung mit Standard-FTP-, -SSL- und -SSH-Verbindungen verwendet werden.

Sichere Übertragungsverschlüsselung mit OpenPGP verwenden

Sie können Ihre Dateien vor, während und nach der Übertragung mit 256-Bit AES, der stärksten Verschlüsselung von WS_FTP, oder anderen Verschlüsselungsmethoden schützen. Der integrierte OpenPGP-Modus zur Verschlüsselung einzelner Dateien in Verbindung mit den sicheren Dateiübertragungsverfahren über SSL/FTPS und SSH/SFTP ist eine sichere Methode zum Übertragen und Speichern persönlicher und vertraulicher Dateien.

FIPS 140-2-validierte Kryptografie verwenden

Info über FIPS (WS_FTP Professional)

FIPS (Federal Information Processing Standard) ist ein vom NIST (U. S. National Institute of Standards and Technology, US-amerikanisches nationales Institut für Standards und Technologie), eines Teilbereichs ohne Regulierungsrechte des U. S. Department of Commerce (US-amerikanisches Handelsministerium) herausgegebener Standard. NIST gibt verschiedene Standards vor, an die sich das US-Militär und verschiedene Regierungsbehörden halten müssen. Aus diesem Grund müssen sich alle Lieferanten und Dienstleister, die mit US-Regierungsbehörden und dem Militär zusammenarbeiten, ebenfalls wo notwendig an diese Standards halten. Zusätzlich zur Tatsache, dass FIPS ein US-amerikanischer Standard ist, gelten bei kanadischen Regierungsbehörden ähnliche Richtlinien, die FIPS-validierte Software vorsehen.

WS_FTP Der FIPS-Modus enthält Triple DES-, AES- und HMAC SHA-1-Verschlüsselung von Dateien während der Übertragung. Es werden keine weiteren Verschlüsselungsmodi unterstützt, wie in FIPS 140-2 angegeben.

Verwenden des FIPS-Modus (WS_FTP Professional)

Wenn der FIPS-Modus aktiviert ist, verschlüsselt WS_FTP sämtliche Dateien, die über SSH, FTP mit SSL oder HTTPS gesendet werden, mit einem FIPS 140-2-validierten Chiffre. Sie können Dateien auch unverschlüsselt oder mit geringerer Verschlüsselungsebene über reguläres FTP senden.

So aktivieren Sie den FIPS-Modus:

1. Wählen Sie **Extras > Optionen**. Das Dialogfeld "Programmeinstellungen" wird geöffnet.
2. Wählen Sie im linken Fenster "FIPS" aus.
3. Wählen Sie die Option **Kryptografisches Modul in FIPS 140-2-Modus betreiben**.

Wenn die Option bereits ausgewählt, aber ausgegraut ist, bedeutet dies, dass Ihr Systemadministrator den FIPS-Modus standardmäßig aktiviert hat. In diesem Fall kann der FIPS-Modus nur vom Administrator deaktiviert werden.

4. Starten Sie die WS_FTP-Anwendung neu (schließen, erneut öffnen).

Alle von der WS_FTP-Anwendung initiierten Übertragungen, die FTPS- (FTP über SSL), SSH- oder HTTPS-Protokolle verwenden, werden mithilfe des FIPS 140-2-validierten kryptografischen Moduls gesendet.

Übertragungen über FTP sind weiterhin zulässig, sie können aber den FIPS-Modus nicht verwenden.

Weitere Informationen finden Sie unter Info über FIPS (siehe Seite 3).

Prüfen der Dateiintegrität

WS_FTP bietet die Möglichkeit, die Integrität der übertragenen Dateien mittels Algorithmen zu überprüfen. Diese stellen sicher, dass die Dateien während der Übertragung zwischen Ausgangs- und Zielort nicht manipuliert wurden. Weitere Informationen finden Sie unter Übertragungsintegrität (siehe Seite 15).

Firewalls konfigurieren

Bei WS_FTP können Sie Informationen über eine bestimmte Firewall in eine Firewall-Konfiguration eintragen, die Sie dann für die Verbindung zu einem FTP-Server hinter jener Firewall auswählen können. Sie können die Firewall in diesem Fall einmal konfigurieren und dann die Konfiguration immer wieder für Server verwenden, für die sie gebraucht werden. Weitere Informationen finden Sie unter Firewalls verwenden (siehe Seite 33).

KAPITEL 2

Secure Sockets Layer (SSL)

In diesem Kapitel

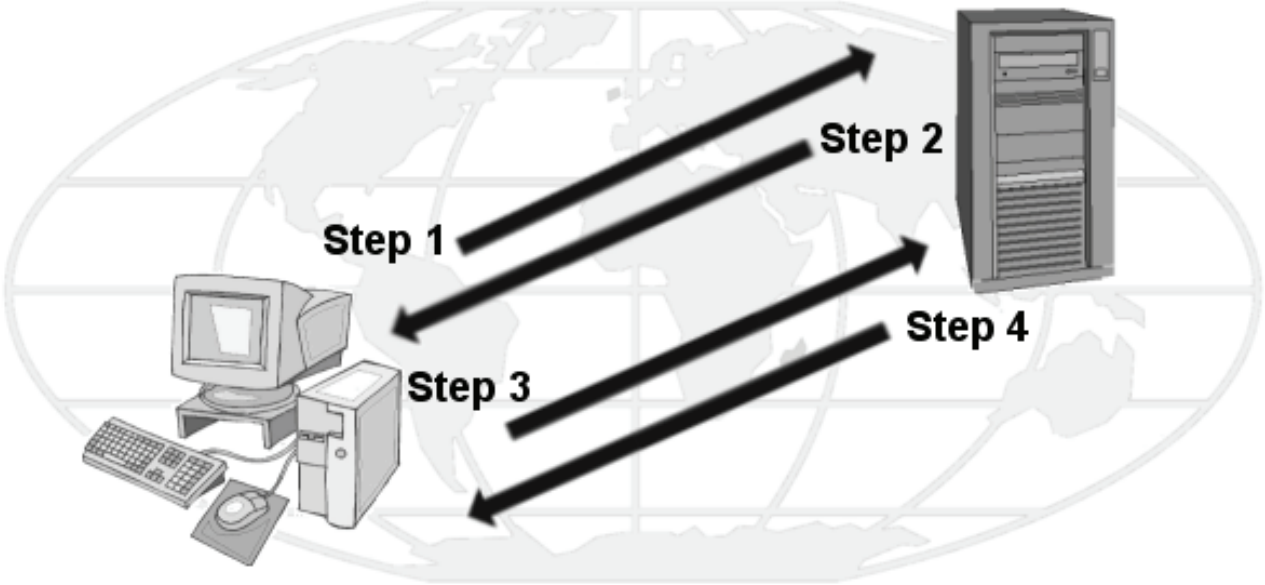
SSL (Secure Socket Layer) kann zusammen mit FTP verwendet werden, um für zusätzliche Sicherheit bei Standard-FTP-Übertragungen zu sorgen. Dieses Kapitel bietet eine Übersicht über das SSL-Protokoll und beschreibt die Funktionsweise von SSL innerhalb von WS_FTP.

Übersicht

SSL (Secure Socket Layer) kann zusammen mit FTP verwendet werden, um für zusätzliche Sicherheit bei Standard-FTP-Übertragungen zu sorgen. Dieses Kapitel bietet eine Übersicht über das SSL-Protokoll und beschreibt die Funktionsweise von SSL innerhalb von WS_FTP.

SSL (Secure Socket Layer) ist ein Protokoll zum Verschlüsseln und Entschlüsseln von Daten, die über direkte Internetverbindungen übertragen werden. Wenn ein Client eine SSL-Verbindung mit einem Server aufbaut, werden die an diesen Server gesendeten und die von diesem Server empfangenen Daten mit einem komplexen mathematischen Algorithmus verschlüsselt; ggf. abgefangene Daten können nur schwer entschlüsselt werden.

Im Folgenden wird Schritt für Schritt erklärt, wie SSL funktioniert.



Schritt 1. Der Client stellt eine initiale Verbindung mit dem Server her und fordert eine SSL-Verbindung an. Wenn implizites SSL verwendet wird, ist die initiale Verbindung verschlüsselt. Mit der Einstellung SSL explizit wird die Anmeldung nicht verschlüsselt.

Schritt 2. Wenn der Server ordnungsgemäß konfiguriert wurde, überträgt der Server sein Zertifikat und seinen öffentlichen Schlüssel an den Client.

Schritt 3: Der Client vergleicht das Zertifikat des Servers mit einer Liste der vertrauenswürdigen Server. Wenn das Zertifikat dort vorkommt, vertraut der Client dem Server und geht zu Schritt 4 über. Ist das Zertifikat dort nicht gespeichert, wird Schritt 4 erst dann ausgeführt, wenn der Benutzer das Zertifikat in die Liste der vertrauenswürdigen Server eingegeben hat.

Schritt 4. Der Client verschlüsselt einen Sitzungsschlüssel mit dem erhaltenen öffentlichen Schlüssel und sendet diesen Sitzungsschlüssel an den Server. Wenn der Server in Schritt 2 das Zertifikat des Client angefordert hat, muss der Client das Zertifikat nun seinerseits senden.

Schritt 5. Wenn der Server so eingerichtet wurde, dass Zertifikate empfangen werden können, vergleicht er das empfangene Zertifikat mit den in der Liste der vertrauenswürdigen Server gespeicherten Zertifikaten, um die Verbindung anschließend zu bestätigen oder abzulehnen.

Wird die Verbindung abgelehnt, überträgt der Server eine entsprechende Fehlermeldung an den Client. Nimmt der Server die Verbindung an oder wurde der Server so konfiguriert, dass er keine Zertifikate empfängt, entschlüsselt er den Sitzungsschlüssel des Client mit seinem eigenen privaten Schlüssel und sendet eine Erfolgsmeldung an den Client, um auf diese Weise einen sicheren Datenkanal zu eröffnen.

Die Funktionsweise von SSL ist am besten anhand der Wirkungsweise der in SSL enthaltenen Elemente zu verstehen. Im Folgenden werden diese Elemente und ihre jeweiligen Aufgaben beschrieben:

- **Client**. In diesem Fall Ipswitch WS_FTP Professional
- **Zertifikat**. Die Zertifikatdatei enthält die Anmeldeinformationen des Clients bzw. des Servers. Mit diesen Informationen weisen die beiden Parteien sich beim Aushandeln der Verbindung aus. Gelegentlich muss das Client-Zertifikat durch das Server-Zertifikat unterzeichnet werden, damit eine SSL-Verbindung hergestellt werden kann. Zertifikatdateien tragen die Endung .crt.
- **Sitzungsschlüssel**. Mit dem Sitzungsschlüssel verschlüsseln Client und Server ihre Daten. Der Sitzungsschlüssel wird vom Client erzeugt.
- **Öffentlicher Schlüssel**. Mit dem öffentlichen Schlüssel verschlüsselt der Client einen Sitzungsschlüssel. Der öffentliche Schlüssel existiert nicht als Datei, sondern entsteht beim Erstellen eines Zertifikats und eines privaten Schlüssels. Mit einem öffentlichen Schlüssel verschlüsselte Daten können nur mit dem privaten Schlüssel entschlüsselt werden, mit dem dieser öffentliche Schlüssel erzeugt wurde.
- **Privater Schlüssel**. Der private Schlüssel entschlüsselt den mit einem öffentlichen Schlüssel verschlüsselten Sitzungsschlüssel des Clients. Die private Schlüsseldatei trägt die Erweiterung .key. Private Schlüssel sollten GRUNDSÄTZLICH NIEMANDEM mitgeteilt werden.
- **Zertifikats-Signieranforderung**. Zertifikats-Signieranforderungen werden beim Erstellen eines Zertifikats generiert. Die betreffenden Dateien werden benötigt, wenn Sie Ihre Zertifikate unterzeichnen müssen. Sobald die Zertifikats-Signieranforderung unterzeichnet wurde, wird ein neues Zertifikat erzeugt und kann dann anstelle des nicht unterzeichneten Zertifikats verwendet werden.

Warum wird SSL verwendet?

SSL verbessert die Sicherheit von Standard-FTP-Übertragungen durch die Verschlüsselung und den Schutz der meisten Elemente einer Verbindung. Sie können Ihre Dateien während der Übertragung mit 256-Bit AES, der stärksten Verschlüsselung von WS_FTP Professional, oder anderen Verschlüsselungsmethoden schützen. Diese Verschlüsselungsmethode gewährleistet bei Übertragungen über SSL/FTPS die Sicherheit und den Schutz vertraulicher Dateien.

HinweisSSL können Sie nur dann nutzen, wenn der FTP-Server SSL unterstützt und für die Annahme von SSL-Verbindungen konfiguriert wurde. Wenn Sie SSL verwenden möchten, Ihr Server SSL aber nicht unterstützt, wenden Sie sich bitte an den Systemverwalter Ihres FTP-Servers.

So stellen Sie eine SSL-Verbindung her:

So stellen Sie eine SSL-Verbindung mit einem für SSL konfigurierten Server her:

1. Erstellen Sie ein Server-Profil und wählen Sie den Server-Typ **FTP/SSL implizit** oder **FTP/SSL (AUTH SSL)**.
2. Wenn Sie auf **Verbinden** klicken, wird der FTP-Server informiert, dass Sie eine SSL-Verbindung herstellen möchten. Der FTP-Server sendet dann ein Zertifikat, das den Server gegenüber dem Client ausweist. Wenn dieses Zertifikat in der Liste der vertrauenswürdigen Server enthalten ist, wird die Verbindung hergestellt.
3. Wenn das Zertifikat nicht enthalten ist, wird eine entsprechende Meldung angezeigt.
4. Wählen Sie die gewünschten Einstellungen, und klicken Sie auf **OK**. Wenn der FTP-Server nicht seinerseits ein Zertifikat erfordert, wird die sichere Verbindung hergestellt. Alle zwischen dem lokalen PC und dem FTP-Server übertragenen Daten werden verschlüsselt.

Wenn der Server, zu dem Sie eine Verbindung herstellen möchten, WS_FTP auffordert, ein Zertifikat zurückzusenden, befolgen Sie die Anweisungen zum Senden eines Client-Zertifikats.

Client-Zertifikat senden

Wenn der Server, zu dem Sie eine Verbindung herstellen möchten, von Ihrem Client die Übertragung eines Zertifikats fordert, verfahren Sie wie folgt:

1. Erstellen Sie ein Server-Profil und wählen Sie den Servertyp **FTP/SSL implizit** oder **FTP/SSL (AUTH SSL)**.
2. Erstellen Sie ein Zertifikat. Beachten Sie dazu bitte auch die Hinweise im Abschnitt Zertifikate generieren (siehe Seite 8).
3. Senden Sie die Datei mit der Zertifikats-Signieranforderung an den Systemverwalter des FTP-Servers.
4. Sobald der Systemverwalter des FTP-Servers die Zertifikats-Signieranforderung unterzeichnet hat, wird die Anforderung wieder an Sie zurückgeschickt.
5. Wenn Sie die Datei erhalten, wählen Sie das neue Zertifikat für das Feld Zertifikat (siehe Seite 10) aus, wie im Abschnitt **Zertifikate auswählen** beschrieben.
6. Stellen Sie die Verbindung zum Server her.

Zertifikate generieren

So erstellen Sie ein SSL-Zertifikat:

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Das Fenster "Programmeinstellungen" wird angezeigt.
2. Wählen Sie **Client-Zertifikate** aus.
3. Klicken Sie auf **Erstellen**. Der Assistent **Client-SSL-Zertifikat erstellen** wird geöffnet.

4 Geben Sie in das Feld **Zertifikat** einen Namen ein. Unter diesem Namen wird das Zertifikat in WS_FTP generiert.

5 Wählen Sie ein Datum aus, an dem das Zertifikat ablaufen soll.

6 Geben Sie eine Kennphrase für Ihr Zertifikat ein und wiederholen Sie die eingegebene Kennphrase zur Bestätigung. Die Kennphrase wird zur Verschlüsselung des privaten Schlüssels verwendet.

Hinweis Diese Kennphrase dürfen Sie keinesfalls vergessen. Die Kennphrase kann aus beliebigen Wörtern, Symbolen, Leerzeichen und Ziffern bestehen.

7 Klicken Sie auf **Weiter**, um fortzufahren.

Nehmen Sie in den übrigen Feldern im Bereich Zertifikatsdaten die erforderlichen Eingaben vor.

Ort. Der Ort, in dem Sie sich befinden. (z. B. Düsseldorf)

Bundesland/Kanton. Landesteil, in der sich dieser Ort befindet. (z. B. Nordrhein-Westfalen)

Organisation. Name der Firma oder des Benutzers.

Kurzbezeichnung. Entweder der Name des Benutzers, der das Zertifikat erstellt, oder der vollständige Domänenname des zum Host gehörenden Servers.

E-Mail. E-Mail-Adresse des Zertifikatinhabers.

Abteilung. Name der organisatorischen Einheit. (z. B. Forschung und Entwicklung)

Land. Land, in dem Sie sich befinden; geben Sie einen gültigen Ländercode mit zwei Buchstaben ein. (z. B. US)

8 Nachdem Sie alle Eingaben ordnungsgemäß vorgenommen haben, klicken Sie auf **Weiter**, um den Vorgang fortzusetzen. Sie können erst dann fortfahren, wenn alle Felder definiert wurden.

9 Überprüfen Sie im letzten Dialogfeld die angezeigten Informationen und klicken Sie auf **Fertigstellen**, um das Zertifikat zu erstellen.

Wenn Sie ein Zertifikat für WS_FTP erstellen, sollten Sie dem Systemverwalter des FTP-Servers die Zertifikats-Signieranforderung per E-Mail zusenden. Der Systemverwalter wird das Zertifikat unterzeichnen und ggf. an Sie zurückschicken. Sobald Sie das Zertifikat erhalten haben, müssen Sie das Zertifikat in Ihre Zertifikat-Datenbank importieren.

Zertifikate importieren

Wenn Sie ein Zertifikat verwenden möchten, das an Sie übermittelt wurde oder das Sie mit Ipswitch WS_FTP Server erstellt haben, müssen Sie das Zertifikat in Ihre Zertifikat-Datenbank importieren.

So importieren Sie ein Zertifikat:

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Das Dialogfeld **Programmeinstellungen** wird angezeigt.
2. Wählen Sie **Client-Zertifikate** aus und klicken Sie dann auf **Importieren**. Der Assistent **Zertifikat importieren** wird geöffnet.
3. Wählen Sie das gewünschte Zertifikat aus und klicken Sie auf **Weiter**.
4. Wählen Sie die Datei mit dem privaten Schlüssel für dieses Zertifikat aus und klicken Sie auf **Weiter**.
5. Geben Sie die zum Erstellen des Zertifikats verwendete Kennphrase ein und klicken Sie auf **Weiter**.
6. Geben Sie den Namen ein, unter dem das Zertifikat in Ihrer Datenbank gespeichert werden soll, und klicken Sie auf **Weiter**.
7. Überprüfen Sie im zuletzt geöffneten Dialogfeld die angezeigten Informationen und klicken Sie auf **Fertigstellen**, um das Zertifikat zur Datenbank hinzuzufügen.

Zertifikate auswählen

Zertifikate werden auf Server-Ebene verwendet; daher müssen Sie für alle von Ihnen erzeugten FTP-Server-Profile ein Zertifikat auswählen. (Sie können allerdings für all Ihre Server das gleiche Zertifikat verwenden).

Zertifikate werden über das Dialogfeld **Einstellungen für FTP-Server** auf dem Dialogfeld **SSL** definiert, indem Sie die gewünschten Zertifikate aus dem Listenfeld **Client-Zertifikat** auswählen. In diesem Listenfeld werden sämtliche Zertifikate angezeigt, die auch über das Dialogfeld **Programmeinstellungen**: Client-Zertifikat“ dargestellt werden können:

Vertrauenswürdige Server

Im Dialogfeld **Vertrauenswürdige Server** wird eine Liste der Zertifikate angezeigt, denen der betreffende Benutzer vertraut. So öffnen Sie die Liste der vertrauenswürdigen Server:

1. Wählen Sie **Extras** > **Optionen** aus. Das Dialogfeld **Programmeinstellungen** wird geöffnet.
2. Klicken Sie auf **SSL**, um die SSL-Optionen anzuzeigen, und klicken Sie anschließend auf **Vertrauenswürdige Server**. Die Liste der vertrauenswürdigen Server wird geöffnet.

Angezeigte Daten

Ausgestellt für. Gibt an, für wen das Zertifikat ausgestellt wurde.

Ausgestellt von. Gibt an, von wem das Zertifikat unterzeichnet wurde.

Ablaufdatum. Gibt an, wann das Zertifikat ungültig wird.

Zertifikate hinzufügen

So fügen Sie ein Zertifikat zur Datenbank hinzu:

1. Klicken Sie auf die Schaltfläche **Importieren** und wählen Sie Pfad und Namen der Zertifikatdatei. Das Dialogfeld **Zertifikat hinzufügen?** wird angezeigt.
2. Überprüfen Sie den Inhalt dieses Dialogfelds, und klicken Sie auf **Ja**, um das Zertifikat in die Datenbank aufzunehmen.

Zertifikate exportieren

So exportieren Sie ein Zertifikat aus der Liste der vertrauenswürdigen Server:

1. Wählen Sie das zu kopierende Zertifikat aus Ihrer Liste aus.
2. Klicken Sie auf **Exportieren**.
3. Wählen Sie den Ordner aus, in den das Zertifikat kopiert werden soll, und geben Sie den Namen ein, unter dem die Zertifikatdatei gespeichert werden soll.
4. Klicken Sie auf **OK**.

Zertifikate entfernen

So entfernen Sie ein Zertifikat:

1. Wählen Sie das zu entfernende Zertifikat aus.
2. Klicken Sie auf **Entfernen**.
3. Vor Ausführung des Befehls wird die Empfehlung angezeigt, das Zertifikat zu exportieren. (Mit dem Entfernen des Zertifikats löschen Sie die Zertifikatdatei.)
4. Klicken Sie auf **OK**, um den Löschbefehl ausführen zu lassen.

Nicht als vertrauenswürdig gekennzeichnetes Zertifikat

Wenn Sie eine Verbindung zu einem Server als SSL-Verbindung herstellen, sendet der Server Ihnen ein Zertifikat. Wird dieses Zertifikat in der Liste der vertrauenswürdigen Server nicht angezeigt, oder wurde das Zertifikat nicht mit einem in der Liste enthaltenen Zertifikate unterzeichnet, wird dieses Dialogfeld angezeigt:

Zertifikatsdaten

Ausgestellt für. Name der Person oder der Firma, der das Zertifikat gehört.

Ausgestellt von. Name der Person oder der Firma, von der das Zertifikat signiert wurde.

Aktiv ab. Datum, an dem das Zertifikat aktiviert wurde.

Ablaufdatum. Datum, an dem das angezeigte Zertifikat ungültig wird.

Zertifikatoptionen

Nur diese Verbindung zulassen. Wenn diese Option aktiviert ist, wird die Verbindung zwar hergestellt, das Zertifikat wird von WS_FTP aber nicht als vertrauenswürdig eingestuft. Und wenn Sie das nächste Mal versuchen, eine Verbindung zu diesem Server herzustellen, wird das Dialogfeld erneut angezeigt.

Diesem Zertifikat vertrauen. Wenn diese Option ausgewählt wurde, wird die Verbindung hergestellt und das Zertifikat zum Registerblatt Vertrauenswürdige Server hinzugefügt. Alle weiteren Verbindungen mit dem Server werden automatisch hergestellt, ohne dass Sie diese erneut bestätigen müssten.

Verbindung nicht zulassen. Wenn diese Option aktiviert ist, wird die Verbindung abgebrochen.

NAT-Firewall verwenden

Wenn Sie eine Firewall mit NAT (Network Address Translation) verwenden, können in Verbindung mit SSL-Verschlüsselungen Probleme auftreten. Um diese Probleme zu vermeiden, sollten Sie WS_FTP und die Firewall so konfigurieren, dass in Ihrem Rechner eingehende Verbindungen zugelassen werden. WS_FTP muss dann den Server anweisen, eine Verbindung zu der externen IP-Adresse herzustellen, die dann von der Firewall an Ihren PC weitergeleitet werden sollte. Außerdem sollten Sie die Anzahl der Ports beschränken, die die Firewall für diese Verbindungen öffnet. In den meisten Fällen wird dieses Vorgehen die Verwendung von SSL über eine NAT-Firewall ermöglichen.

HinweisWenn Sie Windows XP verwenden, können Sie Ihre Firewall automatisch so konfigurieren, dass die benötigten Ports geöffnet sind und Sie die externe IP-Adresse mit UPnP einholen können. UPnP kann im Dialogfeld **Programmeinstellungen**: Firewall“ aktiviert werden.

So konfigurieren Sie SSL über eine NAT-Firewall:

1 Wählen Sie **Extras > Optionen** aus. Das Fenster "Programmeinstellungen" wird angezeigt.

2 Wählen Sie im linken Fenster **Firewall** aus.
3 Wählen Sie **PORT-IP-Adresse erzwingen** und geben Sie dann die IP-Adresse der NAT-Firewall ein.
4 Wählen Sie **Lokalen Port-Bereich begrenzen** und geben Sie dann die **Untergrenze** und die **Obergrenze** für den Port-Bereich ein.
5 Klicken Sie auf **OK**.

KAPITEL 3

Dateiübertragungsintegrität

In diesem Kapitel

Übersicht

WS_FTP bietet die Möglichkeit, die Integrität der übertragenen Dateien mittels Algorithmen zu überprüfen. Diese stellen sicher, dass die Dateien während der Übertragung zwischen Ausgangs- und Zielort nicht manipuliert wurden.

Dieses Kapitel bietet einen Überblick über die Funktion "Dateiübertragungsintegrität" und beschreibt deren Verwendung in WS_FTP.

Wenn durch den FTP-Server unterstützt, kann WS_FTP Überprüfungen der Dateiintegrität aller übertragener Dateien durchführen. Als Teil des letzten Schritts bei der Übertragung führen sowohl der FTP-Client als auch der FTP-Server einen kryptografischen Hashtest der übertragenen Datei durch. Wenn die Werte übereinstimmen, "wissen" beide Seiten, dass die übertragene Datei vollständig identisch mit dem Original ist. Die Ergebnisse der Dateiüberprüfungen werden im WS_FTP-Protokoll angezeigt.

Diese Funktion erkennt Date Datenveränderungen, sodass Hijacking-Angriffe (bei denen der Angreifer Dateien während der Übertragung liest und Codeelemente einfügt und somit die Datei ändert) bei Dateiübertragungen erkannt werden können.

Wie also funktioniert die Überprüfung der Dateiübertragungsintegrität?

Die Überprüfung der Dateiübertragungsintegrität wird erstens pro Server aktiviert, indem Sie die Option **Auf Übertragungsfehler prüfen** im Dialogfeld "Einstellungen für FTP-Server > Übertragung" aktivieren.

Nachdem sie einmal aktiviert wurde, wird sie über das Dialogfeld "Programmeinstellungen> Dateiintegrität" gesteuert. WS_FTP verwendet dann alle ausgewählten Dateiintegritätsalgorithmen in der angezeigten Reihenfolge, um die Datei nach der Übertragung zu überprüfen. WS_FTP überprüft (mithilfe des FEAT-Befehls) für jeden Algorithmus die Antwort vom Server, um zu sehen, ob dieser Algorithmus vom Server unterstützt wird. Wenn der Server einen Fehler zurückgibt (normalerweise "500 Befehl nicht unterstützt"), versucht WS_FTP den nächsten Algorithmus und merkt sich die Algorithmen, die der Server nicht unterstützt. Die Unterstützung eines bestimmten

Dateiintegritätsalgorithmus kann überprüft werden, indem Sie nach einer Dateiübertragung im Verbindungsprotokoll nach Fehlern suchen.

WS_FTP verwendet die folgenden Befehle an den Server, um die Dateiintegrität zu überprüfen: XCRC, XMD5, XSHA1, XSHA256 und XSHA512. Obwohl keiner dieser Befehle die Bedeutung als Internetstandard erlangt hat, unterstützen die meisten Server, die eine Überprüfung der Dateiintegrität unterstützen, mindestens einen dieser Befehle.

Wenn vom Server keiner der Dateiintegritätsalgorithmen unterstützt wird und die Option **Größenvergleich bei Überprüfung der Übertragungsintegrität verwenden** ausgewählt ist, überprüft WS_FTP als letzten Ausweg die Größe der zuletzt übertragenen Datei, indem der Befehl "SIZE" an den externen Server gesendet wird. Wenn der Server den Befehl "SIZE" unterstützt (das tun nicht alle Server), vergleicht WS_FTP den vom Server zurückgegebenen Wert mit der Größe der lokalen Datei. Obwohl dies keine Garantie für die Dateiintegrität ist, stellt es eine Minimalüberprüfung dessen dar, dass die richtige Anzahl an Bytes übertragen wurde.

Die Algorithmusstärke hat ebenfalls Einfluss auf die Dauer der Überprüfung der Dateiintegrität. Je stärker der Algorithmus, desto länger dauert die Übertragungsüberpüfung.

Überprüfen der Dateiübertragungsintegrität einrichten

In den **Einstellungen für FTP-Server** können Sie die Überprüfung der Dateiübertragungsintegrität für einzelne Server aktivieren, in den **Ordner-Einstellungen** (Standard-Server-Optionen) aktivieren Sie die Überprüfung für alle neu eingerichteten Server. Nach der Aktivierung der Dateiübertragungsintegrität legen Sie die Einstellungen und Prioritäten des Dateiintegritätsalgorithmus fest.

Die Dateiintegritätsalgorithmen können nur für binäre Übertragungen von und zu Servern verwendet werden, die die optionalen FTP-Befehle unterstützen. (Dies sind zum Beispiel SSH- und HTTP-Server.) Der Administrator des FTP-Servers kann Ihnen mitteilen, ob die Algorithmen von dem FTP-Server, an den Sie Dateien senden möchten, unterstützt werden.

Option Dateiübertragungsintegrität aktivieren

So aktivieren Sie die Überprüfung der Übertragungsintegrität für ein Server-Profil:

1. Klicken Sie in der Symbolleiste auf die Schaltfläche **Verbinden**, um den Server-Manager zu öffnen.
2. Wählen Sie den Server, für den Sie die Überprüfung der Übertragungsintegrität aktivieren möchten, und klicken Sie dann auf **Bearbeiten**. Nun wird das Dialogfeld **Einstellungen für FTP-Server** angezeigt.

3 Das Dialogfeld **Einstellungen für FTP-Server** wird angezeigt.
4 Klicken Sie auf die Registerkarte **Übertragung**.
5 Aktivieren Sie das Kontrollkästchen **Auf Übertragungsfehler prüfen**.
6 Klicken Sie auf **OK**, um das Dialogfeld **Einstellungen für FTP-Server** zu schließen.

So aktivieren Sie die Überprüfung der Übertragungsintegrität in den Standard-Server-Optionen:

1 Klicken Sie in der Symbolleiste auf die Schaltfläche **Verbinden**, um den Server-Manager zu öffnen.
2 Wählen Sie **Server** aus und klicken Sie auf **Bearbeiten**. Das Dialogfeld **Ordner-Einstellungen** wird angezeigt.
3 Klicken Sie auf die Registerkarte **Übertragung**.
4 Aktivieren Sie das Kontrollkästchen **Auf Übertragungsfehler prüfen**.
5 Klicken Sie auf **OK**, um das Dialogfeld **Ordner-Einstellungen** zu schließen.

Nach der Aktivierung der Option **Auf Übertragungsfehler prüfen** in den Ordner-Einstellungen überprüfen neue Server die Übertragungsintegrität automatisch. Bestehende Server behalten ihre vorherigen Einstellungen bei; wenn Sie die Überprüfung der Übertragungsintegrität für bestehende Server aktivieren möchten, müssen Sie ihre Profile individuell ändern.

Algorithmus für die Überprüfung der Dateiübertragungsintegrität konfigurieren

Die Einstellung des Dateiübertragungsintegritäts-Algorithmus umfasst sowohl die Auswahl als auch die Priorität des Algorithmus. Nicht alle FTP-Server unterstützen die optionalen Integritätsalgorithmen und FTP-Befehle zur Berechnung der Prüfsumme oder des Hash-Werts übertragener Dateien - gerade hier zeigen sich erhebliche Unterschiede zwischen den Servern. Der Administrator des FTP-Servers kann Ihnen mitteilen, ob die Algorithmen von dem FTP-Server, an den Sie Dateien senden möchten, unterstützt werden.

So legen Sie den Dateiintegritätsalgorithmus für Dateiübertragungen fest:

1 Klicken Sie in der Symbolleiste auf **Optionen** (oder wählen Sie **Extras** > **Optionen**), um das Dialogfeld **Programmeinstellungen** zu öffnen.
2 Klicken Sie im linken Fenster auf **Übertragungen** > **Dateiintegrität**. Das Dialogfeld **Dateiintegrität** wird geöffnet.
3 Wählen Sie die gewünschten Optionen zur Überprüfung der Dateiintegrität aus: CRC32, SHA1, MD5, SHA256 und SHA512.
4 Mit den Pfeilen nach oben und nach unten stellen Sie die Algorithmuspriorität für Dateiübertragungen ein.
5 Klicken Sie auf **Übertragungsintegrität durch Größenvergleich feststellen**, wenn der FTP-Client zur Überprüfung der Übertragungsintegrität die Dateigrößen vergleichen soll. Dies bietet sich an, wenn der FTP-Server die ausgewählten Dateiintegritätsalgorithmen nicht unterstützt.

KAPITEL 4

Secure Shell (SSH)

In diesem Kapitel

Das Secure Shell-Protokoll (SSH), auch als SFTP (Secure File Transfer Protocol) bezeichnet, kann bei FTP-Übertragungen verwendet werden, um die Sicherheit von Standard-FTP-Verbindungen zu erhöhen. In diesem Kapitel wird beschrieben, wie das SSH Protokoll in Ipswitch WS_FTP Professional verwendet wird.

Übersicht

SSH ist ein Sicherheitsprotokoll, mit dem sichere Verbindungen zu Servern aufgebaut werden können, auf denen die Protokolle SSH und SFTP (Secure File Transfer Protocol) eingerichtet wurden.

SSH verschlüsselt die gesamte Kommunikation zwischen Client und Server. Bei SSH-Verbindungen werden sämtliche Funktionen mit SFTP ausgeführt. Im Interesse größtmöglicher Sicherheit führt Ipswitch WS_FTP Professional SSH-Übertragungen mit SSH2 und dem SFTP-Protokoll aus.

FTP-Server überwachen gewöhnlich Port 21 auf Verbindungen. Bei SSH-Servern wird Port 22 überwacht.

Hinweis SSH kann in Verbindung mit verschiedenen Authentifizierungsverfahren eingesetzt werden. Allerdings unterstützt WS_FTP nur einfache Kennwort-Authentifizierung, Authentifizierung mit öffentlichem Schlüssel und interaktive Authentifizierung über die Tastatur.

WS_FTP Professional unterstützt nur SFTP/SSH2.

Warum wird SSH verwendet?

SSL verbessert die Sicherheit von Standard-FTP-Übertragungen durch die Verschlüsselung des gesamten Datenverkehrs einschließlich der Verbindungsdaten und der Kennwörter. Dadurch werden Lauschangriffe, Verbindungs-Hijacking und andere Angriffe verhindert.

Sie können Ihre Dateien während der Übertragung mit 256-Bit AES, der stärksten Verschlüsselung von WS_FTP, oder anderen Verschlüsselungsmethoden schützen. Diese Verschlüsselungsmethode gewährleistet bei Übertragungen über SSH/SFTP die Sicherheit und den Schutz vertraulicher Dateien.

SSH können Sie nur dann nutzen, wenn der FTP-Server SSH unterstützt und für die Annahme von SSH-Verbindungen konfiguriert wurde. Wenn Sie SSH verwenden möchten, Ihr Server SSH aber nicht unterstützt, wenden Sie sich bitte an den Systemverwalter Ihres FTP-Servers.

Öffentliche SSH-Schlüssel exportieren

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Das Dialogfeld **Programmeinstellungen** wird angezeigt.
2. Wählen Sie **SSH > Client-Schlüssel**.
3. Klicken Sie auf **Exportieren**. Das Dialogfeld **Speichern unter ...** wird angezeigt.
4. Geben Sie einen Dateinamen ein und klicken Sie auf **Speichern**.

Herstellen einer SSH-Verbindung (WS_FTP Professional)

Das Herstellen einer SSH-Verbindung erfordert kaum zusätzliche Konfigurationsarbeiten an neuen oder vorhandenen Server-Profilen.

Beim Erzeugen eines neuen Server-Profils mit dem Verbindungs-Assistenten definieren Sie nach der entsprechenden Aufforderung im Assistenten für die Option "Server-Typ" einfach die Einstellung **SFTP/SSH**.

Wenn Sie ein vorhandenes Server-Profil bearbeiten:

1. Klicken Sie in der Symbolleiste auf die Schaltfläche **Verbinden**, um den Server-Manager zu öffnen.
2. Wählen Sie den gewünschten Server aus der Liste der konfigurierten Server aus und klicken Sie auf **Bearbeiten**. Das Dialogfeld "Einstellungen für FTP-Server" wird angezeigt.
3. Klicken Sie auf **Weitere Optionen.**

4 Wählen Sie den **Server-Typ** "SFTP/SSH" aus. Klicken Sie auf **OK**.
5 Wählen Sie das gewünschte Authentifizierungsverfahren aus:

- **Kennwort:** Wenn auf Ihrem Server eine Kennwort-Authentifizierung erfolgt, ist die Konfiguration damit abgeschlossen. Wenn Sie sich das nächste Mal auf diesem Server anmelden, wird Ihre Verbindung automatisch mit SSH geschützt.
- **Öffentlicher Schlüssel:** Erfolgt auf Ihrem Server eine Authentifizierung mit öffentlichen Schlüsseln, wählen Sie **Weitere Optionen > SSH**. Wählen Sie unter **SSH-Schlüsselpaar** das benötigte Schlüsselpaar aus. Wenn keine Schlüsselpaare angezeigt werden, können Sie ein Schlüsselpaar erstellen (siehe Seite 21).
- **Interaktiv über Tastatur:** Wenn Ihr Server für die Verwendung der interaktiven Authentifizierung über Tastatur konfiguriert ist, werden Sie zur Eingabe aufgefordert. Ipswitch WS_FTP Professional zeigt automatisch ein Dialogfeld an, in dem der Benutzer gefragt wird, ein Kennwort einzugeben, eine Sicherheitsfrage zu beantworten usw., jeweils abhängig von den Sicherheitseinstellungen des Servers.

Wenn Sie "Interaktive Authentifizierung über Tastatur" auswählen, dann ist eine *manuelle Benutzereingabe* erforderlich. Das bedeutet, dass automatisierte Dateisynchronisierungsvorgänge einen *Benutzereingabe in Echtzeit* erfordern (im Gegensatz zu anderen Methoden, bei denen automatisch auf das gespeicherte Kennwort oder die Schlüsseldatei zugegriffen wird).

6 Klicken Sie auf **OK, um das Dialogfeld "Einstellungen für FTP-Server" zu schließen.**
7 Klicken Sie auf **Verbinden, um die Verbindung herzustellen.**

Wenn Sie nun eine Verbindung mit diesem Profil herstellen, versucht der Client automatisch eine SFTP/SSH-Verbindung über Port 22 dieses Servers aufzubauen.

Über **Extras > Optionen > SSH** legen Sie fest, welche Chiffren/MAC (Message Authentication Code) für die SSH-Verbindung verwendet werden.

SSH-Schlüsselpaare generieren

Sie können ein SSH-Schlüsselpaar zur Authentifizierung einer SSH-Verbindung zu einem Server generieren. Die von Ihnen erstellten SSH-Schlüssel können Sie erst dann verwenden, wenn Sie dem für den betreffenden Server zuständigen Administrator Ihren öffentlichen Schlüssel zur Installation auf diesem Server übermittelt haben.

So generieren Sie ein SSH-Schlüsselpaar:

1 Wählen Sie im Hauptfenster **Extras > Optionen**. Das Fenster "Programmeinstellungen" wird angezeigt.
2 Wählen Sie **SSH > Client-Schlüssel**.
3 Klicken Sie auf **Erstellen**. Der SSH-Client-Schlüsselpaargenerierungs-Assistent wird gestartet.

4 Befolgen Sie die Anweisungen des Assistenten.

Importieren eines SSH-Schlüsselpaars (WS_FTP Professional)

Sie können einen öffentlichen SSH-Schlüssel importieren, der mit der Funktion Ihres SSH-Servers zur Authentifizierung mit öffentlichen Schlüsseln verwendet werden soll.

So importieren Sie einen öffentlichen SSH-Schlüssel:

1 Wählen Sie im Hauptfenster **Extras > Optionen**. Das Fenster "Programmeinstellungen" wird angezeigt.
2 Wählen Sie **SSH > Client-Schlüssel**.
3 Klicken Sie auf **Importieren**. Folgen Sie den Anweisungen auf dem Bildschirm, um den Importvorgang abzuschließen.

KAPITEL 5

OpenPGP

In diesem Kapitel

Übersicht

OpenPGP kann mit FTP verwendet werden und bietet eine bessere Sicherheit als Standard-FTP. WS_FTP Professional bietet den ersten voll integrierten OpenPGP-Modus zur Verschlüsselung einzelner Dateien. Dies ermöglicht auch vor und nach der Übertragung eine sichere Verwaltung der Dateien. OpenPGP-Dateien können nur von Benutzern entschlüsselt werden, die über den passenden Schlüssel verfügen. Der OpenPGP-Modus unterstützt auch AES, 3DES und andere Chiffren, er bietet Signatur- bzw. Schlüsselstärken von 1.024 bis 4.096 Bit und er unterstützt RSA- und Diffie-Hellman-Schlüsseltypen mit zeitlich begrenzbaren Schlüsseln.

Zusätzlich können Sie Ihre Dateien vor, während und nach der Übertragung mit 256-Bit AES, der stärksten Verschlüsselung von Ipswitch WS_FTP Professional, oder anderen Verschlüsselungsmethoden schützen. Der integrierte OpenPGP-Modus zur Verschlüsselung einzelner Dateien in Verbindung mit den sicheren Dateiübertragungsverfahren über SSL/FTPS und SSH/SFTP ist eine sichere Methode zum Übertragen und Speichern persönlicher und vertraulicher Dateien.

In diesem Kapitel wird erläutert, wie OpenPGP in WS_FTP funktioniert. Dazu werden die einzelnen Schritte beim Übertragen von mit OpenPGP verschlüsselten Dateien beschrieben. Anhand von Beispielen wird dargestellt, wie Sicherheitsprobleme mit OpenPGP gelöst werden können.

Die in WS_FTP enthaltene Software beruht teilweise auf Standards, die mit dem von der OpenPGP Working Group der IETF (Internet Engineering Task Force) vorgeschlagenen Standard RFC 2440 definiert wurden. Ipswitch WS_FTP Professional kann mit OpenPGP-, PGP- oder GPGP-Schlüsseln betrieben werden.

Funktionsweise

OpenPGP ist ein auf Schlüsseln beruhendes Verschlüsselungsverfahren, mit dem Dateien so verschlüsselt werden, dass nur der vorgesehene Empfänger die Daten erhalten und entschlüsseln kann. OpenPGP ist besonders im E-Mail-Verkehr verbreitet, kann aber auch für FTP-Übertragungen genutzt werden.

OpenPGP schützt Dateien mithilfe von zwei Kryptografieschlüsseln: Dateien werden mit einem öffentlichen Schlüssel verschlüsselt. Die verschlüsselten Dateien können nur mit dem jeweils passenden privaten Schlüssel entschlüsselt werden.

Anders als SSL und SSH ist OpenPGP kein Verbindungstyp, sondern ein Verfahren zur Verschlüsselung hochzuladender Dateien. In dieser Funktion kann der OpenPGP-Modus in Verbindung mit Standard-FTP-, -SSL- und -SSH-Verbindungen verwendet werden.

Im Folgenden wird Schritt für Schritt erklärt, wie OpenPGP in Verbindung mit FTP-Übertragungen funktioniert.

Schritt 1: Die hochzuladende Datei wird mit einem öffentlichen Schlüssel verschlüsselt, den der vorgesehene Empfänger der Datei zuvor übermittelt hat.

Schritt 2. Die verschlüsselte Datei wird auf den FTP-Server hochgeladen.

Schritt 3. Der vorgesehene Empfänger lädt die Datei vom FTP-Server.

Schritt 4: Mit dem privaten Schlüssel (der zusammen mit dem ursprünglich zum Verschlüsseln der Datei verwendeten öffentlichen Schlüssel ein Schlüsselpaar bildet) entschlüsselt der vorgesehene Empfänger die Datei, damit er den Inhalt der Datei anzeigen kann.

Sie können OpenPGP auch dazu verwenden, lokal gespeicherte Dateien zu ver- und entschlüsseln.

Verwenden des OpenPGP-Modus zum Verschlüsseln und Entschlüsseln von Dateien zur Übertragung (WS_FTP Professional)

Mit dem OpenPGP-Modus können Sie zu übertragende Dateien mit OpenPGP-Schlüsseln verschlüsseln. Dateien, die Sie mit OpenPGP verschlüsseln, können ausschließlich von Personen entschlüsselt werden, die den passenden Schlüssel besitzen. Wenn für einen externen Server der OpenPGP-Modus aktiviert ist, erfolgt die Verschlüsselung wie folgt:

- Wenn Sie eine Datei auf den Server hochladen, wird die Datei, sofern sie es nicht bereits ist, verschlüsselt und dann übertragen. Dies geschieht immer, egal ob es sich um einen lokalen oder externen Server handelt.
- Wenn Sie eine verschlüsselte Datei (eine Datei mit der Erweiterung .pgp) von einem externen Server herunterladen und über den entsprechenden privaten Schlüssel verfügen, wird die Datei übertragen und anschließend entschlüsselt. Bei der entschlüsselten Datei fällt dann die Erweiterung ".pgp" weg. Wenn die verschlüsselte Datei zum Beispiel "Tabelle.xls.pgp" hieß, hat die entschlüsselte Datei den Namen "Tabelle.xls".
- Wenn Sie eine verschlüsselte Datei von einem lokalen Server herunterladen, also eine Übertragung von lokal zu lokal, wird die Datei so übertragen, wie sie ist, und nicht automatisch entschlüsselt.

Aktivieren des OpenPGP-Modus

Den OpenPGP-Modus können Sie für alle Verbindungen mit einem FTP-Server aktivieren.

1. Stellen Sie eine Verbindung mit einem externen Server her.
2. Klicken Sie auf die Registerkarte des FTP-Servers.
3. Wählen Sie **Extras > OpenPGP-Modus** aus, oder klicken Sie in der Symbolleiste auf **OpenPGP-Modus**. Das Dialogfeld "OpenPGP-Modus" wird geöffnet.
4. Wählen Sie aus den angezeigten Optionen das gewünschte OpenPGP-Übertragungsverfahren aus:
   - Verschlüsseln Sie die Dateien mit einem Schlüssel aus Ihrem Keyring. Wählen Sie die zum Verschlüsseln der Dateien zu verwendenden Verschlüsselungsschlüssel aus.
   - Unterzeichnen Sie die Dateien mit Ihrem privaten Schlüssel als digitaler Signatur. Wählen Sie den zum Unterzeichnen der Dateien zu verwendenden Signierschlüssel aus. Geben Sie die **Kennphrase** des Signierschlüssels ein.
   - Wenn Sie beide Funktionen gleichzeitig verwenden möchten, wählen Sie die Einstellung **Verschlüsseln und unterzeichnen** aus.
5. Klicken Sie auf **OK**. Damit ist der OpenPGP-Modus aktiviert, bis die betreffende Verbindung beendet wird bzw. bis Sie den OpenPGP-Modus ausdrücklich wieder deaktivieren.

6 Laden Sie Dateien hoch oder herunter. Beim Hochladen werden Sie aufgefordert, das zu verwendende OpenPGP-Schlüsselpaar auszuwählen.

Ihre Schlüsselpaare, also jene, die Sie entweder generiert oder importiert haben, werden unter **Programmeinstellungen > OpenPGP > Schlüsselpaare** angezeigt.

Die hochgeladenen und mit OpenPGP-verschlüsselten Dateien haben folgende Dateierweiterung: .pgp. Wenn Sie zum Beispiel eine Datei namens "Kapitel4.pdf" hochladen, wird die Datei in dem Ordner auf dem externen Server als "Kapitel4.pdf.pgp" gespeichert.

Wenn Sie eine Datei mit der Erweiterung .pgp herunterladen und über den entsprechenden privaten Schlüssel verfügen, wird die Datei entschlüsselt und überprüft. Der Dateiname wird wieder mit seiner ursprünglichen Erweiterung angezeigt.

Deaktivieren des OpenPGP-Modus

1 Während eine Verbindung mit einem externen Server besteht, wählen Sie dessen Registerkarte.
2 Wählen Sie **Extras > OpenPGP-Modus** aus, oder klicken Sie in der Symbolleiste auf **OpenPGP-Modus**.
3 Damit ist der OpenPGP-Modus deaktiviert.

Aktivieren des OpenPGP-Modus als Voreinstellung für einen Server(WS_FTP Professional)

Sie können einen Server so konfigurieren, dass der OpenPGP-Modus automatisch aktiviert wird, sobald eine Verbindung zu diesem Server hergestellt wurde.

Der OpenPGP-Modus kann beim Hochladen von Dateien mit der Synchronisierungs-Utility und mit der Skript-Utility nicht verwendet werden.

Ihr Keyring muss mindestens einen OpenPGP-Schlüssel enthalten, damit Sie einen Server so konfigurieren können, dass der OpenPGP-Modus automatisch als Voreinstellung angenommen wird, wenn Sie eine Verbindung zu diesem Server herstellen. Weitere Informationen finden Sie in den Abschnitten Generieren eines OpenPGP-Schlüsselpaars (siehe Seite 28) und Importieren von OpenPGP-Schlüsseln (siehe Seite 29).

Diese Serveroption kann unabhängig von den Programmeinstellungen für den OpenPGP-Modus festgelegt werden.

So aktivieren Sie OpenPGP-Modus als Voreinstellung für einen Server:

1 Klicken Sie in der Symbolleiste auf die Schaltfläche **Verbinden**, um den Server-Manager zu öffnen.

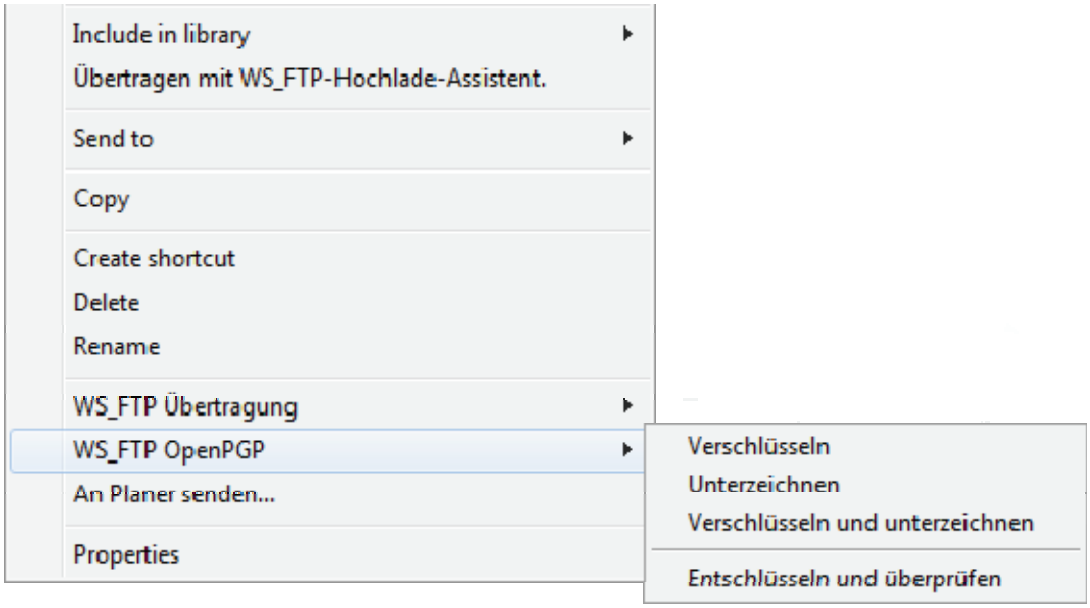
2. Wählen Sie aus der Liste den Server aus, für den der OpenPGP-Modus grundsätzlich als Voreinstellung für den Verbindungsaufbau aktiviert werden soll. Klicken Sie auf **Bearbeiten**. Das Dialogfeld "Einstellungen für FTP-Server" wird angezeigt.
3. Klicken Sie auf **Weitere Optionen**, und wählen Sie die Option **OpenPGP.** Die OpenPGP-Optionen werden angezeigt.
4. Definieren Sie die Einstellung **Nach Verbinden OpenPGP-Übertragung.**
5. Wählen Sie aus den angezeigten Optionen das gewünschte OpenPGP-Übertragungsverfahren aus:
6. Verschlüsseln Sie die Dateien mit einem Schlüssel aus Ihrem Keyring. Wählen Sie die zum Verschlüsseln der Dateien zu verwendenden **Verschlüsselungsschlüssel** aus.
7. Unterzeichnen Sie die Dateien mit Ihrem privaten Schlüssel als digitaler Signatur. Wählen Sie den zum Unterzeichnen der Dateien zu verwendenden **Signierschlüssel** aus. Geben Sie die **Kennphrase** des Signierschlüssels ein**.**
8. Wenn Sie beide Funktionen gleichzeitig verwenden möchten, wählen Sie die Einstellung **Verschlüsseln und unterzeichnen** aus.
9. Klicken Sie auf **OK**, um Ihre Eingaben zu speichern und das Dialogfeld zu schließen.

Verschlüsseln und Entschlüsseln lokaler Dateien (WS_FTP Professional)

Sie können auf die Verschlüsselungsbefehle für lokale Dateien und lokale Übertragungen auf zweierlei Art zugreifen:

- OpenPGP-Menü in der Symbolleiste des lokalen Fensters.

- Das Kontextmenü zu einer lokalen Datei oder einem lokalen Ordner bietet dieselben Menüoptionen wie oben angegeben:



Beide Menüs enthalten dieselben OpenPGP-Befehle:

- **Verschlüsseln:** Diese Option wählen Sie, wenn Sie die Datei vor der Übertragung mit einem Schlüssel verschlüsseln möchten.
- **Unterzeichnen:** Mit dieser Option können Sie Ihre digitale Signatur zur Datei hinzufügen. Indem Sie Ihre digitale Signatur hinzufügen, erhöhen Sie gegenüber anderen Benutzern, die diese Datei herunterladen und die Sie kennen und Ihnen vertrauen, die Vertrauenswürdigkeit Ihrer Datei.
- **Verschlüsseln und unterzeichnen:** Diese Option wählen Sie, wenn Sie die Datei zunächst verschlüsseln und dann Ihre digitale Signatur hinzufügen möchten.
- **Entschlüsseln und überprüfen** - Wählen Sie diese Option aus, um eine mit OpenPGP verschlüsselte Datei zu entschlüsseln (identifiziert durch die Erweiterung .pgp) und die digitale Signatur zu überprüfen, falls eine vorhanden ist.

Beim Verschlüsseln und signieren von Dateien werden Sie aufgefordert, ein zu verwendendes Schlüsselpaar auszuwählen. Ihre Schlüsselpaare, nämlich jene, die Sie entweder generiert oder importiert haben, werden unter **Programmeinstellungen > OpenPGP > Schlüsselpaare** angezeigt.

Generieren eines OpenPGP-Schlüsselpaars (WS_FTP Professional)

Wenn Sie nicht bereits ein persönliches OpenPGP-Schlüsselpaar besitzen, können Sie mit WS_FTP ein Schlüsselpaar generieren.

So generieren Sie ein OpenPGP-Schlüsselpaar:

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Das Fenster "Programmeinstellungen" wird angezeigt.
2. Wählen Sie **OpenPGP > Schlüssel**.
3. Klicken Sie auf **Erstellen**. Der OpenPGP-Schlüsselgenerierungs-Assistent wird gestartet.
4. Befolgen Sie die Anweisungen des Assistenten, um den Schlüssel zu erstellen.

Verwandte Themen

Importieren von OpenPGP-Schlüsseln (siehe Seite 29)

Exportieren von OpenPGP-Schlüsseln (siehe Seite 29)

OpenPGP-Schlüssel importieren

Um Dateien mit dem OpenPGP-Modus zu verschlüsseln, müssen Sie deren OpenPGP-Schlüssel importieren. Sie können Schlüssel und Keyrings auch importieren, die Sie in einem anderen OpenPGP-Programm verwenden.

WS_FTP kann nur auf Schlüssel und auf Keyrings zugreifen, die als eigenständige Dateien gespeichert wurden. Wenn die zu importierenden Schlüssel oder Keyrings in eine E-Mail oder in ein OpenPGP-Programm eingebettet sind, müssen Sie die Schlüssel bzw. Keyrings zunächst exportieren, um die Schlüssel bzw. Keyrings dann wieder in WS_FTP zu importieren.

So importieren Sie einen OpenPGP-Schlüssel oder einen Keyring:

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Das Fenster "Programmeinstellungen" wird angezeigt.
2. Wählen Sie **OpenPGP > Schlüssel**.
3. Klicken Sie auf **Importieren**. Der OpenPGP-Schlüsselimport-Assistent wird gestartet.
4. Befolgen Sie die Anweisungen des Assistenten, um den Schlüssel zu importieren.

OpenPGP-Schlüssel exportieren

Sie können OpenPGP-Schlüssel zur Verwendung mit anderen Programmen exportieren.

So exportieren Sie einen OpenPGP-Schlüssel:

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Das Fenster "Programmeinstellungen" wird angezeigt.
2. Wählen Sie **OpenPGP > Schlüssel**.

3 Wählen Sie den zu exportierenden Schlüssel aus, und klicken Sie auf **Exportieren**. Nun wird der OpenPGP-Schlüsselexport-Assistent geöffnet.
4 Befolgen Sie die Anweisungen des Assistenten zum Exportieren der Schlüssel.

Szenario: Verschlüsseln von Dateien für die Übertragung an oder von einem externen Server (WS_FTP Professional)

Sie können die OpenPGP-Verschlüsselung verwenden, um eine Übertragung von einem lokalen Server auf einen externen Server zu sichern. In einem typischen Szenario ist eine Übertragung zu sichern, die ein Kollege Ihnen von einem entfernten Server senden möchte. Gehen Sie in diesem Fall wie folgt vor:

1 Sie senden Ihrem Kollegen per E-Mail einen öffentlichen Schlüssel. Verwenden Sie WS_FTP, um einen Schlüssel zu generieren (siehe Seite 28) oder zu exportieren (siehe Seite 29).
2 Der Kollege kopiert den öffentlichen Schlüssel (aus der E-Mail), verschlüsselt die Datei mithilfe Ihres öffentlichen Schlüssels und lädt sie folgendermaßen hoch:
    1 Importiert den Schlüssel (siehe Seite 29) in WS_FTP.
    2 Stellt eine Verbindung zum FTP-Server des Unternehmens her.
    3 Aktiviert den OpenPGP-Modus (siehe Seite 25) mit Ihrem für die Verschlüsselung verwendeten Schlüssel.
    4 Lädt die Datei hoch.
3 Sie laden die Datei herunter und entschlüsseln sie mithilfe Ihres privaten Schlüssels.

    Zum Entschlüsseln einer mit OpenPGP verschlüsselten Datei (identifiziert durch die Erweiterung .pgp) klicken Sie mit der rechten Maustaste auf die Datei in WS_FTP und wählen **OpenPGP > Entschlüsseln und überprüfen** aus.

Verwandte Themen

Verwenden des OpenPGP-Modus zum Verschlüsseln und Entschlüsseln von Übertragungen an oder von einem externen Server (siehe Seite 25)

Verschlüsseln und Entschlüsseln von Dateien für lokale Speicherung oder lokale Übertragungen (siehe Seite 27)

Verschlüsseln und Entschlüsseln lokaler Dateien (siehe Seite 27)

Szenario: Verschlüsseln von lokalen Dateien und lokalen Übertragungen (WS_FTP Professional)

Sie können die OpenPGP-Verschlüsselung verwenden, um Dateien zu sichern, die auf Ihrer lokalen Festplatte gespeichert sind, oder um eine Übertragung von einem lokalen Ordner in einen anderen lokalen Ordner zu sichern. In diesem Fall erfolgen die Übertragungen auf demselben Rechner. Typische Szenarien sind:

- Sie haben Dateien auf Ihrem Laptop, die sie als Vorsichtsmaßnahme vor Ort sichern möchten. Sie können die Dateien mit Ihrem öffentlichen Schlüssel verschlüsseln und sie bei Bedarf mit Ihrem privaten Schlüssel entschlüsseln.
- Sie archivieren Dateien regelmäßig auf einer externen Festplatte und möchten Sie beim Übertragen auf die Festplatte verschlüsseln.

Verschlüsseln von Dateien für die Speicherung:

1. Verwenden Sie WS_FTP, um einen Schlüssel zu generieren (siehe Seite 28) oder zu importieren (siehe Seite 29).

   Wenn Sie bereits einen OpenPGP-Schlüssel haben, können Sie ihn für die Verschlüsselung auswählen. Sie können die verfügbaren Schlüssel im Dialogfeld **Programmeinstellungen > OpenPGP->Schlüssel** anzeigen.

   Wenn Sie der Einzige sind, der die Datei verwendet, müssen Sie den öffentlichen Schlüssel an niemand sonst senden.
2. Wenn Sie eine Datei auswählen, klicken Sie mit der rechten Maustaste und wählen Sie **WS_FTP> OpenPGP > Verschlüsseln** aus.

   Die verschlüsselte Datei wird mit einer .pgp-Erweiterung gespeichert.

   Informationen über alle OpenPGP-Befehle finden Sie unter Verschlüsseln und Entschlüsseln lokaler Dateien (siehe Seite 27).
3. Wenn Sie die Datei verwenden möchten, wählen Sie sie aus, klicken Sie mit der rechten Maustaste und wählen Sie dann **OpenPGP > Entschlüsseln und überprüfen** aus.

   Ihr privater Schlüssel wird verwendet, um die Datei zu entschlüsseln. Die Datei muss mit dem zugewiesenen öffentlichen Schlüssel verschlüsselt worden sein.

Verschlüsseln von Dateien für die lokale Übertragung:

1. Stellen Sie eine Verbindung mit dem lokalen Server her und wählen Sie den Zielordner aus.
2. Aktivieren Sie den OpenPGP-Modus (siehe Seite 25) mit Ihrem für die Verschlüsselung verwendeten Schlüssel.
3. Laden Sie die Datei hoch. Die Datei wird im Zielordner gespeichert; sie trägt dann eine .png-Erweiterung.
4. Wenn Sie die Datei verwenden möchten, laden Sie sie herunter, klicken Sie mit der rechten Maustaste und wählen Sie dann **OpenPGP > Entschlüsseln und überprüfen** aus.

Verwandte Themen

Verschlüsseln und Entschlüsseln lokaler Dateien (siehe Seite 27)

Szenario: Verschlüsseln von Dateien für die Übertragung an oder von einem externen Server (siehe Seite 30)

KAPITEL 6

Verwenden von Firewalls

In diesem Kapitel

Mehrere Firewalls

Es gibt mehrere gute Gründe, mehr als eine Firewall-Konfiguration zu erstellen. Wenn Sie beispielsweise mit einem Notebook-Rechner an verschiedenen Standorten arbeiten, die sich hinter unterschiedlichen Firewalls befinden, dann müssten Sie eine Firewall-Konfiguration für jeden Standort einrichten, sodass Sie für jeden Standort auf die dort benötigte Firewall-Konfiguration zugreifen können.

Möglicherweise müssen auch mehrere Firewall-Konfigurationen eingerichtet werden, weil in einem Netzwerk mehrere Routerverbindungen als Firewall vorkommen. In diesem Fall wären abhängig vom Teil des Netzes, in dem Sie gerade arbeiten, ebenfalls verschiedene Firewall-Konfigurationen erforderlich.

Außerdem könnte es beispielsweise eine Reihe vertrauenswürdiger Server geben (z. B. FTP-Server, die Ihr Unternehmen selbst unterhält), für die eine andere (oder gar keine) Firewall verwendet wird.

Firewall-Typen

In der folgenden Tabelle sind alle bekannten Firewall-Typen sowie die Informationen zusammengestellt, die Sie in WS_FTP zu diesen Typen jeweils eingeben müssen.

Typ einer Firewall	**Angaben in WS_FTP**
SERVER-Host-Name	Host-Name (oder Adresse), Benutzername (ID)
BENUTZER nach Anmeldung	Host-Name (oder Adresse), Benutzername (ID), Kennwort

Typ einer Firewall	Angaben in WS_FTP
BENUTZER ohne Anmeldung	Host-Name (oder IP-Adresse)
GEÖFFNETER Proxy	Host-Name (oder IP-Adresse)
USER Externer_Name @Externer_Host Firewall	Host-Name (oder Adresse), Benutzername (ID), Kennwort
USER Firewall@Externer_Host	Host-Name (oder Adresse), Benutzername (ID), Kennwort
Transparent	Benutzername (ID), Kennwort
USER remoteID@fireID @remoteHost	Host-Name (oder Adresse), Benutzername (ID), Kennwort
SOCKS4 und SOCKS5	Host-Name (oder Adresse), Benutzername (ID), Kennwort
HTTP	Host-Name (oder Adresse), Benutzername (ID), Kennwort

Firewall konfigurieren

Um eine Firewall konfigurieren zu können, benötigen Sie vom zuständigen Systemverwalter Daten zur betreffenden Firewall. Weitere Informationen finden Sie im vorstehenden Abschnitt **Firewall-Typen**.

Bei manchen Router-Firewalls werden Sie wahrscheinlich den passiven Modus verwenden. In diesem Modus werden Datenverbindungen nicht über den FTP-Server, sondern über den FTP-Client (in unserem Fall also WS_FTP) aufgebaut.

So konfigurieren Sie eine Firewall:

1. Wählen Sie **Extras > Optionen** aus.
2. Öffnen Sie das Dialogfeld **Firewall**.
3. Klicken Sie auf **Neu**.
4. Geben Sie im Dialogfeld **Firewall-Eigenschaften** folgende Informationen ein. Weitere Informationen zu den Firewall-Eigenschaften finden Sie in der Online-Hilfe.
   - **Name** - Ein Name, der diese Firewall-Konfiguration beschreibt. Verwenden Sie einen beliebigen Namen, der den Einsatz der Firewall-Konfiguration bezeichnet.
   - **Host-Name** - Der Host-Name oder die IP-Adresse des Firewall-Rechners.
   - **Benutzername** - Falls erforderlich, tragen Sie den Benutzernamen für die Firewall ein.

- **Kennwort** - Tragen Sie Ihr Kennwort für die Firewall ein. Aktivieren Sie die Einstellung **Kennwort speichern** nur, wenn Sie der einzige Benutzer Ihres Computers sind.
- **Typ** - Wählen Sie den Firewall-Typ aus der Liste.
- **Port** - Geben Sie die Port-Nummer an.

5 Wenn Sie auf **Fertig stellen** klicken, wird die Firewall der Liste **Konfigurierte Firewalls** hinzugefügt.

Die gespeicherte Firewall-Konfiguration können Sie auch dem Server zuweisen, wie im folgenden Abschnitt Konfigurierte Firewalls verwenden (siehe Seite 35) beschrieben.

Verwenden einer konfigurierten Firewall

Nach der Konfiguration einer Firewall können Sie die Konfiguration auf einen FTP-Server anwenden.

1 Klicken Sie in der Symbolleiste auf **Verbinden**, um den Server-Manager zu öffnen.
2 Aktivieren Sie ein Server-Profil. Klicken Sie auf **Bearbeiten**.
3 Klicken Sie auf **Weitere Optionen**.
4 Wählen Sie in der Liste **Firewall** die gewünschte Firewall-Konfiguration.

Verwenden von UPnP

Wenn Sie Windows XP verwenden, können Sie Ihre Firewall automatisch so konfigurieren, dass die benötigten Ports geöffnet sind und Sie die externe IP-Adresse mit UPnP einholen können.

So aktivieren Sie UPnP:

1 Klicken Sie in der Symbolleiste auf **Optionen**, oder wählen Sie **Extras > Optionen** im Menü. Das Fenster "Programmeinstellungen" wird angezeigt.
2 Wählen Sie **Firewall** aus.
3 Wählen Sie **Für PORT-Befehle UPnP benutzen**.

APPENDIX A

FireScript-Editor

Was ist ein FireScript?

Dieser Anhang beschreibt Zweck und Syntax der Befehlssprache FireScript und erläutert, wie Sie mit FireScript-Befehlen eine FTP-Verbindung über eine Firewall herstellen können.

Mit FireScripts können Sie die beim Anmelden in FTP-Servern verwendeten Befehls- und Antwortketten individuell konfigurieren. Diese individuelle Konfiguration kann erforderlich sein, wenn auf Ihrem FTP-Server vor oder nach dem Anmelden besondere Befehle eingegeben werden müssen oder wenn sich zwischen Client und Server ein besonderer Firewall-Typ befindet.

FireScripts werden in der Programmiersprache FireScript geschrieben, die speziell für die Verwendung in Ipswitch WS_FTP Professional entwickelt wurde. FireScripts können die gleichen Funktionen ausführen, die WS_FTP auch im internen Programmcode beim Verbinden mit einem Host oder einer Firewall verwendet. Mit FireScripts können Sie jedoch festlegen, ob und wann die betreffenden Funktionen tatsächlich ausgeführt werden. Insbesondere entscheiden FireScripts darüber, wann der Host-Typ automatisch erkannt und wann eine sichere SSL-Verbindung hergestellt werden soll. Mit dem Skript können Sie definieren, ob der die Anweisung xauth verwendet werden soll und ob Anmeldungen mit Benutzernamen und Kennwort über ein Benutzerkonto erfolgen sollen

Der Aufbau von FireScripts

FireScripts bestehen aus drei Abschnitten: **fwsc**, **comment** und **script**. Wie bei einer Initialisierungsdatei in Windows steht der Name der Abschnitte in eckigen Klammern jeweils in einer eigenen Zeile gefolgt vom eigentlichen Text der Abschnitte.

Der Abschnitt **fwsc** enthält verschiedene Namen-/Wertepaare (ebenfalls wie in Windows-Initialisierungsdateien). Diese Paare enthalten Informationen zum Skript und geben an, welche Variablen vom Skript benötigt werden.

Der Abschnitt **comment** besteht aus Kommentaren in natürlicher Sprache, die nicht an eine bestimmte Form gebunden sind. Sie werden beim Ausführen des Skripts ignoriert.

Der Abschnitt **script** schließlich enthält die von der Steuerung auszuführenden Skripts und ist entsprechend der FireScript-Syntax zu formulieren.

Im Folgenden sehen Sie ein typisches FireScript mit den beschriebenen Abschnitten:

[fwsc]

… Weitere Werte (hier nicht abgedruckt) könnten z. B. 'required=' und 'version=' lauten.

[comment]

Dies ist ein Beispielskript zum Herstellen einer Verbindung mit einem FTP-Proxy. Das Skript ist nicht vollständig. Viele erforderliche Befehle wurden aus Gründen der Übersichtlichkeit weggelassen. Mit diesem Skript soll hauptsächlich der dreigliedrige Aufbau von FireScript illustriert werden.

```
[script]
send ("OPEN %HostAddress") {}
tryssl;
send (“USER %HostUserId”)
{
case (300..399) :
```

… Aus Platzgründen kann der größte Teil des Skripts nicht abgedruckt werden.

Der Abschnitt fwsc

Im Abschnitt **fwsc** können Sie Informationen zu Ihrem Skript ähnlich wie in der Win.ini-Datei definieren. Die meisten Parameter dienen nur zur Information. Dazu gehören die Felder **author** und **version**. Aus bestimmten Parametern entnimmt die Steuerung, ob der Anmeldedialog angezeigt und welche IP-Adresse verwendet werden soll.

Der Parser erkennt und speichert die Werte der folgenden Parameter:

fwsc-Parameter	
Parameter	Bedeutung und Werte
author	Nur zur Information. Autor des FireScript.
version	Nur zur Information. Versionsnummer der Skriptdatei.
verdate	Nur zur Information. Aktualisierungsdatum der Version.
required	Eine Feldliste mit Kommata als Trennzeichen, die zur Ausführung der FireScripts benötigt wird. Wenn nicht alle erforderlichen Felder definiert wurden, wird der Anmeldedialog angezeigt, und die Schaltfläche **Verbinden** ist erst dann verfügbar, wenn alle benötigten Angaben vorgenommen wurden.
preask	Eine Feldliste mit Kommata als Trennzeichen; die Felder sind nicht unbedingt erforderlich; wenn die Felder aber nicht definiert wurden, erscheint der Anmeldedialog.
connectto	'Firewall' oder 'host'. Aus diesem Parameter entnimmt Ipswitch WS_FTP Professional, zu welcher IP-Adresse eine Verbindung hergestellt werden

	soll.

Nicht erkannte Parameter werden übergangen.

Der Abschnitt comment

Im Abschnitt **comment** können Sie die vom FireScript auszuführenden Funktionen beschreiben. FireScripts sollten möglichst gut beschrieben werden, damit die Skripts auch später noch nachzuvollziehen und ggf. leichter zu aktualisieren sind. Beim Ausführen des FireScripts wird der Abschnitt **comment** ignoriert.

Sie können auch in den Abschnitt script Kommentare einfügen. Diese Kommentare müssen Sie allerdings ähnlich wie in C++ und in Java mit dem Trennzeichen **//** versehen. (Der Parser übergeht sämtliche Eingaben in Zeilen, denen das Zeichen **//** vorangestellt ist.)

Der Abschnitt script

Der Abschnitt **script** besteht aus einer Folge von Anweisungen, mit denen Befehle an die Firewall bzw. an den FTP-Server übertragen werden. Manche Anweisungen führen zu bestimmten Ergebnissen oder lösen Reaktionen der Firewall oder des FTP-Servers aus. Mit einer einfachen Befehlsstruktur können abhängig vom Ergebnis der Reaktionen unterschiedliche Abläufe eines Skripts veranlasst werden.

Der Verbindungsaufbau

Verbindungsanforderungen an FTP-Server werden durch Eingaben in der Benutzeroberfläche (klassisch oder Explorer) oder durch Funktionen der in Ipswitch WS_FTP Professional verfügbaren Utilities (z.B. Suchen oder Synchronisieren) veranlasst. Auch der FTP-Manager fordert gelegentlich Verbindungen an. Alle Verbindungen werden mit der Funktion CreateConnection der Ipswitch WS_FTP Professional API hergestellt.

Der Verbindungsaufbau besteht aus zwei Phasen:

- Phase 1: Aufbau der Verbindung mit der Firewall oder mit dem FTP-Server
- Phase 2: Übertragung von Befehlen zum Anmelden und zum Autorisieren des verbundenen Benutzers. In dieser Phase werden die Befehle eines FireScript ausgeführt.

Die erste Phase läuft unabhängig davon, ob Ipswitch WS_FTP Professional ein FireScript ausführt oder eine der definierten Firewall-Konfigurationen verwendet, immer gleich ab. Vor der Ausführung eines Skripts prüft Ipswitch WS_FTP Professional, ob im Abschnitt **fwsc** Felder mit den Attributen **required** und **preask** gekennzeichnet wurden. Fehlen Angaben in entsprechend gekennzeichneten Feldern, zeigt WS_FTP Pro den Anmeldedialog an. Nachdem der Benutzer alle erforderlichen Informationen

eingegeben und auf die Schaltfläche **Verbinden** geklickt hat, prüft Ipswitch WS_FTP Professional den Eintrag im Feld **connectto**. Abhängig von dem für dieses Feld definierten Wert wird eine Verbindung zu IP-Adresse und Port der betreffenden Firewall bzw. des betreffenden FTP-Servers aufgebaut. Ist das Feld nicht vorhanden, nimmt Ipswitch WS_FTP Professional per Voreinstellung die IP-Adresse der Firewall an (wenn definiert).

Nach erfolgreichem Verbindungsaufbau und nach dem Öffnen eines gültigen Socket veranlasst Ipswitch WS_FTP Professional die Ausführung des FireScript durch die FireScript-Steuerung. Nach erfolgreicher Ausführung und entsprechender Bestätigung der Anmeldung mit Hilfe des FireScript wird die Kontrolle über die Verbindung von der Funktion CreateConnection an den Benutzer übergeben.

Die FireScript-Sprache

Die FireScript-Sprache besteht aus einer bestimmten Anzahl an Elementen, die Ihnen vielleicht bekannt sind, wenn Sie bereits Skripte oder Programme in sonstigen Sprachen geschrieben haben. Die Sprache beruht auf Variablen, Erklärungen und Anweisungen zur Ausführung von Aktionen und zur Steuerung von Programmabläufen. In den folgenden Abschnitten werden diese Elemente beschrieben.

FireScript-Anweisungen enden grundsätzlich mit einem Semikolon. Daher können sich FireScript-Anweisungen auch über mehrere Zeilen erstrecken, und einzelne Zeilen können auch mehrere Anweisungen enthalten. Innerhalb von Zeichenfolgen dürfen allerdings keine Zeilenwechsel vorkommen. Die Abführungszeichen müssen in der gleichen Quellcodezeile stehen wie die Anführungszeichen. Der folgende Code z. B. wäre gültig definiert:

Dieser Code dagegen wäre nicht gültig:

FireScript-Variablen

Für die FireScripts werden die in Ipswitch WS_FTP Professional definierten Anmeldedaten verwendet. Dies sind zumindest Benutzernamen und Kennwörter sowie IP-Adresse und Port des FTP-Servers. Gelegentlich werden auch die IP-Adresse und der Port der Firewall verwendet. Diese Felder werden häufig einem Server-Profil, einer FTP-Adresse oder der Befehlszeile entnommen. Wenn erforderliche Angaben nicht vollständig definiert wurden, wird wie bereits beschrieben beim Verbindungsaufbau das Anmeldedialogfeld angezeigt, damit die Benutzer die fehlenden Eingaben manuell vornehmen können. Vor der Ausführung der entsprechenden Befehle speichert die Skript-Steuerung diese Informationen in einer Reihe interner Variablen. Außerdem wird das Ergebnis des zuletzt ausgeführten Befehls in internen Variablen gespeichert. Die Variablen werden nach Ausführung der betreffenden Anweisungen von der Skript-Steuerung definiert.

Die Syntax der Variablen hängt von den Anweisungen oder Ausdrücken ab, in denen die Variablen vorkommen. In der folgenden Liste sind sämtliche internen Variablen zusammengestellt:

Interne FireScript-Variablen	
Variable	Bedeutung und Verwendung
FwUserId	Benutzername des betreffenden Benutzers in der Firewall. Bei manchen Firewalls müssen sich die Benutzer anmelden, damit weitere Verbindungen über die Firewall zugelassen werden.
FwPassword	Kennwort des betreffenden Benutzers in der Firewall. Das Kennwort ist erforderlich, wenn sich die Benutzer in der Firewall anmelden müssen.
FwAccount	Konto in der Firewall. Diese Eingabe ist erforderlich, wenn die Benutzer ein Konto in der Firewall definieren müssen. Der Vollständigkeit halber aufgenommen, kommt in der Praxis aber so gut wie nie vor.)
FwAddress	Die IP-Adresse der Firewall. Diese Eingabe ist erforderlich, wenn die Benutzer eine Verbindung mit der Firewall herstellen und veranlassen müssen, dass die Firewall ihrerseits als Proxy eine Verbindung zum FTP-Server herstellt.
HostUserId	Benutzername des Benutzers für den FTP-Server. Fast immer erforderlich. Wenn für einen Benutzer auf dem Server kein Name definiert wurde, ist die Eingabe 'anonymous' vorzunehmen.
HostPassword	Kennwort des Benutzers für den FTP-Server. Fast immer in Verbindung mit dem Benutzernamen erforderlich. Wurde der Benutzername **anonymous** verwendet, geben Sie Ihre E-Mail-Adresse als Kennwort ein.
HostAccount	Konto des Benutzers auf dem FTP-Server. Bei FTP-Servern mit bestimmten Betriebsystemen muss nach erfolgreicher Anmeldung außer dem Benutzernamen und Kennwort noch ein Konto angegeben werden.
HostAddress	IP-Adresse des Host. Die Skript-Steuerung stellt entweder eine Direktverbindung mit dieser Adresse her oder übermittelt die Adresse an eine Firewall, die als Proxy-Server fungiert.
LastFtpCode	Dreistelliger numerischer Code der letzten Antwort, die vom FTP-Server oder der Firewall erhalten wurde (nach einer erfolgreichen Anmeldung z. B. 230).
LastReply	Text der letzten Antwort des Servers (z. B. "230 user logged in."

Da benutzerdefinierte Variablen in FireScripts nicht benötigt und nicht verwendet werden, kommen auch Variablendeklarationen nicht vor. Außerdem kann das Skript

nicht direkt Werte für interne Variablen festsetzen, d. h. es brauchen auch keine Zuweisungen definiert zu werden.

Zeichenfolgenerweiterung

Bei manchen FireScript-Befehlen und -Funktionen werden Zeichenfolgen als Argumente verwendet. Diese Zeichenfolgen können eine Variable oder eine Zeichenfolgenkonstante (in Anführungszeichen) sein (z. B. **Dies ist eine Zeichenfolge**). In solche Zeichenfolgen können Sie Zitate einbetten, indem Sie den Anführungszeichen für das Zitat jeweils ein Prozentzeichen (%) voranstellen. Das Prozentzeichen (%) fungiert also als Codeumschalter zum Einbetten von Variablen und Anführungszeichen in Zeichenfolgen.

Aus der Zeichenfolge %% wird %.
Die Zeichenfolge %" wird ersetzt durch ".
%, gefolgt vom Namen einer Variablen, wird durch den Wert der Variablen ersetzt.

Eine Skript-Anweisung könnte z. B. so aussehen:

Wenn beim Aufrufen des Skripts HostAddress gleich **ftp.ipswitch.com** ist, wird der Befehl folgendermaßen umgesetzt:

Der Ausdruck

wird beim Ausführen des Skripts umgesetzt in:

und die Anweisung

wird beim Ausführen des Skripts so umgesetzt:

Die Übergabe einer Zeichenfolgenvariablen entspricht der Übergabe einer Zeichenfolgenkonstante, die die Variable umsetzt, ist aber schneller.

Beispiel:

bedeutet dasselbe wie, ist aber schneller als

Funktionsausdrücke

Die FireScript-Sprache unterstützt zurzeit keine komplexeren Ausdrücke. Sie beinhaltet aber zwei Funktionsausdrücke mit einigen Booleschen Operatoren zur Statusbestimmung von Variablen. Diese sind **contains** und **isempty**. Die Booleschen Operatoren sind **not** und **and**.

Die Funktion **contains** vergleicht zwei Zeichenfolgen und gibt den Wert **true** aus, wenn die zweite Zeichenfolge in der ersten Zeichenfolge vorkommt. Bei der Suche wird zwischen Groß- und Kleinschreibung unterschieden. Beide Zeichenfolgen werden zunächst erweitert.

Die Funktion **isempty** durchsucht eine Zeichenfolge und gibt den Wert **true** aus, wenn die Zeichenfolge tatsächlich keine Zeichen enthält. Mit dieser Funktion können Sie feststellen, ob für eine interne Variable ein Wert definiert wurde.

Der Boolesche Operator **not** kehrt den vom Funktionsausdruck ausgegebenen Wert um.

Beispiel:

Nehmen wir an, die Variable HostAccount enthält den Wert 'usr987i'
isempty (HostAccount) ergibt dann den Wert **false**, aber
not isempty(HostAccount) ergibt den Wert **true**.

Der Boolesche Operator **and** geht davon aus, dass alle angegebenen Bedingungen **true** sind.

Beispiel: Nehmen wir an, die Variable HostAccount enthält den Wert 'usr987I'
und die letzte Antwort des Servers lautet "230 User logged in, please send account".

Für den folgenden Ausdruck ergibt sich dann der Wert **true**:

FireScript-Anweisungen

Die FireScript-Sprache umfasst verschiedene Arten von Anweisungen. Anweisungen bewirken, dass bestimmte Aktionen ausgeführt werden oder ein bestimmter Skript-Ablauf gesteuert wird. Die unterschiedlichen FireScript-Anweisungen werden in den folgenden Abschnitten beschrieben.

Switch-Anweisungen

Die Anweisungen **send** und **xauth** werden als Switch-Anweisungen bezeichnet, weil sie abhängig vom Verhalten des Servers sofort eine bestimmte Entscheidung treffen. Die Switch-Anweisung enthält **case**-Anweisungen, die den Case-Anweisungen in Java und C++ sehr ähnlich sind. Allerdings stellen die Bedingungen keine Konstanten dar, die mit einem einzelnen Ausdruck verglichen werden.

Unmittelbar nach einer Switch-Anweisung wie z. B. **send** oder **xauth** stehen grundsätzlich Case-Anweisungen in geschweiften Klammern { <Case-Anweisungen> }. Die Case-Anweisungen können leer sein (d.h. zwischen den geschweiften Klammern steht keine Eingabe); die Klammern müssen aber in jedem Fall gesetzt werden.

Beispiel einer Switch-Anweisung:

Hinter der Anweisung **send** steht als einziges Argument die an den Server zu sendende Zeichenfolge. Die Zeichenfolge wird vor dem Senden in ihre Langform umgesetzt. Die maximale Länge der ausgeschriebenen Zeichenfolge ist auf etwa 512 Byte - die maximale Länge einer FTP-Zeile - begrenzt. Der Befehl wird also gesendet und die erhaltene Antwort mit den Bedingungen der einzelnen Case-Anweisungen verglichen.

Für die Anweisung **xauth** werden keine Argumente definiert. Die Anweisung bewirkt, dass geprüft wird, ob das Begrüßungs-Banner ein unsichtbar vom Ipswitch WS_FTP-Server bereitgestelltes XAUTH enthält. Wenn keine Verbindung zu einem Ipswitch WS_FTP-Server besteht oder wenn die Einladung nicht gefunden wird, hat die Anweisung **xauth** keinerlei Auswirkungen, und die Case-Anweisungen werden nicht überprüft. Wird die Einladung nicht gefunden, so werden der Benutzername und das Kennwort codiert, und die Anweisung xauth wird an den Server gesendet. Wie bei der Anweisung **send** wird anschließend die Antwort des Servers abgewartet und auf die Case-Anweisungen überprüft.

Case-Anweisungen

Case-Anweisungen sind in Switch-Anweisungen eingebettet. Eine Case-Anweisung enthält eine Liste von Bedingungen, die in der Server-Antwort erfüllt sein müssen, damit der betreffende Fall aktiviert wird.

Diese Liste von Bedingungen endet mit einem Doppelpunkt ':'.

Case-Anweisungen werden in der Reihenfolge ihres Auftretens verarbeitet, bis der erste passende Eintrag erreicht ist.

Sobald für die Bedingungen in der Case-Anweisung eine Entsprechung gefunden wird, werden die eingebetteten Anweisungen ausgeführt.

Mögliche Inhalte für Case-Anweisungen sind eine Liste von FTP-Codes bzw. FTP-Code-Bereichen, ein Funktionsausdruck und die Sonderfälle **any** und **timeout**.

FTP-Codes bzw. FTP-Code-Bereiche müssen immer vor den jeweiligen Funktionsausdrücken stehen. Die Listeneinträge stehen in Klammern und sind durch Kommata getrennt. Jeder Listeneintrag muss entweder ein dreistelliger Code oder ein durch zwei dreistellige Codes bestimmter und durch einen doppelten Punkt (..) eingegrenzter Bereich sein. Mindest- und Höchstwert des Bereichs sind jeweils im Bereich enthalten, und nach Möglichkeit sollte zunächst der Mindestwert definiert werden.

Die Sonderfälle **any** und **timeout** müssen separat verwendet werden.

Beispiele für Case-Anweisung

Die folgenden Case-Anweisungen treffen zu, wenn der Server den FTP-Code 226 oder 231 meldet:

Die folgenden Case-Anweisungen sind zutreffend, wenn der Server den FTP-Code 226 oder 231 meldet oder wenn der gemeldete Code zwischen 250 und 299 liegt, wobei die Grenzwerte jeweils zum betreffenden Bereich zählen. Entsprechend gilt die Anweisung für den Wert 250 sowie für die Werte 251, 252 usw. bis 299

Die folgenden Case-Anweisungen treffen zu, wenn der Server einen FTP-Code von 300 bis 399 meldet und die gemeldete Zeichenfolge den Text "email address" enthält:

Die folgenden Case-Anweisungen treffen zu, wenn der Server einen FTP-Code von größer oder gleich 500 meldet und die gemeldete Zeichenfolge die angegebene Fehlermeldung enthält:

Wenn ein Fall mehrere Bedingungen enthält, müssen diese durch **and** getrennt sein. Der Operator **and** bestimmt, dass alle angegebenen Bedingungen erfüllt sein müssen. Auf das letzte Beispiel bezogen heißt dies, dass der FTP-Code zwischen 500 und 599 liegen muss UND dass die letzte Antwort die angegebene Zeichenfolge enthalten muss. Beide Bedingungen müssen also erfüllt sein. Ist eine Bedingung nicht erfüllt, wird die Case-Anweisung nicht ausgeführt.

Der Operator **not** kehrt das Ergebnis einer Funktion um. Nehmen wir z.B. an, wir möchten sicherstellen, dass eine bestimmte Zeichenfolge in der letzten Antwort nicht vorkommt. Zum Beispiel:

Der Operator **or** kommt nicht vor. Nach demselben Muster können auch mehrere Case-Anweisungen verwendet werden.

Die folgende Case-Anweisung trifft zu, wenn das Zeitlimit für die Anweisung send überschritten wurde:

Der Fall **any** trifft immer zu und sollte deshalb ggf. am Ende der Liste stehen. Weitere Anweisungen im Anschluss an diesen Fall werden nicht berücksichtigt.

Die folgende Case-Anweisung beispielsweise trifft immer zu:

Bei sich überschneidenden Case-Anweisungen - d.h. wenn innerhalb einer Liste mehrere Anweisungen auf die Server-Antwort zutreffen - wird nur die erste ausgeführt.

Beispiel:

Wenn die Anweisung mit der Funktion contains hinter einem Fall ohne diese Anweisung stünde, würde diese Anweisung nicht berücksichtigt.

Continue-Anweisungen

Anders als bei C und C++ wird nach Ausführung einer Case-Anweisung nicht automatisch die nächste Case-Anweisung ausgeführt. Es werden nur die im aktivierten Fall explizit aufgelisteten Anweisungen ausgeführt. Danach wird die nächste Anweisung nach der umgebenden Switch-Anweisung ausgeführt. Die Anweisung **continue** bewirkt die Ausführung der an die begrenzende Switch-Anweisung anschließenden Anweisung. Sie hat somit die gleiche Funktion wie die Anweisung Break in Switch-Anweisungen in C/C++; diese Anweisung ist allerdings nicht unbedingt erforderlich.

Switch-Anweisung können nicht verschachtelt werden. Die Anweisungen **send** und **xauth** dürfen also nicht innerhalb einer **Case**-Anweisung stehen.

Anweisungen jump und label

Eine Sprunganweisung (jump) bewirkt, dass die Skript-Steuerung zu einem anderen Teil des Skripts wechselt. Die betreffende Stelle muss im Skript als Sprungziel (label) definiert sein. In den Beispiel-FireScripts von Ipswitch sind in Case-Anweisungen Sprünge zu verschiedenen Codesequenzen definiert, d. h. der auszuführende Code richtet sich danach, welcher Fall aktiviert wurde.

Sprungzieldeklarationen bestehen aus dem Wort label gefolgt vom Namen des Sprungziels und einem Semikolon.

Sprunganweisungen bestehen aus dem Wort **jump** gefolgt vom Namen des Sprungziels und einem Semikolon.

Innerhalb von Case-Anweisungen können keine Sprungziele definiert werden. Sprünge von außen in eine Case-Anweisung sind nicht möglich.

Anweisung return

Diese Anweisung verhält sich insofern wie eine Funktion, als sie nur einen einzigen Parameter (**true** oder **false**) kennt und mit diesem Parameter anzeigt, ob eine Funktion erfolgreich ausgeführt wurde. Sie beendet die Ausführung des Skripts und übergibt die Kontrolle dem Benutzer. Mit dem Wert **true** wird angenommen, dass die Anmeldung erfolgreich ausgeführt wurde und die Autorisierung erfolgt ist. Wird der Wert **false** ausgegeben, kann der Benutzer seinen Anmeldeversuch wiederholen oder die Anmeldung abbrechen.

Automatisch

Die Anweisung **autodetect** (automatische Erkennung) ermittelt anhand der letzten Antwort des Servers den Host-Typ des FTP-Servers, zu dem eine Verbindung hergestellt wird. Da die Anweisung autodetect das Begrüßungs-Banner auswerten soll, sollte die Anweisung so angeordnet werden, dass sie unmittelbar nach Empfang des Begrüßungs-Banners erfolgt. Im Folgenden sehen Sie als Beispiele die Banner von zwei verbreiteten FTP-Server-Typen. Die Anweisung autodetect würde den ersten Server als Microsoft NT-Server und den zweiten als Ipswitch WS_FTP-Server erkennen:

Wenn nach einer Direktanmeldung bei einem FTP-Host das Begrüßungs-Banner noch vor Ausführung des Skripts empfangen wurde, sollte die Anweisung **autodetect** die erste Anweisung des Skripts sein. Wenn die Anmeldung bei einer Firewall erfolgt und das Begrüßungs-Banner des Host-FTP-Servers erst empfangen wird, nachdem das Skript gestartet wurde, sollte die Anweisung **autodetect** an der entsprechenden Stelle im Skript stehen. Wenn das Begrüßungsbanner des FTP-Hosts von der Firewall verschluckt bzw. ersetzt wird oder aus sonstigen Gründen nicht zum FTP-Client durchdringt, lassen Sie die Anweisung **autodetect** weg. In diesem Fall versucht Ipswitch WS_FTP Professional, den Host-Typ nach Ablauf des Skripts zu bestimmen.

Wenn für den Host-Typ im Server-Profil nicht die Einstellung Automatisch eingestellt wurde, hat die Anweisung autodetect keine Funktion. Die Anweisung **autodetect** wird dann nicht berücksichtigt und hat keine Auswirkungen auf den Ablauf des Skripts.

SSL-Anweisungen

Die Anweisungen **tryssl** und **gossl** versuchen, mit SSL einen sicheren Kanal zum Server zu eröffnen. Wenn dies nicht möglich ist, wird nach der Anweisung **gossl** der Versuch abgebrochen und der Wert **false** ausgegeben; schlägt die Anweisung **tryssl** fehl, wird das Skript trotzdem fortgesetzt. Diese Anweisungen können in einem Skript auch mehrfach verwendet werden. Beide Anweisungen haben keine Auswirkungen, wenn eine sichere Verbindung nicht angefordert oder bereits hergestellt wurde. Wenn keine sichere Verbindung hergestellt werden kann, wird der Benutzer mit einer entsprechenden Meldung gefragt, ob er ungesichert verbunden bleiben möchte, ob die SLL-Verbindung im weiteren Verlauf des Skripts erneut versucht werden soll, oder ob die Verbindung abgebrochen werden soll. Wenn der Benutzer sich für die ungesicherte Verbindung entscheidet, werden weitere Aufrufe von **tryssl** oder **gossl** übergangen.

Am Ende des Skripts wird der SSL-Status durch die Skript-Steuerung überprüft, um sicherzustellen, dass die Anforderung einer sicheren Verbindung berücksichtigt wurde. Wenn der Benutzer im Server-Profil die Einstellung **SSL verwenden** definiert hat, macht die Skript-Steuerung den Benutzer ggf. mit einem Warnhinweis darauf aufmerksam, dass die Verbindung nicht sicher ist. Der Benutzer kann die Verbindung nun wieder abbrechen. Dieser Warnhinweis wird ausgegeben, wenn versucht wurde, eine gesicherte Verbindung aufzubauen, und der Benutzer entschieden hat, die ungesicherte Verbindung beizubehalten.

Je nach Art der Firewall zwischen Client und Server kann sehr wichtig sein, an welcher Stelle im Skript die SSL-Befehle eingefügt sind.

Schlüsselwörter für FireScripts

Die folgende Liste enthält alle Schlüsselwörter, die in der FireScript-Sprache verwendet und verstanden werden. Diese Wörter dürfen nicht als Sprungzielnamen verwendet werden.

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

false	

Reservierte Wörter für FireScripts

Die folgenden Wörter sind für spätere Versionen der FireScript-Sprache und des Analysealgorithmus reserviert. Diese Wörter sollten Sie ebenfalls nicht als Sprungzielnamen verwenden.

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

FireScript-Anweisungen

gossl	tryssl	autodetect
send	xauth	jump
return	continue	

Interne FireScript-Funktionen

contains	isempty

Interne FireScript-Variablen

FwUserId	FwPassword	FwAccount
FwAddress	HostUserId	HostPassword
HostAccount	HostAddress	LastFtpCode

LastReply		

Themenübersicht

Index