IPSWITCH

WS_FTP

Professional

Sicherheitshandbuch

IPSWITCH

Ipswitch Inc.
10 Maguire Rd
Suite 200
Lexington, MA 02421
USA

Website: http://www.ipswitch.com
Telefon: 781.676.5700
Fax: 781.676.5710

Copyright

Versionsverlauf

Version 9.0 Ausgabe: Juni 2004

Version 2006 Ausgabe: Juni 2005

Anhang A FireScript-Editor

Sichere Dateiübertragung

Anhang 1

Dieses Handbuch beschreibt die in Ipswitch WS_FTP Professional verwendeten Protokolle: SSL, SSH und OpenPGP. Außerdem wird erläutert, wie Sie Ipswitch WS_FTP Professional so konfigurieren, dass mit diesen Protokollen sichere Verbindungen hergestellt werden können.

Dieses Kapitel bietet einen Überblick über die verschiedenen Protokolle und vergleicht die Protokolle miteinander, damit Sie entscheiden können, welches Protokoll für Ihre Anforderungen am besten geeignet ist.

Ein sicheres Übertragungs-verfahren 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?	Eigentliche Datei verschlüsselt?
SSL	Ja	Ja	Ja	Ja	Ja	Nein
SSH	Ja	Ja	Ja	Ja	Ja	Nein
OpenPGP	Ja	Nein	Nein	Nein	Nein	Ja

Themenübersicht

HINWEIS: Sowohl bei SSL als auch bei SSH kann der Administrator des Servers, zu dem Sie eine Verbindung aufbauen möchten, Ihnen mitteilen, welcher Server-Typ unter der betreffenden Adresse eingerichtet wurde. 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.

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.

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.

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

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.

SSL (Secure Sockets Layer)

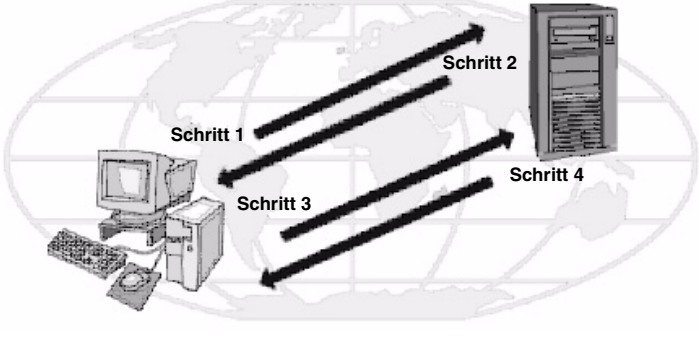
Anhang 2

SSL (**Secure Sockets Layer**) kann in Verbindung mit FTP eingesetzt werden, um die Sicherheit von Standard-FTP-Verbindungen zu erhöhen. Dieses Kapitel bietet einen Überblick über das SSL-Protokoll und beschreibt, wie SSL in Ipswitch WS_FTP Professional genutzt wird.

Übersicht

SSL 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 erläutert, wie SSL funktioniert.



Schritt 1: Der Client meldet sich im Server an und fordert die SSL-Verbindung an. Wenn Einstellung SSL implizit verwendet wird, erfolgt die Anmeldung beim Server bereits in verschlüsselter Form. Mit der Einstellung SSL explizit wird die Anmeldung nicht verschlüsselt.

Themenübersicht

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 werde können, vergleicht er das empfangene Zertifikat mit den in der Liste der vertrauenswürdigen Server gespeicherten Zertifikate, 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.

HINWEIS: SSL können Sie nur dann nutzen, wenn der FTP-Server 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.

Vorgehen zum Herstellen einer SSL-Verbindung

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 dem FTP Professional-Server mitgeteilt, 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, erscheint eine entsprechende Meldung.
4. Wählen Sie die gewünschte Einstellung, 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 FTP-Server beim Client seinerseits von Ipswitch WS_FTP Professional ein Zertifikat anfordert, befolgen Sie die Anweisungen zum Senden eines Client-Zertifikats.

Client-Zertifikat senden

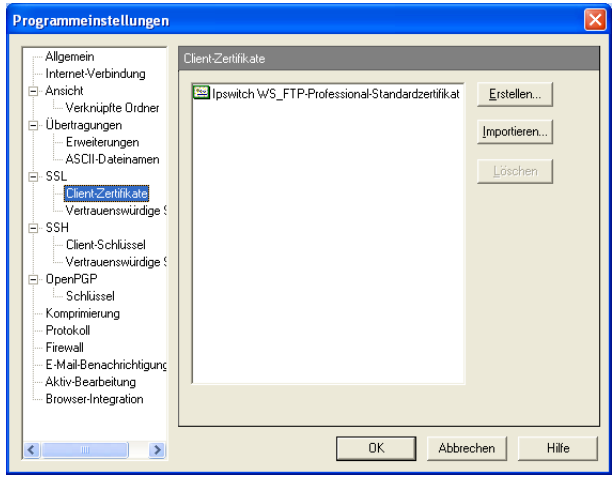
Wenn der Server, zu dem Sie eine Verbindung herstellen möchten, von Ihrem Client die Übertragung eines Zertifikats fordert, verfahren Sie wie folgt:

1. Erstellen Sie ein Server-Profil und wählen Sie den Server-Typ **FTP/SSL implizit** oder **FTP/SSL (AUTH SSL)**.
2. Erstellen Sie ein Zertifikat. Beachten Sie dazu bitte auch die Hinweise im Abschnitt „Zertifikate generieren“ auf Seite 6.
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** aus, wie im Abschnitt „Zertifikate auswählen“ auf Seite 9 beschrieben.
6. Stellen Sie die Verbindung zum Server her.

Zertifikate generieren

So erstellen Sie ein SSL-Zertifikat:

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Nun erscheint das Fenster Programmeinstellungen.
2. Wählen Sie die Einstellung **Client-Zertifikate**.



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 Ipswitch WS_FTP Professional 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 müssen Sie sich merken. Die Kennphrase kann aus beliebigen Wörtern, Symbolen, Leerzeichen und Ziffern bestehen.

7 Klicken Sie auf **Weiter**.

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

Ort: Stadt oder Ort, an dem sich der lokale PC befindet. (z.B. Lüneburg)

Bundesland / Kanton / Bundesstaat: Landesteil, in dem sich dieser Ort befindet. (z.B. Niedersachsen)

Organisation: Unternehmen oder Benutzername

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

E-Mail: E-Mail-Adresse der Person, die das Zertifikat ausgestellt hat

Abt.: Name der Abteilung. (z.B. Forschung und Entwicklung)

Land: Land, in dem Sie sich befinden; geben Sie einen gültigen Ländercode mit zwei Buchstaben ein. (z.B. AT, CH oder DE.)

9 Nachdem Sie alle Eingaben ordnungsgemäß vorgenommen haben, klicken Sie auf **Weiter**. Sie können erst dann fortfahren, wenn alle Felder definiert wurden.

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

Wenn Sie ein Zertifikat für Ipswitch WS_FTP Professional 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**. Nun erscheint das Fenster Programmeinstellungen.

1 Wählen Sie die Einstellung **Client-Zertifikate**, und klicken Sie dann auf die Schaltfläche **Importieren**. Nun erscheint der Assistent zum Importieren von Zertifikaten.

2 Wählen Sie das gewünschte Zertifikat aus und klicken Sie auf **Weiter**.

3 Wählen Sie die Datei mit dem privaten Schlüssel für dieses Zertifikat aus und klicken Sie auf **Weiter**.

4 Geben Sie die zum Erstellen des Zertifikats verwendete Kennphrase ein und klicken Sie auf **Weiter**.

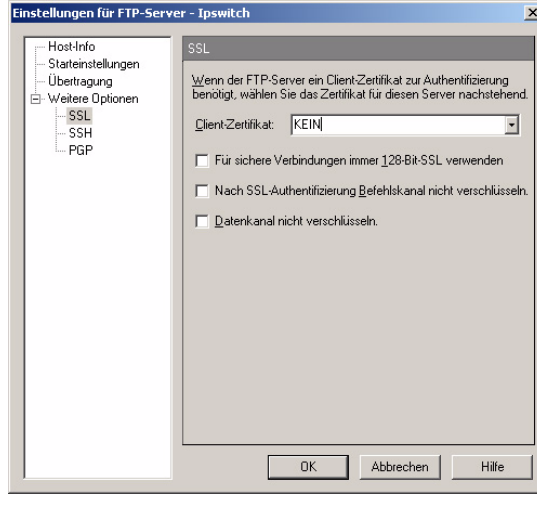
5 Geben Sie den Namen ein, unter dem das Zertifikat in Ihrer Datenbank gespeichert werden soll, und klicken Sie auf **Weiter**.

6 Überprüfen Sie im zuletzt geöffneten Dialogfeld die angezeigten Informationen und klicken Sie auf **Fertigstellen**, um das Zertifikat zur Datenbank hinzuzufügen.

Zertifikate auswählen

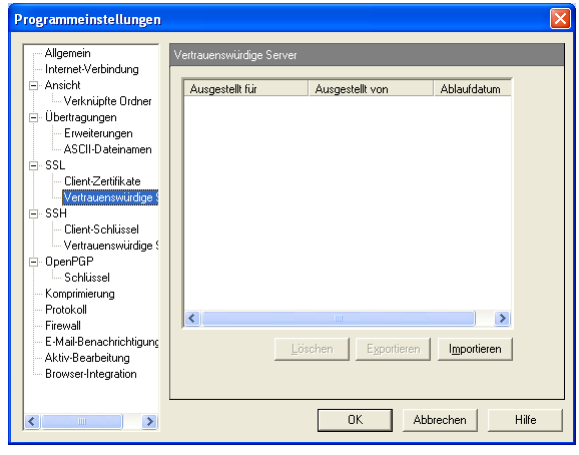
Zertifikate werden auf Server-Ebene verwendet; daher müssen Sie für alle von Ihnen erzeugten FTP-Server-Profile ein Zertifikat auswählen. (Sie können allerdings für all Ihre Server das gleiche Zertifikat verwenden.)

Zertifikate werden über das Dialogfeld Einstellungen für FTP-Server auf dem Registerblatt **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.



Angezeigte Daten:

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

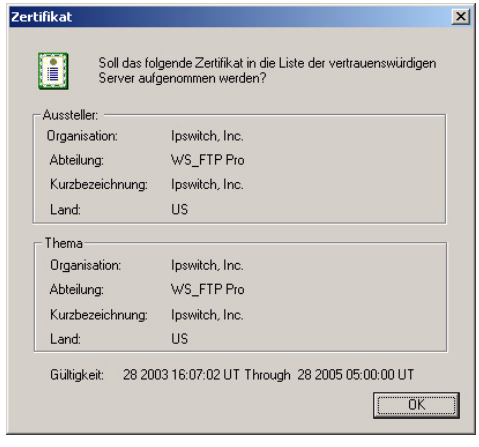
Ausgestellt von: Gibt an, von wem das Zertifikat unterzeichnet wurde.

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. Nun erscheint das Dialogfeld **Zertifikat**.



2 Überprüfen Sie im zuletzt geöffneten Dialogfeld die angezeigten Informationen und klicken Sie auf **Ja**, um das Zertifikat zur Datenbank hinzuzufügen.

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 die Schaltfläche **Exportieren**.

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.

3 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 erscheint die Empfehlung, 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, erscheint dieses Dialogfeld:

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 unterzeichnet wurde

Aktiv ab: Datum, an dem das Zertifikat aktiviert wurde

Ablaufdatum: Datum, an dem das angezeigte Zertifikat ungültig wird

Einstellungen

Nur diese Verbindung zulassen: Wurde diese Option ausgewählt, wird die Verbindung zwar hergestellt, Ipswitch WS_FTP Professional stuft das Zertifikat aber nicht als vertrauenswürdig ein. 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 ausgewählt wurde, wird die Verbindung abgebrochen.

NAT-Firewall in der Praxis

Wenn Sie eine Firewall mit NAT (Network Address Translation) verwenden, können in Verbindung mit SSL-Verschlüsselungen Probleme auftreten. Um diese zu beheben, sollten Sie Ipswitch WS_FTP Professional und die Firewall so konfigurieren, dass sie ankommende Verbindungen zum PC akzeptieren. Ipswitch WS_FTP Professional muss den Server anweisen, eine Verbindung zur externen IP-Adresse herzustellen, und die Firewall sollte diese eingehenden Verbindungen an Ihren Rechner weiterleiten. Außerdem sollten Sie die Anzahl der Ports beschränken, die die Firewall für diese Verbindungen öffnet. Häufig können Sie dann SSL auch bei NAT-Firewalls verwenden.

SSL über eine NAT-Firewall konfigurieren:

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Nun erscheint das Fenster Programmeinstellungen.
2. Wählen Sie **SSL**. Nun erscheint das Dialogfeld SSL.
3. Wählen Sie die Option **Port-IP-Adresse erzwingen**.
4. Geben Sie die **IP-Adresse** der NAT-Firewall Ihres Client ein.
5. Wählen Sie die Option **Lokalen Port-Bereich begrenzen**.
6. Geben Sie den **Mindest** und den **Maximalen Port-Bereich** ein.
7. Klicken Sie auf **OK**.

SSH (Secure Shell)

Anhang 3

Das SSH-Protokoll (SSH = **Secure Shell**) 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.

HINWEIS: Ipswitch WS_FTP Professional unterstützt nur SFTP/SSH2.

Warum wird SSH verwendet?

SSH erhöht die Sicherheit von Standard-FTP-Verbindungen, indem sämtliche Elemente der Verbindung und des Übertragungsvorgangs verschlüsselt und geschützt werden.

HINWEIS: SSH können Sie nur dann nutzen, wenn der FTP-Server SSH-Verbindungen 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.

Themenübersicht

SSH-Verbindungen herstellen

Um SSH-Verbindungen herzustellen, brauchen neue und bestehende Server-Profile kaum geändert zu werden.

Beim Erzeugen eines neuen Server-Profils mit dem **Verbindungsassistenten** 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. Wählen Sie den gewünschten Server aus der Liste **Konfigurierte Server** aus.
2. Klicken Sie auf die Schaltfläche **Bearbeiten**.
3. Klicken Sie auf das Registerblatt **Erweiterte Optionen**.
4. Wählen Sie im Pulldown-Listenfeld **Server-Typ** die Option **SFTP/SSH**. Klicken Sie auf **OK**.
5. Wählen Sie das gewünschte Authentifizierungsverfahren aus:
   - **Kennwort** - Wenn Ihr Server eine Kennwort-Authentifizierung erfordert, ist die Konfiguration damit abgeschlossen. Bei der nächsten Anmeldung bei diesem Server wird die Verbindung dann mit SSH geschützt.
   - **Öffentlicher Schlüssel** - Erfolgt die Authentifizierung auf Ihrem Server mit einem öffentlichen Schlüssel, wählen Sie **Erweiterte Optionen > SSH**. Wählen Sie unter **SSH-Schlüsselpaar** das benötigte Schlüsselpaar aus. Werden keine Schlüsselpaare angezeigt, können Sie das nötige Schlüsselpaar selbst erstellen oder importieren.
6. Klicken Sie auf **OK**, um das Dialogfeld Einstellungen für FTP-Server zu schließen.
7. Klicken Sie auf **Schließen**, um das Dialogfeld Server-Manager zu schließen.

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

SSH-Schlüsselpaare generieren

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Nun erscheint das Fenster Programmeinstellungen.
2. Wählen Sie **SSH > Client-Schlüssel**.
3. Klicken Sie auf **Erstellen**. Nun erscheint der SSH-Client-Schlüsselpaar-Generierungsassistent.
4. Befolgen Sie die Anweisungen des Assistenten.

Öffentliche SSH-Schlüssel exportieren

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

OpenPGP

Anhang 4

OpenPGP kann die Sicherheit von Standard-FTP-Verbindungen erhöht werden. In diesem Kapitel wird erläutert, wie OpenPGP in Ipswitch WS_FTP Professionalfunktioniert. Dazu werden die einzelnen Schritte beim Übertragen von mit OpenPGP verschlüsselten Dateien beschrieben. Anhand eines Beispiels wird dargestellt, wie Standardprobleme im Geschäftsalltag mit OpenPGP gelöst werden können.

HINWEIS: Die in Ipswitch WS_FTP Professional 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.

Übersicht

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

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.

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

Themenübersicht

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.

Den OpenPGP-Modus aktivieren

Der PGP-Modus wird nach dem Herstellen der Verbindung zum Server aktiviert.

1. Klicken Sie in Ipswitch WS_FTP Professional auf das Registerblatt des FTP-Servers.
2. Wählen Sie **Extras > OpenPGP-Modus** oder klicken Sie in der Symbolleiste auf **OpenPGP-Modus**. Nun erscheint das Dialogfeld OpenPGP-Modus.
3. Wählen Sie aus den angezeigten Optionen das gewünschte OpenPGP-Übertragungsverfahren aus.
   - **Verschlüsseln** Sie die Dateien mit einem Schüssel 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.
   - Wählen Sie die Einstellung **Verschlüsseln und Unterzeichnen**, wenn Sie beide Optionen gleichzeitig verwenden möchten.
4. Klicken Sie auf **OK**, um das Dialogfeld zu schließen. Damit ist der OpenPGP-Modus aktiviert, bis die betreffende Verbindung beendet wird bzw. bis Sie den PGP-Modus ausdrücklich wieder deaktivieren.

Den OpenPGP-Modus als Voreinstellung für einen Server aktivieren

HINWEIS: 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.

1. Wählen Sie im Hauptfenster **Verbinden > Server verwalten**. Nun erscheint das Dialogfeld Server verwalten.
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**. Nun erscheint das Dialogfeld Einstellungen für FTP-Server.
3. Klicken Sie auf **Erweitert** und wählen Sie die Option **OpenPGP**. Nun erscheint das Dialogfeld OpenPGP-Optionen.
4. Definieren Sie die Einstellung **Nach Verbinden OpenPGP-Übertragung**.
5. Wählen Sie aus den angezeigten Optionen das gewünschte OpenPGP-Übertragungsverfahren aus.
   - Verschlüsseln Sie die Dateien mit einem Schüssel 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.
   - Wählen Sie die Einstellung **Verschlüsseln und Unterzeichnen**, wenn Sie beide Optionen gleichzeitig verwenden möchten.
6. Klicken Sie auf **OK**, um Ihre Einstellungen zu speichern und das Dialogfeld zu schließen.

Schlüsselpaare generieren

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Nun erscheint das Fenster Programmeinstellungen.
2. Wählen Sie **OpenPGP > Schlüssel**.
3. Klicken Sie auf **Erstellen**. Nun erscheint der OpenPGP-Schlüssel-Generierungsassistent.
4. Befolgen Sie die Anweisungen des Assistenten, um den Schlüssel zu erstellen.

Schlüssel importieren

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Nun erscheint das Fenster Programmeinstellungen.
2. Wählen Sie **OpenPGP > Schlüssel**.
3. Klicken Sie auf **Importieren**. Nun erscheint der OpenPGP-Schlüssel-Importassistent.
4. Befolgen Sie die Anweisungen des Assistenten, um den Schlüssel zu importieren.

Schlüsselpaare exportieren

1. Wählen Sie im Hauptfenster **Extras > Optionen**. Nun erscheint das Fenster Programmeinstellungen.
2. Wählen Sie **OpenPGP > Schlüssel**.
3. Wählen Sie den zu exportierenden Schlüssel aus und klicken Sie auf **Exportieren**. Nun erscheint der OpenPGP-Schlüssel-Export assistent.
4. Befolgen Sie die Anweisungen des Assistenten, um die Schlüssel zu exportieren.

Beispiel

Im folgenden Praxisbeispiel sehen Sie, wie OpenPGP im Geschäftsalltag verwendet werden kann.

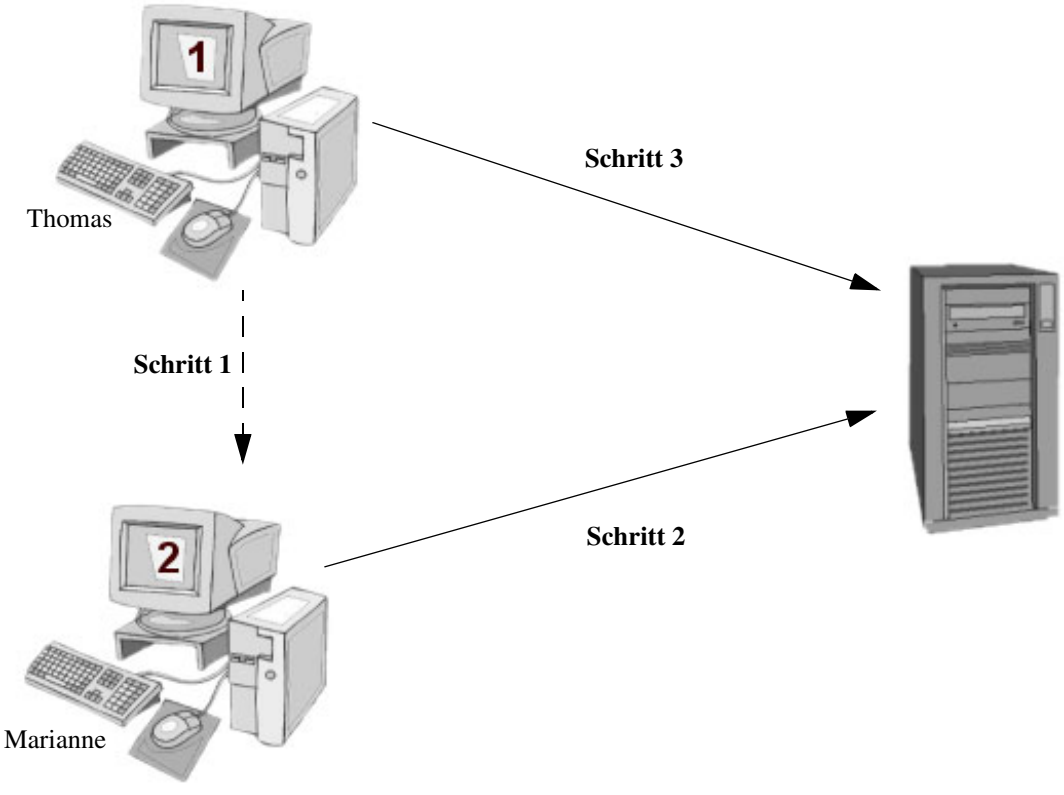
Aufgabe

Marianne, die im Regionalbüro eines Unternehmens beschäftigt ist, muss regelmäßig vertrauliche Mitarbeiterdaten aus ihrem Büro an Thomas am Hauptsitz ihres Unternehmens senden. Bisher hat sie die Daten auf eine CD-ROM gebrannt, die dann auf dem Postweg geschickt wurde. Nun sucht sie eine kostengünstigere und zeitsparendere Lösung.

Einschränkungen

Die zu übertragenden Daten sind für den Versand per E-Mail viel zu umfangreich. Außerdem müssen die Übertragungen geschützt werden, damit niemand außer Thomas auf die Daten zugreifen kann.

Lösung



1 Thomas sendet seinen öffentlichen Schlüssel per E-Mail an Marianne. Er kann mit Ipswitch WS_FTP Professional einen Schlüssel erzeugen oder exportieren.

2 Marianne verschlüsselt die Datei und lädt die Datei mit dem öffentlichen Schlüssel von Thomas hoch.

- **a** Sie importiert den Schlüssel in Ipswitch WS_FTP Professional.
- **b** Sie stellt eine Verbindung zum FTP-Server des Unternehmens her.
- **c** Sie aktiviert den OpenPGP-Modus mit dem Schlüssel, den Thomas für die Verschlüsselung verwendet hat.
- **d** Sie lädt die Datei hoch.

3 Thomas lädt die Datei herunter und entschlüsselt die Datei mit seinem privaten Schlüssel.

Firewalls verwenden

Anhang 5

In manchen Unternehmen ist das lokale Netz durch eine Firewall oder ein Gateway vom übrigen Internet getrennt. Firewalls sind Hardware- oder Softwarelösungen, die bestimmte Arten von Daten oder Zugangsanforderungen in einem Netz unterbinden sollen. Die meisten Firewalls blockieren den Datenfluss in das lokale Netz, wobei jedoch bestimmte Benutzer auf Ressourcen außerhalb des Netzes zugreifen können.

Bei Ipswitch WS_FTP Professional können Sie Informationen zu Firewalls jeweils in eine Firewall-Konfiguration eintragen, die Sie dann für die Verbindung zu einem bestimmten FTP-Server auswählen können. Sie können die Firewall in diesem Fall einmal konfigurieren und dann die Konfiguration immer wieder für die Server verwenden, für die diese Konfiguration benötigt wird.

Mit dem FireScript-Editor können Sie Firewall-Skripts individuell konfigurieren. Weitere Informationen finden Sie in „Anhang A: FireScript-Editor“ auf Seite 29.

Mehrere Firewalls

Aus verschiedenen Gründen kann es erforderlich bzw. empfehlenswert sein, mehrere Firewall-Konfigurationen zu erstellen. Wenn Sie beispielsweise mit einem Notebook an verschiedenen Standorten arbeiten, die sich hinter unterschiedlichen Firewalls befinden, müssen Sie eine Firewall-Konfiguration für jeden Standort einrichten, damit Sie für jeden Standort die erforderliche Konfiguration zur Verfügung haben.

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

Oder es könnte eine Reihe vertrauenswürdiger Server existieren (z.B. FTP-Server, die Ihr Unternehmen selbst betreibt), für die eine andere Firewall zu verwenden ist (oder bei denen auch völlig auf eine Firewall verzichtet wurde).

Themenübersicht

Firewall-Typen

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

Firewall-Typ	**Informationen, die Sie in Ipswitch WS_FTP Professional eingeben müssen**
GEÖFFNETER Proxy	Host-Name (oder IP-Adresse):
SERVER-Host-Name	Host-Name (oder Adresse), Benutzername (ID)
Transparent	Benutzername (ID), Kennwort
BENUTZER nach Anmeldung	Host-Name (oder Adresse), Benutzername (ID), Kennwort
BENUTZER fireID@remoteHost	Host-Name (oder Adresse), Benutzername (ID), Kennwort
BENUTZER remoteID@fireID @remoteHost	Host-Name (oder Adresse), Benutzername (ID), Kennwort
BENUTZER remoteID @remoteHost fireID	Host-Name (oder Adresse), Benutzername (ID), Kennwort
BENUTZER ohne Anmeldung	Host-Name (oder IP-Adresse):
SOCKS4 und SOCKS5	Host-Name (oder Adresse), Benutzername (ID), Kennwort

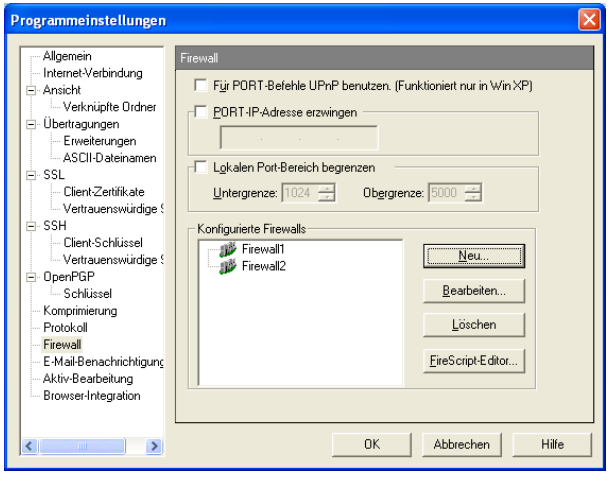
Firewalls konfigurieren

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

HINWEIS: 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 Ipswitch WS_FTP Professional) aufgebaut.

So konfigurieren Sie eine Firewall:

1. Wählen Sie **Extras > Optionen**.
2. Öffnen Sie das Dialogfeld **Firewall**.
3. Klicken Sie auf **Neu**.
4. Folgen Sie den Anweisungen im Assistenten **Neue Firewall**.
5. Wenn Sie auf **Fertigstellen** klicken, wird die Firewall zur Liste **Konfigurierte Firewalls** hinzugefügt.



Die gespeicherte Firewall-Konfiguration können Sie nun dem Server zuweisen, wie im folgenden Abschnitt **Konfigurierte Firewalls verwenden** beschrieben.

Konfigurierte Firewalls verwenden

Sobald Sie eine Firewall konfiguriert haben, können Sie Ihre Konfiguration auf einen FTP-Server anwenden.

Dazu verfahren Sie im Server-Manager wie folgt:

1. Wählen Sie einen Server aus.
2. Klicken Sie auf die Schaltfläche **Bearbeiten**.
3. Klicken Sie auf **Weitere Optionen**.
4. Wählen Sie im Feld **Firewall** die gewünschte Firewall-Konfiguration.

UPnP verwenden

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

Aktivierung vo UPnP:

1. Klicken Sie in der Symbolleiste auf **Optionen**, oder wählen Sie **Extras > Optionen**. Nun erscheint das Fenster Programmeinstellungen.
2. Wählen Sie **Firewall**.
3. Wählen Sie **&Für PORT-Befehle UPnP benutzen**.

FireScript-Editor

Anhang A

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

Was ist ein FireScript?

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 kann die gleichen Funktionen wie Ipswitch WS_FTP Professional zur Verbindung mit einem Host oder Firewall-Typ verwenden. 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.

Themenübersicht

Der Abschnitt **comment** besteht aus Kommentaren in natürlicher Sprache, die nicht an eine bestimmte Form gebunden sind. Die Steuerung übergeht diese Kommentare. Sie werden von dem von der Steuerung auszuführenden Skript 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]

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

… 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 (300.0,399) :

```
continue ;

case any :

return (false) ;
}
```

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

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

Der Abschnitt fwsc

Im Abschnitt **fwsc** können Sie Informationen zu Ihrem Skript ähnlich wie in Windows-Initialisierungsdateien 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.
zwingend erforderlich	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. Die FireScript-Steuerung übergeht den Abschnitt comment.

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 Authorisieren 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:

```
contains
(

lastreply,
"Willkommen auf meinem coolen FTP-Server"
)
;
```

Dieser Code dagegen wäre nicht gültig:

```
contains (lastreply, „Willkommen auf
meinem coolen FTP-Server");
```

FireScript-Variablen

Für die FireScripts werden die in Ipswitch WS_FTP Professional definierten Anmeldedaten benutzt. 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, erscheint wie bereits beschrieben beim Verbindungsaufbau der Anmeldedialog, 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 auf dem FTP-Server; diese Eingabe ist fast immer erforderlich. Wenn für einen Benutzer auf dem Server kein Name definiert wurde, ist die Eingabe ‚anonymous' vorzunehmen.
HostPassword	Kennwort des Benutzers auf dem FTP-Server. Wenn ein Benutzername eingegeben werden muss, ist auch diese Eingabe fast immer 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
„%“ und aus der Zeichenfolge „%“ plus Variable wird der Wert der Variable ohne %.

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

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

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

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

Der Ausdruck

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

wird beim Ausführen des Skripts umgesetzt in:

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

und die Anweisung

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

wird beim Ausführen des Skripts so umgesetzt:

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

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

Beispiel:

```
isempty(FwPassword)
```

bedeutet dasselbe wie, ist aber schneller als

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

Funktionsausdrücke

Die FireScript-Sprache unterstützt zurzeit keine komplexeren Ausdrücke. Sie beinhaltet aber zwei Funktionsausdrücke mit einigen Booleschen Operatoren zur Statusbestimmung von Variablen. Diese Ausdrücke heißen **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, und
not isempty(HostAccount) ergibt den Wert true.

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

Nehmen wir z.B. an, die Variable HostAccount z.B. 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:

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

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:

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

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 nach die xauth-Einladung des Ipswitch WS_FTP-Servers 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 wurde, 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 Funktionsausdrükken 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:

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

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

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

```
case (300..399) and contains(lastreply, "email address") :
```

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:

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

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 999 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. Die betreffende Anweisung könnte dann z.B. so lauten:

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

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:

```
case timeout :
```

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:

```
case any:
```

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

Beispiel:

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

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

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

Anweisung continue

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 anschließendenAnweisung. Sie hat somit die gleiche Funktion wie die Anweisung Break in Switch-Anweisungen in C/C++; diese Anweisung is 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 Authorisierung erfolgt ist. Wird der Wert „false“ ausgegeben, kann der Benutzer seinen Anmeldeversuch wiederholen oder die Anmeldung abbrechen.

Anweisung autodetect

Die Anweisung **autodetect** (automatische Erkennung) ermittelt aufgrund 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:

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

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

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 Servers erst empfangen wird, nachdem das Skript gestartet wurde, sollte die Anweisung **autodetect** an der entsprechenden Stelle im Skript stehen. Wenn das Begrüßungsbanner des Servers 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

Mit den Anweisungen **tryssl** und **gossl** wird versucht, 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	und	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	

Spezifische Funktionen für FireScripts

contains	isempty

Interne FireScript-Variablen

FwUserId	FwPassword	FwAccount
FwAddress	HostUserId	HostPassword
HostAccount	HostAddress	LastFtpCode
LastReply		