zurück

FTP, FTPS & SFTP – Grundlagen

FTP (**F**ile **T**ransfer **P**rotocol) ist eines der ältesten Protokolle zur Dateiübertragung im Netzwerk. Heute wird es oft durch sicherere Alternativen ersetzt, ist aber weiterhin in vielen Umgebungen notwendig (Industrie, Embedded, Router, Firmware, Legacy-Anwendungen).

Diese Seite erklärt:

- FTP (klassisch, unsicher)
- FTPS (FTP + TLS)
- SFTP (SSH File Transfer Protocol)

---

1. FTP – File Transfer Protocol

FTP ist ein unverschlüsseltes Übertragungsprotokoll aus den 1970ern.

Ports & Funktionsweise

FTP verwendet **zwei Verbindungen**:

- **Port 21 (TCP)** – Kontrollkanal
- **Port 20 (TCP)** – Datenkanal (aktiv)
- oder dynamische Ports (passiv)

```
Client ——— Port 21 (Kontrolle)
Client ——— Port 20 oder Pasv-Port (Daten)
```

Aktiver vs. Passiver Modus

Aktiver Modus (active FTP)

Server verbindet sich zum Client → Problem bei Firewalls.

```
Client ── Server:21  (Kontrolle)

Server ── Client:random_port  (Daten)
```

Passiver Modus (passive FTP)

Heutiger Standard, da firewall-freundlicher.

```
Client ⟷ Server:21  (Kontrolle)

Client ⟷ Server:random_pasv_port  (Daten)
```

Nachteile von FTP

- KEINE Verschlüsselung
- Passwörter im Klartext
- leicht durch MITM[1)] angreifbar
- sehr unflexibel in modernen Netzwerken

In modernen Umgebungen sollte FTP nur in **isolierten VLANs** betrieben werden.

---

2. FTPS – FTP über TLS

FTPS fügt TLS-Verschlüsselung hinzu – vergleichbar mit HTTPS.

Zwei Varianten:

- **Explicit FTPS**
  - Start auf Port 21
  - Client fordert TLS via `AUTH TLS` an
- **Implicit FTPS**
  - direkt verschlüsselt
  - Port **990**
  - heute eher legacy, aber noch unterstützt

Vorteile:

- Verschlüsselt
- Nahezu identisch zu FTP
- Besser für Compliance als „plain FTP“

Nachteile:

- komplexer wegen vielen Ports
- Firewalls müssen passive Portbereiche erlauben

---

3. SFTP – SSH File Transfer Protocol

WICHTIG:
SFTP hat **NICHTS** mit FTP zu tun, außer dass der Name ähnlich ist.

Es basiert auf **SSH**, nutzt also:

- **Port 22**
- vollständige Verschlüsselung
- starke Authentifizierung (Passwort, Schlüssel)
- nur EIN Datenkanal (Firewall-freundlich)

```
Client ⟷ Server:22 (SSH) ⟷ SFTP-Session
```

SFTP ist moderner und sicherer und wird heute für nahezu alle professionellen Transfers genutzt (Backup, Automatisierung, CI/CD, Scripts).

---

4. FTP, FTPS, SFTP – Vergleich

Protokoll	Port	Verschlüsselt	Firewallfreundlich	Sicherheit
FTP	21/20	✗ nein	✗ nein	✗ unsicher
FTPS	21/990	✔ ja	✗ kompliziert	✔ gut
SFTP	22	✔ ja	✔ sehr gut	✔✔ sehr sicher

---

5. Wann verwendet man was?

FTP

- nur in alten Systemen oder isolierten VLANs
- Firmware-Uploads alter Hardware
- Industrieanlagen

FTPS

- wenn alte Software zwingend FTP benötigt
- aber Sicherheit vorgeschrieben ist
- Banking, Unternehmensübertragungen

SFTP

- Standard in modernen Architekturen
- automatisierte Skripte
- sichere Dateiübertragungen
- Backup-Jobs
- CI/CD Pipelines
- DevOps

---

6. Beispiel-Befehle

FTP

```
ftp server.example.com
get file.txt
put upload.bin
```

FTPS (Explicit)

```
lftp -e "set ftp:ssl-force true" ftps://server.example.com
```

SFTP

```
sftp user@server.example.com
sftp> get logs.zip
sftp> put backup.tar.gz
```

Für automatisierte Jobs:

```
sftp -i ~/.ssh/key user@host
```

7. Sicherheitsempfehlungen

- normaler FTP NICHT über das Internet
  - am besten ersetzen durch SFTP
    * passive Ports bei FTPS klar definieren
    * Server in eigene VLANs packen
    * nur sichere Ciphers verwenden
    * Logs überwachen (Auth-Logs, Brute Force)

Zusammenfassung

- FTP = alt und unsicher
  - FTPS = FTP mit TLS, sicherer aber komplex
    * SFTP = modernes, SSH-basiertes Protokoll, Standard heute
    * in echten Umgebungen ist SFTP die richtige Wahl
    * FTP/FTPS nur nutzen, wenn notwendig oder altlastbedingt

1)

Man in the Middle Angriff