

[zurück](#)

# TFTP & PXE - Grundlagen

TFTP (**T**rivially **F**ile **T**ransfer **P**rotocol) und PXE (**P**reboot **E**xecution **E**nvironment) werden häufig für Netzboot, Firmware-Updates, Konfigurationsuploads und automatisierte Betriebssysteminstallationen genutzt.

## TFTP - Trivial File Transfer Protocol

TFTP ist ein sehr einfaches Dateiübertragungsprotokoll.

Es verzichtet bewusst auf Funktionen wie:

- Authentifizierung
- Verschlüsselung
- Verzeichnisstrukturen
- Rechteverwaltung

Es wird eingesetzt in:

- Netzwerkgeräten (Switches, Router)
- PXE-Boot
- Embedded-Systemen
- einfachen Firmware-Downloads

### Port & Protokoll

- **69/UDP**
- verbindungslos, schnell, aber unsicher

## Funktionsweise von TFTP

Ablauf:

1. Client sendet Anfrage (RRQ/WRQ) an Port 69
2. Server öffnet einen **neuen Port** für die Datenübertragung
3. Datei wird blockweise übertragen (512 Byte)
4. Client bestätigt jeden Block
5. Übertragung endet bei kleinem Block (<512 Byte)

ASCII:

```
Client → UDP/69 → TFTP-Server  
→ Dynamischer Port → Datenübertragung
```

## Häufige TFTP-Befehle (Linux)

```
tftp 192.168.1.10  
tftp> get config.cfg  
tftp> put firmware.bin
```

## Vor- und Nachteile von TFTP

### Vorteile

- extrem leichtgewichtig
- perfekt für Bootloader, kleine Systeme
- ideal für PXE

### Nachteile

- keine Sicherheit
- keine Authentifizierung
- leicht manipulierbar
- langsamer als moderne Protokolle

## PXE - Preboot Execution Environment

PXE ermöglicht es, **einen Computer über das Netzwerk zu booten**, ohne Festplatte oder lokales Betriebssystem.

Es basiert auf:

- DHCP
- TFTP
- einem Netzwerk-Bootloader

## PXE-Boot Ablauf

Kompletter Boot-Vorgang Schritt für Schritt:

1. Client startet (BIOS/UEFI)
2. NIC sendet DHCP-Discover mit PXE-Option
3. DHCP-Server sendet Adresse + PXE-Optionen (Option 66 & 67)
4. Client lädt Bootloader via TFTP
5. Bootloader lädt Kernel + Initrd via TFTP
6. Betriebssystem startet Installationsroutine oder Live-System

## PXE - Wichtige DHCP-Optionen

- **Option 66** → TFTP-Server-Adresse
- **Option 67** → Name des Bootfiles (z. B. pxelinux.0)

Beispiel:

```
option tftp-server-name "192.168.1.10";
option bootfile-name "pxelinux.0";
```

## Verzeichnisstruktur eines PXE-Servers

Typisch bei Linux PXE-Umgebungen:

```
/tftpboot/
/tftpboot/pxelinux.0
/tftpboot/ldlinux.c32
/tftpboot/menu.cfg
/tftpboot/images/debian/
/tftpboot/images/windows/
/tftpboot/initrd.img
/tftpboot/vmlinuz
```

## Anwendungsfälle von PXE

- Masseninstallation von PCs (Schulen, Firmen)
- Automatisierte Linux-Deployments (z. B. via Foreman, MAAS)
- Diskless Clients
- Rettungssysteme / Live-Systeme

- Imaging (Clonezilla, FOG Server)

## Sicherheit bei PXE & TFTP

Da beide Protokolle **keine Sicherheit** bieten:

- immer in isolierten VLANs verwenden
  - Zugang zu TFTP-Server **einengen**
    - \* Boot-Pfade schützen
    - \* alternative sichere Mechanismen prüfen (iPXE via HTTPS)

Modernere PXE-Varianten (iPXE) unterstützen:

- HTTP/HTTPS
  - Authentifizierung
    - \* fortgeschrittenes Boot-Scripting

## ASCII-Diagramm: PXE Bootkette

```
Client —DHCP—> DHCP-Server (Option 66/67)
      ↓
lädt Bootloader via TFTP
      ↓
lädt Kernel + Initrd via TFTP
      ↓
Start des Installers / Betriebssystems
```

## Zusammenfassung

- TFTP = einfaches Datei-Transferprotokoll (UDP 69), keine Sicherheit
  - PXE = Netzwerkboot, nutzt DHCP + TFTP
    - \* Option 66/67 steuern PXE-Server und Bootloader
    - \* ideal für Masseninstallationen und OS-Deployments
    - \* moderne Varianten nutzen iPXE (auch HTTP/HTTPS möglich)

From:  
<http://wiki.nctl.de/dokuwiki/> - □ **Veni. Vidi. sudo rm -rf / vici.**

Permanent link:  
[http://wiki.nctl.de/dokuwiki/doku.php?id=it-themen:grundlagen:netzwerkdienste:tftp\\_pxe&rev=1764590817](http://wiki.nctl.de/dokuwiki/doku.php?id=it-themen:grundlagen:netzwerkdienste:tftp_pxe&rev=1764590817)

Last update: **01.12.2025 13:06**

