Troubleshooting Cisco Devices

In diesem Kapitel erfahren Sie

- welche Tools zur Informationsermittlung und Fehlererkennung auf Cisco Geräten zur Verfügung stehen
- wie show Kommandos – als umfangreichstes Tool – verwendet werden können
- wie debug Kommandos arbeiten
- wie das Kommando ping arbeitet und eingesetzt werden kann
- wie das Kommando traceroute arbeitet und eingesetzt werden kann
- wie telnet verwendet werden kann um TCP-basierte OSI Applikationen zu testen

Troubleshooting Übersicht

Troubleshooting bezeichnet gebräuchlicher Methoden um auf Cisco Geräten
- Informationen zu ermitteln (System, Status, Funktionen)
- Konfigurationen und Funktionen zu überprüfen
- Fehler zu finden und zu beheben
- Multiple Geräte bzw. ganze Netzwerke zu überwachen

Mit Troubleshooting Kommandos können Eigenschaften und Arbeitsweise vieler Funktionen ermittelt werden – sie können (und sollten) daher exzessiv verwendet werden, um
- Konfigurationen und Funktionalitäten zu verifizieren (immer direkt nach Konfiguration)
- sich einen Überblick über die Funktionsweise der Geräte und Protokolle zu verschaffen.

Grundlegende Möglichkeiten (werden in diesem Modul näher erläutert)
- **show** Kommando
  Mächtigstes Tool auf Cisco Geräten: Anzeige nahezu aller möglichen Informationen
- **debug** Kommando
  Anzeige von Informationen in Echtzeit ("was das Geräte gerade tut")

- **ping** Kommando (OSI 3)
  Verwendet ICMP ("Echo Request" und "Echo Reply"), um aktive Hosts zu ermittle
- **traceroute** Kommando (OSI 3)
  Verwendet ICMP ("TTL exceeded), um den Weg eines Pakets nachzuverfolgen
- **telnet** Kommando (OSI 7)
  Mit Telnet kann die korrekte Funktionsweise von Diensten auf OSI Layer 7 überprüft werden

Einige erweiterte Möglichkeiten (werden teilweise in nachfolgenden Modulen näher erläutert)
- **CDP** und **LLDP** (OSI 2)
  Cisco Discovery Protocol (Cisco) und Link Layer Discovery Protocol (Standard). Protokolle für einen Layer-2 Informationsaustausch zwischen den Geräten.
- **Logging**
  Steuerung der Log-Ausgaben zur Überwachung des Netzwerks: Ausgabe von Log- und Debug-Meldungen können nicht nur lokal auf die Console, auf ein Termial (remote Verbindungen via Telnet/SSH) und in einen internen Puffer in den RAM des Geräts geschrieben werden, sondern können auch an remote einen zentralen SYSLOG Server gesendet werden. Dort werden sie dann dauerhaft gesichert.
- **SNMP – Simple Network Managment Protocol**
  Mit SNMP sendet das Cisco Gerät (Server) Status- und Systeminformationen – auf Anfrage oder unaufgefordert – an ein NMS – Network Management System (Client). Ein NMS bereitet die Informationen z.B. grafisch auf, so dass sie zur Überwachung zur Verfügung stehen. Mit einem leistungsfähigen NMS können multiple Funktionen aller Geräte und somit gesamte Netzwerke effizient überwacht werden.
- **Netflow**
  Einfache Methode, um TCP Datenströme (TCP Flows) zu überwachen.
  Von Cisco entwickelt, um Datenverkehr zu analysieren – Statistiken über die Nutzung unterschiedlicher TCP Verbindungen - mittlerweile Standard.
  Ein Cisco Gerät senden Informationen zu den Flows an einen Netflow Collector (eigenes Programm oder oft als Bestandteil eines "guten" NMS zu finden), der diese dann grafisch aufbereitet darstellt.

show Kommandos

Das wohl wichtigste IOS Werkzeug zum Troubleshooting sind **show** Kommandos.
Mit show Kommandos können Informationen über das Gerät und über alle Funktionen auf dem Gerät ermittelt werden.

Übersicht wichtiger show Kommandos

Kommando	**Informationen**
Allgemeine Informationen: System, Speicher, Konfiguration	
# **show version**	Hard- und Software des Geräts, Uptime, verwendete IOS Datei, Speichergrössen, verfügbare Schnittstellen
# **show flash**	Inhalt und Grösse des Flash Memory
# **show running-config**	Inhalt der aktuellen Konfigurationsdatei im RAM
# **show startup-config**	Inhalt der Konfigurationsdatei im NVRAM
# **show clock**	aktuelle Systemzeit: Uhrzeit, Zeitzone und Datum
Informationen zu Schnittstellen	
# **show ip interface brief**	Tabellarische Übersicht der Schnittstellen mit IP Adressiee und Status
# **show protocols**	Aktive L3 Funktionen: IP Routing und IP Schnittstellen mit IP Adresse und Maske
# **show interfaces** [*IF-Typ IF-Nr*]	Nähere Informationen zu Schnittstellen: Status, IP Adresse und Maske, MAC Adresse, MTU, Bandbreite, Statistiken zum Datenverkehr
# **show controllers** [*IF-Typ IF-Nr*]	OSI 1 Funtionalität der Schnittstellen
# **show ip arp**	Inhalt des ARP Cache
Informationen zum Switching, Routing	
# **show mac-address-table** # **show mac address-table**	nur Switch: Inhalt der MAC Address Table
# **show ip route**	nur Router: Inhalt der Routing Tabelle
Informationen zu Funktionen	
# **show logging**	Einstellungen zum Logging und Inhalt des Log Buffer
# **show users**	Angemeldete Benutzer bzw. aktive "Lines" auf dem Gerät
# **show sessions**	Aktive Telnet/SSH Verbindungen im Hintergrund
Macro show Kommando → multiple Informationen	
# **show tech-support**	zur Kommunikation mit Cisco TAC – Technical Assistant Center: das Macro führt multiple show Kommandos hintereinander aus

show Kommandos filtern und umleiten

Einige show Kommandos bieten zusätzliche Parameter um die Ausgabe zu filtern, d.h. auf die gewünschte Information zu reduzieren.

Ein Beispiel im folgenden Kommando, das die Ausgabe der running-config durch einen zusätzlichen Parameter filtert: es wird lediglich der Konfigurationsabschnitt (section) der angegebenen Schnittstelle innerhalb der running-config ausgegeben – sehr sinnvoll um schnell die Konfiguration einer Schnittstelle zu überprüfen.

show running-config interface *IF-Typ IF-Nr*

Ausgaben von show Kommandos können jedoch auch durch Verwendung des pipe-Zeiches " | " gefiltert oder sogar umgelenkt werden, d.h. als Textdatei auf einem externen Server gesichert werden.

→ Filterung
show *kommando* **|** { **include** | **exclude** | **begin** | **section** } *Zeichenfolge*

include	zeigt alle Zeilen die Zeichenfolge beinhalten
exclude	zeigt alle Zeilen die Zeichenfolge nicht beinhalten
begin	beginnt Ausgabe bei Zeile mit erstem Vorkommen von Zeichenfolge
section	zeigt alle Zeilen die Zeichenfolge beinhalten und alle untergeodneten Zeilen (aus dem ensprechenden SubConfiguration Mode) .. mit Section können Konfigurationssabschnitte dargestellt werden

Beispiele:
show running-config **| include** ip route
show ip interface brief **| exclude** unassigned
show logging **| begin** Log Buffer
show running-config **| section** line vty

→ Umleitung
show *kommando* **|** { **redirect** | **tee** | **append** } *file-system:dateiname*

redirect	leitet die Ausgabe in eine Textdatei z.B. auf einem TFTP Server um
tee	leitet die Ausgabe in eine Textdatei um und zeigt die Ausgabe zusätzclich auf dem Bildschirm an
append	hängt die Ausgabe an eine bestehende Textdatei an (nur Linux Server)

Beispiele:
show tech-support **| redirect** tftp://10.1.1.1/RTA-tech-support
show ip route **| tee** tftp://10.1.1.1/RTA-show-ip-route

debug Kommandos

debug Kommandos zeigen in Echtzeit an, was momentan verarbeitet wird. Diese Anzeige ist im Gegensatz zu show, eine dynamische Anzeige, d.h. jede Veränderung wird sofort angezeigt.

Debug Ausgaben werden von Cisco als Log Meldungen (SYSLOG Log-Level 7) betrachtet.

PROBLEMATIK:
- debug Kommandos sind **CPU intensiv** und können viel Ausgabe erzeugen.
- debug Ausgaben **auf der Console haben höchste Priorität**

Intensives Debugging auf der Console, kann somit dazu führen,
- dass kein Eingabe-Prompt mehr erscheint, dass kein Eingabe mehr möglich ist oder eingegebenen Kommandos nicht mehr ausgeführt werden
- dass das Gerät nicht mehr funktional ist - das Gerät alle anderen Aufgaben, wie z.B. Switching oder Routing vernachlässigt, um die Ausgaben zu erzeugen.

ACHTUNG: Grundsätzlich NIEMALS das Kommando **# debug all** verwenden !
Ein # debug all auf einem Cisco Gerät ist ein erfolgreicher DoS – Denial of Service Angriff.

Empfehlung:
Intensives Debugging auf einer VTY Line durchführen – ACHTUNG: muss aktiviert werden – und die Ausgabe von Debug-Meldungen auf der Console deaktivieren .. siehe auch Kapitel "Logging". Dazu eine Remote Verbindung starten und folgende Kommandos verwenden:

# **terminal monitor**	→ aktiviert Logging/Debugging auf der VTY Line
(config)# **logging console 6**	→ Console Debugging deaktivieren

Sollte das Debugging doch mal auf der Console gestarten worden sein, folgend Workarounds bei intensiven Ausgaben:
- Sollte kein Eingabeprompt mehr erscheinen das Kommando **# u all** einfach **blind eingeben** .. sobald Meldung "All possible debugging has been turned off" erscheint, endet die Ausgabe nach Leerschreiben des Puffers
- Sollte das nicht funktionieren, kann man nur noch dafür sorgen das das debug Kommando keine "Nahrung" mehr bekommt, z.B. Entfernung von Kabelverbindungen.
- Scheint alles verloren :(.. hilft nur noch ein Reboot.

Kommandos zum Debugging

# **debug** *kommando*	startet Debugging für die angegebene Funktion
# **show debugging**	zeigt alle aktiven Debugging Funktionen an
# **no debug** *kommando*	deaktiviert eine aktive Debugging Funktion
# **no debug all**	deaktivert alle aktiven Debugging Funktion
# **undebug all** (# **u all**)	

Beispiel für debug Kommandos:
→ Zeigt Ereignisse der Routing Tabelle (Änderungen)
debug ip routing
.. dann z.B. mal ein Interface auf shutdown und wieder auf no shutdown ..

ping

Das ping Kommando verwendet das ICMP Protokoll um Layer-3 Verbindungen zu testen.
Das ping Kommando ist auf allen Arten von Geräten und Betriebssystemen verfügbar.

Dabei wird eine **ICMP "Echo Request"** (Type 8) Nachricht an ein Zielgerät gesendet, das im Normalfall mit einer **ICMP "Echo Reply"** (Type 0) Nachricht antwortet.

Das ping Kommando steht auf Cisco in 2 Varianten bereit:
Simple ping: ohne zusätzliche Optionen – auch im UserEXEC Mode

> **ping** { *IP* | *IPv6* | *Hostname* }

Extended ping: mit zusätzlichen Optionen (interaktiv oder teilweise über Kommandozeile) – nur im PrivilegEXEC Mode

→ Optionen werden interaktiv abgefragt
ping

Protocol [ip]:	*Layer-3 Protokoll (ip, ipv6)*
Target IP address:	*DST IP Adresse*
Repeat Count [5]:	*Anzahl der Pakete*
Datagram size [100]:	*Größe der Pakete (Bytes)*
Timeout in seconds [2]:	*Timeout*
Extended commands [n]: yes	*.. erweiterte Einstellungen:*
Source address or interface:	*SRC IP Adresse*
Type of service [0]:	*IP Level für QoS setzen*
Set DF bit in IP header? [no]:	*Dont Fragment Bit setzen*
Validate reply data? [no]:	*Prüfsumme überprüfen*
Data pattern [0xABCD]:	*Datenmuster im UDP Segment*
Loose, Strict, Record, Timestamp, Verbose[none]:	*IP und Ausgabeoptionen*
Sweep range of sizes [n]:	*Pakete in untersch. Grössen*

→ Optionen auf der Kommandozeile
ping { *IP* | *IPv6* | *Hostname* } [**repeat** *anzahl*] [**size** *bytes*] [**timeout** *sekunden*]
[**source** *src-IP*] [**df-bit**]

Wichtige Ausgaben des ping Kommandos

Ping Ausgabe	Erläuterung
.	Kein Erhalt eines echo reply vor Ablauf des Timeout
!	Erhalt eines echo reply vor Ablauf des Timeout
N	ICMP Nachricht eines Routers auf dem Weg zur DST IP: Network Unreachable - das Netzsegment existiert in der Routingtabelle nicht
U	ICMP Nachricht eines Routers auf dem Weg zur DST IP: Destination Unreachable - Segment erreichbar, aber Host nicht

Anwendungsbeispiel: **extended ping auf einem Cisco Gerät**
MTU – Maximum Transfer Unit auf einer Verbindung **ermitteln**

# **ping**	
Target IP address: 10.1.1.1	*DST-IP Adresse*
Extended commands [n]: yes	*.. erweiterte Einstellungen:*
Set DF bit in IP header? [no]: yes	*Dont Fragment Bit setzen*
Sweep range of sizes [n]: y	*Pakete in untersch. Grössen*
Sweep min size [36]: 1490	*→ kleinste Grösse*
Sweep max size [18024]: 1510	*→ grösste Grösse*
Sweep interval [1]: 2	*→ Abstand zw. Paketgrössen*

Optionen für das extended ping Kommando:

- Das ping Paket wird an die DST IP 10.1.1.1 gesendet
- Das "Dont Fragment Bit" im IP Header wird gesetzt (Fragmentierung verboten)
- Es werden Pakete in unterschiedlichen Grössen gesendet
  - das erste Paket mit einer min Grösse von 1490 Bytes
  - folgende Pakete dann jeweils 2 Byte grösser
  - das letzte Paket mit einer max Grösse von 1510 Bytes

Da dem Gerät eine Fragmentierung der Pakete verboten wird, können Pakete mit einer Grösse oberhalb der zulässigen MTU nicht mehr gesendet werden.

Anwendungsbeispiel: **ping auf einem PC**
allgemeines Vorgehen zur **schrittweisen Überprüfung der Verbindung** des PCs

1. TCP/IP Protokoll-Software korrekt installiert und aktiv?
   c:\> **ping 127.0.0.1**
   c:\> **ping ::1**

2. NIC – Network Interface Card korrekt installiert, konfiguriert und aktiv?
   c:\> **ping** *eigene-IPv4/IPv6*

3. Lokale, physikalische Verbindung funktional?
   c:\> **ping** *Default-Gateway-IPv4/IPv6*

4. Verbindung in entfernte Netzwerke funktional?
   c:\> **ping** *entfernte-IPv4/IPv6*

traceroute

Mit traceroute können die einzelnen (Router-)Stationen auf dem Weg eines Pakets von der Quelle bis zum Zielnetzwerk am Bildschirm ausgegeben werden.
Es wird überlicherweise verwendet, um Routingprobleme (OSI 3) zu lösen.

Traceroute arbeitet in erster Linie mit **ICMP „Time To Live Exceeded"** (Type 11) Nachrichten.

IP Time to Live (TTL) Funktion

- Wenn eine Station ein Paket sendet, wird im Time To Live Headerfeld eines IP Pakets eine numerischer Startwert eingetragen (z.B.: 128).
- Jeder Router reduziert bei der Weiterleitung eines Paketes, den Wert des entsprechenden TTL Feldes um 1.
- Wenn der Wert des Time To Live Headerfeldes 0 (Null) erreicht hat, verwirft der Router das Paket und sendet eine ICMP "TTL Exceeded" Nachricht an den eigenlichen Sender der Daten zurück, um diesen von der Entsorgung des Pakets in Kenntnis zu setzen.

Arbeitsweise traceroute

- Traceroute generiert nacheinander Pakete mit aufsteigendenden TTL Werten – beginnend von 1 an.
  Dabei können beliebiege Daten übertragen werden. Ciscos traceroute sendet UDP Segemente, Windows tracert sendet ICMP Echo Request Pakete.
- So muß jeder Router auf dem Weg von der Quelle zum Ziel das entsprechende Paket auf 0 (Null) reduzieren, das Paket verwerfen und eine ICMP Nachricht an den Sender zurücksenden.
- Die Absende-Informationen der ICMP TTL exceeded Nachricht (IP Adressen der einzelnen Router) werden dann – ebenfalls nacheinander - am Bildschirm ausgegeben.

Das traceroute Kommando steht auf Cisco in 2 Varianten bereit:
Simple traceroute: ohne zusätzliche Optionen – auch im UserEXEC Mode

> **traceroute** { *IP* | *IPv6* | *Hostname* }

Extended traceroute: mit zusätzlichen Optionen (interaktiv oder über Kommandozeile) – nur im PrivilegEXEC Mode

→ Optionen werden interaktiv abgefragt
traceroute

Protocol [ip]:	*Layer-3 Protokoll*
Target IP address:	*DST IP Adresse*
Source address:	*SRC IP Adresse*
Numeric display [n]:	*Namensauflösung an/aus*
Timeout in seconds [3]:	*Timeout*
Probe count [3]:	*Anzahl der Probes*
Minimum Time to Live [1]:	*Start TTL Wert*
Maximum Time to Live [30]:	*End TTL Wert*
Port Number [33434]:	*DST Port der UDP Nutzlast*
Loose, Strict, Record, Timestamp, Verbose[none]:	*IP und Ausgabeoptionen*

→ Optionen auf der Kommandozeile
traceroute { *IP* | *IPv6* | *Hostname* } [**source** src-IP][**numeric**]
[**timeout** *sekunden*][**probe** anzahl] [**ttl** *min max*] [**port** *port-Nr*]

Telnet – als Troubleshooting Tool

Mit Telnet kann man nicht nur remote Verbindungen zu Geräten realisieren, sondern TCP Verbindungen zu beliebigen Serverdiensten aufbauen, indem man eine ensprechende Ziel-Port Nummer angibt:

telnet { *IP* | *IPv6* | *Hostname* } *dst-Port*

Dadurch kann man - über die Kommandozeile - auch die Erreichbarkeit von OSI 7 Funktionen auf beliebigen Zielgeräten überprüfen.

Bei Aufbau der TCP Session wird eigentlich nur ein Ein-/Ausgabe Kanal zwischen Termial und Serverdienst eingerichtet, d.h. die Ausgaben des Servers erscheinen auf dem Bildschirm und die Eingaben werden, falls für den Server verständliche, korrekte Protkollanweisung, von diesem verarbeitet.

Beispiel für die Überprüfung eines Webservers 10.1.1.1 (HTTP Protokoll, TCP Port 80)

- Nach Eingabe des Telnet Kommandos wird angezeigt, dass die Verbindung offen ist.
- Mit der Eingabe "GET / HTTP/1.0" wird ein HTTP GET Request an den Server gesendet, der die Startseite anfordert – ACHTUNG: der GET Request muss 2x mit ENTER bestätigt werden.
- Nach zweimaliger Besätigung des GET Request, wird die Antwort des Servers (Header und Seiteninhalt) angezeigt und die TCP Verbindung wieder beendet.

telnet fd00::1 80
Trying fd00::1, 80 ... Open
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 30 Dec 2019 19:22:10 GMT
Server: Apache/2.4.7 (Ubuntu)
Last-Modified: Tue, 25 Nov 2014 06:38:01 GMT
ETag: "5c0-508a92716aeb5"
Accept-Ranges: bytes
Content-Length: 1472
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
[.. Inhalt der Dokuments ..]

Selbstkontrolle – Aufgaben und Übungen

1. Nennen Sie 2 Troubleshooting Tools, mit denen OSI 3 Funktionen überprüft werden können.

2. Notieren Sie das notwenidge show Kommando, um alle Schnittstellen in tabellarischer Form am Bildschirm auszugeben, auf denen das "shutdown" Kommando nicht gesetzt ist.

3. Notieren Sie das notwendige show Kommando, um zu ermitteln, wann ein Cisco Router zum letzten Mal neu gebootet wurde.

4. Notieren Sie das – kürzeste – Kommando, um alle aktiven debug Ausgaben zu stoppen.

5. Notieren Sie 2 unterschiedliche Kommandos, um die Konfiguration einer Schnittstelle innerhalb der running-config zu überprüfen. Dabei soll nur die Konfiguration der Schnittstelle ausgegeben werden.

6. Welche ICMP Nachrichten werden von traceroute verarbeitet?

Selbstkontrolle – Lösungen

1. Nennen Sie 2 Troubleshooting Tools, mit denen OSI 3 Funktionen überprüft werden können.

ping, traceroute

2. Notieren Sie das notwenidge show Kommando, um alle Schnittstellen in tabellarischer Form am Bildschirm auszugeben, auf denen das "shutdown" Kommando nicht gesetzt ist.

 # show ip interface brief | exclude Adminstrativly

3. Notieren Sie das notwendige show Kommando, um zu ermitteln, wann ein Cisco Router zum letzten Mal neu gebootet wurde.

 # show version

4. Notieren Sie das – kürzeste – Kommando, um alle aktiven debug Ausgaben zu stoppen.

 # u all

5. Notieren Sie 2 unterschiedliche Kommandos, um die Konfiguration einer Schnittstelle innerhalb der running-config zu überprüfen. Dabei soll nur die Konfiguration der Schnittstelle ausgegeben werden.

 # show running-config interface IF-Typ IF-Nr
 # show running-config | section IF-Typ IF-Nr

6. Welche ICMP Nachrichten werden von traceroute verarbeitet?

ICMP Type 11 "TTL exceeded"