

Zusammenfassung: Jahr 2

Inhaltsverzeichnis

1	Differenzierungskurs: Cisco	1
1.1	Grundlagen	1
1.2	Lab 2.1.4.7 - Establishing a Console Session	1
1.2.1	Connect to Cisco switch using rollover console cable	1
1.2.2	Display and Configure Basic Device Settings	1
1.3	Lab 2.3.3.3 - Building a Simple Network	1
1.3.1	Set Up the Network Topology (Ethernet only)	2
1.3.2	Configure PC Hosts	2
1.3.3	Configure and Verify Basic Switch Settings	2
1.4	Lab 2.3.3.4 - Configuring a Switch Management Address	3
1.4.1	Configure a Basic Network Device	3
1.4.2	Verify and Test Network Connectivity	3
1.5	Gesammelte Befehle	4

1 Differenzierungskurs: Cisco

CCNA Routing and Switching

1.1 Grundlagen

1.2 Lab 2.1.4.7 - Establishing a Console Session

1.2.1 Connect to Cisco switch using rollover console cable

Bei Cisco werden die DB9-RJ45-Kabel zur Verbindung mit einem seriellen Interface als *rollover console cable* bezeichnet. Da kaum ein Notebook noch einen DB9-Anschluss besitzt, sollte ein DB9-zu-USB-Adapter verwendet werden. Die Verbindung wird dadurch erleichtert, beispielsweise muss unter Linux die Baud-Rate nicht händisch angepasst werden, da dies der Adapter erledigt. Ohne den Adapter müsste auch eingestellt werden, ob 8- oder 7-Bit verwendet werden.

	baud rate:	9600
	data:	8 bit
Cisco Defaults:	parity:	none
	stop:	1 bit
	flow control:	none

Tera Term ist ein Terminal Emulator für Windows. Im Prinzip macht das Programm das, was unter Linux mit einem Terminal und dem Programm 'screen' gemacht werden kann. Ich verwende *screen* unter Linux und werde nicht auf die Verwendung von Tera Term eingehen; kurz gesagt funktioniert Tera Term so: „Tera Term starten, Defaults korrekt?, Ok klicken, ..., Profit.“ Verbindung unter Linux per ...

```
> sudo screen /dev/ttyUSB0
```

1.2.2 Display and Configure Basic Device Settings

Nach dem Start des Switches wird gefragt, ob die initiale Konfigurationsdialog gestartet werden soll. Dies verneint man an dieser Stelle. Um die Uhrzeit anzupassen, muss man quasi root werden. Dazu wird unter IOS der Befehl *enable* verwendet.

```
# clock set ?
hh:mm:ss Current Time
# clock set 17:00:00 ?
<1-31> Day of the month
MONTH Month of the year
# clock set 17:00:00 Oct 26 ?
<1993-2035> year
# clock set 17:00:00 Oct 26 2012
# show clock
17:00:03.205 UTC Fri Oct 26 2012
```

An dieser Stelle fällt auf, dass die Hilfestellung, die durch ? angezeigt wird, sich immer auf die jeweilige Option an der Stelle des Fragezeichens bezieht und nicht wie unter Linux die Option *-help* die gesamte Syntax anzeigt.

1.3 Lab 2.3.3.3 - Building a Simple Network

Netzwerke bestehen in der Regel aus drei Komponenten: Hosts, Switchen und Routern. In diesem Lab werden zwei Switches und zwei Hosts so konfiguriert, dass sie sich zu einem einfachen Netzwerk verbinden.

1.3.1 Set Up the Network Topology (Ethernet only)

Dieser Abschnitt des Labs beschreibt, wie die Verkabelung geschehen soll und welche Reaktionen die Switches zeigen sollen. Im Grunde soll dabei die folgende Verbindung hergestellt werden:

```
host1 <---> switch1 <---> switch2 <---> host2
```

Die LEDs des Switches am jeweiligen Port, wo die Hosts eingesteckt sind, sollte grün leuchten.

1.3.2 Configure PC Hosts

In diesem Abschnitt wird beschrieben, wie unter Windows eine statische IP vergeben wird. Unter Linux wird eine statisch IP wie folgt vergeben:

```
Beispiel: # ip address add 192.168.1.123/24 dev eth0
```

Der Befehl lässt sich noch auf *ip a a ...* reduzieren. Unter Windows wäre es notwendig sich durch die Systemsteuerung und die Netzwerk-Optionen hin zu den Interfaces zu klicken, wo dann über die Eigenschaften von IPv4 die IP und Subnetzmaske (255.255.255.0 = /24) eingestellt werden muss. Unter Windows könnte man nun in *cmd* per *ipconfig* die Einstellung von eth0 prüfen. Mit Linux funktioniert es sehr ähnlich:

```
> ip r oder  
> ifconfig
```

Anschließend sollte versucht werden, den gegenüberliegenden Host zu pingen:

```
> ping <ip.host>
```

Warum mussten wir an dieser Stelle die Switches nicht verbinden? Das passiert bei Cisco automagisch.

1.3.3 Configure and Verify Basic Switch Settings

Als erstes muss sich wie in Lab 2.1.4.7 beschrieben per Konsole mit dem Switch verbunden werden und der root-Modus per *enable* aktiviert werden.

```
# configure terminal  
(config) # hostname switch1  
(config) # no ip domain-lookup  
(config) # enable secret <some.password>  
(config) # line con 0  
(config-line) # password <another.password>  
(config-line) # login  
(config-line) # exit
```

Als kleines Gimmick lässt sich ein Banner setzen, der angezeigt wird, wenn man sich per Telnet mit dem Switch verbindet. <EOF-char> kann an dieser Stelle jedes Symbol sein, welches nicht im Banner vorkommt; meist #.

```
(config) # banner motd <EOF-char>  
$BANNER_MESSAGE <EOF-char>  
(config) # exit  
# copy running-config startup-config  
# show running-config  
# show version
```

Bei einem Switch ist es wichtig, den Zustand der Interfaces zu kennen. Mit dem Befehl *show ip interface brief* (beachte: *interface* und **nicht** *interfaces*!) lassen sich einige Informationen über die Interfaces ermitteln. Zu diesen Informationen gehören: Geschwindigkeit des Interfaces, IP-Adresse

und Status. Um den Switch wieder in den Ausgangszustand zu versetzen, müssen die Änderungen rückgängig gemacht werden:

```
# show flash
# delete vlan.dat
# erase startup-config
# reload
```

1.4 Lab 2.3.3.4 - Configuring a Switch Management Address

Cisco Switches besitzen ein spezielles Interface, das sogenannte switch virtual interface (SVI). Diesem kann eine IP zugewiesen werden und anschließend kann man sich per 'telnet' auf das Interface verbinden.

1.4.1 Configure a Basic Network Device

Als erstes muss sich wie in Lab 2.1.4.7 beschrieben per Konsole mit dem Switch verbunden werden und der root-Modus per *enable* aktiviert werden. Anschließend werden wie in Lab 2.3.3.3 Abschnitt 3 (Configure and Verify Basic Switch Settings) beschrieben Passwörter gesetzt. Danach wird ein VLAN und die IP des SVIs gesetzt.

```
> enable
Can I haz password?: <some.password>
# config t
(config) # interface vlan 1
(config-if) # ip address <svi.ip> <subnet>
(config-if) # no shut
(config-if) # exit
(config) # line con 0
(config-line) # password <another.password>
(config-line) # login
(config-line) # exit
(config) # line vty 0 4
(config-line) # password <another.password>
(config-line) # login
(config-line) # end
```

1.4.2 Verify and Test Network Connectivity

Auf dem Switch sollte die Konfiguration überprüft werden:

```
> enable
Can I haz password?: <some.password>
# show run
# show ip interface brief
```

Vom PC aus sollte zunächst gecheckt werden, ob sich die IP des SVIs anpingen lässt. Anschließend kann man sich per *telnet* mit dem Switch verbinden.

```
> ping <svi.ip>
> telnet <svi.ip> 23
Can I haz password?: <another.password>
```

Auf dem Switch lässt sich dann wie in Lab 2.3.3.3 die Konfiguration mit *copy* speichern.

1.5 Gesammelte Befehle

Nicht alle Befehle lassen sich in jedem Modus verwenden. Welche Befehle zu welchem Modus gehören, wird in diesem Abschnitt nicht behandelt. An dieser Stellen werden lediglich die Befehle aufgelistet. Für Informationen zu den jeweiligen Modi sollen die obigen Abschnitte zu Rate gezogen werden.

```
banner motd <EOF-char>
clock set ?                # ? ~ —help
clock set hh:mm:ss month year
configure terminal        # enters global config-mode
config t                  # t = terminal
copy running-config <filename> # speichert die config nach filename
delete <vlan>             # bsp: vlan.dat
enable                    # enter root-mode
enable secret <password>  # set password
erase <config>            # bsp: statup-config
hostname $HOSTNAME
line con 0                # wechselt in console-config-mode
line vty 0 4              # config vty; 0-4: 5 sessions moegl
login                     # aktiviert passwort-abfrage
interface vlan 1          # config vlan 1
ip address <ip> <subnet>  # set ip and subnet
no ip domain-lookup       # disable dns
show clock
show flash
show ip interface brief
show version
show running-config       # zeigt aktuelle config
show run                  # run = running-config
```