Basic Cisco IOS Commands for Router and Switch Management

Josef Kaderka

Version 46

Inspiration Boson

The commands are given in their basic form, without context (i.e., the current mode), and knowledge of it or cultivation of Cisco IOS intuition is assumed. For example, the command to assign an IP address to interface ip address {adr} {sm} is given. However, to enter it, you must first enter privileged mode (enable command), then global configuration mode (configure terminal command), and then specific configuration mode (interface {int} command).

The Cisco IOS operating system comes in several versions for a single device. Not all versions (especially older versions) support all of the commands listed here. Pre-XR IOS has been used.

Router Management

Configuration modes - meaning of the prompt	
User EXEC mode	**Router >**
Privileged EXEC mode	**Router #**
Global configuration mode	**Router (config) #**
Specific configuration mode - interface configuration - Logical interface configuration - routing configuration - link configuration (CON, AUX)	**Router(config-if)#** **Router (config-subif) #** **Router(config-router) #** **Router (config-line) #**

Basic router operations	
Enter privileged EXEC mode	**enable**
Return to user EXEC mode	**disable**
Logging out of the router	**exit**, **logoff**
Restart the router operating system (unsaved changes lost)	**reload**
Previous command	**<Arrow Up>** or **<Ctrl><p>**
Next command	**< Arrow Down>** or **<Ctrl><n>**
Move one character to the right	**< Arrow Right>** or **<Ctrl><f>**
Move one character to the left	**< Arrow Left>** or **<Ctrl><b>**
Break operation	**<Shift><Ctrl><6><x>**
Refresh display content (without inserting a command)	**<Ctrl><L>**
Auto-completion of command and parameters	**<Tab>**
Help (always context-sensitive)	**<?>** or **help**
Just enough characters to make the command unambiguous	**sh run** instead of **show running-config**
Number of console lines per page (on very dumb terminal)	**terminal length {n}**

Discovering router information	
IOS version, memory sizes and configuration register value	**show version**
Show running configuration (stored in RAM)	**show running-config**
Show saved configuration (from NVRAM, flash)	**show startup-config**
Processor usage	**show processes cpu**
Flash memory contents, free, occupied and total space	**show flash:**
Flash memory contents	**dir flash:**
Summary of the status of all interfaces (their system designations, IP addresses, physical and link layer status), can be shortened as follows: **sh ip int br**	**show ip interface brief**

Router Configuration	
Deleting a saved configuration file	**erase startup-config**
Restart (if prompted, do not save anything!)	**reload**
Switch to global configuration mode	**configure terminal**
The router will be named Brno	**hostname Brno**
Go back one level of configuration	**exit**
Return from any level to basic EXEC mode	**end, Ctrl-z**
Copying from tftp server to RAM	**copy tftp running-config**
From hard memory (NVRAM) to RAM; use only if no configuration has already been made - a mixture could be created	**copy startup-config running-config**
From hard memory (NVRAM) to RAM; the current configuration in RAM will be overwritten	**configure replace nvram:startup-config**
From tftp server to flash memory	**copy tftp flash**
From flash memory to tftp server	**copy flash tftp**
Save the current configuration in RAM to NVRAM	**copy running-config startup-config**
Save the current configuration in RAM to NVRAM - an older alternative	**write**
Exact specification of the IOS (file containing it) to be booted from flash memory (use if there are multiple IOSes in flash)	**boot system flash {*filename*}**
Exact specification of the IOS (file containing it) to be booted from the tftp server (IP address will be requested)	**boot system tftp {*filename*}**

Create a local user and assign a password	**username {user} password {password}**
Create a local user account with administrator rights	**username {*user*} privilege 15 password {password}**
Creating a local user and assigning a password; this will be saved after processing by the chosen algorithm	**username {user} algorithm-type {md5|scrypt|sha256} secret {password}**

Passwords, remote access	
The minimum password length will be 8 characters	**security passwords min-length 8**
Setting a "class" password for console access	**line console 0** **password class** **login**
Set "class" password for remote access (telnet), up to 5 users at the same time (virtual terminals 0 to 4)	**line vty 0 4** **password class** **login**
Number of minutes until automatic logout (0 - never)	**exec-timeout {n}**
Setting the password "cisco" to enter privileged mode	**enable password cisco**
Hashing of the password "cisco" to enter privileged mode by the selected algorithm	**enable algorithm-type {md5|scrypt|sha256} cisco**
Encryption of all passwords (with weak algorithm nr. 7)	**service password-encryption**

Remote access using ssh (scp)	
Need to change the default device name (Router, Switch)	**hostname Brno**
Set domain name (any)	**ip domain-name skoleni.org**
Generate asymmetric keys	**crypto key generate rsa**
The ssh protocol version 2 will be used	**ip ssh version 2**
Create a local user	**username {*user*} password {*password*}**
Access to the virtual terminal using ssh only (privileged mode password must be already set!)	**line vty 0 4** **transport input ssh**
Activate the scp server (secure copy)	**ip scp server enable**

Basic serial interface configuration	
Configure interface (numbers indicate the position of the module in device architecture)	**show controller serial 0/1/0**
Configure interface (numbers indicate the "position" of the module)	**interface serial 0/1/0**
Set clock rate on serial DCE inerface	**clock rate 64000**
Only for path cost computing [kb/s], has no clock rate meaning!	**bandwidth 64**
Interface activation	**no shutdown**
Verification of the interface status	**show interface serial 0/1/0**

Create a virtual interface (loopback) and configure its IP address	
Creating a loopback interface with the some number (0 here)	**interface loopback 0**
Assign IP address to loopback interface 0	**ip address 10.0.0.1 255.255.255.255**

Cisco Discovery Protocol (CDP) – proprietary	
Start CDP (runs by default, sends multicast frame every 60 s, dead interval 180 s)	**cdp run**
Overview of direct neighbour Cisco devices (name, local interface identifier, properties, type, remote interface identifier)	**show cdp neighbors**
Additionally, operating system, IP address, and hardware details	**show cdp neighbors detail**
CDP shutdown	**no cdp run**

Link Layer Discovery Protocol – standard IEEE (uses EtherType 0x88CC)	
LLDP startup (frames multicast 30 s, dead interval 120 s)	**lldp run**
Disable transmitting of LLDP frames to the specified interface Disable receiving of LLDP frames from a specified interface	**interface gigabitethernet 0/0** **no lldp transmit** **no lldp receive**
LLDP status	**show lldp**
Overview of directly connected devices (name, local interface	**show lldp neighbors**

identifier, properties, remote interface identifier)	
Additionally, details about the operating system, VLAN, IP address of neigbour, hardware, etc.	**show lldp neighbors detail**
Disable LLDP	**no lldp run**

TCP/IP	
Disable IPv4 routing (enabled by default)	**no ip routing**
Enable IPv6 routing (disabled by default!)	**ipv6 unicast routing**
Setting IP addresses on interfaces and enabling them	**interface serial 0/1/0** **ip address 157.89.1.3 255.255.0.0** **no shutdown** **interface fastethernet 0/0** **ip address 208.1.1.4 255.255.255.0** **no shutdown**

Static routing	
Static routing entry - destination network, mask, our egress interface	**ip route 160.216.0.0 255.255.0.0 Fastethernet 0/0**
Static routing entry - destination network, mask, neighbour router (157.89.10.1)	**ip route 160.216.0.0 255.255.0.0 157.89.10.1**
Static routing entry for the default path (default router/gateway - 157.89.10.1)	**ip route 0.0.0.0 0.0.0.0 157.89.10.1**

Dynamic Routing – RIP, RIPv2	
RIP version 2 routing protocol configuration (default v1) Network addresses 157.89.0.0 and 208.1.1.0 will be advertised	**router rip** **version 2** **network 157.89.0.0** **network 208.1.1.0**
Propagation of a locally defined static route into routing protocol	**redistribute static**
Authentication (RIP v2 only) - local name of a password (key) Local key number Password itself - shared between neighboring routers Enable authentication (set on adjacent interfaces) Same using MD5	**key chain KLIC1** **key 1** **key-string heslo1234** **ip rip authentication key-chain KLIC1** **ip rip authentication mode md5**

Dynamic Routing - EIGRP	
EIGRP Routing Protocol Configuration, Autonomous System 1, do not aggregate subnet address (required if there are multiple subnets of the same network separated by other networks) Network addresses 157.89.0.0 and 208.1.1.0 will be advertised	**router eigrp 1** **network 157.89.0.0** **network 208.1.1.0** **no auto-summary**
EIGRP authentication - local name of password (key) Local key number Password itself Enable authentication (set on adjacent interfaces) Password specification	**key chain MYCHAIN** **key 1** **key-string heslo1234** **ip authentication mode eigrp 10 md5** **ip authentication key-chain eigrp 10 MYCHAIN**

Dynamic Routing – OSPFv2 - IPv4	
OSPFv2 (IPv4) routing protocol configuration, this instance of the OSPFv2 process has a locally valid number of 1, area 0 Network addresses 157.89.0.0 and 208.1.1.0 will be advertised	**router ospf 1** **network 157.89.0.0 0.0.255.255 area 0** **network 208.1.1.0 0.0.0.255 area 0**
No OSPF information will be sent through Fastethernet 0/0, but the address of this network will be propagated into OSPF	**passive-interface fastethernet 0/0**
Redistributing of statically set default path by OSPF	**default-information originate**
Let the path cost (metric) is 47; it is set on the interface	**ip ospf cost 47**
Summation of 8 Class C network addresses originating from OSPF area 19 (on ABR, ASBR only); the summarized data is sent to area 0	**area 19 range 192.168.0.0 255.255.248.0**
Authentication – password to be set on adjacent interfaces Authentication - all router interfaces within area 0 Password is sent as a clear text in bot cases	**ip ospf authentication-key heslo1234** **router ospf 1** **area 0 authentication**
OSPF neighbor authentication using MD5, set on adjacent interface(s)	**ip ospf message-digest-key 1 md5 cisco12345** **ip ospf authentication message-digest**
OSPF neighbour authentication using SHA – set password name Set password identifier (e.g. number) Password itself Hashing algorithm will be SHA256 The password will be used on the fastethernet 0/1 interface Password has been set	**key chain JMENO** **key KEY-ID** **key-string cisco12345** **cryptographic-algorithm hmac-sha-256** **interface fastethernet 0/1** **ip ospf authentication key-chain JMENO**

Dynamic Routing - OSPFv3 - IPv6 (traditional configuration)	
IPv6 packet routing must be explicitly enabled	**ipv6 unicast routing**
Traditional OSPFv3 routing protocol configuration The OSPFv3 router ID must be always specified explicitly IPv6 address setting; in addition the interface will automatically get an additional link-local address Redistribution of static paths and default paths as with OSPFv2	**ipv6 router ospf 1** **router-id 6.6.6.6.** **interface gigabitethernet 0/0** **ipv6 address 2001:DB8:CAFE:1::1/64** **ipv6 ospf 1 area 0**

Dynamic Routing - IPv4 and IPv6 - OSPFv3 (new configuration style)	
Common IPv4 and IPv6 router configuration In addition to the specified IPv6 address, the interface will get an link-local address generated (can also be entered manually) Static paths default paths redistribution as in OSPFv2 The same for passive interface setting	**router ospfv3 1** **..address-family ipv4 unicast** **router-id 1.1.1.1** **address-family ipv6 unicast** **router-id 6.6.6.6** **interface gigabitethernet 0/0** **ip address 192.168.1.1 255.255.255.0** **ipv6 address 2001:DB8:CAFE:1::1/64** **ospfv3 1 ipv4 area 0** **ospfv3 1 ipv6 area 0**

Routing - listing, debugging	
Show IPv4 routing table	**show ip route**
Information about all running routing processes	**show ip protocols**
Basic debug of data exchanged by RIP	**debug ip rip**
Debug EIGRP exchanged data	**debug ip eigrp events** **debug ip eigrp transactions**
Debug of OSPF exchanged events Show OSPF neighbours; show status of adjacency Show OSPF configuration summary Detailed OSPF parameters on the gi0/0 interface Summary details of OSPF parameters on all interfaces	**debug ip ospf events** **show ip ospf neighbor** **show ip ospf** **show ip ospf interface gi0/0** **show ip ospf interface brief**

Access Control Lists (ACLs) – selection	
Meaning of Access Control Lists (ACL) numeric ranges	
IP standard access list (only source IP address is matched)	**<l-99>**
IP extended access list (protocol, source and destination IP addresses, source and destination ports, TCP ACK flag)	**<100-199>**
Appletalk access list	**<600-699>**

48-bit MAC address access list	**<700-799>**
IPX standard access list	**<800-899>**
Extended 48-bit MAC address access list	**<1100-1199>**
IPX summary address access list	**<1200-1299>**
IP standard access list (expanded range)	**<1300-1999>**
Which ACLs are assigned to a given interface?	**show ip interface serial 0/1/0**
Show all ACLs Show only IPv4 ACLs	**show access-lists** **show ip access-list**

Numbered standard access lists (l-99), filter only by source IP address (i.e. sender)	
Purpose - to prevent nodes on subnet 200.1.1.0 255.255.255.0 from sending packets over interface Fastethernet 0/0	
Disable the source subnet	**access-list 1 deny 200.1.1.0 0.0.0.255**
Explicitly allow all other networks – implicit setting is "deny any"	**access-list 1 permit any**
Assign an ACL to the appropriate interface and direction	**interface fastethernet 0/0** **ip access-group 1 in**

Numbered extended access list (l00-199), filter protocol, source and destination IP addresses, ports, etc.	
Purpose - not to allow machine 1.1.1.1 to telnet over interface fa0/0 to machine 2.2.2.2 and not to allow any surfing for subnet 3.3.3.0/24 users	
Syntax: access-list {number} deny|permit protocol source_host destination_host port	**access-list 100 deny tcp host 1.1.1.1 host 2.2.2.2 eq 23**
Disable surfing (http) to users from network 3.3.3.0/24	**access-list 100 deny tcp 3.3.3.0 0.0.0.255 any eq 80**
Explicitly allow other traffic (implicitly "deny any").	**access-list 100 permit ip any any**
Assign an ACL to the appropriate interface and direction	**interface fastethernet 0/0** **ip access-group 100 out**

Named ACL (keywords standard, extended)	
Advantage: better editing possibility, even a single line of a multi-line ACL can be edited instead of having to delete the entire ACL and recreate it as in case of numbered ACLs	**ip access-list standard COOLLIST** **deny 1.1.1.1** **permit any**
Assign an ACL to the appropriate interface and direction	**interface fastethernet 0/0** **ip access-group COOLLIST in**

PPP (mostly obsolete, for information)	
Commands on **router_a,** mirrored on **router_b**, link between **router-a** and **router-b** using serial interfaces,	
We need to create user "router-b", shared password is"cisco"	**username router-b password cisco**
Frame encapsulation is PPP (deafult is cHDLC))	**encapsulation ppp**
Authentication will be via the chap protocol	**ppp authentication chap**
Determination of encapsulation type, activated link layer protocols (LCP), etc.	**show interface serial 0/1/0**
Debug authentication process	**debug ppp authentication**

PPP multilink (aggregation of several physical serial interfaces into a single logical one)	
Create logical interface number 1 and configure it	**interface multilink 0** **ip address 1.1.1.2 255.255.255.0** **ppp multilink** **ppp multilink group 1**
Configure all physical interfaces by the same way and associate them with multilink number 1	**interface serial 0/1/0** **no ip address** **encapsulation ppp** **ppp multilink** **ppp multilink group 1**

Frame-Relay (for information - obsolete protocol)	
Enabling Frame-Relay on a given interface and specifying the encapsulation type	**encapsulation frame-relay ietf**
LMI type specification (detected automatically by IOS since version 11.2)	**frame-relay lmi-type ansi**
If reverse ARP will not work, map the remote IP address to our	**frame-relay map ip 3.3.3.100 broadcast**

(local) DLCI number	
It is also possible to enable broadcasting and specify the encapsulation type	
Define local DLCI (if LMI is not working)	**frame-relay local-dlci 100**
Set the period for verifying of connection	**keepalive 10**
Show LCI and LMI informati	**show interface serial 0**
Show statistics on PVC operation	**show frame-relay pvc**
VýpisShow routing map (static or dynamic)	**show frame-relay map**
Show LMI information	**show frame-relay lmi**
Converting a router to a Frame Relay switch role (for lab purposes)	
Note – appropriate commands must be entered on both DCE interfaces that are connected by Frame Relay	
Enable Frame-Relay switching (on the DCE side)	**frame-relay switching**
Tell the DCE side to support the frame-relay on that interface	**frame-relay intf-type dce**
Tell the DCE side on which other local interface {int_o} and DLCI {dlci_o} to switch DLCI {dlci_i} from the currently configured interface	**frame-relay route {dlci_i} interface {int_o} {dlci_o}**
Set the clock rate [b/s] on the DCE interface	**clock rate 64000**

Router as DNS server	
IP address of the real name server (up to six)	**ip name-server 169.223.2.2**
Own domain name	**ip domain-name skoleni.org**
Router will serve as a name server (cache type)	**ip dns server**
Do not translate names to IP addresses (on local router)	**no ip domain-name lookup**

Router as DHCP server	
Explicit activation of DHCP server (some IOSes)	**service dhcp**
Do not assign IP addresses from these range	**ip dhcp excluded-address 157.89.1.1 157.89.1.2**
Pool naming and definition of parameters sent to clients (max 124 addresses, domain name, IP addresses of default router, DNS and netbios servers, lease for 2 days).	**ip dhcp pool MOJE_ZASOBARNA** **network 157.89.1.0 255.255.255.128** **domain-name unob.cz** **default-router 192.168.12.1** **dns-server 192.168.12.100 192.168.12.101** **netbios-name-server 192.168.12.99** **lease 2**
IP address of remote DHCP server (can not be reached by broadcast). Set on the router interface connecting DHCP client network.	**ip helper-address 169.223.2.2**
The router interface obtains the IP address from the DHCP server	**interface fa0/0** **ip address dhcp**
To whom the IP address has been assigned	**show ip dhcp bindings**

NAT (PAT)	
Internal (private) network interface	**interface FastEthernet0** **ip nat inside**
External (usually public) network interface	**interface FastEthernet1** **ip nat outside**
Trigger – ACL that governs, which traffic will be translated (other traffic passes without translation!); any in this case :-)	**access-list 10 permit any**
The entire internal network will be hidden behind a single public IP address (done by **overload**), set on **FastEthernet1**, because ACL 10 will be used.	**ip nat inside source list 10 interface FastEthernet1 overload**

Configuration register (16 bits)	
RXBOOT (special diagnostic mode, continue with "b")	**confreg 0x2000**
Boot system from ROM, load configuration file (when flash upgrade - for routers that boot IOS from flash)	**confreg 0x2101**
Boot from ROM, do not load configuration file (disaster recovery)	**confreg 0x2141**
Boot from flash, load configuration file (normal state)	**confreg 0x2102**
Boot from flash, do not load configuration file (for password recovery)	**confreg 0x2142**

Router password recovery – (only from console)	
1. Break IOS boot	**<Ctrl><Break>**
2. Boot IOS from flash, do not load configuration file from NVRAM	**confreg 0x2142**
2a. Different syntax valid only for old devices	**o/r 0x2142**
3. Reboot the IOS	**reset**
4. Go into privileged mode; because no configuration file loaded, no passwords applied	**enable**
5. Copy the configuration file from NVRAM to RAM - the router comes alive including unknown passwords, but remains in privileged mode	**copy startup-config running-config**
6. Overwrite the unknown enable password to "NoveHeslo"	**enable password NoveHeslo**
7. Save the configuration to NVRAM (i.e. with the new password)	**copy running-config startup-config**
8. Next start of the router should be normal (IOS from flash, configuration file from NVRAM)	**config-reg 0x2102**

Restore missing IOS operating system (Ethernet interface on router must be present)	
The IOS must be backed up in advance - it cannot be freely downloaded. In an emergency, the same IOS from another router of the same series can be sometimes used. If the IOS is deleted from the flash, but the router is still running, do not switch it off (!), but proceed as standard - **copy tftp flash** (you must have tftp server with backup IOS). For routers with removable storage (Compact Flash, USB flash), the IOS can be written to it on an external device (PC) and loaded or copied into internal flash.	
Connect the Ethernet interface with the lowest ID (e.g. fa0/0) Verify the settings of the listed variables (see example). If necessary, then set (change) the variables similarly as shown in this listing	**rommon 1 > set** **IP_ADDRESS=172.18.16.76** **IP_SUBNET_MASK=255.255.255.192** **DEFAULT_GATEWAY=172.18.16.65** **TFTP_SERVER=172.18.16.2** **TFTP_FILE=c2600-ik9o3s3-mz.123-13.bin**
Example of setting/changing the value of a variable	**TFTP_SERVER=160.216.1.3**
Start IOS download and installation	**tftpdnld**
Reboot the router	**reset**

Restore missing IOS operating system (for non-Ethernet routers only)	
If an Ethernet interface is not available, a low-speed console port can be used to install IOS.	
Connect the serial port of the PC to the console port of the router. Use a terminal program on the PC that supports the Xmodem protocol (Hyperterminal, TeraTerm, modified putty).	
Increase baud rate to maximum according to the router type (0x3822 = 115.2 kb/s, 0x2102 = 9.6 kb/s), set the same on the terminal emulator. Restart the router	**rommon 1 > confreg 0x3822** **rommon 2 > reset**
Start the IOS installation, wait for the end of the transfer (about 30 minutes for IOS 15 MB and 115.2 kb/s, about 4.5 hours at 9.6 kb/s!) (For recovery it is advisable to use as small IOS as possible, e.g. old, boot router and then install the target version from it - already over the network)	**rommon 1 > xmodem c2600-ik9o3s3-mz.123-13.bin**
Set the default value of the configuration registry	**config-register to 0x2102**
Restart the router, return the terminal emulator to 9600 b/s!	**reset**

Precise time - NTP	
This is the source of the exact time: **tik.cesnet.cz**	**ntp server tik.cesnet.cz**
The time zone should be named CET, the offset from UTC is +1 hour	**clock timezone CET 1**

Event log – syslog protocol	
This is the syslog server, this is where the messages will be written	**logging 172.16.1.1**
The message will have the facility **local5**	**logging facility local5**
Send messages of type (priority) debugging and higher	**logging trap debugging**

Network Management - SNMP	
Setting the password "admins" for reading and writing SNMP data	**snmp-server community admins rw** **snmp-server community topsecret rw 60**

Setting the password "topsecret" to read and write SNMP data only from 10.1.1.1	**access-list 60 permit 10.1.1.1**
Setting the password "others" for reading SNMP data (common value is "public")	**snmp-server community others ro**
This is the router master	**snmp-server contact Josef Kaderka**
This is where the router is located	**snmp-server location Brno, Sumavska 4, 3/11a**
SNMP manager, there to send messages (traps) with community public	**snmp-server host 10.1.1.1 public**
Enable to send messages when any event occurs	**snmp-server enable traps**
Send messages only when an event of a given type occurs	**snmp-server enable traps config** **snmp-server enable traps envmon temperature**

Resilient IOS and configuration file	
IOS resilience	**secure boot-image**
Creating a resilient copy of the startup configuration file	**secure boot-config**
Verify IOS resilience status	**show secure bootset**
Restore deleted configuration file (two steps), reflected in the running configuration	**secure boot-config restore flash:archived-config** **configure replace flash:archived-config**
Erasing the resilient copy of the startup configuration file.	**no secure boot-config**
Undo IOS resilient feature	**no secure boot-image**
Update the resilient copy of the startup configuration file, if necessary	**no secure boot-config** **secure boot-config**

Switch management

(Basic operations are the same as for routers)

Determining the status of the switch	
IOS version, hardware, etc. (configuration register differs from routers)	**show version**
Show saved configuration (from hard memory - NVRAM)	**show startup-config**
Show current configuration (from RAM)	**show running-config**
Show flash memory contents	**show flash:** nebo **dir flash:**
Show interface security settings (many variations)	**show port-security**
Show status of all interfaces (many variations)	**show interfaces**
Show interface capabilities and their current settings	**show interfaces fa0/1 capabilities**

Setting the switch to the default state	
Prevent the switch from communicating with neighboring switches by blocking the interface (or discconnect cable)	**interface fastethernet 0/1** **shutdown**
Deleting the stored VLAN database	**delete flash:vlan.dat**
Deleting a saved configuration file	**erase startup-config**
Restart (if prompted, do not save anything)	**reload**

Basic switch operations	
Configure the IP address that allows remote access to the switch. Always block all previously set VLAN interfaces first, then enable the desired one.	**interface VLAN1** **shutdown** **interface VLAN99** **ip address 192.168.1.2 255.255.255.0** **ip default-gateway 192.168.1.1** **no shutdown**
Show the switches table of known MAC addresses	**show mac-address-table**
Show number of MAC addresses in the table (useful when overflow is suspected)	**show mac-address-table count**
Clear MAC address table	**clear mac-address-table**

Enabling IPv6 support	
Only for some switches (according to IOS), subsequent restart required	**sdm prefer dual-ipv4-and-ipv6 default**

Configure the interface for the end device connection	
Interface selection	**interface gigabit 0/1**
Multiple interfaces selection (can be a list)	**interface range fastethernet 0/1–12**
Full duplex option (if not specified, duplex mode will be negotiated)	**duplex full**
100 Mbps speed selection (if not specified, the speed will be negotiated)	**speed 100**
Only the station will be connected to the interface	**switchport mode access**
The interface goes up immediately when the device is connected, no waiting for STP	**spanning-tree portfast**

Switch interface security	
Setting the access interface mode (not trunk)	**switchport mode access**
Enable security on the interface (otherwise other commands will not work)	**switchport port-security**
Only stations with a given MAC address can communicate over the interface	**switchport port-security mac-address {adr}**
No more than **{n}** stations can communicate over the interface	**switchport port-security maximum {n}**
After **{n}** minutes of inactivity, the heard address will be discarded	**switchport port-security aging time {n}**
The interface learns the MAC addresses of devices and writes them to the running configuration so they can be saved to the startup configuration	**switchport port-security mac-address sticky**
Unauthorized communications will be discarded, authorized ones will not	**switchport port-security violation protect**
Same, plus a log entry is made, ev. SNMP trap is sent	**switchport port-security violation restrict**
Interface will be blocked, manual intervention required (default setting)	**switchport port-security violation shutdown**
Automatically unblock the interface after a certain time:	**errdisable recovery cause psecure-violation**

	errdisable recovery interval 60

DHCP Snooping	
Global enabling of DHCP Snooping	**ip dhcp snooping**
Enable DHCP Snooping only in VLAN 10	**ip dhcp snooping vlan 10**
DHCP packets can pass through this interface without restriction	**interface f0/1** **ip dhcp snooping trust**
DHCP packets can pass through these interfaces, but up to a maximum of 5 per second	**interface f0/18** **ip dhcp snooping limit rate 5**
Did DHCP Snooping catch anything?	**show ip dhcp snooping binding**

Protocol Spanning Tree (STP)	
Getting the MAC address of our switch	**show interface vlan 1**
Listing the spanning tree table and finding out who is the root switch	**show spanning-tree**
Set root switch by setting the priority {n} (lowest one wins)	**spanning-tree priority {*n*}**

Spanning Tree Protocol (STP) Security	
If the interface receives a BPDU packet, it will be blocked.	**spanning-tree bpduguard enable**
Unblocking an interface blocked in this way (option 1)	**errdisable recovery cause psecure_violation**
Unblocking an interface blocked in this way (option 2)	**disable** **enable**
Option - fast activation of an interface that has no switch behind it (so no need to wait for STP convergence)	**switchport mode access** **spanning-tree portfast**

Remote management via web interface	
Disable http (access even without password is enabled by default; if set, password is used to enter privileged mode)	**no ip http server**
Enable https	**ip http secure-server**
Create a local user account with administrator privileges and enable local authentication	**username {*user*} privilege 15 password {*password*}** **ip http authentication local**

Password recovery (verified on 29xx/35xx/36xx switches)	
1. Turn off the power to the switch (pull out power cord)	
2. Press and hold the "Mode" button on the front panel of the switch	**<mode>**
3. Turn on the switch power and wait, STAT LED will blink fast	
4. Release the "Mode" button when the STAT LED turns off (or blink amber/green)	
5. Wait for the end of boot, until ROMMON prompt (**switch:**) appears	
6. Enter a following sequence of commands (depending on the switch; not always both).	**flash_init** **load_helper**
7. Rename the configuration file (**config.text**, stored in flash) to	**rename flash:config.text flash:config.old**
8. Boot operating system	**boot**
9. Skip the configuration dialog, go into privileged mode	**enable**
10. Restore the configuration file name	**rename flash:config.old flash:config.text**
11. Load the saved configuration, i.e. with the old password	**copy startup-config running-config**
12. Overwrite the unknown enable password to “class”	**enable secret class**
13. Save the current configuration, i.e. with the new password	**copy running-config startup-config**

Missing IOS operating system recovery (procedure for 29xx/35xx/36xx switches)	
The IOS must be backed up in advance (tftp server) - it cannot be freely downloaded. In an emergency, the same IOS from another switch of the same series can be used. If the IOS is deleted from the flash but the switch is still running, do not shut it down (!), but follow the standard procedure - copy tftp flash (i.e. start tftp server, prepare backup IOS). If the IOS is deleted, it must be installed from a backup on the local PC via the console port using a terminal emulator with X-modem protocol support (Hyperterminal, Tera Term; modified putty) - beware, the transfer takes tens of minutes. For recovery it is advisable to use as small IOS as possible, e.g. old one, boot router from it and then install the target version already over the network	
Check the state of the flash memory (the prompt is in the form **switch:**), especially if there is enough space.	**flash_init** **dir flash:**

Set the highest possible console port speed (here 115 200 b/s).	**set BAUD 115200**
Set the same speed in the terminal emulator (Tera Term), otherwise communication with the switch will not be possible.	**Setup->Serial port->Speed->115200**
Activate the X-modem protocol in the switch in receive mode, using the backup IOS file name.	**copy xmodem: flash: c2960-lanbasek9-mz.150-2.SE8.bin**
In the terminal emulator, start sending the IOS file from the given directory.	**File->Transfer->XMODEM->Send**
Return the baud rate on the switch to the default state.	**unset BAUD**
Return the baud rate in the terminal emulator to 9600 bps.	**Setup->Serial port->Speed->9600**

Virtual LAN (VLAN) and trunking	
Creating VLAN number 20 and name it "KITCHEN".	**vlan 20** **name KUCHYNE**
Assign the interface to VLAN number 20. If it did not exist before, it will be created, explicitly unnamed VLANs will be named VLANxxxx, where xxxx is its number (with leading zeros); can be changed	**interface fastethernet 0/1** **switchport mode access** **switchport access vlan 20**
List of virtual LANs and the interfaces assigned to them	**show vlan**
If the IOS supports two types of encapsulation (standard 802.1q or historical Cisco proprietary ISL), select the desired	**interface fastethernet0/2** **switchport trunk encapsulation dot1q** **switchport mode trunk**
Explicit trunk creation with native (untagged) VLAN 5	**switchport trunk native vlan 5**
Put untagged frames in VLAN 5 (by default they go to VLAN 1)	**switchport trunk allowed vlan 5,10,20**

Virtual LAN (VLAN) and trunking on older switches	
Create VLAN number 20 and name it "KITCHEN"	**vlan database** **vlan 20 name KUCHYNE**
Assign the interface to VLAN20	**interface ethernet 0/1** **vlan static 20**
List of virtual LANs and the interfaces assigned to them	**show vlan-membership**
Choosing encapsulation (ISL or 802.1q; only if IOS supports both) and creating a trunk	**interface fastethernet0/2** **switchport trunk encapsulation isl** **switchport mode trunk**

Routing between virtual LANs (router on a stick method)	
There is a single physical link between the switch and the router, configured as a trunk on the switch side, and a logical interface (subinterface) is created for each VLAN on the router side.	
Configuring the physical interface of the router	**interface fastethernet 0/0** **no shutdown**
Create a logical interface (any number, preferably the same as the VLAN)	**interface fastethernet 0/0.20**
Select the encapsulation and specify the VLAN number	**encapsulation dot1q 20**
Assign an IP address to the logical interface	**ip address 192.168.5.20 255.255.255.0**

Aggregation of several interfaces into a single interface with cumulative speed (Etherchannel, LACP)	
Select interfaces (all must be configured the same; i.e., in trunk or access mode) and selecting a group number, proprietary PAgP protocol	**interface range FastEthernet0/1 - 4** **channel-group 1 mode on**
Select interface (all must be configured the same; i.e., in trunk or access mode) and group number selection, IEEE LACP protocol	**interface range FastEthernet0/1 - 4** **channel-group 1 mode auto**
Status verification	**show etherchannel 1 summary**

Monitoring traffic on one or more interfaces or VLANs with another interface (SPAN - Switched Port Analyzer)	
Traffic sources selection (all interfaces must be configured the same)	**monitor session 1 source interface FastEthernet0/1** **monitor session 1 source interface FastEthernet0/2**
This is where the traffic will be monitored	**monitor session 1 destination interface gigabitEthernet0/1**
Status verification	**show monitor session 1**

Remote Switched Port Analyzer (RSPAN) Remote monitoring of traffic on one or more interfaces or VLANs on another switch interface	
Create a VLAN for the transfer of monitored data in	**vlan 30**

the monitored and monitoring switchech (trunk must be created between them)	**name RSPAN-VLAN** **remote-span**
Monitored switch - select the data source (physical interface) and copy it to the specified VLAN	**monitor session 1 source interface Gi0/1 rx** **monitor session 1 destination remote vlan 30**
Monitoring switch - selecting the data source (VLAN) and copying it to the specified physical interface	**monitor session 1 source remote vlan 30** **monitor session 1 destination interface Gi0/2**
Status verification	**show monitor session 1**

Virtual private network between two routers - variant - IPSec tunnel	
Establishing ISAKMP protocol policy 10 - phase 1 Encryption will be done using the AES algorithm A shared password will be used Diffie-Hellman group 14 (2048 bits) Set shared password and the IP address of the other side of the tunnel	**crypto isakmp policy 10** **encryption aes** **authentication pre-share** **group 14** **crypto isakmp key heslo1234 address 192.168.23.3**
Specification of acceptable cryptographic protocol combinations (otherwise known as "IPSec proposals") - Phase 2	**crypto ipsec transform-set MOJE esp-des esp-sha-hmac**
Creating an IPSec policy (crypto map)	**crypto map MOJEMAPA 10 ipsec-isakmp** **set peer 192.168.23.3** **set transform-set MOJE** **match address 101**
This traffic will go through the tunnel	**access-list 101 permit ip 172.16.1.0 0.0.0.255 172.16.3.0 0.0.0.255**
Applying the cryptomap to the interface	**interface FastEthernet0/0** **ip address 192.168.12.1 255.255.255.0** **crypto map MOJEMAPA**
Verification of tunnel status - Phase 1	**show crypto isakmp sa** **show crypto ipsec sa**