Orientace v systému, konfigurace a zabezpečení switche

Nejdřív průzkum

Jak rychle zjistit, ze kterého obrazu se bootovalo:

```
show boot
```

Podrobnější informace:

```
show version
```

Které soubory jsou v různých pamětech:

```
show flash          … flash PROM, je tam obraz systému a základní konfigurace
dir flash:          … totéž, případně jenom „dir“, nebo může fungovat dir flash0:
dir nvram:          … NVRAM, je tam startup-config
dir usbflash0:      … pokud je připojena USB flash (případně „dir ?“, když tápeme)
```

Práce se soubory:

Mažeme pomocí `delete`, `erase`

Zobrazujeme obsah pomocí `more`, například:

```
more flash:config.text
```

Funguje pro flash a usbflash, pro soubory v RAM (running-config) a v NVRAM (startup-config) používáme příkaz `show`.

Pokud se chceme vrátit k některému příkazu, můžeme buď používat šipky, nebo si zobrazit historii použitých příkazů:

```
show history
```

Uživatelé:

```
show users   … uživatelé, má smysl v případě používání lokální databáze
show login   … informace o přihlášených uživatelích a parametrech
```

Informace o syslogu:

```
show logging
```

Fyzikální parametry (teplota, větráčky, napájení apod.) – nefunguje v Packet Traceru:

```
show env all        … zobrazí se vše, co se dá monitorovat na komponentách
show env fan        … k větráčkům
show env power all  … vše k napájení (k jednomu či dvěma zdrojům, jaké tam jsou)
show env ?          … seznam možností, vybereme si, co potřebujeme
show int g0/1 status
show interfaces status    … případně další klíčová slova:flowcontrol,…
show int g0/1 capabilities      … vlastnosti portu, nefunguje v Packet Traceru
show interfaces accounting      … statistika pro různé protokoly na rozhraních
```

Filtrování výstupu

U show příkazů lze filtrovat výstupy pomocí „roury“ (nefunguje na příkaz `more` a jiné).

Jak zjistit, které filtry se dají použít:

```
show run | ?
```

(symbol roury je třeba obklopit mezerami).

```
show run | begin line con 0          … od tohoto řádku dále (stačí začátek)
show run | section line vty          … sekce (části) takto začínající
show run | section spanning-tree
sh ip int br | include up            … řádky obsahující dané slovo (zde „up“)
sh ip int br | exclude down          … řádky neobsahující dané slovo
show run | include service           … řádky z running-config o službách
```

Tabulka MAC adres

Tím je myšlena dynamicky doplňovaná přepínací tabulka (také CAM = Content-Addresable Memory).

Vytvořte si jednoduchou síť se dvěma switchi, ke každému jeden nebo dva počítače, každému počítači přiřaďte IP adresu z rozsahu 10.0.0.0/8. Počítače mezi sebou „pingněte“, aby se naplnily tabulky switchů.

Zobrazení obsahu:

```
sh mac address-table            NEBO:
sh mac-address-table
```

Smazání tabulky:

```
clear mac address-table         NEBO:
clear mac-address-table
clear mac-address-table dynamic
```

To poslední je praktické tehdy, pokud jsou v tabulce i statické záznamy, budeme mazat jen ty dynamicky naučené.

Konfigurace parametrů portu

Zobrazíme si parametry konkrétního portu, zajímají nás (kromě jiného tyto informace: jestli je spoj aktivní (1), jaká je MAC adresa portu (2), v jakém duplexu a jaké rychlosti port funguje (3):

```
Switch#sh int g0/1
GigabitEthernet0/1 is up, line protocol is up (connected)
  Hardware is Lance, address is 00d0.bc48.a019 (bia 00d0.bc48.a019)
 BW 1000000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s
  input flow-control is off, output flow-control is off
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:08, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
```


Pokud není v pořádku duplex či rychlost:

```
int g0/1
      duplex full
      speed 1000
```

(pro plný duplex s rychlostí 1 Gb/s, ovšem pokud to port umí).

Pro zjištění chyby MDIX (rozpoznávání křížení) – nefunguje v Packet Traceru:

```
show controllers ethernet g0/1 phy
show controllers ethernet g0/1 phy | include auto-mdix
```

Pokud řádek vypadá takto:

```
Auto-MDIX : On [AdminState=1 …]
```

- AdminState=1 znamená, že je tato funkce zapnuta, =0 by znamenalo, že je vypnutá
- On znamená, že port na této straně linky provádí křížení, Off znamená, že neprovádí křížení

Pokud je funkce Auto-MDIX vypnutá, na portu ji zapneme takto:

```
int g0/1
      mdix auto
```

Port Security

Účelem je zajistit, aby v případě, že se k portu připojí někdo „nepovolaný“ (podle MAC adresy), došlo k vhodné bezpečnostní reakci – podle toho, co nastavíme.

Port musí být v přístupovém módu a musí mít zapnutou funkci port security, pak můžeme tuto funkci nastavovat. Takže nejdřív (pro port g0/1):

```
switchport mode access
switchport port-security
```

Můžeme určit maximum pro počet bezpečných MAC adres (pro 2 adresy):

```
switchport port-security maximum 2
```

Jsou tři možnosti, jak určit bezpečné MAC adresy:

- staticky, tedy ručně určíme bezpečnou MAC adresu (uloží se do running-config, po restartu zmizí, ale u portu je uložena, tedy při vypnutí/zapnutí portu se neztratí),
- dynamicky, první připojená MAC se zapamatuje (uloží se do RAM, ale ne do running-config, po restartu zmizí, při vypnutí/zapnutí portu taktéž a učí se znovu),
- sticky port-security: učí se jako dynamicky (uloží se do running-config, po restartu zmizí, ale při vypnutí/zapnutí portu se neztratí) – něco mezi statickým a dynamickým určením MAC.

Pokud máme MAC adresu naučenou v running-configu (staticky nebo sticky), pak je vhodné uložit nastavení do startup-config.

Dynamické a statické určení MAC se dá kombinovat.

Příklad:

Statické určení MAC adresy pro port:

```
int f0/1
      switchport mode access
      switchport port-security
      switchport port-security maximum 5
      switchport port-security mac-address 0001.42a1.5ebd
```

Učení typu sticky:

```
int f0/1
      switchport mode access
      switchport port-security
      switchport port-security maximum 2
      switchport port-security mac-address sticky
```

Reakce při připojení MAC adresy, která není bezpečná:

- protect: provoz od „cizí“ MAC adresy je zahazován, jinak se nic neděje
- restrict: „cizí“ provoz je zahazován, zvýší se counter sledující počet narušení bezpečnosti, je vyvoláno hlášení o narušení bezpečnosti (syslog)
- shutdown: port je vypnut, dostane se do stavu err-disabled, musí se ručně vypnout+zapnout (tj. použít příkazy shutdown a noshutdown); zvýší se counter (defaultní stav)

Používáme příkaz `switchport port-security violation ….`

Příklad:

Pokud tedy chceme nastavovat MAC adresu stylem sticky a chceme, aby port fungoval v režimu restrict, zadáme:

```
int f0/1
       switchport mode access
       switchport port-security
       switchport port-security maximum 2
       switchport port-security mac-address sticky
       switchport port-security violation restrict
```

Pro jiné módy bychom zadali klíčová slova protect nebo shutdown (ale shutdown je výchozí, to není nutno zadávat).

Ověření:

```
show port-security int f0/1
show run | section interface
show port-security
show port-security address
show port-security int f0/1
```