Cisco IOS 1 - úvod, příkaz show

Vložil/a Samuraj [1], 13 Srpen, 2007 - 15:49

- Networks & Protocols [2]

Firma Cisco je v oblasti aktivních prvků známa asi každému. Velká část jejich zařízení je vybavena jednotným operačním systémem zvaným IOS (Internetwork Operating System), který nabízí široké možnosti konfigurace a správy. V této první části série článků o IOSu se věnuji obecnému popisu IOSu, jeho používání a velmi důležitému příkazu show, který slouží k zobrazení informací.

Co je to IOS a jak pracuje

IOS je zkratka pro **Cisco's Internetwork Operating System**, což je operační systém, který používá většina Switchů a Routerů firmy Cisco (dále budu mluvit o switchích, ale obecně je vždy možno uvažovat i o routerech). Celý IOS je uložen v jednom image souboru s příponou bin a v aktuálních switchích má velikost kolem 5 MB. Ve flash paměti switche může být ještě uloženo webové rozhraní, které bývá ve složce HTML.

IOS je propracovaný a na míru provedený systém. Nabízí velké množství možností pro konfiguraci, a pokud víme, co chceme konfigurovat, není to již příliš složité. Obsluha IOSu je založena na **CLI – Command Line Interface**, tedy na příkazové řádce.

Pro správné používání/konfiguraci switche je nutné se orientovat v různých pamětech, které se používají a vědět jak s nimi switch pracuje. Typy pamětí jsou popsány v další kapitole. Důležitý je také postup, jakým probíhá startování switche:

* zapojíme napájení do switche
* spustí se mikrokód z ROM paměti
o pokud nenalezne validní verzi IOSu, nastartuje do omezené funkce RxBoot
* nalezne IOS image (standardně ve Flash paměti) a zavede jej
* načte se a aplikuje startup-config z NVRAM (tento textový soubor obsahuje sekvenci příkazů IOSu, která se vykonává stejně, jako při zadávání přes CLI)
o pokud není nalezen, tak se vyvolá express setup
* pokud existuje, tak se z Flash paměti načte vlan.dat

Pokud zadáváme nějaké konfigurační příkazy IOSu, tak ty se okamžitě provádějí, ale ukládají se pouze do **running-config**, který je uložen v RAM (při startu se do něj kopíruje obsah startup-config). To znamená, že aktuálně jsou platné, ale po restartování switche se vymažou. Pokud bychom tedy provedli nějakou konfiguraci, kterou bychom nemohli vrátit zpět, stačí restartovat switch a ten je v takovém stavu, jako při posledním uložení konfigurace. Pokud však chceme zachovat naše změny, je třeba vždy překopírovat běžící konfiguraci do startovací.

Pozn.: Pro zdůraznění. Všechny změny, o které nechceme přijít při restartu switche, můsíme uložit! To se provede překopírováním běžící konfigurace do startovací.

```
SWITCH#copy running-config startup-config  // uložit
Destination filename [startup-config]?     // dotaz na jméno, stiskn?te ENTER
Building configuration...
[OK]
```

Typy paměti

Cisco switche a routery používají pět různých pamětí (většinou se uvádí pouze čtyři, ale já myslím, že i ten poslední je důležitý).

Paměť ROM

ROM paměť je možno pouze číst a je nezávislá na napájení. Obsahuje procesy, které se provádí při bootování (startování) switche. Ve starých verzích switchů obsahovala i IOS. V ROM paměti je uloženo několik základních funkcí:

* **POST (Power-on Self Test)** - po zapnutí switche provede tento mikrokód test funkčnosti hlavních částí (paměť, CPU, interfacy)
* **Bootstrap Program** - inicializace bootování, nahrává IOS
* **ROM monitor** - speciální diagnostický mód pro řešení problémů
* **RxBoot** - pokud se nenalezne funkční IOS, tak se načte tato omezená verze IOSu, která umožní instalovat správný IOS image

Paměť Flash

Flash je paměť typu NVRAM (Non-volatile random access memory), do které je možno zapisovat a při odpojení napájení zůstane obsah zachován. Ve flash paměti je primárně uložen IOS (může zde být i více verzí), dále kopie startup-config v souboru config.text a informace o VLANech v vlan.dat. Teoreticky zde můžeme uložit cokoliv. Výpis paměti se provede pomocí dir flash: nebo show flash:.

Paměť NVRAM

Non-volatile random access memory (NVRAM) je obdobně jako Flash zapisovatelná paměť nezávislá na napájení. Je použita pro uložení startup-config. Výpis paměti se provede pomocí dir nvram:.

Paměť RAM

Random Access Memory (RAM) je operační paměť. Jedná se o rychlou zapisovatelnou paměť, která je ovšem závislá na napájení a při restartu switche se vymaže. Standardně je tato paměť rozdělena na dvě části - hlavní paměť procesoru a sdílenou paměť I/O. V hlavní části je uložena běžící konfigurace (running-config), běžící IOS, routovací a ARP tabulky. Sdílená paměť je použita jako buffer pro uložení aktuálně zpracovávaných paketů. Zobrazit její obsah můžeme pomocí show memory.

Externí paměť - TFTP

Poslední typ paměti není součástí zařízení, takže by se zde nemusel počítat, ale využívá se pro řadu činností. Používá se zde TFTP (Trivial FTP) server, který můžeme rozběhnout na libovolném počítači. Upozorňuji, že to není FTP, ale TFTP. Nejčastěji jej využijeme při provádění zálohy či upgradu IOSu, ale je možné i zavádět IOS přímo z TFTP serveru.

Základní práce s IOSem

Nejprve je třeba se připojit ke switchi, abychom mohli pracovat v CLI. Můžeme tak učinit přes consoli, tzn. sériovým kabelem přes speciální port na switchi. To je základní připojení a někdy jediné možné. Nebo pokud již máme provedenu základní konfiguraci, přes telnet či ssh. Aby fungovala tato připojení, musí být nastavena a musí být vždy nastaveno přihlašovací heslo. Pro některé úkony, třeba i základní konfiguraci, se můžeme připojit přes webové rohranní či užitečný program Cisco Network Assistant. Ale ač to tak na první pohled nemusí vypadat, nakonec je jednodušší provádět většinu nastavení přes CLI. Pro připojení přes consoli použijeme hyperterminál nebo putty, který poslouží i pro ssh a telnet.

CC-BY-SA Security-Portal.cz | secured by paranoid sense | we hack to learn

Zadávání příkazů

Veškeré příkazy IOSu se dají zadávat zkráceně, stačí zadat první znaky, které jednoznačně určí příkaz (tedy, aby v daném kontextu neexistoval jiný příkaz, začínající těmito znaky).

Klávesa tabulátor doplňuje příkaz. Zadáme prvních pár písmen příkazu a po stisknutí TAB se příkaz doplní, pokud je jednoznačný, nebo se doplní část, která je pro více příkazů společná.

Zadáním **?** (otazníku) se zobrazí seznam příkazů s krátkým popiskem, které můžeme na aktuálním místě použít. Také můžeme zadat prvních pár písmen příkazu a otazník, aby se vypsal seznam příkazů s tímto začátkem. Většina příkazů se skládá z posloupnosti klíčových slov, pokud zadáme příkaz ?, dostaneme seznam argumentů či klíčových slov, která se dají zadat na tomto místě.

```
SWITCH#?
Exec commands:
             Session number to resume
  access-enable    Create a temporary Access-List entry
  access-template  Create a temporary Access-List entry
  archive          manage archive files
 --More--

SWITCH#co?
configure  connect  copy
```

Klávesa šipka nahoru a šipka dolu slouží k procházení historie, prochází dříve zadané příkazy.

Pokud se na obrazovku vypisují údaje, listují se po stránkách. Při vypsání stránky se výpis zastaví. Klávesou SPACE se zobrazí další stránka, klávesou ENTER se zobrazí další řádek.

U příkazů, které generují nějaký informační výstup, můžeme použít výstupní modifikátor | (svislá čára) a pomocí něj omezovat výstup.

```
- Zobrazení MAC adres na portu 16
SWITCH#show mac-address-table | include Fa0/16
```

Pokud jsme v CLI na nějakém switchi, tak se můžeme připojit k jinému switchi zadáním jednoho z příkazů v uživatelském/privilegovaném módu. ssh adresa, telnet adresa, connect adresa či pouze zadáním samotné adresy, pokud neodpovídá nějakému klíčovému slovu. Adresa je buď IP adresa nebo hostname. Pro přepnutí zpět bez ukončení aktuální session slouží Ctrl+Shift+6 a potom x.

Rušení příkazů

Pokud zadáme nějaký příkaz IOSu a odešleme jej pomocí klávesy ENTER, tak se okamžitě uplatní a uloží do běžící konfigurace. Pokud chceme zrušit nějaké nastavení, tedy odvolat dříve zadaný příkaz, použijeme klíčové slovo no a za ním stejný příkaz, jako pro zadání. Tedy všechna nastavení se dají opět rušit pomocí příkazu no. Často není třeba zadávat všechny parametry příkazu. Takto provedené změny se opět uplatní pouze na běžící konfiguraci.

```
SWITCH(config-if)#no ip address // zrušení IP adresy pro interface
```

Příkazové módy

Uživatelské rozhranní IOSu je děleno do řady různých módů, které nám umožňují provádět jiné činnosti. Hlavní módy jsou

* uživatelský mód (EXEC) – user EXEC – SWITCH>
v tomto módu jsme hned po přihlášení a má pouze omezené příkazy
* privilegovaný mód (EXEC) – privileged EXEC - SWITCH#
výchozí mód pro přestup do dalších konfigurací, umožňuje zobrazovat různé údaje
* globální konfigurační mód – global configuration - SWITCH(config)#

CC-BY-SA Security-Portal.cz | secured by paranoid sense | we hack to learn

zde se konfigurují funkce, které ovlivní celý systém
* konfigurace interfacu – interface configuration - SWITCH(config-if)#
v tomto módu konfigurujeme vlastnosti určitého interfacu

Podobných módů jako konfigurace interfacu je celá řada. Módy jsou od sebe odlišeny různým promptem.

```
SWITCH>                                 // uživatelský mód
SWITCH>enable                           // p?epnutí  do privilegovaného modu
SWITCH#configure  terminal              // p?epnutí do globální konfigurace
SWITCH(config)#interface {jméno}       // p?epnutí do  konfigurace interfacu
SWITCH(config-if)#                      // konfigurace interfacu
```

Pro přestup z vyšší urovně do nižší slouží příkaz exit, pokud chceme přejít z několika nadřazených úrovní do privilegovaného módu, použijeme Ctrl+Z.

```
SWITCH(config-if)#exit
SWITCH(config)#
```

Základní kontrolní (informační) příkazy - show

Tyto příkazy slouží k výpisu nastavení a informací o aktuální položce. Většina se zadává v privilegovaném módu pomocí příkazu show, který zobrazuje informace z běžící konfigurace.

Obecné příkazy

```
- Výpis obsahu flash pam?ti (dva  shodné p?íkazy)
SWITCH#show flash:
SWITCH#dir

- Informace o switchi a verzi IOSu
SWITCH#show version

- Vypsání b?žící konfigurace
SWITCH#show running-config

- Vypsání startovací konfigurace
SWITCH#show startup-config

- Informace o loggování a poslední záznamy
SWITCH#show logging

- Seznam naposled zadaných p?íkaz?
SWITCH#show history

- Zobrazení informací o okolních Cisco  switchích pomocí protokolu CDP
SWITCH#show cdp neighbors
SWITCH#show cdp neighbors detail

- Informace o využití procesoru a b?žících procesech
SWITCH#show processes
```

Připojení

```
- Informace o aktuálních telnetových  spojeních
SWITCH#show sessions
```

```
- Informace o aktuálních ssh spojeních
SWITCH#show ssh

- Informace o p?ihlášených uživatelích
SWITCH#show users

- Informace o linkách
SWITCH#show line
```

Interface

```
- Podrobný výpis informací o interface
SWITCH#show interfaces

- Podrobný výpis informací o jednom  interface
SWITCH#show interfaces {jméno-interfacu}
SWITCH#show interfaces f0/1
```

Interface – rozhranní jsou hlavně fyzické porty na switchi a VLANy. Porty se adresují typem, dnes hlavně fastEthernet (stačí f) a gigabitEthernet (stačí g), a číslem portu (může jít o kombinaci čísel, kde je například pořadí ve stacku, ale poslední je vždy číslo portu).

```
- Informace o portu  s podrobnostmi o VLAN
SWITCH#show interfaces {jméno-interfacu} switchport

- Vypsání seznam fyzických interfac?  s popisem a stavem v p?ehledné form?
SWITCH#show interfaces status

- Stru?ný seznam všech interfac? se  statistikou p?enos?
SWITCH#show interfaces summary

- Seznam statistik pro interfacy
SWITCH#show interfaces stats

- Stru?né informace o VLANech a  p?i?azení port?
SWITCH#show vlan

- Informace o existujících truncích
SWITCH#show interfaces trunk

- Zobrazení CAM tabulky – MAC adresy a  porty komunikujících za?ízení
SWITCH#show mac-address-table

- Zobrazení ARP tabulky
SWITCH#show arp

- Zobrazení informací o ACL a routování na interface
SWITCH#show ip interface

- Zobrazení sm?rovací tabulky
SWITCH#show ip route
```

Samuraj, http://www.samuraj-cz.com/ [3]

URL článku:

https://www.security-portal.cz/clanky/cisco-ios-1-%C3%BAvod-p%C5%99%C3%ADkaz-show

Odkazy:
[1] https://www.security-portal.cz/users/samuraj
[2] https://www.security-portal.cz/category/tagy/networks-protocols
[3] http://www.samuraj-cz.com/

CC-BY-SA Security-Portal.cz | secured by paranoid sense | we hack to learn