Bezpečnostní funkce směrovače Cisco 2800
Role-Based CLI Access

Petr Vyletělek

2.7.2007

Obsah

1. Co je Role-Based CLI Access

Role-Based CLI Access je určeno pro zvýšení bezpečnosti při správě síťových prvků.

Tato technologie umožňuje administrátorovi pro daný síťový prvek definovat různé uživatelské „pohledy“(views). Přičemž pohled je podmnožina všech dostupných příkazů daného prvku, ať už z privilegovaného nebo konfiguračního režimu.

Pohledy umožňují přesně určit, které příkazy jsou použitelné a které konfigurační údaje se mohou zobrazit, čímž dávají administrátorovi možnost kontrolovat přístup dalších uživatelů k jednotlivým funkcím síťového prvku.

2. Základní informace

2.1. Výhody použití

Prvky Cisco umožňují spravovat přístup k příkazové řádce(CLI) pomocí různých úrovní privilegovaných režimů. Tento postup však nemusí ve spoustě případů poskytovat potřebnou míru detailu, neboť nastavení povolených či zakázaných funkcí je pro každý privilegovaný režim přednastaveno a není možné je jakkoli měnit.

To je právě důvod použití Role-Based CLI Access, jež umožňuje mnohem detailnější a přesnější popis uživatelských práv pro práci s příkazovou řádkou. Každému uživateli povolí použití pouze definovaných příkazů a zbývající příkazy zůstanou nedostupné. Nedostupné příkazy nejen, že není možné použít, ale nezobrazují se ani při výpisu konfigurace ani při výpisu všech použitelných příkazů, což opět slouží k větší bezpečnosti, neboť uživatel přihlášený v daném pohledu nemá možnost jakkoli zjistit, zda vůbec a jaké konkrétní příkazy má zakázány.

2.2. Typy pohledů

2.2.1. Root View

Root view je předdefinovaný v každém IOSu a není možné jej odebrat ani jakkoli měnit jemu dostupné příkazy.

Uživatel přihlášený do Root view má v podstatě neomezená práva pro správu daného prvku stejně jako uživatel přihlášený do privilegovaného režimu s úrovní 15. Rozdíl je jen v možnosti spravovat další pohledy(views).

Pouze uživatel přihlášený do Root view může přidávat, editovat a odebírat pohledy. Toto právo nemá žádný privilegovaný režim a není ani možné toto právo přidat do žádného jiného pohledu.

Přihlášení do Root view se dá provést v uživatelském nebo privilegovaném režimu zadáním příkazu bez parametrů:

```
enable view
```

2.2.2. CLI View

CLI view je základním typem pohledu. Umožňuje používání pouze nadefinovaných příkazů, všechny ostatní příkazy jsou nedostupné. Nedostupné příkazy pak nejen nelze použít, ale není možné je ani nijak zobrazit, nezobrazují se ani v příkazové nápovědě(tlačítko `TAB` a znak `?`) ani ve výpisu konfigurace.

Před konfigurací musí mít nastaveno heslo, které však při přihlašování nemusí být požadováno, toto je závislé od konkrétní konfigurace AAA subsystému.

Standardně má každý pohled povoleny jen následující tři příkazy(všechny ostatní jsou zakázány):

```
exit
enable
show parser view
```

Jedinými vždy přístupným příkazy,které není možné odebrat, jsou příkaz pro opuštění pohledu(**`exit`**) a pro změnu pohledu nebo privilegovaného režimu(**`enable`**).

Příkaz pro zobrazení aktuálního pohledu (**`show parser view`**) je možné z kteréhokoliv pohledu odebrat.

Žádné CLI view nejsou předdefinovány.

2.2.3. Superview

Superview je speciální typ pohledu, který sdružuje různé již definované CLI pohledy dohromady a vytváří tak z nich pohled nový.

Vlastnosti superview:

- každý superview musí obsahovat nejméně jeden CLI pohled
- každý CLI pohled může být ve více superviews
- příkazy nemohou být přidávány přímo do superview
- uživatelé přihlášení do daného superview mohou využívat všechny příkazy povolené alespoň v jednom CLI pohledu patřícího do tohoto superview
- každý superview má přiděleno heslo
- při smazání superview zůstanou všechny CLI pohledy, které obsahoval, zachovány
- CLI pohledy mohou být přidávány do superview až poté, co bude mít nastaveno heslo

2.3. Omezení

Role-Based CLI Access je implementováno jako standardní součást Cisco IOSů až od verze 12.3(7)T. Pro práci s Role-Based CLI Access je tedy potřeba, aby na síťovém prvku byl nainstalován Cisco IOS s verzí 12.3(7)T nebo vyšší.

CLI pohledů a superviews dohromady může být maximálně 15, přičemž root view se do tohoto počtu nezapočítává.

3. Práce s pohledy

3.1. Vytvoření pohledu

Aby vůbec mohlo být Role-Based CLI Access nakonfigurováno, je nezbytně nutné mít povolenou funkci AAA, jež je ve výchozím nastavení zakázána.

Dále je nutné, aby při konfiguraci pohledů byl administrátor přihlášen pod root view.

3.1.1. Authentication, Authorization and Accounting (AAA)

AAA slouží pro definování způsobu ověřování uživatelů. Tato komponenta disponuje rozsáhlými konfiguračními možnostmi a nabízí mnoho různých variant konfigurace jako např. přihlášení bez hesla, přihlášení jen heslem, přihlášení pomocí uživatelského jména a hesla, přičemž pro ověření mohu být použita data z lokální nebo vzdálené databáze. Podrobnosti viz. [2].

Pro inicializaci AAA slouží příkaz:

```
aaa new-model
```

Pro využití lokální databáze(definováno příkazem **username**) při ověřování uživatelů je potřeba zadat příkaz:

```
aaa authentication login default local
```

3.1.2. Konfigurace

Nový pohled se vytvoří v konfiguračním režimu zadáním příkazu:

```
parser view view-name [superview]
```

- **superview** specifikuje, že nově vytvářený pohled je typu superview

Prvním krokem při definování nového pohledu musí být přiřazení hesla. Dokud nemá pohled nastaveno heslo, není možné provádět žádnou jinou konfiguraci. Pro přiřazení hesla pohledu slouží příkaz:

```
secret password
```

Hlavní část konfigurace pak spočívá v definování jednotlivých příkazů dostupných v daném pohledu a provádí se pomocí příkazu:

```
commands parser-mode {include | include-exclusive | exclude}
[all] [command]
```

- *parser-mode* určuje, v jakém módu se daný příkaz nachází (např. pro privilegovaný režim **exec**, pro konfigurační režim **configure**, pro režim konfigurace rozhraní **interface**)
- dále je nutné určit, v jakém vztahu je zadaný příkaz k danému pohledu. Existují tři možnosti:

include	přidá příkaz do pohledu, aniž by nějak ovlivnil ostatní pohledy příkaz nesmí být v žádném jiném pohledu ve vztahu **include-exclusive**

include-exclusive	přidá příkaz do pohledu a zabrání jeho přidání do jakéhokoliv jiného pohledu příkaz nesmí být v žádném jiném pohledu ve vztahu **include** ani **include-exclusive**
exclude	vyřadí příkaz z pohledu, aniž by nějak ovlivnil ostatní pohledy

- **all** zahrne všechny příkazy začínající stejnými znaky
- *command* příkaz, pro který specifikujeme vztah k pohledu

Include a **exclude** se navzájem neruší. Pokud je nějaký příkaz ve vztahu **include**, není možné tento vztah zrušit zadáním **exclude.** Systém takový postup nepřipouští a vypíše informační hlášení, že daný příkaz již má specifikovaný vztah ke konfigurovanému pohledu. Pro korektní zrušení vztahu **include** je tedy potřeba zadat příkaz:

```
no include …
… konkrétní argumenty rušeného příkazu
```

Obdobné je to i v případě vztahu **exclude**.

Zadání neexistujícího příkazu systém bohužel vždy nehlásí a tak je vhodné občas si zkontrolovat aktuální konfiguraci pohledu (např. pomocí **do show run | beg view**).

Příkazy je možné omezovat pouze podle jejich pevné části. Tedy omezování podle nepovinných částí nebo hodnot argumentů není možné.

Pokud přidáte do pohledu nějaký složený příkaz (skládající se z více slov, např. **show ip**), systém automaticky přidá jednotlivé části tohoto příkazu do vztahu **include** pro daný pohled bez ohledu do jakého vztahu tento složený příkaz přidáváte(viz. příklady 4.1.1. a 4.1.2.).

Všechny příkazy nevyskytující se v konfiguraci pohledu jsou zakázány(až na výjimky uvedené v kapitole 2.2.2.).

Z předchozího také plyne hlavní účel **exclude**, který spočívá ve vynechávání některých příkazů z příkazových výčtů vytvořených pomocí nepovinného argumentu **all**. Při konfiguraci nějakého takového příkladu je navíc nutné si uvědomit, že **exclude** nelze použít na příkazy v nějakém jiném vztahu a proto je vždy potřeba, aby definování vztahů **exclude** předcházelo definicím vztahů **include** a **include-exclusive**. Při zadávání pravidel je tedy nutné postupovat od konkrétnějších pravidel k obecnějším(obdobně jako při vytváření ACL, viz. příklad 4.1.3.). Toto pravidlo neplatí jen pro pořadí zadávání příkazů v rámci pohledu, ale i pro pořadí zadávání jednotlivých pohledů.

Při přidávání příkazů je navíc nutné pamatovat na to, aby byl přístupný i režim, ve kterém se daný příkaz nachází - např. jestliže přidáte nějaký příkaz z konfiguračního režimu(**configure**), musíte přidat do privilegovaného režimu příkaz pro vstup do konfiguračního režimu, jinak příkaz přidaný do konfiguračního režimu nebude dostupný, přestože bude povolený. Takovéto případy systém automaticky neřeší a je třeba na to myslet.

3.2. Přístup k pohledům

Způsob přístupu k pohledu závisí na konkrétní konfiguraci AAA subsystému. Pohled může být přístupný bez hesla, pouze pod heslem atd.

Mezi pohledy je možné se přepínat pomocí příkazu:

enable view *view-name*

Pokud zadaný pohled není nakonfigurován, systém přihlásí uživatele do privilegovaného režimu.

Jména pohledů a hesla rozlišují velká a malá písmena(jsou „case senzitivní“).

4. Konfigurace

Pro všechny příklady platí, že všechny příkazy, které nejsou explicitně zmíněny v zadání jsou ve stavu automaticky předdefinovaném po vytvoření pohledu, což znamená, že až na výjimky(viz. 2.2.2) jsou všechny nezmíněné příkazy zakázány.

4.1. Konfigurace CLI pohledu

4.1.1. Jednoduchý CLI pohled

Cíl

Smyslem je vytvoření jednoduchého pohledu umožňujícího zobrazení verze IOSu, přepnutí do konfiguračního režimu z terminálu a použití všech příkazů začínajících **`show ip`**.

Řešení

```
Router(config)# parser view first
00:11:40:%PARSER-6-VIEW_CREATED:view 'first' successfully created.
Router(config-view)# secret firstpass
Router(config-view)# command exec include show version
Router(config-view)# command exec include configure terminal
Router(config-view)# command exec include all show ip
Router(config-view)# exit

!Vypis konfigurace.
Router(config-view)# do show run | beg view
parser view first
 secret 5 $1$MCmh$QuZaU8PIMPlff9sFCZvgW/
 commands exec include configure terminal
!Automaticky pridane pravidlo.
 commands exec include configure
 commands exec include all show ip
 commands exec include show version
!Automaticky pridane pravidlo.
 commands exec include show
```

Pozn.: Na tomto příkladě jsou vidět i jednoduché případy automatického přidání **`include`** pro složené příkazy **`show ip`** a **`configure terminal`**.

4.1.2. Složené příkazy

Cíl

Příklad vytvoření pohledu umožňujícího používat příkaz **`show ip interface`** ve výhradním režimu(**`include-exclusive`**) a bez možnosti konfigurace z terminálu.

Řešení

```
Router(config)# parser view slozeny
00:13:42:%PARSER-6-VIEW_CREATED:view 'slozeny' successfully created.
Router(config-view)# secret slozenypass
Router(config-view)# command exec include-exclusive show ip interface
Router(config-view)# command exec exclude configure terminal
Router(config-view)# exit

!Vypis konfigurace.
parser view slozeny
 secret 5 $1$iP2M$R16BXKecMEiQesxLyqygW.
 commands exec include-exclusive show ip interface
 commands exec include show ip
 commands exec include show
 commands exec exclude configure terminal
 commands exec include configure
```

Pozn.: Je dobré si uvědomit, že v tomto příkladě je definice pravidla **`exclude`** v podstatě zbytečná a že má jen demonstrovat případ, kdy systém automaticky generuje pravidlo **`include`** při zadání pravidla **`exclude`**.

4.1.3. Použití argumentu `all`

Cíl

Konfigurace pohledu umožňující použití všech příkazů začínajících na písmena „**`co`**“, aniž by byl přístupný konfigurační režim. Použití všech příkazů **`show ip`** bude ve výhradním režimu s tím, že příkazy **`show ip`** začínající písmenem „**`i`**“ ve výhradním režimu nebudou a příkaz **`show ip interface`** nebude přístupný vůbec.

Řešení

```
Router(config)# parser view allTest
00:13:42:%PARSER-6-VIEW_CREATED:view 'allTest' successfully created.
Router(config-view)# secret allTestpass
Router(config-view)# command exec exclude all configure
Router(config-view)# command exec include all co
Router(config-view)# command exec exclude show ip interface
Router(config-view)# command exec include all show ip i
Router(config-view)# command exec include-exclusive all show ip
Router(config-view)# exit
```

Pozn.: Na tomto příkladě si také můžete všimnout správného řazení příkazů od konkrétnějších k obecnějším.

4.2. Konfigurace Superview

4.2.1. Jednoduché superview

Cíl

Účelem je vytvoření dvou pohledů, kdy oba mohou používat všechny příkazy **`show ip`**. První pak bude mít navíc přístupný konfigurační režim a druhý si bude moci nechat zobrazit aktuální konfiguraci.

Řešení

```
!Vytvoreni prvniho CLI pohledu.
parser view one
 secret onepass
 command exec include all show ip

!Vytvoreni druhého CLI pohledu.
parser view two
 secret twopass
 command exec include all configure

!Vytvoreni tretiho CLI pohledu.
parser view three
 secret threepass
 command exec include show running-config

!Vytvoreni prvniho superview.
parser view su_view1 superview
 secret firstsupass
 view one
 view two

!Vytvoreni druhého superview.
parser view su_view2 superview
 secret secondsupass
 view one
 view three
```

Pozn.: Výsledkem jsou dva superview, kdy společné funkce mají definovány pomocí CLI pohledu **`one`** a rozdílné funkce pak pomocí CLI pohledů **`two`** a **`three`**.

4.3. Komplexní příklad

Cíl

Cílem uvedeného příkladu je vytvořit pohledy pro pět uživatelů: Peter, John, Paul, Vera a Marek. Marek má přístup ke všem **show** příkazům až na zobrazení konfigurace. Peter má pak přístup ke všem příkazům **show** i s příkazy na výpis konfigurace. Paul a John mají mít stejná práva. Budou si moci zobrazit pouze konfiguraci, můžou konfigurace kopírovat a mohou daný síťový prvek restartovat. Uživatel Vera bude mít všechna práva již uvedených uživatelů a navíc přístup ke globálnímu konfiguračnímu režimu, kde může ovlivňovat všechny volby týkající se nastavení směrování, a konfiguračnímu režimu rozhraní, kde bude mít možnost měnit pouze nastavení spojené s protokolem IP.

Řešení

```
!Prihlaseni do root view.
enable view

configure terminal

!Inicializace AAA.
aaa new-model
!Nastaveni overovani uzivatelu pri logovani vyuzitim lokalni databaze.
aaa authentication login default local

!Konfigurace pohledu jedna.
parser view jedna
 secret jedna
 command exec include-exclusive show running-configuration
 command exec include-exclusive start-configuration
 exit

!Konfigurace pohledu dva.
parser view dva
 secret dva
 command exec include all show
 exit

!Konfigurace pohledu tri.
parser view tri
 secret tri
!Povoleni moznosti restartovani daneho zarizeni.
 command exec include reload
!Povoleni moznosti kopirovat konfigurace.
 command exec include copy
 exit

!Konfigurace pohledu ctyri.
parser view ctyri
 secret ctyri
 command exec include all configure
!Povoleni moznosti zmenit nastaveni smerovani.
 command configure include all router
!Zpristupni rezimu konfigurace rozhrani.
 command configure include interface
!Povoleni moznosti menit vsechna nastaveni spojena s protokolem IP.
 command interface all ip
 exit
```

```
!Konfigurace superview su_jedna.
parser view su_jedna superview
 secret su_jedna
 view jedna
 view dva
 exit

!Konfigurace superview su_dva.
parser view su_dva superview
 secret su_dva
 view jedna
 view tri
 exit

!Konfigurace superview su_tri.
parser view su_tri superview
 secret su_tri
 view jedna
 view dva
 view tri
 view ctyri
 exit

!Naplneni lokalni databaze uzivatelu a prirazeni pohledu.
username Peter view su_jedna password peter
username John view su_dva password john
username Paul view su_dva password paul
username Vera view su_tri password vera
username Marek view dva password marek
```

4.4. Shrnutí

Výsledky všech příkladů odpovídaly cílům a představám a i přes drobné zádrhele, vytváření řešení příkladů nebylo obzvlášť složitým.

Konfigurace Role-Based CLI Access je opravdu jednoduchá a intuitivní a pro většinu případů často existuje stručné a přehledné řešení, které ale vždy nemusí být zřejmé na první pohled. Proto vřele z vlastní zkušenosti doporučuji držet se pravidla: „Dvakrát měř a jednou řež“, kdy si nejdříve dobře rozmyslíte, co vlastně chcete konfigurovat a zkusíte vymyslet alespoň dvě možná řešení.

Dále bych při konfiguraci Role-Based CLI Access doporučoval, pokud to jde, vyhýbat se používání pravidel s **`include-exclusive`**. Jejich použití totiž přímo ovlivňuje konfiguraci ostatních pohledů, neboť příkaz ve vztahu **`include-exclusive`** již není možné přidat do žádného jiného pohledu, což může být komplikací při případném dodatečném přidávání pohledů. Proto doporučuji definovat všechny pohledy nejlépe jen s využitím **`include`** a **`exclude`**.

Drobnou nevýhodou, kterou tato zkoušená komponenta má, je nemožnost použití funkce dokončení nebo okamžité nápovědy při přidávání(odebírání) příkazu a komplikujícím faktorem je také to, že ať je příkaz zadán správně nebo špatně, systém ve většině z těchto případů nic neohlásí a je tak nezbytné průběžně konfiguraci kontrolovat už během jejího zadávání.

Další nepříjemnosti pak nastávají při nastavování přístupu k jednotlivým pohledům. Pokud budete chtít nějakým způsobem(ať už pomocí hesla nebo uživatelských účtů) ověřovat přístup k pohledům, což se týká valné většiny případů, budete se muset utkat s AAA subsystémem. Konfigurace AAA je na celé záležitosti definování pohledů asi nejzáludnější a oproti samotné konfiguraci pohledů také mnohem komplikovanější. Je vhodné dříve než se pustíte do jeho konfigurace alespoň zběžně projít dokumentaci a příklady. Ušetříte si tak spoustu zbytečných problémů.

5. Závěr

Role-based CLI Access není jen zbytečným a neupotřebitelným doplňkem Cisco IOSů, ale jedná se o opravdu zcela funkční a použitelnou utilitu, která dokáže výrazně zjednodušit a zbezpečnit správu síťových zařízení a která si beze sporu najde své uplatnění.

I přes pár drobností mírně znepříjemňujících konfiguraci, Role-Based CLI Access nemá žádné vážnější nedostatky. Jedinou možnou výtkou na adresu této komponenty tak může být snad jen nedostatek volně dostupných informací, kdy po odmyšlení pár několikařádkových zmínek ve většině dostupné literatury, zůstává jen odkaz na oficiální manuálové stránky firmy Cisco[1].

6. Zdroje

[1] Role-Based CLI Access [online]. c2006, poslední revize 9.11.2006 [cit. 2007-06-18]. <http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cg/hsec_c/part30/hclivws.htm>.

[2] Configuring Authentication [online]. c2006, poslední revize 24.6.2006 [cit. 2007-06-18]. <http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fsaaa/scfathen.htm>.