

VŠB - Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra informatiky

Směřované a přepínané sítě

Možností použití skriptování v jazyku Tcl v Cisco IOS

2007

Mario Ulivelli

Jan Kordas

Úvod

V tomto článku jsou popsány možnosti skriptování v jazyku Tcl (Tool Command Language) v Cisco IOS. Po teoretickém úvodu následují ukázky skriptů a práce s nimi. Dále je popsán EEM (Embedded Event Manager), který slouží pro provádění akcí na základě událostí. V závěru je uveden příklad pro omezení zadávání hesla, který kombinuje možnosti Tcl a EEM.

Tcl v Cisco IOS

Tcl shell (tclsh) byl navrhnut k tomu, aby umožňoval uživatelům spouštět Tcl příkazy přímo z Cisco IOS CLI (Command Line Interface).

Dnešní Cisco IOS software sice obsahuje určité subsystemy jako např. ESM (Embedded Syslog Manager) a IVR (Interactive Voice Response), které používají Tcl interprety jako součást své implementace, ale tyto subsystemy mají své vlastní proprietární příkazy a volby klíčových slov, které tclsh nepodporuje.

Bylo navrženo několik metod ke spouštění Tcl skriptů a Tcl příkazů přímo uvnitř Cisco IOS. Když je Tclsh povolen, můžeme zadávat Tcl příkazy řádku po řádce. Poté co zadáme Tcl příkaz, je tento příkaz odeslán na Tcl interpreter. Pokud je tento příkaz rozpoznán, je spuštěn a výsledek je poslán na výstup. Jestliže příkaz rozpoznán není, je příkaz poslán na Cisco IOS parser. Jestliže ani v tomto případě není příkaz rozpoznán jako příkaz Cisco IOS, je vypsána chybová hláška.

Není nutné psát Tcl příkazy přímo do Tcl shellu. Předdefinované Tcl skripty mohou být vytvořeny mimo Cisco IOS a poté nahrány do paměti a spouštěny odtamtud.

Tcl shell můžeme také použít ke spouštění Cisco IOS CLI EXEC i konfiguračních příkazů uvnitř Tcl skriptů. Díky tomu můžeme automatizovat některé činnosti prováděné CLI příkazy, vytvořit “menu” pro práci, či příručku vedoucí nováčky krok po kroku při obsluze Cisco IOS.

EEM

EEM (Embedded Event Manager) poskytuje možnost monitorování událostí a provedení akcí po jejich vzniku. EEM politiky jsou entity, které definují akce, které mají být provedeny, když nastane definovaná událost. EEM politiky mohou být definovány jako applety, nebo jako skripty v jazyce Tcl.

Applety slouží k definování akcí, které se mají provést, když nastane zvolená událost. Skládají ze tří částí – spouštěcí události, akcí a nastavování proměnných. Nevýhoda appletů oproti skriptům v jazyce Tcl spočívá v jejich relativně omezené funkčnosti.

V Tcl skriptech je možno definovat více různých akcí, které nemusí být prováděny jen sekvenčně a mohou být prováděny na základě dalších definovaných podmínek. Skripty jsou tvořeny mimo Cisco zařízení a následně do něj nahrány. Na druhou stranu applety jsou vhodné, kde jimi poskytované možnosti stačí, pro jejich jednoduchost a snadnost implementace (nemusí se nikde nic nahrávat a konfigurace je snadnější).

Následující kapitoly jsou zaměřeny především na poskytnutí informací, které jsou nutné pro vyřešení omezení zadávání hesla. Další informace o manažeru událostí naleznete ve [2], [3] a [4].

Detektory událostí

Detektory událostí se liší od verzí Cisco IOS. Pro naši aplikaci je nutná podpora CLI detektoru událostí.

Event Detector	12.0(26)S 12.3(4)T	12.2(25)S	12.3(14)T 12.2(18)SXF5 12.2(28)SB 12.2(33)SRA	12.4(2)T 12.2(31)SB3 12.2(33)SRB	12.2(18)SXF4 Cisco IOS Software Modularity
Application-Specific	---	Ano	Ano	Ano	Ano
CLI	---	---	Ano	Ano	Ano
Counter	---	Ano	Ano	Ano	Ano
Enhanced Object Tracking	---	---	---	Ano	---
GOLD	---	---	---	---	Ano
Interface counter	---	Ano	Ano	Ano	Ano
None	---	---	Ano	Ano	Ano
OIR	---	---	Ano	Ano	Ano
Resource	---	---	---	Ano	---
RF	---	---	---	Ano	Ano
SNMP	Ano	Ano	Ano	Ano	Ano
Syslog	Ano	Ano	Ano	Ano	Ano
System manager	---	---	---	---	Ano
Timer		Ano	Ano	Ano	Ano
Watchdog (Cisco ISO)	---	Ano	Ano	Ano	Ano
Watchdog (Cisco ISO Software modularity)	---	---	---	---	Ano

Tabulka 1.: Zdroje událostí v závislosti na Cisco IOS (převzato ze [4])

Dále jsou vybrány některé zajímavé detektory událostí a jejich popis.

- CLI Event Detector – událost nastane, když CLI příkaz v plné formě odpovídá zadanému regulárnímu výrazu. CLI Event Detector podporuje tři módy publikování události:
 1. Synchronní publikování – CLI příkaz není proveden, dokud neskončí EEM politika, která udává, zda se má příkaz provést.
 2. Asynchronní publikování – CLI příkaz je proveden nezávisle na běhu EEM politiky.
 3. Asynchronní publikování s přeskočením příkazu – CLI příkaz není proveden.
- Counter Event Detector – tato událost nastane, když konkrétní čítač dosáhne specifikované hodnoty.
- None Event Detector – událost nastane, když je po zadání CLI příkazu **event manager run <název appletu>** spuštěna EEM politika. Politika může být také spuštěna z jiné politiky definované např. jiným appletem.
- Timer Event Detector – událost nastane v jednom z následujících čtyř případů:
 1. Nastane absolutní čas, který byl specifikován.
 2. Countdown čítač dosáhne nuly.
 3. Watchdog časovač dosáhne nuly.
 4. Na základě CRON časovače, který je podle UNIX standardu.

Akce

Akce, které mohou být provedeny po vzniku události se také liší v závislosti na verzi operačního systému. V našem skriptu níže žádnou akci neprovádíme, proto je pro nás důležitá hlavně předchozí tabulka.

Action	12.0(26)S 12.3(4)T	12.2(25)S	12.3(14)T 12.2(18)SXF5 12.2(28)SB 12.2(33)SRA	12.4(2)T 12.2(31)SB3 12.2(33)SRB	12.2(18)SXF4 Cisco IOS Software Modularity
Execute a CLI command	---	---	Ano	Ano	Ano
Generate a CNS event	Ano	Ano	Ano	Ano	Ano
Generate an SNMP trap	---	Ano	Ano	Ano	Ano
Generate a prioritized syslog message	Ano	Ano	Ano	Ano	Ano
Manually run an EEM policy	---	---	Ano	Ano	Ano
Publish an application-specific event	---	Ano	Ano	Ano	Ano
Read the state of a tracked object	---	---	---	Ano	---
Reload the Cisco IOS software	Ano	Ano	Ano	Ano	Ano
Request system information	---	---	Ano	Ano	Ano
Send a short e-mail	---	---	Ano	Ano	Ano
Set or modify a counter		Ano	Ano	Ano	Ano
Set the state of a tracked object	---	---	---	Ano	---
Switch to a secondary RP	Ano	Ano	Ano	Ano	Ano

Tabulka 2.: Akce, které mohou být provedeny podle verze Cisco IOS (převzato ze [4])

Proměnné

V EEM politikách mohou být využívány proměnné prostředí. Tcl umožňuje definovat globální proměnné, které jsou viditelné pro všechny procedury v Tcl skriptu. EEM umožňuje definování proměnných prostředí pro EEM politiky pomocí CLI příkazu (event manager environment). Všechny proměnné prostředí jsou automaticky přiřazeny globálním Tcl proměnným ještě před spuštěním Tcl skriptu. Proměnné prostředí se dělí na tři typy:

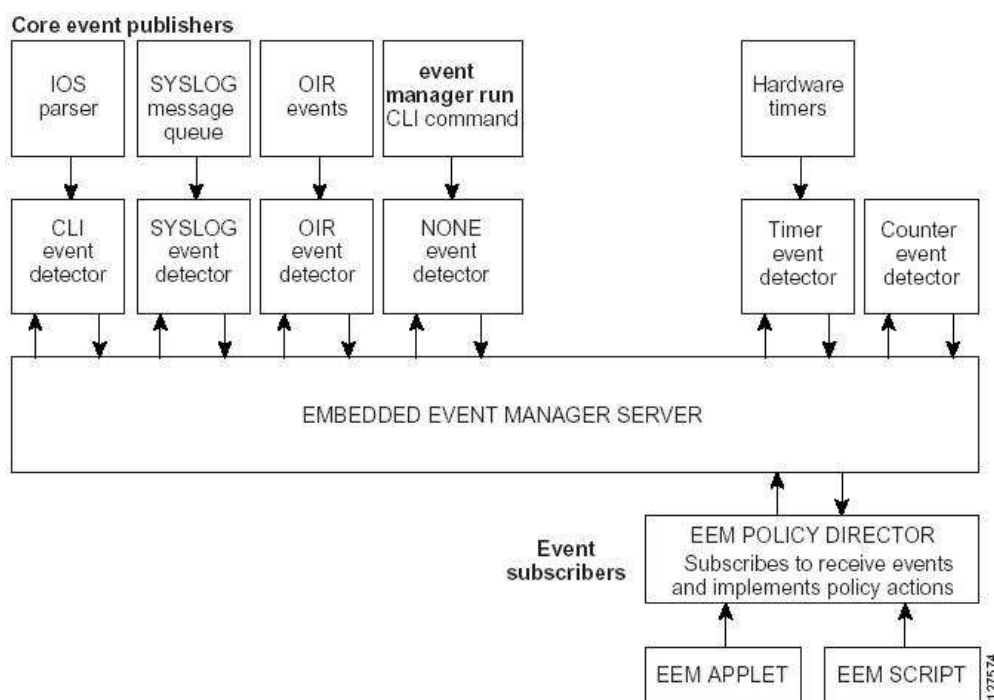
1. Uživatelsky definované – definované uživatelem, při vytváření proměnné prostředí v politice.
2. Cisco definované – definované Ciscem pro specifické vzorové politiky, jako je např. proměnná `_email_to` sloužící pro definování příjemce mailu, který je dané politice odeslán.
3. Cisco vestavěné (dostupné pouze v EEM apletech) – definované Ciscem jako podмноžina Cisco definovaných proměnných. Proměnná `_exit_status` umožňuje nastavení stavu ukončení politiky spouštěné od synchronních událostí. Tímto je možné definovat, zda se provede CLI příkaz, který byl zdrojem události. Když je `_exit_status` 0, CLI příkaz se neprovede, když je 1, příkaz se provede.

Cisco proměnné začínají vždy podtržítkem, proto se důrazně doporučuje, aby uživatelské proměnné podtržítkem nezačínaly. Tím se předejde případným konfliktům.

Pro naše potřeby je nutný EEM verze 2.1, který je podporován ve verzích Cisco IOS Release 12.3(14)T, 12.2(18)SXF5, 12.2(28)SB, 12.2(33)SRA a novějších.

EEM server

Na následujícím obrázku je zobrazen vztah mezi EEM serverem, hlavními detektory událostí, odběrateli událostí a politikami. Události jsou publikovány, když je detektorem událostí zjištěna shoda s událostí, která je definována odběratelem události (politika). EEM server se stará o zprostředkování komunikace mezi detektory událostí a odběrateli událostí.



Obrázek 1.: EEM server (převzato ze [2])

Příklady

Spouštění Tcl příkazů

Postup spouštění jednotlivých TCL příkazů:

1. Vstup do privilegovaného EXEC režimu
enable
2. (Volitelný) vstup do globálního konfiguračního módu
configure terminal
3. (Volitelné) specifikování default cesty externích kódovacích souborů, které jsou využívány TCL příkazem encoding:
scripting tcl encdir <url>
4. (Volitelné) specifikování inicializačního skriptu, který se má spustit, když je TCL shell povolen (příkazem tclsh):
scripting tcl init <url>
5. Opuštění globálního konfiguračního módu
exit
6. Povolení interaktivního TCL shellu a vstup do Tcl konfiguračního módu, ve kterém je možno zadávat Tcl příkazy:
tclsh
7. Zadání příkazu Tcl
<tcl command>
8. Opuštění Tcl konfiguračního módu a návrat do privilegovaného EXEC režimu:
exit

Předefinovaný skript se spustí z privilegovaného EXEC režimu pomocí CLI příkazu **tclsh device:skript.tcl** (kde device je zařízení, na kterém je Tcl skript umístěn).

Pro spouštění IOS CLI EXEC příkazů se používá Tcl příkaz **exec** "<exec-cmd>". Pro modifikaci konfigurace routeru se používá Tcl příkaz **ios_config** "<cmd>" "<subcmd>". Všechny argumenty a podpříkazy musí být napsány na jednu řádku.

Vytvoření a spuštění Tcl skriptu

1. Vytvoření jednoduchého Tcl skriptu Hello.tcl jehož obsahem bude:

```
puts "Type your name, please:"  
gets stdin name  
puts "Hello $name."
```
2. Uložení skriptu Hello.tcl na TFTP server s IP adresou 192.168.0.10
3. Spuštění skriptu:
tclsh tftp://192.168.0.10/Hello.tcl

Vytvoření EEM politiky

EEM politika se vytváří následovně:

1. Výběr zdroje události, která politiku spouští (např. CLI příkaz).
2. Definování voleb detektoru událostí (např. synchronní, asynchronní provedení politiky)
3. Definování proměnných prostředí (volitelné)
4. Výběr akcí, které budou provedeny, když nastane událost.

V následujících dvou kapitolách jsou uvedeny příklady dvou politik. Jedné vytvořené pomocí appletu a druhé vytvořené pomocí Tcl skriptu.

Vytvoření EEM politiky definované appletem

Na následujícím příkladě je ukázán postup definice appletu, který po zadání CLI příkazu `show int` provede reload zařízení.

1. Vytvoření Appletu (v konfiguračním režimu)
event manager applet muj
2. Nastavení spouštěcí události (zadání CLI příkazu obsahujícího `show int`)
event cli pattern "show int" sync yes
3. Nastavení akce, která se má provést (reload)
action 1 reload
4. Nastavení `exit_status` (0 – CLI příkaz, který byl zdrojem události, se neprovede, 1 – CLI příkaz se provede). V této konkrétní politice již příkaz nemusí být, je zde jen pro demonstrování možností.
set 2 _exit_status 0

Další informace k EEM politikám, které jsou definovány pomocí appletů jsou ve [4].

Vytvoření EEM politiky definované skriptem

Postup registrování EEM politiky reprezentované Tcl skriptem.

1. Vytvoření skriptu
2. Nahrání skriptu `cmddis.tcl`, který je uveden v další části, do flash paměti z tftp serveru:
copy tftp://192.168.0.10/cmddis.tcl flash:tcl
3. Přejít do konfiguračního režimu
configure terminal
4. Nastavení cesty k uživatelským skriptům s politikami (adresář `tcl` na flash paměti)
event manager directory user policy flash:tcl
5. Registrace skriptu v EEM
event manager policy cmddis.tcl type user

Další informace k EEM politikám, které jsou definovány pomocí skriptů jsou ve [3].

Blokování vybraných CLI příkazů

Blokování vybraných CLI příkazů se provede následovně:

1. Spuštění politiky, která je definovaná Tcl skriptem při každém zadání CLI příkazu
2. Kontrola, zda se zadaný příkaz včetně parametrů shoduje s některým z povolených příkazů. Pokud se shoduje, zadaný příkaz se provede, jinak se pokračuje dalším bodem.
3. Když se zadaný CLI příkaz shoduje s některým příkazem z množiny zakázaných příkazů, zadaný příkaz se neprovede a vypíše se hláška. Jinak se zadaný CLI příkaz provede.

Definování spuštění politiky na základě zadání libovolného příkazu je snadné, stačí zaregistrovat CLI událost se vzorem, kterému vyhoví libovolný CLI příkaz. Zadaný příkaz včetně zadaných parametrů je ve skriptu dále rozebírán pomocí regulárních výrazů.

Získání posledního zadaného příkazu, který vyvolal událost, už tak snadné není. V appletu je pro tento účel definovaná vestavěná proměnná `_cli_msg`. Pro Tcl skripty bohužel tato proměnná definována není. Proto se musí poslední příkaz získat ze seznamu historie zadaných příkazů. Ten je možné získat Tcl funkcí `sys_reqinfo_cli_history`, který je definována v jmenném prostoru `::cisco::eem`. Výsledek je ve tvaru:

```
rec_list {{CLI history string 0}, {CLI history str 1},...}
```

Jako druhá položka vráceného seznamu je řetězec, který obsahuje poslední zadané příkazy. Poslední zadaný příkaz je na konci řetězce. Řetězec s historií příkazů má následující tvar:

```
time_sec %ld time_nsec %ld cmd {%s}
```

Pro nás je důležitá část řetězce za podřetězcem `cmd`. Regulární výraz, který bude popisovat poslední zadaný příkaz tedy bude vypadat takto `{.* cmd \{ [^\}]*\}\$}`. Za podřetězcem `cmd` musí následovat mezera a znak `{`, po kterém následuje samotný příkaz, který nesmí obsahovat znak `}`. Výraz je ukončen dvojicí znaků `}`. Z podřetězce, který vyhověl výše uvedenému příkazu, se odstraní počáteční a koncové znaky, aby se získal řetězec, který obsahuje pouze poslední zadaný příkaz a jeho argumenty.

Nyní můžeme snadno definovat regulární výrazy popisující námi povolené, nebo zakázané příkazy. Výraz pro povolení hesla `cisco` vypadá takto `{.*password cisco.*}`. Aby bylo možno zadané heslo zrušit musíme také povolit příkazy vyhovující regulárnímu výrazu `{.*password$}`. Definicí regulárních výrazů `{.*password.*}`, `{.*cmddis.*}`, `{.*event manager.*}` popisujících zakázané příkazy, zakážeme zadávání hesla kromě dříve povolených vzorů, příkazy obsahující `cmddis` a `event manager`.

Příkazy obsahující `cmddis` zakazujeme, protože Tcl skript s námi definovanou politikou se jmenuje `cmddis.tcl`. Tímto se znesnadní modifikace, či smazání našeho skriptu. Také zakazujeme práci s `event managerem`, aby uživatel nemohl definovanou politiku zrušit.

Následuje okomentovaný skript (`cmddis.tcl`), který řeší blokování vybraných příkazů. Regulární výrazy, které definují zakázané příkazy, jsou v seznamu, který je uložen do proměnné `patterns0`. Seznam regulárních výrazů, které definují povolené příkazy, je v proměnné `patterns1`. Pro testování, zda vstupní řetězec odpovídá regulárnímu výrazu slouží Tcl funkce `regexp`.

```

#Skript cmddis.tcl:

# definování spouštěcí události = zadání libovolného příkazu
::cisco::eem::event_register_cli pattern ".*" sync yes

# definování vzorů příkazů, které mají být povoleny
set patterns1 [list {.*password cisco.*} {.*password$}]

# definování vzorů příkazů, která mají být zakázána
set patterns0 [list {.*password.*} {.*cmddis.*} {.*event manager.*}]

# získání historie zadanych CLI příkazů do proměnné cmds
set cmd_list [::cisco::eem::sys_reqinfo_cli_history]
set cmds [lindex $cmd_list 1]

# získání posledního zadaného příkazu do proměnné cmd
regexp {cmd \{[^\}]*\}\}$} $cmds cmd1
set len [string length $cmd1]
set len [expr $len - 3]
set cmd [string range $cmd1 5 $len]

# test, zda zadaný příkaz je mezi explicitně povolenými
foreach x $patterns1 {
    if {[regexp $x $cmd]} {
        # ukončení skriptu a následné provedení zadaného příkazu
        exit 1
    }
}

# test, zda zadaný příkaz je mezi zakázanými
foreach x $patterns0 {
    if {[regexp $x $cmd]} {
        puts "Command is not permitted!\n"
        # ukončení, bez provedení CLI příkazu, který vyvolal událost
        exit 0
    }
}
exit 1

```

Závěr

Tcl poskytuje zajímavé rozšíření Cisco IOS, které jistě najde uplatnění v nejrůznějších aplikacích. V současné době se již např. na internetu objevuje řada užitečných skriptů pro zjednodušení práce obsluhy Cisco routerů. Pomocí Tcl se dají také vyvíjet “uživatelská menu“, odesílat emaily při změně stavu routeru, či automatizovat určité činnosti.

Na našem praktickém příkladě jsme demonstrovali propojení Tcl s EEM. Uvedený příklad demonstruje jen jednu z mnoha možností, které EEM spolu s Tcl poskytuje. Drobnou modifikací skriptu lze definovat další zakázané, nebo naopak povolené příkazy.

Seznam zkratek

CLI	Command Line Interface
CNS	Cisco Networking Services – viz. [10]
EEM	Embedded Event Manager
GOLD	Generic Online Diagnostic – viz. [7]
OIR	Online Insertion and Removal – viz. [8]
RF	Redundancy Framework
RP	Route Processor
SNMP	Simple Network Management Protocol – viz. [9]
Tcl	Tool Command Language

Reference

- [1] Cisco IOS Scripting with Tcl
http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801a75a7.html
- [2] Embedded Event Manager Overview
http://www.cisco.com/en/US/products/ps6441/products_configuration_guide_chapter09186a00807c676c.html
- [3] Writing Embedded Event Manager Policies Using Tcl
http://www.cisco.com/en/US/products/ps6441/products_configuration_guide_chapter09186a00807c6784.html
- [4] Writing Embedded Event Manager Policies Using the Cisco IOS CLI
http://www.cisco.com/en/US/products/ps6441/products_configuration_guide_chapter09186a00807c6773.html
- [5] Tutoriál k jazyku Tcl
<http://www.tcl.tk/man/tcl8.5/tutorial/tcltutorial.html>
- [6] Writing Embedded Event Manager Policies
http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a008041231a.html
- [7] The Generic Online Diagnostic
http://www.cisco.com/en/US/products/ps7081/products_qanda_item0900aecd804c224e.shtml
- [8] Online Insertion and Removal
http://www.cisco.com/warp/public/63/oir_17940.shtml
- [9] Simple Network Management Protocol
<http://www.cisco.com/warp/public/535/3.html>
- [10] Cisco Networking Services
http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a0080455963.html