NWSY

Scripting auf Routern

Ausarbeitung

Alexander Rieppel

2. Juni 2014

5AHITT

Inhaltsverzeichnis

1 Einführung

1.1 Was heißt Scripting?

Scripting wird meist unter Verwendung von so genannten Skript-Sprachen betrieben. Genauer beschrieben handelt es sich dabei um Programmiersprachen die das Erstellen von Skripts ermöglichen. Skripts sind Programme die in einer Laufzeitumgebung interpretiert werden. Das heißt, dass die Laufzeitumgebung Befehle erkennt die Befehl für Befehl ausgeführt werden. Dies könnte analog hierzu natürlich auch von einem Menschen erledigt werden der jeden Befehl einzeln in die Kommandozeile eintippt. Da dies allerdings für wiederkehrende Prozeduren sehr aufwendig und zeitraubend ist, gibt es Skripts die diese Vorgänge automatisieren können. Scripting findet in zahlreichen verschiedenen Umgebungen Anwendung, wie Software Applikationen, Webseiten in einem Web-Browser, in der Kommandozeile eines Betriebssystems und auch in Embedded Systems. Skript-Sprachen werden auch oft als "very high-level" Programmiersprache bezeichnet, da sie sich auf einem hohen Abstraktionslevel befinden und meist für die einfache Bedienung von bestimmten Komponenten verantwortlich sind.

Mit dem Begriff Skript-Sprache wird ebenfalls oft auf die sogenannten dynamischen high-level Mehrzweck-Sprachen verwiesen. Unter diesen Begriff fallen beispielsweise Sprachen wie Perl, TCL und Python, wo ein Skript meist ein kleines Programm (bis zu wenigen tausend Zeilen Code) darstellt.

Viele dieser Sprachen wurden ursprünglich für die Verwendung in spezifischen Umgebungen entwickelt und erst später als portable domänenspezifische oder Mehrzweck-Sprache konzipiert. Das Spektrum der Skript-Sprachen reicht von sehr kleinen und hoch domänenspezifischen Sprachen bis hin zu Mehrzweck-Programmiersprachen die fürs Scripting verwendet werden.

Im Prinzip kann allerdings jede Sprache fürs Scripting verwendet werden. Offiziell kann gesagt werden, dass es immer von der Verwendung der jeweiligen Sprache abhängt und genauer wie die Sprache für bestimmte Aufgaben eingesetzt wird. Deshalb ist es nicht ohne weiteres geklärt welche Sprachen als Skript-Sprache angesehen werden können und welche nicht. Generell sind allerdings viele Sprachen nicht für den Einsatz als Skript-Sprache ausgelegt und folglich auch sehr selten als solche in Verwendung. Typischerweise sind Skript-Sprachen so ausgelegt, dass diese eine relativ simple Syntax und Semantik besitzen. Als Beispiel ist es nicht gängig die Programmiersprache Java als Skript-Sprache zu verwenden, da sie eine lange Syntax und restriktive Regeln, welche Klasse in welchen

Dateien liegt, besitzt. Im Gegensatz dazu steht bspw. Python wo es einfach möglich ist einige Funktionen in einer Datei kurz zu definieren. Eine Skript-Sprache wird normalerweise aus Sourcecode zu Bytecode interpretiert. Skript-Sprachen sind im eigentlichen Sinne für den Endnutzer eines Programms konzipiert, damit kleine Programme selbst entwickelt werden können, ohne die verschiedenen Datentypen und das Speichermanagement beachten zu müssen. [Wikb]

1.2 Gängige Skript-Sprachen

1.2.1 Glue Languages

Eine "glue language" ist eine normalerweise interpretierte Skript-Sprache welche dafür designet und konzipiert wurde um "glue code" zu schreiben - Code der Software Komponenten miteinander verbindet. Sie sind in erster Linie nützlich für:

- eigene Befehle für eine Kommandozeile
- kleinere Programme (als diese die besser in einer kompilierten Sprache entwickelt werden)
- "Wrapper" Programme für Anwendungen, wie bspw. ein Batch-File, dass Dateien innerhalb eines Betriebssystems verschiebt oder bearbeitet
- Skripte die sich oft ändern
- schnell entwickelte Prototypen einer Softwarelösung, die später in einer anderen, normalerweise kompilierten, Sprache umgesetzt werden

Glue Language Beispiele:

- Erlang
- Unix Shell scripts (ksh, csh, bash, sh und andere)
- Windows PowerShell
- ecl
- DCL
- Scheme
- JCL
- m4
- VBScript
- JScript und JavaScript

- Apple Script
- LUA
- Python
- TCL
- Ruby
- Perl
- PHP
- Pure
- REXX
- XSLT

[Wikb]

2 Scripting auf Routern

Beschreibt das automatisierte Ausführen von bestimmten Konfigurationszeilen oder andere automatisierte Aufgaben. Diese können sowohl innerhalb des Betriebssystems des Routers selbst oder auch per externem Programm ausgeführt werden. Oft gibt es ebenfalls die Möglichkeit oben beschriebene standardisierte Skript-Sprachen zu verwenden, wie bspw. Tcl oder ähnliches. Skripte werden für gewöhnlich zu fixen Zeiten, fortlaufend oder als Reaktion auf ein bestimmtes Ereignis ausgeführt. Auch gibt es meist die Möglichkeit Skripte zu schreiben, die vom Netzwerkadministrator manuell ausgeführt werden, um wiederholende Abläufe einfach abrufen zu können.

2.1 Nutzen

Wie bereits erwähnt dienen Skripte meist dazu um bestimmte Aubläufe zu automatisieren. Die Hauptvorteile liegen daher in der Arbeitsersparnis des Netzwerkadministrators, da dieser nicht jeden Befehl immer einzeln eingeben muss und auch in der schnellen Reaktionszeit nach bestimmten Ereignissen. Kein Netzwerkadministrator kann so schnell auf ein Ereignis reagieren wie der Router selbst. Ein einfaches Beispiel für ein solches Ereignis wäre das unerwartete herunterfahren eines FastEthernet Ports, der anhand eines einfachen Skriptes automatisch wieder hochgefahren werden kann.

2.2 Möglichkeiten

2.2.1 Expect

Expect ist eine Erweiterung zu Tcl, welche Interaktionen mit Programmen automatisieren kann. Vorraussetzung hierfür ist, dass ein Textterminal für die Kommunikation mit dem Gerät verwendet wird. Es wird für die automatische Kontrolle von interaktiven Applikationen verwendet, wie telnet, ftp, passwd, fsck, rlogin, tip, ssh und andere. Ursprünglich war Expect nur für Unix Systeme verfügbar, ist aber bereits für Microsoft Windows und andere Betriebssysteme verfügbar. Für die Kommunikation werden Pseudo-Terminals im Fall von Unix oder eine Konsolenemulation im Fall von Windows verwendet.

Einfaches “Hello World“ Beispiel:

```
1 #!/usr/bin/expect
2 expect "hello"
3 send "world"
```

In diesem Beispiel wird in der Konsole auf den String “Hello“ gewartet. Wenn der erwartete String erscheint, gibt Expect den String “world“ zurück.

Obwohl Expect eigentlich nicht primär für Router vorgesehen ist, kann es dennoch erfolgreich für bestimmte Abläufe verwendet werden. Dies könnten z.B. Ereignisse sein die in der Kommandozeile vom Router ausgegeben werden. Wenn diese Zeilen erkannt werden schaltet sich das Skript ein und kann entsprechend der Konfiguration vorgehen. [Wika]

2.2.2 Ciscoworks LAN Management Solution (LMS)

LMS ist ein Programm mit Management Funktionen, die die Konfiguration , Administration, Monitoring und Troubleshooting von Cisco Netzwerken erleichtert. LMS erlaubt dem Netzwerkadministrator das managen des Netzwerks über ein Browser basiertes Interface, das jederzeit und überall im Netzwerk abgerufen werden kann. Einmal installiert kann die Software durch Monitoring und Troubleshooting Dashboards schnellstmöglich auf Fehler im Netzwerk aufmerksam machen, damit Netzwerkadministratoren diesen isoliere und beheben können. Dem Administrator steht auch ein Template Center zur Verfügung in dem Cisco validierte Designs und Links zu Router und Switch-Konfigurationen angeboten werden. [Cisb]

Auch wenn hier keine direkte Skript-Sprache im Hintergrund läuft, erfüllt diese Software trotzdem das was ein Skript auch können soll. Und zwar Vorgänge automatisieren, auf Ereignisse im Netzwerk schnell reagieren und dem Netzwerkadministrator damit seine Arbeit zu erleichtern.

2.2.3 Tcl

Tcl (Tool Command Language) ist eine mächtige und einfach zu lernende dynamische Programmiersprache, passend für eine große Variation an Aufgaben. Sie kann bspw. für Web- und Desktop-Applikationen eingesetzt werden, aber auch für Netzwerk-, Administrations- und Testing-Aufgaben. Tcl ist Open-Source, cross-platform und stetig weiterentwickelnd. Ein Tcl-Script besteht aus einer Reihe von Befehlen. Jeder einzelne Befehl besteht seinerseits aus dem Befehlnamen, z.B. “set“ und aus einer Reihe von String-Argumenten. Die Anzahl und Art der Argumente ist von Befehl zu Befehl unterschiedlich. [TCL] Cisco IOS besitzt für TCL-Skripte eine eigene Shell - die Cisco IOS Tcl Shell. Sie wurde designet um Benutzern zu ermöglichen Tcl Befehle, direkt von der Cisco IOS prompt auszuführen.

“Several methods have been developed for creating and running Tcl scripts within Cisco IOS software. A Tcl shell can be enabled, and Tcl commands can be entered line by line. After Tcl commands are entered, they are sent to a Tcl interpreter. If the commands are recognized as valid Tcl commands, the commands are executed and the results are sent to the TTY device. If a command is not a recognized Tcl command, it is sent to the Cisco IOS CLI parser. If the command is not a Tcl or Cisco IOS command, two error messages are displayed. A predefined Tcl script can be created outside of Cisco IOS software, transferred to flash or disk memory, and run within Cisco IOS software. It is also possible to create a Tcl script and precompile the code before running it under Cisco IOS software.“[Cisc]

```
1  tclsh
2  proc get_bri {} {
3      set check ""
4      set int_out [exec "show␣interfaces"]
5      foreach int [regexp -all -line -inline "(^BRI\[0-9]/\[0-9])"
           $int_out] {
6          if {![string equal $check $int]} {
7              if {[info exists bri_out]} {
8                  append bri_out "," $int
9              } else {
10                 set bri_out $int
11             }
12             set check $int
13         }
14     }
15     return $bri_out
16 }
```

Obiges Beispiel zeigt ein, auf einem Router ausführbares Tcl-Skript, welches den Befehl “show interfaces“ auf einen bestimmten Output hin filtert. In diesem Fall handelt es sich um einen Filter der ausschließlich BRI Interfaces auf dem Router anzeigt.

2.2.4 Cisco IOS EEM

Der Embedded Event Manager ist im Auslieferungszustand (ab IOS Version 12.2SX) in jedem Cisco IOS Router und Switch-Betriebssystem vollständig integriert. Es kann dazu verwendet werden um Prozesse zu automatisieren und das Verhalten des Betriebssystems in bestimmten Fällen zu beeinflussen. Dazu wird vom EEM eine eigene Skript-Sprache bereitgestellt, mit deren Hilfe Skripts erstellt werden können. Diese Skripte werden in der Routerkonfiguration abgelegt und sind auch dort als Konfigurationszeilen einseh- und einspielbar. Prinzipiell kann jede denkbare Form von Skript die auf einem Router sinnvoll ist, in der Skript-Sprache von Cisco EEM umgesetzt werden.[Cisa] Zur Veranschaulichung ein Beispiel:

```
1  event manager applet tech
2   event none
3   action 1.0 cli command "enable"
4   action 1.1 cli command "sh␣fl␣|␣i␣tech1"
5   action 1.04  info type routername
6   action 1.2 if $_cli_result eq $_info_routername#
7   action 1.3 cli command "sh␣tech␣|␣redirect␣flash:/tech1.txt"
8   action 1.31 syslog msg "tech-support␣1␣created"
9   action 1.4 cli command "delete␣/force␣tech2.txt"
10  action 1.5 else
11  action 2.0 cli command "sh␣fl␣|␣i␣tech2"
12  action 2.1 if $_cli_result eq $_info_routername#
13  action 2.2 cli command "sh␣tech␣|␣redirect␣flash:/tech2.txt"
14  action 2.21 syslog msg "tech-support␣2␣created"
15  action 2.3 cli command "delete␣/force␣tech3.txt"
16  action 2.4 end
17  action 2.5 end
```

Hier wird ein Beispielskript gezeigt, das die Aufgabe hat tech-supports im flash des Routers abzuspeichern. Es speichert immer zwei tech-supports ab. Wenn ein dritter tech-support gespeichert werden soll wird der älteste gelöscht. Das Erstellen eines tech-supports stellt in vielerlei Hinsicht eine sinnvolle Logging-Möglichkeit dar, da sämtliche Informationen einschließlich Hardware und Konfiguration darin enthalten sind. Da in diesem Fall “event none“ eingetrage ist muss das Skript von Hand vom Netzwerkadmin ausgeführt werden. Stattdessen kann dieses auch bspw. mit einer Uhrzeit versehen werden, sodass das Skript immer zu einer bestimmten Uhrzeit ausgeführt wird. Dies und Informationen zu den einzelnen Befehlen, können der Cisco EEM Dokumentation und den Tutorials zu EEM auf der Cisco-Website entnommen werden.

2.2.5 SNMP

SNMP oder auch Simple Network Management Protocol ist ein von der IEFT entwickeltes Netzwerkprotokoll welches zur Überwachung und Steuerung von Netzwerkkomponenten dient. Gedacht ist dies so, dass von einer zentralen Station aus auf die Netzwerkkomponenten zugegriffen wird und dann Änderungen über dieses Protokoll möglich sind. Wenn man sich nun auf die Cisco Seite beschränkt ist es bspw. möglich zuvor erstellte Konfigurationen via FTP oder TFTP auf einen Router oder Switch über SNMP einzuspielen. Allerdings ist es ebenfalls möglich ganze Konfigurationsschritte mittels SNMP (über walk, set, get, wobei set hierbei fürs ändern zuständig ist) durchzuführen. Hierzu kann von der zentralen Konfigurationsstelle wiederum ein Skript ausgeführt werden welche alle Konfigurationsschritte über SNMP automatisiert vornimmt. Unter folgendem Link befindet sich ein Beispiel zur Konfiguration eines VLANs auf einem Cisco Switch.

http://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-

snmp/45080-vlans.html#topic1[Cisd]

Literaturverzeichnis

[Cisa] Cisco: *Cisco IOS Embedded Event Manager - http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-embedded-event-manager-eem/datasheet_c78-692254.html*

[Cisb] Cisco: *Cisco Works LAN Management Solution - http://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/ciscoworks-lan-management-solution-4-0/data_sheet_c78-610760.html*

[Cisc] Cisco: *Scripting with Tcl on Cisco IOS - http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ios_tcl/configuration/12-4t/ios-tcl-12-4t-book/nm-script-tcl.html*

[Cisd] Cisco: *SNMP-Beispiel - http://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-snmp/45080-vlans.html#topic1*

[TCL] TCL/TK: *Tcl Overview - http://www.tcl.tk/*

[Wika] Wikipedia: *Expect - http://en.wikipedia.org/wiki/Expect*

[Wikb] Wikipedia: *Scripting Languages - http://en.wikipedia.org/wiki/Scripting_language*