Nix

De tudo um pouco: experiências de viagens, foos, fotos, letras de música, linux e outros assuntos…

Comando expect

09/07/201306/04/2014 Nix

Recentemente, tive que fazer a instalação de um software em diversos (ok, muitos) servidores. Imediatamente, pensei em automatizar a tarefa quando me deparei com um problema: a instalação era feita em modo interativo, isto é, esperava receber algumas respostas. Pesquisando no google e conversando com algumas pessoas, fui apresentada ao expect [**1**], que me ajudou a resolver o problema (viva a comunidade opensource! \o/).

O expect é uma ferramenta para automatizar um processo que recebe comando interativos, isto é, que exibe um prompt e espera que o usuário digite alguma resposta. Alguns exemplos de aplicações interativas são: passwd, ftp, fsck, ssh, entre outras.

Três comandos compõe a estrutura do expect: *spawn*, *expect* e *send* e o uso dele é simples. O comando spawn inicia o processo, expect aguarda por uma sequência de strings de um processo e send envia uma string para um processo (comandos, senhas e outros).

Instalação

expect é uma extensão Tcl [**2**] e pode ser instalado em praticamente todas distribuições Linux, seja através de um gerenciador de pacotes como por exemplo apt-get no Debian, yum no Red Hat ou ainda através do código fonte [**3**]. No meu caso, estou utilizando o Linux Mint (derivado do Ubuntu, que por sua vez deriva do Debian) e para instalar executei:

$ sudo apt-get install expect

Como o expect trabalha

Scripts expect são escritos como a maioria de outros scripts. Assim como no Bash ou Perl, onde a primeira linha com o shebang[**4**] indica o programa ou shell que será utilizado para executar os comandos do script, expect também utiliza essa formatação. Para localizar o binário do expect, basta utilizar o comando which:

$ which expect
/usr/bin/expect

É necessário saber o path do binário pois essa informação será colocada na primeira linha do script.

A maioria dos scripts começam com o processo que se quer interagir, como por exemplo “spawn ssh ivani@localhost” ou “spawn ftp localhost”. Em uma forma simples, o processo é iniciado com o comando spawn que gera uma saída onde o expect irá procurar padrões para em seguida, com base nos resultados, enviar comandos através do comando send.

O problema

Pacote que durante a instalação faz algumas perguntas tais como aceite de licença e path dainstalação.

Solução encontrada

1) Criação da lista de hosts onde será feita a instalação.

2) Criação do script install_app.exp com os comandos a serem executados e as respostas esperadas. Lembrando que o Linux não reclama com esse negócio de extensão, apenas utilizei .exp para ficar organizado e eu saber que se trata de um script expect.

3) Criação de um shell script que executa o script install_app.exp em cada host que consta na lista criada no item 1.

No lab, a simulação da instalação do pacote foi feita com um instalador java antigo (jre-6u7-linux-i586.bin) que tem o mesmo comportamento da aplicação que precisei instalar.

1) Lista de hosts

Arquivo texto com os hosts listados linha a linha. Exemplo:

```
$ cat hosts.txt
192.168.1.100
```

2) Script install_app.exp

Corpo do script expect que faz a instalação remota da aplicação; linhas foram numeradas para facilitar a explicação.

```
1 #!/usr/bin/expect -f
2 set HOST [lindex $argv 0]
3 exp_internal 1
4 spawn ssh ivani@$HOST
5 expect “%”
6 send “/home/ivani/jre-6u7-linux-i586.bin \r”
7 send “%q”
8 expect “% EOF”
9 send “yes \r”
10 expect “% Done.”
11 send “exit \r”
12 expect eof
```

Linha 1 – #!/usr/bin/expect

Indica o path do interpretador de comandos que será utilizado.

Linha 2 – set HOST [lindex $argv 0]

Atribui variáveis. Scripts criados com o expect devem ter o formato Tcl; assim, não é possível utilizar variáveis como $1, por exemplo, para pegar parâmetros passados na linha de comando. É necessário utilizar a variável argv junto com o comando [lindex $argv 0], que faz com o primeiro parâmetro indicado na linha de comando (no caso, o nome do host) seja colocado na variável $HOST.

Linha 3 – exp_internal 1

Ativa a função de debug e irá mostrar todas as ações que o expect está realizando. Para desligar basta retirar a linha do script ou simplesmente trocar 1 por 0.

Linha 4 – spawn ssh ivani@$HOST

Inicia o processo da aplicação interativa, no caso, o ssh.

Linha 5 – expect "%"

Aguarda uma instrução (nesse caso, o prompt).

Linha 6 – send "/home/ivani/jre-6u7-linux-i586.bin \r"

Envia um comando para ser executado remotamente, e oparâmetro \r é o **<enter>** esperado.

Linha 7 – send "%q"

No pacote que está sendo instalado, o comando more é acionado para exibir uma licença e é necessário pressionar espaço para ler ou ir direto para o fim do arquivo, onde o usuário será questionado se quer ou não aceitar. Aqui, foi enviado "q" para sair da licença.

Linhas – 8 e 9:

expect "% EOF"
send "yes \r"

Quando se pressiona "q" para sair da licença, é enviado um "EOF" indicando saída da leitura da licença e aparece para o usuário se quer ou não aceitar a licença; nesse ponto, o script envia a resposta yes.

Linhas 10 e 11:

expect "% Done."
send "exit \r"

Ao encontrar o termo "Done", é enviado o comando exit para desconectar da sessão ssh.

Linha 12 – expect eof

Garante que o script aguarde a execução dos comandos antes de retornar o controle para o script .sh e finalizar a conexão SSH.

3) Script install_java.sh

Shell script que chama o script expect para instalar o pacote nos servidores.

Esse shelll é bem simples, foi criado um laço que irá ler o arquivo hosts.txt e para cada linha nesse arquivo, será executado o script java.exp.

A seguir o script:

```
1 #!/bin/bash -x
2
3 for host in $(cat hosts.txt)
4 do
5 /home/ivani/Scripts_Learning/java.exp $host
6 done
```

Abaixo, trechos da execução do script install_java.sh (partes foram suprimidas para o post não ficar tãão longo…).

ivani@nix ~/Scripts_Learning $ **pwd**
/home/ivani/Scripts_Learning
ivani@nix ~/Scripts_Learning $ **./install_java.sh**
++ cat hosts.txt
+ for host in '$(cat hosts.txt)'
+ /home/ivani/Scripts_Learning/java.exp 192.168.1.100
spawn ssh ivani@192.168.1.100
Linux lab 2.6.32-5-686 #1 SMP Fri Feb 15 15:48:27 UTC 2013 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

[…]
Last login: Tue Jul 9 20:58:21 2013 from 192.168.1.101
ivani@lab:~$ **/home/ivani/jre-6u7-linux-i586.bin**

Sun Microsystems, Inc. Binary Code License Agreement

for the JAVA SE RUNTIME ENVIRONMENT (JRE) VERSION 6

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THE
SOFTWARE IDENTIFIED BELOW TO YOU ONLY UPON THE CONDITION
THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY
CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS
(COLLECTIVELY "AGREEMENT"). PLEASE READ THE AGREEMENT
CAREFULLY. BY DOWNLOADING OR INSTALLING THIS SOFTWARE, YOU
ACCEPT THE TERMS OF THE AGREEMENT. INDICATE ACCEPTANCE BY
SELECTING THE "ACCEPT" BUTTON AT THE BOTTOM OF THE
AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY ALL THE
TERMS, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THE
AGREEMENT AND THE DOWNLOAD OR INSTALL PROCESS WILL NOT
CONTINUE.

[…]

Do you agree to the above license terms? [yes or no]

yes
Unpacking…
Checksumming…
Extracting…
UnZipSFX 5.50 of 17 February 2002, by Info-ZIP (Zip-Bugs@lists.wku.edu).
creating: jre1.6.0_07/
creating: jre1.6.0_07/bin/
inflating: jre1.6.0_07/bin/java
[…]

Done.
ivani@lab:~$ **exit**
logout
Connection to 192.168.1.100 closed.
ivani@nix ~/Scripts_Learning $

Até o próximo post :)!
Referências

[1] http://en.wikipedia.org/wiki/Expect (http://en.wikipedia.org/wiki/Expect)

[2] http://en.wikipedia.org/wiki/Tcl (http://en.wikipedia.org/wiki/Tcl)

[3] http://www.nist.gov/el/msid/expect.cfm (http://www.nist.gov/el/msid/expect.cfm)

[4] http://en.wikipedia.org/wiki/Shebang_(Unix) (http://en.wikipedia.org/wiki/Shebang_(Unix))

Getting Started With Expect (http://oreilly.com/catalog/expect/chapter/ch03.html) – O'Reilly

Learning automation, command, expect, Linux, script, Tcl, toolbox, tools

7 comentários sobre "Comando expect"

1. **FABIO**
   25/10/2013 / 3:22 PM
   Ótimo tutorial mas eu me perdi quando é que você chama o hosts.txt?

   NIX
   30/10/2013 / 12:16 AM
   Fabio, obrigada pela visita e pelo comentário.
   O arquivo hosts.txt é lido no script install_java.sh, no item 3 do post :).

2. **BRUNO**
   03/10/2014 / 9:58 PM
   Muito bom, Parabéns!

3. **BORGEEK**
   27/08/2015 / 2:48 PM
   Aquele momento em você vai pesquisar um comando e chega no blog da Nix 🙂

NIX
08/09/2015 / 11:34 PM
Hey Bruna!!!! Obrigada pela visita! Mundo virtual é pequeno 🙂 Por onde andas?

4. **THIAGO HENRIQUE**
   22/02/2016 / 4:10 PM
   Boa tarde, como que faz para mandar a saída do servidor para um arquivo?

5. **THIAGO TAMIOZZO**
   15/10/2016 / 1:23 PM
   Material muito bom, parabéns!!

Os comentários estão desativados.