

Running a VM nated behind my Laptops IP

SSH from my Laptop to my VM

File: Commands.txt

```
show ip int bri | e una
show version | .bin
```

File: Devices.txt

```
192.168.88.99
10.1.2.3
ch-switch-1
br-switch-1
jp-switch-1
CONFIG_ENDE
```

```
VM# ./job-creator-script.sh
VM# ./cisco.exp.joblist.sh

Check output

VM# cd output

VM# more * | egrep "(hostname|ntp peer)"
```

commands.txt:

```
sh ip int brief | e una
conf t
int lo0
description BLA
end
sh run int lo0
```

Devices.txt:

```
ch-switch-01
ch-switch-02
br-switch-01
de-switch-01
jp-switch-01
...
CONFIG_ENDE
```

job-creator-script.sh
Creates the cisco.exp.joblist.sh
And displays what will be
configured or fetched.

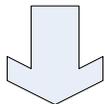
Expect.script.sh
(Can be used manually to,
check ./expect.script.sh -h)

```
[colin@LINUX expect.script]$ ./expect.script.sh
ERROR: Missing required args, must have -h, -u, -p

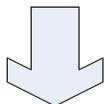
Usage: vty_runcmd.exp <options>

REQUIRED OPTIONS:
-h <hostname|ip> = hostname or ip address
-u <username>    = username to login with
-p <password>    = password for login

Other OPTIONS:
-e <enable password> = Enable password
-t <seconds>        = timeout in seconds
-m <ssh|telnet>    = use either ssh or telnet, default telnet
-f <filename>      = command file, defaults to STDIN
```



cisco.exp.joblist.sh
Run to apply the configurations done in
commands.txt
to the devices from
Devices.txt



Configures/show
outputs of the
selected devices

Generates output per device

ch-switch-01.output.txt

ch-switch-02.output.txt

br-switch-01.output.txt

Step 1

Download [2012.12.10_expect.script.v01.zip](#)

Step 2

Extract contents of zip file into your preferred folder, in my case I had placed it into:

```
mkdir /home/USERNAME/expect/  
  
/home/USERNAME/expect/
```

Step 3

Adjust the "job-creator-script.sh" file according to your needs. The username/password will be the one sent out via SSH/Telnet. YES, the password is clear text in this file!!

```
# AUSWAHL der DEVICE Liste:  
pathDEVICElist="/home/USERNAME/expect.script/Devices.txt"  
exp_username="USERNAME"  
exp_command_list="commands.txt"  
exp_password="DEMO-ONLY"  
output_dir="/home/USERNAME/expect.script/output"
```

Step 4

```
chmod +x job-creator-script.sh  
chmod +x expect.script.sh
```

You will have to create the output folder:

```
mkdir /home/USERNAME/expect/output/
```

Step 5

Ensure you have the expect module installed on your favorite linux version.

Example for Debian: `apt-get install expect`

Step 6

Edit the Devices.txt file according to your needs, by either using fixed IP addresses or DNS names.

```
Devices.txt  
-----  
Switch-1  
Switch-2  
192.168.7.22  
10.1.2.3  
Switch-4  
CONFIG-ENDE  
-----
```

Ensure you have "CONFIG_ENDE" as your last line. I used that as exit criteria out within the script. Yep, crappy programming, but the crap works and is easy to use!

Step 7

Edit the commands.txt file, that will be the commands sent out to the devices.

I would recommend that you initially send out a couple of show commands until you are familiar with those scripts before you do some conf t stuff.

```
commands.txt  
-----  
  
! -----  
show ip interface brief  
! -----  
show ip arp  
! -----  
sh ip route 0.0.0.0  
! -----  
  
-----
```

Step 8

Ensure you are located in the correct folder:

```
cd /home/USERNAME/expect
```

Now run the script, which will create the cisco.exp.joblist.sh file, and in the same moment lists the commands which you are about to send out to the selected list of router or switches specified in the Devices.txt file.

```
./job-creator-script.sh
```

Step 9

Verify the the output before you hit the trigger and push the configuration out to all the selected devices!

Step 10

Send the config/commands to the devices:

```
./cisco.exp.joblist.sh
```

(In case you want to abort the roll-out, hit CTRL-C)

Step 11

Check the output of the session either on screen, or find the output per device in the /home/USERNAME/expect/output/ folder!

