

Script to execute CLI commands at short regular intervals

Task requirement

While performing POC tests, we usually need to capture data regularly with CLI commands. This script issues a set of commands at regular intervals.

Here we save the command set in a file named cli.txt.

```
$ cat cli.txt
show clock
show session info
show running resource-monitor second last 10
show counter global filter delta yes
debug dataplane pool statistics
show running ippool
show system resources | match ': '
debug log-receiver statistics | match 'Log '
show clock
$
```

It also requests TSF to be generated at the end, and log the whole session in a file cli-*.log

The script

The script content is as follows. You may want to customize it before use, including change of the following variables.

uname - the login name "admin"
pass - the login password "password"
host - skipped, the host taken from the command line input
prompt - skipped, CLI command prompt
cli_file - name of the CLI file

Script to execute CLI commands at short regular intervals

time_delay - initial delay in seconds, before the command set to be issued

time_interval - time interval in seconds, between the command set to be issued

intervals - number of intervals, iterations

```
$ cat pancli.exp
#!/usr/bin/expect -f

#set uname [lindex $argv 0]
#set pass [lindex $argv 1]
#set host [lindex $argv 2]

set uname "admin"
set pass "admin"
#set pass "POCpoc123_"

set host [lindex $argv 0]
set prompt "> $"

set cli_file "cli.txt"

set time_delay 0
set time_interval 20
set intervals 18

#
#
#

log_user 1

set mypid [pid]
#log_file cli.log
log_file -noappend "cli-$mypid.log"
```

```
spawn /usr/bin/ssh -l $uname $host
```

```
expect {  
    "Password: " {  
        sleep 1  
        send "$pass\r"  
    } timeout {send_user "error"}  
}
```

```
set timeout 1
```

```
expect -re $prompt  
send "set cli pager off\r"  
expect -re $prompt  
send "show system info\r"
```

```
sleep $time_delay
```

```
set count $intervals
```

```
set f [open "$cli_file"]  
set commands [split [read $f] "\n"]  
close $f
```

```
while {$count > 0} {  
    foreach command $commands {  
        expect -re $prompt  
        send "$command\r"  
    }  
}
```

```
set count [expr $count-1]
```

```
sleep $time_interval
```

```
}
```

Script to execute CLI commands at short regular intervals

```
#expect -re $prompt
#send "request tech-support dump\r"
expect -re $prompt
send "exit\r"

#
#
#

puts "CLI session captured and saved in cli-$mypid.log"

exit

# End
$
```

Script environment

This is an Expect script, which runs on Linux. All you need is a CentOS box with Expect installed.

Use this command to install Expect.

```
$ yum install expect
```

Run the script with this syntax.

```
$ ./pancli.exp 10.10.1.123
```

where admin/password is the login on PA and 10.10.1.123 is the management IP.

Script to execute CLI commands at short regular intervals

You may also run the script in background, and detach it from the connected terminal. This can prevent the script from being killed when you are disconnected.

```
$ nohup pancli.exp 10.10.1.123 &
```

To bring it back to the foreground, just enter ``fg``.

```
$ fg
```

You can stop it by pressing Ctrl-C, or kill it as a detached process with its PID.

```
$ ps -aef | grep pancli
terence 29877 29663 0 09:08 pts/0 00:00:00 /usr/bin/expect -f /home/terence/bin/
pancli.exp 10.10.1.123
terence 29896 29663 0 09:11 pts/0 00:00:00 grep --color=auto pancli
$ kill -TERM 29877
```

Notes

Some CLI commands take more time to return or just fail if PA is committing changes.

One example is ``show rule-hit-count vsys vsys-name vsys1 rule-base security rules all``

There are few options addressing the issue.

1. Adjust accordingly the timeout value in the script.

```
set timeout 10
```

2. Perform data collection more frequently.

3. If you are testing with ``commit``, move those affected CLI commands in the commit script. Both scripts can run independently at the same time.

Script to execute CLI commands at short regular intervals

Reference

- [Script to perform CLI `commit` against a list of saved config](#)
- [Script to generate PA XML config for objects, policy rules and routes](#)
- [Scripts to collect browser content or Panachrome repeatedly over time](#)