Log collection tool

songping 07/18/2009 ChongQing

Email: ping.song@ericsson.com
MSN: itestitest@hotmail.com
Skype: songpingwebphone
Yahoo: spirentping@yahoo.com
Gtalk: songpingemail@gmail.com

ERICSSON
TAKING YOU FORWARD

Log collection tool: logtool

- Agenda
  - Introductions
  - Basic usage
    - Basic uage – shell mode
    - Basic uage – CLI mode
    - Basic uage and examples
  - Input and Output
    - Files Output
    - Emails Output
    - Samples
  - Configuring this tool
    - Configuration file
  - Current limitations
  - To be added

Introduction

- Initial purpose:
  - Automate Information/log collection based on pre-defined commands
  - Long time periodically running without external intervention
  - issue monitoring & detection based on pre-defined criterions
  - Event triggered further check & actions
  - Log all events/commands with accurate timestamps

- Current features:
  - All basic functions (as initially expected) done
  - Configurable/scalable
    - Shell mode (Unix-style) commands
    - Command line options
    - Configuration file
    - Environmental variables setup
  - Running well without modification on Solaris/Windows/Linux
  - Applicable to all telnet manageable devices
    - Redback/Cisco/Extreme/Force10/Solaris/etc…

Important notes:

- This tool is provided for free
- The goal is to aid the log collection tasks in SmartEdge-involved networks
- It is under no conditions with any sense of ‘Ericsson official’ tool, hence there will be no official support for it at this stage either.
- In theory there should be no harm for the devices or/and networks under check by this tool (because of the “show” command nature), but this is not under ‘absolute guarantee’ (if there is such thing ever☺). Simply don’t use it if you don’t trust it.
- hence any impact/issues caused by it should not go to the author :D

Basic usage

- The tool now can be running with 2 different methods:
  - Shell mode: Setup parameters then run it afterward, under a UNIX-style shell
    - "logtool –shell" to enter shell mode
    - Now setup options ("host MYHOST", etc)
    - Double check it ("show current [host]", etc)
    - Run it ("run")
    - Unix style:
      - Double Tab for some hints/prompts
      - Single Tab to complete current name
      - ?/help To ask for help
      - …
  - CLI options mode: setup parameters and run all at one go, with a single command line
    - Logtool –optionsA .. –optionB.. –optionC ...

Basic usage

- E:\perl2\logtool>perl logtool.pl
- usage:
- #help and usage
- logtool -help #more detail instructions
- logtool -usage #this message, brief usage

- #run in shell mode (unix style,more friendly,recommended mode)
- #set up options,review or modify it,then run it
- logtool -shell [OPTIONS]

- #run in CLI option mode
- #provide all options and run it all in one go
- logtool [-online] [-offline|noonline] [-rounds ROUNDS]
- [-sleep SECONDS] [-timing local|remote]
- [-debug 0|1|2|3] [-quiet]
- [-ziplog] [-size2zip BYTES][-log_dir LOGDIR]
- [-configlog_file CONFIGLOGFILENAME]
- [-checkonhit_file CHECKONHITFILENAME]
- [-cmdlog_file CMDLOGFILENAME] [-dotrue] [-dofalse]
- [-sendemail] [-smtpserver SMTPSERVER] [-emailfrom EMAILFROM]
- [-emailfakefrom EMAILFAKEFROM] [-emailto EMAILTO]
- [-emailreplyto EMAILREPLYTO]
- [-emailsubj EMAILSUBJECT] [-emailmax EMAILMAX]
- [-clock_cmd CLOCK_CMD] [-times TIMES]
- [-prompt PROMPT] [-init_prompt INIT_PROMPT]
- [-telnet_timeout ROUNDS] [-telnet_buffer TELNET_BUFFER]
- [-repeat] [-norepeat] [-history_shell HISTORY_SHELLFILE]
- -host HOSTNAME

Basic usage – shell mode

```
E:\perl2\logtool>perl logtool.pl -shell

(you are now running under unix-style shell mode)


logtool [4] >>
cmdlog_file     size2zip        emailfrom       checklog_file   ziplog
exit            stat            bye             telnet buffer   online
rename          sendemail       emailmax        smtpserver      timing
emailto         ?               emailfakefrom   help            reset_all
view            pwd             config_file     checkonhit_file times
debug           emailpref       ls              usage           clock_cmd
logout          delete          dir             debug_complete  dofalse
list            rounds          cli_prompt      quit            sleep
dotrue          repeat          log_dir         cd
show            init prompt     cat             emailsubj
emailreplyto    telnet_timeout  run             host
```

Basic usage – shell mode

```
logtool [6] >> host cqlab
value is set to:cqlab

logtool [8] >> online 1
value is set to:1

logtool [13] >> show current
Current settings are:
%current = (
             'cmdlog_file' => 'show tech.txt',
             'emailto' => 'ericsson team <itestitest@hotmail.com>',
             'debug' => 0,
             'emailreplyto' => 'ericsson team <ping.song@ericsson.com>',
             'size2zip' => '1000',
             'log_dir_ori' => 'CMCC-monitor%t',
             'sendemail' => 0,
             'emailpref' => 'the issue were detected,for more info please see at
tachments.
 info collected at the failure point:


,
 ,
             'rounds' => '10000',
             'repeat' => 1,
             'init_prompt' => '/[\\$%#>]\\ ?$|login: $|password: $|username: $/i
,
 ,
             'telnet_timeout' => 60,
```

Basic usage – shell mode

```
logtool [17] >> show current host

host is: cqlab

logtool [19] >> show default host
no default value for this option!


logtool [21] >> run

PREHANDLE:##############preparing dir and files for logs and reports####################

Prehandle: backed up log_dir name base: CMCC-monitor%t
PREHANDLE: backed up checklog file name base: checklogfile%h%t.txt
PREHANDLE: backed up checklog_file name base: found%h%t.txt
PREHANDLE: Created a dir CMCC-monitor Sat Jul 18 21 10 43 2009...
PREHANDLE: Enter dir CMCC-monitor_Sat_Jul_18_21_10_43_2009...
PREHANDLE: get a file name for checklog_file base: found%h%t.txt

PREHANDLE: online set,login ...

LOGIN: create an initial input logs file (local timestamp only!): checklogfile_c
qlab_Sat_Jul_18_21_10_43_2009.txt
LOGIN: connecting to 10.190.3.220...
```

Basic usage – shell mode

```
logtool [23] >> ?
                cd -- Change to directory DIR
       checklog_file -- all checking logs
     checkonhit file -- checking logs when hitting the issue
          cli_prompt -- cli_prompt used by telnet to chunk the cmdoutput
           clock_cmd -- cmd used to get the current clock on the remote device u
nder check (usually [show clock|date])
         cmdlog_file -- name of log file to be checked
         config_file -- configuration file name
               debug -- debug mode:[0|1|2|3]
      debug_complete -- Turn on completion debugging
              delete -- Delete FILEs
            dofalse -- forcing the dofalse statement:[0|1]
             dotrue -- forcing the dotrue statement:[0|1]
       emailfakefrom -- fake emailfrom:[e.g.:routermonitor@ericsson.com]
           emailfrom -- emailfrom:[e.g.:pingsong@redback.com ]
            emailmax -- max. size of email text:[0..N]
           emailpref -- some headword in the email:[there is a link down issue,f
or more info please see attachment...]
        emailreplyto -- email reply to:[e.g.:pingsong@ericsson.com]
           emailsubj -- email subject:[emergency issue:link is down!]
             emailto -- send email to:[e.g.:wang@cmcc.com]
                help -- Print helpful information
                host -- name of host to be checked
```

Basic usage – shell mode

```
        init_prompt -- prompt used by telnet to input username/password
               list -- List files in DIRs
            log_dir -- dir for all checking logs
             online -- online mode:[0|1]
                pwd -- Print the current working directory
               quit -- Quit this program
             rename -- Rename FILE to NEWNAME
             repeat -- repeat mode:[0|1]
          reset_all -- Resetting all parameters to defaults
             rounds -- rounds of checks:[1..N]
                run -- run the checking
          sendemail -- send email once detecting any issues:[0|1]
               show -- checking settings
            size2zip -- file size max limit before get zipped
              sleep -- sleep interval between each round:[0..N]
         smtpserver -- smtpserver:[e.g.: smtp.redback.com]
               stat -- Print out statistics on FILEs
       telnet buffer -- telnet buffer in Megabits:[1..N]
     telnet_timeout -- telnet timeout in second:[1..N]
              times -- times of a cmd to repeat before execute the next one:[0.
.N]
             timing -- time used in the log:[local|remote]
              usage -- usage
               view -- View the contents of FILEs
             ziplog -- zip the log
```

Basic usage – CLI mode

- logtool [-online] [-offline|noonline] [-rounds ROUNDS]
- [-sleep SECONDS] [-timing local|remote]
- [-debug 0|1|2|3] [-quiet]
- [-ziplog] [-size2zip BYTES][-log_dir LOGDIR]
- [-configlog_file CONFIGLOGFILENAME]
- [-checkonhit_file CHECKONHITFILENAME]
- [-cmdlog_file CMDLOGFILENAME] [-dotrue] [-dofalse]
- [-sendemail] [-smtpserver SMTPSERVER] [-emailfrom EMAILFROM]
- [-emailfakefrom EMAILFAKEFROM] [-emailto EMAILTO]
- [-emailreplyto EMAILREPLYTO]
- [-emailsubj EMAILSUBJECT] [-emailmax EMAILMAX]
- [-clock_cmd CLOCK_CMD] [-times TIMES]
- [-prompt PROMPT] [-init_prompt INIT_PROMPT]
- [-telnet_timeout ROUNDS] [-telnet_buffer TELNET_BUFFER]
- [-repeat] [-norepeat] [-history_shell HISTORY_SHELLFILE]
- -host HOSTNAME

Basic examples – CLI mode

- Example 1: online mode
  - #remotecheck –online –host BKP-RI
  - #remotecheck –online –host BDK-RII
  - This will run the check for some rounds, halt for a period of time between each rounds, create a directory and put all log files in it
  - All options/variable are configurable:
    - Rounds, inter-rounds interval, directory name, log file name, etc.
    - Configured via different method:
      - CLI, configuration files, environment var, or default values

Basic examples – CLI mode (cont.)

- Example 2: control everything via CLI
  - #remotecheck –online –host BKP-RI [-rounds 100000] [-sleep 5][-log_dir singtel_checking][-checklog_file mylog%h%t.txt –hit logonhit%h.txt]
  - This will run the check:
    - for 100,000 rounds,
    - sleep 5 seconds between each round,
    - put all logs into a directory ‘singtel_checking’ (will be created if not yet existing, otherwise not),
    - checking log file is named mylog_BKP-RI_CURRENTLOCALTIME.txt
    - When the monitored issue is detected, put it on a separate file named logonhit_BKP-RI.txt

Basic examples – CLI mode (cont.)

- Example3: debug mode
  - #remotecheck –online –host BKP-RI –rounds 1 –debug 3 > debug.txt
  - Debug mode. This will collect all information about the script running status, send this when it is not functioning as expected
- Example4: quiet mode
  - #remotecheck –online –host BKP-RI –quiet
  - This will generate minimum output about the script itself
- Example5: using time of remote device (remote mode)
  - #remotecheck –online –host BKP-RI –timing remote
  - This will timestamp each command in the log using the clock in remote device (by default use time of local machine)
  - Useful when remote device has a different timezone with local machine

More usage – CLI mode

- Example6: offline mode
  - #remotecheck –offline cmdlog mylog.txt –host BKP-RI
  - This will check log file and see if you can find some thing you are looking for
  - Useful on large logfiles (say, a file larger than 100M)
  - Mostly for test only

Input and output of this tool

- Input:
  - A configuration file, containing all hosts login information and batches of commands that will be executed on remote machine(s)
  - By default the script seek for a file NAME.cfg while ‘NAME’ is the same as script name, configurable by “-config_file” option
  - May contain a bit complex syntax
- Output:
  - Log files recording all commands and their corresponding outputs during the login period. Log files will be put in a DIR.
  - Name of all files and DIR can be tagged with the hostname (%h) under check, and/or timestamps (%t) of the logging moments.
  - The timestamps (%t) can be based on either local machine (“-timing local”) or remote machine (“-timing remote”), depending on configurations)
  - Real time running status, by default to STDOUT

Outputs: directories and files

- Files/directories:
  - A directory holding all log files, can be configured by option:

    -log_dir mylogdir%h%t

  - A Log file that records every commands ever been checked, and all corresponding outputs, in the raw format

    File name is configurable by option:

    -checklog_file mylog%h%t.txt

  - A log file, records only commands and outputs when the issues under monitoring (triggers) were just hit, and the corresponding further checking (actions) right after that, in a formatted form.

    file name configurable by option:

    -checkonhit_file myissuelog%h%t.txt

Outputs: compress big log files

- Long time periodical checking may generate huge and growing log files that sometimes overuse disk spaces
- By default, all files, when growing bigger than 10M Bytes, will be truncated and zipped, periodically, in flavor of saving disk spaces
  - Controlled by “–ziplog” flag
  - configurable via “–size2zip BYTES” option
- In the case you don’t want all log stored in plain text format, use “–nozip” option

Outputs: email

- If option –sendemail is set, the script will prepare an email and send via SMTP protocol:
  - Use the following options to set email facility:
    - Smtpserver/emailfrom/emailfakefrom/emailto/emailreplyto
  - Use the following options to setup email title/subjects/headwords:
    - emailsubj/emailpref/emailmax
  - brief the issue it detected based on those commands that hit the issue
  - Attach the issue problem log file: checkonhit_file
  - Attach the all check log file: checklog_file

- CQCMCC network does not open SMTP port, so no use on their network

Outputs samples: ‘checklog’ files (before zipped)

- bash-3.2$ cd singtel-monitor/
- bash-3.2$ ls
- checklogfile_bkp-ri_Thu_May_28_17_06_18_2009.txt
- checklogfile_bkp-ri_Tue_May_26_17_21_37_2009.txt
- checklogfile_bkp-ri_Tue_May_26_18_05_26_2009.txt
- checklogfile_bkp-ri_Tue_May_26_18_06_15_2009.txt
- checklogfile_bkp-ri_Tue_May_26_18_06_55_2009.txt
- checklogfile_bkp-ri_Tue_May_26_18_07_45_2009.txt
- checklogfilebkp-ri_Tue_May_26_15_06_35_2009.txt
- checklogfilebkp-ri_Tue_May_26_15_09_47_2009.txt
- checklogfilebkp-ri_Tue_May_26_17_14_57_2009.txt
- found_bkp-ri_Fri_May_29_11_39_30_2009.txt
- found_bkp-ri_Fri_May_29_11_49_58_2009.txt
- found_bkp-ri_Fri_May_29_11_53_49_2009.txt

Outputs sample : ‘checkonhit’ files

- ==========
- issue was hit (the 1th time) at the following commands
- ==========
- 1) show port detail (at Mon May 25 15:19:15 2009)
- ----------
- ethernet 1/1 state is Up
- Description : Connection to BD03M-BP03B DARK-01 Dist Sw MegaPop LREF DN0574089
- Line state : Up
- Admin state : Up
- ……
- 2) show arp (at Mon May 25 15:19:16 2009)
- ----------
- Total number of arp entries in cache: 38
- Resolved entry : 38
- Incomplete entry : 0
- Host Hardware address Ttl Type Circuit
- 10.251.130.33 00:17:df:ec:cc:00 2598 ARPA 6/2 vlan-id 123
- 10.251.130.34 00:00:5e:00:01:02 - ARPA 6/2 vlan-id 123
- ……

Outputs sample : checklog files

- root@ems # less checklogfile_bkp-ri_Fri_May_29_04_07_21_2009.txt
- login: root
- Password:
- Last login: Fri May 29 04:06:44 from 10.252.75.212
- Sun Microsystems Inc. SunOS 5.10 Generic January 2005

- Unauthorised access is prohibited
- Sourcing //.profile-EIS.....
- root@ems # telnet 10.251.130.50
- Trying 10.251.130.50...
- Connected to 10.251.130.50.
- Escape character is '^]'.

- BKP_RAN_R-I
- ^Mlogin: redback
- Password:
- [local]BKP_RAN_R-I>enable
- login: root
- Password:
- Last login: Fri May 29 04:06:44 from 10.252.75.212
- Sun Microsystems Inc. SunOS 5.10 Generic January 2005

- Unauthorised access is prohibited
- Sourcing //.profile-EIS.....
- root@ems # telnet 10.251.130.50
- Trying 10.251.130.50...
- Connected to 10.251.130.50.
- Escape character is '^]'.

- BKP_RAN_R-I
- ^Mlogin: redback
- Password:
- [local]BKP_RAN_R-I>enable
- Password:
- [local]BKP_RAN_R-I#term len 0

- [local]BKP_RAN_R-I#show clock
- Fri May 29 04:08:06 2009 GMT
- [local]BKP_RAN_R-I#pwd

- [local]BKP_RAN_R-I#pwd
- /flash
- [local]BKP_RAN_R-I#show version

- Redback Networks SmartEdge OS Version SEOS-6.1.3.5p1-Release
- Built by sysbuild@lx-dev3 Mon Mar 16 09:23:22 PDT 2009
- Copyright (C) 1998-2009, Redback Networks Inc. All rights reserved.
- System Bootstrap version is Mips,rev2.0.2.22
- Installed minikernel version is 11.7
- Router Up Time - 65 days, 10 hours 28 minutes 12 secs
- [local]BKP_RAN_R-I#show clock
- Fri May 29 04:08:07 2009 GMT
- [local]BKP_RAN_R-I#show vrrp stat

- --- VRRP Virtual Router eth-3/1.157/2 (Backup) ---

- Master Transitions: 5
- Advertisement Recv: 0 Advertisement Sent: 0
- Priority 0 Recv : 0 Priority 0 Sent : 0
- Bad Type Errors : 0 Wrong Owner Errors: 0

Outputs sample : email

- ----------
- From: <routermonitor@ericsson.com>
- Sent: Monday, May 25, 2009 1:19 AM
- To: "ericsson team" <itestitest@hotmail.com>
- Subject: ping issue and vrrp state change report

- > the issue were detected,for more info please see attachments.
- > info collected at the failure point:
- >
- >
- >
- >
  ==========
- > issue was hit (the 2th time) at the following commands
- >
  ==========
- > show arp
- > ping 10.251.137.199 20 flood
- > ping 10.251.133.199 20 flood
- > ping 10.251.135.199 20 flood
- > ping 10.251.139.199 20 flood
- > show vrrp

Outputs sample : email (cont.)

- > 1) show arp (at Mon May 25 01:19:26 2009)
- > ----------
- > Total number of arp entries in cache: 4
- > Resolved entry : 4
- > Incomplete entry : 0
- >
- > Host Hardware address Ttl Type Circuit
- > 20.0.0.1 00:30:88:12:b9:68 - ARPA 1/3
- > 20.0.0.2 00:10:94:00:00:01 1889 ARPA 1/3
- > 192.168.251.141 00:30:88:12:b9:69 - ARPA 1/4
- > 192.168.251.142 00:30:88:11:b7:5f 1282 ARPA 1/4
- >
- >
- >
- >
- > 2) ping 10.251.137.199 20 flood (at Mon May 25 01:19:26 2009)
- > --
- > .....snipped.....
- >
- > Address List:
- > 10.251.131.33
- ……

Outputs sample : email (cont.)

- > ==========
- > following commands were checked right after the failure was detected
- > ==========
- > show ism client arp log det cct handle 3/1:1023:63/1/2/13
- > show ism client arp log det cct handle 4/1:1023:63/1/2/17
- > show ism client arp log det cct handle 2/1:1023:63/1/2/9
- > show clock
- > show ism client arp log det cct handle 1/1:1023:63/1/2/5
- > show arp all
- > !!!!ping failure detected!!!!
- > show arp interface
- > show arp circuit

- > thanks
- > regards
- > from remotecheck script

Outputs: others

- Others:
  - Running status of the script itself

    By default it writes to standard output, can be redirected by "|"

    amount of output can be tuned by option:

    -debug 0~3

    -quiet = -debug 0

Configuring the script

- Currently the options can be configured via the following methods, with a descending privilege:
  - Shell mode
    - Highest priority, override all other sources of options
    - More friendly, interactive user interface, “setup-and-run” model, Recommended working mode
  - CLI options
    - Override all other sources of options (except shell mode)
    - Useful when used in other script or UNIX shell
  - Configuration file
    - Once get proper values for your network, put them all in configuration file, no need CLI options anymore
  - Environmental
    - Export rounds=100
  - Default values
    - logtool [8] >> show default config_file
    - config_file defaults to: logtool.cfg

Options: Basic

- Usage:
- -help :print this usage
-
- **COMMONLY USED OPTIONS:**
- -online -(host|hostname) <name>
- :online mode switch
- :host name must exist in cfg file
- -[offline|noonline] -cmdlog_file <FILE>
- :offline mode, default
- :cmdlog_file - operation log file as input
- for analysis under offline mode, def: showtech.
- -rounds <10> :run 10 rounds of commands blocks and stop,default 3
- -sleep <15>s :sleep for 15s between each rounds, def 10s
- -nosleep :turn off sleep, run as quick as possible
- -debug :debug mode: 0/1/2/3, def 1
- -nodebug|quiet :turn off debug mode, generate minimum output (same as -deb 0)
-

options: more

- **SOME USEFULL OPTIONS**:
- -(config_file|cfg) <FILE>
- :configuration file, default: name-of-script.cfg
- -log_dir <FILE> :directory for log files, def: current dir
- -checklog_file <FILE>:log file for all checkings and outputs,
- default: remotechecklog%h%t.txt
- -(checkonhit_file|hit) <FILE>
- -prompt <PROMT> :
- -timing <remote|local> :specify the source of timestamps
- :from local machine or from remote device
- -clockcmd <CMD> :specify the command used to get remote clock
- :”show clock” in smartedge and a lot of other devices
-

Options:Email

- **EMAIL ALARTING OPTIONS:**
- -sendemail :send email switch,by default not send email
- -nosendemail :not to send email
- -smtpserver :smtpserver, def smtp.sina.com.cn
- -emailfrom :from which emailbox, def routermonitor@sina.com
- -emailfakefrom :fake email address, def routermonitor@ericsson.com
- -emailto :email to, def routermonitor@sina.com
- -emailreplyto :reply to, def routermonitor@sina.com
- -emailsubj :email subject,default: failure report
- -emailpref :some head words in email,
- defalt: the issue were detected,for more info ...
- -emailmax :max charactors put in email text

Options: Even more …

- **RARELY USED OPTIONS:**
- -repeat [-times <2>]: repeat mode, repeat each cmd for 2 times before the next one, times default to 1
- -norepeat :norepeat mode,default execute each cmd only once
- -(checkonhit_file|hit) <FILE> :log file that records only the round of commands hitting the issue and corresponding actions taken afterward,
- default checkonhit+timestamp.txt

- -dotrue :force the program to check (dotrue) batch,
- :bypassing any (check) batch analysis, mostly for diagnostics
- -dofalse :force the program to check (dofalse) batch,
- :bypassing any (check) batch analysis, mostly for diagnostics

Configuration file

- Clause based configuration file
  - <GlobalSettings>
    - Global options for all hosts
    - rounds, sleep, checklog_file, checkonhit_file, debug, email setups,etc…
  - </GlobalSettings>
  - <LoginInfo>
    - Login information for different hosts to be checked
  - </LoginInfo>
  - <Data HOSTNAME>
    - commands you want to execute, in what manner, how to define an ‘issue’ or issues, what to do when an issue (or issues) is/are hit
    - support multiple hosts in a configuration file
  - </Data HOSTNAME >
  - Support comments
    - any line preceding with“#” will be ignored
  - 2nd priority in the configuration methods
    - CLI options→configuration file→Environment var→default

Configuration file: <globalsettings>

```
<GlobalSettings>
        online          0           #will telnet online and check
#       repeat          on          #will repeat each cmd,..
#       times           1           #...for how many times before next one
        sleep           5           #5s interval bet. each rounds
        rounds          10000       #run 10000 rds, continuously
        debug           1           #less output on screen
        log_dir         singtel-monitor         #dir for logs
        checklog_file   checklogfile%h%t.txt    #log file name
                                    #h wil be replaced by host name (i.e. BKP-RI)
                                    #t will be replaced by a timestamp :
                                    #       Tue_May_26_17_14_57_2009
        checkonhit_file found%h%t.tx
                                    #log file, only record issues found
#       sendemail       on          #send email report, and settings follow
          smtpserver    smtp.sina.com.cn
          emailfrom     routermonitor <routermonitor@sina.com>
          emailfakefrom routermonitor@ericsson.com
          emailto       ericsson team <itestitest@hotmail.com>
          emailreplyto  ericsson team <ping.song@ericsson.com>
          emailsubj     ping issue and vrrp state change report
</GlobalSettings>
```

Configuration files: <logininfo>

- <LoginInfo>
  - BKP-RI t1:10.252.75.22 u1:root p1:root t2:10.251.130.50 u2:redback p2:redback e2:redback
  - BKP-RII t1:10.252.75.22 u1:root p1:root t2:10.251.130.51 u2:redback p2:redback e2:redback
  - BDK-RI t1:10.252.75.22 u1:root p1:root t2:10.251.130.18 u2:redback p2:redback e2:redback
  - BDK-RII t1:10.252.75.22 u1:root p1:root t2:10.251.130.19 u2:redback p2:redback e2:redback
- </LoginInfo>
- Currently supported parameters:
  - T1/T2:telnet level1/2
  - U1/U2:username level 1/2
  - P1/P2:password level 1/2
  - E1/E2:enable password level 1/2
  - “t1:10.252.75.22 u1:root p1:root” : this is EMS login
  - “t2:10.251.130.50 u2:redback p2:redback e2:redback”: this is Redback login
- Don’t support SSH yet

Configuration files: <data HOST>

- The most complex clause
- Define key ‘checking’ and ‘action’s pairs, and execute in a round loop:
  - what CLIes to be executed , outputs of which will be logged
  - What data is the interested part in the CLI output, capture them
  - save captured data into named variable, for later analysis
  - Calculate if the issue is ‘seen’ (true) or not (false) by logic operations
  - Based on hit or not, decide what to do next
  - Go to execute a branch (“dotrue”) if we are hit (true)
  - Go to execute another branch (“dofalse”) if nothing happened (false)

<data HOST>: items

- Currently supported Items:
  - Precheck
    - Do some pre-checking, before entering the round loop
    - Typically general check/log like: show clock, terminal length 0, show port, show crash, etc
  - Check
    - Check task in the round loop
    - Capture data using regular expressions
    - Based on the check result, different actions will be taken
  - Checkstate/Checkchange/checkfinal
    - ‘calculate’ overall we are hit/‘true’ or not hit/‘false’
    - after we capture data by check statements, how then to determine the issue of concern has appeared or not. and then go to different action branch based on it
  - Dotrue/Dofalse
    - When checkfinal result is true (we are hit), execute commands defined in dotrue ; otherwise dofalse
  - Postcheck
    - Some post-checking when all loop of checking are done

<data HOST>: supported formats

- Currently 3 statement formats are supported
  - Precheck1210 show_ver = {show version}

  A prefix: precheck(keyword)digits blanks

  var name = {CLI }

  (store the captured data)

  Here we capture all output of 'show version'
  - Check1450 ping_result1={ping 10.251.133.199 20 flood}:{(\.\.\.\.\.)}

    var name = {CLI }:{regular expressions}

    like UNIX grep

    Here we capture only the "….." (5 packet loss) among ping outputs
  - Check1450 slot7_sn={show hardware detail}:{Slot\s+: 7\s+Type\s+: xcrp[34].*\nSerial No\s+: }:{ Hardware Rev\s+: \d\d -\s+\n}

    var name = {CLI }:{regular expressions1}:{regular expressions2}

    Here we want to capture anything between the 2 regexes.

<data HOST> ‘precheck’ item (Singtel)

- #some pre-checking
- precheck1210 show_ver={show version}
- precheck1220 show_clock={show clock}
- precheck1230 vrrp_trans={show vrrp stat}
- These will do the following pre-checkings before entering the round loop:
  - capture output of these commands,
  - save them into following named variables respectively:
    - show_ver,
    - show_clock and
    - vrrp_trans,

<data HOST>:'Check' item (singtel)

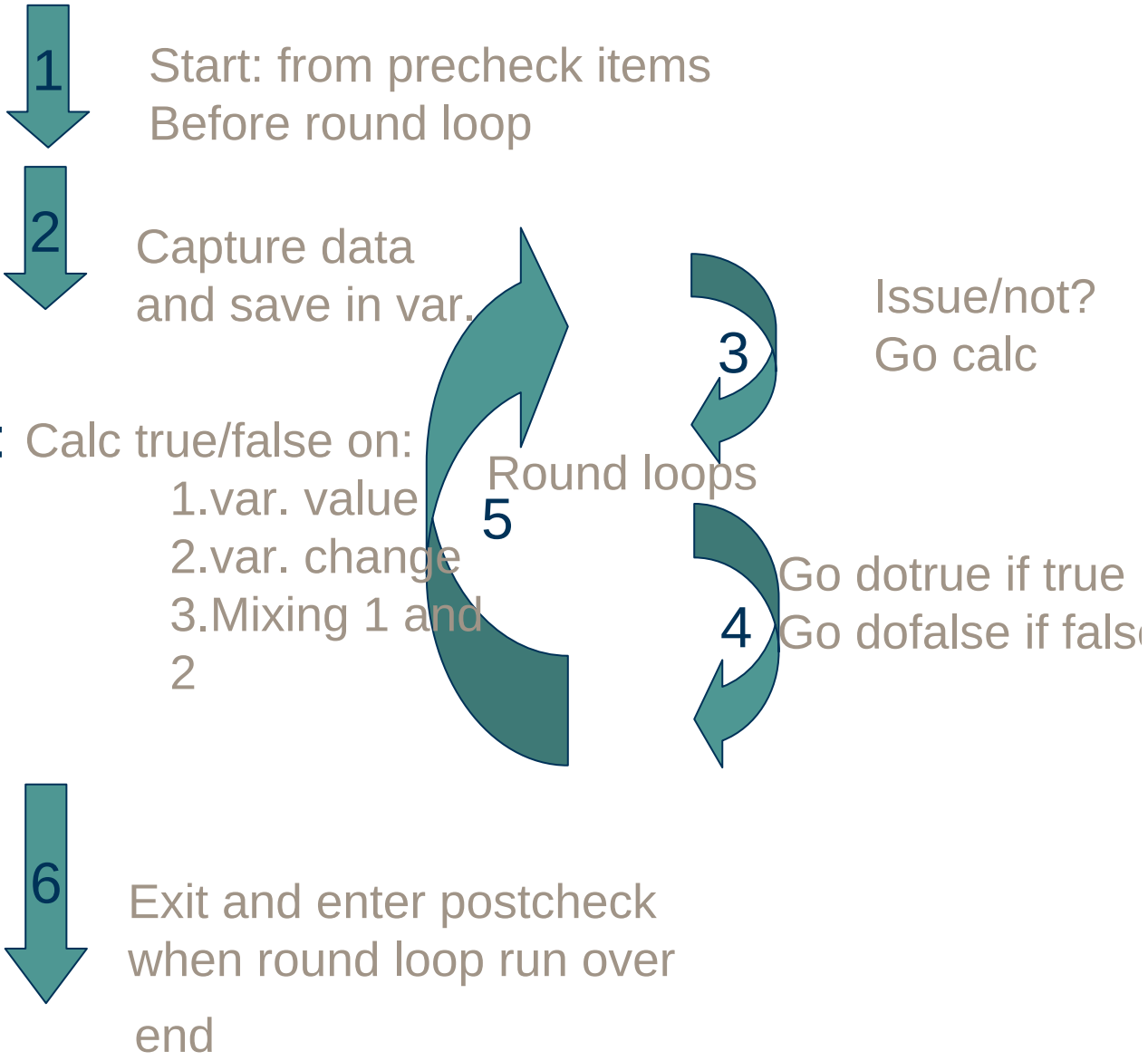
- ping monitor
  - check1450 ping_result1={ping 10.251.133.199 20 flood}:{(\.\.\.\.\.)}
  - check1460 ping_result2={ping 10.251.135.199 20 flood}:{(\.\.\.\.\.)}
  - check1470 ping_result3={ping 10.251.137.199 20 flood}:{(\.\.\.\.\.)}
  - check1480 ping_result4={ping 10.251.139.199 20 flood}:{(\.\.\.\.\.)}
  - These are to capture the situation when there are 5 consecutive packets loss
- VRRP state monitor
  - check1500 show_vrrp={show vrrp}:{state.*?\n}
  - This is to capture any VRRP state
- ARP table monitor
  - check1490 show_arp={show arp}:{.*(10.251.132.201).*(10.251.134.201).*(10.251.138.201)}
  - This is to check if our own IP entry show up in ARP table
- Port state monitor
  - check1600 all_port_up={show port}:{(1\/1\s+\w+\s+Up)\s+(2\/1\s+\w+\s+Up)\s+(4\/1\s+\w+\s+Up)}
  - This is to check the 3 ports are up/down

Issue definition

- how to make the script detect and ‘keep in mind’ a particular issue that is under my concern?
  - example1: I’m only interested in the ping loss to dest IP 1.1.1.1 only when ping 2.2.2.2 is FINE. If both fails I don’t care.
  - Example2: I’m only interested in the VRRP state CHANGE, means I think there must be a problem when the router transit from master → backup or init, or from backup → master or init, etc. but I don’t really care it’s state currently is master/backup or init.
- This is done by issue definition statements:
  - Checkstate/checkchange/checkfinal

issue Determination logic/flow

- Precheck
- Check
- Issue Calculation:
  - Checkchange
  - Checkstate
  - Checkfinal
- Dotrue
- dofalse
- postcheck



issue determination: 3 steps calculations

- support issue definitions under mixing/complex criterions
  - Fully support 3 basic logical operator: AND OR NOT
- Checkstate statement:
  - to calculate issue based variables which stores captured data
  - True if wanted data is captured
  - False otherwise
  - useful when an issue can only be defined by many different components
- Checkchange statement:
  - To calcualate issue based on any detected ‘changes’ of the variable, comparing with previous captured values in it.
  - True if it is changing
  - False otherwise
  - Useful to detect issue defined by any network change
- Checkfinal statement:
  - To combine checkstate & checkchange with logical operations

issue determination example: step 1-checkstate

- Statement in singtel:
  - checkstate ((ping_result1 or ping_result2 or ping_result3 or ping_result4) and (not show_arp) and all_port_up)
- Consider the issue are hit (to be true), only if:
  - any of the 4 defined ping fails,
  - AND
  - arp table has a problem (NOT able to capture wanted entries),
  - AND
  - While all monitored ports are still up
  - This is a monitored issue under EV118251
- B.T.W here we think it's not an issue if ping has packet loss, when port is in down status (maybe maintenance activities)

issue determination: step 2-checkchange

- Statement in singtel:
  - checkchange show_vrrp or all_port_up
- Consider the issue are hit (true), if:
  - VRRP state “changed” bet. 2 consecutive checking
  - OR
  - Port up/down state “changed” (i.e. there was a link flap)

issue determination: step3-finalcheck

- Statement is Singtel:
  - checkfinal (checkstate or checkchange)
- Eventually consider the issue are hit (to be true), if either:
  - checkstate is true (means the ARP issue EV118251 is surfacing)
  - OR
  - Checkchange is true (means if there is link flap or vrrp state transition)

issue determination: some defaults

- When checkfinal is not configured
  - by default do OR:
  - Checkstate OR checkchange
- When checkstate/checkchange/checkfinal are all absent,
  - by default use OR on all check items
  - should we be able to capture anything, we think it’s a hit!

Actions

- Once well-defined an issue, we can use the script to keep monitoring (true or false)
- Once script find a situation matching all defined criterion (true) in a round, actions must be taken (dotrue) before we go for next round check
  - Collect specific information (for DE/TAC)
  - Do something as planned:
    - Service recovery
    - shutdown/reload/switchover/…

Actions when issue is “seen”: dotrue

- Statements in singtel:
  - dotrue100 show_clock={show clock} #check time
  - dotrue110 show_arp_all={show arp all} #check arp table
  - dotrue120 show_arp_int={show arp interface}
  - dotrue130 show_arp_cct={show arp circuit}
  - dotrue140 show_ism_client={show ism client arp log det cct handle 1/1:1023:63/1/2/5}
  - dotrue150 show_ism_client={show ism client arp log det cct handle 2/1:1023:63/1/2/9}
  - dotrue160 show_ism_client={show ism client arp log det cct handle 3/1:1023:63/1/2/13}
  - dotrue170 show_ism_client={show ism client arp log det cct handle 4/1:1023:63/1/2/17}
- This is to check DE/TAC required info when issue EV118251 surfaces again

More Actions when issue is “seen”

- #further check path to RII
- dotrue180 ping_BKP_RII={ping 192.168.251.142 100 flood}
- dotrue185 ping_BDK_RI={ping 192.168.251.157 100 flood}
- dotrue186 ping_BDK_RII={ping 192.168.251.132 100 flood}
-
- #further check ospf/route/vrrp statistics
- dotrue190 ospf={show ospf nei}
- dotrue200 route={show ip route}
- dotrue210 show_vrrp_tran={show vrrp stat}
-
- #further port details checking(tx/rx power) if port down
- dotrue220 show_port_det={show port detail}

Actions when issue is not “seen”: dofalse

- Singtel statement:
  - dofalse100 show_clock={show clock}
- Nothing much need to be done if there is no issue detected, just go next round.

Current limitations/issues

- Need more test
- Need more bug report
- No audit codes
- No SSH support
- Some issues and workarounds:
  - telnet timeout on ‘big’ command
    - When network is slow or delay to remote device is long, telnet may timeout before it get all outputs (typically some SmartEdge macros with large amount of output, like show tech), current timeout value is 60s.
    - Workaounds: set telnet_timeout option to 120s or 150s
  - Can’t telnet successfully
    - This may caused by no proper match on the prompt (usually ‘#’ or ‘%’)
    - Workarounds: set init_prompt option to a proper string

To be added

- Multi-process support: login and monitor 50 nodes in parallel
- SNMP trigger real time checking
- Audit cpu/process/memory/dumpfile/warning/tempratures/…
- More clauses
  - <PerlHandle>
    - Customization code, still to be defined
    - One good thought is to add audit code here
  - </PerlHandle>
  - <SimpleReport>
    - Descriptions, still to be defined
  - </SimpleReport>

Any bug reports/suggestions

- Check and fix it by yourself.
  - Free soft
- Run debug mode and attach the outputs
  - #remotecheck –online –host BKP-RI –rounds 1 –debug 3 > debug.txt
- Send email to :

  Ping.song@ericsson.com


ERICSSON
TAKING YOU FORWARD