

tags: cisco

使用簡單的 BASH 腳本檔擷取設備狀態資訊

- 使用簡單的 BASH 腳本檔擷取設備狀態資訊
- 測試條件
- 目錄架構
 - 裝置清單 device_list.txt
 - 命令清單 cmd_list.txt
 - 函式庫及變數 getConfig
 - 主要執行腳本檔 getswInfo
- 執行結果
- 結論

只要透過簡單的 **BASH 腳本檔**就可以輕鬆擷取網路設備的組態檔或狀態資訊。另外可以透過 **crontab** 定期作為組態備份或狀態查詢確認使用。

由於測試的網路設備及 IOS 版本過舊，暫時就以一般 telnet 的方式進行存取。若為較新設備及安全考量，還是以 **SSH** 連線為主。

測試條件

為符合測試環境及概念性測試，僅作以下需求及腳本檔編寫目標：

- 設備清單 device_list.txt：提供連線設備的 IP 位址及登入資訊。
- 命令清單 cmd_list.txt：提供設備連線後要執行的命令集。
- 函式定義檔 getConfig：提供設備連線所需腳本執行函式功能及變數定義。
- 主要執行檔 getswInfo：主要腳本執行檔，透過組態檔及相關函式功能執行，將取得資訊儲存於 logs/{{date_time}} 目錄，並按照設備及命令分別儲存。

目錄架構

使用 `tree` 命令，顯示整體命令執行目錄架構。

```
richard@dora ~/Projects/test/getSwInfo
└─ tree
   .
   ├── config
   │   ├── cmd_list.txt
   │   ├── cmdLog_list.txt
   │   └── device_list.txt
   ├── getswInfo
   └── lib
       └── getConfig
```

裝置清單 `device_list.txt`

將欲連線設備按照以下格式完成組態輸入：

```
1  #switch ip address:telnet password:enable password
2  10.7.1.254:cisco:Cisco!
3  10.7.1.241:cisco1:Cisco1!
4  10.7.1.242:cisco12:Cisco12!
5  ...
6  (以下省略)
```

在此僅作概念性驗證，生產環境應採用 SSH 連線，或將機敏登入資訊採用安全方式儲存使用。

命令清單 `cmd_list.txt`

將要執行的設備命令逐行條列即可，若不要執行，請在行首加上 `#` 標註即可。

```
1  sh ver
2  #sh running
3  sh ip int brief
4  sh int status
5  #sh int trunk
6  #sh env all
7  #sh etherchannel summary
8  ...
9  (以下省略)
```

函式庫及變數 `getConfig`

以下為簡要範例參考。

```
1  #!/bin/bash
2  #
3  # edited by semigod
4  # 2017/12/17
5
6  function access2File () {
7      (echo "${2}";\
8      echo "enable";\
9      echo "${3}";\
10     echo "term len 0";\
11     echo "${4}";\
12     echo "exit";\
13     sleep 5) | telnet ${1} > "${5}"
14 }
15
16 ## directories
17 configDir="./config"
18 loggingDir="./logs"
19 tmpDir="./tmp"
20 timeline=`date +%F_%H-%M`
21 workingDir="${tmpDir}/${timeline}"
22 if [ ! -d ${workingDir} ]; then
23     mkdir -p ${workingDir}
24 fi
25 workingLogDir="${loggingDir}/${timeline}"
26 if [ ! -d ${workingLogDir} ]; then
27     mkdir -p ${workingLogDir}
28 fi
29
30 deviceFile="${configDir}/device_list.txt"
31 cmdFile="${configDir}/cmd_list.txt"
32 cmdLogFile="${configDir}/cmdLog_list.txt"
33 cat "${cmdFile}" | sed 's/ /\_/' > ${cmdLogFile}
34
35 function getDevice () {
36     hostIp=(`cat ${deviceFile} | sed '/^#/d' | awk -F': ' '{print $1}'`)
37     passwd=(`cat ${deviceFile} | sed '/^#/d' | awk -F': ' '{print $2}'`)
38     enPasswd=(`cat ${deviceFile} | sed '/^#/d' | awk -F': ' '{print $3}'`)
39     hostCount=`echo ${hostIp[@]}`
40 }
```

主要執行腳本檔 getswInfo

以下執行腳本檔範例。

```

1  #!/bin/bash
2  #
3  # edited by semigod
4  # 2017/12/17
5
6  #tput clear
7  b=`tput bold`
8  r=`tput setaf 9`
9  y=`tput setaf 11`
10 g=`tput setaf 10`
11 p=`tput setaf 13`
12 l=`tput setaf 14`
13 o=`tput sgr0`
14
15 ## library directory
16 libDir="./lib"
17 source ${libDir}/getConfig
18
19 oldIFS="$IFS"
20 IFS=$'\n'
21 cmdList=(`cat ${cmdFile} | sed '/^#/d'`)
22 cmdLog_list=(`cat ${cmdLogFile} | sed '/^#/d'`)
23 IFS="$oldIFS"
24 cmdListCount=`echo ${#cmdList[@]}`
25
26 ## get device list
27 getDevice
28
29 ## connect devices and get messages
30 for i in $(seq 0 `expr ${hostCount} - 1`)
31 do
32     echo -e "${g} =====${o}"
33     echo -e " 嘗試連接 ${b}${hostIp[${i}]}${o} , 請稍待!"
34     echo -e "${g} =====${o}"
35
36     if [ ! -d ${workingDir}/${hostIp[${i}]} ]; then
37         mkdir -p ${workingDir}/${hostIp[${i}]}
38     fi
39
40     if [ ! -d ${workingLogDir}/${hostIp[${i}]} ]; then
41         mkdir -p ${workingLogDir}/${hostIp[${i}]}
42     fi
43
44     for j in $(seq 0 `expr ${cmdListCount} - 1`)
45     do
46         echo " > Getting the information of \"${y}${cmdList[${j}]}${o}\""
47         access2File ${hostIp[${i}]} ${passwd[${i}]} ${enPasswd[${i}]} "${cmdList[${j}]}
48         cp "${workingDir}/${hostIp[${i}]}/${cmdList[${j}]} .txt" "${workingLogDir}/${
49         echo " >> Done!"
50     done
51 done

```

執行結果

- getswInfo 執行結果及顯示:

```
richard@dora ~/Projects/test/getSwInfo
> ./getSwInfo
```

嘗試連接 10.7.1.254，請稍待！

```
> Getting the information of "sh ver"
>> Done!
> Getting the information of "sh ip int brief"
>> Done!
> Getting the information of "sh int status"
>> Done!
```

嘗試連接 10.7.1.241，請稍待！

```
> Getting the information of "sh ver"
>> Done!
> Getting the information of "sh ip int brief"
>> Done!
> Getting the information of "sh int status"
>> Done!
```

嘗試連接 10.7.1.242，請稍待！

```
> Getting the information of "sh ver"
>> Done!
> Getting the information of "sh ip int brief"
>> Done!
> Getting the information of "sh int status"
>> Done!
```

嘗試連接 10.7.1.250，請稍待！

```
> Getting the information of "sh ver"
>> Done!
> Getting the information of "sh ip int brief"
>> Done!
> Getting the information of "sh int status"
>> Done!
```

- 紀錄儲存目錄架構：

```
richard@dora ~/Projects/test/getSwInfo
> tree ./logs
./logs
├── 2022-03-14_22-12
│   ├── 10.7.1.241
│   │   ├── sh_int_status.log
│   │   ├── sh_ip_int_brief.log
│   │   └── sh_ver.log
│   ├── 10.7.1.242
│   │   ├── sh_int_status.log
│   │   ├── sh_ip_int_brief.log
│   │   └── sh_ver.log
│   ├── 10.7.1.250
│   │   ├── sh_int_status.log
│   │   ├── sh_ip_int_brief.log
│   │   └── sh_ver.log
│   └── 10.7.1.254
│       ├── sh_int_status.log
│       ├── sh_ip_int_brief.log
│       └── sh_ver.log
```

- 顯示設備的 `sh_int_status.log` 內容。

該紀錄檔內容並未進行資料篩選，若需其他用途，可再撰寫 Parser 進行資料擷取利用。

```
richard@dora ~/Projects/test/getSwInfo
└─ more logs/2022-03-14 22-12/10.7.1.242/sh int status.log
Trying 10.7.1.242...
Connected to 10.7.1.242.
Escape character is '^['.

User Access Verification

Password:
KH-Data-C2960_1.242>enable
Password:
KH-Data-C2960_1.242#term len 0
KH-Data-C2960_1.242#sh int status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1	To_KH-Data-C3560X_	connected	trunk	a-full	a-100	10/100BaseTX
Fa0/2		notconnect	10	auto	auto	10/100BaseTX
Fa0/3		notconnect	1	auto	auto	10/100BaseTX
Fa0/4	FisherYu-1	notconnect	1	auto	auto	10/100BaseTX
Fa0/5	FisherYu	notconnect	1	auto	auto	10/100BaseTX
Fa0/6		notconnect	1	auto	auto	10/100BaseTX
Fa0/7		notconnect	1	auto	auto	10/100BaseTX
Fa0/8		notconnect	1	auto	auto	10/100BaseTX
Fa0/9		notconnect	1	auto	auto	10/100BaseTX
Fa0/10	DanielChen-NB	connected	1	a-full	a-100	10/100BaseTX
Fa0/11	GinnyWu-NB	notconnect	1	auto	auto	10/100BaseTX
Fa0/12	NicoLin-NB	notconnect	1	auto	auto	10/100BaseTX
Fa0/13		notconnect	1	auto	auto	10/100BaseTX
Fa0/14		notconnect	1	auto	auto	10/100BaseTX
Fa0/15	JASON-NB	notconnect	1	auto	auto	10/100BaseTX
Fa0/16	Yumei-NB	notconnect	1	auto	auto	10/100BaseTX
Fa0/17	EASON-NB	notconnect	1	auto	auto	10/100BaseTX
Fa0/18		notconnect	1	auto	auto	10/100BaseTX
Fa0/19	GRACELEE-NB	notconnect	1	auto	auto	10/100BaseTX
Fa0/20	Doohan-NB	notconnect	1	auto	auto	10/100BaseTX
Fa0/21		notconnect	1	auto	auto	10/100BaseTX
Fa0/22	Mendy-NB	notconnect	1	auto	auto	10/100BaseTX
Fa0/23	Print Sharp MX-M46	notconnect	1	auto	auto	10/100BaseTX
Fa0/24		connected	1	a-half	a-10	10/100BaseTX
Gi0/1		notconnect	1	auto	auto	10/100/1000BaseTX
Gi0/2		notconnect	1	auto	auto	10/100/1000BaseTX

結論

透過簡單實用，客製化的腳本檔，可用來解決常見的問題及重複的工作，無非增加了點管理及工作效率。