RouterOS Centralized Backup with PERL

Herry Darmawan
Spectrum Indonesia

How Important a Backup is?

- Backup = Insurance
  - Replace the broken one faster
  - No worries
- Configuration change-log

Concept of Centralized Backup

- Use *.backup or *.rsc files
- Collect them to one server
- Scheduled (daily/ hourly/ weekly/ etc)
- Alert when there are some errors

Kinda Methods

- Using procmail and PERL
  - Create a scheduler in the RouterOS to create backup and send them by email to server, then parse the email received
- Using Telnet/FTP and PERL
  - Use PERL packages to Telnet (to create backup) and FTP (to get the files)

Using Procmail and PERL

- Passive Servers
- Active RouterOS
- Suitable for small scale RouterOS Network
- SMTP and POP Server availability is a must
- Works only for RouterOS (or Router that support sending email with attachment)

Using Procmail and PERL

Using Procmail and PERL

- How to set it up?
  - Create a scheduler on each RouterOS to backup and send the backup files via Email to a Backup Server
  - Backup Server installed with Procmail to receive the mail and the PERL will parse them and put them in some directory
- Complete instruction see MikroTik WiKi

Using Procmail and PERL

- Drawbacks
  - Uncentralized database
  - Works only for RouterOS (or Router that come with sending email capability)
  - Security Issues (spam, etc), user firewall
  - Used only for Backup

Using Telnet/FTP

- Concept
  - Use PERL package Net::Telnet and Net::FTP
  - Must have database of Devices
  - Firewall and Service for FTP and Telnet must be opened
  - Server active - RouterOS passive

Before We Begins

- Before Begins
  - Prepare the data-list

```
[RouterName]/[RouterIP]/[Username]/[Password]

Nebula/192.168.10.100/herry/testing
Saturn/10.1.100.100/spectrum/pass
MilkyWay/172.16.16.1/spectrumindo/pass
```

Read the Data-Lists

```
sub getRouterList
{
  my $filename=pop(@_);
  my @list;
  open(MYFILE,$filename);
  while (<MYFILE>)
  {
    if (substr($_,0,1) ne “#”)
    {
      push(@list,$_);
    }
  }
  close(MYFILE);
  return (@list);
}
```

Telnet Subroutine

```
sub createBackup
{
  $param=pop(@_);
  (my $routername,
   my $ip,
   my $username,
   my $password) = split(/\//,$param);

  ### Creating The session and Login
  my $telnet = Net::Telnet->new(Errmode => 'return');
  $telnet->open("$ip");
  $telnet->login("$username","$password");

  ### Create the Backup script
  $telnet->print("export file=$routername");
  $telnet->waitfor('/> $/');
  $telnet->close();
}
```

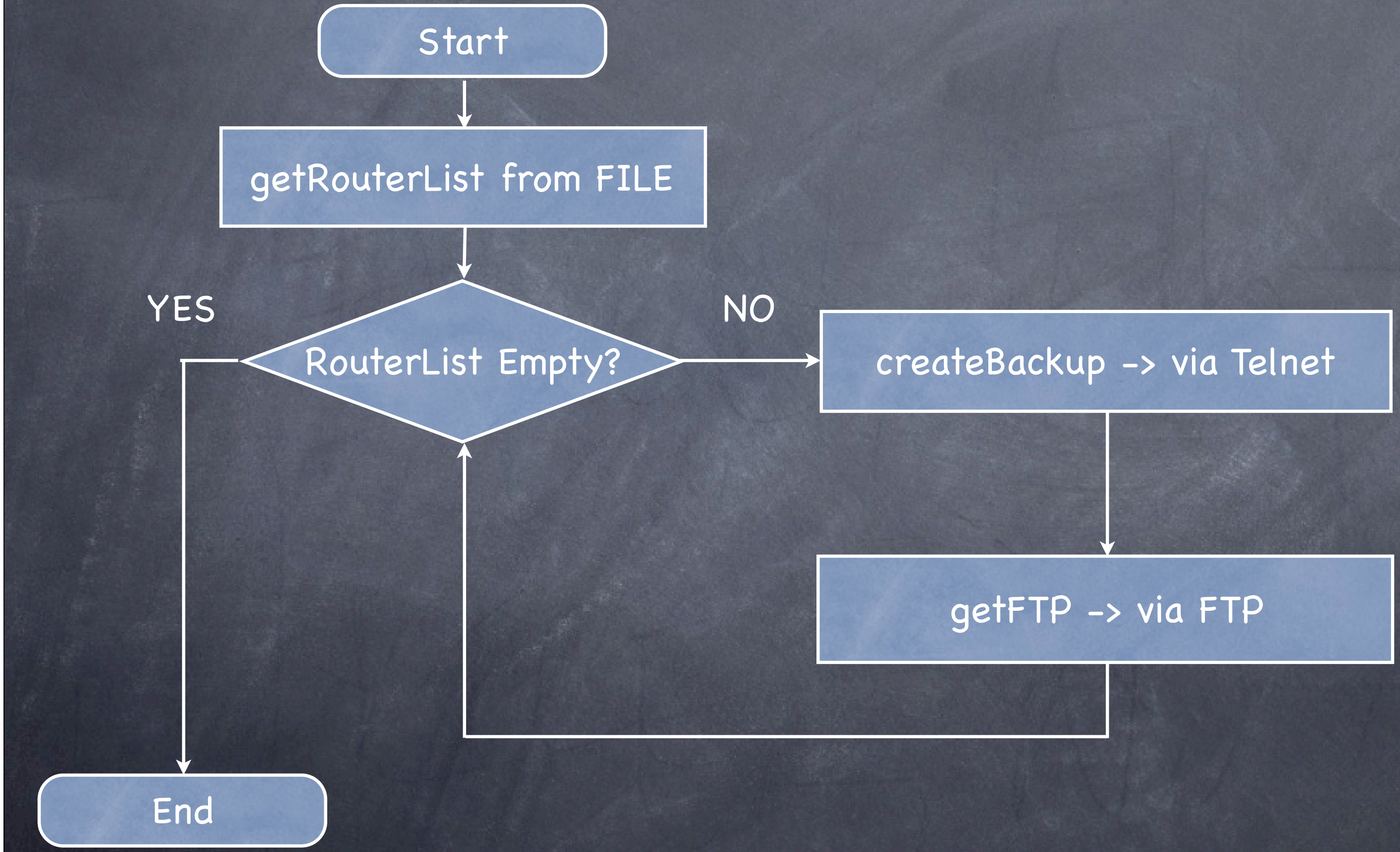
How to FTP

```
sub getFTP
{
  my $param=pop(@_);
  (my $routername,
   my $ip,
   my $username,
   my $password) = split(/\//,$param);

  ### Creating The session and Login
  my $ftp = Net::FTP->new($ip);
  $ftp->login($username,$password);

  ### Get the files
  $ftp->ascii();
  $ftp->get("$routername.rsc");
  $ftp->quit;
}
```

Program FlowChart



Main Program

```
# Created by Herry Darmawan - October 2009
#
# Desc :
#    RouterOS Centralized Backup using Telnet and FTP

use Net::Telnet();
use Net::FTP();

############### Main Program ###################
my $filename = "device.txt";

@router_list = getRouterList($filename);
foreach $line (@router_list)
{
  createBackup($line);
  getFTP($line);
}
```

Let's Try

- Get the subroutines from 10.1.1.1 (via FTP or WinBox)
- Put them on the same files with the sequence (do not reverse the sequence)
- Try to run the script

Drawbacks

- Each new/updated RouterOS information (IP, username or password) must be manually updated to the list
- Time access will increased (without multi-thread executable)
- Simultaneous / multi-thread executable will significantly increased the server load

Combined Method

- Use RouterOS scheduler to create the backup and Net::FTP to get the files
- Use Net::Telnet to create the backup and send a trigger from Server to Send an Email

How Far Can it be Used?

- Use Cron to scheduled the backup process
- Use Databases to store the List-of-IP
- Set the dst-directory into something recognized
- Net::Telnet can be used to do centralized configuration instead of Backup-only

RouterOS Centralized Configuration using PERL

- The same method we use to create backup via Telnet can be used to send any configuration to the Router
- If you need a return value (ie. see what shows on the terminal after command is executed in RouterOS), use
  - $telnet->cmd("<command>")

Concept

- Telnet to a RouterOS or some Routers
- Send a command or interact (get the status or information)
- Used to configure a lot of Router with same configuration pattern

The use of $telnet->cmd

```
##### Creating the Session and Login
my $telnet = Net::Telnet->new(Errmode => 'return');
$telnet->open($ip);
$telnet->login($username,$password);

##### Get the System Identity
my @temp = $telnet->cmd(
              ":put [/system identity get name]"
                       );
($routername) = $temp[0] =~ /\s*([\S ]+)$/;
print "The RouterName is : $routername \n";

$telnet->close();
```

Other Way

- Use combination of Net::Telnet, “fetch” and RouterOS Scheduler
  - Prepare a scripts of configuration in a web server
  - Telnet to a Router then send command to fetch the configuration in the web server
  - send command and create a scheduler to run the script

Thank You !

Herry Darmawan
herry@spectrumindo.com