VIRTUAL ACCESS

SMART NETWORK SOLUTIONS
FOR SERVICE PROVIDERS

natRedirect: Monitoring availability of a LAN device and switching NAT rule to avail of a backup LAN device

Issue: 1.0

Date: 03 June 2014

1 Introduction

The natRedirect script is designed to poll for main server on LAN using ping and modify an incoming NAT rule to forward to an alternate server when unavailable.

The script periodically pings the main server IP and takes a number on consecutive ping failure.

- Change desired incoming NAT rule to point to a backup server IP
- Generate an event to allow other scripts to trigger

The script is commonly used in a scenario shown in the figure below. The WAN connection is in the example is via a GSM link using PPP.

Note: this script currently only modifies one NAT rule but could be modified to handle multiple NAT rules. This would allow it to be used on multiple interfaces and/or handle multiple incoming SNAT translations.

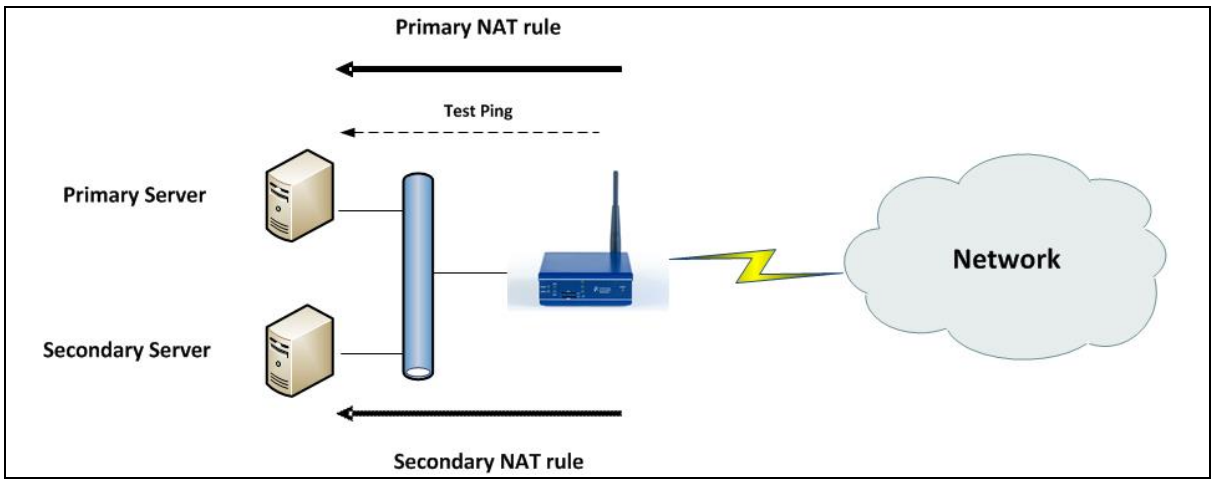

Figure 1: Network architecture

2 Configuring the natRedirect script

2.1 Script overview

The script can only be run once. It is designed to be run on boot.

On boot the script does the following:

- Waits 5 seconds to allow for time for interfaces to connect.
- Sends pings to LAN device destination at configurable durations. Pings allow for a reply wait time to be specified.
- A ping from target signifies that the LAN device is operating as normal.
- When a number of configurable consecutive ping failures are detected the script will change the specified incoming NAT rule internal IP to the specified backup device IP. An INFO event is also generate for visibility of the change and also to allow other scripts to trigger.
- On consecutive pings success the NAT rule is modified so the internal IP points to the original LAN device IP.

2.2 Script requirements

- This script is designed for a single NAT rule.
- The script requires that the NAT incoming rule is enabled on boot and ideally forwards to the main LAN device IP.

2.3 Script parameters

The script name is natRedirect and it takes in five required parameters and a further one optional parameter:

natRedirect	[nat index] [main server IP] [backup server IP] [ping wait] [maximum fail] [ping reply wait]

These parameters are described in the example and table below.

natRedirect 1, 1.1.1.1, 1.1.1.2, 10, 3, 1

Parameter	Type	Description	Default
1	Required	The NAT rule index to be modified.	n/a
1.1.1.1	Required	The main server IP.	n/a
1.1.1.2	Required	The backup server IP.	n/a
10	Required	The wait between pings in seconds.	n/a
3	Required	The number of consecutive pings failures/successes to cause the NAT rule to be modified.	n/a
1	Optional	The time to wait for a ping reply in seconds.	1

Table 1: natRedirect parameter descriptions

2.4 Configuring the script

To use the script, first paste the script from Section5 'natRedirect script' into the script editor and then use the scheduler to run the script at boot up.

From the start page, click **Advanced** to open the Expert View menu.

2.4.1 Pasting the script into the script editor

If you are using 9.09.xx firmware: in the Expert View menu, click **system > scripts->script editor**. The Script Editor page appears

If you are using 10.00.xx firmware: in the Expert View menu, click **system > management > scripts > script editor**. The Script Editor page appears.

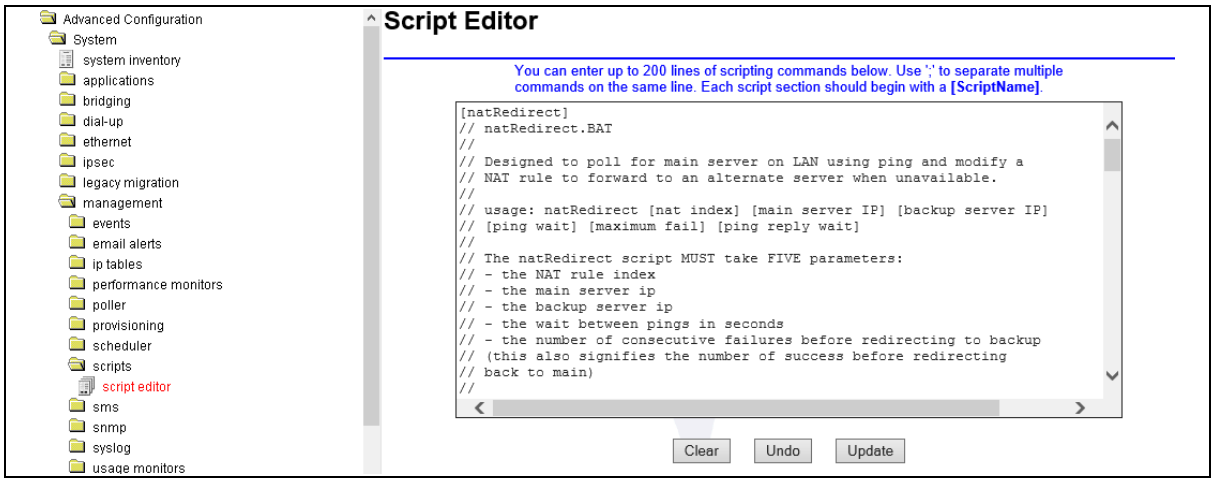

Figure 2: The script editor page

Paste in the script from Section5 'natRedirect script' from this document. The first line of the script should begin with the script name in square brackets, [natRedirect]. Use this name to call the script using the scheduler.

If you need to reduce the number of script lines, you can omit any line beginning with //, as this denotes a comment tag. Also, you can enter multiple script lines onto the same script editor line separated by ';' (semi colon). When you have completed the script, click **Update**.

2.5 Scheduling the script to run on boot

If you are using 9.09.xx firmware: in the Expert View menu, click **system > scheduler > scheduler tasks**. The Scheduler Task Entry page appears.

If you are using 10.00.xx firmware: in the Expert View menu, click **system > scheduler > scheduler tasks**. The Scheduler Task Entry page appears.

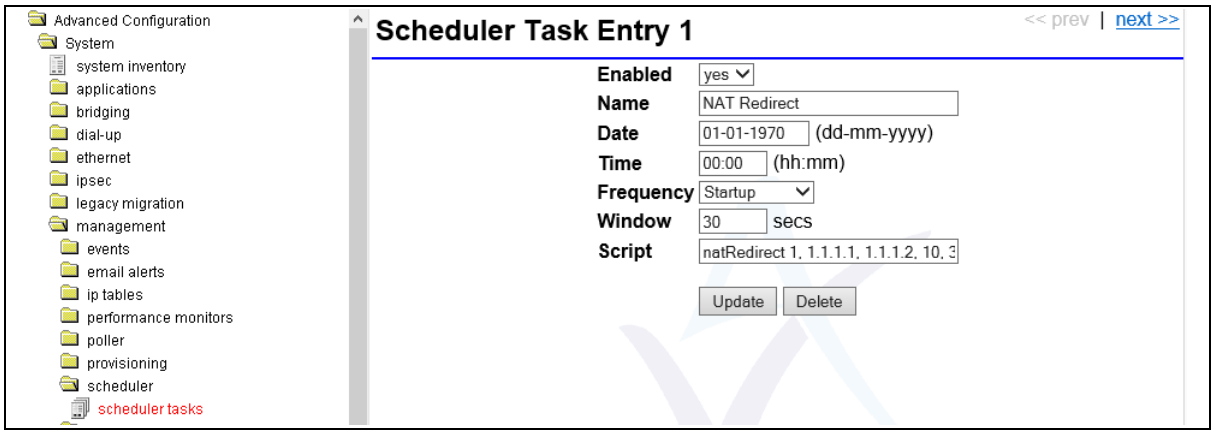

Figure 3: The scheduler task entry page

Field	Description
Enabled	Enables or disables a particular schedule. Set to **Yes**.
Name	The name associated with the schedule. Enter a descriptive name.
Date	The date the script initiates. This field is ignored when frequency is set to start up. Leave as default.
Time	The time the script initiates. This field is ignored when frequency is set to start up. Leave as default.
Frequency	Sets the frequency the script executes. Set to **startup**.
Window	This parameter sets how long the system will wait if it is busy before executing the script. For example if the script is set to execute at 10:00 and the window is set to 30 seconds, the system will try executing the script within this window only. Set to **30**.
Script	The name of the script to be executed. Enter the script name, followed by the relevant parameters as shown in the above image. Separate the parameters by commas. Example: natRedirect 1, 1.1.1.1, 1.1.1.2, 10, 3, 1

Table 2: The scheduler task fields and their descriptions

3 Debugging commands

Useful debug commands via command line are described in the table below.

Diagnostic Command	Description
Show tasks	Displays all running tasks.
Show task <tasknum>	Displays running task. Also indicates position task is currently at.
Show task vars <tasknum>	Displays variables and variable values associated with task.
Show ip route	Shows routing table
Show ipat incoming	Shows incoming NAT rules
Show events	Displays event log.
Show change log	Displays recent configuration changes.
Dir scripts	Displays all scripts embedded in the firmware.
Show config script ALL	Displays all scripts in the script editor.
Show config script <scriptname>	Displays the <scriptname> script as configured in script editor. Includes line numbers.
Show config script –n <scriptname>	Displays the <scriptname> script as configured in the script editor. Does not include line numbers.

Table 3: Debug command lines and their descriptions

Useful trace commands via the command line are described in the table below.

Trace command	Description
++All 6	Traces all INFO events
++ip	Traces IP traffic
++ip:icmp	Traces ICMP IP traffic
++script	Traces script events
--script	Stops script event tracing
--	Stops all event tracing
Trace on <script_name>	Traces each line in a script as it executes
Trace off <script_name>	Turns off tracing for script

Table 4: Trace command lines and their descriptions

4 Script events

Severity	Class	Subclass	Text
INFO	49	40	natRedirect polling <mainServerIP> with NAT <natIndex> backup <backupServerIP> every <pingWait> secs
INFO	49	40	natRedirect incorrect incoming address translation index <natIndex> (incorrect IP)
INFO	49	40	natRedirect incorrect incoming address translation index <natIndex> (rule disabled)
INFO	49	40	natRedirect configuration using main server <mainServerIP>
INFO	49	40	natRedirect configuration using backup server <backupServerIP>
INFO	49	40	natRedirect main server online <mainServerIP>
INFO	49	40	natRedirect main server offline $mainServerIP using backup <backupServerIP>

Table 5: Script events

5 natRedirect script

```
// natRedirect.BAT
//
// Designed to poll for main server on LAN using ping and modify a
// NAT rule to forward to an alternate server when unavailable.
//
// usage: natRedirect [NAT index] [Main Server IP] [Backup Server IP]
//                       [ping wait] [maximum fail] [ping reply wait]
//
// The natRedirect script MUST take FIVE parameters:
// - the NAT rule index
// - the main server ip
// - the backup server ip
// - the wait between pings in seconds
// - the number of consecutive failures before redirecting to backup
//   (this also constitutes the number of success before redirecting
//    back to main)
//
// It can optionally take ONE parameters:
// - the time to wait for a ping reply in seconds (default: 1)
//
// EXAMPLES
// --------
// natRedirect 1, 1.1.1.1, 1.1.1.2, 10, 3
// (Ping 1.1.1.1 every 10 seconds. Waits 1 secs for ping reply
// After 3 consecutive failures change incoming NAT rule index 1 to
// forward to the backup server.

!echo off
!arg natIndex, mainServerIP, backupServerIP, pingWait, pingCount

$pingReplyWait = $6

//defaults
!if pingReplyWait = ''
```

```
  $pingReplyWait = 1
!endif
$mainActive=1
$matchIP = 0
$i = 1
$ping_reply_wait_msec = 1000
!while $i < $pingReplyWait
  !add ping_reply_wait_msec, 1000
  !inc i
!endwhile

!log natRedirect polling $mainServerIP with NAT $natIndex backup
$backupServerIP every $pingWait secs

//checking
!if "`sh ip address translation entry configured $natIndex`" = "yes"
   !if "`sh ip address translation entry ip address $natIndex`" =
"$mainServerIP"
     $matchIP = 1
   !endif
   !if "`sh ip address translation entry ip address $natIndex`" =
"$backupServerIP"
     $mainActive = 0
     $matchIP = 1
   !endif
   !if $matchIP <> 1
      !log natRedirect incorrect incoming address translation index
$natIndex (incorrect IP)
      !exit
    !endif

   //config using backup or main?
   !if $mainActive <> 0
     !log natRedirect configuration using main server $mainServerIP
   !else
     !log natRedirect configuration using backup server $backupServerIP
   !endif
```

```
!else
  !log natRedirect incorrect incoming address translation index $natIndex
(rule disabled)
  !exit
!endif

!pause 5
$pingFail = 0
$pingPass = 0
!while 1

  //route check
  $z = `st ping results reset`
  $z = `quiet ping $mainServerIP -w $ping_reply_wait_msec`
  !pause $pingReplyWait
  $result = `sh ping replies`
  !if $result > 0
    $pingFail = 0
    !inc pingPass
  !else
    $pingPass = 0
    !inc pingFail
  !endif

  !if $mainActive <> 1
    !if $pingPass >= $pingCount
      $z = `set ip address translation entry ip address $natIndex
$mainServerIP`
      $z = `commit`
      !log natRedirect main server online $mainServerIP
      //reset fail count
      $pingFail = 0
      $mainActive = 1
    !endif
  !else
    !if $pingFail >= $pingCount
```

```
      $z = `set ip address translation entry ip address $natIndex
$backupServerIP`
      $z = `commit`
      !log natRedirect main server offline $mainServerIP using backup
$backupServerIP
      //reset pass count
      $pingPass = 0
      $mainActive = 0
    !endif
  !endif

  !pause $pingWait
!endwhile
```