rPatIN

Table of Contents

1 Introduction

Version: 1.0

Purpose of test

- Remote Patin (rPatin) is a network stress test using file operations. It performs file write, read, compares to disk drive across a network. The workload portion executes on "client" machines that will each target share drives on the unit under test (server).
- rPatin consists of two separate binaries - pControl and pGen.
- pControl starts up pGen on connected clients from a Telnet session. Once pGen is up and running, it sends messages to pControl using network sockets. pControl monitors the sockets and watches for timeout conditions.
- pControl is visible to the CTC test executive and is called out in the PKX file.

Module Test Time

- Execution time is dependent on user parameters. The key contributor to the execution time is the RunTime parameter. By default, the snippet (SNX) file defines a 15 minute RunTime, but this will generally be overridden by PKX parameters.

Other Notes

- There are requirements on the network infrastructure and client configuration to use rPatin. Please see the Installation help page.

1.1 Installation

- **Files needed to run rPatin Test**
  - pControl - Executable binary
  - pGen.exe - Executable binary
  - rPatin.snx - Snippet file
  - rPatIN.pdf - This help file
- **How to set up this module**
  - Import the SNX and/or PKX file. Auto-configure on a system with usable clients.
- **Lab prerequisites before using this module**

- **Windows Client side**

o All clients must be running Windows Server 2003 or later (2008/2008r2/2012 are recommended)

o XP, Vista, Win7, Win8 on clients should also work but have not been rigorously tested

o Clients are recommended to have ECC memory to avoid potential for client-side data corruption
(this is the same requirement as with any previous client-based stress apps)

o The TELNET service must be enabled by default (a one time change to some OS versions).

o Username/password must be set to Administrator/password

o Increase concurrent Telnet Server connections with "tlntadmn config maxconn=X" (with X being minimum number of connections expected). The default is maxconn=2.

o All firewalls must be DISABLED.

Note: Clients should not create a performance bottleneck; they need sufficient performance to drive the network(s) under test to saturation levels. Lower performing clients or Ethernet controllers will result in less network saturation.

- **Linux SUT side**

o Linux SUT must have Windows file sharing turned on (CIFS/SAMBA) with full rights given to user: Administrator/password

o All SUT firewalls must be DISABLED

o Advanced network features such as Load Balancing, NIC teaming, link aggregation have not been validated with rPatin

o rPatin will use all machines visible on the network as “clients” with an open telnet port; it does not have an IP range exclusion at this time. Possible workarounds include:

  - Before doing a RECONFIGURE, temporarily disable or disconnect any port on the SUT that connects to machines that you don't want as part of the configuration. For example, a connection to a lab PXE server. Reconnect after RECONFIGURE

  - Another option is to use IPV6 for things like a dedicated lab network, PXE server, etc. rPatin does not recognize IPV6 networks

- **Network**

o Networks must use IPV4 class C addressing scheme to limit SUT visibility to the desired clients on the desired subnets.

- Simultaneous sharing of common clients between different SUT test beds has not been tested but should work if on different subnets
- Avoid any SUT from having visibility to another SUT through a common subnet; otherwise each will treat the other as a client
- TCP/IP network switches must be capable of handling wire-speed saturation levels on all server side ports.

1.2 Example Configuration

RHEL7 SUT Example Installation Procedure

- **rPatin** requires a System Under Test (**SUT**) and remote network host(s) (**Clients**). All clients are presumed to be running Windows Server 2012 or Server 2012R2 with the Telnet Server service installed and started. The **iIVSS** stress tool is installed on the SUT only.
- While it is possible to directly connect the SUT to a single client in a back-to-back scenario, it is preferred to run the stress test across a network infrastructure to more closely simulate a real world environment.

Setting up the SUT - RHEL7 OS Install

- Install a 64-bit version of RHEL 7 to the SUT using the steps below.
  - After finishing the disk partitioning part of the install, go to **Software Selection** and under **Base Environment** select **Server with GUI**
  - To the Add-Ons section to the right check **Directory Server, File and Storage Server, Compatibility Libraries, and Development Tools** then click **Done**.
  - Click **Begin Installation** and let the install complete
  - After the install is complete and the system reboots, continue the initial setup. RHEL 7 requires you to add an additional user, but we can use the root account too after we activate it.
  - Log in as root and from a console type the following command
    # su –u root
  - Make sure the root password is set to "**password**". Install the latest LAN drivers and other drivers necessary to meet your particular hardware requirements.

- **Perform the following additional setup items:**
  - Stop and disable the Firewall Service on RHEL 7.
    - From a command line, type: **systemctl stop firewalld.service,** then **systemctl disable firewalld.service.**
    - To verify the Firewall Service is stopped, type: **systemctl status firewalld.service.**
  - To verify the Samba Service is installed and running, from a command line, type **systemctl status smb.service.**
  - Set up your network so each NIC on the SUT corresponds to the subnets of your remote clients. Verify you can ping the remote clients.

Setting up the Windows Clients

The remote clients are presumed to be Windows-based with either Windows Server 2012 or Windows Server 2008R2. All OS drivers should be installed and the client NICs should have IP addresses set up to match the subnets on the SUT. Make sure you can ping each NIC on the SUT. Make sure the clients also have the same password, and Firewalls disabled.

Install Telnet Server on the Clients

In order for rPatin to properly communicate, the Telnet Server feature must be installed on the Clients. It does not have to be installed on the SUT. Take the following steps.

- From **Server Manager Dashboard**, click on **Add roles and features**.
  (In **W2K8R2**, go to **Server Manager**, click on **Features** and select **Add Features**).
- Click **Next > Next > Next** until you see **Telnet Server** feature listed. Check the checkbox next to **Telnet Server** and and click **Next** once more.
- Click **Install** to begin the installation. Wait until the service is installed and reboot if prompted to.

Start the Telnet Service on the Clients

Once the **Telnet Server** feature is installed, make sure to enable the **Telnet service**, which is disabled by default.

- From **Server Manager Dashboard**, click on **Tools > Computer Management**.
- Expand **Services and Applications** and click on **Services**.
- Under the list of services, scroll down to find the **Telnet** service.
- Right-click **Telnet** and click on **Properties**.
- Under **Startup Type**, change it to **Automatic (Delayed Start)**, then **Apply > Okay**.
- Under **Service Status**, click **Start**, then **Okay**.

NOTE: By default Windows Telnet Service allows only two concurrent telnet

connections. Increase the number of concurrent connections from a com shell using the command "**tlntadm config maxconn=X**", with **X** being the **minimum** number of connections expected.

Installing iLVSS on the SUT

- Obtain a copy of **ilVSS** version **3.6.13** and install it.
  - Extract the tarball (as root… **tar zxvf ilvss-<version>.tar.gz)** to a directory of your choice.
  - Change to the newly created ilvss directory **(cd /opt/ilvss-<version>)**
  - With the RHEL 7 install DVD in a DVD drive, run installation script **(./install**).
  - Your installation should appear in **/opt/ilvss.X** (where "X" is the n'th installation of ilVSS you have installed on that SUT…0 is first).
  - Run CTC (**./ctc**) to launch the user interface.

Configuring ilVSS to run an S3 test using rPatin

- After starting CTC to launch ilVSS, Click **File > Open** and double-click the **.pkx** package file that corresponds to your platform type. For the Purley family, use "**stress_purley.pkx"**.
- After selecting the package, a final **Configuration Selection** dialog box appears. Click the drop-down arrow and change it from **Template Configuration** to **Purley** (or whatever platform you're testing). Then click **OK**.
- In the upper right **Flow** selection window, click the drop-down arrow and select **S3**.
- Now click the **Reconfigure** button to open the **Reconfigure Options** window.
- By default all of the available device modules are set to **Reconfig-'YES'**. The other modules can also be run concurrently at your discretion, but for purposes of this BKM, we are only using **rPatin** and setting the remainder to **Reconfig-'NO'**. Make sure **rPatin** is set to **Reconfig-'YES'** and hit **Reconfigure**.

2 Subtests

rPatIN is targeted to support the following tests

1. Auto-configuration
2. rPatIN

2.1 Auto/Reconfiguration

Auto-Configuration is a required subtest of any module CTC supports. It is designed to initialize test parameters based off of the platform configuration. If auto/reconfiguration fails, it is highly unlikely any other subtest will function correctly.

2.2 rPatIN Test

The Synchronous I/O subtest is designed to perform reads and writes synchronously using the operating system's basic blocking "read()/write()" functions. The test will wait for a return value from these functions before progressing.

3 Parameters

The following list of parameters are available to rpatin.

3.1 DeleteDatFiles

Files created on the target drives are named <eight_random_hex_digits>.dat. If a test is prematurely aborted, the DAT file is left on the target. If the test exits normally, it will delete the DAT file it created if this parameter = 1. For debug purposes, one may want to examine all .DAT files and can set this parameter to 0.

DeleteDatFiles=0 (default if not specified)

DeleteDatFiles=1 (do not remove DAT files)

3.2 DisableFileErrorLogging

If debugging and there are many errors being generated, file logging just slows down throughput, when analysis with instrumentation is desired. Set to 1 if you are debugging and don't care about permanent error logs. Default is 0.

3.3 DriveListParameter

This parameter allows the user to specify the name of an array parameter that will hold the drives to be used. This is used in place of, and overrides the DrivesToBeUsed parameter. There must be a matching parameter array with the same name as the value specified by this parameter.

Examples:

DriveListParameter=LOCAL

LOCAL=<a parameter array holding the list of local drives>

DriveListParameter=REMOTE

REMOTE=<a parameter array holding the list of remote drives>

This parameter is usually defined as a CHILD under a specific subtest instance. This enables unique groups of target drives to be collected together.

3.4 HaltOnError

Specifies how the pControl module should behave after an error is detected.

0=Continue running to completion after an error; restart any failed threads at the next pattern change. (Default)

1=Kill any failed thread, do not restart the thread at the next pattern change.

2=End the module if an error is detected

3.5 Itrations

Number of times the entire program (set of parameters) will be executed. 0 results in an infinite run time. Default is 1.

SecondsPerPattern and RunTime both override this parameter.

3.6 Local

This is a parameter array that is populated during autoconfigure. It is not used by the test, but because of code leftover from Patin, it is still required to be present otherwise an error will be created.

This parameter can be ignored.

3.7 Master

This parameter array contains a matrix of client machines and the target drives on the server.

The first row contains the list of target drive letters on the server

The first <n> columns contain the list of clients on each subnet.

The number that intersects these (3 in the example above) indicates how many threads to execute.

This array is automatically populated by auto-configure and uses the ThreadsPerDrive parameter.

3.8 NumberOfBytesPerTransfer

This parameter specifies the amount of data transferred in the WriteFile and ReadFile APIs.

Special requirements exist if ReadOnly=1:

NumberOfBytesPerTransfer must be less than or equal to the size of the pattern file

Pattern file size MOD NumberOfBytesPerTransfer must = 0

For most cases, set NumberOfBytesPerTransfer=131,072. This has been shown to be optimal for both network and hard drive transfers.

For CDROM, the optimal value is 4096.

Special rules for this parameter

If NumberOfBytesPerTransfer < 65536:

Pattern file size MOD NumberOfBytesPerTransfer must = 0 (pattern file size can be assumed to be 65536 if it is smaller)

If NumberOfBytesPerTransfer > 65536:

NumberOfBytesPerTransfer MOD pattern file size must = 0

3.9 OutputPatternFileSize

The size of the output file that PatIN writes. Values are specified in MB. This is a REQUIRED parameter and must be defined.

OutputPatternFileSize=1 (minimum size, 1MB or 1,048,576 bytes)

OutputPatternFileSize=4096 (creates a 4GB file)

3.10 PatternDir

The directory where the pattern files are stored (usually /LinuxVSS/patterns) When READONLY mode is set, for testing CDROM for example, this is still the location for the reference pattern. Patterns are read from the readonly device from its root directory.

The size of the pattern file is tied in with NumberOfBytesPerTransfer (see above). It can be 2 bytes minimum, but 65536 MOD pattern file size must be 0. Internally it is built up to 65536 if smaller than this.

3.11 PatternFile

The pattern file names to be run. Total string length limited to 1000 characters. This parameter is mandatory if PatternFilesArray is not available.

3.12 PatternFilesArray

The pattern file names to be run, saved in a CTC array. This allows many more patterns. It takes precedence over PatternFile.

3.13 PerfInterval

Specifies the interval in seconds between reporting performance data.If the parameter is not

specified, the performance is reported only every pattern change.

3.14 RandomOrder

This parameter causes the software to internally randomize the order that patterns from PatternFilesArray will be run.

0=Run the patterns in the order they appear (default)

1=Scramble the order of patterns

3.15 ReadOnly

Defaulted to '0' - for normal write-read-compare. To test read only media such as CDROM, set this to 1.

Read-only devices/media are expected to already contain the pattern files being used.

ReadOnly=0 (default, normal operation)

ReadOnly=1 (for read-only devices)

3.16 RunTime

Specifies the number of seconds to run the test. This value is divided by the number of patterns to get the SecondsPerPattern. The default value is 0 seconds which is to ignore this parameter.

3.17 SecondsPerPattern

Specifies the number of seconds each pattern file will be executed. Minimum=1.

If specified, this parameter will be overridden by the RunTime parameter.

3.18 SocketTimeout

Specifies the number of seconds before a warning message is generated. This indicates that a network socket is having difficulty communicating across the network. If there are 5 consecutive socket timeouts, an error is generated.

The pGen process instance associated with the socket that timed out will be in an unknown state; it will continue "as is" until the pControl module ends.

3.19 ThreadsPerDrive

Number of threads per target drive. Every thread creates its own output file on the target drive.

Default=1 (if not specified)

Maximum=10

3.20 TimeoutSeconds

This parameter defines the number of seconds to wait for a response after the WriteFile() or ReadFile(). If the API response exceeds this timeout value, an advisory message is generated. The advisory message continues to be generated every TimeoutSeconds until the operation is completed.

Valid range = 1 to 32 million

Default = 10 (if the parameter is not specified)

In some lab environments, these messages may be more likely to appear. This is not an indication of an error in the SUT, but is more likely to be a result of inferior network infrastructure (switches) or low performance clients. Increasing the value of this parameter will not change anything other than how frequently the advisory message appears in the log.

3.21 WriteOnceReadAndCompareContinuous

This parameter will cause the test to continuously read and compare the output file until the number of seconds per pattern file is consumed.

0=Write the file, read the file (default)

1=Write the file once, then read continuously

Index

- I -

- P -

- S -