3PAR Nagios check script v0.1

This script is provided "as is" without warranty of any kind and 3PAR specifically disclaims all implied warranties of merchantability, non-infringement and fitness for a particular purpose. In no event shall 3PAR have any liability arising out of or related to customer's 'use of the script including lost data, lost profits, or any direct or indirect, incidental, special, or # consequential damages arising there from.

In addition, 3PAR reserves the right not to perform fixes or updates to this script

Contents

Introduction and requirements

This check script is intended to provide active monitoring of 3PAR InServs with Nagios

The following items can be monitored:

- Controller nodes status
- Physical disks status
- Fibre Channel ports status
- Logical disks status
- Virtual volumes status
- Fibre Channel disks used capacity
- NearLine disks used capacity

The minimum version of Inform OS required is v2.3.1

Installing the check script

Copy the check script "check_3par" in the "libexec" directory of Nagios (/usr/local/nagios/libexec for example)

Make the script executable:

```
chmod +x check_3par
```

The check script requires a temporary directory. It is set by default to /tmp

The "nagios" user needs to have write permissions to this directory

To change the temporary directory edit the check_3par file and change the following variable:

```
TMPDIR=/tmp
```

Configuring the check script and Nagios to use the 3PAR CLI

Note : This step is not required if the check script will communicate with the InServ using SSH

Copy the 3PAR CLI 2.3.1 binaries to the Nagios server and install it

Edit the script check_3par and uncomment the following line:

```
CONNECTCOMMAND="/opt/3PAR/inform_cli_2.3.1/bin/cli -sys $INSERV -pwf
$USERNAME"
```

Note: Change the path the the CLI binary if it was not installed in /opt/3PAR/inform_cli_2.3.1

For each InServ that Nagios will need to connect to, do the following:

Run the 3PAR CLI and connect to the InServ:

```
nagios@nagios:~$ /opt/3PAR/inform_cli_2.3.1/
bin/        jre/        lib/        log/        readme.htm  resource/
ssh_keys/   uninstall/
nagios@nagios:~$ /opt/3PAR/inform_cli_2.3.1/bin/cli
system: 192.168.47.64

user: 3paradm
password:
mktg-ins2 cli%
```

Create a password file:

```
mktg-ins2 cli% setpassword -saveonly -file mktg-ins2.pwf
Current password:
mktg-ins2 cli%
```

Copy the password file to the "libexec" directory of Nagios (/usr/local/nagios/libexec for example)

Check that the nagios user can successfuly run the check script on the InServ:

```
nagios@nagios:~$ /usr/local/nagios/libexec/check_3par 192.168.47.64
/usr/local/nagios/libexec/mktg-ins2.pwf check_node
OK : All nodes have normal status
```

Configuring the check script and Nagios to use SSH

Note : This step is not required if the check script will communicate with the InServ using the 3PAR CLI

Edit the script check_3par and uncomment the following line:

```
CONNECTCOMMAND="ssh $USERNAME@$INSERV"
```

Log in to the server with the "nagios" user

Create a SSH key with with "ssh-keygen"

Display the content of the public key :

```
nagios@nagios:~$ cat .ssh/id_rsa.pub

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA0WWWeOfuzItlXeokeBtaGdKnwbKZvkz/b3qCWWCfX3l/L
kAfyMM8U+Xxb0Z3xmQKvbpwFHtSe/an+hShZC5E9YOkMFr/ucHAbGRmwvNQUkXlJyo7BbVwfu
BHqJiY+HxJjqcvP2HEm4fJZ1ggApbxzQVh/vp75XAGyXr327DOUonSU5acvkOVW5au/uvwaho
srVMP7kxrF+ULO5ynV7U9nDEmgeONF/k4YHwze29auARw3u5kJ4rgZUcV/sWL+Xys1cg5hcO5
hq18+x5COnifzM4kjHXlxyeKpWTWRyBjwbxxXLsK31Cho3AWVIEqfrYG5qqjo9fvh1YnU7oZZ
uhUqw== nagios@nagios
```

For each InServ that Nagios will need to connect to, do the following:

Connect using SSH to the InServ (with the user that will be used for running the checks) and add the key fingerprint to the list of known hosts:

```
nagios@nagios:~$ ssh 3paradm@192.168.47.64
The authenticity of host '192.168.47.64 (192.168.47.64)' can't be
established.
RSA key fingerprint is 68:a6:c9:60:a1:cb:12:e2:76:1a:ee:47:b0:8b:f9:ee.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.47.64' (RSA) to the list of known
hosts.
Password:
```

Add the SSH public key to the InServ:

```
mktg-ins2 cli% setsshkey -add
Please enter the SSH public key below.  When finished, press enter twice.
The key is usually long.  It's better to copy it from inside an editor
and paste it here.  (Please make sure there are no extra blanks.)
The maximum number of characters used to represent the SSH key
```

```
(including the "from" option, key type, and additional comments) is 4095.

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA0WWWeOfuzItlXeokeBtaGdKnwbKZvkz/b3qCWWCfX3l/L
kAfyMM8U+Xxb0Z3xmQKvbpwFHtSe/an+hShZC5E9YOkMFr/ucHAbGRmwvNQUkXlJyo7BbVwfu
BHqJiY+HxJjqcvP2HEm4fJZ1ggApbxzQVh/vp75XAGyXr327DOUonSU5acvkOVW5au/uvwaho
srVMP7kxrF+ULO5ynV7U9nDEmgeONF/k4YHwze29auARw3u5kJ4rgZUcV/sWL+Xys1cg5hcO5
hq18+x5COnifzM4kjHXlxyeKpWTWRyBjwbxxXLsK31Cho3AWVIEqfrYG5qqjo9fvh1YnU7oZZ
uhUqw== nagios@nagios

SSH public key successfully set.
```

Check that the nagios user can now connect without having to enter a password:

```
nagios@nagios:~$ ssh 3paradm@192.168.47.64
mktg-ins2 cli%
```

Check that the nagios user can successfuly run the check script on the InServ :

```
nagios@nagios:~$ /usr/local/nagios/libexec/check_3par 192.168.47.64
3paradm check_node
OK : All nodes have normal status
```

Creating the Nagios commands definitions

Edit the content of the commands file (/usr/local/nagios/etc/objects/commands.cfg in Nagios 3.x)

Add the following lines to the file:

```
define command{
        command_name    check_3par_pd
        command_line    $USER1$/check_3par $HOSTADDRESS$ $ARG1$ check_pd
        }

define command{
        command_name    check_3par_node
        command_line    $USER1$/check_3par $HOSTADDRESS$ $ARG1$
check_node
        }

define command{
        command_name    check_3par_ld
        command_line    $USER1$/check_3par $HOSTADDRESS$ $ARG1$ check_ld
        }

define command{
        command_name    check_3par_vv
        command_line    $USER1$/check_3par $HOSTADDRESS$ $ARG1$ check_vv
        }

define command{
        command_name    check_3par_cap_fc
        command_line    $USER1$/check_3par $HOSTADDRESS$ $ARG1$
check_cap_fc
        }

define command{
        command_name    check_3par_cap_nl
        command_line    $USER1$/check_3par $HOSTADDRESS$ $ARG1$
check_cap_nl
        }

define command{
        command_name    check_3par_port_fc
        command_line    $USER1$/check_3par $HOSTADDRESS$ $ARG1$
check_port_fc
        }
```

Adding the Nagios objects and service definitions

Edit the main Nagios configuration file (/usr/local/nagios/etc/nagios.cfg for example)

Add the following line to the configuration file:

```
cfg_file=/usr/local/nagios/etc/objects/3par.cfg
```

Note: If the Nagios object directory is not /usr/local/nagios/etc/objects, change the line to match the real directory

Create the 3par.cfg file in the objects directory (/usr/local/nagios/etc/objects for example)

For each InServ that will be monitored do the following:

Add a host definition with the name of the InServ and the DNS name or IP address:

```
define host{
        use                     linux-server
        host_name               mktg-ins2
        address                 192.168.47.64
        }
```

If the Nagios server uses the 3PAR CLI to communicate with the InServ:

Add the following lines:

```
define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             Physical disks status
        check_command                   check_3par_pd!$USER1$/mktg-
ins2.pwf
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             Nodes status
```

```
        check_command                   check_3par_node!$USER1$/mktg-
ins2.pwf
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             Logical disks status
        check_command                   check_3par_ld!$USER1$/mktg-
ins2.pwf
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             Virtual volumes status
        check_command                   check_3par_vv!$USER1$/mktg-
ins2.pwf
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             FC disks capacity
        check_command                   check_3par_cap_fc!$USER1$/mktg-
ins2.pwf
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             NL disks capacity
        check_command                   check_3par_cap_nl!$USER1$/mktg-
ins2.pwf
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             FC ports status
        check_command                   check_3par_port_fc!$USER1$/mktg-
ins2.pwf
       }
```

Note: Replace "mktg-ins2.pwf" with the name of the password file that was created earlier

If the Nagios server uses SSH to communicate with the InServ:

Add the following lines:

```
define service{
```

```
        use                             generic-service
        host_name                       mktg-ins2
        service_description             Physical disks status
        check_command                   check_3par_pd!3paradm
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             Nodes status
        check_command                   check_3par_node!3paradm
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             Logical disks status
        check_command                   check_3par_ld!3paradm
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             Virtual volumes status
        check_command                   check_3par_vv!3paradm
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             FC disks capacity
        check_command                   check_3par_cap_fc!3paradm
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             NL disks capacity
        check_command                   check_3par_cap_nl!3paradm
       }

define service{
        use                             generic-service
        host_name                       mktg-ins2
        service_description             FC ports status
        check_command                   check_3par_port_fc!3paradm
       }
```

Note: If necessary replace “3paradm” by the user name that will connect to the InServ

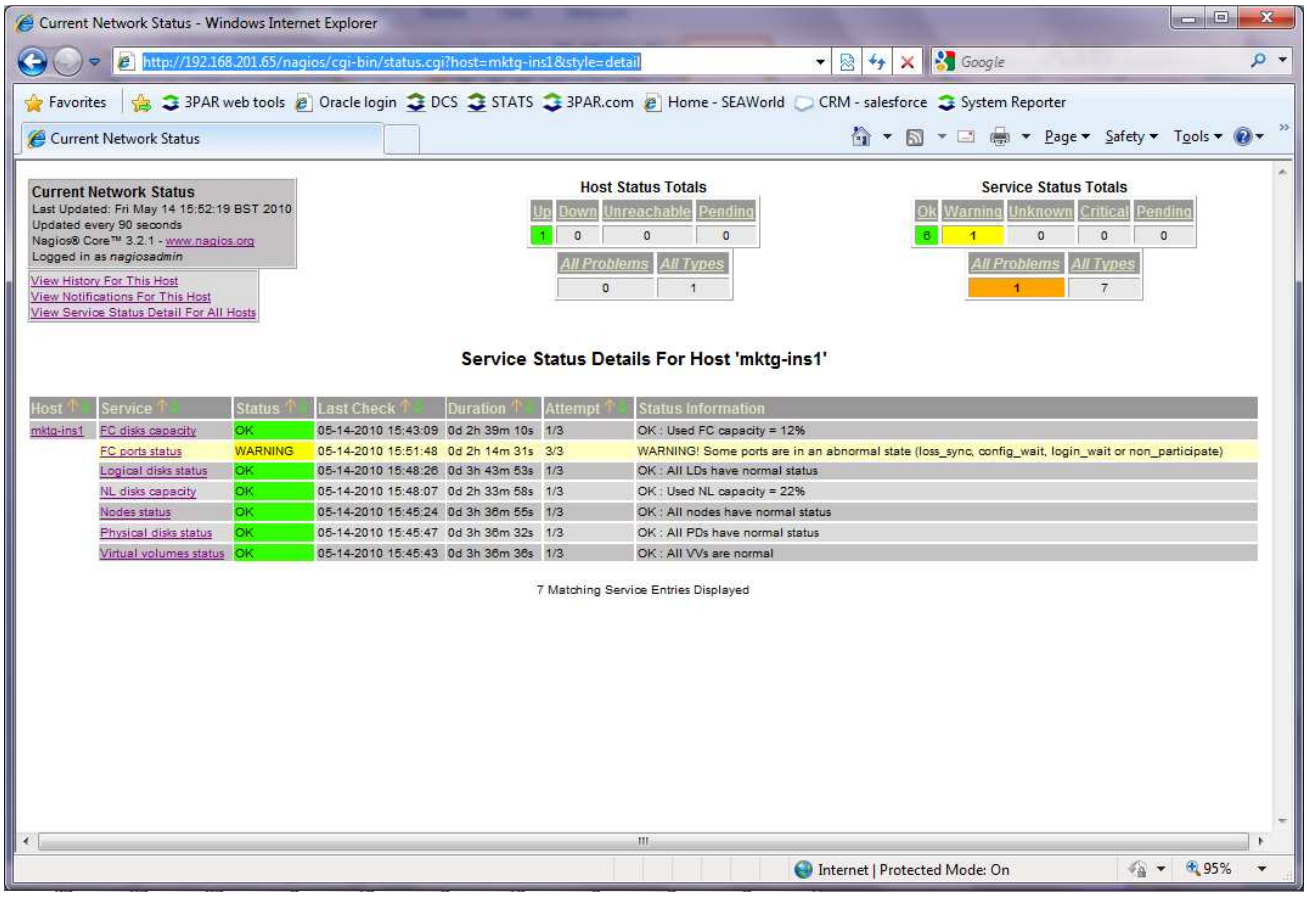
Restart Nagios and check that no error is reported:

```
root@nagios:/usr/local/nagios/etc/objects# service nagios restart
Running configuration check...done.
Stopping nagios: done.
Starting nagios: done.
```

Checking the InServ status in Nagios

Connect the Nagios web page and check the hosts

The InServs should be displayed and have the services monitored:



Configuring the thresholds for warning and critical capacity usage

By default the script will set warning level at 80% used capacity a critical level at 90% used capacity for FC and NL disks.

To change these levels edit the check_3par script and edit the following variables:

```
PCCRITICALFC=90
PCWARNINGFC=80
PCCRITICALNL=90
PCWARNINGNL=80
```

- PCCRITICALFC : Percentage of used FC capacity that will set the Critical threshold
- PCWARNINGFC : Percentage of used FC capacity that will set the Warning threshold
- PCCRITICALNL : Percentage of used NL capacity that will set the Critical threshold
- PCWARNINGNL : Percentage of used NL capacity that will set the Warning threshold