Introduction and Purpose

Nagios is a leading application for system and network monitoring, licensed under GNU GPL. It includes many precompiled functions for monitoring traditional hosts and devices. Through the add-ons (nsclient, nsca, nrpe), Nagios can be extended to monitor a wide range of devices and services.

Increased deployment of storage area network (SAN) and network attached storage (NAS) technology has complicated the task of monitoring hosts and services that rely on this technology. SAN and NAS vendors typically use proprietary operating systems that resist standard Nagios plugins and add-ons.

This document describes techniques to use the nsca add-on to monitor an EMC storage system. The specific case includes CLARiiON CX-340 and CX4-240C SANs, Celerra NS40 NAS gateway, and RecoverPoint v3.2 appliances. With minor adjustments, the techniques should work with other EMC equipment combinations.

Design Goals

- Utilize a subset of the EMC command line interfaces (CLI) to provide first-level Nagios monitoring and diagnosis. This monitoring is not intended to replace the GUI monitoring functions, rather to integrate EMC equipment into enterprise monitoring and alerts.
- Select monitoring commands that summarize the operational state of the EMC equipment. For example, paired CLARiiON storage processors can be examined with a single command. Likewise, a RecoverPoint cluster can be monitored.
- Use generalized Nagios services to report the command results, and parse the command outputs to extract meaningful diagnostic information. For example, a CLARiiON might support dozens of RAID groups or a Celerra might support fifty filesystems. It would be tedious to define them all to Nagios as individual services. Instead, define one service and identify any problems in the message text.
- Develop an extensible architecture to support additional monitoring that may be needed. The constraint is that CLI outputs are not uniform in syntax. The design utilizes external command files to identify monitoring commands, with individual Perl subroutines to parse their outputs. Hopefully the Perl is maintainable.
- Operate within limits imposed by the EMC architecture, e.g. Celerra has no **c** compiler and provides no ability to install compilers. RecoverPoint appliances provide a restricted shell with no console access to the operating system. These constraints prevent typical nsca installations.

EMC CLARiiON SAN

EMC CLARiiON is a family of SAN systems that share a similar architecture. Storage services are provided by storage processors (SP) that are inaccessible to typical operating system commands. To communicate with an SP, EMC provides both a web-based GUI and a command-line interface (CLI). The CLI is operated from a Windows or Linux host that connects to one or more SPs.

Information sources

Perusal of the EMC Navisphere Command Line Interface Reference led to selection of several commands that provide useful monitoring information:

faults -list	list of any faulted components on the storage system
getall -cache	list of states of read & write cache for both SPs
getlun -type -state	list of states of LUNs
getrg -type -state	list of states of RAID groups

Installation and configuration

- Using Navisphere, create a monitoring account on the SP to be monitored (typically SP A). This procedure assumes an account name *nagios*.
- Install Navisphere CLI on a convenient Linux host. Because CLI operates across the network, a single installation can be used to monitor multiple storage systems.
- If not already done, install the gcc compilers and Perl on this host. Install Nagios NSCA per instructions on Sourceforge.
- Logon to the Linux host and create a corresponding local account, e.g. *nagios*. Suggested home directory is /opt/nsca.
- As root, make a directory /opt/nsca and import required files from list below. Mark the Perl and shell scripts as executable. Change directory owner to the local account.
    - clariion.cmd
    - nsca_clariion.cron
    - nsca_clariion.pl
    - nsca_clariion.sh
    - send_nsca.cfg
- Create a CLI security file using the command below and the monitoring password:

```
naviseccli -AddUserSecurity -Password <pwd> -Scope 0
```

- Edit the clariion.cmd file to create conforming Nagios service names and point to the monitored SP.
- Edit the Perl program nsca_clariion.pl for compliance with directory and file names as needed. If desired, adjust the message severity between warning and critical levels.
- Run the Perl program nsca_clariion.pl from the command line. It should produce a stream of messages and write a file nsca_clariion.dat in nsca tab-delimited format. Correct any errors.
- Edit the shell script nsca_clariion.sh for compliance with directory and file names as needed. On the send_nsca line, revise the Nagios host as needed. Edit the nsca configuration file send_nsca.cfg as needed.
- Run the shell script nsca_clariion.sh from the command line. It should produce the nsca_clariion.pl outputs and send nsca packets to the Nagios host. Correct any errors.
- Edit the sample crontab nsca_clariion.cron for compliance with directory and file names. Then edit the actual crontab for the monitoring account.
- On the Nagios host, create required entries for the CLARiiON and its services. See example following.

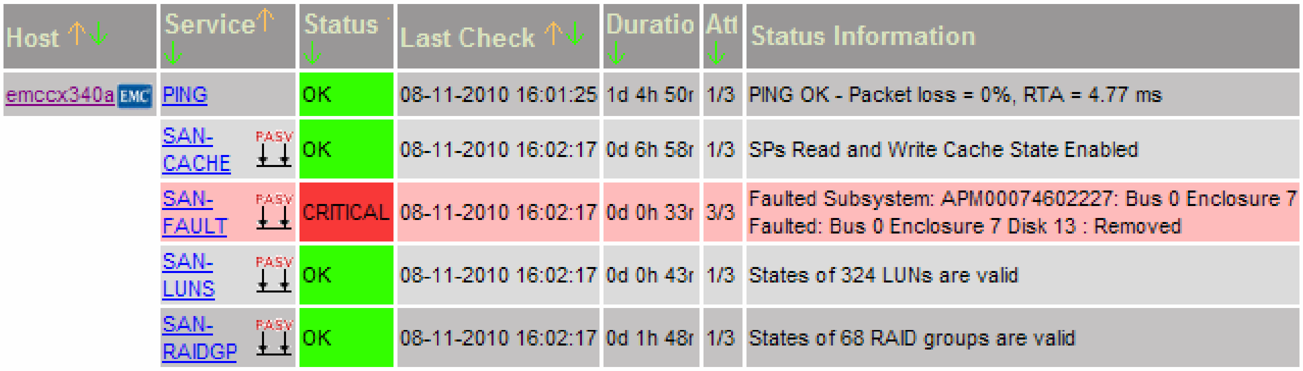
Sample CLARiiON command file

```
#Control file of EMC CLARiiON naviseccli commands for Nagios nsca
#Tab delimited, # in col 1 to skip
#Service       Command                  SP-name
SAN-FAULT      faults -list             cx4-240-spa
SAN-CACHE      getall -cache            cx4-240-spa
SAN-RAIDGP     getrg -type -state       cx4-240-spa
SAN-LUNS       getlun -type -state      cx4-240-spa
```

Sample Nagios service

```
Define  service{
        use                         generic-service ; template
        host_name                   cx4-240-spa
        service_description         SAN-FAULT
        contact_groups              sc-admins,sc-alerts
        check_command               check_dummy
        active_checks_enabled       0
        }
```

Sample Nagios display



Host ↑↓	Service↑↓	Status ↓	Last Check ↑↓	Duratio ↓	Att ↓	Status Information
emccx340a EMC	PING	OK	08-11-2010 16:01:25	1d 4h 50r	1/3	PING OK - Packet loss = 0%, RTA = 4.77 ms
	SAN-CACHE PASV	OK	08-11-2010 16:02:17	0d 6h 58r	1/3	SPs Read and Write Cache State Enabled
	SAN-FAULT PASV	CRITICAL	08-11-2010 16:02:17	0d 0h 33r	3/3	Faulted Subsystem: APM00074602227: Bus 0 Enclosure 7 Faulted: Bus 0 Enclosure 7 Disk 13 : Removed
	SAN-LUNS PASV	OK	08-11-2010 16:02:17	0d 0h 43r	1/3	States of 324 LUNs are valid
	SAN-RAIDGP PASV	OK	08-11-2010 16:02:17	0d 1h 48r	1/3	States of 68 RAID groups are valid

EMC Celerra NAS

EMC Celerra is a NAS system that can be configured to share storage with a CLARiiON SAN. Celerra is essentially a Linux-based NFS server with a CIFS listener. The Linux version is customized and hardened; it appears to be based on Red Hat. File services are implemented as a set of over 100 Linux commands, augmented with a web-based GUI.

Information sources

Perusal of the “man” pages for over 100 Celerra commands led to selection of several that provide useful monitoring information:

enclosure_status -v -e 0	list of status and alarms for both device heads
nas_fs -list	list of filesystems, which can be detailed with -size option
nas_server -list	list of servers (data movers) including state and type
nas_inventory -list	list of components with status and type

Installation and configuration

- Using the Celerra Manager GUI, create a monitoring account with operator privileges on the Celerra. This procedure assumes an account name *nagios*.
- Logon to the Celerra Control Station. As root, make a directory /opt/nsca and import required files from list below. Mark the Perl and shell scripts as executable. Change directory owner to the monitoring account.
    - celerra.cmd
    - nsca_celerra.cron
    - nsca_celerra.pl
    - nsca_celerra.sh
    - send_nsca.cfg
- Import a compiled send_nsca program from a similar Linux environment that has compilers. For example, a version compiled using gcc version 4.0.0 on Fedora Core release 4 (Stentz) worked successfully on EMC Celerra Linux release 2.0 (NAS 5.6.47).
- Edit the celerra.cmd file to create conforming Nagios service names.
- Edit the Perl program nsca_celerra.pl for compliance with directory and file names as needed. If desired, adjust the message severity between warning and critical levels.
- Edit the shell script nsca_celerra.sh for compliance with directory and file names as needed. On the send_nsca line, revise the Nagios host as needed. Edit the nsca configuration file send_nsca.cfg as needed.
- Run the shell script nsca_celerra.sh from the command line. It should produce a stream of messages, write a file nsca_celerra.dat in nsca tab-delimited format, and send nsca packets to the Nagios host. Correct any errors, and if necessary try a different version of send_nsca.
- Edit the sample crontab nsca_celerra.cron for compliance with directory and file names. Then edit the actual crontab for the monitoring account.
- On the Nagios host, create required entries for the Celerra and its services. See example following.

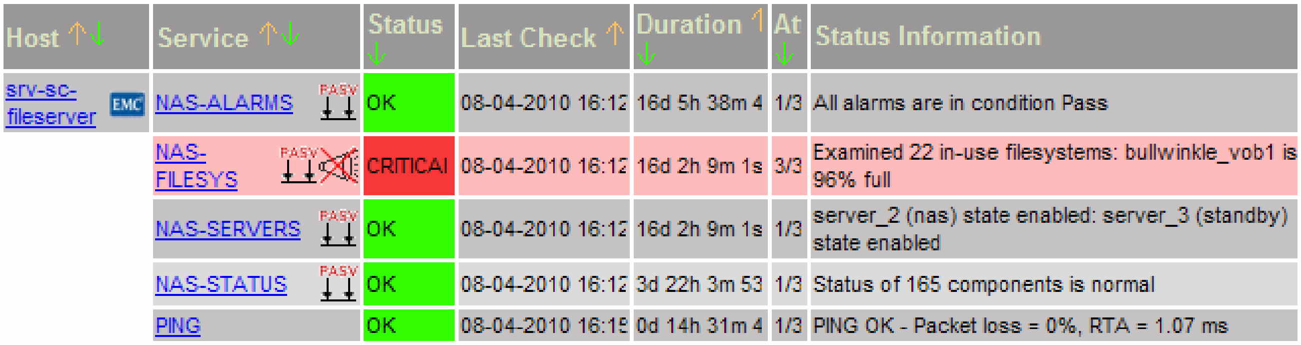
Sample Celerra command file

```
#Control file of EMC Celerra commands for Nagios nsca
#Tab delimited, # in col 1 to skip, alt source to read file instead
#Service        Celerra_nas_command
NAS-ALARMS      enclosure_status -v -e 0
NAS-FILESYS     nas_fs -list
NAS-SERVERS     nas_server -list
NAS-STATUS      nas_inventory -list
```

Sample Nagios service

```
Define  service{
        use                         generic-service ; template
        host_name                   srv-sc-fileserver
        service_description         NAS-ALARMS
        contact_groups              sc-admins,sc-alerts
        check_command               check_dummy
        active_checks_enabled      0
        }
```

Sample Nagios display



Host ↑↓	Service ↑↓	Status ↓	Last Check ↑	Duration ↑↓	At ↓	Status Information
srv-sc-fileserver	NAS-ALARMS	OK	08-04-2010 16:12	16d 5h 38m 4	1/3	All alarms are in condition Pass
	NAS-FILESYS	CRITICAL	08-04-2010 16:12	16d 2h 9m 1s	3/3	Examined 22 in-use filesystems: bullwinkle_vob1 is 96% full
	NAS-SERVERS	OK	08-04-2010 16:12	16d 2h 9m 1s	1/3	server_2 (nas) state enabled: server_3 (standby) state enabled
	NAS-STATUS	OK	08-04-2010 16:12	3d 22h 3m 53	1/3	Status of 165 components is normal
	PING	OK	08-04-2010 16:1	0d 14h 31m 4	1/3	PING OK - Packet loss = 0%, RTA = 1.07 ms

EMC RecoverPoint Appliance

EMC RecoverPoint Appliance (RPA) provides continuous replication of storage systems (including CLARiiON) to peers in remote locations. RPA is based on a customized and hardened 64-bit Linux implementation. Users get a restricted shell that allows only RPA commands, not operating system commands. So monitoring must be done externally.

Information sources

Perusal of the EMC RecoverPoint CLI Reference Guide led to selection of several commands that provide useful monitoring information:

• get_group_state	consolidation status for all groups
• get_monitored_parameters	list of any parameters exceeding limits
• get_system_status	list of any problems on the RPA system

Installation and configuration

- Select a convenient Linux host for monitoring; this can be the same host used to monitor the CLARiiON SPs.
- If not already done, install the gcc compilers and Perl on this host. Install Nagios NSCA per instructions on Sourceforge.
- Logon to the Linux host and create a corresponding local account, e.g. *nagios*. Suggested home directory is /opt/nsca.
- As root, make a directory /opt/nsca and import required files from list below. Mark the Perl and shell scripts as executable. Change directory owner to the local account.
    - recoverpoint.cmd
    - nsca_recoverpoint.cron
    - nsca_recoverpoint.pl
    - nsca_recoverpoint.sh
    - send_nsca.cfg
- On the Linux monitoring host, generate a public/private key pair using

  ```
  ssh-keygen -t rsa
  ```

  Accept the default key location and choose a blank passphrase. Find the generated key location, e.g. ../.ssh/id_rsa.pub. Display the key using cat or more.

- Select an RPA in the cluster for monitoring purposes. Use an ssh client (e.g. PuTTY) to connect to the RPA. Logon as *monitor* with password *monitor*; this account is present by default. At the restricted shell prompt, enter
  ```
  add_ssh_key
  ```
  Supply the key name *nagios*. Supply the public key one line at a time, using copy/paste, to avoid introducing line breaks in the middle of the key.
- On the Linux monitoring host, test the key with the command
  ```
  ssh -l monitor <RPA host name> get_system_status
  ```
  First time only, the host will prompt for confirmation to connect to the RPA host. Subsequent iterations should return status lines without prompting and without requesting a password.
- Edit the recoverpoint.cmd file to create conforming Nagios service names and point to the monitored RPA.
- Edit the Perl program nsca_recoverpoint.pl for compliance with directory and file names as needed. If desired, adjust the message severity between warning and critical levels.
- Run the Perl program nsca_recoverpoint.pl from the command line. It should produce a stream of messages and write a file nsca_recoverpoint.dat in nsca tab-delimited format. Correct any errors.
- Edit the shell script nsca_recoverpoint.sh for compliance with directory and file names as needed. On the send_nsca line, revise the Nagios host as needed. Edit the nsca configuration file send_nsca.cfg as needed.
- Run the shell script nsca_recoverpoint.sh from the command line. It should produce the nsca_clariion.pl outputs and send nsca packets to the Nagios host. Correct any errors.
- Edit the sample crontab nsca_recoverpoint.cron for compliance with directory and file names. Then edit the actual crontab for the monitoring account.
- On the Nagios host, create required entries for the RPA and its services. See example following.

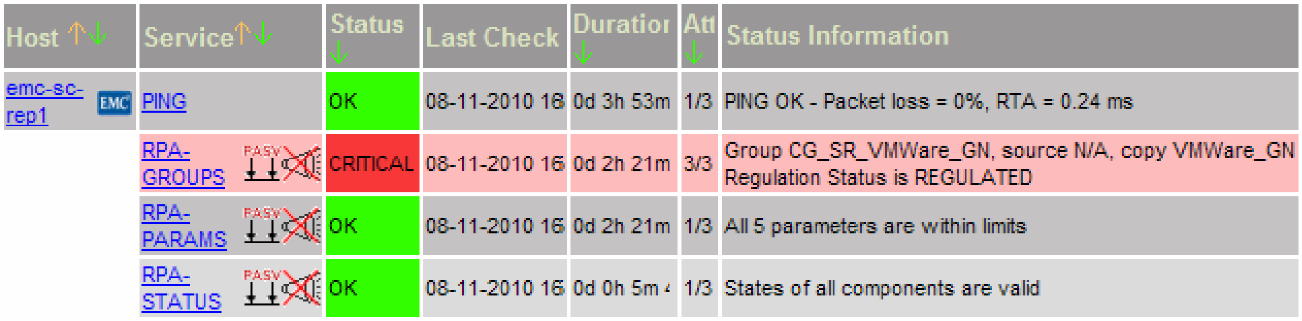
Sample RecoverPoint command file

```
#Control file of EMC RecoverPoint CLI commands for Nagios nsca
#Tab delimited, # in col 1 to skip
#Service        Command                        RPA-name
RPA-STATUS      get_system_status              emc-sc-rep1.symyx.com
RPA-GROUPS      get_group_state                emc-sc-rep1.symyx.com
RPA-PARAMS      get_monitored_parameters       emc-sc-rep1.symyx.com
```

Sample Nagios service

```
Define  service{
        use                         generic-service ; template
        host_name                   emc-sc-rep1
        service_description         RPA-STATUS
        contact_groups              sc-admins,sc-alerts
        check_command               check_dummy
        active_checks_enabled       0
        }
```

Sample Nagios display



Host	Service	Status	Last Check	Duratior	Att	Status Information
emc-sc-rep1	PING	OK	08-11-2010 16	0d 3h 53m	1/3	PING OK - Packet loss = 0%, RTA = 0.24 ms
	RPA-GROUPS	CRITICAL	08-11-2010 16	0d 2h 21m	3/3	Group CG_SR_VMWare_GN, source N/A, copy VMWare_GN Regulation Status is REGULATED
	RPA-PARAMS	OK	08-11-2010 16	0d 2h 21m	1/3	All 5 parameters are within limits
	RPA-STATUS	OK	08-11-2010 16	0d 0h 5m	1/3	States of all components are valid

References

Celerra Product Description Guide in http://www.comparex.sk/download/whitepapers/productguide.pdf

EMC Navisphere Command Line Interface in http://www.emc.com/microsites/clariion-support/pdf/300-003-628.pdf

EMC RecoverPoint CLI Reference Guide (Login account required) in https://powerlink.emc.com/nsepn/webapps/btg548664833igtcuup4826/km/live1//en_US/Offering_Technical/Technical_Documentation/300-010-642.pdf

GCC compiler installation in http://gcc.gnu.org/install/

Nagios documentation in http://support.nagios.com/knowledgebase/officialdocs

Nagios nsca procedure in http://nagios.sourceforge.net/download/contrib/documentation/misc/NSCA_Setup.pdf