nSANity
Users Guide

Contents

Revision History 3
What is nSANity? 4
How it works 4
Capabilities 5
Component 5
Protocol 5
Usage 6
URL Schemes 6
Authentication 7
API and Command Execution 7
Data Output 8
Role Based Access Control 8
Examples 10
Built-In Help 10
Collecting data from multiple Data ONTAP systems 11
Resources 12
MFC90 Requirement 12
FCINFO for Windows 2003 12

Revision History

Date	Comments
2009-04-19	Initial Version
2009-06-15	Update for nSANity 1.1.4
2010-04-09	Update for nSANity 1.1.14
2010-04-20	Update for nSANity 1.1.16
2010-06-26	Update for nSANity 1.1.18
2010-08-18	Update for nSANity 1.1.20
2011-03-25	Update for nSANity 1.2.0
2012-02-14	Update for nSANity 1.2.8

What is nSANity?

Diagnosing dynamic and complex SAN environments is a persistent challenge. Fault symptoms manifest in a variety of ways and rarely directly with the responsible faulty component. Identifying and isolating the faulty component can be tedious process which requires investigating several components in the SAN to build the complete picture. A typical investigation requires collecting configuration and diagnostic data from one end of the Fibre Channel fabric to the other - including the host servers, fabric switches and storage controllers.

nSANity is a one-shot diagnostic and configuration data collection utility. nSANity is a single solution that is capable of collecting diagnostic and configuration from all common SAN components including:

- Data ONTAP 7G
- Cisco MDS, Nexus and Catalyst switches
- Brocade switches
- McData switches
- QLogic switches
- VMware ESX
- Windows 2003, 2008
- Linux
- Solaris
- AIX, PowerVM
- HP-UX

How it works

nSANity is run by a Systems Administrator from a host that has access to each SAN component's management network interface. nSANity makes a connection to the SAN component over an IP network using protocols native to each component. Once a connection has been established and authorized, Data Collection will begin executing commands and API calls against the component. As each command and API call is executed, nSANity records the response data and error status. The response data is recorded in an output file for a follow-up analysis. The output file may be sent to technical support for analysis and fault isolation.

Capabilities

The following table details the connection protocols available for use with nSANity for each component type.

Component	Protocol
Data ONTAP 7G	Manage ONTAP API, SSH
Cisco Switches	SSH
Brocade Switches	SSH
McData Switches	SSH
QLogic Switches	SSH
Windows 2003, 2008	WMI
VMware ESX	SSH
Linux (various distributions)	SSH
Solaris 9, 10	SSH
AIX 5L, 6, PowerVM 2.0	SSH, Telnet
HP-UX 11i	SSH

SSH - Secure Shell
WMI - Windows Management Instrumentation

Usage

nSANity is run by specifying a set of connection definitions using the familiar URL format used in a web browser. nSANity supports a variety of options to alter the function of the program.

```
C:\> nSANity.exe --help

Usage: nsanity [options] url1 url2 url3 ...

Options:
  --version         show program's version number and exit
  -h, --help        show this help message and exit
  -i INPUT_FILE     Read URLs from an input file
  -T                disable verbose output during data collection
  -f NAME           Specify output file name string or case number. e.g.
                    20090221120000_<custom string>_nsanity.xml.gz
  -d DEST           Specify output directory
  -L COMPONENT      List commands to be run for each component type.
                    [ all | windows | vmware | ... ]
  -W                Write predefined commands to external file.
  -S SCOPE          Scope selection, general (default), survey, snapdrive,
                    snapmanager
  -x EXTRACT        Extract existing nSANity output to individual files and
                    directories.
  -u                Disable sudo support for non-root user accounts on
                    VMware, Linux, Solaris, AIX and HPUX. (default is
                    Enabled)
  -U                This option replaces sudo with sesu for platforms that
                    support seos.
```

URL Schemes

The standard URL format is *scheme://username:password@host*.

nSANity supports the following URL schemes:

ontap://
cisco://
brocade://
mcdata://
i10k://
qlogic://
windows://
vmware://
linux://
solaris://
aix://
powervm://
hpux://

Authentication

Authentication is handled by through the usual means of each component. When specifying a connection definition the appropriate credentials, username and password, must be provided. The password can be set directly on the command line or if a * character is given, then Data Collection will prompt for the password to be entered during the program's run. Connection definitions and credentials may optionally be saved in a file that nSANity can read when run with the *-i* option.

Data ONTAP allows any user defined through the *useradmin* interface to be authorized for management functions, including root or other user with the appropriate role based access control definitions.

Brocade, Cisco and McData all support role based administration in addition to using the default admin user. When defining role base access controls, be sure to allow enough privileges to execute all data collection commands.

Windows hosts can use specific user credentials or the current user's credentials. nSANity must be provided administrative user credentials for the remote Windows host. The credentials may be for the local Administrator, domain administrator or user that belongs to the administrator group. A username and password must not be specified for connections to the local system, they are only used when connected to a remote system.

UNIX and UNIX like systems including VMware, Linux, Solaris, AIX and HP-UX all have very similar authentication systems. nSANity must be provided with root credentials or the credentials of a user that can run all of the defined commands. nSANity can be provided a specific user name or it can take advantage of existing SSH public key infrastructure. Public key authentication allows nSANity to be authenticated without specifying a password. nSANity may also use sudo when a non-root user is specified. The requirements for sudo are as follows

- The user must be allowed to run sudo with the NOPASSWD flag
- /etc/sudoers must allow sudo without requiring a TTY, remove the *requiretty* setting.

API and Command Execution

nSANity executes a predefined list of API calls and commands against the indicated component and type of component. A list of commands can be displayed by running nSANity with the *-L*. The *-L* options may be run with one additional argument that specifies the component type or *all* for all component types.

The predefined command list may be overridden by placing a file called *commands.xml* in the current working directory. nSANity will always check for the presence of *commands.xml* and override the predefined list of commands. The purpose of *commands.xml* is to allow the

user to augment or eliminate the current command list in favor of a customized command list without the need to rebuild and republish the application.

Data Output

Once nSANity has completed, it will write all collected data to a single output file in the user's current working directory. The name of the output file is displayed when the program completes and is in the form of *<date>_<custom string_>.dc.xml.gz*. The default name and directory where the file is saved can be controlled through the *-f* and *-d* options respectively. Specifying the -f option fills in the *custom string* field with the desired string, such as a case number. The *-d* option changes the save location to the directory of choice. If the output directory does not exist, then it will be created automatically. The output is also GZIP compressed which dramatically shrinks the size of the data.

The compressed data file is often small enough to be emailed directly to technical support for analysis. If the output file is larger than 7MB, then technical support will provide details for an alternate mechanism to upload the output file.

The contents of the output file are stored in a simple XML schema and may be reviewed in any XML viewer application such as a web browser. nSANity can also extract the data, creating a file for each command output for each component. Use *nsanity –x <filename.xml.gz>* to perform the extraction.

Role Based Access Control

Many component types support role based access control (**RBAC**). RBAC is useful when different components are managed by different organizations. In many IT departments the storage controller administrators, fabric administrators and host administrators may each independently manage their components. RBAC can be employed by each group to allow limited, read-only, access to each group's components. An RBAC user can be defined on the storage controllers, fabric switches and host servers to allow members of the other groups to collect important diagnostic and configuration data without having to directly engage the other groups.

When implementing RBAC users it is import to refer to the respective vendor documentation, since each component's vendor implements RBAC in a different way. The follow list of resources may be useful when setting up RBAC users.

Scope Control

nSANity can alter the data collection of components based on the Scope option. The scope comes in the following forms, each adding additional commands when run against certain components.

The scopes are:

- Survey – Provides commands to collect the vital product data including configuration information
- General – A detailed yet generalized data collection, adding log files and diagnostic data to the commands provided by the survey scope.
- SnapDrive – Adds data collection specific to the SnapDrive product including log files and debug data
- SnapManager - Adds data collection specific to the SnapManager product line including log files and debug data

To examine the commands that will be run use the command line option –L all. The –L option instructs nSANity to list all of the commands for each component type and the scope in which they will run.

Examples

Built-In Help

C:\> nsanity.exe

nSANity version: 1.2.0
Using external commands from ./commands.xml
Running with scope general
Usage: nsanity [options] url1 url2 url3 ...

Options:

--version	show program's version number and exit
-h, --help	show this help message and exit
-i INPUT_FILE	Read URLs from an input file
-T	disable verbose output during data collection
-f NAME	Specify output file name string or case number. e.g. 20090221120000_<custom string>_nsanity.xml.gz
-d DEST	Specify output directory
-L COMPONENT	List commands to be run for each component type. [all | windows | vmware | ...]
-W	Write predefined commands to external file.
-S SCOPE	Scope selection, general (default), survey, snapdrive, snapmanager
-x EXTRACT	Extract existing nSANity output to individual files and directories.
-u	Disable sudo support for non-root user accounts on VMware, Linux, Solaris, AIX and HPUX. (default is Enabled)
-U	This option replaces sudo with sesu for platforms that support seos.

url specifies the connection definition for each storage controller,
switch or host with the format scheme://username:password@host

Storage Controllers
ontap://username:password|*@hostname[:port] (Manage ONTAP API and SSH)

Switches
cisco://username:password|*@cisco_switch (IOS, SANOS, NXOS) (SSH)
brocade://username:password|*@brocade_switch (SSH)
i10k://username:password|*@i10k_switch (McData Intrepid 10k EOSn) (SSH)
mcdata://username:password|*@mcdata_switch (EOS) (SSH, Telnet)
qlogic://username:password|*@qlogic_switch (SSH, Telnet)

Hosts
windows://[[domain|host\]username[:password | *]@]windows_server (WMI)
vmware://[username[:password | *]@]server[:port] (SSH)
linux://[username[:password | *]@]server[:port] (SSH)
solaris://[username[:password | *]@]server[:port] (SSH)
aix://[username[:password | *]@]server[:port] (SSH, Telnet)
powervm://[padmin[:password | *]@]server[:port] (SSH, Telnet)
hpux://[username[:password | *]@]server[:port] (SSH)

Specify '*' for the password to be prompted.

Use quotes around each connection definition if specifying a password with special characters to avoid shell expansion and interpretation.

username and password are optional in many cases. The program will use the username of the current user and prompt for a password when necessary.

Connections to Linux, VMware, Solaris, AIX and HP-UX can use SSH public key authentication if setup in your user's environment.

WMI connections to Windows hosts will use the active user's credentials if a domain\username and password are not present in the URL. Connections to the local Windows host must not contain a a username or password.

Multiple urls may be used on the command line to collect data from multiple components at one time.

c:\> nsanity ontap://root:*@storage1 ontap://root:*@storage2
cisco://admin:*@switch windows://administrator:*@windows_host

The scope option, -S, controls which data will be collected.

'survey' is the most light weight data collection used for obtaining brief configuration
'general' adds diagnostic data and logs to what is collected in survey mode
'snapdrive' adds SnapDrive specific diagnostic log files to general mode
'snapmanager' adds SnapManager product log files to snapdrive mode

Collecting data from multiple Data ONTAP systems

C:\> nsanity.exe ontap://root:*@10.61.65.160 ontap://root:*@10.61.65.165

nSANity version: 1.1.20
Using internal command definitions
Enter password for root@10.61.65.160:
Enter password for root@10.61.65.165:
Starting collection @ 2010-04-20 13:43:28
Beginning data collection for Data ONTAP @ 10.61.65.160
Data collection complete for 10.61.65.160
Beginning data collection for Data ONTAP @ 10.61.65.165
Data collection complete for 10.61.65.165
Ending collection @ 2010-04-20 13:47:27
Collected data saved to file, 20100420174328_nsanity.xml.gz

Resources

MFC90 Requirement

The Windows executable requires MFC90 Runtime libraries, which are included with Windows 7. If your Windows host does not have the required libraries then they may be downloaded from Microsoft at the following URL.

http://www.microsoft.com/downloads/en/details.aspx?FamilyID=a7b7a05e-6de6-4d3a-a423-37bf0912db84

FCINFO for Windows 2003

Microsoft Windows 2003 hosts require an additional package in order to allow complete data collection of HBA information. The Microsoft package is call *fcinfo*, which provides the HBA-API and WMI classes to access the API. This package may be downloaded directly from Microsoft, http://www.microsoft.com/downloads/details.aspx?familyid=73d7b879-55b2-4629-8734-b0698096d3b1&displaylang=en