



Global Distributed Storage Engineering

PowerBroker Implementation for SAN

GDSE-05006-S

Solution Design

1.4

1. Version Control

Version	Date	Author	Reviewed	Comments
0.1	24-Jun-05	Ludwig Derodel	GDSE	Draft
1.0	22-July-05	Ludwig Derodel	Storage Integration, GDSE	Final
1.1	09-Aug-05	Ludwig Derodel	Storage Integration	Updated mailing script
1.2	28-Sept-05	Ludwig Derodel	Stephen Hopkins	Fixed typo in pd.conf
1.3	09-Nov-05	Gary Johnson	Ludwig Derodel	Added HDS CLI, CCI
1.4	17-Oct-06	Stephen Hopkins	Ludwig Derodel	Added Centera CLI

2. Legal Notice

Information contained herein is for Internal Use and may be used only for business purposes authorized by Citigroup.

CITI, CITIBANK, CITICORP, CITIGROUP and the Umbrella Device are trademarks and service marks of Citicorp and are used and registered throughout the world. Citibank, Citicorp, Citigroup and their subsidiaries also claim rights in certain other trademarks and service marks contained in these pages.

3. Contact

Please address any comments or questions regarding the information contained in this document via email to the “*GT Global Distributed Storage Engineering” distribution list or alternatively to gdse@citigroup.com.

4. Latest Versions

It is recommended that the latest version of the document be used from the GDSE website <http://gdse.ny.ssmb.com>. Newer versions may include expanded scope, or comments on issues discovered after deployment.

5. Table Of Contents

1. VERSION CONTROL	2
2. LEGAL NOTICE.....	3
3. CONTACT.....	3
4. LATEST VERSIONS	3
5. TABLE OF CONTENTS	4
6. EXECUTIVE SUMMARY	5
7. SCOPE	5
8. SOLUTION DESIGN	5
9. PROBLEM ANALYSIS	6
9.1. CURRENT ENVIRONMENT.....	6
9.2. LIMITATIONS WITH THE CURRENT ENVIRONMENT	6
10. GOALS.....	6
11. ALTERNATIVE SOLUTIONS.....	6
12. FUTURE DIRECTIONS.....	6
13. REFERENCES.....	7
14. APPENDICES.....	8
14.1. APPENDIX A: SAMPLE PB.CONF	8
14.2. APPENDIX B: LOG ROTATION/ARCHIVAL SCRIPT.....	10
14.3. APPENDIX C: SAMPLE POWERBROKER CONFIGURATION.....	13

6. Executive Summary

PowerBroker has been a standard security product at Citigroup for a number of years. PowerBroker main functionalities include user access control as well as audit capabilities at a user level. Given the above functionalities, GDSE has chosen PowerBroker as the tool to enhance security and logging capabilities of command-line products such as EMC Solutions Enabler, HDS Device Manager and EMC Centera CLI. EMC Solutions Enabler is a tool that is used to manage all the Symmetrix storage arrays. HDS Device Manager is a tool that is used to manage the USP storage arrays. Centera CLI is a tool used to manage EMC Centera storage arrays.

7. Scope

The scope of this project is limited to the environment listed below

- Applies only to a SAN Control Workstations running Solaris 2.8
- Applies to EMC Solutions Enabler CLI commands, HDS Device Manager CLI commands or Centera CLI commands.

8. Solution Design

The SAN implementation of PowerBroker is a fairly simple design. This implementation intends to leverage existing PowerBroker infrastructure in order to support the SAN Control Workstations. However, if such an infrastructure were not currently in place, a new environment would need to be deployed for this implementation. ([see Appendix C](#))

PowerBroker is a Client/Server architecture design where the client must authenticate to a master server in order to gain access to define commands. Furthermore, keystrokes logging can be enabled where all commands and arguments are sent to a log server in real-time.

Below are the steps and components necessary to setup PowerBroker

1. Install the PoweBroker Client on the SAN Control Workstation (**PBclnt 3.5.2-01_A1**) according the documentation released by Unix Engineering ([see References](#))
2. If using existing PowerBroker Master and Log servers, copy pb.key to PowerBroker client. If using separate environment, obtain and download the server software of PowerBroker (**PBmastrS8 v3.5.2-01_A0**) and install it according to documentation released by Unix Engineering. Please note that is installing on SSM system that does not have USM package install, please install it first and then remove the USM package after installing PowerBroker Master. ([see References](#))
3. Setup **pb.conf** on Master Server to grant access to PowerBroker client ([see appendix](#) for SAN specific access)

4. Setup **iolog** directory according to the entry in **pb.conf** for keystrokes logging. (used for keystroke logging)
5. All user ids requiring access can be managed by creating a local or netgroup.
6. Add client IP address in **/etc/host** file if necessary.
7. Setup environment variable in **pb.conf** file on Master Server.
8. Only allow symcli or other SAN related commands to run as root.

9. Problem Analysis

9.1. *Current Environment*

EMC solutions enabler, HDS Device Manager (CLI) and EMC Centera CLI are all command line tools used to control specific storage arrays and SAN infrastructure. All of these tools lack extensive security and logging functionality to meet Citigroup's security requirement.

9.2. *Limitations with the current environment*

- Lack of security restriction on Solutions Enabler command.(commands can only be run as root)
- Lack of audit logs for each commands

10. Goals

- Implement a compensating control environment to address existing security gaps from EMC Solutions Enabler.
- Implement a compensating control environment to address existing security gaps from HDS Device Manager.
- Implement a compensating control environment to address existing security gaps for EMC Centera CLI.

11. Alternative Solutions

Scsudo is a tool very much like PowerBroker, which provides functional access to Solaris systems. However, the storage management group has stated that the current implementation of Scsudo has introduced significant operation overhead into their daily functions.

12. Future Directions

The current implementation of PowerBroker does not support a log review process through a GUI or a Web Browser interface. All log review functions in the interim will be done using a command line method until Unix Engineering has

certified the Web Browser. This functionality will be incorporated in the next release of the software.

Unix Engineering has recently announced that PowerBroker has been chosen as a strategic tool to address such security functions mentioned above.

13. References

Document	Location
PowerBroker Client v3.5-2	PBclnt PV3.5.2-01 A1 DV1.0
PowerBroker Master v3.5-2	PBmastrS8 v3.5.2-01 A0

14. Appendices

14.1. Appendix A: Sample pb.conf

```
#####
# The example below explicitly lists all the admin users as well the
# EMC related commands.
# NIS groups can be used to simplify user or group access based on
# existing NIS tables.
# The ilog parameter specifies where to place the keystroke log file.
# A log file is created for each user and for each command.
#
# Any existing scripts will need to be added to the pb.conf file as
# well, to ensure a seem less integration.
#
#
# All EMC and HDS related commands
adminusers = {"ludwig", "am37267", "mb27571", "sk34158"};
Commands = {"fpath", "fcompare", "fconfig", "vcfind", "srdf",
"symacl", "symapierr", "symbcv", "symcfg", "symcg", "symchg",
"symchksum", "symcli", "symconfigure", "symdev", "symdg", "symdrv",
"symgate", "symhostfs", "syminq", "symioctl", "symlabel", "symld",
"symlmf", "symlv", "symmir", "symerase", "symhost", "sympd", "symqos",
"symrdb", "symrdf", "symreplicate", "symreturn", "symrslv", "symstat",
"symvg", "symaudit", "symclone", "symdisk", "symevent", "symmask",
"symmaskdb", "symoptmz", "symsnap", "powermt", "inq", "hc", "HC",
"HCLI.sh", "hcli.sh", "HiCommandCLI.sh", "raidscan", "paircreate",
"pairevtwait", "pairedisplay", "pairresync", "pairsplit", "pairmon",
"pairvolchk", "paircurchk", "pairsyncwait", "horcmstart.sh",
"horcmshutdown.sh", "horcctl", "inqraid", "mkconf", "horctakeover",
"raidvchkset", "raidvchkdsp", "raidvchkscan"};
#
# Centera related commands
#
Centera_mgmt_hosts = {"Centera-mgt1.nsroot.net", "Centera-
mgt2.nsroot.net"} ;
Centera_users = {"xx12345", "yy67890" };
Centera_progs = { "CTICenteraCli" };

#logstdoutlimit = 200;
#logstderrlimit = 4096;

if (user in adminusers && basename(command) in Commands) {
runuser = "root";
rungroup = "!g!";
rungroups = {"!G!"};
runcommand = basename(command);

ilog = logmktemp("/var/log/pb/pb."+ user + "." + basename(command) +
".XXXXXX");
```

```
setenv("PATH",
"/bin:/usr/bin:/usr/sbin:/export/opt/emc/SYMCLI/V5.5.0/bin:/export/opt/
Navisphere/bin:/export/opt/HiCommandCLI:/export/opt/HORCM");
accept;
}

if ((user in Centera_users) && (basename(command) in Centera_progs) &&
(submithost in Centera_mgmt_hosts)) {

runuser = "root";
rungroup = "!g!";
rungroups = {"!G!"};
runcommand = basename(command);

iolog = logmktemp("/var/log/pb/pb."+ user + "." + basename(command) +
".XXXXXX");

setenv("PATH",
"/bin:/usr/bin:/usr/sbin:/export/opt/EMC/Centera/3_0/SystemOperator/bin
");
accept;
}

#
####
```

14.2. Appendix B: Sample log rotation/archival script

The script below should be setup to run in "cron" daily and performs 2 functions:

- 1) run pblog on the pb.eventlog to create an ascii file format that can be fed into a log collector like E-Trust Audit.
- 2) Archive the keystroke logs into a backup directory using tar and compress.

```
#!/bin/ksh
# mail_plog.ksh
#
# Author: Ludwig Derodel
# Contact info: ludwig.derodel@citigroup.com
#
# This script parses the archive copy of the pb.eventlog.YYYYmmdd. To run the script for another
# day simply modify the ELOG parameter to reflect the filename. The script also sends out
# an email to a desired email group or individual for review.
# All the server names must be defined in "CWS[*]" fields.
# Furthermore, a "pblog" entry line must be created for each SAN server
# i.e. pblog -p 'sprintf("%s %s %-8s %-10s %-40s %s %s", date, time, user, submithost, argv, event,
# iolog)' -c'submithost=="sangs7"' > .sangs7.log
#
# Notice that both "submithost" argument as well as the file name contains the exact server name
# defined in the "CWS[*]" field.
#
#
#
TZ=GMT+24
TODAY=`date +%Y%m%d`
SERVER=$(hostname)
BINPATH=/export/opt/pb/sbin
ELOGDIR=/var/opt/pb
KLOGDIR=/var/log/pb
MAIL_LOG=/var/opt/pb/.pb_server.log
ELOG=pb.eventlog.${TODAY}
NO_DATE=`date "+%Y/%m/%d %H:%M:%S%"`
#
# Define Control Workstation server names below
CWS[0]=sancw01
CWS[1]=sancw02
CWS[2]=sancw03
CWS[3]=sancw04
CWS[4]=sancw05
CWS[5]=sancw06
CWS[6]=sancw07
CWS[7]=sancw08
#
# End of Definition
#
#####
```

```
# This is the function for sending out the mail. The "To:" field should be modified accordingly.
mail_users() {
/usr/lib/sendmail -oi -t <<EOM
From: PBroker Log Server
To: ludwig.derodel@citigroup.com
Subject: PowerBroker log for $SERVER
`cat $MAIL_LOG`

EOM
}
#####
#
#
# Checks to see if the log directory exists
if test -d $ELOGDIR
then
    cd $ELOGDIR
    cp /dev/null $MAIL_LOG
    if test -s $ELOG
    then
        $BINPATH/pblog -p 'sprintf("%s %s %-8s %-10s %-40s %s %s", date, time, user,
submithost, argv, event, iolog)' -c'submithost=="sancw01"' > .sancw01.log
        $BINPATH/pblog -p 'sprintf("%s %s %-8s %-10s %-40s %s %s", date, time, user,
submithost, argv, event, iolog)' -c'submithost=="sancw02"' > .sancw02.log
        $BINPATH/pblog -p 'sprintf("%s %s %-8s %-10s %-40s %s %s", date, time, user,
submithost, argv, event, iolog)' -c'submithost=="sancw03"' > .sancw03.log
        $BINPATH/pblog -p 'sprintf("%s %s %-8s %-10s %-40s %s %s", date, time,
user, submithost, argv, event, iolog)' -c'submithost=="sancw04"' > .sancw04.log

        for server in ${CWS[*]};
        do
            if test -s ${server}.log
            then
                /bin/cat ${server}.log >> $MAIL_LOG
            else
                printf "%s %s %-8s %-10s %-40s %s \n" ${NO_DATE} root
${server} "NO CHANGE" Accept >> $MAIL_LOG
            fi
        done

    else
        exit 0
    fi

else
    exit 0
fi

mail_users
```

Sample output of the pb_etrust.log

The pb_server.log file is created every night by a script called *mail_pblog.ksh*. This script basically parses the pb.eventlog file and sends the output into pb_etrust.log file.

Date	Time	User	Server	Command	Status	lolog
2005/07/28	16:28:53	ludwig	sangs10	test	Reject	
2005/07/28	16:29:09	ludwig	sangs10	ls	Reject	
2005/07/28	17:02:47	ludwig	sangs10	symdg list	Reject	
2005/07/28	17:03:09	ludwig	sangs10	symdg list	Accept	/var/log/pb/pb.ludwig.symdg.bSaO7y
2005/07/28	17:04:49	ludwig	sangs10	symdg list	Accept	/var/log/pb/pb.ludwig.symdg.v6aa9y
2005/08/08	15:17:39	root	sancw02	NO CHANGE	Accept	
2005/08/08	15:17:39	root	sancw03	NO CHANGE	Accept	
2005/08/08	15:17:39	root	sancw04	NO CHANGE	Accept	
2005/08/08	15:17:39	root	sancw05	NO CHANGE	Accept	
2005/08/08	15:17:39	root	sancw06	NO CHANGE	Accept	
2005/08/08	15:17:39	root	sancw07	NO CHANGE	Accept	
2005/08/08	15:17:39	root	sancw08	NO CHANGE	Accept	
2005/08/08	15:17:39	root	sancw09	NO CHANGE	Accept	

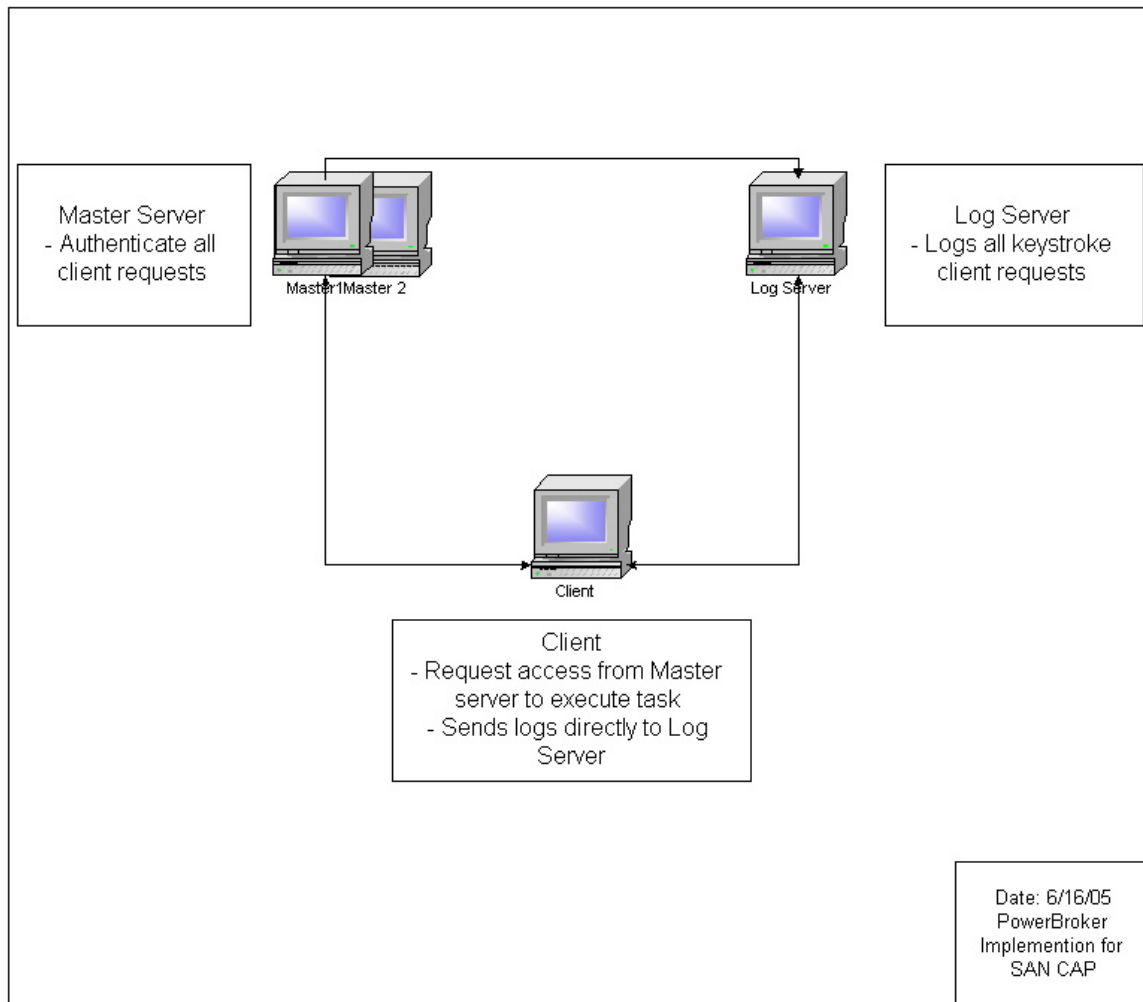
Pbrun example

Once PowerBroker is enabled on the client, the operator must precede each privilege commands with "**pbrun**".

i.e. pbrun symcfg list or pbrun symrdf -g testdg query

Further considerations need to be made in order to allow any existing perl or shell scripts. The main reason is that shell scripts can be modified if not permission accordingly.

14.3. Appendix C: Sample PowerBroker Configuration



- The Master and Log Servers can be combined into one server if hardware resource is an issue