SANS ANALYST PROGRAM

February, 2006

Sponsored by Symark

PowerBroker vs. Sudo

By Jerry Shenk & Steve Mancini

Contents

Author Bios

Jerry Shenk: Jerry currently serves as Senior Analyst for the SANS Institute. Since 1984, he has consulted with companies, financial and educational institutions on issues of network design, security, forensic analysis and penetration testing. His experience spans small home-office systems to global networks. Along with some vendor-specific certifications, Jerry holds 5 GIAC GOLD certifications: GCIA, GCIH, GCFW, GSNA and GCFA: all completed with honors.

Steve Mancini: Steve currently works as a Senior Information Security Analyst for the Intel Corporation where he is the technical lead to a business unit's risk management team. In his spare time he volunteers with the Hillsboro Police Department's Police Reserve Specialists and serves as the police department's principle digital forensics examiner and lab technologist. Steve has obtained 3 GIAC certifications over the last several years: GSEC, GCIH, and GSNA (honors).

Executive Summary

Symark PowerBroker provides centralized Authorization management across a network of UNIX/Linux hosts. There are two primary driving forces behind the need for PowerBroker in a UNIX/Linux environment; restricting what commands can be run by users and maintaining an audit trail of what users have done or have tried to do. PowerBroker allows for policy defined authorization controls which allow administrators to define where and when their end-users can access other accounts they are authorized to use - up to and including root, the traditional "keys to the kingdom". It does so in a fashion that not only restricts access, but also executes with sufficient logging to satisfy the most stringent logging requirements.

Many UNIX/Linux administrators have historically used sudo as a means to provide separation of duties where root level permissions are needed to perform certain duties. Sudo is a free utility that is available for most UNIX variants that allows a user to assume the privileges of another user, including the super-user account. Whereas sudo is often bundled with many UNIX operating systems free of charge, PowerBroker is a commercial program that also allows similar but expanded functionality. PowerBroker offers more granularity in access control, extensive logging options, and the ability to verify a program's validity.

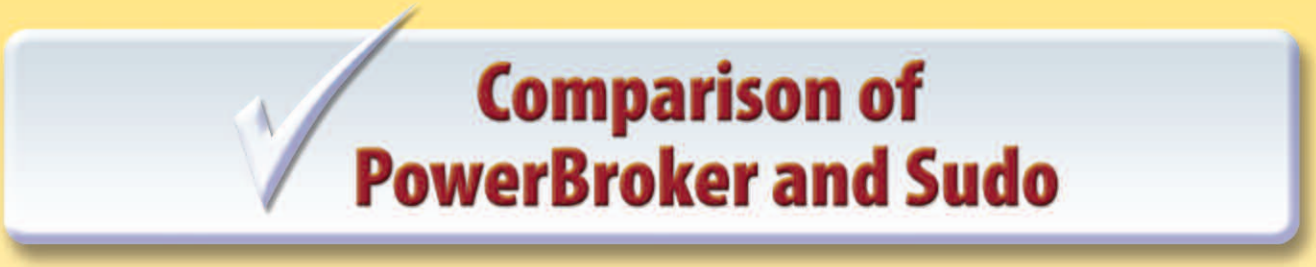
PowerBroker's session level keystroke logging feature provides several benefits when implemented in an enterprise wide deployment. Keystroke logs are protected through access control and are stored in plain text data format by default. The logs can also be encrypted. These logs can provide a detailed forensics trail of commands executed under PowerBroker (which are critical to any detailed incident analysis). These logs can demonstrate a company's commitment to compliance with several US regulatory acts. The Sarbanes-Oxley Act (SOX) of 2002[1] states that "…information related to any audit report, in sufficient detail to support the conclusions reached…" be maintained for 7 years. HIPAA, the Graham-Leach-Bliley Act of 1999, the VISA Cardholder Information Security Program, and other regulations recommend the retention of data to prove the separation of duties and verify what operations were performed by any given individual.

One risk associated with granting users increased access when running specific programs is that many programs can be exploited to allow the user to run other programs they may not be authorized to run. If the initial program is being run as root, then any subsequent program inherits that same level of access with its special privileges. As part of the PowerBroker suite, Symark provides replacements for several of the programs that are most often associated with this exploit thereby decreasing the associated risk. PowerBroker also has the ability to verify the called program or script and restrict its access if the checksum of the program is different from the registered version. This keeps an operator from running a modified program.

The greater functionality, granularity, and versatility of PowerBroker does come at a cost which large UNIX installations will have to take into consideration; PowerBroker is licensed on a ***per-node basis****. In this paper, we will explore the places where PowerBroker's capabilities exceed sudo's capabilities. This paper will help companies determine where the implementation of PowerBroker will enhance their security posture. Ultimately each company must decide if the risks addressed by PowerBroker as well as its contribution toward compliance with US regulations, outweigh the potential cost of installation.*

[1] http://news.findlaw.com/hdocs/docs/gwbush/sarbanesoxley072302.pdf

In a UNIX-type system, root access provides full access to all system processes and files. One of the basic tenets of a sound security posture is the Principle of Least Privilege which advocates that a user be granted only the minimum access necessary to complete a legitimate task. When a user must be given access to root privileges, an administrator has a limited number of choices. One option would be to give user the root password and let them log in as root whenever they need to run an application that requires root access. This provides absolutely no control or accountability and violates the Principle of Least Privilege. Another option would be to provide them access to the UNIX/Linux su command to temporarily run that one command. Unfortunately even this option requires that the user be given root's password. A final option would be to write suid programs allowing the end user to execute the necessary commands. This can result in excessive overhead creating the scripts in such a fashion that they cannot be exploited. In each of these cases there is an absence of command logging. This introduces challenges to forensic procedures related to investigating an issue involving any of these options.



Comparison of PowerBroker and Sudo

PowerBroker and sudo fall into the same general category of access management tools for UNIX environments. Both systems support a similar OS base. Both PowerBroker and sudo can control and log what a user can do on a system. PowerBroker is a commercial application with technical support options while sudo is an open source program with support coming primarily from the user community, newsgroups, and electronic and print articles. Sudo is designed for a single-computer implementation although a large-scale deployment is possible when used in conjunction with a suitable file distribution system. PowerBroker is designed for installation on a large number of machines. All references and comparisons are based upon PowerBroker version 4.0.0-10 and Sudo version 1.6.8p8.

The biggest difference between the two programs is the sheer volume of options. With PowerBroker there is a full suite of options for controlling how programs run. Some of these additional options are; how much logging will be done, what environmental variables will be inherited, what systems the commands will be executed on, when the program can be used, and what if any environment changes should be made to run the program.

Both systems can do keystroke logging; with PowerBroker, those logs can be encrypted and that functionality is built into the program while sudo logs are in clear text by default and the keystroke logging is not integrated into the application. Keystroke logging can be added to sudo by using add-on programs[2].

One of the known issues with sudo is that many programs will allow an experienced user to escape out of a program. For example, in `/usr/bin/less` *or* `/bin/vi`*, a user can "bang out" by using an exclamation and calling a shell (!sh) from the colon prompt in the application. PowerBroker has hardened utilities to replace vi, less and a number of other utilities that have similar functionality. One possible solution for sudo users is to use the* ***noexec*** *switch. This is included in most recent[3] versions of sudo but does have compatibility issues. The* ***noexec*** *switch requires that sudo be compiled to support it and that the OS also support it. Most current Operating Systems do support it, AIX and Unixware are two exceptions.*

[2] http://sourceforge.net/projects/sudosh

[3] The noexec switch was originally added in 1.6.8, released in Aug '04

Centralized Administration

PowerBroker makes use of a centralized "Master Host" which lends itself to organizations with a large installed base of UNIX/Linux computers. Within the central administration paradigm PowerBroker allows for the existence of numerous policy files which can all be incorporated into an enterprise level mapping of access policies. This can be particularly useful in an environment where different administrators control access to one or more systems. The use of separate files allows for delegation of control without having to provide configuration management control of a central file (and thus to systems they do not administer). Each delegated administrator can be given access to their respective configuration file and thereby control access to specific computers in the environment. Configuration settings also provide for the existence of redundant master hosts to ensure that no single point of failure is introduced into the environment. These other master hosts can be positioned throughout the network where they can respond to local authorization requests. Such a design can be extremely valuable to a company with a large WAN or global presence where a lapse in network connectivity could otherwise impact usage. By establishing redundant local Master hosts the service uptime is protected against such an outage.

Client-Server Architecture

Another benefit of PowerBroker is inherent in the client-server architecture design. All policy files are controlled through a central server (or servers if deployed with redundancy). This allows for central administration of access so that if a client is compromised, PowerBroker and its configurations cannot be modified. Depending on the deployment scheme for sudo, changes to its configurations must be pushed out to the local sudoers file on each affected host which would then be vulnerable to integrity attacks targeted against the sudo configuration file. The PowerBroker policy files can be further protected by encrypting them.

Logging Features

PowerBroker has extensive logging capabilities. The main log is the event log which records the date and time of the request, what programs a user attempts to run, what user requested the program, what computer they were on, where they requested the program to be executed, if the request was accepted or rejected, who they are running the program as (root, another provided account or another user account (the runuser)) as well as many other program and user-defined variables.

There is also an iolog function which can log individual keystrokes as well as what is displayed on the screen. There are a number of options for tuning the amount of data that will be logged. In some cases, this type of logging would be quite helpful but in other cases, there could be some serious confidentiality issues. PowerBroker makes it possible to iolog only the specific programs and users where this is in keeping with the company's security and privacy policies. The keystroke logs (iologs) are stored in distinct files for each logged session. This data is stored independently from the event logs.

PowerBroker also allows for the designation of a log server to be defined in the policy files as well as the Master host's configuration file. This allows for PowerBroker logs to be routed either to specific hosts or to a central logging system. Sudo's reliance upon syslog requires that non-sudo logs which are logged to the same facility/level be sent as well. PowerBroker can be configured to log eventlog data separately from other data or integrated into the same file.

Data File Encryption

PowerBroker can optionally encrypt the policy files, event logfiles, and keystroke log files (iolog). By default, the logfiles and policy files are stored in human-readable text. In highly secure environments, it is desirable to encrypt as much data as possible for added precaution against accidental disclosure or intentional compromise. When using sudo, all log data is unencrypted unless the standard syslog services have been upgraded.

Hardened Shells

PowerBroker includes two hardened shells to replace the shells that are normally included with UNIX-type operating systems. These PowerBroker shells are configured with policy files. By default, the policy file is pbshells.conf. It is recommended to block certain dangerous commands from most users but that is fully configurable in the policy files. As with the rest of the PowerBroker system, there are a number of options for configuring the logging including full session logging.

These hardened shells can be set up as drop-in replacements for the standard shells by modifying the shell portion of a user's entry in the `passwd` file. This is normally done by changing the last field in the line of `/etc/passwd` that relates to a particular user. The following example is an entry for a Linux system. Check your system documentation and understand exactly what's going on before duplicating this setup. In this case, /bin/bash is the shell that will be run when testuser logs in but when testpbuser logs in, the pbksh shell will be used instead. Testpbuser will be subject to the PowerBroker policies.

```
testuser:x:521:521:Test User:/home/testuser:/bin/bash
testpbuser:x:522:522:Test PBUser:/home/testpbuser:/usr/local/bin/pbksh
```

Host Redundancy

PowerBroker can be set up with redundancy. Both the master host and the log host can be configured to support redundancy. This is a significant advantage for environments where availability is a high priority. These redundant hosts can be set up with a variety of timing configurations based on the network design. If there is primarily a single host that should handle all requests, the default configuration allows for that host to reply and if there is no reply within 500 ms, then the next host is sent the request.

Checksum Verification

PowerBroker can verify the checksum of a file prior to permitting it to be run. By matching the checksum of a file to a previously stored checksum, PowerBroker can prevent Trojan programs or rogue program modification from causing a security compromise. One way attackers often attempt to compromise a system is by replacing a critical file with a similar file configured to facilitate their control of a system. These programs have been used in the past to transmit credit card information to the attacker, extract usernames and passwords as well as a variety of other violations of availability, confidentiality, and integrity. These changes can be attempted by either an insider with criminal intent or accidentally by a web or e-mail borne attack. There is no comparable functionality included in sudo.

Other Features

The granularity of PowerBroker's configuration allows for policies to be defined for the program that is attempted to be run, the users seeking to run them, the systems they seek to run them on, the user they wish to run them as, and also provides time based rules. Sudo does not provide the ability to accept or reject requests based upon the time and does not have the ability to run programs on another system. Time-based policy configurations are valuable to companies which seek to either restrict after hours access or who have implemented a "follow the sun" support approach and wish to authorize specific users to have administrative access only during specified hours.

PowerBroker has the ability to delegate resource accounts to end-users without the need to share a password among them. Sharing passwords among users is a violation of a number of core security principles but has become commonplace in many organizations. Access and revocation to the resource account is easier to administer than the management of a shared password.

PowerBroker can do more than just log, it can send an email alert, restrict subsequent access from that host, start screen and keystroke logging, generate syslog messages, snmp traps, etc. When a PowerBroker command has been rejected by the pbmaster, it is possible to configure the policy file to execute a script file.

PowerBroker includes a script language that can be used to write interactive menus. This feature can be used to provide a menu-driven "front end" for an application that includes options that a user should not have access to. This should be used with caution because an incorrectly written script could allow a user to gain access as a high-level user.

One area of emerging value may involve the ability to delegate specific tasks to employees who reside in countries which the US government may deem to be 'Controlled" with regard to export laws. Currently US regulations restrict employees of controlled countries to only possess administrative access to systems in their country of national affiliation. PowerBroker's ability to restrict access to an interactive login may allow a company to empower employees to perform defined tasks on systems where they never could before.

Master hosts can be configured to verify clients before allowing them to authenticate. This feature will strengthen the security of a PowerBroker installation by making it more difficult for a hostile client to attempt to gain access to the network.

One feature provided by Sudo natively is the use of timestamp files to implement a "ticketing" system. These tickets allow successfully (sudo) authorized users to issue additional authorized sudo commands within a configurable timeframe without requiring that they re-enter their password. Each subsequent sudo command updates the ticket for another 5 minutes. This allows "heavy users" of sudo to work interactively without the productivity hit taken by entering a password for each command they wish to execute. This also helps to reduce risk associated leaving a root shell where others can physically get to your keyboard or potentially hijack an idle session. Ticket removal can also be forced at any time or configured to be removed, such as in a user's .logout file.

The following policy code will provide the same functionality as the sudo timestamp files.

```
timestamp = "/tmp/" + runuser + ".timestamp";
pass = getuserpasswd (runuser, "Your Password: ", 1, timestamp, 300);
if (pass != 0)
{
  system("/bin/touch" + " " + timestamp);
  runcommand = basename(command);
  accept;
}
```

Feature summary table:

Feature ("out of box")	Sudo	PowerBroker
Central administration	No	Yes
GUI administration	No	Yes
Hardened utilities	No[4]	Yes
Hardened shells	No[4]	Yes
Client-server architecture	No	Yes
Verification of clients	No	Yes
Keystroke Logging	No[4]	Yes
Ability to run root commands without an interactive login shell	Yes	Yes
Checksum validation of program to be executed	No	Yes
Ability to log to syslog	Yes	Yes
Ability to encrypt log data	No	Yes
Ability to log to designated file	Yes	Yes
Technical Support	No	Yes
Authorization "Tickets"	Yes	Yes
Tiered Configuration Files	No	Yes
Log server designation for program files only	No	Yes
Protected authorization channels	Yes	Yes

Supported Platforms

Both Sudo[5] and PowerBroker[6] support a wide range of operating systems. Current versions of Compaq Tru64 UNIX, Debian, Digital UNIX, generic Linux, HP-UX, IBM RS/6000, Red Hat Enterprise, SCO OpenServer, SCO Unixware, SUN, and SuSE as well as quite a few other operating systems are supported by both products. There is also support for a wide range of legacy systems.

[4] This feature is available using addon packages or custom scripting. To some degree most options could be made available with extensive scripting.

[5] http://www.sudo.ws/sudo/runson.html

[6] ftp://ftp.symark.com/unix/pr/pb_datasheet.pdf

PowerBroker Overview

PowerBroker is a suite of privilege management programs that run on UNIX/Linux computers. The core functionality is broken into 4 processes that can be run on separate machines or all on a single computer. These processes are the *Master Host*, the *Log Host*, the *Submit Host*, and the *Run Host*. The *Submit Host* and the *Run Host* are often the same computer while the Master Host and the Log Host are normally running on separate computers to increase security. Separation of duties is an important security control and in this case, it protects the logging and control from the computer that is being used to execute the commands.

Master Host - This is the computer where the control policy files are located. This is normally a computer dedicated to approving the access to run requested programs. In the most secure configuration, this would be a hardened computer with tightly controlled access, file integrity checks performed against the PowerBroker policy files, and a minimal list of other services running on the system.

Log Host - This is one or more computers that log the commands, keystrokes, and attempts (successful and failed) to execute applications through PowerBroker. In the strictest implementations of PowerBroker these logs should be periodically written to a write-once media such as a CD or DVD to ensure that a tamper-proof copy of the logs exist.

Submit Host - this is the computer where the request is submitted. This is normally the same computer as the *Run Host*.

Run Host - this is the computer where the application is actually running. In most cases, this computer is the same as the *Submit Host* but in some cases, it could be a separate computer. Separating the *Submit Host* from the *Run Host* is useful when a system administrator needs to design a program in which people may submit "jobs" to be run on a system with limited access. PowerBroker can be designed to allow an end-user to share the resources on a system without actually having to connect to the system through an interactive login. In many cases, the end user may not even realize where the command is actually running. Since many vulnerabilities require local access to exploit them, restricting access in this manner further reduces such risks.

Sudo Overview

Sudo is a utility that has been used by UNIX and Linux administrators for over 20 years[7]. It was initially developed in the educational arena and has been used widely to give administrators a way to delegate responsibility. The main use of sudo is to give administrators a way to allow normal users to access programs that must run with root access without giving the user full root access. Sudo can also be configured to grant non-root access in recent versions. This is sometimes used for mail or web servers to allow an administrator to manipulate a server without having to log in as the server user.

Sudo is designed to utilize several well know secure authentication procedures. At compile time you can choose to use any of the following authentication strategies:

- One Time Passwords is supported for both s/key and opie.
- SecurID
- Kerberos IV and V.
- PAM (Red Hat 5.x+, Solaris 2.6+, HPUX 11.0+)
- AFS (Andrew File System)
- Shadow passwords

One of the appealing features of sudo is the simplicity of establishing a rudimentary configuration file. Extended Backus-Naur Form (EBNF) is used to define the grammars used in the sudoers file. This is a simple to understand format that lends itself to easy adoption. The basic format for all sudo rules adheres to the following pattern:

```
User_Alias  Host_Alias = (Runas_Alias) Cmnd_Alias
```

The *User_Alias* provides the ability to define a user by their username, their uid, or their inclusion in a UNIX group or a netgroup. The *Host_Alias* can be defined by ip address, system name, netgroups, or CIDR notation. By modifying the *Host_Alias* entries, we could entitle or revoke access to a sudo account without having to impact all of the remaining users (such as you would if you suddenly needed to change the password for an account). The *Runas_Alias* option expands the ability to use sudo for shared accounts without the need to actually share a password. The *Cmnd_Alias* expanded the ability to empower end-users on a granular level by allowing the establishment of specified commands which they are permitted to run.

Summary

PowerBroker is a full-featured solution with a rich suite of security options. Because of these features, PowerBroker can be much more complicated to set up if all the security options are used. In a strict comparison with sudo however; using only the features available to sudo, the installation and maintenance is no more complicated.

Sudo can be an effective solution for organizations where the primary need is to restrict access for cooperative users and to avoid errors. If there is a need for a system that will provide an audit trail and solidly enforce security policy such as is required by HIPAA, SOX, etc. to prove regulatory compliance, PowerBroker has the built-in tools to handle that task.

PowerBroker is issued on a ***per node*** *license which can become costly for a large scale corporation. The cost is one reason some companies may not uniformly deploy PowerBroker and may instead use it as a point solution for mission critical or sensitive data systems.*