Paper SAS1761-2015

Proven Practices for Managing the Enterprise Administrators of a SAS Software Deployment

Clifford Meyers and Rob Collum, SAS Institute Inc.

ABSTRACT

There can be a need to provide multiple administrators with the ability to manage your software. The rationale will span from separation of roles and responsibilities (for example, installer and configuration manager) to changing job responsibilities or even just providing coverage while the primary administrator is on vacation. To meet that need, it's tempting to share the logon credentials of your SAS installer account, but doing so can potentially compromise your security and fail a corporate audit.

The SAS Installer user ID on a UNIX (or Linux) system is typically named something generic, such as "sas" or "sasinst." That account, like others associated with SAS and other software services, is often not tied to any single person in the enterprise. The role of SAS Administrator could be a job function of one or several people – over time or at any point in time – as professionals follow their careers.

This paper focuses on a basic IT practice and utility to help you diligently manage the administration of your SAS software, while properly ensuring that access is secured and auditability is maintained.

INTRODUCTION

Privilege separation and control are fundamental security paradigms implemented in UNIX and Linux operating systems. Regular user IDs operate with limited privileges in order to limit the scope of their influence to their own environment and not to the wider operating system and user base. Third-party administrative user IDs have similar privileges for their associated software. However, changes made to the third-party software by these administrative user IDs can have a major impact on a broad user base and must be carefully managed.

At times, it might be necessary for administrators with regular user ID privileges to act on behalf of a third party administrative account. The need might be based on well-defined, separate roles and responsibilities, or simply to provide coverage while the owner of the primary administrative account is out of the office. These needs can be met either by sharing the administrative account user ID and password or by creating tightly defined, controlled, and auditable roles.

The practice of sharing an administrative user ID and password among two or more individuals is unacceptable. Once more than one person knows the administrative password, plausible deniability becomes a distinct possibility. If multiple people are able to simultaneously log on to the system with the same user ID, then it is challenging to determine which person performed exactly which tasks. When attempting to troubleshoot an issue, each actor might point the finger at the others, and no insight as to why the change was made can be provided.

In this article, we will discuss how multiple regular user IDs can correctly and securely obtain specific administrative privileges of the SAS Installer user in a password-less fashion. In addition, we will discuss how actions that are run on behalf of the SAS Installer user ID by a regular user ID can be monitored.

OPTIONS

There are three fundamental options for granting multiple regular users access to a single third-party administrative user ID:

- direct login to the administrative user ID
- su command to the administrative user ID from a regular user ID
- sudo command to the administrative user ID from a regular user ID

Let's explore those options below and see how well they accomplish the goals of secure access, control, and auditability.

DIRECT LOGIN

The direct login option provides access to the administrative account is provided through server login prompts such as a console login, telnet, or SSH. This means that any person who wants to administer the site with that account must know the account's user ID and password and enter it directly. Therefore, the account's credentials must be shared among those individuals.

This technique has the following implications:

- **The security of the account is marginal, at best.** The credentials are not private to a single person. Regular password changes must be communicated among the members of the administrative team. The potential for inadvertent leaks of this confidential information is high.
- **Privileges are uncontrolled.** The administrative account has full privileges to modify the areas in its domain. Anyone who knows the user ID and password has the ability to make changes without restriction. The larger the administration team, the less likely it is that all personnel have the same skill set and expertise for managing the environment. A well-intentioned administrator could use the unrestricted privileges to make unfortunate changes to the environment.
- **Auditing of administration activity is difficult.** If multiple people log on to the administrative account directly and at the same time, then it can be very challenging and time consuming to determine which person performed which specific actions.

For these reasons, relying only on direct login to the administrative account presents an unacceptable risk to the enterprise. We can do better.

SU COMMAND

The su command – su is short for "substitute user" – is a command-line utility that enables the current user to log on directly as a different user on the same system. In this scenario, a user would first log on to the system with his or her own personal credentials. When the user needs to make an administrative change, he or she would "su" over to the administrative account to run a single command or enter an interactive shell. The su utility requires the user to provide the full set of logon credentials for the administrative account to gain access.

The implications of this technique are the same as for direct login option described previously:

- The administrative account's credentials must be shared. Therefore, security is marginal..
- Any user of the administrative account has ability to make unrestricted changes to the environment because the privileges are uncontrolled..
- It will be very challenging to trace individual actions of multiple concurrent users of the administrative account. Therefore, auditing of administrative activity is still difficult.

For these reasons, relying on the su command to gain administrative access to the system also presents an unacceptable risk to the enterprise. We can do better.

SUDO COMMAND

The sudo command – "sudo" is short for "substitute user do" – is a richly featured utility that provides granular controls for fine-tuning the activities of users. The sudo utility is often used to provide root-level (also known as super user) access to normal user accounts. However, it can be configured to provide access to any account, not just root.

Unlike the su command, the sudo utility can be configured so that regular users must authenticate with their own credentials. Definitive confirmation of identities serves as a precaution against users leaving their terminals unattended and a bad actor surreptitiously acting on their behalf. Furthermore, the sudo utility can be configured to only permit the execution of specific commands on a per-user basis.

This technique has the following implications:

- **Security of the account is good.** The credentials of the administrative account do not need to be shared at all. Users authenticate with their personal credentials instead.
- **Privileges are controlled.** While the administrative account has full privileges to modify the areas in its domain, sudo can be configured to only allow a subset of those abilities to any individual user.
- **Auditing of administrative activity is good**. The sudo utility keeps a detailed audit log of activity, allowing a straightforward investigation into which users performed which operations.

For these reasons, setting up the sudo command for the administrative account is the recommended approach for maintaining control of the account's usage by multiple users.

WHY SUDO

- The password for the administrative account does not need to be shared with every individual who needs to perform administrative tasks.
- Each user can be restricted to execute only a specified list of commands.
- Authorization can be managed in a central location.
- A detailed audit trail is created.
- The "I can do anything" tendency on the part of regular users is avoided.
- Administrative rights can be transferred easily by adding and removing user IDs from groups, while not compromising the administrative account.
- The sudo command supports setting up a fine-grained security policy.
- Authentication with sudo automatically expires after a specified period of time.

MAIN COMPONENTS OF SUDO (REDHAT BASED)

The sudo command has the following three main components:

- the /etc/sudoers configuration file
- the visudo file editor to safeguard proper syntax
- the audit files updated by sudo

SUDOERS CONFIGURATION FILE

The sudoers file (located in /etc) controls which regular user IDs can run what commands as a defined administrative user ID on specific machines. It can also control special configurations such as whether a password is needed for particular commands. The file contains alias types (variables) and user specifications (that control who can run what). Here is an example of a sudoers file:

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particular
## users or groups.
##
## This file must be edited with the 'visudo' command.

## User aliases
## Runas_Alias
## Host aliases
## Command aliases
## User specifications
```

ALIAS TYPES

Four alias types are available: User, Runas, Host, and Cmnd (Command).

- User (answers the question “Who?”)
  - User aliases specify groups of users. You can specify user names, system groups (prefixed by a %), and netgroups (prefixed by a +).
- Runas (answers the question “As whom?”)
  - Runas aliases specify the user ID that other users can run commands as.
- Host (answers the question “Where?”)
  - Host aliases specify the host names, IP addresses, networks, and net groups that sudo can run on.
- Cmnd (answers the question “What?”)
  - Command aliases specify groups of commands and directories that sudo can access.

ALIAS FORMATTING

- Format:
  - [ALIAS TYPE] [ALIAS NAME] = [VALUE_1, VALUE_2, VALUE_N]
- Where:
  - ALIAS TYPE User_Alias, Runas_Alias, Host_Alias, Cmnd_Alias
  - ALIAS NAME user-defined name for the alias
  - VALUE(S) values associated with the alias name

For example:

- To define a User alias called “ORIGMEMBERS” that contains the users jerry, bob, pigpen, phil, and bill, you would add the following entry to the sudoers file:
  - User_Alias ORIGMEMBERS = jerry, bob, pigpen, phil, bill
- To define a Host alias called “VENUE” that contains the host name redrocks.denver-colorado.com, you would add the following entry:
  - Host_Alias VENUE = redrocks.denver-colorado.com
- To define a Runas alias called “THEBAND” that contains the third-party administrative user ID “theodead,” you would add the following entry to the sudoers file:
  - Runas_Alias THEBAND = theodead
- To define a Command alias called “PLAY” that contains the command /usr/local/bin/darkstar, you would add the following entry to the sudoers file:
  - Cmnd_Alias PLAY = /usr/local/bin/darkstar

After you add these entries, your sudoers file would look similar to the following:

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particular
## users or groups.
##
## This file must be edited with the 'visudo' command.
```

```
## User aliases
User_Alias      ORIGMEMBERS       = jerry, bob, pigpen, phil, bill

## Runas_Alias
Runas_Alias     THEBAND           = theodead

## Host aliases
Host_Alias      VENUE             = redrocks.denver-colorado.com

## Command aliases
Cmnd_Alias      PLAY        = /usr/local/bin/darkstar

## User specifications
```

USER SPECIFICATIONS

The user specifications section of the sudoers file determines who can run what as whom. All of the other aliases have been set up specifically for use in this key section of the file.

- Format:
  - <User Alias> <Host Alias> = <(Runas Alias)> <Cmnd alias>

For example, to allow the User alias “ORIGMEMBERS” to run the Cmnd alias “PLAY” on Host alias “VENUE” as Runas alias “THEBAND,” you would add the following entry to the sudoers file:

- ORIGMEMBERS VENUE = (THEBAND) PLAY

After you add the user specification section, your sudoers file would look similar to the following:

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particular
## users or groups.
##
## This file must be edited with the 'visudo' command.

## User aliases
User_Alias      ORIGMEMBERS       = jerry, bob, pigpen, phil, bill

## Runas_Alias
Runas_Alias     THEBAND           = theodead

## Host aliases
Host_Alias      VENUE             = redrocks.denver-colorado.com, redrocks

## Command aliases
Cmnd_Alias      PLAY        = /usr/local/bin/darkstar

## User specifications
ORIGMEMBERS     VENUE             = (THEBAND) PLAY
```

Now, the original members of the Grateful Dead can play the song “Dark Star” at Red Rocks Amphitheater!

SUDOERS CONFIGURATION FILE EDITOR

The sudo command is configured through the sudoers control file, which is owned by the root user. All sudo related activity is logged through a secure log file, which is also owned by the root user.

Never edit the sudoers control file with a normal text editor. Always use the visudo utility instead. The visudo utility validates the syntax of the sudoers control file upon saving. This validation is necessary because improper syntax in the sudoers control file can result in a system in which it is impossible to obtain elevated privileges.

SUDOERS AUDIT FILE

The sudo command provides a comprehensive audit trail. Each successful authentication is logged to the file /var/log/messages. Each issued command, along with the issuer's user name, is logged to the file /var/log/secure.

Suppose the following sudo command is executed by user ID “pigpen” on the server redrocks:

```
sudo -u theodead /usr/local/bin/darkstar
```

After pigpen is prompted for a password, pigpen will see the details of the command as if it had been executed by the userid theodead. From an audit perspective, the following text is stored in the audit file /var/log/secure:

```
Jan 19 12:19:24 redrocks sudo: pigpen: TTY=pts/0; PWD=/home/pigpen;
USER=theodead; COMMAND=/usr/local/bin/darkstar
```

Suppose the same command is attempted by “cdion,” which is a non-member general user ID. After cdion is prompted for a password, cdion will see the following message:

```
Sorry, user cdion is not allowed to execute '/usr/local/bin/darkstar ' as
theodead on redrocks.denver-colorado.com.
```

The following text is stored in the audit file:

```
Jan 19 12:22:18 redrocks sudo: cdion: command not allowed; TTY=pts/0;
PWD=/home/cdion; USER=theodead; COMMAND=/usr/local/bin/darkstar
```

DEMONSTRATION

In the following practical example, we want the regular user ID “clmeye” to be able to execute both the SAS sas.servers and UNIX env commands as the SAS administrative installer user ID “sas” on deadhead.na.sas.com.

Note that the SAS administrative user ID should still be password managed and directly accessible by one well-defined individual.

Here are the high-level planning details:

- SAS administrative installer
  - sas
- Regular user ID:
  - clmeye
- Host
  - deadhead.unx.sas.com
  - deadhead
- Commands
  - /saswork/Analytics/VA7.1/config/Lev1/sas.servers
  - /bin/env

Refinement:

- User_Alias
  - SASSTART_GROUP = clmeye
- Runas_Alias
  - SASINSTALLER = sas
- Host_Alias
  - SASSERVERS = deadhead.unx.sas.com, deadhead
- Cmnd_Alias
  - SASSTART_SERVICES = /saswork/Analytics/VA7.1/config/Lev1/sas.servers, /bin/env
- User specification:
  - SASSTART_GROUP SASSERVERS = (SASINSTALLER) SASSTART_SERVICES

The sudoers file would look similar to the following after you edit it with the visudo command:

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particular
## users or groups.
##
## This file must be edited with the 'visudo' command.

## User aliases
User_Alias      SASSTART_GROUP    = clmeye

## Runas_Alias
Runas_Alias     SASINSTALLER      = sas

## Host aliases
Host_Alias      SASSERVERS  = deadhead.unx.sas.com, deadhead

## Command aliases
Cmnd_Alias      SASSTART_SERVICES =
/saswork/Analytics/VA7.1/config/Lev1/sas.servers, /bin/env
```

```
## User specifications
SASSTART_GROUP = SASSERVERS = (SASINSTALLER) SASSTART_SERVICES
```

Suppose clmeye goes on a well-deserved vacation, and we want to temporarily grant the same SASSTART_SERVICES administrative privileges to regular user ID “rocoll.” No password changes are required! The only requirements are a Change Record indicating the modification of the User alias and an update to the sudoers file.

MISCELLANEOUS

This paper has provided a mostly introductory look at sudo. Before using this tool, you should have a good understanding of how it works and how powerful it is. For further reading about sudo, issue the following command, which opens the manual page for the sudo command:

```
man sudo
```

To access the manual page for the sudoers configuration file, issue the following command:

```
man sudoers
```

Numerous third-party options are available to help manage access to a third-party administrative user ID. Please review your needs and options with your security team and UNIX administrators. Keep in mind that the bottom-line objectives are to protect access to your SAS installation, manage activities efficiently, and be able to audit actions.

THE BIG PICTURE

Administration of your SAS deployment is not solely in the domain of the operating system. The approach that is described here is less applicable to many other common situations that administrative staff must contend with, including:

- SAS metadata
- Integration and management of SAS and other third-party data sources such as the following:
  - SAS LASR Analytic Server tables
  - SAS data sets
  - RDBMS
  - Hadoop
- Grid computing deployment and maintenance
- Third-party authentication and authorization providers
- Application-level user profiles

In SAS deployments, we see the sudo command used most frequently in support of the SAS services software running at the OS level, performing tasks such as the following:

- Starting and stopping the SAS services
- Applying software updates and hotfixes
- Investigation and troubleshooting
- Configuration and other enhancements

With care and planning, the sudo command will be a powerful addition to your SAS administration toolset.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Clifford Meyers
Manager, Mainframe and Unix Interface
Technical Support Engineering
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
(919) 531-5321
Clifford.Meyers@sas.com
www.sas.com

Rob Collum
Sr. Technical Architect, Global Architecture & Technology Enablement
Professional Services Division
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
(919) 531-0295
Rob.Collum@sas.com
www.sas.com