WHITE PAPER

# Understanding Symantec® Privileged Access Manager Server Control

Symantec by Broadcom

## TABLE OF CONTENTS



## Overview

Symantec® Privileged Access Manager Server Control (PAM SC) improves security enforcement and administration of critical servers. PAM SC is a host-based access control solution that provides defense in depth, as an additional layer of security on top of the native operating system.

PAM SC improves the level of control that system administrators have to protect data, applications, and the operating system. This security is even enforceable against unauthorized use of the superuser accounts.

The PAM SC technology is based on a security extension to the operating system kernel. This patent-pending architecture called soft-hooks allows for real-time startup and shutdown of the security by authorized administrators only. Importantly, this is done without requiring a kernel rebuild and without replacing any operating system binaries.

A critical and frequently overlooked capability of security products is self protection. Because PAM SC protects itself, it can be trusted to reliably protect any resource, even from a user or program with root access. With unauthorized superuser access controlled, several other benefits immediately result:

- Superuser delegation or partitioning can be enforced
- System monitoring logs and audit trails will have integrity
- Vulnerability assessments to find the latest hole will be less critical
- Continuous maintenance of system patches is less essential
- Network protections can be implemented

With PAM SC, authorized access to superuser is still allowed, which enables implementing stronger security without impacting essential system administration. Only authorized system administrators are able to make system changes, and the newly secured log and audit reports enable determining who was the actual source of changes.

PAM SC IS A PROACTIVE SECURITY SOLUTION, RATHER THAN THE COMMON REACTIVE APPROACH TAKEN BY MOST UTILITIES AND PRODUCTS

PAM SC is a proactive security solution, rather than the common reactive approach taken by most utilities and products. There are many useful shareware and commercial security tools available including sudo, tripwire, tiger/cops, and TCP wrappers. Unlike vulnerability assessments, security hole patching, security monitoring, auditing, or unenforceable superuser delegation utilities, PAM SC allows you to get ahead of system attackers, stay ahead, and significantly raise the bar on Unix/Linux security.

## Server Security Background

In distributed computing environments, critical information such as financial transactions and trade secrets resides on servers. Controlling access to this data is a key business requirement.

Unfortunately, open-system servers do not offer adequate levels of data security. The main security issues with decentralized servers are unauthorized access and holes in the underlying operating systems. These threats are due to the power of the superuser in Unix/Linux and MS Windows, which grant a vulnerable user account full access to all applications, data, and audit logs. Attempting to manually enforce security in this environment is extremely time intensive and ineffective.

PAM SC solves these problems by strengthening security and reliably controlling access to critical information. With PAM SC, servers automatically protect themselves, avoiding the need for manual human intervention. PAM SC supports most leading client/server systems so security policies can be centrally managed in mixed computing environments.

## How PAM SC Works

### PAM SC Architecture

PAM SC ALLOWS YOU TO GET AHEAD OF SYSTEM ATTACKERS, STAY AHEAD, AND SIGNIFICANTLY RAISE THE BAR ON UNIX/LINUX SECURITY

Soft-hooks is the patent-pending technology that is at the heart of PAM SC. The design goals for soft-hooks provide insights to the architecture of the technology:

- Minimize impact to administration procedures and performance
- Maximize security, granularity, and flexibility.

PAM SC is loosely inspired by the mainframe world as it applies many of the same proven concepts to distributed environments.

A true server security solution needs to be implemented at the level of the operating system. Solutions that are not implemented in this way can be bypassed for authorization checking and auditing. The unique design of PAM SC allows it to become a virtual part of the operating system, without changing a single binary file or rebuilding the operating system kernel. PAM SC accomplishes this strong degree of security without rewriting the operating system and without making system maintenance an impossible task.

The operating system has a system call (or vector) table that contains memory address pointers for each system call. The pointers point to a location in memory where the actual kernel code of the system calls resides. PAM SC stores the address pointers for the security-sensitive system calls and then redirects the pointers to the corresponding PAM SC system call code, which is located elsewhere in memory.

**THE UNIQUE DESIGN OF PAM SC ALLOWS IT TO BECOME A VIRTUAL PART OF THE OPERATING SYSTEM, WITHOUT CHANGING A SINGLE BINARY FILE OR REBUILDING THE OPERATING SYSTEM KERNEL**

When a user process issues as security-related system call, the PAM SC system call code is called and performs a lookup in the PAM SC database. If the action is authorized, PAM SC uses the original address pointer to pass the call to the original operating system call code; processing continues, as it would have without PAM SC installed. If the action is not authorized, PAM SC returns an error code to the user process.

This implementation gives PAM SC a tremendous advantage over security products that are implemented outside of the operating system. Products that replace some system utilities (such as `/bin/su`) can still be bypassed and therefore lead to a false sense of security. They also complicate system administration and maintenance.

PAM SC is also unlike other security products that permanently modify the operating system. These approaches complicate system administration, and can actually void the warranty with the operating system vendor, making technical support difficult or impossible.

With soft-hooks, PAM SC does not replace or modify any system binary files. Current customers run PAM SC across a wide range of platforms and get the same level of vendor support with or without PAM SC running. PAM SC is both backward and forward compatible, so rules and commands that worked for older versions also work with new versions. New features are not supported by older versions, but interactions and administration continue to function.

### Resource Classes

PAM SC categorizes server resources into classes for controlling access. For example, there is a FILE class for defining access control for files. There is a TCP class for defining access controls related to inbound and outbound TCP services.

Each PAM SC class has objects that represent resources of a specific type, with their own appropriate settings. For example, in the FILE class, you might define a resource called `/etc/passwd`. Each class can have default settings, which include specifying owner, access, audit, and warning mode. In addition, you can authorize access for specific users or groups outside of the default specified for the resource.

You can also authorize access based on the program. For example, you can specify that accounting data can only be accessed through the accounting application. The access can be exclusive or inclusive, in other words, access can be denied except for specified users, or access can be allowed except for specified users.

### System Integrity

Ensuring the integrity of an application and its data is a key requirement. PAM SC has a number of features designed to maintain integrity. First, the PAM SC daemons are self-checking. At startup, a database integrity check is automatically performed. All of the PAM SC files (except the boot-time startup scripts) reside in their own directory structure.

In the event of a system shutdown, the PAM SC database is automatically shut down. An internal locking mechanism provides for exclusive database use by the PAM SC database tools. Audit logs are also locked by PAM SC so that all users, including superusers, cannot tamper with the audit trails.

**PAM SC PROVIDES COMPLETE FILE PROTECTION BY INTERCEPTING EVERY FILE ACCESS REQUEST AND DECIDING IF THE USER IS AUTHORIZED TO ACCESS THE FILE IN THE REQUESTED MANNER, ACCORDING TO ITS ACLs**

## PAM SC Controls

PAM SC provides granular access control over a variety of server resources

The main controls are detailed below.

### File Access Controls

Native operating systems do not provide file security against attackers with superuser (root, administrator) access; they also don't provide a method to secure files with a generic, wildcard definition to determine and maintain access to groups of files. PAM SC provides complete file protection by intercepting every file access request and deciding if the user is authorized to access the file in the requested manner, according to its Access Control Lists (ACLs).

With PAM SC, the superuser account is treated, and can be restricted, just like any other user. PAM SC file access rules are more flexible than those provided by native operating systems, and specific rules can be established for multiple users and groups, each with their own access modes. With this feature, customers can limit access to any files, including configuration files, log and audit files, databases, and personnel files. With the generic file rule feature of PAM SC, you can specify one rule to apply to many subdirectory trees on your system, or have it match a specific filename no matter where it is found on the machine.

**WITH PAM SC, THE SUPERUSER ACCOUNT IS TREATED, AND CAN BE RESTRICTED, JUST LIKE ANY OTHER USER**

PAM SC can protect the contents of entire directories, for example, `/etc/*` or `$DIR/webserver/ht-docs/*`. PAM SC rules can also protect files such as `$HOME/*/.rhosts` to protect all users' `.rhosts` files. You can also set a rule to protect a set of files with the same name: `/app/config*`. This would protect `/app/config.dat` and `/app/config.tar`. If the need arises to protect files that don't map into a directory or common naming convention, you can use the GFILE class. This class allows you to define specific files and apply a common set of access control rules to all of the files.

Additionally, with Program Access Control Lists (PACLs), you can ensure that sensitive resources are only being accessed using approved programs, for example, personnel database files are only being written to using the database application.

Sometimes, you need to restrict how someone accesses a particular file—you may want them to be able to view a file through a specialized reader, but not through the Unix/Linux vi, cat, or other commands. With PACLs, or program pathing, you can define a file so that by default it cannot be read. You would then authorize read access to the file through the program `/usr/local/bin/reader` (reader is the authorized program in this example).

## Superuser Containment

### Substitution Protection

**PAM SC INTERCEPTS AND VERIFIES EVERY REQUEST TO CHANGE USER IDENTITY AND MAINTAINS A RELIABLE AUDIT TRAIL**

Accountability is an important part of security, and a reliable audit trail of events identifies which users are responsible for performing certain actions. Native Unix/Linux can be fooled when the user ID (`uid`) or group ID (`gid`) is changed using the `su` command, and the audit trail for further actions will not reflect the original user ID.

PAM SC intercepts and verifies every request to change user identity and maintains a reliable audit trail. Unlike other solutions that work by replacing the `/bin/su` program binary, PAM SC maintains control over this event regardless of the program used to change the `uid`. In this way, PAM SC provides assurance that the original login ID is never lost, and that it is present on every log record generated for the user. Furthermore, the user's permissions are always governed by the original login ID. Even taking over the root account does not grant the user any additional privileges.

There are a number of reasons why many people, in addition to the system administrators, may have root access. For example, database administrators may know the root password, or the help desk may know the superuser password in order to help users who forget their own. There are several accounts including root that are often generic shared IDs, to which many people in the organization may require access. In native Unix/Linux, there is no way of ensuring that this access is not misused. Consequently, limiting who may assume this authority, and what they are capable of doing while using it, is a major concern.

With PAM SC you can impose control over root ID access by forcing users to log in with their private user IDs rather than using root. In native Unix/Linux, organizations may disable root logins except from a physically secured console. Users must log in over the network and then access the root account. However, once on the system, they can gain access to the root account in many different ways. PAM SC allows only authorized individuals to `su` to root after they login. Unauthorized users cannot `su` to root even if they know the password. Furthermore, PAM SC can log every operation root executes, and keep track of the original identity of the operator.

### Delegating Authority

**PAM SC ENABLES ADMINISTRATORS TO SHARE SUBSETS OF ROOT AUTHORITY AMONG DIFFERENT ADMINISTRATORS BASED ON FUNCTIONAL ROLES**

System administrators often share the root ID in order to perform a variety of privileged functions. This is problematic as the privileges can be used with no mechanism for accountability. Native Unix/Linux cannot provide a reliable audit trail of root activity, and the audit trail cannot connect the activity to each individual person.

PAM SC enables administrators to share subsets of root authority among different administrators based on functional roles. The organization can then eliminate the concept of one all-powerful superuser. If this doesn't make sense because of existing organizational procedures, some administrators can continue to have full root access while others have a subset. For example, some organizations make a distinction between administrators and operators. PAM SC can facilitate a limited set of root access for operators, and then PAM SC can monitor the activity of administrators that have full root access.

PAM SC uses the SUDO class facility to delegate authority to users for specific actions to be run as root or other users (such as Oracle, Sybase, or web). The SUDO class allows an organization to define the Unix/Linux process, under which account it runs, which users are authorized to start the process, and what parameters are allowed to be given to this process. This feature can be used, for example, to allow operators to initiate a backup, mount a file system, or perform a database export.

### Application Jailing with Logical Users

There are OS daemons that run under superuser authority, and it is frequently necessary to limit the access of such daemons to system resources. PAM SC allows a logical user to be assigned to each such daemon. PAM SC assigns all access rights to this daemon based on logical user access rights, and not the OS user id (common root/administrator). For example, if the `sendmail` daemon runs under root, you can define a user called `sendmail` and specify it as the logical user for the `sendmail` daemon. You can then specify the `sendmail` user in ACLs to control the access of the `sendmail` daemon to resources.

**PAM SC PROVIDES THE ABILITY TO CREATE JAILS FOR APPLICATIONS, WHICH HELPS PREVENT ZERO-DAY ATTACKS ON CRITICAL APPLICATIONS**

This feature is implemented with the SPECIALPGM class. Each program and daemon specified in the SPECIALPGM class is associated with a logical user, as specified in the command attributes.

PAM SC provides the ability to create jails for applications. The PAM SC application jails create logical users, and then use the SPECIALPGM resource class to assign authorization controls that can be applied for a specific application.

This functionality helps prevent zero-day attacks on critical applications.

## Program Controls

### Back Door and Trojan Horse Protection

PAM SC provides solutions to the problems of back doors and Trojan horses. Some of the most popular mechanisms for attacking Unix/Linux systems are based on back doors and Trojan horses:

- Back doors can be created intentionally by malicious developers or unintentionally through poor programming and weaknesses in the operating system and programming languages. Through these back doors, attackers can trick the operating system into giving them greater access.
- Trojan horses are programs that perform one function when they are supposed to perform another, for example, a program that an administrator normally runs that secretly gives an attacker access.

Through the PAM SC PROGRAM and SECFILE classes, administrators can prevent these attacks from taking place or control the damage that can be done based on these attacks.

### Privileged Programs

Through the use of privileged (`suid/sgid`) programs, non-root users can temporarily assume root authority. If the PROGRAM class is active, PAM SC only allows execution of `suid/sgid` programs or scripts that are defined. PAM SC denies execution of all other programs and scripts, as well as any defined programs or scripts that have been modified. PAM SC constantly verifies that registered trusted programs are unchanged. If a trusted program is modified or replaced, PAM SC marks it as untrusted and prohibits its execution.

**PAM SC ONLY ALLOWS EXECUTION OF SUID/SGID PROGRAMS OR SCRIPTS THAT ARE DEFINED**

In addition to creating these authorized programs and scripts, an administrator can define how these programs can be executed. This program pathing functionality works in the same way as it does for the file controls.

### Monitored Files

Through the PAM SC SECFILE class, administrators can define critical files that are not supposed to change very often. If these files are modified, the process that checks the privileged programs will find that the files have changed and write an audit record, which can be routed in many ways.

This functionality is similar to that provided by the freeware tripwire program, and provides the extra advantage that the audit information and the baseline information can be protected even from an attacker with root access.

## Process Controls

In Unix/Linux, web servers, database servers, and all other services operate as daemons in the system and are constantly active processes, answering calls and requests from clients. A primary security objective is to maintain the availability of these services. Maintaining process availability can become a challenge when unauthorized individuals seek to impede operations. Also, any user running as root may shut down a server (using the `kill` command) whether accidentally or maliciously.

Using the PAM SC PROCESS class, an administrator can control the circumstances under which authorized users may terminate sensitive processes, including time, day, and where from.

## Network Controls

### Controlling Network Access

**PAM SC PROVIDES ACCESS CONTROLS FOR BOTH INBOUND AND OUTBOUND NETWORK COMMS, WHICH GIVES EVERY PAM SC SECURED SERVER ITS OWN FIREWALL FUNCTIONALITY**

Computing is about connectivity and communications. The challenge is to allow network access to authorized users while denying access to unauthorized users. PAM SC provides access controls for both inbound and outbound network communications. This gives every PAM SC secured server its own firewall functionality. PAM SC provides configuration of network security through the TERMINAL, HOST, CONNECT, and TCP classes.

The TERMINAL class controls access to a machine based on source host (terminal) and account name.

The TCP class controls inbound and outbound connections based on service and host. For outbound connections, it also provides controls based on the program used to try to establish the connection. The TCP class organizes the access control rules by service. The controls can also be enforced for specified time of day and day of week.

For example, you have a machine called savings1 that is running PAM SC, and you want to allow web traffic so employees can view their 401(k) information. To accomplish this, set up a TCP resource called TCP1 with default read access for inbound traffic. Then deny all other TCP access by setting default access to the TCP resource class `_default` to none. For administration, you can selectively authorize other TCP services

**UNLIKE TCP WRAPPERS, PAM SC PROTECTION CAN BE APPLIED TO ALL NETWORK PORTS REGARDLESS OF WHAT PROGRAM IS BEING USED TO ACCESS THE NETWORK**

Unlike TCP wrappers, PAM SC protection can be applied to all network ports regardless of what program is being used to access the network. PAM SC can also be configured to lock specific users into a machine and prevent them from jumping off the system to other machines. This is often useful in situations where vendors are providing remote support for an application. They legitimately need access to that one machine, but they have no reason to go to any other machine on your network.

### Auditing Controls

Good audit trails are an important and necessary component of any security solution. Administrators need to track changes to system configuration, and incident response depends on reliable audit data. If the need for criminal action arises, audit data with integrity is far more powerful evidence in court. Audit data must be complete and accurate.

With native Unix/Linux, you can never be sure that the information you see in log files is either accurate or complete. Anyone who obtains root access can delete or modify any of the log files of the operating system to cover their tracks. PAM SC audit files cannot be tampered with by anyone while PAM SC is running on the machine. Additionally, any change of PAM SC rules will always be audited. PAM SC protects the audit records and has facilities for routing logs anywhere, making the audit data truly reliable.

Audit events can be recorded on the following instances: success, failure, both or none, all in accordance with the administrator's discretion and with the sensitivity of the resource. In addition to auditing access to specific resources, PAM SC can be configured to audit the actions of individual users. PAM SC also has a dynamic trace facility that monitors all system events as they occur. The trace can be used for troubleshooting, or if extremely detailed information is needed about user actions.

PAM SC has an audit routing mechanism, which uses the log-routing and log-collection daemons `selogrd` and `selogrcd`. The source machine runs `selogrd` and routes the events to the collector machine that runs `selogrcd`. An organization can have multiple collection points that can receive different versions of the data based on the filtering rules created. One highly effective way to secure the audit logs would be to route them to a collector behind a firewall that only allows the PAM SC audit data to pass through it. PAM SC does not have to, but should, be run on the collector machine.

To further secure PAM SC logs, use the powerful log routing facility to consolidate the audit logs of many machines onto one isolated server. The log routing facility provides many filtering options so you can decide to send some or all of your logging to the central server. The audit data can also be passed along to any event monitoring or tracking system. This is configured through the log routing configuration files.

**Audit Events**

In addition to the default PAM SC startups and shutdowns, administrators can select which events to audit, including user logins, logouts, access to resources, and change directory commands.

**PAM SC CONTROLS WHO CAN DEFINE AND VIEW AUDIT RECORDS**

PAM SC controls who can define and view audit records. The audit flag is a special permission flag in PAM SC that allows users to view the audit records. Administrators do not need the audit flag, so a separate group can be created to do the auditing of system activity and events. Filtering can be inclusive or exclusive, support wildcards, and define by class, user, resource, access mode, success mode, date and time, logins, and logouts.

PAM SC audit data also includes Warning Mode events. When a security rule is set in Warning Mode, the access control restrictions for that rule will not be enforced but they will be logged. This is useful for the development and introduction of new rules.

Audit events can be viewed with a command line utility called `seaudit`, which has options to format the output and filter output based on categories such as resource and time of day. Typical Unix/Linux utilities and scripts can be used to generated customized reports.

You can use existing auditing systems, such as syslog, together with PAM SC. PAM SC protects the syslog entries, making them more reliable.

## Central Policy Management

Large enterprises use PAM SC to secure thousands of servers.

**PAM SC PROVIDES CENTRAL POLICY MANAGEMENT THROUGH THE PAM WEB INTERFACE**

To solve the problem of securing a large number of servers, PAM SC provides central policy management through the PAM web interface. The policies (groups of rules) you create are deployed are propagated based on device (server) or device group assignment. The policy administration can be centralized at a corporate level, decentralized by business unit, geography, application area, or any combination of these using the role-based access control.

The policy administration is also available through REST API, so it can easily be integrated with any automation and administration solution. PAM SC supports remote administration through the standard command line administration utility (`selang`).

An important feature of PAM SC is that security for a given server could be enforced even if the network connectivity was not working properly. As a result, every instance of PAM SC has its own authorization and enforcement engine and the corresponding security rules database. This security rules database is called the PAM SC Runtime Database. The result is that rules are decentralized and each server can exist securely on its own.

## Conclusion

**PAM SC IMPROVES THE LEVEL OF SECURITY ON AN INDIVIDUAL MACHINE BASIS, THEN SEAMLESSLY MANAGES HOST SECURITY FOR ALL SYSTEMS**

In today's sensitive IT environments, security needs to be every organization's top priority. While IT managers should have a choice of UNIX, Linux and Windows application platforms to meet business needs, that choice should not compromise security. Having multiple operating systems does not mean that numerous error-prone manual security procedures need to be used.

PAM SC improves the level of security on an individual machine basis, then seamlessly manages host security for all systems by placing a small, unique set of patented technologies on each server. These technologies enable powerful user access control, rich distributed policy management, and compliant auditing. These and other unique security management features make Symantec Privileged Access Manager Server Control the premier host solution.

For more information, visit our website at: www.broadcom.com

Copyright © 2025 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

PAM-SC-WP100 September 29, 2025