helpsystems

GUIDE *(Cybersecurity)*

The Essentials: Privileged Access Management

Administrator accounts have privileges to access any data and execute any application or transaction, typically with little or no tracking or control. These accounts, which in some enterprises number in the hundreds, are frequently not tied to specific individuals, so the accounts can be used to do virtually anything, with little or no possibility of detection.

In this white paper, discover how you can effectively and efficiently control privileged UNIX and Linux accounts using centralized access management.

Properly defining, controlling and monitoring administrative privileges in IT systems continue to be significant challenges for organizations of all sizes. And while in the past, controlling privileged accounts made good business sense, today, it is mandated by many regulations.

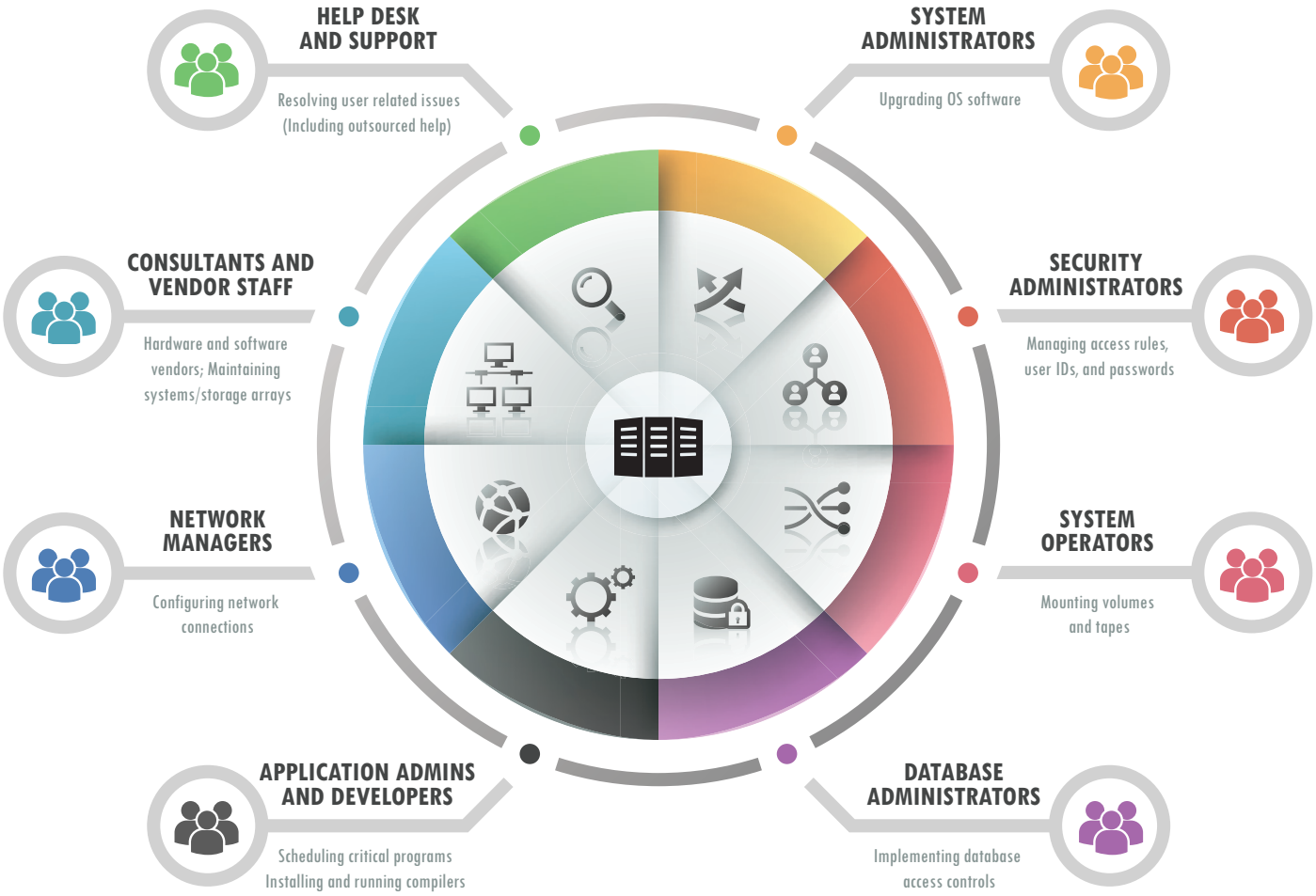
In addition to the increased potential for failing IT security audits, sharing root and other privileged accounts can lead to a significant increase in the risk of fraudulent activities by employees, an even bigger threat to corporate value.

In this guide, discover how you can effectively and efficiently control privileged accounts using the latest in adaptive access controls management. You will also learn about how to leverage the latest in Role-Based Access Controls.

What's the Problem With Privileged Accounts?

All computer operating systems require some kind of unrestricted administrative access to enable system management. Security models based on group policies and privileges, overlaid and accumulated through group membership and security principals, as is the case in Microsoft Windows environments, generates one set of challenges. On UNIX and Linux systems, the unrestricted "root" account poses a different and particularly troublesome situation. And similar super user issues can also arise with privileged accounts for database administrators.

The real challenge is that the problems associated with controlling privileged and root accounts increases quickly as the number of people who need powerful administrative access for various job functions grows. Examples of roles and tasks that may require privileged accounts include:



Considering all of the work that needs to be done using privileged permissions, most organizations find that controlling the ability to switch to these privileged accounts can quickly become difficult to manage and monitor. Furthermore, granted permissions are rarely reviewed or revoked, which means that users unintentionally accumulate more and more privileges over time as their job functions change and new access rights are granted. This review cycle can become even murkier when outsourced partnerships are involved.

As delivered, standard UNIX and Linux systems have poor support for controlling delegation of privileged accounts. The easy way out for support staff is to share superuser passwords with a variety of administrative "controls":

- Write the password on a piece of paper
- Create multiple user accounts with different passwords that share the root UID 0
- Create emergency superuser accounts, that are protected with dual controls, by splitting passwords in half so two people need to communicate prior to use
- Implement a software-based solution that generates random passwords for emergency superuser account access

Another common practice is delegation through suid or sgid, allowing a process to run in the context of the program file's owner or group. However, delegation through suid or sgid is also a much appreciated door-opener for hackers, since such processes can be readily exploited.

Regardless of which of the above approaches is being used, all fail to provide efficient, effective protection of privileged accounts. Even an elaborate process for password check-out and subsequent password resets can remain dependent on someone with unlimited privileges.

And there are some other considerations to the challenge of controlling privileged accounts:

Securing data in transit: Even if a user has been granted administrative access in a controlled fashion, used in a clear-text telnet session, the password can be intercepted and shared in an uncontrolled way by someone listening in on network traffic. And asking your users to switch to a peer-to-peer solution, such as user-managed SSH, only partially improves the situation; it doesn't scale, you remain vulnerable to man-in-the middle attacks, and you have lost control to random support staff who may want to make use of port-forwarding techniques for purposes that are not aligned with your business objectives.

Passing IT audits: Even if you trust your process for password sharing, you still won't be able to satisfy your auditors. When the audit log shows that "root" did something, your auditor will want to know the real name behind "root".

Capturing superuser's actions: Once root privileges have been acquired, the omnipotent user can do just about anything, including deleting log files and then re-configuring the operating system for future private use. You need a way to effectively monitor and record what the root user is doing once they have acquired their privileges.

Maintaining security during mergers and acquisitions: Chances are your organization has put a lot of effort into standardizing the identifier used by your staff to login to the network and access corporate data. Most organizations also have decent control over how a person's identity is managed and removed from these central databases. Non-personal accounts (system and software accounts) are typically not allowed in your "people" database by your auditors. Let's consider what happens if your organization acquires another company. Chances are that their personal ID's have a different structure, that their group memberships will vary, and that their change management controls will not be the same. Now your support team needs to manage two sets of user stores, and two divergent standards for non-personal accounts. How quickly can you convince your auditors or industry regulator you're in control of access to your systems and data?

What's really required to protect privileged accounts?

To effectively and efficiently control privileged accounts, a combination of adaptive access management capabilities is required:

Centralized administrative management of user accounts across all servers (both "real" and "virtual"): Centralized administration of user accounts across your diverse UNIX and Linux server estate ensures that you can monitor and audit which specific user has what type of access on which machine. Centralized management will also facilitate automatic provisioning and rapid disabling of user accounts as needed across the security domain.

Seamless, agnostic integration with existing corporate directories: None of us live in an ideal world; web-enabled business processes, external supply chains, and outsourcing models mean staff from multiple organizations may have access to corporate data. It is entirely possible that there isn't a single "corporate ID database". Therefore, access management solution needs to integrate seamlessly with multiple corporate directories and identity management systems so that team and group identities can be associated automatically to the correct systems, business applications, and data.

Contextual authentication: Authentication is the first step in any access request process. You must first authenticate that the user is who they say they are before you can authorize the access. Whilst there are numerous authentication offerings on the market, the real trick is being able to adapt the authentication required based on the context of the access request. And while you could apply multi-factor authentication for all resources and access requests, it would be over-kill for many day-to-day tasks and overly expensive and management-intensive to implement.

Basically, contextual authentication enables organizations to target strong authentication to particular servers and roles that bring a higher level of risk. This requires flexibility in the access management solution and the ability to work in concert with the authorization rules discussed in the next section. Consider a group of developers. You may want this particular developer role to be able to access a time-reporting system using password authentication, but to use a smart card when accessing a sensitive code repository. You may even want these authentication rules to adapt depending on where the developer is accessing the servers from.

Targeting authentication methods based on the context of the request enables you to avoid using blanket strong authentication, saving money while enabling high levels of security.

Granular authorization: Instead of allowing functional accounts such as "root" or "sysdba" to login, you need to have proactive, enforceable authorization rules that mandate the use of individual and auditable user accounts. Using access controls, any switch to a privileged functional account for specific job functions or tasks will be tied to the named user. The authorization rules should be configurable and central policy automatically applied based on who is making the access request, where that request is being made from, which server they are wanting to access, how they want to access the server (e.g. Secure Shell) and when they are making that request.

Centralized management of adaptive, granular authorization rules that can be enforced throughout the security domain means that controlled switching to a privileged function will not require privileged password sharing. In fact, in normal support related functions (with policy-driven access management), 90% of privileged operations can be controlled with a combination of escalating authentication challenges. Depending on the access policy and where they are on the network, the staff member may be again challenged with their own password or other authentication mechanisms such as tokens, Kerberos tickets, etc.

Secure communications that are mandated by the access rules: It is not sufficient to provide users with the option to use a secure connection rather than clear-text telnet. You must be able to enforce its usage where needed. As well, it

is not sufficient to delegate the task to establish encrypted connections to individual users (e.g. letting them maintain user and host public SSH keys as they find suitable). In reality, delegating the task introduces a new authentication authority which in turn subverts centralized management and enforcement of access rules. Look for a solution that enables you to define the service level required for a particular access request including the ability to authenticate and authorize SSH at the sub-service level.

Consolidated audit logging: An effective approach to protecting privileged accounts includes centralized audit logging with a detailed record of user activities. However, consolidating and cross-referencing local audit logs from hundreds of thousands of diverse servers is a difficult task. Look for a solution that can deliver consolidated audit logs and reports from across your server domains. It is also important that these logs are kept on a separate security domain so they can be trusted.

Consolidated operational reports: Gaining control of your privileged accounts is crucial to protecting your business from insider fraud. As well, proving that you have control to your regulators or auditors will keep your executives out of jail. However, it is also important to provide senior management with real-time visibility (across business units) on access control status and patterns of behavior. This "operational data" will enable your organization to proactively identify access control issues and mediate problems. Look for solutions that provide both operational and compliance reports, and because your organization will have specific requirements, make sure the reporting functionality is easily customized.

Secure keystroke logging: For sensitive sessions, you must also have the ability to adaptively enforce full keystroke logging such that administrator activities can be tracked in detail. Changes to data privacy regulations will likely mandate that these keystroke logs are not accessible by your system security staff. Look for a solution that not only securely holds these keystroke session logs, but can also limit release to approved auditors.

Adaptability and maintenance: One thing is quite certain, you will need to adapt your access management solution to meet your current and ever changing needs. Look for a solution that is easy to configure and a vendor that is focused on the access management space to future proof your investment.

What options are out there?

Organizations struggling to achieve effective protection of privileged and root accounts often evaluate three different alternatives:

1. Create a home-grown solution based on operating system capabilities, available utilities such as "sudo", clever password management procedures, and lots and lots and lots of scripts. Except for in very small organizations, the attempt to create a home-grown solution will often become extremely costly, as well as requiring system administrators to do programming instead of their day-jobs. The home-grown solution is often found insufficient from an auditor's perspective as well. Even if they provide an acceptable level of password management capabilities, they often fail to fully address auditing requirements. Start-up companies often begin in this position, looking to "save money" as they build their architecture. However, even in start-up companies, particularly in the Life Sciences sector, sensitive information such as clinical trial information may be subject to insider threats. Before attempting to go public, their auditors require internal controls to be toughened up, and the home grown solution is superseded.

2. Combine various commercial or open source point solutions to create an operating system environment that provides an effective approach to protecting privileged accounts. This typically involves using one solution for user provisioning, another for centrally managed secure communications (SSH), a third for password management, and possibly another tool for audit log consolidation. While the combined solutions can amount to something powerful, in the end, one important aspect is lost: centralized access

management on one security system. Combining multiple technical solutions into one leaves conceptual gaps, which in turn lead to security flaws and inefficient management. Ironically, while cost-awareness may well be the primary driver for exploring this option, it typically ends up costing an organization more than other options. Organizations in this bind often plan to outsource their piece-meal security problems to a third party. You may move the problem, but responsibility for control failures still reside with your CIO and CISO.

3. Invest in an Identity and Access Management solution (IAM). Several of the leading IAM solutions do provide several of the components required to control super user privileges, but these systems are very expensive and complex to install and operate and still do not fully address the capabilities needed to effectively enforce authorization of privileged accounts with enough granular detail and managed in a risk-adjusted manner. Implementations tend to be lengthy, slowing the overall ROI.

4. Invest in an agile and extensible Access Management solution. There is another option. The Powertech Identity & Access Manager (BoKS) solution provides all of the components needed for effectively protecting privileged and root accounts both proactively and adaptively, without all of the overhead, costs, and complexities of full-blown IAM infrastructures. Identity & Access Manager is both directory- and IAM- system agnostic, and includes connectivity to simplify integration and support strategic changes you make over the next few years.

Protected Delegation of Privileged Accounts

Identity & Access Manager enables you to centralize administration, authentication, authorization and audit across UNIX and Linux servers. Entire domains of servers running heterogeneous operating systems can be securely managed from one central, web-based administration console with a single policy set.

In addition to providing fine-grained access rules to manage which users can access what network or local service on which server, when, and from where, Identity & Access Manager also delivers additional functionality for effectively controlling privileged and root accounts. First, many of the common management functions in your server domain can be performed using the secure administration interface via a web browser. This greatly reduces the amount of operations requiring administrators to know privileged passwords. You can also define sub-administrators who are allowed to work on certain tasksor on specific portions of your server environment and track their actions.

On protected servers, SU can only be performed if the user has a specific access route allowing him or her to do so. Even if a user knows a privileged account password, they cannot use it to become the privileged user unless they have been granted permission to run SU.

Identity & Access Manager includes programs that enable your administrators to delegate privileged command execution. SUEXEC allows users to perform specific operations as another user. Again, users can only perform SUEXEC controlled operations if they have explicitly been assigned permissions to do so. For ease of management, you can specify, down to program argument level, exactly what operation the user can carry out as another user, and group the program command permissions.

You can configure the system so that users can run SU or run SUEXEC, using their own passwords or tokens, which removes the need for any of your administrators to know privileged account passwords.

There are still rare cases where you will need the root or administrator password. Password Vault, an optional add-on module, is a solution that manages the checkout of privileged account passwords and automatically changes passwords after the configurable checkout period has ended. Password Vault removes the need to share passwords, enforces limits of which users can check out which passwords, and helps you avoid having privileged account passwords active in the system too long.

Identity & Access Manager also helps you make the next step, and make the most effective use of existing security protocols such as SSH with centralized management of the keys and policies. With Identity & Access Manager, you can easily deploy SSH public keys to hosts and users correctly (ask your technical team, it's a real headache). Even more importantly… you can enforce individual policies for access requests centrally. For example, within the configurable authorization rules, you can define SSH access on sub-service level, and set up a rule that allows the transfer of files using SFTP w/o granting access to interactive terminal session for any given access request.

SUEXEC (a sudo replacement) operations can be keystroke logged based on policy, with a configurable level of keyboard input and on-screen output recorded for reference. Keystroke logging provides a forensic level of traceability, and ensures that no user in your organization can perform any subversive activity in the guise of a privileged user.

And finally, Identity & Access Manager also enforces use of encrypted communications to protect privileged account passwords.

Organizations using Identity & Access Manager can avoid sharing privileged account passwords, and indeed, once the solution is configured, there very limited need to use these passwords in day-to-day activities. When operations are traceable back to a specific person, organizations can significantly improve the security over their IT assets and greatly simplify their IT audits.

About Identity & Access Manager

Identity & Access Manager transforms your multi-vendor Linux and UNIX server environment into one centrally managed security domain. It simplifies your organization's ability to enforce security policies, and control access to critical systems and information. With full control over accounts, access and privilege, IT and security teams can proactively prevent internal and external critical system attacks before they start.

Identity & Access Manager protects your most critical systems and data so that you can focus on what is more important—accelerating the growth of your business.

Learn More | Request a Demo

helpsystems

www.helpsystems.com

About HelpSystems

Organizations around the world rely on HelpSystems to make IT lives easier and keep business running smoothly. Our software and services monitor and automate processes, encrypt and secure data, and provide easy access to the information people need.

(RJB-PTX-ESSPAM-GD-0918-R1)