

Realtime  
publishers

"Leading the Conversation"

# The Essentials Series

# Managing Access to Privileged Accounts

*sponsored by*



*by Ed Tittel*

---

Article 1: Understanding Account Access Management .....	1
Types of Access .....	2
User Level.....	2
Administrator Level .....	2
System Level.....	2
Application to Application.....	3
Application to Service (A2S).....	3
Issues with Privileged Access .....	3
Privileged Password Management.....	3
Programmatic Access.....	4
Session Control and Audit .....	4
Logs and activity tracking.....	4
Compliance Issues .....	5
Summary .....	5
Article 2: Privileged Password Management Systems .....	6
Types of Accounts .....	6
Guest or Anonymous Accounts .....	6
User Accounts.....	6
Privileged Accounts .....	7
Shared System Accounts.....	7
Service Accounts .....	8
Historical Approaches to Password Management .....	8
Modern Password Management.....	9
Basic Requirements and Functionality .....	10
Advanced Requirements and Functionality .....	11
Summary .....	12
Article 3: Privileged Session Controls.....	13
Providing Access: Key Resources, Systems, and Information .....	13
Compliance and Audit Requirements .....	13
Types of Privileged Sessions .....	14
Vendors.....	14
Consultants.....	14
Remote Employees .....	14

---

Remote Administration.....	15
Internal Access to Sensitive Systems.....	15
Privileged Session Characteristics and Requirements .....	16
Inability to Enforce Security Policy Requirements .....	16
Inability to Enforce Remote Access Methods and Controls.....	16
Span of Control and Access Control Issues.....	17
A Solution for Remote Privileged Sessions.....	17
Summary.....	18

---

## Copyright Statement

© 2008 Realtime Publishers, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers, Inc or its web site sponsors. In no event shall Realtime Publishers, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at [info@realtimepublishers.com](mailto:info@realtimepublishers.com).

---

## Article 1: Understanding Account Access Management

One of the biggest IT challenges facing organizations today relates to the uses of privilege. This does not mean use or even abuse of personal prerogatives and powers but rather the ways in which high levels of access are assigned, managed and controlled, and tracked on systems and network infrastructure elements inside the firewall. There are many factors that can contribute to lack of sufficient controls and transparency, with significant concomitant risk and exposure, inherent to traditional methods for managing accounts and their passwords.

In this first of three articles on account access management, privileged account management, and privileged session management, we explore the general terrain inside which access occurs, how it is managed, where exposures can (and do) occur, and how regulatory compliance and industry best practices play into access and its management.

Throughout these articles, we will make extensive use of the following terms to frame and explore this discussion

- *Privileged account*: Any account that includes rights or permissions that enable its user to access sensitive data, control system or network infrastructure element configurations and behaviors, manage users or other system and network resources, or manage applications, file systems, and other major IT system building blocks.
- *Privileged password*: Any password that enables users to operate within a privileged account. Likewise, other types of credentials also used for authentication, such as biometrics, smart cards, token devices, and so forth, may be called *privileged credentials*.
- *Privileged session*: Any network session that links a client on one system to one or more other systems or network infrastructure components that uses a privileged account. If the client and host(s) that participate in a session are on the same local network, the term local can be included to identify a local privileged session. If the client is outside the firewall and the host(s) is on a local network, the term remote is included to identify a remote privileged session.

As you'll see in the following discussion, access to systems and resources come in various kinds. Any kind that meets the criteria for a privileged account as defined in the preceding paragraph also qualifies as privileged access.

---

## Types of Access

When any particular system, service, or application is accessed, such access occurs with some level of associated rights and privileges. Some kinds of access may involve little or no rights and privileges; others may involve carte blanche when it comes to rights and privileges, granting associated individuals or accounts the ability to change anything and everything under their purview.

### ***User Level***

At the user level, access involves the ability to see and use shared public system resources but affords little or no (mostly no) control over such things. The same goes for applications and services, and the ability to define or redefine system behavior, resources, and, of course, rights and permissions. Users will usually have broad rights and permission where their own files, documents, email messages, preferences, and desktop settings are concerned, but no rights or permissions for similar kinds of resources belonging to others.

An important principle for managing access is known as the “principle of least privilege.” It can be best summarized as a principle to “provide access only to resources, services, and applications necessary to do the job, no more, no less.” This principle applies equally at all levels of access, but at the user level, this generally entails making sure that no user account’s rights and privileges err on the side of too much access (too little access invariably leads to complaints, which then leads to necessary corrections). Regular audits of the rights and permissions associated with user-level accounts are the best way to ensure that the principle of least privilege is honored in practice as well as in theory.

### ***Administrator Level***

At the administrator level, elevated rights and permissions come into play. Thus, this is one of the privileged types of access. Administrators may have privileged access to single systems, collections of systems, or entire networks and all their systems and infrastructure elements.

On any system where an account is granted administrator-level access, anyone who can use that account can install or uninstall updates, applications, and services; perform backup and restore operations; manage configurations (including the operating system—OS); and do anything they see fit with local file systems. That said, OSs might prevent even administrators from accessing or altering certain key files, but administrators can also escalate their rights and privileges to circumvent such behaviors when they must.

### ***System Level***

At the system level, anything and everything on a particular system becomes accessible, and may be added, altered, or deleted at will. That’s because this level of access matches the same level that the OS or control program itself enjoys at its maximum rights and permissions, which touch everything under its purview. On many network infrastructure elements, administrators may share a single, common system-level account that they use to perform updates and installs, manage configurations, and perform other maintenance tasks.

---

## ***Application to Application***

When applications interact with one another, generally one application will initiate such interaction. This means the calling application must have rights to access the called application plus related system resources, data files, and interfaces. This type of access is properly identified as an application-to-application (A2A) account. As applications are being developed, they may be granted system- or administrator-level access to other systems to make them easier to test and debug. Although the principle of least privilege argues strong against leaving this alone in production settings, ignorance or convenience often argue otherwise—and win!

## ***Application to Service (A2S)***

When applications interact with a service, the application will typically initiate such interaction. This means the application must have rights to access a called service along with related system resources, data files, and interfaces. This type of access is properly identified as an application-to-service (A2S) account. As with A2A, as applications are under development, they may be granted system- or administrator-level access to services to make them easier to test and debug. Also, as with A2A, A2S accounts' rights and permissions far too often end up exceeding what the principle of least privilege would dictate be granted to them.

## **Issues with Privileged Access**

Of the preceding types of accounts, all of them except user level often enjoy (or assume, as with unchanged instances of A2A and A2S) privileged access to systems, infrastructure elements, applications, services, and so forth. Such access involves all kinds of potential security issues as well as potential risk and exposure to financial losses, legal penalties, and other undesirable consequences. All these factors help to explain why managing privileged access is critical and why tracking privilege access activity is essential for regulatory and best practices compliance.

## ***Privileged Password Management***

In many ways, managing privileged access is all about managing the passwords or other credentials used to access privileged accounts. Proper management of privileged passwords requires that the passwords themselves comply with governing security policy so that such passwords are sufficiently long, strong, and complex to defeat attack. Proper management of privileged passwords may also require that passwords change at specified intervals and that individuals who use them provide additional proof of identity as the resources and assets they access become increasingly sensitive.

Under some circumstances, in fact, users may never see or know what passwords they're actually using to access systems or network infrastructure components. That's because a password management system may sometimes provide all access by proxy. In this kind of situation, users request privileged access from the password management system, and are granted such access pursuant to sufficient proof(s) of identity (password, retinal or fingerprint scan, Smart Card, token device, or whatever factors may be involved in multi-factor identification schemes). The password management system stores passwords so that users can't interact with them directly, though they can use them as needed and as permitted. Among other things, such an approach provides a mechanism to coordinate shared passwords and to manage exclusive access to accounts for which passwords are shared. The system can even automatically change the password every time it's used if security considerations argue that one-time use is called for.

---

## ***Programmatic Access***

When applications must call on other applications or services to perform specific tasks or access certain resources, such access often ends up hard-coded into configuration files, batch files, or scripts that are invoked whenever an application needs to call on another application or service for any reason. The problem here is that passwords may be stored in plain text in these various files, where any user with privileged access can open and read them, even if the principle of least privilege argues that such information should not be made available to those individuals.

Furthermore, the dispersion of and knowledge about such files may pose potent barriers against enforcing security policies that require passwords adhere to specific length, strength, and complexity rule. Likewise, widespread, undocumented use of such passwords also erects major obstacles to adhering to mandated password changes at regular intervals.

What programmatic access really requires is some kind of well-documented application programming interface (API) that can work with strong credentials, such as digital certificates or private/public key pairs. This approach enables ready access to a password management system instead of storing password information directly (possibly in inappropriate or improperly maintained forms and formats) so that passwords can be centrally managed and stored independently from application scripts, configuration files, and other static and dangerous forms. Such an approach also makes it easy to enforce security policy requirements governing password length, strength, complexity, and frequency of change, and makes it completely unnecessary for programmers to know or store passwords at all.

## ***Session Control and Audit***

Passwords may be shared by design or as circumstances permit, but this defeats the notions of individual and organizational accountability. Individual accountability means that individuals' actions and activities must be distinguishable so that changes, additions, and deletions to systems, configurations, data collections, and so forth can be properly ascribed to those who enacted them. Organizational accountability means (and sometimes legally requires) that the organization be able to furnish logs or audit trails of such actions to demonstrate proper prudence, due diligence, and (where relevant) compliance with applicable laws or regulations governing access, confidentiality, integrity, and security.

## ***Logs and activity tracking***

Modern computing systems often involve interactions with graphical user interfaces (GUIs) where mouse movement and clicks are as important as keystrokes (or replace them entirely) in recording and tracking user activity. When it comes to logging activity, this means that modern systems must record all mouse movement and activity on a per-account/per-session basis as well as recording keystrokes. Only in this way can user activity be replayed or analyzed for evidence of adherence to or violation of security policy, employee or contractor guidelines, acceptable use policies, and so forth. Any capable system must be able to completely reconstruct what happened at any given moment; what resources, systems, or infrastructure components were involved; and what results ensued from the actions that occurred.

---

## **Compliance Issues**

Various forms of law and regulation require specific types of recordkeeping for industries that include financial services, healthcare, and others. This information is also subject to specific confidentiality and privacy restrictions, both as it is stored digitally, and whenever and however it is transferred from one party to another. Likewise, all publicly-held companies must comply with legislation that governs how accounting information is acquired, stored, audited, and reported.

Though this may seem to have little to do with IT at first blush, because privileged accounts can access information related to all these areas and concerns, actions on any of this data from privileged accounts must be logged so that it can be audited, analyzed, and possibly serve a probative purpose when circumstances call for legal investigation or proceedings to occur. Though activity logging and tracking may serve a variety of purposes, nowhere else are these capabilities as important as when it comes to adhering to laws and regulation that require formal proof of compliance, due diligence, and proper care and treatment of information and accounts, and related transactions or treatments and outcomes.

## **Summary**

When all the various factors related to privileged access are considered—especially activities undertaken inside privileged accounts—the issues involved require capable effective management of passwords, strong authentication, and tracking or logging of privileged account activities. In the articles that follow, we will explore how managing privileged passwords and sessions can help to mitigate the issues involved, and reduce the risks and exposures occasioned thereby.

---

## Article 2: Privileged Password Management Systems

Privileged password management systems provide a framework within which passwords for privileged access may be established, managed, and maintained, along with services whereby password requests may be handled, whether those passwords originate from human users (interactive) or from an operating system (OS), application, or service (programmatic).

Privileged password management systems generally apply specifically to privileged accounts. Though these relate to the levels of access described in the preceding article, we begin our discussion here with a recitation of the various types of accounts in use in most enterprise environments and the characteristics associated with each.

### Types of Accounts

To some extent, the levels of access described in the preceding article map to various accounts created to service users at varying levels of privilege. That said, this makes it imperative to understand that a “user account” and an “ordinary user account” are two different ways of identifying the most common and frequently occurring type of account. Likewise, it’s also important to understand that the kind of account used for A2A or A2S access (called a “service account” in the sections that follow) may or may not be endowed with privileged access. What’s more important in this designation is that the term refers to accounts that are accessed programmatically rather than interactively.

#### ***Guest or Anonymous Accounts***

Guest or anonymous accounts not only can be accessed without a password or other credentials of any kind but also, on most systems, they define the level of access that is permitted to anyone who accesses that system. The principle of least privilege argues forcefully that guest or anonymous accounts should be endowed with very few rights and privileges, if not denied access altogether. In fact, the more sensitive a system (as a general rule, there is no such thing as guest or anonymous access to network infrastructure components) or its contents, the less likely it becomes that guest or anonymous accounts will be available.

#### ***User Accounts***

These accounts are defined for normal, ordinary users to conduct typical, everyday activities associated with their jobs: running applications, accessing services, reading email, browsing the Web, and so forth. User accounts normally include a network file store of some kind, wherein the account owner can create, modify, and delete files. The group or department for which a user works will normally go a long way toward defining what kinds of applications and services he or she is allowed to use, what kinds of network resources he or she is allowed to access, and whether and what kind of Internet access is enabled. Aside from their own desktops (and sometimes not even there), ordinary users have little or no control over systems, configurations, data repositories, network infrastructure components, and applications.

---

## ***Privileged Accounts***

As defined in the first article, a privileged account includes any account with rights or permissions that enable its user to access sensitive data, control system or network infrastructure element configurations and behaviors, manage users or other system and network resources, or manage applications, file systems, and other major IT system building blocks. By definition, this includes administrator-level and system-level access. Whether A2A- or A2S-level access falls under this umbrella depends on whether the rights and permissions associated with such access meets this definition. By default or accident, too many of such applications and services inherit or obtain more privilege than they need, and associated accounts therefore qualify as privileged; but this need not be the case.

Of course, managing privileged accounts properly is essential because such accounts confer the keys to the kingdom—at least, for the systems or network components for which they are valid—to those who use them. Password management systems not only help to ensure that sufficient proof of identity are furnished to obtain privileged access but also provide tools for tracking and auditing that let users with access to privileged accounts know that they will be held accountable to definite and specific standards of conduct, behavior, and activity while using those accounts.

Without such admonitions, and the logs to back them up, there's nothing to stop savvy administrators from making unauthorized changes, installing unauthorized software, or performing other illicit actions, and then erasing the tracks of their activities from logs and audit trails to which they already have access. The presence of an external watchdog cannot prevent such things from happening, but these actions will leave an indelible trail in logs which those administrators can neither access nor alter.

## ***Shared System Accounts***

This term applies to any privileged account, be it for a computing system or network infrastructure component of some kind, that is shared by more than one user, typically an administrator or a designated third party (such as a consultant or a service provider, whether working locally or remotely). Shared system accounts are particularly challenging to manage because more than one person can use such an account, and more than one person might be logged into such an account at any given moment. From the standpoints of accountability and auditability, password management systems bring order to the potential chaos that shared system accounts can cause.

---

## **Service Accounts**

This term applies to any kind of application or service that makes programmatic access to some kind of account, be it privileged or otherwise. The value of password management systems for this kind of access is manifold. First and foremost, it eliminates the need for any program or service to store passwords internally or in related scripts, batch, or configuration files related to their use. In turn, this makes it trivial for enterprises to enforce security policy password requirements even for programmatic access, including password length, strength, and complexity requirements, as well as frequency of change requirements. Where one-time passwords are deemed necessary for programmatic access, in fact, a good password management system can provide them via a single configuration parameter.

Password management systems can require that programmers only know how to reference the right digital credentials to establish the authenticity and veracity of a calling application or service, usually in the form of a digital certificate or public/private key pair. Even this data can be made transparent to an application or service by storing such information as a “call by secure reference” rather than an instantiation by reference in the code. Finally, programmers need not be given privileged passwords to use with the applications or services they build. Because such passwords might confer other privileged access to which those programmers are not entitled, this helps to close a potential security exposure that sharing such passwords might entail in some situations.

## **Historical Approaches to Password Management**

In the absence of a formal password management system, IT professionals have resorted to all kinds of tools and techniques to manage them informally. These include numerous forms of paper records, from the infamous Post-It note on a terminal or desktop somewhere, to more sophisticated forms of secure password storage under controlled physical access (in a safe, locked filing cabinet, or something similar), usually in the hands of a separate security department somewhere.

Digital analogs to such static forms of storage also abound. These might include simple text files, Excel or other spreadsheets, and even one or more special database files. These electronic password stores may reside in plain-text files, or they might be encrypted in some form or fashion.

Whether on paper or in electronic form, all these historical approaches are subject to security risks. Some of them may also be subject to availability risks. To begin with, the presence of a plain-and-simple password store of any kind poses a security risk related to access. Whether encrypted or not, anyone who is allowed to access the store can also access any of the passwords it contains. This may or may not be consonant with the principle of least privilege, where administrators who are granted access to one or some systems may obtain access to all systems for which passwords are stored. Depending on the type of store involved, and how secure that store might be, the possibility of unauthorized or illicit access may be more or less troublesome.

---

Availability risk becomes especially apparent when other departments outside IT become involved in password access. Let's examine the hypothetical case in which a security department maintains a safe where passwords are stored so that they may be retrieved when needed. Once an administrator determines that such a password is necessary, he or she must contact the other department, provide sufficient proof of identity to warrant access, then accept delivery of the password. If working after hours or on a holiday weekend when security department staffing may be low, slow, or unavailable, it might take hours or days to obtain necessary passwords to restore a backup, obtain access to, or reinstall some specific system. In some cases, it should be obvious that such delays could not only be unacceptable, they could also involve financial losses or legal exposures.

All these historical systems also suffer from resistance to password change, update, and automated policy enforcement mechanisms. Whether on paper or in digital form, these approaches all require manual updates and changes, and rely on the individuals who use them to ensure compliance with password security policy stipulations. The work involved in making such changes may itself be a powerful deterrent to enforcing policy. Perhaps more important, the manual update technology that applies also means there are no detailed accounting, logging, or auditing capabilities built-in to this type of password management.

Ultimately, the root problem with historical approaches to password management arises from the static and ad-hoc nature of the password storage and access mechanisms involved. A growing need for dynamic storage backed up with automatic enforcement of security policy, 24/7 access to authorized parties that provide adequate and acceptable proofs of identity, and automatic logging of use and activity drives the need for the types of modern password management systems we describe in the sections that follow.

## **Modern Password Management**

A modern password management solution involves a secure, automated facility that provides centralized and sophisticated services to establish and manage passwords, to control access to such passwords only to authorized personnel, and to provide secure delivery of passwords for use in real time. In addition, modern password systems either include or integrate with powerful authentication, and deliver robust, reliable programmatic access to applications and services.

---

## **Basic Requirements and Functionality**

Any capable password management system must include all the following capabilities in some form or fashion:

- *Secure, centralized password management and storage:* Whether delivered in the form of a standalone hardware appliance or a hardened server-based applications, modern password management systems must deliver a highly secure and encrypted system that may be managed from anywhere and be available everywhere it's needed.
- *Complete access coverage and control tied:* Modern password management systems must use strong authentication or identity management technology, or integrate with enterprise systems in place to supply such functionality, to enable authorized users to obtain privileged access to systems and network infrastructure components. Also, all communications between clients, the password management system, and managed systems and network infrastructure components should be encrypted so as to prevent eavesdropping or replay attacks.
- *Automated password check-in/check-out facilities:* For service accounts, the password management system must enforce access policies and lock out other authorized users of shared accounts where required. They must also establish records so that individual accountability can be maintained for use of service accounts so that audit logs may be correctly ascribed to responsible parties.
- *Automated update and policy enforcement:* Any modern password management system must be able to accommodate and enforce password security policy criteria and requirements on the passwords it manages, including password length, strength, and complexity criteria; frequency of change criteria; and so forth.
- *Accountability and activity logging:* Any modern password management system must be able to record individual account activity once a privileged account session is underway. This involves the ability to capture and store mouse movement and clicks, as well as keystrokes, so that complete and accurate replay of such sessions is enabled. This makes it possible to reconstruct activity perfectly, and to see exactly what was done, what resources were affected, and so forth.
- *Programmatic interfaces:* For service accounts related to applications (A2A) and services (A2S), a modern password management system must offer APIs, digital authentication mechanisms, and controls to enable programmers to access needed applications and services without having to hard-code passwords into applications themselves, or into related script or batch files. This also prevents programmers from obtaining access to privileged passwords.

---

## ***Advanced Requirements and Functionality***

Beyond the must-have functions described in the preceding section, the following items qualify as “nice-to-haves” in more capable password management systems. These items include the following:

- *Ability to meet real-world requirements:* Enterprise-scale password management systems must be able to accommodate tens of thousands to hundreds of thousands of privileged users, and thousands to tens of thousands of systems and network infrastructure components. It’s essential to choose a system architecture that scales to meet your needs.
- *Reliability and availability:* Modern password management systems must be sufficiently reliable to withstand various types of failures (network access, appliance or server, and so forth). They must also be sufficiently responsive to authorized access requests so as not to impede reasonable login or access times for systems and network infrastructure components, even on global-scale networks.
- *Dual/multiple approvals for password use:* For modern password management systems, it’s helpful to impose dual or multiple approval schemes for access to particularly sensitive or important systems and network infrastructure elements. This is the digital analog to the dual-key systems used for access to weapons systems and classified information. It provides an extra level of security for extremely critical access.
- *One-time passwords, plus automatic reset on check-in or expiration:* Some systems are sufficiently sensitive that no exposure of passwords can be tolerated. In those circumstances, it makes sense to enforce one-time passwords and to automatically reset those passwords when checked-in or after a timeout period has elapsed. The best of the modern password management systems make this functionality available.
- *Proxy intervention prevents password exposure:* For particularly sensitive systems, or where the principle of least privilege dictates that account users not be entrusted with passwords, modern password management systems can proxy all interaction between clients and systems or network components. This prevents users from learning passwords and enables all host communications and interactions to be inspected (and possibly rejected for security or other reasons) before they can be effected.
- *Advanced programmatic capabilities:* The best modern password management systems support both application and command-line interfaces (APIs and CLIs) that support automated password management and maintenance. These systems also enable proxy intervention for applications so that no exposure of actual passwords ever occurs.
- *Agentless operation, appliance based:* The best modern password management systems use secure protocols and work via standard Web browsers so that clients can interact with the system from their platform of choice, without requiring any software to be installed on the client side of the connection. And by housing the password management system in a secure, hardened, standalone network appliance, access to that system (and the systems and network infrastructure components it controls) can be completely controlled (and proxied, if deemed desirable).

---

## Summary

Modern password management systems offer more than centralized, secure access to privileged passwords for systems and network infrastructure components. They include automated enforcement of password security policy, auditing and logging for personal and organizational accountability, and capable programmatic interfaces to permit applications to access other applications and services as they like, without having to hard-code passwords into the application itself or related files or scripts. Modern password management systems also regulate use of shared passwords for service accounts, prevent unnecessary exposure of passwords, and ensure that passwords remain secure and inviolate, both in storage and as communicated to systems and network infrastructure components.

---

## Article 3: Privileged Session Controls

The problem with providing access to key resources through privileged accounts is that clients, consultants, and vendors are essentially handed the keys to the kingdom. A common security protocol is the principle of least privilege, which dictates that consultants, developers, and vendors should be given only sufficient access to get the job done. Providing limited, task-specific access to these key resources, systems, and information is crucial to preserving the protected environment.

### **Providing Access: Key Resources, Systems, and Information**

Areas most often supported by remote vendors include application management, desktop management, system management, platform development, and system security. Accordingly, privileged session control necessitates new and often entirely different requirements than traditionally exercised over internal employees. The enterprise landscape is comprised of multiple tiers of application and server platforms operating at various levels of infrastructure, each requiring individual security constraints and posing separate remote access challenges.

### ***Compliance and Audit Requirements***

Additional laws and federal regulations impact organizations differently. In fact, even the implementation of overlapping compliance coverage differs greatly among similar organizations. This creates an entirely new set of challenges and compromises.

For an enterprise operating in compliance with the Gramm-Leach-Bliley Act (GLBA), there's no reliable means of ascertaining whether a vendor has taken full precautionary measures to avoid violating customer privacy while rendering contractual services. Likewise, a company complying with the Health Insurance Portability and Accountability Act (HIPAA) cannot ensure that consultants haven't compromised or exposed patient data—as has happened in the recent past with security contractors retaining patient records only to have them stolen offsite. Companies can only be assured of their own compliance or negligence under such conditions, while outside sources may not be held to the same standard or governed by the same processes.

---

## Types of Privileged Sessions

An increasing interest in outsourcing gives rise to privileged session access, which only further complicates the privileged session scenario. In an ongoing effort to maximize uptime and reduce cost of ownership, more enterprises are permitting IT equipment vendors and service providers to access network and networked systems to diagnose and resolve issues remotely. Clients, consultants, and remote vendors are also becoming increasingly involved with company IT operations where cost-cutting efforts result in outsourcing responsibility for administrating, maintaining, and repairing on-site applications and equipment.

Granting, managing, and controlling access among these expanding external groups further challenges IT administrators within the company. Remote vendors staff personnel that are beyond the scope of review by company insiders, which creates difficult issues for maintaining accountability—particularly with any shift in the remote vendor’s staff or user base.

Administrator accounts are universally present, but network devices and security appliances often utilize a single administrator account without support for creation of sub-accounts. UNIX systems and Windows domain controllers are fully supportive of non-administrative account creation, though few restrictions apply to lower-priority administrative roles.

### ***Vendors***

An enterprise cannot expect to dictate the rules of engagement for a remote vendor operating entirely out of their own interests in accordance with their own guidelines. This is problematic for the home team environment particularly where compliance requirements, product implementations, and security specifications clash. Furthermore, each vendor is likely governed by entirely separate site-specific security requirements and governmental regulations originating within an entirely different country.

### ***Consultants***

Hiring consultants to perform on-site or remote services is equally challenging within a protected enterprise environment. Consultants aren’t bound to the same regulatory and compliance practices as the enterprise environments they work within, even if there’s a contractual obligation to retain privacy and maintain certain ethical standards. Companies and consultants are bound to entirely separate codes of conduct and modes of operation even where there is significant overlap in protocol or procedure.

### ***Remote Employees***

Even when employees work remotely, especially those who must operate in privileged sessions or make use of privileged accounts, many of the same concerns that apply to outsiders also apply to them. Although employees can be held accountable to codes of conduct and expected to comply with security policy and best practices, their remote sessions need extra levels of protection, inspection, and control.

---

## Remote Administration

Current approaches to remote administration are incomprehensive and incomplete. In a “jump box” scenario, vendors have access to a few defined machines from which they launch their sessions. Although this scenario creates a defined point of entry capable of monitoring keystrokes and providing session replay, it works only in limited situations (such as command-line activities). VPNs enforced by access control lists (ACLs) also permit limited connectivity to select systems with defined access but provide no replay and still creates administrative burden.

Enterprises need granular authorization for administrative connections permitting strong authentication at every entry point into the protected environment. Remote administration should operate under a segmentation strategy with secure connections to ensure privacy. Proxy deployment interrupts direct system-level connections to prevent creating a bridge for remote vendor malware from creeping into the protected environment.

## Internal Access to Sensitive Systems

Application-to-application (A2A) or application-to-service (A2S) interactions often call upon elevated privileges to perform tasks involving sensitive data or processes. Without proper user account management, these interactions too often involve storage of credentials in configuration files or registry entries in plain text where anyone with elevated privileges can access such information. In particular, the following types of data demand more probative and careful treatment:

- *Human Resources (HR) data:* HR information is also protected under compliance requirements that dictate secure handling and processing of confidential data. This encompasses both financial data and health records but includes other sensitive information. Administrators must keep track of who goes where and what authorized parties are allowed to access and change.
- *Credit card information:* The Payment Card Industry (PCI) Data Security Standard (DSS) is a non-governmental mandate defining the requirements for handling and processing credit card data. Non-compliance with terms and conditions of PCI’s DSS results in contractual penalties and revocation of the right to process credit transactions. Protecting against the prying eyes of internal employees is not enough—external clients, visiting consultants, and telecommuting vendors must also be considered.
- *Developer access to production systems:* Both internal and external developers often require access to production systems to pilot and implement new features or enhanced functionality—this is an unavoidable fact of the IT environment and product life cycles. Granting access to production systems in either case should be treated with equal amounts of discretion, even though the conditions and mechanisms for that access will vary conditionally with each case.

---

## **Privileged Session Characteristics and Requirements**

Managing privileged user accounts for remote vendor access involves entirely separate concerns and conditions than with typical remote employee access. Managing and monitoring access to sensitive systems must still be centralized, policy-driven, and automated, but the underlying characteristics and governing requirements are entirely different.

### ***Inability to Enforce Security Policy Requirements***

When it comes to dealing with remote sessions, privileged or otherwise, it's often difficult if not impossible to enforce security policy requirements for firewalling, antivirus, and other anti-malware controls, platforms, applications, and so forth. Thus, when it comes to creating a cooperative environment among diverse and regionally dispersed systems and users, it's improbable to expect uniformity among operating protocols and platforms. Different organizations address system and network concerns differently, using dissimilar operating systems (OSs), network protocols, and security paradigms.

With this collaboration among consultants and vendors comes a distinct inability to enforce compatible and consistent firewall and antivirus applications. Unfortunately, this simple disparity can present a significant stepping stone for an attacker—whether automated or manual, man or man-made—to gain foothold into one or the other organizations. Where one company enforces timely updates for signature databases, firewall rules, and application updates, another may be lax or lenient; this can cause significant problems for a well-protected environment.

### ***Inability to Enforce Remote Access Methods and Controls***

Also accompanying a difference in operating platforms, protocols, and procedures is a distinct inability to enforce compliance among VPN and remote access software. Clients, consultants, and vendors utilize individually or organizationally defined products to achieve remote access with partner companies, and this includes likely non-compliance with firewall policies as well. Furthermore, dial-up connections and VPNs introduce security vulnerability and inherently lack sufficient auditing capabilities, making it virtually impossible to track external access and maintain consistent data center security.

A likely scenario is that a participating partner uses different VPN software or an incompatible communications protocol. VPN connectivity is neither consistent nor universal, which poses real problems for ground-level IT workers whose job it is to make connectivity work. Many protected environments do away with modem connections entirely for improved security, but not all organizations are inclined to make such sacrifices and continue to utilize this most insecure communications medium.

---

## **Span of Control and Access Control Issues**

The span of control issues scales in size with the remote vendor and whatever complexities it brings to the table. First and foremost, remote vendor staff operates well beyond company control. On-site staff often has no idea who they're working with remotely and even a simple change in remote vendor staff creates significant accountability issues. Trust within protected environments is a fragile thing.

As an organization expands in scale and scope, the number of administrators accessing sensitive data and systems grows as well. Organizations encounter growing pains with the realization that old standby methods of account management—sealed envelopes, sticky notes, spreadsheets, and so on—are insufficient and incompatible with modern auditing requirements. Yet the need to continue granting access to protected systems remains constant, with access now reaching an external scope of outside help. This calls for more highly granular access controls, along with comprehensive controls over remote access (who gets in) and privileged session activity (who may touch which resources, and what actions they may perform).

## **A Solution for Remote Privileged Sessions**

Numerous characteristics must be present for remote privileged sessions to be properly handled. We review the key characteristics in a workable solution for managing remote privileged sessions in the list items that follow:

- *Clientless agentless implementation:* This kind of solution enables remote vendors, consultants, and employees to utilize any tools and platforms they like for network and system access, with no strict need to enforce requirements on connecting clients. Clientless application-based solutions (often written in Java) ensure better controls and proper auditing features, while the proxy architectures minimize the chances that remote malware might spread to local host platforms inside the Internet boundary.
- *Activity logging mechanisms:* Activity logging captures all mouse and keyboard activity, permits simple replay, and supports compact session recording to maintain compliance with governmental laws and federal regulatory practices. An added benefit of keystroke recording and playback is that capturing entire sessions can span applications, platforms, and network equipment.
- *Logical separation stymies malware:* Proxying connections end-to-end between company and remote systems creates a logical separation and creates no system-level connections to company hosts. Thus, there is a firewalling effect present that gives malware no chance to migrate from a remote system to any local system. By using secure HTTP (such as HTTPS) and secure shell (SSH) protocols, remote users have no opportunities to employ NetBIOS or other insecure protocols and services.
- *Strong reliable authentication for remote sessions:* Multi-factor authentication, including tokens and smart devices, strengthens authentication and limits access exclusively to authorized parties. Opting for biometric access controls facilitates accountability particularly for remote partners, service providers, and other third parties to whom companies must extend a certain degree of trust.

- 
- *High availability sustains service levels and fosters reliable session security:* High-availability permits uninterrupted access without frequent or significant company IT activity or intervention. Once privileged sessions are initiated, management systems must remain available to avoid negatively impacting the IT infrastructure where service levels aren't adequately maintained. It may involve dynamic DNS, load balancers, and round-robin strategies to maintain service levels.
  - *Hardened intermediate appliances deter attack and unwanted access:* The use of hardened network appliances as delivery vehicles protects privileged session management from various forms of attack and compromise. Following the principle of least privilege, even where a vendor may access a device directly, proxy servers and services can ensure that vendor staff members never obtain access to a device password.
  - *Specialized APIs support developer access:* Special application programming interfaces (APIs) and command-line interface (CLI) capabilities also facilitate compliant interaction for privileged sessions. Many privileged account management vendors use specialized APIs for provisioning users into an account management system that involves no development effort. In other situations, developers can access production systems under strict control and supervision that only lets them touch components and information relevant to their software, without enabling carte blanche access to its entire contents and capabilities.

## Summary

Privileged sessions and the accounts they use continue to demand close scrutiny and tight control throughout the enterprise, primarily because they can bypass ordinary IT user access and management controls. Privileged sessions are only as strong as the weakest link in a long security chain. Privileged accounts should be handled with discretion, but even that is not enough to prevent authorized users from making unauthorized changes to system configurations or operational parameters. That's why account tracking, tight access controls, and activity logging are so important to delivering workable solutions that comply with security policy and regulatory requirements.