Realtime
publishers

The Essentials Series: Managing Access to Privileged Accounts

Understanding Account Access Management

sponsored by

e·DMZ SECURITY

by Ed Tittel

Copyright Statement

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.