Realtime publishers

The Essentials Series: Managing Access to Privileged Accounts

Privileged Password Management Systems

sponsored by

e·DMZ SECURITY

by Ed Tittel

Privileged Password Management Systems....1
Types of Accounts....1
Guest or Anonymous Accounts....1
User Accounts....1
Privileged Accounts....2
Shared System Accounts....2
Service Accounts....3
Historical Approaches to Password Management....3
Modern Password Management....4
Basic Requirements and Functionality....5
Advanced Requirements and Functionality....6
Summary....7

Copyright Statement

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.

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.