Privileged Access Management

Frequently Asked Questions
# Contents

1. What are the business problems related to privileged accounts?  
2. How does a privileged access management system work?  
3. How often should passwords on privileged accounts be changed?  
4. How should access to privileged accounts be controlled?  
5. Can these systems support a "two keys to launch" scenario?  
6. What about securing passwords to Windows service accounts?  
7. How about mainframes, ERP applications and other systems?  
8. Are there alternatives to displaying passwords to administrators?  
9. Can single sign-on be included?  
10. What happens a login to the physical console of a server is needed?  
11. What about privileged accounts on laptops?  
12. Can the administrator actions be recorded?
1 What are the business problems related to privileged accounts?

It can be difficult to securely manage access to thousands of privileged accounts. Consequently, in many organizations, the passwords to privileged accounts are:

- known to many people, possibly including former staff,
- often the same on many systems,
- rarely if ever changed, and
- stored in plaintext, by people and by applications.

There are serious consequences to these password management practices, including:

- There is no accountability for use of shared, privileged accounts. This is both a security / regulatory compliance problem and a problem with diagnosing operational problems.
- Former staff may retain sensitive access.
- Attackers have an easier time compromising these dangerous accounts.
- If one system is compromised (e.g., an IT user's PC or an application server), the attacker can leverage passwords stored or typed on that system to compromise additional systems.

2 How does a privileged access management system work?

There are several technological approaches to more securely managing privileged passwords:

<table>
<thead>
<tr>
<th>Approach</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Eliminate shared passwords entirely and assign personal administrator-level accounts to each IT user, on each asset.</td>
<td>Individual accountability for configuration changes.</td>
</tr>
<tr>
<td>2</td>
<td>Create and delete personal administrator-level accounts for users on demand.</td>
<td>Individual accountability for configuration changes.</td>
</tr>
<tr>
<td>3</td>
<td>Modify operating systems and applications to check whether users are allowed to perform privileged actions, in real time. Manage access control policies centrally.</td>
<td>Fine-grained control over user access.</td>
</tr>
<tr>
<td>4</td>
<td>Use software installed on each device to periodically change local passwords. Send a copy of these passwords to a secure vault, shared by many systems.</td>
<td>Works even in complex, segmented networks.</td>
</tr>
<tr>
<td>5</td>
<td>Software on a central system periodically pushes new passwords to each device and keeps copies in a secure vault.</td>
<td>Minimal footprint on managed systems.</td>
</tr>
</tbody>
</table>
By far the most common approach to securing privileged accounts is to randomize their passwords regularly. Normally this process is initiated by a central server, to eliminate the need for change control on each managed system.

3 How often should passwords on privileged accounts be changed?

A good rule of thumb is daily. With a daily password change, if a system administrator quits, he would only have access to a few accounts (on systems where he did work on his last day) and all that access would automatically expire within 24 hours.

Longer password change intervals introduce the possibility of access retention for more time, creating a longer window of vulnerability.

Shorter password change intervals may interfere with work. For example, an administrator may need to sign into a system for several hours to make a complex change, and an hourly password change might interfere with this work.

4 How should access to privileged accounts be controlled?

There are two scenarios where people need access to privileged accounts:

1. **Routine access:** for example, Windows server administrators need frequent access to privileged accounts on Windows, Linux administrators on their servers, desktop support on user PCs, etc.

2. **One-time access:** for example, data center staff may need access to systems during an emergency in the middle of the night or programmers may need temporary access to diagnose a production problem.

Each of these scenarios calls for its own access control strategy.

For routine access, administrators should be pre-authorized to gain access whenever they need it. For them, a privileged access management system becomes a launch-pad for single sign-on to accounts they use to do their job.

For one-time access, an approvals workflow is required. Users can sign into request access to specified accounts. Other users are then invited via e-mail or SMS to review these requests. They respond by signing into a web portal and approving or rejecting the proposed access. Approved logins behave just like routine access, but only for a limited time.

In either case, user roles and account groups play a part in reducing the complexity of system setup. Users should be assigned roles, such as “Windows administrator.” Systems and privileged accounts should be collected into groups. User roles can then be assigned rights to system groups.
5 Can these systems support a "two keys to launch" scenario?

Using the one-time access workflow described above, a privileged access management system should support multiple authorizers. For example, user A might request access to privileged account P on system S. This request could be routed to multiple people to review – say users B, C and D. If any two of B, C and D approve the request, then user A will be allowed to sign into P.

6 What about securing passwords to Windows service accounts?

On the Windows operating system, service programs are run either using the SYSTEM login ID, which possesses almost every privilege on the system (and consequently can do the maximum harm) and which requires no password or using a named local or domain account. Services are run in the security context of a named account in order to reduce the privileges available to them at runtime.

When Windows services are run with a named account, the password for that account is needed to start the service process. This means that the Service Control Manager (SCM), IIS web server, Scheduler, etc. all need to know both the ID and password of service accounts, when they are configured to run jobs, services, application pools or similar contexts as a named account, rather than as SYSTEM.

Service account passwords differ from administrator passwords in that they appear in at least two places:

1. Hashed, in the security database – e.g., the local SAM database or Active Directory, just like any other account.
2. Reversibly encrypted or plaintext, in the registry or a configuration file, where the program that starts the service (e.g., Service Control Manager, Scheduler, IIS, ...) can retrieve the password value when starting a new service process.

Some Windows components, notably IIS, are able to periodically change the passwords of local service accounts they use. Unfortunately, this capability does not extend to domain service accounts, used to run services on multiple systems and does not apply to all types of service accounts. This means that many Windows service account passwords remain static by default.

A privileged access management system should be able to change service account passwords:

1. Automatically discover that an account is used to run services on one or more computers.
2. Randomize passwords on service accounts.
3. Notify the service control manager, scheduler, IIS, etc. of new password values.
4. Queue and retry notification in the event of communication problems.
7 How about mainframes, ERP applications and other systems?

Some privileged access management systems include a rich set of connectors and can manage passwords across the enterprise, rather than just on Windows servers and via scripted SSH sessions.

It is helpful to deploy a system that can handle the majority of systems in an organization, rather than having to use different applications for each platform.

8 Are there alternatives to displaying passwords to administrators?

Displaying passwords from the vault should only be available as a last resort. In most cases, where connectivity is available to the system in question, one of the following mechanisms should be used instead:

1. The privileged access management system can launch login sessions using Terminal Services (RDP), SSH (PuTTY), VMWare vSphere, SQL Studio, etc. and inject passwords from the vault, providing the IT user single sign-on and eliminating the need to display plaintext passwords.

2. A copy of the password could be placed in the user's copy buffer, so that the user can paste it into a login screen. The copy can also be automatically removed from the copy buffer after a minute or two.

3. The authorized user's personal (and normally unprivileged) Active Directory account can be temporarily attached to security groups on Active Directory or on domain-member computers.

4. The authorized user's public SSH key can be temporarily appended to the .ssh/authorized_keys file of a privileged account on Unix or Linux.

9 Can single sign-on be included?

Yes, as described above, this can be done by launching RDP, SSH or similar sessions; by temporarily adding a user's AD account to security groups or by temporarily creating SSH trust relationships.

10 What happens a login to the physical console of a server is needed?

Password display is needed where a login to a system's console is required. This access disclosure mechanism should be handled via the one-time access disclosure workflow, rather than in the context of routine access.
11 What about privileged accounts on laptops?

A password management system can easily make connections to servers, which have fixed network addresses, are always on and are continuously connected to the network. It is much harder for a central password management server to connect to mobile laptops, for several reasons:

- Laptops frequently move from site to site.
- Even when they remain in one place, laptop IP addresses may change dynamically, due to use of DHCP.
- Laptops are often turned off and do not respond to network inquiries when deactivated.
- Laptops may be unplugged from the network, either to move them or for periods of disuse.
- Laptops may be protected by a firewall that blocks network connections inbound to the PC.

In short, while it is easy for laptops to contact a central server, it is nearly impossible for the reverse to happen reliably.

To secure privileged accounts on laptops, a privileged access management system must include client-side code, which initiates password changes from the laptop, rather than from the central server.

This architecture supports:

- Laptops that are periodically disconnected or powered down.
- Laptops behind firewalls or with un-routable IP addresses (NAT).
- Laptops with dynamic IP addresses.

12 Can the administrator actions be recorded?

Modern privileged access management systems support session recording. This technology is used to record login sessions made by administrators to privileged accounts and later search and playback of these sessions.

The approaches used to accomplish this vary widely:

1. Instrumentation installed in advance on the user’s PC.
2. Instrumentation on the user’s PC implemented dynamically via ActiveX.
3. Instrumentation on managed servers.
4. A proxy server which intercepts, records and analyzes connection protocols.