Modern PAM Buyer's Guide

2022

Teleport

Modern PAM Buyer's Guide

Privileged Access Management (PAM) is a go-to solution to prevent privilege misuse and insider threats, and limit malware propagation. After all, properly protecting and monitoring the keys to the kingdom is always a good practice. Privileged Access Management has been even more critical in recent times. With the advent of the cloud where infrastructure is provisioned with a single API call and authenticated with a single API key, the risk of someone misusing these credentials is far higher. Now adversaries can edit or even delete your entire infrastructure with a single API call. While the concept and strategy of Privileged Access Management remain the same, traditional PAM solutions struggle to adapt to current DevOps operations and, in many cases, disturb engineers' workflow, slowing them down.

This Modern PAM Buyer's Guide details the questions to ask vendors when evaluating a PAM solution for modern, cloud-native applications. It is essential that a Modern PAM solution combine strong functionality for managing privileged accounts while enabling ease of use for developers and maintainability for ops teams.

Table of Contents

Shortcomings of traditional PAM solutions for modern, cloud-native applications

Before we discuss what is needed in a modern PAM solution for cloud-native applications, let's look at why traditional PAM solutions struggle to meet the privileged access management needs of modern infrastructure. Traditional PAM solutions were built at a time when infrastructure was much more static, with fleets of Windows Servers, Remote Desktops, and SSH via Putty, hosted in co-located or on-premises data centers. Additionally, the administrative workflows to manage this infrastructure were primarily GUI-based. A PAM user who needs to access a server will log into the PAM solution, retrieve their credential, then use that credential to log into the server — all manually.

This workflow is no longer useful with modern cloud operations and DevOps practices where most interactions with infrastructure assets are based on API automation and CLI tools, not GUIs. As we will discuss below, the shortcoming of traditional PAM systems is their inability to adapt to current DevOps and cloud-native workflows.

Requirements for a modern PAM solution

Buyers looking for a PAM solution for modern apps need to consider three broad categories of functionality: features, ease of use for modern workflows, and maintainability of the PAM solution itself. This section highlights what to look for in each of these categories.

FEATURES AND FUNCTIONALITY

Essential PAM functionality is well-understood by many buyers. Here is a brief overview of some of the most important features to look for in a PAM solution:

1.

Does the PAM solution enable zero standing privileges?

In order to minimize risk of privilege abuse and stolen credentials, your PAM solution should be able to define roles that do not have any privileges by default, a concept known as zero standing privileges.

2.

Does the PAM solution enable just-in-time privilege escalation?

Because engineers need access to systems to do their jobs and the concept of zero standing privileges by definition prevents access by default, your PAM solution should enable just-in-time (JIT) privilege escalation, wherein an engineer can request access to a system, and be granted access quickly via an automated approval workload.

3.

Does the PAM solution provide robust audit capabilities for multiple types of systems?

Security and compliance dictate not only that you control who can access which system, but that you can demonstrate the controls that you are using. It is essential that your PAM solution support robust audit capabilities across all supported protocols. For instance, it must be able to log sys calls for Linux, queries run for MongoDB, and kubectl exec commands for Kubernetes.

4.

Does the PAM solution support session recordings?

Session recordings are a special case of audit. While audit can be defined as simply logging behavior, session recordings take that one step further and provide a complete view of what happened during a privileged session, making review easier. Session recordings can also be used for training purposes.

5.

Does the PAM solution support dual authorization of privilege escalation requests?

Dual authorization is the concept that some privilege actions must be approved by more than one person. In line with zero standing privileges and just-in-time access, your PAM should support workflows that enable dual authorization.

6.

Does the PAM solution support per-session MFA?

Your PAM solution should be able to enforce multi-factor authentication for privileged sessions as needed. This capability enables access to some system access with a single authentication, while requiring a MFA for more sensitive systems.

7.

Does the PAM solution support moderated sessions?

Dual authorization requires more than one approver before a privileged session is started. Moderated sessions require that more than one person is connected to a shared session when accessing sensitive systems. Your PAM solution should support both.

8.

Does the PAM solution support Linux and Windows?

While a majority of modern applications run on Linux, many enterprise applications still run on Windows. Your PAM solution should provide management for privileged users on both platforms with a single solution.

9.

Does the PAM solution provide access controls for modern applications running on Kubernetes?

A modern PAM solution must support not only multiple operating systems but also cloud-native applications managed via Kubernetes. Because containers provide isolated runtime environments separate from the Linux kernel, relying on host OS restrictions for containerized applications is insufficient.

10.

Does the PAM solution provide native support for SQL and NoSQL databases?

In the past it was common for enterprise applications to be supported by one, large Oracle or Microsoft SQL Server database. However, with the advent of microservices, a single end-user facing application might be supported by many different SQL and NoSQL databases such as MySQL, PostgreSQL, MongoDB, Redis and CockroachDB. It is essential that your PAM solution provides native support for a range of databases used in your organization.

EASE OF USE

While the features and functionality of PAM solutions are fairly well-known, it is less well-understood how ease of use for modern developer workflows impacts the effectiveness of the solution. This section looks at critical PAM capabilities from an ease of use perspective.

1. Does the PAM solution require developers to log into the PAM for each new connection?

Accessing the infrastructure necessary to run modern applications requires numerous developer and administrative tools that speak different protocols (e.g. HTTP, SSH, RDP, MongoDB, Postgres, MySQL, Jenkins, GitLab). Most traditional PAM solutions have a loose integration with these tools and protocols and require engineers to separately authenticate to the PAM for each new connection in order to access their resource. Switching tools and context like this slows developers down dramatically because it creates friction.

Let's look at one simple example: SSHing into a remote server. With a traditional PAM solution, the engineer needs to authenticate with the PAM portal each time they need to retrieve a key to authenticate with SSH. This sounds like a small ask for a security benefit. But if we consider the number of servers and the frequency of remote access for daily administrative tasks, this workflow dramatically slows engineers down because it gets engineers out of "the zone."

Now, consider the same inefficient workflow for every infrastructure resource an engineer needs access to: a database, a Kubernetes cluster, a monitoring dashboard, a CI/CD environment, a version control system. Each time an engineer needs to access infrastructure, they first need to go to the PAM tool to check out their credentials, and then go to the resource to continue their work. This is a waste of time and when a workflow slows an engineer down, they find workarounds, which in many cases negate the purpose of the PAM in the first place. For example, it is common for engineers using traditional PAM tools to check a private key out of the PAM vault one time, and keep it on a local notes app for easy reuse without logging into the vault each time. While this may be prohibited for certain levels of access by the PAM, with supply chain attacks on the rise, a vulnerability at any point in the system can have far-reaching implications.

For maximum security and efficiency, what you want is a workflow wherein the engineer goes directly to the resource they want to access. Once they are at the resource, they are then challenged for their identity via integration with an IDP provider such as Active Directory or Okta. And once they have authenticated one time, they don't have to do it again until the time-to-live (TTL) on the request expires. This is dramatically simpler, more efficient, and discourages insecure workarounds.

2. Does the PAM integrate with the native tools engineers love?

Developers gain muscle memory through years of working with preferred tools. When you make them use a different tool to complete the same task, they have to relearn commands that have become second nature, which results in a productivity loss. Unfortunately, this is quite common with traditional PAM portals that are generally limited on features compared to native tooling.

For example, in order to manage privileged access for certain resources, like databases, PAM solutions encourage access via their GUI, instead of via the native database client. This effort to control access hampers the database administrative workflow because the database clients used by PAM providers are not as feature-rich as the database client built by the database or developer tool companies.

The best security solutions are those that bring security to the user's existing workflow. It's hard to change users' habits, and the user will always find a way to trick and bypass the security system. So deep support at the protocol level that allows engineers to use the tools they are already using, with a little or no modification, is the best way to increase engineers' productivity and security.

3. Does your PAM solution support just-in-time access via modern ChatOps?

Modern engineering teams use ChatOps platforms such as Slack, Microsoft teams, and Mattermost for communication. With the increase in distributed workforces and remote work, these platforms have become a crucial piece of technology for team collaboration across the entire organization. Since most engineers already use these platforms for day-to-day communication and receiving event alerts, PAM workflows such as just-in-time (JIT) access requests and approval are more efficient if integrated with ChatOps platforms. This extends to integrations with the rest of the DevOps tool chain. Can filing a Jira ticket kick off a privilege escalation request? Can that privilege escalation request be automatically approved if an SRE is on-call in PagerDuty?

A modern PAM has integration points across a range of communication tools to streamline developer workflows without sacrificing security.

4. Does the PAM solution promote best practices for passwordless authentication?

Most of the PAM solutions on the market are built around a core feature: the password manager. Almost all the security controls that protect privileged access (e.g. privilege escalation, just-in-time access) are implemented during the password retrieval process. Not only are passwords risky, but they also create operational overhead for engineers. Maintaining password vaults and implementing timely password and key rotation take a significant amount of engineering time.

While the whole security industry is going passwordless due to security concerns, it makes less sense to go for password management solutions that are both risky and create additional operational overhead. Instead, you should expect that privileged access management capabilities are delivered in a passwordless manner.

MAINTAINABILITY

While it is important to understand the features & functionality, and ease of use capabilities of your PAM solution, it is also critical to take into consideration the maintainability of the PAM solution itself. If running the PAM introduces a maintenance burden on your operations team, then it will increase the total cost of ownership of the solution. Here are some key considerations for deploying and maintaining your PAM solution.

1.

Does the PAM solution support applications running in the cloud as well as on-prem with a single integrated solution?

As we discussed in the introduction, traditional PAM solutions were built at a time when infrastructure was much more static, with fleets of Windows Servers, Remote Desktops, and SSH via Putty, hosted in a co-located or on-premises data center. While modern applications often run in the cloud, or on-prem in dynamic cloud-like environments, many traditional applications still exist. Enterprises need a PAM solution that encourages modern security best practices like passwordless access, but for all their applications regardless of environment.

2.

Can the PAM solution itself be managed in a cloud-native way using solutions such as containers and Kubernetes?

A modern PAM solution must be easy to install and manage. That means that your PAM solution should itself be able to be deployed as a container and operated inside a Kubernetes cluster like any other modern application.

Traditional PAM solutions that require heavyweight Windows Servers and that cannot be installed and run in a cloud-native way put an additional burden on operations teams who must maintain traditional server administration practices for the PAM solution, while using modern DevOps practices for applications. If the engineering culture of your organization is to go all-in with cloud-native infrastructure, this is a step backward.

3.

Does the PAM support infrastructure-as-code management patterns?

Infrastructure-as-code, policy-as-code, detection-as-code — these new methods to manage infrastructure remove manual intervention that slows DevOps teams down and reduces security.

Being able to configure and maintain infrastructure using a programming language drastically increases the speed of administrative workflows and allows teams to automate security, reducing opportunity for human error.

Furthermore, change management is an essential part of infrastructure management. Configuration-as-code allows maintaining deployment state in a version-controlled system which means increased visibility, automated testing, and the ability to easily roll back changes if needed.

Unfortunately, most traditional PAM solutions do not support these modern practices, limiting their use with teams who have embraced the modern infrastructure management ethos.

4.

Does the PAM adapt to dynamic auto-scaling infrastructure?

Modern cloud and private cloud environments are highly dynamic. Servers, containers, and Kubernetes clusters are spun up and spun down. Yet while these services are running, privileged access and audit need to be maintained. Does the PAM solution support dynamic auto-scaling infrastructure such that you do not need to manually register resources in order for the service to be protected?

Teleport

Modernizing your PAM with Teleport

Traditional PAM solutions were developed in a corporate IT management era where workflows were very different from today's fast-paced, code-driven IT operations. While Traditional PAM solutions are beginning to deliver features to support privilege management in cloud infrastructure (albeit sometimes with an add-on tool), they struggle to adapt to distributed team workflows and integrate very loosely with modern DevOps tools and cloud-native infrastructure.

At Teleport, we believe that security solutions should adapt to users' workflow. Forcing engineers to change their workflow slows them down, costing the organizations they work for money. Further, when security controls break engineers' workflow, they often try to trick and bypass the system, posing more risk to the organization. Finally, the inability of traditional PAM solutions to adapt to modern, dynamic, cloud-native infrastructure and workflows means your engineers will spend administrative time maintaining the traditional PAM solutions instead of using them for secure infrastructure access.

Teleport is a cloud-native PAM solution that unifies access for Linux and Windows servers, Kubernetes clusters, SQL and NoSQL databases, and DevOps applications like CI/CD, version control, and monitoring dashboards. Teleport eliminates the risks of passwords in infrastructure by allowing certificate-based remote access. In addition, it supports configuration via code, integrates well with Kubernetes and cloud-native infrastructure, and supports zero standing privileges and JIT access workflows from modern ChatOps platforms, making it versatile for modern infrastructure operations.

You need a cloud-native strategy and tools to succeed in cloud-native operations.
Try Teleport today or contact us for a demo.

Appendix

Buyer's Guide for Modern PAM: PAM solution comparison checklist
Use this Template to evaluate multiple PAM vendors across categories of features & functionality, ease of use, and maintainability. Add your own requirements as needed. A version of this template is available in Google Sheets here.

CATEGORY: FEATURES

Does the PAM solution enable zero standing privileges?		
Teleport	**Vendor 2**	**Vendor 3**
Yes, any Teleport role can be configured without any preassigned privileges, and the required privileges can be assigned in an ad hoc fashion just for the session via Teleport JIT integration with preferred ticketing systems such as Jira or Service Now.		
Does the PAM solution enable just-in-time privilege escalation?		
Yes. Teleport access requests can be used to implement just-in-time infrastructure access by allowing users to request to have their access temporarily elevated. After their request is approved, the user's short-lived certificates are updated to their temporary role until the certificate time to live (TTL) is exceeded. The Access Request API makes it easy to dynamically approve or deny these requests.		
Does the PAM solution provide robust audit capabilities for multiple types of systems?		
Teleport provides audit logs for all support protocols (SSH, RDP, Kubernetes, SQL & NoSQL Databases and private applications). Teleport streams security events to a centralized destination of choice such as SIEM solutions to help integrate access events into a single source of truth.		

Does the PAM solution support session recordings?		
Teleport	**Vendor 2**	**Vendor 3**
Teleport maintains a list of live sessions across all protocols and environments, providing an up-to-date picture of what is happening. Additionally, interactive sessions across the entire infrastructure are recorded and stored in a storage solution of choice. Session recording can be useful for forensic or educational purposes. Recordings can even be copy/pasted from session recordings since they play as a video, but are actually in json.		
Does the PAM solution support dual authorization of privilege escalation requests?		
Yes, Teleport can require the approval of multiple team members to perform some critical actions. This allows to satisfy FedRAMP AC-3 dual authorization control that requires approval of two authorized individuals.		
Does the PAM solution support per-session MFA?		
Yes, Teleport supports multi-factor authentication on a session-by-session basis. This is an advanced security feature that protects users against compromises of their on-disk Teleport certificates.		
Does the PAM solution support moderated sessions?		
Yes, Moderated Sessions allows Teleport administrators to define requirements for other users to be present in a session. Depending on the requirements, these users can observe the session in real time, participate in the session, and terminate the session at will.		

Does the PAM solution support applications and infrastructure running in the cloud and on-premises?		
Teleport	**Vendor 2**	**Vendor 3**
Yes, the same Teleport instance can be used to access any supported infrastructure component, regardless of whether it is running in the cloud, on-premises or at the edge.		
Does the PAM solution support Linux and Windows?		
Yes, Teleport supports access to Linux and Windows hosts.		
Does the PAM solution provide access controls for modern applications running on Kubernetes?		
Yes, Teleport supports secure access for Kubernetes clusters.		
Does the PAM solution provide native support for SQL and NoSQL databases?		
Yes, Teleport supports a wide variety of SQL and NoSQL databases including MySQL, PostgreSQL, Microsoft SQL, MongoDB, CockroachDB, Redis, and Cloud-hosted databases such as Amazon RDS, Aurora and Redshift.		

CATEGORY: EASE OF USE

Does the PAM solution require developers to log into the PAM for each new connection?		
No. With Teleport, an engineer authenticates one time via their SSO provider and receives a short-lived certificate that provides instant access to all infrastructure resources that they are entitled to. There are no keys to check out, and no hopping back and forth between the PAM and servers, databases or internal applications.		

Does the PAM integrate with the native tools engineers love?		
Teleport	**Vendor 2**	**Vendor 3**
Yes, Teleport integrates with OpenSSH, kubectl, and native database clients such as pgsql. As such, developers get a native experience interacting with their favorite tool, but everything is logged and auditable behind the scenes, improving both productivity as well as security.		
Does your PAM solution support just-in-time access via modern ChatOps?		
Yes, Teleport allows users to request elevated privileges in the middle of their command-line sessions. Requests can be approved or denied via ChatOps in Slack & PagerDuty or anywhere else via flexible Authorization Workflow API.		
Does the PAM solution promote best practices for passwordless authentication?		
Yes, all access to protected applications and services are based on certificates. Further, Teleport supports modern authentication standards such as WebAuthn as a second authentication factor. WebAuthn support includes hardware devices, such as YubiKeys or SoloKeys (tsh and Web UI), as well as biometric authenticators like Touch ID and Windows Hello (Web UI only).		

CATEGORY: MAINTAINABILITY

Does the PAM solution support applications running in the cloud as well as on-prem with a single integrated solution?

Teleport	Vendor 2	Vendor 3
Yes, the same Teleport instance can be used to manage access to infrastructure, regardless of whether it is in the cloud, on-premises or at the edge.		

Can the PAM solution itself be managed in a cloud-native way using solutions such as containers and Kubernetes?

Yes, Teleport can be installed and managed in many ways, including as a container, using a Helm chart and managed via Kubernetes.		

Does the PAM support infrastructure-as-code management patterns?

Yes, Teleport's internal configurations and policies can all be managed with a YAML file that can be maintained in a version control system like Git. Further, all the administrative functions can be managed and automated via API or a built-in CLI client.		

Does the PAM adapt to dynamic auto-scaling infrastructure?

Yes, Teleport supports auto-joining and auto-discovery capabilities so that access can be provided to resources as they scale up and scale down without manual intervention.		