SOLUTION BRIEF

15 Server Privilege Management Use Cases for Unix & Linux

This guide has been prepared so that IT and security administrators can quickly understand how BeyondTrust Endpoint Privilege Management for Unix & Linux and BeyondTrust Active Directory Bridge can help you progressively mature your privileged access security controls and eliminate sudo, while centralizing administration, auditing, and reporting.

Table of Contents

Executive Summary

The concept of least privilege states that all users should have the lowest level of access privileges required to effectively conduct their jobs. However, many basic operating system, management, application, and software functions (e.g. configuration utilities) require more than just basic privileges. Consequently, end users traditionally need to possess elevated privileges in the form of root or administrative usernames and passwords. To overcome this inherent security risk of excessive privileges, organizations must remove the need to distribute and maintain root and administrative credentials – or even reveal these credentials to end users at all – without impacting user or administrator productivity.

The BeyondTrust Endpoint Privilege Management solution enables IT organizations to exert granular control over who can access Unix, Linux, and Windows servers and what those users can do with that access. BeyondTrust enables organizations to improve server system security, while simplifying privileged access management deployments and ensuring productivity.

BeyondTrust Endpoint Privilege Management includes capabilities for:

- Discovering, managing, and monitoring access to root, admin, and other privileged passwords and SSH keys automatically
- Elevating commands and delegating privileges with fine-grained policy controls, providing robust control over what Windows, Unix, and Linux users can do once they are logged on to a system
- Enabling single sign-on (SSO) and simplified policy by consolidating accounts across multiple platforms into a single set of access control policies
- Providing risk visibility into applications targeted for privilege elevation
- Analyzing, logging, recording, and reporting on privileged password, user, and account behaviour

This white paper explains common use cases for privilege management on Unix/Linux servers, including capabilities available in Privilege Management for Unix & Linux and BeyondTrust Active Directory Bridge (AD Bridge) which comprise part of the BeyondTrust Endpoint Privilege Management solution. These use cases are progressive, with the most straight- forward use cases listed first, followed by those of increasing maturity level.

15 Common Server Privilege Management Use Cases

1. Removing the Need to Login as Root

Many system and application users of Unix and Linux use the phrase, “I need root,” indicating that they can only perform their daily job functions if they can logon using the most powerful user on the system, “root”. Root is often referred to as the “God” user as there is practically nothing that the root user cannot do.

Allowing the usage of the root account removes the ability to audit an individual’s actions (essentially promoting account sharing) and inhibits the use of a strong, changeable password for the root account due to the need for multiple persons to use the account and leverage its privileges at any given time. Use of the root account, and practices such as account sharing, dramatically increase organizational risk from insider threats via malicious and accidental behaviors as well as external threats.

Privilege Management for Unix & Linux implements a true least privilege delegation model that can eliminate, or highly restrict, use of the root account—without hampering user workflows. With the BeyondTrust solution, you can allow users to run any command at a higher privilege level as dictated by your centralized policy. Removing the need for users to logon as root allows the root user account to have much tighter security controls or be moved to a password management system such as BeyondTrust Password Safe.

2. Achieving Compliance for the Root Account

The most senior admins may, from time to time, have a legitimate need to use the root account due to the types of system-level changes being made, or simply because of the ad-hoc nature of the commands they may need to issue.

However, the compliance team needs to monitor ALL activity and ensure accountability for actions. To allow system administrators access to such a highly-privileged account, the compliance teams need the right tools and processes in place to be able to identify who was using the root account, when they were using the root account, and what changes were made by that user with the root account. In addition, it is imperative that log files are immune to any sort of tampering.

Privilege Management for Unix & Linux allows standard named user accounts to elevate to a root level with full session logging, providing a centralized, indelible audit trail, and transparent accountability for each individual system administrator.

3. When Sudo Doesn't Cut It Anymore

Sudo has existed for a long time, but as the number of systems and users has grown, management of sudo has become inordinately time-consuming. Coupled with limitations of the controls available in sudo, systems now seem perilously exposed to an increasing number of internal and external security threats.

Privilege Management for Unix & Linux provides a far more flexible policy language, allowing for infinitely more granular policies to be created at both the command and system level. Privilege Management for Unix & Linux increases security in several ways, including moving the policy and log data off the users' workstation or server, and utilizing the latest encryption technology for data both in transit and at rest.

4. Controlling File System Permissions and Access

File systems allow for the setting of individual file and folder permissions such as read, write, execute, etc. However, file and folder permissions are very static in nature and need to be set on each host. There are also risks when allowing highly privileged accounts (such as root) access to a server or workstation, as an account with that type of privilege can easily manipulate permissions and ownership settings using commands such as "chmod" and "chown". Giving an account unlimited access to files and data represents a significant risk.

Privilege Management for Unix & Linux solves this challenge by allowing the application of centrally controlled, runtime security settings that can secure valuable information from even the most highly privileged accounts. Privilege Management for Unix & Linux's Advanced Control and Audit (ACA) policies can be dynamically applied, allowing for the enforcement of file system permission rules during certain times, or allowing access only if certain conditions are met, such as the input of a two-factor authentication or by supplying a valid ticketing system incident number.

With this capability, organizations can grant a full 'root' level shell, but at the same time, block that root user from being able to access any user files stored in '/home', or by allowing scripts to be executed, but not viewed or edited.

5. Controlling Script Access and Auditing Script Actions

Unix and Linux administrators rely heavily on the use of scripts to perform daily system administration duties. Many of these scripts have to run as a privileged user, such as 'root', or call functions that in turn require high levels of privilege. Using a least privilege solution is essential when handing out rights to users that need these elevated rights. Traditionally, the best way to audit these users' activities was to enable some form of session recording, i.e. record stdin and stdout, or log everything the user types and everything the user sees on the screen. The problem with this approach is that when recording these actions, logging the fact that someone typed 'update.sh' on the command line provides no way to see what actions were performed within the script, what data may have been affected, or how the system may have been manipulated.

By using Privilege Management for Unix & Linux and its Advanced Control and Audit (ACA) feature, not only can scripts be more tightly controlled (i.e. scripts can be executed and viewed, but not modified—even by root), but organizations can also log and record all the system level calls that are made, regardless of their source, during the session.

This capability allows for actions, such as the execution of a script, to be logged along with every call that the script makes (i.e. reads, writes, executes, etc.). In addition to logging all the script activities, Privilege Management for Unix & Linux also has the capability to control individual actions within the script and selectively allow or block those actions without stopping the overall processing of the script.

6. Controlling Access Down to the System Level

Traditional least privilege tools, such as sudo, focus on controlling what the user types on the command line rather than what the system actually tries to process. The rules defined in such tools focus on what the user is typing and not what the system is actually going to do. In most modern operating systems, however, these two actions can be very different. For example, on Windows, it is simple to create a shortcut to an item, like 'notepad.exe', and name that shortcut 'My Fav Text Editor.lnk'. If an organization has a policy that stops someone from clicking on an icon named Notepad.exe, what is going to happen when someone double clicks on the new shortcut just created?

The same type of scenario plays out in the Unix and Linux world, only being a command line operating system, security policies focus heavily on what the user types at the command prompt. The problem is that shortcuts (known as symbolic links and hard links) can also be used to call binaries (executables) and scripts, or reference files. In addition, binaries (executables) can also be renamed to circumvent security policies. Imagine if someone copied the 'shutdown' binary and named it 'ping'. Shutdown may be blocked by sudo, but ping may be just fine. A sneaky admin runs 'sudo ping' and the system is going to power down without anyone able to figure out what just happened.

Privilege Management for Unix & Linux Advanced Control and Audit (ACA) policy rules enable one-line controls to be placed around files and folders, allowing for the target of individual files, files, and folders that match a naming template or generically apply to entire folders and all the subfolders. These controls allow for granular permissions to be set against the target, but more importantly, ACA operates at the system level and not the command line level. Regardless of how the 'shutdown' command is called, whether it be directly '/usr/sbin/shutdown', a hard or soft link to '/usr/sbin/shutdown' or a renamed copy of the shutdown command, ACA is intelligent enough to block the execution of the issued command.

7. Consolidating Event Logs, Reporting, and Analytics

By their command line nature, Unix and Linux systems don't lend themselves to easily- consumed reporting. However, reporting is essential – especially when conducting forensic investigations on logs, or detecting anomalies.

BeyondTrust solves this problem with integration between Privilege Management for Unix & Linux and the central platform, BeyondInsight. Privilege Management for Unix & Linux sends event log data to BeyondInsight for presentation on the dashboard; specifically, the who, what, where, and when of events. This data can then be presented in easy-to-consume reports, with access to the data cube for building custom reporting. This data can be correlated with other data in the Auditor feature of BeyondInsight for anomaly detection and reporting.

This integration makes reporting on Unix and Linux events simpler, faster, and easier to read so stakeholders can have a clearer picture of their privilege risks.

8. Forensics – Time is of the Essence

Logging all Unix/Linux user activity can quickly become overwhelming. When a forensic investigation needs to be performed, organizations can waste time and manpower performing investigations.

With Privilege Management for Unix & Linux, event logs can be dynamically named, centrally located, and access controlled in the BeyondInsight central console. Privilege Management for Unix & Linux utilizes SOLR to index all recorded sessions, with all information accessible via command line or REST API.

9. Session Recording (Everything Typed, Everything Seen)

Least privilege is an ideal state for most security groups, but sometimes you just need to turn over a privileged shell, such as a root level shell. Strict auditing is an effective way to keep honest people honest. So, for trusted admins, a full root shell often presents no issue as long as the admin's activity is recorded in a tamperproof way. This level of oversight helps ensure that admins don't abuse their rights, while also helping an organization meet compliance requirements around privileged access.

One simple line in Privilege Management for Unix & Linux policy turns on full session recording, which is then dynamically named and automatically indexed using SOLR. This capability enables organizations to view the session in many different ways: by interactive playback, video style playback, view session transcript, view the command history, or in a searchable index. This capability provides flexibility to quickly turn on and search use activity, helping you to better manage risk.

10. Achieving Enterprise Password Management and Least Privilege

For a password storage solution to operate, a second level account (functional account) is required for the password safe to be able to drive password changes should the managed account password become out of sync or unknown to the safe. This is also true for any application that needs higher privileges, such as scanning tools, automated remote administration, etc. These accounts pose a risk on the target system due to their required privilege level of 'user password manager'.

By installing Privilege Management for Unix & Linux with BeyondTrust Password Safe and adding a simple policy, any user account – even one with extremely limited rights on the target system – can be granted the privileges needed by Password Safe to function. This ensures increased security and tighter compliance by not having accounts with privileges typically required for functional accounts to operate, while lowering the attack surface of the given host.

11. Integrating Privileged Password Management with Command Execution

Sometimes, administrators need to perform an action on a target host that requires the use of an account/password, which may not reside on the host itself. Running a command via Privilege Management for Unix & Linux, i.e. pbrun sqlplus, Privilege Management for Unix & Linux can retrieve credentials from a password storage solution for use in the execution of the requested command. This integration ensures that organizations can not only control their privileged credentials, but also enable very granular command control on target systems once the credential is retrieved.

12. Centrally Managing Sudo Policy Controls

For even the most mature users of an enterprise least privilege tool, there will almost always be some machines that still have access to sudo. With so many different iterations of the sudoers policy file used by the various groups inside the organization, the need to control and track changes made to each sudoers file quickly becomes an unmanageable task. Almost every user of sudo soon runs into the challenge of appropriately managing the individual sudoers policy files that get created on each Unix or Linux host.

Manual synchronization tools and homegrown LDAP/database solutions may, at first, appear appealing, but reliability, complexity, and security controls end up dramatically reducing the effectiveness of such configurations, often causing more problems than they solve. There is no effective way to undo changes to one or more sudoers files, or jump back to a certain point of time/version of a sudoers policy file.

Privilege Management for Unix & Linux provides a way to rapidly centralize one or more sudoers files, enabling change management and version control. Policy changes can be validated before making changes to the policy file live, or quickly compared with highlighted differences between any two versions of a sudoers file. The roll-back/roll-forward functionality allows for fast switching between any two saved versions of the sudoers file that are being managed by PMUL for Sudo. Connecting hosts can be optionally grouped or run in a hybrid of one-to-one plus grouped hosts. This allows simple and controlled access to specific sudoers files located on one or more centralized servers, based on the requesting host’s group membership.

13. Centrally Storing Sudo Audit Data

Using sudo means placing log data in different locations on different systems. Depending on the version of Unix or Linux, the event data ends up in different syslog files and is also mixed in with other system event data. A better way to securely move and store the sudo log data and sudo session log data is clearly needed.

Privilege Management for Unix & Linux provides a secure network connection to a centralized server that stores event and session data (encrypted if desired) to a centralized and secure location as the log data is written, removing the ability of submitting users to tamper with the logs and allowing for much faster log review and forensics when required.

14. Consolidating Accounts and Directories

When users and administrators require access to a system, a user account needs to be created on each host to provide system access for the user. Rights for these user accounts are often bloated, and clean-up of accounts, along with their associated rights, often goes unchecked when an employee changes roles or leaves an organization.Organizations need a way reduce the number of accounts being created, control which server those consolidated accounts can logon to, and exert control over what rights that user has after they have been authenticated.

BeyondTrust Active Directory Bridge (AD Bridge), coupled with Privilege Management for Unix & Linux, provides account consolidation along with consolidated system access rights and a least privilege system to control user rights, post-logon. A user's Active Directory group membership can control what servers that user can access, and a centralized policy will tightly control and audit a user's activities during each authenticated session.

As an additional benefit, AD Bridge allows users to log onto Unix, Linux, or Mac systems using their Active Directory (AD) usernames and passwords, without requiring additional infrastructure or password synchronization.

With this integrated solution, you can facilitate migration from multiple authentication mechanisms, identities, and directories to a single Active Directory-based infrastructure for all systems and users. This centralizes control and speeds user onboarding and off-boarding.

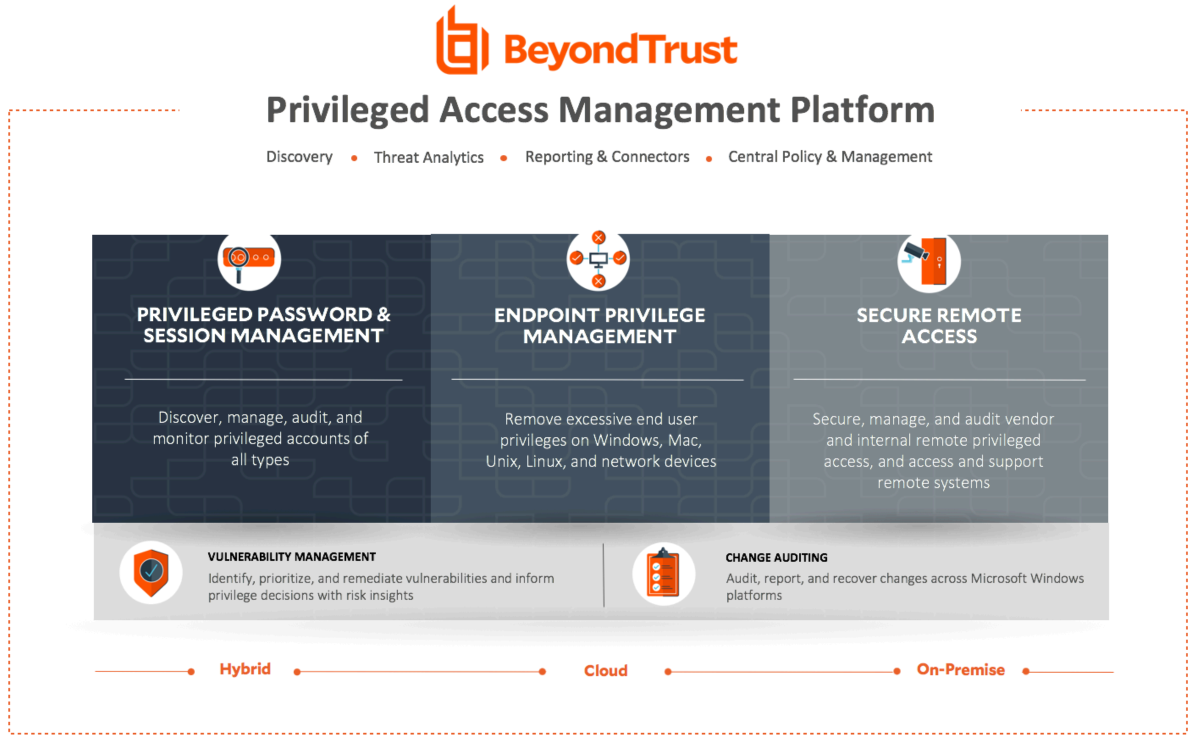
15. Extending Group Policy to Non-Windows Systems

Some organizations choose to centrally manage their Windows, Unix, and Linux systems for security, compliance, and efficiency benefits, but can struggle with ensuring configuration policies are consistently applied enterprise-wise.

BeyondTrust Active Directory Bridge (AD Bridge), coupled with Privilege Management for Unix & Linux, enables consistent configuration enterprise-wide by extending native Group Policy management tools to include specific Group Policy settings for Unix, Linux, and Mac. This capability supports compliance with SOX, PCI, HIPAA, and other regulations across all systems by replacing NIS with an Active Directory infrastructure.

The BeyondTrust Privileged Access Management Platform

Privilege Management for Unix & Linux and Active Directory Bridge (AD Bridge) are part of BeyondTrust's solutions for Endpoint Privilege Management and integrate with other privileged access management solutions in the BeyondTrust Privileged Access Management Platform. The platform is an integrated solution to provide control and visibility over all privileged accounts and users.


BeyondTrust
Privileged Access Management Platform
Discovery • Threat Analytics • Reporting & Connectors • Central Policy & Management
PRIVILEGED PASSWORD & SESSION MANAGEMENT
Discover, manage, audit, and monitor privileged accounts of all types
ENDPOINT PRIVILEGE MANAGEMENT
Remove excessive end user privileges on Windows, Mac, Unix, Linux, and network devices
SECURE REMOTE ACCESS
Secure, manage, and audit vendor and internal remote privileged access, and access and support remote systems
VULNERABILITY MANAGEMENT
Identify, prioritize, and remediate vulnerabilities and inform privilege decisions with risk insights
CHANGE AUDITING
Audit, report, and recover changes across Microsoft Windows platforms
Hybrid
Cloud
On-Premise

Next Steps

This document presented common use cases for fine-grained, policy-based privilege delegation and command elevation for Unix and Linux servers. For more on how BeyondTrust can help, visit www.beyondtrust.com/endpoint-privilege-management.