# Best Practices: Enforcing Least Privilege Access for Linux Servers With Okta

okta

# Index

# Introduction

In their 2019 Data Breach Investigation Report, Verizon revealed an alarming increase in 2018 cybersecurity attacks to over 375,000 incidents – almost 65% were server attacks. Today, information security (IS) and information technology (IT) professionals must guard against a seemingly endless variety of attack vectors while balancing on a narrow Identity Access Management (IAM) tightrope. Often, adequate security controls are one side while user productivity is on the other. Exposing sensitive company or customer Personal Information can result in catastrophic consequences, including customer loss, brand damage, compliance fines, lawsuits, and more. The role of IAM can be critical in determining who has access to which servers and what tasks they can perform through strong authentication, dynamic authorization, and strict policy enforcement.

**375,000**
cybersecurity attacks in 2018 revealed by Verizon

**65%**
of incidents were server attacks

Gartner defines IAM as a critical security discipline that enables the right individuals to access the right resources at the right times for the right reasons. IAM addresses the mission-critical need to comply with increasingly complex compliance mandates while ensuring appropriate access to resources across heterogeneous server environments. Gartner says IAM is “a crucial undertaking for any enterprise,” and firms that develop mature IAM capabilities can reduce costs and increase agility to support new business initiatives.

If your firm utilizes Infrastructure as a Service (IaaS) vendors, such as Amazon Web Services, Google Cloud Platform, or Microsoft Azure to deploy more workloads, cloud or hybrid IAM could present difficult challenges. Gartner stresses using a “high-value, high-risk” approach, and any IAM project should support multifactor authentication (MFA), along with Least Privilege Access.

This whitepaper outlines best practices to ensure the success of an IAM initiative by enforcing the principle of Least Privilege Access on Linux servers.

# Overview

For almost any digital initiative, many organizations are adopting DevOps practices to better automate the delivery and operations of software in the cloud. Given an ever-changing surface area, automation often opens the door to potential risks. Forrester Research estimates that 80% of breaches involved some form of privileged credentials, which grant administrative rights to the holder. A core challenge with these credentials (SSH Keys in the case of Linux servers) is the lack of any clear tethers to identity. This approach exposes firms to the equivalent of Adverse Possession, wherein “trespassers become property owners.”

With the introduction of Advanced Server Access, Okta provides a seamless approach to ensuring secure access to Linux servers without using static SSH Keys. Unlike traditional privileged access solutions, Advanced Server Access delivers an Identity-First approach where policy and activity are securely linked to users. Any successful IAM initiative needs to incorporate Least Privilege Access. As the name implies, this means only granting minimum rights to perform specific functions. Proper implementation ensures adherence to role-based access controls across critical infrastructure resources. Any written IAM policies that can’t be enforced become ineffective.

# Best Practices for Least Privilege Access

The following best practices focus on Linux servers and the core functionality of Okta Advanced Server Access.

## Identity-First Zero Trust Model

Forester's Zero Trust Model advocates "verifying before trusting" in every instance. Using shared accounts is an anti-pattern to securing access to servers in a Zero Trust fashion. Strongly authenticated sessions are based on the user and the device from which they are connecting. Having a clear and secure link to identity strengthens access controls and ensures a clear audit log. For Linux servers, we recommend provisioning local user accounts that are directly tied to your Identity Provider. Provisioning directly to downstream servers eliminates the need to run a synchronous directory interface, such as an LDAP Pluggable Authentication Module, as an intermediary.

Advanced Server Access focuses on users by making a clear and direct connection between a company directory and a server fleet. A lightweight agent is installed on each server across the fleet using an automation service of choice, such as Terraform, Chef, Puppet, Ansible, or others, to communicate with the Okta API and register changes in user status, group membership, or entitlements in near real-time. The agent updates the server to ensure accounts and policies are always up-to-date at any level of scale or elasticity.

### Best Practice Recommendations:

- Never use shared accounts. Zero Trust Model best practices advise that everything should be based on a user's identity, directly tied to your organization's system of record.
- Users and groups should be automatically provisioned and deprovisioned directly from the system of record to the local account database (/etc/passwd) with no intermediary (e.g; LDAP PAM).
- Focus on users by making a clear and direct connection between a company directory and a server fleet.

## Ephemeral Credentials

Using SSH Keys, even when password protected, is also an anti-pattern with Zero Trust. Making a dynamic, contextual access decision needs an on-demand credential mechanism to match. The best approach allows for an abstraction of Public Key Infrastructure (PKI) management complexity while providing server access through an advanced Zero Trust architecture, which can eliminate hours of needless effort by senior engineers.

Advanced Server Access is backed by a “Programmable Certificate Authority (CA)” that mints short-lived, tightly scoped client certificates for every request. Each certificate is linked to a user and a device logging into a server and expires within three minutes. Self-revocation renders the credential useless after use within the authorized scope. Every Advanced Server Access Project, which is a collection of servers and their respective permissions, has its own backing Certification Authority that mints a certificate on-demand for each individual request only after full authentication and authorization through an SSO and MFA workflow.

### Best Practice Recommendations:

- Eliminate static SSH Keys in favor of one-time client certificates minted on-demand.
- Reduce Public Key Infrastructure (PKI) management complexity by abstracting through a SaaS layer
- Restrict non-certificate access to certain groups, eliminate authorized_keys for individual users and only keep a single admin “break glass” key stored in a vault for emergency situations.

## Role-Based Access Controls

With any IAM initiative, recommended best practices include assigning policies to roles where a group of users are able to share a permission model. In this way, users can be easily added and removed manually or via automation. This recommended approach should function in an identical fashion for servers on-premises, in the cloud, or in hybrid infrastructures.

The Linux permission model is centered around users and groups. Similar to Identity-First, local group accounts and respective memberships should be provisioned directly from the backing Identity Provider. With Advanced Server Access, as they are with user accounts, local group accounts are created on the downstream servers using a lightweight agent. With Okta Identity Cloud as the backing directory, group membership is maintained and any changes are identified in near real-time.

### Best Practice Recommendations:

- Assign users to groups that share a permission model (e.g; admin, web, qa, database, etc.).
- Allow for the addition or removal of users without having to manage individual permissions.
- Local group accounts and respective memberships should be provisioned directly from the backing Identity Provider.

## Sudo Entitlements

With users and groups forming the foundation for local accounts, entitlements are the enforcement mechanism to ensure adherence to the Least Privilege Access principle. Specific commands executed by users should be whitelisted based on group membership. Certain administrators may have full privileges on servers, but others should be more restricted based on roles and requirements. The Linux construct for privilege escalation is sudo, which stands for “Super User Do.” Sudo is backed by a policy file named “sudoers” wherein specific executables and directory paths can be explicitly granted to specific users and groups. Sudo enforces command-level Least Privilege Access.

Advanced Server Access works directly with the local sudo construct, allowing administrators to manage entitlements through a central control plane, provisioned as local sudoers drop-in files via the lightweight agent in a similar manner as user and group accounts. Sudo entitlements with Okta allows greater control and flexibility to server administrators by explicitly permitting select commands to select users. This allows managers to better enforce Least Privilege Access across critical infrastructure resources.

### Best Practice Recommendations:

- Restrict permissions to a specific whitelist, with the default as “no privilege,” and only explicitly grant privileges as required.
- Configure the local sudoers file (/etc/sudoers) to drop-in permissions files (/etc/sudoers.d/) for each individual entitlement.
- Administrators should be able to manage entitlements through a central control plane provisioned as local sudoers drop-in files via a lightweight agent.

## Bastion Architecture

A proper Zero Trust architecture eliminates the need for a VPN as the network is no longer a binary access decision point. Shifting controls to Layer 7 enables contextual access controls, however, it is still a recommended best practice to incorporate network controls and avoid lateral movement. Whether deploying in the cloud or on-premise, production hosts that power workloads and store sensitive data should reside in private networks and only allow inbound access from bastion hosts that live in a public subnet. For Linux servers, ingress SSH access should be limited to port 22.

A properly configured bastion architecture eliminates the need for a Virtual Private Network (VPN), extending seamless authentication workflows from any location. Okta views bastions as a first-class feature with Advanced Server Access, allowing the configuration of target systems with bastion hosts where the authentication and transport can occur transparently. SSH traffic can be mutually encrypted across all hops and independently authenticated and credentialed. Bastion hosts can share a configuration for High Availability (HA), wherein the SSH connection from clients arbitrarily choose one of many hosts.

### Best Practice Recommendations:

- Bastion configuration and transport should be transparent to the end user logging into a machine.
- SSH traffic should be mutually encrypted across all hops and independently authenticated and credentialed.
- Bastion hosts should share a configuration for high availability, where the SSH connection from the Client arbitrarily chooses one of many hosts to hop through.

# Okta Advanced Server Access

Okta Advanced Server Access extends the full power of the Okta Identity Cloud to the machine level. Seamless Single Sign-On with Adaptive MFA workflows are integrated natively within the SSH protocol. Universal Directory is the source of truth for server users and groups, with Lifecycle Management handling the provisioning and deprovisioning of local accounts.

## Zero Trust Architecture

Advanced Server Access allows for an abstraction of PKI management complexity while providing server access through an advanced Zero Trust architecture to dramatically reduce complexity and effort. Unauthorized access is prevented by minting ephemeral client certificates on-demand for each individual request after full authentication and authorization through an SSO and MFA workflow.

## Designed for Scale & Automation

With Advanced Server Access, any change in user status, group membership, or policy is automatically detected and reflected across an entire server infrastructure in near-real-time.

Okta becomes the source of ultimate truth for users and groups, and automatically provisions or deprovisions downstream as local machine accounts. A lightweight Server Agent runs on each machine and captures login activity as a log entry that can be further analyzed via the dashboard or third party SIEM service. Enrollment, provisioning, and configuration can all be fully automated, ensuring simplicity and scalability.

## Centralized Least Privilege Access

With Okta Identity Cloud as the unified Identity layer, Advanced Server Access provides a central access control plane to servers across any environment. Our agent-based approach works the same for servers on-prem as it does on cloud instances across AWS, GCP, or Microsoft Azure, abstracting the complexity of managing server IAM. Sudo entitlements can be managed centrally and pushed downstream to each server based upon explicit group assignments. Authorization can extend to an explicit time window for specific users and servers for such cases when a user should only be granted temporary access. When a user is deactivated, the local user account is immediately disabled, eliminating user access risks.

# Conclusion

The prevalence and sophistication of cyber security attacks is escalating at an alarming rate. Recent research validates that server attacks account for a majority of the targets and traditional privileged access solutions have failed to adequately protect sensitive information stored on Linux servers. They have also injected several complexities and inadequacies stemming from static credentials that can be lost, stolen, or misused. Also, from difficult and time-consuming manual provisioning, a lack of direct ties to identity profiles, and risky shared access across systems.

To address these issues, IS and IT professionals are reexamining their IAM effectiveness and maturity. The primary targets for attackers are privileged or empowered accounts, requiring the need for comprehensive and updated IAM best-practices to mitigate risks. Gartner recommends using a "high-value, high-risk" approach that supports MFA. Least Privilege Access best practices should be followed for any IAM initiative, or to bolster a firm's overall security posture.

Okta Advanced Server Access allows IS and IT professionals to implement proven Least Privilege Access best practices and deliver a seamless user experience that works "out of the box" to protect your firm's Linux servers from unauthorized access.

Learn more about Okta Advanced Server Access at
https://www.okta.com/products/advanced-server-access/

## About Okta

Okta is the leading independent provider of identity for the enterprise. The Okta Identity Cloud enables organizations to securely connect the right people to the right technologies at the right time. With over 6,000 pre-built integrations to applications and infrastructure providers, Okta customers can easily and securely use the best technologies for their business. Over 6,550 organizations, including 20th Century Fox, JetBlue, Nordstrom, Slack, Teach for America and Twilio, trust Okta to help protect the identities of their workforces and customers. For more information, visit us at www.okta.com or follow us on www.okta.com/blog.