How Teleport Delivers Security Best Practices for Privileged Access Management of Cloud-Native Infrastructure

Teleport

Table of Contents

Abstract 1
Challenges of Modern Privileged Access Management 2
Security Best Practices Included in Teleport 3
- Enforcing the use of Proxies / Bastions 3
- Ephemeral Certificates Instead of Static Keys 4
- Role Based Access Controls 4
- Single Source of Identity 5
- Forensic Level Audit Logging 6

Additional Benefits 6
- Single Control Plane for Modern and Legacy Access Patterns 6
- Accommodates Existing Behaviors to Improve User Adoption 6
- Open Sourced and Leverages Open Standards 7
- Third-Party Security Auditing 7

Conclusion and Additional Resources 7
Appendix A: Teleport Diagram 8

Abstract

This paper is an overview of how Teleport delivers security best practices necessary for providing secure privileged access for multi-team organizations accessing dynamic, multi-cloud and on-prem infrastructure environments. It assumes knowledge of Public Key Infrastructure (PKI), Public Key Cryptography, Secure Shell (SSH) and Kubernetes.

Challenges of Modern Privileged Access Management

In the days when servers were precious pets and applications were large monolithic code bases, securely accessing application environments for updates and maintenance was a fairly simple process of installing the public keys of authorized users on servers and using SSH. The use of key management or server configuration tooling could be used to automate the management of keys for larger or more dynamic teams.

Due to its simplicity and popularity, SSH keys are a popular attack vector for the unauthorized access of server infrastructure, even with the wide availability of key management tooling. In 2014, a Ponemon Institute survey of more than 2,100 systems administrators at Global 2000 companies discovered that three out of four enterprises are vulnerable to root-level attacks against their systems because of their failure to secure SSH keys. These concerns led to the National Institute of Standards and Technology ("NIST") to publish a 37 page guide on the best practices for securing server access management using SSH. More recently, the increased adoption of elastic, cloud infrastructure and dynamic, micro-service architecture using containerized application services (aka, "cloud-native" applications), has resulted in the additional complexity of having application services that can migrate across dynamic server infrastructure. This makes managing access to applications and their infrastructure through SSH more complicated and more prone to security threats.

Three out of four enterprises are vulnerable to rot-level attacks against their systems because of their failure to secure SSH keys.

These new methodologies bring new software and technologies for accessing and managing application environments. Using SSH to access infrastructure is increasingly becoming an emergency ("break-glass") measure of last resort and administrators are relying on more modern tools like Kubernetes to manage immutable infrastructure by replacing components rather than doing in-place management of patches and/or upgrades. While this may reduce the number of people and processes that need SSH access, SSH is rarely abandoned completely as the foundational method for accessing servers. The additional technologies also lead to a larger surface area of workflows and tools for which to manage privileged access and role-based permissions. Unfortunately, as is the case with any new technology or methodology, the tools to properly secure its use usually lag behind the usage, especially with many teams in large organizations adopting tools independent of central IT (aka, "shadow IT"). For example, Tesla was a recent victim when attackers gained entry to its cloud servers through an unsecured Kubernetes admin console and used the servers for crypto mining. The vulnerability was discovered during a study by security researchers at Redlock, who found hundreds of Kubernetes administration consoles accessible over the internet without any password protection during their study.

Security researchers found hundreds of Kubernetes administration consoles accessible over the internet without any password protection.

If that wasn't enough complexity, organizations are increasingly adopting a hybrid infrastructure strategy of using one or more third-party cloud providers, in addition to on-premise or collocated data centers.

All of this results in CSOs and their staff having to deliver on the difficult mandate of securing their company infrastructure while minimizing the disruption to existing developer and DevOps workflows. In this paper, we will go through how Teleport delivers security best practices for modern Privileged Access Management.

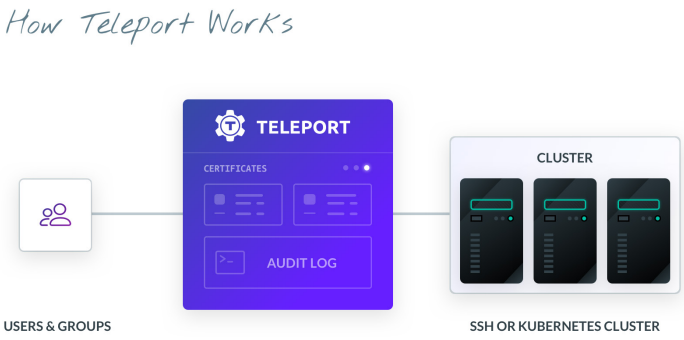
Security Best Practices Included in Teleport

Teleport was designed to deliver the latest security best practices for managing privileged access to infrastructure. In addition, we are dedicated to maintaining and updating these best practices as new technologies and methodologies emerge and evolve.

Enforcing the use of Proxies / Bastions

Teleport enforces the use of proxy or bastion endpoints when accessing the infrastructure behind them. Bastion servers are access endpoints set up as the only way to get into the rest of the cluster and provide a way to expose only a small part of the infrastructure. Teleport users / clients must first connect to the Teleport Proxy Service before accessing the nodes behind it. The Proxy Service is designed by default to be stateless and it does not have access to any unencrypted data. The Proxy Service enables the following security best practices:

- A proxy exposes only a limited portion of the infrastructure to the outside world.
- Limits additional attack vectors if the Proxy is breached (assuming it is configured to only pass through encrypted traffic).
- Limiting access points allows for easier monitoring of traffic and more easily shut down access in the case of security threats like a lost laptop.



Read more about the Teleport Architecture in the Teleport Documentation.

Ephemeral Certificates Instead of Static Keys

As mentioned in the challenges above, SSH keys represent a common attack vector. Using static keys can often lead to breaches due to common security events like:

- Lost or stolen devices.
- Accidental exposure of private keys (most commonly on shared repositories like Github).
- Mismanagement of employee or contractor public keys.

```
John Smith (js@example.com)
ROLES: developer, intern

ALLOWED UNIX LOGINS:
    - jsmith, admin

EXPIRES: July 26, 13:20
```

Teleport requires the use of a Certificate Authority which issues ephemeral or short lived certificates to determine authorization. Each user / client has to authenticate who they are each time a certificate expires. This limits the scope of the attack vectors above. Another benefit of using certificates is that they can contain metadata to determine the permissions a specific user/identity should have in the system.

Teleport Certificate Authorities can also be configured to trust each other and share permissions. This makes it easy to manage access and permissions across multiple clusters and teams.
Read more about certificates in the Teleport documentation.

Read more about certificates in the Teleport documentation.

Role Based Access Controls

The use of certificates allows for the specification and enforcement of fine-grained permissions that are tied to a specific user or identity (aka, role based access controls or RBAC). This allows for additional controls beyond the authentication and authorization of users. Some typical use cases would be:

- Contractors can only access staging servers.
- Developers can not write to production databases.
- Only highest clearance employees can access customers' private health data.

```
John Smith (js@example.com)
ROLES: developer, intern

ALLOWED UNIX LOGINS:
    - jsmith, admin

EXPIRES: July 26, 13:20

PERMISSIONS:
    - Staging servers: ALLOWED
    - Port forwarding: ALLOWED
    - Terminal: ALLOWED
    - Prod servers: DENIED
    - Secure file copy: DENIED
```

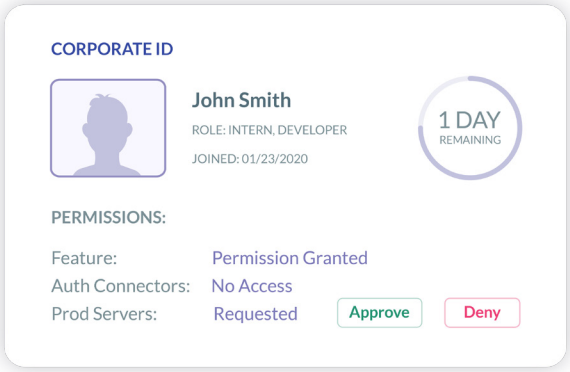
Teleport also allows for the ability to use labels to link these permissions to dynamic services that may move across servers, which is useful with microservice application architectures.

Teleport also integrates with Kubernetes RBAC for those that are using more modern, cloud-native tooling.

Read more about RBAC in the Teleport Documentation.

Single Source of Identity

Teleport is designed to integrate with existing identity management services through common protocols like SAML and OIDC. This means that Teleport uses an organization's single source of identity for managing access to its infrastructure. This also means there is no need to manage an additional source of identity. Some of the identity management services supported include:

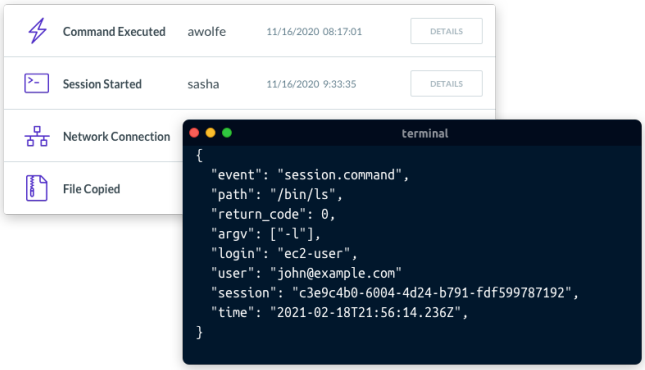


- Okta
- Centrify
- ADFS
- One Login
- Auth0
- Google GSuite

Review the Okta integration guide in the Teleport Documentation.

Forensic Level Audit Logging

Restricting access and granting specific permissions through RBAC is the first step to securing infrastructure. The next step is to log all activity across the infrastructure.



Secure Audit Logs. Teleport logs events like successful user logins along with the metadata like SSH events, remote IP address, time and the session ID. Teleport does not store the audit logs on the machines where the activity is occuring. Logs are either stored on the secure Auth Server or can be shipped to external storage like DynamoDB or logging services like Splunk.

Session Recording. Teleport goes beyond just recording events and also records all of the SSH sessions, stores them and makes them available for playback so you have further visibility into what's happening during the session.

Cluster Level Logging. Finally, because Teleport is designed for clusters, it will log activity across an entire cluster or environment, not just on a specific machine. This allows you to track activity across your infrastructure more easily.

Read about the audit log in the Teleport documentation.

Additional Benefits

While not features per se, Teleport was designed and is maintained with principles that help to improve the security at organizations where it is used.

Single Control Plane for Modern and Legacy Access Patterns

In addition to managing SSH access, Teleport integrates with Kubernetes RBAC to control access and permissions for cloud-native workflows. Kubernetes' `kubectl exec` command are logged and sessions are recorded. This gives the administrators of the infrastructure the ability to use one tool for modern workflows through Kubernetes and legacy or emergency access using SSH. Learn more about the Kubernetes integration in the Teleport Documentation.

Accommodates Existing Behaviors to Improve User Adoption

Teleport is designed to stay out of the way of existing behaviors and workflows. The best security systems are useless if no one wants to use them. That is why Teleport is fully compatible with OpenSSH and supports traditional SSH commands and workflows. Similarly, Teleport works seamlessly with Kubernetes workflows. If a user doesn't want to take advantage of the additional Teleport feature set, they should not even realize they are using it.

Open Sourced and Leverages Open Standards

Teleport code is open sourced and available on Github , with over 6,700 stars. Teleport is also committed to using open standards. For example, Teleport does not use its own implementation of cryptography. Instead, Teleport is built on top of the high-quality Golang SSH implementation developed at Google.

Third-Party Security Auditing

Teleport is audited regularly by best-in-class, third-party security consultants. These audits include full pen testing and code reviews. Below is a quote from our most recent audit.

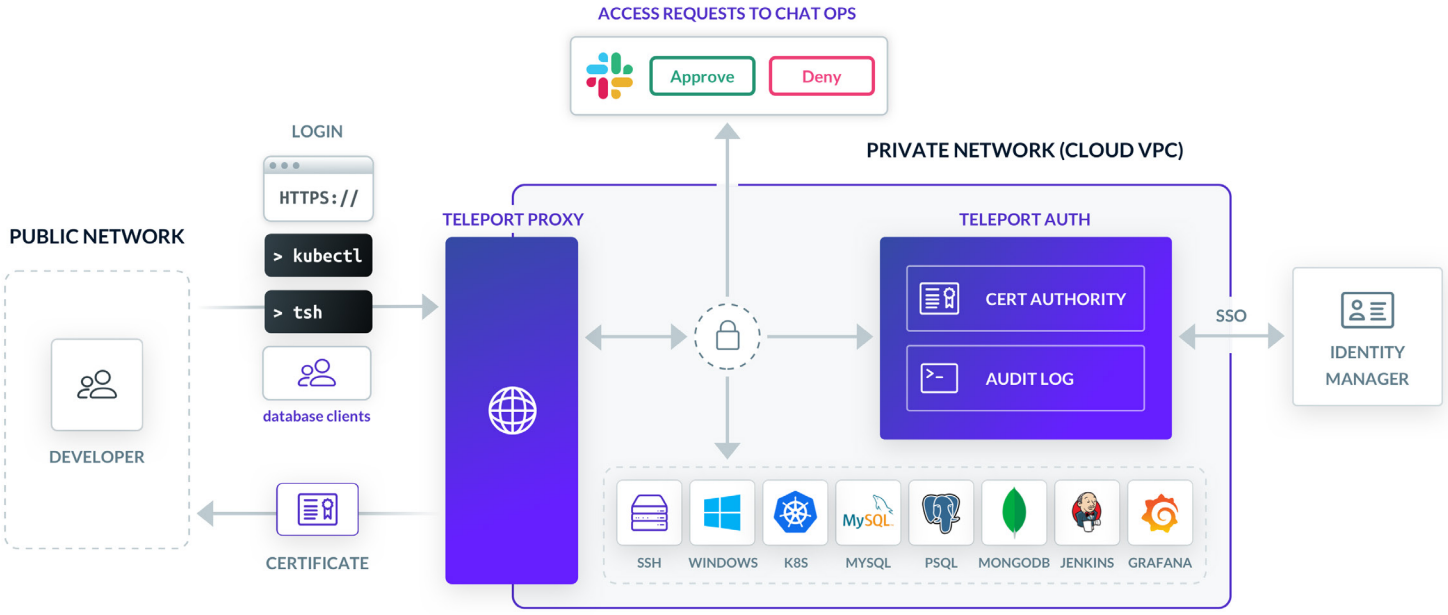
"The results of this second-run Cure53 security assessment of the latest release of the Teleport software are once again very positive. With the first Cure53- Teleport collaboration already yielding good results, the fact that this time the findings are few and are between across the board is very much praiseworthy.

The overall feeling of coherence caused by good design decisions, the achieved good test coverage, as well as the choices of relying on a modern implementation language, translated to the conviction that Teleport should be seen as a mature software package. The product can subsequently be recommended for general deployment in security critical environments."

Conclusion and Additional Resources

We hope this paper has given you a better understanding of how Teleport can help you manage privileged access across your infrastructure footprint and implement security best practices in your organization.
You are also welcome to review additional information in the online Teleport Documentation, the Teleport Resources Library (where we have the security audits and other white papers) or feel free to reach out to us in person for more information or a demo.

Appendix A: Teleport Diagram



To learn more about Teleport architecture, please visit https://goteleport.com/teleport/how-it-works/