SCYTALE

# Scytale Enterprise

## Easily and Securely Extend Kerberos to the Cloud

## THE PROBLEM

As organizations adopt cloud-based infrastructure, application services running in the cloud must be able to establish trusted communication with on-premise systems of record. To allow rapid cloud adoption without compromising security, IT security and engineering teams want to extend their existing security investments into cloud environments without significant modification to existing tooling, practices or client libraries. Extending existing Kerberos-based authentication, in particular, becomes a priority.

However extending existing Kerberos-based authentication into cloud- and container-based environments is often stymied by the following obstacles:

- **Manually provisioned credentials are incompatible with cloud automation:** Services that participate in Kerberos authentication must first establish a connection to the Kerberos-enabled Identity Provider (IdP) using a keytab—a long lived credential. Any actor who obtains this credential can impersonate this service, and often do so undetected. In traditional environments, a trusted human operator generates this credential manually and delivers it to the node running the service. However, in cloud-based or container-based environments where nodes and/or services are provisioned dynamically (for example, because of elastic scaling in the public cloud, or dynamic scheduling in a container orchestrator), this process must necessarily be automated. This, in turn, requires complete trust in the automation process as well as the workload itself, and any human operator who interacts with it.
- **Long lived credentials become an attractive target:** Since keytabs are long lived credentials, they become an attractive target for a malicious actor because they can be used to impersonate a service long after it has been exfiltrated.
- **Providing network line-of-sight between an identity provider and cloud service is complex and poses a risk:** In traditional Kerberos authentication, a source service requires a network line-of-sight not only to the destination service it is calling but also to the Kerberos identity provider itself (for example, an Active Directory deployment). This can be logistically challenging (given the dynamic nature of cloud networking) and demands that the security team perform additional investigation of the workload and its security privileges, which slows time to delivery.
- **Load spikes on the IdP disrupt service performance:** In traditional environments, a Kerberos keytab is typically exchanged for a short lived ticket-granting ticket when the service starts or first authenticates, a cryptographically expensive operation. However, provisioning services rapidly through automation, elastic scaling, or dynamic scheduling can lead to unexpected load spikes on the IdP, which can have a widespread impact on services dependent on that provider.
- **Limited audibility of the ticket issuance makes compliance difficult:** Compliance and security best practices often require detailed auditing of when and where authentication credentials are issued, which entitlements they confer, and for how long credentials are valid. Conventional IdP's auditing Kerberos ticket issuance typically rely on recording the IP address of the user principal that requested the ticket. In cloud- or container-based deployments, an IP address is not a useful, stable identifier for auditability.

## SOLUTION OVERVIEW

Scytale Enterprise, an industry-first service identity platform, allows you to easily and securely extend Kerberos-based authentication infrastructure to the cloud. The solution securely issues short lived credentials from on-premise identity providers (IdPs) such as Active Directory to cloud and container-based services. It also enables cloud services to access on-premise services without exposing IdPs to the public internet or breaking or changing existing risk policies.

For information please visit **scytale.io**

## HOW DOES IT WORK?

**Scytale Enterprise achieves the above through two core capabilities:**

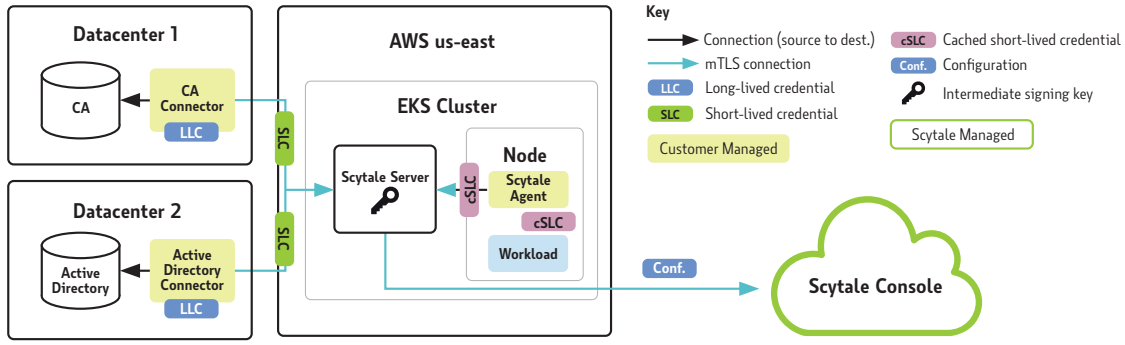
**1. Multifactor service authentication based on industry standards (SPIFFE):** Scytale Enterprise performs a zero-trust attestation process that leans on a configurable union of trusted third parties (such as cloud provider control planes, container orchestrators, signatures from CI/CD pipelines, and trusted OS kernels, Hardware Security Modules, and existing machine identity frameworks) to provide a strongly attested identity for cloud (and optionally, on-premise) services.

Service identity is thus conferred by a detailed set of identifying attributes of the service in question, rather than by the presence of a service ticket or an IP address. Scytale Enterprise thus provides a highly trusted identity in dynamic cloud- and container-based environments and avoids the pitfalls and complexities of conventional secrets management.

The identities Scytale Enterprise issues conform to the SPIFFE open standard, which is backed by the Cloud Native Computing Foundation (CNCF). Organizations can use these identities to authenticate services in a wide variety of applications.

**2. Identity brokering from a Kerberos identity provider to SPIFFE-identified services.** Having established a service's identity through multifactor authentication, an application must then be able to deliver to it a scoped, short lived Kerberos ticket. Scytale Enterprise includes an identity brokering capability in which a SPIFFE-identified service may obtain a short lived Kerberos service ticket from an identity provider. A lightweight, secure Scytale Connector maintains trust between Scytale Server and the identity provider—such that these sensitive long lived credentials need never be delivered to directly target services—and handles direct interactions with the Kerberos Identity Provider to generate short lived service tickets. Short lived credentials are then delivered directly to the service via a channel secured by the multifactor authentication process described above.

FIGURE A




### Example Deployment Model

Figure A shows a typical deployment, in which a hosted Scytale console oversees manages customer-managed Scytale Connectors, Scytale Server and Scytale Agents. The Server, Agent, and Connectors provide isolation and quarantine privilege to the components that need it but require minimal configuration, as most of the configurations are pulled from the Scytale console and integrate with standard monitoring and SIEM tooling.

The console maintains policy and configuration and never requires access to customer-controlled secrets, including the long lived credentials required by other access identity providers or the signing key used to generate SPIFFE identities for a service.

## BENEFITS

**Strengthen your security posture and protect your existing investments:** Multifactor policies establish greater trust in provisioned identities. Short lived credentials reduce threat radius. The ability to extend your existing on-premise IdPs—such as Active Directory—to cloud and container services strengthens your security posture, reduces the risk of compromise, and eliminates the need to rewrite code or re-architect for cloud and container platforms.

**Boost staff and developer productivity** API-driven, automated controls mean that your developers spend less time writing code for security controls or waiting for tickets; and operations teams are able to deploy, operate, and scale authentication easily across dynamic, heterogeneous infrastructures.

**Speed cloud and container adoption** Leverage a common, scalable identity model that works with your existing IdP and is designed for cloud- and container-based environments to seamlessly and securely authorize services across any platform.

**Reduce time to market from weeks to minutes** Eliminate manual processes and development efforts to authorize a service across multiple platforms. Automated, uniform service identity management reduces application onboarding times from weeks to minutes.

For information please visit **scytale.io**