凭据管理系统

产品简介

腾讯云

文档目录

产品简介

产品概述

最近更新时间：2022-12-14 17:36:30

什么是凭据管理系统

凭据管理系统（Secrets Manager，SSM）为用户提供凭据的创建、检索、更新、删除等全生命周期的管理服务，结合资源级角色授权及全面细致的审计管控，轻松实现对敏感凭据的统一管理。用户或应用程序可通过调用凭据管理系统 API，规避敏感配置及敏感凭据硬编码等风险问题，同时可有效避免敏感信息泄密以及权限失控带来的业务风险。

产品优势

企业级凭据管理

凭据管理系统专注于解决用户敏感凭据管理问题，有效避免程序硬编码导致的明文泄密，以及权限失控带来的业务风险。

全生命周期管理

凭据管理系统可以为用户提供凭据的创建、检索、更新、删除、权限管控等全生命周期的管理服务，结合资源级角色授权及全面细致的审计管控，轻松实现对敏感凭据的统一管理。

安全可靠

凭据管理系统架构采用集群化部署方式，通过分布式数据库存储系统实现数据存储与容灾备份。业务侧用户也可多地域创建同样的凭据，实现业务侧的跨区域容灾。

加密存储

凭据通过腾讯云密钥管理系统进行加密存储，基于第三方认证的硬件安全模块（HSM）来生成和保护加密密钥。检索凭据时，通过 TLS 安全传输到服务器本地。

按需付费

用户在使用凭据管理系统时，仅按实际使用量收费。用户可按照在凭据管理系统中管理的凭据数量和 API 调用次数进行付费，无最低费用和设置费用，详情请参见 购买指南 。

产品功能

最近更新时间：2024-04-16 11:20:41

针对敏感配置、敏感凭据硬编码带来的泄露风险问题，所有的凭据由密钥管理系统（KMS）进行加密保护，并且提供简单易用的 API 和 SDK，能够降低用户的使用成本和管理成本。 用户可通过凭据管理系统轻松实现对数据库凭证、API 密钥和其他密钥、敏感配置的集中检索、管理以及加密存储，有效避免程序硬编码导致的明文泄密，以及权限失控带来的业务风险。

安全可控的凭据检索

从应用程序的源代码中删除硬编码凭据，将代码中的硬编码凭据替换为对凭据管理系统 API 调用，以便用编程的方式动态检索及管理凭据。

凭据加密存储与传输

凭据管理系统使用 密钥管理系统（KMS）安全保护的主密钥（CMK）作为加密密钥，并对所管理的凭据内容进行加密存储，使用凭据时，将通过 TLS 安全传输到服务器本地。

应用层凭据轮换

用户可通过凭据管理系统按周期更新敏感凭据内容，依赖该凭据的所有应用点将自动完成同步，安全实现凭据轮换管理，同时确保依赖该凭据业务的连续性。

存储多类型凭据

通过 Name-Value 的方式存储多种类型数据，Value 部分最大支持4096字节，例如数据库凭据、账号密码及 IP 端口等。

资源级访问授权

凭据管理系统与腾讯云 访问管理（CAM）集成，通过身份管理和策略管理方式确保只有授权用户可以访问或修改凭据，并可以将这些策略附加到用户或角色，指定这些用户或角色可以访问哪些凭据。

精细化监管审计

凭据管理系统与腾讯 操作审计（CloudAudit）结合，支持对用户的腾讯云账号进行监管、合规性检查、操作审核和风险审核等，同时可记录凭据管理操作和凭据使用情况。

高可用容灾备份

凭据管理系统架构采用集群化部署方式，通过分布式数据库存储系统实现数据固化与容灾备份，实现业务侧的跨区域容灾。

安全合规性说明

凭据管理系统与密钥管理系统（KMS）相关联，密钥管理系统底层使用经过第三方认证的硬件安全模块（HSM）来生成和保护密钥，符合监管和合规要求。

自动凭据轮换

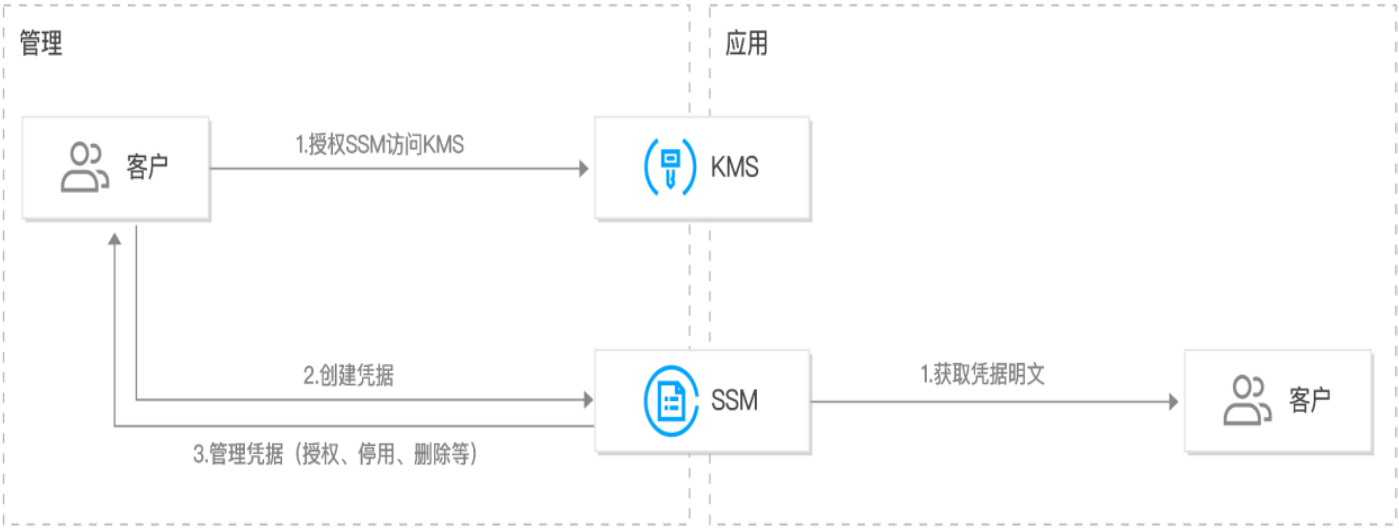
目前云上产品的账号密码面临管理权限失当、账号密码长时间不变更、配置中的密钥信息是明文等问题，导致数字资产损失。
针对于这些风险，数据库凭据对凭据进行定期轮换，自动创建高强度密码和管理敏感配置信息，在降低账号的风险与安全威胁时，也能提高业务数据的安全性。

应用场景

最近更新时间：2024-04-16 11:20:41

凭据集中管控

- **场景痛点**：为保障业务开发敏捷性，系统中往往存在大量的敏感信息，例如账户信息、Tokens、证书、SSH 密钥及 API 密钥等，因此需要对敏感凭据进行统一的存储、检索及使用等全生命周期管控。
- **场景举例**：多应用敏感配置信息凭证加密存储、查询管理等生命周期管理。
- **面临挑战**：敏感凭据硬编码、权限管理混乱、凭据托管管理难。
- **解决方案**：业务开发者可通过 凭据管理系统控制台 、SDK 或命令行界面创建、使用、存储敏感配置信息的凭据。通过凭据管理系统与访问控制 CAM、操作审计 CloudAudit 产品的结合，业务管理者可实现对企业凭据全生命周期的统一管理。



敏感凭据检索管理

- **场景痛点**：当用户访问应用程序或服务时，需创建身份验证的数字证书，例如密码、令牌、证书、SSH 密钥或 API 密钥等各种类型机密信息，通常直接使用明文方式嵌入在应用程序的配置文件中，安全性较低。通过凭据管理系统可有效避免敏感凭据硬编码等风险问题。
- **场景举例**：数据库凭据、API 密钥、账号密码等。
- **面临挑战**：敏感凭据信息泄露。
- **解决方案**：用户可以将代码中的硬编码凭证（包括密码）替换为对凭据管理系统 API 的调用，以便用编程的方式动态查询凭据，由于该凭据中不包含敏感信息，所以可以保证密钥不被泄露。

配置文件： app.ini
[mysql]
connString=user:pwd@tcp(10.x.x.x:1234)/db_a?charset=utf8

初始化：
```
func DbInitDemo() {
    dbConnect, _ := sql.Open("mysql", conf.MysqlConnStr)
    dbConnect.Ping()
}
```

配置文件： app.ini
[mysql]
connStringName=DB_A
versionId=V1.0

初始化：
```
func DbInitDemoSsm(client *ssm.Client) {
    request := ssm.NewGetSecretValueRequest()
    request.SecretName = conf.MysqlConnStrName
    request.VersionId = conf.MysqlConnVersion
    rsp, _ := client.GetSecretValue(request)

    dbConnect, _ := sql.Open("mysql", rsp.Response.SecretString)
    dbConnect.Ping()
}
```

凭据轮换

- **技术痛点：**为提升系统安全性，需要对敏感凭据进行定期更新，通过凭据管理系统可以实现更新。
- **场景举例：**应用层凭据轮换。
- **面临挑战：**凭据轮换时要求对目标凭据具备依赖性的应用或配置同步更新，多应用系统凭据更新容易遗漏，可能带来应用中断风险。
- **解决方案：**在凭据管理系统中通过控制台新增凭据版本或通过调用 API 更新目标凭据内容，用户可自主选择全量或者灰度轮换凭据，实现对依赖目标凭据的所有应用点的同步更新。