百度智能云

CAS 文档

目录

产品动态

最新公告

免费SSL证书有效期调整通知

免费SSL证书有效期调整通知

尊敬的百度智能云SSL证书客户，您好：

受上游厂商免费1年期SSL数字证书调整策略影响，自2024年8月1日0时起，从百度智能云新购的TrustAsia品牌DV证书单域名版（测试版）免费期限将从1年调整为3个月。具体如下：

1. 自2024年8月1日0时起，新申请的TrustAsia品牌DV证书单域名版（测试版），签发后有效期为3个月。

> 2024年8月1日0时起，已购买但未申请的证书在提交申请时，其有效期将变为3个月；
>
> 2024年8月1日0时起，处于“申请中”状态的证书，若申请通过，证书的有效期仍为12个月，但如果取消申请并重新提交申请，证书的有效期也将变为3个月；
>
> 证书的有效期自签发之日起计算。

2. 在2024年8月1日0点之前已签发的原1年期TrustAsia品牌DV证书单域名版（测试版），证书有效期仍为1年。在证书到期后您可以自行决定是否续费，续费后的新证书有效期为3个月。

感谢您对百度智能云的支持与关注！

有关TrustAsia品牌根证书的切换说明

尊敬的百度智能云用户，您好：

为了提高SSL证书的稳定性和兼容性，加快签发速度，TrustAsia品牌SSL证书已于2025年01月14日进行升级。 本次升级主要表现为根证书的变更，其它部分无发生变化。

根证书变更详情

根证书变更前	根证书变更后
USERTrust RSA Certification Authority	DigiCert Global Root G2
USERTrust ECC Certification Authority	DigiCert Global Root G3
AAA Certificate Services	DigiCert Global Root CA

本次升级影响

- 2025年01月14日之前签发的所有SSL证书不受影响，均可正常使用。
- 2025年01月14日之后签发的TrustAsia证书，会自动为您替换为新根证书进行签发，其他品牌SSL证书不受影响。

百度自有品牌证书BaiduTrust下线通知

尊敬的百度智能云用户，您好：

因SSL证书战略调整，百度自有品牌**BaiduTrust SSL证书将于2024年8月31日0点下线**，详情如下：

在2024年8月17日0点后，**将不能通过百度SSL证书控制台及API接口下单**BaiduTrust SSL证书；

在2024年8月17日0点前**已购买但未申请的BaiduTrust SSL证书**，将为您保留至2024年8月23日23点59分59秒；在2024年8月24日0点后，已购买但未申请的BaiduTrust SSL证书**系统将自动退款**；

在2024年8月24日0点前**已申请但未签发的BaiduTrust SSL证书**，我们将于7个自然日内完成签发，如遇签发失败或**未签发**将正常退款；

在2024年8月31日0点前**已签发的BaiduTrust SSL证书**，在2024年8月31日0点后将无法续费；

在2024年8月31日0点后**多年期已签发且未到期的BaiduTrust SSL证书**，可正常使用；

如果您的网站或小程序商城参与售卖BaiduTrust SSL证书，**即日起请尽快调整相应页面**，以免引起不必要的损失。

以上通知即日起生效。

在此期间，您可以购买其它品牌证书，点击百度智能云SSL证书官网可查看其它品牌证书，百度智能云SSL证书团队将继续为您服务。

感谢您对百度智能云的理解和支持！

TrustAsia 免费证书 CT 政策问题

尊敬的百度智能云SSL证书用户，您好：

本次通知事件涉及范围为 2024 年 9 月 18 日 15:35（UTC+8） 至 2024 年 9 月 18 日 16:37（UTC+8）期间签发的部分 TrustAsia 品牌 90 天免费证书，不涉及 1 年期收费证书。

我们遇到了一个比较罕见的问题：

1.事件背景：行业规范要求小于 180 天证书需要发布至两家 CT 运营商的两个 CT 日志服务器；大于 180 天的证书要求提交并发布在两家 CT 运营商的三个 CT 日志服务器。而本次受影响的证书（皆为 90 天证书），被提交到了 Google 旗下的两个 CT日志服务器，故不符合需要分布在两家运营商的要求。

2.事件影响：如您在该期间内申请并签发了此类证书，会因为不符合行业规范而导致浏览器不受信问题。

3.解决方案：我们建议您针对该时间段内申请的证书立即**重新申请**。

我们对为此造成您的不便而感到抱歉，非常感谢您的理解与配合！

DigiCert 品牌根证书升级通知

尊敬的百度智能云用户，您好：

受全球知名信任库 Mozilla的根证书最新信任策略影响，DigiCert将逐步停用旧根体系(G1)颁发TLS/SSL证书，并开始使用新根体系(G2)颁发TLS/SSL证书，以确保TLS/SSL证书在主流浏览器中继续受到信任。**因此，自2024年7月1日起，在百度智能云申请的GeoTrust、SecureSite品牌的SSL证书将陆续使用全新的中级证书和根证书签发新证书。**

此次Mozilla的新策略具体为：全球所有CA的可信根证书生成后最少15年更换一次，超过时间的可信根会逐渐被Mozilla停止信任。

百度智能云相关产品将做出对应调整，**自 2024年7月1日起，GeoTrust、SecureSite品牌SSL证书将使用新的G2根体系签发证书，逐步停用旧的G1根体系。**

升级影响

1. 2024年7月1日前，**已签发证书不受影响**，仍有效直至证书过期，证书有效期内进行重颁发，重颁发的证书仍使用旧根签发；
2. 2024年7月1日后，提交的DigiCert品牌SSL证书新订单将使用新根签发；
3. 2024年7月1日后，DigiCert品牌SSL证书多年期订单在下一年续期证书时，将使用新根签发；
4. 新的 G2 根体系依然兼容当前主流的操作系统和移动设备，**不存在兼容性变化问题**，并采用更高安全性的 SHA256 签名

> 算法。

升级详情

证书品牌	证书类型	原中间证书	新中间证书	原根证书	新根证书
GeoTrust	GeoTrust (DV) SSL	GeoTrust TLS ECC CA G1	不变	DigiCert Global Root G3	不变
GeoTrust	GeoTrust (DV) SSL	GeoTrust CN RSA CA G1	GeoTrust TLS RSA CA G1	DigiCert Global Root CA	DigiCert Global Root G2
GeoTrust	GeoTrust (OV) SSL	GeoTrust ECC CN CA G2	GeoTrust G2 TLS CN RSA4096 SHA256 2022 CA1	DigiCert Global Root CA	DigiCert Global Root G2
GeoTrust	GeoTrust (OV) SSL	GeoTrust RSA CN CA G2	GeoTrust G2 TLS CN RSA4096 SHA256 2022 CA1	DigiCert Global Root CA	DigiCert Global Root G2
GeoTrust	GeoTrust (EV) SSL	GeoTrust EV RSA CA G2	GeoTrust EV G2 TLS CN RSA4096 SHA256 2022 CA1	DigiCert Global Root G2	不变
GeoTrust	GeoTrust (EV) SSL	DigiCert TLS Hybrid ECC SHA384 2020 CA1	GeoTrust EV G3 TLS CN ECC P-384 SHA384 2022 CA1	DigiCert Global Root CA	DigiCert Global Root G3
Secure Site	SecureSite (OV) SSL	DigiCert Secure Site CN CA G3	DigiCert Secure Site OV G2 TLS CN RSA4096 SHA256 2022 CA1	DigiCert Global Root CA	DigiCert Global Root G2
Secure Site	SecureSite (OV) SSL	DigiCert Secure Site ECC CN CA G3	DigiCert Secure Site OV G3 TLS CN ECC P-384 SHA384 2022 CA1	DigiCert Global Root CA	DigiCert Global Root G3
Secure Site	SecureSite (OV Pro) SSL	DigiCert Secure Site Pro CN CA G3	DigiCert Secure Site Pro G2 TLS CN RSA4096 SHA256 2022 CA1	DigiCert Global Root CA	DigiCert Global Root G2
Secure Site	SecureSite (OV Pro) SSL	DigiCert Secure Site Pro ECC CN CA G3	DigiCert Secure Site Pro G3 TLS CN ECC P-384 SHA384 2022 CA1	DigiCert Global Root CA	DigiCert Global Root G3
Secure Site	SecureSite (EV) SSL	DigiCert Secure Site EV CN CA G3	DigiCert Secure Site EV G2 TLS CN RSA4096 SHA256 2022 CA1	DigiCert High Assurance EV Root CA	DigiCert Global Root G2
Secure Site	SecureSite (EV) SSL	DigiCert Secure Site EV ECC CN CA G3	DigiCert Secure Site EV G3 TLS CN ECC P-384 SHA384 2022 CA1	DigiCert High Assurance EV Root CA	DigiCert Global Root G3
Secure Site	SecureSite (EV Pro) SSL	DigiCert Secure Site Pro EV CN CA G3	DigiCert Secure Site Pro EV G2 TLS CN RSA4096 SHA256 2022 CA1	DigiCert High Assurance EV Root CA	DigiCert Global Root G2
Secure Site	SecureSite (EV Pro) SSL	DigiCert Secure Site Pro EV ECC CN CA G3	DigiCert Secure Site Pro EV G3 TLS CN ECC P-384 SHA384 2022 CA1	DigiCert High Assurance EV Root CA	DigiCert Global Root G3

公告原文

DigiCert 公告原文

https://knowledge.digicert.com/generalinformation/digicert-root-and-intermediate-ca-certificate-updates-2023.html

SSL证书有效期变更通知及应对策略

【重要通知】2020年9月1日起，SSL证书最长有效期将缩短为13个月

尊敬的百度智能云SSL证书客户，您好：

由于苹果和谷歌根存储策略的变化，从2020年9月1日起，苹果系统、谷歌和火狐浏览器中受信SSL证书的最长有效期将从825天缩短至398天（13个月）。对此，各CA厂商也将缩短SSL证书的最长有效期。受此影响，百度智能云SSL证书预计于2020年8月25日起，停止售卖2年期SSL证书。平台受影响证书品牌包括：CFCA、GeoTrust、Globalsign、Digicert SecureSite、TrustAsia。

1. 有什么变化？

   由于苹果和谷歌根存储策略的变化，从9月1日开始，所有新的SSL/TLS证书的最长有效期不得超过13个月。

2. 此变更何时生效？

   2020年9月1日。

3. 刚刚购买了有效期为2年的SSL证书，2020年9月1号之后会被信任吗？

   在2020年9月1号之前颁发且有效期大于398天的SSL证书将继续受到信任，使用不受影响。

4. 更改生效后，重颁发现有的2年期证书会怎么样？

   如果您在9月1日之后重颁发现有的2年期证书，我们需要把重颁发出来的新证书有效期限制为398天。

5. SSL/TLS有效期缩短对网站管理者有哪些影响？

   SSL/TLS证书有效期缩短将导致证书和私钥的管理成本提高，但好处是确保技术人员采用最新加密标准的SSL证书保护网站安全性，减少证书被盗用的风险。

6. 如何应对SSL证书的有效期策略变化，降低您的证书运营风险？

（1）如果您想要获得有效期超过13个月的SSL证书，请在2020年8月31日截止日期之前抓紧时间完成签发。目前百度智能云平台SSL证书2年期火热促销中，活动持续到8月25日，需要的尽快选购。

（2）及时检查SSL证书有效期，建议对企业网络中所有证书导入百度智能云证书管理中心进行全生命周期的持续监控和管理。

（3）您可在平台选择BaiduTrust品牌证书，最长支持5年购买，分5次自动签发，免除您的后顾之忧！

SSL证书产品官网：https://cloud.baidu.com/product/ssl.html

感谢您一直以来对百度智能云SSL证书的关注和支持，如您有任何疑问可随时拨打平台热线咨询：400-920-8999，我们将竭诚为您提供更优质的服务！

Symantec证书品牌名及品牌标识变更通知

【公告】Symantec SSL证书品牌名及品牌标识变更通知

尊敬的百度智能云客户：

您好！

因CA认证机构DigiCert收购 Symantec 安全认证业务原因。DigiCert将于2020年4月30日起，在全球范围内停止使用与赛门铁克（Symantec）相关的营销及品牌行为。百度智能云SSL证书服务团队将会配合此次变更。除以下变更外，您的“原Symantec”证书使用不会受到任何影响。

1. Symantec品牌SSL证书更名DigiCert Secure Site品牌SSL证书。

2. 此外，Symantec遵守与NortonLifeLock Inc.(以前称为赛门铁克公司)之间的协议。 诺顿安全认证签章同步进行更新，如下所示。

截止2020年4月30日，我司原Symantec品牌相关商务文件将进行统一更名。如您的网页或资料中有使用Symantec品牌信息，请进行更换，否则将可能会承担法律责任。 在此期间，如您有任何异议，请随时与我们联系，我们将一如既往竭诚为您服务！

最新活动

【活动下线通知】下单BaiduTrust SSL证书指定版本送百度网民保障权益计划！

尊敬的用户，您好：

因业务调整，BaiduTrust SSL证书赠送网保计划活动已在7月21日下线，届时请关注后续活动通知。

产品描述

介绍

目前互联网安全威胁愈演愈烈，各类入侵、劫持事件层出不穷，欺诈、钓鱼网站比比皆是。https加密传输方案在传输层可有效防止他人截获，同时客户端浏览器的强制验证手段，可有效帮助用户甄别真实网站，避免上当受骗，因此https也成为越来越多的互联网站的标配。

采用https的服务器必须从CA（Certificate Authority）申请一个用于证明服务器用途类型的证书。该证书只有用于对应的服务器的时候，客户端才信任此主机。证书可自行签发，但自签发证书无第三方监督审核，不受浏览器信任，网站访问会出现安全警示甚至拒绝访问。这就需要由第三方证书服务商签发具备严格用户验证流程、各浏览器版本认证的客户证书服务。

百度智能云已与第三方SSL证书服务平台对接，用户可通过百度智能云自助申请各类SSL证书（包括免费证书和收费证书），并可完成自动化或半自动化的证书核验和签发、部署，帮助用户快捷地将原有服务升级成HTTPS加密传输服务。

将站点原有服务升级为HTTPS后，可有效防止用户信息被他人截取，同时也可以帮助用户甄别真实网站，提升企业形象。在站

点部署DV或OV证书后，浏览器地址栏标记为绿色，表示用户和网站之间建立是私密链接，如下图所示：

在站点部署EV证书，浏览器地址栏标记为绿色并显示认证通过的企业名称，表示该网站是一个受信任网站，如下图所示：

核心概念

SSL：SSL（Secure Sockets Layer）即安全套接层，及其继任者传输层安全（Transport Layer Security，TLS）是为网络通信提供安全及数据完整性的一种安全协议。TLS与SSL在传输层对网络连接进行加密。

HTTPS：HTTPS（Hyper Text Transfer Protocol over Secure Socket Layer）是以安全为目标的HTTP通道，是HTTP的安全版，即在HTTP下加入SSL层。

SSL数字证书（SSL证书）：数字证书的一种，类似于驾驶证、护照和营业执照的电子副本。因为配置在服务器上，也称为SSL服务器证书。SSL 证书就是遵守 SSL协议，由受信任的数字证书颁发机构CA（如GlobalSign，wosign），在验证服务器身份后颁发，具有服务器身份验证和数据传输加密功能。

CA：即证书授权中心（CA, Certificate Authority）。CA是负责签发证书、认证证书、管理已颁发证书的机关。用户向CA提出申请后，CA负责审核用户信息，然后对关键信息利用私钥进行"签名"，并公开对应的公钥。客户端可以利用公钥验证签名。

CSR：CSR（Certificate Signing Request）即证书请求文件，也就是证书申请者在申请数字证书时由CSP（加密服务提供者）在生成私钥的同时也生成证书请求文件，证书申请者只要把CSR文件提交给证书颁发机构后，证书颁发机构使用其根证书私钥签名就生成了证书公钥文件，也就是颁发给用户的证书。

DV证书：域名级别验证的证书：Class 1。适合小型网站。

OV证书：企业身份验证的证书：Class 3。需要提交企业营业执照等企业有效资质。适合企事业单位。

EV证书：加强验证型证书：Class 4。需要提交企业营业执照等企业有效资质，进行最严格的验证。属于等级最高的SSL证书。浏览器地址栏标记为绿色并显示认证通过的企业名称。适合互联网公司、大型网站、政府、金融、保险等行业。

优势

知名品牌合作，安全可信赖

百度智能云SSL证书服务全面对接国际、国内最值得信赖的第三方数字证书颁发机构(CA)，确保数字证书认证可信力和证书加密强度、保障权益最大化，让用户真正安全可信赖。

免费版证书，0成本安全加密

百度智能云联合国内知名数字证书品牌TrustAsia（亚洲诚信）打造推出DV免费版本证书，更好的浏览器兼容性，更可靠的证书品牌，让SSL证书人人用得起，推动互联网实现全网加密。

全线HTTPS支持，轻松部署

百度智能云具备Web服务能力的全线产品，包括BLB、CDN、BAE、BCH等均已支持HTTPS服务，实现一键选择证书，自动化部署HTTPS服务，有效降低使用门槛。

一键申请，快捷高效

只需简单一步，即可完成证书申请提交、CSR生成、信息认证、所有者验证、证书签发等环节，全线上自动流程，DV证书分钟级签发，0技术基础用户也能快速申请，大幅提升申请效率。

统一管理，方便易用

申请或购买的SSL证书信息可自动导入到证书管理中心，并支持手动导入自有证书，方便用户对各级别、不同Web应用的证书进行统一管理，简化证书管理成本。

使用场景

企业网站安全加密

启用企业网站全站https安全加密，激活绿色安全标识(DV/OV)或地址栏企业名称标识(EV)，为潜在客户带来更可信、更放心的访问体验，极大增强企业诚信力和用户信赖感，有效提升成单率。

企业应用安全加密

越来越多的企业将OA、CRM、进销存、ERP、HRM等企业应用系统部署于云端，享受云计算的高效和便捷性。而升级为https安全加密，可进一步提升系统安全性，确保敏感信息不被劫持。

政务信息安全加密

公信力是政务平台要打造的最重要特性。而越来越多的钓鱼欺诈网站和信息劫持手段，对政务平台的信息安全带来严重威胁。启用权威认证的SSL证书能最大化保障信息安全和网站公信力。

支付体系安全加密

支付环节是用户最敏感也最容易受到安全威胁的部分，极易成为不法用户信息劫持和伪装欺诈的重要目标。因此，实现网站支付环节的https信息传输加密，已经成为各大网站的标配。

API接口安全加密

API接口是第三方网站进行信息交互的重要形式，因为大多涉及敏感信息或重要操作指令的传输，因此其安全性至关重要。使用SSL证书进行信息传输的高强度加密，可有效杜绝信息劫持。

产品定价

产品定价

计费概述

目前百度智能云现在提供DV、EV、OV型证书申请服务。证书价格请参考：CAS价格详情。

- 为了加强证书管理，确保免费版证书不被滥用，免费版TrustAsia证书不会发送证书信息到您的个人邮箱，您只可通过百度智能云的产品使用免费证书服务。
- 不同品牌OV/EV证书申请所需材料不同，详情参见各品牌OV/EV证书申请材料列表。

> 说明：
>
> - 展示价格为最低域名数量配置价格，可选择域名数量的证书价格依据实际购买域名数量， 按照多域名证书价格计算公式展示实际价格。

申请退款

百度智能云证书拥有无理由退款特权。在退款有效期内，用户可以通过提交工单的方式进行退款申请，为了保证退款快速受理，请务必在工单中说明需要退款的产品ID。如需退款，请点击申请退款。

证书退款期限

根据证书品牌的不同，申请退费的有效期限也不同。

证书品牌	退费期限
TrustAsia 付费证书	7天
CFCA	7天
SecureSite	7天
GlobalSign	7天
GeoTrust	7天
sslTrus	7天
TLC	7天
UniTrust	7天
Digicert	7天
TrustAsia 免费证书	不支持

> **注意：**
>
> - 证书退费期限指的是**证书签发后**的无理由退费期限。

操作指南

购前准备

证书品牌类型

目前，支持在百度智能云售卖的SSL证书品牌包括：

Globalsign：GlobalSign是一家声誉卓著，备受信赖的CA中心和SSL数字证书提供商，服务超过14万用户。GlobalSign所颁发的SSL证书具有2048位防解密，支持128位至256位SSL加密级别；兼容性高，通用所有浏览器、服务器及各式移动设备；支持通配型SSL证书、SANS证书及签名证书等特点。

CFCA：中国金融认证中心（CFCA）作为国内第一家与国外SSL服务器证书厂商媲美的电子认证服务机构，在国内银行业证书占有量第一。CFCA品牌的SSL证书严格按照国际标准提供电子认证服务，并结合我国国情，在密码算法、安全技术服务等方面兼容国际和国产算法。**政务用户请首选此品牌证书。**

SecureSite：SecureSite曾用名Symantec，是全球最大的信息安全服务商，被Digicert收购后品牌升级为SecureSite。该品牌为Digicert的原厂品牌，适用于对安全品牌有较高要求的用户。

GeoTrust：GeoTrust是Digicert旗下的性价比品牌。该公司各种先进的技术使得中小型机构和公司能安全地低成本地部署SSL数字证书和实现各种身份认证，从而确保数据传输的安全。

TrustAsia（亚数）：亚洲诚信是亚数信息科技（上海）有限公司应用于信息安全领域的品牌， 专业为各行业提供国际知名品牌数字证书及网络信息安全管理解决方案。TrustAsia品牌SSL证书市场占有率在国内保持领先地位。客户覆盖电子商务、互联网金融、银行及政府机构、保险证券、医疗机构、系统与软件开发商等各个领域。

TLC（泰尔认证）：泰尔认证SSL证书是一款由中国信息通信研究院全资子公司泰尔认证中心面向信息通信及工业互联网行业的众多客户提供的证书品牌，**欢迎政务用户选择此品牌证书。**

UniTrust（万维信）：万维信是一款由上海市数字证书认证中心自主研发的SSL证书，符合国内信息安全标准，确保客户的信息和审核数据不出境，**欢迎政务用户选择此品牌证书。**

sslTrus（锐安信）：锐安信是上海锐成信息的自研品牌，基于全球技术前沿的CA基础服务，为用户提供高性价比的SSL证书，**政务用户请避免选择此品牌证书。**

Digicert：DigiCert是一家具有创新性的企业，89%的《财富》500强公司和全球100家顶级银行中的97家银行选择DigiCert为他们的网络服务器和物联网终端进行身份验证和加密服务。

证书验证类型

百度智能云联合有资质的CA中心提供以下几种数字证书配置组合方案。

- **域名型DV**：DV证书属于域名验证型证书，支持线上5分钟内快速签发，无需人工干预。用户仅需对域名有权验证，即可签发证书。
- **企业型OV**：针对网站的域名及所有权进行严格的书面审查程序，通过审核后证书标示企业组织机构详情，让网友直接了解拥有该网站的企业真实身份，强化对该站的信任感。而网友也可于浏览器中发现「黄色小锁」符号已经激活。
- **增强型EV**：除了进行严格的网站所有权的真实身份验证之外，还加入第三方验证，证书标示增强组织机构详情，强化信任度，增强型SSL证书（EVSSL）最大的特色便是激活网站浏览器的栏位使其变成绿色。

域名数量类型

- 单域名版：此类产品仅支持一个标准域名。eg：baidu.com，不限制域名层级。
- 多域名版：一张数字证书同时保护多个域名、服务器，节省大量管理成本； 可以任意添加任何域名、子域名、IP 地址和本地服务器的组合，数量最多250个，每个证书品牌默认含有的标准域名数量不同，详见多域名版证书标准域名数量限制，申请证书时无需一次性提交，签发后可通过重新签发证书增加域名。
- 通配符域名版：一张数字证书下可同时保护同一个域下多个子域名，如为域名 *.baidu.com申请证书, 那么该证书支持主域名本身`\*.baidu.com`以及`a1. baidu.com`、 `a2.baidu.com`等子域名。

> **注意:**
>
> - 多域名版证书支持标准域名与通配符域名，如果用户选择了此类型则需要在购买信息中分别方分别选择两种域名的数量。
> - 一个通配符证书只能保护主域名的下一级子域名，且只有当主域名为一级域名时，证书才能同时保护主域名本身。如主域名为*.baidu.com时证书可以保护主域名本身，但当主域名为*.cloud.baidu.com时证书不能保护主域名本身。

多域名版证书标准域名数量限制

证书品牌	默认数量（个）	最低数量（个）
GeoTrust	5	5
TrustAsia	1	1
SecureSite	2	1
Globalsign	2	1
CFCA	2	1

> **注意：**
>
> TrustAsia、SecureSite（原Symantec）、Globalsign及CFCA品牌多域名证书的购买信息中，域名总数（标准域名数量与通配符域名数量之和）必须**不低于2**，当总数为1时将无法结算。

选择证书有效期

各个品牌的数字证书均有有效期年限限制：

除sslTrus支持购买1-5年外，

GlobalSign、SecureSite、GeoTrust、TrustAsia、CFCA、UniTrust、TLC、Digicert 等品牌的证书仅支持购买1-3年。

域名相关准备

1. 在申请证书前，用户应先确保已具备独立域名；如果没有域名，请先完成域名注册。
2. 申请付费版SSL证书时，将验证您的域名注册邮箱，因此请务必保证该邮箱真实、正确、有效。如果您不确定注册邮箱信息，可以到域名信息查询网站查询该域名WHOIS信息。

```
Registry Registrant ID:
Registrant Name: Zhiyong Duan
Registrant Organization: Beijing Baidu Netcom Science Technology Co., Ltd.
Registrant Street: 3F Baidu Campus No.10, Shangdi 10th Street Haidian District
Registrant City: Beijing
Registrant State/Province: Beijing
Registrant Postal Code: 100085
Registrant Country: CN
Registrant Phone: +86.1059924216
Registrant Phone Ext:
Registrant Fax: +86.1059927435
Registrant Fax Ext:
Registrant Email: domainmaster@baidu.com
Registry Admin ID:
Admin Name: Zhiyong Duan
Admin Organization: Beijing Baidu Netcom Science Technology Co., Ltd.
Admin Street: 3F Baidu Campus No.10, Shangdi 10th Street Haidian District
Admin City: Beijing
Admin State/Province: Beijing
Admin Postal Code: 100085
Admin Country: CN
Admin Phone: +86.1059924216
Admin Phone Ext:
Admin Fax: +86.1059927435
Admin Fax Ext:
Admin Email: domainmaster@baidu.com
Registry Tech ID:
```

如果您在注册时所留邮箱已经无法正常使用，请进入域名注册商管理控制台进行修改，或联系域名注册商协助变更。

> **注意:**
>
> - TrustAsia 域名型DV 单域名版证书为免费证书，申请时不需要验证邮箱。

购买证书

DV证书申请流程图如下：

> **注意：**
>
> DV免费证书现已支持下载。DV免费证书与收费证书对比
>
> 由于DV免费证书加密和验证方式有限，用户辨识度与网站官方可信度不足，易受到钓鱼网站利用危害正常网站经营者的利益，建议用户仅把免费证书用于测试环境，不建议用于生产环境。点此购买付费版DV证书

具体操作步骤分为以下几步：

第一：购买DV证书；
第二：提交信息；
第三：域名验证；

购买DV证书

1. 登录百度智能云官网，点击右上角的“管理控制台”，快速进入控制台界面，选择“产品服务>安全和管理>SSL证书服务”，进入“已购证书列表”。
2. 点击“购买新证书”按钮，进入“证书购买”页面。

3. 您可以根据自己的业务需求选择相应的SSL证书，完成支付并进入配置流程。

> **注意**
>
> - 单个多域名版证书订单内含的域名，主域名最多1个，标准域名和通配符域名总数不能超过249个，总订单不能超过250个。
> - 单个通配符版证书订单仅能支持1个通配符域名。

4. 同时，百度智能云为用户免费提供免费DV证书套餐，套餐默认配置如下：

项目	详情
证书品牌	TrustAsia
证书类型	域名型DV
产品类型	单域名版
购买时长	3个月

> **说明：**
>
> - 每个用户最多支持申请20张免费DV型证书（TrustAsia品牌），超过配额后您需要付费购买证书。
> - 域名型DV证书属于域名验证型证书，支持线上快速签发，且仅支持单域名。
> - 选择TrustAsia免费证书时会默认免费DV证书套餐中的配置，且不能修改。
> - 免费证书的加密等级较低，安全性相比付费证书来说较差；同时因OCSP服务器设置在国外，网站的首次访问速度将会稍慢一些。因此免费证书多作为测试使用，强烈建议您在商用时选择付费证书！

管理证书

1. 选择“产品服务>安全和管理>SSL证书服务”，进入“已申购证书列表”。
2. 已购证书中分为“超级SSL证书”及“普通SSL证书”两类。您购买的BaiduTrust证书将归类在“超级SSL证书”下，其它品牌的证书将归类在“普通SSL证书”下。
3. 点击“证书信息管理”可进入证书信息管理服务，该服务主要用于统筹管理百度智能云平台所购买的证书及其它服务商处购买的证书（需上传），此栏目下的证书可以推送到百度智能云的云资源下，无需下载和上传过程，以便于您更快捷的启用证书。

> 关于证书管理的详细介绍，请参看证书管理。
>
> 如果用户已经通过其它服务提供商申请了证书，也可以手动录入证书，关于证书的上传方法，请参看上传证书。

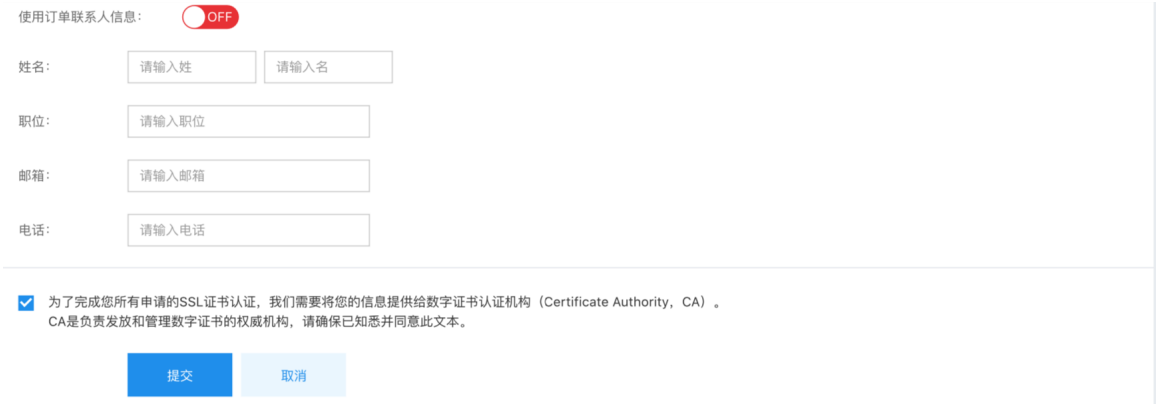
签发证书（DV）

DV证书申请

1. 完成证书购买之后，进入已购证书列表页面。
2. 根据产品ID、证书状态等信息选择需要填写申请信息的证书，在相应的操作栏下点击**证书申请**。
3. 根据提示填写相应申请资料，填写方式可选择 **新建表单** 或 **使用已有的CSR**。

- 新建表单方式


温馨提示：提交后不可修改，请谨慎填写。 DV型证书签发流程详解>>
填写
*填写方式：
新建表单
使用已有的CSR
域名信息
产品ID：
证书类型：
BaiduTrust 域名型DV 多域名版
证书有效期：
1年
*主域名：
请输入标准域名
*多域名：
请输入域名，多个域名用换行隔开
*验证方式：
DNS验证
验证方式
公司信息
公司名称：
请输入公司名称
部门：
请输入部门
城市：
中国
北京市
北京市
详细地址：
请输入详细地址
邮政编码：
请输入邮政编码
联系电话：
请输入电话(座机号码有利于加速申请)
温馨提示：
请输入公司的真实信息，与工商部注册的地址保持一致可以加速数字证书认证机构（Certificate Authority，CA）审核。
订单联系人信息
姓名：
请输入姓
请输入名
职位：
请输入职位
邮箱：
请输入邮箱
电话：
请输入电话
温馨提示：
证书审核过程中，CA会需要联络订单联系人进行信息核验。请保证邮箱地址和联系电话能够联系到订单联系人。

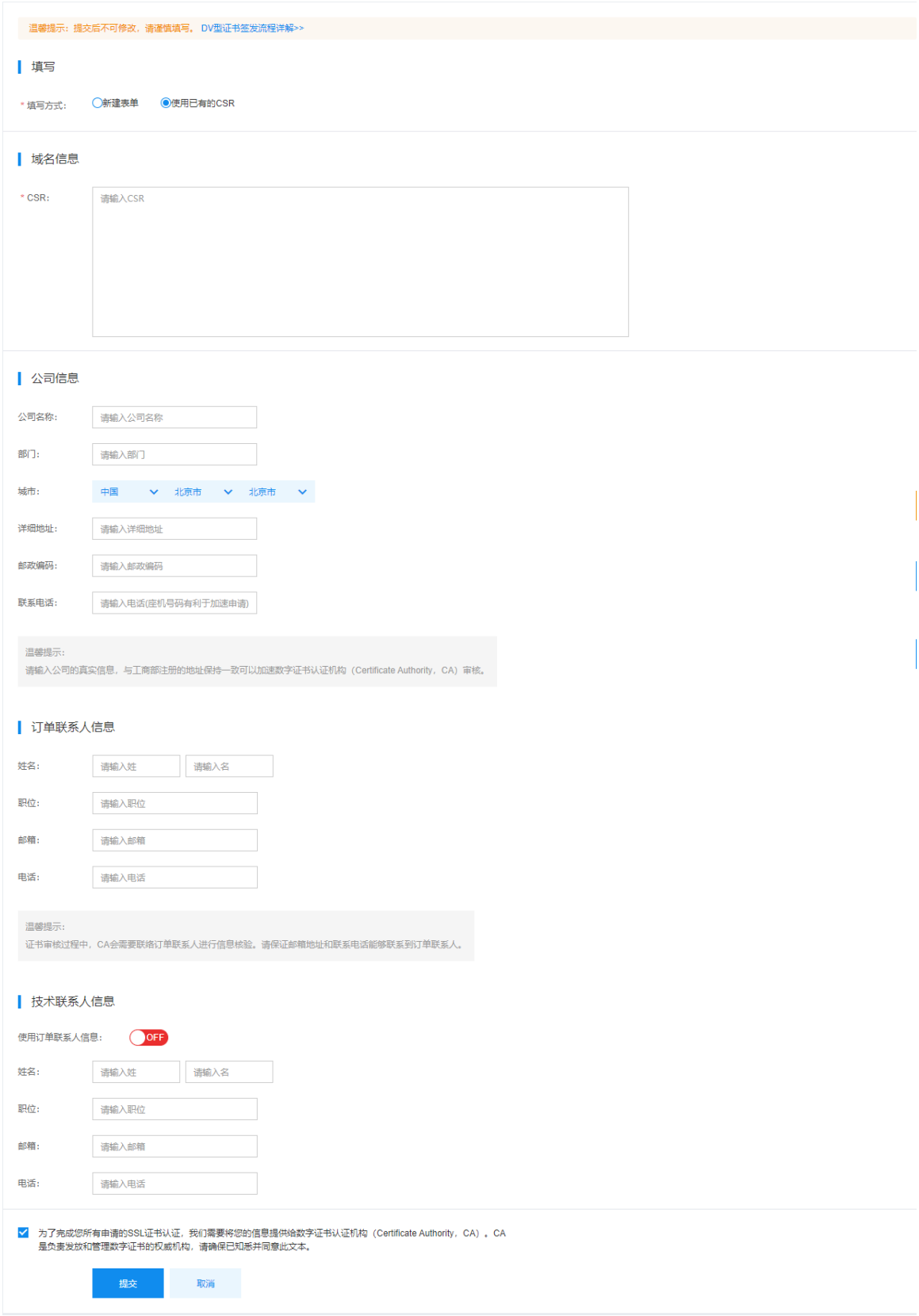


注意：

- 域名为单域名，必填项。
- 企业信息、订单联系人信息，技术联系人信息为选填项。
- 域名信息部分需要选择域名验证方式，如下图所示

- 使用已有的CSR方式

CSR是证书申请者在申请数字证书时由CSP（加密服务提供者）在生成私钥的同时也生成证书请求文件，用户可以使用其已有CSR进行证书申请，同样可以实现证书快速签发。



域名验证

完成上述信息填写后，进入域名验证环节，域名验证方式在域名填写时选择。

1. DNS验证

   - 1.1 如果您的域名是在百度智能云注册的域名，系统将自动添加指定的DNS解析记录，记录被检测成功后，验证自动完成。
   - 1.2 若您的域名为非百度智能云注册的域名，请按照控制台提示文字（如图），前往域名注册商添加对应的解析记录：

- 1.3 若您无法使用DNS验证方式， 可尝试将域名转入百度智能云，以享受更完备的解析服务。

2. 文件验证

   按指定文件目录、文件名、文件内容新增文件例如

名称	详情
待验证域名:	ttany.baidu.com
验证文件路径:	/.well-known/pki-validation/fileauth.txt
验证文件值:	201708070854422nuyjdq5xdpie8xze7uz23qfohtutkqz7jqzws7pgobkmryhqp

   如果申请文件验证的域名是ttany.baidu.com，则进行访问的链接地址为：http://ttany.baidu.com/.well-known/pki-validation/fileauth.txt或者https://ttany.baidu.com/.well-known/pki-validation/fileauth.txt

   如果申请文件验证的域名是泛域名*.baidu.com，那么进行验证访问的链接地址是 http://baidu.com/.well-known/pki-validation/fileauth.txt或者 https://baidu.com/.well-known/pki-validation/fileauth.txt

   访问链接可获取到内容为201708070854422nuyjdq5xdpie8xze7uz23qfohtutkqz7jqzws7pgobkmryhqp

完成域名验证后，DV证书自动签发，其申请流程完成。

> **注意：** 文件验证步骤中的文件验证路径不能更改，如因路径中的特殊符号造成验证流程无法继续，请选择DNS验证方式。

查看域名验证结果（以TXT解析类型为例）

DV域名签发速度较快，如果申请时间较长仍未签发，请通过以下方法校验域名验证结果。如域名验证已成功，请耐心等待。

运行dig 主机记录 命令进行检测或者运行dig 主机记录 @8.8.8.8命令指定使用谷歌DNS进行验证。

例如：

```
dig txt baidu.com @8.8.8.8
```

```
baidudeMacBook-Pro-10:~ $ dig txt b @8.8.8.8

; <<>> DiG 9.10.6 <<>> t. @8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36186
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;b com.                  IN      TXT

;; ANSWER SECTION:
b                       119     IN      TXT     "20170802113925588 6f1hox2i99gd6e36hfy76b87g7tcq0hrk8ijwh55eokg911

;; Query time: 104 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Thu Mar 19 15:57:58 CST 2020
;; MSG SIZE  rcvd: 154
```

- 如果返回结果中存在类似图示中的TXT记录，且记录值与DNS设置

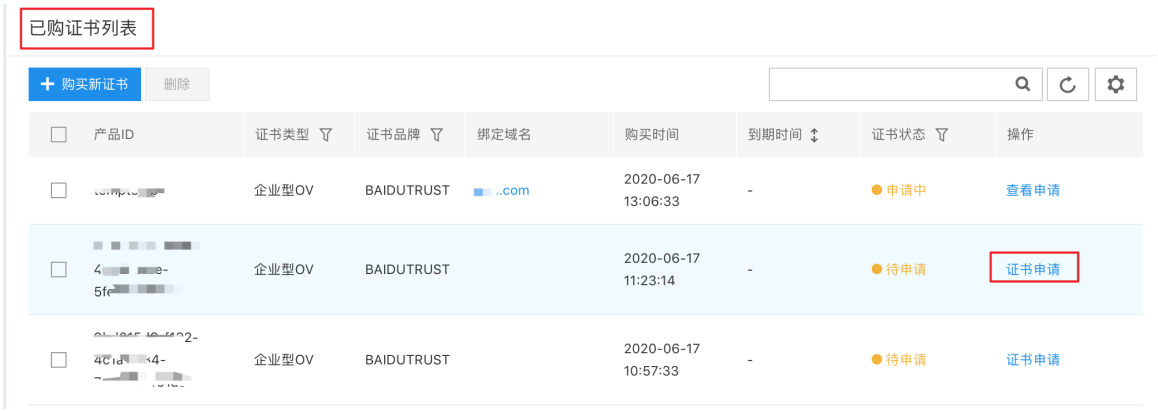
值20170802113925588 6f1hox2i99gd6e36hfy76b87g7tcq0hrk8ijwh55eokg911一致，表示您的DNS配置正确且已生效；如果记录值不同，请使用该值在您的DNS域名解析服务商处更新该记录值。

- 如果返回结果中不存在TXT记录，可能是DNS解析配置有误或者配置未生效。如DNS解析配置错误，请使用该值在您的DNS域名解析服务商处更新该记录值；如果配置长时间未生效，请联系您的域名托管商。

签发证书（OV与EV）

OV/EV证书申请

1. 完成证书申购之后，进入已购证书列表页面。选择要申请的证书资源，点击“证书申请”。



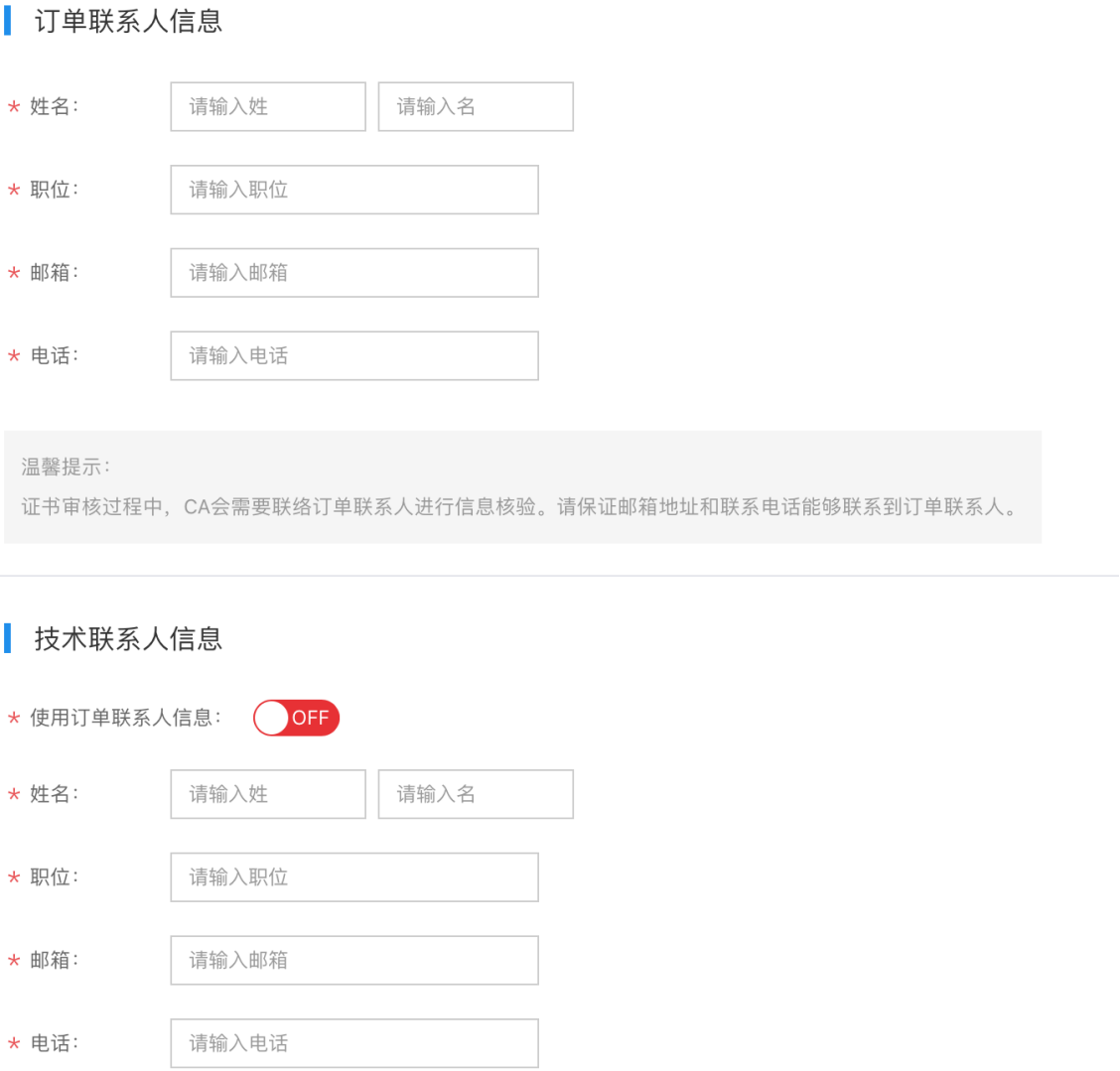
2. 进入表单填写页，根据提示填写相应申请资料，填写方式可选 **新建表单** 或 **使用已有的CSR**。

- 新建表单方式

- 使用已有的CSR方式

  CSR是证书申请者在申请数字证书时由CSP（加密服务提供者）在生成私钥的同时也生成证书请求文件，用户可以使用其已有CSR进行证书申请，同样可以实现证书快速签发。


填写
*填写方式：
新建表单
使用已有的CSR
如果已有证书CSR，可选择使用已有的CSR，粘贴CSR申请暂不支持证书下载和自动导入证书管理中心。
域名信息
*CSR：
请输入CSR
CSR
*多域名：
请输入域名，多个域名用换行隔开
*验证方式：
DNS验证
验证方式
公司信息
* 公司名称：
请输入公司名称
* 部门：
请输入部门
* 城市：
中国
北京市
北京市
* 详细地址：
请输入详细地址
* 邮政编码：
请输入邮政编码
* 联系电话：
请输入电话(座机号码有利于加速申请)
温馨提示：
请输入公司的真实信息，与工商部注册的地址保持一致可以加速数字证书认证机构（Certificate Authority，CA）审核。



表单填写说明：

1. 申请证书的域名请确保状态正常，如域名状态未过期，未被停止服务，已通过实名认证，所有者归属权清晰等。
2. 表单中需要用户填写设置的密码为用户部署证书时所需要用到的密码，该密码非常重要，若遗忘只得通过工单方式申请证书重新颁发。请务必保管好签发密码。
3. 请按照提示和要求填写所有必填信息，包括申请域名、密码、公司信息、订单联系人和技术联系人的相关信息。信息一经提交不可修改，请谨慎填写。
4. 公司信息最好和财务年报的信息保持一致，如此有利于加速CA审核。例：GlobalSign OV型证书，若订单信息中的固定电话和公司的财务年报中的电话信息一致，且可以通过此电话号码联系到申请人，即可免去确认函盖章上传环节，加速证书签发。
5. 证书人工审核过程中，审核部会通过订单联系人电话联络确认信息，或要求通过电子邮箱发送补充文件，联络不到申请人会导致签发延时，请确保联络方式畅通。

3. 域名验证完成后，CA审核机构会拨打申请公司的认证电话号码进行信息核验。公司的认证电话号码通常是由工商局登记，公司对外公布年报等登记的电话号码。请确保公司认证电话可以联络到订单联系人，或有能力进行核验电话的接听。若不行，请参考步骤4。
4. 在平台下载确认函模板，填好信息后与您的审核人员进行核对后，上传至平台。以上操作都有CA审核人员通过订单联系人的邮箱或电话协助您完成。

申请材料下载上传说明

1. 确认函上有公司信息和联系人信息等，请确认信息无误后加盖公司印章（公章/合同章/财务章/项目章均可，印章需清晰显示公司全名），然后上传完成确认函的扫描件即可。
2. 扫描件大小请勿超过2M，格式不限。
3. 不同的品牌对材料的要求不同，以下载的材料包为准。各品牌OV/EV证书申请材料列表参见列表

> 注意： 订单信息中的固定电话和公司的财务年报中的电话信息一致，且可以通过此电话号码联系到申请人，即可免去确认函盖章上传环节。

5. 人工审核。人工审核过程根据品牌不同，需要2-5个工作日。用户需确保订单信息中的电话号码保持畅通，留意收到的邮件。若错过电话或忽略邮件通知，会延长审核时间，可发工单申请重新联络。
6. 证书成功签发。用户可在线下载证书。

证书申请常见问题

- DV SSL证书申请需要多久？
- 申请了主域名SSL证书，是否还需要申请www域名的？
- 为什么会出现安全审核失败？
- 常见的证书申请状态有哪些？
- CNAME冲突如何解决？
- 各品牌OV/EV证书申请材料都有哪些？

重新签发证书

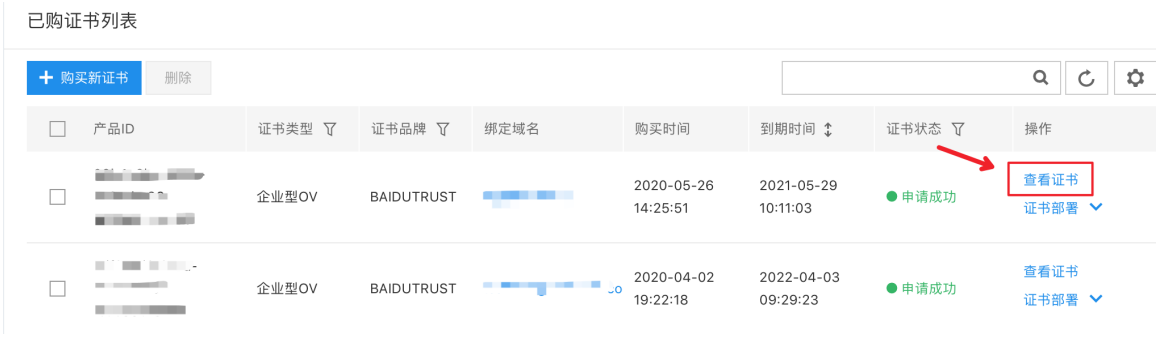
应用场景

证书重新颁发功能适用于以下场景：

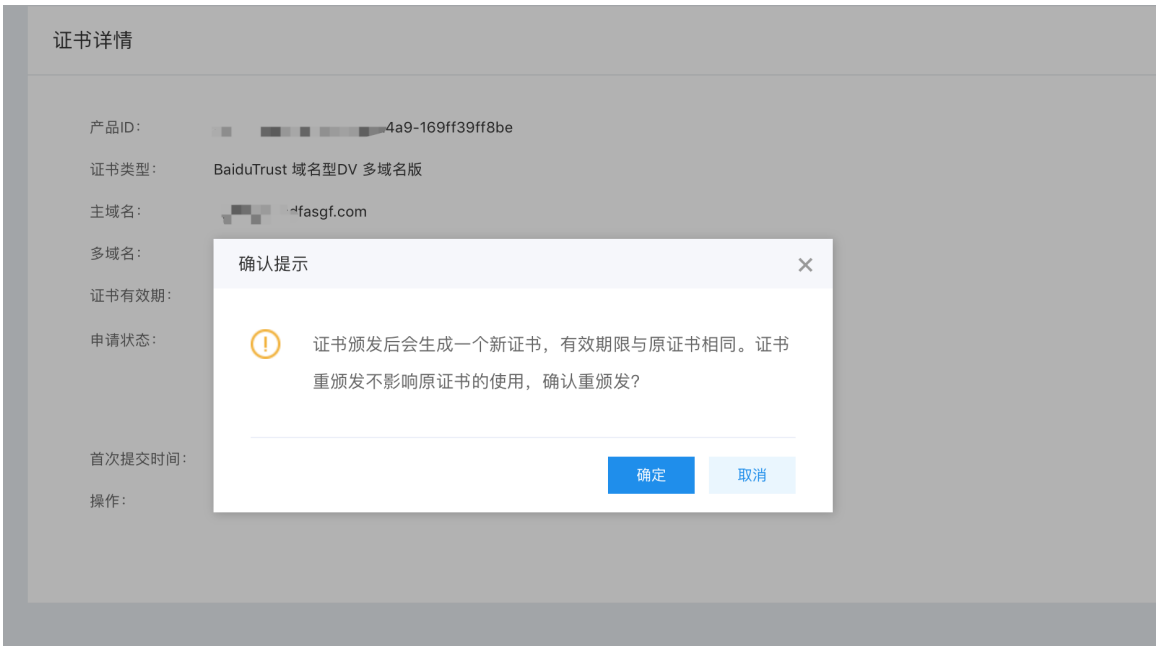
1. 多域名证书补充域名。
2. OV-EV证书忘记密码需要修改密码时。

操作步骤

1. 登录百度智能云控制台，选择“产品服务 > SSL证书服务 > 已购证书列表”。
2. 选择需要重新颁发的证书，点击**查看证书**（BaiduTrust证书为“管理证书”）。



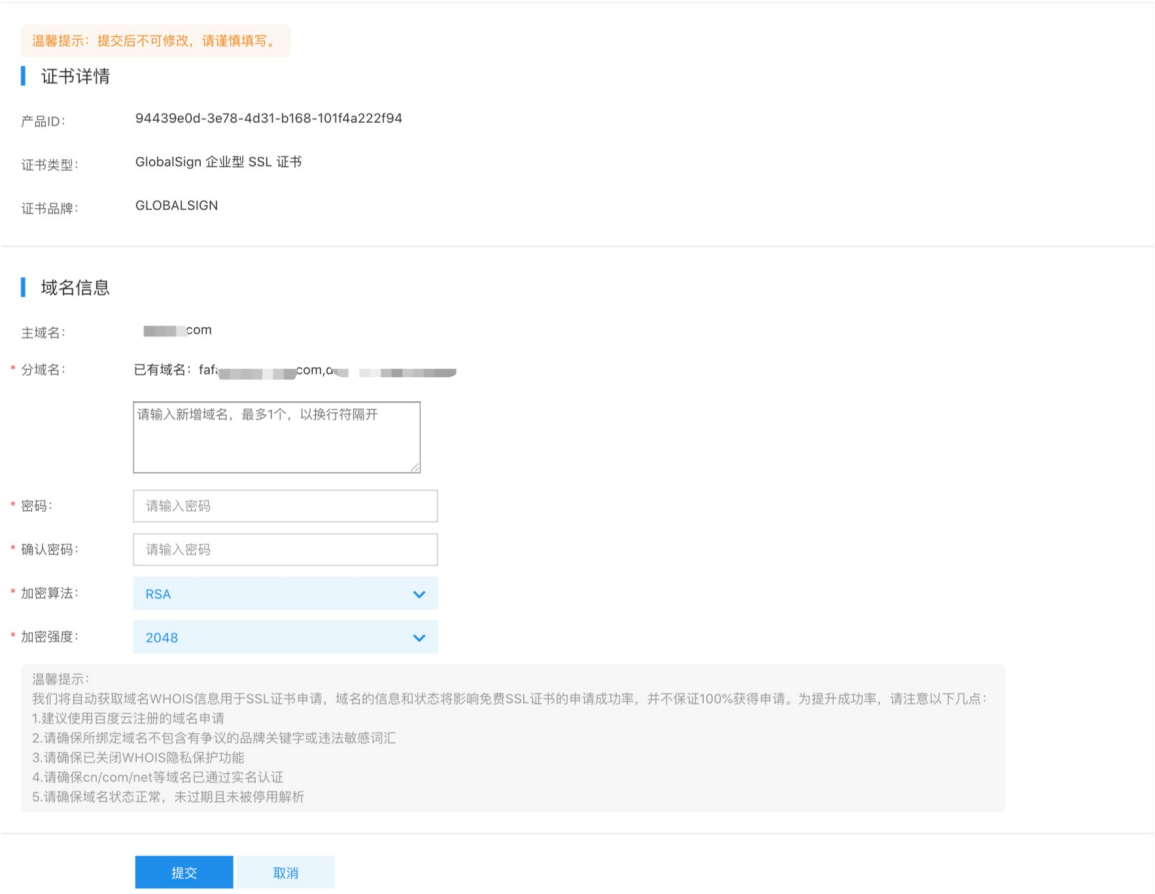
3. 选择**证书重新签发**，并确认重新颁发。



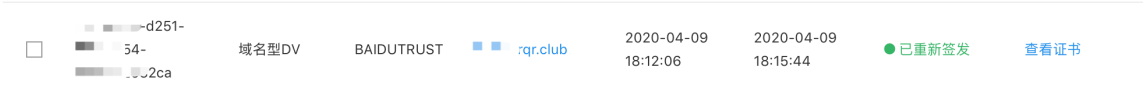
4. 在域名信息中输入需要新增加的域名，如有多个域名可以通过“回车键”换行添加。完成后点击提交，证书状态将变为**重新签发中**。

   - DV证书重新颁发只需要输入新增加的域名即可。

- OV/EV证书需要输入新的密码及加密算法。



5. 在已购证书证书列表中可查看重新签发中的证书。重新签发成功后的证书状态将变为**已重新签发**。



> **注意：**
>
> 1. 重新签发证书需要前往域名注册商添加TXT解析记录完成域名验证。
> 2. 百度智能云域名将自动添加解析记录，无需手动添加验证。
> 3. 若您无法使用我们提供的验证方式，可尝试将域名转入百度智能云，参考域名转入。

部署证书

百度智能云内自动部署

证书申请后，将自动以高度加密形式导入到“证书信息管理”服务中。您可通过不同服务的证书选择功能选择对应证书，快速部署服务。

> **说明：** 为了加强证书管理，确保免费版证书不被滥用，免费版证书不会发送证书信息到您的个人邮箱，您只可通过百度智能云的产品使用免费证书服务。

目前支持证书部署的产品服务有BCH、CDN、BLB、BAE专业版和LSS等，关于HTTPS具体部署方法，请查看对应产品的操作指导：

- BCH
- CDN

- BLB
- BAE专业版
- LSS

手工部署或部署到其它位置

针对不同平台OV/EV证书部署方法不同，如Apache、Nginx和Tomcat等平台部署方法请参见：

- 在IIS服务器上安装SSL证书
- 在Nginx或Tengine服务器上安装证书
- 在Apache服务器上安装SSL证书
- Tomcat服务器安装SSL证书

续费证书

概述

您可通过本文档了解百度智能云平台证书即将到期时，如何进行操作续费。

SSL证书到期前若不及时替换新证书，一旦过期会导致网站出现不安全提示、站点无法访问等。百度智能云现提供证书续费及提醒功能，具体注意事项及操作流程如下：

证书续费须知

1. 续费时间：证书在**到期日前1个月**，管理控制台证书列表会开放续费入口，您可选择相应待续费证书进行续费。
2. 流程简化：证书续费无需重新填写原申请信息（包括域名及联系人信息等），您只需下单支付成功后，根据控制台提示完成新证书的域名验证及审核流程。
3. 时长增补：到期日前1个月内续费证书，除正常续费时长外，原证书到期前剩余时长会补偿到新证书中，可有效保障证书续费而不损失原证书时长。

例如：原证书到期时间为2020年9月10日，您在9月1日操作续费1年，9月3日签发成功，则新证书有效时间为2020年9月3日到2021年9月10日。

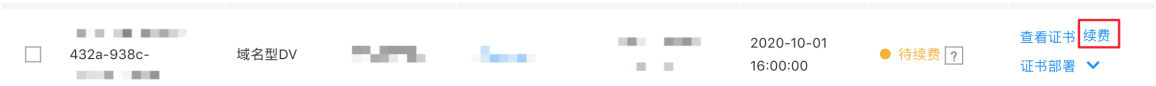
> **注：免费证书续费不支持赠送时长。**

4. 信息一致：续费证书域名及证书品牌型号将与原证书完全一致。如需更改请重新购买证书。

续费证书

1. 查看待续费证书

登录百度智能云证书控制台，定位到待续费的证书，点击右侧**续费**。



2. 确认订单信息

1）在信息确认页面，确认**域名**及选择**域名验证方式**。

2）在公司信息和联系人处，点击右侧**展开**可确认或修改联系人信息。

3）选择续费年限，点击**下一步**完成支付。

4）支付成功后，控制台会新增一张证书记录，状态为**申请中**。

5）返回证书列表页面，定位到新证书，点击右侧**查看申请**。

3. 域名验证与CA审核

DV证书仅需验证域名即可完成签发，OV/EV证书除域名验证外，还需验证企业信息，验证成功后即可完成签发。

1）续费证书域名验证流程与新购证书验证方法一致，请参考域名验证方式

2）完成域名验证后，可返回**查看申请**页面，使用**校验工具**检测域名是否解析成功。**解析成功后DV证书即可签发。**

3）OV及EV证书在完成域名验证后，还需电话验证企业信息。

BaiduTrust OV/EV证书流程参考：OV证书验证/EV证书验证

4）完成以上步骤，即可等待CA机构完成审核并签发。

> **DV证书最快5~10分钟完成签发；OV和EV证书最快2个工作日完成签发。**

5）完成证书签发后，可在**证书详情**页面下载最新证书，重新部署在服务器上。

BaiduTrust云端配置

区别于传统SSL证书的下载自行部署方式，BaiduTrust品牌证书提供“云端配置”能力，实现免部署、网站访问加速、搜索收录加速、搜索权重提升等优势权益。

> 注：如果您采用了传统手工下载后自行部署的安装方式，则无法享受我们提供的这些增值服务内容。

如果您期望使用这些能力，可以进入“管理证书”后，点选左侧的“云端配置”，下图示意“1”的位置：

- 位置2：多域名下拉切换
- 位置3：当前域名的启动和停止控制
- 位置4：详细配置项目

常规配置

基本信息

- CNAME地址：您需要将您当前域名DNS解析地址修改为此地址
- 源站地址：请在此处设置您网站原本的IP地址或者CNAME地址 以上两项设置完成后，则已具备免部署的条件，及享受其它BaiduTrust证书提供的优势权益，您可以根据需要使用顶部的开关按钮，启动和停止服务。

> **如何理解上面两个地址的用途呢？**
>
> 举例来讲，假如您的服务器地址为`10.10.10.10`，域名是`test.com`，您的域名`test.com`如果当前访问正常的话，应该是直接指向了`10.10.10.10`，那么：
>
> **【不推荐】**在传统方式下，您应该下载证书文件后，在`10.10.10.10`的服务器上安装证书文件，才可以正常使用证书提供的安全能力。
>
> **【推荐】**在新的免部署方式下，您**无需**下载文件及进行安装。仅需要将您的域名`test.com`修改解析指向至本功能中`CNAME地址`上（以实际为准），以及将您的服务器地址`10.10.10.10`设置入本功能中`源站地址`上（以实际为准）。

用户流量

因BaiduTrust证书赠送网站加速服务，网站加速将以流量计费，此处将显示您网站剩余的可用高速流量，超出后网站加速速率将无法保证，您可以充值高速流量，以继续享受更快的网站加速效果。

> **关于回源与缓存策略的说明**
>
> - 云端配置中默认不做任何缓存策略的设置，但会遵循源站的缓存规则进行缓存
> - 如您希望进行缓存，请在源站Response Header中增加Cache-Control或者Expires相关字段的设定

TLS版本控制

此处可以切换设置您的站点当前所使用的TLS版本。

OCSP Stapling

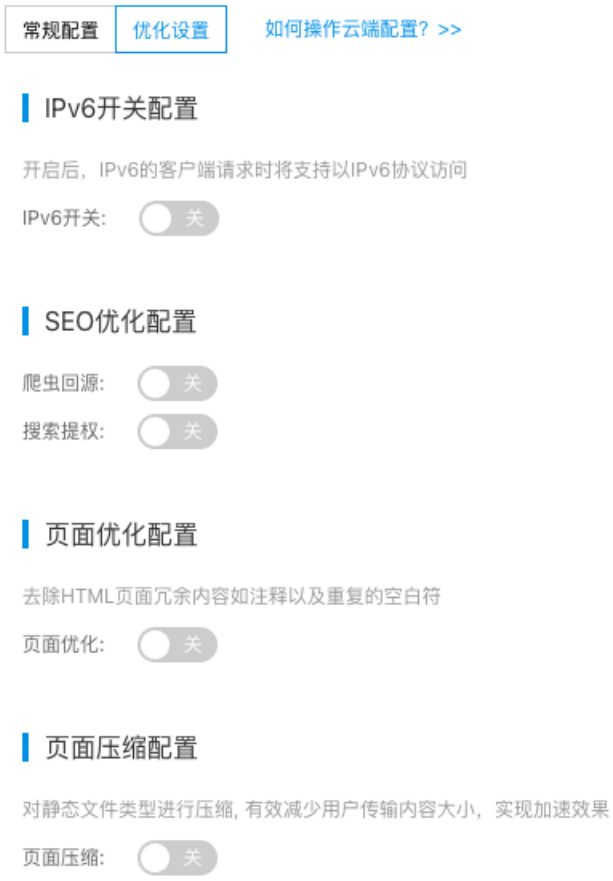
OCSP Stapling功能是由CDN服务器查询OCSP（Online Certificate Status Protocol）信息，可以降低客户端验证请求延迟，减少等待查询结果的响应时间。此处可以开启或关闭OCSP Stapling。

QUIC协议配置

此处可以开启或关闭QUIC协议。

优化设置

您可以根据提示选择是否开启当中的服务。



BaiduTrust签发证书

EV证书验证与签发

EV证书验证与签发

具体流程如下：

1. 选择待申请的证书，完成 证书申请。
2. 完成 域名验证。
3. 完成 Agreement填写
4. 完成 企业信息验证

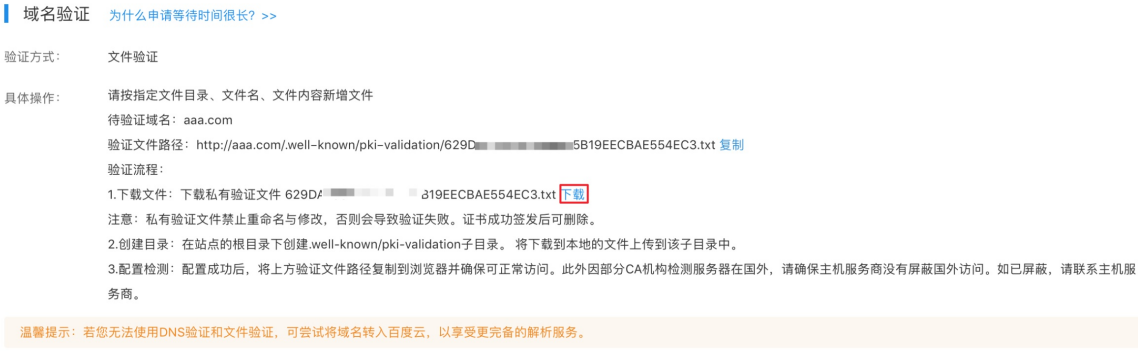
5. 等待签发成功。

EV证书申请

1. 进入已购证书列表页面选择待申请的EV证书，点击 **证书申请**。

2. 在申请页面填写域名并选择域名验证方式，Baidu Trust证书默认可选 DNS验证与 文件验证 两种方式。

3.点击 **提交** 完成证书申请，继续对域名进行验证。



EV证书域名验证

EV证书域名验证方式与DV证书验证方式一致，请参见 DV域名验证。

EV证书签署协议

EV证书在域名验证完成后，CA会发送Agreement协议邮件至申请人邮箱，可选择方便快捷的线上填写，点击邮件中的链接，输入邮件中的验证码即可。

> **注意：**邮件验证码有效期为72小时，请及时签署。 **协议签署规范：**

协议链接打开之后是需要填写如下信息的，与订单中填写信息保持一致即可：

1）First Name（名字）

2）Surname（姓氏）

3）Email（邮箱）

4）Contact Phone No（电话）

5）Job Title（职位）:需填写具体职位，如IT。

EV证书企业信息验证

1.在签署协议后，CA将会验证企业信息，企业信息验证参见 OV企业信息验证。

2.EV证书只能使用第三方平台的企业信息完成企业验证，不能使用企业域名后缀邮箱验证。

EV证书签发

完成上述域名验证、协议签署及企业信息验证，EV证书即可签发。

OV证书验证与签发

OV证书验证与签发

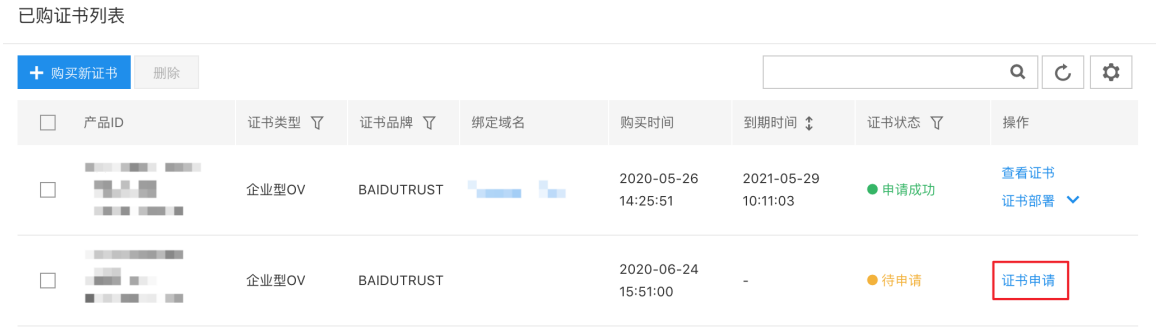
Baidu Trust OV证书需验证域名与企业信息，验证成功后可在1～3个工作日签发。

具体流程如下：

1. 选择待申请的证书，完成 证书申请。
2. 完成 域名验证。
3. 完成 企业信息验证。
4. 等待签发成功。

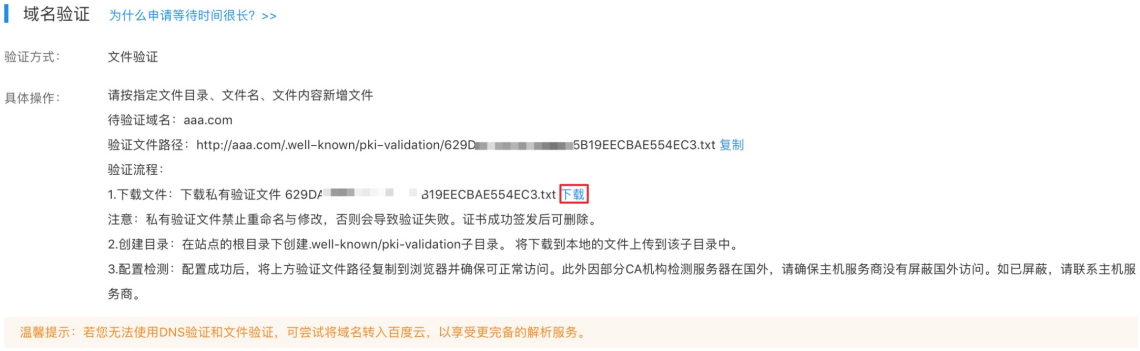
OV证书申请

1. 进入已购证书列表页面选择待申请的OV证书，点击 **证书申请**。



2. 在申请页面填写域名并选择域名验证方式，Baidu Trust证书默认可选 DNS验证与 文件验证 两种方式。

3.点击 **提交** 完成证书申请，继续对域名进行验证。



OV证书域名验证

OV证书域名验证方式与DV证书验证方式一致，请参见 DV域名验证。

OV证书企业信息验证

1. OV证书提交订单时请尽量提供企业域名后缀的邮箱。（如申请证书的域名为：baidu.com，则尽量提供xxx@baidu.com这样的邮箱）
2. 如提交订单时提供了企业域名后缀的邮箱，在域名验证完成后，CA机构会发送确认邮件至该邮箱，请收到邮件后回复“同意”，完成验证。
3. 如提交订单时提供的邮箱为qq/163/126/gmail/hotmail等免费邮箱，则CA机构将使用第三方平台查询到的电话或邮箱完成验证。
4. 关于电话验证：电话验证是CA机构联系申请证书的企业，确认是否申请证书、公司信息、申请域名等的一个验证环节。如果无法进行电话验证或者企业信息有误，可能会导致证书签发延迟，甚至被拒签。 关于电话验证内容：CA需要确认申请证书的企业是真实存在且合法经营的，此外CA要确保证书是该企业真实申请，而不是非法人员冒充申请。
5. 关于验证电话：CA机构会选择以下第三方平台信息进行电话验证

企查查：https://www.qcc.com/

爱企查：https://aiqicha.baidu.com/

114百事通：http://www.114best.com/

6. 请确保以上任一平台企业电话可用，如发生变更或失效，请及时修改更正。
7. 请确保企业电话可联系到订单申请人或电话接听人员知晓证书申请事宜，并配合完成验证。
8. 如电话确认无误及可用，CA将会在工作日24小时内（8:00am～16:00pm）拨打电话验证，请在此期间保持企业电话畅通。
9. 完成上述域名验证与企业电话验证之后，OV证书即可签发。

DV证书验证与签发

DV证书验证与签发

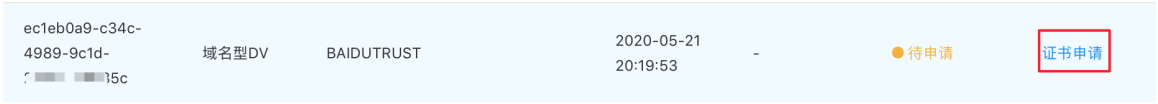
BaiduTrust DV证书仅需验证域名，验证成功后证书将在10分钟左右签发。

具体流程如下：

1. 选择待申请的证书，完成 证书申请。
2. 完成 域名验证。
3. 等待签发成功。

DV证书申请

1. 进入已购证书列表页面选择待申请的DV证书，点击 **证书申请**。



2. 在申请页面填写域名，并选择域名验证方式，Baidu Trust证书默认可选 DNS验证与 文件验证 两种方式。

3.点击 **提交** 完成证书申请，继续对域名进行验证。

DV证书域名验证

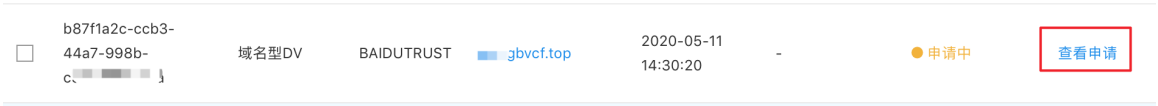
DNS验证

如绑定域名为申请者本人所有，则推荐选择DNS验证方式，系统会自动判断所提交的域名。

- 如果证书绑定域名在百度智能云账号内，则系统会自动添加一条CNAME解析记录，您只需等待5～10分钟后查看证书是否签发成功即可，除此之外无需任何操作。
- 如果证书绑定域名在非百度智能云账号内，则需前往该域名注册商处手动添加一条CNAME记录。

以百度智能云页面为例（不保证所有平台流程相同，如有疑问请咨询域名所在的注册商平台）具体操作如下：

1. 在管理控制台已购证书列表定位申请中证书并查看申请。



2. 复制保存名称与设置值（请确保复制完整）。

3. 进入域名管理列表操作解析，需确保所解析域名与证书绑定域名一致。

4. 点击左上角 **添加解析**。

5. 完成配置项填写，并点击 **确定**。

配置项说明：

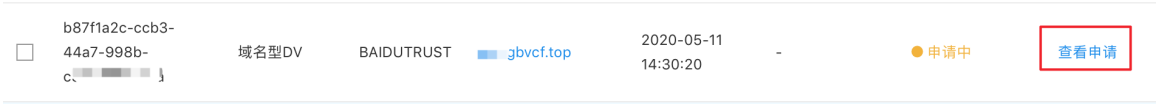
配置项	配置说明
主机记录	• 绑定域名为主域名，主机记录：名称.主域名（仅复制名称即可，无需手动加.） • 绑定域名为二级域名，主机记录：名称.二级域名前缀.主域名（复制名称+.二级域名前缀） > 百度智能云默认无需填写主域名，如其他注册商平台未设置域名，请手动添加（例如名称为：abcd，主域名：baidu.com，则完整主机记录为：abcd.baidu.com；如申请域名为二级域名，如cloud.baidu.com，则完整主机记录为abcd.cloud.baidu.com）
记录类型	cname记录
解析线路	默认
记录值	完整复制 **设置值** 即可
TTL	默认

文件验证

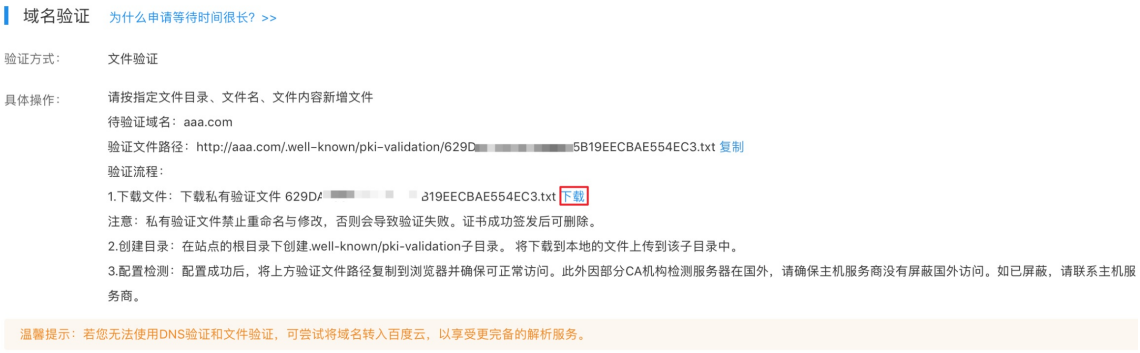
如您非域名所有者但有服务器管理员权限，则可选择文件验证。

具体操作如下：

1. 在管理控制台已购证书列表定位申请中证书并查看申请。



2. 在证书申请页的 **域名验证** 处，点击 **下载**。



3. 在站点的根目录下创建 `.well-known/pki-validation` 子目录。注意第一层目录是带点的隐藏目录，Windows下命令为：md ".well-known"。 将TXT文档上传到该子目录中。如果您的站点由于某种原因无法创建隐藏目录，选择其它DNS验证方式。

4. 上传成功后可自行检测配置

检测链接：`http://域名/.well-known/pki-validation/文件名.txt`

1）请确保主机服务商没有屏蔽国外访问。如已屏蔽，请联系主机服务商。

2）为确保国外CA正常访问域名，请添加以下IP为白名单（如无IP控制，请忽略）：

91.199.212.132

91.199.212.133

91.199.212.151

91.199.212.176

> **注意： DV证书需要等待10～15分钟完成验证，如较长时间未签发，请提交工单咨询；OV/EV证书除域名验证外还需配合完成企业信息验证。**

多用户访问控制

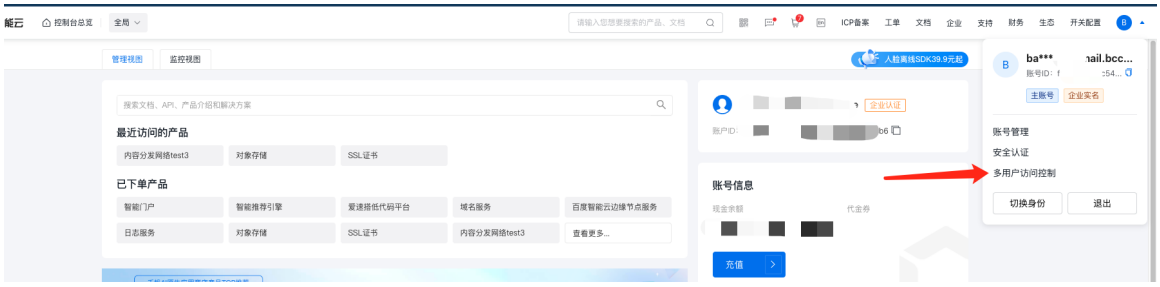
介绍

多用户访问控制，主要用于帮助用户管理云账户下资源的访问权限，适用于企业内的不同角色，可以对不同的工作人员赋予使用产品的不同权限，当您的企业存在多用户协同操作资源时，推荐您使用多用户访问控制。 适用于下列使用场景：

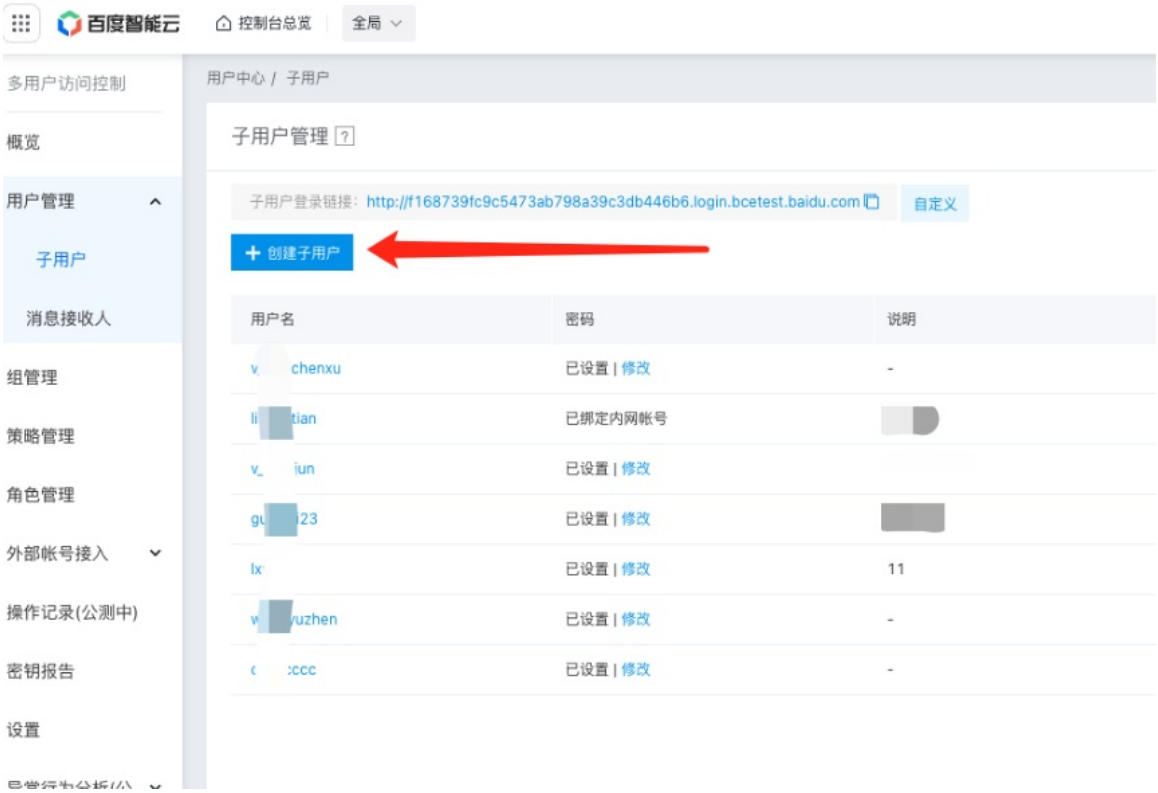
- 中大型企业客户：对公司内多个员工授权管理
- 偏技术型vendor或SAAS的平台商：对代理客户进行资源和权限管理
- 中小开发者或小企业：添加项目成员或协作者，进行资源管理

创建用户

1.主账号用户登录后在控制台选择“多用户访问控制”进入用户管理页面。



2. 在左侧导航栏点击“用户管理”，在“子用户管理列表”页，点击“新建子用户”（如子用户已存在则跳过新建子用户这一步）。填写创建子用户的表单



3. 在弹出的“新建子用户”或编辑已有子用户权限对话框中，完成填写“用户名”和确认，返回“子用户管理列表”区可以查看到刚刚创建的子用户。

配置策略

SSL证书支持系统策略和用户自定义策略两种，分别实现对SSL证书的产品级权限和实例级权限控制。

- 系统策略：SSL证书服务为管理资源而预定义的权限集，这类策略可直接为子用户授权，主账户或子用户账户管理员只能使用而不能修改。
- 自定义策略：由用户自己创建，更细化地管理资源的权限集，可以针对单个实例配置权限，更加灵活地满足账户对不同用户的差异化权限管理。

系统策略

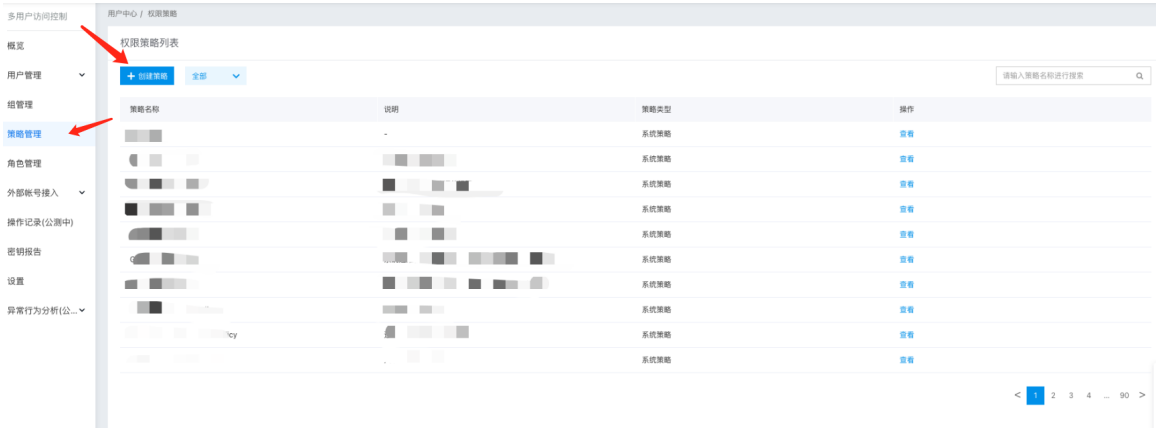
系统策略包含管理权限、运维权限和只读权限3种策略，权限范围详细如下：

策略名称	权限说明	权限范围
CASFullControlPolicy	完全控制管理SSL证书的权限	拥有购买SSL证书、删除SSL证书的权限以及CASOperateAccessPolicy的所有权限
CASOperateAccessPolicy	运维操作SSL证书的权限	拥有申请SSL证书、取消SSL证书申请以及CASReadOnlyAccessPolicy的所有权限
CASReadOnlyAccessPolicy	只读访问SSL证书的权限	拥有查看SSL证书列表、查看SSL证书详情、查看SSL证书申请、下载SSL证书的权限

自定义策略

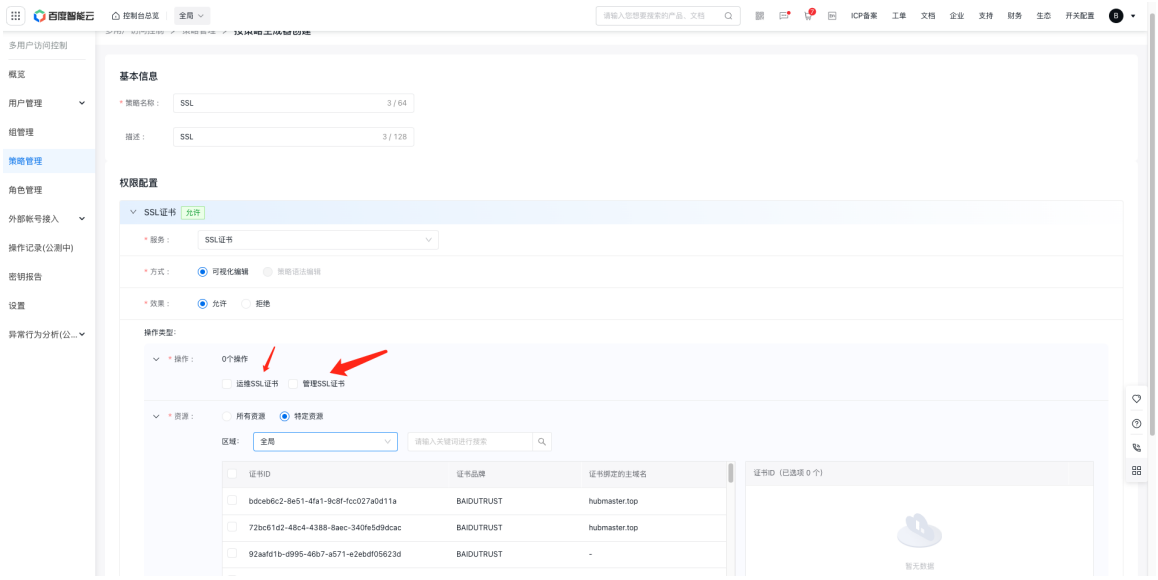
自定义策略是从实例维度进行授权，与系统策略不同，只对选定的实例生效。

用户先通过左侧导航栏进入【策略管理】，然后点击“创建策略”，用户填写策略名称并选择服务类型为【SSL证书】，其中策略生成方式默认为策略生成器，不需要修改。



点击【按策略生成器创建】

填写好策略名称后，服务选择**SSL证书**，效果选择**允许**，操作可以选择运维SSL证书、管理SSL证书，分别对应上表中的CASOperateAccessPolicy、CASFullControlPolicy对应的权限，同时支持选择**所有资源**或**特定资源**，支持使用证书ID进行筛选，选择好以后点击**确定**，自定义策略即创建成功



自定义权限范围详细如下：

权限名称	权限范围
管理权限	拥有购买SSL证书、删除SSL证书的权限以及运维权限的所有权限
运维权限	拥有申请SSL证书、取消SSL证书申请以及只读权限的所有权限
只读权限	拥有查看SSL证书列表、查看SSL证书详情、查看SSL证书申请、下载SSL证书的权限

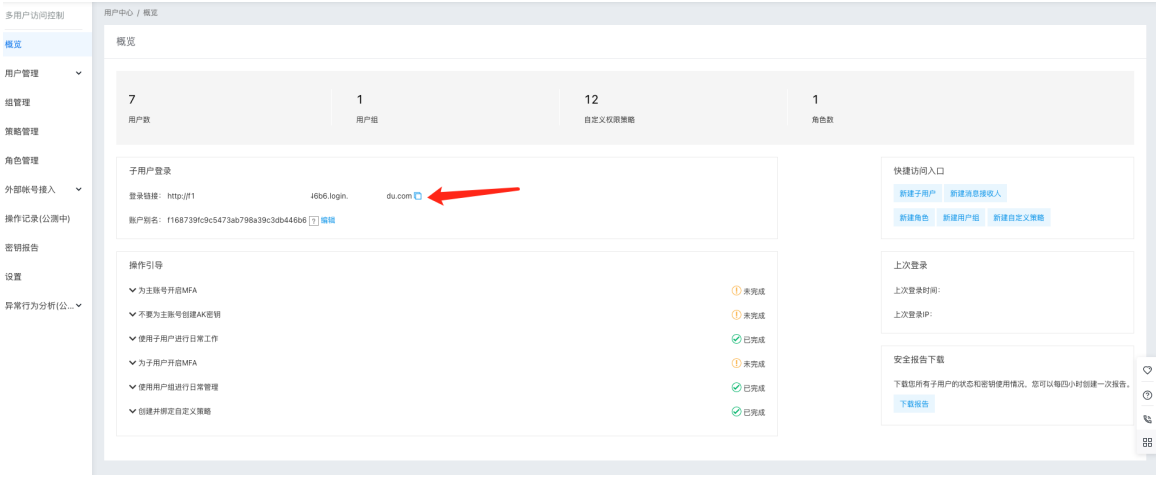
用户授权

在“用户管理->子用户管理列表页”的对应子用户的“操作”列选择“添加权限”，并为用户选择系统权限或自定义策略进行授权。

证书如给子用户赋予自定义策略权限，请务必同时给用户勾选系统策略中的CASReadOnlyAccessPolicy，给子用户授权读取所有SSL证书的权限。

子用户登录

主账号完成对子用户的授权后，可以将子用户登陆链接发送给子用户；子用户可以通过IAM用户登录链接登录主账号的管理控制台，根据被授权的策略对主账户的SSL证书资源进行操作和查看。



其他详细操作参考：多用户访问控制。

证书相关概念

主流数字证书都有哪些格式？

主流Web服务软件

一般来说，主流的Web服务软件，通常都基于OpenSSL和Java两种基础密码库。

- Tomcat、Weblogic、JBoss等Web服务软件，一般使用Java提供的密码库。通过Java Development Kit （JDK）工具包中的Keytool工具，生成Java Keystore（JKS）格式的证书文件。
- Apache、Nginx等Web服务软件，一般使用OpenSSL工具提供的密码库，生成PEM、KEY、CRT等格式的证书文件。
- IBM的Web服务产品，如Websphere、IBM Http Server（IHS）等，一般使用IBM产品自带的iKeyman工具，生成KDB格式的证书文件。
- 微软Windows Server中的Internet Information Services（IIS）服务，使用Windows自带的证书库生成PFX格式的证书文件。

如何判断证书文件是文本格式还是二进制格式？

您可以使用以下方法简单区分带有后缀扩展名的证书文件：

- *.DER或*.CER文件： 这样的证书文件是二进制格式，只含有证书信息，不包含私钥。
- *.CRT文件： 这样的证书文件可以是二进制格式，也可以是文本格式，一般均为文本格式，功能与 *.DER及*.CER证书文件相同。
- *.PEM文件： 这样的证书文件一般是文本格式，可以存放证书或私钥，或者两者都包含。 *.PEM 文件如果只包含私钥，一般用*.KEY文件代替。
- *.PFX或*.P12文件： 这样的证书文件是二进制格式，同时包含证书和私钥，且一般有密码保护。

您也可以使用记事本直接打开证书文件。如果显示的是规则的数字字母，例如：

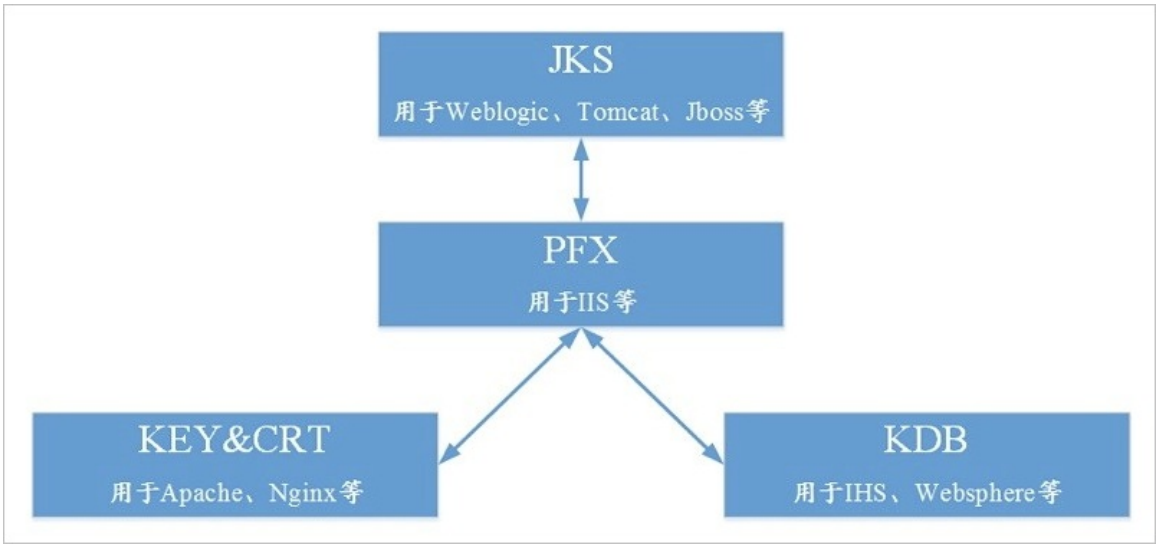
```
—–BEGIN CERTIFICATE—–
MIIE5zCCA8+gAwIBAgIQN+whYc2BgzAogau0dc3PtzANBgkqh.....
—–END CERTIFICATE—–
```

那么，该证书文件是文本格式的。

- 如果存在`——BEGIN CERTIFICATE——`，则说明这是一个证书文件。
- 如果存在`—–BEGIN RSA PRIVATE KEY—–`，则说明这是一个私钥文件。

证书格式转换

以下证书格式之间是可以互相转换的。



您可使用以下方式实现证书格式之间的转换：

说明： SSL证书服务统一使用 PEM 格式的数字证书文件。

- 将JKS格式证书转换成PFX格式

  您可以使用JDK中自带的Keytool工具，将JKS格式证书文件转换成PFX格式。例如，您可以执行以下命令将server.jks证书文件转换成server.pfx证书文件：

  ```
  keytool -importkeystore -srckeystore D:\server.jks -destkeystore D:\server.pfx
      -srcstoretype JKS -deststoretype PKCS12
  ```

- 将PFX格式证书转换为JKS格式

  您可以使用JDK中自带的Keytool工具，将PFX格式证书文件转换成JKS格式。例如，您可以执行以下命令将server.pfx证书文件转换成server.jks证书文件：

  ```
  keytool -importkeystore -srckeystore D:\server.pfx -destkeystore D:\server.jks
      -srcstoretype PKCS12 -deststoretype JKS
  ```

- 将PEM/KEY/CRT格式证书转换为PFX格式

  您可以使用 OpenSSL工具，将KEY格式密钥文件和CRT格式公钥文件转换成PFX格式证书文件。例如，将您的KEY格式密钥文件（server.key）和CRT格式公钥文件（server.crt）拷贝至OpenSSL工具安装目录，使用OpenSSL工具执行以下命令将证书转换成server.pfx证书文件：

  ```
  openssl pkcs12 -export -out server.pfx -inkey server.key -in server.crt
  ```

- 将PFX转换为PEM/KEY/CRT

  您可以使用 OpenSSL工具，将PFX格式证书文件转化为KEY格式密钥文件和CRT格式公钥文件。例如，将您的PFX格式证书文件拷贝至OpenSSL安装目录，使用OpenSSL工具执行以下命令将证书转换成server.pem证书文件KEY格式密钥文件（server.key）和CRT格式公钥文件（server.crt）：

  - `openssl pkcs12 -in server.pfx -nodes -out server.pem`
  - `openssl rsa -in server.pem -out server.key`
  - `openssl x509 -in server.pem -out server.crt`

  说明： 此转换步骤是专用于通过Keytool工具生成私钥和CSR申请证书文件的，并且通过此方法您可以在获取到PEM格式证书公钥的情况下分离私钥。在您实际部署数字证书时，请使用通过此转换步骤分离出来的私钥和您申请得到的公钥证书匹配进行部署。

SSL证书安装指南

在IIS服务器上安装SSL证书

您可将下载的百度云SSL证书安装到IIS服务器上，使您的IIS服务器支持HTTPS安全访问。

前提条件

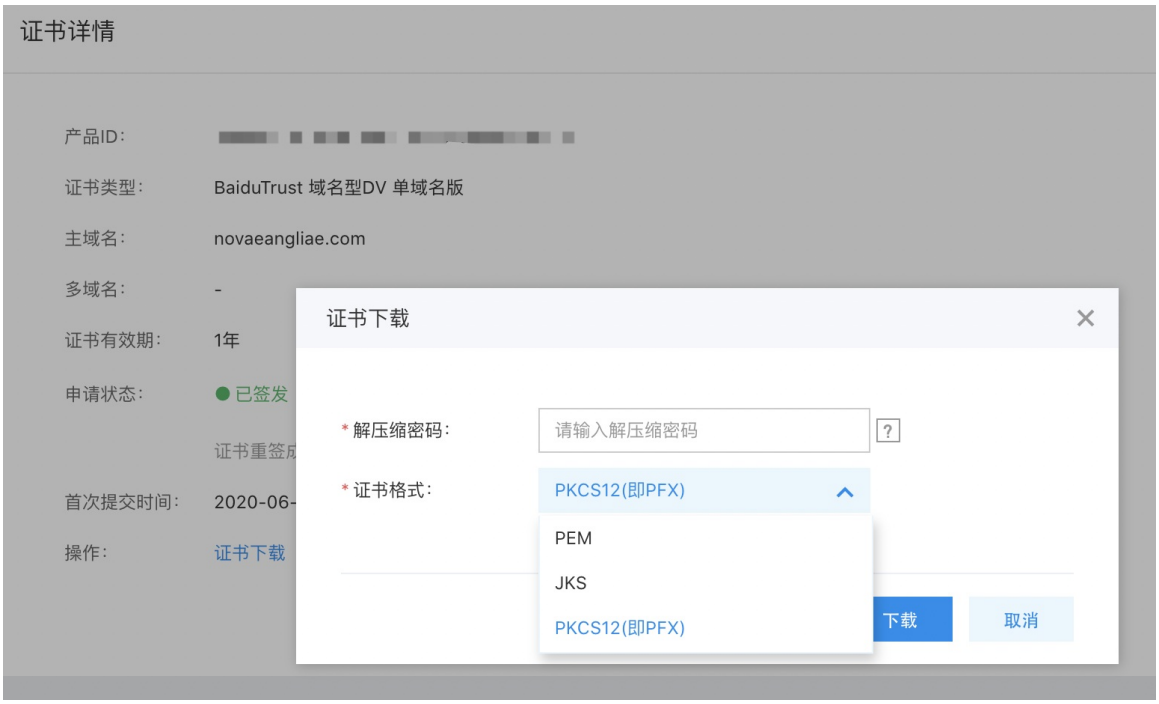
- 已安装IIS服务器，且您的IIS服务器上已经开启了443端口（HTTPS服务的默认端口）。
- 已安装OpenSSL工具。
- 已下载IIS服务器所需要的证书文件。

> 说明:
> 申请证书时需要选择**系统自动创建CSR**。
> 申请证书时如果选择**手动创建CSR**，则不会生成证书文件。您需要选择**其他服务器**下载.crt证书文件后，使用openssl命令将.crt文件的证书转换成.pfx格式。

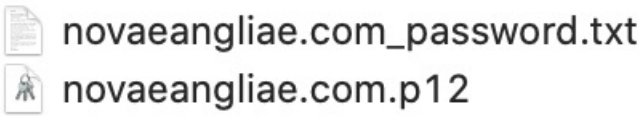
操作指南

1. 登录百度云SSL证书控制台。
2. 在SSL证书页面，定位到需要下载的证书并单击证书条目右下角的**查看证书**

3. 打开后点击**证书下载**对话框。选择PFX格式并且键入证书压缩密码（注意不是证书密码也不是订单密码）

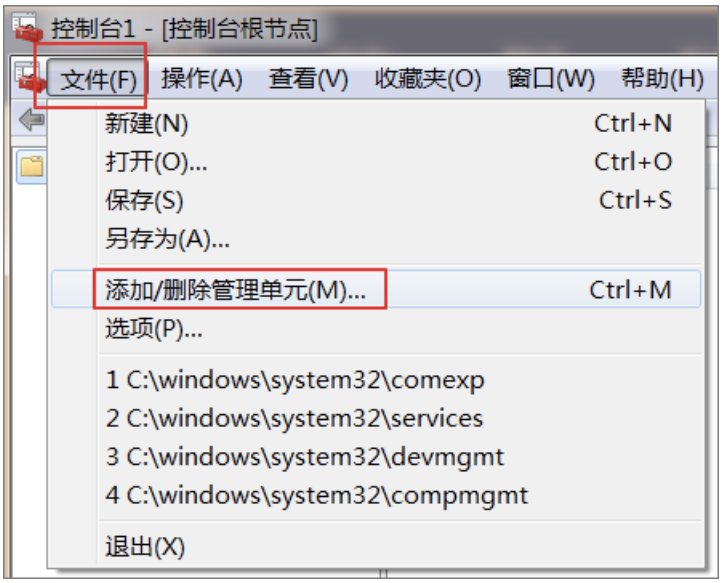


4. 解压Tomcat证书。您将看到文件中有一个以.p12为后缀或文件类型的证书文件。（若是百度自有品牌BaiduTrus证书，还会有一个密码文件，以.txt为后缀或文件类型）。

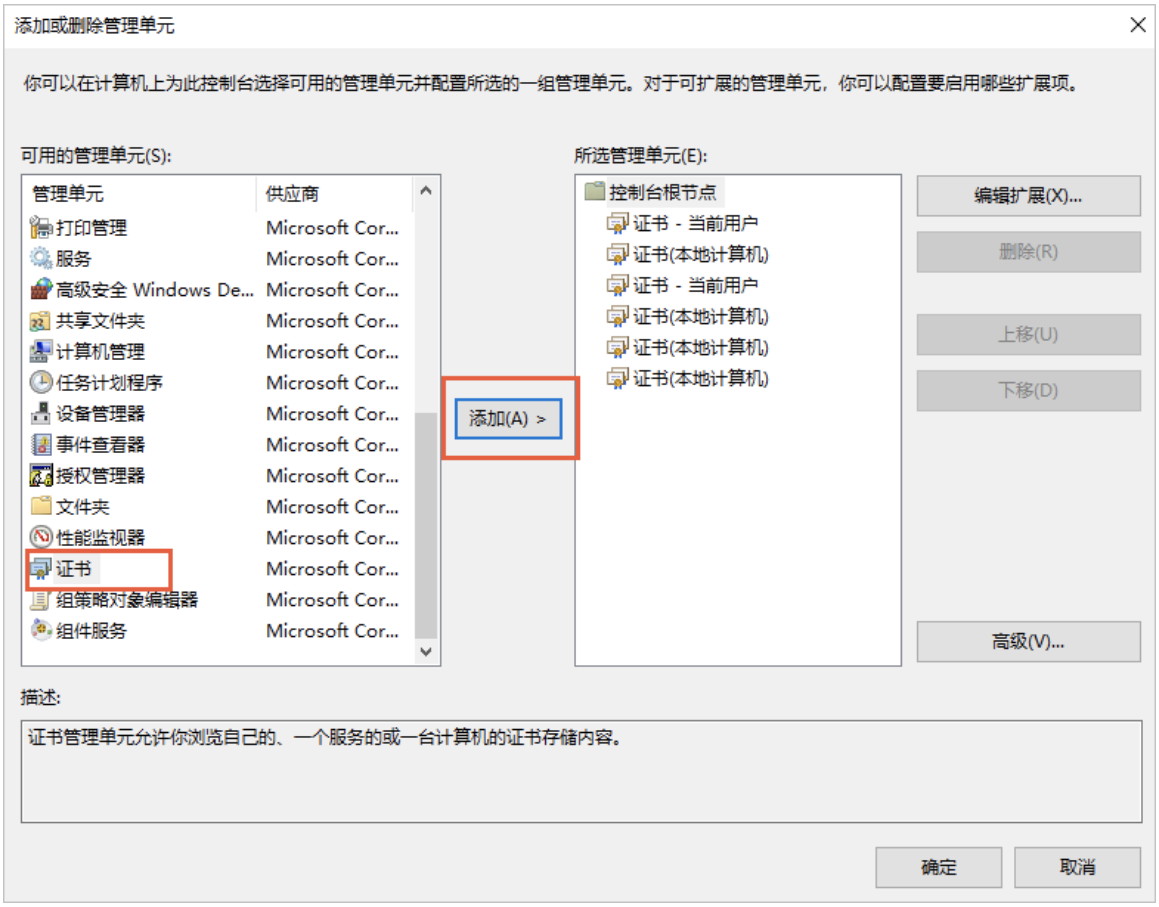


说明：
每次下载证书都会产生新的密码，该密码仅匹配本次下载的证书。如果需要更新证书文件，同时也要更新匹配的密码文件。

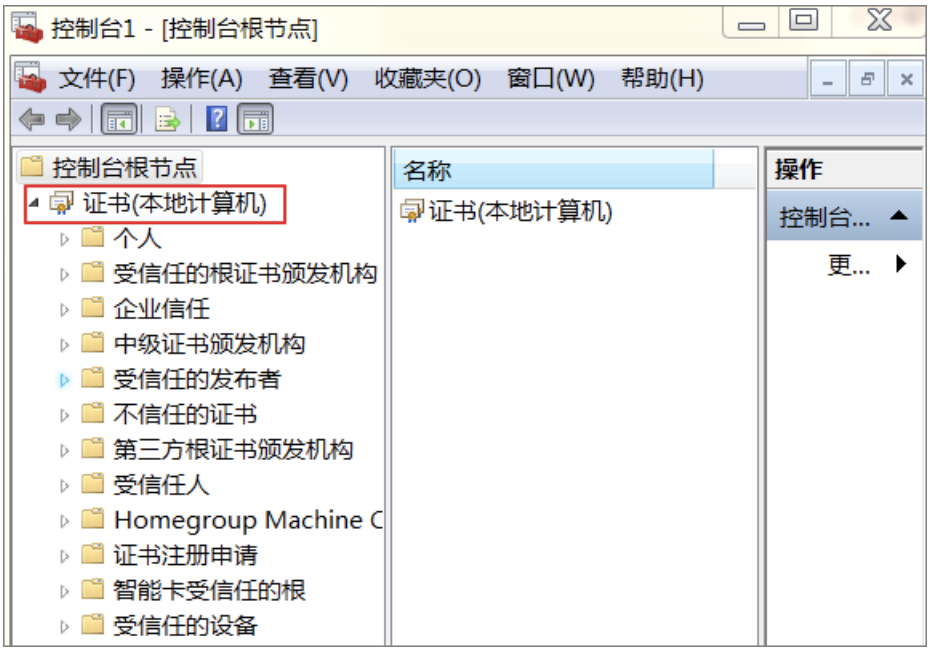
5. 在**控制台**操作对话框中导入您下载的IIS证书文件。
    1. 单击**开始** > **运行** > **MMC**打开控制台。
    2. 单击**文件** > **添加/删除管理单元**打开**添加/删除管理单元**对话框。



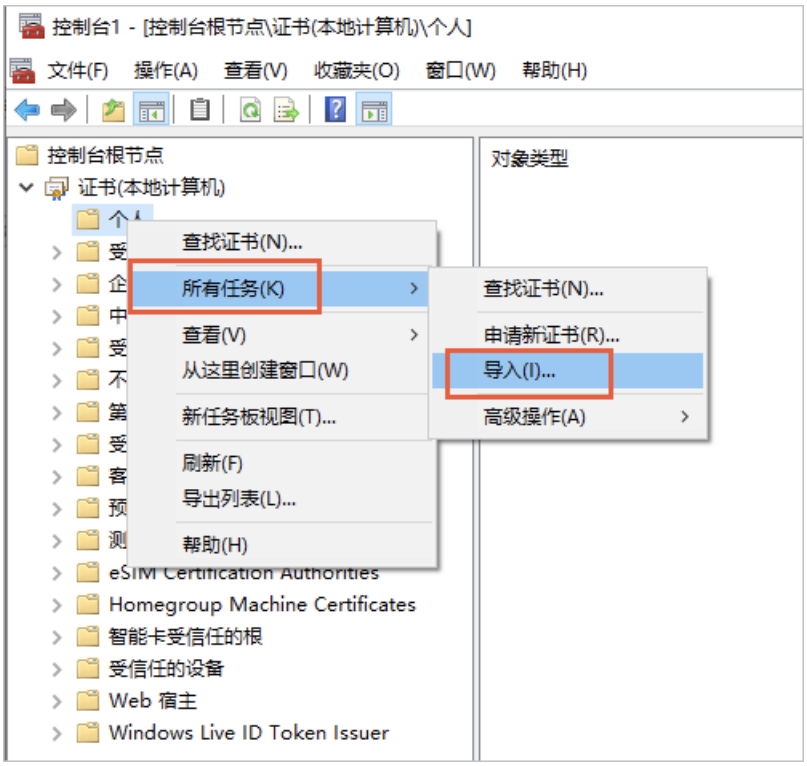
3. 在**可用的管理单元**中单击**证书** > **添加** > **计算机账户** > **本地计算机（运行此控制台的计算机）** > **完成**。



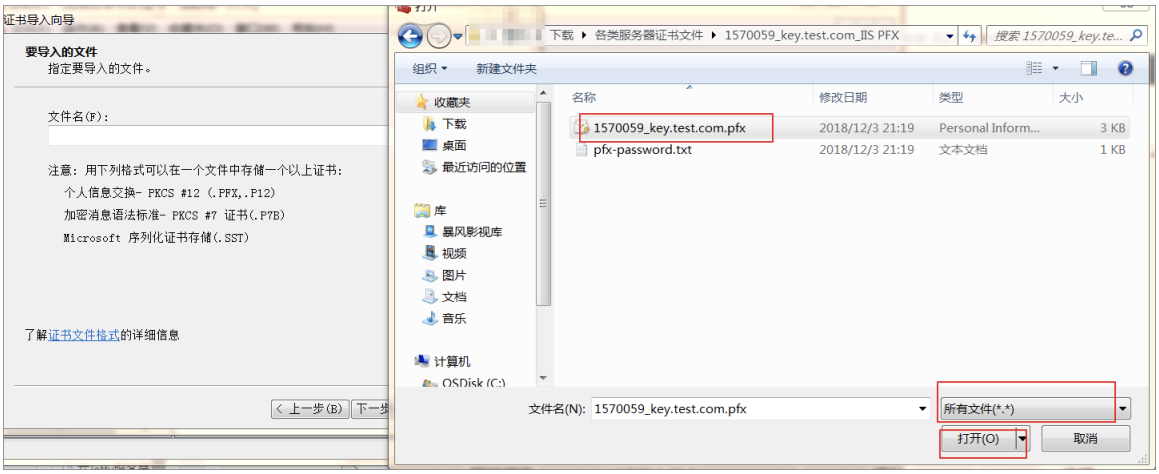
4. 在控制台左侧导航栏单击**控制台根节点**下的**证书**打开证书树形列表。



5. 单击**个人** > **证书** > ** > 所有任务 > 导入**打开**证书导入向导**对话框。



6. 在**要导入的文件**页面单击浏览导入下载的PFX格式证书文件，并单击**下一步**。

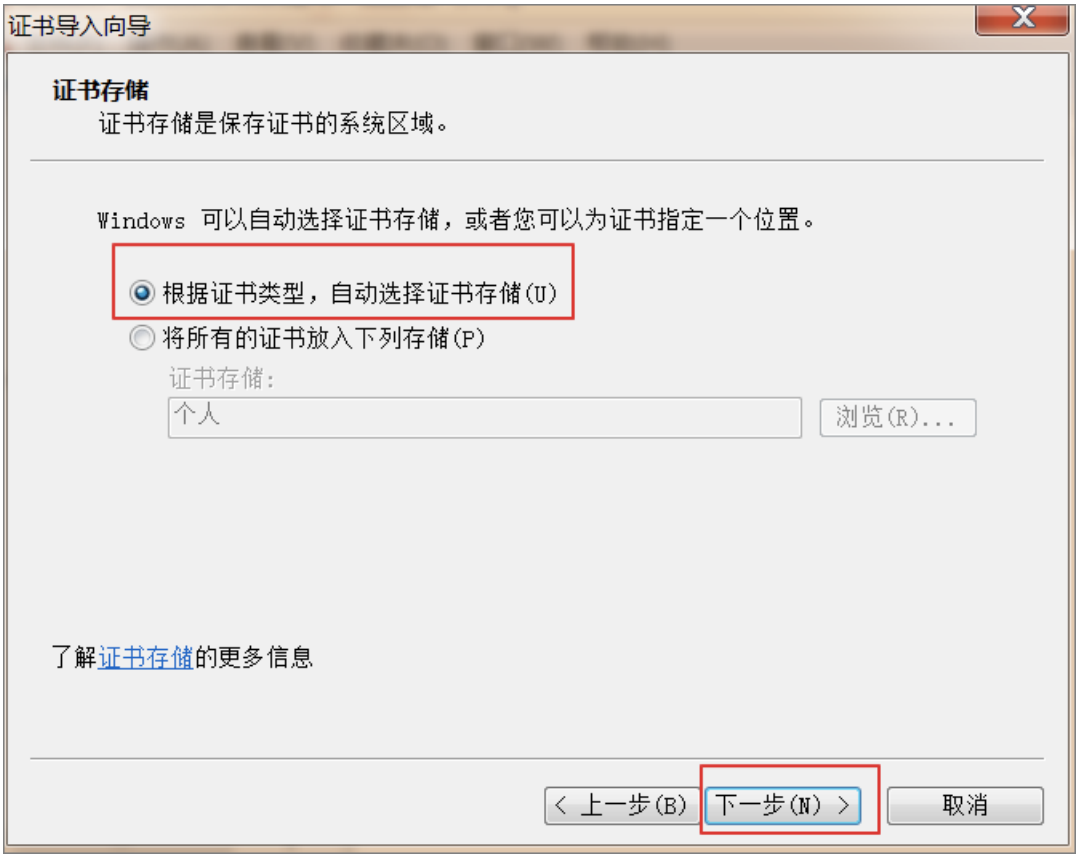


说明： 在导入证书文件时，**文件名**右侧文件类型下拉菜单中请选择**所有文件**类型。

7. 输入证书秘钥文件里的密码。

   您可在下载的IIS证书文件中打开password.txt文件查看证书密码。

8. 勾选**根据证书类型，自动选择证书存储**并单击**下一步**完成证书的导入。



6. 分配服务器证书。

1. 打开IIS8.0 管理器面板，定位到待部署证书的站点，单击**绑定**。

2. 在**网站绑定**对话框中单击**添加** > **选择https类型** > **端口选择443** > **导入的IIS证书名称** > **确定**。

说明： SSL 缺省端口为 443 端口，请不要修改。 如果您使用其他端口如：8443，则访问网站时必须输入`https://www.domain.com:8443`)。

后续操作

证书安装完成后，可通过登录证书绑定域名的方式验证证书是否安装成功。

```
https://domain:port  #domain name替换成证书绑定的域名,默认443端口可以忽略不输入
```

如果网页地址栏出现绿色小锁标志，表示证书安装成功。

验证证书是否安装成功时，如果网站无法通过https正常访问，需确认您安装证书的服务器443端口是否已开启或被其他工具拦截。

在Nginx或Tengine服务器上安装证书

百度智能云SSL证书服务支持下载证书安装到Nginx、Tengine服务器上，本文介绍了证书安装的具体操作。

前提条件

您的Nginx或Tengine服务器需具备以下条件：

- 服务器已开启了443端口（HTTPS服务的默认端口）。
- 服务器上已安装了http_ssl_module模块（启用SSL功能）。

背景信息

本文档以CentOS 7、Nginx 1.15.6为例。

本文档证书名称以 `domain` 为示例，如证书文件名称为 `domain.crt`，证书密钥文件名称为 `domain.key`。

下载的Nginx证书压缩文件解压后包含：

- .crt：证书文件，crt扩展名的证书文件采用Base64-encoded的PEM格式文本文件，可根据需要修改成 `.pem` 等扩展名。
- .key：证书的密钥文件。申请证书时如果选择使用**已有的CSR方式**申请证书，则下载的证书文件压缩包中不会包含.key文件（私钥文件），key文件在您生成CSR时已经生成了，需要您将自己手动生成的私钥文件拷贝到 `cert` 目录下并命名为 `domain.key`。

证书的格式详见 主流数字证书都有哪些格式。

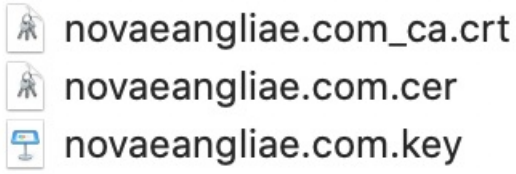
操作指南

1. 登录百度智能云 SSL证书控制台。
2. 在SSL证书页面，定位到需要下载的证书并单击证书条目右下角的**管理证书**

3. 打开后点击**证书下载**对话框。选择 **PEM_nginx&apache（非百度证书下载PEM_nginx）** 格式并且键入证书压缩密码（注意不是证书密码也不是订单密码），OV、EV证书还会有订单密码输入框，请输入申请证书是填写的密码

4. 解压Nginx证书，您将看到文件夹中有3个文件：

   - 证书文件（以 .cer 、 crt、pem 为后缀或文件类型）
   - 密钥文件（以 .key 为后缀或文件类型）
   - 证书链文件（以 _ca.cer 或 _ca.crt 为后缀或文件类型）

5. 在Nginx安装目录下创建 cert 目录，并将下载的证书文件，和密钥文件拷贝到 cert 目录中。

   说明： 如果您在申请证书时选择使用**已有的CSR**申请证书，请将对应的密钥文件放到 cert 目录中，并命名为 domain.key。

6. 打开**Nginx安装目录** > **conf文件夹** > **nginx.conf文件**，在 nginx.conf 文件中找到以下属性：

```
  # HTTPS server
 server {
  listen 443;
  server_name localhost;
  ssl on;
  ssl_session_timeout 5m;
  ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
  ssl_ciphers ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP;
  ssl_prefer_server_ciphers on;
  location / {
  ……
```

   修改 nginx.conf 文件如下：

```
 # 以下属性中以ssl开头的属性代表与证书配置有关，其他属性请根据自己的需要进行配置。
server {
    listen 443;
    server_name localhost;  # localhost修改为您证书绑定的域名。
    ssl on;  #设置为on启用SSL功能。
    root html;
    index index.html index.htm;
    ssl_certificate cert/domain_ca.crt;  # 将domain_ca.crt替换成您证书链的文件名(如下载文件没有domain_ca.crt则使用domain.crt或domain.pem)。
    ssl_certificate_key cert/domain.key;  # 将domain.key替换成您证书的密钥文件名。
    ssl_session_timeout 5m;              # 指定SSL/TLS会话的超时时间
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;  #使用此加密套件。
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  #使用该协议进行配置。
    ssl_prefer_server_ciphers on;
    location / {
        root html;  #站点目录。
        index index.html index.htm;
    }
}
```

7. 保存 `nginx.conf` 文件后退出。
8. 重启Nginx服务器。
9. （可选步骤）设置http请求自动跳转https。

   在需要跳转的http站点下添加以下 `rewrite` 语句，实现 **http** 访问自动跳转到 **https** 页面

```
 server {
  listen 80;
  server_name localhost;
  rewrite ^(.*)$ https://$host$1 permanent;
  location / {
  index index.html index.htm;
}
```

后续操作

证书安装完成后，可通过登录证书绑定域名的方式验证证书是否安装成功。

```
https://domain:port  #domain替换成证书绑定的域名,默认443端口可以忽略不输入
```

如果网页地址栏出现 **小锁** 标志，表示证书安装成功。

验证证书是否安装成功时，如果网站无法通过 **https** 正常访问，需确认您安装证书的服务器443端口是否已开启或被其他工具拦截。

在Apache服务器上安装SSL证书

在Apache服务器上安装SSL证书

您可以将从百度智能云SSL证书控制台下载的证书安装到您的Apache服务器上，使Apache服务器支持HTTPS安全访问。

前提条件

- 已安装OpenSSL。
- Apache服务器已安装mod_ssl.so模块（启用SSL功能）。

- 如未安装，可执行 `yum install -y mod_ssl` 命令安装。安装后，可执行 `httpd -M | grep 'ssl'` 检查 `mod_ssl.so` 是否安装成功。

安装成功效果图：

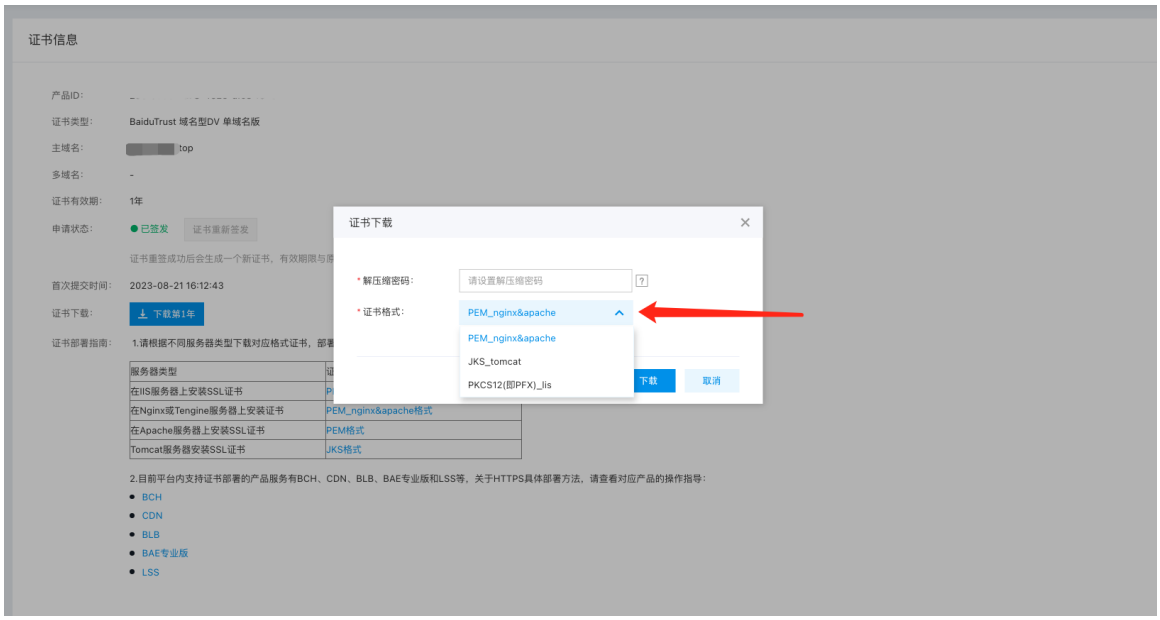
- 本文档证书名称以 `domain` 为示例，如证书文件名称为 `domain.crt`，证书链文件名称为 `domain_ca.crt`，证书密钥（私钥）文件名称为 `domain.key`。
- 申请证书时如果未选择**系统自动创建CSR**，证书下载压缩包中将不包含 `.key` 文件。

说明：`.crt` 扩展名的证书文件采用Base64-encoded的PEM格式文本文件，可根据需要修改成 `.pem` 等扩展名。

操作指南

1. 登录百度智能云 SSL证书控制台。
2. 在SSL证书页面，定位到需要下载的证书并单击证书条目右下角的**管理证书**

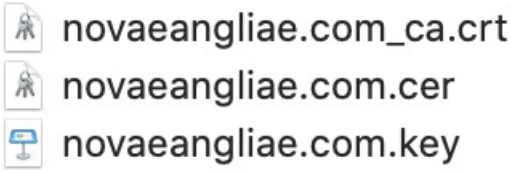
3. 打开后点击**证书下载**对话框。选择**PEM_nginx&apache**格式并且键入证书压缩密码（注意不是证书密码也不是订单密码）将Apache版证书压缩包下载到本地。



4. 解压Apache证书。

   您将看到文件夹中有3个文件：

   - 证书文件（以 `.crt` 或 `.cer` 或 `.pem` 为后缀或文件类型）
   - 证书链文件（以 `_ca.crt` 或 `.cer` 或 `.pem` 为后缀或文件类型）
   - 秘钥文件（以 `.key` 为后缀或文件类型）



5. 在Apache安装目录（一般在 /etc/httpd）中新建 cert 目录，并将下载的Apache证书、 证书链文件和秘钥文件拷贝到 cert 目录中。

说明：如果申请证书时选择了**使用已有的CSR**方式，请将手动生成创建的秘钥文件拷贝到 cert 目录中并命名为 domain.key。

6. 在Apache安装目录下，打开 Apache/conf/httpd.conf，在 httpd.conf 文件中找到以下参数并进行配置。

```
 #LoadModule ssl_module modules/mod_ssl.so  #删除行首的配置语句注释符号“#”加载mod_ssl.so模块启用SSL服务，Apache默认是不启用该模块的。如果找不到该配置，请重新编译mod_ssl模块。
#conf.modules.d/*.conf #（用于加载SSL配置目录），检查是否被注释，如果被注释，请删除#注释。
```

7. 保存 httpd.conf 文件并退出。

8. 配置证书

   8.1. 打开 Apache/conf/extra/httpd-ssl.conf(如没有找到则参考8.2)，在 httpd-ssl.conf 文件中找到以下参数并进行配置。证书路径建议使用绝对路径。

```
 SSLProtocol all -SSLv2 -SSLv3  # 添加SSL协议支持协议，去掉不安全的协议。
SSLCipherSuite HIGH:!RC4:!MD5:!aNULL:!eNULL:!NULL:!DH:!EDH:!EXP:+MEDIUM  # 使用此加密套件。
SSLHonorCipherOrder on
SSLCertificateFile cert/domain.crt  # 将domain.crt替换成您证书文件名。
SSLCertificateKeyFile cert/domain.key  # 将domain.key替换成您证书的秘钥文件名。
SSLCertificateChainFile cert/domain_ca.crt  # 将domain_ca.crt替换成您证书链文件名；证书链开头如果有#字符，请删除。
```

   8.2. 如没有找到上述文件，则编辑 Apache/conf.d/ssl.conf，在 ssl.conf 配置文件中，定位到以下参数，按照中文注释修改

```
<VirtualHost _default_:443> # 也可能是 <VirtualHost *:443>
   ServerName www.xxx.com #修改为申请证书时绑定的域名。
   SSLCertificateFile cert/domain.crt  # 将domain.crt替换成您证书文件名。
   SSLCertificateKeyFile cert/domain.key  # 将domain.key替换成您证书的秘钥文件名。
   SSLCertificateChainFile cert/domain_ca.crt  # 将domain_ca.crt替换成您证书链文件名；证书链开头如果有#字符，请删除。
</VirtualHost>
```

9. 保存 httpd-ssl.conf 文件配置并退出。

10. 重启Apache服务器使SSL配置生效。

    10.1. 在Apache bin 目录下执行以下命令停止Apache服务。

```
    apachectl -k stop # 如发现无法停止服务，可以尝试使用 `systemctl stop httpd` 命令
```

    10.2. 在Apache bin 目录下执行以下命令开启Apache服务。

```
    apachectl -k start # 如发现无法启动服务，可以尝试使用 `systemctl start httpd` 命令
```

11. （可选步骤）设置Apache http自动跳转https。

    在 httpd.conf 文件中，在 <VirtualHost *:80> </VirtualHost> 中间，添加以下重定向代码。

```
 RewriteEngine on
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^(.*)$ https://%{SERVER_NAME}$1 [L,R]
```

后续操作

证书安装完成后，可通过登录证书绑定域名的方式验证证书是否安装成功。

```
https://domain  #domain替换成证书绑定的域名
```

如果网页地址栏出现 **小锁** 标志，表示证书安装成功。

验证证书是否安装成功时，如果网站无法通过**https**正常访问，需确认您安装证书的服务器443端口是否已开启或被其他工具拦截。

Tomcat服务器安装SSL证书

安装JKS格式证书

您可以将下载的证书安装到Tomcat服务器上。Tomcat支持PFX格式和JKS两种格式的证书，您可根据选您Tomcat的版本择其中一种格式的证书安装到Tomcat上。本文档介绍了JKS格式证书安装的具体步骤。

前提条件

您的Tomcat服务器上已经开启了443端口（HTTPS服务的默认端口）。
已安装OpenSSL工具。
已下载Tomcat服务器所需要的证书文件。

> **说明**
>
> 申请证书时如果选择指定提交CSR信息，证书下载压缩包中将不包含.txt文件。需要您选择其他类型服务器下载.crt证书，并使用openssl命令生成jks文件。 如果您自己拥有其他证书，可使用openssl命令将您自己的证书文件转化为相应格式的文件，安装到Tomcat服务器上。

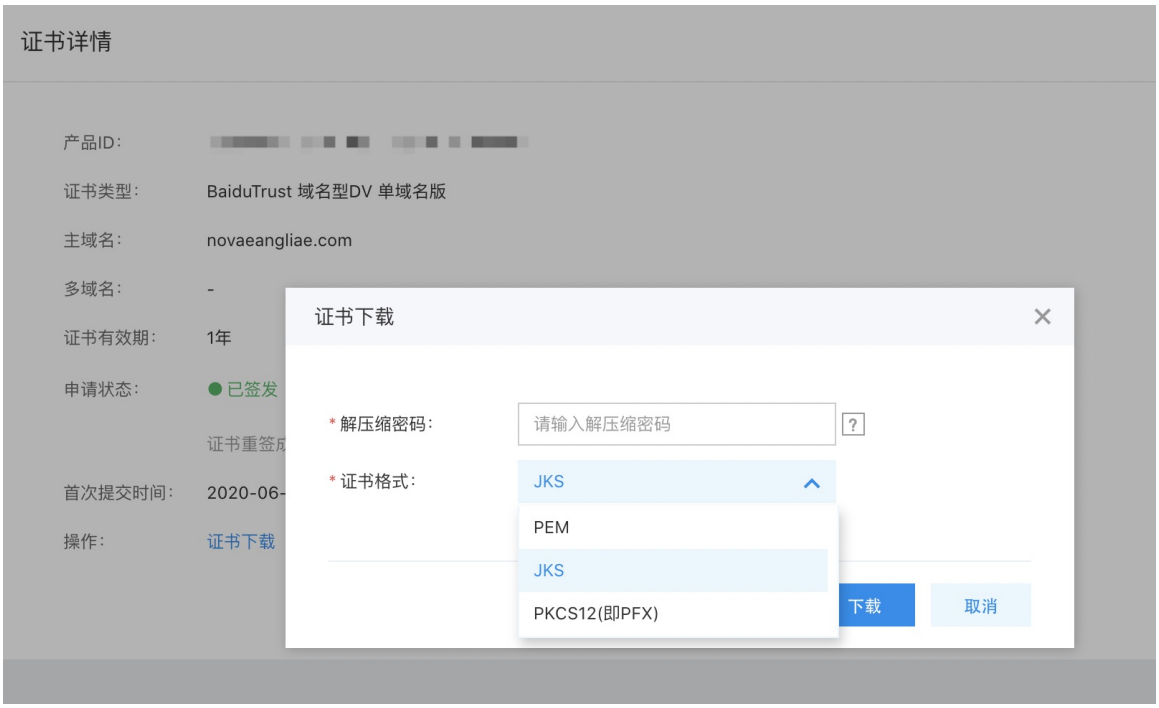
背景信息

- 本文档证书名称以**domain.com**为示例，如证书文件名称为**domain.com.jks**，证书密码文件名称为**domain.com_password.txt**。
- 申请证书时如果未选择**系统自动创建CSR**，证书下载压缩包中将不包含.txt文件。需要您选择**PEM类型**下载.crt证书，并使用openssl命令生成pfx文件。

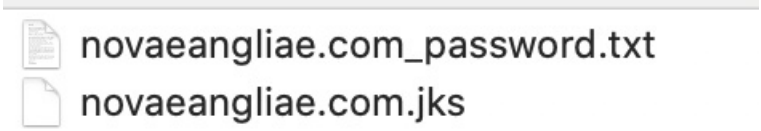
操作指南

1. 登录百度云SSL证书控制台。
2. 在SSL证书页面，定位到需要下载的证书并单击证书条目右下角的**查看证书**

3. 打开后点击**证书下载**对话框。选择JKS格式并且键入证书压缩密码（注意不是证书密码也不是订单密码）



4. 解压Tomcat证书。您将看到文件中有一个以.jks为后缀或文件类型的证书文件（若是百度自有品牌BaiduTrus证书，还会有一个密码文件，以.txt为后缀或文件类型；如果**没有密码文件**，则密码为用户下载证书时设置的**解压密码**）。



说明： 每次下载证书都会产生新的密码，该密码仅匹配本次下载的证书。如果需要更新证书文件，同时也要更新匹配的密码文件。

5. 在**Tomcat**安装目录下新建**cert**目录，将证书和密码文件拷贝到**cert**目录下。

6. 打开**Tomcat安装目录** > **conf文件夹** > **server.xml文件**，在**server.xml文件**中找到 <Connector port="8443"标签并进行修改。

```
 <!--
 <Connector port="8443"
protocol="HTTP/1.1"
port="8443" SSLEnabled="true"
 maxThreads="150" scheme="https" secure="true"
 clientAuth="false" sslProtocol="TLS" />

-->
```

参考以下完整配置（其中port属性请根据您的实际情况修改）：

```
 <Connector port="443"
   protocol="HTTP/1.1"
   SSLEnabled="true"
   scheme="https"
   secure="true"
   keystoreFile="cert/domain.jks"  #此处keystoreFile代表证书文件的路径，请用您证书的文件名替换domain。
   keystoreType="PKCS12"
   keystorePass="证书密码"   #请用您证书密码文件中的密码替换“证书密码”。
   clientAuth="false"
   SSLProtocol="TLSv1+TLSv1.1+TLSv1.2"

ciphers="TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_CB
>
```

7. 保存server.xml文件配置。

8. （可选步骤）配置web.xml文件开启HTTP强制跳转HTTPS。

```
 #在</welcome-file-list>后添加以下内容：
<login-config>
    <!-- Authorization setting for SSL -->
    <auth-method>CLIENT-CERT</auth-method>
    <realm-name>Client Cert Users-only Area</realm-name>
</login-config>
<security-constraint>
    <!-- Authorization setting for SSL -->
    <web-resource-collection >
        <web-resource-name >SSL</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

9. 重启Tomcat。

后续操作

证书安装完成后，可通过登录证书绑定域名的方式验证证书是否安装成功。

```
https://domain:port   #domain name替换成证书绑定的域名,默认443端口可以忽略不输入
```

如果网页地址栏出现绿色小锁标志，表示证书安装成功。

验证证书是否安装成功时，如果网站无法通过https正常访问，需确认您安装证书的服务器443端口是否已开启或被其他工具拦截。

典型实践

CentOS系统Tomcat 8.5或9部署SSL证书

本文档介绍了CentOS系统下Tomcat 8.5或9部署SSL证书的操作说明。

环境准备

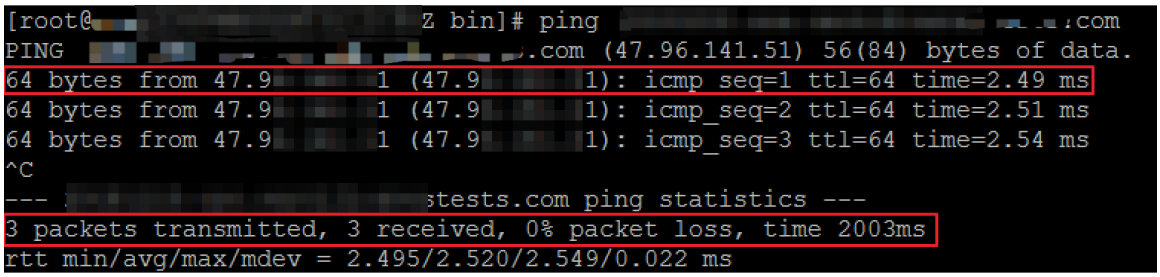
- 操作系统：CentOS 7.6 64位

- Web服务器：Tomcat 8.5或9

> **注意**：Tomcat服务器需要提前安装JDK环境变量，请前往Tomcat官网查看推荐的JDK兼容配置。

前提条件

- 已从百度智能云SSL证书服务控制台下载Tomcat服务器证书（包含PFX格式证书文件和TXT格式密码文件）。
- 您申请SSL证书时绑定的域名已完成DNS解析、实现了该域名指向您Tomcat服务器的IP地址。
  域名解析设置完成后执行 `ping www.yourdomain.com` 命令，如果返回了您所设置解析的主机IP地址，说明解析成功。



操作步骤

1. 解压Tomcat证书。

> **注意：** 每次下载证书都会产生新的密码，该密码仅匹配本次下载的证书。如果需要更新证书文件，同时也要更新匹配的密码。

2. 将下载的证书和密码文件拷贝到Tomcat的 **conf** 目录下。

> **注意：** 如果需要安装JKS格式证书，可使用以下命令将PFX格式证书转化成JKS格式。`keytool -importkeystore -srckeystore domain name.pfx -destkeystore domain name.jks -srcstoretype PKCS12 -deststoretype JKS`

3. 打开Tomcat/conf/server.xml，在server.xml文件中找到以下参数并进行修改。

```
 <Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
#找到以上参数，去掉<!- - 和 - ->这对注释符并修改为如下参数，对HTTPS默认端口进行配置：
<Connector port="80" protocol="HTTP/1.1"    #将Connector port修改为80。
           connectionTimeout="20000"
           redirectPort="443" />     #将redirectPort修改为SSL默认端口443，让HTTPS请求转发到443端口。
```

```
<Connector port="8443"
        protocol="org.apache.coyote.http11.Http11NioProtocol"
        maxThreads="150"
        SSLEnabled="true">
      <SSLHostConfig>
        <Certificate       certificateKeystoreFile="cert/keystore.pfx"
         certificateKeystorePassword="XXXXXXX"
                    certificateKeystoreType="PKCS12" />
  #找到以上参数，去掉<!- - 和 - ->这对注释符并修改为如下参数：
  <Connector port="443"   #将Tomcat中默认的HTTPS端口Connector port 8443修改为443。8443端口不可通过域名直接访问、需要在域名后加上端口号；443端口是HTTPS的默认端口，可通过域名直接访问，无需在域名后加端口号。
        protocol="org.apache.coyote.http11.Http11NioProtocol"   #server.xml文件中Connector port有两种运行模式（NIO和APR），请选择NIO模式（也就是protocol="org.apache.coyote.http11.Http11NioProtocol"）这一段进行配置。
        maxThreads="150"
        SSLEnabled="true">
      <SSLHostConfig>
        <Certificate       certificateKeystoreFile="/usr/local/tomcat/cert/证书域名.pfx"   #此处certificateKeystoreFile代表证书文件的路径，请用您证书的路径+文件名替换证书域名.pfx，例如：certificateKeystoreFile="/usr/local/tomcat/cert/abc.com.pfx"
         certificateKeystorePassword="证书密码"   #此处certificateKeystorePassword为SSL证书的密码，请用您证书密码文件pfx-password.txt中的密码替换，例如：certificateKeystorePassword="bMNML1Df"
         certificateKeystoreType="PKCS12" />   #证书类型为PFX格式时，certificateKeystoreType修改为PKCS12。
      </SSLHostConfig>
  </Connector>
```

```
  <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
#### 找到以上参数，去掉<!- - 和 - ->这对注释符并修改为如下参数：
<Connector port="8009" protocol="AJP/1.3" redirectPort="443" />  #将redirectPort修改为443，让HTTPS请求转发到443端口。
```

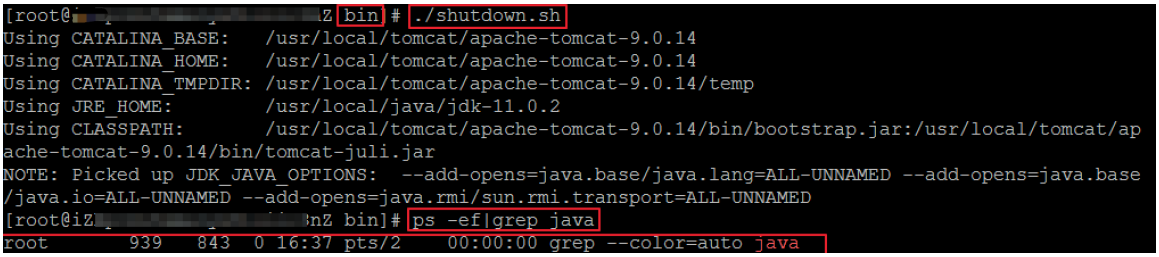
4. 保存server.xml文件配置。

5. （可选步骤）在web.xml文件最底部添加以下内容，实现HTTP自动跳转为HTTPS。

```
<security-constraint>
    <web-resource-collection >
        <web-resource-name >SSL</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
            <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

6. 重启Tomcat服务。

    1. 在Tomcat下的bin目录中执行./shutdown.sh关闭Tomcat服务。



    2. 在Tomcat下的bin目录中执行./startup.sh开启Tomcat服务。

```
[root@ .Z bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat/apache-tomcat-9.0.14
Using CATALINA_HOME:   /usr/local/tomcat/apache-tomcat-9.0.14
Using CATALINA_TMPDIR: /usr/local/tomcat/apache-tomcat-9.0.14/temp
Using JRE_HOME:        /usr/local/java/jdk-11.0.2
Using CLASSPATH:       /usr/local/tomcat/apache-tomcat-9.0.14/bin/bootstrap.jar:/usr/local/tomcat/ap
ache-tomcat-9.0.14/bin/tomcat-juli.jar
Tomcat started.
```

后续操作

Tomcat服务重启成功后，您可在浏览器中输入您SSL证书绑定的域名 `https://www.yourdomain.com` 验证证书安装结果。浏览器地址栏显示绿色的小锁标识说明证书安装成功。

Ubuntu系统Apache 2部署SSL证书

本文档为您介绍了如何在Ubuntu系统以及Apache2中安装百度智能云SSL证书。

环境准备

- 操作系统：Ubuntu
- Web服务器：Apache 2

前提条件

- 已从 SSL证书服务控制台 下载Apache服务器证书。
- 已安装Open SSL。

操作步骤

1. 运行以下命令在apache2目录下创建ssl目录。

```
mkdir /etc/apache2/ssl
```

2. 运行以下命令将下载的百度智能云证书文件复制到ssl目录中。

```
cp -r YourDomainName_public.crt /etc/apache2/ssl
```

```
cp -r YourDomainName_chain.crt /etc/apache2/ssl
```

```
cp -r YourDomainName.key /etc/apache2/ssl
```

3. 运行以下命令启用SSL模块。

```
sudo a2enmod ssl
```

```
root@ :~# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
```

SSL模块启用后可执行 `ls /etc/apache2/sites-available` 查看目录下生成的default-ssl.conf文件。

> **注意：**443端口是网络浏览端口，主要用于HTTPS服务。SSL模块启用后会自动放行443端口。若443端口未自动放行，可

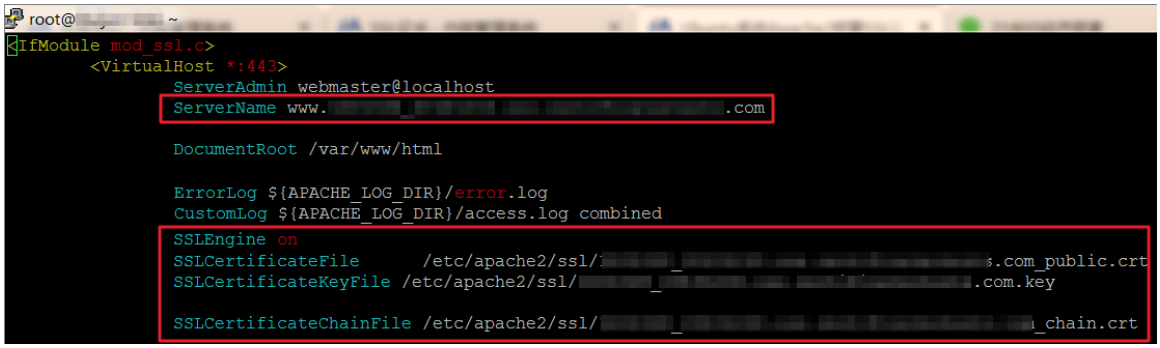
> 执行 vi /etc/apache2/ports.conf 并添加Listen 443手动放行。

4. 运行以下命令修改SSL配置文件default-ssl.conf。

```
vi /etc/apache2/sites-available/default-ssl.conf
```

在default-ssl.conf文件中找到以下参数进行修改后保存并退出。

```
 <IfModules mod_ssl.c>
<VirtualHost *:443>
ServerName  #修改为证书绑定的域名www.YourDomainName.com。
SSLCertificateFile /etc/apache2/ssl/www.YourDomainName_public.crt  #
将/etc/apache2/ssl/www.YourDomainName.com_public.crt替换为证书文件路径+证书文件名。
SSLCertificateKeyFile /etc/ssl/apache2/www.YourDomainName.com.key  #
将/etc/apache2/ssl/www.YourDomainName.com.key替换为证书密钥文件路径+证书密钥文件名。
SSLCertificateChainFile /etc/apache2/ssl/www.YourDomainName.com_chain.crt  #
将/etc/apache2/ssl/www.YourDomainName.com_chain.crt替换为证书链文件路径+证书链文件名。
```



/sites-available：该目录存放的是可用的虚拟主机；/sites-enabled：该目录存放的是已经启用的虚拟主机。

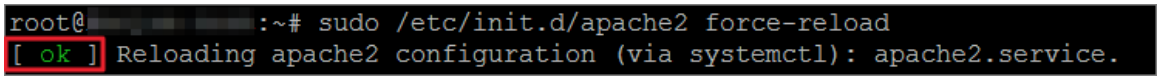
> **注意：**default-ssl.conf文件可能存放在/etc/apache2/sites-available或/etc/apache2/sites-enabled目录中。

5. 运行以下命令把default-ssl.conf映射至/etc/apache2/sites-enabled文件夹中建立软链接、实现二者之间的自动关联。

```
sudo ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/001-ssl.conf
```

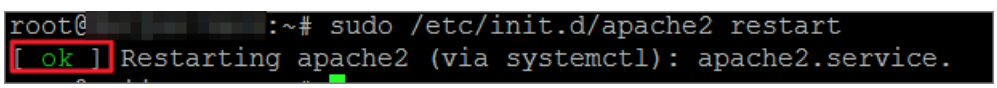
6. 运行以下命令重新加载Apache 2配置文件。

```
sudo /etc/init.d/apache2 force-reload
```



7. 运行以下命令重启Apache 2服务。

```
sudo /etc/init.d/apache2 restart
```



后续操作

Apache 2服务重启成功后，您可在浏览器中输入 https://www.YourDomainName.com 验证证书安装结果。浏览器地址栏显示绿色的小锁标识说明证书安装成功。

HTTPS安全典型实践

SSL和TLS部署

SSL/TLS 是一种简单易懂的技术，它很容易部署及运行。但想要部署的安全通常是不容易的。这也使系统管理员和开发者不得不去了解 SSL 和 TLS 相关的技术，掌握如何配置一个安全的 web 服务器或应用。无疑会耗费很大的精力去看相关的技术文档，乏味且宽泛。

本篇文档的目的在于如何让系统管理员或开发者用尽可能少的时间部署一个安全的 web 站点或应用，即 SSL 和 TLS 部署典型实践。

1 证书和私钥

在TLS中，所有的安全性都从服务器的密码标识开始；需要一个强大的私钥来防止攻击者进行模拟攻击。同样重要的是要有一个有效的和强大的证书，这将授予私有密匙作为一个特定主机名的权利。没有这两个基本的构建块，就没有其他东西可以安全了。

1.1 使用 2048 位私钥

对于大多数的 web 站点，提供一个 2048 的 RSA key 是足够安全的。RSA 的公钥算法是被普遍支持的，这使得这个类型的 key 作为默认是足够安全的。对于 2048 位，这些 key 提供了大约 112 位的安全性。如果您想要比这更多的安全性，请注意 RSA key 的伸缩性不太好。想要获得 128 位的安全性，你需要 3072 位 RSA key，这会很大的影响性能。ECDSA key 提供了一种提供更好安全性和更好性能的替代方法。对于 256 位，ECDSA key 提供 128 位安全性。少数古董客户端不支持 ECDSA，但现代客户端是支持的。如果您不介意管理这样一个设置的开销，那么您可以同时部署 RSA 和 ECDSA 密钥。

1.2 保护你的私钥

把你的私钥视为一项重要的资产，尽可能最大的使用你的私钥，限制最小的员工的访问。建议的政策包括以下内容:

在可信计算机上用足够的熵生成私有密钥。一些 CA 为您提供生成私钥的功能，请尽量不要这样做。 密码保护 key 最初就不要存储在备份系统中。私钥密码在生产环境中起不了什么作用，因为有知识的攻击者总是能够从进程内存中检索密钥。有硬件设备(被称为硬件安全模块，或 HSMs)，即使在服务器折衷的情况下，也可以保护私有密匙，但是它们是昂贵的，因此仅适用于具有严格安全性需求的组织。 妥协后，撤销旧证书并生成新密钥。 每年更新证书，如果您可以自动化过程，则更频繁。大多数网站都应该假定不可靠的妥协证书被撤销; 因此，具有较短使用寿命的证书在实践中更加安全。 除非保持相同的密钥对于公钥密钥很重要，否则每当获得新证书时，还应该生成新的私钥。

1.3 覆盖您的域名

确保您的证书涵盖您希望与网站一起使用的所有名称。您的目标是避免无效的证书警告，这会混淆用户，削弱他们的信心。

即使您期望只使用一个域名，请记住，您无法控制用户到达该网站的方式或其他人如何链接到该网站。在大多数情况下，您应该确保该证书与 www 前缀有关（例如，它适用于 example.com 和 www.example.com ）。经验法则是，安全的 Web 服务器应该具有对配置为指向它的每个 DNS 名称有效的证书。

通配符证书能满足更广泛的需求，但如果准备将密钥暴露给更多的人员，特别是跨团队或部门，则避免使用它们。换句话说，访问私钥的人越少越好。还要注意，证书共享会创建一个可以被滥用的将漏洞从一个网站或服务器传输到使用相同证书的所有其他站点和服务器（即使底层私钥不同，只要证书域名匹配）的绑定。

1.4 从可信 CA 获取证书

选择对其证书业务和安全性可靠和认真的认证中心（CA）。选择 CA 时，请考虑以下条件：

安全状态 所有CA都经过定期审核，但有些则比其他 CA 更为严重。弄清哪些在这方面做的更好并不容易，但一个选择是检查他们的安全历史，更重要的是，他们如何反应妥协，如果他们从错误中学到了经验，这将更有利。

业务重点 CA 的活动构成其业务的重要组成部分，如果事情发生严重错误，其所有事情都将丢失，并且在其他地方追逐潜在的更有利可图的机会可能不会忽视其证书部门。

提供的服务 至少，您选择的 CA 应提供对证书吊销列表（CRL）和在线证书状态协议（OCSP）撤销方法的支持，具有稳定的网

络可用性和性能。许多网站对域验证的证书感到满意，但您也应该考虑是否需要扩展验证（EV）证书。在任一种情况下，您都应该选择公钥算法。大多数网站今天使用 RSA，但由于其性能优势，ECDSA 在未来可能会变得重要。

证书管理 选项如果您需要大量证书并在复杂环境中运行，请选择一个 CA，为您提供良好的管理工具。

支持选择一个 CA，如果需要的话可以给您很好的支持。

> **注意：**
> 为了获得最佳效果，请提前获得证书，并在部署到生产之前至少一周。这种做法（1）有助于避免在计算机上没有正确时间的一些用户的证书警告;（2）有助于避免与 CA 需要额外时间的 CA 失败的撤销检查，以向 OCSP 响应者传播有效的新证书。随着时间的推移，尝试将这个“热身”期延长至 1-3 个月。同样，不要等到你的证书即将到期以替换它们。留下额外的几个月也会帮助时钟不正确的人在另一个方向。

1.5 使用强签名算法

证书安全性取决于（1）用于签署证书的私钥的强度，（2）签名中使用的散列函数的强度。直到最近，大多数证书都依赖于 SHA1 散列函数，现在被认为是不安全的。因此，我们正在向 SHA256 转型。截至 2016 年 1 月，您无法从公共 CA 获取 SHA1 证书。现有的 SHA1 证书将继续工作（在某些浏览器中有警告），但只能到 2016 年底。

2 配置

使用正确的 TLS 服务器配置，您可以确保将凭据正确呈现给站点的访问者，仅使用安全的加密原语，并减轻所有已知的缺陷。

2.1 使用完整的证书链

在大多数部署中，仅服务器证书不够的; 需要两个或多个证书来建立完整的信任链。当部署具有有效证书但没有所有必要的中间证书的服务器时，会发生常见的配置问题。为避免这种情况，只需使用 CA 提供给您的所有证书。

无效的证书链有效地使服务器证书无效并导致浏览器警告。实际上，这个问题有时难以诊断，因为一些浏览器可以重构不完整的链，有些浏览器不能重建。所有浏览器都倾向于缓存和重用中间证书。

2.2 使用安全的协议

SSL／TLS 系列中有五种协议：SSL v2，SSL v3，TLS v1.0，TLS v1.1和TLS v1.2：

SSL v2 是不安全的，不能使用。此协议版本非常糟糕，即使它们位于完全不同的服务器（DROWN 攻击）上也可以用来攻击具有相同名称的RSA 密钥和站点。 当与 HTTP（POODLE 攻击）一起使用时，SSL v3 是不安全的，当与其他协议一起使用时，SSL v3 是弱的。它也是过时的，不应该被使用。 TLS v1.0 也是不应该使用的传统协议，但在实践中通常仍然是必需的。其主要弱点（BEAST）在现代浏览器中得到缓解，但其他问题仍然存在。 TLS v1.1 和 v1.2 都没有已知的安全问题，只有 v1.2 提供了现代的加密算法。 TLS v1.2 应该是您的主要协议，因为它是唯一提供现代认证加密（也称为 AEAD）的版本。如果您今天不支持 TLS v1.2，则缺乏安全性。

为了支持较旧的客户端，您可能需要继续支持 TLS v1.0 和TLS v1.1。但是，您应该计划在不久的将来退出 TLS v1.0。例如，PCI DSS 标准将要求所有接受信用卡付款的网站在 2018 年 6 月之前移除对 TLS v1.0 的支持。

目前正在开展设计 TLS v1.3 的工作，其目的是消除所有过时和不安全的功能，并进行改进，以保持我们的通信在未来几十年内的安全。

2.3 使用安全的套件

为了安全通信，您必须首先确定您正在与所需方（而不是通过将窃听的其他人）直接沟通并安全地交换数据。在 SSL 和 TLS 中，密码套件定义了如何进行安全通信。它们由不同的建筑组成，通过多样性实现安全。如果发现其中一个构建块软弱或不安全，那么您应该可以切换到另一个。

您应该主要依靠提供强身份验证和密钥交换，前向保密和至少 128 位加密的 AEAD 套件。还有一些其他较弱的套房可能仍然得到支持，只要它们只能与不支持任何更好的老客户进行协商。

有几个过时的加密原语必须避免：

- 匿名 Diffie-Hellman（ADH）套件不提供身份验证。
- NULL 密码套件不提供加密。
- 导出密码套件在连接中协商时不安全，但也可以针对更喜欢更强大的套件（FREAK攻击）的服务器使用。
- 弱密码（通常为 40 和 56 位）的套件使用可以轻松破坏的加密。
- RC4 是不安全的。
- 3DES 缓慢而虚弱。

使用以RSA和ECDSA键为基础的以下套件配置，作为起点：

```
 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
```

> **警告：**
> 我们建议您始终首先在分段环境中测试TLS配置，仅在确定所有内容按预期工作时将更改应用到生产环境。请注意，以上是一个通用列表，并不是所有系统（特别是较旧的）支持所有套件。这就是为什么测试很重要，推荐您使用《SSL/TLS安全评估》进行检查。

上述示例配置使用标准 TLS 套件名称。一些平台使用非标准名称; 有关详细信息，请参阅您的平台的文档。例如，以下套件名称将与OpenSSL 一起使用：

```
 ECDHE-ECDSA-AES128-GCM-SHA256
ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-ECDSA-AES128-SHA
ECDHE-ECDSA-AES256-SHA
ECDHE-ECDSA-AES128-SHA256
ECDHE-ECDSA-AES256-SHA384
ECDHE-RSA-AES128-GCM-SHA256
ECDHE-RSA-AES256-GCM-SHA384
ECDHE-RSA-AES128-SHA
ECDHE-RSA-AES256-SHA
ECDHE-RSA-AES128-SHA256
ECDHE-RSA-AES256-SHA384
DHE-RSA-AES128-GCM-SHA256
DHE-RSA-AES256-GCM-SHA384
DHE-RSA-AES128-SHA
DHE-RSA-AES256-SHA
DHE-RSA-AES128-SHA256
DHE-RSA-AES256-SHA256
```

2.4 选择合适的协议

在SSL v3及更高版本的协议版本中，客户端提交他们支持的密码套件列表，服务器从列表中选择一个用于连接的套件。然而，并不是所有的服务器都做得很好，有些将从客户端列表中选择第一个支持的套件。使服务器主动选择最佳可用加密套件对于实现最佳安全性至关重要。

2.5 使用 FS

前向保密（有时也称为完全前向保密）是一种协议功能，可实现不依赖服务器私钥的安全对话。对于不提前向保密的密码套件，可以恢复服务器的私钥的人就可以解密所有较早记录的加密对话（也就是可以先大量记录密文，再解密，比如您的证书到期后没有正确销毁，它的私钥就能用来解密非PFS的密文）。您需要支持并喜欢 ECDHE 套件，以便通过现代网络浏览器实现前向保密。为了支持更广泛的客户，您还应该使用 DHE 套件作为 ECDHE 后备。避免 RSA 密钥交换，除非绝对必要。我在2.3节中提出的默认配置只包含提供前向保密的套件。

2.6 使用强的密钥交换算法

对于密钥交换，公共站点通常可以选择经典的短暂的 Diffie-Hellman密钥交换（DHE）和其椭圆曲线变体 ECDHE。还有其他的密钥交换算法，但是它们通常是以某种方式不安全的。RSA 密钥交换仍然很受欢迎，但不提供前向保密。

2015 年，一批研究人员发表了对 DHE 的新攻击; 他们的工作被称为Logjam 攻击。研究人员发现，较低强度的 DH 密钥交换（例如768 位）容易被破坏，一些知名的 1024 位 DH 组可被国家机构破坏。为了安全起见，如果部署 DHE，请至少配置 2048 位的安全性。一些较老的客户端（例如Java 6）可能不支持这种强度。出于性能原因，大多数服务器应该更喜欢 ECDHE，这是更强大和更快。在这种情况下，secp256r1命名曲线（也称为 P-256）是一个很好的选择。

3 减轻已知问题

近几年来已经发生了几次严重的 SSL 和 TLS 攻击，但是如果您正在运行最新的软件并遵循本指南的建议，那么它们通常不会关心您。（如果没有，我建议您使用 MYSSL 测试您的系统，并从中进行测试）。但是，没有什么是完全安全的，所以为了保持对安全性的了解，这是一个很好的做法。如果供应商补丁可用，请及时提供; 否则，依靠解决方案进行缓解。

4 性能

安全是我们在本指南中的主要重点，但我们也要注意表现; 一个不符合性能标准的安全服务无疑将被丢弃。通过正确配置，TLS 可以相当快。使用现代协议（例如 HTTP/2），甚至可能比明文通信更快。

4.1 避免过度安全

用于建立安全连接的密码握手是一种操作，其费用受私钥大小的高度影响。使用太短的密钥是不安全的，但使用太长的密钥将导致“太多”的安全性和缓慢的操作。对于大多数网站，使用超过 2048 位的 RSA 密钥和强大于 256 位的 ECDSA 密钥会浪费

CPU 功耗，并可能会损害用户体验。类似地，增加短暂密钥交换的强度对于 DHE 为 2048 位以及 ECDHE 为 256 位几乎没有什么好处。使用高于 128 位的加密没有明显的好处。

4.2 使用 session 恢复

会话恢复是一种性能优化技术，可以节省昂贵的密码操作的结果，并重复使用一段时间。残疾或非功能性会话恢复机制可能会引起显着的性能损失。

4.3 使用 WAN 优化和 HTTP/2

这些天，TLS 开销不是来自 CPU 饥饿的加密操作，而是来自网络延迟。只有在 TCP 握手完成后才能启动TLS握手，需要进一步交换数据包，并且离开服务器的距离更远。最小化延迟的最佳方法是避免创建新的连接 - 换句话说，保持现有的连接长时间（keep-alives）。提供良好结果的其他技术包括支持现代协议（如HTTP / 2）和使用WAN优化（通常通过内容传送网络）。

4.4 隐藏公共内容

通过TLS进行通信时，浏览器可能会认为所有流量都是敏感的。它们通常会使用内存来缓存某些资源，但一旦关闭浏览器，所有内容可能会丢失。为了获得性能提升，并能够长期缓存一些资源，将公共资源（例如图像）标记为公开。

4.5 使用 OCSP Stapling

OCSP 装订是 OCSP 协议的扩展，可以直接从服务器提供撤销信息作为 TLS 握手的一部分。因此，客户端不需要联系 OCSP 服务器进行带外验证，并且总体 TLS 连接时间显着减少。OCSP 装订是一种重要的优化技术，但您应该注意，并不是所有的网络服务器都提供了可靠的 OCSP 装订实现。结合具有缓慢或不可靠的 OCSP 响应者的 CA，这样的 Web 服务器可能会产生性能问题。为了获得最佳效果，请模拟故障条件，看看它们是否会影响您的可用性。

4.6 使用快速加密

除了提供最佳的安全性，我推荐的密码套件配置也提供了最好的性能。尽可能使用支持硬件加速 AES 的 CPU。之后，如果您真的想要进一步的性能优势（大多数网站可能不需要），请考虑使用 ECDSA 密钥。

5 HTTP 和 应用安全

HTTP 协议和 Web 应用交付的周边平台在 SSL 诞生后继续快速发展。作为这一进化的结果，该平台现在包含可用于打败加密的功能。在本节中，我们列出了这些功能，以及安全使用它们的方法。

5.1 加密无处不在

加密是可选的事实可能是今天最大的安全问题之一。我们看到以下问题：

- 没有 TLS 需要它的网站
- 具有 TLS 但不执行 TLS 的站点
- 混合 TLS 和非 TLS 内容的网站，有时甚至在同一网页内
- 编程错误的网站会颠覆 TLS

尽管如果您确切了解您正在做的事情，许多这些问题可以被缓解，可靠地保护网站通信的唯一方法是无一例外地执行加密。

5.2 消除混合内容

混合内容页面是通过 TLS 传输但是包含不通过 TLS 传输的资源（例如，JavaScript 文件，images，CSS 文件）的页面。这样的页面不安全。一个活跃的中间人（MITM）攻击者可以搭载一个单独的未受保护的 JavaScript 资源，例如劫持整个用户会话。即使您遵循上一节的建议并对整个网站加密，您仍然可能会最终从第三方网站中检索未加密的一些资源。

5.3 使用可信第三方

网站通常使用通过从另一个服务器下载的 JavaScript 代码激活的第三方服务。这种服务的一个很好的例子是 Google Analytics（分析），用于 Web 的大部分。这种包含第三方代码创建一个隐含的信任连接，有效地使对方完全控制您的网站。第三方可能不是恶意的，但是这些服务的大型提供商越来越被视为目标。推理很简单：如果大型提供程序受到威胁，攻击者将被

自动访问所有依赖该服务的站点。

如果您遵循第4.2节的建议，至少您的第三方链接将被加密，从而避免 MITM 攻击。但是，您应该进一步了解：了解您使用的服务和删除服务，将其替换为更安全的替代方案，或接受其继续使用的风险。一种称为子资源完整性（SRI）的新技术可用于通过第三方资源来减少潜在的风险。

5.4 安全 cookie

要正确安全，网站需要 TLS，而且所有的 Cookie 在创建时都被明确标记为安全的。未能保护 cookies 可以让活跃的 MITM 攻击者通过聪明的技巧来挑逗一些信息，即使在 100％ 加密的网站上也是如此。为了获得最佳效果，请考虑为您的 Cookie 添加加密完整性验证或甚至加密。

5.5 安全 HTTP 压缩

2012 年 CRIME 攻击显示 TLS 压缩无法安全实施。唯一的解决方案是完全禁用 TLS 压缩。次年，随后再发生两次攻击。TIME 和 BREACH 专注于使用 HTTP 压缩压缩的 HTTP 响应实体中的秘密。与 TLS 压缩不同，HTTP 压缩是必需的，不能关闭。因此，为了解决这些攻击，需要对应用程序代码进行更改。

TIME 和 BREACH 攻击并不容易实现，但是如果某人有足够的动力使用它们，则这种影响大致相当于成功的跨站点请求伪造（CSRF）攻击。

5.5 部署 HSTS

HTTP 严格传输安全（HSTS）是 TLS 的安全网。它旨在确保即使在配置问题和实施错误的情况下，安全性仍然保持不变。要激活 HSTS 保护，您可以向您的网站添加一个新的响应头。之后，支持 HSTS（此时所有现代浏览器）的浏览器执行它。

HSTS 的目标很简单：激活后，它不允许与使用它的网站进行任何不安全的通信。通过自动将所有明文链接转换为安全的链接，实现了这一目标。作为奖励，它还会禁用点击式证书警告。（证书警告是活动的 MITM 攻击的指标，研究表明大多数用户点击这些警告，所以绝对不要让他们感兴趣）。

添加对 HSTS 的支持是您可以为您的网站的 TLS 安全性做出的最重要的改进。新站点始终应设计为 HSTS，旧站点转换为尽可能快地支持。为了获得最佳安全性，请考虑使用 HSTS 预加载，将HSTS配置嵌入到现代浏览器中，从而使您的网站的第一个连接安全。

以下配置示例将在主主机名及其所有子域上激活一段时间为一年的 HSTS，同时还允许预加载：

```
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
```

5.6 部署 CSP

内容安全策略（CSP）是网站可以用来限制浏览器操作的安全机制。尽管最初旨在解决跨站点脚本（XSS），CSP 不断发展，并支持对增强TLS安全性有用的功能。特别地，它可以用于限制混合内容，当涉及到第三方网站，HSTS没有帮助。

要部署CSP以防止第三方混合内容，请使用以下配置：

```
Content-Security-Policy: default-src https: 'unsafe-inline' 'unsafe-eval';
                         connect-src https: wss:
```

> **注意：**
> 这不是部署 CSP 的最佳方法。为了提供不破坏混合内容以外的任何内容的示例，我不得不禁用某些默认安全功能。随着时间的推移，当您了解 CSP 的更多信息时，您应该更改您的策略以使其恢复。

5.7 不要缓存敏感内容

所有敏感内容必须仅传达给预定方，并由所有设备进行相应处理。虽然代理没有看到加密的流量，并且不能在用户之间共享内容，但是使用基于云的应用交付平台正在增加，这就是为什么在指定什么是公共的时候需要非常小心的是什么。

5.8 考虑其它威胁

TLS 旨在仅解决安全机密和您与用户之间通信的完整性的一个方面，但还有许多其他威胁需要处理。在大多数情况下，这意味着确保您的网站没有其他弱点。

6 验证

有许多配置参数可用于调整，预先知道某些变化会产生什么影响。此外，有时会意外地进行更改; 软件升级可以静默地引入更改。因此，我们建议您最初使用全面的 SSL/TLS 评估工具来验证您的配置，以确保您开始安全，然后定期确保您保持安全。对于公共网站，我们建议您免费使用SSL实验室服务器测试。

6.1 高级主题

以下高级主题目前不在我们的指南范围之内。他们需要更深入地了解 SSL/TLS 和公钥基础设施（PKI），而且他们仍然被专家辩论。

6.2 使用 HPKP

公共密钥固定旨在使网站运营商有权限制哪些 CA 可以为其网站颁发证书。Google 已经部署了这个功能了一段时间（硬编码到他们的浏览器，Chrome），并且已被证明是非常有用的，以防止攻击并使公众了解它们。在 2014 年，Firefox 还增加了对硬编码固定的支持。现在可以使用一种称为 HTTP 的公钥固定扩展标准。公钥绑定解决了 PKI 最大的弱点（事实上，任何 CA 都可以为任何网站发布证书），但是这是一个成本; 部署需要大量精力和专业知识，并造成失去对您站点控制的风险（如果最终导致无效的固定配置）。你应该考虑固定很大程度上只有当你

6.3 使用 DNSSEC 和 DANE

域名系统安全扩展（DNSSEC）是一种增加域名系统完整性的技术。今天，一个活跃的网络攻击者可以轻松地劫持任何 DNS 请求并伪造任意的响应。使用 DNSSEC，所有响应都可以加密地跟踪到 DNS 根目录。命名实体的基于 DNS 的身份验证（DANE）是建立在 DNSSEC 之上的单独标准，用于提供 DNS 和 TLS 之间的绑定。DANE 可用于增强现有基于 CA 的 PKI 生态系统的安全性，或者完全绕过它。

即使不是每个人都同意，DNSSEC 是互联网的一个很好的方向，但对其的支持仍在继续改善。浏览器还不支持 DNSSEC 或 DANE（更喜欢 HSTS 和 HPKP 提供的类似功能），但有一些迹象表明它们正在开始用于提高电子邮件传递的安全性。

安全加固

当你的网站上了 HTTPS 以后，可否觉得网站已经安全了？这里提供了一个 HTTPS 是否安全的检测工具，你可以试试。

本篇正文讲述的是 HTTP 安全的典型实践，着重在于 HTTPS 网站的 Header 的相关配置。

1 连接安全性和加密

1.1 SSL/TLS

传输层安全（TLS）及其前身安全套接字层（SSL），通过在浏览器和 web 服务器之间提供端到端加密来促进机密通信。没有 TLS，就谈不上什么安全。TLS 是 HTTP 安全性的基础。

想要部署 TLS 是非常容易的，但其难点在于如何使用安全的配置来保障站点的安全。 尤其是 Protocol 版本和 Cipher 需要小心选择和配置。你可以通过本站 工具 体检你的网站，发现并解决这些细节的问题。

建议

所有本地和链接的资源需要正确的配置，且要使用 TLS。

1.2 HTTP Strict Transport Security (HSTS)

指示浏览器只使用 HTTPS 连接到目标服务器。这可以防止一些潜在的中间人攻击，包括 SSL 剥离，会话 cookie 窃取（如果没有被 适当保护）。如果遇到任何与证书相关的错误，它还可以阻止浏览器连接到网站。当浏览器访问一个设置相应 HTTP header 的 HTTPS 网站时，HSTS 将被激活。

HSTS 有一个固定期限，由 max-age 字段值控制。这个值可以是静态的，也可以是相对于将来某个特定日期的，你可以设置成 SSL 证书的过期时间。

在浏览器中，HSTS 首选项可以通过提交到 Chromium's HSTS preload list 来硬编码，这是所有实现 HSTS 使用的浏览器。

注意，HSTS 确实有陷阱。它提供了 include subdomains 选项，这在实践中可能是太宽泛了。此外，客户端错误可能会造成严重的后果——客户端错误的时钟导致它认为服务器的 SSL 证书无效或过期，或者缺少根 CA 证书——将不再导致浏览器中的证书错误。浏览器将完全拒绝访问页面，并且可能会显示让安全专家之外的完全无法理解的错误。

建议

设置 HSTS header 长的生命周期，最好是半年及以上。

```
Strict-Transport-Security: max-age=31536000
```

1.3 Public Key Pins

HTTP PKP（HPKP）指示浏览器只与提供的 SSL/TLS 的 HASH 相符或存在于同一证书链的服务器相连接。换句话说，如果 SSL/TLS 证书以一种意想不到的方式发生了变化，浏览器就无法连接到主机。这主要是针对受信任证书颁发机构（CA）或流氓 CA 证书颁发的伪造证书，用户可能会被骗安装。

例如，浏览器连接到 https://example.com，它存在这个头。header 告诉浏览器，如果证书 key 匹配，或者在发出证书链中有一个 key 匹配，那么在将来才会再次连接。其他的指令组合是可能的。它们都极大地减少了攻击者在客户端和合法主机之间模拟主机或拦截通信的可能性。

像 HSTS 一样，HPKP 在实现之前需要仔细的思考和计划。错误可以将用户锁定在您的站点之外，并且不容易修复。

建议

确定是否需要为您的站点使用 PKP。如果是这样的话，那么从一个较小的实践开始，如果在一段时间之后没有遇到问题，就增加它。如果 SSL/TLS 密钥需要更新，建立备份计划。优先创建备份密钥和离线存储。

示例HTTP头:

```
Public-Key-Pins: max-age=5184000; pin-sha256="+oZq/vo3Kcv0CQPjpdwyInqVXmLiobmUJ3FaDpD/U6c="; pin-sha256="47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU="
```

1.4 Mixed HTTPS and HTTP Content

主站点通过 HTTPS 安全地服务，但是在 HTTP 上加载一些文件（images、js、css）。这是一个巨大的安全漏洞，破坏了 HTTPS 提供的安全性。受影响的站点可能会泄漏会话 cookie 或用户行为信息。它们也可能容易受到注入和其他 MITM 攻击的攻击，而 HTTPS 通常会阻止这种攻击。

建议

如果 HTTPS 部署在主站上，请将任何地方的所有内容都 HTTPS 化（全站 HTTPS）。

2 Content security

2.1 Content Security Policy

为浏览器提供关于网站内容类型和行为的明确说明。良好的内容安全策略（CSP）可以帮助抵御跨站点脚本（XSS）和其他注入攻击等攻击。CSP 支持所有主要的浏览器，尽管只是部分地之前在 IE 11。

一个好的 CSP 是基于白名单的方法，不允许任何东西，除了明确允许的内容。它还限制了 javascript 的来源和允许操作。

CSP 很难启用遗留代码库。为了简化实现，CSP 提供了一个 report-only 模式，在浏览器中，CSP 的违规被发送到一个网站端点，但是该策略没有被强制执行。 新项目应该从一开始就使用 CSP。

建议

从限制性政策开始，在必要时放松。禁止所有的例子:

```
Content-Security-Policy: default-src 'none';
```

现在让我们允许自托管 scripts、images、CSS、fonts 和 AJAX，以及 jQuery CDN 托管脚本和 Google Analytics:

```
Content-Security-Policy: default-src 'none'; script-src 'self' https://code.jquery.com https://www.google-analytics.com;
img-src 'self' https://www.google-analytics.com; connect-src 'self'; font-src 'self'; style-src 'self';
```

要注意的是，不要让所有的东西都破坏你的网站，例如，如果你使用 `child-src` 指令，而浏览器不支持它。一个不那么严格的政策可能从以下开始:

```
Content-Security-Policy: default-src 'self';
```

甚至更少的限制性政策甚至可以使用 `default-src '*'`，然后添加限制。我建议你不要这么做，除非你完全明白其中的含义。否则，你可能会依赖 CSP，它只会给你一种错误的安全感。

2.2 Frame Options

控制站点是否可以放置在 `<iframe>`，`<frame>` 或 `<object>` 标签。不允许使用框架可以防止 clickjacking 攻击。例如，从 2015 年 2 月起，Internet Explorer's universal cross-site-scripting bug 就被这个消息头减轻了。

X-Frame-Options 是一个非标准的 header，在内容安全策略级别 2 中被 frame ancestor 指令所取代。然而，frame ancestor 还没有得到普遍的支持，而 X-Frame-Options 得到了广泛的支持。

建议

确定你的网站是否需要被允许呈现在一个 frame 中。完全不允许使用 sameorigin 拒绝或允许同源框架的选项。避免由于受限或 bug 浏览器支持而允许的选项。示例 HTTP 头:

```
X-Frame-Options: deny
```

2.3 XSS Protection

跨站点脚本（XSS 或 CSS）的保护被构建到大多数流行的浏览器中，除了 Firefox 之外。这种保护是用户可配置的，可以关闭。因此，明确要求浏览器在你的网站上使用它的 XSS 过滤器是个好主意。

相反，网站可以要求 XSS 保护在页面的基础上被禁用。这绝对不是一个好主意。

建议

使用入校 HTTP header:

```
X-Xss-Protection: 1; block
```

2.4 Cache Control

表示缓存页面输出的首选项。适当的值随网站数据的性质而变化，但强烈推荐使用偏好。否则，它取决于浏览器和代理来选择是否缓存内容。不恰当的选择可能会导致性能问题、安全问题，或者两者都有。

建议

开发缓存策略，然后将缓存首选项包括为 HTTP 头。

```
Cache-Control: public*
```

其中的一个 public，private，no-cache 或 no-store。如果允许缓存，则应该将 max-age 值包含在 Cache-Control 以及 Etag 头文件中，以允许客户端缓存验证。

2.5 Content Type Options

当浏览器以不同的方式处理来自服务器的文件时，MIME 嗅探就是服务器指令。当一个网站承载不受信任的内容（如用户提供的）时，这是很危险的。假设服务器允许用户上传 image。如果用户上传 HTML 文档，浏览器可能会将其呈现为 web 执行 scriptpage，即使服务器明确表示正在发送 image。非标准的标头 `X-Content-Type-Options` 选项指示浏览器不做任何模仿指定类型的 MIME。

建议

总是设置 header:

```
X-Content-Type-Options: nosniff
```

2.6 Subresource Integrity

浏览器通常从外部域加载大量资源、javascript 和样式表。内容交付网络经常被使用。如果外部资源被破坏，依赖站点的安全性也可以。子资源完整性允许浏览器验证 javascript 或样式表未被意外修改。

建议

设置外部 javascript 和样式表的完整性属性。

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js" integrity="sha384-6ePHh72RI3hKio4HiJ841psfsRJveeS+aLoaEf3BWfS+gTF0XdAqku2ka8VddikM"></script>
```

注意

您应该始终提供外部脚本的本地副本，并实现一种方法，以便在外部负载失败的情况下重新加载它们。否则你的网站可能会崩溃。例子:

```
<script>window.jQuery || document.write('<script src="/jquery.min.js"><\/script>')</script>
```

2.7 Iframe Sandbox

iframe 在 WWW 上随处可见。网站平均有 5.1 iframe，主要用于装载第三方内容。这些 iframe 有很多方法来伤害托管网站，包括运行脚本和插件和重新引导访问者。`sandbox` 属性允许对 iframe 中可以进行的操作进行限制。

建议

设置 iframe 的 `sandbox` 属性，然后添加所需的权限。

```
<iframe src="https://example.com" sandbox="allow-same-origin allow-scripts"></script>
```

2.8 Server Clock

服务器包括所有响应的时间戳。不准确的时钟不会给客户机浏览器带来问题。然而，当与其他系统或服务交互时，问题就会出现。

建议

使用网络时间协议（NTP）来保持服务器时钟的准确性。

3 Information disclosure

3.1 Server Banner

大多数 web 服务器设置报头来识别自己和他们的版本号。这只服务于信息目的和实际用途是非常有限的。去掉整个头，而完全可以接受，通常是不必要的。但是，建议从头中删除版本号。在特定 web 服务器版本中存在 bug 的情况下，包括版本号可以作为对脚本 kiddy 的邀请来尝试对服务器的攻击。

建议

包含服务器名称但去掉版本号；

```
Server: nginx
```

3.2 Web Framework Information

许多 web 框架设置 HTTP 头，识别框架或版本号。除了满足用户的好奇心，而且主要作为技术堆栈的广告，这几乎没有什么作用。这些头是不标准的，对浏览器渲染站点的方式没有影响。

虽然它们没有什么实际用途，但对于搜索运行过时版本的软件的机器人或蜘蛛来说，这些标头是无价的，因为这些软件可能包含安全漏洞。如果没有定期更新，这些头文件可以使网站的目标变得容易。

建议

从服务器响应中删除这些标头: `X-Powered-By`, `X-Runtime`, `X-Version` 和 `X-AspNet-Version`。

4 Cookies

4.1 Cookie Security

包含敏感信息的 cookie，特别是会话 id，需要标记为安全的，假设网站是通过 HTTPS 传输的。这会阻止 cookie 通过 HTTP 发送明文文本。另一种方法是通过 HSTS 来阻止非安全 cookie 在 HTTP 上传输。建议使用安全 cookie 和 HSTS。

会话 cookie 应该与 HttpOnly 值进行标记，以防止它们被 javascript 访问。这可以防止攻击者利用 XSS 窃取会话 cookie。其他 cookie 可能不需要这样标记。但是，除非有明确的需要从 javascript 中访问他们的值，否则最好还是呆在安全的一边，把所有 cookie标记为 HttpOnly

建议

标记所有 cookie 安全和 HttpOnly。

```
Set-Cookie: Key=Value; path=/; secure; HttpOnly, Key2=Value2; secure; HttpOnly
```

服务器软件

我们在典型实践文章中建议大家如何去配置协议和密码套件，但是如果服务器软件（nginx、apache等）所使用的ssl协议库存在SSL漏洞，或者不支持那些现代化的密码套件和特性，那么无论你如何去修改配置都无法改善现在的安全问题。

所以我们在配置前，或者发现按照推荐配置进行了调整《SSL/TLS安全评估报告》还是无法满足要求，那么可以检查下所使用的OpenSSL等加密库是否版本过低。

如何检查OpenSSL版本

nginx

`nginx -V`

```
 nginx version: nginx/1.10.2
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-4) (GCC)
built with OpenSSL 1.0.1e-fips 11 Feb 2013
TLS SNI support enabled
```

或者通过openssl命令查看(适用于非自己通过openssl源码编译的)

```
openssl version
```

推荐的OpenSSL版本

1. OpenSSL 1.0.2用户需更新到1.0.2h 以上。
2. OpenSSL 1.0.1用户需更新到1.0.1t 以上。
3. OpenSSL官方已停止对 0.9.8和 1.0.0 两个版本的升级维护，请使用这两个版本的用户将其升级至1.0.2h版本以上。

> OpenSSL 1.0.1以下不支持tls1.2
> 升级前请做好测试

漏洞事件

心血漏洞

OpenSSL 心血（HeartBleed）漏洞是openssl在 2014-04-07 公布的重大安全漏洞（CVE-2014-0160）这个漏洞使攻击者能够从服务器内存中读取64 KB的数据，甚至获取到加密流量的密钥，用户的名字和密码，以及访问的内容。

- OpenSSL 1.0.1g 已修复该漏洞
- OpenSSL 1.0.0 分支版本不受此漏洞影响
- OpenSSL 0.9.8 分支版本不受此漏洞影响
- OpenSSL 1.0.2 Beta2 不受此漏洞影响

Heartbleed检测工具>>

水牢漏洞（DROWN跨协议攻击TLS漏洞）

水牢漏洞可以允许攻击者破坏使用SSLv2协议进行加密的HTTPS网站，读取经加密传输的敏感通信，包括密码、信用卡账号、商业机密、金融数据等。

- OpenSSL 1.0.1h+
- OpenSSL 1.0.0m+
- OpenSSL 0.9.8za+

“密文填塞”（Padding Oracle）漏洞

“密文填塞”（Padding Oracle）漏洞，只要网络连接采用的是AES-CBC密码和支持AES-NI的服务器,那么MITM攻击者就可以使用Padding Oracle攻击解密通信

- OpenSSL 1.0.2用户需更新到1.0.2h 。
- OpenSSL 1.0.1用户需更新到1.0.1t 。
- 使用包管理系统的用户可以直接更新到2016年5月3日 之后的版本。

CBC padding oracle检测 检测工具>>

CSS 注入漏洞

OpenSSL CCS漏洞，此漏洞是 OpenSSL ChangeCipherSpec 设计缺陷造成，被称为 CCS 注入漏洞。

- OpenSSL 1.1.0 应升级到 1.1.0a 或更高版本。
- OpenSSL 1.0.2 应升级到 1.0.2i 或更高版本。
- OpenSSL 1.0.1 应升级到 1.0.1u 或更高版本。

OpenSSL CCS 检测工具>>

注意事项

OpenSSL官方已停止对 0.9.8和 1.0.0 两个版本的升级维护，请使用这两个版本的用户将其升级至更高版本。

OpenSSL心血漏洞（Heartbleed）修复方案

概述

OpenSSL 心血(HeartBleed)漏洞 是openssl 在 2014-04-07 公布的重大安全漏洞（CVE-2014-0160）这个漏洞使攻击者能够从服务器内存中读取64 KB的数据，甚至获取到加密流量的密钥，用户的名字和密码，以及访问的内容。

主要影响版本 OpenSSL 1.0.1 到 OpenSSL 1.0.1f 以及 OpenSSL 1.0.2 Beta1

不受此漏洞影响的 OpenSSL版本信息：

- OpenSSL 1.0.1g 已修复该漏洞
- OpenSSL 1.0.0 分支版本不受此漏洞影响
- OpenSSL 0.9.8 分支版本不受此漏洞影响
- OpenSSL 1.0.2 Beta2 不受此漏洞影响

确认站点是否存在此漏洞

1. 可以直接利用在线工具检测 https 站点是否存在此漏洞(快捷、准确)。

2. 查看提供加密服务的OpenSSL版本是否在受影响的版本范围（1.0.1—1.0.1f / 1.0.2 Beta1）

操作系统openssl版本查看：

命令：`openssl version`

结果：OpenSSL 1.0.0-fips 29 Mar 2010

nginx openssl版本查看：

命令： `nginx –V`

结果：

nginx version: nginx/1.5.13

built by gcc 4.4.6 20120305 (Red Hat 4.4.6-4) (GCC)

TLS SNI support enabled

configure arguments: --prefix=/usr/local/nginx-1.5.13 --with-select_module --with-http_ssl_module --with-openssl=/home/user/openssl-1.0.1g --with-http_spdy_module --with-pcre=/home/user/pcre-8.33 --with-zlib=/home/user/zlib-1.2.8

windows Apache openssl版本查看：

命令：`D:\httpd-2.4.7-x64\Apache24\bin>openssl version`

结果：

WARNING: can't open config file: /apache24/conf/openssl.cnf

OpenSSL 1.0.1e 11 Feb 2013

修复方案

Openssl:

升级OpenSSL 1.0.1g

Nginx:

官方在 2014-04-08 发布了最新版本 1.5.13 修复了此漏洞请升级至最新版本

Apache：

建议下载最新版本 openssl 和 Apache 重新编译安装。

其他受影响的WebServer：

建议到官方更新不受此漏洞影响的版本。

漏洞分析

OpenSSL 是 Apache 和 nginx 网络服务器的默认安全协议，此外大量操作系统、电子邮件和即时通讯系统也采用OpenSSL加密用户数据 通讯。而此次发现的bug已经存在两年之久，这意味着攻击者可以利用该bug获取大量互联网服务器与用户之间的数字证书私钥，从而获取用户账户密码等敏感 数据。由于攻击者不会在服务器日志中留下痕迹，因此网站系统管理员将无法得知系统漏洞是否已经被黑客利用，也无从得知用户数据和账号是否已经被黑客扫描获

OpenSSL已经发布了1.0.1g修正bug，Debian发行版也在半小时修复了bug，Fedora发布了一个权宜的修正方案。 该bug是在2011年引入到OpenSSL中，使用OpenSSL 0.9.8的发行版不受影响，但Debian Wheezy、Ubuntu 12.04.4、 Centos 6.5、Fedora 18、SuSE 12.2、OpenBSD 5.4、FreeBSD 8.4和NetBSD 5.0.2之后的版本都受到影响。如果你运行存在该bug的系统，那么最好废除所有密钥。

检测OpenSSL-DROWN漏洞

最近OpenSSL官方发布了新的安全公告，公告中提及修复了一个高危漏洞——DROWN跨协议攻击TLS漏洞（水牢漏洞）。

DROWN漏洞的影响

可以允许攻击者破坏使用**SSLv2协议**进行加密的HTTPS网站，读取经加密传输的敏感通信，包括密码、信用卡账号、商业机密、金融数据等。

DROWN漏洞的利用难度较高

DROWN漏洞的潜在影响虽然严重，但需要满足多个条件才能被利用：

1. 漏洞只存在于支持 SSLv2协议的服务端中，而这个是一个古老的协议，官方已经建议禁用；
2. 依靠OpenSSL的应用程序必须配置使用基于DSA的group去生成基于Diffie Hellman密钥交换的临时密钥;
3. 利用paddingoracle的攻击方式解密密文，破解密钥需要使用一定性能的计算集群。

DROWN漏洞处理建议

通过DROWN漏洞的攻击原理可以知道：只要服务端支持SSLv2 弱强度加密组件就容易受到影响。

SSLv2是一种古老的协议，官方已经建议禁用SSLv2，并且Microsoft (Windows Server):iis 7和以上的版本默认已经禁止了SSLv2，实践中也有许多客户端已经不支持使用SSLv2。但是由于错误配置，许多网站仍然支持SSLv2。

建议这些网站及时升级openssl，并禁用所有服务器中的SSLv2版本协议。

1. 在CentOS、Redhat系统，可以通过如下命令升级：

```
#### yum update openssl
```

2. ubuntu等版本可以通过如下命令升级：

```
#### apt-get upgrade opensll
```

3. 通过如下配置禁用apache的SSLv2:

```
SSLProtocol all -SSLv2
```

4. 通过如下配置限制nginx中只使用TLS协议:

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
```

当然了，我们也有关于DROWN攻击的好消息带给大家 – 幸好这一漏洞是由安全研究人员所发现的。但坏消息就是，既然这一漏洞的详细信息已经被公开了，那么不出意外的话，攻击者很快就会利用这项攻击技术来对网络中的服务器进行攻击。

密文填塞漏洞

在OpenSSL官方(2016/5/3)发布的安全公告中，公开了新的高危漏洞CVE-2016-2107。虽然此漏洞的利用难度很高，但是一旦在实际中被利用，可以窃取到用户数据、凭据、财务和个人信息。

漏洞情况分析

(CVE-2016-2107)："密文填塞"（Padding Oracle）漏洞

OpenSSL公布的说明中这样说，"只要网络连接采用的是AES-CBC密码和支持AES-NI的服务器,那么MITM攻击者就可以使用Padding Oracle攻击解密通信。"

据安全公司High-Tech Bridge进行的一项分析显示，Alexa排名前10000的网站中有许多易受到MITM攻击,与1829家顶级网站(占比18.29%)相连的网络和邮件服务器都是脆弱易感的。

漏洞利用难度很苛刻

这是一种中间人攻击的方式，攻击者首先要满足中间人攻击能达成的以下条件：

- 能控制受害者进行多次通信连接
- 能在受害者明文头部添加数据（加密前）
- 能修改受害者明文数据（加密前）
- 能截断和修改受害者发送的密文
- 能获得服务器返回的数据

检测方式

可使用在线漏洞检测工具，一键检测您的站点是否存在漏洞：

Padding Oracle漏洞处理建议

受影响版本：

- OpenSSL 1.0.2 < 1.0.2h
- OpenSSL 1.0.1 < 1.0.1t
- OpenSSL 1.0.0
- OpenSSL 0.9.8

将 openssl 升级到以下版本可修复漏洞：

- OpenSSL 1.0.2用户需更新到1.0.2h 。
- OpenSSL 1.0.1用户需更新到1.0.1t 。

使用包管理系统的用户可以直接更新到2016年5月3日之后的版本。

特别注意！

- 升级前请做好测试。

- OpenSSL官方已停止对 0.9.8和 1.0.0 两个版本的升级维护，请使用这两个版本的用户将其升级至 1.0.2h 版本。

OpenSSL-CCS注入漏洞修复方案

关于 OpenSSL CCS 漏洞

在昨天 2014-06-05 OpenSSL 发布了关于漏洞 CVE-2014-0224 的安全公告，并发布了已修复此漏洞的最新 OpenSSL版本。由于此漏洞是 OPenSSL ChangeCipherSpec 设计缺陷造成，被称为 CCS 注入漏洞。攻击者可以发起中间人攻击并利用此漏洞篡改或监听SSL加密传输的数据。

受影响的OpenSSL版本包括：

- OpenSSL 1.0.1 through 1.0.1g
- OpenSSL 1.0.0 through 1.0.0l
- all versions before OpenSSL 0.9.8y

未影响版本：

- OpenSSL 1.0.1h
- OpenSSL 1.0.0m
- OpenSSL 0.9.8za

检测漏洞

您可通过在线工具检测SSL服务端是否存在这个漏洞。

应对措施

Windows环境下的apache及nginx等使用openssl的webserver直接重新下载最新的webserver版本就可解决。

Linux环境下

- **Nginx：**

使用ldd nginx指令检查是否有的libssl.so和libcrypto.so的调用，如果用，直接升级openssl就可以了，如果没有，那么需要重新编译Nginx

- Apache：
  - 如果你的apache是使用管理安装的，那么直接使用包管理升级openssl就可以了。
  - 如果你的apache是自己编译的，那么您需要检查apache的编译参数，看有没有指定自己的openssl目录；
  - 如果没有那么也可以直接使用包管升级openssl。
  - 如果您编译安装的时候使用了自己的openssl目录而非系统的openssl，那请升级对应目录的openssl或重新编译apache.

  升级openssl：对于使用包管理安装的openssl直接使用包管理升级就可以了，升级后版本号可能不在安全的版本号之只，但是它们是被打过补丁的，是安全的。

  从源码编译安装openssl：从openssl下载自己希望使用的openssl版本的最新文件。编译时请加上参数shared以打开共享库。完成后需按自己的环境来安装。

- 其他受影响的WebServer：

  建议到官方更新不受此漏洞影响的版本。

漏洞分析

SSL握手过程中两端会发起**ClientHello**和**ServerHello**握手消息。

在握手过程中双方会协商一些会话参数,如协议版本、加密套件、会话密钥等。SSL协议中允许双方在握手阶段通过使用ChangeCipherSpec（CCS）来修改连接的加密策略。

按照标准，CCS消息应该是在 **握手加密参数协商完成之后** 和 **最终确认消息发送之前** 来发送。但Openssl没有这么设计，它允许CCS消息在加密参数协商完成之前发送。

中间人攻击可以利用这点，在一个 SSL 握手过程中向客户端和服务端分别发送一个 CCS 包并用长度为零的预主密钥来协商会话密钥，这样攻击者就可以知道会话密钥并可以篡改或 截获SSL通信数据。

API参考

概述

欢迎使用百度智能云的核心产品——百度智能云SSL 证书服务。您可以使用本文档介绍的API对SSL证书进行灵活的操作。

通用说明

API调用遵循HTTP协议，CAS为全局服务，使用统一一个域名，具体域名为cas.baidubce.com。 数据交换格式为JSON，所有request/response body内容均采用UTF-8编码。

API认证机制

所有API的安全认证一律采用Access Key与请求签名机制。 Access Key由Access Key ID和Secret Access Key组成，均为字符串。 对于每个HTTP请求，使用下面所描述的算法生成一个认证字符串。提交认证字符串放在Authorization头域里。服务端根据生成算法验证认证字符串的正确性。 认证字符串的格式为`bce-auth-v{version}/{accessKeyId}/{timestamp}/{expirationPeriodInSeconds}/{signedHeaders}/{signature}`。

- version是正整数。
- timestamp是生成签名时的UTC时间。
- expirationPeriodInSeconds表示签名有效期限。
- signedHeaders是签名算法中涉及到的头域列表。头域名之间用分号（;）分隔，如host;x-bce-date。列表按照字典序排列。（本API签名仅使用host和x-bce-date两个header）
- signature是256位签名的十六进制表示，由64个小写字母组成。

当百度云接收到用户的请求后，系统将使用相同的SK和同样的认证机制生成认证字符串，并与用户请求中包含的认证字符串进行比对。如果认证字符串相同，系统认为用户拥有指定的操作权限，并执行相关操作；如果认证字符串不同，系统将忽略该操作并返回错误码。

鉴权认证机制的详细内容请参见鉴权认证机制。

通信协议

支持HTTP和HTTPS两种调用方式。为了提升数据的安全性，建议通过HTTPS调用。

请求结构说明

数据交换格式为JSON，所有request/response body内容均采用UTF-8编码。

请求参数包括如下4种：

参数类型	说明
URI	通常用于指明操作实体，如:POST /v{version}/instance/{instanceId}
Query参数	URL中携带的请求参数，通常用来指明要对实体进行的动作
HEADER	通过HTTP头域传入，如:x-bce-date
RequestBody	通过JSON格式组织的请求数据体

响应结构说明

响应值分为两种形式：

响应内容	说明
HTTP STATUS CODE	如200,400,403,404等
ResponseBody	JSON格式组织的响应数据体

API版本号

参数	类型	参数位置	描述	是否必须
version	String	URI参数	API版本号，当前值为3	必须

幂等性

当调用创建资源的接口时，如果遇到了请求超时或服务器内部错误，用户可能会尝试重发请求，导致资源的超量创建。这时用户可以通过clientToken参数避免创建出比预期要多的资源，即保证请求的幂等性。

幂等性基于clientToken，clientToken是一个长度不超过64位的ASCII字符串，通常放在query string里，如`http://cas.baidubce.com/v1/instance?clientToken=be31b98c-5e41-4838-9830-9be700de5a20`。

如果用户使用同一个clientToken值调用创建接口，则服务端会返回相同的请求结果。因此用户在遇到错误进行重试的时候，可以通过提供相同的clientToken值，来确保只创建一个资源；如果用户提供了一个已经使用过的clientToken，但其他请求参数（包括queryString和requestBody）不同甚至url Path不同，则会返回IdempotentParameterMismatch的错误代码。

clientToken的有效期为24小时，以服务端最后一次收到该clientToken为准。也就是说，如果客户端不断发送同一个clientToken，那么该clientToken将长期有效。

日期与时间规范

日期与时间的表示有多种方式。为统一起见，除非是约定俗成或者有相应规范的，凡需要日期时间表示的地方一律采用UTC时间，遵循ISO 8601，并做以下约束：

- 表示日期一律采用`YYYY-MM-DD`方式，例如`2014-06-01`表示2014年6月1日。
- 表示时间一律采用`hh:mm:ss`方式，并在最后加一个大写字母Z表示UTC时间。例如`23:00:10Z`表示UTC时间23点0分10秒。
- 凡涉及日期和时间合并表示时，在两者中间加大写字母T，例如`2014-06-01T23:00:10Z`表示UTC时间2014年6月1日23点0分10秒。

规范化字符串

通常一个字符串中可以包含任何Unicode字符。在编程中这种灵活性会带来不少困扰。因此引入“规范字符串”的概念。一个规范字符串只包含百分号编码字符以及URI（Uniform Resource Identifier）非保留字符（Unreserved Characters）。 RFC 3986规定URI非保留字符包括以下字符：字母（A-Z，a-z）、数字（0-9）、连字号（-）、点号（.）、下划线（_）、波浪线（~）。

将任意一个字符串转换为规范字符串的方式是:

- 将字符串转换成UTF-8编码的字节流。

- 保留所有URI非保留字符原样不变。
- 对其余字节做一次RFC 3986中规定的百分号编码（Percent-Encoding），即一个%后面跟着两个表示该字节值的十六进制字母。字母一律采用大写形式。

示例： 原字符串：`this is an example for 测试`，对应的规范字符串：`this%20is%20an%20example%20for%20%E6%B5%8B%E8%AF%95`。

服务域名

区域	服务端点Endpoint	协议
全局	cas.baidubce.com	HTTP and HTTPS

公共请求头与公共响应头

公共请求头

头域	说明	是否必须
Authorization	包含Access Key与请求签名。具体请参考鉴权认证	必须
Content-Type	application/json; charset=utf-8	必须
x-bce-date	该请求创建的时间，表示日期一律采用YYYY-MM-DD方式，例如2014-06-01表示2014年6月1日。如果用户使用了标准的Date域，该头域可以不填。当两者同时存在时，以x-bce-date为准。	必须

公共响应头

头域	说明
Content-Type	application/json; charset=utf-8
x-bce-request-id	对应请求的requestId

错误码

错误码格式

当用户访问API出现错误时，会返回给用户相应的错误码和错误信息，便于定位问题，并做出适当的处理。请求发生错误时通过Response Body返回详细错误信息，遵循如下格式：

参数名	类型	说明
code	String	表示具体错误类型。
message	String	有关该错误的详细说明。
requestId	String	导致该错误的requestId。

例如：

```
 {
   "code":"IllegalRequestUrl",
   "message":"The requested url belongs to domain which is not under acceleration",
   "requestId":" 81d0b05f-5ad4-1f22-8068-d5c9de60a1d7"
}
```

公共错误码

错误码	错误消息	HTTP状态码	描述
AccessDenied	Access denied.	403 Forbidden	无权限访问对应的资源。
InappropriateJSON	The JSON you provided was well-formed and valid, but not appropriate for this operation.	400 Bad Request	请求中的JSON格式正确，但语义上不符合要求。如缺少某个必需项，或者值类型不匹配等。出于兼容性考虑，对于所有无法识别的项应直接忽略，不应该返回这个错误。
InternalError	We encountered an internal error. Please try again.	500Internal Server Error	所有未定义的其他错误。在有明确对应的其他类型的错误时（包括通用的和服务自定义的）不应该使用。
InvalidAccessKeyId	The Access Key ID you provided does not exist in our records.	403 Forbidden	Access Key ID不存在。
InvalidHTTPAuthHeader	The HTTP authorization header is invalid. Consult the service documentation for details.	400 Bad Request	Authorization头域格式错误。
InvalidHTTPRequest	There was an error in the body of your HTTP request.	400 Bad Request	HTTP body格式错误。例如不符合指定的Encoding等。
InvalidURI	Could not parse the specified URI.	400 Bad Request	URI形式不正确。例如一些服务定义的关键词不匹配等。对于ID不匹配等问题，应定义更加具体的错误码，例如NoSuchKey。
MalformedJSON	The JSON you provided was not well-formed.	400 Bad Request	JSON格式不合法。
InvalidVersion	The API version specified was invalid.	404 Not Found	URI的版本号不合法。
OptInRequired	A subscription for the service is required.	403 Forbidden	没有开通对应的服务。
PreconditionFailed	The specified If-Match header doesn't match the ETag header.	412 Precondition Failed	详见ETag。
RequestExpired	Request has expired. Timestamp date is XXX.	400 Bad Request	请求超时。XXX要改成x-bce-date的值。如果请求中只有Date，则需要将Date转换为datetime。
Idempot			

entParameterMismatch	The request uses the same client token as a previous, but non-identical request.	403 Forbidden	clientToken对应的API参数不一样。
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided. Check your Secret Access Key and signing method. Consult the service documentation for details.	400 Bad Request	Authorization头域中附带的签名和服务端验证不一致。

查询相关接口

获取已购证书列表

接口描述

本接口用于查询用户已购买的证书，使用分页

请求结构

```
 > GET http://cas.baidubce.com/v3/openapi/query/?pageNo=1&pageSize=3
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
pageNo	int	是	Query参数	页码数
pageSize	int	是	Query参数	单页大小，最大1000
brand	String	否	Query参数	证书品牌
certType	String	否	Query参数	证书类型 选值见附录 CertType
status	String	否	Query参数	证书状态 选值见附录 CertStatus

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
totalCount	int	一共有证书数量
result	list	证书列表

请求示例

```
 > GET http://cas.baidubce.com/v3/openapi/query/?pageNo=1&pageSize=3
> Authorization: bce-auth-v1/5c36beca06ab4616b7969fe0e768d660/2020-04-27T02:43:18Z/3600/host;x-bce-console-rpc-id;x-bce-date/841f0ebe49d9b3fd2abfc33b71d53e534a6460fc55e6bfbe96a6f1cc5d163ae6
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

响应示例

```
 < Cache-Control: no-cache
< Content-Length: -1
< Content-Type: application/json;charset=UTF-8
< Date: Mon, 27 Apr 2020 02:43:21 GMT
< Server: BWS
< transfer-encoding: chunked
< X-Bce-Gateway-Region: NJ
< X-Bce-Request-Id: c994f260-c77a-417f-9aa9-26cefecdec36
{
  "totalCount": 32,
  "result": [{
      "certType": "DV",
      "productType": "SINGLE",
      "expireTime": "2018-07-17T23:59:59Z",
      "createTime": "2017-07-17T07:10:01Z",
      "duration": 0,
      "productId": "90988831-bd80-11e7-a065-2c44fd7f89bd",
      "brand": "SYMANTEC",
      "domainNumber": 0,
      "wildcardNumber": 0,
      "status": "EXPIRED",
      "domainName": "www.lss-qa.com"
  }, {
      "certType": "DV",
      "productType": "SINGLE",
      "expireTime": "2018-07-17T23:59:59Z",
      "createTime": "2017-07-17T07:10:18Z",
      "duration": 0,
      "productId": "90988ac4-bd80-11e7-a065-2c44fd7f89bd",
      "brand": "SYMANTEC",
      "domainNumber": 0,
      "wildcardNumber": 0,
      "status": "EXPIRED",
      "domainName": "abc.lss-qa.com"
  }, {
      "certType": "DV",
      "productType": "SINGLE",
      "expireTime": "2018-07-28T23:59:59Z",
      "createTime": "2017-07-28T04:10:33Z",
      "duration": 0,
      "productId": "909b6d34-bd80-11e7-a065-2c44fd7f89bd",
      "brand": "SYMANTEC",
      "domainNumber": 0,
      "wildcardNumber": 0,
      "status": "EXPIRED",
      "domainName": "mysandbox.lss-qa.com"
  }]
 }
```

免费证书购买限制检查

接口描述

本接口用于查询用户是否可以继续购买免费证书

请求结构

```
 > GET http://cas.baidubce.com/v3/openapi/query/check
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
check	String	是	Query参数	接口要求，必带，value为空，不要填

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
freeCount	int	一共有免费证书配额
enablePurchaseDV	boolean	当前是否可以继续免费购买

请求示例

```
  > GET http://cas.baidubce.com/v3/openapi/query/check
 > Authorization: bce-auth-v1/5c36beca06ab4616b7969fe0e768d660/2020-04-27T07:30:50Z/3600/host;x-bce-console-rpc-id;x-bce-date/c359d7207b0bef14c68e79532948056134a9f385fc5fed3a4454bd196e88ddd5
 > Host: cas.baidubce.com
 > x-bce-console-rpc-id: 883bacf4-3e9e-41e1-92f4-8da87ae4f99f
 > x-bce-date: 2020-04-27T07:30:50Z
```

响应示例

```
  < Cache-Control: no-cache
 < Content-Length: -1
 < Content-Type: application/json;charset=UTF-8
 < Date: Mon, 27 Apr 2020 07:30:52 GMT
 < Server: BWS
 < transfer-encoding: chunked
 < X-Bce-Gateway-Region: SZ
 < X-Bce-Request-Id: 420bdb80-e273-4e97-a804-b95a71502413
{"freeCount":5,"enablePurchaseDV":true}
```

价格相关接口

计算证书价格

接口描述

本接口用于查询用户选择的证书价格

请求结构

```
 > POST http://cas.baidubce.com/v3/openapi/price
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
{
 "certType" : "DV",
 "productType" : "SINGLE",
  "orderType" : "NEW",
  "brand" : "BAIDUTRUST",
  "domainNumber" : 1,
  "wildcardNumber" : 0,
  "purchaseLength" : 2
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
brand	String	是	BODY	证书品牌 取值见附录CertBrand
certType	String	是	BODY	证书类型 取值见附录CertType
productType	String	是	BODY	产品类型 取值见 ProductType
domainNumber	int	是	BODY	标准域名数量
wildcardNumber	int	是	BODY	通配符域名数量
duration	int	是	BODY	购买年限 （每个品牌每种证书不一定相同，具体参见控制台）
orderType	String	是	BODY	订单类型 取值见附录 OrderType

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
price	int	价格

请求示例

```
 > POST http://cas.baidubce.com/v3/openapi/price
 > Authorization: bce-auth-v1/5c36beca06ab4616b7969fe0e768d660/2020-04-27T07:41:42Z/3600/host;x-bce-console-rpc-id;x-bce-date/7ffe6b3a3e3f9ea2f1e3d73d6ac91680d5dbdbeccfcb6c74f1807926e2b484a8
 > Content-Type: application/json
 > Host: cas.baidubce.com
 > x-bce-console-rpc-id: 31ee5f3c-eb4f-4496-bd0c-d715d3286b06
 > x-bce-date: 2020-04-27T07:41:42Z
{
  "certType" : "DV",
  "productType" : "SINGLE",
  "orderType" : "NEW",
  "brand" : "BAIDUTRUST",
  "domainNumber" : 1,
  "wildcardNumber" : 0,
  "purchaseLength" : 2
}
```

响应示例

```
 < Cache-Control: no-cache
 < Content-Length: -1
 < Content-Type: application/json;charset=UTF-8
 < Date: Mon, 27 Apr 2020 07:41:44 GMT
 < Server: BWS
 < transfer-encoding: chunked
 < X-Bce-Gateway-Region: BJ
 < X-Bce-Request-Id: b1783bd8-1c16-41b2-8d83-45528aec1866
{"price":"1512.00"}
```

订单相关接口

SSL新购订单并自动支付

接口描述

本接口用于购买SSL证书，并且自动支付，自动支付可以使用 代金券，账号返点，账号余额

> 关于 API 支付逻辑的说明
>
> 默认会尽量使用代金券全额支付，不足部分使用默认逻辑处理。
>
> 默认逻辑即有账期默认账期支付
>
> 无账期则按照返点 > 现金的顺序支付；
>
> 代金券顺序：优先使用最快过期的、满足金额的代金券

请求结构

```
 > POST http://cas.baidubce.com/v3/openapi/order?new
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
{
 "certType" : "DV",
 "productType" : "SINGLE",
  "orderType" : "NEW",
  "brand" : "BAIDUTRUST",
  "domainNumber" : 1,
  "wildcardNumber" : 0,
  "purchaseLength" : 2
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
new	String	是	Query参数	必带标志参数，值必须为空
brand	String	是	BODY	证书品牌 取值见附录CertBrand
certType	String	是	BODY	证书类型 取值见附录CertType
productType	String	是	BODY	产品类型 取值见 ProductType
domainNumber	int	是	BODY	标准域名数量
wildcardNumber	int	是	BODY	通配符域名数量
purchaseLength	int	是	BODY	购买年限 （每个品牌每种证书不一定相同，具体参见控制台）
timeUnit	int	否	BODY	时间单位默认为YEAR (可选MONTH YEAR，免费证书purchaseLength必须为3，timeUnit必须为MONTH)
orderType	String	是	BODY	订单类型 取值见附录 OrderType

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
bceOrderId	String	订单ID
certIds	List< String>	证书ID

请求示例

```
 > POST http://cas.baidubce.com/v3/openapi/order?new
 > Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-27T09:29:38Z/3600/host;x-bce-console-rpc-id;x-bce-date/fe48acdd7ac62c215fe567fa0536f29c7096d5103ccd3eaa3ad5e1fbd163b7b4
 > Content-Type: application/json
 > Host: 10.133.101.20
 > x-bce-console-rpc-id: 63c91ede-f2c6-4aa3-b6e7-23a599a6a5f7
 > x-bce-date: 2020-04-27T09:29:38Z
{
  "certType" : "DV",
  "productType" : "SINGLE",
  "brand" : "GEOTRUST",
  "domainNumber" : 1,
  "wildcardNumber" : 0,
  "purchaseLength" : 1,
  "timeUnit" : "YEAR"
}
```

响应示例

```
 < Content-Type: application/json;charset=UTF-8
 < Date: Mon, 27 Apr 2020 09:30:18 GMT
 < Server: BWS
 < Transfer-Encoding: chunked
 < x-bce-request-id: b1d08f72-cc3f-4d8e-b368-e35d3d2a94ce
{
    "bceOrderId": "989bdbc4e7da42e892dd07d4b2834a08",
    "certIds": ["b7a3e152-7ebd-4c39-8a1c-60b79c2f3367"]
}
```

证书相关接口

申请证书

接口描述

本接口用于对已购买的证书进行申请

请求结构

```
 > POST http://cas.baidubce.com/v3/openapi/certs/{certId}
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
{
  "company" : "default company",
  "address" : "default address",
  "postalCode" : "default code",
  "region" : {
    "province" : "000000",
    "city" : "000000",
    "country" : "086"
  },
  "domain" : "ba.com",
  "verifyMode" : "DNS",
  "multiDomain" : [ ],
  "department" : "default division",
  "companyPhone" : "17647371024",
  "orderGivenName" : "default first name",
  "orderFamilyName" : "default last name",
  "orderPosition" : "default title",
  "orderEmail" : "ssl_no_reply@baidu.com",
  "orderPhone" : "17647371024",
  "techGivenName" : "default first name",
  "techFamilyName" : "default last name",
  "techPosition" : "default title",
  "techEmail" : "ssl_no_reply@baidu.com",
  "techPhone" : "17647371024"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
certId	String	是	Url参数	操作的证书ID
domain	String	是	BODY	主域名
password	String	否（DV证书可不填写）	BODY	订单密码（设置阶段，务必保存）
algorithm	String	否（DV证书不需要填写）	BODY	加密算法 取值见附录 EncryptionType
strength	String	否 （DV证书不需要填写）	BODY	加密强度 取值见附录 EncryptionStrength
csr	String	否	BODY	CSR方式申请时必填
verifyMode	String	是	BODY	域名验证方式 取值见附录 AuthType
multiDomain	List< String>	否	BODY	提交多域名列表
company	String	否（DV证书可不填写）	BODY	公司名称
department	String	否 （DV证书可不填写）	BODY	部门名
address	String	否（DV证书可不填写）	BODY	地址
postalCode	String	否（DV证书可不填写）	BODY	地区编码
companyPhone	String	否（DV证书可不填写）	BODY	公司联系电话
region	Region	否 （DV证书可不填写）	BODY	公司地址，取值见附录 Region
orderGivenName	String	否（DV证书可不填写）	BODY	管理人姓
orderFamilyName	String	否（DV证书可不填写）	BODY	管理人名
orderPosition	String	否（DV证书可不填写）	BODY	管理人职位
orderEmail	String	否（DV证书可不填写）	BODY	管理人邮箱
orderPhone	String	否（DV证书可不填写）	BODY	管理人电话
techGivenName	String	否（DV证书可不填写）	BODY	技术联系人姓
techFamilyName	String	否（DV证书可不填写）	BODY	技术联系人名
techPosition	String	否（DV证书可不填写）	BODY	技术联系人职位
techEmail	String	否（DV证书可不填写）	BODY	技术联系人邮箱
techPhone	String	否（DV证书可不填写）	BODY	技术联系人电话

响应头域

除公共头域外，无其它特殊头域。

响应参数

无

请求示例

```
 > POST http://cas.baidubce.com/v3/openapi/certs/4454ebda-7ff1-4d66-9599-6a81ca510b51
 > Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-27T11:48:29Z/3600/host;x-bce-console-rpc-id;x-bce-date/0f8b8306c6540bd2dde8c7608c053662b74eeadb95cfd8191444292c2348b1f2
 > Content-Type: application/json
 > Host: 10.133.101.20
 > x-bce-console-rpc-id: be1bdfb5-fa30-4684-acbf-11b7bc9a747d
 > x-bce-date: 2020-04-27T11:48:29Z
{
  "company" : "default company",
  "address" : "default address",
  "postalCode" : "default code",
  "region" : {
    "province" : "000000",
    "city" : "000000",
    "country" : "086"
  },
  "domain" : "ba.com",
  "verifyMode" : "DNS",
  "multiDomain" : [ ],
  "department" : "default division",
  "companyPhone" : "17647371024",
  "orderGivenName" : "default first name",
  "orderFamilyName" : "default last name",
  "orderPosition" : "default title",
  "orderEmail" : "ssl_no_reply@baidu.com",
  "orderPhone" : "17647371024",
  "techGivenName" : "default first name",
  "techFamilyName" : "default last name",
  "techPosition" : "default title",
  "techEmail" : "ssl_no_reply@baidu.com",
  "techPhone" : "17647371024"
}
```

响应示例

```
 < Cache-Control: no-cache
< Content-Length: 0
< Date: Mon, 27 Apr 2020 11:48:34 GMT
< Server: BWS
< x-bce-request-id: de3f0c3e-86f6-43e7-b1bd-35a837f4b84f
```

下载确认函模板

接口描述

本接口用于对需要提交确认函的证书，提供下载确认函模板的功能

请求结构

```
 > GET http://cas.baidubce.com/v3/openapi/certs/{certId}/letter
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
certId	String	是	Url参数	操作的证书ID

响应头域

除公共头域外，无其它特殊头域。

响应参数

无，直接将response的返回流保存到文件即可

请求示例

```
 > GET http://cas.baidubce.com/v3/openapi/certs/4454ebda-7ff1-4d66-9599-6a81ca510b51/letter
> Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-27T12:53:02Z/3600/host;x-bce-console-rpc-id;x-bce-date/c72c2e21d2ae7e470b534590b7d48034424a3b3308e09fcb385cbc581e097cc5
> Host: 10.133.101.20
> x-bce-console-rpc-id: 950a47e0-1b63-4554-b710-53230565a365
> x-bce-date: 2020-04-27T12:53:02Z
```

响应示例

```
 < Cache-Control: no-cache
< Content-Length: 0
< Date: Mon, 27 Apr 2020 11:48:34 GMT
< Server: BWS
< x-bce-request-id: de3f0c3e-86f6-43e7-b1bd-35a837f4b84f
二进制流
```

上传确认函

接口描述

本接口用于对需要提交确认函的证书，提供填写确认函后上传的功能

请求结构

```
 > PUT http://cas.baidubce.com/v3/openapi/certs/{certId}/letter\
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
二进制流
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
certId	String	是	Url参数	操作的证书ID

响应头域

除公共头域外，无其它特殊头域。

响应参数

无

请求示例

```
 > PUT http://cas.baidubce.com/v3/openapi/certs/d579e580-4aa3-44a2-afb6-68891e050ec2/letter
> Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-27T13:00:44Z/3600/host;x-bce-console-rpc-id;x-bce-date/43e7bd1ec6fafd73140ac54b529b7d153d995b24a691c8db1ea262068b092ad0
> Content-Type: application/octet-stream
> Host: 10.133.101.20
> x-bce-console-rpc-id: 507139d7-4209-4e0d-a526-620439434424
> x-bce-date: 2020-04-27T13:00:44Z
二进制文件
```

响应示例

```
 < Cache-Control: no-cache
< Content-Length: 0
< Date: Mon, 27 Apr 2020 11:48:34 GMT
< Server: BWS
< x-bce-request-id: de3f0c3e-86f6-43e7-b1bd-35a837f4b84f
```

取消申请

接口描述

本接口用于取消申请流程中的证书

请求结构

```
 > DELETE http://cas.baidubce.com/v3/openapi/certs/{certId}?cancel
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
certId	String	是	Url参数	操作的证书ID
cancel	String	是	Query参数	接口标志，必传，值为空

响应头域

除公共头域外，无其它特殊头域。

响应参数

无

请求示例

```
 > DELETE http://cas.baidubce.com/v3/openapi/certs/32f10579-5c5c-4c29-912b-45ab26fb0a9b?cancel
> Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-27T13:07:04Z/3600/host;x-bce-console-rpc-id;x-bce-date/2a10207e379a6d46d5c5d33bc97edc2332cb90f39f46a0d3a022f146f5f7bcb9
> Host: 10.133.101.20
> x-bce-console-rpc-id: 10d86ecb-5aff-4e84-87c8-500a9cccedc4
> x-bce-date: 2020-04-27T13:07:04Z
```

响应示例

```
 < Cache-Control: no-cache
< Content-Length: 0
< Date: Mon, 27 Apr 2020 13:07:07 GMT
< Server: BWS
< x-bce-request-id: 4e6bba7f-13c4-4e09-b9ba-030524713358
```

删除申请

接口描述

本接口用于删除失败或者到期的证书，释放免费DV证书配额。

请求结构

```
 > DELETE http://cas.baidubce.com/v3/openapi/certs/{certId}?delete
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
certId	String	是	Url参数	操作的证书ID
delete	String	是	Query参数	接口标志，必传，值为空

响应头域

除公共头域外，无其它特殊头域。

响应参数

无

请求示例

```
> DELETE http://cas.baidubce.com/v3/openapi/certs/32f10579-5c5c-4c29-912b-45ab26fb0a9b?delete
> Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-27T13:07:04Z/3600/host;x-bce-console-rpc-id;x-bce-date/2a10207e379a6d46d5c5d33bc97edc2332cb90f39f46a0d3a022f146f5f7bcb9
> Host: 10.133.101.20
> x-bce-console-rpc-id: 10d86ecb-5aff-4e84-87c8-500a9cccedc4
> x-bce-date: 2020-04-27T13:07:04Z
```

响应示例

```
< Cache-Control: no-cache
< Content-Length: 0
< Date: Mon, 27 Apr 2020 13:07:07 GMT
< Server: BWS
< x-bce-request-id: 4e6bba7f-13c4-4e09-b9ba-030524713358
```

下载证书

接口描述

本接口用于下载已经颁发成功的证书。**不支持下载重新颁发的证书**

请求结构

```
> POST http://cas.baidubce.com/v3/openapi/certs/{certId}?download
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
certId	String	是	Url参数	操作的证书ID
download	String	是	Query参数	接口标志，必传，值为空
format	String	是	BODY	文件类型 取值见附录 CertFileType
orderPassword	String	否 （DV证书不需要）	BODY	订单密码
filePassword	String	是	BODY	即将下载的压缩文件密码

响应头域

除公共头域外，无其它特殊头域。

响应参数

无

请求示例

```
> DELETE http://cas.baidubce.com/v3/openapi/certs/32f10579-5c5c-4c29-912b-45ab26fb0a9b?download
> Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-27T13:07:04Z/3600/host;x-bce-console-rpc-id;x-bce-date/2a10207e379a6d46d5c5d33bc97edc2332cb90f39f46a0d3a022f146f5f7bcb9
> Host: 10.133.101.20
> x-bce-console-rpc-id: 10d86ecb-5aff-4e84-87c8-500a9cccedc4
> x-bce-date: 2020-04-27T13:07:04Z
```

响应示例

```
< Cache-Control: no-cache
< Content-Length: 0
< Date: Mon, 27 Apr 2020 13:07:07 GMT
< Server: BWS
< x-bce-request-id: 4e6bba7f-13c4-4e09-b9ba-030524713358
二进制文件
```

证书详情

接口描述

本接口用于查看证书详情。**不支持下载重新颁发的证书**

请求结构

```
> POST http://cas.baidubce.com/v3/openapi/certs/{certId}/detail
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
certId	String	是	Url参数	操作的证书ID

响应头域

除公共头域外，无其它特殊头域。

响应参数

返回 CertDetails 对象

请求示例

```
 > POST http://cas.baidubce.com/v3/openapi/certs/d579e580-4aa3-44a2-afb6-68891e050ec2/detail
 > Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-28T02:31:50Z/3600/host;x-bce-console-rpc-id;x-bce-date/7c3adfcd4d1463950711186a5c593a18acc9f39d513ff2228b82a2d6de3da0cf
 > Content-Type: application/json
 > Host: 10.133.101.20
 > x-bce-console-rpc-id: d8e0f464-3bf3-41d8-85c6-0268f822d528
 > x-bce-date: 2020-04-28T02:31:50Z
{
  "productId" : "d579e580-4aa3-44a2-afb6-68891e050ec2"
}
```

响应示例

```
 < Cache-Control: no-cache
 < Content-Length: -1
 < Content-Type: application/json;charset=UTF-8
 < Date: Tue, 28 Apr 2020 02:31:53 GMT
 < Server: BWS
 < Transfer-Encoding: chunked
 < x-bce-request-id: 707dada2-9a69-4c84-abb9-988fc9fc214e
{
    "productName": "GeoTrust 域名型SSL证书",
    "certType": "DV",
    "productType": "SINGLE",
    "applyTime": "2020-04-14T03:05:40Z",
    "downloadSupported": false,
    "bindDomains": [],
    "duration": 1,
    "fromBaidu": false,
    "productId": "d579e580-4aa3-44a2-afb6-68891e050ec2",
    "brand": "GEOTRUST",
    "domainNumber": 1,
    "wildcardNumber": 0,
    "processStatus": "DELETED",
    "domainName": "ba.com"
}
```

PKI信息

接口描述

本接口用于查看证书pki信息详情。**不支持下载重新颁发的证书**

请求结构

```
 > GET http://cas.baidubce.com/v3/openapi/certs/{certId}/pki
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
certId	String	是	Url参数	操作的证书ID
productId	String	是	BODY	证书ID

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
domainName	String	绑定域名
company	String	公司
department	String	部门
address	String	地址
region	String	地域，取值见附录Region
algorithm	String	算法
strength	String	强度
csrPem	String	csr pem
certPem	String	证书pem
certCaPem	String	证书链pem
startTime	String	证书生效时间
expireTime	String	证书到期时间

请求示例

```
 > GET http://cas.baidubce.com/v3/openapi/certs/d579e580-4aa3-44a2-afb6-68891e050ec2/pki
> Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-28T02:38:48Z/3600/host;x-bce-console-rpc-id;x-bce-date/a551c0c274ba9619ddab0c08c23a802fd0089dc0bd75f80f5c3aeb775cfb823a
> Host: 10.133.101.20
> x-bce-console-rpc-id: bb72144f-0b31-45ae-a3b6-0d40f5f1e2f6
> x-bce-date: 2020-04-28T02:38:48Z
```

响应示例

```
  < Cache-Control: no-cache
 < Content-Length: -1
 < Content-Type: application/json;charset=UTF-8
 < Date: Tue, 28 Apr 2020 02:38:50 GMT
 < Server: BWS
 < Transfer-Encoding: chunked
 < x-bce-request-id: 8deba927-eb82-44f3-bf71-a633fde4f672
{
    "company": "*******",
    "address": "*******",
    "region": {
        "province": "000000",
        "city": "000000",
        "country": "086"
    },
    "algorithm": "SHA256withECDSA",
    "strength": "2048",
    "csrPem": "-----BEGIN CERTIFICATE REQUEST-----
\nMIICoTCCAYkCAQAwXDEPMA0GA1UEAwwGYmEuY29tMRgwFgYDVQQKDA9kZWZhdWx0\nIGNvbXBhbnkxEDAOBgNVBAcMB0
-----END CERTIFICATE REQUEST-----",
    "certPem": "-----BEGIN CERTIFICATE-----
\nMIIDCTCCAq+gAwIBAgIIdhgvbRaqPjEwCgYIKoZIzj0EAwIwUjEiMCAGA1UEAxMZ\nTVBLSSBTeW1hbnRlYyBTU0wgVEVTVCBDQT
-----END CERTIFICATE-----",
    "certCaPem": "\n-----BEGIN CERTIFICATE-----
\nMIICwzCCAaugAwIBAgIIBFIpXRbKwCUwDQYJKoZIhvcNAQELBQAwVzEnMCUGA1UE\nAxMeTVBLSSBTeW1hbnRlYyBTU0wgVE
-----END CERTIFICATE-----",
    "expireTime": "2020-05-13T08:00:00Z",
    "domainName": "ba.com",
    "department": "default division",
    "startTime": "2020-04-14T08:00:00Z"
}
```

公司及联系人信息

接口描述

本接口用于查看证书公司及联系人信息。

请求结构

```
 > GET http://cas.baidubce.com/v3/openapi/certs/{certId}/contact
> Authorization: authorization string
> Host: cas.baidubce.com
> x-bce-console-rpc-id: e69c8fff-166b-4c82-87c1-7aa3ba309d5c
> x-bce-date: 2020-04-27T02:43:18Z
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
certId	String	是	Url参数	操作的证书ID

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
company	String	公司
department	String	部门
address	String	地址
companyPhone	String	公司电话
orderName	String	联系人名
orderPosition	String	联系人职位
orderEmail	String	联系人邮件
orderPhone	String	联系人电话
techName	String	技术联系人名
techPosition	String	技术联系人职位
techEmail	String	技术联系人邮件
techPhone	String	技术联系人电话
region	Region	公司地址，取值见附录 Region

请求示例

```
 > GET http://cas.baidubce.com/v3/openapi/certs/d579e580-4aa3-44a2-afb6-68891e050ec2/contact
> Authorization: bce-auth-v1/f624059c33b948319ad3b8a4a6c5b670/2020-04-28T02:38:48Z/3600/host;x-bce-console-rpc-id;x-bce-date/a551c0c274ba9619ddab0c08c23a802fd0089dc0bd75f80f5c3aeb775cfb823a
> Host: 10.133.101.20
> x-bce-console-rpc-id: bb72144f-0b31-45ae-a3b6-0d40f5f1e2f6
> x-bce-date: 2020-04-28T02:38:48Z
```

响应示例

```
  < Cache-Control: no-cache
 < Content-Length: -1
 < Content-Type: application/json;charset=UTF-8
 < Date: Tue, 28 Apr 2020 03:15:50 GMT
 < Server: BWS
 < Transfer-Encoding: chunked
 < x-bce-request-id: 90a89673-1444-4213-84fe-8a553493dd1a
{
    "company": "*******",
    "address": "*******",
    "postalCode": "default code",
    "region": {
        "province": "000000",
        "city": "000000",
        "country": "086"
    },
    "department": "default division",
    "companyPhone": "*******",
    "orderName": "default last namedefault first name",
    "orderPosition": "default title",
    "orderEmail": "*******@email",
    "orderPhone": "*******",
    "techName": "default last namedefault first name",
    "techPosition": "default title",
    "techEmail": "*******@email",
    "techPhone": "*******"
}
```

证书过户

接口描述

本接口用于对已购买的证书进行转移账号

请求结构

```
 > POST http://cas.baidubce.com/v3/openapi/trans/batch/{newUserId}
> Authorization: bce-auth-v1/c4a25a8a12304b5ca7a7b35d12acd058/2020-09-15T07:54:23Z/3600/host;x-bce-console-rpc-id;x-bce-date/5113458088e76548ad9a2b85ea133cf452cc2dd429fb51e4ebcbbe22e64969e6
> Content-Type: application/json
> Host: gzns-store-sandbox009.gzns.baidu.com
> x-bce-console-rpc-id: c4ce6e7c-4c00-45d5-886c-f501a41c2598
> x-bce-date: 2020-09-15T07:54:23Z
{
  "params" : [ "a9277059-6193-4a41-87c2-cfe4c10f68bd" ]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
params	List< String>	是	BODY	操作的证书ID列表
newUserId	String	是	URL参数	过户目标账号ID

响应头域

除公共头域外，无其它特殊头域。

响应参数

无

请求示例

```
 > POST http://cas.baidubce.com/v3/openapi/trans/batch/f168739fc9c5473ab798a39c3db446b6
> Authorization: bce-auth-v1/c4a25a8a12304b5ca7a7b35d12acd058/2020-09-15T07:54:23Z/3600/host;x-bce-console-rpc-id;x-bce-date/5113458088e76548ad9a2b85ea133cf452cc2dd429fb51e4ebcbbe22e64969e6
> Content-Type: application/json
> Host: gzns-store-sandbox009.gzns.baidu.com
> x-bce-console-rpc-id: c4ce6e7c-4c00-45d5-886c-f501a41c2598
> x-bce-date: 2020-09-15T07:54:23Z
{
  "params" : [ "a9277059-6193-4a41-87c2-cfe4c10f68bd" ]
}
```

响应示例

```
 < 200
< Cache-Control: no-cache
< Content-Length: 0
< Date: Tue, 15 Sep 2020 07:54:25 GMT
< Server: BWS
< x-bce-request-id: ccf10a98-c763-452b-8b89-d6ac8d11d4e7
```

云SSL相关接口

站点列表

接口描述

站点列表，查看用户所拥有的云端证书部署的站点

请求结构

```
 POST /host/list_host
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	HostlistRequest	null	Body

请求体参数

HostlistRequest

参数名称	参数类型	是否必须	描述	示例值
host	**String**	是	站点	"a.com"

响应参数

无

请求示例

POST /host/list_host

```
 <公共请求头>
{
 "host" : "host"
}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
[
   {
      "host":"aaa.com",
      "status":"RUNNING",
      "resource_id":"xxx"
   },
   {
      "host":"bbb.com",
      "status":"SUSPEND",
      "resource_id":"yyy"
   }
]
```

站点详情

接口描述

站点详情

请求结构

```
 POST /host/detail
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	DetailRequest	null	Body

请求体参数

DetailRequest

参数名称	参数类型	是否必须	描述	示例值
host	**String**	是	站点	"a.com"

响应参数

无

请求示例

POST /host/detail

```
 <公共请求头>
{"host":"a.com"}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

站点停止

接口描述

站点停止

请求结构

```
 POST /host/shutdown
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	查看请求体参数	{"host":"a.com"}	Body

请求体参数

参数名称	参数类型	是否必须	描述	示例值
host	**String**	是	站点	"a.com"

响应参数

无

请求示例

POST /host/shutdown

```
 <公共请求头>
{"host":"a.com"}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

更新IPv6

接口描述

更新ipv6

请求结构

```
 POST /host/update_ipv6
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	Updateipv6Request	null	Body

请求体参数

Updateipv6Request

参数名称	参数类型	是否必须	描述	示例值
host	String	是	站点	"a.com"
ipv6	String	是	是否优化，可选值"true" "false"	"true"

响应参数

无

请求示例

POST /host/update_ipv6

```
 <公共请求头>
{"host":"a.com","ipv6":"true"}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

更新TLS协议版本控制

接口描述

更新tls协议版本控制

请求结构

```
 POST /host/batch_update_tls
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	BatchUpdateTLSRequest	null	Body

请求体参数

BatchUpdateTLSRequest

参数名称	参数类型	是否必须	描述	示例值
host	String	是	站点	www.baidu.com
ssl_protocols	List<String>	是	协议，包含 (TLS10,TLS11,TLS12,TLS13)	["TLS10","TLS11"]

响应参数

无

请求示例

POST /host/batch_update_tls

```
 <公共请求头>

{"host":"a.com","ssl_protocols":["TLS10","TLS11"]}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>

{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

更新OCSP

接口描述

更新ocsp

请求结构

```
 POST /host/update_ocsp
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	UpdateOCSPRequest	null	Body

请求体参数

UpdateOCSPRequest

参数名称	参数类型	是否必须	描述	示例值
host	**String**	是	站点	"ac.com"
ocsp_stapling	**String**	是	是否启用，可选值 "true" "false"	"true" 或"false"

响应参数

无

请求示例

POST /host/update_ocsp

```
 <公共请求头>
{"host":"a.com","ocsp_stapling":"true"}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "Enable1":false,
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

更新QUIC

接口描述

更新quic

请求结构

```
 POST /host/update_quic
<公共请求头>
```

请求参数

UpdatequicResponse

参数名称	参数类型	是否必须	描述	示例值
quic	String	是	是否启用，可选值 "true"或"false"	"true"
host	String	是	站点	"a.com"

响应参数

无

响应状态码

200

请求示例

POST /host/update_quic

```
 <公共请求头>
{"host":"a.com","quic":"true"}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "Enable1":false,
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

更新源站地址

接口描述

更新源站地址

请求结构

```
 POST /host/update_origin
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	UpdateOriginRequest	null	Body

请求体参数

UpdateOriginRequest

参数名称	参数类型	是否必须	描述	示例值
host	String	是	站点	"abc.com"
origins	Origin	是	源站数组	[{"addr":"2.2.2.2","http_port":"80","https_port":"443","type":""}]

响应参数

无

请求示例

POST /host/update_origin

```
 <公共请求头>
{
    "host":"abc.com",
    "origins":[
        {
            "addr":"2.2.2.2",
            "http_port":"80",
            "https_port":"443",
            "type":""
        }
    ]
}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "Enable1":false,
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

更新压缩

接口描述

更新压缩

请求结构

```
 POST /host/update_compress
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	UpdateCompressRequest	null	Body

请求体参数

UpdateCompressRequest

参数名称	参数类型	是否必须	描述	示例值
host	String	是	站点	"a.com"
compress	String	是	是否压缩，可选值"true" "false"	"true"

响应参数

无

请求示例

POST /host/update_compress

```
 <公共请求头>
{
 "host": "a.com",
 "compress": "true"
}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "Enable1":false,
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

更新SEO

接口描述

更新seo

请求结构

```
 POST /host/update_seo
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	UpdateSEORequest	null	Body

请求体参数

UpdateSEORequest

参数名称	参数类型	是否必须	描述	示例值
host	**String**	是	站点	"a.com"
seo_push_record	**String**	是	是否主动推数据，可选值"true" "false"	"true"
seo_directly_origin	**String**	是	是否直接回源"true" "false"	"true"

请求示例

POST /host/update_seo

```
 <公共请求头>
{
 "host": "a.com",
 "seo_push_record": "true",
 "seo_directly_origin": "false"
}
```

响应参数

无

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

页面多余字符裁剪

接口说明

页面多余字符裁剪

请求结构

```
 POST /host/update_file_trim
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	UpdateFileTrimRequest	null	Body

请求体参数

UpdateFileTrimRequest

参数名称	参数类型	是否必须	描述	示例值
host	String	是	站点	"a.com"
trim	String	是	是否优化，可选值"true" "false"	"false"

响应参数

无

请求示例

POST /host/update_file_trim

```
 <公共请求头>
{"host":"a.com","trim":"true"}
```

响应示例

```
 HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "Enable1":false,
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

HTTPS强制跳转

接口描述

https强制跳转

请求结构

```
POST /host/update_mandatory_https
<公共请求头>
```

请求参数

参数名称	参数类型	是否必须	描述	示例值	参数位置
body	Object	否	UpdateMandatoryHTTPSRequest	null	Body

请求体参数

UpdateMandatoryHTTPSRequest

参数名称	参数类型	是否必须	描述	示例值
host	**String**	是	站点	"a.com"
mandatory_https	**String**	是	是否启动，可选"true","false"	"false"

响应参数

无

请求示例

POST /host/update_mandatory_https

```
<公共请求头>
{"host":"a.com","mandatory_https":"true"}
```

响应示例

```
HTTP/1.1 200 OK
<公共响应头>
{
    "id":0,
    "resource_id":"",
    "host":"",
    "user_id":"",
    "cname_record":"",
    "iam_cert_id":"",
    "enable":"",
    "status":"",
    "route_type":"",
    "origins":null,
    "updated_at":"0001-01-01T00:00:00Z",
    "created_at":"0001-01-01T00:00:00Z",
    "ssl_versions":null,
    "ocsp_stapling":"",
    "mandatory_https":"",
    "file_trim":"",
    "hsts":"",
    "quic":"",
    "cache":"",
    "seo_push_record":"",
    "seo_directly_origin":"",
    "traffic_limit_rate":0,
    "compress":"",
    "quota_id":0
    }
```

附录

Model对象定义

CertDetails

参数名称	类型	描述
productId	String	证书ID
productName	String	产品名称
brand	String	证书品牌
productType	String	产品类型
domainNumber	int	标准域名数量
wildcardNumber	int	通配符域名数量
status	String	证书状态
processStatus	String	申请状态
applyTime	Timestamp	申请时间
submitTime	Timestamp	上传时间
expireTime	Timestamp	到期时间
createTime	Timestamp	创建时间
errorMessage	String	失败原因
verifyMode	String	验证方式
domainName	String	主域名
value	String	dns 或 file验证内容
fileUrl	Strubg	file验证时候要求的地址
dnsName	String	dns验证时要求的name
leftTime	String	验证剩余时间
downloadSupported	boolean	证书是否可下载
bindDomains	List	绑定域名列表
duration	int	证书年限
fromBaidu	boolean	是否为百度证书

CertStatus

取值	对应状态
APPLY_PENDING	待申请
APPLYING	申请中
NORMAL	申请成功
FAILED	申请失败
RENEW_PENDING	待续费
REPLACING	重新颁发
REPLACING_FAILED	重新颁发失败
REPLACING_NORMAL	重新颁发成功
REPLACING_CONFIRM	重新颁发待确认
REPLACE_CA	重颁发信息确认后，需要提交信息至CA
EXPIRED	已过期
DELETED	已删除

CertType

取值	对应类型
DV	DV(域名型)
OV	OV(企业型)
EV	EV(增强型)
OVPRO	企业型专业版（OV PRO）证书
EVPRO	增强型专业版（EV PRO)证书

CertBrand

取值
SECURESITE
GEOTRUST
GLOBALSIGN
CFCA
TRUSTASIA
BAIDUTRUST

ProductType

取值	对应类型
SINGLE	单域名
MULTI	多域名版
WILDCARD	通配符域名版
SINGLE_PRO	单域名专业版
MULTI_PRO	多域名专业版
WILDCARD_PRO	通配符域名专业版

OrderType

取值	对应类型
NEW	购买
RENEW	续费

NewOrderConfig

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号
brand	String	是	BODY	证书品牌 取值见附录CertBrand
certType	String	是	BODY	证书类型 取值见附录CertType
productType	String	是	BODY	产品类型 取值见 ProductType
domainNumber	int	是	BODY	标准域名数量
wildcardNumber	int	是	BODY	通配符域名数量
duration	int	是	BODY	购买年限 （每个品牌每种证书不一定相同，具体参见控制台）
orderType	String	是	BODY	订单类型 取值见附录 OrderType

EncryptionType

取值
RSA
ECDSA

EncryptionStrength

取值	对应加密类型
ECDSA_PRIME256V1	ECDSA
RSA_2048	RSA
RSA_4096	RSA

AuthType

验证方式
DNS
FILE
HTTP
CNAME

Region

注意以下取值需要使用地区编码

字段	释义
province	省
city	市
country	国家

CertFileType

取值
PEM
PEM_APACHE
PEM_NGINX
PEM_HAPROXY
JKS_TOMCAT
JKS
PKCS12

常见问题

常见问题总览

一般问题

- HTTP和HTTPS有什么区别？
- 什么是SSL证书？
- 什么是CSR？
- 什么是DV证书？
- SSL证书有什么优势？
- 免费与付费DV证书区别
- DV型、OV型、EV型SSL证书对比

SSL证书申请问题

- DV SSL证书申请需要多久？
- 申请了主域名SSL证书，是否还需要申请www域名的？
- 为什么会出现安全审核失败？
- 常见的证书申请状态有哪些？
- 如何获取WHOIS信息？
- CNAME冲突如何解决？
- 各品牌OV/EV证书申请材料都有哪些？
- 证书为什么无法提交域名？

SSL证书部署问题

- 如何导入证书？
- 是否可以下载SSL证书到本地？
- 百度申请的SSL证书能否部署在别的云服务器上？

一般问题

HTTP和HTTPS有什么区别？

HTTP（Hypertext Transfer Protocol）超文本传输协议是用来在Internet上传送超文本的传送协议，它可以使浏览器更加高效，使网络传输减少。但HTTP协议采用明文传输信息，存在信息窃听、信息篡改和信息劫持的风险。

HTTPS(Secure Hypertext Transfer Protocol) 安全超文本传输协议是一个安全的通信通道，它基于HTTP开发，用于在客户计算机和服务器之间交换信息。HTTPS使用安全套接字层(SSL)进行信息交换，简单来说HTTPS是HTTP的安全版，是使用TLS/SSL加密的HTTP协议。

什么是SSL证书？

SSL证书是数字证书的一种，遵守 SSL协议，由受信任的数字证书颁发机构CA（如：Symantec）在验证服务器身份后颁发，具有服务器身份验证和数据传输加密功能。

服务器部署了SSL证书后可以确保从用户电脑到服务器之间的传输链路上是高强度加密传输的，用户在浏览器上输入的机密信息和从服务器上查询的信息是不可能被非法篡改和窃取的。同时向网站访问者证明了服务器的真实身份，此真实身份是通过第三方权威机构验证的。

什么是CSR？

CSR 即证书签名请求文件（Certificate Signing Request），是证书申请者在申请数字证书时由CSP(加密服务提供者)在生成私钥的同时也生成证书请求文件，证书申请者只要把CSR文件提交给证书颁发机构后，证书颁发机构使用其根证书私钥签名就生成了证书公钥文件，也就是颁发给用户的证书。

什么是DV证书？

DV证书是域名级别验证的证书，通常无需人工审核，快速颁发，立即生效。适合小型网站。百度智能云提供免费的DV型证书，每个用户ID可最多申请20张证书，申请成功但不使用的证书也占用配额。

SSL证书有什么优势？

1. 简单快捷,只需要申请一张证书,部署在服务器上,就可以在有效期内不用再做其他的操作。
2. 显示直观,部署SSL证书后,通过https访问网站,能在地址栏或地址栏右侧直接看到加密锁标志能直观的表明网站是加密的!使用EV证书,甚至能直接在地址栏看到公司名称。
3. 身份认证,这是别的加密方式都不具备的,能在证书信息里面看到网站所有者公司信息,进而确认网站的有效性和真实性,不会被钓鱼网站所欺骗。

免费与付费DV证书区别

对比项目	免费证书	商业收费证书
支持购买数量	每个用户支持20张免费证书申请	不限
支持域名数量	同一主域名累计仅支持20张	合规域名均可支持
支持域名格式	仅支持www.abc.com/abc.com 单域名格式	全面支持各种域名格式， 支持www.abc.com/abc.com 单域名格式；*.abc.com泛域名；单域名和泛域名混合在一张证书里的多域名格式
算法支持	仅支持RSA加密算法	支持更安全的ECC + RSA两种加密算法
验证方式	仅支持DNS／文件两种验证方式	支持DNS/文件，邮件，电话，人工律师函多种验证方式
签发成功率	存在一定的签发失败率，不保障100%成功签发	合规域名保障100%签发，保障用户使用
伴随服务	不提供人工服务	提供全生命周期技术服务支持
适用客户	个人用户	商业用户

DV型、OV型、EV型SSL证书对比

产品类型	DV型SSL证书	OV型SSL证书	EV型SSL证书
适合场景	适合场景适合小微企业/API服务/个人网站	适合企业应用（OA、CRM、ERP、HRM等）、企业官网、电商	适用于金融/安全行业企业、大中型企业或对可信品牌更重视的企业、政府机关
场景特点	使用SSL证书进行信息传输的高强度加密，可有效杜绝信息劫持	提升系统安全，确保敏感信息不被劫持，增强企业诚信力和用户信赖感	最大程度保障信息安全和网站公信力，大网站标配
验证内容	验证域名归属	验证企业身份	最高验证级别
部署形态	普通加锁标记	普通加锁标记	绿色安全地址栏
主流浏览器兼容	100%	100%	100%
签发时间	10分钟内	1-2个工作日	1-3个工作日

SSL证书申请问题

DV SSL证书申请需要多久？

DV SSL证书无需验证所有者资质资料，审核流程相对简单，因此可快速签发。但部分域名信息可能会触发不同等级的安全审查机制，必要时需要人工介入进行审查签发，因此，SSL证书签发时间可能会从10几分钟到3个工作日不等。如果您的证书仍在审核中，请耐心等待。

证书为什么无法提交域名？

CA行业规定，带下划线的域名不能申请证书，请检查域名是否符合规范。

证书申请提交很久了，为什么还是审核中？

申请证书后，您需要查看您证书绑定的域名是否验证成功。域名验证成功后，CA中心才会对证书进行签发。

您需要到您的域名解析服务商提供的系统中进行配置，参考域名验证。完成域名验证后，您需要校验域名验证结果，参考查看域名验证结果。

> 如果您的域名中包含某些敏感词（例如bank、pay、live等），可能会触发人工审核机制，审核时间会比较长，请您耐心等待。

申请了主域名SSL证书，是否还需要申请www域名的？

- DV型免费证书 申请主域名如baidu.com的SSL证书，默认会在证书域名中添加www域名如www.baidu.com，因此无需重复申请；同样，如果您申请www的二级域名，将同时为您添加主域名使用权限。但如果您是申请的三级域名，如www.bce.baidu.com，则只能支持该域名，不会支持bce.baidu.com。
- 付费型证书

	GlobalSign	TrustAsia	SecureSite	GeoTrust	CFCA
通用名称 www. 开头送上级域	送，不论www.在二级还是三级都会送上级域	送，不论www.在二级还是三级都会送上级域	送	送	送
通用名称 不带www. 送www.子域	送，只有通用名称为主域 才送www.子域	送，只有通用名称为主域 才送www.子域	送	送	送
通用名称 通配符证书 送上级域	送，不论 *. 处于哪一级 都会送上级域	送，不论 *. 处于哪一级 都会送上级域	送	送	送

续表

	sslTrus	UniTrust	TLC	Digicert
通用名称 www. 开头送上级域	送	送，不论www.在二级还是三级都会送上级域	送，不论www.在二级还是三级都会送上级域	送
通用名称 不带www. 送www.子域	送	送，只有通用名称为主域 才送www.子域	送，只有通用名称为主域 才送www.子域	送
通用名称 通配符证书 送上级域	送	送，不论 *. 处于哪一级 都会送上级域	送，不论 *. 处于哪一级 都会送上级域	送

备注：注意存在特殊顶级后缀情况，如：.com.cn，.net.cn...则：通用名称.com.cn也是主域名

为什么会出现安全审核失败？

用户域名中包含非法关键字可能导致安全审核失败，非法关键字由CA定义，例如cctv，icbc等。安全审核失败后，将由CA人工介入进行审核，请耐心等待。

提交证书申请时，申请信息不要包含中文，否则可能出现“安全审查失败”。中文信息建议用拼音代替。

常见的证书申请状态有哪些？

在申请证书过程中，可能出现以下状态：

- 申请中：已提交证书申购，等待用户验证域名所有权。
- 待验证：正在进行域名所有权验证。
- 待签发：已通过域名所有权验证，等待CA签发证书。
- 已签发：证书已签发成功，可以使用。
- 验证失败：域名所有权验证失败，请确认后重新提交验证。
- 已过期：证书已过期，请尽快续费。
- 安全审核失败：用户域名中包含非法关键字导致安全审核失败，非法关键字由CA定义，例如cctv，icbc等。安全审核失败后，将由CA人工介入进行审核，请耐心等待。

如何获取WHOIS信息？

您可以到专业的域名查询网站获取域名的WHOIS信息。以whois官网为例，在搜索框中输入域名，点击“WHOIS”，即可获取该域名的WHOIS信息。

CNAME冲突如何解决？

当用户在DV证书的域名验证选择了DNS方式验证时，域名下面已经存在CNAME记录，有可能会导致TXT记录不生效，简称CNAME冲突。遇到这种情况用户可以通过以下几种方法解决。 1．用户可以改用文件验证的方式，继续域名的验证步骤。 2．用户可以在原来TXT记录基础前面加一级_dnsauth. 来避免冲突，如果域名为主域的话，原来的配置中的@修改为_dnsauth。

各品牌OV/EV证书申请材料都有哪些？

各品牌OV/EV证书申请材料列表如下：

- Globalsign / UniTrust / sslTrus / GeoTrust / SecureSite / Digicert / CFCA品牌

  证书申请表或确认函，或企业对CA机构的授权书

- TLC品牌（以下三种方式均支持）

1. 使用企业邮箱认证：必须是该企业官网域名的邮箱
2. 使用证件和授权书认证：①申请人有效证件（身份证、驾驶证、护照）②提供签字盖章版授权书，并加盖公司章，部门章需体现公司全名（若申请人为公司法人，则无需提供）
3. 使用盖章后信息确认函认证：审核时间将延长1-2个工作日

- TrustAsia 品牌

  证书申请表、企业信息确认函

> 注意：资料都需要加盖公章

SSL证书部署问题

如何导入证书？

证书申请成功后将自动导入证书管理模块，无需进行手动操作。

如果您是通过其它服务提供商申请的证书，也可以手动导入证书，具体操作步骤请参看上传证书。

是否可以下载SSL证书到本地？

无论是免费证书或是付费证书，证书申请后都可在证书管理中进行自助下载。

百度申请的SSL证书能否部署在别的云服务器上？

可以正常部署在其他服务器上，无需其他操作，正常部署即可。

SSL证书生效问题

如何在浏览器中检查SSL证书是否生效？

浏览器中受信任的根证书颁发机构存储区验证服务器身份后，会颁发SSL数字证书，该证书具有网站身份验证和加密传输的双重功能。以下是在浏览器中，检查SSL证书是否生效的操作步骤。

1. 打开浏览器，在浏览器地址栏中输入以下格式的URL。`https://[$Domain]`

> 说明：[$Domain]为您的数字证书绑定的域名。

2. 按回车键，访问上一步的URL，确认可以成功访问网站，同时浏览器地址栏左侧显示绿色安全锁标志，说明您的SSL数字证书已生效。

服务器IP地址更换后原来的SSL证书能否生效？

SSL证书都是绑定域名的，不受服务器更换IP地址的影响。只要证书绑定的域名不变，就可以重新解析到新的IP地址，原来的SSL证书仍然可以生效，不需要更换新的证书。

终端的浏览器提示证书不可信的排查方法

以下介绍电脑或者手机的浏览器提示证书不可信的排查方法。

排查终端类型

确认您所购买的数字证书品牌和提示证书不可信的终端类型。部分品牌的数字证书在某些终端中不被支持，请参考该品牌的数字证书相关介绍。目前市场中的主流设备兼容Symantec和GeoTrust两个品牌的数字证书。

> 说明：Chrome 53版本存在已知的问题导致不兼容Symantec和GeoTrust品牌证书，详细信息请参见以下文档：
>
> - Chrome 53 Bug Affecting Symantec SSL/TLS Certificates
> - Warning | Certificate Transparency error with Chrome 53

排查证书的配置与部署

若数字证书与终端兼容，建议您参考以下步骤检查证书的配置与部署：

1. 使用检查工具 GeoCerts SSL Checker 进行检查数字证书。以下是不同检查结果的排查方法：
    1. 如果检查结果中的证书品牌、证书类型或域名与您购买的证书中的配置不一致，请确认已正确配置服务器的数字证书相关信息。
    2. 如果检查结果显示证书链信息不完整，请确认已正确配置服务器的数字证书相关信息。

    > 说明：证书服务提供的PEM格式数字证书包含两段内容，两段内容中任何一段都不能丢失。如果两段内容之间存在空白行，请删除空白行。配置修改完成后重启Web服务，并重新检查。

2. 确认数字证书配置中已关闭了不安全的协议，例如SSL v3等存在隐患的协议。
3. 检查服务器站点的网页是否引用了HTTP资源。在部分浏览器中，HTTPS站点引用HTTP资源会被视为不安全操作。
4. 如果一个域名被解析到多台服务器中，请确认每台服务器都正确部署了证书。

浏览器访问相关问题

Chrome浏览器提示错误

使用Chrome 53版本，QQ浏览器9.5.1版本（内置chrome53内核）在访问HTTPS网站时，出现以下报错信息。

```
NET::ERR_CERTIFICATE_TRANSPARENCY_REQUIRED 错误
```

问题原因

该问题为Chrome 53版本浏览器的BUG，导致显示HTTPS网站异常。

解决方案

采用非53版本的Chrome浏览器就可避免上述报错。

Chrome浏览器使用域名访问网站失败

已经将证书导入至Web应用，且在SSL设置中启用证书。将Web应用的域名与SSL证书绑定之后，在Chrome浏览器中使用域名访问网站失败。

问题原因

服务器端加密套件、协议未优化，需要更新配置。

解决方案

1. 在服务器中 下载工具。
2. 运行工具，单击最佳配置>应用。
3. 然后重启服务器，使配置生效。
4. 确认使用域名访问网站成功。

访问网站时显示为旧证书

已正确配置SSL证书，但访问网站时仍显示为旧证书。

问题原因

前端设备配置了旧证书。

解决方案

检查前端设备是否部署了旧的证书，例如SLB、CDN和WAF等产品，若已配置旧证书，则请更新证书。

在IIS部署服务证书后访问资源出现404报错

在IIS中部署服务证书后，访问资源出现404报错。

问题原因

可能存在的部分原因如下所示：

- HTTPS和对应的HTTPS服务绑定的站点不统一。
- 站点信息配置错误。

解决方案

成功部署证书后，通过HTTP协议访问资源正常，通过HTTPS协议无法访问资源并出现404错误提示。如果您在IIS服务器中配置了证书，且防火墙开启了443端口，可参考以下两个方面排查问题。

- HTTP和HTTPS可以设置不同的网站根目录。在IIS服务器中，检查站点的443端口绑定情况，并确认443端口绑定的站点与期望显示的HTTP服务80端口的站点相同。

- 检查端口绑定情况时，检查设置站点的IP地址、主机名的正确性。