PKI資料格式

目錄

簡介

本檔案介紹最常用的公開金鑰基礎架構(PKI)資料格式和編碼。

必要條件

需求

思科建議您瞭解以下主題：

- 公鑰加密（基本概念）。
- 公開金鑰基礎架構（基本概念）。

採用元件

本文件所述內容不限於特定軟體和硬體版本。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路正在作用，請確保您已瞭解任何指令可能造成的影響。

慣例

如需檔案慣例的相關資訊，請參閱思科技術提示慣例。

ASN.1表示法

抽象語法表示法1(ASN.1)是一種形式化語言，用於定義資料型別和值，以及在各種資料結構中如何使用和組合這些資料型別和值。該標準的目的是定義資訊的抽象語法，而不限制資訊如何被編碼以供傳輸。

以下是一個摘自*X.509 RFC*的範例：

```
   Version  ::=  INTEGER  {  v1(0), v2(1), v3(2)  }
CertificateSerialNumber ::= INTEGER
Validity ::= SEQUENCE {
notBefore Time,
notAfter Time }
Time ::= CHOICE {
utcTime UTCTime,
generalTime GeneralizedTime }
```

請參閱國際電信聯盟(ITU-T)標準站點中的以下文檔：

- X.680 ASN.1:基本符號的規範
- X.681 ASN.1:資訊對象規範
- X.682 ASN.1:約束指定
- X.683 ASN.1:ASN.1規範引數化

ITU-T建議搜尋 — 在記錄**中搜尋X.509或standard,Language設定為ASN.1**。

BER/CER/DER編碼

ITU-T定義了一種將ASN.1中描述的資料結構編碼為二進位制資料的標準方法。X.690定義基本編碼規則(BER)及其兩個子集，規範編碼規則(CER)和可分辨編碼規則(DER)。 這三個欄位都基於封裝在分層結構中的**type-length-value**資料欄位，該分層結構由**SEQUENCE**、**SET**和**CHOICE**構建，具有以下差異：

- BER提供多種對相同資料進行編碼的方法，不適合於加密操作。
- CER提供明確的編碼，使用不定長度的資料，並在特定情況下使用資料結束標籤。
- DER提供明確的編碼，並在特定情況下使用明確的長度標籤。
- 在這三種協定中，DER是處理PKI和加密有效負載時通常遇到的協定。

範例：在DER中，20位值1010 101 1100 1101 1110編碼為：

- **tag**:0x03(位字串)
- **長度**:0x04(位元組)
- **值：** 0x**04**ABCDE**0**
- **完整的DER編碼**：0x030404ABCDE0

前導**04**表示必須捨棄編碼值的最後4位(等於尾部**0**位)，因為編碼值不會在位元組邊界上結束。

請參閱TU-T標準網站中的以下文檔：

- X.690 ASN.1編碼規則：基本編碼規則(BER)、規範編碼規則(CER)和可分辨編碼規則(DER)的規範

在維基百科網站上，請參閱以下文檔：

- 基本編碼規則
- 規範編碼規則
- 可分辨編碼規則

DER十六進位制轉儲

Cisco IOS、Adaptive Security Appliance(ASA)和其他裝置使用**show running-config**命令將DER內容顯示為**十六進位制轉儲**。 以下為輸出內容：

```
crypto pki certificate chain root
certificate ca 01
30820213 3082017C A0030201 02020101 300D0609 2A864886 F70D0101 04050030
1D310C30 0A060355 040B1303 54414331 0D300B06 03550403 1304726F 6F74301E
170D3039 30373235 31313436 33325A17 0D313230 37323431 31343633 325A301D
...
```

這種十六進位制轉儲可以通過多種方式轉換回DER。例如，可以刪除空格字元並將其引導到**xxd程式**:

```
$ cat ca.hex | tr -d ' ' | xxd -r -p -c 32 | openssl x509 -inform der -text -noout
```

另一種簡單方法是使用此Perl指令碼：

```
#!/usr/bin/perl
foreach (<>) {
s/[^a-fA-F0-9]//g;
print join("", pack("H*", $_));
}

$ perl hex2der.pl < hex-file.txt > der-file.der
```

此外，還有一種轉換**證書轉儲**的簡潔方法，每個轉儲之前都通過**bash**命令列手動複製到副檔名為**.hex**的檔案，如下所示：

```
for hex in *.hex; do
b="${hex%.hex}"
hex2der.pl < "$hex" > "$b".der
openssl x509 -inform der -in "$b".der > "$b".pem
openssl x509 -in "$b".pem -text -noout > "$b".txt
done
```

每個檔案都會產生以下結果：

- **file.hex — 原始**檔案（必須僅包含十六進位制數字）。
- **file.der -** DER（二進位制）格式的證書。
- **file.pem -** PEM（Base64 +頁首/頁尾）格式的證書。
- **file.txt — 用**戶友好且可讀的憑證版本。

Base64編碼

Base64編碼表示只有64個可列印字元(A-Za-z0-9+/)的二進位制資料，與uuencode**類似**。在二進位制到Base64的轉換中，原始資料的每個6位塊都被編碼成一個8位可列印的ASCII字元和一個轉換表。因此，編碼後的資料大小增加了33%（資料乘以8除以6位，等於1.333）。

24位緩衝器用於將三(3)組八(8)位轉換為四(4)組六(6)位。因此，在輸入資料流的末尾可能需要一(1)或二(2)位元組的填充。填充在Base64編碼的資料的末尾以一個等號(=)表示，每組八(8)入。

請參閱來自Wikipedia的示例。

請參閱RFC 4648中的最新資訊：Base16、Base32和Base64資料編碼。

PEM編碼

Privacy Enhanced Mail(PEM)是一種完整的Internet工程任務組(IETF)PKI標準，用於交換安全郵件。它不再被廣泛使用，但其封裝語法被廣泛借用，以便格式化和交換Base64編碼的PKI相關資料。

PEM RFC 1421第4.4節：封裝機制，定義由封裝邊界(EB)分隔的PEM訊息(基於RFC 934)，格式如下：

```
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
Header: value
Header: value
...

Base64-encoded data
...
-----END PRIVACY-ENHANCED MESSAGE-----
```

在今天的實踐中，當分發PEM格式的資料時，會使用以下邊界格式：

```
-----BEGIN type-----
...
-----END type-----
```

type可以使用其他金鑰或證書，例如：

- RSA
-
-
-
- X509 CRL

附註：雖然RFC並未將此規定為必填，但EB中的前導和尾隨短劃線(-)的數量相當大，應該始終為五(5)。 否則，某些應用程式（例如OpenSSL）會阻塞輸入。另一方面，其他應用（例如Cisco IOS）完全不需要EB。

如需詳細資訊，請參閱以下最近的RFC:

- RFC 1421 :PEM第I部分：報文加密和驗證程式
- RFC 1422 :PEM第II部分：基於憑證的金鑰管理
- RFC 1423 :PEM第III部分：演算法、模式和識別符號
- RFC 1424 :PEM第四部分：關鍵認證和相關服務

X.509憑證和CRL

X.509是X.500的子集，它是有關開放系統互連的擴展ITU規範。它專門處理證書和公鑰，並被IETF調整為Internet標準。X.509提供在RFC中用ASN.1表示法表示的結構和語法，以儲存憑證資訊和憑證撤銷清單。

在X.509 PKI中，CA會發出繫結公共金鑰的證書，例如：Rivest-Shamir-Adleman(RSA)或數位簽章演算法(DSA)金鑰用於特定可分辨名稱(DN)，或替代名稱(如電子郵件地址或完全限定域名(FQDN))。 DN遵循X.500標準中的結構。以下是範例：

```
CN=common-nameOU=organizational-unitO=organizationL=locationC=country
```

由於ASN.1的定義，X.509資料可以編碼為DER以便以二進位制形式交換，並可選轉換為Base64/PEM用於基於文本的通訊方式，例如終端上的複製貼上。

- 請參閱此ITU-T標準文檔X.509開放系統互聯 — 目錄：公鑰和屬性證書框架。
- 請參閱RFC 5280:X.509憑證和憑證撤銷清單(CRL)設定檔以瞭解詳細資訊。

PKCS標準

公開金鑰加密標準(PKCS)是來自RSA Labs的規範，已部分發展為行業標準。最常遇到的問題涉及這些主題；然而，它們並不都處理資料格式。

PKCS#1(**RFC 3347**) — 涵蓋基於RSA的加密（加密原語、加密/簽名方案、ASN.1語法）的實施方面。

PKCS#5(**RFC 2898**) — 涵蓋基於密碼的金鑰派生。

PKCS#7(**RFC 2315**)和**S/MIME RFC 3852 - 定義用於傳輸已簽名和加密的資料以及相關證書的消息語法。通常僅用作X.509證書的容器。

PKCS#8 — 定義消息語法以傳輸明文或加密的RSA金鑰對。

PKCS#9(**RFC 2985**) — 定義其他對象類和身份屬性。

PKCS#10(**RFC 2986**) — 定義憑證簽署請求(CSR)的訊息語法。CSR由實體傳送到CA，並包含要由CA簽署的資訊，例如公開金鑰資訊、身分和其他屬性。

PKCS#12 — 定義一個容器，用於將相關PKI資料(通常為entity keypair + entity cert + root和中繼CA證書)打包到單個檔案中。它是Microsoft個人資訊交換(PFX)格式的演變。

請參閱以下資源：
- 關於PKCS的維基百科文章
- PKCS上的RSA Labs頁面

相關資訊

- **技術支援與文件 - Cisco Systems**