Formatos de datos PKI

Contenido

Introducción

Este documento describe los formatos de datos y las codificaciones mas comunes del Public Key Infrastructure (PKI).

Prerequisites

Requisitos

Cisco recomienda que tenga conocimiento sobre estos temas:

- Cifrado de clave pública (conceptos básicos).
- Public-Key Infrastructure (conceptos básicos).

Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando,

asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Convenciones

Consulte Convenciones de Consejos Técnicos de Cisco para obtener información sobre las convenciones sobre documentos.

Notación ASN.1

El Abstract Syntax Notation One (ASN.1) es un lenguaje formal para la definición de los tipos de datos y de los valores, y cómo utilizan y se combinan a esos tipos de datos y valores en las diversas estructuras de datos. La meta del estándar es definir el abstract syntax de la información sin obligar cómo la información se codifica para la transmisión.

Aquí está un ejemplo extractado del *X.509 RFC*:

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

Refiera a estos documentos de los sitios de los estándares de la Unión Internacional de Telecomunicaciones (ITU-T):

- X.680 ASN.1: Especificación de la notación básica
- X.681 ASN.1: Especificación del objeto de la información
- X.682 ASN.1: Especificación del obstáculo
- X.683 ASN.1: Parametrización de las especificaciones ASN.1

Búsqueda de las recomendaciones ITU-T - Búsqueda para el **X.509** en el **Rec. o estándar** con el **lenguaje** fijado al **ASN.1**.

Codificaciones BER/CER/DER

El ITU-T ha definido a un modo estándar de estructuras de datos de la codificación descritas en el ASN.1 en los datos binarios. X.690 define los Basic Encoding Rule (BER) y sus dos subconjuntos, reglas canónicas de la codificación (CER) y reglas distinguidas de la codificación (DER). Los tres se basan en los campos de datos del **Type Length Value** pila de discos en una estructura jerárquica, que se construye de las **secuencias**, de los **conjuntos**, y de las **opciones**, con estas diferencias:

- El BER proporciona las diferentes formas de codificar los mismos datos, que no se adapta para las operaciones crypto.
- El CER proporciona la codificación inequívoca y utiliza los datos indefinidos de la longitud, con una etiqueta de plástico fin-de-DATA en los casos específicos.

- El DER proporciona la codificación inequívoca y utiliza las etiquetas explícitas de la longitud en los casos específicos.
- Entre los tres, el DER es el que se encuentra generalmente al tratar del PKI y de las cargas útiles crypto.

Ejemplo: En el DER, 20-bit el valor `1010 1011 1100 1101 1110` se codifica como:

- **etiqueta**: `0x03` (el bitstring)
- **longitud**: `0x04` (bytes)
- **valor**: `0x04ABCDE 0`
- **codificación completa DER**: `0x030404ABCDE0`

Los `04` principales significa que los bits del último 4 (iguala los **`0`** dígitos que se arrastra) del valor codificado deben ser desechados porque el valor codificado no termina en un límite de bytes.

Refiera a estos documentos del sitio de los estándares TU-T:

- Reglas de la codificación X.690 ASN.1: La especificación de los Basic Encoding Rule (BER), de las reglas canónicas de la codificación (CER) y de la codificación distinguida gobierna (el DER)

Del sitio de Wikipedia, refiera a estos documentos:

- Basic Encoding Rule
- Reglas canónicas de la codificación
- Reglas distinguidas de la codificación

Vaciado Hex DER

Cisco IOS, dispositivo de seguridad adaptante (ASA), y contenido de la visualización DER de los otros dispositivos como **vaciado Hex** con el **comando show running-config**. Aquí está la salida:

```
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
...
```

Esta clase de vaciado Hex se puede convertir de nuevo al DER en las distintas maneras. Por ejemplo, usted puede quitar los caracteres de espacio y transmitirlos al **programa del xxd**:

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

Otra forma sencilla es utilizar este script Perl:

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

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

Además, una manera compacta de convertir los **volcados CERT**, cada uno copiado previamente

manualmente a un archivo con la extensión **.hex**, de una línea de comando del **golpe** como se muestra aquí:

```
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
```

Cada archivo da lugar a:

- **file.hex -** El archivo original (debe contener los dígitos hexadecimales solamente).
- **file.der - Certificado** en el formato (binario) DER.
- **file.pem - Certificado** en el formato PEM (base64 + encabezado/pie de página).
- **file.txt -** Versión convivial, legible del certificado.

Codificación del base64

La codificación del base64 representa los datos binarios con solamente 64 caracteres imprimibles (`A-Za-z0-9+/`) semejantemente al **uuencode**. En la conversión del binario al base64, cada bloque 6-bit de las informaciones originales se codifica en un carácter ASCII imprimible de 8 bits con una tabla de traducción. Por lo tanto, el tamaño de los datos después de que la codificación haya aumentado en el 33 por ciento (datos mide el tiempo de 8 divididos por 6 bits, iguales 1.333).

Un buffer 24-bit se utiliza para la traducción de tres (3) grupos de ocho (8) bits en cuatro (4) grupos de seis (6) bits. Por lo tanto un (1) o dos (2) bytes de relleno se pudieron requerir en el extremo de la secuencia de datos de entrada. El relleno es indicado en el final de los datos codificados en base64, por uno iguala (`=`) `la` muestra para los bits de cada relleno del grupo de ocho (8) agregados a la entrada durante la codificación.

Refiera a este ejemplo de Wikipedia.

Refiera a la mayoría de la información reciente en el RFC 4648: El Base16, el Base32, y las codificaciones de los datos del base64.

Codificación PEM

Privacy Enhanced Mail (PEM) es un estándar completo de la Fuerza de tareas de ingeniería en Internet (IETF) (IETF) PKI para intercambiar los mensajes seguros. Es no más ampliamente utilizado como tal, pero su sintaxis de la encapsulación se ha pedido prestado extensamente para formatar e intercambiar los datos PKI-relacionados codificados en base64.

El RFC 1421 PEM, sección 4.4: El mecanismo de encapsulación, define los mensajes PEM según lo delimitado por los límites de la encapsulación (EBs), que se basan en el RFC 934, con este formato:

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

```
...

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

En la práctica hoy, cuando se distribuyen los datos PEM-formatados, se utiliza este formato del límite:

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

el tipo puede estar con otras claves o Certificados por ejemplo:

- `CLAVE PRIVADA RSA`
- `CLAVE PRIVADA CIFRADA`
- `CERTIFICADO`
- `PEDIDO DE CERTIFICADO`
- `X509 CRL`

Note: Aunque los RFC no hagan este obligatorio, el número de rociadas principales y que se arrastran (-) en el EBs es significativo y debe siempre ser cinco (5). Si no, algunas aplicaciones, tales como OpenSSL, obstruyen en la entrada. Por otra parte, otras aplicaciones, tales como Cisco IOS, no requieren EBs en absoluto.

Refiera a estos RFC más recientes para más información:

- RFC 1421: Parte I PEM: Cifrado y procedimientos de autenticación del mensaje
- RFC 1422: Parte II PEM: Administración de claves basada en el certificado
- RFC 1423: Pieza III PEM: Algoritmos, modos, e identificadores
- RFC 1424: Pieza IV PEM: Certificación y servicios relacionados dominantes

Certificados X.509 y CRL

El X.509 es un subconjunto de X.500, que es una especificación ITU extendida sobre el OSI (Open Systems Interconnection). Se ocupa específicamente de los Certificados y de las claves públicas y ha sido adaptado como norma de Internet por el IETF. El X.509 proporciona una estructura y un sintaxis, expresados en el RFC con la notación ASN.1, para salvar la información del certificado y los Lista de revocación de certificados (CRL).

En un X.509 PKI, CA publica un certificado que ate una clave pública, por ejemplo: una clave del Rivest-Shamir-Adleman (RSA) o del Digital Signature Algorithm (DSA) a un Nombre distintivo (DN) determinado, o a un nombre alternativo tal como una dirección de correo electrónico o un nombre de dominio completo (FQDN). El DN sigue la estructura en los estándares X.500. Aquí tiene un ejemplo:

```
CN=common-name, OU=organizational-unit, O=organization, L=location,
C=country
```

Debido a la definición ASN.1, los datos X.509 se pueden codificar en el DER para ser intercambiado en la forma binaria, y opcionalmente, convertida a Base64/PEM para los medios de la comunicación basados texto, tales como copia-goma en una terminal.

- Refiera a este OSI (Open Systems Interconnection) del documento X.509 de los estándares

ITU-T - el directorio: La clave pública y el atributo certifican los marcos.
- Refiera al RFC 5280: Perfil del certificado X.509 y del Listas de revocación de certificados (CRL) para más información.

Estándares PKCS

Los estándares del Cifrado de clave pública (PKCS) son especificaciones de los laboratorios RSA que se han desarrollado en parte en los estándares de la industria. Ésos encontrados lo más a menudo posible, trato con estos temas; sin embargo, no todos trato con los formatos de datos.

PKCS#1 (RFC 3347) - Cubre los aspectos de la implementación de la criptografía RSA-basada (primitivos, los esquemas del cifrado/de la firma, el sintaxis crypto ASN.1).

PKCS#5 (RFC 2898) - Cubre la derivación dominante basada en la contraseña.

PKCS-7 (RFC 2315) y **RFC 3852 S/MIME -** define un sintaxis del mensaje para transmitir firmado y los datos encriptados y los Certificados relacionados. De uso frecuente simplemente como envase para los Certificados X.509.

PKCS#8- define un sintaxis del mensaje para transportar el texto claro o los pares de claves RSA cifrados.

PKCS#9 (RFC 2985) - Define las clases de objeto y los atributos de identidad adicionales.

PKCS-10 (RFC 2986) - Define un sintaxis del mensaje para los pedidos de firma de certificado (CSR). Un CSR es enviado por una entidad a CA y contiene la información que se firmará por CA, tal como información de clave pública, identidad, y atributos adicionales.

PKCS-12 - Define un envase para los datos relacionados de empaquetado PKI (típicamente, **keypair de la entidad + entidad CERT + raíz y los Certificados de CA del intermedio**) dentro de un archivo único. Es una evolución del formato del intercambio de la información personal de Microsoft (PFX).

Refiera a estos recursos:
- Artículo de Wikipedia sobre el PKCS
- Página de los laboratorios RSA en el PKCS

Información Relacionada

- Soporte Técnico y Documentación - Cisco Systems