Formatos de dados PKI

Índice

Introdução

Este documento descreve os formatos de dados e as codificações os mais comuns do Public Key Infrastructure (PKI).

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- criptografia de chave pública (conceitos básicos).
- Public-Key Infrastructure (conceitos básicos).

Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

Convenções

Consulte as Convenções de Dicas Técnicas da Cisco para obter informações sobre convenções de documentos.

Notação ASN.1

O Abstract Syntax Notation One (ASN.1) é uma linguagem formal para a definição dos tipos de dados e dos valores, e como aqueles tipos de dados e valores são usados e combinados em várias estruturas de dados. O objetivo do padrão é definir o abstract syntax da informação sem forçar como a informação é codificada para a transmissão.

Está aqui um exemplo excerpted do *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 }
```

Refira estes documentos dos locais dos padrões da união de telecomunicação internacional (ITU) (ITU-T):

- X.680 ASN.1: Especificação da notação básica
- X.681 ASN.1: Especificação do objeto da informação
- X.682 ASN.1: Especificação da limitação
- X.683 ASN.1: Parametrização das especificações ASN.1

Busca das recomendações do ITU-T - Busca para o **X.509 no Rec. ou padrão** com a **língua** ajustada ao **ASN.1**.

Codificações BER/CER/DER

O ITU-T definiu uma maneira padrão das estruturas de dados da codificação descritas no ASN.1 em dados binários. X.690 define os Basic Encoding Rule (BER) e os seus dois subconjuntos, regras canônicas da codificação (CER) e distintas regras da codificação (DER). Todos os três são baseados nos campos de dados do **Type Length Value** embalados em uma estrutura hierárquica, que seja construída das **sequências**, dos **grupos**, e das **escolhas**, com estas diferenças:

- O BER fornece formas múltiplas de codificar os mesmos dados, que não são seridos para operações criptos.
- O CER fornece a codificação inequívoca e usa dados indefinidos do comprimento, com um marcador fim--DATA em casos específicos.
- O DER fornece a codificação inequívoca e usa etiquetas explícitas do comprimento em casos específicos.
- Entre os três, o DER é esse que é encontrado geralmente ao tratar o PKI e as cargas úteis

criptos.

Exemplo: No DER, 20-bit o valor `1010 1011 1100 1101 1110` é codificado como:

- **etiqueta**: `0x03` (bitstring)
- **comprimento**: `0x04` (bytes)
- **valor**: `0x04ABCDE 0`
- **termine a codificação DER**: `0x030404ABCDE0`

`Os 04 de` condução significam que os últimos 4 bit (iguala os **`0`** dígitos de arrasto) do valor codificado devem ser rejeitados porque o valor codificado não termina em um limite de byte.

Refira estes documentos do local dos padrões TU-T:

- Regras da codificação X.690 ASN.1: A especificação dos Basic Encoding Rule (BER), das regras canônicas da codificação (CER) e de distinta codificação ordena (o DER)

Do local de Wikipedia, refira estes documentos:

- Basic Encoding Rule
- Regras canônicas da codificação
- Distintas regras da codificação

Cópia parcial da memória de HEX DER

Cisco IOS, ferramenta de segurança adaptável (ASA), e índice do indicador DER dos outros dispositivos como uma **cópia parcial da memória de HEX** com o **comando show running-config**. Está aqui a saída:

```
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
...
```

Este tipo da cópia parcial da memória de HEX pode ser convertido de volta ao DER em várias maneiras. Por exemplo, você pode remover os caracteres de espaço e conduzi-los ao **programa do xxd**:

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

Uma outra maneira fácil é usar 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
```

Além, uma maneira compacta de converter **descargas CERT**, cada um copiado previamente manualmente a um arquivo com extensão **.hex**, de uma linha de comando da **festança** como mostrado aqui:

```
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 arquivo conduz a:

- **file.hex -** O arquivo original (deve conter encantam dígitos somente).
- **file.der - Certificado** no formato (binário) DER.
- **file.pem - Certificado** no formato PEM (Base64 + encabeçamento/pé de página).
- **file.txt -** Versão fácil de usar, legível do certificado.

Codificação de Base64

A codificação de Base64 representa os dados binários com os somente 64 caráteres imprimíveis (`A-Za-z0-9+/`) similarmente ao **uuencode**. Na conversão do binário a Base64, cada bloco 6-bit dos dados originais é codificado em um caractere ASCII imprimível de 8 bits com uma tabela de tradução. Consequentemente, o tamanho dos dados depois que codificar aumentou por 33 por cento (dados cronometra 8 divididos pelos bit 6, pelos iguais 1.333).

Um buffer 24-bit é usado para uma tradução de três (3) 8) bit dos grupos de oito (em quatro (4) grupos de seis (6) bit. Consequentemente um (1) ou dois (2) bytes de preenchimento puderam ser exigidos na extremidade do fluxo de dados de entrada. O estofamento é indicado no fim dos dados Base64-encoded, por um iguala (`=`) o sinal para bit de cada estofamento do grupo de oito (8) adicionados à entrada durante a codificação.

Refira este exemplo de Wikipedia.

Refira a maioria de informação recente no RFC 4648: O Base16, o Base32, e as codificações dos dados de Base64.

Codificação PEM

O Privacy Enhanced Mail (PEM) é um padrão completo do Internet Engineering Task Force (IETF) PKI a fim trocar mensagens seguras. É já não amplamente utilizado como tal, mas sua sintaxe do encapsulamento foi pedida extensamente a fim formatar e trocar dados PKI-relacionados Base64-encoded.

O RFC 1421 PEM, seção 4.4: O mecanismo de encapsulamento, define mensagens PEM como limitadas por limites do encapsulamento (EBs), que são baseadas no RFC 934, com este formato:

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

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

Na prática hoje, quando os dados PEM-formatados são distribuídos, este formato do limite é usado:

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

o tipo pode ser com outros chaves ou Certificados como:

- CHAVE PRIVADA RSA
- CHAVE PRIVADA CIFRADA
- CERTIFICADO
- PEDIDO DO CERTIFICADO
- X509 CRL

Note: Embora os RFC não façam este imperativo, o número de traços de condução e de arrasto (-) no EBs é significativo e deve sempre ser cinco (5). Se não, alguns aplicativos, tais como o OpenSSL, bloqueiam na entrada. Por outro lado, outros aplicativos, tais como o Cisco IOS, não exigem EBs de todo.

Refira estes RFC os mais recentes para mais informações:

- RFC 1421: Peça PEM mim: Criptografia e procedimentos de autenticação da mensagem
- RFC 1422: Parte II PEM: Gerenciamento chave Certificado-baseado
- RFC 1423: Parte III PEM: Algoritmos, modos, e identificadores
- RFC 1424: Peça IV PEM: Certificação e serviços relacionados chaves

Certificados X.509 e CRL

O X.509 é um subconjunto do X.500, que é uma especificação de ITU prolongada sobre o Open Systems Interconnection. Trata especificamente os Certificados e as chaves públicas e foi adaptado como um padrão do Internet pelo IETF. O X.509 fornece uma estrutura e uma sintaxe, expressadas no RFC a notação ASN.1, a fim armazenar a informação do certificado e as listas de revogação de certificado.

Em um X.509 PKI, CA emite um certificado que ligue uma chave pública, por exemplo: uma chave de Rivest-Shamir-Adleman (RSA) ou do Digital Signature Algorithm (DSA) a um nome destacado (DN) particular, ou a um nome alternativo tal como um endereço email ou o nome de domínio totalmente qualificado (FQDN). O DN segue a estrutura nos padrões X.500. Aqui está um exemplo:

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

Devido à definição ASN.1, os dados X.509 podem ser codificados no DER a fim ser trocado no formulário binário, e opcionalmente, convertido a Base64/PEM por meios de uma comunicação baseados texto, tais como a cópia-pasta em um terminal.

- Refira este Open Systems Interconnection do documento X.509 dos padrões do ITU-T - o diretório: A chave pública e o atributo certificate estruturas.
- Refira o RFC 5280: Perfil do certificado X.509 e do Certificate Revocation List (CRL) para mais informação.

Padrões PKCS

Os padrões da criptografia de chave pública (PKCS) são especificações dos laboratórios RSA que evoluíram em parte em padrões para indústria. Aqueles encontrados o mais frequentemente, negócio com estes assuntos; contudo, não todo negócio com formatos de dados.

PKCS#1 (RFC 3347) - Cobre os aspectos da aplicação da criptografia RSA-baseada (primitivos criptos, esquemas da criptografia/assinatura, sintaxe ASN.1).

PKCS#5 (RFC 2898) - As tampas senha-basearam a derivação chave.

RFC 3852 **PKCS#7 (RFC 2315)** e **S/MIME -** define uma sintaxe da mensagem para transmitir assinado e dados criptografados e Certificados relativos. Usado frequentemente simplesmente como um recipiente para os Certificados X.509.

PKCS#8- define uma sintaxe da mensagem para transportar o texto não criptografado ou pares de chave RSA cifrados.

PKCS#9 (RFC 2985) - Define classes de objeto e atributos de identidade adicionais.

PKCS#10 (RFC 2986) - Define uma sintaxe da mensagem para as solicitações de assinatura de certificado (CSR). Um CSR é enviado por uma entidade a CA e contém a informação a ser assinada por CA, tal como a informação de chave pública, a identidade, e atributos adicionais.

PKCS-12 - Define um recipiente para dados relacionados de empacotamento PKI (tipicamente, **keypair da entidade + entidade CERT + raiz e certificados de CA do intermediário**) dentro de um arquivo único. É uma evolução do formato da troca de informação pessoal de Microsoft (PFX).

Refira estes recursos:

- Artigo de Wikipedia em PKCS
- Página dos laboratórios RSA em PKCS

Informações Relacionadas

- **Suporte Técnico e Documentação - Cisco Systems**