PKI-gegevensformaten

Inhoud

Inleiding

In dit document worden de meest gebruikelijke bestandsindelingen en -coderingen voor de openbare sleutelinfrastructuur (PKI) beschreven.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- cryptografie van de openbare sleutel (basisconcepten).
- infrastructuur van de openbare sleutel (basisconcepten).

Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

Conventies

Raadpleeg Cisco Technical Tips Convention voor informatie over documentconventies.

ASN.1 Opmerking

Abstract Syntax notatie 1 (ASN.1) is een formele taal voor de definitie van gegevenstypen en -waarden, en hoe deze gegevenstypen en -waarden in verschillende gegevensstructuren worden gebruikt en gecombineerd. Het doel van de standaard is de abstracte syntaxis van informatie te definiëren zonder te beperken hoe de informatie voor transmissie wordt gecodeerd.

Hier is een voorbeeld uit de *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 }
```

Raadpleeg deze documenten van de standaardinstellingen van de International Telecommunication Union (ITU-T):

- X.680 ASN.1: Specificatie van basisnotatie
- X.681 ASN.1: Specificatie van informatieobject
- X.682 ASN.1: Beperkingsspecificatie
- X.683 ASN.1: parameterbepaling van ASN.1-specificaties

Zoeken op ITU-T-aanbevelingen - Zoeken naar **X.509** in **Rec. of standaard** met **Taal** ingesteld op **ASN.1**.

BER/CER/DER-encoderingen

De ITU-T heeft een standaardmanier gedefinieerd om gegevensstructuren te coderen die in ASN.1 zijn beschreven in binaire gegevens. X.690 definieert Basis-coderingsregels (BER) en de twee subgroepen daarvan, Canonical Encoding Rules (CER) en gescheiden coderingsregels (DER). Alle drie zijn gebaseerd op gegevensvelden **van het type-lengte** die in een hiërarchische structuur zijn ingepakt en die zijn opgebouwd uit **SEQUENCEs**, **SETs** en **KEUZEs**, met deze verschillen:

- BER biedt verschillende manieren om dezelfde gegevens te coderen, wat niet geschikt is voor cryptooperaties.
- CER biedt een duidelijke codering en gebruikt gegevens met een onbepaalde lengte, met een end-of-data markering in specifieke gevallen.
- DER biedt duidelijke codering en gebruikt in specifieke gevallen expliciete lengte tags.
- Onder de drie is DER degene die gewoonlijk wordt aangetroffen bij het omgaan met PKI en crypto payloads.

Voorbeeld: In DER wordt de 20-bits waarde 1010 1011 100 1101 1110 gecodeerd als:

- **tag**: `0x03` (bitstring)
- **lengte**: `0x04` (bytes)
- **waarde**: `0x04 ABCDE0`
- **volledige versleuteling DER**: `0x030404ABCDE0`

Het leidend **04** betekent dat de laatste 4 bits (gelijk aan het cijfer **0**) van de gecodeerde waarde moet worden weggegooid omdat de gecodeerde waarde niet op een bytegrens eindigt.

Raadpleeg deze documenten op de website van de TU-T-standaard:

- X.690 ASN.1-coderingsregels: Specificatie van basis-coderingsregels (BER), Canonical Encoding Rules (CER) en onderscheiden coderingsregels (DER)

Raadpleeg de volgende documenten op de Wikipedia-site:

- Basiscoderingsregels
- Canonische coderingsregels
- Gedifferentieerde coderingsregels

DER hex Dump

Cisco IOS, Adaptieve security applicatie (ASA), en andere apparaten tonen de inhoud van DER als een **hex stortplaats** met de **show in werking stellen-configuratie** opdracht. Dit is de output:

```
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
...
```

Dit soort hex-afvalhoop kan op verschillende manieren worden omgezet naar DER. U kunt bijvoorbeeld de ruimtetekens verwijderen en het naar het **xxd-programma** buigen:

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

Een andere makkelijke manier is om dit Perl script te gebruiken:

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

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

Bovendien, een compacte manier om **bepaalde dumps** te converteren, die vroeger handmatig naar een bestand met extension **hex** werden gekopieerd, van een **bash** opdrachtregel zoals hieronder wordt getoond:

```
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
```
Elk bestand resulteert in:

- **file.hex -** het oorspronkelijke bestand (mag alleen hex cijfers bevatten).
- **file.der -** certificaatmodel in DER (binair) formaat.
- **file.pem -** certificaatmodel in PEM (Base64 + header/voettekst)-indeling.
- **bestand.txt -** gebruikersvriendelijke, leesbare versie van het certificaat.

Base64-codering

Base64 encodering representeert de binaire gegevens met slechts 64 printbare tekens (`A-Za-z0-9+/`) vergelijkbaar met **uencode**. In de conversie van binair naar Base64 wordt elk 6-bits blok van de oorspronkelijke gegevens gecodeerd in een 8-bits printbaar ASCII-teken met een vertaaltabel. Daarom is de grootte van de data na codering met 33 procent toegenomen (data times 8 gedeeld door 6 bits, is gelijk aan 1.333).

Een 24-bits buffer wordt gebruikt voor het vertalen van drie (3) groepen van acht (8) bits in vier (4) groepen van zes (6) bits. Daarom kan één (1) of twee (2) bytes van padding nodig zijn aan het eind van de invoergegevensstroom. Het opvullen is aangegeven aan het eind van de Base64-gecodeerde gegevens, door één gelijk (`=`) teken voor elke groep van acht (8) opvulbits die tijdens het coderen aan de ingang zijn toegevoegd.

Raadpleeg dit voorbeeld op Wikipedia.

Raadpleeg de meest recente informatie in RFC 4648: De Base16, Base32, en Base64 Data-encodings.

PEM-versleuteling

Privacy Enhanced Mail (PEM) is een volledige IETF-PKI-standaard (Internet Engineering Task Force) om beveiligde berichten uit te wisselen. Het wordt niet langer op grote schaal als dusdanig gebruikt, maar de insluitingssyntaxis is is wijdverspreid om met Base64-gecodeerde PKI-gegevens te kunnen opmaken en uitwisselen.

PEM RFC 1421, paragraaf 4.4: Het mechanisme van de insluiting definieert PEM-berichten als gedefinieerd door Encapsulation Boundaries (EB's), die gebaseerd zijn op RFC 934, met dit formaat:

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

Base64-encoded data
...
-----END PRIVACY-ENHANCED MESSAGE-----
```
In de huidige praktijk, wanneer PEM-geformatteerde gegevens worden verspreid, wordt dit

grensformaat gebruikt:

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

het type kan worden gebruikt met andere sleutels of certificaten, zoals:

- `RSA-PRIVÉ-TOETS`
- `VERSLEUTELDE PRIVATE SLEUTEL`
- `CERTIFICAAT`
- `CERTIFICAATVERZOEK`
- `X509 CRL-modulator`

Opmerking: Hoewel de RFC's dit niet verplicht maken, is het aantal leidende en handelingsdashes (-) in de ECB's aanzienlijk en dient dit altijd vijf te zijn (5). Anders kunnen bepaalde toepassingen, zoals OpenSSL, op de ingang klikken. Aan de andere kant hebben andere toepassingen, zoals Cisco IOS, helemaal geen EBs nodig.

Raadpleeg deze meest recente RFC's voor meer informatie:

- RFC 1421 : PEM Deel I: Procedures voor versleuteling en verificatie van berichten
- RFC 1422: PEM Deel II: Op certificaten gebaseerd hoofdbeheer
- RFC 1423 : PEM Deel III: Algoritmen, modi en identificatiemiddelen
- RFC 1424: PEM Deel IV: Belangrijke certificering en verwante services

X.509-certificaten en CRL's

X.509 is een subset van X.500, wat een uitgebreide ITU-specificatie is van Open Systems Interconnect. Het gaat specifiek om certificaten en openbare sleutels en is door de IETF als Internet-norm aangepast. X.509 biedt een structuur en syntaxis, uitgedrukt in de RFC met ASN.1-notatie, om certificaatinformatie en certificaatherroeping-lijsten op te slaan.

In een X.509 PKI geeft een CA een certificaat uit dat een openbare sleutel bindt, bijvoorbeeld: een Rivest-Shamir-Adleman (RSA) of Digital Signature Algorithm (DSA)-toets naar een bepaalde onderscheidde Naam (DN) of naar een alternatieve naam zoals een e-mailadres of volledig gekwalificeerd domeinnaam (FQDN). De DN volgt de structuur in de X.500 standaarden. Hierna volgt een voorbeeld:

```
CN=common name,OU=organisatorische
eenheid,O=organisatie,L=locatie,C=land
```

Vanwege de ASN.1-definitie kunnen X.509-gegevens in DER worden gecodeerd zodat ze in binaire vorm kunnen worden uitgewisseld en naar keuze worden geconverteerd naar Base64/PEM voor op tekst gebaseerde communicatiemiddelen, zoals kopieerpasta op een terminal.

- Raadpleeg dit ITU-T standaard document X.509 Open Systems Interconnect - The Directory: Publiek-sleutel en attribuut certificaatkaders.
- Raadpleeg RFC 5280: X.509-profiel (certificaatherroeping en -lijst (CRL) voor meer informatie.

PKCS-normen

De Public Key Cryptography Standards (PKCS) zijn specificaties van RSA Labs die gedeeltelijk tot industrienormen zijn geëvolueerd. De meest ondervraagden behandelen deze onderwerpen; niet alle hebben echter betrekking op gegevensformaten .

PKCS#1 (RFC 3347**)** - bestrijkt de implementatieaspecten van cryptografie op basis van RSA (crypto-primitieven, encryptie/signatuurschema's, syntaxis van ASN.1).

PKCS#5 (RFC 2898**)** - heeft betrekking op een op wachtwoord gebaseerde sleutelafleiding.

PKCS#7 (RFC 2315**)** en **S/MIME** RFC 3852 **-** definieert een berichtsyntaxis om getekende en versleutelde gegevens en verwante certificaten te verzenden. Vaak gebruikt eenvoudig als houder voor X.509-certificaten.

PKCS#8- definieert een berichtsyntaxis om duidelijk tekst of versleutelde RSA-toetsenborden te transporteren.

PKCS#9 (RFC 2985**)** - definieert extra doelklassen en identiteitskenmerken.

PKCS#10 (RFC 2986**)** - definieert een berichtsyntaxis voor certificaatsignaleringsaanvragen (CSR's). Een CSR wordt door een entiteit naar een CA verzonden en bevat de informatie die door de CA moet worden ondertekend, zoals openbare sleutelinformatie, identiteit en extra eigenschappen.

PKCS#12 - definieert een container voor verpakkingsgerelateerde PKI-gegevens (doorgaans **entiteiten** + **entiteiten** cert + **wortel- en intermediaire CA-certificaten**) in één bestand. Het is een ontwikkeling van Microsoft's Mobile Information Exchange (PFX)-formaat.

Raadpleeg deze middelen:

- Wikipedia-artikel over PKCS
- RSA-labelpagina op PKCS

Gerelateerde informatie

- **Technische ondersteuning en documentatie – Cisco Systems**