Guiões das Aulas Práticas

Segurança Informática

Licenciatura em Tecnologias da Informação

Hélder Gomes

Escola Superior de Tecnologia e Gestão de Águeda
Universidade de Aveiro

2018–2019

Conteúdo

2

Validação de certificados em Java

Resumo:

- Assinaturas digitais.
- Dispositivos PKCS#11
- Assinaturas digitais com o Cartão de Cidadão.

2.1 Introdução

Os certificados digitais de chave pública são definidos na norma X.509v3 e são geridos no contexto de PKIs (Public Key Infrastructures - Infraestruturas de Chave Pública), que definem as políticas que regulam a sua utilização. A maioria ds normas que regulam a utilizaç ao de certificados digitais X.509 na Internet, foram definidas pelo PKIX Working Group[1] da IETF.

Este guião aborda a validação de certificados. Um certificado é considerado válido, para ser utilizado para um fim específico, se as seguintes condições se verificarem: (i) o certificado é/foi usado dentro do seu período de validade, (ii) o certificado foi emitido (assinado) por uma entidade (CA) de confiança e (iii) as políticas no certificado permitem a sua utilização para o fim pretendido. Neste guião, apenas iremos abordar as duas primeiras condições acima indicadas. Para obter informação adicional, pode consultar o Java PKI Reference Guide[2].

Para a realização deste guião é necessário que tenha o software do Cartão de Cidadão instalado no seu computador.

2.2 Intervalo de validade dos certificados

Os certificados X.509 contém a definição de um intervalo de validade que define o período temporal em que eles podem ser usados.

Implemente um pequeno programa que leia um certificado do seu Cartão de Cidadão e verifique se a data actual está dentro do período de validade do certificado.

Sugestão: Use a classe X509Certificate.

2.3 Certificados âncora de confiança

Os certificados de âncora de confiança são os certificados no qual o utilizador confia e que, tipicamente, são certificados raiz (i.e., auto-assinados).

2.3.1 Leitura de certificados a partir de uma keystore

Normalmente os certificados âncora de confiança são fornecidos pelo sistema operativo ou pelos programas que deles necessitam (e.g., Firefox, Thunderbird, etc.), mas podem também ser fornecidos pelo utilizador. Estes certificados devem ser protegidos numa `Keystore`, para evitar a adição ou a

[1] `https://datatracker.ietf.org/wg/pkix/documents/`

[2] https://docs.oracle.com/javase/8/docs/technotes/guides/security/certpath/CertPathProgGuide.html

remoção de algum deles, intencionalmente ou não. O Java inclui um conjunto de certificados raiz (âncoras de confiança), que podem ser lidos do ficheiro `lib/security/cacerts`, na pasta da versão de Java que está a usar. Pode usar a seguinte linha de código para obter o nome completo do ficheiro `cacerts`.

```
String filename = System.getProperty("java.home") +
        "/lib/security/cacerts".replace('/', File.separatorChar);
```

O ficheiro `cacerts` contém uma `keystore` que pode carregar (load) usando a *password* `"changeit`).

Os certificados de âncora de confiança (ou raiz de confiança) são importantes para a validação de caminhos de certificação (ou cadeias de certificados). Qualquer caminho de certificação válido tem de terminar num certificado âncora de confiança. A classe `PKIXParameters` do Java, que especifica o conjunto de parâmetros para o algoritmo de validação de certificados definido pelo PKIX, o que inclui os certificados âncora de confiança, pode ser usada para ler todos os certificados âncora de confiança contidos numa `keystore`. O seguinte excerto de código, em que `ks` é a variável que contém o `keystore`, ilustra como isso pode ser feito:

```
PKIXParameters par = new PKIXParameters(ks);
for(TrustAnchor ta : par.getTrustAnchors() )
{
    X509Certificate c = ta.getTrustedCert();
    System.out.println( c.getSubjectDN().getName());
}
```

Implemente um pequeno programa que mostre todos os certificados de confiança (âncoras de confiança) do Java.

2.3.2 Carregamento de uma keystore com certificados

Por vezes os certificados estão sob a forma de ficheiros numa pasta, como acontece com os certificados de confiança da Mozilla, que estão na pasta `/usr/share/ca-certificates/mozilla`. Pode pois ser conveniente lê-los para uma `keystore` (por exemplo, para iniciar um objecto `PKIXParameters`). No guião sobre a utilização do Cartão de Cidadão pode ver como ler um certificado a partir de um ficheiro. O seguinte excerto de código mostra como inserir um `array` de certificados (variável `certs`) numa `keystore` vazia.

```
KeyStore ks = KeyStore.getInstance(KeyStore.getDefaultType());
ks.load(null); //empty keystore
for(Certificate c : certs ){
   X509Certificate xc = (X509Certificate) c;
   ks.setCertificateEntry(xc.getSubjectDN().getName(), c);
}
```

Implemente um pequeno programa que leia todos os certificados raiz de confiança da Mozilla e os coloque numa `keystore`.

Sugestão: Use a classe `File`.

2.4 Construção de um caminho de certificação

Uma cadeia de certificados, ou caminho de certificação, é o conjunto de certificados que compõem a cadeia de confiança, desde a âncora de confiança até ao certificado do utilizador final. Frequentemente, para validar um certificado de um utilizador final é necessário construir uma cadeia de certificação usando um conjunto de possíveis certificados intermédios. Outras vezes, o utilizador final fornece a cadeia de certificação em conjunto com o seu certificado.

O seguinte excerto de código mostra como construir uma cadeia de certificação, dados um conjunto de certificados âncora de confiança, um conjunto de possíveis certificados intermédios, e um certificado de utilizador final.

```
//defines the end-user certificate as a selector
X509CertSelector cs = new X509CertSelector();
cs.setCertificate((X509Certificate)myCert);
//Create an object to build the certification path
CertPathBuilder cpb = CertPathBuilder.getInstance("PKIX");
//Define the parameters to buil the certification path and
//provide the Trust anchor certificates (trustAnchors) and
//the end user certificate (cs)
PKIXBuilderParameters pkixBParams =
     new PKIXBuilderParameters(trustAnchors,cs);
pkixBParams.setRevocationEnabled(false); //No revocation check
//Provide the intermediate certificates (iCerts)
CollectionCertStoreParameters ccsp =
     new CollectionCertStoreParameters(Arrays.asList(iCerts));
CertStore store = CertStore.getInstance("Collection", ccsp);
pkixBParams.addCertStore(store);
//Build the certification path
```

```
CertPath cp = null;
try {
     CertPathBuilderResult cpbr = cpb.build(pkixBParams);
     cp = cpbr.getCertPath();
     System.out.println(
          "Certification path built with success!");
} catch (CertPathBuilderException ex) {
     System.out.println(
       "It was not possible to build a certification path!");
}
```

Nota: Durante a construção da cadeia de certificação, é possível verificar o estado de revogação dos certificados. No entanto, por simplicidade, o excerto de código acima não faz essa verificação. A revogação de certificados é o assunto da secção seguinte.

Descarregue da página da disciplina o conjunto de todos os certificados intermédios do Cartão de Cidadão e implemente um pequeno programa para construir o caminho de certificação para o seu certificado de autenticação, usando os certificados âncora de confiança do Java.

2.5 Validação de um caminho de certificação

Dado um caminho de certificação, podemos proceder à sua validação, i.e., verificar se as assinaturas de todos os certificados estão válidas e se nenhum dos certificados foi revogado, entre outras verificações definidas pelas normas PKIX. Um certificado revogado é um certificado que foi tornado inválido antes de terminar o seu intervalo de validade.

O PKIX define dois mecanismos para permitir verificar se um certificado foi ou não revogado: As listas de certificados revogados (CRL - Certificate Revocation Lists) e o protocolo para a verificação online do estado dos certificados (OCSP - Online Certificate Status Protocol). O Java 8 disponibiliza a classe `PKIXRevocationChecker` que pode ser configurada para verificar o estado de revogação de uma cadeia de certificação usando os dois mecanismos atrás indicados. Também permite a validação de certificados baseada em propriedades específicas contidas nos certificados, mas isso não será aqui abordado.

O seguinte excerto de código mostra como pode ser feita a validação de uma cadeia de certificação em Java8, usando o protocolo OCSP para verificar o estado de revogação dos certificados (a variável `cp` contém o caminho de certificação para validar, incluindo o certificado do utilizador final, e a variável trustAnchors contém o conjunto de certificados raíz de confiança).

```
PKIXParameters pkixParams = new PKIXParameters(trustAnchors);
//Class that performs the certification path validation
CertPathValidator cpv = CertPathValidator.getInstance("PKIX");
//Disables the previous mechanism for revocation check (pre Java8)
pkixParams.setRevocationEnabled(false);
//Enable OCSP verification
Security.setProperty("ocsp.enable", "true");
//Instantiate a PKIXRevocationChecker class
PKIXRevocationChecker rc =
     (PKIXRevocationChecker) cpv.getRevocationChecker();
//Configure to validate all certificates in chain using only OCSP
rc.setOptions(EnumSet.of(
        PKIXRevocationChecker.Option.SOFT_FAIL,
        PKIXRevocationChecker.Option.NO_FALLBACK));
PKIXCertPathValidatorResult result = null;
try {
  //Do the velidation
  result =
    (PKIXCertPathValidatorResult) cpv.validate(cp, pkixParams);
  System.out.println("Certificado Válido");
  System.out.println("Issuer of trust anchor certificate: " +
    result.getTrustAnchor().getTrustedCert().getIssuerDN().getName());
} catch (CertPathValidatorException cpve) {
  System.out.println("Validation failure, cert[" +
    cpve.getIndex() + "] :" + cpve.getMessage());
}
```

Implemente um pequeno programa para validar o caminho de certificação que construiu na secção anterior.

NOTA: Pode usar o `Wireshark` para verificar as mensagens trocadas com os servidores de OCSP para a obtenção do estado de revogação dos certificados na cadeia de certificação.

2.6 Desafios

Altere o seu programa, desenvolvido na secção anterior, para usar apenas CRL para verificar a revogação de certificados na validação da cadeia de certificação.

Altere o seu programa, para usar o OCSP em primeiro lugar e as CRL caso o OCSP falhe.

Altere o seu programa para verificar apenas o estado de revogação do certificado do utilizador final.

2.7 Bibliografia

http://docs.oracle.com/javase/tutorial/security/index.html
http://docs.oracle.com/javase/8/docs/technotes/guides/security
https://docs.oracle.com/javase/8/docs/technotes/guides/security/certpath/CertPathProgGuide.html
https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html https://docs.oracle.com/javase/8/docs/api/ https://datatracker.ietf.org/wg/pkix/charter/ https://en.wikipedia.org/wiki/X.509