***X.509***

O *X.500* foi a primeira versão da série aprovada pelo Setor de Padronização de Telecomunicações da *União Internacional de Telecomunicações (ITU-T)* no ano de 1988. O *X.500* é um conjunto de padrões para redes de computadores que tratam sobre serviços de diretório, entre eles está a definição do *DAP* (*Directory Acess Protocol*) que é a base para o *LDAP* e diversos outros serviços de diretórios. Um padrão da série *X.500* é o *X.509* que trata sobre certificados de chave pública e, devido à sua grande importância acabou até mesmo extrapolando o escopo da série.

O *X.509* está na versão 3 que foi publicada em 1996 em uma colaboração entre a *ISO/IEC*, *ITU-T* e *ANSI*. Uma característica fundamental deste certificado é a existência de um emissor, o que significa que além dos dados de identificação do titular, tem também a identificação de quem emitiu o certificado e, portanto, de quem certifica a validade dos dados. **Alguns certificados também são capazes de serem usados para emitir outros certificados.**

Entre os campos do certificado temos a versão do *X.509*, o número serial que é o identificador do certificado no escopo do emissor, o algoritmo de assinatura, a identificação do emissor, a validade do certificado, a entidade, chave pública, entre outros.

Se um certificado digital *X.509* deve ter explicitamente um emissor, este deve servir para atestar a veracidade do certificado por ele emitido. Dessa forma, se o emissor fosse alguma pessoa ou entidade desconhecida, a assinatura no certificado não ofereceria nenhuma garantia e poderia ser desconsiderada. Por isso que o padrão *X.509* estabelece que a emissão de certificados digitais aos usuários finais deve ser realizada por uma entidade confiável como uma *Autoridade Certificadora* (ou *Autoridade de Certificação - AC*).

As *Autoridades Certificadoras* devem controlar todo o ciclo de vida dos certificados que ela emite prezando pelo sigilo da própria chave privada usada na assinatura. Por isso que uma *AC* tem uma estrutura bastante robusta, com mecanismos de auditoria, tolerância a falhas, segurança com salas-cofre e documentação de procedimentos e eventos.

No entanto, as atividades mais burocráticas de recolhimento e conferência da documentação dos usuários não faz parte do escopo de uma *AC* restando isso para as *AR* (*Autoridade de Registro*) que funciona como uma subsidiária operacional da *AC*.

Para mais informações sobre como funciona a cadeia de certificação no Brasil e a ICP veja mais informações sobre o artigo de certificação digital publicada no portal DevMedia.

Na próxima seção veremos na prática **como podemos gerar um certificado digital próprio**, fazendo a nossa própria cadeia de certificação.

**Cadeia de Certificação e ICP**

Conforme vimos anteriormente alguns **certificados podem assinar outros certificados**, essa característica é especialmente útil para as *ACs* quando se quer ter um nível intermediário entre si e os usuários finais, muito em função da complexidade do cenário. Assim, as *ACs* podem ter *ACs* subsidiárias que possuem seus certificados assinados pelo certificado da *AC* principal. Os certificados dos usuários finais são assinados pelas *ACs* subsidiárias.

Esse tipo de estrutura pode ter um elevado número de níveis e de ramificações em cada nível, tornando a hierarquia bastante complexa. Porém, na prática temos três a quatro níveis para a maioria das aplicações. Nessa estrutura em árvore temos também o conceito de delegação postergada de confiança, onde a *AC* emite o certificado de um usuário final atestando a sua autenticidade, e quem atesta a autenticidade do certificado dessa *AC*, além da confiabilidade das suas práticas, é *AC* de nível superior. Dessa forma, a cada nível que subimos na árvore, a confiança é creditada para a *AC* que assina os certificados daquele nível. Esse processo de confiança vai até a *AC Raiz* que está no topo da árvore e o seu certificado é auto assinado, ou seja, ela é considerada como possuidora de confiança absoluta para seus devidos fins, de forma que nenhuma outra atividade assina o seu certificado.

Uma *infraestrutura de Chaves Públicas* (*ICP*, do inglês *Public Key Infrastructure* - *PKI*) é todo o conjunto formado pela comunidade de usuários finais, *AC Raiz* e *ACs* intermediárias. Além disso, também engloba normas e procedimentos regradores desse conjunto, e as estruturas físicas e lógicas que dão o suporte necessário ao funcionamento de todo o sistema. Uma *ICP* pode ter âmbito organizacional, acadêmico, nacional, entre outros.

No ano de 2001 o governo federal do Brasil instituiu a *ICP Brasil* (*Infra-Estrutura de Chaves Públicas Brasileira*). Essa norma especificou as bases para a criação das estruturas e das organizações que dão suporte a essa *ICP*. Entre algumas das definições o Instit*uto Nacional de Tecnologia da Informação (ITI)* foi transformado em autarquia e passou a desempenhar a função de *Autoridade Certificadora Raiz*, e foi criado um *Comitê Gestor*, que é responsável pela definição de políticas e fiscalização da *AC Raiz*. Entre as atribuições do comitê estão: adotar medidas necessárias e a coordenação da implantação e do funcionamento da *ICP-Brasil*, estabelecer a política, critérios e normas técnicas para o credenciamento das *ACs*, *ARs* e demais prestadores de serviço de suporte à *ICP-Brasil*. Outras atribuições incluem o ajuste e revisão de procedimentos e práticas da *ICP-Brasil* para garantir a compatibilidade e atualização tecnológica do sistema e conformidade com as políticas de segurança.

O papel da *ITI*, o *AC Raiz*, é de executor das normas aprovadas pelo comitê, assim como emitir, expedir, distribuir, revogar e gerenciar os certificados das *AC* de nível imediatamente subsequente. Também cabe a *AC Raiz* fiscalizar e auditar as *ACs* e *ARs* que fazem parte da *ICP*.

**Gerando um Certificado Digital Próprio**

**Gerar um certificado e inclusive nos tornarmos a nossa própria *Autoridade Certificadora*** é simples como veremos nessa sessão, basta utilizarmos as ferramentas corretas como a *OpenSSL*.

Para isso devemos primeiramente baixá-la em [***http://www.slproweb.com/products/Win32OpenSSL.html***](http://www.slproweb.com/products/Win32OpenSSL.html)*. Se quisermos baixar para Linux devemos baixá-lo no site*[***www.openssl.org***](http://www.openssl.org/).

A **Figura 1** mostra a página de download para plataforma *Windows* do *OpenSSL*. Escolha a primeira opção “*Win32 OpenSSL v1.0.1i Light*”.

**Figura 1**. Download do *OpenSSL*.

Após o download e a instalação do *OpenSSL* devemos ir até o diretório de instalação do *OpenSSL* em “*C:\OpenSSL-Win32\bin*” e executar o aplicativo *openssl.exe*. Não devemos esquecer de executá-lo como administrador, para isso basta clicar com o botão direito do mouse em cima do ícone do *prompt* e clicar em “*Executar como administrador*”.

Segue na **Figura 2** o *prompt* de comando do *OpenSSL* aberto.

**Figura 2**. *Prompt* de comando do *OpenSSL*.

Aberto o aplicativo do *OpenSSL* podemos começar gerando um certificado auto assinado (o de uma *Autoridade Certificadora*) através do comando:

|  |  |
| --- | --- |
| 1 | genrsa -des3 -out ca.key 4096 |

Este comando gera uma chave *rsa* de *4096 bits*, armazena a chave no arquivo *ca.key* e protege este arquivo cifrando-o com o *3DES*.

Após executar o comando a tela da **Figura 3** será exibida, esta tela mostra o comando sendo processado.

**Figura 3**. Comando sendo processado pelo *OpenSSL* para geração da chave *rsa*.

Após o processamento ser realizado o *openSSL* solicitará a digitação da *phrase* que irá cifrar esta chave. Não podemos nos esquecer de memorizar a *phase*. Para fins de exemplo digitamos “*testphrase*”. Segue a **Figura 4,** onde mostra a solicitação da *phrase.*

**Figura 4**. Solicitação da *phrase* para cifrar a chave.

Após isso será solicitado para confirmar a *phrase* digitada anteriormente, apenas devemos colocar a mesma *phrase* conforme digitado anteriormente.

Na **Figura 5** podemos observar a criação do arquivo *ca.key* no diretório.

**Figura 5**. Arquivo *ca.key* criado no diretório.

Após isso devemos gerar um certificado *x509* que será válido por 10 anos (3650 dias) que conterá a chave pública do arquivo *ca.key* e será armazenado no arquivo *ca.crt*. Após executarmos o comando será necessário entrarmos com várias informações que farão parte do certificado. Portanto, basta executarmos o comando abaixo:

|  |  |
| --- | --- |
| 1 | req -new -x509 -days 3650 -key ca.key -out ca.crt |

Após executar o comando podemos verificar a solicitação da *phrase* conforme já digitado anteriormente. Segue a **Figura 6** que mostra a solicitação da *phrase.*

**Figura 6**. Digitando o comando para criação do certificado e o *pass phrase*.Se o *pass phrase* for digitado incorretamente teremos um erro apresentado na tela conforme mostra a **Figura 7.**

**Figura 7**. *Pass phrase* digitado incorretamente na geração do certificado.

Digitando corretamente o comando o *OpenSSL* será solicitado para digitarmos o “*Country Name*” que deverá conter duas letras. No nosso caso digitamos *BR* de Brasil conforme mostra a **Figura 8.**

**Figura 8**. Solicitação do *Country Name* na geração do certificado.

Após isso será solicitado o “*State*”. No nosso exemplo digitamos “*Rio Grande do Sul*” conforme mostra a **Figura 9.**

**Figura 9**. Solicitação do *State* na geração do certificado.

Agora será solicitada a “*City*”. No nosso exemplo de certificado colocamos “*Porto Alegre*” conforme mostra a **Figura 10.**

**Figura 10**. Solicitação da *City* na geração do certificado.

Agora será solicitado a “*Organization Name*”. No nosso exemplo colocaremos “*Devmedia Organization*” conforme mostra a **Figura 11.**

**Figura 11**. Solicitação da *Organization* na geração do certificado.

A próxima solicitação é para digitarmos a “*Organizational Unit Name*”. No nosso exemplo colocaremos “*Devmedia Certification*” conforme mostra a **Figura 12.**

**Figura 12**. Solicitação da “*Organizational Unit Name*” na geração do certificado.

Após isso será solicitado o “*Common Name*”. No nosso exemplo colocaremos um nome próprio “*Higor Medeiros*” conforme mostra a **Figura 13.**

**Figura 13**. Solicitação do “*Common Name*” na geração do certificado.

Agora será solicitado o “*Email Address*”. No nosso exemplo colocaremos o e-mail próprio *higor@devmediacertification.com*, conforme mostra a **Figura 14.**

**Figura 14**. Solicitação do “*Email Address*” na geração do certificado.

Podemos verificar a criação do certificado “*ca.crt*” no diretório conforme mostra a **Figura 15.**

**Figura 15**. Geração do certificado *ca.crt* no diretório.

Clicando duas vezes no certificado podemos verificar diversas informações como: para quem e por quem o certificado foi emitido, a validade do certificado, o seu algoritmo de assinatura, sua versão, algoritmo de identificação, chave pública, entre outras informações. As**Figuras 16 a 18** mostram o certificado criado e cada uma das suas abas.

**Figura 16**. Informações da aba “*Geral*” do certificado.

**Figura 17**. Informações da aba “*Detalhes*” do certificado.

**Figura 18**. Informações da aba “*Caminho de Certificação*” do certificado.

Após gerarmos o certificado raiz podemos gerar um certificado de um servidor. Para isso devemos executar o comando abaixo:

|  |  |
| --- | --- |
| 1 | genrsa -des3 -out server.key 4096 |

Este comando gera uma chave *rsa* para o servidor (*server.key*). A **Figura 19**mostra o processamento do comando.

**Figura 19**. Processamento do comando para geração da chave *rsa* para o servidor.

Após isso será solicitado a *pass phrase*, nesse caso digitaremos “*testphrase*” conforme mostra a **Figura 20.**

**Figura 20**. Solicitação do *pass phrase* para *server.keys*.

Após isso devemos confirmar a *pass phrase* digitando a mesma *pass phrase* digitada anteriormente.

Na **Figura 21**podemos observar a criação do arquivo *server.key* no diretório.

**Figura 21**. Arquivo *server.key* criado no diretório.

A seguir faremos uma requisição de certificado (*server.csr*) com o comando abaixo:

|  |  |
| --- | --- |
| 1 | req -new -key server.key -out server.csr |

Após executar o comando acima teremos uma série de solicitações para que possamos cria-lo. Porém primeiramente devemos inserir o *pass phrase* digitado anteriormente conforme mostra a **Figura 22.**

**Figura 22**. Solicitação do *pass phrase* para execução do comando.

Digitando corretamente o comando o *OpenSSL* será solicitado para digitarmos o “*Country Name*” que deverá conter duas letras. No nosso caso digitamos *BR* de Brasil conforme mostra a **Figura 23.**

**Figura 23**. Solicitação do *Country Name* na geração do certificado.

Após isso será solicitado o “*State*”. No nosso exemplo digitamos “*Rio Grande do Sul*” conforme mostra a **Figura 24.**

**Figura 24**. Solicitação do *State* na geração do certificado.

Agora será solicitada a “*City*”. No nosso exemplo de certificado colocamos “*Porto Alegre*” conforme mostra a **Figura 25.**

**Figura 25**. Solicitação da *City* na geração do certificado.

Agora será solicitado a “*Organization Name*”. No nosso exemplo colocaremos “*Devmedia Organization*” conforme mostra a **Figura 26.**

**Figura 26**. Solicitação da *Organization* na geração do certificado.

A próxima solicitação é para digitarmos a “*Organizational Unit Name*”. No nosso exemplo colocaremos “*Devmedia Certification*” conforme mostra **Figura 27.**

**Figura 27**. Solicitação da “*Organizational Unit Name*” na geração do certificado.

Após isso será solicitado o “*Common Name*”. No nosso exemplo colocaremos um nome próprio “*Higor Medeiros*” conforme mostra a **Figura 28.**

**Figura 28**. Solicitação do “*Common Name*” na geração do certificado.

Agora será solicitado o “*Email Address*”. No nosso exemplo colocaremos o e-mail próprio *higor@devmediacertification.com* conforme mostra a **Figura 29.**

**Figura 29**. Solicitação do “*Email Address*” na geração do certificado.

Podemos verificar a criação do certificado “*ca.crt*” no diretório conforme mostra a **Figura 30.**

**Figura 30**. Geração do certificado *server.csr* no diretório.

Feito isso devemos assinar o certificado com o certificado raiz. Para isso devemos utilizar o comando abaixo:

|  |  |
| --- | --- |
| 1 | x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -set\_serial 10102014 -out server.crt |

Após a execução do comando serão apresentadas algumas informações e a solicitação do *pass phrase*novamente, conforme mostra a **Figura 31.**

**Figura 31**. Execução da assinatura do certificado com o certificado raiz.

Este comando assina a requisição *server.csr* com o certificado *ca.crt* e gera um certificado *server.crt*. Uma diferença é que este certificado tem validade de um ano (365 dias) e um número série igual a 10102014.

Após todos esses passos realizados já temos um certificado digital raiz (*ca.crt*) e um certificado para o servidor (*server.crt*) que foi assinado pela nossa própria autoridade certificadora. Para usá-los também precisaremos das chaves *ca.key* e *server.key*.