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

Utilização de certificados para autenticação na Web

Resumo:

- Criação de mini PKI local com dois níveis
- Configuração e gestão de entidades certificadoras
- Configuração de um servidor HTTPS (Apache) para autenticaçao mútua;
- Autenticação de utentes via SSL com certificados.
- Revogação de certificados.

2.1 Introdução

Os certificados podem ser usados para autenticar remotamente os seus titulares, nomeadamente no âmbito das tecnologias Web. Neste guião iremos mostrar como se consegue configurar a interação entre o navegador cliente e o servidor Web de forma a conseguir usar certificados para autenticar servidores web perante os utentes e também para autenticar utentes em interações com servidores Web (autenticação mútua).

Este guia de laboratório serve um duplo objetivo. Por um lado, mostra como se configura um servidor Web seguro, i.e., cujo acesso é feito através de um canal seguro SSL. Tal permitirá que os clientes (navegadores Web) verifiquem a autenticidade do servidor. Por outro lado, considerando-se que alguns conteúdos são restritos a um número limitado de indivíduos, como é que o acesso a esses conteúdos pode ser controlado impondo uma autenticação prévia dos utentes usando certificados. Tal como a autenticação do servidor (perante o navegador), a autenticação dos utentes com certificados (perante o servidor) será feita no âmbito do estabelecimento da sessão segura SSL.

Finalmente, e por uma razão conjuntural, este guia serve ainda um terceiro objetivo: o de mostrar como se pode criar e gerir, de forma simplificada, uma PKI (*Public Key Infrastructure*) simples composta por duas Entidades Certificadoras (EC), uma EC raiz de confiança e uma EC de segundo nível (subordinada) cujo certificado foi emitido pela EC raiz de confiança. Esta EC subordinada será usada para emitir certificados de chave pública para as várias entidades, nomeadamente o certificado para o servidor Web, que será apresentado por este e que terá de ser validado pelos navegadores clientes, e o certificado para o utilizador, que será apresentado por este e validado pelo servidor Web.

2.2 Ambiente de trabalho

Para a realização deste trabalho iremos usar um sistema Linux, que pode ser uma máquina virtual ou uma máquina física, onde irão correr todas as aplicações necessárias para a realização deste trabalho. A razão para utilizarmos apenas uma máquina é a simplificação do trabalho, devido à complexidade que seria termos de lidar com várias máquinas em laboratório. Assim, note que esta configuração não é aconselhável para um ambiente de produção real, nos quais, por razões de segurança, é conveniente instalar as ECs e o servidor Web em máquinas separadas.

2.3 Configuração das Entidades de Certificação

As PKI são vitais para a autenticação de serviços em toda a Internet. Elas são consideradas de confiança e essa confiança é herdada pelos certificados que as respectivas ECs emitem (assinam). Embora muitas PKI tenham uma abragência global e sejam confiáveis à escala mundial, isso não é necessariamente uma exigência em todas as circunstâncias e podem existir PKI confiáveis numa menor escala. Se os utentes dos certificados emitidos, direta ou indiretamente, por uma dada EC local confiarem na sua operação e na correção da sua chave pública, então podem confiar nos certificados em cuja validação a EC surge como raiz de confiança.

Neste guião irá implementar uma PKI de dois níveis, ou seja, com duas ECs: Uma EC raiz de confiança, que iremos designar como EC Raiz, e uma EC emissora de certificados, que iremos designar por EC Subordinada. A EC Raiz destina-se a ser a raiz de confiança para a PKI, pelo que apenas emitirá o certificado para a EC Subordinada, sendo de seguida desligada, por razões de segurança (Idealmente deveria estar num computador dedicado exclusivamente a este fim (EC Raiz), sendo o seu disco guardado num cofre, após o desligar a máquina).

Quanto à EC Subordinada, cujo certificado é assinado pela CA Raiz, destina-se a emitir certificados para os vários utilizadores (pessoas, máquinas ou outros). Idealmente deveria haver alguma segmentação dos utilizadores, para evitar que um eventual comprometimento alastre a toda a organização. Esta segmentação pode ser feita de acordo com vários critérios, desde o tipo de utilizador, distribuição geográfica, etc.

Para a implementação das ECs, usaremos o Software XCA, que pode ser facilmente instalado usando o comando `apt-get` (o pacote chama-se `xca`). Por simplicidade, e como este trabalho tem apenas um objectivo pedagógico, vamos criar ambas as ECs na mesma máquina, algo que na prática **nunca** deve ser feito.

2.3.1 Criação e configuração da EC Raiz

O primeiro passo é iniciar a aplicação XCA e criar uma nova base de dados para a EC Raiz (root CA). Não se esqueça de especificar uma senha (que servirá para proteger dados secretos da EC, nomeadamente a sua chave privada de assinatura de certificados)! Esta base de dados servirá para guardar toda a informação usada na gestão corrente da EC.

Em seguida gere o certificado de raiz (autoassinado), que será a raiz de confiança para as cadeias de certificados desta PKI. Clique na aba `Certificates` e depois no botão `New Certificate`. Na aba `Source` da janela `Create X509`

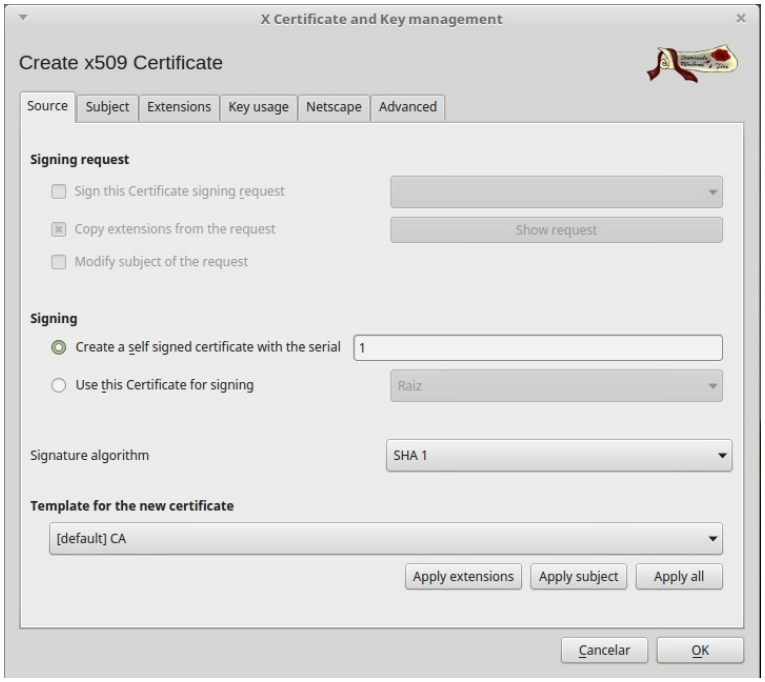

Figura 2.1: Criação do certificado autoassinado para a EC Raiz.

`Certificate` (Figura 2.1) pode introduzir informação sobre como vai o novo certificado ser assinado (self-signed) e qual o tipo de certificado (CA). Verifique se estas duas opções estão seleccionadas, e clique no botão `Apply Extensions`, para aplicar ao novo certificado as extensões caracterśiticas de um certificado de uma EC. De seguida clique na aba `Subject` onde irá fornecer os dados para a identificação da EC. Nesta introduza os dados para identificar a EC raiz, tendo o cuidado de não introduzir espaços nem acentos, e clique no botão `Generate a new key` para abrir a janela para gerar o par de chaves para a EC (Figura 2.2), que convém que seja robusta pelo que deve gerar uma de 4096 bits.

Na aba `Extensions` confirme nas `Basic Restrictions` que está seleccionado o tipo `Certification Authority` e que ão seleccionadas as opções `Critical` e `Subject Key Indentifier`. Na aba `Key Usage` confirme que as opções `Certificate Sign` e `CRL Sign` estão seleccionadas, e na aba `Netscape` confirme que as opções `SSL CA`, `S/MIME CA` e `Object Signing CA` estão seleccionadas. Por fim, clique no botão `OK` para gerar o certificado. Este pode agora ser visto na aba `Certificates` (Figura 2.3).

Explique o que é um certificado auto assinado, por que razão o certificado da CA Raiz é um certificado autoassinado e quais as consequências práticas disso.

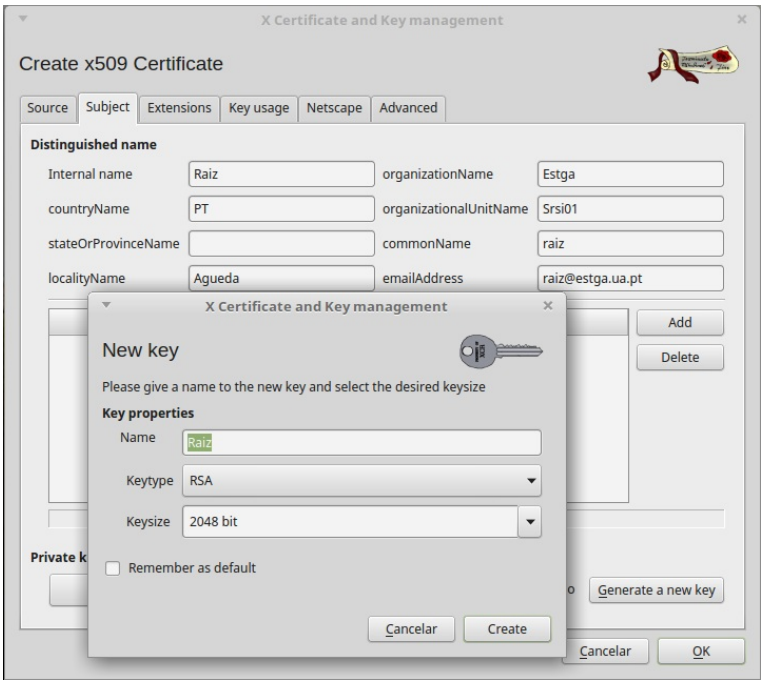

Figura 2.2: Criação do par de chaves para a EC Raiz.

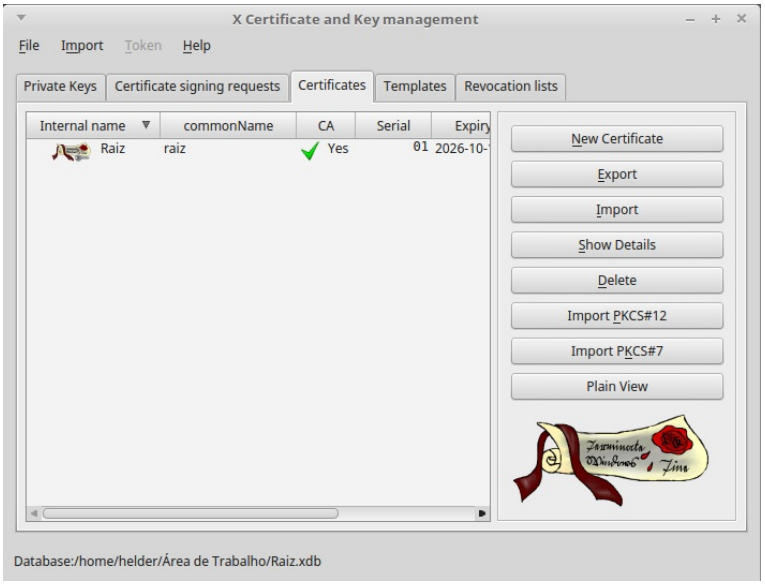

Figura 2.3: Certificado da EC Raiz.

2.3.2 Criação e configuração da Ec Subordinada

Na prática, a EC Subordinada deveria estar numa máquina dedicada, o que, para simplificar, não é o caso neste trabalho. Para a EC Subordinada vamos usar uma nova instância do programa XCA. Abra o programa XCA (abra uma nova instância) e crie uma nova base de dados.

Ao contrário do que sucede com a EC Raiz, o certificado da EC Subordinada não é um certificado autoassinado, mas sim um certificado assinado pela EC Raiz. No entanto, a chave privada da EC Subordinada deve ser da sua exclusiva posse, pelo que deve gerar um par de chaves e fazer um pedido de certificação para a chave pública (pedido para a criação de um certificado para a chave pública).

Porque razão se usam CAs subordinadas? Explique de forma justificada.

Geração do par de chaves e do pedido de certificação da chave pública

Para gerar o par de chaves e o pedido de certificação da chave pública, vá para a aba `Certificate Signing Requests` e clique no botão `New Request`. Na janela `Certificate Signing Request`, tal como fez para a geração do certificado da EC Raiz, introduza os dados para a EC Subordinada, garanta que o pedido é de facto para uma CA e gere o par de chaves. No final clique no botão `Ok` para gerar o pedido de certificação da chave pública, que pode ver na aba `Certificate Signing Requests` (Figura 2.4).

Exporte o pedido de assinatura, botão `Export`, para ficheiro, para poder ser levado para a EC Raiz.

Explique a razão por que o par de chaves e o certificado da EC Subordinada não são logo gerados na EC Raiz, em vez de ter de se levar o pedido de certificação para lá.

Assinatura do certificado da EC Subordinada

Na EC Raiz (instância anterior do programa XCA), seleccione a aba `Certificate Signing Request` e clique no botão `Import` para importar o ficheiro com o pedido de certificação da chave da CA Subordinada. Depois de importada, seleccione o pedido, clique com o botão direito do rato para abrir um menu com as várias operações disponíveis e seleccione a operação `Sign` para assinar o certificado da CA Subordinada (Figura 2.5).

Na janela `Create X509 Certificate` garanta que está seleccionada a chave da EC Raiz para fazer a assinatura, e que as opções `Sign this Certificate Signing Request` e `Copy Extensions from the request` se encontram seleccionadas. Clique no botão `Ok` para criar o novo certificado.

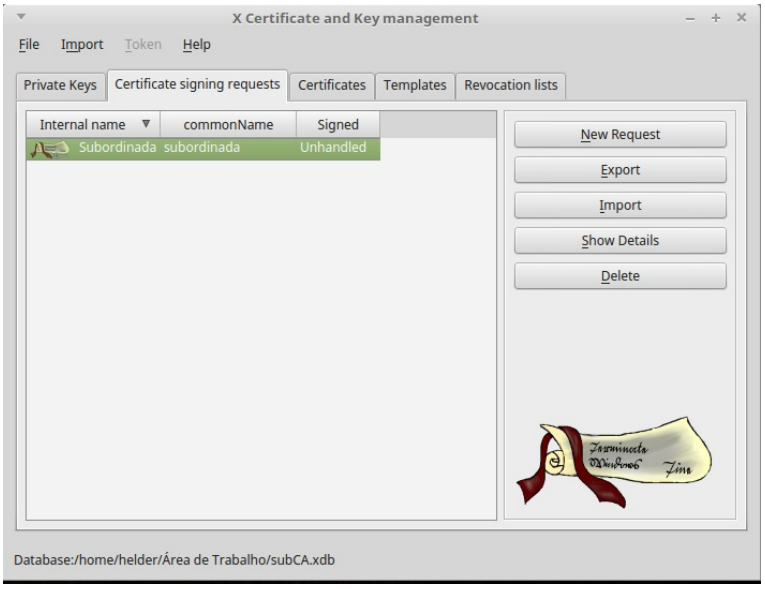

Figura 2.4: Pedido de certificação da chave pública da EC Subordinada.

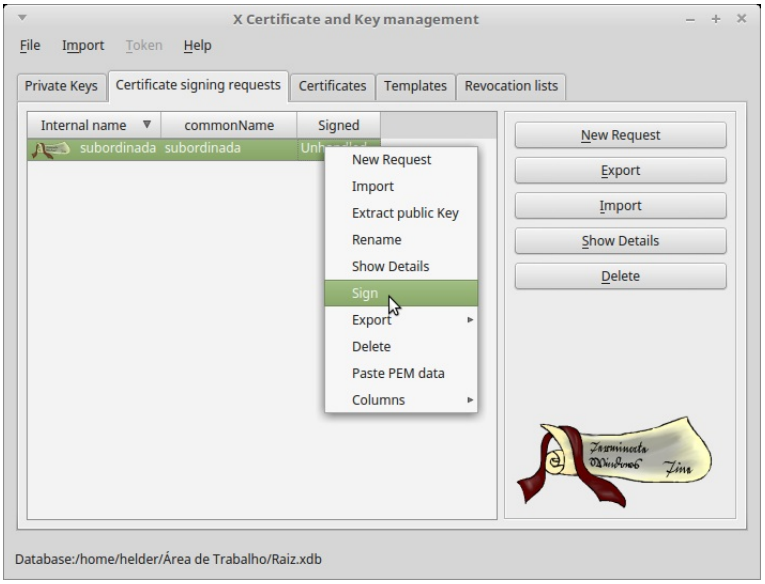

Figura 2.5: Início da criação do certificado da chave pública da EC Subordinada.

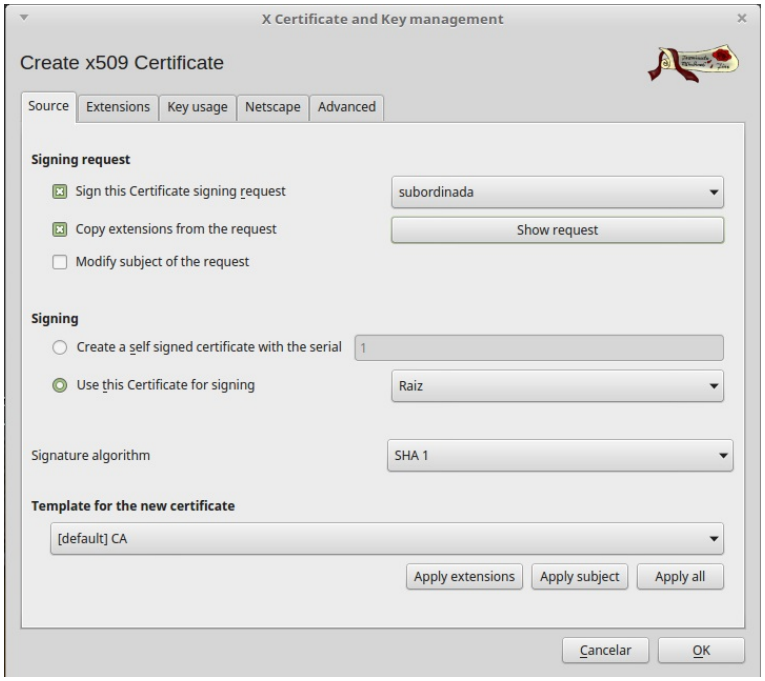

Figura 2.6: Criação do certificado da chave pública da EC Subordinada.

Pode ver o novo certificado na aba `Certificates` (Figura 2.7). Seleccione-o e exporte-o para ficheiro, clicando no botão `Export`, e usando o formato `PKCS#7 with certificate chain`. A EC Raiz já fez o seu trabalho. Agora deve ser desligada e a sua chave privada guardada em local seguro. Sai da instância do XCA correspondente à EC Raiz.

Por que razão se deve desligar e proteger a EC Raiz?

Importação do certificado da EC Subordinada

Agora, na EC Subordinada vamos importar o novo certificado. Para isso, vá para a instância do programa XCA correspondente à EC Subordinada e importe o novo certificado, clicando no botão `Import PKCS#7`. Deve ver dois certificados (o da EC Raiz e o da EC Subordinada), a cadeia de certificados completa (Figura 2.8). Importe os dois, após o que os pode ver na aba `Certificates`. No entanto, eles ainda não são de confiança.

Explique por que é que os certifcados que importou não são de confiança?

Para os tornar de confiança, clique no certificado da EC Raiz com o botão do lado direito do rato, seleccione `Trust` e indique `Always trust this`

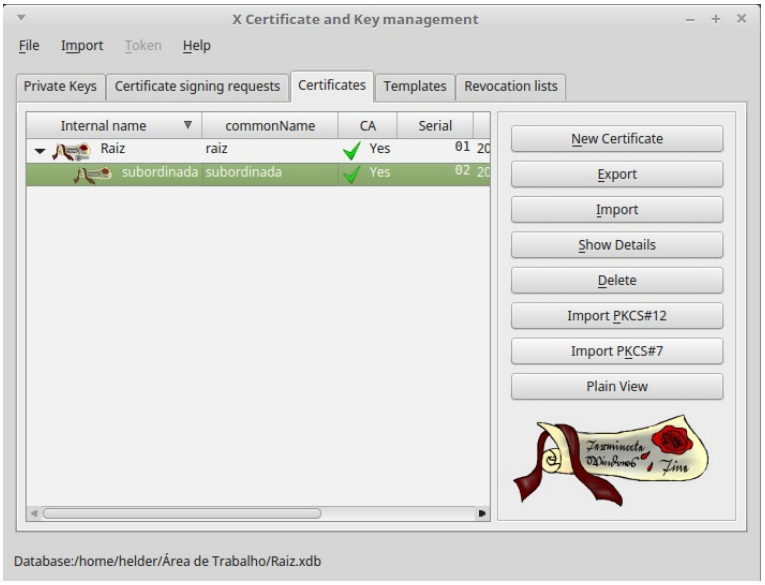

Figura 2.7: Certificado da chave pública da EC Subordinada.

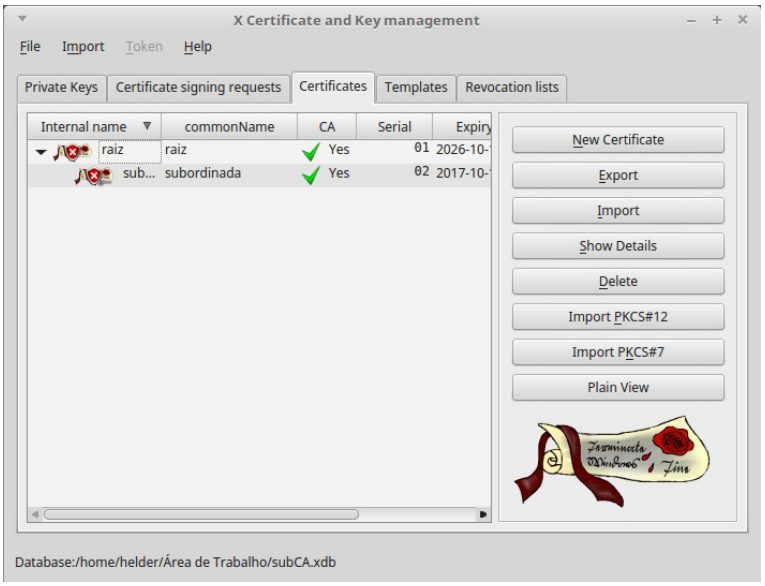

Figura 2.8: Importação do certificado da chave pública da EC Subordinada.

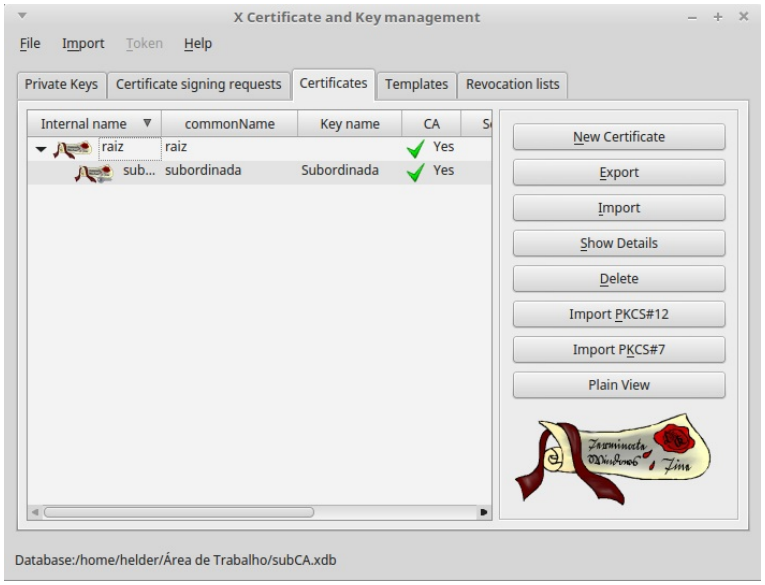

Figura 2.9: Cadeia de certificação da EC Subordinada.

`certificate`. Faça o mesmo para o certificado da EC Subordinada, com a diferença de que deve seleccionar `Only trust this certificate, if we trust the signer`. Neste momento, a EC Subordinada já está configurada com o seu certificado e a respectiva cadeia de certificação (Figura 2.9), pelo que já pode ela própria emitir certificados.

2.4 Instalação do servidor HTTPS

Um servidor HTTPS é um servidor Web (HTTP) cuja comunicação com o cliente é feita um cana seguro SSL. Um dos serviços de segurança que o SSL fornece é a autenticação do servidor, que implica que o servidor tenha um par de chaves assiméticas com a respectiva chave pública certificada (certificado X.509v3). Assim, nesta secção vamos gerar um par de chaves para o servidor e emitir um certificado para a sua chave pública usando a EC Subordinada que atrás criou.

Os servidores HTTPS estão normalmente localizados noutras máquinas que não aquela onde se localiza a EC que lhe fornece o certificado para a comunicação segura. No entanto, dado o carácter exclusivamente pedagógico deste guião, e para facilitar as operações, iremos instalá-lo na mesma máquina. Para ele será, então, necessário gerar um par de chaves e um certificado para a sua chave pública.

O servidor HTTPS que vamos usar é um servidor Apache, mais propria-

mente o `apache2`, que deve ser instalado com o comando `apt-get`.

2.4.1 Geração do par de chaves do servidor e do pedido de certificação para a sua chave pública

Para a geração do par de chaves para o servidor HTTPS, vamos usar o OpenSSL, disponível através da linha de comandos. Use o seguinte comando para gerar um par de chaves para o algoritmo RSA de 2048 bits, cifrar a chave privada usando o algoritmo Triple-DES e guardar o par de chaves no ficheiro `server.key` na sua pasta corrente:

```
openssl genrsa -des3 -out server.key 2048
```

Use agora o seguinte comando para criar um pedido de certificado de chave pública (CSR) para o servidor, que ficará guardado no ficheiro `server.csr` na sua pasta corrente:

```
openssl req -new -key server.key -out server.csr
```

Durante a execução do comando se-lhe-á pedida informação para a identificação do servidor HTTPS, que deve fornecer. Tenha especial atenção quando lhe for pedido o `Common Name` porque este deverá ser igual ao nome DNS do servidor HTTPS (para este trabalho use www.srsiX.pt, em que o X é o número do seu grupo)

O ficheiro server.csr contém o pedido de certificação e deve ser enviado para a EC da qual se pretende o certificado, no nosso caso para a EC Subordinada que está na mesma máquina.

2.4.2 Geração do certificado para o servidor HTTPS

Usando o XCA com a base de dados da EC subordinada, importe o pedido de certificação. Em seguida dê início ao processo para a sua assinatura. Garanta que o certificado é assinado usando a chave da CA Subordinada, que contém todas as extensões características de um servidor HTTPS e que define o seguinte URI para a localização onde irão ser colocadas as CRL (Listas de Certificados Revogados):

```
URI:http://www.ecsubordinada.pt/crl/subordinada.crl
```

As listas de certificados revogados serão abordadas mais à frente.

Exporte o certificado usando o formato PEM `PEM with certificate chain`, tendo o cuidado de não proteger o seu conteúdo com senha, para que o Apache o posso ler..

2.4.3 Instalação do certificado e chave privada no servidor

Para configurar o servidor HTTPS, ativamos o seu módulo SSL através do comando

```
a2enmod ssl
```

A ativação deste módulo cria automaticamente um par de chaves para o servidor e um certificado autoassinado da respetiva chave pública; porém, não vamos usar estas credenciais.

Na diretoria `/etc/apache2/sites-available` está um ficheiro `default-ssl.conf`, que contém a configuração por omissão do SSL usada pelo servidor. Copie este ficheiro para `/etc/apache2/sites-enabled` e edite-o de forma a considerar as credenciais de autenticação anteriormente criadas. Em particular, altere as seguintes variáveis de configuração:

SSLCertificateKeyFile: referência ao ficheiro PEM com a chave privada do servidor.

SSLCertificateFile: referência ao ficheiro PEM com o certificado do servidor.

SSLCertificateChainFile: referência ao ficheiro PEM com o certificado do servidor.

Feitas estas alterações, reinicie o servidor através do comando

```
service apache2 restart
```

2.5 Importação do certificado raiz pelos navegadores

Verifique que o navegador local consegue aceder ao servidor Web usando o URL `https://`*`servername`*, em que *servername* é o nome que colocou no campo `CommonName` no certificado que gerou para o servidor. Verificará que não consegue aceder ao servidor. Porquê?

Edite o ficheiro /etc/hosts e adidicone uma linha com o endereço IP localhost (127.0.0.1) (é na sua máquina que o servidor está a correr) e o nome que deu ao servidor no campo `CommonName` do certificado. Tente novamente aceder ao servidor.

Verificará que consegue aceder (no sentido em que ele existe e está contactável), mas obtém um erro do SSL porque o seu navegador não consegue

validar o certificado apresentado pelo servidor. Analise, no seu navegador, o certificado que o servidor apresentou e a respectiva cadeia de certificação. Qual a razão para o erro que acontece?

O problema poderá ser resolvido através da confiança depositada pelos navegadores clientes na EC que emitiu o certificado do servidor. Nesse sentido, os navegadores terão que obter e importar, para o seu repositório de certificados raiz (confiáveis), o certificado da EC Raiz.

Voltando à EC Subordinada (XCA), exporte o certificado da EC Raiz para um ficheiro. Importe esse ficheiro para os certificados raiz do navegador que usou anteriormente e verifique que já conseguirá aceder ao servidor Web que criou sem problemas.

Experimente agora aceder ao servidor usando um endereço tecnicamente equivalente ao anterior: `https://127.0.0.1`. poderá constatar que obtém novamente o erro que acontecia quando não tinha a EC nos repositórios do navegador. Explique a origem do problema.

2.6 Análise do tráfego

Arranque com um analizador de tráfego (e.g. o Wireshark) e inicie uma captura de tráfego sobre a sua interface de rede para a Internet. Com o seu *browser* aceda ao site da Câmara Municipal de Aveiro (`http://www.cm-aveiro.pt`).

Termine a captura e analise o tráfego capturado quanto à confidencialidade, integridade e autenticidade. Indique quais as suas conclusões.

Inicie uma nova captura, agora sobre a interface de rede com que acede ao seu site na Web. Inicie o seu *browser* e aceda ao seu servidor Web usando o URL https://*servername*, em que servername é o nome que deu ao servidor.

Termine a captura e analise o tráfego capturado. O que conclui?

No tráfego capturado identifique os pacotes onde o servidor envia certificados. Analise os certificados enviados. Indique que certificados são enviados e explique porque razão o servidor não pode enviar apenas o seu certificado.

2.7 Autenticação de utilizadores Web com certificados

O próximo passo consiste na autenticação dos clientes Web, e, neste caso particular, autenticar o utente do navegador cliente por meio certificados digitais de chave pública.

2.7.1 Geração do certificado para o utilizador

Em primeiro lugar é necessário gerar um par de chaves assimétricas e o respectivo certificado de chave pública para o utilizador. Para simplificar, use a `EC Subordinada` que instalou atrás instalou para gerar o par de chaves e o respectivo certificado digital de chave pública para o utente. Note no entanto, que numa situação real possivelmente a chave privada deveria ser gerada pelo próprio utilizador.

O `XCA` não tem nenhum perfil para certificados de utilizador, por isso tenha o cuidado de, na aba `Source`, não clicar em nenhum dos três botões para aplicar as definições de perfil. Tenha o cuidado de indicar que a chave para assinar o certificado é a chave da `EC Subordinada`.

Na aba `Subject` introduza os dados que identificam o utilizador, à semelhança do que fez para o certificado do servidor. Use o seu nome para o campo `CommonName`. Não se esqueça de gerar a chave privada.

Na aba `Extensions` escolha o tipo `EndEntity` nas `Basic constraints` e seleccione as opções `Subject Key Identifier` e `Authority Key Identifier`.

Na aba `Key usage` seleccione `TLS Web Client Authentication` na lista `Extended key usage` (à direita), e não seleccione nada na lista `Key usage` (à esquerda).

Por fim, na aba `Netscape` seleccione `SSLClient`

Depois de termianda a configuração, pode gerar o certificado e exportá-lo. Tenha atenção que tem também de exportar a correspondente chave privada pelo que deve usar o formato `PKCS#12 with Certificate chain`.

2.7.2 Importação do certificado no navegador

Deve agora, no seu navegador, importar o certifica atrás gerado e a correspondente chave privada. No `Firefox` isso faz-se na janela do `Gestor de Certificados`, à qual pode aceder clicando no botão `Ver Certificados` das `Opções Avançadas` das `Preferências`.

2.7.3 Configuração do servidor web

Para terminar, é preciso indicar na configuração do servidor HTTPS que (i) deverá autenticar os clientes e (ii) como deve orientar orientar os clientes na escolha das suas credenciais. Para indicar ao Apache que este deve autenticar o cliente através de certificados, deverá editar o ficheiro `default-ssl.conf` antes editado, procurar a secção `VirtualHost *:443` (correspondente às definições de serviços que usam SSL) e no final da secção adicionar uma entrada com o seguinte conteúdo.

```
<Location /secure>
    SSLVerifyClient require
    SSLVerifyDepth 10
    SSLOptions +OptRenegotiate +StdEnvVars +ExportCertData
</Location>
```

Além disso, para que ele diga ao cliente que certificados podem ser usados, é necessário que a variável `SSLCACertificateFile` contenha o nome completo do ficheiro com o certificado da EC Raiz.

Uma vez alterado este ficheiro, crie a diretoria `/var/www/html/secure` e coloque na mesma um ficheiro `index.php` com uma mensagem de sucesso.

Para interpretar PHP precisa de instalar o respetivo módulo, o que pode fazer usando `apt-get` e o pacote `libapache2-mod-php5`. Reinicie o servidor como fez anteriormente.

Tente agora aceder com o navegador ao URL `http://`*`<servername>`*`/secure`, em que *servername* é o nome DNS do seu servidor. Verá que aparece uma janela com a informação do seu certificado para que confirme se é aquele o certificado que quer usar para se autenticar. Caso possua mais do que um certificado é-lhe dada a possibilidade de ver os dados dos certificados, para que possa seleccionar o certificado que pretende usar. Confirme que pretende usar o certificado e verá aparecer o conteúdo que colocou na página `index.php`.

Note que o servidor foi configurado para, no acesso a este URL, requerer uma autenticação do cliente com um certificado cuja raiz de confiança seja a EC Raiz. Use o `Wireshark` para observar o diálogo entre o navegador e o servidor e constatar a lista de certificados enviada pelo servidor para guiar a autenticação do cliente. Explique como a observou.

2.7.4 Identificação do cliente pelo servidor

Um aspecto importante da autenticação de clientes Web por meio de certificados é o de identificar claramente o indivíduo na camada aplicacional (por exemplo, aplicação PHP). Para isso, edite o ficheiro `index.php` que criou anteriormente e adicione o seguinte conteúdo:

```
<html>
<head>
    <meta charset="UTF-8">
</head>
<body>
    <pre>
    <?php print_r($_SERVER); ?>
```

```
    </pre>
</body>
</html>
```

Se aceder novamente a página `https://localhost/secure`, poderá obsarver os dados do utente que estão à disposição do servidor Web através do seu certificado de autenticação, usado na prova de identidade do utente cliente.

Compare a autenticação através de certificados com a autenticação através de login e *password* e identifique as vantagens e desvantagens de cada um deles.

2.8 Revogação de certificados

Como sabe, os certificados podem ficar inválidos antes de expirar. Nestas circunstância é necessário informar essa ocorrência às entidades que validam certificados. Para esse efeito existem as CRL (*Certificate Revocation Lists*) e o OCSP (*Online Certificate Status Protocol*). Vamos agora configurar o Apache para verificar numa CRL se o certificado do utilizador está revogado.

Primeiro é necessário gerar uma lista de certificados revogados. Para isso, dirija-se à EC Subordinada, invalide o certificado do utilizador que usou na secção anterior e gere uma lista de certificados revogados. Explique como o fez e mostre evidências disso com uma captura de ecrã.

Exporte a CRL em formato PEM.

De seguida, no ficheiro de configuração do SSL do Apache, `default-ssl.conf`, altere a linha com a directiva `SSLCARevocationFile` para referênciar o ficheiro com a CRL que atrás exportou. Na linha anterior, adicione a seguinte directiva `SSLCARevocationCheck leaf`, tal como ilustrado no seguinte excerto, em que fullpathtoCRL.pem é o caminho completo para o ficheiro com a CRL:

```
SSLCARevocationCheck leaf
    SSLCARevocationFile fullpathtoCRL.pem
```

Explique o objectivo da directiva `SSLCARevocationCheck leaf`, indicando quais os valores possíveis para ela e respectivas funcionalidades que activam.

Experimente agora aceder novamente à página `secure` do servidor Web.

Indique algumas situações da vida real onde ache que pode ser útil a verificação da revogação de certificados.

Que outros mecanismos de revogação de certificados conhece e como funcionam?

2.9 Bibliografia

- Apache2 ModSSL, `http://httpd.apache.org/docs/2.2/mod/mod_ssl.html`
- XCA, `http://sourceforge.net/projects/xca`