Segurança Informática e nas Organizações

Guiões das Aulas Práticas

João Paulo Barraca[1] e Hélder Gomes[2]

[1]Departamento de Eletrónica, Telecomunicações e Informática
[2]Escola Superior de Tecnologia e Gestão de Águeda
Universidade de Aveiro

2015–2016

Conteúdo

2

Canais seguros SSL: autenticação mútua com certificados X.509

Resumo:

- Criação de uma Entidade Certificadora;
- Criação de um servidor HTTPS;
- Autenticação de utentes via SSL com *smartcards*.
- Exploração do Cartão de Cidadão em autenticações de utentes em sessões SSL.

2.1 Introdução

Os *smartcards* podem ser usados para autenticar remotamente os seus titulares, nomeadamente no âmbito das tecnologias Web. Neste caso, o *smart card* é apresentado pelo utente ao navegador, e não ao servidor. Neste guião iremos mostrar como se consegue configurar a interação entre o navegador cliente e o servidor Web de forma a conseguir usar *smartcards*, e em particular Cartões de Cidadão, para autenticar utentes em interações com servidores Web.

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 do utentes com um *smart card*. Tal como a autenticação do servidor (perante o navegador), a autenticação dos utentes com o seu *smart card* (perante o servidor) será feita no âmbito do estabelecimento da sessão segura SSL. Como *smart card* vamos usar os Cartões de Cidadão.

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 Entidade Certificadora (EC) instalada numa máquina Linux. Esta EC será usada para emitir certificados de chave pública, nomeadamente o certificado apresentado pelo servidor Web e que terá de ser verificado pelos navegadores clientes.

2.2 Ambiente de trabalho

Para simplificar, nesta aula iremos usar apenas uma máquina virtual Linux, ou um sistema nativo Linux caso os alunos o tenham. Tanto o servidor Web como o navegador que acede ao mesmo são executados na mesma máquina. A esta máquina deverá ser igualmente ligado um leitor de *smart card*.

2.3 Criação de uma EC

As EC são vitais para a autenticação de serviços em toda a Internet. Elas são consideradas de confiança e essa confiança é herdada pelos certificados que emitem (assinam). Embora muitas ECs tenham uma abragência global e sejam confiáveis à escala mundial, isso não é necessariamente uma exigência em todas as circunstâncias e podem existir ECs 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 surge como raiz de confiança.

Nesta etapa vamos criar uma EC local, a fim de gerar certificados para um servidor Web pessoal. Para isso, usaremos o Software XCA, pode ser facilmente instalado usando o comando `apt-get` (o pacote chama-se `xca`).

O primeiro passo é iniciar a aplicação XCA e criar uma nova base de dados. 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 um novo par de chaves assimétricas para a EC, que será designado neste guião como `CAKey`. Normalmente, as chaves das ECs são consideravelmente mais robustas do que as de certificados de servidor, pelo que deverá usar um número elevado de bits (4096 bits se possível).

Em seguida, crie o certificado de chave pública raiz da EC selecionando *Certificates* → *certificate*. Não se esqueça de selecionar a chave que você acabou de criar, assim como o modelo padrão de CA (*CA template*, deve selecioná-lo e, em seguida, escolher *Apply*). Além disso, deve preencher os dados de identidade da EC (aba *Subject*) e definir o prazo de validade do certificado (aba *Extensions*, escolha alguns anos).

Depois de ter gerado o certificado raiz da EC, podemos gerar novos certificados assinando pedidos de emissão dos mesmos, e assinar estes pedidos com a chave privada da EC que acabou de criar.

2.4 Emissão do certificado de um servidor HTTPS

Os servidores HTTPS são servidores HTTP que interagem com os clientes sobre um canal seguro SSL. Tipicamente esses servidores têm de se autenticar usando um par de chaves assimétricas e uma chave pública certificada (certificado X.509v3). Assim, nesta secção vamos gerar um certificado para um servidor HTTPS emitido pela EC criada na secção anterior.

Vá até a aba *Certificate signing requests* e selecione *New Request*. Neste caso, escolha o modelo *HTTPS_server* para o novo certificado. Na aba *Subject* preencha os dados de identificação do servidor, tendo o cuidado de escolher `localhost` como nome comum (*Common name*). Nessa mesma aba indique que pretende usar um novo par de chaves assimétricas nesse

certificado (selecione *Generate a new key*).

Estando o pedido criado, o certificado pode ser finalmente gerado assinando o pedido com a chave privada da CA (`CAKey`). Selecione o pedido feito com o botão direito do rato e escolha selecione *sign*.

2.4.1 Exportação da chave e certificado do servidor HTTPS

A etapa final consiste as credenciais do servidor anteriormente geradas: a sua chave privada e o ser certificado de chave pública. Exporte ambos como PEM.

Exporte o certificado de chave pública para um ficheiro e copie-o depois para a diretoria `/etc/ssl/certs` (ou exporte-o diretamente para esta diretoria se estiver a executar o XCA como `root`). Esporte também a chave privada para um ficheiro, sem proteger o seu conteúdo com uma senha, e copie-o posteriormente para a diretoria `/etc/ssl/private`.

2.5 Instalação do servidor HTTPS

Para o servidor HTTPS vamos usar `apache2`, que deve ser instalado com o comando `apt-get`. Em seguida 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, estas credenciais não vão ser usadas.

Na diretoria `/etc/apache2/sites-available` está um ficheiro `default-ssl`, 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:

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

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

Feitas estas alterações, reinicie o servidor através do comando

```
service apache2 restart
```

Verifique que o navegador local consegue aceder ao servidor usando o URL `https://localhost`. verificará que consegue aceder (no sentido em que ele existe e está contactável), mas obtém um erro do SSL porque não consegue validar o seu certificado.

2.5.1 Importação do certificado raiz pelos navegadores

O problema indicado acima deverá 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 EC raiz (confiáveis), o certificado da EC antes criada.

Voltando ao XCA, exporte o certificado da EC que criou 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 aceder ao servidor usando um endereço tecnicamente equivalente ao anterior: `https://127.0.0.1`. poderá constatar que obtém novamente o erro anterior, quando não tinha a EC nos repositórios do navegador. Explique a origem do problema.

2.5.2 Autenticação de utentes Web com o Cartão de Cidadão

O próximo passo consiste na autenticação dos clientes Web, e, neste caso particular, identificar o utente do navegador cliente por meio do Cartão de Cidadão.

O primeiro passo passa por descarregar todos os certificados que podem ser usados em cadeias de certificação das chaves de autenticação de Cartões de Cidadão e instalar os mesmos numa diretoria temporária (e.g. `/tmp/PTEID`). Como o servidor HTTPS usado carrega certificados no formato PEM, e os certificados em causa são disponibilizados num outro formato (DER ou CER), poderá ser preciso convertê-los, o que pode ser feito com o seguinte comando:

```
openssl x509 -in infile.cer -out outfile.pem -inform DER -outform PEM
```

Os certificados que deverá colocar na diretoria acima referida deverão ser:

- Se tem o `middleware` do Cartão de Cidadão instalado, todos os que estão na diretoria `usr/local/bin/eidstore/certs`.

- Todos os certificados das ECs emissoras de certificados de autenticação para Cartões de Cidadão. Os seus certificados pode ser obtidos através de um URL

  https://pki.cartaodecidadao.pt/publico/certificado/cc_ec_cidadao_autenticacao/EC de Autenticacao do Cartao de Cidadao XXXX.cer

  onde a parcela `XXXX` deverá ter os valores `0001`, `0002`, etc., até ao da EC do ano corrente (`0008` em 2013).

Uma vez obtidos todos os certificados, crie na diretoria `/etc/ssl/certs/` uma subdiretoria (e.g. `PTEID`) um único ficheiro (e.g. `Auth.PEM`) contendo todos os certificados que podem ser necessário para autenticar os clientes SSL:

```
cat /tmp/PTEID/*.pem > /tmp/Auth.pem
cp /tmp/Auth.pem /ssl/certs/PTEID
```

Para terminar, é preciso indicar na configuração do servidor HTTPS que (i) deverá autenticar os clientes e (ii) deverá usar os certificados anteriormente agrupados para orientar os clientes da escolha das suas credenciais. Para isso, deverá editar o ficheiro `default-ssl` antes editado, procurar a secção `VirtualHost *:443` (correspondente às definições de serviços que usem SSL) e criar uma entrada nova com o seguinte conteúdo:

```
<Location /secure>
    SSLVerifyClient require
    SSLVerifyDepth 10
    SSLOptions +OptRenegotiate +StdEnvVars +ExportCertData
    SSLCACertificateFile /etc/ssl/certs/PTEID/Auth.pem
</Location>
```

Uma vez alterado este ficheiro, crie a diretoria `/var/www/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://localhost/secure`. Sem uma configuração apropriada nem acesso a um Cartão de Cidadão, o navegador não consegue autenticar-se como pedido pelo servidor e o estabelecimento da sessão SSL falha. Note que o servidor foi configurado para, no

acesso a este URL, requer uma autenticação do cliente com as credenciais de autenticação de um Cartão de Cidadão.

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.

2.6 Configuração do navegador

Para que o navegador consiga usar corretamente um Cartão de Cidadão para autenticar o seu utente, terá de ser configurado para usar a biblioteca PKCS#11 do Cartão de Cidadão. Tal depende de navegador para navegador; no `Firefox`, entre em *Preferences*, escolha a aba *Advanced*, nesta a aba *Encryption* e selecione *Security Devices*. Usando o botão *Load*, adicione um novo módulo PKCS#11, usando para o efeito o ficheiro `/usr/local/lib/libpteidpkcs11.so`.

Nota: se no passo anterior der erro na instalação do módulo, tal pode dever-se a uma ausência de um serviço que é fundamental para aceder a *smartcards* (PCSCD). Para resolver esse problema, ligue um leitor de *smartcards* e execute o seguinte comando

```
service pcscd restart
```

Seguidamente já deverá conseguir carregar o módulo.

Uma vez feita esta configuração, sempre que esteja disponível um leitor de *smartcards* e o mesmo tenha um Cartão de Cidadão inserido, o navegador terá acesso aos certificados de chave pública e (indiretamente) às chaves privadas disponibilizados pelo Cartão de Cidadão, e pode usá-los sempre que adequadamente solicitado por um servidor.

Neste momento, e caso se use um Cartão de Cidadão, já se consegue aceder ao URL `https://localhost/secure` após uma introdução correta do seu PIN de autenticação.

Use uma vez mais o `Wireshark` para observar o diálogo entre o navegador e o servidor e constatar a resposta enviada pelo cliente à solicitação de autenticação do servidor. Reinicie o navegador para este efeito.

2.7 Identificação no cliente pelo servidor

Um aspecto importante da autenticação de clientes Web por meio de *smartcards* é o de identificar claramente o indivíduo (Cidadão Português) 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.

2.8 Bibliografia

- Cartão de Cidadão, `http://www.cartaodecidadao.pt`
- Apache2 ModSSL, `http://httpd.apache.org/docs/2.2/mod/mod_ssl.html`
- XCA, `http://sourceforge.net/projects/xca`