Relatório

Segurança 2014/2015

Grupo 7 - Prática 4

Ivo Silva, 64833

Rui Filipe Pedro, 65828

Índice

1. Introdução

Nesta segunda parte deste projecto, o objectivo é adicionar autenticação ao sistema criado. A autenticação pode ser feita de duas formas: usando o Cartão de Cidadão ou usando uma password. Para este efeito foram feitas alterações ao sistema existente, tal como a introdução de novos conceitos no projecto.

2. Registo do utilizador

No registo é necessário que seja atribuído um nome de utilizador, uma password e ter o CC introduzido.

Esta password introduzida é utilizada como password para a chave privada do utilizador mas também para a autenticação do utilizador no servidor (no caso de autenticação por username e password).

É gerado um salt aleatório e é feito um hash do mesmo com a password (PBKDF2). Tanto este hash como o salt gerado são enviados para a base de dados do servidor. Desta maneira quando se pretender autenticar um utilizador através de password é necessário pedir o salt ao servidor e proceder ao hash da password com esse mesmo salt e depois enviar esse hash para o servidor comparar com a hash que está na base de dados. Para garantir que a hash enviada pelo cliente é a que chega ao servidor, o cliente cifra a hash com a chave pública de RSA do servidor antes de a enviar.

No registo é também retirado do CC o expoente e o módulo do “CITIZEN AUTHENTICATION CERTIFICATE”, ou seja, do certificado de chave pública que irá permitir validar uma assinatura feita com a chave privada de autenticação do CC.

No caso de autenticação através do CC irá ser necessária a introdução da password apenas para o utilizador executar operações que necessitem da chave RSA privada do utilizador (a chave privada da PBOX e não a do CC). A password irá apenas ser usada para fins de encriptação e desencriptação de ficheiros e não para fins de autenticação. Essa autenticação é feita só pelo cartão de cidadão.

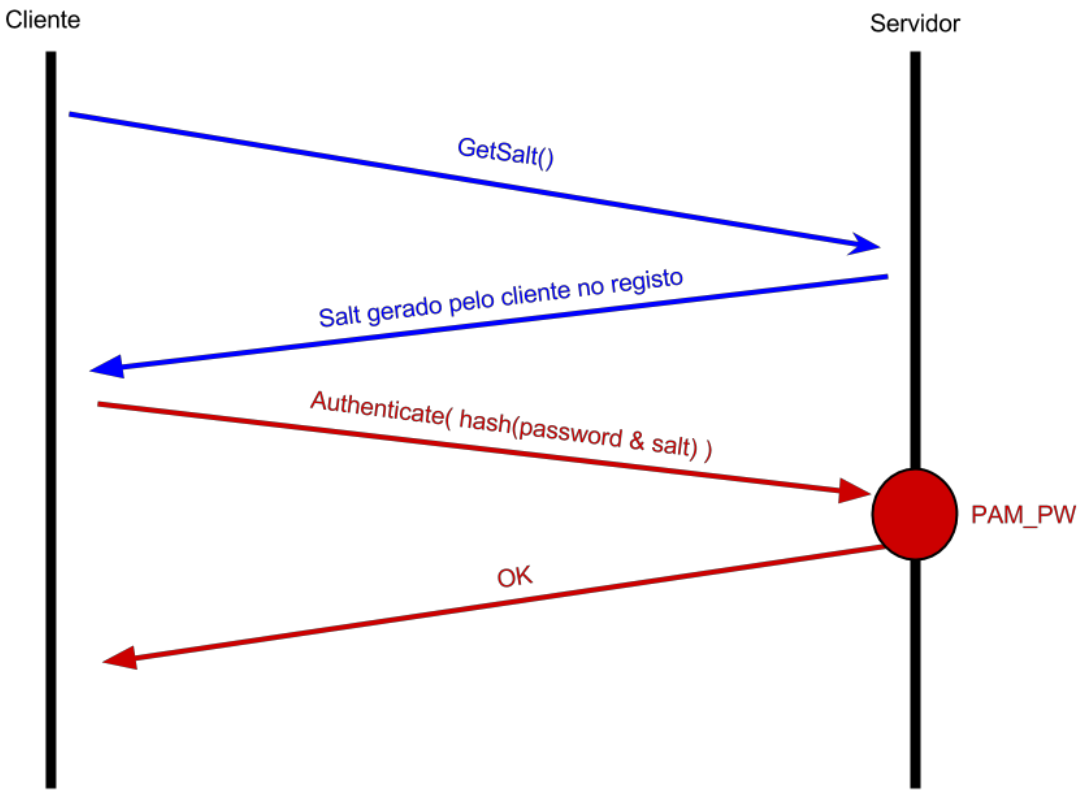
3. Autenticação

A autenticação pode ser feita através de username e password ou através do cartão de cidadão. Ambos os casos vão ser tratados através de um módulo PAM. O PAM referente ao sistema está configurado da seguinte maneira:

```
auth    sufficient      pam_pw.so
auth    sufficient      pam_PTEIDCC.so
```

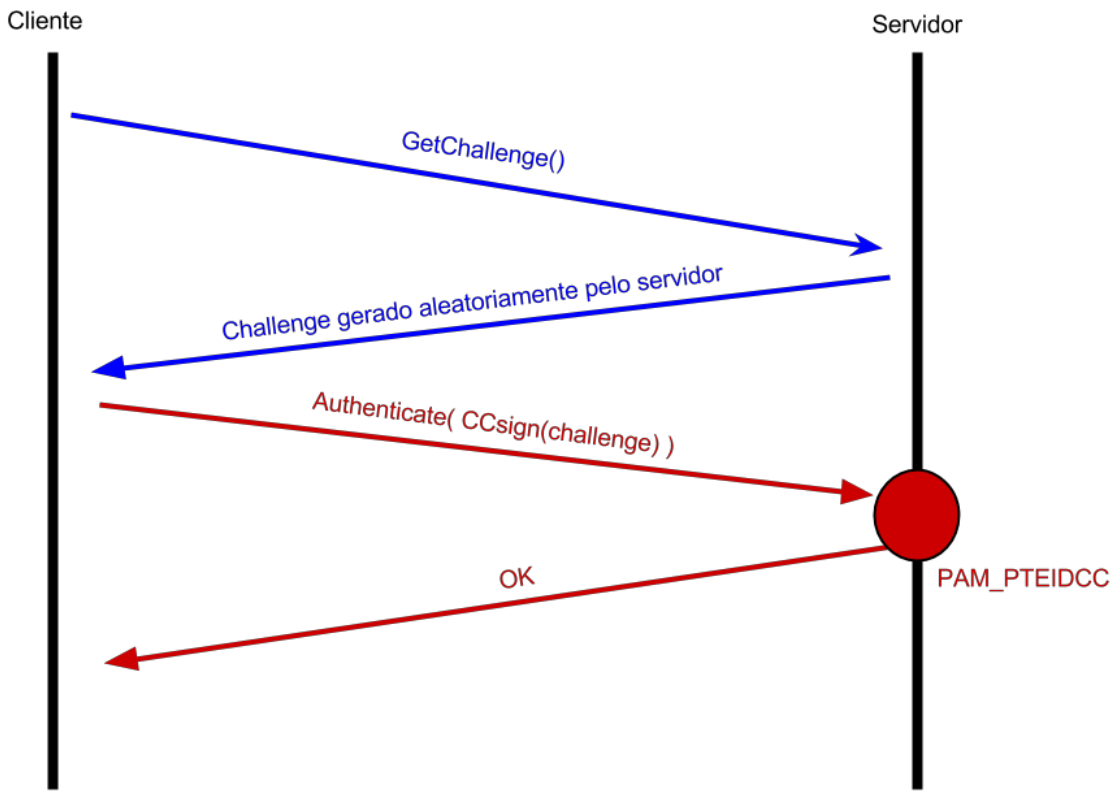
Ou seja, primeiro há uma tentativa de autenticar através da password. Se esta tiver sucesso o utilizador está autenticado. Pelo contrário, se não for possível a autenticação através de password, a mesma é feita através do cartão de cidadão.

PAM_PW.so



No módulo PAM responsável pela autenticação por password é comparada a hash recebida do cliente com a hash armazenada na base de dados no registo. Se estas forem iguais, o utilizador será autenticado com sucesso.

PAM_PTEIDCC.so



Para a autenticação através do cartão de cidadão foi usado o método de challenge-response.

Quando o cliente indica que quer autenticar-se, é gerado um challenge de 64 bytes. Devido a alguns problemas com a funcionalidade de assinatura digital do CC no que toca a strings dessa dimensão, decidimos fazer um digest (SHA-1) desse challenge e só depois o enviar para o cliente. Quando o cliente recebe o digest(challenge), é feita a assinatura do mesmo, com a 'CITIZEN AUTHENTICATION KEY' do CC, e enviada para o servidor.

O módulo PAM detém, nesse momento, todos os dados que necessita para fazer a verificação da assinatura. Esses dados são o challenge (o digest do challenge, neste caso), a

assinatura feita pelo cliente, o expoente e o módulo da chave pública de autenticação do CC do cliente.

A verificação da assinatura traduz-se então em:

verify(digest(digest(challenge)), signature(digest(challenge)), RSA key(e, m))

Se esta verificação for efetuada com sucesso, então o utilizador considera-se autenticado.

4. Validação da cadeia de certificação e listas de revogação de certificados

Para a validação da cadeia de certificação foi criado um módulo chamado wrapper. Este módulo tem como objectivo fazer com que as operações com os certificados do Cartão de Cidadão sejam transparentes do ponto de vista do servidor e do cliente, permitindo também exportar e importar os certificados para poder transferi-los do cliente para o servidor.

A validação da cadeia de certificação é feita no servidor. Quando um utilizador tenta autenticar-se perante o servidor, os seus certificados são lidos do Cartão de Cidadão e é enviado para o servidor a cadeia de certificados. Aqui a cadeia de certificação é validada usando o wrapper.

Depois da validação da cadeia, o wrapper é usado para obter as listas de revogação de certificados referentes a cada certificado. De seguida é feito o download desses documentos e neles são procurados os serials dos certificados do utilizador. Se algum serial for encontrado numa lista de revogação então o utilizador não é autenticado pelo servidor.

Tanto a validação da cadeia de certificação como a validação dos certificados perante as listas de revogação de certificados é apenas feita uma vez por dia. Para isso, a tabela User da base de dados possui um campo, denominado de, last_date, que guarda a última data em que estas verificações foram feitas. A verificação desta data é feita sempre que um utilizador se tenta autenticar com o servidor.

5. Session Key

Após a autenticação de um utilizador no sistema é criada uma session key. Esta session key será uma chave única naquela sessão, gerada sempre que um utilizador se autentique, partilhada entre o cliente e o servidor.

Em todas as comunicações feitas do cliente para o servidor é gerado um HMAC da concatenação de todos os argumentos passados ao servidor e esse HMAC também é enviado. Ou seja, todas as comunicações levam um argumento extra. Este HMAC é gerado com a session key partilhada entre o cliente e o servidor.

Em todas as comunicações o servidor irá concatenar os argumentos recebidos e irá também gerar um HMAC dessa concatenação com a sua session key referente ao utilizador. Se o HMAC gerado não for igual ao HMAC recebido do cliente, assume-se que os dados não são fidedignos e, por isso, são descartados e a suposta operação não chega a realizar-se.

A session key é criada no servidor e é enviada cifrada com a chave pública do cliente, deste modo garante-se que ninguém, à exceção do cliente e do servidor, tem acesso à session key.