Relatório de Segurança

Sistemas Distribuídos

3ª Entrega



Grupo 67:

André Nunes 64728

André Vieira 79591

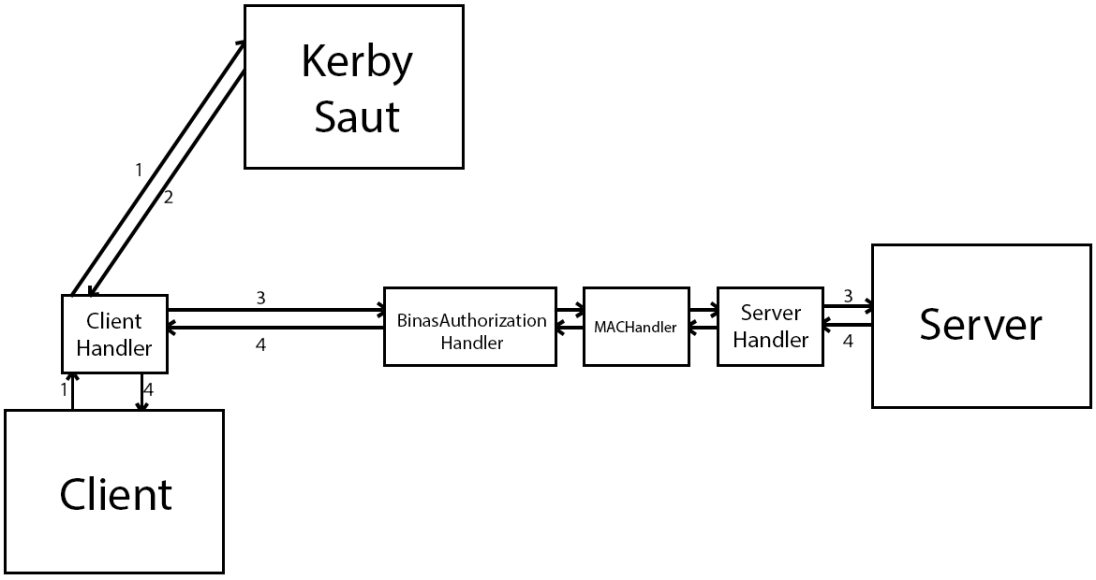
Ricardo Marques 81778

URL GitHub : https://github.com/tecnico-distsys/A67-SD18Proj

Introdução

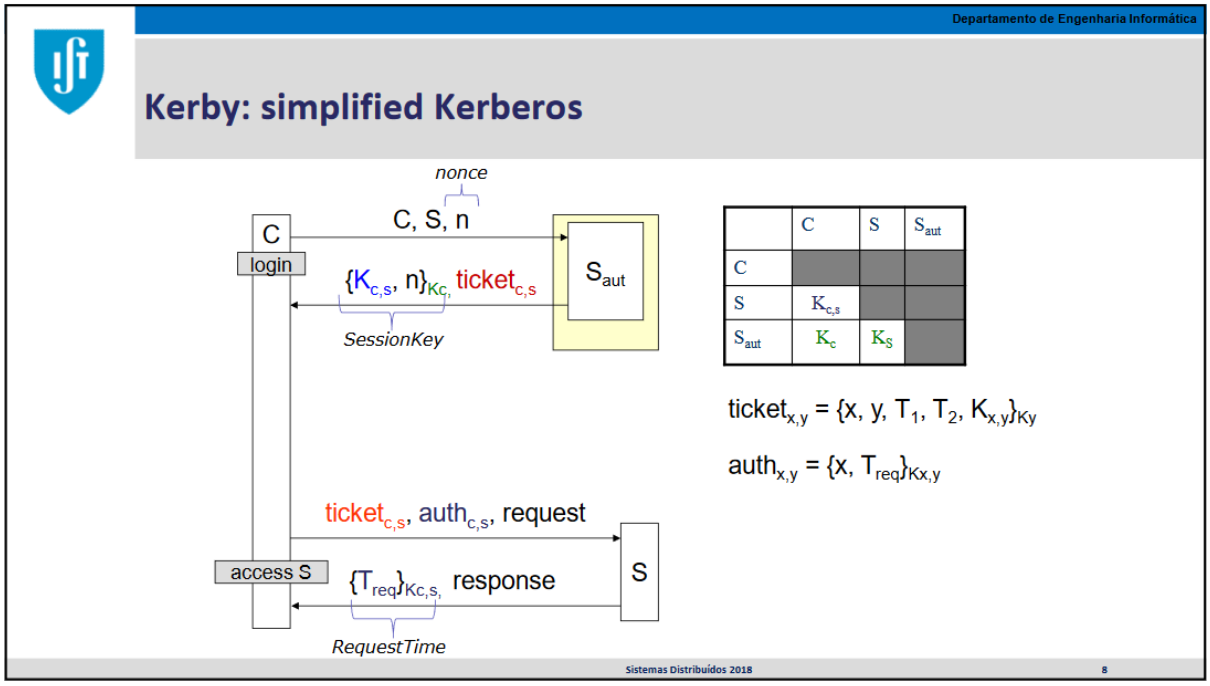
O objetivo da terceira parte do projeto consiste em implementar mecanismos de segurança, seguindo políticas de segurança e precavendo um modelo de ameaças que até aqui poderiam atacar o nosso sistema.
O servidor Binas utiliza uma versão simplificada do protocolo Kerberos (versao V5) sem servidores TGS autónomos, exigindo apenas um pedido-resposta para o cliente obter o ticket necessário para invocar o servidor.
O servidor de autenticação foi desenvolvido pelos docentes. Este servidor permite aos clientes do Binas pedirem uma nova sessão no Binas através de um web service SOAP com um WSDL e URL bem conhecido.
Este servidor conhece um conjunto de contas de utilizadores que se assumem per-registadas. No nosso projeto guardamos esses dados em variáveis static no Handlers de segurança, tais como nomes de utilizadores e respetivas senhas secretas.

Modelo de funcionamento



O Cliente autentica-se com a sua mensagem SOAP do seu pedido a ser interceptada pelo KerberosClientHandler, onde este comunica com o Sauth no Kerby (1), recebendo de seguida uma chave de sessão e um ticket (2). Com isso, o utilizador fica autenticado. A mensagem segue então para o servidor, sendo intercetada pelo handler BinasAuthorizationHandler que verifica se é um utilizador genuíno, por um MACHandler que verifica se a mensagem foi alterada, e pelo KerberosServerHandler que verifica a frescura da mensagem e finalmente chega ao servidor (3) onde o pedido é processado e a resposta é devolvida (4).

Solução detalhada



- Autenticação de utilizadores

A autenticação dos clientes é feita no Kerby. Este possui os dados do utilizadores e respetivas passwords. O cliente autentica-se comunicando com o Kerby onde este devolve uma chave de sessão para que possa ser utilizada, depois, nos pedidos aos servidores.

- Controlo de acessos

O cliente decifra a chave de sessão com a sua chave e verifica se o nounce que recebeu do servidor é o mesmo que enviou no pedido (gerado aleatoriamente), garantido assim que a resposta corresponde ao pedido. O ticket recebido pelo cliente inclui um intervalo de tempo para o qual um pedido do utilizador é válido, garantindo assim a frescura da mensagem e evitando ataques de replay onde um atacante poderia enviar uma mensagem com um timestamp antigo (demonstrado no caso seguinte). Sendo assim, quando o cliente quiser fazer um pedido, envia-o e com ele estará o ticket para o servidor saber que é um utilizador autenticado. A interceção e acesso às mensagens SOAP é feita não só no lado do cliente mas também no lado do servidor. Para além do tempo dos tickets, a favor de manter a frescura, existe o handler BinasAuthorizationHandler que verifica se o email que está no pedido é o mesmo que está no SOAP correspondente, assim confirma-se que a mensagem não foi adulterada e que um utilizador não está a fazer-se passar por outro.

- Integridade dos pedidos e das respostas

A integridade dos pedidos e das respostas é feito através de uma inserção de um resumo ao header das mensagens SOAP, usando o algoritmo HmacSHA256 e consequente verificação.
Isto é possível através do handler (MAC Handler) , que concatena o body da mensagem com a chave de sessão e gera um resumo, sendo esse resumo colocado no header da mensagem SOAP que será comparada pelo cliente/servidor que a receber. Este irá repetir o processo, gerar um novo resumo e compará-lo com o resumo obtido no header da mensagem. Se os resumos forem iguais, garante-se que a o corpo da mensagem não foi modificada, garantido assim a integridade e autenticidade, pois concluí-se que a mensagem foi de facto emitida pelo emissor legítimo.

Mensagens SOAP

Abaixo representamos a sequência de mensagens SOAP trocadas entre o cliente e servidor para a operação *activateUser*

1 - Mensagem Outbound cliente, onde é possível verificar que no header foram adicionadas as tag Ticket que corresponde ao ticket cifrado, Auth que corresponde ao autenticado cifrado, e mac que corresponde ao resumo

2 - Mensagem Inbound no servidor, onde se verifica que efetivamente se trata da mensagem enviada pelo cliente, podendo os Handlers do servidor aceder às informações enviadas no header

```
 IN BOUND SOAP MESSAGE:
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <t:ticket xmlns:t="http://ticket">Cjx0aWNrZXQgeG1sbnM6bnMyPSJodHRwOi8va2VyYnkuc2Rpcy50ZWNuaWNvLnVsaXNib2EucHQvIj4KICAgIDxkYXRhPmtuZWQ
8zd1MySTVkc0RmdDVJY2VGT1QxQkFNQWRZbkZwQXFaamllTzJIZTRtdVlMeHdmaDhJUzVoL044NEE0ZE9xd215c3VBZnFXSWdLaFBWNmg2QjNKSXZ2QjRnN2JQelQzeVUwWFBvNHZ
hBN09jOW93K2U1Q08rZHZLMzZrcmdVb2pBQ0RtVGY1ZXBnVlRwR0hDamN0dWIwQUJkY011bnIzRmRSaVcvQTdSRTN4MOxkZE5EwUxUSDZBRHpxT1pDbzlwdDg2WTlCRzN4QzJ0Z1F
4KPC90aWNrZXQ+Cg==</t:ticket>
    <a:auth xmlns:a="http://auth">CjxhdXRoIHhtbG5zOm5zMj0iaHR0cDovL2tlcmJ5LnNkaXMudGVjbmljby51bGlzYm9hLnB0LyI+CiAgICA8ZGF0YT5OcmpZdEx0ZlJ
IrTkFEVmd3dWhBwkFEU2tPK2RDVUoxYk5zOWRqSVkzclA1Q2Y3Vy9JVjZWRlpFYOpTT24rTldDSkFLM09pbVUrNEZzNUhqTStQdEVHOHRyTzFyOVVkUFBOVTgyWWN4Z2RRMDZyWWx
RoPgo=</a:auth>
    <m:mac xmlns:m="http://mac">hKBKMeoZEzQsnyiLl6ZZMqF5a/4nZHDO+17r3fn3VDU=</m:mac>
  </SOAP-ENV:Header>
  <S:Body>
    <ns2:activateUser xmlns:ns2="http://ws.binas.org/">
      <email>alice@A67.binas.org</email>
    </ns2:activateUser>
  </S:Body>
</S:Envelope>
```

3- Mensagem Oubound no servidor, onde foi adicionada a tag Treq que corresponde ao timestamp que o handler do servidor extraiu do autenticador, e um novo resumo representado na tag mac, para garantir a integridade da resposta do servidor

```
 OUT BOUND SOAP MESSAGE:
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <m:mac xmlns:m="http://mac">9rhg540F/dswDWWA0jdUeJyNLJEI15xkkOIEn45/Kkg=</m:mac>
    <r:treq xmlns:r="http://treq">Cjx0cmVxIHhtbG5zOm5zMj0iaHR0cDovL2tlcmJ5LnNkaXMudGVjbmljby51bGlzYm9hLnB0LyI+CiAgICA8ZGF0YT56U3dsb2Y1Nld
hnRDVDT1l5MkYzRjJ6aDh1RVBDODYvT2o3MVc1LzVOZzdHSFc5Z3MwYTVBSnFrNUs5wHZSR0NndGJERWowMHgxdlQ3QnhxZWIxMGJUdXFRbHg0QUdmYwJieHluem92VkgxN0dGY3d
  </SOAP-ENV:Header>
  <S:Body>
    <ns2:activateUserResponse xmlns:ns2="http://ws.binas.org/">
      <user>
        <email>alice@A67.binas.org</email>
        <hasBina>false</hasBina>
        <credit>10</credit>
      </user>
    </ns2:activateUserResponse>
  </S:Body>
</S:Envelope>
```

4 – Mensagem Inbound no Cliente, onde se confirma a correta receção da mensagem enviada pelo servidor, e onde o cliente acede ao resumo e a Treq

```
 IN BOUND SOAP MESSAGE:
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <m:mac xmlns:m="http://mac">9rhg540F/dswDWWA0jdUeJyNLJEI15xkkOIEn45/Kkg=</m:mac>
    <r:treq xmlns:r="http://treq">Cjx0cmVxIHhtbG5zOm5zMj0iaHR0cDovL2tlcmJ5LnNkaXMudGVjbmljby51bGlzYm9hLnB0LyI+CiAgICA8ZGF0YT56U3dsb2Y1Nl
hnRDVDT1l5MkYzRjJ6aDh1RVBDODYvT2o3MVc1LzVOZzdHSFc5Z3MwYTVBSnFrNUs5wHZSR0NndGJERWowMHgxdlQ3QnhxZWIxMGJUdXFRbHg0QUdmYwJieHluem92VkgxN0dGY3
  </SOAP-ENV:Header>
  <S:Body>
    <ns2:activateUserResponse xmlns:ns2="http://ws.binas.org/">
      <user>
        <email>alice@A67.binas.org</email>
        <hasBina>false</hasBina>
        <credit>10</credit>
      </user>
    </ns2:activateUserResponse>
  </S:Body>
</S:Envelope>
```