



TÉCNICO LISBOA

Sistemas Distribuídos

2.º Semestre 2017/2018

SD-Binas

<https://github.com/tecnico-distsys/A58-SD18Proj>

Relatório - Segurança



79758 Luísa Santo

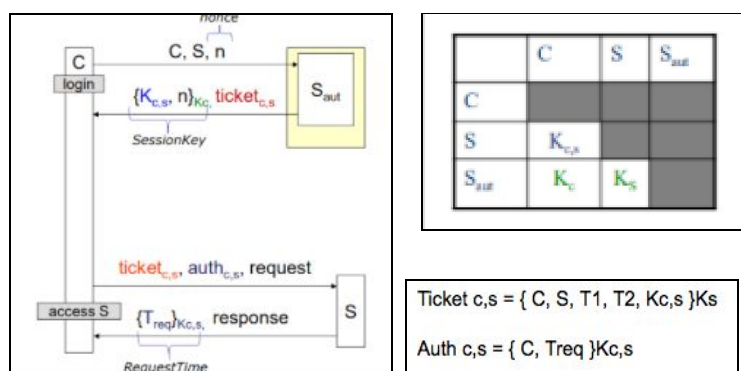


83517 Sofia Martins



87823 Kevin Vos

Protocolo Kerberos Simplificado



Nomenclatura:

C: Cliente Binas
Kc: Chave do Cliente C
S: Servidor Binas.
Ks: Chave do Servidor S
Saut: Servidor de autenticação Kerberos
Kcs: Chave de sessão do cliente C e do servidor S
T1 e T2: Intervalo de validade do Ticket.
n: Nounce, número aleatório gerado no cliente antes do envio de cada pedido.

Fig. 1 -Protocolo Kerberos Simplificado

Assume-se que as contas do **C** e do **S** encontram-se pré-registradas no **Saut**. A chave K_c é conhecida apenas pelo **C** e pelo **Saut**, a chave K_s é conhecida apenas pelo **S** e pelo **Saut**. O protocolo Kerberos permite os utilizadores autenticarem-se e fazer pedidos ao servidor Binas de forma segura, confidencial e manter a integridade das mensagens enviadas através de funções digest.

Troca de mensagens

O procedimento de autenticação consiste no **C** enviar para **Saut** o nome do cliente, o nome do servidor que pretende comunicar e um nounce.

O **Saut** gera um Ticket, cifrado com a chave K_s , e cria uma chave de sessão K_{cs} cifrada com a chave K_c . O Ticket e a chave de sessão são enviados do **Saut** para o **C** como resposta.

Após resposta do **Saut**, o **C** necessita de abrir a chave de sessão K_{cs} com a sua chave K_c , permitindo construir um pedido a **S**. Antes de enviar o pedido, é construída uma mensagem SOAP, cujo corpo possui informações sobre o pedido, e o cabeçalho possui o Ticket e um Auth, cifrados com a chave de sessão K_{cs} . Por fim, **S** abre e valida o ticket recebido com a sua chave K_s , e verificar se o nome do cliente no Ticket coincide com o nome do cliente no Auth, rejeitando pedidos caso contrário.

Controlo de acessos

O controlo de acessos consiste em rejeitar pedidos de utilizadores que pretendam invocar operações binas em nome de outro utilizador, isto é, rejeitar um pedido de um utilizador que pretende alterar o saldo de outro utilizador. O controlo de acessos é feito quando o **BinasAuthorizationHandler** interceta a mensagem SOAP, comparando o nome do utilizador do ticket com o do autenticador, e se existir, com o nome do utilizador presente no corpo da mensagem. Caso 2 dos nomes sejam diferentes, o pedido é rejeitado.

Replay Attacks

Os replay attacks são ataques que consistem em obter uma mensagem enquanto passa pela rede, e a reenviar com o objetivo de executar o pedido mais do que uma vez. Para combater isto, é efectuado um registo da data de envio das mensagens SOAP. Se a diferença entre a data de envio e a data atual for acima de 10 segundos, a mensagem não é processada. Também é necessário combater o caso em que o atacante volte a re-enviar a mensagem dentro dos 10 segundos. Tratamos deste caso através da geração de um nounce, um número aleatório suficientemente grande, registado numa coleção do Servidor. Caso haja um pedido com um nounce já pertencente a esta coleção, o pedido é rejeitado. É possível um pedido ser rejeitado devido a nounce repetido, mesmo se foi gerado pelo cliente antes de enviar, mas a probabilidade de 2 nounce serem iguais é extremamente pequena.

Handlers

Ao invocar um método remoto, os handlers interceptam o pedido antes de este sair para a rede. A ordem dos handlers que interceptam é KerberosClientHandler e MacHandler na saída. Na entrada a ordem é MacHandler, KerberosServerHandler e BinasAuthorizationHandler.

KerberosClientHandler

No BinasClient, existe uma TicketCollection que entradas constituídas por SessionKeyAndTicketView e um tempo limite de validade, permitindo verificar a validade do pedido quando invocado o método getTicket() da TicketCollection. Caso a TicketCollection esteja vazia, o KerberosClientHandler ao interceptar instancia um KerbyClient para obter um Ticket e uma nova chave de sessão do **Saut** e guarda na TicketCollection. O Ticket e a chave de sessão são adicionados ao cabeçalho da mensagem SOAP antes do envio como duas CipheredView.

MACHandler

Este handler permite garantir a integridade das mensagens. Gera-se um MAC (Message Authentication Code) com o algoritmo SHA-256. Este algoritmo permite “resumir” uma mensagem na forma de um código, de forma a que o número de colisões seja mínima, e não seja invertível. Na saída de um pedido do **C** ao **S**, gera-se o resumo M do corpo da mensagem e adiciona-se ao cabeçalho da mensagem SOAP. Na entrada de um pedido, é gerado um novo resumo M' do corpo da mensagem. Caso M e M' sejam diferentes, sabe-se que a mensagem foi alterada entre **C** e **S** e o pedido é rejeitado. Caso contrário garantiu-se a integridade da mensagem.

FreshnessHandler e TimeHandler

Estes handlers permite a frescura na comunicação. Esta é garantida através do TimeHandler presente na handler-chain **S** que regista a data de envio no Header das mensagens SOAP. Se a diferença entre esta data e a data atual for acima dos 10 segundos, a mensagem é descartada. No entanto, durante o intervalo de 10 segundos pode existir ataques de repeticao. Esta situação é resolvida com a introdução de um número aleatório **nonce** através do FreshnessHandler guardado no lado do Servidor. A chegada de uma nova mensagem, confirma-se se o nonce já foi anteriormente lido. Os números aleatório são guardados num ficheiro tsv, de modo a que os dados possam ser preservadas.

Exemplo de uma mensagem SOAP para a operação activateUser, do cliente binas para o servidor Binas.

```
2018-05-18T17:18:59.296 OUTbound 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>
    <ns1:ticket xmlns:ns1="urn:ticket"><?xml version="1.0" encoding="UTF-8" standalone="no"?>&lt;ticket
      xmlns:ns2="http://kerby.sdis.tecnico.ulisboa.pt/"&gt;&lt;data&gt;SJoLxhyXymH4W9emFsJY1Kmxs0oQnqfNsStmqGE9nbda
      cI8nK+GsJ0fdb5eeMURk0Hg2axwYMMpolKQ0jP1Ho8MtH/YqPGMeKX7Y8As2/Q2tcU/x96fP20YYkw52FiUEp2qWPYFuJhrFXH5veYo3BG06Re
      NJbyvjHNCZwiqmXBI6TnV3sIzf9hz4Hy7GAR0YZc3RoF2BCp0hZ0mUYiUJEUf0aCnERPBG1Qdf1UKSwW64BWjnhYBT1GyBN0ompe1lryU7sh7
      /sw9ShYVmxGs2bwA5zmvKyaQj+jic/1iuyvRa/aw6Q0RH2+977gRy6U29DxRrE370PyFTy8TnyB5/LxRrd5x+7IhelFs+jXM/L80rGbX8af9wL
      +is0JZeRA0Ft6vww0dJe2rmm0ITfjQ=&lt;/data&gt;&lt;/ticket&gt;</ns1:ticket>
    <ns1:auth xmlns:ns1="urn:auth"><?xml version="1.0" encoding="UTF-8" standalone="no"?>&lt;auth xmlns:
      ns2="http://kerby.sdis.tecnico.ulisboa.pt/"&gt;&lt;data&gt;1Vm/ty907A08rMHvnFnEavGCMkf6d+pfI8bPSGzZ/K4CRsdeZ7d
      nefBmAAHAq05MwaQYff46GTa0JmehqjKac5GGFI4SKx9l8/NgNFAZcmIi20Mc5z3wKPLR2c5L3yIP8qWUWN/GgcsWZ8av2XUgmFmxgrHMETaxL
      yztI26o6kvyGY1TynP3f1JXqBBpYnrxtgSVPbqQr+00t/WqQJ01cEDwtMEQcRNNXryC5JIjiE=&lt;/data&gt;&lt;/auth&gt;</ns1:aut
      h>
    <ns1:digest xmlns:ns1="urn:digest">p22owhUWd/2LTunUsAnKh7hl+RuTXqiXhrx+pEU+7xo=</ns1:digest>
  </SOAP-ENV:Header>
  <S:Body>
    <ns2:activateUser xmlns:ns2="http://ws.binas.org/">
      <email>alice@A58.binas.org</email>
    </ns2:activateUser>
  </S:Body>
</S:Envelope>
```

Ticket cifrado na forma de CipheredView serializado

Auth cifrado na forma CipheredView serializado

Resumo gerado através do corpo (Digest)

Corpo da mensagem SOAP

Observa-se que a mensagem possui no cabeçalho um ticket e um auth cifrados e possui um digest gerado a partir do corpo da mensagem