Segurança de Dados em Transmissões de Stream de Vídeo

RESUMO:

Este trabalho consiste na abordagem das questões de segurança numa transmissão streamming de vídeo de uma rede de arquitetura cliente/servidor. Toda a transmissão para o cliente deve ser feita usando algum tipo de criptografia de forma que não permita que pessoas não autorizadas acessem tais transmissões. Não iremos nos ater às questões de performances ou perda de pacotes na transmissão do streamming, pois o trabalho visa abordar a autenticação e a confidencialidade dos dados entre cliente/servidor.

INTRODUÇÃO:

Será utilizado um exemplo já pronto desenvolvido em Java de arquitetura cliente/servidor de stream de vídeo. O software irá garantir que usuários não autorizados fiquem impossibilitados de acessarem os pacotes. Para a autenticação, será feita através de chave de sessão com chaves simétricas, pois criptografar/descriptografar é comprovadamente mais rápida (em software ou hardware) do que criptografia de chaves assimétricas (chaves-públicas). Já para a criptografiar os dados será utilizada biblioteca de criptografia da linguagem Java.

PROPOSTA:

O trabalho propõe em implementar em Java na camada de aplicação um software que aborde características de segurança de uma rede baseada na arquitetura cliente/servidor, onde inicialmente qualquer usuário (ou cliente) que deseja acessar uma transmissão deve-se autenticar no servidor e ser aceito. Após a autenticação será gerada uma chave de sessão única entre cliente-servidor.

Toda a transferência de vídeo para o cliente será feito usando essa chave de sessão, para isso usaremos um algoritmo de chave simétrica. Para tal usaremos algoritmo Rjindael (AES – 128 bits).

VALIDAÇÃO:

Para demonstrar que os dados estão sendo criptografados entre cliente/servidor iremos utilizar um sniffer de rede chamado WireShark.

Para demostrar autenticação, partiremos de um exemplo onde um usuário não consiga se autenticar no sistema ele não terá uma chave de sessão compartilhada com o servidor, logo sem a chave o usuário não irá conseguir descriptografar os dados.

IMPLENTAÇÃO:

Para a criptografia/descriptografia dos dados, utilizaremos a biblioteca JCE (Java™ CryptographyExtension). Enquanto para a autenticação do cliente/servidor, utilizaremos chave de sessão simétrica através do algoritmo Rjindael (AES – 128 bits). Com ele a criptografia se faz em blocos, diferente de outros algoritmos que faz a criptografia de bits em

bits. Este algoritmo é o sucessor do DES (atualmente considerado inseguro para muitas aplicações, pois isto se deve principalmente a pequena chave de 56-bit, sem contar que esse algoritmo permite se reverter criptografia).

TESTE:

Num mesmo computador com a JVM (Máquina virtual Java) iremos rodar a aplicação cliente e servidor. Além disso teremos uma aplicação que será responsável pela autenticação do cliente que funcionará como um servidor de autenticação e distribuição de chaves. No momento que o cliente estiver autenticado e autorizado o servidor irá enviar os dados criptografados usando a chave de sessão compartilhada entre o cliente. Nesse momento será possível capturar esses dados pelo programa sniffer e notar os pacotes criptografados.

O objetivo do trabalho é que não haja um aumento de delay na transmissão do vídeo. Porém como a transmissão será feita no mesmo computador não há como testar se os efeitos numa rede real haveria perda de performance.

CONCLUSÃO:

Com a implementação desse trabalho, podemos perceber a dificuldade em se acessar um dado criptografado e a importância de uma autenticação com o servidor para obtenção da chave de sessão. Também podemos perceber a importância da segurança da informação se utilizando em algoritmos de criptografia que impedem em um tempo computacionalmente viável que um usuário não autorizado acesse ao conteúdo do dado.