Projeto de Sistema de Arquivos Criptografados

Vitor Vinicius Gomes da Silva, Elise Yumi Tanaka, Felipe S. Uehara

***Resumo*—Projeto Final da discliplina Segurança e Auditoria de Sistemas ministrada pelo Prof. Lucas Dias H. Sampaio na Univesidade Tecnológica Federal do Paraná em Cornélio Procópio - PR**

***Index Terms*—Arquivos, Sistema, Criptografia, Segurança, Servidor, Seguro.**

I. Introdução

SEGURANÇA em sistemas de armazenamento de dados é um fator decisivo para isolar a informação somente aos usuários que a detêm. Muitos sistemas de arquivos são compartilhados por diversos usuários o que faz com que o controle de acesso aos arquivos entre eles deva ser feito de forma eficaz: um usuário não pode ter acesso as informações de outro sem permissão. Neste ponto, surge a necessidade de se criptografar as informações armazenadas no sistema. Entretando, muitos sistemas de arquivos legados não possuem suporte nativo à criptografia de arquivos, o que prejudica a segurança da informação, tornando-os assim sistemas não confiáveis para se armazenar dados.

II. Proposta

Baseada no tema proposto pelo professor nas notas de aula da disciplina, a proposta deste trabalho é criar um sistema de armazenamento de informações na Web utilizando servidores sem suporte nativo à criptografia e que permitam a listagem das informações contidas neles. A informação seria previamente criptografada no cliente ou no servidor de aplicação e então registrada no servidor web. Todos os usuários do sistema poderiam ter acesso à listagem e ao download dos arquivos, entretando, somente aqueles que detêm a chave de descriptografia poderiam ler e manusear as informações. Quando o usuário que detêm a chave, desejar o acesso à informação, ela é automaticamente descriptografada no processo de aquisição somente para o usuário em questão. Isto poderá ser feito pelo servidor de aplicação ou pelo aplicativo do cliente.

A. Tecnologias e Ferramentas

Para realizar este projeto foram utilizadas as seguintes tecnologias e ferramentas:

1) Criptografia AES: AES (sigla para "Advanced Encryption Standard"), é uma criptografia de blocos com chave simétrica e pretende-se usar o modelo AES-256 no sistema desenvolvido neste projeto. O padrão de criptografia AES utiliza o algoritmo de Rijndael, que utiliza métodos de substituição e permutação para criptografar uma mensagem. O AES foi anunciado oficialmente em 2001 e tornou-se um padrão em 2002. O AES atualmente é dos algoritmos mais populares usados para criptografia de chave simétrica.

2) CryptoJS: O CryptoJS é uma coleção de algoritmos criptográficos padrões e seguros implementados em JavaScript usando as melhores práticas e padrões de segurança. Têm suporte a muitas tecnologias e algoritimos de criptografia e produção de hashes tais como: AES, TripleDES, RC4, MD5, SHA-1, SHA-256 entre outros. Esta ferramenta foi utilizada neste projeto para implementar a criptografia AES no aplicação no navegador do cliente.

3) Servlet: Servlet é uma tecnologia usada para criar páginas web geradas dinamicamente baseadas em HTML, XML ou outros tipos de documentos usando como base a linguagem de programação Java. Esta tecnologia será utilizada para processar as requisições dos usuários como os pedidos de download e upload de arquivos. O software desenvolvido será executado sob esta plataforma e ficará responsável por armazenar e disponibilizar os dados dos usuários conforme suas demandas. Para implantar e executar o sistema proposto em Servlet, um servidor web compatível, como Apache Tomcat, Jetty ou Glassfish, será necessário.

4) Ambiente: A aplicação do lado do servidor é um sistema desenvolvido em Java e portanto, será capaz de ser executado em nos principais sistemas operacionais existentes: Windows, Linux, etc. A aplicação que é executada no lado do cliente é desenvolvida em HTML, CSS e Javascript e dessa forma, compatível com a grande maioria dos navegadores web modernos. A comunicação entre a aplicação do lado cliente e a aplicaçãod o lado servidor é feita usando o protocolo HTTP (Hypertext Transfer Protocol).

III. Metodologia

O sistema de armazenamento criptografado segue uma metodologia para cada um dos seguintes casos:

1) Envio de Arquivo: O usuário, utilizando-se do aplicativo web, indica o arquivo à ser armazenado e solicita o envio. A aplicação no lado do navegador do cliente criptografa o arquivo usando a ferramenta CryptoJS e o algoritimo AES com sua chave de criptografia e envia-o para o servidor. O servidor de aplicação recebe o arquivo criptorafado e o armazena. O fluxo desta metodologia de envio de arquivo pode ser visto na Figura 1.

2) Recebimento de Arquivo: O usuário, utilizando-se do aplicativo no navegador web, indica o arquivo desejado. O servidor de aplicação busca o arquivo no servidor web e o disponibiliza para download pelo cliente. Após o download pela aplicação no lado do cliente, o processo de descriptografia é feito diretamente no navagador usando usando a ferramenta CryptoJS, o algoritimo AES e a chave de criptografia do cliente. O fluxo desta metodologia de descarregamento (download) de arquivo pode ser visto na Figura 2.

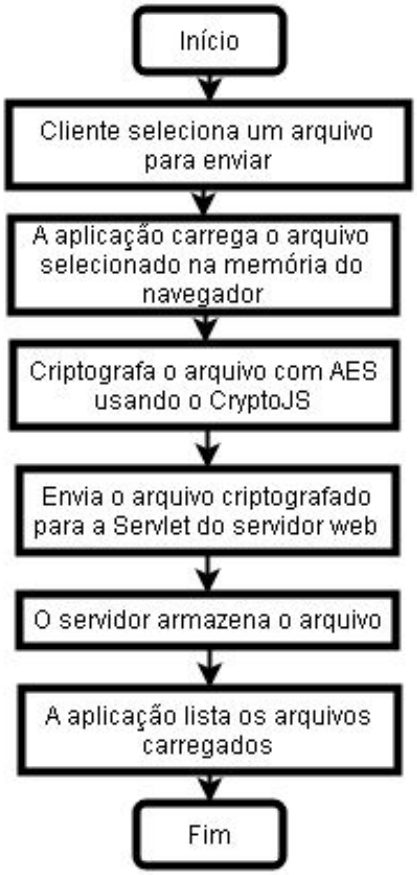

Figura 1. Processo de envio de arquivo.

Figura 2. Processo de descarregamento de arquivo.

IV. Casos de Testes

A. Teste de conteúdo de pacotes com criptografia habilitada

Este teste busca verificar se o conteúdo dos arquivos permanecem ilegíveis durante envio e recebimento pela rede. Usando a ferramenta de de captura de pacotes SmartSniff, os pacotes trocados entre o cliente e servidor foram capturados para análise. Para isto, foi usado um arquivo de texto chamado "Test.txt" com o conteúdo de texto literal igual a "Informação Secreta".

B. Teste de conteúdo de pacotes com criptografia desabilitada

Este teste busca verificar se o conteúdo dos arquivos permanecem legíveis durante envio e recebimento pela rede com a criptografia desabilitada. Usando a ferramenta de de captura de pacotes SmartSniff, os pacotes trocados entre o cliente e servidor foram capturados para análise. Para isto, foi usado um arquivo de texto chamado "Test.txt" com o conteúdo de texto literal igual a "Informação Secreta".

V. Resultados dos Testes

A. Pacote de envio com criptografia habilitada (APÊNDICE A)

Decodificando o parametro "data", que corresponde à informação criptografada do conteúdo do arquivo, do padrão Base64 para UTF-8, temos que data é igual à: "Salted__"+ @W ht v -zG. ĐN&l #". Como a informação é ilegível significa que ela realmente está sendo enviada de maneira criptografada.

B. Pacote de recebimento com criptografia habilitada (APÊNDICE B)

Decodificando o conteúdo do pacote recebido do padrão Base64 para UTF-8, temos que ele é igual à: "Salted__"+ @W ht v -zG. ĐN&l #". Como a informação é ilegível significa que ela realmente está sendo recebida de maneira criptografada.

C. Pacote de envio com criptografia desabilitada (APÊNDICE C)

Decodificando o trecho "SW5mb3JtYcOnw6NvIFNlY3JldGE=", que corresponde à informação não criptografada do conteúdo do arquivo, do parametro "data"do padrão Base64 para UTF-8, temos que data é igual à: "Informação Secreta". Como a informação é legível significa que ela realmente está sendo enviada de maneira não-criptografada, comprometendo a segurança da informação.

D. Pacote de recebimento com criptografia desabilitada (APÊNDICE D)

Decodificando o trecho "SW5mb3JtYcOnw6NvIFNlY3JldGE="do conteúdo do pacote, que corresponde à informação não

criptografada do conteúdo do arquivo, do padrão Base64 para UTF-8, temos que data é igual à: "Informação Secreta". Como a informação é legível significa que ela realmente está sendo recebida de maneira não-criptografada, comprometendo a segurança da informação.

VI. Conclusão

O sistema deste projeto foi pensado para solucionar um problema de segurança no armazenamento de dados em um ambiente inseguro como um servidor web comum. Como pôde ser visto com os resultados dos testes, o sistema realmente faz o que é proposto por este trabalho. Demais funcionalidades como Controle de Usuários e Grupos podem ser implementadas em trabalhos futuros.

Referências

[1] José Moraes, *Segurança de dados com AES*, Disponível em: https://portal.vidadesilicio.com.br/seguranca-de-dados-com-aes/, Acesso em 18 de Dezembro de 2018.

[2] Wikipédia, a enciclopédia livre, *JavaServer Pages*, Disponível em: https://pt.wikipedia.org/wiki/JavaServer_Pages, Acesso em 18 de Dezembro de 2018.

Apêndice A
Pacote de envio com criptografia habilitada

POST /upload HTTP/1.1 Host: 192.168.0.100:8080 Connection: keep-alive Content-Length: 137 Cache-Control: max-age=0 Origin: http://192.168.0.100:8080 Upgrade-Insecure-Requests: 1 Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8 Referer: http://192.168.0.100:8080/index.html Accept-Encoding: gzip, deflate Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7

data=U2FsdGVkX1+BT1VqgIFo0PuG/7BBzWKVusexMRgFQ0nkihKk9OvT565tDR/aoIKD3d0jF64atH7YxJoQEf0QJC+xd6uJ6Sr8jpoiuUotDzo=&name=Test.txtGET /index.html HTTP/1.1

Apêndice B
Pacote de recebimento com criptografia habilitada

HTTP/1.1 200 OK Server: GlassFish Server Open Source Edition 4.1.1 X-Powered-By: Servlet/3.1 JSP/2.3 (GlassFish Server Open Source Edition 4.1.1 Java/Oracle Corporation/1.8) Content-disposition: attachment; filename=Test.txt Content-Type: text/plain;charset=UTF-8 Date: Fri, 15 Jun 2018 15:23:42 GMT Transfer-Encoding: chunked

6e U2FsdGVkX1+BT1VqgIFo0PuG/7BBzWKVusexMRgFQ0nkihKk9OvT565tDR/aoIKD3d0jF64atH7YxJoQEf0QJC+xd6uJ6Sr8jpoiuUotDzo=

0

Apêndice C
Pacote de envio com criptografia desabilitada

POST /upload HTTP/1.1 Host: 192.168.0.100:8080 Connection: keep-alive Content-Length: 80 Cache-Control: max-age=0 Origin: http://192.168.0.100:8080 Upgrade-Insecure-Requests: 1 Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8 Referer: http://192.168.0.100:8080/index.html Accept-Encoding: gzip, deflate Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7

data=data:text/plain;base64,SW5mb3JtYcOnw6NvIFNlY3JldGE=&name=Test.txtGET /index.html HTTP/1.1

Apêndice D
Pacote de recebimento com criptografia desabilitada

HTTP/1.1 200 OK Server: GlassFish Server Open Source Edition 4.1.1 X-Powered-By: Servlet/3.1 JSP/2.3 (GlassFish Server Open Source Edition 4.1.1 Java/Oracle Corporation/1.8) Content-disposition: attachment; filename=Test.txt Content-Type: text/plain;charset=UTF-8 Date: Fri, 15 Jun 2018 15:37:57 GMT Transfer-Encoding: chunked

35 data:text/plain;base64,SW5mb3JtYcOnw6NvIFNlY3JldGE=

0