Projeto de Programação IMD0040 - Representando Comportamento de Usuários para Detecção de Ameaças Internas

Carlos José L. Júnior[1], Judson Matheus F. de Andrade[2]

Instituto Metrópole Digital - Universidade Federal do Rio Grande do Norte
Caixa Postal 1524 - 59078-970 - Natal - RN - Brasil
judson.matheus.andrade@gmail.com, carlosjsl95@gmail.com

Abstract. *This paper is aimed to present the report of the Programming Language II third unity project's course developed by the students Carlos José Leonel Júnior and Judson Matheus Ferreira de Andrade. There will be presented the goal of this project; the description of the code; the analysis of the code's complexity; which procedures were made to complete the activity; the class diagram and, at last, the results.*

Resumo. *Este documento tem como objetivo apresentar o relatório do projeto da terceira unidade da disciplina Linguagem de Programação II, dos alunos Carlos José Leonel Júnior e Judson Matheus Ferreira de Andrade. Nele serão apresentados: O objetivo do projeto; a descrição do código; a análise de complexidade do código; quais foram os procedimentos para a realização da atividade; o diagrama de classes e por fim os resultados obtidos.*

1. Introdução

Com o avanço constante das tecnologias se faz necessário desenvolver novos modos de garantir a segurança das pessoas que as utilizem. No contexto empresarial, hoje, existem várias formas de agentes externos conseguirem realizar ataques e roubar dados vitais das empresas, mas uma outra forma de ataque que não é tão destacada, mesmo sendo muito importante, são os ataques internos.

Uma das formas de lidar com essas ameaças é o uso de técnicas de detecção de anomalias, método onde os usuários têm suas atividades registradas e analisadas a fim de detectar possíveis atividades maliciosas. Esse projeto consiste exatamente de um sistema que permite representar os perfis de usuários, e com eles analisar suas atividades, assim, gerando informações vitais para a segurança interna da empresa que o utilize.

2. Descrição do Problema Abordado

O projeto tem quatro objetivos principais que devem ser abordados que são: ler arquivos de logs como entradas de dados, montar os perfis dos usuários baseado nos logs, visualização dos perfis dos usuários e detecção de anomalias com análise dos dados.

Esses objetivos são propostos com intuito de promover um sistema capaz de garantir a segurança dos agentes internos à empresa que o utilize, onde os logs são os registros das atividades realizadas por cada usuário nos equipamentos da empresa. Sendo registrado, nesses logs, o computador utilizado, o dia e a hora da atividade e qual foi a atividade realizada.

3. Estruturas de Dados Utilizadas

Na implementação do sistema foram utilizadas duas estruturas de dados, cada uma responsável por guardar e gerenciar dados específicos do programa. As estruturas são uma árvore binária de busca e uma árvore genérica.

Uma árvore binária de busca consiste em uma estrutura de dados de árvore binária baseada em nós, onde todos os nós da subárvore esquerda possuem valor menor que a raiz e todos os nós da subárvore direita possuem valor maior que a raiz. Ela foi escolhida por sua complexidade ser da ordem de O(log n) no melhor caso, sendo fundamental para poder trabalhar com uma grande quantidade de dados.

Já a árvore genérica consiste em uma estrutura do tipo árvore que não possui número de filhos fixo, não possuindo qualquer regra de inclusão com ordenação, ou qualquer limitação neste sentido. Foi decidido utilizar essa estrutura devido a necessidade de trabalhar com uma quantidade grande e indeterminada de dados.

4. Descrição da Abordagem de Solução do Problema

Primeiramente, para solucionar o problema, é necessário que os dados seja fornecidos e gravados pelo sistema, pois sem os dados nada pode ser feito. Para isso foi planejado métodos para ler os arquivos informados pelo usuário que posteriormente é

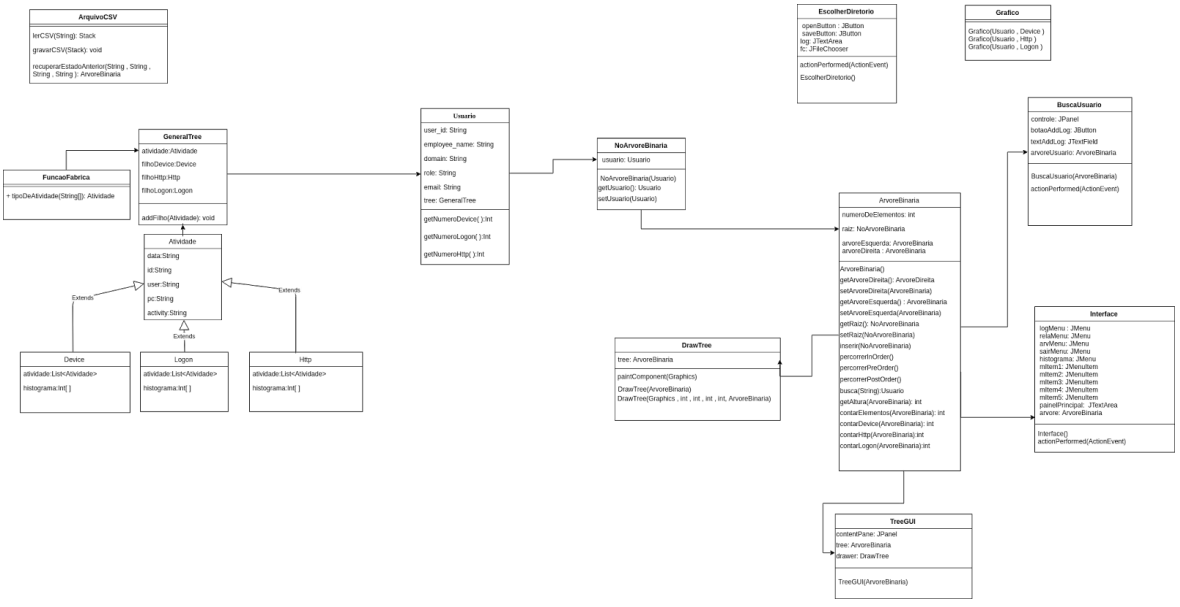
gravado em uma pasta de backup. O sistema utiliza os dados contidos na página de backup para montar o perfil e realizar a análise dos usuários.

Para montar o perfil do usuário foi necessário definir especificamente as atividades que ele realiza. Sendo definida três atividades chamadas de Logon, Http e Device. Logon se trata dos acessos aos computadores, tendo apenas duas possibilidades, Login ou Logoff. Http se refere às páginas Web acessadas pelo usuário, sendo registrado a "url" acessada. Device monitora as atividades de inserir e remover um pendrive, tendo duas possibilidades, connect e disconnect. Dessa forma basta definir que o usuário pode realizar essas atividades, e que cada atividade pode ser feita múltiplas vezes.

Para um melhor resultado do sistema se torna necessário possuir um forma eficiente para visualizar estes perfis, pois se as informações não podem ser compreendidas todo este trabalho terá sido inútil. Dessa forma, foi proposta a utilização de uma interface gráfica simples e intuitiva para apresentar as informações..

5. Detalhes de Projeto OO

Todas as classes utilizadas para criar o sistema proposto podem ser visualizadas no diagrama de classes apresentado a seguir. Nele é possível ver toda a estrutura hierárquica das classes, seu parâmetros e métodos mais importantes.



A classe ArquivoCSV é responsável por trabalhar com os logs. Ela possui três métodos. O primeiro é o lerCSV, que é responsável por ler um arquivo informado, salvando todos os dados em um Stack. O segundo é gravarCSV, que recebe como parâmetro uma Stack e cria arquivos específicos em uma pasta de backup. Esses arquivos são identificados através de uma classe *factory*, ela é responsável por determinar o tipo de dado que a classe ta trabalhando. A terceira é a recuperarEstadoAnterior, ela é responsável por acessar todos os arquivos de backup e restaurar a sessão para o estado da última vez utilizada.

A classe usuário é responsável por representar as diferentes pessoas que serão cadastradas no sistema. Para isso ela possui diversos atributos como: user_id, employee_name, domain, role e email. Esta classe também possui como atributo um árvore genérica, que será utilizada para registrar as diversas tarefas que o usuário possa realizar.

Como já foi dito, anteriormente, a classe GeneralTree tem a função de registrar as diferentes atividades que um usuário possa fazer. Para isso, ela possui três filhos, um do tipo Device, outro Http e outro Logon. Cada um é responsável por registrar o respectivo tipo de atividade.

Atividade é a classe que representa as tarefas que o usuário fez. Ela possui diversos parâmetros responsáveis por fornecer informações sobre a atividade que são, data, id, user, pc e activity. Da classe Atividade se estende três outras classes, Device, Http e Logon. Elas são responsáveis por diferenciar o tipo de atividade, cada uma possuindo uma lista e um histograma. A lista é responsável por gravar as diversas atividades de um mesmo tipo e o histograma é responsável por gravar as ocorrências das atividades em intervalos de horas.

6. Explicação Detalhada dos Algoritmos Utilizados

Para o bom funcionamento do programa é necessário garantir que todas as estruturas utilizadas sejam capaz de realizar suas atividades em um tempo aceitável. Para isso é importante que a complexidade de todos os algoritmos sejam as melhores possíveis - o mais próximo de O(1).

Dessa forma podemos analisar o programa baseando-se na complexidade dos principais métodos utilizados no sistema, que podem ser vistos na tabela apresentada a seguir.

Métodos	Complexidade	Descrição
lerCSV(String):Stack	O(n)	Sendo 'n' igual ao número de linhas do arquivo.
gravarCSV(Stack): void	O(n)	Sendo 'n' igual ao número de dados na Stack.
recuperarEstadoAnterior(String, String, String, String): ArvoreBinaria	O(n)	Sendo 'n' relacionado ao tamanho dos arquivos de backup.
inserir(NoArvoreBinaria): void	O(n)	Sendo 'n' relacionado à altura da árvore.
busca(String): Usuario	O(log n)	Para o melhor caso, quando a árvore está balanceada. Pode chegar até O(n).
addFilho(Atividade):void	O(1)	Adiciona em uma lista, que é O(1).

7. Conclusão

Com todo esse desenvolvimento podemos concluir que é possível estruturar os dados obtidos por meios de logs em um sistema. Onde, assim, podemos realizar análises a fim de identificar atividades suspeitas, ou até mesmo mostrar atividades ilegais dentro da empresa. Essa é uma das armas contra atividades maliciosas que uma empresa pode utilizar para se proteger nos dias de hoje.

8. Referências

Sociedade Brasileira de Computação (20??). "Instructions for Authors of SBC Conferences Papers and Abstracts", http://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos, Junho.

Universidade de São Paulo (2005). “Árvores binárias de busca (BSTs)”, https://www.ime.usp.br/~pf/estruturas-de-dados/aulas/st-bst.html , Junho.

Edson Prestes. “Complexidade de Algoritmos”, http://www.inf.ufrgs.br/~prestes/Courses/Complexity/aula1.pdf, Junho.

Donald Bell (2016). “O diagrama de classes: Uma introdução aos diagramas de estrutura em UML 2”, https://www.ibm.com/developerworks/br/rational/library/content/RationalEdge/sep04/bell/index.html, Junho