Senior Project Proposal

Title: Detection of Compromised Accounts Through Analysis of Server Logs

Author: Scott McCoy

Supervisor: Dr. Wilson

Degree: B.S. Computer Science

Background

Application security is a growing problem throughout the field of enterprise computing. It is widely accepted the primary goal of security is to prevent malicious agents from gaining access to a system. But what happens when they do? An equally important security question is how are attackers identified and disabled once they gain access to a system. One common method for identification of attackers is through the analysis of server log files. Log analysis is something that can be useful in a wide variety of applications, both small and large [1].

Motivation

The initial motivation for this project came from my employer who administers the computer systems for a university library. An issue that we have there is that when even user level accounts get compromised they can pose a serious risk to library assets. This is because the library pays for subscriptions to large research paper databases for its patrons. The problem here is that if a malicious user gains access to an account they may run a script that attempts to download all papers from a database. When a breach like this occurs the database provider will generally block access to its databases for the entire university causing inconvience to students and faculty and effectively wasting university resources.

Method

Stage 1
In the first stage of this project focus will be centered around identifying different

ways that potentially compromised accounts may be detected. Examples of this may include reverse DNS lookup, geolocation, and pattern analysis. Design priorities and a solid code plan will be developed in this stage.

Stage 2
In the second stage of this project the code will be written to target the specific problem at hand (our library, running our systems).

Stage 3
In the third stage of this project efforts will be made to make the project more generic as well as enhancing the overall functionality of the program. For example, automatic log file format detection, selectable detection options and prioritization schemes, and/or GUI interfaces. These improvements would be aimed at making the program more useful, more usable, and allowing it to be potentially useful for libraries beyond our own.

Prior Experience

I have previously made attempts to solve this problem. The most recent attempt involved using geolocation to detect when our systems were being accessed from outside the country, possibly indicating the use of a VPN. The problem with this approach was that I was unable to find any geolocation APIs that were free to use and that updated their geolocation databases often enough to be useful. There were some that were available to use but they cost roughly $300 per year.

Timeline

The following list represents some broad goals for the timeline of the project.

1) **First week of the semester**
Identify existing programs to base this project off of, identify all of the library functions that may be useful to have throughout the development of this project, create a flow chart/plan for how the program will work, choose the language to write in and the platform to target.

2) **By February 1st**
Have all of the library functions written and obtain the log files from Renaldo.

3) **By March 1st**
All the basic logic is written and the program is functional at a base level.

4) **March-April**
Improve the portability of the program, generalize the program, clean up the code and extend the documentation, add statistical reporting functionality, add any other features that may be interesting or useful if time permits.

5) **April-May**
Deploy the program to an actual university server, write the final report and create

the final presentation, troubleshoot any remaining issues.

6) **May**
Done/give presentation if I haven't already done so.

References

[1] 7 Authors, Beehive: Large-Scale Log Analysis for Detecting Suspicious Activity in Enterprise Networks (arijuels.com)