SERVER LOG ANALYSIS

May 1, 2018

Scott McCoy
Dr. Todd Wilson

Introduction

New computer systems are created and used daily that affect businesses and institutions both large and small. As the prevalence of enterprise computer systems grows, so too does the profitability of attacking such systems. With the quickly changing culture of software development security holes can easily be introduced as new APIs and frameworks become available but these security risks can go months or years without being patched. This means that often times it is not a question of if a user account will be compromised but when a user will be compromised. When a user of one of these systems is compromised a myriad different types of information is at risk of being stolen by attackers. Libraries risk losing their databases of research papers, businesses risk databases of customer information, and governments risk their national security.

The purpose of this project was to combat such attackers by attempting to identify compromised users and alert relevant information security personel before damage can be done. This was achieved through analyzing the log files of a server in real time in order to identify and respond to any users that exhibit suspicious behavior. The initial motivation for this project came from a university library that was having trouble with compromised user accounts that would download as many academic papers and journals as possible for resale. This would result in the affected databases being locked to the entire university until administrators were able to prove to the database provider that the attacker had been dealt with.

Libraries and Resources

Inspirations

There were several programs that this project drew inspiration from. The first piece of software that was studied was Splunk. Splunk is security information and event management program with a variety of features includeing data collection, searching, indexing, and analysis. The principle feature that drawn from Splunk was the real time analysis of log files with the intent of finding strange data patterns. Splunk many several advanced features such as data visualization and searching functionality but the goal of this project was to be much more light weight than something like Splunk in terms of size and overhead. Another program that inspired this project was IBM QRadar. The feature that we were most interested in replicating with this project was the detection of insider threats. Often times the most common threat in library systems the authenticated so it was important that our program would be able to detect threats from authenticated users. The final program that we drew inspiration from was Graylog. The inspiration from Graylog was taken primarily from its alerting mechanisms. Greylog is capable of sending an email or Slack message, spawning a new server to balance load, and blocking IP ranges in a firewall when a threat is detected.

Libraries and Tools

There were several external libraries and tools that were used in the creation of this project. The first major tool that was used was the Unix tail command. The tail command outputs the last 10 lines of a file to standard output. When used with the -f option, tail continuously outputs any line that is appended to the file. By using tail -f on the log file and then piping the output into the program using the Unix pipe operator. Another library that was used was the Boost library. One of the specific files that were included were the Boost lexical cast file ($< boost/lexical\_cast.hpp >$) which was used for catching errors during the processing of a raw line provided by the tail command. The other was the Boost string file ($< boost/algorithm/string.hpp >$) which was used in the division of the raw line.

Data Structures

[REVISE THESE SECTIONS LATER TO ENSURE CORRECTNESS]

Userdata and Free List

The Userdata class provided the abstraction for each line of data that was parsed. When a line of data was obtained it was immediately sent to the free list via the getNode function for parsing. This function extracted the needed data from the line (username and IP address), put it into a node from the freelist, and then returned the node. Once this node was obtained by the main loop it was time stamped with the current time added with the appropriate lookahead distances.

Priority Queue

The priority queue was the data structure responsible for managing when data should be expired from each of the map data structures. [ADD MORE HERE]

Flood Map

The flood map was responsible for detecting attacks that would cause a flood of requests to appear in the log file over a short period of time such as DoS, spidering, or any type of mass download script. This is achieved by mapping each user to a number of requests recieved from that user over a set period of time (the period of time and the number of requests are determined by values read in from the config file). If this threshold is crossed then an alert is sent to the administrator for further action. The flood lookahead distance will typically be relatively short as the purpose of this data structure is to detect bursts of activity.

IP Map

The IP map is a map from an IP address to a map from a username to a timestamp. The purpose of this data structure is to keep track of how many usernames are associated with each IP address. This allows for the detection of attackers who have compromised multiple accounts and are switching between them in an effort to avoid detection to potentially be detected. There is also a timestamp associated with each username which allows the association between the IP address and that particular user account to be broken after a period of time determined in the config file. The number of users associated with a given IP before flagging the administrator can also be set via the config file.

User Map

[WRITE HERE WHEN FIXED]

ALGORITHMS

[talk about algorithms used]

PARADIGMS

[talk about online processing as a paradigm and why this was a necessary approach to the problem]

or was it batch processing?

FUTURE WORK

[talk about future work that could be done for this project and projects that could take inspiration from this one]