Intrusion Detection: An Analysis of User Behavior

Stefan Maurer, Hiram College, Hiram Ohio, USA

Problem and Motivation:

No matter how hard we try, there is no definite way to insure a computer's absolute security. The best we can offer are ways to increase a computer's security, just as we have been trying to do since the creation of the password. However, all forms of security have their flaws. Passwords, of course, lose their effectiveness as soon as they are compromised. Another form of security is the use of physical measures, such as keys or magnetic cards, which are obviously not very secure as they can be lost or stolen. As these two main forms of security can be easily compromised, a more secure form of security is needed. As a result, biometric security was born; the method used to authenticate a specific individual as such. Biometric authentication has many applications, such as ear recognition, fingerprint recognition, retina recognition, and voice recognition [5]. Recognition, although fairly effective, enables authentication, but only when the user is gaining access to a building, computer, or a particular room. Once authentication occurs, there is no guarantee that the user that gained initial authentication is still the same person. Host-based intrusion detection systems attempt to remedy this dilemma by providing continuous authentication once a user has logged onto a computer. The developed system as described here performs behavior analysis on the current user as to authenticate the user, or identify them as an intruder.

Background & Related Work

In the past, there have been many approaches to creating an effective intrusion detection system, particularly concerning the methods used to analyze a user's behavior. Regardless of the method used, however, every intrusion detection system should comply with several attributes of a "good" intrusion detection system. These attributes include run-time efficiency, ease of use, system security, interoperability between systems, and transparency of use [1]. Our system, although only available for simulated use, either already supports each of these attributes, or would be easily adapted to do so. The intrusion detection system described in this work is able to operate efficiently without the user's knowledge, as well as only require the user to interact with the system upon installation and when an intruder has been detected – thus fulfilling the efficiency, ease of use, and transparency requirements of a "good" intrusion detection system [1]. The interoperability of our system is able to be maintained so long as the system is given permission to monitor and record user activity, as well as prevent intruders from additional activity. Finally, so long as the intruder is not aware of the intrusion detection system and the recorded behavior is kept encrypted, our system remains secure.

The system developed here uses a bioinformatics technique, the Smith-Waterman alignment algorithm, to perform behavior analysis. However, it was not the first to do so. "Intrusion Detection: A Bioinformatics Approach" and "Sequence Alignment for Masquerade Detection" by Scott E. Coull, Joel W. Branch, Boleslaw K. Szymanski, and Erik A. Beimer describe the implementation and results of a Smith-Waterman based intrusion detection system using sequential audit data as behavioral data, and provided the inspiration to use an alignment algorithm for behavior analysis. Other approaches to intrusion detection include behavior anomaly detection using a statistical approach, as seen in "Temporal Signatures for Intrusion Detection" by Anita Jones and Song Li. Anita Jones and Song Li analyzed audit data just as in the previous work, but concerned themselves with the use of temporal data as well. Yet another approach to host-based intrusion detection is to detect attacks, rather than trace the behavior of the user, as was used in "Improved Offline Intrusion Detection using a Genetic Algorithm" by Pedro A. Diaz-Gomez and Dean F. Hougen. Although each of these approaches to intrusion detection is unique, there are many other methods for host-based detection, all of which cannot be described here.

Uniqueness of Approach

As shown in the previous section, many host-based intrusion detection systems use behavior data obtained from command line audit data. However, some portion of this behavior does not represent the user's actions so much as the patterns created by a variety of processes run by the system. For this reason, our intrusion detection system uses a sequential list of process use while a particular user is logged in. Specifically, the system was designed to monitor the list of currently

running processes, and record when either a new process has begun executing, or has terminated. We believe that recording and monitoring behavior in this fashion will yield a better representation of a user's behavior, particularly concerning the *order* in which the user began and ceased the use of processes. Of course, there are some processes that the user does not control directly, though these processes seem to have little effect on the accuracy of our intrusion detection system.

In order to determine how similar two behaviors are, the Smith-Waterman alignment algorithm is used. This algorithm is given parameters for the alignment, such as the benefit to the alignment score for matching two individual actions within a behavior, as well as an alignment cost for mismatching two actions [7]. In addition, the algorithm is able to insert gaps into both the current and recorded behavior in order to find the alignment of actions within the behaviors as to produce the maximum final alignment score [7]. There is also a cost to inserting gaps in either behavior list. This bioinformatics algorithm is primarily used to align two sequences of nucleotides or amino acids, but was adapted for use in the intrusion detection system [3]. The first adaptation made was to not penalize the behavior alignment score for use of "padding" gaps. The behavior representation of the current user is generally small (less than 25 actions in length), whereas the past behavior list is fairly large (hundreds in length). During alignment, the Smith-Waterman algorithm is required to insert gaps into either behavior list until they are the same length [3]. Any gap used only to increase the length of either behavior is considered a "padding" gap, and does not penalize the final alignment score. Without this modification, the alignment score would be almost completely influenced by these gaps, causing the alignment score to become consistently negative [3]. The second modification made was to penalize gaps inserted into the recorded behavior differently than those inserted into the current behavior. The reasoning behind this decision lies in the meaning behind the insertion of a gap into the recorded behavior list. When used in the field of bioinformatics, neither sequence is any more relevant than the other. In this application, however, inserting a gap into the recorded behavior is essentially allowing for use of some irregular action within the current behavior. As the current behavior generally complies with the past behavior, a larger penalty is given to a gap in the past behavior than in the current behavior [3].

In addition to attempting to make the behavior analysis portion of the system as effective as possible, a great deal of thought was given to the internal mechanisms of the system as to maintain the security of the system without sacrificing transparency. As is the case with any system charged with making an intelligent decision, minimizing false-positives and false negatives is a priority. In the present situation, false-negatives are certainly dangerous as they allow an intruder full user access to a computer. A high false-positive rate, however, may cause the frustrated user to remove the intrusion detection system from their computer completely, as it represents the rate of locking the user out of the system incorrectly. In order to help prevent a high-false positive rate, we have designed our intrusion detection system with the ability to detect intruders "gradually", as opposed to deciding the current user is an intruder due to a small change in behavior. This method allows for the user to occasionally use an application or process that they may not regularly use without being flagged as an intruder. Of course, the system does not allow for any substantial change in behavior, as using completely different sets of applications and processes would certainly indicate the presence of an intruder.

Another method used to avoid a high false-positive rate while maintaining security was to selectively update the user's recorded behavior. Over time, a user's behavior is likely to change, but not so drastically that they will begin to use completely different applications in sequence. In order to reflect a gradual change in behavior, the system will incorporate the legitimate user's behavior into the recorded behavior so long as an intruder is not detected. Furthermore, as to not allow for behavior no longer adhered to by the legitimate user to be used during behavior alignment, the system only considers more recent recorded behavior. In this way, we are able to further decrease the false-negative rate by reducing the chances an intruder's behavior will correspond to the legitimate users distant past behavior. Disregarding distant past behavior also decreases the ability of the legitimate user's behavior to align well during Smith-Waterman alignment, thus increasing the false-positive rate slightly. However, if we only disregard recorded behavior that is sufficiently distant from the current behavior, this increase in false alerts seems to become negligible.

Due to our specific type of recorded behavior, raw data was required before our system could be optimized and evaluated for effectiveness. Six users, using multiple operating systems, volunteered to allow their process data to be recorded over a period of approximately two and a half weeks. Once data had been collected from each of the six users, the system could be optimized using the data for training. Among various optimization techniques, such as direct optimization, hill climbing, and simulated annealing, a genetic algorithm was chosen for optimization in accordance with other intrusion detection systems that had been optimized in a similar manner [4]. In order find optimal sets of parameters for the

intrusion detection system, five files at a time were used to train the system, followed by the use of the singular remaining file for testing. In order to determine how well the intrusion detection system was able to differentiate the intruder from the correct user, portions of all six behaviors were chosen (18 actions in length), and analyzed against each of the remaining behaviors in segment sizes decided by the genetic algorithm. In order to determine how well the system avoided flagging the correct user as an intruder, the portions of the sixth behavior were analyzed against the remaining portion of the current behavior.

Results and Contributions:

Results:

Upon optimization of the intrusion detection system, we were able to obtain accurate results while maintaining low false-positive and false-negative rates. After using each combination of five training behaviors and one test behavior, we achieved the following results. Of 10,230 tests, 9089 were true positives, 211 were false negatives, 879 were true negatives, and 51 were false positives, thus yielding an overall 97.43% accuracy percentage, with a 97.73% true positive rate and a 94.51% true negative rate. Based on these results, it is clear that the intrusion detection system is capable of accurate results, while maintaining a respectable true-positive and true-negative rates. Keeping in mind that some of the behavior used to determine the true-negative rate may have been extremely anomalous, the slightly lower true-negative rate becomes more acceptable. In some cases it may be impossible for this method to accurately analyze severely abnormal behavior, and the user should expect to be flagged as an intruder.

It should be noted that although supplementary data would enable additional testing to find added optimal parameters for the intrusion detection system, the addition of too much supplementary behavior data would cause an increase in our false-negative rate. As the number of behaviors increases, we are bound to find a pair of people with very similar behaviors, thus allowing an intruder continuous access to the system.

Future Work:

Although the above results indicate that the intrusion detection system is effective, there is still much room for improvement. By collecting different types of data from the legitimate user as he or she uses the computer, we may be able to create a denser behavior set, effectively adding another dimension to the behavior analysis. Additional behavior sets may include specific temporal data concerning the use of processes, particular web sites visited using an internet browser, and the detection of new or malicious activity. For example, temporal data could be used to analyze how often the user keeps each process running, when they are likely to be using the computer, as well as the time lapse between each individual application use. Web site data, on the other hand, could easily be represented in a manner similar to the process behavior, and consequently be analyzed using the Smith-Waterman algorithm as well. Perhaps with just one or two of these advancements, the system developed here could be expanded to run as a trial intrusion detection system on a number of computers.

Contribution:

As our system was certainly neither the first to use a Smith-Waterman algorithm [2][3] for behavior analysis or the first to use process data as our behavior source, we believe our implementation to be unique. Most particularly, our intrusion detection system employs several methods to avoid false positives on behalf of the legitimate user, effectively reducing the risk of being uninstalled by the user due to annoyance. In any case, we feel that the research done here provides significant indication as to the effectiveness of process behavioral data as a means to identify intruders. Furthermore, by incorporating other forms of already-existing intrusion detection systems, it would seem reasonable to predict a dramatic increase in overall accuracy.

Acknowledgements

The author would like to acknowledge the advisor for this research, Dr. Ellen Walker, Hiram College, as well as all of the volunteers that made the development of this system possible.

References

[1] Axelsson, Stefan. "On a Difficulty of Intrusion Detection". Goteborg, Sweden, n.d.

[2] Coull, Scott E., et al. "Sequence Allignment for Masquerade Detection." 2004.

[3] Coull, Scott, et al. Intrusion Detection: A Bioinformatics Approach. New York, n.d.

[4] Diaz-Gomez, Pedro A. and Dean F. Hougen. "Improved Off-Line Intrusion Detection using a Genetic Algorithm." 2005.

[5] Jain, Anil K., Arun Ross and Salil Prabhakar. "An introduction to biometric recognition." IEEE Trans. on Circuits and Systems for Video Technology 14 (2004): 4-20.

[6] Li, Song and Anita Jones. "Temporal Signatures for Intrusion Detection." 2001.

[7] Smith, T F and M S Waterman. "Identification of Common Molecular Subsequences." Journal of Modern Biology (1981): 195-197.