Unsupervised Methods for Detecting a Malicious Insider

Matt Wolff
University of Hawaii
wolffm@hawaii.edu

ABSTRACT

One way a malicious insider can attack a network is by masquerading as a different user. Various algorithms have been proposed in an effort to detect when a user masquerade attack has occurred. In this paper, two unsupervised algorithms are proposed with the intended goal of detecting user masquerade attacks. The effectiveness of these two unsupervised algorithms are then compared against supervised algorithms.

Keywords

User masquerades, insider threat, unsupervised learning, network security.

INTRODUCTION

In many organizations, the most devastating form of attacks against a computer network are not from the elite outside hacker, but from trusted entities that belong to the internal organization. 68% of respondents to a McAfee survey identified the insider threat as the biggest risk to their organizations information (McAfee, 2009). The insider can be considered extremely dangerous, due to an in-depth knowledge of the target and trusted access to vital information on the network (Randazzo et al., 2004).

There have been significant advances in computer security research dealing with the area of external threats. Anti-virus software, firewalls, intrusion detection systems, and other products have been successful as repelling external threats to a computer network. However, there has not been similar successful research in the area of internal threats. This may be due to the visibility and higher profile of external attacks, garnering more research attention then the insider attack (Blackwell, 2009).

One of the research areas pertaining to the insider threat is in the development of automated detection systems that can determine when a user account has been used in a malicious manner. For the purpose of this paper, we define malicious account activity as either an account whose credentials have been compromised, or an account whose legitimate user is the malicious insider knowingly conducting the threatening activity on the account. We then define a user masquerade as a subset of malicious activity, specifically the instances where a user's credentials have been compromised and utilized by a different user to access the legitimate users account. There are various motivations for a malicious actor to use a different user account; to evade detection, to access information that they do not have access to, to sabotage a colleague, or other reasons (Wood, 2000).

PREVIOUS WORK

User masquerade algorithms

There have been various approaches toward detection of this type of activity, with a majority of the previous research focusing on host-based detection systems (Bertacchini and Fierens, 2009). Most host-based detection systems are based on the assumption that a masquerading user will behave differently compared to the legitimate user of the account. In host-based system, data can be collected from a variety of data points to create a user profile. The user profile is then used as a standard upon which all new account activity is tested against. In theory, the difference in the behavior of the masquerading user will be statistically significantly compared to the user profile. Therefore many of the host-based detection systems construct algorithms in an effort to find anomalous user activity, with the idea that if the data is anomalous compared to the user history, then the data is from a masquerading user, as opposed to the legitimate user.

Reviewing Statement: This paper represents work in progress, an issue for discussion, a case study, best practice or other matters of interest and has been reviewed for clarity, relevance and significance.

While there are a variety of datasets available for testing user masquerade detection algorithms, the most popular data set is from Schonlau et al. (2001) (Bertacchini and Fierens, 2009). In the Schonlau et al. (2001) paper the authors collected the names of Unix system calls for a group of users to construct a user profile, from which they tested a variety of algorithms to detect the malicious users. Bertacchini and Fierens (2009) provides a thorough review of the various datasets and detection algorithms that have been applied to user masquerade detection.

Additional techniques and systems have been researched to detect insider threats on a network. Bowen et al. (2009) developed a decoy document system and believability rating for fake documents, and worked on a complete system, known as the RUU project. Maloof and Stephens (2007) describes the design and results from the ELICIT insider detection system. Kirkpatrick et al. (2009) introduce the Contextually Adaptive INsider threat architecture (CAIN), which incorporates contextual and risk-based access control with anomaly detection. Kruegel et al. (2003) proposed system finds intrusions by viewing system call parameters and checking that the parameters are valid. Mathew et al. (2008) develop a static analysis tool called ICMAP to periodically construct CAGs which are then analyzed to uncover possible attacks. Myers et al. (2009) look at detecting malicious insiders who exploit internal web servers. Nguyen et al. (2003) looks at insider hacking of a system via system calls. Ray and Poolsapasit (2005) propose a framework that uses an attack tree to identify malicious activities from authorized insiders. Rocke and Demara (2006) present the CONFIDANT file integrity verification framework focusing on insider defense aspects. Santos et al. (2008) designed a framework that minimized the impacts from cognitive styles. The evaluation showed that four out of five simulated malicious insiders were successfully detected. Schultz (2002) proposed framework defines relevant types of insider attack-related behaviors and symptoms—"indicators" that include deliberate markers, meaningful errors, preparatory behaviors, correlated usage patterns, verbal behavior and personality traits. Spitzner (2003) introduces the idea of honeypots for catching insiders that are actively seeking information. Thompson's (2004) approach focuses on modeling the insider through the insider's interactions with the document control system and textual models of the insider's task, queries, documents accessed, and work product. Calandrino et al (2007) proposed an Intelligent Insider Threat Detection (I2TD) system for monitoring and evaluating insider behavior to detect potentially malicious or otherwise undesirable activity. Ha et al. (2007) ICMAP tool presented in this paper is very effective at modeling insider threats, analyzing vulnerabilities and evaluating sensor deployment locations. Maybury et al. (2005) constructed a complete system by bringing together different parties interested in the insider threat area.

Dataset

For the collection of user data in the Schonlau data set, the authors collected Unix system calls over a 6 month span for 50 separate users, resulting in a collection of 15000 system calls per user. The authors then split the data into a training set and test set. The training set, unique for each user, consist of the first 5000 calls recorded for each user. The remaining 10000 commands were split into blocks of 100 commands each, thus constructing a test set of 100 blocks for each user. One each user test set was constructed, the authors synthesized user masquerades by replacing some of the blocks in certain users test data with blocks of commands from different users (Schonlau et al., 2001).

NEW TECHNIQUES

Issues with the dataset

While the Schonlau data set can provide a metric for testing user masquerade detection algorithms, it relies on the use of training data for training the various algorithms that have been implemented. If an organization were to attempt to construct their own data set, in creating the training data they would need to assert that the training data is free of user masquerades. This may be feasible in a small, controlled environment, but considering the network of corporations, universities, or other organization, it may not be possible to construct a training data set, or user profile, for each user while asserting that the data collected is free of user masquerades. For algorithms that assume the training data is free of masquerades, and based on this assumption classify future blocks of data, then it is likely that the classification may not prove to be as accurate as initially intended. It is possible if simply a small portion of the training data contains user masquerades, it could skew the results of the entire classification, reducing the effectiveness of the classification algorithm.

When considering the issue of possible training data manipulation by user masquerades, combined with the difficultly of trying to assemble training data sets for every user in a network, it is likely that the use of unsupervised techniques would be warranted in this situation. By using unsupervised techniques, the issue of training data poisoning by masquerade data is minimized in situations where a solid majority of the user data is

legitimate information (i.e. not data resulting from a user masquerade). Using unsupervised techniques also removed the problem of having to construct a clean training data set for user masquerade detection.

For the purpose of unsupervised user masquerade detection, I have designed two new techniques for user profile generation. Using the Schonlau data set, the unsupervised techniques construct a user profile, which is a collection of data that represents the standard usage pattern of the user. The user profile is then used to classify all remaining blocks of data as either legitimate user data or as user masquerades.

Minimum Distance

The first technique proposed for user profile generation is Minimum Distance. The premise behind the Minimum Distance profiling algorithm is to find those blocks of data which are most similar to all other blocks in the individual users data set. The Minimum Distance for a data block b, d_b can be calculated as follows

$$d_b = \sum_{i=0}^{150} \sum_{k=1}^{K} | n_{kb} - n_{ki} | \quad (1)$$

Once d_b has been calculated for each block, the user profile is then constructed using the blocks whose d_b value are below a set threshold θ.

Symbol	MEANING
K	Total number of distinct commands
n_{kb}	Number of times command k is in block b
n_{ki}	Number of times command k is in block i
U_k	Number of users who have k in the user profile
U	Total number of users
N_{uk}	Number of times user u has command k in user profile
N_u	Length of users u user profile

Table 2. Explanation of symbols in various formulas.

Compact Cluster

The second technique for unsupervised user profile generation is the Compact Cluster. In this technique, we set

$$m_b = d_{b_{\min}}$$
$$p_u = \{b_0, b_1, ..., b_n\} \quad (2)$$

Where $b_0 = m_b$ and b_n = *nth* farthest block from m_b. The Compact Cluster will find the data block whose total distance from every other block is minimal. Once this block is found, the θ-closest blocks to m_b are chosen to form the user profile, where θ is the threshold value.

Applying to the data set

From previous work on this data set, there exists a collection of supervised algorithms for classification of the data. In an effort to test the effectiveness of the unsupervised user profile generation, I combined the user profile generation algorithms with the Uniqueness algorithm presented in the Schonlau et. al. paper. I will provide a brief summary of the algorithm here, please consult (Schonlau et al., 2001) for a more thorough explanation.

The Uniqueness algorithm works as follows

$$x_u = \frac{1}{n_u}\sum_{k=1}^{K} W_{uk}(1-U_k/U)n_{uk}$$
$$W_{uk} = \begin{cases} 1 \; if \; k \; is \; in \; profile \\ (N_{uk}/N_u)/\sum_u (N_{uk}/N_u) \; otherwise \end{cases} \quad (3)$$

We make some small adjustments to the Uniqueness algorithm to suit our testing. First, we remove the updating technique. Since our user profiling algorithms already pass through all of the data when generating a profile, there is no need for dynamic updating of the profile during classification. If we were only using a subset of the data set to generate a user profile, then there would be a case for reinstating an updating mechanism into the classification algorithm. The second change removed the reliance on the training data in the data set, and instead have substituted the training data with the user profile generated by the Minimum Distance and Compact Clutser algorithms.

RESULTS

In generating results from our tests with the unsupervised user profile generation techniques, we attempted to minimize all other variables aside from user profile generation. To do this, we ran four separate variations of the Uniqueness algorithm:

- Uniqueness algorithm utilizing the training data provided in the data set.
- Uniqueness algorithm with the Minimum Distance user profile technique.
- Uniqueness algorithm with the Compact Cluster user profile technique.
- Uniqueness algorithm with a randomly generated user profile.

For the user profile generation techniques, we employed a θ value of 5000 commands. This gave the user profiles the same size as the training set in the data used by the supervised algorithms. Using these profiles, we then tested the results by classifying the first 50, 150, 300, and 400 most anomalous blocks and compared those results to other techniques. Figure 1 displays the classification results with blue representing the percentage of correctly classified masquerading blocks and red displaying the percentage of blocks that were misclassified as masquerades. The larger the percentage of blocks identified as masquerades, the better the performance of the classification algorithm.

For small to medium classification thresholds, the supervised Uniqueness algorithm generated better results than the three unsupervised user profile generation techniques, as expected. However, for the largest classifying

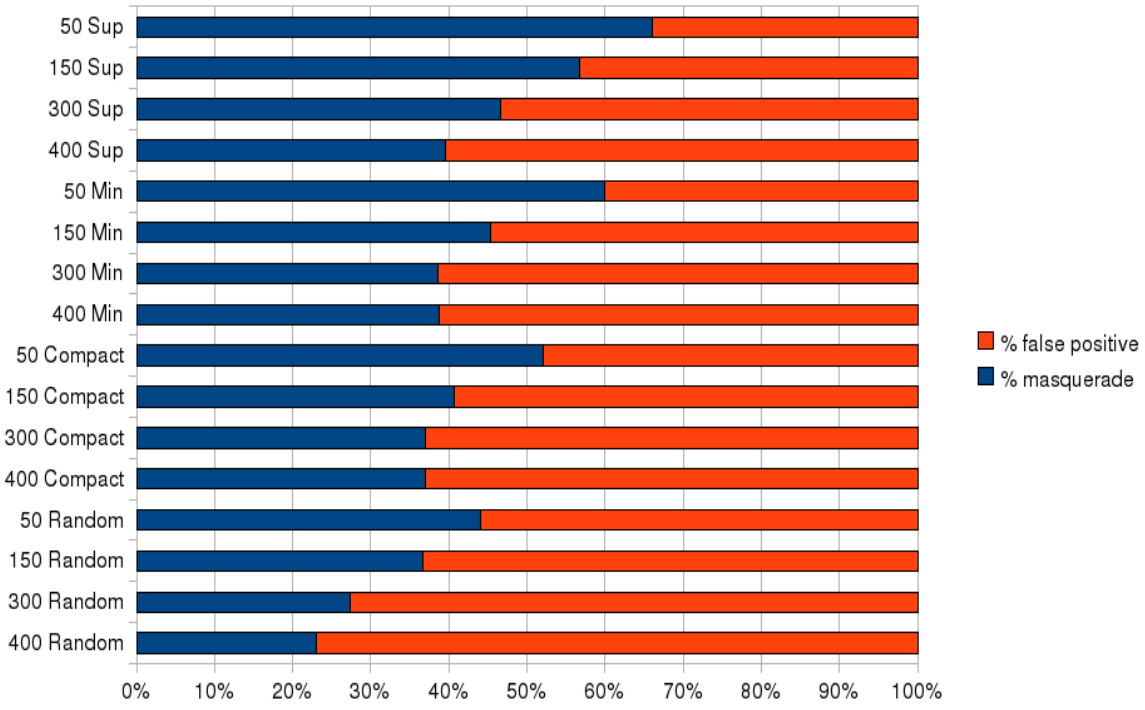

Figure 1. % of correct classifications to incorrect classification for each of the four techniques

threshold, 400 anomalous blocks, the Minimum Distance unsupervised algorithms produced results similar to the supervised Uniqueness algorithm.

CONCLUSION

In the case of user masquerade detection, unsupervised algorithms can be used to avoid having to generate a training data set that is guaranteed to be free of malicious activity. Unsupervised algorithms also take into account all of a users history when generating a user profile, while the supervised algorithm may initially only take in a predetermined subset of data for training. These benefits for unsupervised algorithms in this situation may create a reasonable case in certain situations for their use in determining user masquerades.

REFERENCES

1. Bertacchini, M., and Fierens, P.I.(2009) A Survey on Masquerader Detection Approaches.
2. Blackwell C. (2009) A security architecture to protect against the insider threat from damage, fraud and theft, Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research Cyber Security and Information Intelligence Challenges and Strategies - CSIIRW '09, 1.
3. Bowen, B., Hershkop, S., Keromytis, A, and Stolfo, S. (2009) Baiting Inside Attackers Using Decoy Documents, Columbia University Department of Computer Science Technical Report, CUCS-016-09, 09.
4. Bowen, B., Salem, M. B., Hershkop, S., Keromytis, A, and Stolfo, S (2009) Designing Host and Network Sensors to Mitigate the Insider Threat, IEEE Security & Privacy Magazine, 7, 22-29.
5. Calandrino, J., McKinney, S., and Sheldon, F. (2007) Detection of undesirable insider behavior, Cyber Security and Information Intelligence Research Workshop (CSIIRW), 1-4.
6. Ha, D., Upadhyaya, S., Ngo, H., Pramanik, S., Chinchani, R., and Mathew, S. (2007) Insider threat analysis using information-centric modeling, International Federation For Information Processing-Publications-IFIP, 242, 55.
7. Kirkpatrick, M., Bertino, E., and Sheldon, F. (2009) An Architecture for Contextual Insider Threat Detection, cs.purdue.edu, 1-11.
8. Kruegel, C., Mutz, D, Valeur, F., and Vigna, G (2003) On the detection of anomalous system call arguments, Lecture Notes in Computer Science, 2808, 326–343.
9. Maloof, M., and Stephens, G. (2007) ELICIT: A System for Detecting Insiders Who Violate Need-to-know, Lecture Notes in Computer Science, 4637, 146.
10. Mathew, S., Upadhyaya, S., Ha, D., and Ngo, H. (2008) Insider Abuse Comprehension through Capability Acquisition Graphs, Information Fusion, 11th International Conference on, 1–8.
11. Maybury, M., Chase, P., Cheikes, B., Brackney, D., Matzner, S., Hetherington, T., Wood, B., Sibley, C., Marin, J., Longstaff, T., Spitzner, L., Haile, J., Copeland, J., and Lewandowski, S. (2005), Analysis and detection of malicious insiders, MITRE CORP BEDFORD MA
12. McAfee (2009). Unsecured Economies: Protecting Vital Information, at http://resources.mcafee.com/content/NAUnsecuredEconomiesReport
13. Myers, J., Grimaila, M.R., and Mills, R.F. (2009) Towards insider threat detection using web server logs, Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research Cyber Security and Information Intelligence Challenges and Strategies - CSIIRW '09, 1.
14. Nguyen, N., Reiher, P., and Kuenning, G. (2003) Detecting insider threats by monitoring system call activity, IEEE Systems, Man and Cybernetics Society Information Assurance Workshop,2003., 45-52.
15. Randazzo, M., Keeney, M., Kowalski, E., Cappelli, D., and Moore, A. (2005) Insider hreat study: Illicit cyber activity in the banking and finance sector. U.S. Secret Service and CERT Coordination Center/Software Engineering Institute: Philadelphia, PA. 25
16. Ray, I and Poolsapassit, N. (2005) Using Attack Tress to Identify Malicious Attacks from Authorized Insiders, ESORICS 2005, 231-246.
17. Rocke A. and DeMara, R.(2006) Mitigation of insider risks using distributed agent detection, filtering, and signaling, International Journal of Network Security, 2,141–149.
18. Santos Jr., E., Nguyen, H., Yu, F., Kim, K., Li, D., Wilkinson, J. T., Olson, A., and Jacob, R. (2008) Intent-Driven Insider Threat Detection in Intelligence Analyses, 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 345-349.

19. Schonlau, M, Vardi, Y, Theusan, M, Karr, A.F, Ju, W, and DuMouchel, W (2001) Computer Intrusion: Detecting Masquerades, Statistical Science, 16, 58-74.
20. Schultz, E. (2002) A framework for understanding and predicting insider attacks, Computers & Security, 21, 526–531.
21. Spitzner, L. (2003) Honeypots: Catching the insider threat, Annual Computer Security Applications Conference, 03,170-179.
22. Thompson, P. (2004) Weak models for insider threat detection.
23. Wood, B. (2000) An insider threat model for adversary simulation, SRI International, Research on Mitigating the Insider Threat to Information Systems, 2, 1-3.