Ashwin Narkar

CS 111: Dis 1D

Professor Eggert

15 March 2019

HIDS: Host-based Intrusion Detection System

As technology grows and computers become more and more pervasive in all aspects of our lives, the issue of security of these machines comes into consideration. We see various different methods of attacks of these machines, from unauthorized access to tampering with private personal data. Attackers can exploit small bugs in code to attack these machines or manipulate users through social engineering to trick them into granting them access that was otherwise not granted. To protect users from these attackers, various measures have been put in place at various levels of software. These protections permeate all levels from the operating system to application level authentication. Researchers from the University of Technology in Sydney and Shanhai Jiao Tong University have developed Host-Based Intrusion Detection System (HIDS) with System Calls primarily for embedded systems.

Currently, one of the systems used for security is NIDS which is Network Intrusion Detection System. This system is established to check communication between networks, such as data mining to detect problems and potential threats. The system has been used to detect a denial of service attack using multivariate correlation. This system has its limitations however. It is usually implemented using hardware devices on edges of a network. This means while the system is good at detecting threats from the outside, it struggles to find threats within an Intranet. It also will add overhead and could pose a bottleneck for network flow.

HIDS, the topic of this paper, uses a different approach. It monitors system call logs to detect anomalies which could potentially be threats. It uses similar concepts in algorithms in NIDS to analyze system call sequences. By training the system to detect threatening sequences, the system can be aware of potential security breaches. HIDS has become very popular because there have been a lot of attacks on Linux servers recently. Challenges to HIDS include performance in training and analyzing convoluted system call logs. For NIDS, one would just have to install hardware on a network, but in HIDS, every host has to be updated with the software. This means that every host incurs this overhead. Another method called CIDS, collaborating intrusion detection system, involves combining HIDS, NIDS, and other methods together in a hybrid method.

Intrusion detections systems have to account for detecting two different types of attacks. One type of attack is the misuse attack. In this model, the system has a list of known attack signatures and abnormal behaviors. This list is defined by system administrators. This method is very good at detecting known forms of attack. It is limited by the obvious fact that it can only detect known signatures. It also has a very high miss rate for detecting attacks for the same reason. An alternate, more effective method is using anomaly detection. This is the system that NIDS and HIDS use as discussed previously. The NIDS system uses complex surveying of network traffic to identify anomalies in communications. The HIDS system analyzes system call sequences to identify problematic and threatening sequences of system calls. One of the biggest challenges for anomaly detection is having a high false alarm rate. As new system arise, various new system call sequences arise that do not often pose a problem but appear as new to the system. This means the system might classify them as an anomaly.

The main mechanism for HIDS is the system call trace analysis. By tracing the various system calls that a program makes, its primary functions can be analyzed. Because system calls invoke kernel level code and privileged instructions, an intrusive program might make certain patterns of system calls to gain control of the system or violate security measures that are in place. Using a sliding time window, HIDS can analyze sequences of *n* system calls. HIDS begins by taking normal system call sequences from various programs and creating a database of these. Then, when analyzing a sequence, it computes the Hamming distance from this sequence to all the sequences in the database and finds the minimal distance. Hamming distance is concept used in coding theory and communications where it takes the number of elements that differ between two sequences. Finding the minimal Hamming distance is common operation done for coding theory, for example in convolutional codes in a channel, and in HIDS for anomaly detection. Some problems with Hamming distance and database generation involve time and complexity. Finding minimal Hamming distance would involve checking every single sequence in the database, and every time a program is updated it would have to generate a new sequence for normal sequences in case new system calls are added.

Basically the new HIDS system developed by these researchers involves training a machine learning model to analyze sequences of system calls within a host instead of over a network. Using various complex machine learning models and generating datasets, they have worked to create an efficient and reliable HIDS. Various tools exist for creating system call traces, such as strace. To adapt the technology for embedded systems, one has to account for the smaller processing power and other considerations such as power and speed. On an embedded system, a smaller processor means less computational power to run complex machine learning algorithms and having to worry about things like battery usage and heat. Research has also found ways to use embedded system hardware to ease the computational power and run detection offline, using FPGAs and other hardware technology. Combining these system calls with cloud computing has also shown to be effective, when embedded systems send information to a cloud to do analysis of system calls.

The challenges and improvements in HIDS involve improving the machine learning models to increase the accuracy of detecting intrusions. This goes beyond the scope of operating systems, however. The main point is that using system call traces is an effective way to detect intrusions because anomalies can be seen when programs try to ask the kernel to do certain things that could indicate malicious intent. HIDS is more thorough in analyzing intrusions when compared to NIDS because it can detect intrusions within a network or even within a single system because system call logs are unique to a single system. It is a very important technology. As security becomes more fundamental in today’s interconnected and networked world, computer security because a crucial issue that must be addressed in every system that is connected to the worldwide network.

Works Cited

<http://delivery.acm.org/10.1145/3220000/3214304/a98-liu.html?ip=131.179.54.52&id=3214304&acc=ACTIVE%20SERVICE&key=CA367851C7E3CE77%2E79535EF926D6BC05%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1552715813_1551d19a260fcc1aa4bde2c40cc1f273>