Intrusion Detection Systems

H. Gunes Kayacik, A. Nur Zincir-Heywood, Malcolm I. Heywood
Dalhousie University, Faculty of Computer Science
(Canada)

INTRODUCTION

Along with its numerous benefits, the Internet also created numerous ways to compromise the security and stability of the systems connected to it. In 2003, 137529 incidents were reported to CERT/CC © while in 1999, there were 9859 reported incidents (CERT/CC©, 2003). Operations, which are primarily designed to protect the availability, confidentiality and integrity of critical network information systems, are considered to be within the scope of security management. Security management operations protect computer networks against denial-of-service attacks, unauthorized disclosure of information, and the modification or destruction of data. Moreover, the automated detection and immediate reporting of these events are required in order to provide the basis for a timely response to attacks (Bass, 2000). Security management plays an important, albeit often neglected, role in network management tasks.

Defensive operations can be categorized in two groups: static and dynamic. Static defense mechanisms are analogous to the fences around the premises of a building. In other words, static defensive operations are intended to provide barriers to attacks. Keeping operating systems and other software up-to-date and deploying firewalls at entry points are examples of static defense solutions. Frequent software updates can remove the software vulnerabilities, which are susceptible to exploits. Firewalls provide access control at the entry point, they therefore function in much the same way as a physical gate on a house. In other words, the objective of a firewall is to keep intruders out rather than catching them. Static defense mechanisms are the first line of defense, they are relatively easy to deploy and provide significant defense improvement compared to the initial unguarded state of the computer network. Moreover they act as the foundation for more sophisticated defense mechanisms.

No system is totally foolproof. It is safe to assume that intruders are always one step ahead in finding security holes in current systems. This calls attention to the need for dynamic defenses. Dynamic defense mechanisms are analogous to burglar alarms, which monitor the premises to find evidence of break-ins. Built upon static defense mechanisms, dynamic defense operations aim to catch the attacks and log information about the incidents such as source and nature of the attack. Therefore, dynamic defense operations accompany the static defense operations to provide comprehensive information about the state of the computer networks and connected systems.

Intrusion detection systems are examples of dynamic defense mechanisms. An intrusion detection system (IDS) is a combination of software and hardware, which collects and analyzes data collected from networks and the connected systems to determine if there is an attack (Allen *et. al.*, 1999). Intrusion detection systems complement static defense mechanisms by double-checking firewalls for configuration errors, and then catching the attacks that firewalls let in or never perceive (such as insider attacks). IDSs are generally analyzed from two aspects:

- IDS Deployment: Whether to monitor incoming traffic or host information.
- Detection Methodologies: Whether to employ the signatures of known attacks or to employ the models of normal behavior.

Regardless of the aspects above, intrusion detection systems correspond to today's dynamic defense mechanisms. Although they are not flawless, current intrusion detection systems are an essential part of the formulation of an entire defense policy.

DETECTION METHODOLOGIES

Different detection methodologies can be employed to search for the evidence of attacks. Two major categories exist as detection methodologies: misuse and anomaly detection. Misuse detection systems rely on the definitions of misuse patterns, which are the descriptions of attacks or unauthorized actions (Kemmerer and Vigna, 2002). A misuse pattern should summarize the distinctive features of an attack and is often called the signature of the attack in question. In the case of signature based IDS, when a signature appears on the resource monitored, the IDS records the relevant information about the incident in a log file. Signature based systems are the most common examples of misuse detection systems. In terms of advantages, signature based systems, by definition, are very accurate at detecting known attacks, which are included in their signature database. Moreover, since signatures are associated with specific misuse behavior, it is easy to determine the attack type. On the other hand, their detection capabilities are limited to those within signature database. As the new attacks are discovered, a signature database requires continuous updating to include the new attack signatures, resulting in potential scalability problems.

As opposed to misuse IDSs, anomaly detection systems utilize models of the acceptable behavior of the users. These models are also referred to as normal behavior models. Anomaly based IDSs search for the deviations from the normal behavior. Deviations from the normal behavior are considered as anomalies or attacks. As an advantage over signature based systems, anomaly based systems can detect known and unknown (i.e. new) attacks as long as the attack behavior deviates sufficiently from the normal behavior. However, if the attack is similar to the normal behavior, it may not be detected. Moreover, it is difficult to associate deviations with specific attacks since the anomaly based IDSs only utilize models of normal behavior. As the users change their behavior as a result of additional service or hardware, even the normal activities of a user may start raising alarms. In that case, models of normal behavior should be redefined to maintain the effectiveness of the anomaly based IDS.

In today's intrusion detection systems, human input is essential to maintain the accuracy of the system. In the case of signature based systems, as new attacks are discovered, security experts examine the attacks to create corresponding detection signatures. In the case of anomaly systems, experts are needed to define the normal behavior. Therefore, regardless of the detection methodology, frequent maintenance is essential to uphold the performance of the IDS.

Given the importance of IDSs, It is imperative to test them to determine their performance and eliminate their weaknesses. For this purpose, researchers conduct tests on standard benchmarks (Kayacik, Zincir-Heywood, 2003; Pickering, 2002). When measuring the performance of intrusion detection systems, the detection and false

positive rates are used to summarize different characteristics of classification accuracy. In simple terms, false positives (or false alarms) are the alarms generated by a non-existent attack. For instance, if an IDS raises alarms for the legitimate activity of a user, these log entries are false alarms. On the other hand, detection rate is the number of correctly identified attacks over all attack instances, where correct identification implies the attack is detected by its distinctive features. An intrusion detection system becomes more accurate as it detects more attacks and raises fewer false alarms.

IDS DEPLOYMENT STRATEGIES

In addition to the detection methodologies, data is collected from two main sources: traffic passing through the network and the hosts connected to the network. Therefore, according to where they are deployed, IDSs are divided into two categories, those that analyze network traffic and those that analyze information available on hosts such as operating system audit trails. The current trend in intrusion detection is to combine both host based and network based information to develop hybrid systems and therefore not rely on any one methodology. In both approaches however, the amount of audit data is extensive, thus incurring large processing overheads. A balance therefore exists between the use of resources, and the accuracy and timeliness of intrusion detection information.

Network based IDSs inspect the packets passing through the network for signs of an attack. However, the amount of data passing through the network stream is extensive, resulting in a trade off between the number of detectors and the amount of analysis each detector performs. Depending on throughput requirements, a network based IDS may inspect only packet headers or include the content. Moreover, multiple detectors are typically employed at strategic locations in order to distribute the task. Conversely, when deploying attacks, intruders can evade IDSs by altering the traffic. For instance, fragmenting the content into smaller packets causes IDSs to see one piece of the attack data at a time, which is insufficient to detect the attack. Thus, network based IDSs, which perform content inspection, need to assemble the received packets and maintain state information of the open connections, where this becomes increasingly difficult if a detector only receives part of the original attack or becomes ‘flooded’ with packets.

A host based IDS monitors resources such as system logs, file systems, processor and disk resources. Example signs of intrusion on host resources are critical file modifications, segmentation fault errors, crashed services or extensive usage of the processors. As opposed to network based IDSs, host based IDSs can detect attacks, which are transmitted over an encrypted channel. Moreover, information regarding the software that is running on the host is available to host based IDS. For instance, an attack targeting an exploit on an older version of a web server might be harmless for the recent versions. Network based IDSs have no way of determining whether the exploit has a success chance, or of using a priori information to constrain the database of potential attacks. Moreover, network management practices are often critical in simplifying the IDS problem by providing appropriate behavioral constraints, thus making it significantly more difficult to hide malicious behaviors (Cunningham, Lippmann and Webster, 2001).

CHALLENGES

The intrusion detection problem has three basic competing requirements: speed, accuracy and adaptability. The speed problem represents a quality of service issue. The more analysis (accurate) the detector the higher the computational overhead. Conversely accuracy requires sufficient time and information to provide a useful detector. Moreover, the rapid introduction of both new exploits and the corresponding rate of propagation requires that detectors be based on a very flexible / scalable architecture. In today's network technology where gigabit Ethernet is widely available, existing systems face significant challenges merely to maintain pace with current data streams (Kemmerer and Vigna. 2002).

An intrusion detection system becomes more accurate as it detects more attacks and raises fewer false alarms. IDSs that monitor highly active resources are likely to have large logs, which in turn complicate the analysis. If such an IDS have high false alarm rate, the administrator will have to sift through thousands of log entries, which actually represent normal events, to find the attack related entries. Therefore, increasing false alarm rates will decrease the administrator's confidence in the IDS. Moreover, intrusion detection systems are still reliant on human input in order to maintain the accuracy of the system. In case of signature based systems, as new attacks are discovered, security experts examine the attacks to create corresponding detection signatures. In the case of anomaly systems, experts are needed to define the normal behavior. This leads to the adaptability problem. The capability of the current intrusion detection systems for adaptation is very limited. This makes them inefficient in detecting new or unknown attacks or adapting to changing environments (i.e. human intervention is always required). Although a new research area, incorporation of machine learning algorithms provides a potential solution for accuracy and adaptability of the intrusion detection problem.

CURRENT EXAMPLES OF IDS

Intrusion detection systems reviewed here are by no means a complete list but a subset of open source and commercial products, which are intended to provide readers different intrusion detection practices.

Snort: Snort is one of the best-known lightweight IDSs, which focuses on performance, flexibility and simplicity. It is an open-source intrusion detection system that is now in quite widespread use (Roesch, 1999). Snort is a network based IDS which employs signature based detection methods. It can detect various attacks and Probes including instances of buffer overflows, stealth port scans, common gateway interface attacks, and service message block system Probes (Roesch, 1999). Hence, Snort is an example of active intrusion detection systems that detects possible attacks or access violations while they are occurring (CERT/CC ©, 2001).

Cisco IOS (IDS Component): Cisco IOS provides a cost effective way to deploy a firewall with network based intrusion detection capabilities. In addition to the firewall features, Cisco IOS Firewall has 59 built-in, static signatures to detect common attacks and misuse attempts (Cisco Systems, 2003). The IDS process on the firewall router inspects packet headers for intrusion detection by using those 59 signatures. In some cases routers may examine the whole packet and maintain the state information for the connection. Upon attack detection, the firewall can be configured to log the incident, drop the packet or reset the connection.

Tripwire: When an attack takes place, attackers usually replace critical system files with their versions to inflict damage. Tripwire (Tripwire Web Site, 2004) is an open-source host based tool, which performs periodic checks to determine which files are modified in the file system. To do so, Tripwire takes snapshots of critical files. Snapshot is a unique mathematical signature of the file where even the smallest change results in a different snapshot. If the file is modified, the new snapshot will be different than the old one; therefore critical file modification would be detected. Tripwire is different from the other intrusion detection systems because rather than looking for signs of intrusion, Tripwire looks for file modifications.

FUTURE TRENDS

As indicated above, various machine learning approaches have been proposed in an attempt to improve on the generic signature based IDS. The basic motivation is to measure how close a behavior is to some previously established gold standard of misuse or normal behavior. Depending on the level of a priori or domain knowledge, it may be possible to design detectors for specific categories of attack (e.g. Denial of Service, User to Root, Remote to Local). Generic machine learning approaches include clustering or data-mining in which case the data is effectively unlabeled. The over-riding assumption is that behaviors are sufficiently different for normal and abnormal behaviors to fall into different 'clusters'. Specific examples of such algorithms include artificial immune systems (Hofmeyr and Forrest, 2000) as well as various neural network (Lee and Heinbuch, 2001; Kayacik, Zincir-Heywood and Heywood, 2003) and clustering algorithms (Eskin *et al.*, 2002).

Naturally the usefulness of machine learning systems is influenced by the features on which the approach is based (Lee and Stolfo, 2001). Domain knowledge that has the capability to significantly simplify detectors utilizing machine learning often make use of the fact that attacks are specific to protocol-service combinations. Thus, first partitioning data based on the protocol-service combination significantly simplifies the task of the detector (Ramadas, Ostermann and Tjaden, 2003).

When labeled data is available then supervised learning algorithms are more appropriate. Again any number of machine learning approaches have been proposed, including: decision trees (Elkan, 2000), neural networks (Hofmann and Sick, 2003) and genetic programming (Song, Heywood and Zincir-Heywood, 2003). However, irrespective of the particular machine learning methodology, all such methods need to address the scalability problem. That is to say, datasets characterizing the IDS problem are exceptionally large (by machine learning standards). Moreover, the continuing evolution of the base of attacks also requires that any machine learning approach also have the capability for on-line or incremental learning. Finally, to be of use to network management practitioners it would also be useful if machine learning solutions were transparent. That is to say, rather than provide "black box solutions", it is much more desirable if solutions could be reverse engineered for verification purposes. Many of these issues are still outstanding, with cases that explicitly address the computational overhead in learning against large datasets only just appearing (Song, Heywood and Zincir-Heywood, 2003).

C O N C L U S I O N

Intrusion detection system is a crucial part of the defensive operations, which complements the static defenses such as firewalls. Essentially, intrusion detection is searching for signs of attacks and when an intrusion is detected, intrusion detection system can take an action to stop the attack by closing the connection or report the incident for further analysis by administrators. According to the detection methodology, intrusion detection systems can be categorized as misuse detection and anomaly detection systems. According to the deployment, they can be classified as network based or host based although such distinction is coming to an end in today's intrusion detection systems where information is collected from both network and host resources. In terms of performance, an intrusion detection system gets more accurate as it detects more attacks and raises fewer false alarms.

R E F E R E N C E S

Allen, J., Christie, A., Fithen, W., McHugh, J., Pickel, J., & Stoner, E. (1999). State of the practice of intrusion detection technologies. CMU/SEI Technical Report (CMU/SEI-99-TR-028). http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html Retrieved on June 2004.

Bass T. (2000) *Intrusion Detection Systems and Multisensor Data Fusion*, Communications of the ACM, 43(4), 99-105.

CERT/CC © (2003). *Incident Statistics 1988-2003*, http://www.cert.org/stats/ Retrieved on June 2004.

CERT/CC © (2001), *Identifying tools that aid in detecting signs of intrusion*, http://www.cert.org/security-improvement/implementations/i042.07.html

Cisco Systems Inc. (2003), *Cisco IOS Firewall Intrusion Detection System Documentation*, http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/iosfw2/ios_ids.htm, Retrieved on Jun 2004

Cunningham R.K., Lippmann R.P., Webster S.E. (2001) Detecting and Displaying Novel Computer Attacks with Macroscope. IEEE Transactions on Systems, Man, and Cybernetics – Part A. 31(4) pp 275-280.

Elkan C. (2000), *Results of the KDD'99 Classifier Learning*. In ACM SIGKDD Explorations, volume 1, pages 63–64, 2000.

Eskin E., Arnold A., Prerau M., Portnoy L., and Stolfo S. (2002) A *Geometric Framework for Unsupervised Anomaly Detection: Detecting attacks in unlabeled data.* In D. Barbara and S. Jajodia, editors, Applications of Data Mining in Computer Security. Kluwer, 2002. ISBN 1-4020-7054-3, 2002.

Hofman A., Sick B. (2003) *Evolutionary Optimization of Radial Basis Function Networks for Intrusion Detection*. Proceedings of the International Joint IEEE-INNS Conference on Neural Networks. Pp 415-420.

Hofmeyr S.A., Forrest S. (2000) *Architecture for an Artificial Immune System*. Evolutionary Computation. 8(4) pp 443-473.

Kayacik G. and Zincir-Heywood N. (2003) *A Case Study Of Three Open Source Security Management Tools*. In Proceedings of International Symposium on Integrated Network Management 2003.

Kayacik G., Zincir-Heywood N., and Heywood M. (2003) *On the Capability of an SOM based Intrusion Detection System*. In Proceedings of International Joint Conference on Neural Networks, 2003.

Kemmerer R. A. and Vigna G. (2002), *Intrusion detection: A Brief History and Overview*, IEEE Security and Privacy, April 2002, pp.27-29.

Lee S.C., Heinhuch D.V. (2001) *Training a Neural-Network based Intrusion Detector to Recognize Novel Attacks*. IEEE Transactions on Systems, Man, and Cybernetics – Part A. 31(4) pp 294-299.

Pickering K. (2002), *Evaluating the Viability of Intrusion Detection System Benchmarking,* B.S. Thesis submitted to The Faculty of the School of Engineering and Applied Science, University of Virginia. http://www.cs.virginia.edu/~evans/students.html, Retrieved on June 2004.

Ramadas M., Ostermann S., Tjaden B., (2003) *Detecting Anomalous Network Traffic with Self-organizing Maps*, 6th International Symposium on Recent Advances in Intrusion Detection. Lecture Notes in Computer Science 2820. Springer-Verlag, pp 36-54.

Roesch M., (1999), *Snort – Lightweight Intrusion Detection for Networks*, Proceedings of the 13th Systems Administration Conference, pp. 229-238.

Song D., Heywood M.I., Zincir-Heywood A.N., (2003) *A Linear Genetic Programming Approach to Intrusion Detection*, GECCO'03, Proceedings of the Genetic and Evolutionary Computation Conference.

Tripwire Web Site (2004), *Home of the Tripwire Open Source Project*, http://www.tripwire.org/, Retrieved on June 2004.

Terms and Definitions

Attack vs. Intrusion: A subtle difference, intrusions are the attacks that succeed. Therefore the term attack represents both successful and attempted intrusions.

CERT / CC ©: CERT Coordination Center. Computer security incident response team, which provide technical assistance, analyze the trends of attacks and provide response for incidents. Documentation and statistics are published at their web site www.cert.org.

Logging: Recording vital information about an incident. Recorded information should be sufficient to identify the time, origin, target and if applicable characteristics of the attack.

Fragmentation: When the data packet is too large to transfer on given network, it is divided into smaller packets. These smaller packets are reassembled on destination host. Among with other methods, intruders can deliberately divide the data packets to evade IDSs.

Exploit: Taking advantage of a software vulnerability to carry out an attack. To minimize the risk of exploits, security updates or software patches should be applied frequently.

Light Weight IDS: An intrusion detection system, which is easy to deploy and have smaller footprint on system resources.

Machine Learning: A research area of artificial intelligence, which is interested in developing algorithms to extract knowledge from the given data.

Open Source Software: Software with its source code available for users to inspect and modify to build different versions.

Security Management: In network management, the task of defining and enforcing rules and regulations regarding the use of the resources.