International Journal of Information & Network Security (IJINS)
Vol.2, No.2, April 2013, pp. 197~202
ISSN: 2089-3299

An Architecture of Hybrid Intrusion Detection System

Kanubhai K. Patel*, Bharat V. Buddhadev**
* Ahmedabad University
** Department of Computer Engineering, LDCE, G T University

Article Info

Article history:

Received Nov 4th, 2012
Revised Dec 19th, 2012
Accepted Dec 22th, 2012

Keyword:

Hybrid IDS
Intrusion detection
Misuse detection
Network security
Signature-based

ABSTRACT

Intrusion Detection System (IDS) is renowned and widely-deployed security tool to detect attacks and malicious activities in information system. It is an essential element of any contemporary information system. There are mainly two techniques for intrusion detection: i) misuse (signature-based) detection and ii) anomaly (behavior-based) detection technique. Both the techniques have their advantages and disadvantages. This paper presents research from an ongoing study on the use of features of both the intrusion detection techniques to design a novel and efficient hybrid IDS. An architecture and implementation details of our hybrid IDS are presented. Furthermore, unique characteristics of our hybrid IDS are described. This paper concludes with future research directions and challenges in IDS.

Corresponding Author:

Kanubhai K. Patel,
Ahmedabad University,
Email: kkpatel7@gmail.com

1. INTRODUCTION

Intrusion Detection System (IDS) is renowned and widely-deployed security tool to detect attacks and malicious activities in information system. It is generally deployed as a second line of defense along with other defensive security mechanisms, such as firewall, vulnerability monitor, access control and authentication that protects information system. Intrusion detection involves monitoring network traffic, detecting attempts to gain unauthorized access to a system or resources, and alerting the appropriate persons so that countermeasures can be taken. It also involves developing an understanding of how attacks occur. IDS is an essential element of any modern information system. It is a necessary part of the entire defense system because of following reasons: (1) Numerous long-established systems and applications were developed without security in mind. (2) Systems and applications were developed to work in a different environment and may become vulnerable when deployed in the current environment (for example, a system may be perfectly secure when it is isolated but become vulnerable when it is connected to the Internet). Intrusion detection provides a way to identify, and thus allow responses to, attacks against these systems. (3) Due to the limitations of information security and software engineering practice, computer systems and applications may have design flaws or bugs (e.g., protocol flows) that could be used by an intruder to attack the systems or applications. As a result, certain preventive mechanisms (e.g., firewalls, access control, and authentication) may not be as effective as expected.

We have designed a Hybrid IDS to detect attacks on information system. In this paper, we have presented an architecture and implementation details of our hybrid IDS. The rest of the paper is structured as follows. Section 2 describes various types of intrusion detection techniques along with their advantages and disadvantages. Section 3 differentiates host-based and network-based IDS. Section 4 presents an architecture of our hybrid IDS. While section 5 presents unique characteristics of our hybrid IDS. Section 6 reviews literature of related works. And Section 7 concludes the paper with future research directions and challenges in IDS.

2. Types of Intrusion Detection Techniques

There are mainly two techniques for intrusion detections: i) misuse (signature-based) detection and ii) anomaly (behavior-based) detection [4]. Purpose of both techniques is in attempting to detect any attacks or intrusions in a system.

2.1. Misuse (Signature-based) Detection Technique

The misuse (signature-based) detection is normally used for detecting known attacks. It requires that all known threats will be defined first, and the information regarding these threats to be submitted to the IDS. Thus, the IDS is able to then compare all incoming or outgoing activity against all known threats in its knowledge base and raise an alarm if any activity matches information in the knowledge base. The information stored in this knowledge base is usually known as signatures [4]. The process for actually comparing a signature with an attack include simple string matching – which involves looking for unique key words in network traffic to identify attacks – to more complex approaches such as rule-based matching which defines the behavior of an attack as a signature [4]. Various string-matching (or pattern-matching) algorithms are used to inspect the content of packets and identify the attacks signature in IDS. There are mainly two kinds of algorithms, viz. i) Single-keyword pattern matching algorithms viz., Brute force algorithm, Knuth-Morris-Pratt Algorithm [16], and Boyer-Moore algorithm [7]; and (ii) Multiple-keyword pattern matching algorithms viz., Aho-Corasick [1], Wu-Manber Algorithm [33], Horspool Algorithm [15], Quick search algorithm [31], Piranha [3], and E2xb [2].

Following are the advantages of misuse detection technique: (1) Signatures are very easy to develop and understand, if we know what network behavior we are trying to identify. For instance, we might use a signature that looks for particular strings within exploit particular buffer overflow vulnerability. The events generated by signature-based IDS (SIDS) can communicate the cause of the alert. (2) It has a relatively low rate of false alarms [30], which means that the SIDS has a relatively high precision. This high precision is caused by the fact that a SIDS is explicitly programmed to detect certain known kinds of attacks [30].

One big challenge of SIDS is that every signature requires an entry in the database, and so a complete database might contain hundreds or even thousands of entries. Each packet is to be compared with all the entries in the database. This can be very resource consuming and doing so will slow down the throughput and making the IDS vulnerable to DoS attacks. Some of the IDS evasion tools use this vulnerability and flood the IDS systems with too many packets to the point that the IDS cannot keep up with the traffic, thus making the IDS time out and drop packets and as a result, possibly miss attacks [28]. On modern systems, string-matching can be done more efficiently, so the amount of power needed to perform this matching is minimal for a rule set. For example if the system that is to be protected only communicate via DNS, ICMP and SMTP, all other signatures can be ignored.

Following are the disadvantages of signature-based intrusion detection system (SIDS): (i) The detection rate of attacks is relatively low [30]. (ii) There is a lower recall for new types of intrusions. (iii) An attacker will try to modify a basic attack in such a way that it will not match the known signatures of that attack. The attacker may insert malformed packets that the IDS will see, to intentionally cause a pattern mismatch; the protocol handler stack will then discard the packets because of the malformation. Each of these variations could be detected by an IDS, but more different signatures require additional work for the IDS, which reduces performance [24]. (i) It cannot detect a new attack for which a signature is not yet installed in the database. Ideally, signatures should match every instance of an attack, match subtle variations of the attack, but not match traffic that is not part of an attack. However, this goal is difficult to accomplish in current IDSs [24]. (ii) The efficiency of the SIDS is greatly decreased, as it has to create a new signature for every variation. As the signatures keep on increasing, the system performance deteriorates. Due to this, many SIDS are deployed on systems with multi processors and multi Gigabit network cards. IDS developers develop the new signatures before the attacker does, so as to prevent the novel attacks on the system. The difference of speed of creation of the new signatures between the developers and attackers determine the efficiency of the system.

2.2. Anomaly (Behavior-based) Detection Technique

The anomaly (heuristic-based) detection is based on defining the network behavior. Instead of looking for matches, anomaly intrusion detection looks for behavior that is suspicious [24]. Anomaly-based IDS attempt to characterize normal operation, and try to detect any deviation from normal behavior [30]. The network behavior is in accordance with the predefined behavior, then it is accepted or else it triggers the event in the anomaly detection. The accepted network behavior is prepared or learned by the specifications of the network administrators. It builds a model of acceptable behavior and flag (i.e. label) exceptions to that model; for the future, the administrator can mark a flagged behavior as acceptable so that the anomaly-based IDS will now treat that previously unclassified behavior as acceptable [24].

An anomaly-based detection technique compares normal behavior against the current pattern of behavior in a system. In order to achieve this task, the main challenge in anomaly detection technique is in learning what is considered "normal" behavior. The work by Axelsson [4] describes the two main approaches which are used to achieve this goal: self-learning or programmed anomaly detection.

In the self-learning approach, the anomaly detection system will begin to automatically monitor events, such as live network traffic, on the environment it has been implemented on and attempt to build information on what is considered normal behavior [4]. This is otherwise known as online learning [14].

In the programmed approach, the anomaly-based IDS must manually learn what is considered normal behavior by having a user or some form of function "teaching" the system through input of information [4]. This is otherwise known as offline learning, and may involve feeding the system a network traffic data set which contains normal network traffic [14].

The major advantage of anomaly detection over misuse technique is that a novel attack for which a signature does not exist can be detected if it falls out of the normal traffic patterns. This is observed when the systems detect new automated worms. If the new system is infected with a worm, it usually starts scanning for other vulnerable systems at an accelerated rate filling the network with malicious traffic, thus causing the event of a TCP connection or bandwidth abnormality rule. Following are the disadvantages of anomaly intrusion detection: (i) There is a higher rate of false alarms, which means a lower precision [30]. (ii) It also needs periodic online retraining of the behavior profile. (iii) It tends to be computationally expensive because several metrics are often maintained that need to be updated against every system activity and, due to insufficient data, they may gradually be trained incorrectly to recognize an intrusive behavior as normal due to insufficient data [6].

3. Host-based vs Network-based IDS

Intrusion detection can be implemented either on the hosts that need to be protected or on a network device that can sniff the traffic for all the hosts on the network. Based on the implementation locations, there are two common types of IDS, viz., i) host-based IDS, and ii) network-based IDS.

Host-based IDS (HIDS) examines information at the local host or operating system on which it is installed. It examines actual system calls and system log files. While network-based IDS (NIDS) examines the actual network packets that are traveling across the network. It examines this traffic for known signs of instructive activity. Because NIDS is watching network traffic, any attack signatures detected may succeed or fail. It is usually difficult if not impossible for NIDS to assess the success or failure or the actual attacks. It only indicates the presence of intrusive activity.

4. Architecture of our Hybrid IDS

Our Hybrid IDS consists of six components viz., i) Data acquisition module, ii) Signature database, iii) Analyzer, iv) Anomaly detector, v) Signature generator, and vi) Counter-measure module (see Fig-1).

Data acquisition module has multiple sensors. Sensors are placed either on individual host or in particular network segment. Sensors that are placed on individual hosts observe packets as they enter and leave that host. Sensors that are placed on a particular network segment read packets as they pass into and out of each segment. The sensors need to be positioned in locations where they will be able to capture all of the packets entering and leaving a host or network segment. Sensors that are placed on network segments do not always have the ability, if the traffic level becomes too heavy, to capture every packet. Repositioning the sensors on each network host will improve accuracy even though the effort of installing them can be considerable. The important thing is to be able to capture all packets so that none can potentially circumvent the IDS. For our purposes we use Snort on the Windows operating system using WinPcap.

The Signature database records enable the IDS to have a set of signature, criteria or rules against which they can compare packets as they pass through the sensor. The database of signatures needs to be installed along with the IDS software and hardware itself. After the Signature database is in place, sensors of the Data acquisition module gather data by reading packets from the network and reassemble them. As packets from network can arrive out of order, or can be duplicated. Moreover, packets arrive at a high speed therefore the Data storage is required to store the packets.

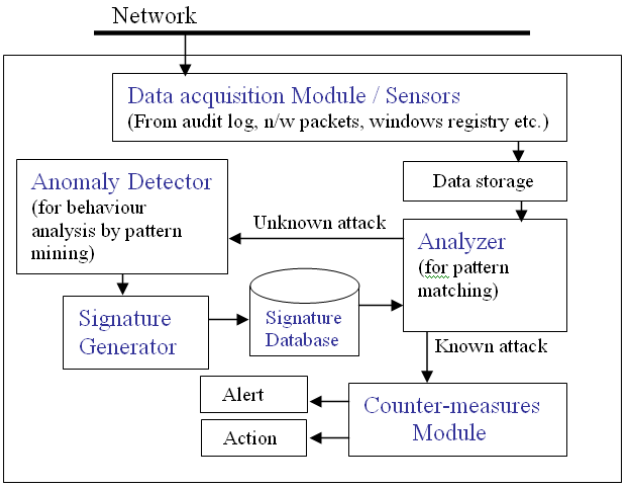

Figure 1. Schematic diagram of Hybrid IDS

The Analyzer module compares the packets it observes with the signatures or rules of normal patterns of behavior stored in Signature database using pattern-matching algorithm. We use the well-known Aho-Corasick algorithm [1] for performing pattern matching. If analyzer finds any match then sends appropriate alert message for known attack to the Counter-measure module. Also it enters entry in log file about the event that caused the alert. If analyzer does not find any match then sends data to Anomaly detector for finding anomaly using pattern mining technique. If Anomaly detector finds any anomaly then send appropriate message to Signature generator. Here Signature generator creates rule or signature and make new entry in Signature database.

When Counter-measure module receives the alert message of known attack from Analyzer, it notifies the administrator in one of several ways that the administrator has configured beforehand. The module might display a pop-up window or sends an e-mail message to the designated individual, for example. Besides the automated response sent to the administrator, this module can be configured to take action at the same time that an alert message is received. Typical actions are: i) Alarm, in which an alarm is sent to the administrator, ii) Drop, in which the packet is dropped without an error message being sent to the originating computer; and iii) Reset, which instructs the IDS to stop and restart network traffic and thus stop especially severe attacks. This module is also used by network administrator to evaluate the alert message and to take proper actions such as dropping a packet or closing a connection. The administrator can anticipate having to fine-tune the signature database to account for situations that seem to the IDS to be intrusions but that are actually legitimate traffic. For example, an adjustment might be made to enable traffic that might otherwise be seen by the firewall as suspicious, such as a vulnerability scan performed by a scanning device located at a particular IP address. The IDS could be configured to add a rule that changes the action performed by the IDS in response to traffic from that IP address from Alarm to Drop.

5. Unique characteristics of our hybrid IDS

The following are the list of unique characteristics of our hybrid IDS.

a) It will be easy to install and configure.
b) It will be adaptive in nature and adapts the changes in user and system behavior.
c) It will run constantly with minimal human supervision. It will create signatures of new attacks.
d) Design of our hybrid IDS makes it fault tolerant, so that it will be able to recover from crashes. It will be able to get its prior state and resume its operation with out any adverse effect.
e) It will be able to monitor itself and detect attacks on it.
f) It will consume less memory to operate. It can be run with less overhead on the systems where it is installed.
g) It will be accurate and thereby there will be less number of false positives and false negatives.

6. Related Work

Some of the well known IDSs that employ signature-based (misuse) detection technique are Snort [9], [29], Bro [23], Tipping Point (http://www.hpenterprisesecurity.com/), NFR NID-200 (NFR Security, Inc. www.nfr.com), RealSecure (http://www.iss.net/), and Cisco Secure IDS (www.cisco.com). Snort is the most popular open source NIDS created by Martin Roesch. It is the most flexible and powerful solution for

intrusion detection and also for intrusion prevention. Other attempts to solve the intrusion detection and response problem can be found in [8], [10], [21], [25].

Langin and Rahimi [17] have covered the State-of-the-Art in the field of intrusion detection using soft computing. While, Rajput and Shrivastava [26] have covered various data mining based database intrusion detection system. Mannila and Toivonen [22] have proposed the concept of frequent episode rules (FERs) first for anomaly-based IDS (AIDS). Subsequently, Lee et al. [20], [19], and Fan et al. [13] have suggested a framework to specify FERs for anomaly detection against normal traffic profiles. Qin and Hwang [25] have refined the rule formulation procedure with an adaptive base-support algorithm to mine normal traffic records. Different axis attribute values apply different thresholds. Many other researchers have studied supervised AIDS by training over attack-free traffic [5], [11], [12], [13], [18], [34]. AIDS is designed by mining FERs [20], [25] over Internet connections. They developed a new weighted signature generation algorithm to characterize anomalous attacks and extract their signatures. The new signatures are generated from anomalies detected by AIDS. This idea was inspired by former works on weighted association rules [27], [32]. This new approach automatically enables HIDS to detect similar anomalous attacks in the future. We have considered close cooperation between the two types of intrusion detection techniques. We have proposed the architecture of HIDS. Our HIDS integrates the flexibility of anomaly-based IDS with the accuracy of signature-based IDS.

7. Conclusion

The main characteristic of misuse (signature-based) intrusion detection technique is in comparing incoming threats against a predefined knowledge base in order to decide whether the threat is considered an attack or intrusion whilst anomaly detection technique involves looking for any unexpected changes in behavior of a system against what is considered normal behavior. Both misuse and anomaly detection techniques have their own advantages and disadvantages. We have used features of both the intrusion detection techniques in our Hybrid IDS. This paper presents research from an ongoing study on the use of features of both the intrusion detection techniques to design a novel and efficient hybrid IDS. An architecture and implementation details of our Hybrid IDS are presented. The proposed design of HIDS, however, aims to be more accurate and it does not require more processing resources, thus offering both speed and accuracy to detect the intrusions. We are encouraged by this pilot design, which suggests that such systems would help network administrators to get effectively secure the information systems. We have planned to evaluate this design both as offline using available dataset, and as online in real environment in the networks of our organization.

REFERENCES

[1] Aho, A and Corasick, M. (1975). “Efficient String Matching: An Aid to Bibliographic Search”. Communications of the ACM, 18, 1975, pp. 333-40.
[2] Anagnostakis, K. G., Markatos, E. P., Antonatos, S. and Polychronakis, M. (2003). E2xB: A domain-specific string matching algorithm for intrusion detection. In Proceedings of the 18th IFIP International Information Security Conference (SEC2003), May 2003.
[3] Antonatos, S., Polychronakis, M., Akritidis, P., Anagnostakis, K.G., and Markatos, E.P. (2005). "Piranha: Fast and Memory-Efficient Pattern Matching for Intrusion Detection", ;in Proc. SEC, 2005, pp.393-408.
[4] Axelsson, S. (2000). Intrusion-detection systems: A taxonomy and survey. Tech. Rep. 99-15, Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, 2000.
[5] Barbara, D., Couto, J., Jajodia, S., Popyack, L. and Wu, N. (2001). “ADAM: Detecting Intrusions by Data Mining,” Proc. IEEE Workshop Information Assurance and Security, 2001.
[6] Botha, M., and von Solms, R., (2003). “Utilizing fuzzy logic and trend analysis for effective intrusion detection”, In Computers & Security, volume 22, 2003, pp 423–434.
[7] Boyer, R. and Moore, S. (1977). "A Fast String Searching Algorithm." CACM, 20, 1977, 762-72.
[8] Burroughs, D. J., Wilson, L.F., & Cybenko, G.V. (2002). “Analysis of Distributed Intrusion Detection Systems Using Bayesian Methods Performance,” Proc. IEEE Int’l Computing and Comm. Conf., 2002, pp. 329-334.
[9] Casewell, b. and Beale, J. (2004). SNORT 2.1. Intrusion Detection, (2nd ed), Syngress, May 2004.
[10] Cuppens F., and Miege, A. (2002). “Alert Correlation in a Cooperative Intrusion Detection Framework,” Proc. 2002 IEEE Symp. Security and Privacy, pp. 187-200, 2002.
[11] Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P., Srivastava, J., Kumar, V., and Dokas, P. (2004). “The MINDS—Minnesota Intrusion Detection System,” Next Generation Data Mining, MIT Press, 2004.
[12] Eskin, E., Arnold, A., Prerau, M., Portnoy, L. and Stolfo, S. (2002). “A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data,” Applications of Data Mining in Computer Security, Kluwer Academic Publishers, 2002.
[13] Fan, W., Miller, M., Stolfo, S., Lee, W., and Chan, P. (2001). “Using Artificial Anomalies to Detect Unknown and Known Network Intrusions,” Proc. First IEEE Int’l Conf. Data Mining, Nov. 2001.
[14] Gong, F. (2003). Deciphering Detection Techniques: Part II Anomaly Based Intrusion Detection [White Paper], McAfee Security, McAfee Security White Paper, 2003, Retrieved October 10, 2012, from

https://secure.mcafee.com/japan/products/pdf/Deciphering_Detection_Techniques-Anomaly-Based_Detection_WP_en.pdf
[15] Horspool, R. N. (1980). Practical fast searching in strings. Software Practice and Experience, 10(6):501–506, 1980.
[16] Knuth, D. E., Morris, J. H., and Pratt, V. R. (1977). "Fast pattern matching in strings". SIAM Journal on Computing, 6(2), June 1977, pp. 323–350.
[17] Langin, C. and Rahimi, S. (2010). "Soft Computing in Intrusion Detection: The State of the Art," Journal of Ambient Intelligent Human Computer, Vol. 1, Issu. 2, pp. 133–145.
[18] Lazarevic, A, Ertoz, L, Kumar, V, Ozgur, A and Srivastava, J. (2003). "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection", Proc. Third SIAM Conf. Data Mining, 2003, http://www.users.cs.umn.edu/~kumar/papers.
[19] Lee, W., and Stolfo, S. (2000). "A Framework for Constructing Features and Models for Intrusion Detection Systems," ACM Trans. Information and System Security (TISSec), 2000.
[20] Lee, W., Stolfo, S.J., and Mok, K. (2000). "Adaptive Intrusion Detection: A Data Mining Approach," Artificial Intelligence Rev., vol. 14, no. 6, Kluwer Academic Publishers, Dec. 2000, pp. 533-567.
[21] Lippmann, R. P., and Haines. J. (2000). "Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation," Proc. Third Int'l Workshop Recent Advances in Intrusion Detection (RAID '00), H. Debar, L. Me, and S.F. Wu, eds., pp. 162-182, 2000.
[22] Mannila, H., and Toivonen, H. (1996). "Discovering Generalized Episodes Using Minimal Occurrences," Proc. Second Int'l Conf. Knowledge Discovery and Data Mining, Aug. 1996.
[23] Paxson, V. (1998). "Bro: A System for Detecting Network Intrusions in Real Time," Proc. Seventh USENIX Security Symp., 1998.
[24] Pfleeger, C. and Pfleeger, S. (2003). Security in computing. Prentice Hall, 2003.
[25] Qin M., and Hwang, K. (2004). "Frequent Episode Rules for Internet Traffic Analysis and Anomaly Detection," Proc. IEEE Network Computing and Applications (NAC '04), Sept. 2004.
[26] Rajput, I. J. and Shrivastava, D. (2012). "Data Mining Based Database Intrusion Detection System: A Survey," International Journal of Engineering Research and Applications (IJERA) Vol. 2, Iss. 4, pp.1752-1755, 2012.
[27] Ramkumar, G. D., Ranka, S., and Tsur, S. (1998). "Weighted Association Rules: Model and Algorithm," Proc. Fourth ACM Int'l Conf. Knowledge Discovery and Data Mining, 1998.
[28] Raven Alder, J. B., Doxtater, A., Foster, J., Kohlenberg, T., and Rash, M. (2004). "Snort 2.1 Intrusion Detection," 2nd ed. Rockland, MA: Syngress (Distributed by O'Reilly and Associates), 2004.
[29] Roesch, M. (1999). "SNORT—Lightweight Intrusion Detection for Networks," Proc. USENIX 13th Systems Administration Conf. (LISA '99), pp. 229-238, 1999.
[30] Stillerman, M., Morceau, C., and Stillman, M. (1999). "Intrusion Detection for Distributed Applications", Communications of the ACM, 42(7), July, 1999, 62-69.
[31] Sunday, D. M. (1990). "A very fast substring search algorithm", Communications of the Association for Computing Machinery, 1990, pp. 132-142.
[32] Tao, F., Murtagh, F., and Farid, M. (2003). "Weighted Association Rule Mining Using Weighted Support and Significance Framework," Proc. Ninth ACM Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD), pp. 661-666, 2003.
[33] Wu, S., and Manber, U. (1992). "Fast Text Searching With Errors." Technical Report TR-91-11, Department of Computer Science, University of Arizona., 1991. To appear in Proceedings of USENIX Winter 1992 Conference, San Francisco, January, 1992.
[34] Xie, Y., Kim, H., O'Hallaron, D.R., Reiter, M.K., and Zhang, H. (2004). "Seurat: A Pointillist Approach to Anomaly Detection," Proc. Seventh Int'l Symp. Recent Advances in Intrusion Detection (RAID '04), 2004.