ISSN:0975-9646

An Intensifying Concept of Intrusion Detection to Accelerate Accuracy and Effectiveness with Knowledge

Vipin Singh[1] ,Himanshu Arora[2]

[1] *M-Tech Student, Arya Institute of Engineering & Technology, Jaipur*
[2] *Associate Professor, Arya Institute of Engineering & Technology, Jaipur*

Abstract: **In the recent years, advancement in the field of communication technology has made computer networks an essential part of our lives. The result is Security attacks through network have grown in recent years. Since machines on a network provide many requisite services and store sensitive information, therefore theses are a prime target of malicious hackers. The term for such activities done by hackers is known as intrusions. Any attempt to compromise the confidentiality, integrity or availability of a resource is termed as an intrusion.**

This research paper directly provides the design of a contextually sensitive post processor for Intrusion Detection Systems with knowledge. It relies on information about constituent hosts to generate a composite score for each received alert. The composite score reflects the impact that the attack will likely have on the system and is made up of two different metrics – the susceptibility of the system to attack and the destructive capabilities of the attack.

***Keyword*: Security, Intrusion, hosts, metrics, attack, networks, alert, knowledge.**

I INTRODUCTION

Today the number of computers connected to a network and the Internet is increasing with every day. This collective with the increase in networking speed has made intrusion detection a challenging process. Protecting computer networks from internal and external threats has become a high main concern. Any security violation in information systems can easily make vulnerable the monetary and structural reliability of an organization or a company, because information is often as critical as the corporal assets that it represents. Intrusion detection is the process of screen the proceedings taking place in a computer system or network and analyzing them for signs of possible incidents, which are violations or coming up threats of breach of computer security policies, acceptable use policies, or security standard practices [1] Since their gain in popularity, intrusion detection systems have begin to be used regularly as one component of an successful covered security model for an organization. Various amendment in monitoring systems [2, 3, 4] prompt interest, and intrusion detection quickly became known as an imperative computer safekeeping tool for individual computers as well as in computer networks.

II TYPES OF INTRUSION DETECTION SYSTEMS

There are different bases of classifying the IDS. We can classify the Intrusion Detection System in many types. Following are types IDS based on the study of [5, 6, 7] have proposed many different methods of classifying IDS. A few of them are explained here.

1 Based on Detection Techniques

This classification relies on underlying methodology used by IDS to detect the attacks. There are two different approaches that can be used

1.1 Behavior Based IDS

An IDS that determines the normal behavior of the protected network and logs any deviations in the behavior beyond a pre-determined threshold, is known as Behavior based IDS or more commonly as Anomaly Based IDS [8]. The steps of operation of Anomaly Based IDS can be summarized [9] as

- Parameterization – The observed instances of target system are represented in pre-established form
- Training Stage – The normal behavior of target system is characterized and a model for the same is built. This step may be done manually or using automatic tools.
- Detection Stage – Once the normal model of system is ready, it can be deployed to compare the observed traffic with normal parameters. In case the deviation is more than a threshold value, an alarm is raised.

1.2 Knowledge Based IDS

An IDS that relies on predefined knowledge about attacks to detect anomalous traffic is known as Knowledge Based IDS or, more commonly as, Signature Based IDS [10, 11]. Much like the traditional virus scanners, Signature based IDS maintain a repository of signatures of attacks that exploit known vulnerabilities of the system. These systems analyze the network packets for presence of signatures and generate an alert if any packet matches a signature. Thus, any action that is not explicitly recognized as attack is considered acceptable. Techniques [10] employed by such systems are

- Expert Systems – They contain a set of rules to describe an attack. The audit events are then translated into facts carrying their semantic significance in expert systems. An inference engine is then used to draw conclusions about the attack. Thus, abstraction of audit data is increased by adding a semantic to it.
- Signature Analysis – This method is not very different from Expert Systems. The semantic description of attack is transformed into information that can be found in the audit trail very easily. Usually, it consists of encoding the payload of packets into signatures.
- Petri nets- Some academic projects have also explored possibility of using Petri nets for knowledge based systems. However, this method has proven unreliable in case of complex attacks.
- State Transition Analysis – The attack is described as a set of goals and transitions and represented as state transition diagrams.

Such systems are highly accurate in identifying known attacks and are very easy to implement and configure, even for a large network. The core of the system, once employed, does not need to change. Newer signatures/definitions can be added using plug-ins. The downside is that the signatures need to be updated regularly. Also, signature writing is an intricate task, which, if not performed correctly, can lead to large number of false alarms or missed attacks. Researchers have also pointed out that attacks can be modeled to either defeat signature based IDS [12] or generate lot of noise, i.e. false alarms [13]. Such Signature based IDS have been developed extensively both by commercial and educational organizations.

2 Based on Coverage

Classifications of IDS are also possible where the sensor is placed and what sort of events it gathers. There are two choices

2.1 Network IDS (NIDS)

Network Based IDS [5, 15] detect attacks by capturing and analyzing network packets. NIDS typically consist of one or more sensors placed at different segments of network. These sensors listen to the network segment unobtrusively and perform local analysis on the traffic. They then transmit the captured information to a central management console. NIDS, hence, provide a macro-level view of the network and help in detecting remote attacks. NIDS are deployed by connecting them to the spanning ports of network infrastructure. NIDS have many advantages. Due to their positioning, they can see the larger picture of the network. E.g. - A slew of probes to port 80 (web-servers) of all the hosts on the network will be quickly flagged as port-scan by the NIDS even though each host only receives a single probe. Even a large network can easily be monitored by placing a few sensors at the correct location. It can detect rogue/unauthorized hosts easily since it can listen to the traffic to or from such hosts. Deployment of NIDS does not affect the existing network infrastructure since host configurations do not need to be modified. Drawbacks of NIDS are that in case of heavy traffic NID may miss out some attacks due to packet drops. Also, NIDS can only sense the presence of attack signatures but cannot predict the success of attacks. As shall be shown later, not every attack leads to successful intrusion. Since NIDS work by analyzing network packets, some attacks have been developed that attempt to crash the IDS by sending malformed/fragmented packets. A classical example is the RPC fragmentation buffer overflow vulnerability in snort.

2.2 Host IDS (HIDS)

Just as the name suggests, HIDS [7, 17] are deployed at individual hosts. Rather than watch network traffic (like NIDS), HIDS monitors the behavior and state of the system for signs of possible intrusion. In order to accomplish this, HIDS may monitor system logs, file systems and other audit trails. HIDS were the first IDS developed during heydays of mainframe computing when protecting a single computer from malicious local users was a priority. In recent times, focus has shifted to communication among various HIDS to protect a network better Advantages of HIDS are that since it looks at the state of the system, it can easily detect novel attacks while at the same time protecting against classical attacks. Reliability of such systems is high due to focus on core computing platform. Once initialized, it will not need any signature updates. On the downside, HIDS need to be initialized on each host. Thus, if newer hosts are to be introduced in the network, they will need to install HIDS software. Management of such distributed system can also be an issue. HIDS do not take into account larger picture.

3 Based on Actions

IDS are also classified based on the response to detected intrusions.

3.1 Active IDS

Active IDS, also known as Intrusion Prevention System (IPS), can perform a variety of actions on detecting an intrusion attempt, in addition to logging the attack and alerting the operator. The erring packet may be dropped or the TCP connection may be reset by injecting RST packets in the stream. It can also modify firewall rules to block future packets from the same source. A next generation IPS can launch counter-attacks against the attacker though this capability is yet to be realized. To be able to generate active response, IPS must be deployed in the network stream rather than on the fringes as in case of traditional IDS [5, 16].

3.2 Passive IDS

On detecting an attack, a passive IDS [5] simply logs the alert source (packet or audit trail) and generates an alert for the system administrator. The alert may be sent via e-mail or flashed on customized monitoring system. However, passive IDS will not in any way interfere with the attack.

III PROPOSED SOLUTION

The proposed system, henceforth called PIKE (Post-processor for IDS alerts using Knowledge-based Evaluation), classifies the IDS alerts by considering the context of the alert. For the purpose of prioritization, PIKE considers the following aspects as context

- Target Configuration – The OS/Services installed on the target host
- Vulnerability Assessment – The vulnerabilities reported in the target host
- Attack Impact Assessment – The impact that the attack can have on any system PIKE collects the network information periodically using established tools and employs accepted vulnerability metrics to measure impact of the attack. On receiving an alert, it calculates two scores
- Affectability Score – The score which tells to what extent the attacked system is vulnerable to the given attack.
- Impact Score – The score which quantifies the likely impact the attack will have on the target.

The two scores are then aggregated to give a Relevance score which, as its name suggests, indicates the relevance of the alert to the monitored environment. Each alert is tagged with this relevance. PIKE then employs a simple threshold classifier to the relevance values to partition the alerts into relevant and non-relevant categories. Note that PIKE does not take any action against alerts on its own although this functionality can be incorporated into it. Figure 1.1 shows PIKE as a part of network security infrastructure.

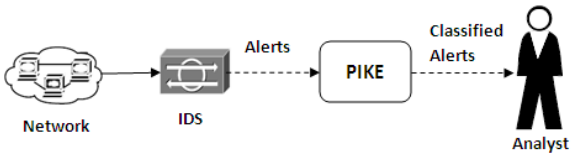

Figure 1.1: PIKE as a part of network security infrastructure

Logically, PIKE can be thought of to be made up of various components. Figure 1.2 shows the conceptual architecture of PIKE.

Existing IDS setup generates alerts while watching over a network. The IDS alerts are passed onto PIKE which prioritizes them. PIKE is made up of two distinct entities – The Knowledge Base which is a repository of contextual information and Post Processor which is a tool that uses the knowledge base to prioritize alerts.

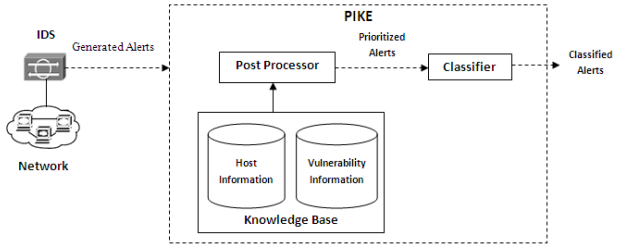

Figure 1.2: Conceptual PIKE Architecture

Knowledge Base is a collection of database and files that holds information regarding systems present in the network and known vulnerabilities. It has two principle components – Host Information Database which holds details of systems connected to the network. These details include the OS installed on the machines, their open ports, services running on these machines and vulnerabilities associated with that machine. The vulnerability information database holds details of known vulnerabilities. It stores information such as impacted/immune systems, remedial measures, cross-references, credits for discovery, discovery date, impact scores etc.

Post Processor is the main entity of PIKE. It receives the alerts from the IDS and uses the knowledge from Knowledge Base to compute the relevance value of each alert. This relevance value acts as a measure of alert priority.

IV RESULT

The test dataset was created for a small network segment. Since we need to know the outcome of each attack, it does not contain any background traffic. It only consists of alerts that were explicitly identified by Snort. To generate attack traffic, we used Metasploit Framework [17], a well known Open Source project that has many known ready to use exploits. In addition to that, we also used code fragments found on some security related blogs. We relied on ready to use exploits because exploit writing in itself is a very wide field consisting of a lot of trial and error.

Figure 1.3 shows the variation of accuracy with the threshold. As is evident higher thresholds have higher accuracy. However, setting the threshold too high decreases the accuracy. This is because some relevant alerts that had a lesser score (due to lower CVSS impact score) are classified as irrelevant and are thus False Negatives (FN). If we consider the costs of false positives and false negatives we get the weighted accuracy. Figure 1.3 shows the weighted accuracy curve with CR = $^{0.3}/_{0.7}$

As can be seen, the fall in accuracy on increasing the threshold is more pronounced because of false negatives having a higher cost. It can also be seen that accuracy does not reach 1 for any threshold levels. This is due to Pseudo-Fail alerts, i.e. alerts that would have been successful if environment had been right. In both the graphs, the maximum values occur at thresholds 0.8 and 0.9. Thus, optimum threshold for given dataset can be taken as 0.8 (initial maximum).

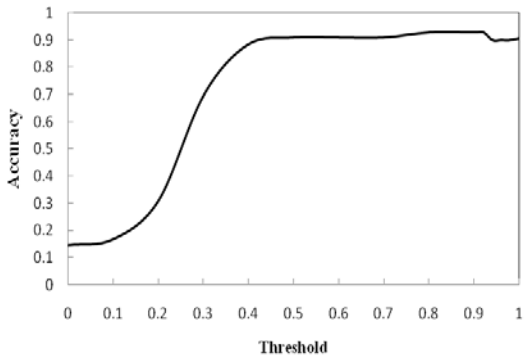

Figure 1.3 Accuracy vs Threshold Graph for PIKE

V CONCLUSION

In this paper we have presented design of a contextually sensitive post processor for IDS alerts, PIKE. It relies on information about constituent hosts to generate a composite score for each received alert. The composite score reflects the impact that the attack will likely have on the system and is made up of two different metrics – the susceptibility of the system to attack and the destructive capabilities of the attack. After calculating the score, PIKE then employs a simple binary classifier to partition the alerts into relevant and irrelevant alerts before presenting them to the analyst. The analyst can analyze the relevant alerts while ignoring or saving for future reference the relevant ones

REFERENCES

[1] http://csrc.ncsl.nist.gov/publications/nistpubs/800-94/SP800-94.pdf Guide to Intrusion Detection and Prevention Systems (IDPS), NIST CSRC special publication SP 800-94, released 02/2007.

[2] J. P. Anderson; Computer security threat monitoring and surveillance in technical report; J. P. Anderson Co., Fort Washington, PA, USA, 1980.

[3] D. E. Denning. An intrusion-detection model. IEEE Transactions on Software Engineering, 13(2):222–232, 1987.

[4] M. Gregg. CISSP Exam Cram 2. Que, New York, NY, USA, 2005.

[5] K. Scarfone, and P. Mell, ―Guide to intrusion detection and prevention systems, NIST, US Dept. of Commerce, Tech. Rep. 800-94, 2007

[6] S. Axelsson, ―Intrusion detection systems: A survey and taxonomy; Department of Computer Engineering, Chalmers University, Tech. Rep. 99-15, 2000.

[7] H. Debar, M. Dacier, and A. Wespi, ―Towards a taxonomy of intrusion-detection systems, Computer Networks, vol. 31, no. 8, pp 805-822, April 1999.

[8] P. Garcıa-Teodoro, J. Dıaz Verdejo, G. Macia Fernandez, and E. Vazquez; Anomaly-based network intrusion detection: Techniques, systems and challenges, Computers and Security, vol. 28, no. 1, pp. 18-28, 2009.

[9] J.M. Estevez-Tapiador, P. Garcıa-Teodoro, and J.E. Dıaz-Verdejo; Anomaly detection methods in wired networks: A survey and taxonomy, Computer Networks vol. 27, no. 16, pp. 1569–1584, 2004

[10] H. Debar, M. Dacier, and A. Wespi, ―Towards a taxonomy of intrusion-detection systems, Computer Networks, vol. 31, no. 8, pp 805-822, April 1999.

[11] T. F. Lunt, R. Jagannathan, R. Lee, A. Whitehurst, and S. Listgarten, "Knowledge- based intrusion detection," in Proceedings of the Annual Conference on AI Systems, 1989, pp.102 107.

[12] J. McDonald, ―Defeating sniffers and intrusion detection systems, Phrack magazine, vol. 8, Dec. 1998

[13] M. M. Yasin, and A. A. Awan, ―A study of host-based IDS using system calls, in Proceedings of the IEEE Networking and Communication Conference, 2004, pp. 36-41.

[14] H. Debar, M. Dacier, and A. Wespi, ―Towards a taxonomy of intrusion-detection systems, Computer Networks, vol. 31, no. 8, pp 805-822, April 1999.

[15] N. Ierace, C. Urrutia, and R. Bassett. ―Intrusion prevention systems, in Ubiquity, pp.2-11, June 2005

[16] Metasploit. Available at www.metasploit.com