ISSN:2249-5789
Ritika Lohiya et al , International Journal of Computer Science & Communication Networks,Vol 5(2),72-78

Current Trends in Network Intrusion Detection Techniques

Ritika Lohiya
Dept of CSE,
Nirma University,
Ahmedabad.
ritikalohiya207@gmail.com

Pranav Varma
Dept of CSE,
Nirma University,
Ahmedabad.
d1.pranav.varma@gmail.com

Yaman patel
Dept of CSE,
Nirma University,
Ahmedabad.
yaman.patel2@gmail.com

Abstract

Security threats in computer networks and internet has increased to a great extent. Zero day attacks appearing continually due to new development in network has led to a severe challenge in implementing adaptive security methods. This paper focus on Network Intrusion Detection Systems which are capable of monitoring the network malicious activities. The paper gives a brief introduction to Network Intrusion Detection Systems followed by the type of attacks detected by the NIDS with common vulnerabilities and the types of NIDS. Lastly, the paper discusses a case study on Snort which is widely accepted NIDS.

1. Introduction

The increase in the internet usage has led to the demand of increased security to protect the cyber infrastructure from various security attacks. Attacks on the network give rise to big problems which swiftly allow illegal activities in the network to compromise the on the three main goals of security and that is authenticity, integrity and availability of the network. Starting with 1988, with the Morris worm, the world of malwares has reached to such a high level where security of the network becomes mandatory, as any mistake in the network cannot be afforded.

Security measures like encrypting and decrypting the information with secret key serves only the means of protecting the information over the network channel. But besides encrypting and decrypting the text, what takes the most important place in detection of intrusion and taking necessary steps for future protection. An intrusion detection system is used to detect several types of malicious behaviours that can compromise the security and trust of the network system.

The hype for intrusion detection system has increased because they are the newest line of defence in network security and combine two levels of protection into one and that is identifying intrusion and prevention. The intrusion detection system monitors the activities on the network by observing the network and generating the reports for the system administrator. Whereas the firewalls which are used by most of the organizations and networks provides a basic line of defence by allowing or blocking connectivity to the network through port connections. The problem here is that the firewall does not investigate the data that is allowed in the network. With this limitation, firewalls cannot be considered for the overall security of the network, and therefore Intrusion detection systems are preferred over firewalls.

Intrusion detection systems perform the following tasks:

- Analysis of the computer or network system activity.
- Auditing the system configuration and vulnerabilities.
- Evaluation of correctness of the system and data files.
- Observing and abnormal activities.

Apart from all these activities, the major challenge of any intrusion detection system is to detect an attack and report it. Computer networks have a dynamic nature in sense that information and data within the network is continuously changing. Therefore, detecting any intrusion accurately and promptly the system has to work in real time. Operating in real time is not just to detect the intrusion but to attempt to the new dynamics of the system.

Intrusion detection system can be into two main categories namely host based intrusion detection

system and network based intrusion detection system. A host based intrusion detection system detects intrusion on a single host system. It monitors the activities like integrity of the system, files, host based traffic, system logs etc. It acts as a passive component of the system. Whereas, on the other hand Network intrusion detection system monitors the network traffic to protect the system from the network based attacks. It tries to detect malicious activities such as denial of service attacks, port scans etc. It examines the network packet in real time for detecting the intrusions and therefore it refers to the active component of the system.

In this paper will we discuss about the Network based intrusion detection system, its strategic importance, methods and tools. Further the challenges and issues regarding the network based intrusion detection are discussed.

2. Strengths and Limitations of NIDS

2.1. Strengths

2.1.1. Lower Cost Network based intrusion detection system are deployed for each node segment in a network. This IDS is responsible for monitoring the network traffic for all the nodes in the network. This kind of architecture nullifies the need of loading the software at different hosts in the network segment, this in turn reduces the management overhead as there is no need to maintain software at the host level.

2.1.2. Easier to Deploy Network based intrusion detection are operating system independent and therefore they are easier to deploy as they don't affect the existing infrastructure of the network. Also, network sensors will monitor for all types of attacks in the network regardless of the type of the operating system the target node is running.

2.1.3. Detection of Network based attacks A network based IDS checks the network headers of all the packets travelling in the network. Therefore, it easily detects the network based attacks which the host based IDS fails to detect. Attacks like TCP SYN flood, denial of service attack, fragment packet attack etc can be identified by looking at the network header of the packet. Network based IDS is capable of detecting such attacks in real time by just checking the header information of the network packet.

2.1.4. Retaining Evidence Network based IDS do real time intrusion detection by analysing the live network traffic. Therefore most of the attackers cannot remove the evidence of attack. This information can also be used for forensic investigation. While on the other hand, a host based intrusion detection system detects the attacks by analysing the log files of the system. And most of the attacker remove the log files after finishing the attack.

2.1.5. Quick response and Real time detection Network based IDS works in real time so they can detect any malicious activity as they occur. Based on how the node is configured, such attack can be stopped even before they can compromise the system.

2.1.6. Detection of failed attacks A network based IDS deployed outside the firewall is capable of detecting malicious attacks on resources behind the firewall, even though the firewall may be rejecting these attempts. This information can be very useful in forensic analysis.

2.2. Limitations

2.2.1. False Positives One major limitations of network intrusion detection system is frequency of false positives. A false positive is an event when NIDS falsely raises a security threat alarm for harmless traffic. No IDS can completely eliminate the possibility of a false positive. However most NIDS may be reconfigured so that a particular false positive does not continue to occur.

2.2.2. TCP Stream Reassembly In order to analyse a TCP/IP connection, the target network must keep track of all of the individual TCP or IP packets. It may happen that a set of TCP packets may arrive out of order and therefore the receiving network may reorder the packets by using the packet sequence numbers. Many attacks like tear drop attack attempt to confuse the process of stream reassembly by causing a buffer overflow through the use of malformed packets. The loophole lies in the fact that the first packet looks no different than an ordinary data packet, so the IDS does not detect the attack.

2.2.3 Encryption The increased use of data encryption in signature based NIDS is another limitation of NIDS. To ensure security, the data packets are encrypted before transmission. Once the payloads are encrypted the existing signatures will become useless in determining the harmful traffic.

3. Common Attacks and Vulnerabilities in Network Intrusion Detection system

Network Intrusion Detection system analyse the network for any evasion or malicious activity. They monitor the network traffic and generate alerts to the administrator if any intrusions are found. A large NIDS server can be setup at the backbone of the network and small systems can be set up at every host like switch, router or gateway.

Intrusion detection systems are implemented in network with increasing use of computing systems which may have potential threats and vulnerabilities. The constantly changing and evolving environment, the need for intrusion detection system has increased to maintain the pace with ongoing technologies. Current NIDS requires substantial amount of human interaction for effective operations. This leads to the importance of understanding the network architectures for the network administrators so that they can interpret the attacks and their mechanisms.

Unlike the host based systems, NIDS are mainly driven off by interpreting the raw network traffic. They detect attacks by examining the patterns of suspicious activity in the traffic. NIDS are capable of correlating attacks on multiple machines on a network and this makes them superior to host based intrusion detection system. But at the same NIDS also has one limitation. They are very poor at determining the actual process occurring on the host computers. For instance it is very difficult to judge what is happening on the system by just analysing the 'shell', 'login' and 'telnet' sessions.

Network intrusion detection system works by analysing the contents of the network packet. They collect the relevant information by parsing the network packets and examining the protocols used for their transmission. And this is achieved by passively monitoring the network. This passive monitoring has an advantage of promiscuous mode which is referred to as 'sniffer'. It is unobtrusive and passively monitors the data packets at the lowest network operation

3.1. Attack Types

- *Confidentiality*: Attacker gains unauthorized access to confidential data or the data which is not accessible.
- *Integrity*: Attacker alters the data without authorization.
- *Availability:* Attacks which breaks the system and make it unavailable to the users. For instance, Denial of service attacks.
- *Control*: Attacks gains full access to the system remotely and can change the access privileges, triggering the above three attacks types

3.2. Common Attacks Detected by NIDS

3.2.1. Scanning Attack This attack is carried out by sending probe packets to the target system in which the vulnerability can be exploited. As soon as the probe packets are sent to the target system, the system responds back. Attacker analyses the response packets to find out the vulnerabilities in the target system and to determine its characteristics. To accomplish this task, attacker often uses port scanner, network scanner or vulnerability scanner which yields information like:

- Network topology of the target system
- Type of firewall used by the target system
- Open ports and vulnerabilities
- Operating system, server applications and software used by the system.

Some well-known malicious scanning include Vertical and Horizontal port scanning, ICMP (ping) scanning, very slow scan, scanning from multiple ports and scanning of multiple IP addresses and ports. NIDS signatures can be devised to identify such malicious scanning activity from a legitimate scanning activity with fairly high degree of accuracy

3.2.2. Denial of Service Attack It is a very common type of attacks where a network of computers is used to send multiple dummy requests to the web server resulting in breakdown of the system. This attack usually slows down or completely shut down the target system so that the legitimate and authorized users can't access the resources. There are many types of denial of service attacks for instance: flaw exploitation DoS attacks in which the attacks exploits the flaw in the server software say for example sending a ping request with larger packet size may crash the target computer. And flooding DoS attacks in which more number of requests are sent to the target system than it can handle.

3.2.3. Penetration Attacks These attacks give the control of the target system to the attacker, who can modify or alter the system files and even change the privileges of the user. The most common types of penetration attacks are:
User to root: A local user gets the full access to every component of the system.
Remote to user: A user across the network gains a user account and the associated controls.
Remote to root: A user across the network gains the complete control of the system.
Remote disk read: An attacker on the network gains access to the inaccessible files stored locally on the host.
Remote disk write: An attacker on the network not only gains access to the inaccessible files stored locally on the host, but can also alter them.

3.3. Vulnerabilities

Since most of the attacks exploit a known or unknown vulnerability in the computer, a NIDS usually reports the type of vulnerability that an attacker is trying to exploit. Such information is

important in keeping the system up to date, fixing the bugs and eliminating the vulnerability. Here we discuss some of the well known vulnerability which has been exploited in the past.

3.3.1. Buffer Overflow A buffer overflow is vulnerability in a program which results in illegal program termination or memory access exception. Buffer overflow is practiced mainly to compromise the system security, for instance the Morris worm, the Code Red worm and the SQL Slammer worm are the result this vulnerability. Also buffer overflows present in licensed Xbox games have been exploited to allow unlicensed software, including homebrew games, to run on the console without the need for hardware modifications, known as modchips. Buffer overflow exploits are generally well fingerprinted and all known exploits have fairly accurate signatures.

3.3.2. Input Validation Error In this vulnerability, the system does not check the input for integrity and correctness before processing them. This can lead to a number of exploits, wherein an attacker can send a specific sequence of inputs that will lead to either the failure of the system, or will give the attacker an unauthorized access. Again a NIDS can easily detect such events, and raise an alarm.

3.3.3. Boundary condition error Boundary condition error is a form of input validation error where the input results in the system crossing some security boundary. For example the system may run out of memory, disk space, and network bandwidth. A simple example of such vulnerability is "divide by zero", wherein a careless implementation may lead to a crash. A NIDS can detect such conditions, and take appropriate actions.

3.3.4. Access control vulnerability This might arise due to the faulty implementation of the access control. This may include giving an unauthorized access to a user, or providing illegitimate remote access between two separate network domains. A NIDS can effectively tackle such exploits by examining the IP and application level headers, and checking the source and destination hosts. Access control vulnerability may also arise due to configuration errors, which can also be detected with the appropriate NIDS signatures.

4. Types of NIDS

The NIDS can be divided into two parts basically Signature Based NIDS and Anomaly Based NIDS[1]. The differences between both of the classification is in the way how they detect the intrusion in the network. Signature Based NIDS as the name suggests matches the signature of malwares from its database if it finds a match then raises an alarm. On the other hand Anomaly Based NIDS uses Machine Learning techniques to find any deviation in the regular behaviour of the network. In this section we will study these classifications of NIDSs in detail of their working and will compare them.

4.1. Signature Based NIDS

The network can be targeted with various kinds of attacks like DoS, ftp write, syn flood etc.. There is a vast compilation of such attacks in [2] Kendall has divided these attacks into four major categories named: Remote to local (R2L), Probes, User to root (U2R), and Denial of service (DoS). To prevent from such attacks the IDS must be given the patterns of connection and the attacks as well. Pattern of network cannot be determined by observing one packet, is must be a series of data packet for a particular protocol. Then there must be patterns for the attacks also in categories as mentioned above.

The signature based IDSs uses pattern matching algorithms to find the attack or malicious activity over the network. Time intervals and multiple parallel sessions between two hosts are the conditions which needs to be taken into consideration for matching patterns. Lee and Stolfo [3] defined a set of connection features between temporal and statistical relations and those were tested on 1999KDD Cup datasets.

As we discussed the Signature Based NIDS uses pattern matching for detection. There are basically two types of pattern matching viz. *Exact String Matching* and *Approximate String Matching*. The name itself suggests their way of workings. Researchers have proposed lots of algorithms proposed in pattern matching; We will discuss here few of them which are most widely implemented in NIDSs

4.1.1. Boyer-Moore Algorithm [4]: It is considered as a benchmark for practical pattern matching algorithms. The major advantage of this algorithm is that it starts the search process from the tail of the given string instead of going from the head. Also it jumps more than one character at a time unlike other algorithms which uses character by character matching. The overview of the algorithm can be describe in following steps:

- Start from the last character of the string.
- Match: Continue
  - All match: pattern found at this location
- Not match: does the character present in the string
  - Yes: Shift to closest such character from the present location.
  - No: Skip the string

- Continue Some implementation of Snort uses this algorithm partially for pattern matching

4.1.2. Aho-Corasick Algorithm [5]: It is pattern matching algorithm which uses character by character matching to find a sub-string in long string., running time of this algorithm is linear depending on the size of the input string. This was implemented using Finite Automata with a trie look alike structure. It starts with the head of the given string, when the pattern is not present in the string the root node represents as blank or null. Sub consecutive nodes are added as the pattern seems present in the input string. A lot of research work has been done with this algorithm to use it in NIDS they have came up with some modification of Aho-Corasick algorithm. [6] shows 30% better performance of the same algorithm; authors used path compression to reduce the memory requirement of original Aho-Corasick algorithm. In [7] authors again try to reduce the space requirement of the base algorithm using multiple binary state machines. FPGA-based design with character pre-decoding along with CAM-based pattern matching is another approach based on the same algorithm.

4.1.3. Regular Expressions Signatures Some researchers found that it would be more efficient and flexible to use Regular Expressions to represent signature in NIDS rather than using exact match approaches. Regular Expressions(RE) are more expressive as they uses unions, closures etc. Finite Automata are the best way to represent RE, these automata are implemented in various ways. Researchers has proposed many hardware specific implementations of such automata.

Automata are of two types: Deterministic Finite Automaton (DFA) and Non-deterministic Finite Automaton (NFA). NFAs are smaller in size as compared to DFAs because of their non deterministic nature. These REs are implemented on-chip using logics and those are hard to update and re program. This make it hard to use in present where there are new kinds of attacks are being launched with difference in their signatures. Because of this drawback REs using memory instead of logic are being preferred in practical implementation of this approach. Some implementations of this approach are TippingPoint X505 and a family of network security appliances from Cisco Systems

4.2. Anomaly Based NIDS

There is a fundamental drawback in Signature Based NIDS that is they cannot detect Zero Day Attacks or the new kind of attacks whose signature is not available a-priory. To overcome such situations another approach has been proposed called as Anomaly Based NIDS. It is also referred as the future of the IDSs, as they can detect the attack on the go. These process in general involves two steps: First one is what is called training phase, in which we the IDS is trained that how a normal network traffic behaves and what are its characteristics are. The other phase is called detection phase in which the IDS is deployed in the network and it start detecting the abnormal behaviour of the network. There are majorly three types of Anomaly Based IDS(A-IDS) statistical based, machine learning based and data mining based. We will discuss about them further in following subsections.

4.2.1. Machine Learning Based A-IDS: This type IDS uses machine learning techniques to train themselves about the network and the tries to find out the anomaly in that later. Though they are said to be self trained but there is always some kind of human interaction is always required to train the system initially. There are various machine leaning algorithms proposed with their own advantages and drawbacks. The most generic ones are: Bayesian networks, Genetic algorithms, Fuzzy logic, Markov model, Neural networks etc. So as the example list shows that almost all major machine learning techniques can be used in A-NIDS. As far as the implementations of such A-NIDS is concern we do not have any commercial tool present at this time but there are manufacturers working on this.

4.2.2. Data Mining Algorithm Based A-NIDS: Taking huge amount of data in input and making profile out of those data is the basic concept behind this technique. Once the data in fetched into the system it derives a normal behaviour of the network from it, and becomes ready for testing, in testing phase if anything happens which can be defined as deviation from the normal behaviour it rings the alarm. For a while in testing phase the system is monitored and some false alarms are rectified; the system adds such cases into its dataset and learns that they should not be notified as alarms. By continuing this process system becomes more and more intelligent and reduces the false alarm rates. Various data mining techniques are used to get the profile or behaviour of the network from input data.

4.2.3. Statistical Based A-NIDS: This type of A-NIDS is most exhaustive type one can say. This uses large number of statistics about the network and its traffic, then is waits for the anomaly in the network from the statistics. A lot of arguments have been made against this approach as it is very easy for a smart attacker to keep his/her presence un-detectable from this type of approach. For example suppose there is an attacker who want to scan open ports on a network systems, so the NIDS based on this approach will detect it on the basis of

number of such port request from the attacker. But if the attacker keeps patient and send only limited number of requests in certain time period then he can be un detectable from the IDS system. Lakhina[8] proposed a way to handle such situations, by using entropy. This makes the system analyze the traffic feature relations and is more successful to detect the attacks. There is another proposal in [9] which suggest address correlations. According to its authors the system must analyse the packet headers and not the payload; wavelet analysis [10] can be used to find co-relations among the headers and thus system will be able to detect the attacks.

5. Case Study: Snort

It is an open source Network Intrusion Detection System that tries to detect malicious activities over the network. Snort uses rule based language combining signature, protocol and anomaly inspection methods. It can act as a packet sniffer, packet logger, Honeypot monitor and NIDS as well. It is a lightweight, small, flexible and highly capable system.

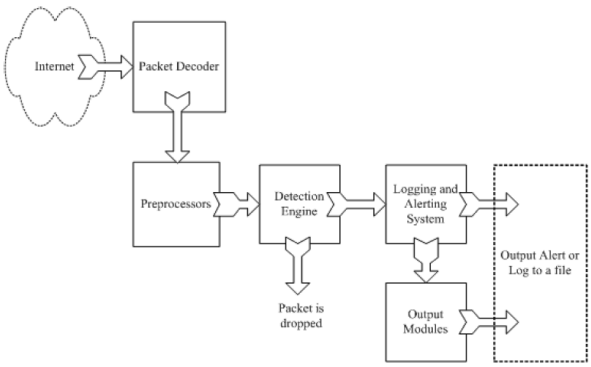

Fig: Architecture of Snort[11]

5.1. Snort Logical Components

Packet Detector: It is the module which is directly connected to internet and all the packet goes through this module. The job of this module is to receive the packets from various network interfaces and then prepare them for preprocessing.

Preprocessor: Once the packets are received by Packet Detector module it transfers those packets to the preprocessor module; the job of preprocessor is to analyze the packet headers and detect any anomaly in the header part only. It includes packets receiving from malicious IP, port no etc. Packets are defragmented here before sending to next module. Decoding of HTTP URI etc are done in this phase itself. Then the packets are sent to next module which is Detection Engine.

Detection Engine: This is the most important module of Snort. It is where the packet payload is analyzed and checked for the malicious data. This module analyses the IP Header of the packet, and then transport layer headers like TCP, UDP, ICMP etc according to the protocol used in data transmission. Afterwards the application layer headers are removed and examined. If the headers do not show any sign of malicious data then payload of the packet is thoroughly examined by the Detection engine.

It uses Boyer-Moore string matching algorithm to apply rules for detection of signatures of malicious data or attack. What it requires is time; to keep the time minimal there have been several variants of Boyer-Moore algorithm proposed as we discussed earlier in previous section. The time depends on other various constraints like number of rules; if there are more number of rules then it will surely take more time to check the signature against all those rules. Traffic load on the network; higher the load on the traffic is more it will take time to analyze the packets. Speed of machine; the machine where Snort is installed must be fast enough to run the algorithm efficiently, hardware support is crucial. Efficiency of the algorithm; as said earlier, there are various algorithms for pattern matching choosing the correct one according to requirement and hardware is as important as installing the NIDS.

Logging and Alert System: In case of finding the attack, the NIDS must log it and raise an alarm to the network administrator. Keeping log of the attacks helps to build signature more strong and sometimes it gives defence approach against similar kind of attacks. Alert must be sent to authorities so that they can take necessary steps to prevent the network from attack. Sometimes there are automated systems installed with NIDS, which on receiving alert acts accordingly in such conditions it is very important that the alert must be very specific.

Output Module: Once the log is generated and alerts been sent this module prepares the final output.

5.2. Challenges in Snort

There are few challenges in Snort:

Rules database is very huge; and it is growing by the time. In the version 2.3.2 there are more than 2600 rules, out of them around 80% are signatures only. Snort spends more than 75% of its total time in pattern matching

5.3. Improvements

Increasing the pre-processing ability by doing partial work from Detection Engine module. Some of the parts of the packets can be matched at the preprocessing phase itself, it will reduce the overhead from the detection engine. Another way to improve detection rate is to use specific hardware, as the software are more flexible but the hardware are gives more throughput, so if the system where software updates are not more likely important hardware can be made specific to the need to get more out of it. Detection algorithm will always be a scope for improvement as there may be more ways to make the pattern matching more efficient in terms of time and space as well. Organizing the rules into proper data structure may also give more throughput in performance of pattern matching.

6. Conclusion

In this survey paper, a brief overview of NIDS is described focusing on specifically two types of NIDS: Signature based and anamoly based. Advantages and limitations of NIDS are dicussed with a number of attacks detected by NIDS and common vulnerabilities. With mounting security concerns, the future of NIDS is surely promising. A centralized NIDS system should be employed in the host machines which can look for the malicious or abnormal behaviour in the network. Thus, the future scope of NIDS lies in standardizing the NIDS mechanisms.

References

[1] S. Axelsson. Research in intrusion detection systems: A survey. *Technical report*, Chalmers University of Technology, 1999

[2] K. Kendall. A database of computer attacks for the evaluation of intrusion detection systems. *Master's thesis*, Massachusetts Institute of Technology, June 1999

[3] W. Lee and S. J. Stolfo. A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information and System Security (TISSEC)*, 3(4):227–261,November 2000

[4] Boyer, Robert S., and J. Strother Moore. "A fast string searching algorithm." *Communications of the ACM* 20.10 (1977): 762-772.

[5] Aho, Alfred V., and Margaret J. Corasick. "Efficient string matching: an aid to bibliographic search." *Communications of the ACM* 18.6 (1975): 333-340

[6] Tuck, Nathan, et al. "Deterministic memory-efficient string matching algorithms for intrusion detection." *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. Vol. 4. IEEE, 2004.

[7] Tan, Lin, and Timothy Sherwood. "A high throughput string matching architecture for intrusion detection and prevention." *ACM SIGARCH Computer Architecture News*. Vol. 33. No. 2. IEEE Computer Society, 2005.

[8] Lakhina, Anukool, Mark Crovella, and Christophe Diot. "Mining anomalies using traffic feature distributions." *ACM SIGCOMM Computer Communication Review*. Vol. 35. No. 4. ACM, 2005.

[9] Kim, Seong Soo, AL Narasimha Reddy, and Marina Vannucci. "Detecting traffic anomalies through aggregate analysis of packet header data." *Networking 2004*. Springer Berlin Heidelberg, 2004.

[10] Graps, Amara. "An introduction to wavelets." *Computational Science & Engineering, IEEE* 2.2 (1995): 50-61.

[11] Rafeeq Ur Rehman, *Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID*