IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Special Issue || Dec. 2019 || PP 66-70
International Conference on Management Practices, Innovations & Research 2019

"Survey Architecture for Network Intrusion Detection and Prevention"

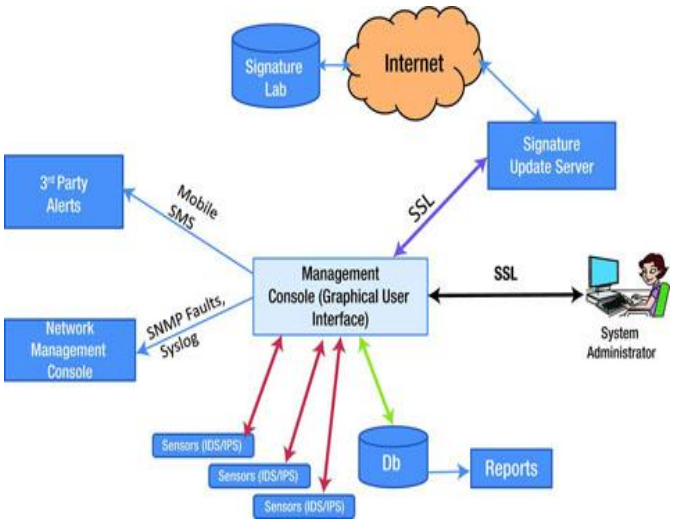
[1]monali Bodkhe, 2Gurudev Sawarkar
VM Institute of engineering & Technology RTMNU,NAGPUR,
Assistant Professor VM Institute of engineering & Technology RTMNU,NAGPUR,

Abstract: *This paper presents an investigation, involving experiments, which shows that current network intrusion, detection, and prevention systems (NIDPSs) have several shortcomings in detecting or preventing rising unwanted traffic and have several threats in high-speed environments. It shows that the NIDPS performance can be weak in the face of high-speed and high-load malicious traffic in terms of packet drops, outstanding packets without analysis, and failing to detect/prevent unwanted traffic.*
Since new threats are potentially more lethal, a number of pro-active designs have been proposed, which can detect new security events such as propagation of a new and unknown virus or worm. Such systems accomplish this by creating a profile of normal Internet traffic, and then using this profile to continuously monitor the network activity for suspicious activity. As the system senses an anomaly, or a dramatic change in traffic characteristics, it takes certain actions such as raising an alarm or discarding certain traffic. In this Survey paper, we will evaluate a number of current NIDS systems and the algorithms they employ to detect and combat security threats, both from technical and economical perspective.
Keywords: *NIDS, Anomaly Detection, Network Security, Security Signature, Pattern Matching*

I. Introduction

A NIDS aims at detecting possible intrusions such as a malicious activity, computer attack and/or computer misuse, spread of a virus, etc, and alerting the proper individuals upon detection. A NIDS monitors and analyzes the data packets that travel over a network looking for such suspicious activities. A large NIDS server can be set up[1] on the links of a backbone network, to monitor all traffic; or smaller systems can be set up to monitor traffic directed to a particular server, switch, gateway, or router. Another class of NIDS can[2] be setup at a centralized server, which will scan the system files, looking for unauthorized activity and to maintain data integrity. There are two primary approaches to NIDS implementation: signature based, and anomaly detection based. [3]The first approach has become a commercial success. A signature based NIDS maintains a collection of signatures, each of which characterizes the profile of a known security threat (e.g. a virus, or a DoS attack). These signatures are used to parse the data streams of various flows traversing through the network [4] link; when a flow matches a signature, appropriate action is taken (e.g. block the flow or rate limit it). Traditionally, security signatures have been specified as a string signature, port signature and header condition signature.

Information technology (IT) in_uences almost every aspect of modern life. Today, various devices are available to meet users' requirements such as high machine processor speed, and fast networks. Alongside our increasing dependence on IT, there has unfortunately been a rise in security incidents. Threats and attacks may range from stealing personal information from a laptop or network server to stealing the most top-secret information stored on a Security Intelligence Service (SIS).



II. Literaturere view:-

In this experiment, WinPcap, Flooder packet and TCP replay tools were used to send flood traffic with signed (known) malicious UDP packets (255 threads per 1mSec) to a physical system at different speeds (see Table 1). The UDP malicious packets were[2][3] interspersed among other packets transmitted at varying speeds. The following rule has been designed to require Snort to detect (alert and log) any UDP threads or malicious packets that contain the variables 'ab.H0..OK..cdef' and time to live (TTL) 132 that comes from any source and port address and goes to any destination address and ports: Alert udp any any -> any any (msg: ''Detect Malicious UDP Packets''; ttl: 132; content:|' 61 62 C2 48 60 AE 97 4F 4B C3 63 64 65 66'|; Sid: 100004;). Flood traffic TCP/IP was sent in different bandwidths (Bps) with 255 malicious[4] UDP packets (threads) in interval packets with a delay of 1 microsecond (1 mSec). The NIDS rule was set up to check the pattern inside the packets and then detect only the malicious UDP threads when the two conditions of (TTL and content) are matched. As shown in Table 1 and Snort NIDS.Analysed every packet that reached the wire.When255malicious UDP packets were[5] [6]sent at a speed of 1 mSec with TCP/IP

Food traffic at 16 bytes per second (16Bps), Snort alerted and logged more than 99% of the total UDP packets that it analyzed. As the flood traffic (speed) was increased to 200, 1200, 4800 and 60000 bytes per second (Bps), Snort alerted and logged packets to a decreasing degree, respectively, at 98.84, 97.17, 49.40 and 35.75% of the total malicious packets analyzed

In this experiment, TCP/IP flood traffic was sent at differing speeds (see Table 2) with 255 malicious UDP packets (threads)alsosentat1microsecond(1mSec)intervals.Snort was set to prevent UDP threads by[6] using two rule conditions (TTL and content) as follows:[7] reject udp any any ->any any (msg: ''Prevent UDPPackets'';ttl:120;content:|'C24860AE974F4BC3 '|; Sid: 100007;). Use of these options will prevent any UDP malicious packet that is matched with the TTL value equal to 120 and a data pattern inside the malicious [8]packet with content ''.H'..OK.''. The hexadecimal number ('C2, 48, 60, AE, 97, 4F, 4B, C3'), which the rule contained, is [7]equal to the ASCII characters ('., H0,.,.,., O, K,.'). As shown in Table 2 Figure 2, When 255 malicious UDP packets were sent at a speed of 1 mSec and TCP/IP flood traffic at 100 bytes per[8] second (Bps), Snort prevented 100% of the total UDP packets that it analyzed.

III. Intrusion Detection System:

Intrusion detection systems are an essential component of defensive measures protecting computer system and network against harm abuse [7].This system becomes important part in the cloud computing infrastructure. The main idea of IDS is to[7] detect attacks and provide the proper response [10]. IDS can be defined as the technique that is used to detect and response to intrusion activities from network or host [8].

Intrusion detection system can be divided into two main categories. [9]They are Host Based Intrusion Detection System (HIDS) and Network Based Intrusion Detection System (NIDS). The IDS can be supposed as a defense system, which can detect hostile activities in the network. It can compromise system security. and prevent the malicious activities. The main feature of intrusion[6] detection system is to provide a view of unusual activity and to issue alerts notifying administrators or blocking a suspected connection. Host Based intrusion Detection System (HIDS) includes [7] software or agent components. It can run on the server, router and switch or network appliance.

Network Based Intrusion Detection System (NIDS) collects network traffic packets such as TCP and UDP. NIDS analyzes the [9]content against a set of RULES or SIGNATURES to determine if a POSSSIBLE event took place. HIDS and NIDS are needed in the Cloud computing environment, which they offer significantly different benefits. For IDS, it is needed to use detection, attack anticipation and prosecution [3] [8]. With the description of the[9] above 3 popular NIDS deployment configurations, we proceed to the NIDS architecture and the algorithms that are used to implement NIDS. NIDS has traditionally been designed with two popular techniques: a signature based detection and a relatively advanced implementation called anomaly based detection. We begin with the description of the signature based design.

Types of IDS	Description	Pros	Cons
Host-Based	Host-based IDSs are installed on a specific machine such as a server and mobile devices that monitor the operating system's audit information for any sign of intrusion. In addition, they detect which programs are accessing which part of the system or resources.	• At the transport layer, it monitors network traffic. • Does not require additional hardware. • Can deal with switched and encrypted environments. • Can help with the detection of a Trojan horse.	• Formation at a host may cause severe limitation-of-the network. • Any other attacks can involve software integrity breaches.
Network-Based	Network-based IDSs monitor network traffic and application protocol activity between any two computers for any type of intrusion.	• Cost-effective. • NIDS can detect attacks that are skipped by HIDS. • Allows for quick response. • It is easy to deploy as it does not affect existing infrastructures.	• It is far from the individual host. • Unable to monitor and analyze encrypted packets. • Requires full-time monitoring.
Hybrid	This is combination of both HIDS and NIDS components using mobile agents and a combination of anomaly- and misuse-based approaches. A system log file checker is performed by the mobile agent traveling to each host, while the overall network can be checked by a central agent for the existence of anomalies.	• Provides in-depth defense. • Gives administrators the ability to quantify attacks. • Provides an additional layer of protection. • Provides protection for the entire network.	• Generates false positives and false negatives. • Reacts to attacks rather than prevents them. • Generates an enormous amount of data to be analyzed. • It is most expensive.
Distributed	Various IDS (HIDS and NIDS) are combined by working as faraway sensors and constructing a report about intrusions. Later submits report to a centralized control, called distributed IDS. Uses remote sensors that can be host-based, network-based or even a combination of host- and network-based.	• It utilizes traffic information from various sources. • Monitoring is controlled by a central server. • Detection and response are also monitored from a central point. • Facilitates advanced network monitoring, incident analysis, and instant attack data.	• The flow of data may generate huge network movement overheads. • The system uses data packets that may be obtained from a network. • A program can be edited or interrupted by an intruder.

3.1 Figures and Tables

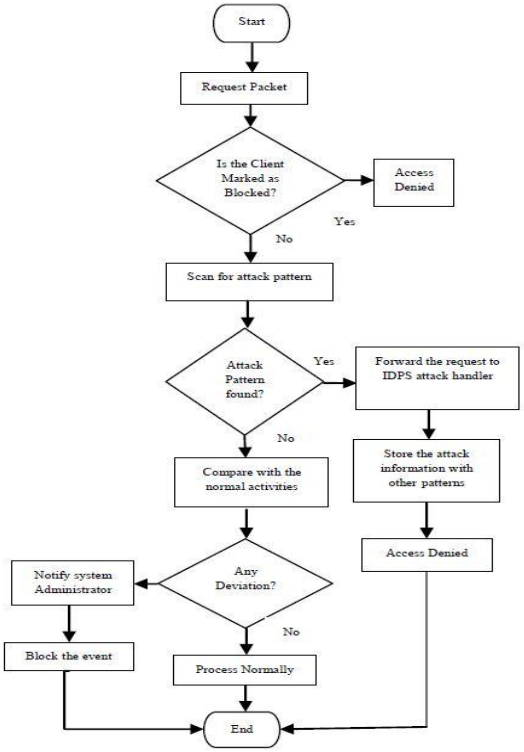
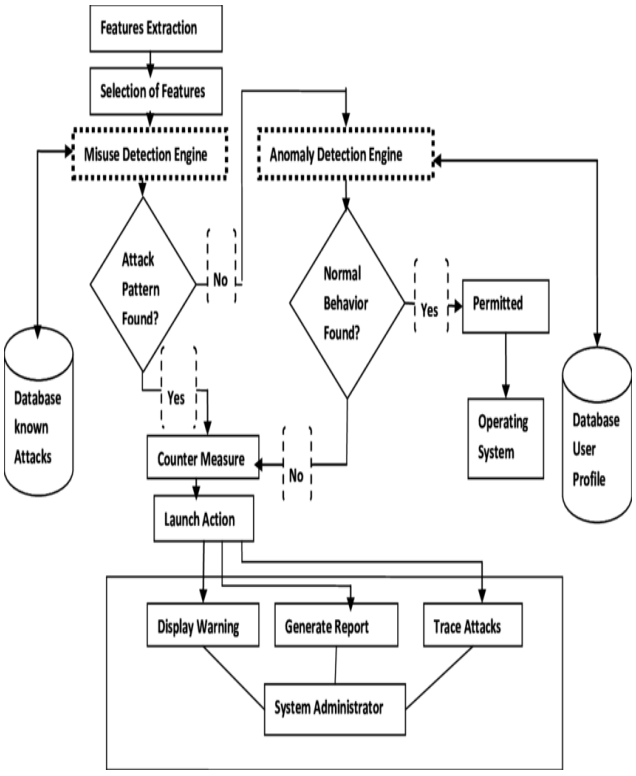


Fig .IDPS Activity Framework

Intrusion detection and prevention system (IDPS) which is an integrated model that consists of two techniques (AD) and (SD). When IDPS are placed behind a firewall, it can introduce new security risks to the internal network, especially if the internal network is not secured against the IDPS through additional firewalls. There is important to distinguish between a setup where the firewall enables access to the IDPS or where access from the Internet is denied. A Honeypot does provide a lot of services and also most of them are not used as exported services to the Internet. They are not forwarded to the IDPS by the firewall. By placing the IDPS behind a firewall, it is inevitable to adjust the firewall rules if access from the Internet should be permitted [1]. The proposed integrated system detected any of the attacks and compare it with the know threats (signature) and produce an alarm in the case of matching according to Signature Based Detection technique.

IV. Proposed System:

The results of the experiments described above in section (II) show that the NIDPS's performance decreases when faced
with heavy and high-speed attacks. This section analyses the problem and then outlines a novel solution to increase NIDPS performance in the analysis, detection, and prevention of malicious attacks.



4.1. Novel Nidps Architecture

Critical analyses were done for the experiments presented in sections II(A) and II(B) (see Figures 1 and 2, respectively). The Figures show that performance of NIDPS throughput is affected when NIDPS is exposed to a high-volume and speed of traffic; more packets will be dropped and left outstanding as the speed of traffic increases. Figure 1 shows that the NIDPS's detection performance decreased when the traffic speed increased. There were more missed alerts and missed logs for packets as the speed of traffic increased. the NIDPS prevention performance decreased when traffic speed increased. When traffic moves through the network interface card (NIC) to the NIDPS node, the packets are stored in the buffer until the other relevant packets have completed transmission to processing nodes. In the event of high-speed and heavy trafficin multiple directions, the buffer will _ll up. Then packets may be dropped or left outstanding [10]_[9]. In this case, there is no security concern about the packets dropped; the packets are dropped outside the system. The existence of outstanding packets that are waiting or have not been processed by a security system (i.e. NIDPS node) affects the system ef_ciency however. Packets can also be lost in a host-based IDPS. Most software tools use a computer program such as the kernel, which manages input/output (I/O) requests from software and decodes the requests into instructions to direct the CPU's data processing. When trafficmoves from the interface (NIC) through the kernel's buffer to the processor space, where most of processing nodes are executed, the packets will be held in the kernel buffer before being processed by the CPU. When some nodes experience a high-volume of data, the buffer will _ll

up and packets may be dropped. There are therefore three (3) places where packets could be dropped: in the network, in the host or in the processor, because all of them are dependent on buffer size and processing speed.

V. Algorithm:

Pseudo code:

Proposed Intrusion Detection Algorithm to find density variance.
Input: Data set of size n, natural no k and threshold value .
Output: Density for each data points and outliers
object.
IDA(H-Distance, T)
For i=1 to n
For j=1 to n
H.D[i,j]= A[i,j] XOR B[i,j]
End For
End For
For k = 1 to n

N (x) H-distance(p) =no of data objects with H-distances less than or equal to Abs H-distance(x)
H-Density=$N_{H\text{-}distance}$ (x)/ H-distance(x)

Density Variance(DV)= $(1/ N_{H\text{-}distance}(x))^* \sum_x (H\text{-Density}(x)-H\text{-Density})^2$

(where H-Density is the mean density.)
If DV[x] <Threshold T
Then x is intruder.
Else
x is normal data.
End if
end_for
end

VI. Conclusion:

In this survey paper, we describe the design and architecture of a number of different NIDS and the various configurations, in which they are employed in the network. Specifically we focus on two important classes of NIDS signature based and anomaly based. We thoroughly investigate their benefits and drawbacks, and discuss a number of attack and vulnerabilities than they can combat. Finally we discuss the future trends in this space, where we argue that a more distributed version of NIDS is on the horizon and that the NIDS mechanisms need to be standardized.

References:

[1]. R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunninghan, and M. Zissman, "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation," DARPA Information Survivability Conference and Exposition, Jan 2000.

[2]. R. Lippmann, J. Haines, D. Fried, J. Korba, and K. Das., "Analysis and Results of the 1999 DARPA Off-line Intrusion Detection Evaluation," Recent Advances in Intrusion Detection (RAID 2000), Oct 2000.

[3]. J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner, "State of the Practice of Intrusion Detection Technologies," Technical Report CMU/SEI-99-TR-028, CMU/SEI, 2000.

[4]. T. H. Ptacek and T. N. Newsham, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," Technical report, Secure Networks Inc., Jan 1998.

[5]. V. Paxson, "Bro: A System for Detecting Network Intruders in Real-time," Computer Networks, 31(23-24), Dec 1999.

[6]. L.G. Roberts, "Beyond Moore's Law: Internet Growth Trends," IEEE Computer, pp. 117-119, Jan 2000.

[7]. P. A. Porras and P. G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," National Information Systems Security Conference, Oct 1997.

[8]. G. Vigna, R. A. Kemmerer, and P. Blix, "Designing a Web of Highly-configurable Intrusion Detection Sensors," Recent Advances in Intrusion Detection (RAID 2001), Oct 2001.

[9]. J. Balasubramaniyan, J. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents," 14th IEEE Computer Security Applications Conference, pp. 13-24, Dec 1998.

[10]. R. Gopalakrishna and E. H. Spafford, "A Framework for Distributed Intrusion Detection Using Interest- driven Cooperating Agents," Recent Advances in Intrusion Detection (RAID 2001), Oct 2001.