ISSN:0975-9646

Sapna S Kaushik et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 982-986

Detection of Attacks in an Intrusion Detection System

Sapna S. Kaushik[#1], Dr. Prof.P.R.Deshmukh[#2]

M.E. II Year, Computer Science and Engg., Sipna College of Engg. Amravati, INDIA

Professor & Head ,CSE and IT , Sipna College of Engg. Amravati, INDIA

Abstract

Intrusion detection is the act of detecting unwanted traffic on a network or a device. A intrusion detection system (IDS) provides a layer of defense which monitors network traffic for predefined suspicious activity or patterns, and alert system administrators when potential hostile traffic is detected. Intrusion detection faces a number of challenges; an intrusion detection system must reliably detect malicious activities in a network and must perform efficiently to cope with the large amount of network traffic.Network based intrusion detection are the most deployed IDS. An IDS can be a piece of installed software or a physical appliance. Many IDS tools will also store a detected event in a log to be reviewed at a later date or will combine events with other data to make decisions regarding policies or damage control. This paper discusses the various types of attacks that can be detected in a simulated network environment. The different types of attacks are Probe attacks,R2L,Dos and U2R attacks.

Keywords **– Intrusion Detection System (IDS), Probe attacks, Dos (Denial of Service) attacks,R2L (Remote to Local) attack, U2R (User to Root) attack,HIDS,Signature and Anamoly Based IDS.**

I. INTRODUCTION

Several types of IDS technologies exist due to the variance of network configurations. Each type has advantages and disadvantage in detection, configuration, and cost.

NIDS (Network Intrusion Detection Systems)

Network Intrusion Detection Systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. Ideally one would scan all inbound and outbound traffic. NIDS analyzes network traffic at all layers of the Open Systems Interconnection (OSI) model and makes decisions about the purpose of the traffic, analyzing for suspicious activity. Most NIDSs are easy to deploy on a network and can often view traffic from many systems at once.

HIDS (Host Intrusion Detection Systems)

Host Intrusion Detection Systems are run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator if suspicious activity is detected. HIDS analyze network traffic and system-specific settings such as software calls, local security policy, local log audits, and more. A HIDS must be installed on each machine and requires configuration specific to that operating system .

Signature Based

A signature based IDS will monitor packets on the network and compare them against a database of signatures or patterns of known malicious threats. This is similar to the way most antivirus software detects malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to your IDS. During that lag time your IDS would be unable to detect the new threat.

Anomaly Based

An IDS which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline will identify what is "normal" for that network, what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other and alert the administrator or user when traffic is detected which is anomalous or significantly different than the baseline.

II.SUBSYSTEMS OF IDS

There are three primary subsystems that make up intrusion detection system: the packet decoder, the detection engine, and the logging and alerting subsystem. These subsystems will provide a portable packet sniffing and filtering capability. Program configuration, rules parsing, and data structure generation takes place before the sniffer section is initialized, keeping the amount of per packet processing to the minimum required to achieve the base program functionality.

2.1Packet Decoder

The decode engine will be organized around the layers of the protocol stack present in the supported data-link and TCP/IP protocol definitions. Each subroutine in the decoder imposes order on the packet data by overlaying data structures on the raw network traffic. These decoding routines get called in order through the protocol stack, from the data link layer up through the transport layer, finally ending at the application layer. Speeds get emphasized in this section, and the majority of the functionality of the decoder consists of setting pointers into the packet data for later analysis by the detection engine. It will provide decoding capabilities for Ethernet, raw (PPP) data-link protocols.

2.2Detection Engine

System maintains its detection rules in a two dimensional linked list of what will be termed Chain Headers and Chain Options. These are lists of rules that will be condensed down to a list of common attributes in the Chain Headers, with the detection modifier options contained in the Chain Options. For example, if forty five CGI-BIN probe detection rules are specified in a given detection file, they generally all share common source and destination IP

addresses and ports. To speed the detection processing, these commonalities are condensed into a single Chain Header and then individual detection signatures are kept in Chain Option structures. These rule chains will be searched recursively for each packet in both directions. The detection engine checks only those chain options which have been set by the rules parser at run-time. The first rule that matches a decoded packet in the detection engine triggers the action specified in the rule definition and returns.

2.3Logging and Altering

The alerting and logging subsystem will be selected at run-time. The logging options can be set to log packets in their decoded, human readable format to an IP-based directory structure, or in tcpdump binary format to a single log file. The decoded format logging will allow fast analysis of data collected by the system. The tcpdump format is much faster to record to the disk and should be used in instances where high performance is required. Logging can also be turned off completely, leaving alerts enabled for even greater performance improvements. Alerts may be sent to system log, logged to an alert text file in two different formats, or sent as popup messages.

III. THE DIFFERENT TYPE OF ATTACKS THAT CAN OCCUR IN AN IDS SYSTEM ARE

3.1 Probe attacks

The probe attacks are aimed at acquiring information about the target network from a source that is often external to the network. Hence, basic connection level features such as the "duration of connection" and "source bytes" are significant while features like "number of files creations" and "number of files accessed" are not expected to provide information for detecting probes.

3.2 DoS Attacks

The DoS attacks are meant to force the target to stop the service(s) that is (are) provided by flooding it with probes illegitimate requests. Hence, for the DoS attack to be detected, traffic features such as the "percentage of connections having same destination host and same service" and packet level features such as the "source bytes" and "percentage of packets with errors" are significant. To detect DoS attacks, it may not be important to know whether a user is "logged in or not."

3.3 R2L Attacks

The R2L attacks are one of the most difficult to detect as they involve the network level and the host level features. We therefore select both the network level features such as the "duration of connection" and "service requested" and the host level features such as the "number of failed login attempts" among others for detecting R2L attacks.

3.4 U2R Attacks

The U2R attacks involve the semantic details that are very difficult to capture at an early stage. Such attacks are often content based and target an application. Hence, for U2R attacks, features such as "number of file creations" and "number of shell prompts invoked," are selected while features such as "protocol" and "source bytes are ignored.

IV. IMPLEMENTATION FOR DETECTION OF ATTACKS

We have used the network simulator NS2 on FEDORA Operating System,We have simulated the network environment using NAM simulator. Data packets are sent from the attacker nodes to the victim node or nodes.Attacks are generated randomly using a random function. The type of attack generated is classified to be a Probe, R2L ,U2R or Dos attack. In case the attack was not generated then the classification would be as a normal packet. The packets that reach the victim are analyzed and the frequency of single characters and frequency of group of characters is displayed. The NAM visualization displays the packets going from attacker to victim nodes.

The result of implemented work is carried out by different simulations which are implemented to demonstrate the different types of intrusion detection in the different types of network architecture. Initially simnids.tcl is implemented to demonstrate network intrusion detection system where there are two attacker nodes, one sender node and one receiving node. For the execution of this tcl script initially all the environment variables are set and the following command is executed on the terminal.

ns simnids.tcl

After the execution out.nam file is created inside the current working directory and we get the following output on the terminal.

Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> asdfg
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> jkhgasdfg
R2L attack
Detected in 122.000000 Seconds.
Packet received
Transaction (set D) ==> locnnmasdfghf
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> oqwdjhkcmd
SET L1={ (a,3) (s,3) (d,5) (f,4) (g,4) (j,2) (k,2) (h,3) (o,2) (c,2) (n,2) (m,2) }
Ln={ (sda,0) (sds,0) (sdd,0) (sdf,0) (sdg,0) (sdj,0) (sdk,0) (sdh,0) (sdo,0) (sdc,0) (sdn,0) (sdm,0) (asd,0) (ssd,0) (dsd,0) (fsd,0) (gsd,0) (jsd,0) (ksd,0) (hsd,0) (osd,0) (csd,0) (nsd,0) (msd,0) }
Cn={ (sdf,3) (asd,3) }
Cn={ (sdfg,3) (asdf,3) }
Cn={ (asdfg,3) }

It can be observed from the output various types of attacks are detected in the network. The probe attack is generated when there is addition of some data bytes inside the original data which is send by the sender or when there is alteration of data. The R2L attack is detected when there is maximum connection duration. The DoS attack is detected when there is packet loss that means the packet doesn't receive to the destination node.

We can run the simulation by executing the following command on the terminal.

nam out.nam

After execution the output is generated inside the network animator. The result of which are shown below.

Fig 1 : NAM visualization of network

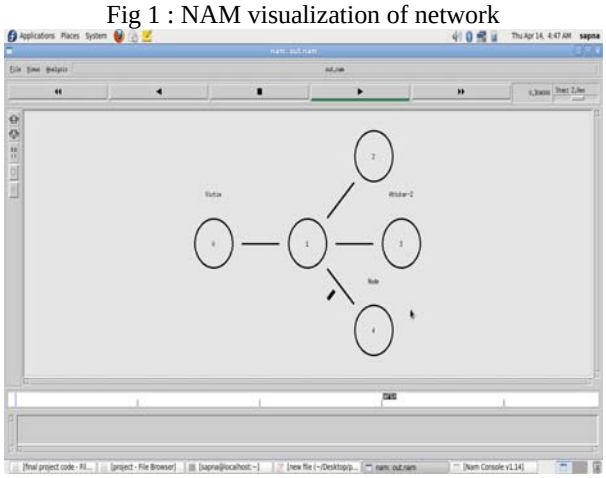
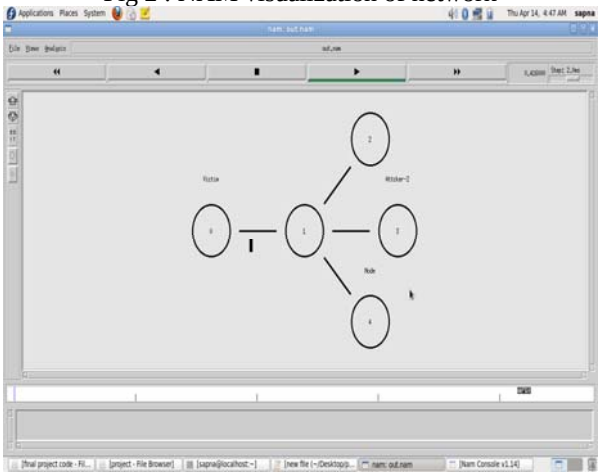


Fig 2 : NAM visualization of network



In the last simnidsnewmorenodes.tcl simulation network intrusion system with more nodes is demonstrated. After the first execution we get the following output.

ns simnidsnewmorenodes.tcl
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> efgh
DoS attack
Detected in 2.000000 Seconds.
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> abcdefghi
Normal Packet:1
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> jkhgasdfg
Normal Packet:1
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> efghijklmn
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> locnnmasdfghf
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> abmlopbxcv
R2L attack
Detected in 122.000000 Seconds.
Packet received
Transaction (set D) ==> oqwdjhkcmd
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> xyz
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> pqrstxyztr
DoS attack
Detected in 2.000000 Seconds.
Normal Packet:1
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> shfhsfhfgkl
SET L1={ (e,3) (f,9) (g,7) (h,9) (a,4) (b,3) (c,4) (d,5) (i,2) (j,3) (k,4) (s,5) (l,4) (m,4) (n,3) (o,3) (p,2) (x,3) (q,2) (y,2) (z,2) (r,2) (t,2) }
Ln={ (sde,0) (sdf,0) (sdg,0) (sdh,0) (sda,0) (sdb,0) (sdc,0) (sdd,0) (sdi,0) (sdj,0) (sdk,0) (sds,0) (sdl,0) (sdm,0) (sdn,0) (sdo,0) (sdp,0) (sdx,0) (sdq,0) (sdy,0) (sdz,0) (sdr,0) (sdt,0) (esd,0) (fsd,0) (gsd,0) (hsd,0) (asd,0) (bsd,0) (csd,0) (dsd,0) (isd,0) (jsd,0) (ksd,0) (ssd,0) (lsd,0) (msd,0) (nsd,0) (osd,0) (psd,0) (xsd,0) (qsd,0) (ysd,0) (zsd,0) (rsd,0) (tsd,0) }
Cn={ (sdf,2) (asd,2) }
Cn={ (sdfg,2) (asdf,2) }
Cn={ (asdfg,2) }

The result of simulation are shown in the following figures.

Fig 3 : NAM visualization of network

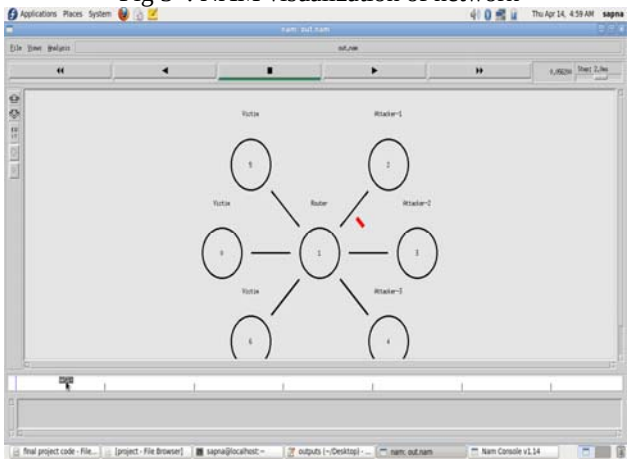
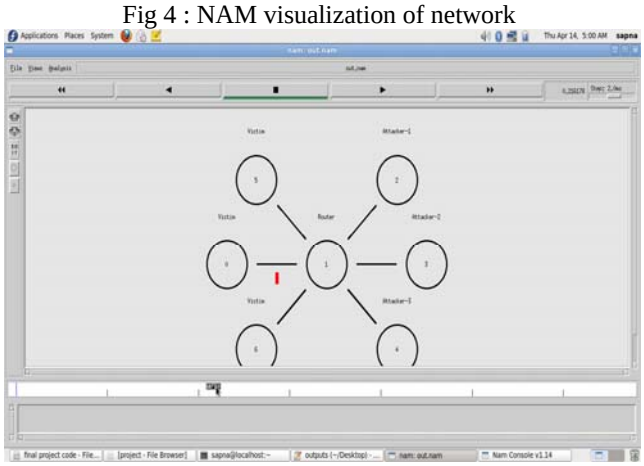


Fig 4 : NAM visualization of network



The next simulation is executed which detects the DoS attack. It can be observed from the displayed output that the indented packet doesn't reach the destination therefore service will not get processed. The output and its simulation is shown in the following figures.

ns simnidsnew.tcl
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> asdfg
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> jkhgasdfg
DoS attack
Detected in 2.000000 Seconds.
Probe attack
Detected in 2.000000 MiliSeconds.
Packet received
Transaction (set D) ==> oqwdjhkcmd

SET L1={ (a,2) (s,2) (d,4) (f,2) (g,3) (j,2) (k,2) (h,2) }
Ln={ (sda,0) (sds,0) (sdd,0) (sdf,0) (sdg,0) (sdj,0) (sdk,0) (sdh,0) (asd,0) (ssd,0) (dsd,0) (fsd,0) (gsd,0) (jsd,0) (ksd,0) (hsd,0) }
Cn={ (sdf,2) (asd,2) }
Cn={ (sdfg,2) (asdf,2) }
Cn={ (asdfg,2) }

Fig 5 : NAM visualization of network

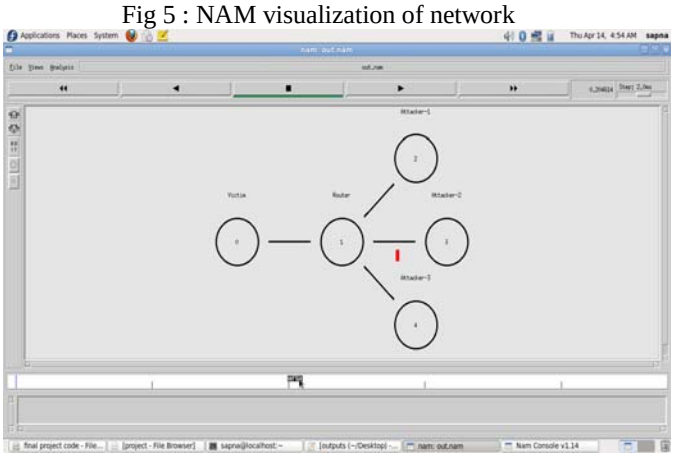


Fig 6 : NAM visualization of network

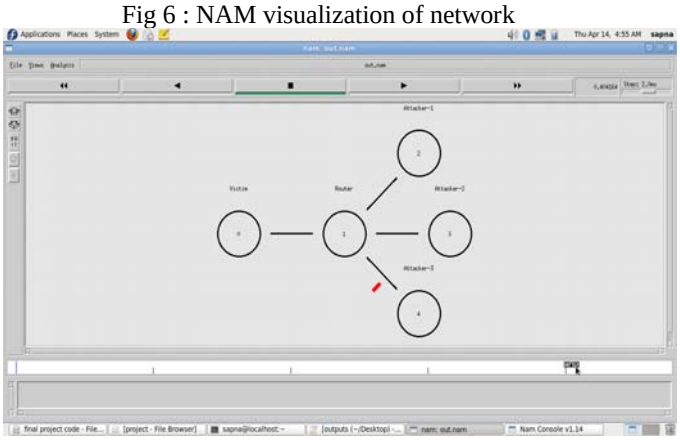
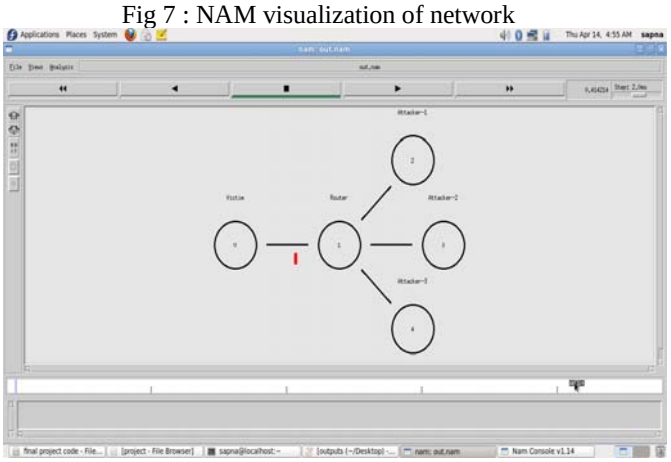


Fig 7 : NAM visualization of network



V. CONCLUSION

There are various approaches to detect the attacks in an Intrusion Detection System. Each of the approaches has its own advantages and disadvantages. Thus it is difficult to choose a particular method to implement an intrusion detection system over the other.New techniques keep emerging which will remove the drawbacks of the previous methods of implementation. Thus a judicious approach has to be made while selecting a mode to implement attack detection in an intrusion detection system.

REFERENCES

[1] Kapil Kumar Gupta, Baikunth Nath, Ramamohanarao Kotagiri ,"Layered Approach Using Conditional Random Fields for Intrusion Detection"

[2] www.snort.org

[3] Renaud Bidou "Denial of Service Attacks"

[4] John Bellardo and Stefan Savage "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions"

[5] Maheshkumar Sabhnani ,Gursel Serpen "KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection "

[6] Khaled Labib and V. Rao Vemuri "Detecting And Visualizing Denial-of-Service And Network Probe Attacks Using Principal Component Analysis"

[7] Vitaly Shmatikov and Ming-Hsiu Wang "Security Against Probe-Response Attacks inCollaborative Intrusion Detection "

[8] Kevin S. Killourhy, Roy A. Maxion and Kymie M. C. Tan " A Defense-Centric Taxonomy Based on Attack Manifestations "