A Java Based Network Intrusion Detection System (IDS)

Allam Appa Rao, P.Srinivas, B. Chakravarthy, K.Marx, and P. Kiran

Department of Computer Science and Systems Engineering,
Andhra University College of Engineering, India
allamapparao@gmail.com

Abstract

The number of hacking and intrusion incidents is increasing alarmingly each year as new technology rolls out. Unfortunately in today's digitally connected world, there is no place to hide. DNS, NSlookup, Newsgroups, web site trawling, e-mail properties etc. are just some of the many ways in which you can be found. In this research project, we designed and build an Intrusion Detection System (IDS) that implements pre-defined algorithms for identifying the attacks over a network. The Java programming language is used to develop the system, JPCap must be used to provide access to the winpcap. The packets in the network are captured online i.e., as they come on the interface of the network. The IDS is designed to provide the basic detection techniques so as to secure the systems present in the networks that are directly or indirectly connected to the Internet.

I. Introduction

An intrusion is somebody (“hacker" or "cracker") attempting to break into or misuse your system. The word "misuse" is broad, and can reflect something severe as stealing confidential data to something minor such as misusing your email system for spam (though for many of us, that is a major issue!). With the emergence of Internet and the World Wide Web, the concept of Global village has taken its inception. There are facilities to virtually achieve any kind of information on the internet. All these advantages have been achieved because of networking computers and associated devices. There has been a rapid progress in this field. Along with this, there is the arms race between the intruders and people who provide security to the systems in networks. This project IDS (detection and protection) [2, 3] runs on the host machines and assists the network Administrators to detect several intrusion attacks and inform to the owner of the system and also provide security by blocking the malicious users based on their IP addresses.

Network Intrusion:

A deliberate attempt to enter a network and break the security of the network and thus breaking the confidentiality of the information present in the systems of the network. The person who tries to attempt such an action is called as an Intruder and the action can be termed as Network

Intrusion. The network administrator is supposed to protect his network from such persons and this software can help his in his efforts.

Intrusion detection systems (IDS)

An Intrusion Detection System (IDS) is a system that is responsible for detecting anomalous, inappropriate, or other data that may be considered unauthorized occurring on a network. An IDS captures and inspects all traffic, regardless of whether it's permitted or not. Based on the contents, at either the IP or application level, an alert is generated.

Need for an IDS:

Intrusion detection devices are an integral part of any network. The internet is constantly evolving, and new vulnerabilities and exploits are found regularly. They provide an additional level of protection to detect the presence of an intruder, and help to provide accountability for the attacker's action.

Four different types of attacks have been identified which makes the need for an IDS critical.

Denial of service
Network-based denial-of-service [1, 2, 3] attacks are one of the easiest types of attacks. It often requires little effort to fully consume resources on the target computer, to starve the target computer of resources, or to cause critical services to fail or malfunction. Internal corporate networks typically do not have internal filtering defenses against common denial-of-service attacks, such as flooding.

Threat to Confidentiality
Some viruses attach themselves to existing files on the system they infect and they send the infected files to others. This can result in confidential [1] information being distributed without the author's permission.

Modification of contents
Intruders might be able to modify news sites, produce bogus press releases, and conduct other activities, all of which could have economic impact.

Masquerade
A masquerade [1, 2, 3] takes place when one entity pretends to be a different entity. Authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

Any system connected to the internet and providing TCP-based network services (such as a Web server, FTP server, or mail server) is potentially subject to this attack. Note that in addition to attacks launched at specific hosts, these attacks could also be launched against your routers or other network server systems if these hosts enable (or turn on) other TCP services (e.g., echo).

The consequences of the attack may vary depending on the system; however, the attack itself is fundamental to the TCP protocol used by all systems.

II. Requirements For IDS

The requirements to develop the system or software can be listed at two levels of abstraction.
High Level Requirements:

- To develop an application that is capable of sniffing the traffic, to and from the host machine.
- To develop an application that is capable of analyzing the network traffic and detects several pre-defined intrusion attacks and mappings.
- To develop an application that warns the owner of the host machine, about the possible occurrence of an intrusion attack and provides information regarding that attack.
- To develop an application that is capable of blocking traffic to and from a machine that is identified to be potentially malicious and that is specified by the owner of the host machine.

Low Level Requirements:

- To develop an application capable of displaying the traffic to and from the host machine in the form of packets to the owner of the host.
- To develop an application capable of detecting occurrence of Denial of Service attacks such as Smurf Attack and Syn-Flood Attack.
- To develop an application capable of detecting attempts to map the network of the host, using techniques such as Efficient Mapping and Cerebral Mapping.
- To develop an application capable of detecting activities which attempt to gain unauthorized access to the services provided by the host machine using techniques such as Port Scanning.
- To develop an application that maintains a "Log Record" of identified intrusion attacks done on the host in the present session and also displays it upon request.
- To develop an application that displays the list the active and inactive methods of which each scans for a specific intrusion attack.
- To provide options to activate or de-activate each of the Attack Detection methods.
- To provide an option to the user of the host to frame Rules which explicitly specify the set of IP addresses that are to be blocked or allowed. These Rules determine the flow of traffic at the host.

III. Purpose And Scope

Purpose of the system: The purpose of the system is to detect certain well-known intrusion attacks on the host system and display warnings to the user and also store information regarding the IP addresses and allow the traffic based on that information.

Scope of the system: The system frames certain rules based upon the input given by the user. It then allows traffic inwards or outwards based upon the rules. The system also detects certain well-known attacks and gives warnings to the user.

System overview

When packets arrive at the system, they are sniffed by the sniffer and then various processing techniques are applied to detect if any attack is being done, in which case a warning is given to the user.The attacks detected are predefined and well known. In this system the following attacks detection are implemented.

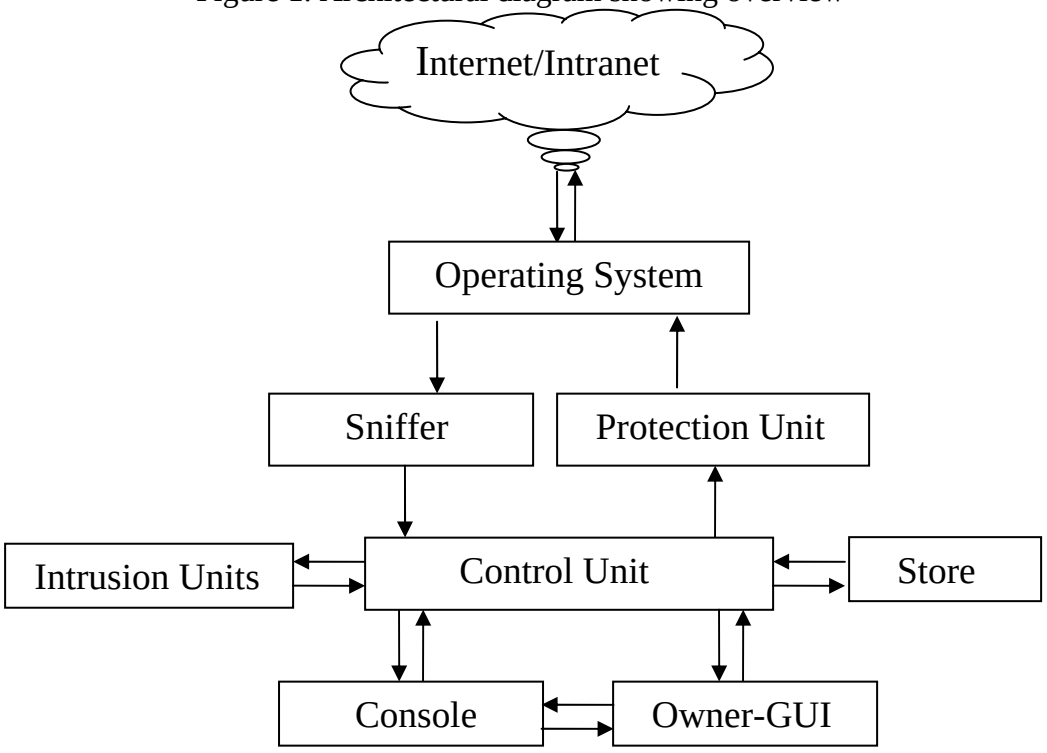
- Port Scanner,
- Smurf Attack,
- SYN Flood Attack,
- Efficient Mapping, and
- Cerebral Mapping.

The system has also provision to block traffic from a specific IP address which may have been recognized to be malicious or troublesome. There is also provision to allow traffic from specific IP addresses for some trusted systems, from which traffic is not monitored. Traffic from unknown hosts is monitored and any potential attacks are informed to the user.

System structuring

The Architectural Design is depicted as a block diagram where each box in the diagram represents a sub-system. The arrows indicate data or control flow in the direction specified by them. The Architectural block diagram presents an overview of the system architecture (Figure 1).

Figure 1. Architectural diagram showing overview



Description

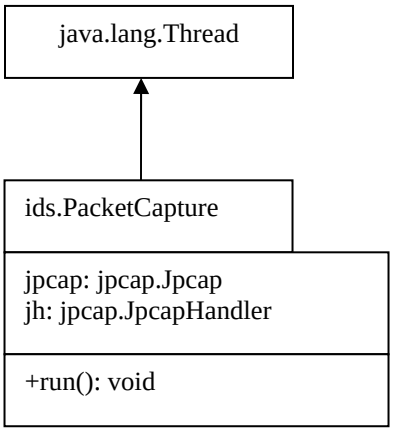
The above diagram is the structural model of architecture for the present system.
In this system, the Sniffer sub-system captures the packets which flow into and out of the system. This sub-system then formats these packets in a format that is convenient for further processing. Jpcap is a part of this Sniffer sub-system .The packet capturing function is accomplished via Jpcap. It provides a Java API to the popular C packet capture library called pcap. While Jpcap is not a complete conversion of the popular C pcap library yet, it does provide the basic functionality we need. There are various Intrusion units each for a specific attack. So, there are individual intrusion units which detect Port Scanning, Smurf Attack, syn-flood Attack, Efficient Mapping and Cerebral Mapping. All these intrusion units are independent of each other and interact only with the Control Unit. They run simultaneously continuously scanning for occurrence of specific attacks and report the attacks to the Control Unit when detected. The Store sub-system stores the various Rules defined and given to it by the Control Unit. It consists of various other sub-systems for data processing. The data in the form of XML files is stored after encryption using Simplified DES algorithm. The Owner-GUI sub-system displays to the user the defined Rules, the attack logs and the running status of the Intrusion units. It also provides facilities for starting and stopping intrusion units, clearing attack logs, adding new Rule to the store and deleting existing Rule from the store.

The console sub-system performs functions similar to the Owner-GUI sub-unit, but displays at the command line. The Control Unit sub-system manages the sub-systems for detection of attacks by taking the packets from the sniffer, sending relevant packets to the Intrusion Units, gives Rules to the store and retrieves them and displays necessary messages to the user through the user interface. The Protection Unit sub-system takes the Rules from the Control unit and provides security by applying the rules on local Operating System as IPSec policies.

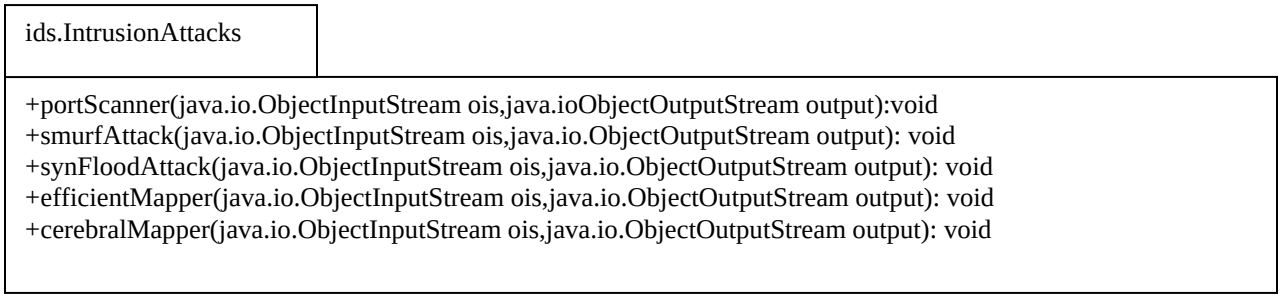
Detailed Class Descriptions:

There system contains a total of 31 classes where 7 of them are inner classes.
The following are the important classes identified,
PacketCapture,
IntrusionAttacks,
IntrusionUnit,
Attack,
Rule,
ProtectionUnit,
SimpleDES,
XmlData,
DataProcessor,
ControlUnit,
Console,
Owner, and
IDSMain.

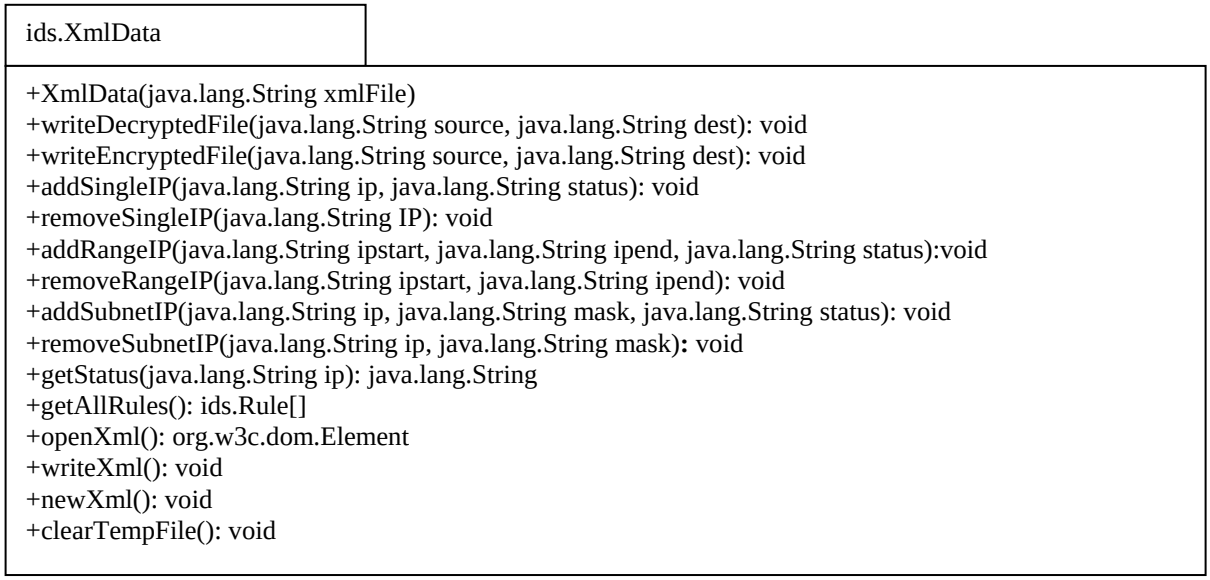
UML Diagram showing PacketCapture



UML Diagram showing IntrusionAttacks



XmlData showing various tags



Design with Reuse as goal

Some of the components present in the system are sequence independent from other components. These are to be designed in such a way that they can be reused by other systems. The system is designed to meet the specifications of "design with reuse". The various components that are developed in the system that can be used as Reusable components are as follows:
PacketCapture,
IntrusionUnit,
SimpleDES,
XmlData, and
Console.

The system is implemented in JAVA, which provides the very important feature called "Platform independence" making the code portable on any Operating system.

IV. Conclusion And Future Work

The IDS is designed to provide the basic detection techniques so as to secure the systems present in the networks that are directly or indirectly connected to the Internet. Performing such a duty always goes in hand on hand diving success as well as failure in fulfilling the objective. At least it does it job. But finally at the end of the day it is up to the Network Administrator to make sure that his network is out of danger. This software does not completely shield network from Intruders, but IDS helps the Network Administrator to track down bad guys on the Internet whose very purpose is to bring your network to a breach point and make it vulnerable to attacks. The following is just a first and of what should be the source of action while using the software and after an attack has been detected by IDS. Unlike other conventional Intrusion Detection Systems the present system also provides facilities for Intrusion Protection. This facilitates for blocking or allowing particular IP, range of IPs or a subnet IPs by applying relevant rule on the Operating system. The IDS system is designed in such a way that it can be reused very easily. A platform is set very clearly in order that some known attacks can be identified. Due to the high end flexibility and extensibility given using the design of the system it will be easy to add more number of attacks to the system in future.

The IDS is written completely in Java. Thus the present system is platform independent, yet it has been tested only on WindowsXP. It can be employed and tested on various other machines which run on different Operating systems and which satisfy the requirements and pre-requisites for the IDS system. The present IDS system employs a log that is valid only for the current session and doesn't store the information about the past sessions. This feature can be extended by enhancing the log capability to store the information about the past sessions. The system may be enhanced by incorporating techniques corresponding to the future works listed below:

The present system just displays the log information but doesn't employ any techniques to analyze the information present in the log records and extract knowledge. The system can be extended by incorporating Data Mining techniques to analyze the information in the log records which may help in efficient decision making. The present system only detects the attacks only

the known attacks. This can be extended by incorporating Intelligence into it in order to gain knowledge by itself by analyzing the growing traffic and learning new Intrusion patterns.
The present system runs on an individual host machine and is not a distributed application. This can be extended to make it a distributed application where different modules of the same system running on different machines may interact with each other thus providing distributed detection and protection for all those machines on which the system is running.

V. References

[1] William Stallings, “Cryptography and Network Security”, Principles and Practices, Third Edition.
[2] D. E. Denning, "An intrusion-detection model". IEEE Transactions on Software Engineering, Vol. SE-13(No. 2):222-232, Feb. 1987.
[3] Stephen Northcutt, Judy Novak, “Network Intrusion Detection”, Third Edition, Pearson Education 2003.

Biography

Allam Appa Rao, Ph.D is a faculty in Department of Computer Science and Systems Engineering at Andhra University College of Engineering, Visakhapatnam, India. He is also President of the Institution. His email address: allamapparao@gmail.com.

P.Srinivas, B. Chakravarthy, K.Marx, and P. Kiran are students in Department of Computer Science and Systems Engineering at Andhra University College of Engineering, Visakhapatnam, India.