International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

Analysis and Design Modeling for Next Generation Network Intrusion Protection Systems

Nareshkumar Harale, B. B. Meshram

Abstract: ***The continued exponential growth of successful cyber intrusions against today's businesses has made it abundantly clear that traditional perimeter security measures are no longer effective. We evolved the network trust architecture from trust-untrust to Zero-Trust, With Zero Trust, essential security capabilities are deployed in a way that provides policy enforcement and protection for all users, devices, applications, data resources, and the communications traffic between them, regardless of location. Information exchange over the Internet, in spite of inclusion of advanced security controls, is always under innovative, inventive and prone to cyberattacks. TCP/IP protocol stack, the adapted standard for communication over network, suffers from inherent design vulnerabilities such as communication and session management protocols, routing protocols and security protocols are the major cause of major attacks. With the explosion of cyber security threats, such as viruses, worms, rootkits, malwares, Denial of Service attacks, accomplishing efficient and effective intrusion detection and prevention is become crucial and challenging too. In this paper, we propose a design and analysis model for next generation network intrusion detection and protection system as part of layered security strategy. The proposed system design provides intrusion detection for wide range of attacks with layered architecture and framework. The proposed network intrusion classification framework deals with cyberattacks on standard TCP/IP protocol, routing protocols and security protocols. It thereby forms the basis for detection of attack classes and applies signature based matching for known cyberattacks and data mining based machine learning approaches for unknown cyberattacks. Our proposed implemented software can effectively detect attacks even when malicious connections are hidden within normal events. The unsupervised learning algorithm applied to network audit data trails results in unknown intrusion detection. Association rule mining algorithms generate new rules from collected audit trail data resulting in increased intrusion prevention though integrated firewall systems. Intrusion response mechanisms can be initiated in real-time thereby minimizing the impact of network intrusions. Finally, we have shown that our approach can be validated and how the analysis results can be used for detecting and protection from the new network anomalies.***

Keywords: Intrusion System, Network Intrusion Detection, Intrusion Prevention, Firewall system; Data Mining, Association rule, Network Security

1. Introduction

Nations without controlled borders cannot ensure the security and safety of their citizens, nor can they prevent piracy and theft. Similarly, computer networks without controlled access cannot ensure the security or privacy of stored data, nor can they keep network resources from being exploited by hackers. When internal network is connected to the internet, there is no inherent central point of security control; in fact there is no security at all. With the persistent development and extensive application of network technology, the security of network system is increasingly outstanding, which has been a big hotspot concerned by governments, companies and individual users. In order to safeguard the secure running of network, people have taken diverse protecting measures, and among them, firewall and intrusion detection are adopted more frequently.

2. Literature Survey

Network security – It is one of the major considerations in computer networking design and implementation. Various types of tools are being used for providing security to networks. Firewall and Intrusion Detection System are majors among them. In this section the extensive literature survey is done on firewall, types of firewall, comparison between firewalls and IDS Lastly the programming languages and tools to be used in the proposed systems are studied.

Data Mining - Data mining has attracted more and more attention in recent years, probably because of the popularity of the "big data" and data management concept. Data mining is the process of discovering interesting patterns and knowledge from large amounts of data [1]. As a highly application-driven discipline, data mining has been successfully applied to many domains, such as business intelligence, Web search, scientific discovery, digital libraries, etc.

Firewall System – It is a software program or device that monitors, and sometimes controls, all transmissions between an organization's internal network and the Internet. However large the network, a firewall is typically deployed on the network's edge to prevent inappropriate access to data behind the firewall. The firewall ensures that all communication in both directions conforms to an organization's security policy. Firewall technologies are configurable. You can limit communication by direction, IP address, protocol, ports, or numerous other combinations. If you have access to the firewall, you can configure it to enable the ports, protocols, and addresses. In some cases, however, your organization's security policy may prevent optimal streaming. For example, firewalls configured to only allow TCP traffic may cause the user to see frequent buffering of clips. User experience of the presentation is compromised; greater latency and startup times affect the time needed to view the clip, and delivery of the clip requires more total bandwidth.

Types of Firewalls

Firewall systems fall into following categories:

- **Packets Filter Firewall**: Packet filtering firewalls work at the network level of the OSI model, or the IP layer of TCP/IP. They are usually part of a router firewall. In a

packet filtering firewall, each packet is compared to a set of criteria before it is forwarded. Depending on the packet and the criteria, the firewall can drop the packet, forward it, or send a message to the originator. Most routers support packet filtering. Even if other firewalls are used, implementing packet filtering at the router level affords an initial degree of security at a low network layer. Important point to remember is that the rule set should be built from more specific to more general rules. And the ordering of rules should be such that the most frequently used rules are at the top of the list.

- **Application Gateway Firewall**: An application-level firewall consists of a proxy server communicating with servers outside the network to control traffic between two networks. When you use an application-level firewall, local network does not directly connect to the Internet. Instead, the proxy server transfers an isolated copy of each approved packet from one network to another, whether the packet contains incoming or outgoing data. The result is that the firewall effectively masks the original address of the initiating connection and protects internal network from malicious intruders who may attempt to obtain network information. In other words, proxy servers are used to hide your IP address, making you anonymous on the Internet. The downfall is that hackers can also use this "service" to hide their IP addresses when attacking a specific server. Because proxy servers recognize network protocols, you can configure your proxy server to control which IP services you want on your network. There are many types of proxy servers available. Each protocol that you screen for requires a new proxy server entry (unlike a screening router). This firewall system works as per the steps given as follows:

  Lack of transparency: When all the above said steps are done, the communication carries on through the gateway. So the communication is no more private. The gateway knows everything going on between the communicating parties.

  i. **SYN Flooding:** This involves sending a stream of forged messages to target computer.
  ii. **Ping Flooding**: This involves sending a stream of number of pings.
  iii. **Malicious applets:** Some applets contain undesired contents but as they are downloaded to computer directly, gateways can't trace them.

- **Guard Firewall:** This is the most complex firewall among all firewalls. When used it can see full text of communication. It can audit activity on network connection. Screening in this firewall is based on interpretation of the message content it follows. Due to its complexity it limits assurance of security.
- **Stateful Inspection Firewall:** A newer method that doesn't examine the contents of each packet but instead compares certain key parts of the packet to a database of trusted information. Information traveling from inside the firewall to the outside is monitored for specific defining characteristics, and then incoming information is compared to these characteristics. If the comparison yields a reasonable match, the information is allowed through. Otherwise it is discarded. The firewall works as per the following steps:

  i. The Packet is inspected to determine whether it is part of an existing, established communication flow by comparing the characteristics of the packet with a connection table of existing, valid connections to see whether there is a match.
  ii. Depending on the protocol, the packet may be inspected further.
  iii. If the packet does not have a corresponding entry in the connection table, the firewall will inspect the packet against its configured rule set.
  iv. If packet is permitted then firewall will forward it to final destination.
  v. The firewall will typically use timers & inspection of a TCP packet with the FIN bit.

- **Personal Firewall**: A personal firewall is an application which controls network traffic to and from a computer, permitting or denying communications based on a security policy. A personal firewall differs from a conventional firewall in terms of scale. Personal firewalls are typically designed for use by end-users. As a result, a personal firewall will usually protect only the computer on which it is installed. Many personal firewalls are able to control network traffic by prompting the user each time a connection is attempted and adapting security policy accordingly. Personal firewalls may also provide some level of intrusion detection, allowing the software to terminate or block connectivity where it suspects an intrusion is being attempted.

Packet Filtering Firewall Systems

At its most basic level, a packet-filtering firewall consists of a list of acceptance and denial rules. These rules explicitly define which packets will and will not be allowed through the network interface. The firewall rules use the packet header fields to decide whether to forward a packet to its destination, to silently throw away the packet, or to block the packet and return an error condition to the sending machine. These rules can be based on a wide array of factors, including the source or destination IP addresses, the source and (more commonly) destination ports, portions of individual packets such as the TCP header flags, the types of protocol, the MAC address, and more. MAC address filtering is not common on Internet-connected firewalls. Using MAC filtering, the firewall blocks or allows only certain MAC addresses. However, in all likelihood you only see one MAC address, the one from the router just upstream from your firewall. This means that every host on the Internet will appear to have the same MAC address as far as your firewall can see. A common error among new firewall administrators is to attempt to use MAC filtering on an Internet firewall. Using a hybrid of the TCP/IP reference model, a packet-filtering firewall functions at the Network and Transport layers, as shown in figure 1. The overall idea is that you

need to very carefully control what passes between the Internet and the machine that you have connected directly to the Internet. On the external interface to the Internet, you individually filter what's coming in from the outside and what's going out from the machine as exactly and explicitly as possible.

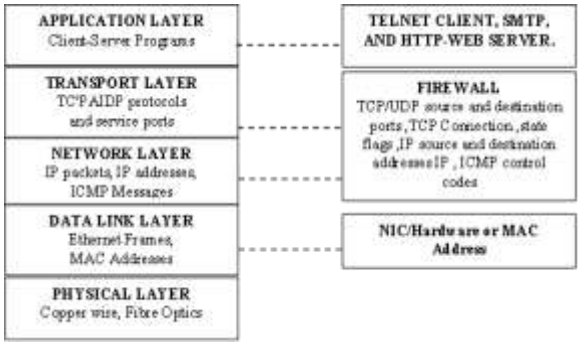

Figure 1: Firewall/IDS placements in TCP/IP reference

Choosing a Default Packet Firewall Policy

As stated earlier in this chapter, a firewall is a device to implement an access control policy. A large part of this policy is the decision on a default firewall policy. There are two basic approaches to a default firewall policy such as Deny everything by default, and explicitly allow selected packets through and Accept everything by default, and explicitly deny selected packets from passing through. Without question, the deny everything policy is the recommended approach. This approach makes it easier to set up a secure firewall, but each service and related protocol transaction that you want must be enabled explicitly as shown in following figure. This means that you must understand the communication protocol for each service you enable. The deny everything approach requires more work up front to enable Internet access. Some commercial firewall products support only the deny everything policy. The accept-everything policy makes it much easier to get up and running right away, but it forces you to anticipate every conceivable access type that you might want to disable . Figure 2 and 3 shows both policies:

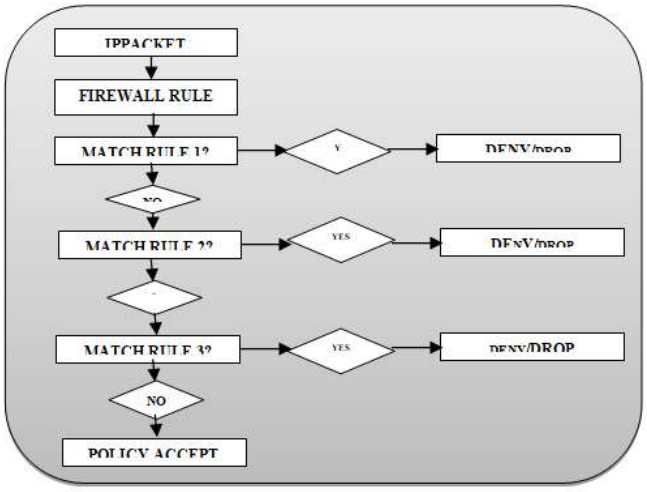

Figure 2: Deny Everything by Default

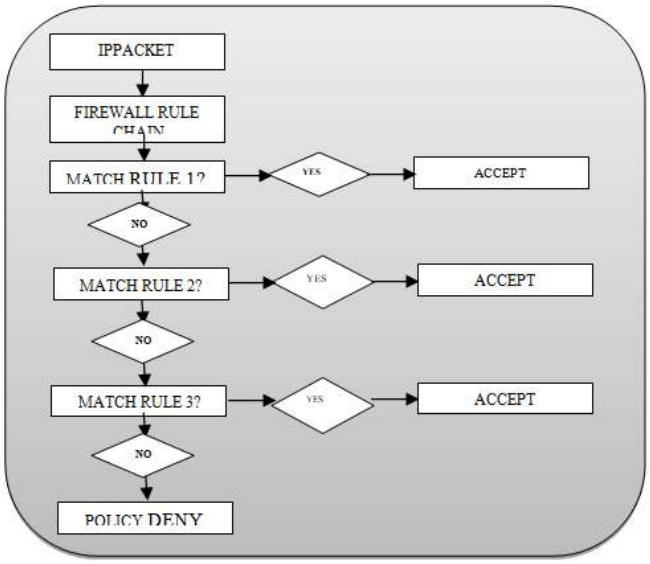

Figure 3: Accept Everything by Default

IP-TABLES - The iptables tool inserts and deletes rules from the netfilter chains Linux kernel. Iptables have two parts, the user-space tools and the kernel-space modules. The kernel-space modules are distributed with the main kernel, and you compile them as you would any other module, be it sounds drivers, a file system or USB support. There is the main iptables module, as well as modules specifically for NAT, logging, connection tracking and so on. These modules perform the appropriate function on the packets which they get sent by netfilter, depending on the rules which they have in their rule-list, or chain. The user-space iptables code comes in the form of a binary called iptables, which is distributed separately from the main kernel tree, and is used to add, remove or edit rules for the modules. [10].

Packet Traversal through filters - The kernel starts with three lists of rules; these lists are called firewall chains or just chains. The three chains are called INPUT, OUTPUT and FORWARD. Figure 4 shows this.

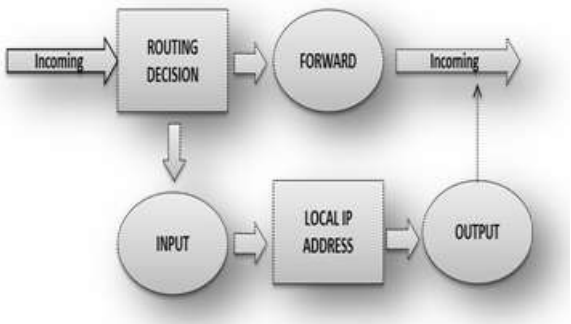

Figure 4: Packet Traversal Process

The three circles represent the three chains mentioned above. When a packet reaches a circle in the diagram, that chain is examined to decide the fate of the packet. If the chain says to DROP the packet, it is killed there, but if the chain says to ACCEPT the packet, it continues traversing the diagram. A chain is a checklist of **rules**. Each rule says 'if the packet header looks like this, then here's what to do with the packet'. If the rule doesn't match the packet, then the next rule in the chain is consulted. Finally, if there are

no more rules to consult, then the kernel looks at the chain **policy** to decide what to do. In a security-conscious system, this policy usually tells the kernel to DROP the packet.

1 When a packet comes in (say, through the Ethernet card) the kernel first looks at the destination of the packet: this is called 'routing'.
2 If it's destined for this box, the packet passes downwards in the diagram, to the INPUT chain. If it passes this, any processes waiting for that packet will receive it.
3 Otherwise, if the kernel does not have forwarding enabled, or it doesn't know how to forward the packet, the packet is dropped. If forwarding is enabled, and the packet is sent for another network interface (if you have another one), then the packet goes rightwards on our diagram to the FORWARD chain. If it is accepted, it will be sent out.
4 Finally, a program running on the box can send network packets. These packets pass through the OUTPUT chain immediately: if it says ACCEPT, then the packet continues out to whatever interfaces it is destined for. [10]

Steps to implement firewall using IPTABLES:

1. Enter the system with identity root, and establish a script named as rc.firewall in the directory /etc/rc.d/. This script only can be written and executed by identity root. Modify files in order that initializing firewall doesn't need to restart the computer at any moment, and this script is automatically executed after computer's start.
2. Initialize the rules of firewall. The main actions are deleting the existing rules and defining the default strategies.
3. Set the rules of filtering data packet. The main actions are enabling the loopback interface, forbidding the data packets from special and reserved IP address of interior network, filtering ICMP message, protecting the services of the non-privilege ports, etc.
4. Activate basic services including DNS service, FTP service, WEB service, UDP service, etc.
5. Forbid to access malevolent websites, Activate the accessing of LAN to Internet and IP masquerade.

Intrusion Detection System

Attacks on network infrastructure presently are main threats against network and information security. With the rapidly growing network activities on the network Intrusion Detection System (IDS) as a component of defense in depth is very necessary because traditional firewall techniques can't provide complete protection against intrusion. The goal of intrusion detection is to positively identify all true attacks and negatively identify all non-attacks. [6]

IDS classification - IDS are primarily of three types:

1. **Network Based IDS (NIDS)**: **A network intrusion detection system** (**NIDS**) is an intrusion detection system that tries to detect malicious activity such as denial of service attacks; port scans or even attempts to crack into computers by monitoring network traffic. If, for example, a large number of TCP connection requests to a very large number of different ports are observed, one could assume that there is someone committing a "port scan" at some of the computer(s) in the network. It also (mostly) tries to detect incoming shell codes in the same manner that an ordinary intrusion detection systems does. A NIDS is not limited to inspecting incoming network traffic only. Often valuable information about an ongoing intrusion can be learned from outgoing or local traffic as well. Some attacks might even be staged from the inside of the monitored network or network segment, and are therefore not regarded as incoming traffic at all. Often, network intrusion detection systems work with other systems as well. They can for example update some firewalls' blacklist with the IP addresses of computers used by (suspected) crackers. [11]
2. **Host Based IDS (HIDS)**: is an intrusion detection system that monitors and analyzes the internals of a computing system rather than on its external interfaces (as a network-based intrusion detection system (NIDS) would do). Host-Based IDS's monitor all or parts of the dynamic behavior and of the state of a computer system. Much as a NIDS will dynamically inspect network packets, a HIDS might detect which program accesses what resources and assure that (say) a word-processor hasn't suddenly and inexplicably started modifying the system password-database. Similarly a HIDS might look at the state of a system, its stored information, whether in RAM, in the file-system, log files or elsewhere; and check that the contents of these appear as expected. One can think of a HIDS as an agent that monitors whether anything/anyone – internal or external – has circumvented the security policy that the operating system tries to enforce. [11]
3. **FUSION IDS:** A hybrid IDS combines a HIDS, which monitors events occurring on the host system, with a NIDS, which monitors network traffic. [11]. Figure 5 is the block diagram of standard IDS system:

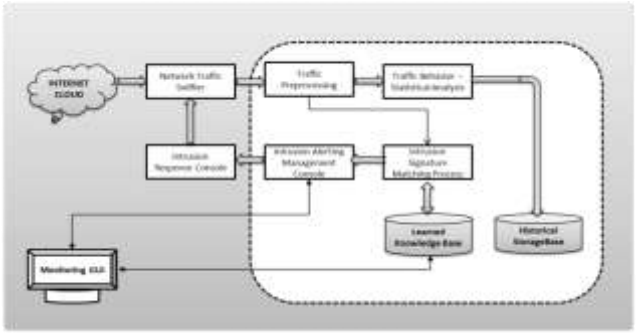

Figure 5: Standard Network IDPS Architecture

Data Mining and Network IDS

A typical approach for anomaly based Network IDS is the analysis of specialized audit trails to spot abnormal patterns of usage. But the problem of audit trails is that

data volumes are so large that analysis can become extremely expensive. Data mining based approaches ha great potentials to help alleviate the problem of automatically detecting anomalous patterns in large amount of audit data. We can apply data mining algorithm on the collected data (related to host pc or from network by sniffer) so as to extract knowledge by which intrusion can be detected. [6].

Strategies for Intrusion Detection

The different approaches that have been pursued to develop intrusion detection systems are described in many papers, including. Figure 6 shows four major approaches to intrusion detection and the different characteristics of these approaches. [12].

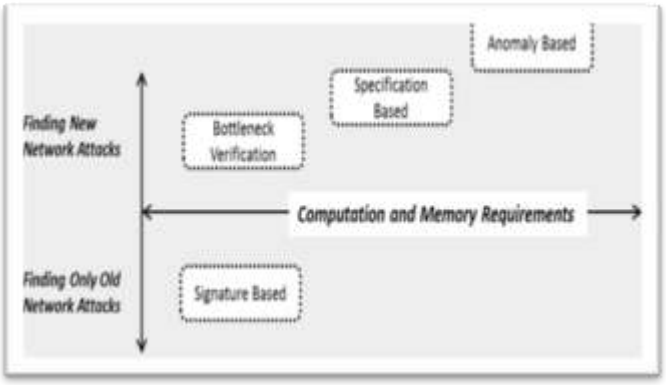

Figure 6: Approaches for network intrusion detections

The lower part of this figure shows approaches that detect only known attacks, while the upper part shows approaches that detect novel attacks. Simpler approaches are shown on the left and approaches that are both computationally more complex and have greater memory requirements are shown towards the right. The most common approach to intrusion detection, denoted as "signature verification" is shown on the bottom of above figure. Signature verification schemes look for an invariant sequence of events that match a known type of attack. For example, a signature verification system that is looking for a Ping of Death denial-of-service attack (an oversize ping packet that causes some machines to reboot) would have a simple rule that says "any ping packet of length greater than 64 kilobytes is an attack."

KDD-99 Cup Training Data Set

This is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99. The Fifth International Conference on Knowledge Discovery and Data Mining. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between ``bad'' connections, called intrusions or attacks, and ``good'' normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment. [17]

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset. Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks.

The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records. A connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a target IP address under some well-defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes. [15]

Network Attacks - Network attacks fall into four main categories:

- DOS: denial-of-service, e.g. syn flood;
- R2L: unauthorized access from a remote machine, e.g. guessing password;
- U2R: unauthorized access to local superuser (root) privileges, e.g., various ``buffer overflow'' attacks;
- Probing: surveillance and other probing, e.g., port scanning.

3. System Analysis and Design – NG2-ProWALL

In software design and development, system analysis is the process of studying and defining the problem to be resolved. It involves discovering the requirements that the system must perform, the underlying assumptions with which it must fit, and the criteria by which it will be judged a success or failure. In this research project work, we have used object oriented system analysis and design approach.

A. System Analysis - Analysis modeling mainly focuses on the user model and structural model views of the system. UML design modeling addresses behavioral model, implementation model, and environmental model views.
B. Use Case Diagram - Use case diagram is used to describe the system from user's perspective. It provides functional description of a system and its major processes. It provides graphic description of the users of the system and what kinds of interaction to expect within that system. In our project NG2-ProWALL as per our problem statement following are the specifications for use case diagram:

Actors: In the proposed IDPS system there are only two actors i.e. security analyst and security administrator who monitor, operates and manage the entire security management system.

Use Cases: Basically in our security analyst has a prime responsibility to do continuous intrusion monitoring which includes majorly the attack log monitoring and analytical reporting, however security administrator has two main activities i.e. to detect whether there are any intrusions or attacks on the system and control the firewall operations. In intrusion detection, there includes processes like invoking intrusion protection manager in case to block the connection and also responsible to store the attack in log database for future reference. In intrusion protection management there will be three main processes mainly adding rule, removing rule and blocking connection through firewall manager. Hence use cases are as below:

1. Intrusion Detection Module which includes:

Detection Manager: This use case shows the functionality that whenever network attack is detected the detector will first alert the security administrator and then invokes firewall manager

Attack Log Store Manager: After detecting network attacks, they are stored in log book for future use (in case of defining new firewall security protection policy)

2. Intrusion Protection Manager which includes

Add firewall rule: In this editor Firewall Manager first takes rule to add as input from administrator, secondly it detects potential anomalies that will occur after insertion of that rule and its probable position in firewall rules. After confirmation from administrator editor adds the rule

Remove firewall rule: It simply takes rule order as input from administrator and removes rule with that order.

Block attack connection or session: This is the specialized form of rule addition, and we can say it's the addition of rule in case of intrusion detection.

- **Intrusion Protection Module with Firewall Manager**: This package mainly concerns with management of firewall. It has following functionality: Adding rule to firewall, Removing rule from firewall, Blocking malicious attack connections or sessions, Identifying Firewall rule statistics

For adding rule, removing rule and blocking connection or session the main class is **editor** of whom these are functionality implemented. Also while adding new rule there are some previous rules that might conflict with new rule hence for that **anomaly detector** is another class. Input to anomaly detector is a systematic process of getting attack log from firewall system, applying association rule mining on it and generalizing of **fit records,** Hence there will be four more classes **log**, **arbdmlf** (association rule based data mining using log frequencies), **generalizer** and **order generator** for rule enforcement. For statistics generator, we get protection rules extracted by armlf and use them with initial firewall rules for identifying decaying and dominant rules; hence there will be another class **statistics generator**.

1. **Intrusion Detection Module**: This package mainly concerns with getting network traffic log as input and detecting intrusive connection or sessions among them. It has three major functionality such as Detection of network intrusion or attacks, Alerting to a security administrator and Storing intrusion or attack logs into the database.
2. **Intrusion Signature Generator Module**: This package is responsible for generating signatures for different network attacks which will be used by intrusion detector for detecting genuine attacks. For intrusion signature generation besides this there are two subsystems mainly **Firewall** and **Packet Sniffer**.
   - Firewall: Firewall mainly follows rules managed by firewall manager and allows or blocks connection according to them. Firewall also generates log that is used mainly for identifying rule statistics.
   - Packet Sniffer: Packet Sniffer sniffs the network traffic filtered by firewall and gives packet information in packet log for detector.

C. Class Diagrams

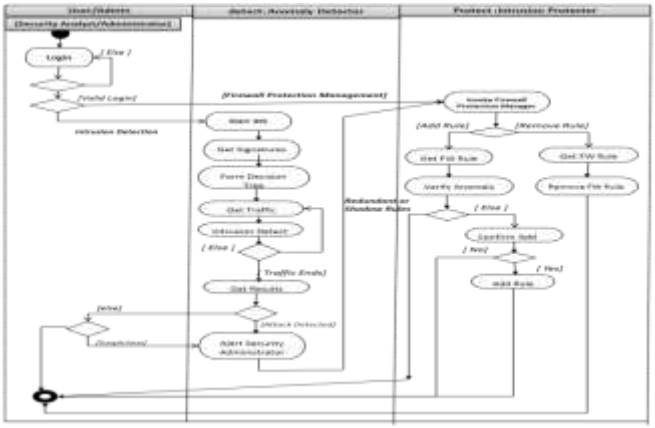

Figure 7: Activity diagram for NG2-ProWALL System

D. Functional Modeling

The functional model consists of multiple data flow diagrams which show the flow of values from external inputs, through operations and internal data stores, to external outputs.

Context Level DFD for NG2-ProWALL System

Figure 8 shows the context level or Level 0 DFD which shows holistic view of information system as single processing unit and gets input from security analyst and or security administrator and gives output to security analyst/administrator.

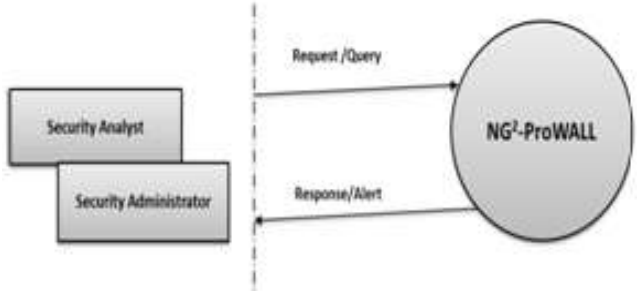

Figure 8: Level 0 – Context Level DFD

Level 1 DFD for NG2_ProWALL System

Figure 9 is the Level 1 DFD for NG2-ProWALL system which breaks the system into five main parts i.e. SystemLogin, IntrusionMonitor, IntrusionDetector, IntrusionProtector (FireMAN) and IntrusionSignature Generator. Security Analyst or Security administrator first have to login into the system. After successful login, they can request either for intrusion monitoring information or intrusion detection by providing network traffic log as input or perform firewall management operation for which interface is given in Network IDS.

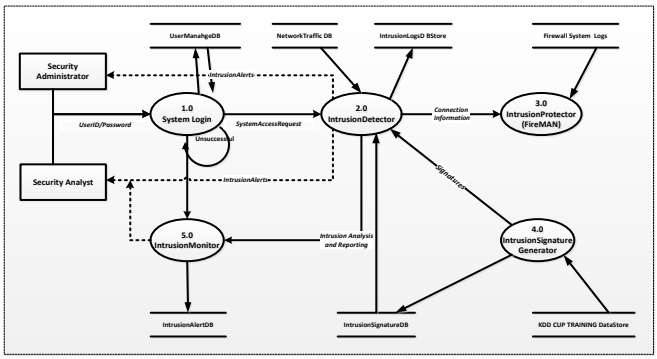

Figure 9: Level 1 -DFD NG2-ProWALL System

Level 2 DFD for Detector

Intrusion Detector process has two main components, Preprocessor – to process the IP packet's and Packet Analyzer for analyzing the contents to confirm an abnormalities. Preprocessor takes the network traffic stored by packet sniffer. It processes the traffic and converts it into connection level format usable by analyzer. For intrusion detection analyzer gets attack signatures from intrusion signature database, forms decision tree and apply decision tree to each connection record available from network connection database. On actual intrusion detection, Analyzer alerts to security analyst as well as security administrator related processes and invokes Intrusion Protector (FireMAN) to block an intrusive or malicious connection. ID Analyzer stores detected attack information in attack log for future reference. Figure 10 shows this:

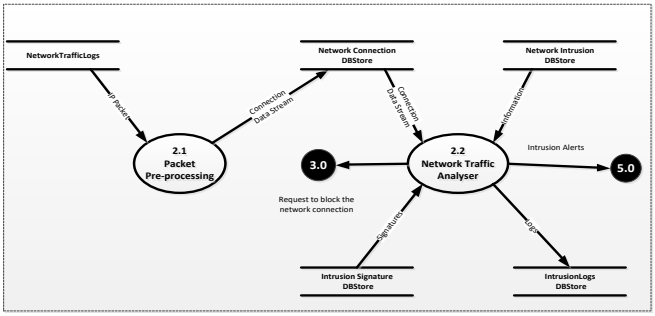

Figure 10: Level 2 - DFD for Intrusion Detector Process

Level 2 - DFD for Intrusion Protector (FireMAN)

In Intrusion Protector (FireMAN) process, ARMLF module takes firewall log as input and generates fit rules using support-confidence values. It also stores support-confidence for each rule (fit as well as unfit) to be used for decaying and dominant rule detection. Generalizer generalizes fit rules and stores them to generalize table. Statistic generator gets its input from support confidence table and also uses firewall rules to identify decaying and dominant rules among them. It stores statistics in statistics table. Order generator gets generalized and initial rules, combines them, orders them and stores them as anomaly input for anomaly detection. Anomaly detectors get its data from anomaly input and give anomalies to editor. Finally editor uses all the results and manages the firewall rule accordingly.

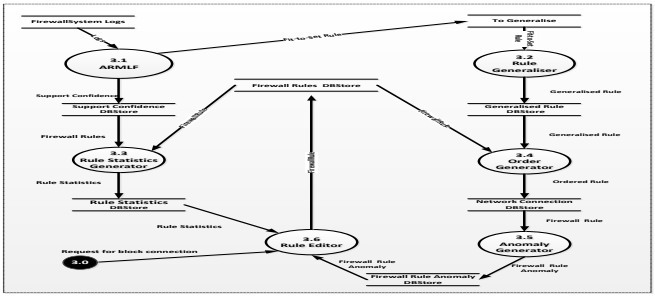

Figure 11: Level 2 - DFD for Intrusion Protector Process (FireMAN)

Level 2 DFD for Signature Generator

In signature generation KDDReader first reads the KDD99 cup training data and stores it to apriori input table. apriori reads data from apriori input calculate fitness of each record and stores them to rule base table. Selector selects best rules among rule base using their fitness and stores them in attack signature table. Finally tester takes generated signature as input and tests them with kdd99 cup testing data and stores test statistics in test results.

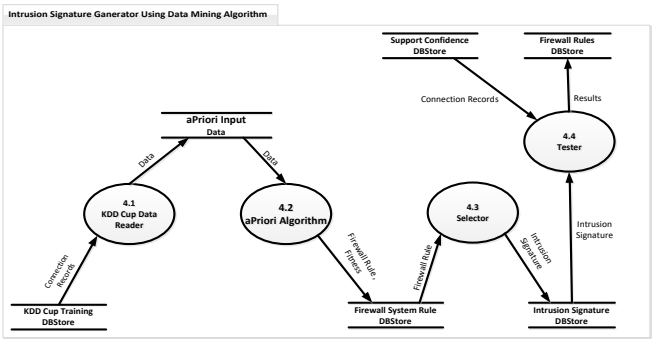

Figure 12: Level 3 - DFD for Signature Generator Process

4. Conclusion and Future Work

The extensive literature survey of classification of IDS and Firewall and Data Mining Algorithms for the implementation of Network IDPS is done. The analysis of the systems using use case diagram, class diagram and activity diagram along with data flow diagram is proposed to capture all the user requirements for the implementation of the Network IDPS. Then the proposed IDPS software is presented with the internal functionality of the Network IDPS. The future work will be carried out for the implementation of the Network IDPS by proposing data structures and algorithms required for the implementation of the software successfully. The proper system GUI will be designed and implemented to display the testing results for the network intrusion detection and protection.

References

[1] Zongpu Jia, Shufen Liu, Guowei Wang, "Research and Design of NIDS Based on Linux Firewall" 2006 1 st International Symposium on Pervasive Computing and Applications
[2] Korosh Golnabi, Richard K. Min, Latifur Khan, Ehab Al-Shaer, "Analysis of Firewall Policy Rules Using Data Mining Techniques", Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP.
[3] Agrawal, R., T. Imielinski, and A. Swami. "Mining Association Rules between Sets of Items in Large Databases" in Proceedings of the 1993 Webb, G.I., Association Rules, in Handbook of Data Mining, N. Ye, Editor, Lawrence Erlbaum: To appear.
[4] Robert Winding, Timothy Wright, and Michael Chapple, "System Anomaly Detection: Mining Firewall Logs", 2006, IEEE
[5] Wenke Lee, "A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems"
[6] Yuebin Bai, Hidetsune Kobayashi, "Intrusion Detection System: Technology And Development", Proceedings of the17 th International Conference on Advanced Information Networking and Applications (AINA'03), IEEE
[7] Eugene Spafford, Diego Zamboni, "Data Collection Mechanisms For Intrusion Detection"
[8] Conference (IM'2003), March 2003. E. Al-Shaer and H. Hamed. "Firewall Policy Advisor for Anomaly Detection and Rule Editing." IEEE/IFIP Integrated Management
[9] Ehab Al-Shaer and Hazem Hamed, "Discovery of Policy Anomalies in Distributed Firewalls" in Proc. of IEEE INFOCOMM'04, vol. 23, no. 1, March 2004 pp. 2605-2616.
[10] D. Chapman and E. Zwicky., Building Internet Firewalls, Second Edition, Orielly & Associates Inc., 2000.
[11] Intrusion Detection & Prevention by Carl Endorf, Eugene Schultz and Jim Mellander McGraw -Hill © 2004
[12] A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems by Kristopher Kendall
[13] C. Elkan, "Results of the KDD'99 classifier learning contest," SIGKDD Explorations. ACM SIGKDD", vol. 1, no. 2, pp. 63–64, 2000.
[14] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection", IEEE Network, 8(3): 26-41, May/June 1994.
[15] MIT Lincoln Laboratory, DARPA datasets, MIT, USA, http://www.ll.mit.edu/IST/ideval/data/data_index.html (accessed in November 2004).
[16] W. Li, "A Genetic Algorithm Approach to Network Intrusion Detection", SANS Institute, USA, 2004.
[17] Dong Song, Malcolm I. Heywood and A. Nur Zincir-Heywood, "Training Genetic Programming on Half a Million Patterns: An Example from Anomaly Detection", IEEE Transactions on Evolutionary Computation, VOL. 9, NO. 3, JUNE 2005