# NIDS: A network based approach to intrusion detection and prevention

Martuza Ahmed, Rima Pal
Dept. of Computer Science & Engineering
Shahjalal University of Science & Technology
Sylhet, Bangladesh
martuzaahmed@gmail.com, rima_cse34@yahoo.com

Md. Mojammel Hossain, Md. Abu Naser Bikas,
Md. Khalad Hasan
Dept. of Computer Science & Engineering
Shahjalal University of Science & Technology
Sylhet, Bangladesh
{bsst33.rezbe, bikasbd }@yahoo.com,
khalad-cse@sust.edu

*Abstract*—**Computer networks have added new dimensions to the global communication. But intrusions and misuses have always threatened the secured data communication over networks. Consequently, network security has come into issue. Now-a-days intrusion detection systems play an important role in security infrastructures. Intrusions typically start with intruders infiltrating a network through a vulnerable host and after that approaching for further malicious attacks. The techniques used for intrusion detection have their particular limitations. Any of the Intrusion Detection Systems proposed so far is not completely flawless. The host based systems as well as the network based systems have their own limitations. So, the quest for betterment continues. In this progression, here we present NIDS: a complete intrusion detection system which reduces some significant problems contained by the previous systems such as trust issues and message spreading problems. In the proposed IDS we don't need to install the system for every host. That reduces the system overhead to a reasonable extent. This system generates steady responses against intrusions and stops the intruder to proceed for further attacks.**

***Keywords- Intrusion Detection System, Intrusion Prevention, Trust Issue, Packet Behavior, Discard Packet.***

## I. INTRODUCTION

*Intrusion* is the act or attempt of using a computer system or computer resources without the requisite privileges, causing willful or incidental damage. *Intrusion Detection* involves identifying individuals or machines that perform or attempt intrusion. *Intrusion Detection Systems* (IDS) are computer programs that tries to perform intrusion detection by comparing observable behavior against suspicious patterns, preferably in real-time. Intrusion is primarily a network based activity. With increasing global network connectivity, the topic of intrusion has gained prominence, spurring active research on efficient IDS. Intrusion detection systems can be classified on the basis of a multitude of factors [1]. Some factors significant to our paper are listed below.

RESPONSE TO INTRUSION: This may be passive or active. A passive system is content with just detecting intrusion, leaving its handling to a second. On the other hand, an active system takes action on own initiative, for example freezing network connections to a suspected host. Obviously, active systems can react more quickly and to more events, but they over-react to deliberately triggered false alarms.

SOURCE OF AUDIT DATA: The data to be examined could be network data or host data.

DATA COLLECTION AND PROCESSING: Collection of data may be centralized or distributed. Again, this data may be processed centrally or at distributed locations. In recent times, there has been a lot of work done in distributed schemes for intrusion detection. When the research community has been active in this area [2-8], most existing intrusion detection systems are passive because they only collect information in a distributed way. The controlling intelligence is centralized in the person of the system administrator(s) who manage the administrative domain. Bringing the exact information to this central entity is a difficult task that needs a fine balance between overloading the administrator and not providing enough related information. Therefore, an autonomous system is needed to replace this central control.

Another thing that requires attention here is, we are proposing a system where we are concerned about less overhead in less installation overhead. As we all know, network data is just a collection of several packets. What we are doing here is testing every packet of data before it reaches the destination host computer. And no host receives any packet from anywhere else than the server of that network. Even if some packet is sent to it directly then it forwards that packet to the server and the server after verification sends the packet back.

The motivations and current design of the NIDS are described in Section II. Section III discusses the knowledge updating for NIDS architecture. In Section IV we describe how NIDS deals with multiple packets at the same time. Sec. V deals with solution of the trust issue and message spreading difficulties in NIDS. Sec. Principle features of NIDS are mentioned in section VI. And section VII puts a conclusion to all the proceedings.

978-0-7695-3653-8/09 $25.00 © 2009 IEEE
DOI 10.1109/IACSIT-SC.2009.96

## II. NIDS

NIDS stands for Network-based Intrusion Detection System.

### A. *Brief overview of NIDS*

NIDS is an intrusion detection tool that applies a network based approach towards an intrusion. In this system we assign a server node for a part of the network. Now if we need to send some data to any node of that specific part of the network, then we send it to the server and then the server redirects the data to its original destination. The server has the system installed in it to detect an intrusion. So, the server detects it and discards the data if it is an intrusion. Otherwise it sends the verified data to the destination node.

### B. *Normal Mode Operation*

- Firstly let's assume that 'S' is the server for the network X. And A, B, C, D are different hosts on that network.
- Now if a host I has to send some data to any of the hosts within network X, then it sends the packet to server S and then S redirects it to the original destination after verification.

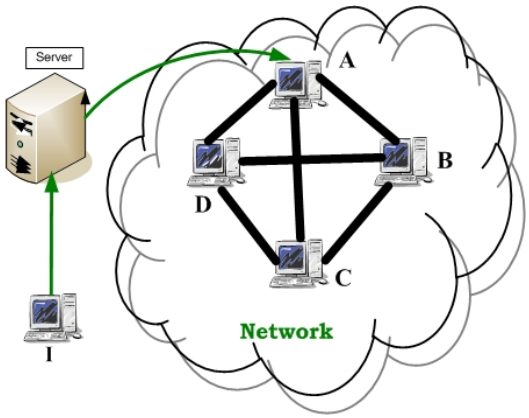


**Fig.1: Normal mode transmission in NIDS**

- Any of the hosts within network X does not receive any data except from the server S. If a node finds that the data has come from any other node than the server, then it forwards it to the server without receiving it.
- The server after verification sends back the data to the node which forwarded it.

### C. *How the intrusion is tackled:*

- Now, let's assume an intruder 'I' sends some intrusive data to node 'A'.
- Primarily 'I' has to send the data through the server 'S'. In that case the server detects the intrusion and discards the data. Now for more convenience let's assume that the intruder

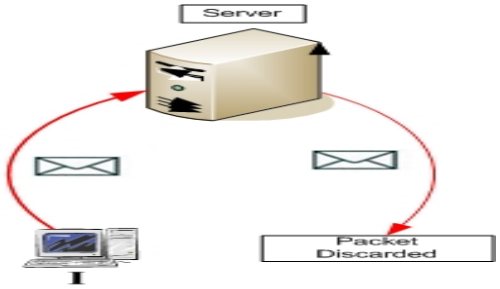


**Fig.2: Intrusion attempt taken through the sever**

somehow gets access to node 'A' through some other way. Then what happens?

- As per the definition of the system, node 'A' forwards the data to server 'S'.
- The server tests the data and detects it as an intrusion. And immediately discards the data. Now the question comes, how the server detects the intrusion?

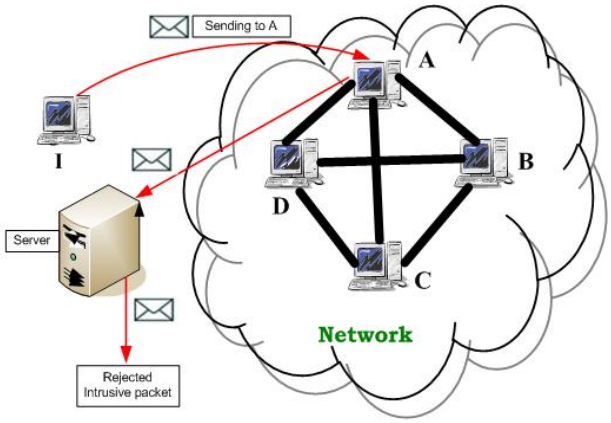


**Fig.3: Intrusion attempt taken directly to A and the intrusive packet is forwarded to the server**

- The server uses a comparing tool and a known attack pattern list of all possible attacks. Whenever the server finds some data set, it starts the comparison.

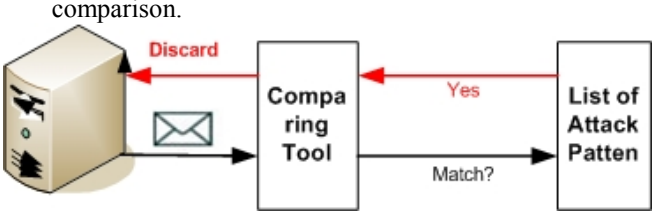


**Fig.4: Intrusion detection process in the server**

- And in the case of an intrusion like this one, quite a good similarity is found with any of the attack patterns in the list.
- Then that portion of data is discarded straight away.

## III. KNOWLEDGE UPDATING IN THE PROPOSED IDS

In the proposed system the server contains a list of known attack patterns. When a packet arrives at the server it first checks the packet behavior with some predefined intrusive behavior list. If the packet behavior matches with any of these then the packet is rejected. We assume that the server will be regularly updated with the recent intrusive behaviors. But just in case, if the packet contains such an intrusive behavior which is not in the predefined list of attack patterns stored in the server then the server have to pass the packet to the appropriate destination with that newly born attack. For such a situation, we will send an identification code with every packet. Server will store the last delivered 30-50 packet behaviors (can be defined by the administrator) along with their identification number. Now, when a host receives a packet and its performance is degraded, comparing to its previous performance then the host sends the identification number back to the server. Performance can be measured by comparing its processing power or some unusual behavior that was not present before the arrival of the particular packet.

The whole process is being executed only to make the updating process dynamic. In this process the server remains aware of all the latest attack patterns. And it does not have to depend on any human agent for this updating procedure.

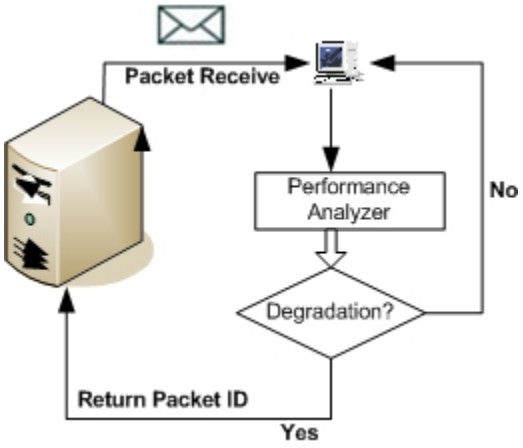


**Fig.5: Knowledge updating in the server**

## IV. WHEN MULTIPLE PACKETS ARRIVE IN THE SERVER AT THE SAME TIME

When a server receives a data set then as stated earlier, it assigns an identification number to each data set and starts matching it with the known attack patterns. But there is a possibility of multiple packet arrival at the same time. In that case the server maintains a queue and after assigning identification numbers to the packets, keeps them in that

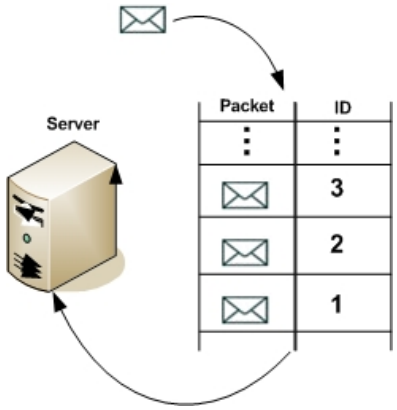


**Fig.6: When multiple packets reach the server at the same time**

queue. The packets are taken from the queue, then processed in the server and at last sent to the required destination.

## V. THE TRUST ISSUE AND MESSAGE SPREADING IN THE PROPOSED IDS

Now the question of trust comes. In this system, we don't have a trusted network. That means we don't need to trust anybody anywhere. As the whole detection process takes place in the server and the server updates it from time to time, there is no necessity of message spreading either. So, this strongly demolishes the possibility of false alarms as there is no need of alarm generating.

## VI. PRINCIPLE FEATURES OF THE PROPOSED IDS

- A network based system which does not require the system to be installed in every node.
- Free from trust problem.
- Free from message spreading difficulties.
- As all the packets of the network are received from the server, the rest of the network does not have to bother with the intrusion.
- As there remains no need of alarm generating, it is clearly understood that there is no probability of false alarm problems.
- The total system is self sufficient.

The following table gives us a brief view of the comparison between IDS and existing IDSs:

| Name of the IDS | Type of response | Data processing | Need of trust | Time Consumption | Multiple attack stability | Message spreading | Required Connection Termination |
|---|---|---|---|---|---|---|---|
| NADIR [2] | Passive | Centralized | Not applicable | Weekly basis analysis | Can't cope | Not applicable | Yes |
| DIDS [10] | Passive | Distributed | Not applicable | Time consuming | Can't cope | Not applicable | Yes |
| GrIDS [3] | Passive | Distributed | Not applicable | Time consuming | Can't cope | Not applicable | Yes |
| CSM [10] | Active | Distributed | Not applicable | Time consuming | Can't cope | Not applicable | Yes |
| EMERALD [6] | Active | Distributed | Needs | Time consuming | Can't cope | Required | Yes |
| AAFID [10] | Passive | Centralized | Not applicable | Time consuming | Can't cope | Not applicable | Yes |
| INDRA [9] | Active | Distributed | Needs | Less time consuming | Can't cope | Required | Yes |
| * NIDS | Passive | Centralized | Needs not | Less time consuming | Can cope | Not required | No |

*indicates the proposed IDS.

**Table 1: A Comparison between the existing IDS and NIDS**

## VII. CONCLUSION

The proposed system has the benefit of less overhead in installing the system on different hosts. The system needs to be installed only for the server of a definite part of the network. The other hosts of the network may remain as they are without bothering about the intrusion. But the server needs to be updated regularly.

## REFERENCES


[1] S. Axelsson. Research in intrusion-detection systems: A survey. Technical Report 98–17, Department of Computer Engineering, Chalmers University of Technology, December 1998.

[2] Judith Hochberg, Kathleen Jackson, Cathy Stallings, J. F. McClary, David DuBois, and Josephine Ford. Nadir: An automated system for detecting network intrusion and misuse. *Computers & Security*, 12(3):235–248, 1993.

[3] S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, J. Rowe, S. Staniford-Chen, R. Yip, and D. Zerkle. The design of grids: A graph-based intrusion detection system. Technical Report CSE-99-2, U.C. Davis Computer Science Department, January 1999.

[4] J. S. Balasubramaniyan, J. O. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni. An architecture for intrusion detection using autonomous agents. Technical Report 98/05, Purdue University, 1998

[5] G. White, E. Fisch, and U. Pooch. Cooperating security managers: A peer-based intrusion detection system. IEEE Network, 10(1):20–23, 1994.

[6] P. A. Porras and P. G. Neumann. EMERALD: event monitoring enabling responses to anomalous live disturbances. In *Proceedings of the 20th National Information Systems Security Conference*, pages 353–365, October 1997.

[7] G. Helmer, J. Wong, V. Honavar, and L. Miller. Intelligent agents for intrusion detection. In *IEEE Information Technology Conference*, pages 121–124, September 1998.

[8] M. Crosbie and G. Spafford. Defending a computer system using autonomous agents. Technical Report 95-022, Dept. of Computer Sciences, Purdue University, Mar 1996.

[9] Ramaprabhu Janakiraman, Marcel Waldvogel, Qi Zhang, "Indra: A peer-to-peer approach to network intrusion detection and prevention."

[10] Hakan Albag, Network & Agent Based Intrusion Detection Systems Tu Munich, Dep. of Computer Science – Exchange Student Istanbul Tech. Uni., Dep. Of Comp. Engineering.