

A Distributed Intrusion Detection System Against Flooding Denial of Services Attacks

XIE Chuiyi^{ab}, ZHANG Yizhi^c, BAI Yuan^d, LUO Shuoshan^b, XU Qin^e

^a School of Mathematics and Information Science, ShaoGuan University, Guangdong, P.R.China

^b Information Security Center, Beijing University of Posts and Telecommunications, Beijing, P.R.China

^c School of Electronics & Information Engineering, Liaoning University of Technology, P.R.China

^d Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology,
Tianjin University of Technology, Tianjin, P.R.China

^e Beijing Safe-Code Technology CO., LTD. China

xcy@sgu.edu.cn, snowbaiyuan@163.com

Abstract—A DIDS(Distributed Intrusion Detection System) named as F-DIDS is built to defend against flooding DoS(Denial of Services) attacks in the paper. F-DIDS is composed of F-IDSes which are settled in nodes who need to be protected. F-DIDS is a complete DIDS, in which each node with F-IDS can be the detection centre. Due to no central node, single failure can be avoided in F-DIDS. In F-DIDS, Traffic tables help to build up traffic matrixes. By analyzing traffic matrixes, flooding DoS attacks could be detected and three normal flooding DoS attacks are classified. Local and global communication methods are proposed to reduce the overhead brought from fully distributed architecture. The simulation results and performance analysis show that F-DIDS works effectively.

Keywords—Denial of Service, Distributed Intrusion Detection System, SYN Flooding, ICMP Flooding, UDP Flooding

I. INTRODUCTION

As a serious treat to network securities, DDoS(Distributed Denial of Service) is always hard to be detected[1]. In DDoS attacks, the hacker sends malicious flows from a wide area to the victim. On the node near to the sources abnormal signs are inconspicuous; however, while the attacks behavior could be felt, the networks have been throttled and then detecting attacks becomes meaningless. It is the major point and also the most difficult to detect DDoS as near to the source as possible, which is discussed in the paper. Since DIDS(Distributed Intrusion Detection System) gathers data with distributed components to analyze data from the whole area, DIDS is a strong tool to detect and defense against DDoS attacks.

Fully distributed architecture is used to build a DIDS named as F-DIDS to detect flooding DDoS attacks in the paper. A single component settled in a specific network element is called F-IDS, which is based on traffic matrixes. In a local network area, all F-IDSs gather data, communicate

with each other and analyze data. All F-IDSes compose F-DIDS. As a fully DIDS, each F-IDS could be the central processor and make decision by itself. Local and global communication methods are proposed to reduce the overhead brought from fully distributed architecture. Fully architecture avoids the single point failure.

This paper is organized as follows. Related work is introduced in the Section 2. Design of F-IDS and F-DIDS are respectively discussed in Section 3 and 4. The performance analysis and simulation are given in Section 5. Finally the conclusions are summarized in the last section.

II. RELATED WORK

IDS(Intrusion Detection System) is the initial system to detect DDoS or DoS, which places detection component in the important network node[2][3]. The data will not be exchanged among IDSs. With the development of networks, DoS attacks arise in distributed way, which makes IDS powerless to defense against DDoS attacks. In order to detect attacks as near to the hacker as possible, DIDS becomes the first selection[4].

There are three architectures of DIDS: Distributed central-control systems, distributed hierarchical systems and fully distributed systems[1]. In central-control systems, data gathered by distributed components are sent to a monolithic engine where data are analyzed and decision is made, which is easy to cause a single point failure. This serious weakness led to research on distributed hierarchical systems[5][6], which describe a cooperative system without centralized analysis components. In this approach, the local intrusion detection components look for local intrusions and pass the results of their analysis to the upper levels of the hierarchy. The components at the upper levels analyze the refined data from multiple lower level components and seek to establish a global view of the system state[7]. The major disadvantages of hierarchical DIDSs are the heavy network load and network latency (the delay between each level in the architecture). Moreover, there is still highest-level entity, which is the

bottleneck of this system and leads a single point failure. In fully distributed systems, each detection node could be central. An intruder can just destroy one point if possible, but other points still work. However, the most common shortcomings in existing fully DIDS architectures are that there are very large amounts of data being transmitted among the detectors which may occupy the networks channel. [7] solves the problem with shorter messages. In the paper, different communication schemes help to reduce the communication overhead. A fully DIDS is also discussed in [1]; however, three main flooding attacks are not classified in [1]. By traffic matrix, the type of flooding DDoS attacks is made clear in F-DIDS.

III. DESIGN OF F-IDS

A F-IDS is composed of traffic gathering module, traffic table, traffic matrix and communication module. The organization of F-IDS is described in Figure 1. Traffic gathering module gathers traffic in order to store it in traffic table to set up traffic matrixes. Different packets' incoming rates, such as SYN, ICMP and UDP ones, will be sampled in traffic gathering module. The content of traffic table comes from either its own traffic gathering module or others' communication modules. Data in traffic table are used to set up traffic matrix so that the flooding DoS attacks' alerts could be classified into four levels denoted as 0th, 1st, 2nd and 3rd warnings.

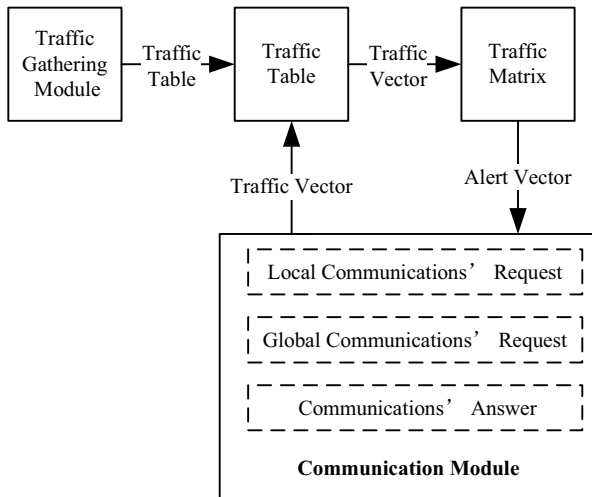


Figure 1. Organization of F-IDS

A. Traffic Gathering Module

Data sampled by traffic gathering module include IPR_i (Incoming Packets' Rate), SYN_i (Incoming SYN Packets' Rate), ACK_i (Incoming Rate of Acknowledgement Packets for SYN), $ICMP_i$ (Incoming ICMP Packets' Rate), UDP_i (Incoming UDP Packets' Rate). IPR helps to detect flooding attacks. SYN and ACK are used to detect SYN flooding attacks, while $ICMP$ and UDP work for ICMP flooding attacks and UDP flooding attacks respectively. All data are quantized into 8 levels: from 0 to 7.

B. Traffic Table

Data in traffic table comes from either Traffic gathering module or communication module. Traffic table is represented as Table 1. *Fresh* is set 1 if traffic vector is valid. Traffic vector is one row of traffic matrix. Arriving time is the time when vector arrives, which is related to *fresh* signal. If *local* is set 1, the node is the local node.

Table 1. TRAFFIC TABLE

Address	Fresh	Local	Traffic Vector	Arriving Time

C. Traffic Matrix

TM (Traffic Matrix) is a $n \times 5$ matrix. Here, n is below the sum of nodes in the networks.

$$\mathbf{TM}_{n \times 5} = \begin{pmatrix} IPR_0, & SYN_0, & ACK_0, & ICMP_0, & UDP_0 \\ IPR_1, & SYN_1, & ACK_1, & ICMP_1, & UDP_1 \\ & & & & \dots\dots \\ IPR_{n-1}, & SYN_{n-1}, & ACK_{n-1}, & ICMP_{n-1}, & UDP_{n-1} \end{pmatrix}$$

The simplest TM is the traffic vector denoted as $\mathbf{TM}_{1 \times 5}$.

$$\mathbf{TM}_{1 \times 5} = (IPR_0, SYN_0, ACK_0, ICMP_0, UDP_0)$$

$$IPR_0, SYN_0, ACK_0, ICMP_0, UDP_0 \in \{0, 1, 2, 3, 4, 5, 6, 7\}$$

AV (Alert Vector) denotes the level of alert.

$$\mathbf{AV} = (Flood, SYN, ICMP, UDP)$$

$$Flood, SYN, ICMP, UDP \in \{0, 1, 2, 3\}$$

Flood denotes the possibility of Flooding DoS attacks. Similarly, *SYN* is for SYN flooding attacks; *ICMP* is for ICMP flooding attacks; and *UDP* is for UDP flooding attacks.

Accuracy factor denoted as α could avoid repetitive communications. In 2nd alert, if α is big enough, which means there are enough traffic vectors in traffic table, the global communications will be avoided and communication overhead will be reduced.

$$\alpha = \frac{m}{m + \left\lfloor \frac{n}{2m} \right\rfloor}$$

" n " is the sum of nodes in the networks. " m " is the sum of nodes whose *fresh* is 1.

$$Flood = \left\lfloor \frac{\sum_{i=0}^{n-1} IPR_i \times \alpha}{2n} \right\rfloor \quad ICMP = \left\lfloor \frac{\sum_{i=0}^{n-1} ICMP_i \times \alpha}{2n} \right\rfloor$$

$$UDP = \left\lfloor \frac{\sum_{i=0}^{n-1} UDP_i \times \alpha}{2n} \right\rfloor \quad S_i = \left\lfloor \log \frac{ACK_i + \alpha}{SYN_i + \alpha} \right\rfloor$$

$$\bar{S} = \left\lceil \sum_{i=0}^n S_i / n \right\rceil \quad SYN = \begin{cases} \bar{S} & \bar{S} < 4 \\ 3 & \bar{S} > 4 \end{cases}$$

IV. DESIGN OF F-DIDS

A. Format of TC Packets

Table 2. FORMAT OF TC PACKETS

Type	ID	Source Address	Destination Address	Operand
------	----	----------------	---------------------	---------

TC(Traffic Communication) packets are used to transmit traffic data. Table 2 shows the format of TC packets.

Type: three types of TC packet are global communications' request, local communications' request and communications' answer.

Packets ID: The Identification is for TC packet.

Source Address and Destination Address: In local and global communications' request packets, source address is the address of the node that sends TC packet and destination address is null. In communications' answer packet, the source address is the address of the node that receives TC packet and the destination address is the last destination of transmission.

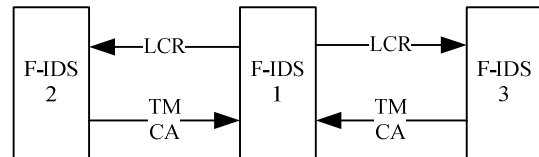
Operand: In local and global communications' request packets, the operand is null. In communications' answer packet, the operand is traffic vector.

B. Local Communications

Two conditions will start local communication: one is *Flood* equals 1 and α is below a threshold, which means the traffic matrix is not "fresh" enough; the other one is to initialize a new node. The procedure of local traffic communication is as follows.

- 1) The source node constructs TC packet with type field set as "local communications' request" and with source address set as the node's address.
- 2) The node sends TC packet to adjacent nodes(local nodes).
- 3) Local node receives TC packet and checks traffic table. If source node is not in traffic table, insert it to traffic table of local node.
- 4) Local node constructs TC packet with type field set as "communications' answer", with operand field set as local node's traffic vector and with destination address set as the address of the source node.
- 5) Local node sends TC packet.
- 6) The source node receives TC packet and then updates traffic table.

Figure 2 describes the procedure of local communications. Here, F-IDS1 sends TC packets to its local nodes:F-IDS2 and F-IDS3. TM of F-IDS2 and F-IDS3 are returned with TC packets.

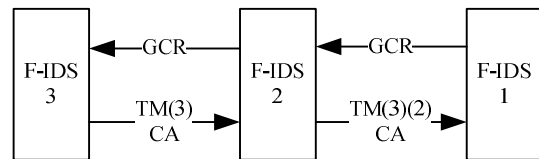


LCR-Local Communications' Request
TM-Traffic Matrix
CA-Communications' Answer

Figure 2. Procedure of local communication

C. Global Communications

When *Flood* equals 2 and α is below a threshold, a global communication will be started. During global communication, traffic vectors spread in the networks. The procedure of global communication is as follows and shown in Figure 3.



GCR-Global Communications' Request
TM-Traffic Matrix
CA-Communications' Answer

Figure 3. Procedure of global communication

- 1) The source node constructs and sends TC packet with type field set as "global communications' request".
- 2) After a node receives TC packet, it will check if it is the first time to receive the packets. If yes, go ahead. If no, discard the TC packet. Type, ID and source address fields of TC packet are the key words during checking and store into a queue with 1K bytes.
- 3) If the source node is new, insert it into traffic table.
- 4) The node transmit TC packet to its adjacent node.
- 5) Construct TC packet with type field set as "communications' answer", with source address set as its address and with destination address set as the source address. TC answer packets also send to its adjacent nodes.
- 6) The node that receives TC answer packet checks if it is the first time to receive the packet. If yes, go ahead. If no, discard the packet.
- 7) If the source node is in the traffic table, go ahead; or else, insert the source node into the traffic table.
- 8) Modify traffic table by traffic vector in operand field of arriving TC packets and check if the destination has been arrived. If not destination, TC packet will be transmitted into next node; or else the transmission will be stopped.

In summary, the flow chart of a F-IDS is represented as Figure 4.

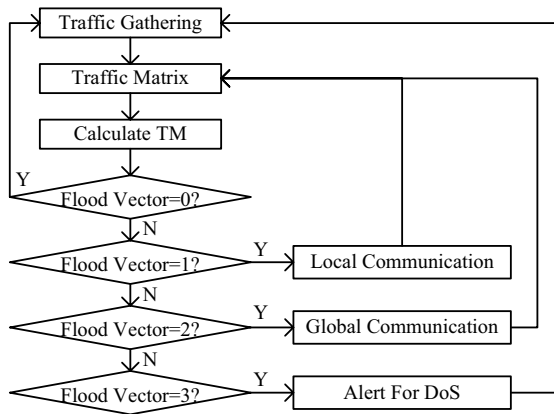


Figure 4. Flow Chart of F-IDS

V. SIMULATION OF F-DIDS

Figure 5 is the topology of the simulated networks. There are 5 nodes with F-IDS who construct a F-DIDS. NS-2 is the tool for simulation.

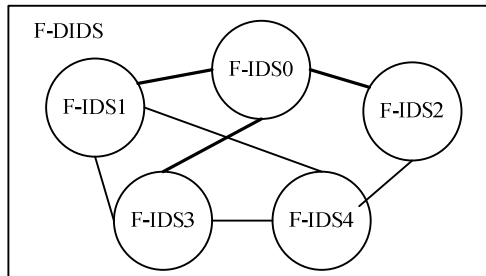


Figure 5. the Topology of Simulated Networks

Assume packets flows sent by DDoS hackers pass through node1 with F-IDS1, node2 with F-IDS2 and node3 with F-IDS3. Node0 with F-IDS0 is the victim. Figure 6 shows the incoming packets. Figure 7 represents the *flood* level of AV(Alert Vector) in F-IDS0. From 3rd to 5th second, a warning of attack is given.

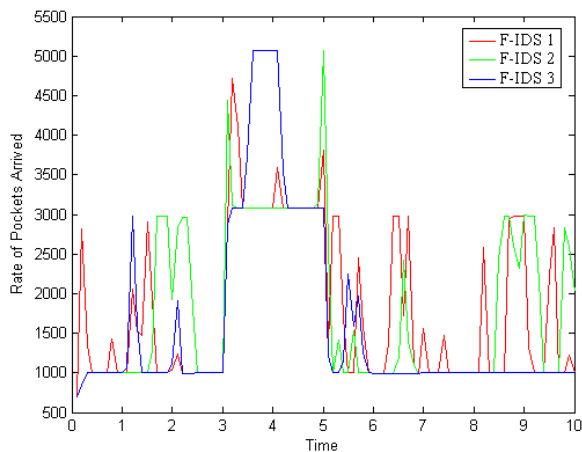


Figure 6. Incoming Packets in Three F-IDS

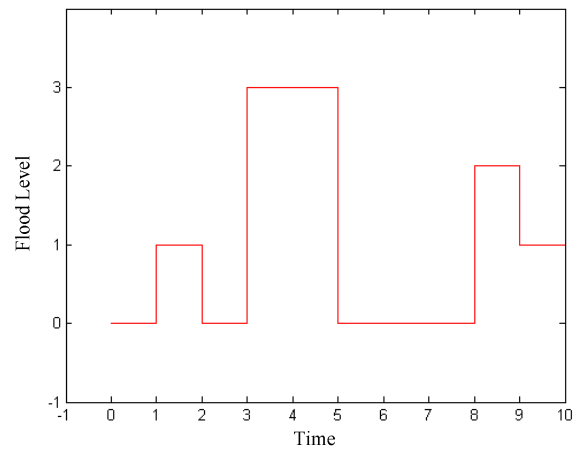


Figure 7. Flood Level for F-IDS0

VI. CONCLUSIONS

Without any central node, single failure can be avoided in F-DIDS. Traffic matrixes help to distinguish three flooding attacks. Local and global communications reduce the overhead of data exchanging.

The simulation results and performance analysis show that F-DIDS works effectively.

ACKNOWLEDGMENT

This study is supported by National S&T Major Project(2009ZX03004-003-03).

REFERENCES

- [1] BAI Yuan, BAI Zhongying: Design and Simulation of a Tree-Based Intrusion Detection System against Denial of Service[J]. Applied Mechanics and Materials, ISSN: 1660-9336.
- [2] D. Novikov, R. Yampolskiy, and L Reznik: Anomaly Detection Based Intrusion Detection. Third International Conference on Information Technology(2006), p. 420- 425.
- [3] H. Kai, H. Zhu, K. Eguchi, N. Sun, and T. Tabata: A Novel Intelligent Intrusion Detection, Decision, Response System (IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences archive, 2006).
- [4] Wei Lin, Liu Xiang, Derek Pao, and Bin Liu: Collaborative Distributed Intrusion Detection System. Second International Conference on Future Generation Communication and Networking (2008), p.172-177.
- [5] Martin Chovanec, Liberios Vokorkos, and Ján Perhák: Security Architecture Based on Multilayer Distributed Intrusion Detection System. 5th International Symposium on Applied Computational Intelligence and Informatics (2009), p.301-306.
- [6] M. Yasin, and A. Awan: A Study of Host-Based IDS using System Calls. International Conference on Networking and Communication (2004), p.36- 41.
- [7] Safaa Zaman and Fakhri Karray: Collaborative Architecture for Distributed Intrusion Detection System. Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (2009), p.1-7.