Design and Implementation of Unauthorized Access Tracing System

Shigeyuki Matsuda, Tatsuya Baba, Akihiro Hayakawa, and Taichi Nakamura
NTT Data Corporation
Department of Information Technology
Kayabacho Tower, 1-21-2, Shinkawa, Chuo-ku, Tokyo 104-0033, Japan
matu@rd.nttdata.co.jp

Abstract

We had proposed a hop-by-hop IP traceback method that can reliably trace a source of an attack. In this paper, we describe the development and the evaluation of our prototype system. The main features of our proposed method are the packet feature, which is composed of specific packet information contained in a packet for identification of an unauthorized packet, and the algorithm using datalink identifier to identify a routing of a packet. We show the development of the prototype system equipped with our tracing functions on routers and its processing result as well as trace time.

1. Introduction

While the Internet as a business infrastructure increases its importance, the number of unauthorized access incidents on the Internet is growing, and such activity tends to cause a great problem.

At present, the access control technologies including firewalls are commonly used to prevent unauthorized access, but some specific way of access cannot be stopped by the access control technologies. Nowadays, installing Intrusion Detection Systems (IDS) coupled with firewalls, and monitoring networks enables us to quickly detect and react to unauthorized access. Figure 1 shows a current dealing with unauthorized access. However, even if these tools can detect unauthorized activities, their sources cannot be identified. The reason is that denial of service (DoS) attacks, which have recently increased in number, can easily hide their sources and forge their IP addresses. Thus, it is not possible for the access control alone to be a deterrence of unauthorized access.

As the measure of unauthorized access, it is necessary to pinpoint the source in order to prevent the unauthorized activity. For this reason, we have been studying a method to identify the source of an authorized activity and developed a prototype system.

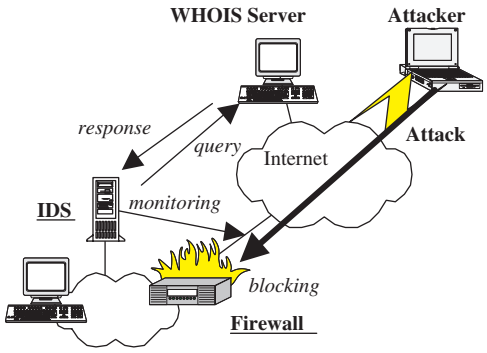

Figure 1. Dealing with Unauthorized Access

2. Related Studies

2.1 Traceback Method

The ability required to perform traceback is "to identify the true IP address of the terminal originating attack packets." If we can identify the true IP address of the attacker's terminal, we can also get information about the organization (e.g. name or telephone number) involved in the attack or the attacking terminal.

As the method of the source pursuit of unauthorized access, some researches using IP (Internet Protocol) are performed. The source pursuit using IP is called IP Traceback. IP traceback methods can be divided into two groups. One group that is categorized as " proactive tracing" prepares information for tracing when packets are in transit. In a case where packet tracing is required, the target of the attack refers information and identifies the source of the packets [1, 2, 3].

Proceedings of the 2002 Symposium on Applications and the Internet (SAINT'02)
0-7695-1447-2/02 $17.00 © 2002 IEEE

The other methods that are categorized as "reactive tracing" start tracing when required. In our study, we have selected reactive tracing that does not increase network traffic at normal times and generates traffic for tracing only when actual tracing is required.

2.2 The Trend of the Reactive Tracing Methods

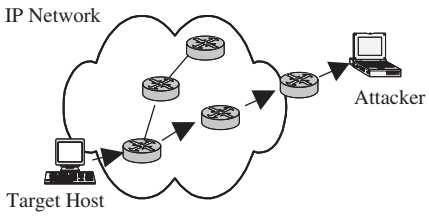

Figure 2. Hop-by-Hop Tracing

The majority of reactive tracing methods trace the attack path from the target back to the source. The challenges involved in this type of method are traceback algorithm and packet matching technique.

(1) Hop-by-Hop Tracing

This method is to trace an IP packet from the target back to the source hop-by-hop, and trace the source based on the incoming packets that arrive one after another during a flood type attack (e.g. the DoSTracker provided by former MCI). Figure 2 shows a flow of trace to detect the source hop-by-hop.

(2) Hop-by-Hop Tracing with Overlay Network

The particular problems involved in tracing routers hop-by-hop are that if there are too many hops, the number of necessary processing for tracing will be increased. As the result, it will take a longer time to trace, and information for tracing can be lost before trace processing is completed. Therefore, a method to build the overlay network for tracing purposes that involves a less number of hops is proposed[4]. With this method, IP tunnels between the edge routers and the special tracking routers have created, and the IP packets are rerouted to the tracking router via IP tunnel. Hop-by-hop tracing is performed over the overlay network that consists of IP tunnels and tracking routers.

(3) IPsec Authentication

Another proposed technique is that when unauthorized access is detected, a Security Association (SA) of the IPsec is created dynamically, and authenticating the packet with IPsec identifies the travel path and the source of the packet[5]. Since this technique uses existing IPsec protocol, it has an advantage that it is not necessary to implement a new protocol.

(4) Traffic Pattern Matching

Another proposed technique traces the forwarding path of the traffic by comparing traffic patterns observed at the entry and exit point of the network based on the network map[6].

3. Traceback Approaches

In the field of reactive tracing study, several methods that identify a source of a packet with forged source IP address have been proposed. Although most of the existing techniques deal with flood type DoS attacks, there are more attacks using only one or a few IP packets such as attacks exploiting IP fragment or UDP. It is important to be able to trace unauthorized access using single packet.

Based on the above, we have proposed a hop-by-hop traceback method [7, 8]. We are developing a system implementing our method even if the attacker forges its source IP address. Our system performs real-time tracing and exactly identifies the source of the specific packet along the attack path.

4. Our Traceback Architecture

4.1 Traceback Method

In general, the source IP address of a packet can easily be forged at the source of the packet. On the other hand, it is difficult for a sender of a packet to forge the datalink-level identifier when sending packets, because, in the event of frame or cell transfer, forwarding unit (such as router) in turn converts the datalink-level identifier to the interface identifier of the unit. Therefore, at each forwarding unit, we can identify its adjacent unit having forwarded a particular packet based on the datalink-level identifier of the adjacent unit and the datalink-level identifier corresponding to the packet.

In our approach, we keep forwarded packets and MAC address corresponding to their datalink-level identifier in each forwarding unit and identify the adjacent unit by searching for the forwarded packet that corresponds to an attack packet. Beginning with the forwarding unit closest to the sensor that has detected unauthorized access, we identify each adjacent forwarding unit along the attack path, and ultimately reach the source of the attack packet even if a forged source IP address is used.

4.2 Our Traceback Model

In this section, we describe our traceback architecture that identifies the source of a packet with forged source IP address. The architecture consists of the following three components:

(1)Sensor

This component has two functions. One is to detect unauthorized access from the network (the same function as existing IDSs have) and another is to request a manager to start tracing.

(2)Tracer

This component implements a function to maintain information about forwarded IP packets as well as a function to trace the source of the forwarded packet along the attack path on forwarding unit.

(3)Manager

In response to a request from a sensor, this component controls traceback tasks.

We can install a tracer and a manager on each unit (Figure 3), or install a single manager as a central manager of the entire network (Figure 4).

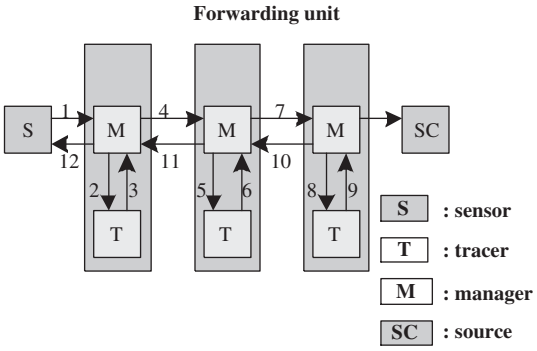

Figure 3. Basic Model of Our Traceback Method

In practical terms, particular network policy may restrict tracing a packet with certain limitation. We cannot trace a packet beyond our own network perimeter if neighboring networks impose different policy. Therefore, we consolidate managers in a specific policy-controlled network perimeter and install an overall manager ("monitoring manager") for each perimeter. Using the monitoring manager, we can give orders for tracing to the different policy-controlled networks and receive the results from them. We select the single manager model that enables us to control and monitor tracing tasks between network perimeters that impose different policy.

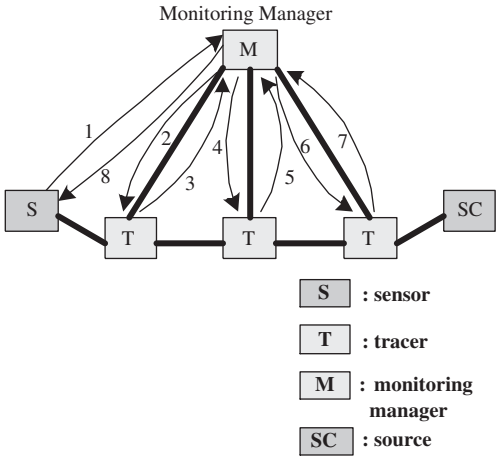

Figure 4. Traceback Model of Single Manager Method

4.3 Traceback Protocol

The basic functions of the traceback protocol define the following tasks:

(1) A trace request from a sensor and a notice of the tracing result to the sensor

(2) A trace order from a monitoring manager to a tracer and a notice of the processing result to the monitoring manager

(3) A trace request and a notice of the tracing result exchanged between monitoring managers

Table 1. Message of Traceback

Message Name	Operation
Trace Request	Trace Request from a sensor to a manager
Notification of Tracing Result	Notification of Traceing Result from a manager to sensor
Trace Order between managers	Trace Order from an original manager to a requested manager
Notification of Tracing Result between managers	Notification of Tracing Result from a requested manager to an original manager
Trace Order	Trace Order from a manager to a tracer
Notification of Processing Result	Notification of Tracing Result from a tracer to a manager

Table 1 describes the protocol commands and their functions.

We explain the tracing flow of Figure 4. If a sensor detects unauthorized access, the sensor notifies the monitoring manager of detection. At this time, unauthorized access's information (packet feature) is notified to the manager. The pursuit of unauthorized access's source starts by this notification (sequence 1:Trace Request). The monitoring manager sends the unauthorized access's information to a tracer with which the sensor is connected, and inquires to the tracer from which tracer the unauthorized access came (sequence 2:Trace Order). The tracer analyzes this information, specifies the tracer by which the unauthorized access came, and returns information on the tracer by which the unauthorized access came (sequence 3:Notification of Processing Result). The monitoring manager decides the next tracer from information returned from the tracer, and puts out the inquiry to the next tracer. This procedure is continued to the tracer with which unauthorized access's source is connected (sequence 4, 5, and 6). The tracer with which unauthorized access's source are connected returns source's information to the monitoring manager, when unauthorized access information from the monitoring manager and source's information is corresponding (sequence 7). The monitoring manager ends the pursuit, and notifies source's information to the sensor (sequence 8:Notification of Tracing Result).

4.4 Packet Feature

Our traceback method uses a packet feature [9] as a parameter for Trace Request and Trace Order. In order to uniquely identify the individual packet, we extract several fields of the IP packet that are not altered by tracers and create a packet feature. The extracted fields are as follows:

- Version
- Header Length
- Identification
- Protocol
- Source and Destination IP addresses
- A part of IP data

If we create a packet feature consisting of only IP header fields, identical packets may exist. Therefore, in order to improve the precision of packet identification, we decide to include a part of IP data field (maximum 20 bytes). Figure 5 shows the structure of the packet feature.

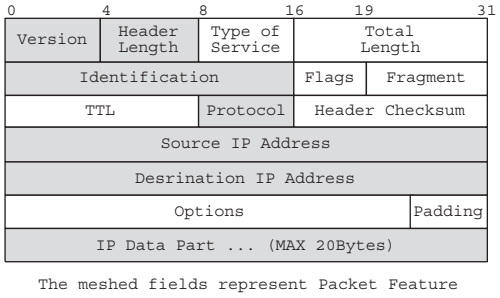

Figure 5. Structure of Packet Feature

5. Implementation of Our Traceback System

We have developed a traceback system based on our architecture and protocol.

5.1 Implementation Model

On our traceback method, we implement sensor for detecting unauthorized access and for making a trace request, tracers for executing trace tasks and a monitoring manager for centrally controlling the tasks.

(1) Sensor
Sensor detects unauthorized activity, then asks for tracing the source and receives the result.

(2) Tracer
Tracer actually traces packets from the victim site to the source of the packets along the attack path. In order to utilize existing forwarding unit, we append the tracing functions to forwarding unit.

(3) Monitoring manager
Monitoring manager orders tracers to start tracing in response to requests from sensor. Additionally, it orders upstream tracer to start tracing. When the source of a packet is identified, monitoring manager returns the result to the sensor. It also monitors the status of tracing tasks based on the status information sent from tracers.

5.2 Implementation Method

This section describes how to implement a tracer, which is one of the major functions in tracing tasks.

5.2.1 Implementation of Tracer

The tracing function consists of 2 modules.

(a) Packet Conversion and Store process

It creates a packet feature from a packet that passes through the tracer, and stores the packet feature.

(b) Trace and Search process

With Trace Order from a manager, the tracer performs tracing process, and returns the result to the manager.

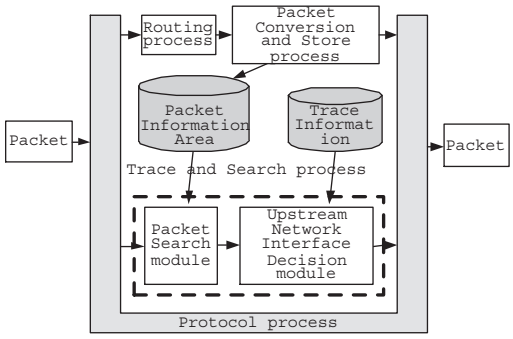

Figure 6. Structure of Tracer

Figure 6 shows the structure and the activities of the tracer.

(1) Packet Conversion and Store process

After routing process, Packet Conversion and Store process gets a packet to forward and creates a record containing the address of the upstream unit (MAC address) and a packet feature extracted from the packet. This record is stored into Packet Information Area in the tracer. Every incoming packet is processed through this procedure.

(2) Trace and Search process

Trace and Search process has two modules: Packet Search module and Upstream Network Interface Decision module. Packet Search module accepts Trace Order and searches for the specified packet feature from Packet Information Area. If a record matching with the trace packet is found, Upstream Network Interface Decision module decides the upstream network interface and notices this trace result to the monitoring manager using Notification of Processing Result.

5.2.2 Trace Algorithm

We have developed and implemented the algorithm that processes Trace Order reception, trace execution for upstream path decision and trace report. Below we describe our algorithm.

When a tracer receives Trace Order, this order accompanied with a packet feature is passed to Packet Search module in Trace and Search process. Packet Search module searches Packet Information Area for the record that matches with the packet feature. On our implementation, searching starts from the latest record in Packet Information Area, and the first record that matches with the packet feature is recognized as the target record to trace.

Next, information of the target record is passed to Upstream Network Interface Decision module. This module compares the address information in the target record (for LAN, this is a MAC address) with the address information (MAC address and IP address) of the connected tracer stored in the trace information. Based on the matching MAC address, the upstream IP address is decided and returned to the monitoring manager as the trace result. The monitoring manager decides the next unit to trace based on the result and order the tracer to start tracing. This process is repeated until the source is detected.

5.2.3 Information Management

There are two types of information used in tracers. One is the packet information that converts traversed packets information into packet features and stores them, and the other is the network interface information that stores network interface information between two units connected each other.

(1) Packet Information Area

For all packets that traverse tracers, we create packet features containing necessary information for tracing. Next, we add network interface information of the unit that forwarded the packet and the forwarding time to the packet feature to create a record, which is finally stored in Packet Information Area. Under LAN configuration, the MAC address is the network interface information. Tracers store these records in the order of creation until the memory capacity becomes full. On our implementation, records are stored in the memory area of the tracer for the purpose of real-time processing. If the volume of Packet Information Area exceeds the memory capacity, the oldest record will be deleted and the latest one will be stored in turn.

(2) Trace Information

We have studied three methods for obtaining network interface information from the unit connected with the tracer.

Method 1: trace table method

Checking the network interface number, IP addresses and physical addresses (e.g. MAC address on LAN) of the connected tracers in advance, and storing them in the unit.

Method 2: ARP table method

Using the ARP table stored in the unit to look up

the IP address and physical address of the connected tracer when Trace Order is received.

Method 3: order-driven query method
Without providing a fixed table, obtaining network interface information using the lower layer protocols (e.g. RARP protocol) in response to Trace Order.

We have reviewed each method and reached the following conclusion: As network interface information is temporarily stored in the ARP table, some information may be changed or may not be available when searching the table; Although the order-driven query method is suitable for obtaining the latest network interface information, the process is complicated and takes longer time because the query task to the adjacent node is called every time a trace order is issued. Therefore, we select the trace table method that provides real-time, reliable, and efficient tracing.

5.2.4 Required Memory of Tracer

We retain packet information in the memory area of the tracer. However, the memory area has a limited capacity. Therefore, we examine possible duration for Packet Information Area to retain packet information based on the network traffic and memory capacity of the unit. We set the following conditions for the examination: the packet stream to the tracer is from 1 Mbps to 10 Mbps, and the packet length is 1,000 bytes. The length of the record retained is 60 bytes, and it has packet feature, time, address of the upstream interface and sequence number fields. Figure 7 explains the estimation of memory capacity necessary for the tracer.

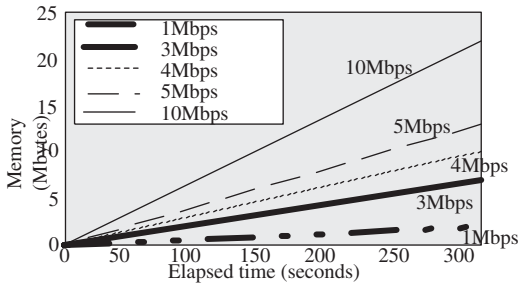

Figure 7. Required Memory of Tracer

6. Implementation of the Traceback System

6.1 Implementation

We have implemented sensor, tracers and monitoring manager on our traceback system. Table 2 shows the specification of the implementation

Table 2. Specification of Implementation

Tracer	Kawasaki Steel A2DIS SV-1000 CPU:MC68360 25MHz Memory:16MB `(Packet Information Area:8MB)` Network Interface:10Mbps OS:INFOS/INCS
Sensor	DELL PowerEdge1300 CPU:600MHz PentiumIII Memory:384MB Network Interface Card:Intel Pro/100+x2 OS:FreeBSD 4.2-RELEASE
Monitoring Manager	Sun Enterprise 250Server CPU:UltraSPARC-II 400MHz x 1 Memory;512MB OS:Solaris2.7

6.2 Experimental Result

We have built a traceback system equipped with 1 sensor, 3 tracers, and 1 monitoring manager. Figure 8 shows a process flow of the trace from an attacked server to a terminal with forged IP address. This figure shows to detect the source of attacking terminal.

6.3 Experimental Result of Trace Time

We have built a traceback system equipped with 1 sensor, 10 tracers, and 1 monitoring manager (Figure 9). In this measurement of a trace time, we have connected a terminal with forged IP address to 1st, 2nd, 3rd, 4th, 6th, 9th, 10th tracer, and have attacked from this terminal to the target server. We have measured the trace time from an issue of Trace Request until an acceptance of Notification of Tracing Result. We have assumed that packet features were filled in Packet Information Area, and Packet Source module has searched a packet feature until the end of Packet Information Area. And only trace packets existed on the experimental network, while tracing.

We have measured the trace time of each measurement point 5 times and calculated the average values of

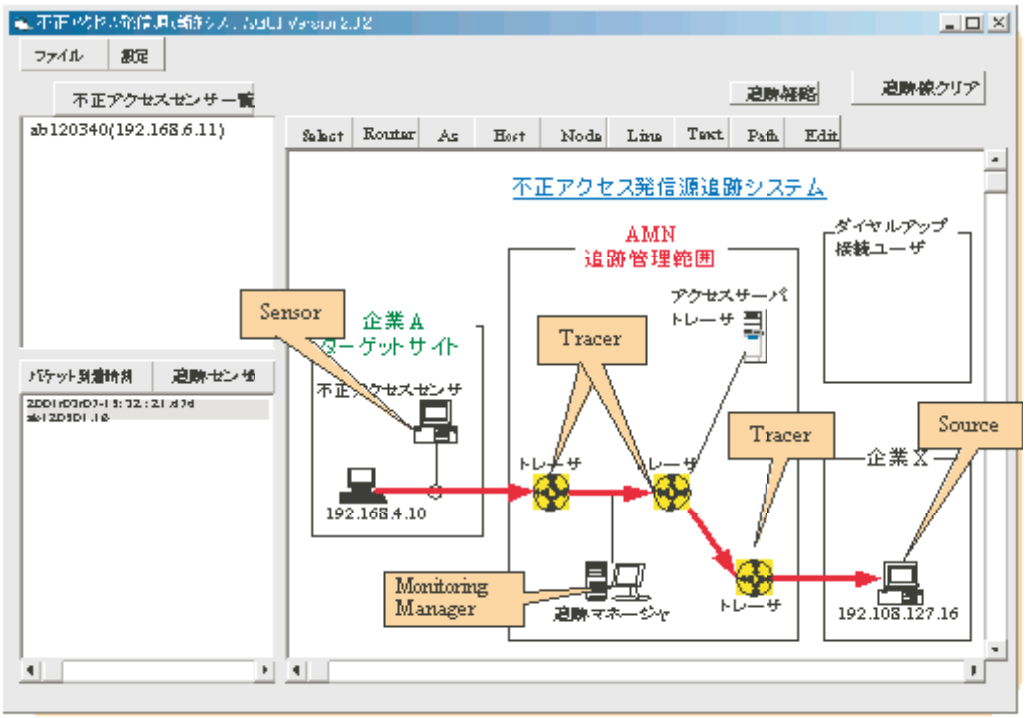

Figure 8. Diagram of A Process of Trace Flow

3 values except a maximum and a minimum value. Table 3 shows our experimental result of trace time [10].

This measurement time contains the communication time between the sensor and the tracer, the monitoring manager, the pursuit processing time in the tracer, and the processing time in the monitoring manager. This experiment result showed that the trace time in the 10 tracers was less than 3 seconds, when there was no network load.

Table 3. Experimental Result of Trace Time

Number of Hop	Elapsed time (sec)
1	0.300
2	0.419
3	0.609
4	0.910
6	1.723
9	2.464
10	2.598

7 Conclusions and Future Work

We have created a traceback system that can pursue the source even if an IP address is forged, and have demonstrated the effectiveness of the traceback processing. We will change the network load and measure the trace time. And we will consider the relationship among trace time, the network load, and the number of tracers.

In the viewpoint of the introduction of the traceback, we have 2 subjects. First subject is method to identify matching packets and identify the sources under DDOS attack where identical packets are sent from different sources. Second subject is method to introduce the tracer function. At the first step, the introduction of this method assumes the limited network such as Intranet. We think that it is possible to implement the tracer function on all the network equipments in such a network environment. However, it is assumed that it is impossible to implement the tracer on all the network environments by the open network. Then, the method that the source can be pursued is needed when the tracer function is partially introduced.

We will further study how to improve the accuracy of the packet search process., and develop the IP

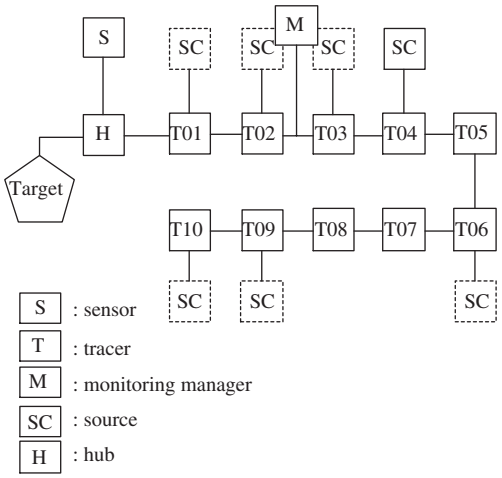

Figure 9. Experimental Environments

Traceback system to catch the source of attacking site against a variety of attacks.

8 Acknowledgements

This work is funded by the Telecommunications Advancement Organization of Japan (TAO).

References

[1] S.Savege, D.Wetherall, A.Karlin, and T.Anderson. "Practical Network Support for IP Traceback," *in Proceedings of the 2000 ACM SIGCOMM Conference*, 30(4):295–306, August 2000.

[2] K.Park and H.Lee. "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack," *Technical Report, Purdue University*, (CSD-TR 00-013), June 2000.

[3] D.X.Song and A.Perrig. "Advanced and Authenticated Marking Schemes for IP Traceback," *Technical Report, University of California at Berkeley*, (UCB/CSD-00-1107), June 2000.

[4] R.Stone. "CenterTrack: An IP Overlay Network for Tracking DoS Floods," *in Proceedings of the 9th USENIX Security Symposium*, pages 199–212, August 2000.

[5] H.Y.Chang, R.Narayan, B.Vetter S.F.Wu, M.Brown, X.Wang, J.Yuill, C.Sargor, F.Gong, and F.Jou. "DecIdUouS: Decentralized Source Identification for Network-Based Intrusions," *in Proceedings of the 6th IFIP/IEEE International Symposium on Integrated Network Management*, pages 701–714, May 1999.

[6] K.Ohta, G.Mansfield, Y.Takei, and Y.Nemoto. "Detection, defense, and tracking of Internet wide illegal access in distributed manner," *in Proceedings of INET 2000*, July 2000. http://www.isoc.org/inet2000/cdproceedings/1f/1f_2.htm.

[7] K.Kokubo, H.Watanabe, S.Matsuda, et al. "A study of unauthorized access tracing system," *in Proceedings of the 60th National Convention of IPSJ(Japanese only)*, vol. 3:283–284, March 2000.

[8] S.Taketsume, S.Matsuda, H.Watanabe, M.Yanagida, and K.Kokubo. "A Study of Architecture for Unauthorized Access Tracing System," *in Proceedings of the 60th National Convention of IPSJ(Japanese only)*, vol. 3:287–288, March 2000.

[9] H.Watanabe, T.Baba, S.Taketsume, and S.Matsuda. "A Study of packet identifier for unauthorized access tracing system," *in Proceedings of the 60th National Convention of IPSJ(Japanese only)*, vol.3:289–290, March 2000.

[10] M.Ikeda, S.Tanaka, A.Hayakawa, and S.Matsuda. "A Study of architecture for unauthorized access tracing system," *in Proceedings of the 62nd National Convention of IPSJ(Japanese only)*, vol.3:285–286, March 2001.