Adaptive and Selective Packet Marking in Communication Networks

YACINE DJEMAIEL, SLIM REKHIS, and NOUREDDINE BOUDRIGA
Communication Networks and Security Research Lab
SupCom
University of the 7th of November at Carthage
Tunisia
http://www.supcom.mincom.tn/cnas/

Abstract: - We propose a novel traceback approach that marks IP traffic by applying selective marking and reducing load mechanisms. Our technique is adaptive and is exploiting any specific properties that help characterizing an activity in communication traffic. It helps reducing problems such as processing overhead, bandwidth overload, detecting security attacks, and handling encrypted traffic. The selective character of our method reduces tremendously the performance cost observed with other marking approaches. The approach is applied to TCP traffic, for which an IP traceback has been implemented and integrated into what is called "Click software router". Finally, a comparative study is made with the Deterministic Packet Marking technique.

Key-Words: - IP traceback, selective marking, mark, SYN traffic, Three-way TCP Handshake, Click.

1 Introduction

In recent years, interconnected networks have been threatened by a large spectrum of cyber attacks, which caused serious damages (e.g., theft of confidential documents and denial of service). To face this situation, a new research field has attracted the security and network community since 1999. The aim of that activity was to study and provide efficient techniques to detect attacks and locate their sources. Since that date, several IP traceback approaches have been proposed. Some approaches have met success, but did not solve problems like the routers processing overhead and the inability to handle specific categories of IP traffic (e.g., encrypted and fragmented traffics), which are considered and defined among the evaluation metrics for IP traceback approaches [1].

To overcome such shortcomings, we propose a novel IP traceback approach, called ***A**daptive and **S**elective **P**acket **M**arking* (ASPM), which marks traffic by applying a selective method and reducing the induced processing load. Unlike previous approaches, ASPM marks only packets that match a set of specific properties assigned to the used protocol and that we have stored in a knowledge database. Marking this kind of packets has provided the opportunity to insert some useful information that helps locating the attack origin.

The remaining of this paper is organized as follows. We present, in Section 2 the background information about major IP traceback approaches. We also discuss some problems that have not been solved yet. Section 3 introduces ASPM marking approach and describes its main components. It also addresses the mechanisms added to improve marking performance. Section 4 details the verification of marks. In the following section, we consider an application of ASPM to the context of TCP traffic. Section 6 gives a validation of ASPM, by performing a simulation. Finally, Section 7 concludes the paper.

2 Marking approaches

Recently, several IP-based traceback approaches have been proposed. These approaches use several mechanisms such as the packet logging, packet marking, sending additional traceback traffic, and link testing. In this section, we give the state of the art related to the marking techniques. We particularly consider the mark reconstruction process and the problems that have not been solved by these techniques.

2.1 Marking Principle

The basic idea of marking is simple: Marking routers insert (deterministically or probabilistically) some useful information, called *mark*, into the traffic packets during their transmission. While deterministic approaches mark all IP traffic, probabilistic techniques mark the traffic packets with a given probability *p*. Most of the proposed techniques use the same information to be inserted in the mark, which contains the IP address of the marking node or the router marking interface. On the other hand, other approaches insert edge information to the mark like the edge sampling algorithm (as described in [7]). Marking may be performed at all nodes that may occur in an attack path for the majority of marking approaches. Deterministic packet marking schemes, however, mark packets at the network entry point only.

Mark is encoded and inserted into IP traffic using different ways. Few techniques divide the marking node IP address into two or more fragments and insert them

probabilistically or deterministically into the IP packets. To do this, most of the proposed marking techniques have chosen to overload the 16-bit IP Identification field used for fragmentation in the IP header. The Deterministic Packet Marking scheme (DPM) presented in [3] divides the IP address into two 16-bit fragments. With a probability equal to 0.5, the identification field of each incoming packet will be assigned one of the 16-bit fragments.

At the victim level, marking approaches behave differently when dealing with marked traffic. While some approaches attempt to reconstruct the whole attack path (this is the case for probabilistic marking schemes [7], for example), the other techniques, like the deterministic packet marking, only reconstruct the IP address of the edge ingress router. The reconstruction process depends on the number of received packets. Typically, marking approaches that divide the mark into more than two fragments may need an important traffic to be able to reconstruct the IP addresses of all nodes belonging to an attack path.

2.2 Schemes Insufficiencies

The majority of IP traceback approaches have provided a means to determine the path to an attack source. Unfortunately, efficient results are not provided all the time they are needed. These approaches introduce problems that may prevent their integration into a real environment, if they are not solved, because of the following three facts: (1) the provided techniques introduce an important processing overload, which is not recommended at the router level; (2) additional traceback information inserted into incoming packets (or sent in separate messages) may introduce a bandwidth overload; and (3) the available approaches do not process all types of traffic. Encrypted and fragmented traffics belong to that category. Furthermore, dividing marks into fragments causes reconstruction problems when the victim does not receive all the fragments needed to reconstruct the correct mark. A new IP traceback approach should propose a solution to these problems and ensure that a victim will be able to trace the majority of attack traffics.

3 The ASPM scheme: The marking process

3.1 ASPM basic scheme

ASPM is an IP traceback approach which adapts its behavior according to the nature of the attack traffic. The basic idea of ASPM is to select and mark attack traffic that matches specific properties in order to reduce or avoid major IP traceback problems (e.g., processing overload and bandwidth overload). ASPM adaptive processing is based on the interaction with a Knowledge Database, which stores protocol specific properties that can be exploited by the scheme to provide marking. Properties are represented as rules in order to be easily interpreted by the ASPM selector component (see Section 3.3.1). The rule may be composite, if many properties are available for a selected protocol and that should be fulfilled together by the incoming traffic. We have defined a SQL-like language for describing rules and retrieving them.

ASPM marking process is enabled at the closest interface to the source of the packet on the edge ingress router. A Knowledge Database, containing the specification of protocol properties, may be made available at each ASPM enabled marking node (or at a central site on an Internet domain, which is less expensive but adds considerable delays to the marking process).

3.2 Properties Description Language

When describing the ASPM principle, we have indicated that properties are translated to rules; and then stored in the knowledge database. These rules should be easily interpreted by the ASPM selector component to avoid confusion or mistreatment. That's why we have chosen to use a simple language that is similar to SQL and which provides the possibility of translating properties into rules that can be viewed as a selection query.

Similarly to SQL language, an ASPM rule is defined using two basic keywords: *select* and *where*. The former instructs the ASPM selector component to select a traffic unit, like a packet, a datagram or a cell, so the first part of the rule presents the action to perform by the selector. The second part of an ASPM rule starts by the *where* keyword which is followed by a set of selection criteria. An ASPM rule is written according to the following syntax:

select *traffic unit* ***where*** $criterion_1$, ..., $criterion_n$

A selection criterion is written following a tree based representation. This syntax is simple and illustrates better the protocol hierarchy. An example of ASPM rule will be provided later for the TCP protocol (see Section 5.1).

3.3 Marking procedure

In this subsection, we will first describe the ASPM marking module then we present some of its features and the mechanisms it provides.

3.3.1 Marking module description

ASPM marking module, depicted by Figure 1, includes four major components. They are: the selector, collector, the key generator, and the ASPM marker.

Selector This component processes the incoming traffic to select the packets to mark. It first checks the traffic protocol type and then queries the knowledge database for the relevant rules to apply. Based on the retrieved

rules, the selector applies the reducing load mechanism to decide whether the packet should be marked or simply forwarded.

Collector This component treats the selected packets in a given traffic to extract the information needed for computing the related HMAC digest, which is appended to authenticate the ASPM mark and the carrying traffic unit. Information are extracted from the invariant fields of the traffic units (e.g., for an IP traffic, one can consider: the protocol type, sequence number, source port, and destination port). In addition, the collector extracts the IP address of the marking interface as well as the marking date and time.

Key generator Authentication keys are used to compute HMAC digests for marks. To generate random keys, we have used a pseudo-random number generator. We added temporal information to the generated pseudo-random number using the XOR operator. We then applied a hash function to generate the authentication key which can be, according to [5], of any length up to the block byte-length of the HMAC hash function, B.

ASPM Marker This component first generates the HMAC digest for the set of information constituted by the marking interface IP address, marking date, marking time, and collected content delivered by the collector. The digest is computed using the authentication key. After generating the digest for the computed HMAC, ASPM marker inserts the marking interface IP address, the date and time of marking, and the digest. Finally, the marked packets are directed to the forwarding module.

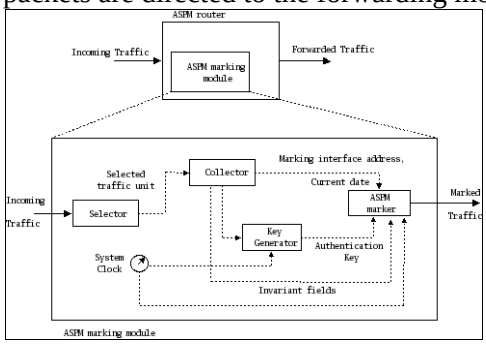

Fig.1 – ASPM generic architecture

3.3.2 Reducing load mechanism

The basic ASPM version marks all outgoing packets that match the rules in the knowledge database at the ingress router interface. This marking scheme may lead the outer to an overloaded state, if an attacker generates an attack with large traffic. To avoid such a situation, we have introduced a reduction mechanism that uses a table, called *TempTbl*, and makes the choice that only the first traffic unit is marked among those units (within the same traffic) for a fixed lifetime period. At this level, a daemon is running and its function is to delete each *TempTbl* entry when the lifetime period has expired compared to the *TempTbl* entry receiving time. Each entry in *TempTbl* contains the source IP address, destination IP address, and receiving time of the selected unit for marking. Before marking a selected traffic unit, the ASPM selector checks if an entry already exists in the *TempTbl* table (i.e., if the unit has been preceded by another unit that has been marked); if that's the case, the selected traffic unit will be passed to the forwarding module; else it will be forwarded to the ASPM marking module after adding a new entry to the *TempTbl* structure.

Following this mechanism, we have not inserted the mark in the *TempTbl* structure. This adopted choice, will keep the memory requirement for *TempTbl* structure as minimal as possible.

3.3.3 Authentication of the mark

Marking traffic units at a network entry point ensures that spoofed marks, inserted by an attacker located ahead of the node, will be overwritten by the ASPM enabled router. If a router located after the network entry point was compromised, an attacker is able to forge a mark instead of the ASPM mark. To overcome this problem, ASPM introduces an authentication mechanism which is similar to the one proposed in [6]. This mechanism computes the HMAC [5] of the mark using time-released Key chains. Chain of keys is obtained for a router R_i according to the following relation: $K_{j,i} = g(K_{j+1,i})$ for a given function g. The first key is obtained by applying a hash function to a randomly selected seed. The first generated key is signed with the ASPM router private key and be published on the ASPM authentication server. The scheme assumes that each ASPM router has a digital certificate.

After expiration of the key lifetime, each ASPM router discloses the key used to authenticate the mark, as well as the hash function under use. The key and hash function are then changed for use in another interval of time. For each interval, the disclosure time is published with the key, the hash function, and the router ID.

4 The ASPM scheme: Verification procedure

At the victim side, received marks are handled by another module called *TraceSrc*. This module is integrated into an Intrusion Detection System, as depicted by Figure 2.

The *TraceSrc* module performs three basic functions, which will be detailed in this section.

4.1 Mark extraction and storage

Marked traffic will be forwarded to a module called *TraceSrc*, which extracts the mark and the fields used to compute the HMAC digest. The receiving time and the source IP address are added to the extracted information and then inserted into the *TraceDB* database. Before

extracting the mark, *TraceSrc* checks the type of the incoming traffic unit and selects from *TraceDB* database the information for the involved protocol. Then, *TraceSrc* checks whether the authentication key is disclosed. If the key is disclosed, the traffic unit is considered not valid and a notification is generated and sent to the analyzer. In the other case, *TraceSrc* module inserts the extracted information in *TraceDB* database.

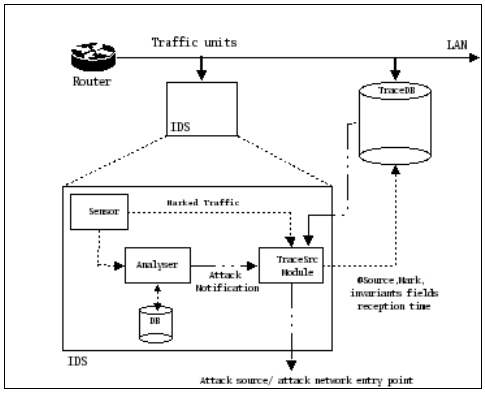

Fig.2 – *TraceSrc* module integrated with an IDS

4.2 Mark validation

To avoid processing overhead, mark validation is not done in real time for ASPM. *TraceSrc* module extracts the marking time from the *TraceDB* database and generates a request to the ASPM authentication server, which processes the request and selects the time interval that contains the marking time. Then it responds by sending the key and the hash function used during the selected interval. After receiving this information, *TraceSrc* starts the mark validation process by computing the HMAC digest using the content of fields extracted from *TraceDB* database and the corresponding authentication key. Next, *TraceSrc* computes the digest from the HMAC using the appropriate hash function. This digest will be compared to the digest extracted from the marked traffic unit. If the two digests are equal then the mark is valid, else it is ignored and an alert is sent to the IDS analyzer. After checking the mark validity, *TraceDB* database is updated to indicate that the marking process has been accomplished.

It is important to notice that both requests and responses exchanged between IDSs and ASPM authentication servers are digitally signed. Consequently, IDSs and ASPM authentication servers must have their digital certificates. *TraceSrc* module may be activated after receiving an attack notification from the IDS analyzer. In that case, it searches the mark in *TraceDB* database and checks whether it is valid.

For published marking approaches, the victim should perform many operations to reconstruct the IP address from different fragments. If there are some fragments missing, the reconstruction process fails and the victim is not able to reconstruct the path or to determine the IP address related to the network entry point node. For our scheme, *TraceSrc* does not have to face the reconstruction problem because the mark, it uses, is not fragmented; it is encapsulated on a single packet. Search requests are made using the content of appropriate fields like the IP source address, TCP source port, TCP destination port, and sequence number. If an entry matches the request, *TraceSrc* module returns the source IP address and the IP address of the ASPM marking interface. In this case, the path to the attack source may be generated using the *traceroute* utility.

5 ASPM Behavior for TCP Traffic

In this section, we consider the marking issues related to the TCP protocol. We present first, the TCP property used by the ASPM scheme for the traffic selection and the behavior of this scheme faced to specific traffic categories.

5.1 Property for TCP

Statistics collected from several sites during the recent five years have shown that the TCP traffic presents the majority of the traffic exchanged on the Internet [4] and that 90% of denial of service attacks have used TCP traffic. Based on these observations, we have focused our study on TCP traffic and have found an interesting property that can be used to select traffic for marking. This property is related to the first TCP packet, exchanged during the three-way TCP Handshake mechanism, which is composed only by the two headers (IP and TCP). For this, the following rule has been generated and added to the knowledge database:

select packet ***where*** tcp.flags.syn = 1

5.2 ASPM behavior against TCP traffic

In this section, we consider different kinds of traffic and detail the ASPM behavior with each one of them.

1. Encrypted TCP traffic In the case of an encrypted connexion, the first three packets used in the Three-Way TCP handshake protocol are not encrypted. Therefore, ASPM is able to mark attack traffic before establishing the tunnel.

2. Fragmented TCP Traffic The fragmentation process occurs when a packet enters a network having a Maximum Transfer Unit (MTU) smaller than the packet length. The original packet will be transformed into a set of fragments where each fragment represents a packet containing a portion of the payload of the original packet. From this definition, we can deduce that ASPM marked packets are not concerned by the fragmentation process because they contain only the IP and TCP header without additional data. Moreover, SYN packets have a small size that is lower than many known MTUs. If an artificial fragmentation has been performed by an intruder, the victim is able to reconstruct the original packet from fragments associated to the ASPM marked packet. This

operation is possible because ASPM does not overload the identification field during the marking process unlike the other marking approaches. In this situation, two cases may occur as detailed in [7]:

- **Upstream fragmentation:** In the case of upstream fragmentation, the packet is fragmented by a router or a host before it reaches the ASPM enabled interface. In that case, ASPM enabled interface will mark the selected fragments by appending the mark to the payload field. At the target site, *TraceSrc* module will extract the mark from the fragment but the victim will not be able to reconstruct the packet and will reject it. Marking this kind of malicious packets can provide the victim the ability to collect information about the attack network; and at the same time it has protected the victim from this malicious traffic by stopping the attacker attempt before opening a TCP connexion.
- **Downstream fragmentation:** If the fragmentation is performed by the downstream routers, the marked traffic will be processed similarly to normal traffic. In this case, the ASPM router is not concerned by fragmentation and does not cause any problem for the reassembly process. When fragments are reassembled, *TraceSrc* module extracts the mark from the packet and stores it in *TraceDB* database.

3. Local Traffic According to ASPM marking principle, it appears that marking the traffic exchanged between to sub networks that belong to the same local area network cannot be achieved. This justifies the need to have some protection and logging mechanisms be deployed at the LAN level, in order to determine the origin of attacks.

6 ASPM Simulation

We have carried out some simulation experiments to compare ASPM and DPM marking processing time for two attack scenarios. We start by defining the simulation model and the input and output simulation parameters. Then, we analyze the given results to highlight the positive impact of ASPM.

6.1 Simulation model

We have implemented and integrated ASPM marking module in the Click modular router [2]. We have also implemented the basic DPM marking procedure which is described in [3]. Our simulation experiments treat only the ASPM router because we are interested at evaluating the router processing time for marking actions. For this purpose, we have connected two segments by the click router. The first segment is used as an attacker network; however, the second (segment # 2) is the location of the victim. Attack traffic is forwarded by *Click* router where ASPM and DPM schemes are enabled. ASPM and DPM marked traffic is captured using Ethereal. This sniffer is connected to the segment # 2.

We have decided to implement DPM technique for two major reasons:

- DPM belongs to the same IP traceback class than ASPM approach.
- DPM mark packets at the interface closest to the source of the packet on the edge ingress router. In a similar way, ASPM is deployed only in a single router.

To fix the end of the simulation, we have defined some thresholds. The first threshold is the number of packets treated by the router before stopping the simulation. This threshold is applied to both schemes integrated into the Click router. For ASPM scheme, we apply this threshold to stop the simulation for a fixed *TempTbl* entry lifetime but we reiterate next for another selected value. The other threshold represents the minimal number of marked packets.

Attack scenarios Two attack scenarios are applied to measure the marking processing time. The first scenario (or scenario #1) is a scan port attack using a well known port scanner, called *Nmap*. This scenario is commonly accomplished by attackers at the reconnaissance phase. The second scenario (or scenario #2) is a denial of service attack simulated using Nessus DoS plug-in.

Simulation parameters Three input parameters are used to give information about attack traffic. They are the number of attack packets, number of SYN attack packets, and attack duration (Δt_{attack}). Some additional input parameters are needed by ASPM and DPM techniques. These parameters are: the marking interface IP address (used for ASPM and DPM) and *the TempTbl* entry Lifetime's (only used for ASPM).

The following output parameters are common for ASPM and DPM: The number of packets selected for marking and the marking processing time(s).

6.2 Analysis of results related to marking time processing

Simulation results have shown the impact of ASPM Selectivity and reducing load mechanisms on marking processing time. In fact, DPM marking processing time is much higher than ASPM. Figure 3 illustrates the difference in time between the two approaches for Scenario #1. The ASPM marking processing time is 4.53 times lower than the time spent by DPM for marking the same traffic (1700 packets).

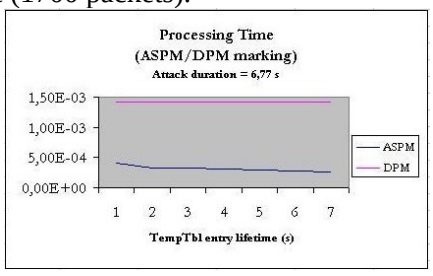

Fig.3 – Comparison of ASPM and DPM processing time for attack scenario #1

The difference between the two techniques becomes more important when the volume of attack traffic increases. For example, ASPM marking processing time is 3.10^1 times lower than the time given by DPM technique for an attack traffic containing 15000 packets. As it can be seen from Figure 4, the processing time depends on the specified lifetime value during the marking process.

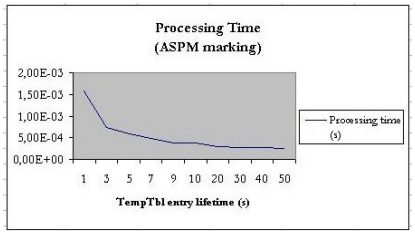

Fig. 4 – Impact of the lifetime value on the marking processing time

ASPM processing time is reduced compared to DPM due to the limited number of marked packets. The number of packets selected for marking becomes minimal (equal to 1) if the *TempTbl* entry lifetime is almost equal to the attack duration. Figure 5 illustrates the number of marked packets for some values chosen for the *TempTbl* entry lifetime (Scenario #2).

In fact, there is a relation between the number of marked packets and the *TempTbl* entry lifetime. Before detailing this relation, we define the following parameters:

- Δt_{attack} : Attack duration.
- $\Delta t_{TempTbl}$: *TempTbl* entry lifetime.
- n_{select}: Number of packets selected for marking.

To deduce the general formula relating n_{select} to Δt_{attack} and $\Delta t_{TempTbl}$, we can distinguish first the following two cases:

First **case:** if $\Delta t_{attack} < \Delta t_{TempTbl}$ then $n_{select} \leq 1$

Second **case:** if $\Delta t_{attack} \geq \Delta t_{TempTbl}$ then we have the following assertions :

$$\Delta t_{TempTbl} \leq \Delta t_{attack} < 2 \cdot \Delta t_{TempTbl} \Rightarrow n_{select} \leq 2$$
$$\Delta t_{TempTbl} \leq \Delta t_{attack} < 3 \cdot \Delta t_{TempTbl} \Rightarrow n_{select} \leq 3$$
$$\Delta t_{TempTbl} \leq \Delta t_{attack} < 4 \cdot \Delta t_{TempTbl} \Rightarrow n_{select} \leq 4$$

etc.

From these two cases we can deduce the following general formula, which determines the upper bound for the number of marked packets given the *TempTbl* entry lifetime and the attack duration:

$$\alpha \cdot \Delta t_{TempTbl} \leq \Delta t_{attack} < (\alpha + 1) \cdot \Delta t_{TempTbl} \Rightarrow n_{select} \leq \alpha + 1$$

where $\alpha \in N$

Typically, n_{select} may be equal to $\alpha + 1$ (the threshold), where there are a traffic that matches selection rules for each time interval, otherwise n_{select} will be less than this threshold. From this analysis, we should notice that it is interesting to assign a *TempTbl* entry lifetime that is close to the attack duration to insure that the number of marked packets is minimal.

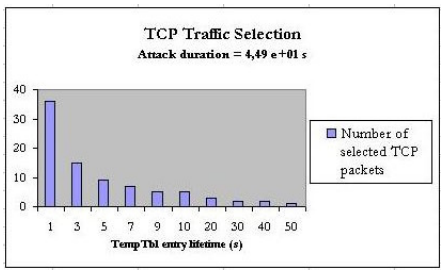

Fig.5 – Relation between n_{select} and $\Delta t_{TempTbl}$

7 Conclusion

In this paper, we have presented a novel IP traceback approach that is robust against some problems essentially related to processing overload introduced in routers and networks. ASPM technique has an adaptive behavior which changes according to the traffic protocol used.

To illustrate this new approach, we have described ASPM behavior for TCP protocol. In fact, the property exploited for TCP is related to the Three-Way TCP Handshake mechanism; the first packet initiating this process has only an IP and TCP headers so its size is minimal. Selecting SYN traffic and applying a reducing load mechanism have provided the opportunity to resolve many problems. Implementing and integrating the novel scheme in the Click modular router has offered a good experimentation of our scheme and has demonstrated its advantages compared to DPM scheme.

References:

[1] A. Belenky and N. Ansari, On IP Traceback, *IEEE Communications Magazine*, Vol. 41, No.7, 2003, pp. 142-153.

[2] E. Kohler, R. Morris, B. Chen, J. Jannotti, M.F. Kaashoek, The Click modular router, *ACM Transactions on Computer Systems*, Vol. 18, No. 3, 2000, pp. 263-297.

[3] A. Belenky, and N. Ansari, IP Traceback with deterministic Packet Marking, *IEEE communications letters*, Vol. 7, No. 4, 2003.

[4] M. Fomenkov, K. Keys, D. Moore and K. Claffy, Longitudinal study of Internet traffic in 1998-2003, tech. report, CAIDA, University of California, San Diego, http ://www.caida.org/.

[5] H. Krawczyk, M. Bellare, and R. Canetti, HMAC: Keyed-hashing for message authentication, Internet RFC 2104, February 1997, www.rfc-editor.org/rfc/rfc2104.txt.

[6] D. X. Song and A. Perrig, Advanced and Authenticated Marking Schemes for IP Traceback, *Proc. INFOCOM*, Vol. 2, 2001, pp. 878-86.

[7] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, Practical network support for IP traceback, *in Proc. Of the 2000 ACM SIGCOM Conference*, August 2000, Stockholm, Sweden.