A Denial-of-Service-Resistant IP Traceback Approach

Bao-Tung Wang and Henning Schulzrinne
Department of Computer Science, Columbia University, New York, NY 10027, U.S.A.

ABSTRACT
Identifying the origins of a Distributed Denial-of-Service (DDoS) attack is among the hardest research topics in the Internet security area. In this paper, we propose a multifunctional, secure, and DoS-resistant ICMP message scheme. This straightforward method allows victims not only to identify the sources of a DDoS attack but also to authenticate the ICMP Caddie messages received. Besides IP traceback, potential applications include packet dropping detection, feedback based routing, Traceroute, packet authentication and authorization, and confirmed anonymous communication. This paper describes the design of the *iCaddie*, and discusses its potentials.

1. INTRODUCTION

IP Traceback is a technique attempting to identify the origin of a specific IP packet. In the last few years, researchers has proposed various IP Traceback mechanisms, such as link testing [7], logging [3], overlay networking [4], probabilistic packet marking [1,2], and ICMP messaging [5,6]. In general, each approach has been designed to conquer some specific IP Traceback difficulties but also introduces new problems. For instance, the route reference approach [7] does not require ISPs to participant the traceback process but it, in fact, originates a new DoS attack on its own network. In addition, theoretically, the SPIE [3] can achieve the ultimate goal of the IP Traceback techniques, the single-packet IP Traceback. However, even though the storage requirement has been significantly reduced to 0.5% of the link capacity per unit of time, the overhead is still considerable, particular for routers at the core of the Internet. The CenterTrack [4] lessen the number of hops required for traceback but need to cooperate with other ISPs in order to continue the traceback across network boundaries. Although probabilistic packet marking methods successfully eliminate all bandwidth and storage overheads on network equipment, the path construction process has become too complex to accomplish accurately and timely. The ICMP Traceback (*iTrace*) [5,6] reduces the computation complexity but increase overall network traffic. Therefore, we consider the following properties for our IP Traceback scheme.

- *Incremental deployment*. Due to the cost and time required for upgrading network equipment, it is not practical to assume that most of them can be enhanced with new hardware or software in a short time.
- *Workload equilibrium*. Some network equipment, particular those at the core of the Internet, are time-sensitive and incapable of performing complex functionalities.
- *Security*. One of the most common problems of all proposed mechanisms is the mark or message authentication because they are relatively expensive.
- *Robustness*. Due to very limited available space in the IP packet header, the PPM breaks information into pieces, but that causes a very high rate of false positives for path construction.
- *Bandwidth overhead*. Except in-band IP Traceback measures [1,2], whether extra traffic load consumes conspicuous bandwidth is a critical issue.
- *Computation overhead*. For most IP Traceback methods, the most critical computation overhead is the attack path construction process, which needs to gather and assemble scattered information in considerable quantity of packets or messages received by the victim.
- *Storage overhead*. Even though the SPIE can reduce the storage requirement, the memory needed is still undesirable. Besides, the PPM and the *iTrace* also require a lot of space at the victim, which collect and store information for path construction.
- *DoS-resistance*. Although IP Traceback mechanisms are designed to defense DoS/DDoS attacks, most of them are suffering DoS/DDoS attacks as well, because they do not prevent information from tempting with as well as need to consume significant resources even when no attack is involved [8].

2. ICMP CADDIE MESSAGES

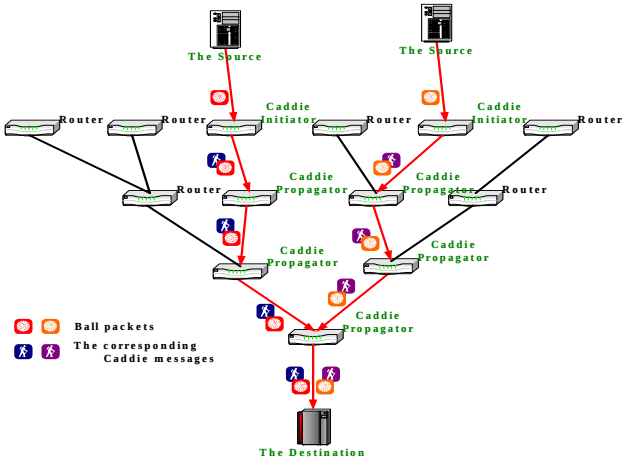


Figure 1: The *iCaddie* Scheme

The ICMP Caddie Messages (*iCaddie*) is a newly proposed ICMP Messages scheme. A *Caddie message* is an extra ICMP message generated by a router or an application, called a *Caddie initiator*, attached with the entire packet routing history of one randomly selected packet, called a *Ball packet*, forwarded by the router. In other words, while a router is forwarding packets, it randomly selects one of the packets as a ball packet, and then generates a Caddie message following the ball packet. The Caddie message will collect the path information about the sequence of the

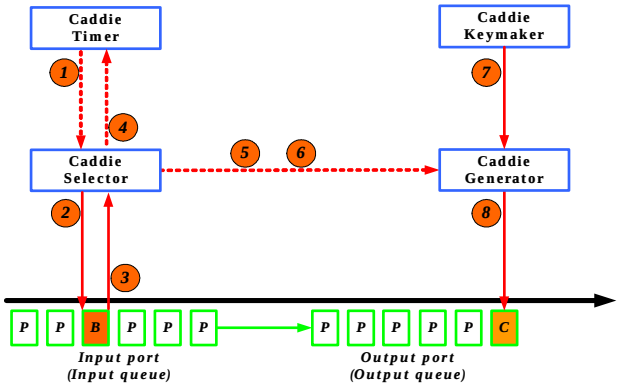
router's identities along the way toward the ball packet's destination.

2.1 Model

A router plays the role of Caddie initiator when it generates Caddie messages, and later it may act as *Caddie propagator* when it forwards Caddie messages for other Caddie initiators (see Fig.1.)

2.2 Caddie Message Generation

Generally speaking, a featured router randomly select a packet, as a ball packet, with a probability of about 1 out of 20K, and then it generates and emits an ICMP Caddie message associated with this ball packet. However, in order to reduce the number of Caddie messages produced, each input port of a router maintains a simple Caddie timer (or a Caddie counter), which indicates how long this port has not received any Caddie message, regardless of its source or destination. If the amount exceeds a specified threshold, the router will start to act as a Caddie initiator (see Fig.2.) Accordingly, a router should not generate any Caddie message unless it has not received any Caddie message from its upstream routers for a certain amount of time.



B. The ball packet
C. The Caddie message
P. Regular packets

1. *Trigger ball packet selection*
2. *Select a ball packet*
3. *Extract the ball packet*
4. *Update the Caddie timer*
5. *Copy the ball packet header*
6. *Trigger Caddie generation*
7. *Generate a session key*
8. *Inject Caddie message in the front of output port*

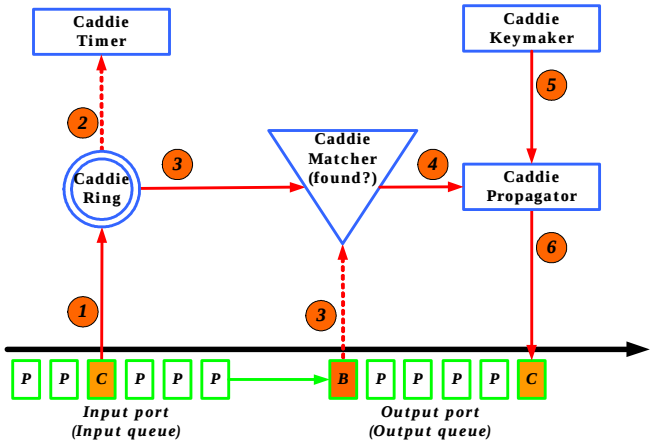
Figure 2: Caddie Message Generation Components

2.3 Caddie Message Matching

The Caddie ring is a circular linked list storing Caddie messages recognized in the routers' input ports; the Caddie matcher mates Caddie messages with their corresponding ball packets. Before overwriting an unmated Caddie message, the router will put the caddie message into the output port based on the hash value of IP address of the destination. The DIGEST field in the header of a Caddie message works as the Caddie ID, which is computed from the invariant portion of the IP header of the ball packet. In other words, while a Caddie message was recognized in an input port, it will be stored in the Caddie ring, and then the Caddie matcher starts to compare it with packets in the output port. If its ball packet is found, the Caddie message will be inserted into the front of the output port (see Fig.3.)

2.4 Caddie Message Propagation

While a router receives a Caddie message and the identified corresponding ball packet, it appends a new element into the ROUTER LIST (RL) field, including the router's IP address, the interface on which the ball packet arrived, and the next hop of the ball packet based on the routing table. After filling the TIMESTAMP (TS) field based on the current time, the router will compute the HMAC, a mechanism for message authentication using cryptographic hash functions, with its current secret key in order to prevent this RL element from being tampered with. Because the TSs are only used for TRKC (discussed in section 2.6) and compared with TSs set by the same router, it is not really matter that clocks on different routers are not accurately synchronized.



B. The ball packet
C. The Caddie message
P. Regular packets

1. *Store Caddie messages in the Caddie ring*
2. *Update the Caddie timer*
3. *Match each Caddie message with its ball packet*
4. *Trigger Caddie propagation*
5. *Generate a session key*
6. *Update and inject Caddie message in the front of output port*

Figure 3: Caddie Message Propagation Components

2.5 Caddie Message Transformation

The DIGEST in a Caddie message header will be updated when the corresponding ball packet is changed. Accordingly, the DIGEST field can always uniquely identify the ball packet and enable the identification across hops in the forward path. Furthermore, because the DIGEST is signed by each router in the RL, even though a compromised router attempts to tamper with the DIGEST field in order to mismatch a ball packet with its Caddie message, the destination still can easily discover the malicious attempt and its identity.

2.6 Time-Released Key Chain

To protect the content inscribed in each Caddie message, the *iCaddie* employs the Time-Released Key Chain (TRKC) [2] to verify the authenticity of a Caddie message and its each RL element. To generate a sequence of secret keys (session keys), each router successively applies a one-way hash function to a randomly selected seed, and then each router reveals the key after a delay at the end of each time interval. So, the destination of a Caddie message can retrieve the newest key, and then compute all the secret keys for previous time intervals to compute and verify the HMACs for every RL element in the Caddie message.

3. DISCUSSION

We have selected following parameters for evaluation.

- *The scalability*. The configuration of the *iCaddie* is totally independent of each other. So, it's scalable.
- *The capability of incremental deployment*. Non-participating routers simply forward Caddie messages as normal ICMP messages based on the destination IP addresses.
- *The workload distribution of participant routers*. The workload at the core routers is relatively lower than that at the edge routers, because they have more chance to receive ICMP Caddie messages from edge routers, and then simply update and propagate the messages, instead of generating new ICMP messages.
- *The number of attack packets required for IP Traceback*. It's thousands, based on the Caddie message generation rate of Caddie initiators.
- *The number of ICMP message generated for IP Traceback*. The number of Caddie messages generated depends on the number of attack sources but not the length of the attack path.
- *The robustness in case of a DDoS attack*. The ICMP Caddie messages scheme produces few false positive because the probability that two packets forwarded by one router in a very short time with the same packet digest is small.
- *The computation overhead of an attack path construction process*. Because all the Caddie messages method are required only to count and append new paths onto the traffic source tree; the complexity increases almost linearly.
- *The security of the marks or messages*. The authentication of the initialization of a Caddie message is depended on the HMAC of the first element of the ROUTER LIST field, which is inscribed by the Caddie initiator, and each element of the ROUTER LIST field is authenticated by the corresponding HMAC field, protected by secret keys of each Caddie propagator.

4. APPLICATIONS

The *iCaddie* is a general protocol with wide applicability.

- *IP Traceback*. A victim can construct a traffic source tree structure by simply composing the Caddie travel paths inscribed in the Caddie messages received.
- *Packet dropping detection*. Based on the principle of packet flow conservation, an victim can deduce the offensive routers by using a revised datacube algorithm.
- *Feedback based routing* [9]. Based on the network topology obtained by Caddie messages, routing decisions can be made by edge routers.
- *Traceroute*. A host can send a dummy packet and then create a Caddie message following the dummy packet.
- *Packet authentication/authorization/QoS*. Any packet can be associated with one Caddie message, on to which is attached the needed information.
- *Confirmed anonymous communication*. Caddie initiators intentionally hide the original source address by replacing it with the Caddie initiator's IP address and sign the Caddie message as a certificate.

5. CONCLUSION

The Internet has changed a lot from its early days, but its essential protocols and standards has not revised significantly, because of the expensive and tedious deployment processes. In this paper, we have proposed a multifunctional ICMP message scheme that is capable of addressing several important network security issues, such as IP Traceback, routing protocol attacks, and packet dropping attacks. In addition, the method also support some other value-added network services, e.g. secure Traceroute, packet authentication and authorization, as well as confirmed anonymous communication. Therefore, the method may be worth to pay off the considerable deployment cost.

6. REFERENCES

[1] S. Savage, David Wetherall, Anna Karlin and Tom Anderson, "Network Support for IP Traceback," *IEEE/ACM Trans. on Networking*, June 2001.

[2] D.X. Song, A. Perrig, "Advanced and authenticated marking schemes for IP traceback," *Proc. IEEE INFOCOM*, 2001.

[3] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, S.T. Kent, W.T. Strayer. "Single-packet IP traceback," *IEEE/ACM Trans. on Networking*, Dec 2002.

[4] R. Stone. "CenterTrack: An IP Overlay Network for Tracking DoS Floods," *Proc. USENIX Security Symposium*, Jul. 2000.

[5] S.M. Bellovin. "ICMP traceback messages", Internet draft ftp://ftp.ietf.org/internet-drafts/draft-ietf-itrace-04.txt. Feb. 2003.

[6] S.F. Wu, L. Zhang, D. Massey, A. Mankin. "On design and evaluation of "intention-driven ICMP traceback," *Proc. Computer Communications and Networks*, 2001.

[7] H. Burch and B. Cheswick. "Tracing Anonymous Packets to Their Approximate Source," *Proc. LISA 2000*, Dec. 2000.

[8] B. Wang, H. Schularinne, "Analysis of Denial of Service Attacks on IP Traceback Techniques", *IEEE GLOBECOM*, Dec. 2003.

[9] D. Zhu, M. Gritter, D.R. Cheriton, "Feedback Based Routing," *ACM SIGCOMM Computer Communication Review*, Vol. 33, No. 1, Jan. 2003.