Defeating Reflector Based Denial-of-Service Attacks using Single Packet Filters

Ashok Singh Sairam
ashok@iitp.ac.in
Dept. of Computer Science and Engineering
Indian Institute of Technology Patna

Late Ashish Subramaniam, Gautam Barua
gb@iitg.ernet.in
Dept. of Computer Science and Engineering
Indian Institute of Technology Guwahati

***Abstract*—Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks are becoming increasingly sophisticated with few practical solutions available. In this paper we consider the issue of filtering reflector based DoS attacks and of identifying attackers. For reflector based attacks, a Signature Conflict Triggered Filtering (SCTF) ([4]) scheme based on Deterministic Edge Router Marking (DERM) ([8]) was proposed. We suggest an enhancement to make the 3-way handshake in SCTF stateless and call it Fast-SCTF. We then propose a framework using BGP for a single-packet handshake. We demonstrate that our proposed scheme is space efficient, more secure, robust and it requires very little cooperation among autonomous systems.**

I. Introduction

Identifying the authentic source of an attack is crucial to institute protection measures against Denial-of-Service (DoS) attacks. The problem of finding the source of a packet is called the IP traceback problem ([9]). A deterministic packet marking scheme, DERM was proposed in [8] to filter attack packets at the victim and also to identify attackers. The scheme fails to identify attackers if attacks are reflected through attackers. In [4], a technique (SCTF) was proposed to filter attack packets upstream at the edge router of reflectors. Through a 3-way handshake, the victim informs upstream routers of attack packets. These routers then filter such attack packets and can also identify the routers from which attacks took place. In this paper we first propose an improvement of this technique (Fast SCTF). We then propose a single packet information scheme which is more efficient in informing upstream routers in the face of attacks.

The paper is organized as follows. In section II we review DERM and its variants. In section III we discuss in detail SCTF and Fast-SCTF. In section IV we detail our proposal for the single packet, authenticated filter propagation and an analysis of the proposal is given in V. A discussion on related work is included in section VI. Conclusions are given in section VII.

II. Tracing DoS attacks using Packet Marking

In [8] an effective packet marking technique called DERM was introduced for defense against DoS attacks. Each packet that enters a network is marked by an edge ingress router. In basic DERM, the scheme is to insert a 16-bit hash of the IP address of the edge router on each incoming packet (in the ID field used to identify fragments). The victim handles an attack in two phases - a filtering phase and an identifying phase. To filter packets from the attacker, the victim maintains a table, each entry of which consists of 3-tuples: a possible hash mark (also called as HashMark or signature), a *RECV* bit and the list of all ingress router addresses that have this HashMark. When an attack packet is identified, the victim notes its HashMark and the *RECV* bit in the corresponding table entry is set. The filtering process simply consists of checking whether the *RECV* bit corresponding to the HashMark of an arriving packet is set and to drop that packet if so. The list of all ingress routers that have the HashMark are identified as attackers.

Note that if the number of ingress routers is greater than 2^{16} there will be more than one ingress address corresponding to each HashMark. Thus HashMarks of other legitimate users might collide with the attacker's HashMark, which will result in dropping of their packets (collateral damage) during the filtering phase or being falsely identified as an attacker. In order to improve attacker identification of basic DERM, multiple hash DERM was proposed. In multiple hash DERM, the 16 bit ID field accommodates two components instead of one, a d bit HashMark and a log(f) bit hash function identifier, where f is the number of hash functions used. On receiving a packet, the ingress router randomly selects one of the HashMarks with equal probability. The HashMark and the hash function identifier are then inserted into the packet. The victims instead of maintaining a single table will now have f tables corresponding to each of the f HashMarks. As before whenever a packet is identified as an attack packet, the hash function identifier is noted and the *RECV* bit is set in the corresponding table. To identify an attacker, the victim checks that the *RECV* bit corresponding to the HashMark of the attacker is set in all the f tables. These additional checks decrease the number of false positives. Another variant of DERM called the skewed multiple hash DERM ([4]) was proposed to reduce the collateral damage during filtering. In skewed multiple hash DERM, multiple hash functions are used but with unequal probabilities.

III. Distribution Reflection Denial of Service Attacks (DRDoS)

In a direct DDoS attack, the attacker uses a number of zombies (slaves or agents) to flood the victim's network links with attack packets. Another technique used by attackers is to use reflectors to flood the victims network ([10]). Using reflectors in a DDoS attack is called a DRDoS attack. Not only

CHINACOM 2010, August 25-27, Beijing, China
Copyright © 2011 ICST 973-963-9799-97-4
DOI 10.4108/chinacom.2010.86

do reflectors hide the identity of slaves, but they may be used to multiply the attack packets. Techniques like DERM and its variants effectively drop packets that have certain HashMarks (signature) making it a victim-side defense. But attack packets reach the victim and so the network gets choked. What is required is that edge routers at the source of attack packets be informed of the HashMarks of attack packets. This has to be done securely as messages to edge routers from victims can themselves be spoofed and constitute an attack on edge routers. The SCTF protocol ([4]) and its improvement suggested here, provide such techniques. They also enable the detection of the source of reflector attacks.

A. Signature Conflict Triggered Filtering (SCTF)

The SCTF proposal installs filter rules at the source edge router (which could either be a zombie in a direct attack or a reflector) so as to mitigate DRDoS attacks before they reach the victim's network. What is described below is for reflector attacks. Direct attacks are handled similarly. When the victim's intrusion detection system informs it of a DRDoS attack, the victim initiates a 3-way handshake with the remote router (the reflector's edge router, RER, in short). At the end of the SCTF handshake, the RER will have the IP address of the victim and the HashMark of the victim's ingress edge router. The two together constitutes the signature of the victim which is maintained by the RER in a Conflict Detection Table (CDT). In figure 1, we illustrate how this signature is used by the RER to filter attack packets. Let a zombie (Z) send an attack packet with the source address spoofed with that of the victim's address (IP_V), and the destination address that of the reflector. As the packet traverses, the zombie's edge router will add its HashMark (IP_{ZR}) to the packet as per DERM. When the packet arrives at RER, it will look up the CDT and find that the edge router's hash of the claimed source (victim) differs from the hash stored in the table. The RER realizes that the packet has been spoofed and hence must be dropped. The RER forwards a packet to the reflector if and only if either the packet's source IP is not present in the CDT or the source IP and its hash match an entry in the CDT.

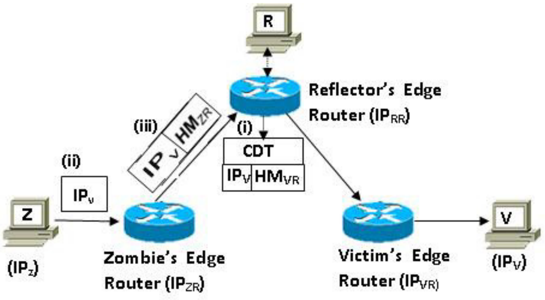

Fig. 1. Filtering attack packets in SCTF.

B. The SCTF Handshake

In the SCTF protocol, a 3-way handshake is used for authenticated information to be propagated to remote routers. In this handshake, a request packet (VERIFY_SIGNATURE) originates from the victim directed towards the RER. The request packet consists of the IP address of the victim and the HashMark of its edge router. The victim also enters this information into a REFLECTOR table. To verify that the request packet was not spoofed, the RER sends a challenge (VERIFY_SOURCE) back to the requester. The challenge packet contains a random number. The RER also makes an entry into a VICTIM table. The victim responds to the challenge by sending a SOURCE_RESPONSE packet to the reflector with the same random number, but only after ensuring that the IP address of the RER exists in its REFLECTOR table. When the RER receives a SOURCE_RESPONSE packet, it checks the VICTIM table and accordingly updates its Conflict Detection Table.

C. The Fast SCTF Handshake

We propose an enhancement to the SCTF handshake (the Fast-SCTF scheme) to make it stateless and faster. In the VERIFY_SIGNATURE packet sent by the victim, in addition to the IP address of the victim and the HashMark of its edge router, a random number RAND_V is also included. The random number is a keyed hash of the IP address of the RER. The key is known only to the victim. In the VERIFY_SOURCE packet sent by the RER, RAND_V is echoed along with its own random number RAND_R. This random number is a keyed hash of the IP address of the victim, and the key is known only to the RER. RAND_R is echoed by the victim in the SOURCE_RESPONSE packet. The victim can verify the VERIFY_SOURCE packet by a single hash computation obviating the need for the REFLECTOR table. The RER can also verify the authenticity of a request on the fly by a single hash computation.

In both the SCTF and Fast-SCTF handshakes, the authentication is weak because any on-path network entities that can eavesdrop on the handshake can provide false claims of identity of either party. A stronger and more secure authentication could use the public key infrastructure (PKI), but the overheads will be much higher. However, as the packet will be traveling only through routers, eavesdropping is unlikely. The other and more serious drawback of the 3-way SCTF handshake is that it cannot be guaranteed to be successfully completed in the face of a DRDoS flood attack, when the victim's bottleneck link will be clogged. During a DRDoS flood attack the link from the victim to the rest of the world will not be choked but the link to the victim (incoming bandwidth) will be heavily choked. This means response packets sent by the RER to a request posed by the victim is unlikely to reach the victim. The handshake packet will suffer the same loss as other legitimate packets when the victim's bottleneck link is clogged during a DRDoS attack. Thus the 3-way handshake will suffer from incomplete handshakes. One way to solve this problem of incomplete handshakes is that a remote router on receiving the first packet of the 3-way handshake temporarily blocks any traffic directed to the source. The remote router then queues its response to the handshake ahead of other packets, thereby prioritizing the completion of the handshake. But a local prioritization by a router still will not guarantee delivery to the source because of a network bottleneck somewhere downstream. We need to globally prioritize completion of the 3-way handshake but this will require cooperation between different networks and is not a practical proposition.

IV. Single-Packet Traceback Scheme

A. Design Overview

In this section we propose a single-packet traceback protocol. The idea is to replace the 3-way SCTF handshake with a single-packet handshake, so as to obviate the problem of incomplete handshakes. The authentication scheme must uniquely authenticate a sender, it should be safe from eavesdropping and it should be robust. The basic functionality of the Single-packet Traceback scheme is exactly similar to the SCTF protocol except for the way the initial handshake is performed. As the path from the victim toward the attacker is un-congested, the victim can successfully inform upstream routers in the path to filter all packets from an attacker. In general, the enhanced single-packet protocol will not only mitigate DRDoS attacks, but can be provisioned to handle generic requests from nodes. Thus, it can be used for direct DoS attacks too.

B. The Handshake Packet

In this work, we propose to use hash chains for authenticated propagation of filter requests. A significant characteristic of hash chains is that they are computationally inexpensive which makes them suitable for online schemes. Leslie Lamport ([3]) originally advocated the use of hash chains as an authentication mechanism in an insecure communication channel.

In this authentication scheme, a user (potential victim) picks a random number *x*, and a hash function *H* is used to compute a hash chain of length n, i.e $h_n = H^n(x)$ where H^n denotes n successive applications of *H*. h_n, is called the anchor of the hash chain (also referred to as secret in this paper) is made available to the remote routers or edge routers. The user authenticates by sending values of the hash chain in reverse order (that is h_{i+1} is released before h_i). The receiver needs to perform a few hash computations at most to identify the user. To prevent attacks against the hash chain, *H* is assumed to be a cryptographic hash function providing pre-image resistance and second pre-image collision resistance. The anchors are released through the distribution service described in the next section, which are then stored by routers for future processing. Filter request packets from a victim will include values of the hash chain in reverse order as authenticator. The content of a handshake packet will thus consist of a hash value (authenticator) and signature of the victim. For example, the first single-packet handshake will consist of $< h_{n-1}, IP_V, HM_{VR} >$, where h_{n-1} is a hash chain value (authenticator) that has not yet been released by the victim. IP_V and HM_{VR} are the IP addresses of the victim and HashMark of the victim's edge router respectively. The remote router computes a hash of the authenticator h_{n-1}. If the computed value matches the hash chain anchor h_n which was earlier released by the victim, the request is authenticated, else it is discarded. To disallow replay attacks, the remote router after successfully authenticating the initiator will replace the stored hash value h_n with h_{n-1}. The next handshake packet from the victim must include h_{n-2} as the authenticator. Updating the stored hash chain anchor at the receiver end after a successful authentication ensures that the receiver needs fewer hash computations to authenticate the sender in the next round of authentication. Filtering of packets will be exactly in the same way as in the SCTF protocol. The remote router will store the signature of the victim in a table (CDT) to filter malicious packets forwarded towards the victim.

Although our proposed authentication scheme use a single-packet handshake, the use of hash chains makes our scheme more secure and robust. An eavesdropper will neither be able to re-transmit the same hash nor generate a previous value of the hash to impersonate the victim. Since the receiver expects a distinct hash with every handshake packet, replay attacks will not be possible. An eavesdropper will not be able to generate a hash value by looking at already known hash values as these would contradict the one-way property of hash functions. Hash chains also have an important property of being robust to a system and its authenticating user getting out of synchrony ([3]). A remote router and a victim may get out of synchrony due to system crashes or packet loss. In the single-packet filter model, a victim sends a filter request then waits for a certain time interval (say round-trip time) for the request to transmit to the RER. After the filter request is successfully transmitted, no attack packets should be directed from the reflector towards the victim. However, if the victim still receives attack packets from the reflector, it will mean the filter packet transmission was unsuccessful (lost or garbled). The victim will resend the filter packet with a new hash value (next from the pre-determined sequence). The remote router can verify the authenticity of the packet by repeatedly applying the hash function. For example, if the remote router has a stored hash value of h_{99}, it would expect a hash value h_{98} in the next filter request from the sender. However, if the filter packet containing the authenticator h_{98} is lost on transit, the victim will include h_{97} as the authenticator in the next filter request it resends. The remote router can verify the authenticity of the victim by applying the hash function twice(h_{99} =$H^2(h_{97})$).

C. Architecture

To limit the amount of storage required by both the sender and receiver, we explored the possibility of aggregating the authenticator for a given IP-prefix. In the remaining part of the paper we assume that the authenticator is common to an autonomous system (AS) and the nodes of the AS themselves are not aware of the state of the shared secret. The task of sending request packets are outsourced from the victim to an entity we term as VICTIM_GATEWAY (VG). This is essential because the filter request needs to carry an authenticator based on a secret that is common to the IP-prefix encompassing the victim or the AS housing the victim. The victim is not aware of the secret and so we need a third party to perform the filter propagation. This is a significant aspect in which this protocol differs from the SCTF proposal.

In an AS there will be at least one BGP speaking router (VG) that is used to propagate reachability information to other BGP peers. We assume that during the first time a BGP router advertises its route, along with the route updates a shared secret (anchor of a hash chain) for the entire AS is also advertised. Since hash chains can be of arbitrary length and yet be stored efficiently, hash-chain anchors are released infrequently, for example once in a year. This means the secret needs to be advertised only once at the beginning of the BGP session. Subsequent route updates between the BGP peers will be as usual and no shared secret needs to be included. The BGP router generates a separate secret for each of its neighbour AS. The shared secret between any two AS is unique. Thus once an AS comes up and has finished announcing its reachability information to all its neighbouring AS, it will also have shared an unique secret with each of the neighbours.

D. Dispatching Filter Request

In order to initiate the process of sending a filter request, an attack packet must be identified. We assume that an intrusion detection system (IDS) determines that an exceptional event has occurred. Upon receipt of an alarm from the IDS, the victim informs the VICTIM_GATEWAY (VG) to dispatch a filter request on its behalf to the attacker's edge router. The propagation of the filter request takes place in three phases. In the first phase the victim initiates the filtering process and informs the VG. This occurs via the fast 3-way SCTF handshake mechanism. In the second phase the VG propagates the filter requests to the ATTACKING_GATEWAY (AG). This occurs via the speedy single-packet mechanism. In the third phase the AG propagates the request to the edge router (ER) of the reflector (also referred to as the reflecting edge router) using again the fast 3-way SCTF handshake. The steps involved in the propagation of a filter request is illustrated in figure 2. We argue that the 3-way handshakes will not suffer from incomplete handshakes when used within a domain. In the first phase, VG upon receiving the first packet of the 3-way handshake from the victim (V) queues the return response to V ahead of any other packets flowing towards V. That is, VG prioritizes completion of the 3-way handshake. If there are intermediate routers in between VG and V they can also be asked to prioritize the response. In fact, if the routers do not have enough buffer space they can temporarily drop any packets directed to V. Local prioritization of response will work in this case since both the sender and the receiver as well as the intermediate routers are under the control of a single administrative domain. Using the same argument, the 3-way handshake between AG and the edge router in the third phase will also be successfully completed. The routers can store the shared secret in a separate table or along with the routing information in the routing table.

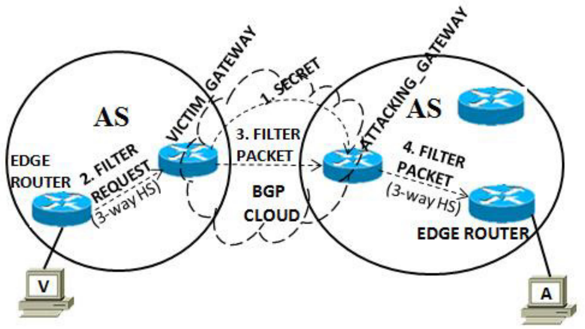

Fig. 2. Steps involved in dispatching a filter request. (1) Include a shared secret along with the BGP route advertisement (2) Victim initiates a 3-way handshake with VG to propagate filter request to ER (or RER) (3)VG forwards the request to the AG using a single-packet filter (4) AG finally propagates the filter request to the ER (or RER) using a 3-way handshake

E. Cumulative Authentication

So far we have outlined steps to propagate a filter request where the victim and attacker belong to neighbouring autonomous systems. The question now is how VG will send a filter request to an attacker which does not reside in its immediate neighbour AS but further down? In such a case the ASes encompassing the VG and the AG will not share a common secret, hence it will not be possible to directly send the filter request. We propose to use a chain of single-handshake packets between neighbouring ASes to dispatch the filter request. The idea is similar to the way a chain of secure neighbour-to-neighbour communication ([6]) is used to propagate route updates securely. Upon receiving a filter request packet, if a VG finds that the packet is not intended for an edge router in its domain, the packet is forwarded. This first VG will look up its routing table and accordingly forward the request packet to a second VG in its neighbouring AS. However, before the packet is forwarded, the shared secret in the filter request packet inserted by the sender is replaced with the common secret the first VG shares with the neighbour AS. The second VG authenticates the filter packet, updates the state of its shared secret and proceeds to forward the packet until the intended destination VG is reached. The destination VG will finally forward the filter request to the intended edge router via the 3-way handshake.

V. Analysis of the Single-Packet Filtering Model

In this section we characterize the performance of our scheme in terms of accuracy in identifying the attacking reflectors router (false positives), storage requirements and overhead of looking up tables. The possible attacks against this scheme will be similar to the ones applicable to the SCTF proposal. Let M be the number of ingress edge routers, N the number of attackers, d the number of marking bits used and f the number of multiple hash functions. The average number of ingress routers (M_{av}) corresponding to a particular HashMark will be given as, $M_{av} = \frac{M}{2^d}$. Let E(HM) be the expected number of HashMarks in a particular table with RECV bit set to 1.

A. False positives

Assuming that the skew multiple hash DERM is used, the number of false positives will be $E(HM) * M_{av} - N$ ([4]). This means in the single-packet filtering model, a victim will send filter request packets to the attacking router as well as to some innocent edge routers. The sending of packets by the victim to innocent edge routers has a side effect of an entry in their CDTs. However, such entries will not lead to dropping of packets. The *signature* of legitimate traffic will not match the entries stored in the CDT and so no packets will be filtered. Thus, the false positives generated during the identification of the reflecting edge router will have no noticeable effect.

B. Storage Requirements

In the single packet filtering model, an AS will have to store a shared secret (anchor of a hash chain) with each of its neighbouring ASes. Assuming that we use a collision-resistant hash function like SHA-2 ([5]), the size of the anchor will be at the most 512 bits. Let an AS *P* have G_p neighbours. The storage requirement of the AS will be, $G_p * 512\ bits$, which is minor.

C. Filtering Overhead

In the reflecting edge routers, a CDT lookup is required before forwarding any packet,. Each entry in the CDT consists of the IP address of the victim and HashMark of the victim's edge router (of length d, which is typically 12 bits). That is, each entry in the CDT will be (32+d) bits. Let r be the number of simultaneous attack packets that uses a particular reflector. The instantaneous size of the CDT will be r*(32+d) bits. If the source IP address matches an entry in the CDT but the source edge router HashMark does not, the packet is marked as an attack packet. The attacking router's hash and the victim's IP address need to be logged in an ATTACKING_ROUTERS table (no duplicates). The storage requirement of this table will be N*(32+d) bits where N is the number of attackers.

VI. Related Work

Several types of DDoS attacks exist, the most basic being host-based DDoS attacks which can be easily traced and managed. Network-based DDoS attacks exploit the weakness of the TCP/IP protocol suite and hence are more difficult to traceback ([9]). A detailed evaluation of the different IP traceback techniques is given in [1]. Park and Lee ([7]) proposed to install distributed packet filters on AS over the Internet to filter packets from *unexpected* links. Their idea was essentially to extend the ingress packet filtering approach ([2]) to the Internet core. For both political and technical reasons, ingress filtering cannot be widely deployed. Moreover, this approach requires BGP messages to carry source information which would significantly increase the BGP message size. Snoeren et. al. ([10]) proposed an auditing technique called Source Path Isolation Engine (SPIE) to traceback individual packets. A serious limitation in SPIE is that traceback must be performed within a specified short-period of time. In our scheme there are no such limitations. The attack packets can be handled immediately while the traceback can be done after the attack has subsided. No techniques have been reported in the literature to help identify reflector attackers, especially when it is a spoofing based reflector attack. SCTF, and the improved protocol presented in this paper provide solutions.

VII. Conclusions

Use of deterministic packet marking strategies can help in identifying attack packets once an attack is detected by an intrusion detection scheme. Marks are placed in IP packets, typically in the ID fields. To help prevent attack packets from reaching a victim, the edge routers from where the packets are coming have to be informed. This issue assumes particular significance in the case of reflector based attacks. Without information from a victim, an edge gateway can only identify the reflector, and cannot filter packets sent to the reflector. The challenge is to securely inform edge routers in the midst of an attack where the large volume of traffic to the victim can prevent two way communication between a victim and an edge router. This paper proposes a single packet mechanism of sending such information from a victim to edge routers. The proposal is built on top of our earlier work which had proposed a deterministic edge marking scheme called DERM. The mechanism proposes extensions to the BGP protocol to enable BGP routers to exchange information securely and efficiently. Hash chains are used for the purpose of providing authentication.

References

[1] A. Belenky and N. Ansari. On IP Traceback. *IEEE Comm.*, 41(7):142–153, 2003.

[2] Ferguson, P. and Senie, D. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. In *RFC 2827*, May 2000.

[3] Leslie Lamport. Password Authentication with Insecure Communication. *Commun. ACM*, 24(11):770–772, 1981.

[4] Mittal, Prateek and Barua, Gautam and Narang, Sameer. Defeating Reflector Attacks: Signature Conflict Triggered Filtering. *Proceedings Fifth European Conference on Information Warfare and Security (ECIW 2006)*, June 2006.

[5] National Institute of Standards and Technology. FIPS PUB 180-3: Secure hash standard. *Gaithersburg, MD, USA, NIST*, October 2008.

[6] F. Palmieri. A Scalable PKI for Secure Routing in the Internet. *Computational Science and Its Applications ICCSA 2004*, 3043/2004:882–894, 2004.

[7] Kihong Park and Heejo Lee. On the Effectiveness of Route-based Packet Filtering for Distributed DoS attack Prevention in Power-law Internets. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 15–26, New York, NY, USA, 2001. ACM.

[8] Rayanchu, Sharavan K. and Barua, Gautam. Tracing Attackers with Deterministic Edge Router Marking (DERM). *International Conference on Distributed Computing and Internet Technology*, 3347:400–409, December 2005.

[9] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical Network Support for IP Traceback. *SIGCOMM Comput. Commun. Rev.*, 30(4):295–306, 2000.

[10] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Beverly Schwartz, Stephen T. Kent, and W. Timothy Strayer. Single-Packet IP Traceback. *IEEE/ACM Trans. Netw.*, 10(6):721–734, 2002.