Computers & Security (2005) **24**, 571—586

Computers & Security

www.elsevier.com/locate/cose

Defending against spoofed DDoS attacks with path fingerprint☆

Fu-Yuan Lee *, **Shiuhpyng Shieh**

Department of Computer Science and Information Engineering,
National Chiao Tung University, Hsinchu 30010, Taiwan

Received 8 September 2004; revised 13 March 2005; accepted 28 March 2005

KEYWORDS
Network security;
Intrusion detection;
DDoS;
IP spoofing

Abstract In this paper, we propose a new scheme, called ANTID, for detecting and filtering DDoS attacks which use spoofed packets to circumvent the conventional intrusion detection schemes. The proposed anti-DDoS scheme intends to complement, rather than replace conventional schemes. By embedding in each IP packet a unique *path fingerprint* that represents the route an IP packet has traversed, ANTID is able to distinguish IP packets that traverse different Internet paths. In ANTID, a server maintains for each of its communicating clients the mapping from the client's IP address to the corresponding path fingerprint. The construction and renewal of these mappings is performed in an on-demand fashion that helps to reduce the cost of maintenance. With presence of the mapping table, the onset of a spoofed DDoS attack can be detected by observing a surge of spoofed packets. Consequently, spoofed attack packets are filtered so as to sustain the quality of protected Internet services. ANTID is lightweight, robust, and incrementally deployable. Our experiment results showed that the proposed scheme can detect 99.95% spoofed IP packets and can discard them with little collateral damage to legitimate clients. It also showed that the higher the aggregated attack rate is, the sooner the attack can be detected.

Introduction

Distributed Denial-of-Service (DDoS) attacks pose a major threat to the availability of Internet services. A DDoS attacker can greatly reduce the quality of a target Internet service or even can completely break the network connectivity of a server by persistently overloading critical network or system resources of the target, such as network bandwidth, router processing capability, or CPU/memory at the target machine. Generally, to achieve resource overloading, a DDoS attacker will first compromise a large number of hosts and

☆ This work is supported in part by III and NSC of Taiwan.

* Corresponding author.
E-mail address: leefy@csie.nctu.edu.tw (F.-Y. Lee).
URL: http://www.csie.nctu.edu.tw/~leefy.

0167-4048/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2005.03.005

subsequently instructs these compromised hosts to attack the service by exhausting a target resource. Due to the lack of built-in security mechanisms in the current Internet infrastructure, conducting a DDoS attack is easy. An attacker can easily get access to a large number of insecure computers with exploit/attack programs, such as Trinoo, TFN and TFN2k (CERT Coordination Center, 1999a,b,c, 2000; Dittrich, 1999a,b). On the other hand, defending against DDoS attacks is extremely difficult because there is usually no explicit attack pattern to distinguish legitimate packets from malicious ones. Moreover, to hide the sources of attack traffic and circumvent DDoS defense mechanisms relying on inspecting IP header fields, DDoS attack programs generally fill IP header fields, especially the 32-bit source IP address, with randomized values. This IP spoofing technique has made the detection and filtering of DDoS traffic extremely difficult, and it has become a common feature of the many DDoS attack tools.

To design an effective and feasible DDoS countermeasure, there are several requirements a DDoS defense mechanism should meet. These requirements are listed as follows.

- *Discrimination*: The ability of discriminating legitimate packets from attacking packets is considered the first fundamental requirement of a DDoS defense mechanism. In fact, this is a very challenging problem. Specifically, since most of current DDoS attack tools generate spoofed IP packets, the ability to identify spoofed IP packets becomes a key in defending against DDoS attacks in which spoofed IP packets dominate a significant share of attack traffic.
- *Lightweight*: The defense of DDoS mechanism should not impose substantial load on both the Internet routing infrastructure and the victim. It is clear that heavy load imposed on Internet core routers will seriously affect the throughput of these routers. For instance, complex packet filtering operations should not be involved on the path of packet forwarding on core routers. Moreover, the load on victims should also be lightweight. Otherwise, the defense mechanism itself will become vulnerable to DDoS attacks.
- *Loose cooperations*: The defending scheme should avoid the assumption that tight cooperation is required among ISPs. This is because cooperation normally requires complex coordinations among ISPs and therefore incurs substantial overhead. This will make deployment of the DDoS defense mechanism difficult in large networks, such as the Internet.
- *Incremental deployment*: A defense mechanism should be incrementally deployed if it requires enhancements on network entities, such as routers. It is unrealistic to assume that all the enhancements can be achieved at the same time. The incremental deployment property allows the defense to gradually gain its effectiveness with respect to the degree of deployment.
- *Accuracy*: An effective DDoS countermeasure should be accurate, in terms of low false positive ratio and low false negative ratio. Low false positive ratio refers to that the defense should not lead to significant collateral damage to legitimate traffic, and low false negative ratio means that only a negligible portion of attack traffic is undetected. More importantly, accuracy must be maintained all the time even when the DDoS defense mechanism is under attacks launched by attackers who possess reasonable and sufficient resources, such as a complete topological map of the Internet and the IP addresses of the Internet routers. Attackers would try all the possibilities to circumvent the defense such that (1) attack traffic can circumvent detection and filtering mechanism, or (2) the defense mechanism will be deceived into misjudging legitimate packets as malicious ones. Thus, it is important for a DDoS defense mechanism to resist sophisticated attacks and keep its accuracy under all circumstances.

To defend against DDoS attacks, many countermeasures have been proposed in the literature in recent years. (These DDoS defense mechanisms are reviewed in Section Related work.) Most of these schemes (Belenky and Ansari, 2003; Bellovin et al., 2003; Dean et al., 2002; Keromytis et al., 2002; Keromytis et al., 2004; Kung et al., 2002; Kung et al., 2003; Li et al., 2001; Ioannidis and Bellovin, 2002; Mahajan et al., 2002; Mirkovic et al., 2002; Sanchez et al., 2001; Savage et al., 2000; Savage et al., 2001; Snoeren et al., 2001; Snoeren et al., 2002; Song and Perrig, 2001) are somewhat weak against DDoS attacks in large networks, such as the Internet. Some of them require supports from all edge routers and complex cooperation among different ISP networks, while others do not provide real time response to an attack. Non-trivial packet filtering operations are usually involved in the process of packet forwarding. Thus, the throughput of routers, which participate in the defense of DDoS attacks, will be significantly reduced. Other schemes (Jin et al., 2003; Peng et al., 2002; Peng et al., 2003; Sung, 2002; Sung, 2003; Yaar et al., 2003) suffer from their inherent design weaknesses. For instance, they are vulnerable to sophisticated DDoS attacks, unable to distinguish DDoS attacks from flash crowd events, or are not effective under spoofed DDoS attacks.

To effectively detect and filter spoofed DDoS attacks, we propose a new anti-DDoS scheme, called ANTID. ANTID focuses on the identification of spoofed IP packets and the filtering of attack packets when a DDoS attack occurs. By weeding out spoofed IP packets constituting a dominant share of DDoS attack traffic, DDoS attackers are forced to use real source IP addresses in attack packets. This allows packet filtering mechanisms to discard packets according to their source IP addresses. Moreover, sophisticated resource management schemes can be used in conjunction with the proposed scheme to sustain the quality of protected Internet services.

Our scheme is inspired by hop-count filtering scheme (Jin et al., 2003) and path identification scheme (Yaar et al., 2003). In the proposed scheme, each Internet router participating in the defense of DDoS attacks deterministically marks each incoming IP packet such that every IP packet can arrive at its destination along with a unique *path fingerprint* representing the route it has traversed. (In the context of this paper, routers that participate in DDoS defense mechanisms are referred to as *participating routers*.) As we shall see shortly in Section Related work, though there are other approaches attempting to create a unique path identifier for each Internet path, they are somewhat weak in defending against sophisticated DDoS attacks. Our approach proposed a new method to create such a unique identifier and can resist those sophisticated attacks.

Since a spoofed IP packet is unlikely to have a path fingerprint identical to that of the source IP address being spoofed, a destination host can identify a majority of spoofed IP packets and then discard these packets when it is under spoofed DDoS attacks. From this point of view, establishing the mapping table, which contains the mappings from communicating peers' IP addresses to their corresponding path fingerprints, is essential for the effectiveness of our approach. Theoretically, it seems that the mapping table should contain the mappings of all live IP addresses so that an Internet server under attacks can judge IP packets sent from every possible IP addresses on the Internet. However, from a practice perspective, it is usually unnecessary to do so since the set of IP addresses which frequently visits a normal site usually takes a relatively small portion of all live IP addresses (Jung et al., 2002; Peng et al., 2003). At the same time, building such a mapping table containing only frequently contacted clients will greatly reduce the storage requirement and lookup time of an Internet server. Thus, in addition to the proposed scheme for identifying/filtering spoofed IP packets with path fingerprints, we give an efficient method that enables an Internet server to construct and update the database in an on-demand fashion. In this way, a mapping of an IP address is created or updated only when the Internet server receives an IP packet from the IP address or when there are changes on the Internet path between the server and the IP address.

There are two execution modes in the proposed scheme, namely *monitor mode* and *filter mode*. By default, the proposed scheme stays in the monitor mode. In this mode, the proposed scheme collects and updates the path fingerprints of clients who want to connect to the protected Internet server. No spoofed packet is discarded in this mode. However, once the rate of spoofed packets received exceeds a predefined threshold, the proposed scheme switches to the filter mode. In the filter mode, spoofed packets and IP packets sent from infrequently contacted clients (that is, clients whose IP addresses and correspondent path fingerprints are not yet recorded) are discarded to guarantee service quality to frequent clients.

ANTID has the advantages of strong incremental deployment property, lightweight processing load for marking, decoding and filtering, and strong incentive of deployment. It does not require cooperations between ISP networks, and the filtering of spoofed DDoS packets is performed on a per packet basis. ANTID also possesses other useful characteristics that are not present in other schemes (Jin et al., 2003; Yaar et al., 2003). First, it can maintain high accuracy (i.e. low false negative ratio and low false positive ratio) even when under a sophisticated DDoS attack (Details will be presented in Section New attacking technique.). Second, it can differentiate Internet paths in which the last 8 or 16 routers are identical. Third, the proposed scheme works well even when attackers are near the victim (in terms of number of hops) and conventional schemes cannot work well. According to the experiment results, ANTID can identify 99.95% spoofed packets. Experiment results also indicate that the higher an aggregated attack rate is, the sooner the attack can be detected.

Note that ANTID is designed to defend against DDoS attacks which mainly consisted of spoofed packets. Other types of DDoS attacks, such as Distributed Reflector DoS (DRDoS) is out of the scope of this paper. In a DRDoS attack, the attack packets sent to the victim server are generally not spoofed and can be handled by conventional schemes. In this case, DRDoS victim will not directly benefit from ANTID. However, since the packets for triggering a DRDoS attack (the packets delivered to the reflectors) are generally spoofed, our approach can detect the spoofed packets, and make it very difficult for attackers to collect a sufficiently large number of reflectors. In other words, with a wide deployment of our scheme, the DRDoS attack can also be hampered.

The paper is organized as follows. Section Related work reviews the conventional DDoS defense mechanisms. Section New attacking technique presents a new attack that can circumvent conventional DDoS defense schemes. Section Proposed scheme presents the details of the proposed path fingerprint scheme. Section Robustness against circumvention analyzes the robustness of our approach against attacks. Section Evaluation presents experimental results and this paper concludes with the last section.

Related work

Many DDoS defense mechanisms have been presented in the literature recently. These schemes can be roughly categorized into four classes: *attacker-end based*, *network-based*, *victim-end based*, and *hybrid*. The attacker-end based approaches (Li et al., 2001; Mirkovic et al., 2002) attempt to identify DDoS attack traffic or spoofed IP packets at attack sources. Once DDoS attack traffic or spoofed packets are detected, proactive filtering mechanisms are activated to stop attack traffic from entering the Internet. Although these approaches can effectively reduce network congestions caused by attack traffic, their effectiveness of defending against DDoS attacks heavily depends on the wide deployment on the Internet. Moreover, the lack of incentive for installing defense mechanisms at sources and the shortage of incremental deployment property will weaken the feasibility of these approaches in large networks, such as the Internet.

The network-based approaches count on Internet routers to defend against DDoS attacks in a cooperative manner. Schemes in this category perform either the traceback of the attack traffic or complex filtering operations on routers. IP traceback schemes (Belenky and Ansari, 2003; Bellovin et al., 2003; Dean et al., 2002; Sanchez et al., 2001; Savage et al., 2000; Savage et al., 2001; Snoeren et al., 2001; Snoeren et al., 2002; Song and Perrig, 2001) focus on identifying the origins of spoofed DDoS attacks, rather than stopping these attacks. Thus, it does not provide immediate help to victims when an attack occurs. On the other hand, the on-line filtering mechanisms (Ferguson et al., 2000; Keromytis et al., 2002; Keromytis et al., 2004; Kung et al., 2002; Kung et al., 2003; Li et al., 2001; Ioannidis and Bellovin, 2002; Mahajan et al., 2002) can immediately alleviate the syndrome of DDoS attacks. However, all these schemes require significant enhancements to the current routing infrastructure, non-trivial filtering operations involved in the packet forwarding process and complex cooperation among different ISP networks. These requirements may increase the difficulty in deploying these schemes, and thus, they may not be put into practice in the near future.

The victim-end approaches (Jin et al., 2003; Peng et al., 2002; Peng et al., 2003) try to enhance the resilience of Internet servers against DDoS attacks. The advantages of the victim-end approaches are that they do not require support from the Internet routing infrastructure and that they strongly motivate the victim to deploy these schemes owing to the direct benefit to the victim itself. These schemes exploit essential characteristics of spoofed DDoS attacks in designing their DDoS countermeasures. That is, the source IP addresses of spoofed DDoS attack packets are usually *spoofed randomly*. In some schemes (Jin et al., 2003), spoofed DDoS packets are identified according to the hop-count information, which refers to the number of routers traversed in an Internet path. The effectiveness of hop-count filtering is based on the assumption that most spoofed IP packets do not have hop-count values identical to those of IP addresses being spoofed. By inferring the hop-count information from the TTL field, spoofed IP packets can be easily identified and then be discarded when the victim is under attack. However, this assumption is somewhat weak. By observing network traffic or using traceroute technique, a DDoS attacker can obtain the hop-count value between the victim and a spoofed IP address. Thus, hop-count filtering is likely to be compromised by sophisticated DDoS attacks that can adjust the initial TTL value according to the collected hop-count information.

Based on the 32-bit source IP address, an IP filtering technique for defending against DDoS attacks is proposed (Peng et al., 2002;

Peng et al., 2003). In this approach, the number of new IP addresses connecting to the protected server is monitored, and IP addresses of frequently contacted clients are learned from past communications. A surge in the number of new IP addresses is considered as a signal of the onset of a spoofed DDoS attack. Then, packets with source IP addresses not found in the IP address database will be discarded during the attack. This approach suffers from several drawbacks. First, it cannot distinguish flash crowd events from real spoofed DDoS attacks. Second, before launching a real DDoS attack, an attacker can slowly pollute the IP address database of the victim by sending the victim a set of malicious packets with an IP address to be spoofed. Then, the attacker can use those IP addresses used before to attack the target. Although it is indicated (Peng et al., 2003) that such an attack can be prevented by increasing the period over which IP addresses must appear to be considered frequent, this approach, on the other hand, will exclude some legitimate clients from the IP address database. Subsequently, the number of legitimate clients allowed to access the service is reduced. In other words, the effectiveness of defending against spoofed DDoS attacks is not significant since some frequent clients may be prohibited from accessing the protected service.

Schemes in the fourth category can be considered a hybrid of network-based and victim-based approaches. These schemes require support from the Internet routing infrastructure and from the victim or victim network. In these schemes, routers mark each incoming IP packet in a deterministic or probabilistic manner. Then, in victims or victim networks, attack packets are identified and discarded on a per packet basis according to marks left by Internet routers (Sung, 2002; Sung, 2003; Yaar et al., 2003). An IP traceback method is employed to construct the attack graph, and subsequently IP packets marked with one of network edges in the attack graph are discarded (Sung, 2002; Sung, 2003). This scheme suffers from the large number of packets required to construct the attack graph. And, it may misclassify legitimate packets as attack packets if legitimate packets ever traversed the network edge in the attack graph.

In another scheme (Yaar et al., 2003), each participating router marks some bits (one or two bits) in the Identification field of an IP packet according to the router's IP address and the TTL value in the IP header. In this way, an IP packet will arrive at its destination along with a unique identifier representing the path it has traversed. Since the marking is deterministic, packets traversing the same path will share an identical path identifier. With this scheme, the path identifier of a single identified attack packet will provide the victim the ability to filter subsequent attack packets with the same path identifier. However, the motivation of using this approach is unclear. Since the victim is capable of detecting a single attack packet, the reason for not using the same detecting facility to detect subsequent attack packets is not mentioned. Moreover, consider that packets traversed the same last 8 routers before they enter the Autonomous System (AS) at which the victim resides, the victim will not be able to distinguish these packets since their path identifiers are identical (in the case of each router marks two bits in the Identification field). As a result, IP packets of legitimate clients that traverse the same last 8 routers with attack packets will be discarded. Furthermore, consider a special case where the number of participating routers between an attack and the victim is smaller then 8. The attack can partially pollute the attack mark list, which represents the marks of attack packets. It is because, in this case, some bits in the Identification field are under the control of the attacker and remain unmarked throughout the path. This may result in mis-classifying a large portion of legitimate packets as attack ones. Finally, this scheme cannot handle DDoS attacks originated from the AS in which the victim resides, since the marking operation is suppressed on routers in the AS.

New attacking technique

In this section, we propose a new attacking technique that can be used to circumvent hop-count-filtering approach. Specifically, we will illustrate using IP SOURCEROUTE option and ICMP echo-request/echo-reply messages to explore the list of intermediate routers between two remote systems. In this way, an attacker will obtain the hop-count information that is sufficient to dodge hop-count filtering.

We will first give a scenario for obtaining the hop-count information between two remote hosts. Later on in this section, a concrete example will be demonstrated to show the feasibility of this scenario. Fig. 1 depicts a simple spoofed attack scenario that an attacker **A** attempts to send spoofed IP packets to a victim **V**, with source IP address **S**. To dodge the hop-count filtering mechanism installed at **V**, the attacker **A** must acquire the number of intermediate routers between **S** and **V**. To achieve this objective, **A** must first obtain the

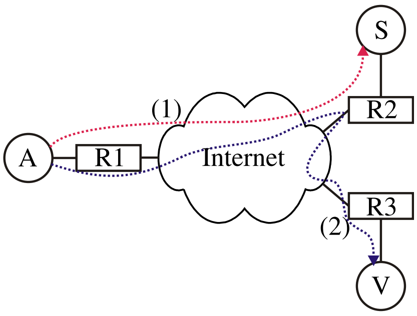
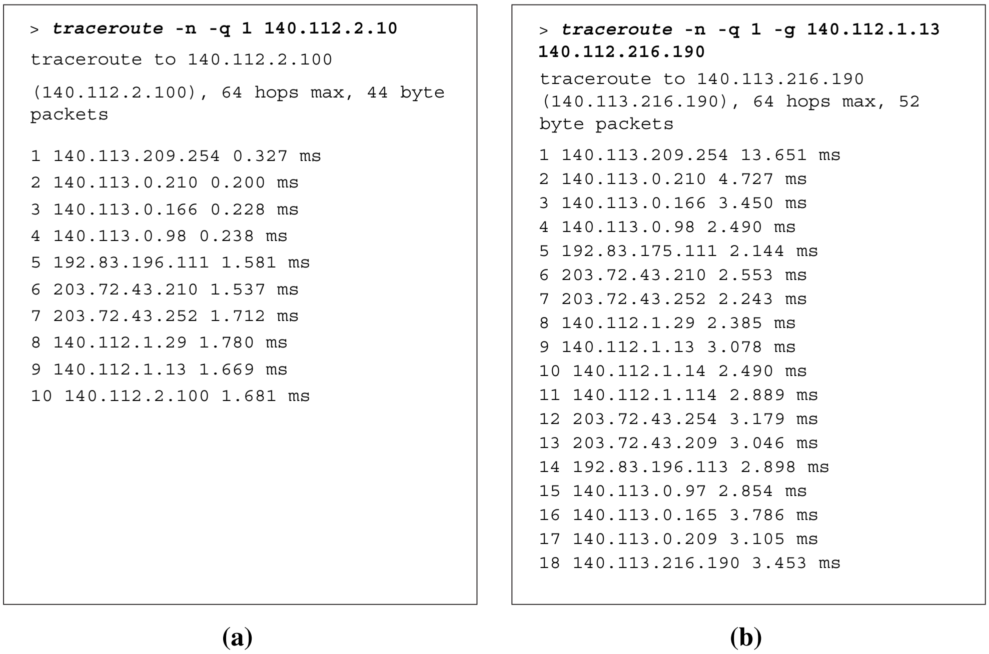


Figure 1 A two-step scenario for remotely exploring the number of hops between two end hosts.

IP address of the default gateway of **S**, i.e. the IP address of **R2**. (Here, it is assumed that an Internet host has the same default gateway for both inbound and outbound traffic. In other words, the ingress and egress routers of an Internet host are identical.) As shown in step (1) of the figure, **A** can easily obtain the IP address of **R2** by using traceroute to obtain the list of routers between **A** and **S**. Next, as shown in step (2), to acquire the number of hops between **S** and **V**, **A** can issue another traceroute command to explore the list of routers between itself and **V**. By setting the IP SOURCE-ROUTE option on, traceroute packets are forced to traverse **R2**, and **A** can successfully obtain the number of intermediate routers and their IP addresses between **S** and **V**.

To illustrate the feasibility of the scenario presented above, an example of exploring the list of routers between two remote sites is demonstrated. The following example is conducted by using the traceroute program in FreeBSD 5.0. In the following example, **A** stands for ''140.113.209.21'' and **S** stands for ''140.112.2.100''. Then, **V** is ''140.113.216.190''. First, the attacker in ''140.113.209.21'' explores the default gateway of ''140.112.2.100'' by issuing the command *traceroute 140.112.2.100*. Fig. 2(a) shows the result of this command, and according to this figure, the IP address of the default gateway of **S** is ''140.112.1.13''. Afterwards, the attacker issues another traceroute command: *traceroute -g 140.112.1.13 140.113.216.190*, and the correspondent result is shown in Fig. 2(b). By comparing the two traceroute results, the attacker can easily find that the number of hops between the two host is 9. Consequently, the attack can successfully dodge the hop-count filtering mechanism by setting an appropriate initial TTL value in the IP packet header.

The aforementioned example shows that hop-count filtering is vulnerable to sophisticated DDoS attacks that can explore the hop-count information between the victim and spoofed IP addresses. Thus, developing a new technique for defending against sophisticated spoofed DDoS attacks is needed.



```
> traceroute -n -q 1 140.112.2.10
traceroute to 140.112.2.100
(140.112.2.100), 64 hops max, 44 byte
packets

1 140.113.209.254 0.327 ms
2 140.113.0.210 0.200 ms
3 140.113.0.166 0.228 ms
4 140.113.0.98 0.238 ms
5 192.83.196.111 1.581 ms
6 203.72.43.210 1.537 ms
7 203.72.43.252 1.712 ms
8 140.112.1.29 1.780 ms
9 140.112.1.13 1.669 ms
10 140.112.2.100 1.681 ms
```

(a)

```
> traceroute -n -q 1 -g 140.112.1.13
140.112.216.190
traceroute to 140.113.216.190
(140.113.216.190), 64 hops max, 52
byte packets
1 140.113.209.254 13.651 ms
2 140.113.0.210 4.727 ms
3 140.113.0.166 3.450 ms
4 140.113.0.98 2.490 ms
5 192.83.175.111 2.144 ms
6 203.72.43.210 2.553 ms
7 203.72.43.252 2.243 ms
8 140.112.1.29 2.385 ms
9 140.112.1.13 3.078 ms
10 140.112.1.14 2.490 ms
11 140.112.1.114 2.889 ms
12 203.72.43.254 3.179 ms
13 203.72.43.209 3.046 ms
14 192.83.196.113 2.898 ms
15 140.113.0.97 2.854 ms
16 140.113.0.165 3.786 ms
17 140.113.0.209 3.105 ms
18 140.113.216.190 3.453 ms
```

(b)

Figure 2 (a) An example of determining the default gateway of an IP address being spoofed, and (b) an example of enumerating the list of routers between a spoofed source and the victim.

Proposed scheme

In this section, we propose a new spoofed packet filtering anti-DDoS scheme, called ANTID. In this scheme, each IP packet is embedded with a unique *path fingerprint* representing the route an IP packet has traversed, and IP packets with incorrect path fingerprints is considered spoofed. The proposed scheme eliminates some weaknesses of conventional schemes, and is designed specifically for defending against spoofed DDoS attacks. It intends to complement, rather than replace existing schemes.

The basic of ANTID is the validation of an IP packet via its source IP address and the path fingerprint embedded in it. In this section, the computation of a path fingerprint is first described, and then the inspection algorithm for identifying spoofed IP packets is presented. Next, an efficient approach for constructing a table that contains the mappings of IP addresses and their path fingerprints is proposed. Finally, the details of detecting a spoofed DDoS attack are shown, and subsequent packet filtering operations are examined.

Path fingerprinting and spoofed packet inspection

To generate a path fingerprint representing the route an IP packet traversed, it is assumed that each participating router assigns each of its network interface a n-bit random number, and these random numbers are *kept securely*. These numbers should not be disclosed. In ANTID, a *path fingerprint* of an IP packet is composed of two fields, a d-bit *distance* field and a n-bit *path identification* (PID) field, where the former represents the number of intermediate routers traversed, and the latter denotes an identifier derived from the random numbers associated with the traversed network interfaces in the route. The path fingerprint of an IP packet is stored in the IP packet header, and thus it is delivered to the destination host along with the packet. Moreover, we also assume that a *pf-flag* bit in the IP packet header is available for indicating the start of path fingerprinting. Later in this section, we will discuss the allocation of $(1 + d + n)$ bits in the IP packet header fields.

The path fingerprinting procedure is presented as follows. Whenever a participating router receives an IP packet, it first examines the pf-flag field. If it is unset, i.e. 0, the receiving router is then aware that it is the first participating router the packet encountered in the path. In this case, the receiving router sets the pf-flag bit to 1, sets the distance field to 1 and sets the path identification field to the random number associated with the incoming interface of the packet. On the other hand, if the flag bit is already on, i.e. 1, the receiving router increments the distance field by one, and updates the path identification field with H(PID, N_i), where PID represents the current value of path identification field in the packet, N_i denotes the random number of the incoming interface, and H is a one-way hash function with weak collision resistance. (Note that H is not a secret and each participating router can choose its hash function.) Algorithm 1 shows the pseudocode for computing path fingerprint on a participating router, and Fig. 3 illustrates an example of the path fingerprinting scenario.

In the example depicted in Fig. 3, a packet traverses from the source **S** to the destination **D** across routers **R1**—**R4**. The first router in the path, **R1**, sets both pf-flag and distance filed to 1 and sets the initial PID value to the random number of the incoming interface, i.e. N_1. Afterwards, each router increases the distance field and updates the PID field according to the previous PID value and the random number of the current incoming interface. In this figure, H denotes a hash function.

To allocate space from the IP packet header for storing a path fingerprint, the 16-bit Identification field in the IP header is chosen to be overloaded. Issues related to the overloading of this field has been studied and reported (Savage et al., 2000). In this paper, the 16-bit Identification field is divided into two sub-fields. The first sub-field is 5-bit long and is used to store the value of distance. It is believed that 5 bits are sufficient (Carter and Crovella, 1997; Theilmann and Rothermel, 2000) since most of Internet paths are shorter than 31

Algorithm 1 Computation of path fingerprints on a participating router

```
Let p denote an incoming IP packet.
Let p.pf-flag, p. distance and p.pid denote the
  pf-flag, distance and path identification fields in
  packet p, respectively.
Let N_i denote the random number associated with
  the incoming interface of p.
if $p.pf\text{-}flag = 0$ then
    $p.pf\text{-}flag \leftarrow 1$
    $p.distance \leftarrow 1$
    $p.pid \leftarrow N_i$
else
    $p.distance \leftarrow p.distance + 1$;
    $p.pid \leftarrow \mathrm{H}(p.pid, N_i)$;
end if
```

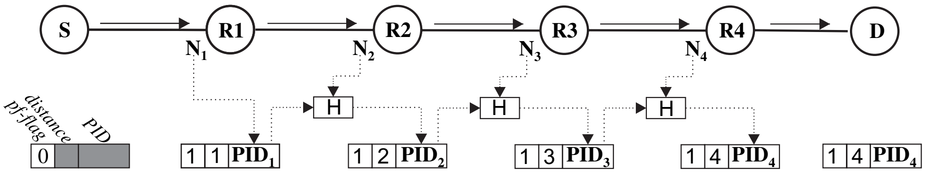


Figure 3 An example of the proposed path fingerprinting scheme.

hops. To deal with Internet paths with more than 31 hops, a simple solution is to use more bits. However, this will reduce remaining bits for storing path identification, and consequently increase the collision rates. To avoid increasing hash collisions, in our scheme, we choose to stop increment the distance field when its value reaches 31. Though in this case, Internet paths that have more than 31 routers supporting our scheme will have the same distance values; the path identification field can still help distinguish them if their path identifications are different. The remaining 11 bits of the Identification field are used to store path identification. Finally, we propose to use the un-used bit of the FLAG field in IP header to store the value of the pf-flag bit.

In this way, filtering of spoofed IP packets will be quite straightforward if the table that contains the mappings of IP addresses and their path fingerprints is present. In the following context of this paper, the table is referred to as *S2PF* (abbreviation of Source to Path Fingerprint) table. Here, we first assume that the S2PF table is available, and its construction will be discussed later.

Algorithm 2 shows the pseudo-code for identifying spoofed IP packets. In this algorithm, the source IP address and path fingerprint of an IP packet are extracted from the IP packet header first. Next, by using the extracted source IP address, the correspondent path fingerprint can be retrieved from the S2PF table. If the path fingerprint of the given IP address cannot be found in the S2PF table, the algorithm returns *UNKNOWN*. Otherwise, the algorithm compares the two path fingerprints. (One from the IP packet and the other from the S2PF table.) If they are identical, the algorithm returns *LEGITIMATE*, or else, it returns *SPOOFED*.

Algorithm 2 Spoofed packet inspection

```
Let src denote the source IP address and pf
  denote the path fingerprint of a given IP packet.
Retrieve the path fingerprint, $pf_{s2pf}$ indexed by
  src in the S2PF table.
if No entry indexed by src found then
  Return UNKNOWN
else if $pf_{s2pf} \neq pf$ then
  Return SPOOFED
else
  Return LEGITIMATE
end if
```

Notice that the inspecting algorithm only determines whether a given IP packet is spoofed or not. Weeding out spoofed packet is not performed by ANTID alone. This is because the inspecting algorithm may mis-classify legitimate packets as spoofed ones when an out-of-date S2PF table is in use. (A S2PF table will become out-of-date when there are topological changes in the Internet, or a participating router re-assigns random numbers to its network interfaces.) To avoid errors caused by an obsolete S2PF table, spoofed packets will only be discarded after a DDoS attack signal is caught by ANTID.

The construction and update of the S2PF table

As mentioned previously, there are two execution modes in the proposed scheme, namely *monitor mode* and *filter mode*. In the monitor mode, the S2PF table is constructed, and its entries will be updated if there are changes in the topology of Internet or in the Internet paths due to dynamic routing. Notice that, ANTID does not attempt to build a S2PF table containing all live IP addresses. Instead, the S2PF table should only have entries for IP addresses that ever connected to the destination in the past communications. It is believed that S2PF constructed in this way is sufficient enough because, according to the report (Jung et al., 2002), the source IP addresses of a given site in normal conditions only take a small set of values. Thus, in ANTID, a new S2PF table entry will be added if the destination host receives IP packets from a new IP address. This allows controlling the size of the S2PF table at a manageable level, and, at the same time, reducing the time for searching.

There are different ways to learn the mapping of an IP address and its path fingerprints from communications. One naive approach is to learn the mappings simply from received IP packets. Whenever an IP packet with a new source IP address arrives, a new entry is inserted into the S2PF table. Similarly, the arrival of an IP packet that carries a new path fingerprint different from the one already stored in the S2PF table would result in an update on the correspondent S2PF entry. In this way, constructing a S2PF table is quite straightforward, however, it is clear that this naive approach will not work for identifying spoofed packets since no validation process is involved. A more complicated method is to learn the mappings from established TCP connections. Since it is very difficult to spoof source IP addresses in TCP connections, the validity of mappings learned in this way is ensured. However, this method is not appropriate for Internet servers that do not run TCP-based services. Herein, we suggest a simple and generic method to obtain the mappings in an efficient and robust manner. That is, the path fingerprint of a specific source IP address is explicitly explored by the use of ICMP echo-request messages. Before an entry of a new source IP address can be inserted into the S2PF table or an update can be made, the destination host sends an ICMP echo-request message to the source IP address. Then, the path fingerprint in the returned ICMP echo-reply message is treated as the most up-to-date path fingerprint of that source IP address. It is worthy to note here that traceroute packets might be blocked for security reasons. Thus, an alternative way is to use the ICMP port unreachable message and the TCP RST message. Specifically, the destination host can send a UDP packet to that specific source IP address and the source port number and destination port number can be set arbitrarily. In this case, it is very likely that no process is serving on that destination port, and thus an ICMP port unreachable message will be sent back to the server. In that ICMP packet, the victim server can learn the path fingerprint between itself and the specific host. Similarly, sending a TCP packet with randomly created destination port number can be used to accomplish the same objective. The corresponding TCP RST packet will carry the path fingerprint. No matter which approach is taken, the S2PF table, constructed in this way, will contain only correct mappings of legitimate clients.

There are two main reasons for invoking exploration of the path fingerprint of a specific IP address. The first refers to the arrival of an IP packet with a new IP address. The second directs to the necessity of updating a S2PF table entry. Consider the first case when the packet from a new client arrives. In this case, the spoofed packet detection algorithm presented in Algorithm 2 returns *UNKNOWN*, and an exploration process will be invoked at probability q, where $0 \leq q < 1$. In this way, we can avoid overloading the Internet and can avoid building a S2PF table containing a large portion of infrequently contacted clients. As to the second case, although the majority of Internet paths are expected to be stable and remain unchanged for a long period of time (Jin et al., 2003; Paxson, 1997), occasionally it is still necessary to update the S2PF table when the routing changes. This update function is important to maintain an up-to-date S2PF table. In the proposed scheme, upon receipt of an IP packet that traversed a new Internet path (assuming that the S2PF table has an entry for the IP address of this packet), the spoofed detection algorithm will classify this packet as spoofed. Then, in this case, an exploration process will be invoked at probability r, where $0 \leq r < 1$. Both q and r are used to prevent our scheme from excessively exploring path fingerprints by ICMP echo-request/echo-reply messages, and at the same time, we preserve the ability to insert and update entries in the S2PF table.

Since a S2PF table cannot accommodate the mappings of all possible IP addresses (there can be at most 2^{32} entries), replacing an old S2PF table entry with a new one is also an important issue that needs to be addressed. Whenever a replacement is needed, we currently recommend that several cache replacement techniques, such as *Least Frequently Used* (LFU), *Least Recently Used* (LRU) and *Most Frequently Used* (MFU), can be used. However, in this paper, we do not make definitive claim that which replacement technique is the best for the S2PF table entry replacement, and determining the best replacement policy warrants further research.

Finally, in ANTID, the number of spoofed packets received is used as a criterion to determine the onset of a spoofed DDoS attack. Thus, in the monitor mode, whenever an exploration process is invoked owing to receipt of a new IP address, the returned path fingerprint is compared with the path fingerprint stored in the IP packet. If they are not identical, a counter *spoofing-cnt*, which records the number of spoofed packet received in one unit of time, will be increased by one. Similarly, whenever the inspecting algorithm returns *SPOOFED*, the spoofing-cnt is also incremented by one unless the exploration process returns an path fingerprint identical to the one in the IP packet. Algorithm 3 shows the pseudo-code for the construction and update of the S2PF table in the monitor mode.

Algorithm 3 The construction and update of the S2PF table

```
Let $p$ denote the incoming packet.
Let $p.src$ and $p.pf$ denote the source IP address
  and the path fingerprint stored in the packet
  header of $p$.
Status ← SpoofInspection(p).
if Status = UNKNOWN then
  Let $x$ be a random number from [0,1).
  if $x < q$ then
    Invoke the path fingerprint exploration process.
    Insert a new table entry of $p.src$ into S2PF table.
    Let $p.pf_{new}$ denote the newly acquired path
      fingerprint of $p.src$
      if $p.pf_{new} \neq p.pf$ then
      Classify packet $p$ as spoofed
      $spoofing\text{-}cnt \leftarrow spoofing\text{-}cnt + 1$
    end if
  end if
else if Status = SPOOFED then
  Let $x$ be a random number from [0, 1)
  if $x < r$ then
    Invoke the path fingerprint exploration
      process.
    Update the entry of $p.src$ with newly acquired
      path fingerprint.
    Let $p.pf_{new}$ denote the newly acquired path
      fingerprint of $p.src$
    if $p.pf_{new} \neq p.pf$ then
      $spoofing\text{-}cnt \leftarrow spoofing\text{-}cnt + 1$
    end if
  else
    $spoofing\text{-}cnt \leftarrow spoofing\text{-}cnt + 1$
  end if
end if
```

State transitions and spoofed packet filtering

As mentioned, the number of spoofed packets received is used as a criterion for transition between the monitor mode and the filter mode. In the proposed scheme, the time is divided into a set of uniform time intervals. At the beginning of each time interval, the spoofing-cnt, which records the number of spoofed packet in the previous time interval, is examined. If the value of this counter exceeds a threshold T_1, ANTID switches to the filter mode. In the filter mode, spoofed packets and packets with source IP addresses absent in the S2PF table are discarded. The proposed scheme switches from the filter mode to monitor mode when IP spoofing ceases. This is achieved by examining the spoofing-cnt. If the value of this counter is smaller than another threshold T_2, ANTID switches back to the monitor mode. In this paper, we provide only a general guideline commonly used in threshold schemes for setting the two thresholds. The basic principle is to let $T_1 > T_2$. It is clear that this can prevent ANTID from alternating between the two execution modes. Another important concern is that T_1 should be set appropriately such that the victim server will not falsely switch into the filter mode. Switching to the filter mode too easily may lead to another form of DoS attack because, in the filter mode, the victim server only serves previously validated clients.

As to the problem of setting the specific values of these parameters, such as q, r, T_1 and T_2, we recommend that they should be configurable by administrators of the Internet servers. These parameters are highly application-dependent, that is, it is largely relied on administrators to determine their own best trade-off between the performance and security of the protected sites. Herein, we only briefly enumerate some factors related to the setting of these parameters. First, q is related to definition of a ''frequent'' visitor, q can be set to 1/10 if a user is considered a frequent user when the user visits the protected site for more than 10 times. Second, r highly depends on the dynamics of Internet topology. Though Internet paths were found to be persistent for days, the Internet topology is assumed to change dynamically. Further investigation on the dynamics of the Internet is needed to provide hints for setting r appropriately. As for T_1 and T_2, their values highly depend on the number of spoofed packets the protected site can tolerate in each time interval. For instance, consider an Internet server whose average packet arrival rate is about 5000 packets per second. Then, T_1 can be set as 2500 (about the 1/2 of the average number of packets), and T_2 can be set as 500 (about the 1/10 of the average number of packets). Such settings would be useful in detecting large scale of spoofed DDoS attacks that attempt to flood the protected site with spoofed packets.

Algorithm 4 shows the pseudo-code of state transition, and Algorithm 5 presents the pseudo-code of spoofed packet filtering in the filter mode.

Robustness against circumvention

In this section, we present possible approaches that a DDoS attacker may take to circumvent the proposed DDoS defense mechanism and show that our scheme is robust against these attacks. The key for an attacker to circumvent spoofed packet detection is the ability to generate attack packets in accordance to the constraint that they can

Algorithm 4 The transition between monitor and filter modes

```
In the begin of each time interval
if current execution mode = monitor then
    if $spoofing\text{-}cnt > T_1$ then
        switch to filter mode
    end if
else
    if $spoofing\text{-}cnt < T_2$ then
        switch to monitor mode
    end if
end if
$spoofing\text{-}cnt \leftarrow 0$
```

finally arrive at the victim along with path fingerprints consistent with the spoofed IP addresses. In the following, we examine two types of approaches to achieve this objective.

- *Simple attack*: The simplest approach is to set random initial values in both the path fingerprint field and the source IP address field. Notice that an attacker *cannot* seed the fingerprint field of attack packets in such a way that the fingerprint of the attack packets will arrive at the victim server along with a correct path fingerprint. This is because the value in the path fingerprint field will be changed securely. Without knowing all the random numbers associated with the traversed links, the attacker has no knowledge of the correct seed. In other words, since the random numbers associated with network links are kept securely, it is very difficult for an attacker to control path fingerprint received by the victim. Thus, the best an attacker can do is to set random seed in the fingerprint field and the source IP field. However, this approach is infeasible since it is very unlikely that the randomly spoofed source IP address will exist in the S2PF table at the destination host or that these attack packets can arrive at the destination along with a correct path fingerprint. The probability for a match is $1/2^{(d+n)}$ (in our scheme, $(d+n) = 16$). Next, consider a more sophisticated case where an attacker can carefully select a spoofed IP address and can set an appropriate value in the distance field (setting an appropriate initial value in the distance field can be achieved by using the technique presented in Section New attacking technique). In this case, only very few attack packets can pass the spoofed packet detection. It is because that the best an attacker can do is to fill the path identification field with a random value and only then the fraction $1/2^n$ (in our scheme, $n = 11$) of attack packets can arrive at the destination along with a correct path identification value. In short, such a simple attack is not useful to dodge the proposed scheme.
- *Detour attack*: As we have shown in Section New attacking technique, an attacker can determine the default gateway of a spoofed IP address. Thus it is reasonable to assume that an attacker can force attack packets to traverse the default gateway of the spoofed IP address by using IP SOURCEROUTE option. In this way, the postfix of the attack path will be identical to the path from the spoofed source to the victim. This type of attack is referred to as *detour attack*. The success of a detour attack relies on the following mandatory conditions: there must not exist any participating router in the path from the attacker to the spoofed IP address (including the default gateway of the spoofed source). If this condition holds, an attacker can successfully conduct a spoofed DDoS attack by using the detour technique presented here. In this case, the victim cannot identify spoofed attack packets since the path fingerprints of these packets are correct. Although this type of attack allows an attacker to dodge path fingerprint filtering, finding an appropriate spoofed source is very difficult if the participating routers are widely distributed over the entire Internet. Moreover, the victim can easily stop attack packets of a detour attack by filtering IP packets with IP SOURCEROUTE option set.

Algorithm 5 Spoofed packet filtering

```
Let $p$ denote an incoming IP packet.
Status = SpoofedInspection(p)
if Status = SPOOFED or UNKNOWN then
    Drop the packet p
    $spoofing\text{-}cnt \leftarrow spoofing\text{-}cnt + 1$
end if
```

According to the above analysis, we can find that both the simple attack and the detour attack are ineffective. Furthermore, from a probability point of view, in the simple attack, attack packets can bypass the spoofed packet detection at probability of $1/2^{11}$ (later on we will confirm this by experiments). In summary, the proposed scheme is robust against circumvention.

Evaluation

In this section, we evaluate the accuracy of the proposed scheme in the identification and filter of spoofed packets under DDoS attacks. We simulate the aggregate of DDoS attack traffic at different attack rates and then present the performance metrics that we measure.

Internet data sets

To simulate the Internet topology, the Internet map (Cheswick et al., 2000) is used as our Internet topological data. The Internet map contains Internet paths (each represented as a list of routers), from a specific host to most of nets on the Internet. In our experiments, the host which originates these traceroute-style path probes is viewed as the victim of a DDoS attack, and attackers and legitimate clients are randomly selected from those hosts at the end of each traceroute path.

Fig. 4(a) depicts the distribution of number of intermediate routers on each completed Internet paths. There are in total 24,772 Internet paths. (We exclude incomplete traceroute probes from our experiments.) As the figure shows, only a few Internet paths consist of more than 32 intermediate routers, and the most popular path length is 16. Fig. 4(b) shows the distribution of the path identifications of these Internet paths. Theoretically, since the numbers associated with network interfaces are randomly assigned, the path identifications of Internet paths should also be random, and Fig. 4(b) confirms this theoretical inference.

Experimental design and performance metrics

In the experiments, 500 end hosts are randomly selected from the Internet map to act as frequently contacted clients of an Internet server. Then, a S2PF table which contains the ''source to path fingerprint'' mappings of the 500 clients is constructed. Moreover, we set q to 1/10, r to 1/100 and T_1 to 2500. Notice that, in our experiments, we do not attempt to find a best suite of values of these configurable parameters for a specific network environment. (As indicated in Section Proposed scheme, the setting of these parameters heavily depends on both specific characteristics of deployed networks and the trade-off between security and performance. Thus, we do not focus on this issue in this paper.) Instead, other behaviors, such as the growth rate of the S2PF table and the false negative ratio at the monitor mode and the filter mode, are explored under various attack rates.

First, we measure the false negative ratio of the proposed scheme before and after it switches from monitor mode to filter mode. Herein, the false negative ratio refers to the ratio of undetected spoofed packets. In this experiment, we simulate the aggregate of attack traffic that have 5000 attack packets in each attack round. (Note that the growth of the number of attackers can only increase the aggregation of attack traffic, but cannot increase the probability of passing the proposed filtering mechanism. Thus, we use the aggregation of attack traffic to test the proposed scheme.) A round of attack, in fact, stands for a period of time. In this paper, we do not impose restrictions on setting the length of an attack round. Instead, we are interested in the number of rounds and the rate required to detect the presence of a DDoS attack. Source IP addresses of these attack packets are randomly selected from the Internet map with a constraint that these addresses are disjoint with legitimate clients. Moreover, by the same technique presented in Section New attacking technique, we let these attack packets carry appropriate distance values

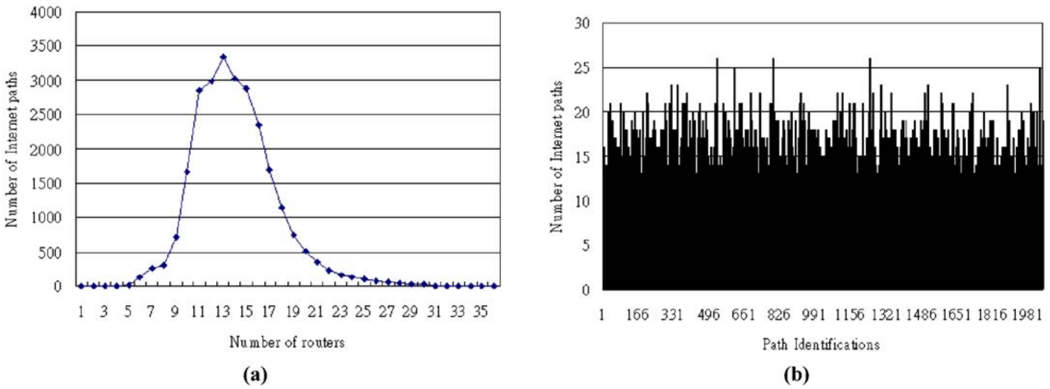


Figure 4 The distribution of: (a) number of intermediate routers, and (b) the value of path identifications.

such that they can arrive at the victim along with consistent distance values with the spoofed IP addresses. As shown in Fig. 5, the false negative rate shows a tendency to decrease as the number of rounds increases, and finally at round 3156, the proposed scheme switches to filter mode. Thus, after round 3156, the false negative ratio steeply drops to around 1/2048. The decrease of false negative ratio at the monitor mode is caused by the growth of the size of the S2PF table. Since the victim will add a new S2PF table entry at probability 1/100, about (1/100) × (number of spoofed IP addresses not in the S2PF table) new entries will be added to the S2PF table after each attack round. At round 3156, there are in total 11,937 entries (about the half of number of Internet paths in Internet map) in the S2PF table.

As to false negatives, in our approach, legitimate packets will not be classified as spoofed ones in the monitor mode. They will be classified as UNKNOWN packets, or their path fingerprints will be inserted into the S2PF table after the fingerprint exploration process. While in the filter mode, our scheme indeed will discard some legitimate packets originated from infrequent clients whose path fingerprints are not in the S2PF table. These packets are still classified as UNKNOWN rather than SPOOFED. Possible cases of mis-classifying legitimate packets as spoofed ones are mostly caused by changes in the Internet topology when our scheme is in the filtering mode. In filtering mode, our scheme stops updating path fingerprints for SPOOFED packets. Such false positives highly depend on the dynamics of the Internet, and therefore are not discussed in the context of this paper.

According to the statistics on the measured DDoS attacks, attack rates range from 500 to 600,000 attack packets per second (Darmohray and Oliver, 2000; Moore et al., 2001). In the following, we will present experiments on how many packets are needed for our scheme to detect the presence of a DDoS attack, and please notice that the sensitivity presented here highly depends on the settings of configuration variables. Fig. 6 shows the number of rounds required to detect the presence of a spoofed DDoS attack under three attack rates: 50,000, 100,000 and 150,000 attack packets per round. This figure shows that the number of rounds needed to detect a spoofed DDoS attack decreases as the attack rate increases. Next, we conduct the same experiments at lower attack rates. In these experiments, we send 5000, 10,000, 15,000, ..., 50,000 attack packets to the victim at each round. We observed the growth of the S2PF table and the number of rounds needed to detected attacks. The experimental result is shown in Table 1. According to the table, we can find that the higher the attack rate is, the fewer rounds and the fewer entries in the S2PF table are required to detect the presence of an attack.

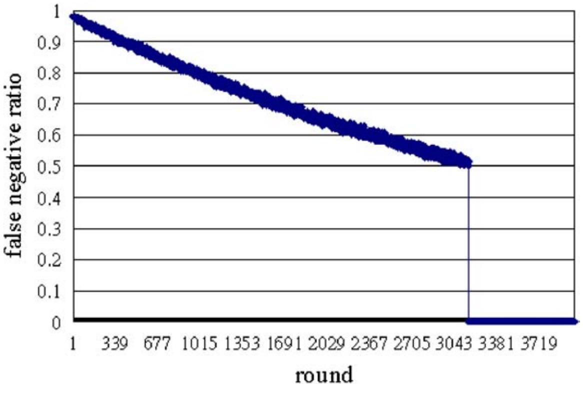


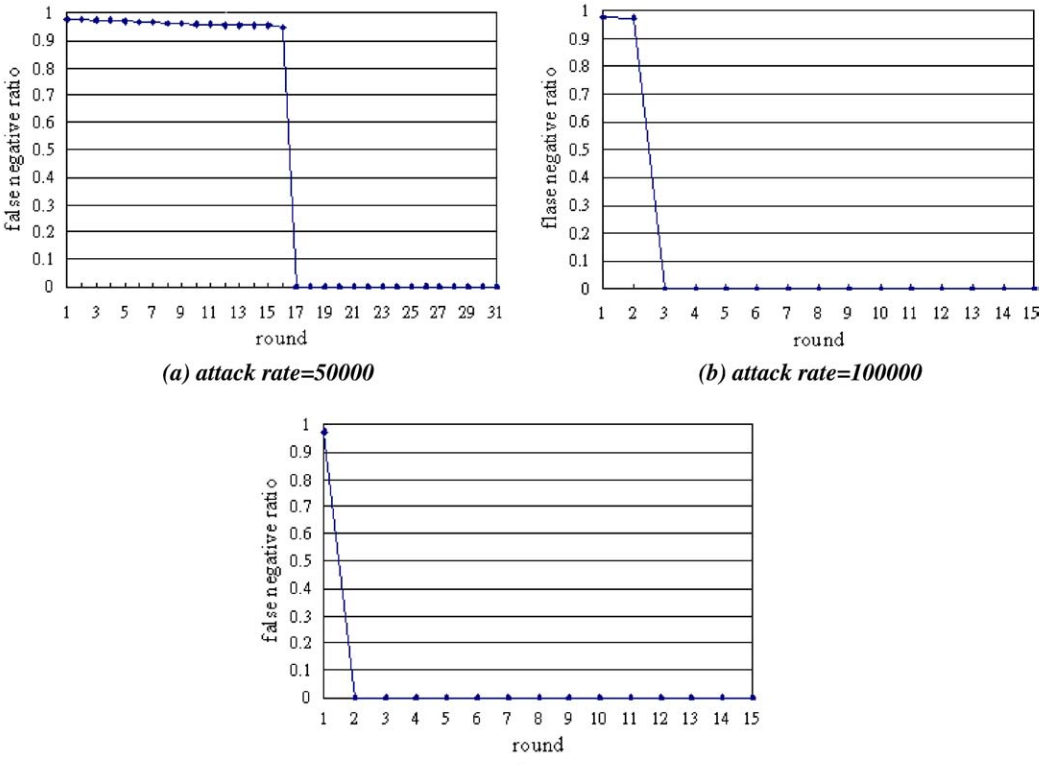
Figure 5 The false negative ratio under the attack rate of 5000 packets per round.

In summary, the experimental results show that a higher attack rate will result in a smaller number of rounds required to detect an ongoing spoofed DDoS attack. After the attack is detected, only a very small fraction (around 1/2048) of attack packets can pass the spoofed packet detection mechanism. This shows that our approach can effectively detect the presence of an attack and subsequently can weed out these attack packets.

Conclusions and future work

In this paper, we presented an anti-DDoS scheme, ANTID, for defending against spoofed DDoS traffic. ANTID intends to complement, rather than replace existing schemes. For instance, the proposed scheme helps to discard spoofed packets before ingress filters are installed on all edge routers. Furthermore, by weeding out a majority of spoofed attack packets, our approach allows some resource management systems, that share resource fair amount many participants, to work better.

In our approach, each IP packet is embedded with a unique path fingerprint that represents the Internet path it has traversed. By learning path fingerprints from past traffic, the victim can efficiently establish the S2PF table which contains the mappings of source IP addresses and corresponding path fingerprints of frequently contacted clients. A spoofed packet can be easily identified



(a) attack rate=50000

(b) attack rate=100000

(c) attack rate=150000

Figure 6 The false negative ratio under the attack rates of 50,000, 100,000 and 150,000 attack packets per round.

by consulting the S2PF table since it is very unlikely that a spoofed packet can have a path fingerprint identical to that of the spoofed IP address. Thus, by identifying and filtering spoofed packets, a spoofed DDoS attack can be identified and prevented. This makes the proposed scheme an effective and efficient approach for defending against spoofed DDoS attacks.

ANTID runs in two execution modes, the monitor mode and the filter mode. We simulate DDoS attacks with variable attack rates to evaluate the performance of our approach. Experimental results showed that ANTID can provide protection against spoofed DDoS attack. Only around 1/2048 attack packets can pass the spoofed packet detection and filter mechanism when our scheme stays in filter mode. The experimental results also show that the time required to detect an attack depends on the attack rate of aggregated attack traffic. The higher the attack rate is, the short the time for detecting will be. Finally, our approach possesses several important properties, such as strong incremental deployment and lightweight for marking, decoding and filtering. No cooperation among ISP networks is needed. More importantly, it is robust against sophisticated DDoS attacks, and it is resistant to the deception by nearby attackers. These properties make the proposed scheme a general and robust approach that is feasible to be deployed in the Internet. There are several issues that require further investigations. For instance, a systematic way for configuring parameters, q, r, T_1 and T_2, for a specific network environment is required. And, an efficient approach for maintaining the S2PF table is needed. Finally, the best strategy for deploying participating routers in the Internet needs to be designed and investigated.

Table 1 At different attack rates, the number of rounds and the number of table entries required to detect the attack

Attack rate	5000	10,000	15,000	20,000	25,000	30,000	35,000	40,000	45,000	50,000
Table size	11,937	5928	3983	2990	2462	2037	1750	1517	1378	1243
Number of rounds	3156	621	259	136	88	52	37	27	20	16

References

Belenky A, Ansari N. IP traceback with deterministic packet marking. IEEE Communications Letters April 2003;7(2):162—4.

Bellovin S, Leech M, Taylor T. ICMP traceback messages [Online]. Available from: http://www.ietf.org/internet-drafts/draft-ietf-itrace-04.txt; Feb. 2003.

Carter RL, Crovella ME. Server selection using dynamic path characterization in wide-area networks. In: Proceedings of the IEEE INFOCOM; Apr. 1997. p. 1014—21.

CERT Coordination Center. CERTR incident note IN-99-07 distributed denial of service tools [Online]. Available from: http://www.cert.org/incident_notes/IN-99-07.html; Jan. 1999a.

CERT Coordination Center. Results of the distributed-systems intruder tools workshop [Online]. Available from: http://www.cert.org/reports/dsit-workshop-final.html http://www.cert.org/reports/dsit-workshop.pdf; Nov. 1999.

CERT Coordination Center. CERTR advisory CA-1999-17 denial-of-service tools [Online]. Available from: http://www.cert.org/advisories/CA-1999-17.html; Dec. 1999.

CERT Coordination Center. CERTR advisory CA-2000-01 denial-of-service developments [Online]. Available from: http://www.cert.org/advisories/CA-2000-01.html; Jan. 2000.

Cheswick B, Burch H, Branigan S. Mapping and visualizing the internet. In: Proceedings of USENIX annual technical conference [Online]. Available from: http://www.usenix.org/publications/library/proceedings/usenix2000/general/cheswick.html; June 2000.

Darmohray T, Oliver R. ''Hot spares'' for DoS attacks;login [Online]. Available from: http://www.usenix.org/publications/login/2000-7/apropos.html; July 2000.

Dean D, Franklin M, Stubblefield, A. ''An algebraic approach to IP traceback''. ACM Transactions on Information and System Security May 2002;5(2):119—37.

Dittrich D. The DoS project's trinoo distributed denial of service attack tool [Online]. Available from: http://staff.washington.edu/dittrich/misc/trinoo.analysis; Oct. 1999a.

Dittrich D. The tribe flood network distributed denial of service attack tool [Online]. Available from: http://staff.washington.edu/dittrich/misc/tfn.analysis; Oct. 1999b.

Ferguson P, Senie D. Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing. Internet engineering task force, RFC 2827 [Online]. Available from: http://www.rfc-editor.org/rfc/rfc2827.txt; May 2000.

Ioannidis J, Bellovin SM. Implementing pushback: router-based defense against DDoS attacks. In: Proceedings of network and distributed system security conference. p. 79—86 [Online]. Available from: http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/ioanni.pdf; Feb. 2002.

Jin C, Wang H, Shin KG. Hop-count filtering: an effective defense against spoofed ddos traffic. In: Proceedings of ACM conference on computer and communications security; Oct. 2003. p. 30—41.

Jung J, Krishnamurthy B, Rabinovich M. Flash crowds and denial of service attacks: characterization and implications for CDNs and web sites. In: Proceedings of IEEE international world wide web conference; May 2002. p. 252—62.

Keromytis AD, Misra V, Rubenstein D. SOS: secure overlay services. In: Proceedings of the 2002 ACM conference on applications, technologies, architectures, and protocols for computer communications; Aug. 2002. p. 61—72.

Keromytis AD, Misra V, Rubenstein D. SOS: an architecture for mitigating DDoS attacks. IEEE Journal on Selected Areas in Communications Jan. 2004;22(1):176—88.

Kung HT, Bradner S, Tan K-S. An IP-layer anonymizing infrastructure. In: Proceedings of MILCOM, vol. 1; Oct. 2002. p. 389—94.

Kung HT, Cheng C-M, Tan K-S, Bradner S. Design and analysis of an IP-layer anonymizing infrastructure. In: Proceedings of the third DARPA information survivability conference and exposition, vol. 1; Apr. 2003. p. 62—75.

Li J, Mirkovic J, Wang M, Reiher P, Zhang L. Save: source address validity enforcement protocol. In: Proceedings of IEEE INFOCOM, vol. 3; June 2001. p. 1157—566.

Mahajan R, Bellovin SM, Floyd S, Ioannidis J, Paxson V, Shenker S. Controlling high bandwidth aggregates in the network. ACM Computer Communications Review July 2002; 32(3):62—73.

Mirkovic J, Prier G, Reiher P. Attacking DDoS at the source. In: Proceedings of international conference on network protocols; Nov. 2002. p. 312—21.

Moore D, Voelker G, Savage S. Inferring internet denial of service activity. In: Proceedings of USENIX security symposium [Online]. Available from: http://www.usenix.org/events/sec01/moore.html; Aug. 2001.

Paxson V. End-to-end routing behavior in the internet. IEEE/ACM Transactions on Networking Oct. 1997;5(5):601—15.

Peng T, Leckie C, Ramamohanarao K. Detecting distributed denial of service attacks using source IP address monitoring. Australia: The University of Melboume; 2002. Tech. rep [Online]. Available from: http://www.ee.mu.oz.au/pgrad/taop/research/detection.pdf.

Peng T, Leckie C, Ramamohanarao K. Protection from distributed denial of service attacks using history-based IP filtering. In: Proceedings of IEEE international conference on communications, vol. 1; May 2003. p. 482—6.

Sanchez LA, Milliken WC, Snoeren AC, Tchakountio F, Jones CE, Kent ST, et al. Hardware support for a hash-based IP traceback. In: Proceedings of the second DARPA information survivability conference; June 2001. p. 146—52.

Savage S, Wetherall D, Karlin AR, Anderson T. Practical network support for IP traceback. In: Proceedings of SIGCOMM conference; Aug. 2000. p. 295—306.

Savage S, Wetherall D, Karlin AR, Anderson T. Network support for IP traceback. IEEE/ACM Transactions on Networking June 2001;3:226—37.

Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F, Kent ST, et al. Hash-based IP traceback. In: Proceedings of the ACM SIGCOMM conference; Aug. 2001. p. 3—14.

Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F, Schwartz B, et al. ''Single-packet IP traceback''. IEEE/ACM Transactions on Networking 2002;10(6):721—34.

Song D, Perrig A. Advanced and authenticated marking schemes for IP traceback. In: Proceedings of IEEE INFOCOM conference; Apr. 2001. p. 878—86.

Sung M, Xu J. IP traceback-based intelligent packet filtering: a novel technique for defending against internet DDoS attacks. In: Proceedings of international conference on network protocols; Nov. 2002. p. 302—11.

Sung M, Xu J. IP traceback-based intelligent packet filtering: a novel technique for defending against internet DDoS attacks. IEEE Transactions on Parallel and Distributed Systems Sep. 2003;14(9):861—72.

Theilmann W, Rothermel K. Dynamic distance maps of the internet. In: Proceedings of the IEEE INFOCOM, vol. 1; Mar. 2000. p. 275—84.

Yaar A, Perrig A, Song D. Pi: a path identification mechanism to defend against DDos attacks. In: Proceedings of the IEEE symposium on security and privacy; May 2003. p. 93—109.

Shiuhpyng Shieh is a Professor and former Chairman of Department of Computer Science and Information Engineering of National Chiao Tung University. He is also the president of Chinese Cryptology and Information Security Association (CCISA), which is the largest and a highly respectable academic organization on information security research in Taiwan. He has worked as advisor to many institutes, such as National Security Bureau, GSN-CERT/CC, National Information and Communication Security Task Force. Before joining NCTU, Dr. Shieh participated in the design and implementation of the B2 Secure XENIX at IBM, Federal Sector Division, Gaithersburg, Maryland. He also designed and developed NetSphinx, a network security product, for Formosoft Inc., which is awarded 1999 network product of the year, Taiwan.

Dr. Shieh received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park. He is a senior member of IEEE, and an editor of ACM Transactions on Information and System Security, Journal of Computer Security, and Journal of Information Science and Engineering. He was in the organizing committees of numerous conferences, such as ACM conference on Computer and Communications Security, IACR Asiacrypt. Dr. Shieh published over a hundred academic articles, including papers, patents, and books. Recently he received the Outstanding Research Award from National Chiao Tung University for his academic achievement in research, and the Outstanding Achievement Award from State Department of Taiwan. His research interests include internetworking, distributed operating systems, and network security.

Fu-Yuan Lee received the BS degree in computer science from National Chiao Tung University in 1998. He is currently a Ph.D. student in the Department of Computer Science and Information Engineering at National Chiao Tung University. His research interests are in the areas of computer networks and network security.