Distributed Packet Filtering Based on IP Traceback

Kobra Khanmohammadi\(^1\) and Mehran S. Fallah\(^2\)

\(^1\) Kobra Khanmohammadi is an M.S. student in the department of computer and IT engineering, Amirkabir University of Technology (AUT), Tehran, Iran, Email: kmohammadi@ce.aut.ac.ir

\(^2\) Mehran S. Fallah is with the department of Computer and IT engineering, Amirkabir University of Technology (AUT), Tehran, Iran, Email: fallah@ce.aut.ac.ir

Abstract—The IP traceback is a reactive mechanism for finding the origin(s) of a flooding denial-of-service (DoS) attack. In order to defeat a DoS attack or alleviate its harmful effects, the IP traceback should be integrated with some complementary defense mechanisms. In this paper, we propose a distributed packet filtering scheme based on the information provided by an IP traceback mechanism. In this method, a number of routers, located on the attack path, filter out the packets probabilistically. The victim uses the information provided by the traceback mechanism to determine the routers that should participate in defense as well as the probability of filtering for each router. The method proposed in this paper differs from the existing IP traceback-based filtering mechanisms in two ways: the method can defeat bandwidth exhaustion attacks; and its filtering is only based on the packets’ destination addresses. The simulation results show the effectiveness of the proposed method.

Key Words—Distributed packet filtering, flooding denial-of-service attacks, IP traceback

1. INTRODUCTION

The IP traceback is a method for locating the origins of a flooding DDoS attack. If the victim knows the sources of the attack, he can defeat the attacker by discarding the packet received from those sources. Therefore, in these attacks, the attacker usually uses spoofed IP addresses to conceal his location(s) on the network from the victim. The current IP traceback mechanisms can find the origins of the attack in a high confidence level. Nonetheless, these mechanisms alone cannot completely defeat the attack, because the time needed for an IP traceback mechanism to decide the origins of the attack is long enough for the attacker to preempt some of the key resources at the victim. Thus, the IP traceback should be integrated with some complementary defense mechanism.

The literature witnesses efforts in combining the IP traceback and filtering aiming at a complete solution to the flooding DDoS attacks \([1, 2]\). In the proposed solutions, the filtering mechanism uses the information obtained from the IP traceback to obstruct the malicious packets more accurately, and therefore, to improve the effectiveness of the method. In these approaches, both IP traceback and filtering are based on packet marking, and the victim uses the marks to decide if a specific packet should be discarded. Moreover, as filtering is done at the end of attack paths, the advertent packets deplete the bandwidth terminating at the victim.

This paper proposes a filtering mechanism which is based on IP traceback and performs the filtering in a distributed manner using some of the routers along the attack paths. In this way, the bandwidth exhaustion attacks can be thwarted. In this approach, the victim deploys the information obtained from IP traceback to select the routers which should participate in filtering. The victim also determines the probability of filtering for each of the selected routers. This probability is calculated on the basis of the distance between the router and the victim as well as the ratio of the packets traversing that router in their journey to the victim.

2. DISTRIBUTED IP TRACEBACK-BASED FILTERING

In the method proposed in this paper, some of the routers on the attack path check the destination addresses of incoming packets, and then filter out the ones targeting at the victim. This filtering is done in a probabilistic manner based on the information provided by the victim. The victim runs an IP traceback mechanism and develops the attack tree concurrently. At any time, before completing the traceback, the victim has some information about the attack tree, though he may not definitely know the leaves (attack origins) of this tree. Indeed, at a given time during the execution of traceback, the victim has developed
a part of the tree which terminates at a number of nodes called leaf routers.

The proposed scheme prefers the filtering to be performed in leaf routers. Nonetheless, this may be impractical in some actual cases because the leaf routers are not equipped with filtering options, or their administrators do not collaborate in the defense scheme. Therefore, a mechanism is required for the victim to select other routers in these cases. Our proposed scheme resolves this by choosing the router that is the nearest one to the corresponding leaf in the attack tree developed so far. This selected router should have the capabilities required for filtering. Otherwise, the procedure above is repeated until a suitable router is selected. The filtering procedure is terminated at a selected router when the following events occur. First, the victim requests the router to terminate the filtering procedure as a result of choosing another filtering router in the same attack path. Second, a specific time period has been elapsed after hiring the router to do filtering. The length of this time period is proportional to the time the victim requires to develop the entire attack tree. It is worth noting that an authentication protocol is also required to support secure communications between the victim and the filtering routers. There are suggestions for such a protocol in the literature, e.g., Intrusion Detection Exchange Protocol [11]. In our proposed mechanism, the existence of such a protocol is assumed. Later, in the paper, we elaborate the proposed scheme in a more formal manner.

The proposed scheme has many advantages, listed below, to the IP traceback-based filtering mechanisms presented in [1] and [2] (to the best of our knowledge, these are the only earlier works in this area.)

- In [1], there exist two types of packet marks; signaling and data. The former is used in IP traceback, while the latter is for filtering. As signaling marks appear in one-twentieth of the marked packets, the construction of attack tree takes twenty times the time taken by the original IP traceback where it is deployed alone. In our proposed method, all the marks are used in IP traceback, and the said shortcoming is resolved.
- In [1], only 5% of marks are signaling, the marks are used for IP traceback. This implies that the reconstruction of the whole attack graph would be 20 times slower than the underlying IP traceback scheme. In the approach presented here all the marks are used for IP traceback.
- The probability of receiving a packet marked $i$ hop away by the victim with the marking probability $p$ is $p(1-p)^{i-1}$. Number of packets have the mark of router $i+1$ are $(1-p)$ times less than the number of packets which have the mark of router $i$. So by filtering packets by checking their mark ([1], [2]), more marked packets of router $i$ would be filtered out. Also, the normal paths and attack paths overlap more near the victim. In this approach we filter packets in routers through the attack path without attention to their mark. So normal packets that their part of path overlaps with an attack path in edges near the victim would be safe from filtering.
- By filtering packets in upstream routers near the victim (perimeter routers) [1], only we could prevent from entering packets to victim system. The bandwidth between attacker and perimeter router would be consumed by the attack traffic. So the bandwidth, an important resource of network, would be unavailable for legitimate users. In this approach by doing filtering in routers in attack path, we can prevent wasting the bandwidth of the network.
- Filtering was done in [2] after the IP traceback mechanism was finished. Our approach like [1] does filtering during the IP traceback. It would mitigate the effect of an attack while it is raging on and help the victim to stand by until the completing the IP traceback mechanism.
- In our approach filtering packets are independent from IP traceback approach so in real implementation it is better to use all other improvement that papers ([8], [9], [10]) suggest for packet marking.

3.1. IP TRACEBACK APPROACH

Like the approach in [1] we suggest AMS [6] scheme to be used for IP traceback. AMS is faster than other packet marking schemes like a scheme...
presented in [4] and has construction time and less false positive. Also because of distance field in its mark, it is safe for *pollution attacks*. In a *pollution attack*, the attacker sends malicious fragments that interfere with path reconstruction [12]. The distance field allows marking injection only at distances greater than the closest attacker. However, large numbers of legacy routers reduce this immunity because distance is only counted in terms of marking-enabled routers.

Packet marking approaches like FIT [10] have more advantages to AMS, for example they can perform traceback even after a very small number of attack packets with minimal processing overhead and without contacting any external entities. In addition, FIT handles legacy routers better than any previous mechanism, as a victim can even detect the presence of legacy routers on the attack path. FIT has some other advantages but in this scheme we use AMS to compare our approach with the schemes that filter packets by using the information obtained from IP traceback like [2]. Also T. Peng et al. [8] suggest a dynamic probability for marking packets in router instead of marking packets with constant probability. We just prefer constant probability because of the reason we don’t use FIT approach.

At all, in our approach, filtering packets are independent from IP traceback scheme, so in real implementation it is better to use all other improvements that papers ([8], [9], [10]) suggest for packet marking.

### 3.2. FILTERING PACKETS IN Routers

Each leaf router would filter out packets by checking their destination IP. For each leaf router, the probability for filtering is depends on these parameters:

1. **Distance from leaf router to the victim:** distance is the hop count form leaf router to the victim. The probability of receiving a packet marked \( i \) hop away by the victim with the marking probability \( p \) is \( p(1-p)^{i-1} \). So in attack graph for edges near the victim, the probability of being this edge in the path of legitimate packets is more than the farther edges. Therefore for packets that have the mark of farther router in attack path, the probability that it would be the attack packet is more than the packet that have the mark from a nearer router. Therefore filtering probability must be more in the farther router in attack path. We show this distance by \( h_i \).

2. **Number of attack packets comes from routers is difference in leaf routers.** So the filtering probability must be different. In routers that the number of attack packets come from them is more, the filtering probability must be more. The victim could compute the percent of attack packets come from node \( i \) \( (w_i) \), from the number of attack packets that have the mark of node \( i \) to all the marked attack packets.

If \( n \) be the leaf router in the snapshot, and the goal be that \( p \% \) of all packets that come to the victim and must be filtered out, it means a packet would be filtered out with probability \( p \). this probability would be decided by the victim. So the filtering probability for each leaf router would be:

\[
p_i = \frac{1}{C - h_i} \cdot w_i \cdot \alpha_i
\]

\( C \) is the number of hop count in the average of packets path. Since most path lengths in the Internet are bounded by 30 [14] [15]. We can take \( C = 30 \) for safety. \( \alpha_i \) is a variable parameter that is calculated from the value, \( p \).

Sum of the \( p_i \) must be equal by \( p \).

\[
\sum_{i=1}^{n} p_i = p
\]

If we suppose that, \( i = 1, \ldots, n \), have the same value, we could find one value for \( \alpha_i \) from the equation (2) and then find the probability for each router by the equation (1).

Victim must compute these probabilities and then send the probability for each router. Figure 1 shows the process is done in victim. It shows that the victim done the IP traceback mechanism and after a period of time finds the filtering probability for each router. Also, figure 2 shows the algorithm for marking and filtering packets in routers.

### 3.3. ANALYSES

The first set of routers could start filtering after a period of time that used for IP traceback and the time needed for making configuration in routers. For the upper bound of the time needed for computing the IP traceback we suppose the
IP traceback process:
generate attack tree
after a period of time
partial-attack-tree:=snapshot of attack tree

Compute filtering probability for each leaf-router
let \( r_i \) is the rate of packets receive from router \( i \)
let \( h_i \) is the hop count distance of router \( i \)
let \( P \) is the filtering probability for all / received packet that initiated by the victim
let \( p_i \) is the filtering probability of router \( i \) initiated to 0

\[
\alpha = \frac{r_i}{\sum \left( \frac{r_i}{r} \cdot \frac{1}{C - h_i} \right)}
\]

for each router \( i \) is a leaf-router do
\[
p_{i} := \alpha \cdot \frac{r_i}{r} \cdot \frac{1}{C - h_i}
\]
send \( p_i \) to leaf router \( i \)

Figure 1. IP traceback and computing filtering probability in victim

let \( P \) represent the filtering probability for \( R \) and equal to 1 in default
let \( Q \) represent the markings probability
let \( mch \) represent the selected marking mechanism
for each packet \( pkt \) do

generate a random number \( x \) within \([0..1]\]
if \( x < P \) then
\( \text{drop} \ pkt \)
else

generate a random number \( x \) within \([0..1]\]
if \( x < Q \) then
\( \text{mark} \ pkt \) with mechanism \( mch \)

Figure 2. Marking and filtering algorithm in router \( i \)

filtering probability is constant and equal to \( \rho \). So packets are filtered out by probability \( \rho \) in routers. If packets come with the average rate \( R \) to the victim, after starting filtering in the first set of routers, packet comes with the rate \( \rho R \).

If IP traceback is completed after \( k \) period, and \( t \) is the time of completing IP traceback in AMS [6], then the maximum time needed for IP traceback is \( \sum_{i=0}^{k-1} \frac{\rho^i}{k-i} \).

The time needed for IP traceback is less than this in reality because some time is needed for making configuration in routers and in this time packets come with rate \( R \) and the victim receives more attack marked packets.

If a packet filtered in this scheme, precisely means that the packet passes from one of the leaf routers in the attack graph and is filtered by the probability of filtering in a passed leaf router. So the probability of filtering for each packet that comes toward the victim is equal by the probability of this packet pass from one of the leaf routers and filtered out by a leaf router. We suppose that \( L_n \) the number of leaf routers in distance \( n \), and \( P_i \) is the probability of filtering packet by the leaf router \( i \). The probability \( P_f \) would be the probability that a packet filtered by one of the leaf routers. If we assume the maximum distance of leaf routers as a distance of leaf routers then the probability of filtering packet by one of the leaf routers would be \( P_f = \sum P_i \cdot \frac{1}{L_n} = P \cdot \frac{1}{L_n} \).

In paper [9] Al-Duwairi et. al shows that the approximate degree distribution in 80% of routers is 4. So we can assume that the average number of routers in distance \( n \) be \( 4^n \). Therefore the approximate calculation of \( P_f \) would be \( \frac{P}{4^n} \).

In AMS [6] number of packets for reconstructing path with distance \( d \) have the upper bound equal to \( E(x) < \frac{\ln(d)}{p(1-p)^{d-1}} \). In our scheme, if we assume the maximum probability for a packet to filter out in its path, then the maximum number of packets is needed for reconstructing a path with distance \( d \) is \( E(x) < \frac{\ln(d)}{(1-\rho)p(1-p)^{d-1}} \).

4. SIMULATION RESULTS
In this paper we use the real internet topology dataset provided by CAIDA’s Skitter map [13]. This dataset is a collected topology originated from a CAIDA-owned host on 04/21/2003. The data collection process is a part of CAIDA’s Skitter project. Note that all topologies are routes from a single origin to multiple hosts on the Internet. In our simulations, we assume that this origin is the victim and the attackers and legitimate clients are randomly distributed among the destination hosts in the topologies.

The simulated metric of our scheme is not dependent on the actual values of the parameters concerning bandwidth and data rates but rather on their relative ratios, we will use abstract “units” for characterizing these parameters instead of the actual rates. We assume that each legitimate user sends at the rate of 1 unit per second and fix the number of legitimate clients to 100 and the bandwidth of the victim site is set to 100 units so that it is 100% utilized by legitimate traffic when there is no attack. For performance evaluation purposes, we also assume that each attacker attacks at the same rate (not needed for the correctness of our scheme). The number of attackers will be a function of the severity of the attack and the bandwidth per attacker, to be discussed later.

The good traffic percentage (GTP) is the metric we would like to improve, which is the arrival rate of the legitimate traffic at the victim divided by the victim bandwidth. GDP represent the percentage of legitimate traffic that is dropped to the entire dropped packet.

In our simulation, routers mark packets with probability 0.04 that is suggested by [4] and [6]. The parameter $g$ represents the percentage of incoming traffic being legitimate and $b$ is the ratio of the rate of an attacker over the rate of a legitimate host. Note that when $g$ and $b$ are set, the number of attackers is calculated as $100 \ast (1 - g) / b \ast g$ (recall that there are 100 legitimate hosts). We use $b=40$, so the number of attackers are 10.

Figure 3 show that by incrementally reconstruction of attack graph, the percentage of legitimate traffic is incremented. Because filtering packet is done in leaf router of reconstructed attack graph. In some distance, GTP change is more, it might be because of finding the edge between the core node and the connected node to this core node in attack path. Overlap of the attack path and legitimate packet path would decrease in these edges. Figure 4 show that how GDP is decrement by attack path reconstruction completed. In section 3 we calculate the $P_f$. In real $P_f$ is equal to $P \ast GDP$.

5. CONCLUSIONS AND FUTURE WORKS

This paper presents intelligent distributed packet filtering by using information obtained from IP traceback. In this scheme IP traceback is started as soon as an attack is detected and during the IP traceback, filtering is done probabilistically in routers through the attack path by filtering out packets which are sent to the victim. The probability for filtering in each router is related to the distance and rate of attack packets which pass this router. This relation would help decrease false drops.

We compute the time needed to complete IP traceback and the packets needed by comparing this scheme with the scheme presented in [6]. This scheme is distributed and leads to save bandwidth between the attacker and the victim. Because of the independence of IP traceback approach and filtering mechanism in this scheme, it has capability for all the improvement suggested in papers for packet marking approach in IP traceback.

If we suppose that the network is divided to some parts and each part has the responsibility to
do IP traceback in its part when the attack starts in each part, IP traceback will be faster because the pass length in each part will be smaller. SPIE [3] is the logging scheme for IP traceback that models the network's parts. So we could present a new distributed IP traceback based packet filtering in a way that each part of the network is responsible for IP traceback and filtering in its part. We are going to try to achieve this scheme in our next work.

6. REFERENCES


