A Stateless Scheme for Single-Packet IP Traceback

Rafael P. Laufer[1] and Pedro B. Velloso[1,2]
Advisor: Otto Carlos M. B. Duarte[1]

[1]Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro
Rio de Janeiro, RJ, Brazil

[2]Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie - Paris VI
Paris, France

I. Overview

On most denial-of-service (DoS) attacks, packets with spoofed source addresses are employed in order to disguise the true origin of the attacker. A defense strategy is to trace attack packets back to their actual source in order to make the attacker accountable and isolate him from the network. To date, the proposed traceback systems require either large amounts of storage space on router-connected devices or a sufficient number of received attack packets. We propose an IP traceback system capable of determining the source of every packet received by the victim without storing state in the network infrastructure. For practical purposes, a generalization of the Bloom-filter theory is developed and evaluated.

II. The Proposed IP Traceback System

We present a new IP traceback technique designed to trace the source of each individual packet. The proposal is based on the packet-marking approach to avoid storing state at routers. Each router inserts a "signature" into the packet, which indicates its presence on the path. A Bloom filter [1] is employed to reduce the amount of information inserted into the packet and to limit the size of this information to a fixed value to avoid packet fragmentation. Therefore, the attack route is stored in a built-in Bloom filter integrated into the packet. In addition, we propose the use of a generalized Bloom filter (Section II-A) to prevent "signature" forgery by the attacker and therefore backtracing failures.

The marking algorithm for this case is quite simple. Just before forwarding a packet, the router inserts the IP address of its output interface into the filter of the packet. To reconstruct the attack path, the following procedure is used. Initially, the victim checks for the presence of all neighbor routers in the Bloom filter of the received attack packet. The router that is recognized as an element of the filter is identified as the upstream router and is, therefore, integrated into the attack path. Afterwards, this selected router receives the Bloom filter from the victim and checks which neighbor router is also recognized as an element of the filter, identifying the next upstream router. This process is recursively repeated on each upstream router to reconstruct the actual path traversed by the packet. When a router does not recognize any neighbor router as an element of the filter, the process stops and this router may be considered the source of the attack.

Some advantages come from the adoption of this approach. First, the complete route of each packet can be individually determined. Such behavior is idealized by every IP traceback system since it permits the identification of each source of a distributed attack, even if it contributed with only one packet. By enabling backtracing of a single packet, the system becomes as scalable as it can be. Besides, no information needs to be stored in the network infrastructure. All traceback data is stored at the victim, who chooses to hold it or not according to the local security policy. Another advantage is the ability of tracing an attack long after it is over and without any help from network operators.

On the other hand, additional processing overhead is introduced during each packet routing. Moreover, the adoption of a Bloom filter introduces false positives into the attack path. During the reconstruction procedure, a false positive implies the incorrect integration of a router into the attack path. If this probability is small enough, false positives do not significantly impact on the reconstruction. There would be some concurrent routes for the same packet but the set of possible attackers would still be reduced. Nevertheless, since the attacker controls the initial content of the packet, he can fill all the filter bits with 1. By saturating the filter, every router is integrated into the attack path during the reconstruction procedure, making impractical to distinguish the real path.

In order to minimize misleading techniques and to make the system less dependent of the initial state of the filter, we propose a generalization of the Bloom filter. The basic idea of the generalized Bloom filter is to employ both hash functions that set and hash functions that reset bits. We show that with the generalized Bloom filter the false-positive probability is reduced and it depends little on the initial condition of the filter. On the other hand, false negatives, which do not exist in standard Bloom filters, are now introduced.

A. The Generalized Bloom Filter

As the standard filter, the generalized Bloom filter is also a data structure used to represent a set $S = \{s_1, s_2, \ldots, s_n\}$ of n elements in a compact form. It is constituted by an array of m bits and by $k_0 + k_1$ independent hash functions $g_1, g_2, \ldots, g_{k_0}, h_1, h_2, \ldots, h_{k_1}$ whose outputs are uniformly distributed over the discrete range $\{0, 1, \ldots, m-1\}$. The generalized filter is built in a similar way to the standard filter. Nevertheless, the initial value of the bits of the array is not restricted to 0 anymore. In the generalized Bloom filter, these bits can begin with any value. For each element $s_i \in S$, the bits corresponding to the positions $g_1(s_i), g_2(s_i), \ldots, g_{k_0}(s_i)$ are set to 0 and the bits corresponding to the positions $h_1(s_i), h_2(s_i), \ldots, h_{k_1}(s_i)$ are set to 1. In the case of a collision between a function g_i and a function h_j within the same element, we arbitrate that the bit is always set

to 0, $\forall i, j$. The same bit can be set to 0 or 1 several times without restrictions. After inserting the elements, membership queries can be easily made. To check if an element x is in S, we check if the bits of the array corresponding to the positions $g_1(x), g_2(x), \ldots, g_{k_0}(x)$ are all set to 0 and if the bits $h_1(x), h_2(x), \ldots, h_k(x)$ are all set to 1. If at least one bit is inverted, then $x \notin S$ with high probability. In the generalized Bloom filter, it is possible that an element $x \in S$ may not be recognized as an element of the set, creating a false negative. Such anomaly may happen when at least one of the bits $g_1(x), g_2(x), \ldots, g_{k_0}(x)$ is set to 1 or one of the bits $h_1(x), h_2(x), \ldots, h_{k_1}(x)$ is set to 0 by another element inserted afterwards. On the other hand, if no bit is inverted, then $x \in S$ also with high probability. This uncertainty is explained by the fact that an element $x \notin S$ may be recognized as an element of the set, creating a false positive. A false positive occurs when the bits $g_1(x), g_2(x), \ldots, g_{k_0}(x)$ are all set to 0 and the bits $h_1(x), h_2(x), \ldots, h_{k_1}(x)$ are all set to 1 due to a subset of elements of S or to the initial condition of the bit array.

III. Preliminary Results

In order to show the advantages of employing a generalized Bloom filter to represent the attack path, we present an analytical comparison between the two versions of our scheme. We compare a simple version that uses the standard Bloom filter and the extended version that uses the new concept of generalized Bloom filter. The analysis comprises three different aspects: false positives, false negatives, and interference of the attacker.

During the path reconstruction procedure, a false positive implies the integration of an incorrect router into the attack path. Thus, the higher the false-positive probability, the greater the number of possible routes from which the packet may have come, which makes harder the attacker identification.

Figure 1 shows the false-positive probability of a generalized Bloom filter f_p as a function of $p_1(n)$, which is defined as the fraction of bits that are marked as 1 after inserting n elements. For the standard Bloom filter (curve $k_0 = 0$), we can notice that the false-positive probability grows as $p_1(n)$ increases, as expected. The other curves represent the generalized Bloom filter. We can observe that for $p_1(n) = 0$ and $p_1(n) = 1$ the false-positive probability equals zero. It occurs because, when we are using at least one function of each type, it is required at least one bit marked as 0 and one bit marked as 1 to have a false positive.

Different from the standard Bloom filter, our generalized version has a bounded maximum false-positive probability. We show that the maximum false-positive probability of a generalized Bloom filter is approximately

$$f_p^{max} = \left(\frac{k_0}{k_0 + k_1}\right)^{k_0} \left(\frac{k_1}{k_0 + k_1}\right)^{k_1}, \qquad (1)$$

which is exclusively determined by system parameters k_0 and k_1 [2]. This characteristic can restrict the attacker interference in the traceback process, as follows.

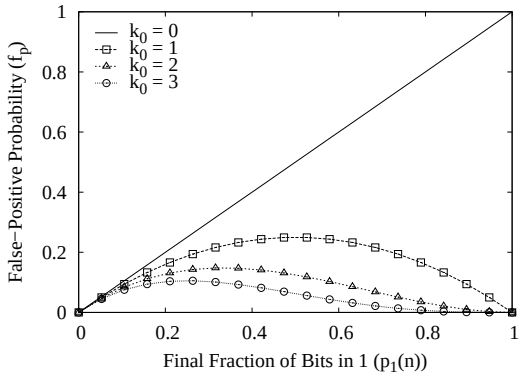

Fig. 1. False-positive probability of a generalized Bloom filter as a function of the final fraction of bits in 1, for $k_1 = 1$.

Since the filter is integrated into the packet, the attacker may interfere in both standard-filter and generalized-filter systems by setting the initial condition of the filter. In a standard Bloom filter, the false-positive probability may reach 100% when the attacker just fills with 1 the bits of the packet corresponding to the filter. Nevertheless, when we use a generalized Bloom filter instead of a standard filter, the interference of the attacker is considerably reduced. For instance, the maximum false-positive probability drops at least 75%, for the worst case where $k_0 = k_1 = 1$.

On the other hand, using a generalized Bloom filter might lead to false negatives during the attack-path reconstruction procedure. A false negative means not detecting a router by which the attack packet has passed. Therefore, just one false negative is enough to stop the reconstruction procedure and avoid finding the real attack path. It is worth mentioning that the attacker cannot interfere in the false-negative probability.

IV. Future Directions

Probabilistically, the proposed system is able to trace an attack back to its source analyzing a single packet. Thus, our approach is scalable and fits well to trace each source of a distributed DoS attack. A generalization of Bloom filters is proposed and employed in the system. We show that, with the generalized Bloom filter, the misleading ability of the attacker in the traceback procedure is drastically reduced. The tradeoff cost is the introduction of false negatives in the system. As future works, we are studying new path-reconstruction procedures that reduce the false-positive probability.

Acknowledgment

This work has been supported by CNPq, CAPES, FAPERJ, FINEP, RNP, and FUNTTEL.

References

[1] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Communications of the ACM*, vol. 7, no. 13, pp. 442–426, July 1970.

[2] R. P. Laufer, P. B. Velloso, D. de O. Cunha, I. M. Moraes, M. D. D. Bicudo, and O. C. M. B. Duarte, "A New IP Traceback System Against Distributed Denial-of-Service Attacks," Grupo de Teleinformática e Automação, COPPE/UFRJ, Tech. Rep., Jan. 2005, available at: http://www.gta.ufrj.br/~rlaufer/publications/TechTraceback.pdf.