Design of IP Traceback System based on Generalized Bloom Filter

Zhouguo Chen, Bo Liu, Shi Pu and Chen Huang
Science and Technology on Communication Security Laboratory
Chengdu, China
e-mail: czgexcel@163.com, boliu1017@163.com, 271193918@qq.com

***Abstract*—With the rapid development of Internet, network security technology has recently attracted a lot of interest, both from academia and industry. By tracing and locating the source of network intrusion, IP traceback technology help defenders to take targeted defensive measures, counter attackers and collect information for cyber forensics. This paper first introduces the basic method of IP traceback, then introduces the theory of Generalized Bloom Filter (GBF) and its application in packet marking. After that, design IP Traceback System based on GBF (For short IPTSGBF) and experimental results show that the system can success reconstruct attack path. Finally summarize this paper and outline future's research directions.**

Keywords- IP traceback; Generalized Bloom Filter

I. INTRODUCTION

As the use of e-commerce and dot-coms continues to expand in various spheres of life; the security breaches in the systems are also expanding correspondingly. Internet has played a very important role in the business, culture, politics etc. So the security of Internet has arrest people's attention. In the last several years, Internet attacks such as DDoS, Worm, have increased in frequency, severity and sophistication, which could hide their identity behind legitimate users or spoof the source address of the packets. Unfortunately, the traditional mechanisms for dealing with it, e.g. IDS, have not advanced at the same pace, not been availability. IP Traceback can be defined as "determining the identity or location of an attacker or an attacker's intermediary". By tracing and locating the source of network intrusion, IP traceback technology help defender take targeted defensive measures, counter-attack attacker and collect information for cyber forensics.

The rest of this paper is organized as follows: 1)IP Traceback techniques are introduced, 2) the Generalized Bloom Filter (GBF) algorithms are presented, and analysis it performance, 3) then IP traceback system design based on GBF, 4) Finally, the conclusion is presented.

II. RELATED WORK

There have been several efforts to reduce the anonymity afforded by IP spoofing. Table 1 provides a subjective characterization of each of these approaches in terms of management cost, additional network load, overhead on the router, the ability to trace multiple simultaneous attacks, the ability trace attacks after they have completed, and whether they are preventative or reactive.

A. Link testing

Link testing [1] is a traceback technique which is based on testing the links connected to the routers on the upstream direction of the victim. It is simply assumed that the attack is still active during the traceback process. Therefore, it navigates through the links on the upstream direction until the source (attacker) is identified. Input debugging and controlled flooding are the two types of link testing technique.

B. Logging

This approach is based on logging the packets on the routers (note: applied to specified routers). By using data mining techniques, the path can be reconstructed. Tracing back the attacker could be carried out at any time after the attack, which is considered to be an advantage for this technique. The famous way of logging is SPIE[2]. To save the cost of storage, Traffic auditing is accomplished by computing and storing 32-bit packet digests rather than storing the packets themselves. Every packet traversing a SPIE-enhanced router is recorded in a digest table; digest tables are paged at a specified rate and are representative of the traffic forwarded by the router during a particular time interval. A cache of digest tables is maintained for recently forwarded traffic.

C. ICMP Traceback

It is proposed by Bellovin [3], as a packet traverses through the network, each router probabilistically generates a separate trace packet called Internet Message Control Protocol (ICMP). To keep a control on the overhead and the number of ICMP packets, the router generates an ICMP packet for only one in 20,000 packets that passes through it. As the packet traverses through routers, it collects useful path information on its way to its destination. The destination tries to glean path information from all the ICMP packets emitted by the chain of routers along a given path and hence can infer the true source.

D. Packet Marking

Packet marking is based on inserting marks (information about the intermediate routers) in the packets [4] [5]. These marks would help the victim to reconstruct the path to identify the source of the attack. Tracing the attack by marking the packets is mentioned by Burch and Cheswick at [6]. Packet marking can vary according to the method or algorithm is used. In general, there are two forms of packet marking: Probabilistic Packet Marking (PPM)[7] and Deterministic Packet Marking (DPM)[8].

TABLE I. QUALITATIVE COMPARISON OF EXISTING SCHEMES FOR IP TRACEBACK

	ISP cooperation	***Need Modifying Router***	***Amount of data***	***carried out after the attack***	***Speed of Traceback***	***Router Overhead***
Link Testing	High	Yes	Little	Poor	Slow	High
Logging	Low	No	Little	Good	Fast	High
ICMP Traceback	Low	Yes	More	Good	Slow	Low
PPM	Low	Yes	More	Good	Slow	Low
DPM	Low	Yes	Little	Good	Fast	Low

III. GENERALIZED BLOOM FILTER

A. *Theory of Generalized Bloom Filter*

A Bloom Filter (BF) is a method for representing a set A = {a_1, a_2, … a_n} of n elements (also called keys) to support membership queries. It was invented by Burton Bloom in 1970 [9]. Bloom Filters have been used in many different computer science areas, including spell checkers, database applications and networking. The idea of BF, illustrated in Figure 1, is to allocate a vector v of m bits, initially all set to 0, and then choose k independent hash functions, h_1, h_2,… h_k, each with range {1, …, m}. For each element a∈A, the bits at positions h_1(a), h_2(a), ⋯, h_k(a) in V are set to 1. A particular bit might be set to 1 multiple times. Given a query for b we check the bits at positions h_1(b), h_2(b), ⋯, h_k(b). If any of them is 0, then certainly b is not in the set A. Otherwise we conjecture that b is in the set although there is a certain probability that we are wrong. This is called a false positive. The parameters k and m should be chosen such that the probability of a false positive is acceptable.

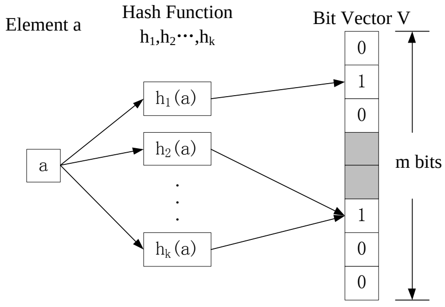


Figure 1. Insertion of an element into a Bloom Filter

Generalized Bloom Filter (GBF) is invented by Rafael P. Laufer, etc.[10]. As the standard filter, the Generalized Bloom Filter is also a data structure used to represent a set A = {a_1,a_2,...,a_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,...,g_{k0}, h_1,h_2,...,h_{k1} whose outputs are uniformly distributed over the discrete range {0,1,...,m - 1}. The GBF is built in a similar way to the standard filter. Nevertheless, the initial value of the bits of the array is not restricted to zero anymore. In the GBF, these bits can be initialized to any value. For each element a_i ∈ A, the bits corresponding to the positions $g_1(a_i)$,$g_2(a_i)$,...,$g_{k0}(a_i)$ are reset and the bits corresponding to the positions $h_1(a_i)$,$h_2(a_i)$,...,$h_{k1}(a_i)$ are set. In the case of a collision between a function g_i and a function h_j for the same element, we arbitrate that the resulting bit in the filter is always reset. The same bit can be set or reset several times without restrictions. Fig. 2 depicts how an element is inserted into a GBF. After inserting the elements, membership queries can be the same to BF. To check if an element x belongs to A, we check if the bits of the array corresponding to the positions g_1(x),g_2(x),...,g_{k0}(x) are all reset and if the bits h_1(x),h_2(x),...,h_{k1}(x) are all set. If any one bit is inverted, then x ∉ A is high probability. In the GBF, it is possible that an element x ∈ A may not be recognized as an element of the set, creating a false negative. Such anomaly happens when at least one of the bits g_1(x), g_2(x),...,g_{k0}(x) is set or one of the bits h_1(x),h_2(x),...,h_{k1}(x) is reset by another element inserted afterwards. Also, it is possible that an element x ∉ A may be recognized as an element of the set, creating a false positive.

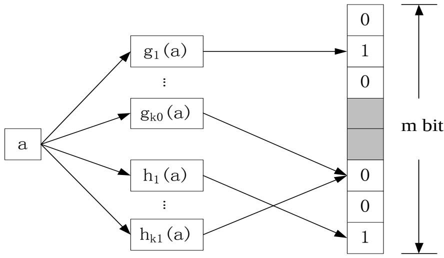


Figure 2. Insertion of an element into a Generalized Bloom Filter

The main advantage of the Generalized Bloom Filter is that both the false-positive and the false-negative ratios are upper bounded. In addition, these upper bounds depend only on the chosen parameters of the filter and not on its initial condition. As a result, it can be used in distributed applications that require a secure and space-efficient set representation. See the reference [11] for details.

B. *Performance of GBF*

To IP Traceback, the false positive means a benign network node (e.g. Router and PC) is recognized as a malicious node in the attacking path. While false negative means a malicious node is not identified. So we should reduce the value of false positive and false negative to improve the efficiency of IP Traceback.

Usually, the space of storage is more lager than the amount of HASH functions, i.e. m>>k_0 and m>>k_1. So when $k=k_0=k_1$ the false-positive probability F_p is maximum [10], the expression as follow:

$$F_p= (1/4)^k \quad (1)$$

The maximum false-positive of GBF is determined by the amount of HASH functions. It can be further reduced if a larger number of hash functions is used, be independent of space of storage m. When $k=k_0=k_1=2$, have a maximum false-positive probability of only 6.3%. If $k_0 = k_1 = 3$ and $k_0 = k_1 = 4$, the maximum false-positive probability drops to 1.6% and 0.4%.

The false-negative probability of GBF is a monotonically decreasing function of the number of inserted elements [10]. Fig3. depicts the behavior of the Generalized Bloom Filter when we increase the number of hash functions. Clearly, the false-negative probability of the first elements increases significantly. It happens because with more hash functions it is more likely that the bit markings of the first elements get inverted by the last elements.

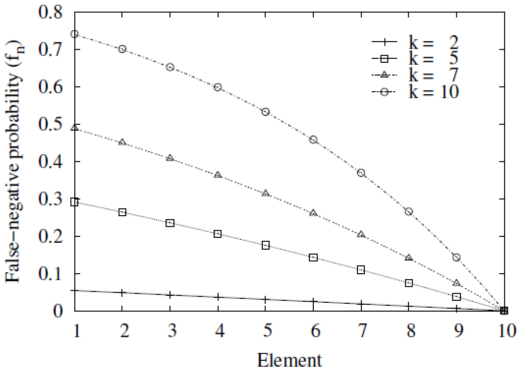


Figure 3. False-negative probability of a GBF for each inserted element

We can think of the GBF as a "memory" buffer where first inserted elements are more likely to be "forgotten" or overwritten by the last ones. This effect depends not only on the number of hash functions used, but also on the relative size of the filter. The intuition behind this last statement is that a larger filter implies a larger output range for the hash functions. With a larger range, the probability of one element overwriting the bits of previous elements decreases.

IV. IP TRACEBACK SYSTEM BASED ON GBF

A. Choice Parameter of GBF

We have proved that the false-positive of GBF can be reduced as a larger number of hash functions used, and the false-negative can be increased while a larger parameters k is used. However the larger value k is, the more number of computing is used. As a result, it is important to choose the appropriate number of hash functions that provide the ideal tradeoff between false positives and false negatives. And we should require the more little data in IP Traceback to avoid network overhead.

Design an IP Traceback system based on GBF, we should determine the number of hash functions (k), and the space to store the value of GBF in IP packet. When $k=k_0=k_1=2$, have a maximum false-positive probability of only 6.3%. And the count of HASH calculating is accepted. So the HASH function is defined as followed:

$$h_{c,d}(x)=[(c[id]x+d[id]) \bmod z] \bmod m \tag{2}$$

According to Eq. (2), x is the value of IP address, z is a prime number which is 4294967291, m is the space of memory in GBF. To change a id of HASH function, we can have many different HASH functions, that $0 <= id < k_0+k_1$. When $0 <= id < k_0$, the bits of the array corresponding to the positions $h_{c,\,d}(x)$ are all reset, while $k_0 <= id < k_0+k_1$, the bits are set. We define that c[id]=2*id+1, d[id]=2*id+2.

In this IP Traceback System, we choice the value of storage memory in GBF (m) is 38B (304bit). According to section 3, when $k=k_0=k_1=2$ the maximum false-positive probability of only 6.3%, in Fig. 4, the value of false-negative probability is clearly observed for each n, which is the number of router in network. Assuming n = 10, means that there are ten routers in attack path, the probability of identifying original router of network attack is almost 20%, and can be reduced with larger of m/n.

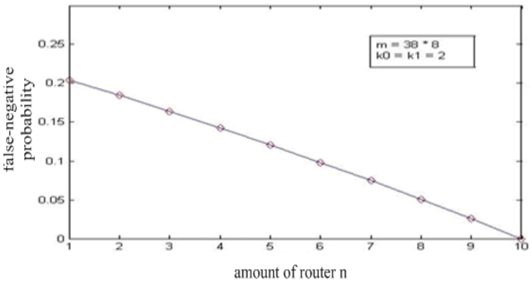


Figure 4. False-negative probability of a GBF for number of routers

In Fig. 5 the data structure of IP packet is designed. We add 40BYTE in the Option after IP Header, the first Byte is flag, the second Byte is the length of Option, so the first WORD is 0x28, and the leaving space of 38 Bytes is for GBF.

<table>
<tr><td colspan="3">Ethernet(14B)</td></tr>
<tr><td colspan="3">IP Header(20B)</td></tr>
<tr><td>flag
(1B)</td><td>size
(1B)</td><td>GBF(38B)</td></tr>
<tr><td colspan="3">Other Data</td></tr>
</table>

Figure 5. The data structure of IP Packet with GBF

B. System Design

Internet is composed of a set of autonomous systems under the control of different administrative entities. In the context of this work, we focus on the implementation of traceback mechanisms between several autonomous systems to identify the source AS of fault packets, and in the AS to determine the origin of attack packet. We define that An Inter-AS traceback is a recursive border tracking on each AS included in an attack path, and an Intra-AS traceback is aimed to seek attack nodes inside an AS. To exactly identify the origin of attack packet, the IP Traceback system can do both of Inter-AS traceback and Intra-AS traceback.

The tracking process has two stages, one is packet marking, the other is tracking. Packet marking is carried out by router, the victim initiate the process of tracking. Fig. 6 depicts the process of IP Traceback based on GBF. process ① is packet GBF marking by router, process ② is initiated tracking by victim, process ③ is that result of traceback sent to victim by the router which is close to attacker.

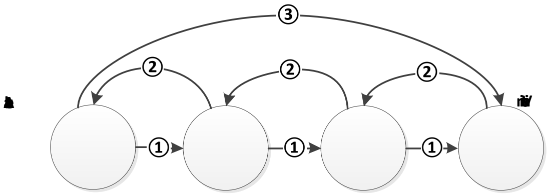


Figure 6. The Process of IP Traceback based on GBF

The IP Traceback System based on GBF (IPTSGBF) is consisting of our modules. The first module is Router Mark (RM), the second is Victim Traceback (VT), the third is Router Traceback (RT) and the last is Manager Traceback (MT). The system infrastructure of IPTSGBF is depicted in Fig 7.

During attacking packet forwarding to victim, it would be marked by router with GBF. If transform n routers, it will be marked with n RM marks. VT can launch two different tracking, each other is independent. RT process is hop-by-hop tracking attack packet in AS, while the MT is responsible for the Inter-AS traceback and Intra-AS traceback. The result of tracking is sent to victim by the RT or MT which is close to attacker.

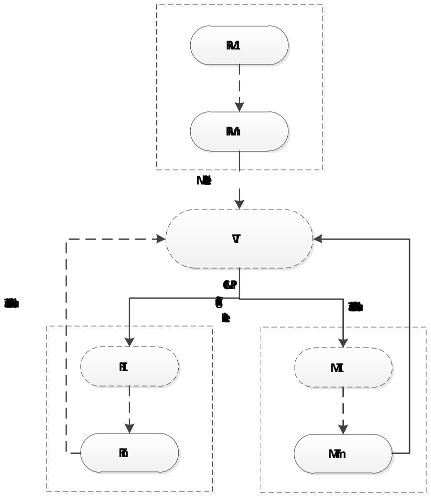


Figure 7. The System Architecture of IPTSGBF

RM is deployed in the router, which can write the router's address into the Option of IP Packet with GBF format. The Option field of IP Packet is not usually used, so this marking should not disturb normal operation in Internet. RM module is implemented in the kernel of router to improve its performance. When marking, RM will detect whether there is GBF storage room in IP Packet. Then if so, RM marks the address of its transmitting export and if not, it will set up GBF memory and mark the address. This operation is depicted in Fig 8.

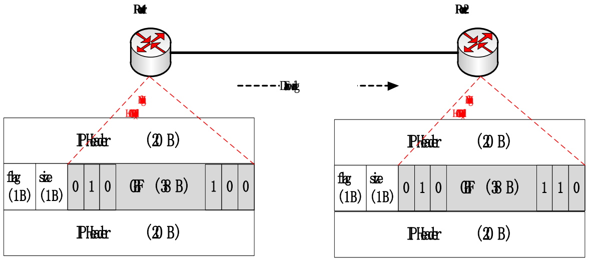
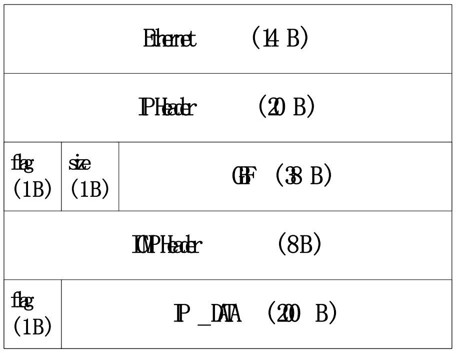


Figure 8. The process of GBF marking

VT module is deployed in the victim, that including PC, Servers etc. VT is the IP Traceback starter and when IDS detects an attack, VT will construct the ICMP traceback request packet to send to RT or MT, which is responsible of tracking. ICMP traceback request packet contains GBF data from IP Packet and space of storing IP address (IP_DATA), depicted in Fig 9. In tracking process every RT or MT will write its address into IP_DATA field, if corresponding router is in the attack path. After finishing tracking, the ICMP traceback request packet will send back from the RT or MT, which is closed to attacker. To parse the content of IP_DATA, VT will determine the attack path and identify the origin of attack.



Ethernet (14 B)		
IP Header (20 B)		
flag (1B)	size (1B)	GBF (38 B)
ICMP Header (8B)		
flag (1B)	IP_DATA (200 B)	

Figure 9. The Structure of ICMP Traceback request packet

RT is also deployed in the router. The principle of RT's work is depicted Fig 10. Usually, malicious IP packet is transformed from R4, R3, R2, and R1 to victim. Victim will launch the request of tracking to R1, router R1 will check the address of router R2 in GBF or not. If it is, R1 will send the ICMP traceback request packet to R2, router R2 receive this packet, and then separately check the address of router R3 and R5 in GBF. Since the address of R5 is not in the GBF, R3 is in, then router R2 send the request packet to router R3. In the whole tracking period, the confirmed router will write its address to the IP_DATA field of ICMP traceback request packet. And so on, the request packet will arrive at router R4, which is only link the router R3, no other neighbor router. The result that original router is R4 will be confirmed, then the router R4 will feedback the result to victim, which the attack path is V-R1-R2-R3-R4.

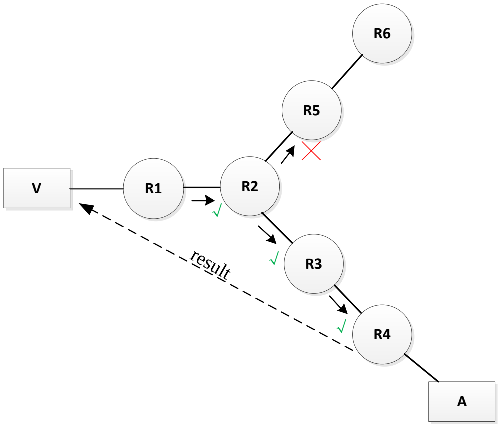


Figure 10. The Process of RT

MT module can be deployed in the manager system of AS, so we suppose that MT module can get the network topology of AS. The process of MT is depicted in Fig.11. Some malicious data, which an attacker A from AS2 sending, is transformed from router R9 to router R1, attack the victim V. V will send request of traceback to MT of AS1 M1, which carries out the process of tracking in AS. Alike the process of RT, M1 can get the result of tracking in AS1, which is R1-R3-R7, and write it to ICMP traceback request packet, then M1 will check its neighbor AS. When it checks the neighbor router from AS2, M1 will send the ICMP traceback request packet to the M2. So M2 will do the same process of MT like M1, has the result of tracking, i.e. R8-R9 to write the IP_DATA of request packet. MT module in AS2 will check its neighbor AS, if it has some neighbor router, it will send request packet to continue. If not, M2 will send the result of tracking to victim V. When V received the result of tracking, it reconstructs the attack path, i.e. V-R1-R3-R7-R8-R9.

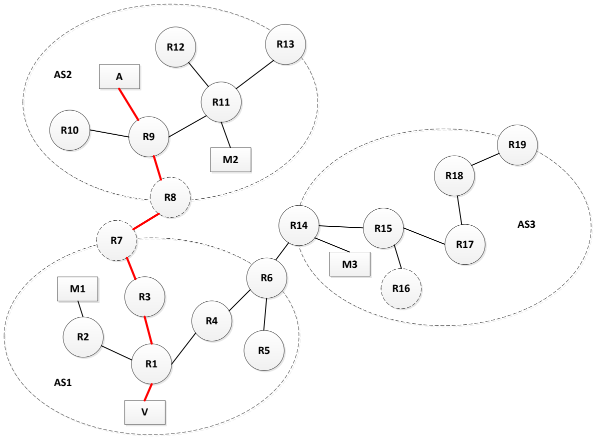


Figure 11. The Process of MT

C. Practical Implementation

For our IPTSGBF prototype, we simulate two autonomous systems, i.e. AS1 and AS2 with virtual machine (VMware). This prototype includes nine local networks, twelve PCs of Linux. Attack machine A , benign machine N1, MT machine M1 and three routers (R1, R2, R3) are in the AS1, while Victim V, benign machine N2, MT machine M2 and three routers (R4, R5, R6) are in the AS2. The detail configuration of prototype is depicted in Fig. 12. "…" in figure 12 means "192.168", for example "…1.2" is "192.168.1.2".

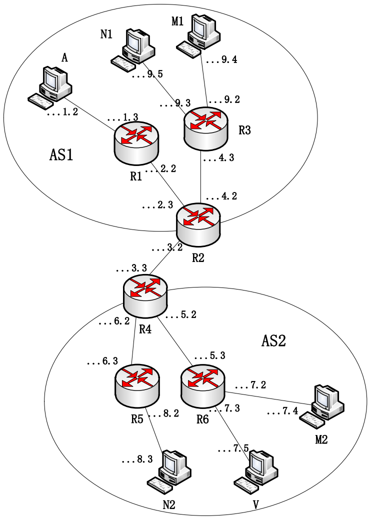
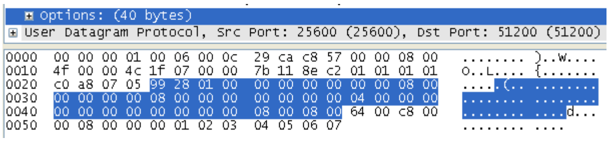


Figure 12. Testbed of IPTSGBF prototype

RM and RT are deployed in the routers (R1,R2,R3,R4,R5 and R6), MT software is worked in the M1 and M2. Attack PC A simulates sending attack packets, e.g. IP spoof. Benign PCs (N1 and N2) simulate normal network operation, e.g. FTP, Web etc. VT is run in the victim machine.

When victim is attacked by A, e.g. SYN-Flood, the IDS of V will alarm and VT receive the IP spoof packets, structure of data is depicted in Fig.13. The fuscous part of figure 13 is 40 Bytes GBF space of IP Option field. 0x99 is sign, 0x28 is its length, and the left is 0x00 which means it is not used for marking.

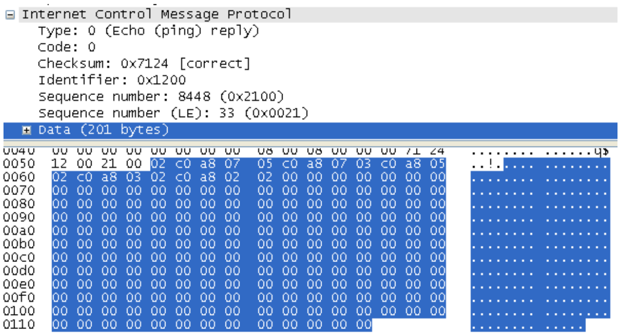


```
⊞ Options: (40 bytes)
⊞ User Datagram Protocol, Src Port: 25600 (25600), Dst Port: 51200 (51200)

0000  00 00 00 01 00 06 00 0c  29 ca c8 57 00 00 08 00   ........ )..W....
0010  4f 00 00 4c 1f 07 00 00  7b 11 8e c2 01 01 01 01   O..L.... {.......
0020  c0 a8 07 05 99 28 01 00  00 00 00 00 00 00 08 00   .....(.. ........
0030  00 00 00 00 08 00 00 00  00 00 00 00 04 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  08 00 08 00 64 00 c8 00   ........ ....d...
0050  00 08 00 00 00 01 02 03  04 05 06 07               ........ ....
```

Figure 13. The Structure of IP Spoof Packet

On finishing the tracking, the VT will receive the ICMP traceback packet, which is the result of traceback, depicted in Figure14. The first byte is 0x02, which means this packet is the result of traceback, the second is 0xc0a80705 which is the IP address of victim, and the following is 0xc0a80703 which is the IP address of router R6. And so on, we can have the attack path that is V-R6-R4-R2-R1.



```
⊟ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x7124 [correct]
    Identifier: 0x1200
    Sequence number: 8448 (0x2100)
    Sequence number (LE): 33 (0x0021)
  ⊞ Data (201 bytes)

0040  00 00 00 00 00 00 00 00  08 00 08 00 00 00 71 24   ........ ......q$
0050  12 00 21 00 02 c0 a8 07  05 c0 a8 07 03 c0 a8 05   ..!..... ........
0060  02 c0 a8 03 02 c0 a8 02  02 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0110  00 00 00 00 00 00 00 00  00 00 00 00 00            ........ .....
```

Figure 14. The Result of IP Traceback base on GBF

V. Discussion and Conclusion

Our IPTSGBF using GBF to store marked data, it has some advantage that its memory space is fixed and can be independent of GBF initialization state. It can track a signal packet and its complexity of computing is low. It also can track after attacking and make little overhead of network. However, the false-positive of GBF is lower with the number of hash function increasing, so how to solve this problem will be the focus in the future. We should select ideal tradeoff between false positives and false negatives in the real network. Even though there are some false positives or false negatives, we can still reconstruct the whole path of attack in the autonomous systems.

References

[1] S. Savage, et al., "Network Support for IP Traceback ," vol.9, JUNE 2001

[2] W. Timothy Strayer, Christine E. Jones, Fabrice Tchakountio, Alex C. Snoeren,Beverly Schwartz, Robert C. Clements, Matthew Condell, and Craig Partridge, Traceback of Single IP Packets Using SPIE, Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX’03),2003

[3] S. M. Bellovin, “ICMP Traceback Messages”. InNetwork Working Group Internet Draft, March 2000.

[4] T. S. D. o. C. S. A.John, Ramanujam School of Mathematics and Computer Science,"DDoS: Survey of Traceback Methods," vol. 1, May 2009.

[5] H. Aljifri, "IP Traceback: A New Denial-of-Service Deterrent?," 2003.

[6] H. Burch and B. Cheswick, "Tracing anonymous packets to their approximate source," Dec 2000.

[7] SAVAGE S, WETHERALL D, KARLIN Anna. Practical Network Support for IP Traceback[D]. Department of Computer Science and Engineering University of Washington Seattle, WA, USA,2000.

[8] BELENKY A., ANSARI N, IP traceback with deterministic packet marking[J]. IEEE Communications Letters, 7(4) (April 2003), pp. 162–164.

[9] 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.

[10] RAFAEL P L, PEDRO B, VELLOSO, Generalized Bloom Filters[OL], COPPE/UFRJ, Tech. Rep. GTA-05-43, 2005 - gta.ufrj.br

[11] Rafael P. Laufer , Pedro B. Velloso, Daniel de O. Cunha, Igor M. Moraes, Marco D. D. Bicudo, Marcelo D. D. Moreira, and Otto Carlos M. B. Duarte, "Towards Stateless Single-Packet IP Traceback," 32nd IEEE Conference on Local Computer Networks - LCN'2007, Dublin, Ireland, October 2007.