A Novel Hybrid Technique for Internet Protocol Traceback

G.VeeraSwamy[1] & A.Arun[2]

1. M-Tech CSE ,Vignana Bharathi Institute of Technology, Ghatkesar,Hyderabad . Telangana.
2. Assistant Professor, Vignana Bharathi Institute of Technology, Ghatkesar , Hyderabad. Telangana.

[1]veera.5c6@gmail.com

Abstract—

Now a days the Internet has been widely applied in a variety of fields, more and more network security issues come into view and catch people's attention. However, adversaries frequently hide themselves by spoofing their own Internet Protocol addresses and then launch attacks. For this reason, researchers have projected a lot of tracebacking schemes to mark out the source of these attacks. Some peoples use only one packet in their packet logging schemes to accomplish IP tracking. Other peoples combine packet marking with packet classification and there after create a hybrid IP traceback methods demanding less storage space but require a longer search.In this paper, we suggest a new hybrid IP traceback scheme with more effective packet logging aiming to have a fixed storage space requirement for each router in the network in packet logging without the required to restore the logged tracking data and to achieving zero false positive and false negative rated values in attack path reconstruction .In addition, to we use a packet's reference field to sensor attack traffic on its upstream routers. Lastly, we reproduce and analyze our proposal, in comparison with some other related investigation,in the aspects of: storage requirement, computation, and accuracy.

KEYWORDS—

Trace back IP; CAIDA ; IP Spoofing; Path reconstruction; Dos Attacks; Router Interface

I.INTRODUCTION

With the fast growth of the Internet, different internet applications are developed for different kinds of users. Due to the reduce cost of Internet access and its increasing availability from a plethora of devices and applications, the impact of assault becomes more significant. To disrupt the service of the server, and the sophisticated attackers may launch a distributed. denial of service (DoS) attack . Based on the number of packets to deny the service of a server, we can categorize D DoS attacks into coding-based attacks and software exploit attacks . The major signature of coding-based attacks is a huge amount of forged source packets to exhaust a victim's partial resources. Another type of DoS attack, software exploit attacks, attacks a host by using the host's vulnerabilities with few packets. Since most edge routers do not check the packet's origin's address of a packet, core routers have difficulties in recognizing the source of packets. These source IP address in a packet can be spoofed when an attacker wants to hide himself from tracing. Therefore, IP address spoofing makes hosts hard to defend against a D DoS attack. For these reasons, developing a mechanism to locate the real source of the impersonation attacks has become an important issue now a day.

For tracing the real source of coding-based attack packets, Burch and Cheswick propose a link test scheme using the UDP charges service to generate an extra load to upstream links. The extra load may compete against the attack packets and perturb the

attack traffic, so that we can find the upstream router through which the attack traffics passes. propose an I Track scheme, which generates an ICMP packet with forward and backward links of the router to leverage the triggering packet. The victim host collects all the ICMP messages to reconstruct the attack path. Because previous schemes need extra packets to trace the origin of attack packets, packet marking approaches are introduced to mark the router or path information on the triggering packets.

Packet marking can be put into two categories, deterministic packet marking (DPM) and probabilistic packet marking (PPM).propose DPM trace back schemes to mark a border routers IP address on the passing packets. However, IP packet's header's identification field is not enough to store the full IP address. For this reason, the border router can divides its IP into several segments and computes the digest of its IP. Then it randomly decide a segment and the digest to mark on its passing packets. When the purpose host receives enough packets, it can use the digest to assemble the dissimilar many segments. On the other hand, Savage et al. propose a PPM scheme with edging sampling which is called FMS. Song and Perrig propose the AMS scheme. Yaar *et al.* propose the FIT scheme. Al-Duwari sarrc and Govindarasu propose the probabilistic pipelined packet marking (PPPM) scheme. These probability-based schemes require routers to mark partial path information on the packets which pass through them with a probability. That has to say, if a victim collects enough marked packets, it can restructure the full attack path.

Since coding-based trace back schemes need to collect a huge amount of attack packets to find the origin of attacks, these schemes are not suitable for tracing the origins of software exploit attacks.

Most current tracing schemes that are designed for software exploits can be classify into three groups: single packet, packet logging, and hybrid IP trace back. The basic idea of packet logging is to log a packet's information on routers. Huffman codes , Modulo/ Reverse modulo Technique (MRT) and Modulo/Reverse modulo (MORE) use interface numbers of routers, instead of partial IP address or link information, to mark a packet's routing information. Each of these scheme marks routers' interface numbers on a packet's IP header along a route. However, a packet's IP header has rather partial space for marking and therefore cannot always afford to record the full route information. So, they put together packet logging into their marking schemes by al-lowing a packet's marking field for the short term logged on routers.

We find these tracing methods still require high storage on logged routers. And also, their schemes cannot avoid the false positive problem because their packet digests in each log table may have conflict, and their schemes even have false negative problem when routers refresh logged data packet A part from these ,we find their comprehensive searching quite inefficient in path reconstruction.

For these many reasons, we propose a traceback scheme that marks routers' interface numbers and integrates packet logging with a hash table (RIHT) to deal with these logging and marking concern in IP trace back. RIHT is a hybrid IP trace back scheme designed to achieve the following properties: 1) Our storage requirement for an arbitrary router is bounded above by the number of paths to the router, and thus every router does not need to refresh logged tracking information. 2) Our scheme achieves positive and

negative rates in attack path reconstruction

In the marking process, each router puts into the data marking field. Possibly the simplest way to encode data is by fixed-length coding. However, such an approach does not use a packet's marking field efficiently if it is not a power of two. Choi and Dai propose a marking scheme using Huffman coding to reduce the bits required for marking on a packet.

It encodes by Huffman coding according to the traffics of each interface. Their analysis shows their proposal has better performance when the traffics allocation for each interface is unequal. Tamilarasi and Malliga propose two traceback schemes, namely MRT and MORE.

MRT uses a 32-bit marking field while MORE uses a 16-bit marking field and it will separates a log table into parts. They use mathematical methods to mark the marking fields . In these marking schemes, the new marking field is computed by the routers to which each a packet is forwarded. In their path reconstruction, the old marking data field marking field is computed by the routers to which a packet is traced back. The upstream interface number marking field is also computed where percentage is the modulo operation, and the packet is sent back to the upstream router along the obtained upstream interface. According to test results in MRT and MORE, the average bits used for marking are a decrease amount of than those in Huffman coding.

II. RELATED WORK

In most of the current single packet traceback schemes be likely to log packets' in sequence on routers. For instance, Snoeren *al.* propose a system SPIE to digest the unaffected parts of the packets and used a bloom filter to filter the digest. Yet, this method requires large storage space and has a false positive problem in a bloom filter. By this reason, Zhang and Guan propose a TOPO to improve the efficiency, precision of SPIE, but TOPO still wants a large storage capacity and inevitably has a false positive problem because of bloom filter. And the hybrid IP tracebacking schemes are introduced to moderate the storage problem of logging based tracebacking methods Gong and Sarac proposed a hybrid IP tracebacking scheme called as Hybrid IP Traceback (HIT) for combining packets marking and packets logging. HIT uses packets marking to reduce the more number of routers required for logging. Other researchers have proposed new models to further reduce the storage requirement for each router logging and to decrease the various number of routers required for logging.

Since these methods are using interface numbers of routers for marking, they are assume a router set R={R1,R2,R3…...Rl} comprises a routers in the networks and require all the routers to support the particular traceback schemes. And also, they use the degree of a router as a parameter in the marking schemes where the degree is the number of interfaces of the router, not together with ports connected to the local networks. Here we are using D{Ri} degree of a router Ri Besides with, these schemes need to maintain an interface table on each and every routers in advance. This table will maps a unique number to each interface of a router along which the router is connected to another router. The interface numbers of a router Ri are between 0 and the D{Ri}-1 For the discussion, we denote by UTi the upstream interface number of a router Ri in a router. In what follows, we use routers and paths interchanged.

In these marking process, each router puts U(Ri} into the marking fields. Perhaps this is the simplest way to encode UTi is by fixed length coding . However,these such an come close to does not use a packets marking field effectively if is not a power of two. Choi and Dai suggest a marking scheme using Huffman coding method to reduce the bits required for marking on a packets. It encodes by Huffman coding according to the traffic of each interface. Their analysis represents their schemes has good performance when the traffic sharing for each interface is unequal. Malliga and Tamilarasi propose two tracebacking schemes,namely MRT and MORE.

Even though the all marking fields of a packet in Huffman codes, MRT, and MORE each can store the path of longer length than in the fixed length coding, these marking fields may be full before the packet reaches its destination. In such a circumstances, they need to log the packet's information on the routers that fail to mark on the marking field. These routers may pair the packet digest with the marking field, and then they can log the pair into a log table. After logging, these routers clear the marking field and to repeat the marking process.It could recover the marking fields by the above steps.

But there are two problems in the following Huffman codes, MRT and MORE's schemes. The First is , after logging, if the marking field of the packet is still 0 on the neighboring downstream router, it will be recognized as a logged router for the packet while tracingbacking. Then it will fail to find the origin. And the Second , since the digests in a log table might have a collision with other, it causes the false positive problem during the path reconstruction of a router.

Due to these problems in the Huffman codes, MRT and MORE models, we propose a tracebacking scheme that marks routers interface numbers and integrate packets logging with a hash table. RIHT has a less storage requirement and better exactness and efficiency than Huffman codes and MRT.

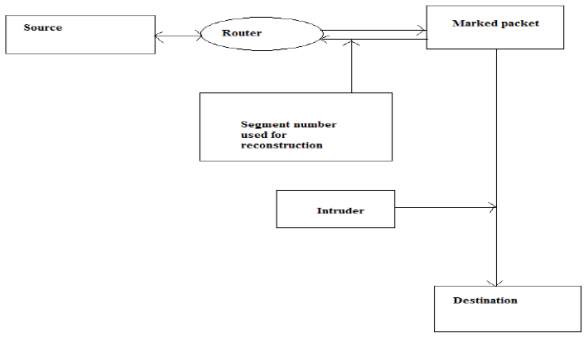

Fig 1.Design of RIHT

III. FRAMEWORK OF RIHT ALGORITHM

3.1. Packet marking

This component is designed in such a way that when an IP packet enters into the protected network, it is noticeable by the interface close to the source of the packets on each router. These source IP addresses is stored in the marking field. These mark will not be overwritten by intermediate routers when the packet traverses the network**.**

3.2.IP Trace back

After all of these segments corresponding to the same router address have been arrived at the reconstruction point of the path, and the source IP address of the each packets can be reconstructed. In order to keep the track of the sets of IP packets that are used for reconstruction, the identities to show the packets coming from the same source must be included.The reconstructed packet will be forwarded by the router to the server by the legitimate client's IP address.

3.3. Reconstruction

Reconstruction is the process of receiving back the packet and sending them one by one

denial of service. This will helps in construction of inappropriate packets and also helps to avoid further the loss of packets. These FDPM involves in the number of the packets count of the reconstructed packets.

IV. RIHT

RIHT marks interface numbers of routers on packets so as to trace the path of packets. Since the marking field of each packet will be limited, our packet marking method may need to log the marking field into a hash table and store the table index on the packet. We repeat this marking (or) logging process until the packet reaches its destination. After that, we can reverse such process to trace back to the origin of attack packets.

4.1. Network Topology and Preliminaries

A router can be connected to a local network or other data routers; or even both .A border router receives packets from its local network packets. A core router receives packets from other routers .For example, serves as a periphery router when it receives packets from host. However, it becomes a core router when a router receiving packets from .

The following are assumptions of our scheme.

1) A router creates an interface table and numbers the up-stream interfaces from 0 to 1 in advance.
2) The router should knows whether a packet comes from a router or a local network.
3) Such a trace back scheme is possible on every router.
4) The traffics route and network topology may be changed, but not often.

John et al..Also point out that over 60% of fragmented packets are attacking packets. Therefore, if attackers try to use Encapsulating Security Payload (ESP) packets to evade IDS, their at random generated ESP packets can never be decrypted at a victim's site because of the lack of proper mutual keys. In such a case, the adversaries can only generate a large volume of forged ESP packets to attack a host, consuming the victim's bandwidth and computation resources. If we mark the ESP packets with a low probability value, the marked packets are enough for us to trace the attackers' source, and the unmarked segmented ESP packets are still able to assemble at the destination host. In some cases, adversaries may compromise a node in the target network. Then they can use ESP packets in their software exploit. And Teardrop attack and LAND attack, which tend to consume destination hosts' buffer and computation resources. If we overwrite the segment field, the attackers are not able to launch Teardrop attacks to deny the service at a victim's site.

As mentioned above, the use of the fragment and the identification fields will not affect most legitimate packets. Besides, fragmentation is commonly used for IDS evasion. Thus, when we overwrite these two fields in our trace back scheme, we avoid attackers using fragmented packets to evade IDS. For this reason, we use an IP header's identification field, flag field, and fragment offset field as a 32-bit marking field.

4.2 Marking and Logging Scheme

When a border router receives a packet from its local net-work, it sets the packet's marking field as zero and forwards the packet to the next core router.

4.3 .Path Reconstruction

When a victim is under attack, it sends to the upstream router a reconstruction request data, which includes the attack

packet's marking field, termed here. When a router receives a reconstruction request, it tries to find the attack packet's upstream router.

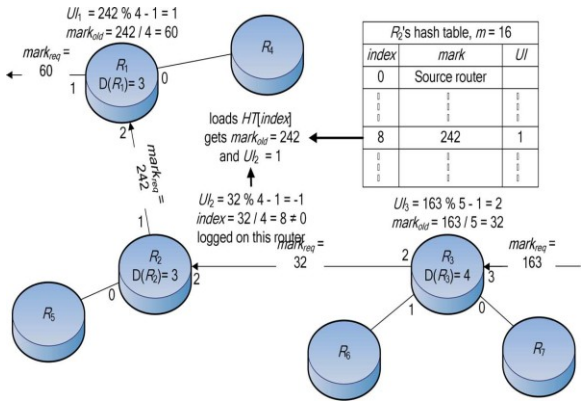

Fig 2. Example for Reconstruction of path

4.4 RIHT Extension

As for the partial deployment issues in our traceback scheme, each router only needs to know its upstream router which com-plies with our scheme. Then, the two routers can use a tunnel for direct communication between them. It means if the adjacent router does not support our traceback, we will not receive any positive feedback and will have to query the next one hop.

On the other hand, if an attack packet reaches a NAT server before any routers that support our traceback scheme, we can only trace its source to the NAT server. That is to say, we can only find the attack's local area network, which, however, is sufficient to locate the origin of an attack.

Also, the modification of a router's port numbers may lower the precision of our scheme. In this case, we can extend our path reconstruction scheme into a two-layer approach to get around this problem. First, each ISP needs to run our trace back scheme separately. Since every ISP is well aware of the port-number modification, they can exactly identify an AS's incoming and outgoing border routers which a packet goes through. Second, the victim site needs to run our scheme to query a traceback server in an AS in order to reconstruct an attack path. With this extension of our scheme, we can guarantee the high accuracy of this approach.

V. COMPUTATION ANALYSIS

In the following, we compare the computing time of logging and path reconstruction in RIHT with that in MRT and MORE. Since RIHT uses a hash table to log, we inevitably have to face a hash table's collision problem. In RIHT, the open addressing method is used to solve this problem. In the opening the addressing method, when a new entry has to be inserted, these slots are examined, starting with the hashed to slot and proceeding in some probe sequence, until an unoccupied slots are found. When searching for an entry, the slots are scanned in the same series, until either the objective record is found or an unused slot is found. Furthermore, to minimize the impact of the collision problem on our system, we adopt the quadratic probing as the probe sequence because it requires only light computation and is proved effective when we try to avoid clustering problem .When we deal with the collision problem, we have to take into consideration a hash table's load factor, which directly affects the number of collisions.

However, the calculation results of collision times may vary because we have two situations, successful search and unsuccessful search, when logging. We explain the two condition and their relations with collision times as follows. Unsuccessful search means that an entry has not been logged in a hash table and therefore is to be

inserted into an empty slot. These probe is performed each time collision occurs. The predictable number of probes in unsuccessful search using open addressing is at most, presumptuous uniform hashing. Successful search means an entry has been logged in a hash table. The likely number of probes in a successful search using open the addressing is at most by uniform hashing.

5.1 Storage Requirement

Our scheme maintains a hash table and an interface table on a router, while MRT and MORE maintain log tables and an interface table on a router. Since the storage requirement of an interface table is insignificant, we leave it out of our storage requirement analysis. In RIHT, the size of a hash table decides how several paths can be logged on a router. For two arbitrary packets in RIHT, they take the same path to a router if and only if they have the similar marking field on the router. Thus, our scheme regards the marking field of a packet as one path to a router. For discussion, we say that a path to router requirements to be logged on if the marking field of every packet taking this path requirement to be logged on. A hash table's load factor, where is the number of logged paths in a hash table. As the analysis in Section IV-A .Therefore, if the number of paths which need to be logged on a each router is, and then the size of the hash table on the router should be set.

R_6

5.2 False Positive and False Negative Rates

R_2 R_4

When a router is mistaken for an attack router, we call it "false positive". When we fail to trace back to an attacker, we call it false negative in MORE and MRT, the size of a log table increases with the number of logged packets, but a router's memory is incomplete. Thus, when those schemes are out of the memory, they have to reload their log tables. The false positive or false negative problem happens when the logged data is refreshed. Unlike MRT and MORE, RIHT's hash table size depends on the number of logged paths, and the table does not have to refresh. Therefore,RIHT has no false positive and false negative problem in this respect In MRT, a router logs the marking fields of packets, which are indexed by the digests of the packets.

In MORE, a router uses different log tables, which are associated with marking fields of packets indexed by the digests of the packets. Therefore, the false positive rates of MRT and MORE are greater than 0 even without refreshing if a collision of assimilate happens in that log table. On the other hand in RIHT, since we mark index on each logged packet's these marking fields, under the guidance of each index, we can just obtain the logged data from the hash table and circumvent the collision problems. Therefore without any chance of collisions in our scheme, our false positive rate is 0, hence higher precision.

5.3. Packet Identity

In the same route, every packet's marking field is the same on an arbitrary router .Hence a packet's marking fields is often seen as a packet recognize and used to help us identify an attack packet's source and then filter malicious packets. But in the MRT and MORE schemes, we are unable to identify a packet's source from its marking field in the following situation. for example, illustrates those packets, which are logged on the same router, say in this case, come from different sources, and turn out to carry the same values in their marking fields when they are heading to the victim. For this reason, the victim as well as and are confused and not capable to identify the packets' source simply from the

contrast of marking fields.

V. CONCLUSION AND FUTURE SCOPE

In this paper, we propose a new hybrid IP tracebacking scheme for efficient packet logging aiming to have a fixed storage requirement in packet logging without the need to refresh the logged tracking information.And also these proposed scheme has zero false positive and false negative rates in an attack-path reconstruction. As a part of these properties, our scheme can also deploy a marking field as a packet identity to filter malicious traffics and secure against DoS/DoS attacks. As a result, with more accuracy and low storage requirement, and fast computation, RIHT will serve as an efficient and protected scheme for the packets As for our future work, we would like to come up with another version of RIHT which uses a 16-bit marking field to avoid the problem caused by packet fragmentation.

REFERENCES

[1] CISCO, "Cisco Visual Networking Index :Global Mobile Data Traffic Forecast Update, 2011," Tech. Rep, 2012.

[2] Y. Li, Y. Zhang, and R. Yuan, "Measurement and Analysis of Large Scale Commercial Mobile Internet TV System," in *ACM IMC*.

[3] T.Taleb and K. Hashimoto, "MS2: A Novel Multi-Source Mobile-Streaming Architecture," in *IEEE Transaction on Broadcasting*,

[4] X. Wang,S.Kim, T.Kwon, H.Kim, Y.Choi, "Unveiling the BitTorrent Performance in Mobile WiMAX Networks," in Passive and Active Measurement Conference, 2011.

[5] A. Nafaa,T. Taleb, and L. Murphy, "Forward Error Correction Adaptation Strategies for Media Streaming over Wireless Networks," 2008.

[6] J. Fernandez, T.Taleb, M. Guizani, , "Bandwidth Aggregation-aware Dynamic QoS Negotiation for Real-Time Video Applications in Next-Generation Wireless Networks,", 2009.

[7] T. Taleb, K. Kashibuchi, A.Leonardi, S. Palazzo, K. Hashimoto, N. Kato, and Y. Nemoto, "A Cross-layer Approach for An Efficient Delivery of TCP/RTP-based Multimedia Applications in Heterogeneous Wireless Networks,"in *IEEE Transaction on Vehicular Technology*, vol. 57, no. 6, pp. 3801–3814, 2008.

[8] K. Zhang, J.Kong, G.L Song, "Multimedia Layout Adaptation Through Grammatical Specifications," in *ACM/*Springe*r* Multimedia Systems, vol.10, 2005.

[9] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-Time System for Adaptive Video Streaming Based on SVC,"

[10] H. Schwarz, D. Marpe, and T.Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," in *IEEE* Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, Sep. 2007.

[11] P. McDonagh, C.Vallati, A. Pande, and P. Mohapatra, "Quality-Oriented Scalable Video Delivery Using H. 264 SVC on An LTE Network," in *WPMC*, 2011.

[12] Q.Zhang,and R.Boutaba, "Cloud Computing:State-of-the-art and Research Challenges," in *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, Apr. 2010.

[13]D. Niu, H. Xu, B. Li, and S.Zhao, "Quality-Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications," in *IEEE INFOCOM*, 2012.

[14]Y.G. Wen, W.WZhang, K. Guan, D. Kilper, and H. Y. Luo, "Energy-Optimal Execution Policy for A Cloud-Assisted Mobile Application Platform," Tech. Rep., September 2011

[15]W.W Zhang, Y.G. and D.P.Wu, "Energy-Efficient Scheduling Policy for Collaborative Execution in Mobile Cloud Computing," in

[16]W.W.Zhang, Y.G. Wen, Z.Z. Chen and A.Khisti, "QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming over Wireless Networks," in *IEEE Transactions on Multimedia*, November 2012.

[17]Z. Huang, C.Mei, L. E.Li, and T. Woo, "CloudStream : Delivering High-Quality Streaming Videos through A Cloud-based SVC Proxy," in *IEEE INFOCOM*, 2011.

[18]N.Davies,"The Case forVM-Based Cloudlets in Mobile Computing," in *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.