IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-12, Dec 2018 E-ISSN: 2347-2693

Spitri: Single Packet ICMP Traceback Using Router Interface

S. Suganya[1], P. Subramaniam[2]

[1,2]Department Of Computer Science Muthayammal College of arts and science Namakkal-637408, Tamilnadu, India

Available online at: www.ijcseonline.org

Accepted: 06/Dec/2018, Published: 31/Dec/2018

***Abstract*:-**Securing the Internet and its services is recognized as one of the most challenging research problems. Amongst the threats imposed on the Internet, Distributed Denial of Service (DDoS) attack has occurred recurrently with a severe impact on the economy of the organization. Regardless of the fact that security experts propose plentiful stupendous solutions to mitigate DDoS attack, it has continued to prevail over a decade. This convolutes the forensic inspection and countermeasures against DDoS offensive. Identifying the origin of the attack is an important and essential step towards deterrence and countermeasures against these attacks. However, they either require huge storage at the routers or require numerous packets to traceback the attack path. Further, most of the marking based traceback schemes are not backward compatible. This proposed system focuses on scrutinize these issues and proposes a feasible solution to identify the origin of Direct Distributed DDoS attack. Backward compatible Single Packet ICMP Traceback scheme using Router Interface (SPITRI) is proposed. It also uses an out-of-band ICMP message to track the attack path. It identifies the origin of an attack packet with a single ICMP message whereas the existing ICMP based traceback scheme requires more number of ICMP packets. Subsequently, SPITRI has undoubtedly reduced the bandwidth overhead provoke by the existing ICMP based traceback scheme. It traces back the attacker with minimal computation overhead and negligible storage at the routers. According to CAIDA dataset, SPITRI tracebacks 13000 attackers with an accuracy of 95.98%.

Keywords; **-** Spoofing; Trace back; Client-Server Authentication; IP forging, Distributed Denial of Service, Single Packet ICMP Traceback scheme using Router Interface.

I. INTRODUCTION

The contribution of Internet has directly benefited communication, education, business, health sector, farm sector and many more. But, the adversary is always on the prowl biding his time to create chaos in the Internet space. Consequently, securing the Internet and its services is a burgeoning issue. Amongst the various attacks hampering the security of Internet, considered as the most pernicious weapon. For fast few years to now,DDoS attack remain as a major threat to the availability of Internet services and it is still evolving. Before ten years, the most powerful DDoS attack was at the rate of 8 Gbps. Today, the record breaking DDoS attack has peaked at 500 Gbps in Hongkong (FORBES 2014) [1].

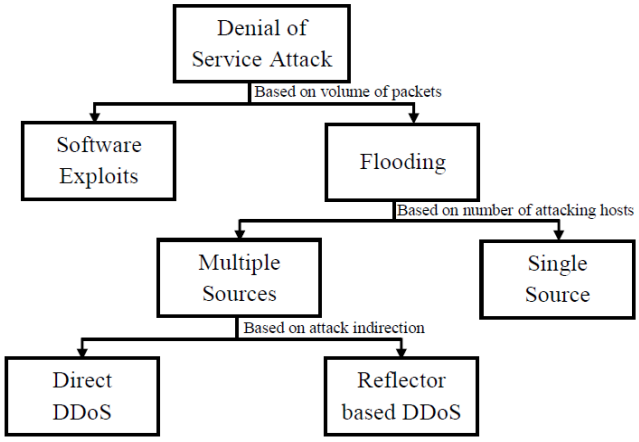

Figure 1 Classification of denial of service attacks

These resources can be network bandwidth, calculate power, memory or OS data structures. It also exploits the weaknesses in software design or implementation. In Distributed Denial of Service (DDoS) attack, the offensive

also known as master establishes its own network of compromised hosts termed as zombies. Consequently, to launch an attack, the master triggers the zombies which generally are huge in number and are distributed across the network to explode the attack traffic towards the target. The target can be a mail server, web server, a DNS server, an Internet gateway, a highly sophisticated power grid or any network resource. DDoS attacks can be broadly classified into flooding attack and software exploits (Hussain et al 2003) [2] based on the volume of packets involved in the attack. Figure 1 represents the various classifications. Flooding attack can further be classified into one source and multiple source attacks based on the number of attacking hosts. Numerous source (DDoS) attacks are additional classified into straightforward DDoS and Reflector based DDoS attack based on the flow indirection of DDoS traffic. The concept of an IP traceback implement to Internet service provider (ISP) defined following real world scenario, see Figure2 A DDoS attack and malicious traffic are detected at a terminal system by intrusion detection systems (IDS).

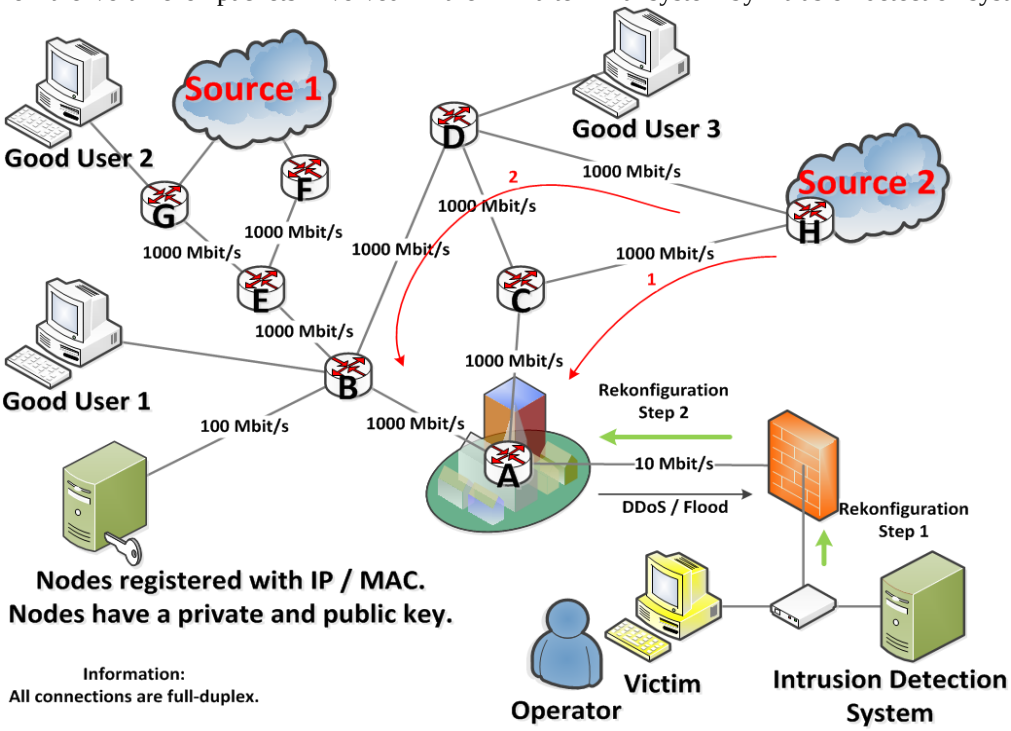

Figure 2: Traceback of sophisticated attackers.

The rest of the proposed paper is referred as follows: - A review of existing works is done in Section II. Section III includes the network model and proposed energy harvesting technique. Simulation results and performance analysis is done in Section IV finally conclusion on section V.

II.LITERATURE REVIEW

Over the past decades, various defense and traceback strategies have been developed. Vincent et al proposed methods of DRS are limited to traceback trace only for 9 hops where as PRS algorithm needs more packets to redefine the path. Belenky et al defined Packet Marking is separated in Edge-Sampling and Node-Sampling. They are not efficient with respect to the capacity. The marking detail instruction in the payload can cause faults. stonr et al proposed, Link Testing by Input Debugging and bruch proposed methods of Link Testing by Controlled Flooding require a continuous attack for complete attack path identification and limit immediate live defense actions. Snoeren et al approached of desertification at a router generate a huge amount of data, needs large resources and limit live feedback actions. The backward ascertained way of the ICM Protocol Izaddoost et al traceback approach does not important match the real way of a packet, because of load accommodate and other authority. The ISP Traceback khan et al and stelte et al methods are allocation to identification of the initial ISP, but not track the direct path. Various conclusion and hybrid results are trace packets as in gong et al modal of traceback.

Table 1: Qualitative comparison of IP traceback categories

Metrics Link Testing	Metrics Link Testing	Packet Marking		Packet Logging	ICMP Traceback	Hybrid Scheme
		PPM	DPM			
Ease of Deployment	Fair	Fair	Fair	Poor	Good	Fair

Scalability	Poor	Poor	Fair	Fair	Poor	Good
Memory at Router	Nil	Nil	Nil	Very High	Nil	Medium
Memory at Victim	Nil	Very High	Medium	Nil	Medium	Nil
Router Processing Overhead	High	Medium	Medium	High	Low	Medium
Flexibility to Partial Deployment	No	Yes o	Yes	No	Yes	No
Prior Knowledge of Topology	Needed	Not needed Faster trace back if known	Not needed Faster trace back if known	Not needed	Not Needed	Not Needed
Post Attack Analysis	Not Possible	Possible	Possible	Possible (short duration)	Possible	Possible
Attackers challenge vs. Scheme survival	poor	Poor	Poor	Poor	High	Poor
Router Involvement during traceback	Very High	Nil	Nil	Very High	Nil	High
Ability to Handle Fragmented Packets	Yes	Yes	Yes	Yes	Yes	No
Handle Packet transformations	Yes	Yes	Yes	Yes	Yes	No
Compatibility existing Protocol implementation	Yes	No	No	Yes	Yes	No
Bandwidth Overhead	Very High	Nil	Nil	Nil	High	Nil

The entire traceback scenario does not accomplish the identified necessity to follow the way of network packets. Table 1 Belenky et al presents many researchers and experts have come out with sound solutions to identify the attackers. It is important to mention here that most of the schemes presented here remain theoretical and have not been implemented in the industry for number of reasons. Consequently, this proposed methods aims solution to trace Direct DDoS attack without altering the fields of an IP header, which has minimal processing and storage accuracy.

III. PROPOSED SYSTEM

As in ITrace, the SPITRI also uses ICMP packet to carry the identify information of the packets. Hence, the existing infrastructure need not be modified and the normal operation of IP packets will not be disturbed. Unlike ITrace, instead of appending the upstream and downstream IP address of the routers, the proposed SPITRI manipulates the trace information with the router interface ID like other router interface based traceback model.

3.1. Basic Assumptions

The proposed SPITRI is motivated by some assumptions made by the existing router interface based traceback approaches. The assumptions are as follows:

1. Every router builds a router interface table and assigns the interface ID from 0 to b-1, where „b" is the total number of links. Even ITrace uses interface identifier in Forward/Backward Link element using character string format.
2. The traceback message generation and construction algorithm is implemented at all the routers.
3. Routers are not generally compromised.
4. Routers can determine the packet from a local network or any router.

3.2. Working of SPITRI

Every traceback message has its own key to unveil the path it traversed. Assume that an Intrusion Detection System (IDS) is running on the victim machine. When it detects an attack, path reconstruction process is initiated. The trace back packet is identified by comparing with the value of Traced Packet Content element received from the SPITRI message.

The Path Information element of that SPITRI message contains the final value. The path to reach the attacker border router and attacker LAN is retrieved from the Path Information value of the SPITRI message itself. So, even if the source IP address of the attack packet is spoofed, the Path Information value of the SPITRI message would be adequate to derive the path to reach the attacker border router and attacker LAN. The attack packet is associated to the SPITRI message by storing first few bytes of the attack packet in the Traced Packet Content element of the SPITRI message. Hence, by that way, logging at the intermediate routers is avoided. By matching the Traced Packet Content element of the SPITRI message with the attack packet, the corresponding SPITRI message could be identified and thereafter, with the Path Information element, attacker LAN could be traced out by employing the process depicted in Figure 4, the traceback algorithm is shown in Figure 3.

Input: SPITRI message with mark value PI.value
Output: Set of upstream interface ID UIId[ ] to reach the attacker LAN from victim
1. **Begin**
2. Let UIId [NR] be Upstream Interface Id Array through which the packet came crossing NR routers, PI value be the current value found in the Path Information of the packet, PI value previous be the Path Information value computed by the router at the previous hop.
3. Identify the ICMP message
4. From the TTL of ICMP message compute the hopcount
5. **While** (hopcount! = 1)
    1. UIId[i] = floor (PI value) – 1
    2. PI value $_{previous}$=(1/(PI value –floor(PI value))-2
    3. PI value = PI value $_{previous}$
    4. i++
    5. hopcount --

End while
6. UIId [i] = PI value
7. Print the Upstream Interface Id array UIId[ ]
8. Generate a Filter request packet travelling through UIId[ ]
9. **End**

Figure 3: Traceback message construction algorithm

From the TimeToLive (TTL) field, the number of hops traversed by the traceback message is computed. The process of identifying the upstream interface ID and the previous value of the Path Information element is repeated till hop count value becomes 1. The upstream interface is identified by Equation (1)

$$\text{UIId} = \text{floor}(\text{PI.value}) - 1 \text{----------1}$$

To discover further upstream routers, the next upstream interface ID from that router has to be identified. To identify this, previous Path Information value of the traceback message is found by Equation (2).

$$PI.value\ previous = (1/(PI.value\ floor(PI.value)) - 2 \quad \text{-----2}$$

where, PI.value $_{previous}$ denotes the previous Path Information value during the construction phase, and *PI.value* denotes the current Path Information value. This process of identifying UIId is continued number of hop times. The output of this traceback process will be a sequence of upstream interface ID that has to be visited to reach the attacker from the victim.

The links from each router is provided with a locally unique interface identifiers starting from 0 to b-1, where „b" is the total number of links available at the router. As mentioned earlier border router is the router which is directly connected to a local network, and it will be the ingress router for the packets going out of that network. Core routers receive packets from other routers. It will be an intermediate router in a network path. A border router can also be a core router. For example, R1 is the border router for the packets from the attacker. If packets from LAN1 choose to travel through R11-R1-R2-R3, then R1 becomes a core router.

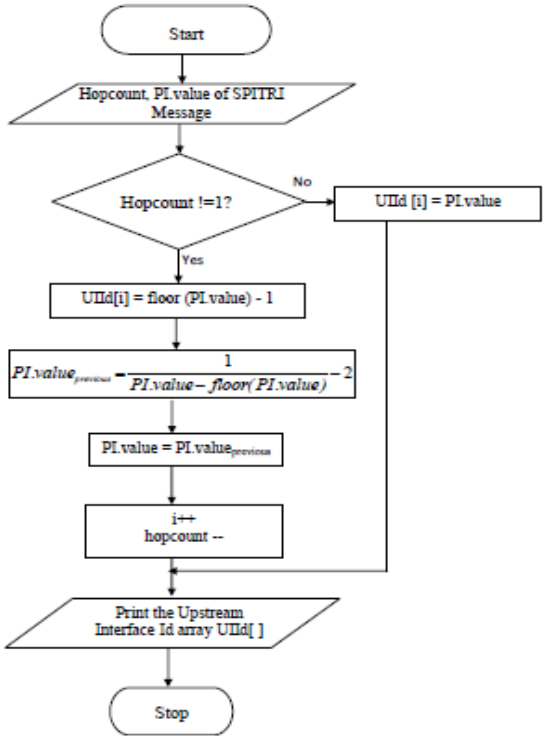

Figure 4: Flowchart for traceback algorithm

SPITRI does not require numerous packets to traceback a large scale DDoS attacks like ITrace. One SPITRI message is adequate to traceback one attacker. Router interface based traceback methods like MRT, MORE and RIHT can also traceback with a single packet, but they demand additional storage at the routers. This would degrade the performance of routers. Also, they are backward-incompatible. They utilize the IP header fields to mark the traceback data, overriding their conventional purpose. Moreover, they burden the routers in the attack path by marking every packet it forwards, and disturbs the routers during the traceback process also. Whereas SPITRI is capable of tracing the attacker from the victim itself with a single traceback message, without demanding additional storage or disturbing the existing infrastructure. MRT, MORE and RIHT can traceback only till the edge router, as they initialize the marking field to „0" at the attacker"s border router in the packet marking process. But, SPITRI identifies the attacker or the attacker LAN, as it initializes the Path Information with the interface ID through which the attack packet enters.

IV. RESULT AND DISCUSSION

The computation overhead and accuracy are analyzed using NS-2 integrated with BRITE generated topology. The storage requirement is analyzed using the CAIDA topology datasets described in chapter 1. The accuracy of SPITRI is also confirmed by simulating SPITRI on CAIDA topology datasets.

The proposed work traced back with the help of ICMP traceback message using router interface. So, the efficiency of the proposed scheme is analyzed by simulating and comparing with 1) Original ICMP traceback and 2) the state-of-the-art router interface based approach RIHT.

SPITRI is evaluated by comparing with the original ICMP traceback using the following performance metrics 1) Number of ICMP Packets Required to Traceback 2) Number of Packets Required Reconstructing the Full Path. 3) Path Reconstruction Time 4) Bandwidth Overhead 5) Accuracy.
Traceback methods dependant on more number of packets is normally time consuming and have the prospect of producing false positives. As mentioned, ITrace message with forward or backward link enables the victim to identify 2 routers in an attack path, and ITrace message with both links facilitate the victim to identify 3 routers in the attack path. Hence, if an attack path is of 'h' hops, at least, h/2' ITrace messages (Forward or Backward Link) would be required to identify one attacker. Likewise, at least „h/3" ITrace messages (Both Links) would be required to identify one attacker.

SPITRI needs only „1" ICMP packet (SPITRI message) generated from the border router to traceback one attacker.

From a one packet, it is capable of follow the way till the boundary router and the interface to which the attacker is associated.

Simulating both SPITRI and ITrace in the BRITE generated topology it is also demonstrated using the graph plotted in Figure 5 that ITrace requires huge number of ITrace message compared to SPITRI.

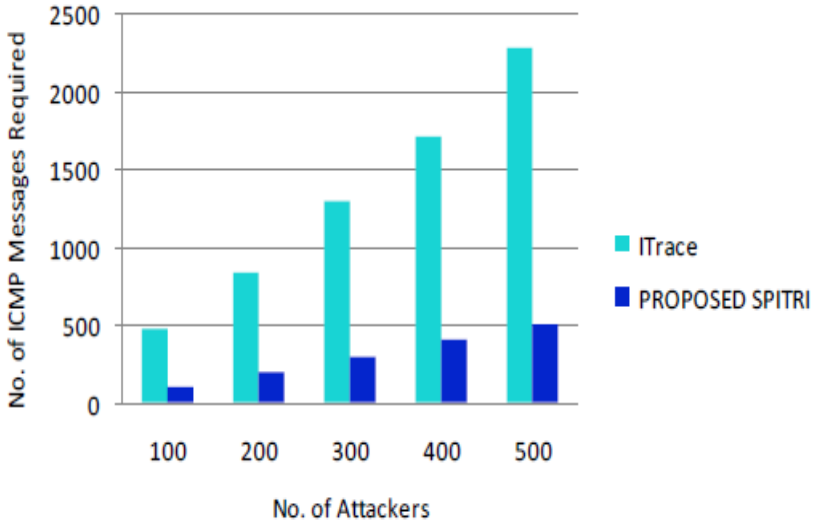

Figure 5 Number of ICMP messages required in reconstructing the attack path

In ITrace scheme, path reconstruction probability is dependent on the path length whereas in SPITRI method, SPITRI message from the border router is sufficient to trace till the attacker. Majority of the network path is less than 32 hops. So, the probabilities of reconstruction of 16 hops and 32 hops attack path on receipt of „Np" packets using ITrace (Forward or Backward Link), ITrace (Both Links) and SPITRI are analyzed. Figure 5 depicts the path reconstruction probability, if SPITRI and ITrace are sent at a low probability of 1/20000. Figure 6 shows that SPITRI requires less number of packets compared to ITrace irrespective of the path length and the probability at which the traceback message is generated.

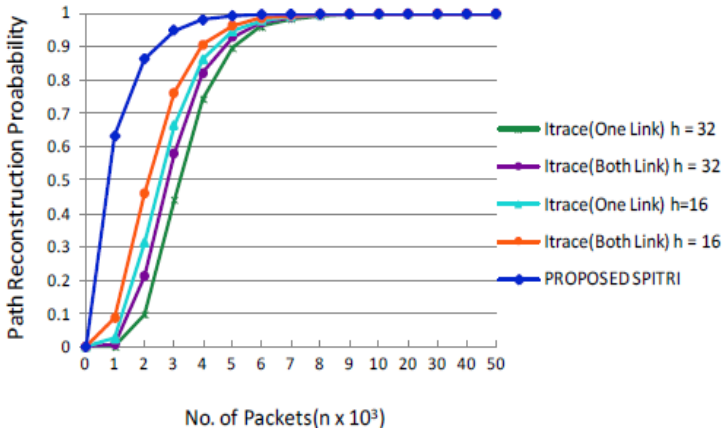

Figure 6: Number of packets required (p=1/2000)

Path reconstruction time: In ITrace simulation it is made sure that all the routers in the path have generated at least one ICMP message. In order to maintain uniformity, in the computation of reconstruction time of ITrace, waiting time is not included. The average path reconstruction time of 50 runs is noted. It is observed that ITrace requires relatively larger time compared to SPITRI due to the dependency of more number of ITrace Packets and the complexity involved. In ITrace, the number of ICMP packets is dependent on the number of ops in the attack path whereas SPITRI will identify any number of hops with a single ICMP packet. In SPITRI, the number of hops merely decides the number of iteration involved in the path reconstruction process. Figure 7 depicts that SPITRI can identify the attacker faster compared to ITrace.

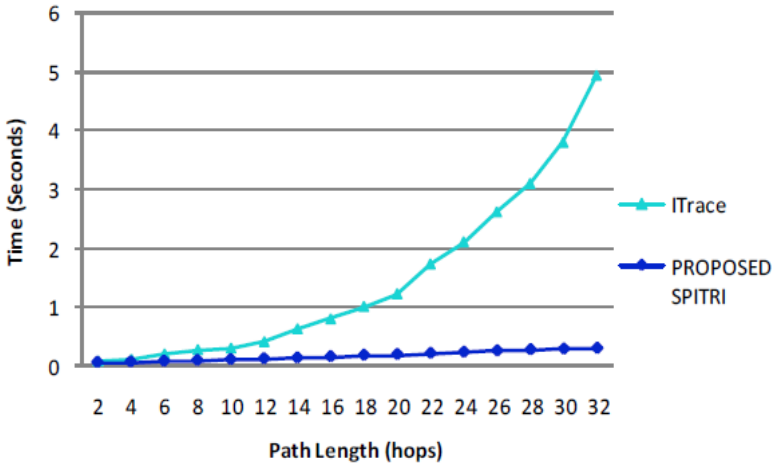

Figure 7: Average reconstruction time

The number of false negative nodes using both the schemes is depicted in Figure 8. The false negative nodes in ITrace are higher with the increase in number of attackers. In ITrace, the attack path can be reconstructed only after collecting all the ITrace messages associated to that path. the number of attackers raised, the expected number of ITrace messages also raised.

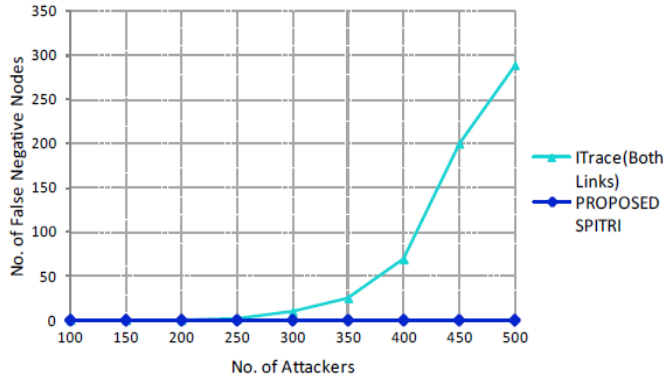

Figure 8 Number of attackers Vs false negative nodes

When a router fails to send an ITrace message, it would lead to incomplete path. Incomplete paths would lead to false negatives. Moreover, the packet to be traced is selected using pseudo random number generator and so, the chances of sending duplicate ITrace messages are also high. Summarizing the above, the evaluation of ITrace and SPITRI is given in Table 2. It indicates that SPITRI is more efficient compared to ITrace in terms of convergence, reconstruction time, bandwidth overhead and accuracy.

Table 2 Summary of evaluation of SPITRI and ITrace

Metrics ITrace	**Proposed**	**SPITRI**
Number of ICMP Packets Required to Traceback (to trace 300 attackers)	1296	300
Number of IP Packets Required to Reconstruct the Full Path (to trace one attacker 16 hops away from victim,	65761	20000
Path Reconstruction Time (to trace an attacker 16 hop away from Victim)	783 ms	141 ms
Net Increase in Traffic near Victim (p □1/ 20000)	0.1%	0.005%
No. of False Negative (to trace 300 attackers)	10	0

V. CONCLUSION

The proposed paper is proved to be efficient than ITrace in terms of number of ICMP packets needed, number of attack packets needed, bandwidth overhead, reconstruction time and accuracy. Bandwidth overhead acquired by SPITRI is nearly number of hop times less than the one in ITrace. According to CAIDA dataset, SPITRI tracebacks 13000 attackers with an accuracy of 95.98%. Compared to the state-of-the-art router interfaced approaches, SPITRI is demonstrated to be efficient in terms of computation overhead, storage overhead, and router processing overhead and backward compatibility. According to CAIDA dataset, it occupies less than 1.25KB in 99.99% of the routers, which is 256 times lesser than the state-of-the-art router interface based technique. SPITRI is backward compatible. It can handle fragmented packets. It does not require major rework in the IP protocol implementation which is an essential feature of a feasible traceback scheme. It can also trace back large scale attacks faster compared to the existing schemes and can identify any number of attackers with their corresponding ICMP packet, victim itself. The processing is removed at the routers in traceback.

REFERENCES

[1] FORBES 2014, The Largest Cyber Attack in History has been Hitting Hong Kong Site, NEW JERSEY.
[2] Hussain, A, Heidemann, J & Papadopoulos, C 2003, „A framework for classifying denial of service attacks", Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ed. Anja, ACM, Karlsruhe, pp. 99-110.
[3] S. Vincent and J. Raja, "A Survey of IP Traceback Mechanisms to overcome Denial-of-Service Attacks," in Proc. networking, VLSI and signal processing (ICNVS), 2010.
[4] A. Belenky and N. Ansari, "IP Traceback With Deterministic Packet Marking," in Proc. IEEE Communications Letters, 2003.
[5] R. Stone, "CenterTrack: An IP Overlay Network for Tracking DoS Floods," in Proc. USENIX Security Symposium (SSYM), 2000.
[6] H. Burch, "Tracing Anonymous Packets to Their Approximate Source," in Proc. 14th Systems Administration Conference (LISA), 2000.
[7] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, B. Schwartz, and S. Kent, "Single-Packet IP Traceback," in Proc. IEEE Transactions on Networking, 2002.
[8] A. Izaddoost, M. Othman, and M. Rasid, "Accurate ICMP Traceback Model Under DoS/DDoS Attack," in Proc. Advanced Computing and Communications (ADCOM), 2007.
[9] Z. Khan, N. Akram, K. Alghathbarl, M. She, and R. Mehmoodl, "Secure Single Packet IP Traceback Mechanism to Identify the Source," in Proc. IEEE Internet Technology and Secured Transactions (ICITST), 2010.
[10] B. Stelte, "ISP Traceback - Attack Path Detection," in Proc. IEEE Communications and Network Securit, 2013.
[11] C. Gong and K. Sarac, "A More Practical Approach for Single-Packet IP Traceback Using Packet Logging and Marking," in Proc. IEEE Transactions on Parallel and Distributed Systems, 2008
[12] Belenky, A & Ansari, N 2003, „IP Traceback with deterministic packet marking", IEEE Communication Letters, Vol. 7, no. 4, pp. 162-164.

