Framework Based Channel State Information with Dynamic Path Identifiers

K.Gandhimathi[1], T.DhivyaBharathi[2], S.Periyanayaki Jenifer Mary[3], R.Pragathi[4]

Idhaya Engineering College for Women, Chinnasalem, Villupuram District.

Abstract

Today anyone with black hat mindset can launch the attack. Availability of tools and cost effective attack service have made DDOS attacks more dangerous and more temporal than ever. These attacks have become very complex with respect to time scale such that existing security algorithms are not those much sufficient to counter and protect against this attacks. In existing approaches Path Identifiers (PIDS) used are static which makes easy for attackers to launch DDOS attacks. The proposed scheme uses Dynamic Path Identifiers (DPID) is simple to implement, introduces no bandwidth overhead, low computational overhead and has low fault probability. In D-PID, PID of an inter domain path connecting two domains is kept secret and changes dynamically.

Keywords—Ddistributed denial-of-service (DDoS) attacks, path identifiers, Security, Inter-domain path

I.INTRODUCTION

Today, Internet is an essential part of everyday life and many important and crucial services like banking, shopping, transport, health, and communication are partly or completely dependent on the Internet. Also it was originally designed for openness and scalability without much concern for security. Security is a fundamental component of every network design. A security policy defines what people can and can't do with network components and resources. For secure communication to take place desirable security aspects such as confidentiality, authentication, message integrity and non-repudiation is to be considered. Classes of attack might include passive monitoring of communications, active network attacks, close-in attacks, exploitation by insiders and attacks through the service provider. A system must be able to limit damage and recover rapidly when attacks occur..In recent years, there are increasing interests in using path identifiers (PIDs) as inter-domain routing on objects to prevent DDOS attack. However, the PIDs used in existing approaches are static, which makes it easy for attackers to launch distributed denial-of-service (DDoS) flooding attacks. To address this issue, the DPID framework has been implemented. In D-PID, two adjacent domains periodically update the PIDs between them and install the new PIDs into the data plane for packet forwarding. Even if the attacker obtains the PIDs to its target and sends the malicious packets successfully, these PIDs will become invalid after a certain period and the subsequent attacking packets will be discarded by the network. Moreover, if the attacker tries to obtain the new PIDs and keep a DDoS flooding attack going, it not only significantly increases the attacking cost but also makes it easy to detect the attacker.

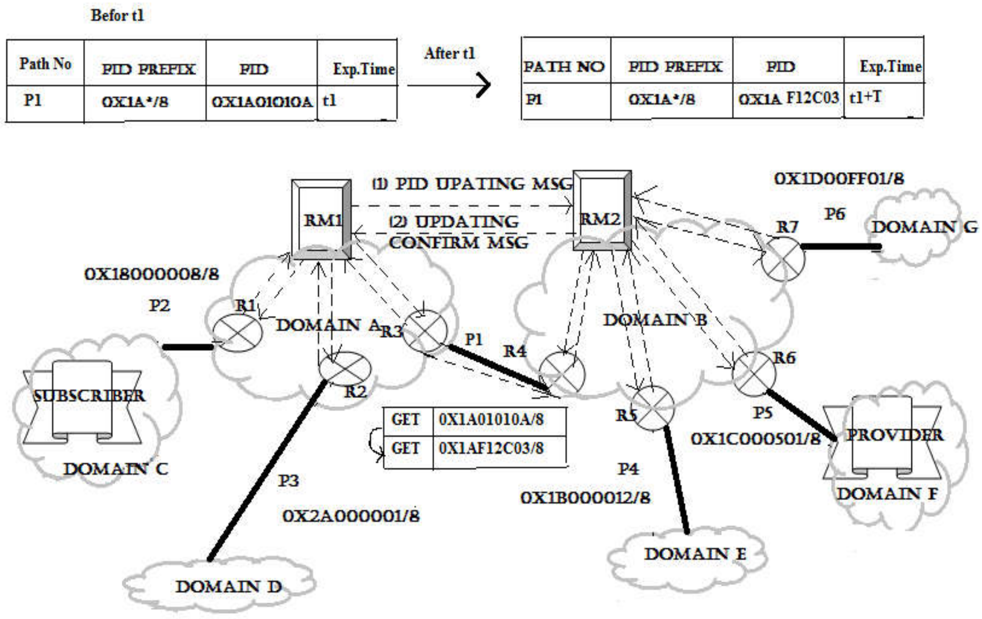
II. PROPOSED SYSTEM

In the proposed work, DDOS attack, PID forgery and Spoofing attacks are concentrated. So a new prototype named (DPID) is proposed. In D-PID, two adjacent domains periodically update the PIDs between them and install the new PIDs into the data plane for packet forwarding. Even if the attacker obtains the PIDs to its target and sends the malicious packets successfully, these PIDs will become invalid after a certain period and the subsequent attacking packets will be discarded by the network. Moreover, if the attacker tries to obtain the new PIDs and keep a DDoS flooding attack going, it not only significantly increases the attacking cost, but also makes it easy to detect the attacker. We Propose the D-PID design by addressing the following challenges. First, how and how often should PIDs change while respecting local policies of autonomous systems (ASes)? Second, since inter-domain packet forwarding is based on PIDs that change dynamically, it is necessary to maintain legitimate communications while preventing illegal communications when the PIDs change. To address this challenge D-PID lets every domain distribute its PIDs to the routers in the domain. Third, the overheads incurred by changing PIDs should be kept as small as possible. This includes not only the overhead in negotiating PIDs by neighboring domains, but also the overhead for a domain to distribute the updated PIDs to routers in the domain, and that for transmitting content request messages resent by content consumers. To address this challenge, the PID prefix assigned to an inter-domain path is unique among the PID prefixes assigned by the two domains connected by the inter-domain path.

ADVANTAGES

- D-PID does help preventing DDoS flooding attacks since it not only imposes significant overhead for the attacker to launch DDoS flooding attacks, but also makes it easier for the network to detect the attacker.
- Incurs little overheads.

III.SYSTEM ARCHITECTURE



Befor t1

Path No	PID PREFIX	PID	Exp.Time
P1	OX1A*/8	OX1AO1O1OA	t1

After t1

PATH NO	PID PREFIX	PID	Exp.Time
P1	OX1A*/8	OX1A F12C03	t1+T

GET	0X1A01010A/8
GET	0X1AF12C03/8

Figure 3.1-System Architecture

3.1 HOW PIDS DYAMICALLY CHANGES

Two adjacent domains periodically update the PIDs between them and install the new PIDs into the data plane for packet forwarding. Even if the attacker obtains the PIDsto its target and sends the malicious packets successfully, these PIDs will become invalid after certain Period and the subsequent attacking packets will be discarded by the network. Thus, the core idea of DPID is to dynamically change the PIDof an inter-domain path. In particular, for a given path connecting two neighboring domains A and B, it is assigned a PIDand an update period TPID. The update period TPIDrepresents how long the PIDof the path should be changed since the PIDis assigned. For instance, if path P1is assigned PID1 at time t, the RMsin the two domains should negotiate a new PID (i.e., PID2) for P1at time t + TPIDand a new update period T′PID, by using the negotiation process described in Sec. III-B. At time t + TPID + T′PID, the two RMs will negotiate another new PID (i.e., PID3) for P1 . Once the new PID (*i.e.*, PID2) is assigned to the path, the RMs in domains A and B then distribute the new PID(*i.e.*,PID2) to the routers in domains A and B. After that, the RMsappend the new PID (i.e., PID2) onto GET messages if the path is chosen to carry the corresponding data packets. At the same time, the border routers forward data packets based on the new PID (i.e., PID2). Since some GET packets are forwarded from domain A (or B) to domain B (or A) by using the old PID (i.e., PID1) of the path, the old PID is still valid until t + TPID + T′PID. That is, the update period of a path is fixed. The new PIDof the path is still known only by the two domains.

3.2 PID NEGOTIATION PROCESS

PID of a path connecting two neighboring domains, it is associated with an update period. At the end of the update period, the initiative RM in the two neighboring domains randomly chooses a new PID from the PID-prefix assigned to the path, and sends the chosen PID to the RMin another domain. If the later one accepts the chosen PID, it sends a confirmation message back to the initiative RM, Otherwise, then later RM chooses another PID from the PID-prefix and sends the chosen PID back to the initiative RM .if the update period of a path is not fixed, the two domains can also negotiate the new period used to update the PID of the path. The RM in a domain Aneeds to distribute the new PIDto the routers in that domain so that the new PID can be used to forward data packets. To achieve this, the RM simply sends a PIDupdate message to every border router in the domain. The PIDupdate message contains the path and its corresponding new PID. When a border router receives the PID update message, it updates its inter-domain routing table. After that, it sends an acknowledgement message to the RM. When the RM receives the acknowledgement messages from all border routers, it then updates its PID table. After that, the RM appends the new PIDinstead of the old one onto the GET messages when it forwards them. Accordingly, the corresponding data packets will be forwarded from the neighboring domain to domain A by using the new PID.

3.3 PID DISTRIBUTION PROCESS

In the PID distribution process, assuming thatdomains Aand Bchange path P1’s PID from 0x1A01010Ato 0x1AF12C03 during the *PID* negotiation process. In addition,before path P1’s new PIDis distributed, the interdomainrouting table of router *R2* is shown at the bottom left corner. Note that in order to maintain legal communications the inter-domain routing tableof router *R2* has two entries for path *P1* : 0x1A01010A thatis used by the RMbefore the negotiation and will be replacedby 0x1AF12C03 after the negotiation, and 0x1AE581FA that Is used in the previous negotiation and has been replaced by 0x1A01010A.

IV.METHODOLOGY

The entire work is divided into five different divisions. They are:

- Network topology Construction
- Path Selection
- Packet Sending
- Packet Marking and Logging
- Path Reconstruction

4.1 NETWORK TOPOLOGY CONSTRUCTION

A network can include multiple connected nodes. The nodes in a network can be configured in various topologies. Once a network topology has been defined, the user can configure one or more end-to-end connections that can span multiple nodes, an operation is referred to herein as provisioning. For end-to-end connection between two nodes, a physical path must be selected and configured. Each set of physical connections that are provisioned creates an end-to-end connection between the two end nodes that supports a virtual point-to-point link (referred to herein as a virtual path or VP). The resulting VP has an associated capacity and an operational state, among other attributes.

In a network, VPs may be provisioned statically or dynamically. For example, a user can identify the nodes which will comprise the virtual path. The selection of nodes may be based on any number of criteria, such as latency, cost, distance traveled in the network and the like. Alternatively, the VP may be provisioned dynamically using any one of a number of methods. The provisioning information may then be forwarded to all the nodes in the network to store information in each node's network topology database. Each node periodically updates this information to efficiently maintain resources and in case of path failure, effectively allocate appropriate resources needed for specific virtual path for path restoration.

4.2 PATH SELECTION

A switch/router dynamically selects a path from multiple available paths between a source destination pair for a frame. A hash function generates a hash value from frame parameters such as source ID, destination ID, exchange ID, etc. The hash value is given as an input to a plurality of range comparators where each range comparator has a range of values associated with it. If the hash value falls within a range associated with a range comparator, that range comparator generates an in-range signal. A path selector module detects which range comparator has generated the in-range signal, and determines a path associated with that range comparator from previously stored information. The frame is transmitted via the selected path. The ranges associated with each range comparator can be non-overlapping and unequal in size. The number of range comparators can be equal to a number of selected multiple paths.

4.3 PACKET SENDING

Packet sending is the basic method for sharing information across systems on a network. Packets are transferred between a source interface and a destination interface, usually on two different systems. When you issue a command or send a message to a nonlocal interface, your system forwards those packets onto the local network. If the destination address is not on the local network, the packets are then forwarded to the next adjacent network, or hop. In many cases this gateway is the router. If the router's forwarding tables know where the packet should go, the router will send the packet out along the appropriate route, If the router does not know where the destination network is, it will forward the packet to its defined gateway, which will repeat the same process.

4.4 PACKET MARKING

One of the main difficulties in the detection and prevention of Distributed Denial of Service (DDoS) attacks is that the incoming packets cannot be traced back to the source of the attack, because (typically) they contain invalid or spoofed source IP address. For that reason, a victim system cannot determine whether an incoming packet is part of a DDoS attack or belongs to a legitimate user. Various methods have been proposed to solve the problem of IP traceback for large packet flows. These methods rely on the assumption that they can gather a sufficient number of packets from the same source, in order to reconstruct the traversed path or to determine the source address. Packet marking scheme is introduced which enables the unique identification of the path that each incoming packet has traversed, relying only on the information inside that packet. In packet marking, routers alter the IP header of the traversing packets (they mark them) in order to notify the end host of their presence on the route. The end host can gather those markings and rebuild the traversed path for large packet flows. Packet marking schemes encode information about the path a packet traverses in the packet itself, usually in rarely-used fields within the IP header. Apart from the well-known issue of finding enough space in the current IP header in which to place traceback information, another common problem with packet marking schemes arises from the additional computational tasks placed on routers during the marking process.

4.5 PACKET LOGGING

In packet logging, the IP packet is logged at each router through which it passes. Historically, packet logging was thought to be impractical because of enormous storage space for packet logs. Hash-based IP traceback approach records packet digests in a space-efficient data structure, bloom filter, to reduce the storage overhead significantly. Routers are queried in order to reconstruct the network path. the information required to achieve traceback is either stored at different points (mostly on routers) along the path that a packet traverses or that path and usually other incidental paths are analyzed to gain information that will be used in traceback.

4.6 PATH RECONSTRUCTION

To reconstruct the path of a packet and identify the source of the attack, the victim requires a map of the routers. The victim matches packet markings with the routers on the map and can thus reconstruct the attack path. Obtaining or constructing this map is not difficult. A number of tools are available that can be used to obtain a map of the the routers and the Internet. If a router commits logging operation on an attack packet, examining digest tables at that router will not only confirm that router is in the attack path, but also find out its upstream router in the attack path since each digest table is annotated with an upstream router's ID number. Given an attack packet and victim, the traceback server could infer the last hop router and whether the last hop router committed logging operation based on the logging flag bit carried by the attack packet.

- If the traceback server infers a router logged the attack packet, examining the digest tables at that router would identify its upstream router in the attack path.
- If the traceback server infers a router didn't log but marked the attack packet, querying the neighbor routers of that router in the RPF manner and examining the digest tables on these neighbor routers would identify the upstream router. The attack graph can be constructed using those two methods alternately.

V IMPLEMENTATION

Eclipse can be used to develop applications. The overall operation can be shown in the following flow diagram 5.1

The selected file is send to the appropriate router to correctly reach the desired destination node. Optimized path is selected dynamically and checked then the destination node receives the file which is send by the source node from the appropriate router and the receiver can receive the file.

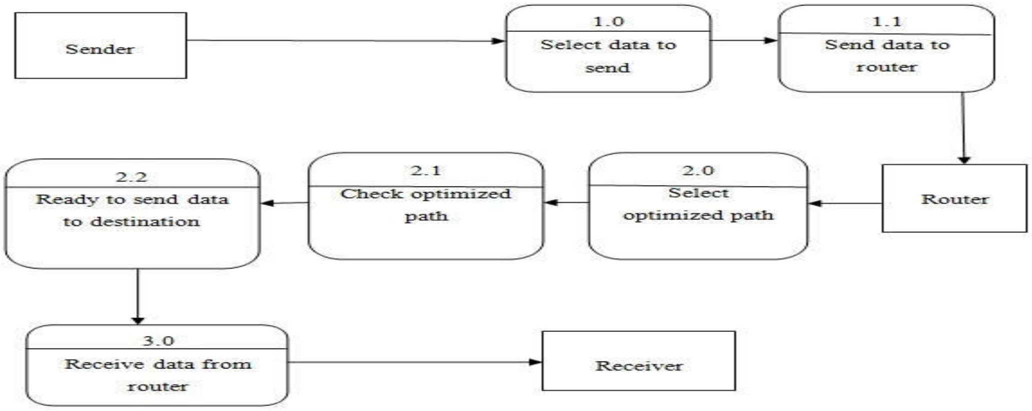


Fig 5.1 Flow diagram

VI.CONCLUSION

The proposed work performs effectively and resolves flooding Attacks with random anonymous secure path identifiers. The existing PID is replaced with dynamic generation feature and then improved as D-PID. Most importantly D-PID gets updated dynamically. Dynamic path identifier is an eminent way to detect and prevent the distributed denial of service attack. The detect information includes the need if anonymous secure unique identifiers and timestamp value. The idea can be implemented in large scale to facilitate better safety to the internet in the future work.

REFERENCES

1. *H. Luo, Z. Chen, J. Cui, H. Zhang, M. Zukerman, C. Qiao, "CoLoR: an information-centric internet architecture for innovations," IEEE Network, vol. 28, no. 3, pp. 4 - 10, May 2014.*
2. *602 Gbps! This May Have Been the Largest DDoS Attack in History.http://thehackernews.com/2016/01/biggest-ddos-attack.html.*
3. *J. Francois, I. Aib, and R. Boutaba, "Firecol: a Collaborative Protection Network for the Detection of Flooding ddos Attacks," IEEE/ACM Trans. on Netw., vol. 20, no. 6, Dec. 2012, pp. 1828-1841.*
4. *P.Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, P. Nikander, "LIPSIN: Line Speed Publish/Subscribe Inter- networking," in Proc. SIGCOMM'09, Aug. 2009, Barcelona, Spain, pp. 195 - 206.*
5. *Z. Duan, X. Yuan, J. Chandrashekar, "Controlling IP Spoofing through Interdomain Packet Filters," IEEE Trans. on Depend. andSecure Computing, vol. 5, no. 1, pp. 22 - 36, Feb. 2008.*
6. *D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, "Accountable Internet Protocol (AIP)," In Proc.SIGCOMM' 08, Aug. 2008, Seattle, WA, USA.*
7. *B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y. Hu, "Portcullis: Protecting connection setup from denial-of-capability attacks," In Proc. SIGCOMM'07, Aug.2007, Kyoto, Japan.*
8. *H. Ballani, Y. Chawathe, S. Ratnasamy, T. Roscoe, S. Shenker, "Off by default!," In Proc. HotNets-IV, Nov. 2005, College Park, MD, USA.*
9. *Hongbin Luo, Zhe Chen, Jiawei Li, and Athanasios V. Vasilakos, "Preventing Distributed Denial-of-Service Flooding Attacks with Dynamic Path Identifiers" in IEEE Transactions On Information And Forensics Security, 2016*
10. *S. T. Zargar, J. Joshi, D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," IEEE Commun. Surv. & Tut., vol. 15, no. 4, pp. 2046 - 2069, Nov. 2013.*
11. *Y. Xiang, K. Li, W. Zhou, "Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics," IEEE Trans. on Inf. Foren. and Sec., vol. 6, no. 2, pp. 426 - 437, May 2011.*
12. *A. Yaar, A. Perrig, and D. Song, "SIFF: a stateless internet flow filter to mitigate DDoS flooding attacks," In Proc. IEEE Symposium on Security and Privacy, May 2004, Oakland, CA, USA.*