IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 3, March 2018

# Preventing and Detecting Distributed Denial-of-Service Flooding Attacks with Dynamic Path Identifiers

**Sable D.V.[1], Sakhare S S.[2], Khule Y.R.[3], Salve P.M.[4], Tonape Y.L[5]**

Student, Computer Science & Engg, Bhagwant Institute of Technology, Barshi, India[1,2,3,4]

Assistant Professor, Computer Science & Engg, Bhagwant Institute of Technology, Barshi, India[5]

**Abstract:** In recent years, there are raising interests in using path identifiers (PIDs) as inter-domain routing objects. However, the PIDs used in existing approaches are static, which made it easy for attackers to launch distributed denial-of service (DDoS) flooding attacks. To address this issue, in this paper, we present the design, implementation, and evaluation of D-PID, In that framework that uses PIDs negotiated between neighbouring domains as inter-domain routing objects. In DPID, the PID of an inter-domain path connecting two domains is kept top secret and changes dynamically. In this paper we describe in detail how neighbouring domains negotiate PIDs, how to maintain ongoing communications when PIDs change. We build a 42-node prototype comprised by six domains to verify D-PID's beneficial and conduct broad simulations to evaluate its effectiveness and cost. The results from both simulations and experiments show that D-PID can effectively prevent DDoS attacks.

**Keywords:** Inter-domain routing, security, distributed denial-of- service (DDoS) attacks, path identifiers.

## I. INTRODUCTION

Distributed denial-of-service (DDoS) flooding attacks are very harmful to the Internet. In a DDoS attack, the attacker uses widely distributed zombies to send a large amount of traffic to the target system, thus preventing legitimate users from accessing to network resources. For example, a DDoS attack against BBC sites in Jan. 2016 reached 602 gigabits per second and "took them down for at least three hours". At the same time, in recent years there are increasing interests in using path identifiers PIDs that identify paths between network entities as inter-domain routing objects, since doing this not only helps addressing the routing scalability and multi-path routing issues, but also can facilitate the innovation and adoption of different routing architectures[10]. For instance, Godfrey et al. proposed pathlet routing, in which networks advertise the PIDs of pathlets throughout the Internet and a sender in the network constructs its selected pathlets into an end-to-end source route. Koponen et al. further argued in their insightful architectural paper that using pathlets for inter-domain routing can allow networks to deploy different routing architectures, thus encouraging the innovation and adoption of novel routing architectures. Luo et al. proposed an information-centric internet architecture called CoLoR that also uses PIDs as inter-domain routing objects in order to enable the innovation and adoption of new routing architectures, as in[10].

There are two different use cases of PIDs in the aforementioned approaches. In the first case, the PIDs are globally advertised. As a result, an end user knows the PID(s) toward any node in the network. Accordingly, attackers can launch DDoS flooding attacks as they do in the current Internet. In the second case, conversely, PIDs are only known by the network and are secret to end users (as in LIPSIN and Color[11] ). In the latter case, the network adopts an information-centric approach where an end user (i.e., a content provider) knows the PID(s) toward a destination (i.e., a content consumer) only when the destination sends a content request message to the end user.

After knowing the PID(s), the end user sends packets of the content to the destination by encapsulating the PID(s) into the packet headers. Routers in the network then forward the packets to the destination based on the PIDs. It seems that keeping PIDs secret to end users , makes it difficult for attackers to launch DDoS flooding attacks since they do not know the PIDs in the network.

However, keeping PIDs secret to end users is not enough for preventing DDoS flooding attacks if PIDs are static. For example, Antikainen et al. argued that an adversary can construct novel zFilters (i.e., PIDs) based on existing ones and even obtain the link identifiers through reverse-engineering, thus launching DDoS flooding attacks. Moreover, attackers can launch DDoS flooding attacks by learning PIDs if they are static.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 3, March 2018

To address this issue, in this paper, we present the design, implementation and evaluation of a dynamic PID mechanism. 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. In particular, our main contributions are twofold. On one hand, 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 .To address this challenge, D-PID lets neighbouring domains negotiate the PIDs for their inter-domain paths based on their local policies. In particular, two neighbouring domains negotiate a PID-prefix (as an IPprefix) and a PID update period for every inter-domain path connecting them. At the end of a PID update period for an inter-domain path, the two domains negotiate a different PID (among the PID-prefix assigned to the path) to be used in the next PID update period. In addition, the new PID of an inter-domain path is still kept secret by the two neighbouring domains connected by the path.

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. For every inter-domain path, the routers in a domain forward data packets based on the PID of the previous PID update period and that of the current PID update period. In addition, D-PID uses a mechanism similar to the one that the current Internet collects the minimum MTU (maximum transmission unit) of networks so that a content consumer knows the minimum update period of PIDs along the path from a content provider to it .Based on this period, the content consumer periodically resends a content request message to the network in order to renew the PIDs along the path. Third, the overheads incurred by changing PIDs should be kept as small as possible. This includes not only the overhead in negotiating PIDs by neighbouring 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.

On the other hand, we build a 42-node prototype comprised by six domains to verify D-PID's feasibility and conduct extensive simulations to evaluate D-PID's effectiveness and overheads. Surprisingly, achieving such benefits only incurs little overheads. Our simulation results show that the number of extra content request messages caused by D-PID is only 1.4% or 2.2% (by using different data traces), when the PID update period is 300 seconds. Even if the PID update period is 30 seconds, the peak PID update rate of a domain is less than 10 per second with a probability higher than 95%, and the maximal PID update rate of all domains is 202 per second, which is significantly less than the peak update rate (1,962 per second) of IP-prefixes in the current Internet [32]. While part of this work has been published in [45], we significantly extend it with the following new contributions. First, we propose an approach for neighbouring domains to negotiate PIDs and to distribute them to routers in a domain. Second, we implemented D-PID in a prototype to verify its feasibility. Third, we conduct extensive simulations to evaluate the effectiveness of D-PID in defending against DDoS flooding attacks.

## II. RELATED WORK

Because of the complexity and difficulty in defending against DDoS flooding attacks, many approaches have been Proposed in past two decades. For instance, filtering-based approaches aim at mitigating DDoS flooding attacks by de-ploying source address filtering at routers[5]. Similarly, IP trace back-based methods trace attacks back through the network toward the attacking sources [8]. In addition, the approaches proposed in, aim at mitigating DDoS attacks by sending shut-up messages to the attacking sources, assuming that they will cooperate and stop flooding. While there are too many literatures, we refer interested readers to [2] for a survey on existing approaches in defending again DDoS flooding attacks. Instead, we outline prior work closely related to this work and compare D-PID with them. A main reason that DDoS flooding attacks proliferate is anode can send any amount of data packets to any destination, regardless whether or not the destination wants the packets. To address this issue, several approaches have been proposed. In the "off by default" approach, two hosts are not permitted to communicate by default. Instead, an end host explicitly signals, and routers exchange, the IP-prefixes that the end host wants to receive data packets from them by using an IP-level control protocol. The D-PID design is similar in spirit, since D-PID dynamically changes PIDs and a content provider can send data packets to a destination only when the destination explicitly sends out a GET message that is routed (by name) to the content provider. However, there are two important differences. First, the "off by default" approach works at the IP-prefix granularity, but D-PID is based on an information-centric network architecture and works at the content granularity. Second, the IP-prefixes that an end host wants to receive packets from are propagated throughout the Internet in the "off by default" approach, which may cause significant routing dynamics if the allowed

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 3, March 2018

IP-prefixes of end hosts change frequently. On the other hand, as pointed out in the capability-based approaches are vulnerable to "denial-of-capability attacks, where compromised computer(s) sends plenty of capability requests to a victim, thus preventing normal users to obtain the capability from the victim.

## III. EXISTING SYSTEM

In existing system, there are increasing interests in using path identifiers (PIDs) as interdomain routing objects. However, the PIDs used in existing approaches are static, which makes it easy for attackers to launch distributed denial-of service (DDoS) flooding attacks. In a DDoS attack, the attacker uses widely distributed zombies to send a large amount of traffic to the target system.

Disadvantages :

- PIDs are not secret to end users makes it easy for attackers to launch DDoS flooding attacks.
- PIDs are static, IP address and routing is same up to the destination.

## IV. PROPOSED SYSTEM

In this paper, PIDs that identify paths between network entities as inter-domain routing objects, since doing this not only helps addressing the routing scalability and multi-path routing issues, but also can facilitate the innovation and adoption of different routing architectures. In DPID, the PID of an inter-domain path connecting two domains is kept secret and changes dynamically. We describe how neighbouring domains negotiate PIDs, how to maintain ongoing communications when PIDs change.

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 invalid after a certain period..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. In particular, our main contributions are twofold. On one hand, 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 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 neighbouring 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 of proposed system:

- In tis system used D-PID is secret to end users since prevention of DDOS attack.
- D-PIDs are dynamically changes from domain to domain
- D-PID helps preventing DDoS flooding attacks it not decreases the overhead for the attacker to launch DDoS flooding attacks, but also makes it easier for the network to detect the attacker.
- Surprisingly, achieving such benefits only incurs little overheads.

4.1Color n/w Architecture

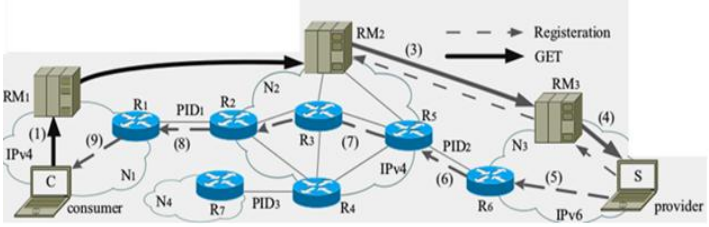


Fig 1: Illustration for the basic operations in Color.

Color is a receiver-driven information centric network architecture that assigns unique and persistent content names (or service identifiers, SIDs) to content chunks. Color assigns intrinsic secure self-certifying node identifiers (NIDs) to network nodes and ASes so that authenticating a node/AS does not require an external authority such as ICANN, thus improving security and privacy. In addition, two neighbouring domains negotiate a PID for every inter-domain path

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 3, March 2018

between them and the PID is only known by them. The two domains then use the PIDs assigned to their inter domain paths to forward packets from one domain to the other. For this purpose, the routers in a domain maintains an interdomain routing table, which records the PID of each inter domain path and the border router that the PID originates, as illustrated in Fig. 1. For instance, the border router in domain N2 connecting PID2 in Fig. 1 is R5. On the other hand, each domain is free to choose its preferred intra-domain routing architecture so that a domain A uses IPv4 for intra-domain routing while another domain B may use IPv6 for intra-domain routing. Furthermore, every domain in the Internet maintains a logically centralized (but may be physically distributed) resource manager (RM) used to propagate the reachability information. Particularly, when a content consumer wants to obtain a piece of content, it sends out a GET message to its local RM. If the desired content is hosted by a local node, the RM forwards the GET message to that node. Otherwise, the RM forwards the GET message to the RM in a neighbouring domain (toward the content provider) over a secure channel between the two RMs (because of the use of intrinsic secure identifiers). During this process, the PIDs of inter-domain paths from the content provider to the content consumer are determined. The content provider then sends the desired content to the content consumer by embedding the collected PIDs into headers of packets for the desired content.

4.2 Why dynamically changing PIDs

In this subsection, we explain why it is necessary to dynamically change PIDs in Color. To this end, we first present two approaches to learning PIDs whey they are static. We then present an example to show that an attacker can launch DDoS attacks when he have learnt some PIDs in the network.

1) Two approaches to learning PIDs: The first approach to learning PIDs is GET Luring, where an attacker uses an end host to register normal content names into the network, thus luring GET messages from content consumers. Since the corresponding PIDs are carried by the GET messages, the attacker then can learn a part of PIDs in the network.

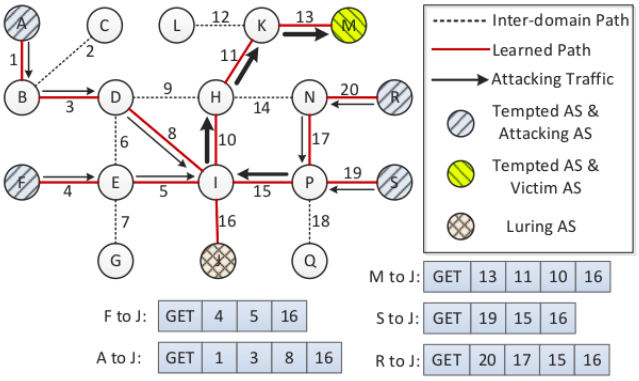


Fig 2. Illustration for the GET luring.

Fig. 2 illustrates the process of GET luring. For ease of presentation, we call the AS where the attacker locates as a luring AS and the ASes that send GET messages to the luring AS as tempted ASes. Each node in Fig. 2 represents an AS in the Internet, AS J is the luring AS, and ASes A, F, M, R, and S are the tempted ASes. At the beginning, ASJ registers content names into the network. Then, ASes A, F, M, R, and S are lured to send GET messages to AS J. The GET messages received by AS J are shown at the bottom of Fig. 2. The attacker then learns the corresponding PIDs in the network, which are represented by solid lines in Fig. 2. Another approach to learning PIDs is botnet cooperation. In botnet cooperation, an attacker is assumed to have controlled a distributed botnet by using various methods such as worms or instant messaging applications. In particular, zombies in the botnet register content names to the network.

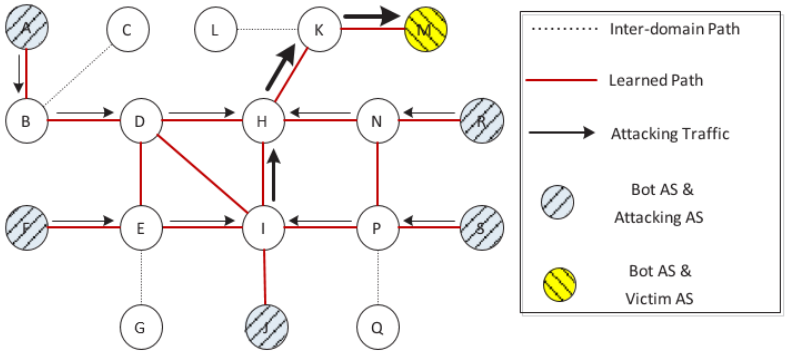


Fig 3. Illustration for the botnet cooperation

and send GET messages mutually, thus learning the PIDs in the network. Fig. 3 illustrates botnet cooperation, assuming that an AS in which one or more bots locate is called a bot AS. In Fig. 3, ASes A, F, M, R, and S are bot ASes. When a zombie in bot AS A sends a GET message to another zombie in bot AS R, the inter-domain paths A−B, B −D, D −H, H −N, and N −R are learned. Similarly, other inter-domain paths marked with bold lines in Fig. 3 can also be learned.

2) Launching DDoS Attacks: Once the attacker has learned a part of PIDs in the network (through GET luring, botnet cooperation, or other possible approaches), it can freely send packets along the paths represented by the learned PIDs. We assume that the attacker can compromise a number of computers along the paths as zombies, by using similar methods with the ones in the current Internet (e.g., by using worms). Note that this is a pessimistic assumption since the integrality of a content in information-centric networking is usually easy to verify. Then the attacker can order the zombies to flood a victim that should also be along the learned paths. We call such a process as the attacking stage.

Fig. 2 and Fig. 3 illustrate the attacking stage. We call the ASes where the compromised computers (that flood the victim) locate as attacking ASes and the AS where the victim locates as the victim AS. Note that an AS may play multiple roles, e.g., a tempted AS at the PID learning stage may be an attacking AS at the attacking stage. In Fig. 2, AS M is the victim AS, and ASes A, F, R, and S are the attacking ASes that are compromised by the attacker and can flood the victim by using the learned PIDs, as illustrated by the arrowed lines in Fig. 2. In Fig. 3, AS M is the victim AS, ASes A, F, J, R, and S are the attacking ASes, and the attacking traffic is represented by the arrowed lines.

From the above descriptions, one can see that it is possible for an attacker to launch DDoS attacks if PIDs are kept secret but static. In addition, since the PIDs carried by data packets are popped out domain-by-domain, the victim does not know the PIDs to the attackers. Accordingly, it cannot trace back them. One may argue that we should not pop out the PIDs when data packets pass through domains. In that case, however, an attacker can try to hide himself by prepending some invalid PIDs at data packets. For instance, the actual PIDs from the content provider S to the content consumer C in Fig. 1 are PID2 and PID1. In order to hide himself, S can prepend an invalid PID (e.g., PID6 not shown in Fig. 1) before PID2 and PID1. This way, the content consumer C cannot easily find S even if we do not pop out PIDs during the packet forwarding process. Therefore, we propose to defend against DDoS attacks by dynamically changing PIDs.

## V. CONCLUSIONS

In this paper, we have presented the design, implementation and evaluation of D-PID, a framework that dynamically changes path identifiers (PIDs) of inter-domain paths in order to prevent DDoS flooding attacks, when PIDs are used as inter-domain routing objects. The proposed approach is very accurate and can be implemented with low overheads. This system is useful in detection and prevention of D-DOS attack and also shows the compromised nodes in network. The results show that the time spent in negotiating and distributing PIDs are quite small (in the order of ms) and D-PID is effective in preventing DDoS attacks.

## REFERENCES

[1]. .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.
[2]. .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.
[3]. .A. Yaar, A. Perrig, D. Song, “StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense,” IEEE J. on Sel. Areas in Commun., vol. 24, no. 10, pp. 1853 - 1863, Oct. 2006.
[4]. .H. Wang, C. Jin, K. G. Shin, “Defense Against Spoofed IP Traffic Using Hop-Count Filtering,” IEEE/ACM Trans. on Netw., vol. 15, no. 1, pp. 40 - 53, Feb. 2007.
[5]. .Z. Duan, X. Yuan, J. Chandrashekar, “Controlling IP Spoofing through Interdomain Packet Filters,” IEEE Trans. on Depend. and Secure Computing, vol. 5, no. 1, pp. 22 - 36, Feb. 2008.
[6]. .M. Sung, J. Xu, “IP traceback-based intelligent packet filtering: a novel technique for defending against Internet DDoS attacks,” IEEE Trans. on Parall. and Distr. Sys., vol. 14, no. 9, pp. 861 - 872, Sep. 2003.
[7]. .M. Sung, J. Xu, J. Li, L. Li, “Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Information-Theoretic Foundation,” IEEE/ACM Trans. on Netw., vol. 16, no. 6, pp. 1253 - 1266, Dec. 2008.
[8]. .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.
[9]. .X. Yang, D. Wetherall, and T. Anderson, “TVA: A DoS-Limiting Network Architecture,” IEEE/ACM Trans. on Netw., vol. 16, no. 3, pp. 1267 - 1280, Jun. 2008.
[10]. [10] .T. Koponen, S. Shenker, H. Balakrishnan, N. Feamster, I. Ganichev, A. Ghodsi, P. B. Godfrey, N. McKwoen, G. Parulkar, B. Raghavan, J.Rexford, S. Arianfar, D. Kuptsov, “Architecting for innovation,” ACM Comput. Commun. Rev., vol. 41, no. 3, July 2011, pp. 24 - 36.
[11]. [11].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.