International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 2 | ISSN : 2456-3307

Detecting and Isolating On Distributed Denial of Service Attack With Dynamic Path Identifiers

Y.Srinivasa Rao*[1], Chandu Tirupathamma[2]
[1]srinivasu7777@gmail.com, chandu.lakshmi095@gmail.com[2]

ABSTRACT

As of late, there are expanding interests in utilizing path identifiers (PIDs) as between area directing objects.However, the PIDs utilized as a part of existing methodologies are static, which makes it simple for assailants to dispatch distributed denial-of-service(DDoS) flooding assaults. To address this issue, in this paper, we show the outline, usage, and assessment of D-PID, a structure that utilizations PIDs consulted between neighboring areas as between space directing articles. In DPID, the PID of a between space path interfacing two areas kept mystery and changes powerfully. We portray in detail how neighboring areas arrange PIDs, how to keep up continuous correspondences when PIDs change. We construct a 42-hub prototype contained six areas to check D-PID's achievability and direct broad recreations to assess its adequacy and expenses. The outcomes from the two reproductions and trials demonstrate that D-PID can adequately anticipate DDoS assaults.

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

I. INTRODUCTION

Distributed Denial of Service (DDoS) is the organized endeavor to bargain the accessibility of system resources or servers as appeared in figure 1. These attacks make money related misfortunes by hindering true blue access servers and online administrations. To moderate the effect of these attacks solid safeguard components are required that can identify and prevent progressing attacks. Numerous resistance instruments have been proposed and sent at different areas in current web. The viability of these systems relies upon the execution exchange offs and cost acquired in deployment.

DDoS recognition systems recognize the deviation of movement from typical conduct. This activity is named attack movement and afterward obstructed by proper resistance instrument. For exactness the recognition system should bring about low false positive and false negative rate.

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 [21], but also can facilitate the innovation and adoption of different routing architectures [22]. For instance, Godfrey et al. proposed pathlet routing [21], 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 [22]. Jokela et al. proposed in LIPSIN [23] to assign identifiers to links in a network and to encode the link identifiers along the path from a content provider to a content consumer into a zFilter (i.e., a PID), which is then encapsulated into the packet header and used by routers to forward packets. Luo et al. proposed an information-centric internet architecture called CoLoR [24] that also uses PIDs as inter-domain routing objects in order to enable the innovation and adoption of new routing architectures, as in [22].

There are two different use cases of PIDs in the afore-mentioned approaches. In the first case, the PIDs are globally advertised (as in pathlet routing [21] and [22]). 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 [23] and CoLoR [24]). In the latter case, the network adopts an information-centric approach [25] - [27] 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 (as in [23], [24]) 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 [28]. Moreover, as it

is shown in Sec. II-B, attackers can launch DDoS flooding attacks by learning PIDs if they are static.

To address this issue, in this paper, we present the design, implementation and evaluation of a dynamic PID (D-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 two fold.

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)? To address this challenge, D-PID lets neigh-boring domains negotiate the PIDs for their inter-domain paths based on their local policies (Sec. III-B). In particular, two neighboring domains negotiate a PID-prefix (as an IP-prefix) 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 neighboring 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 commu-nications when the PIDs change. To address this challenge, D-PID lets every domain distribute its PIDs to the routers in the domain (Sec. III-C). 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 re-sends 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 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.

II. LITERATURE SURVEY

In this segment, we survey the existing literature on Distributed Denial of Service attacks.

S. Yu, et al. [1], proposed a dynamic resource allocation method for securing singular clients of cloud amid DDoS attack guaranteeing quality of service during attack. The cloud condition is fit for controlling the resource allotment since it has vast number of resources to dispense to individual client. The resource allocation system utilized as a part of mists assumes key part in relieving the effect of attack by offering access to resources. In cloud condition the accomplishment of attack or defends relies on who is holding more resources, attacker or cloud client. The dynamic additional resource allocation counteracts starvation, along these lines protecting against DDoS attack. They additionally exhibited line based model of resource portion under different attack situations.

V. A. Foroushani, et al. [2], proposed protection against DDoS attacks containing attack packets with spoofed IP addresses called Trace back based safe defense against DDoS loading attacks. The component is executed shut to attack source, rate-constraining measure of movement sent towards casualty. The execution assessment of the system utilizing true CAIDA DDoS attack datasets showed increment in throughput of real activity forcing less overhead on participating routers.

B. Liu, et al. [3], proposed shared departure filtering for giving insurance against IP spoofing based flooding attacks. They have utilized genuine web dataset for acquiring reenactment comes about. The instrument utilizes the entrance control rundown of autonomous (AS) that contains rundown of tenets for applying entrance/departure separating and unicast reserve path forwarding. This strategy ensures the frameworks which send the component while keeping non-deployers from openly utilizing it.

In [4], A. Compagno, et al. introduced barrier against interest flooding conveyed dissent of administration attacks in Named Data organizing. Interest flooding requires restricted resources to dispatch attack. Pending interest table is kept up at switches for maintaining a strategic distance from copy interests. Poseidon structure is presented for identification and relief of interest flooding attacks. The assessment of the system over system reenactment condition utilizing NS3 demonstrated that it is conceivable to use up to 80% accessible data transfer capacity amid attack utilizing this framework.

C. Chung, et al. [5], proposed distributed intrusion recognition and countermeasure choice component in cloud frameworks. The NICE framework utilizes interruption recognition conspire at each cloud server for distinguishing and dissecting approaching traffic. The strategy works for virtual cloud framework and makes situation attack diagram for ascertaining helplessness to communitarian attacks.

The defenseless frameworks are the exchanged to review state where profound bundle assessment is utilized to stamp potential attack practices.

In [6], S. Rastegari, et al. displayed a quantitative structure for understanding DDoS attack systems and gave defense answers for these attacks. The collaboration amongst aggressor and safe defense is exhibited utilizing Red group Blue group practice where Red group speaks to adversaries and Blue group recognizes conceivable vulnerabilities endeavoring to shield them. The framework was tried utilizing OMNeT++ arrange test system. The reproduction comes about show that one defense technique isn't generally an ideal arrangement; rather it ought to powerfully adjust and enhance as indicated by changing attack strategies.

In [7], L. Jingna has portrayed different Denial of Service attack standards, strategies for recognizing the DoS and DDoS attacks, and safe defense instruments against DDoS attacks. Different attack propelling techniques, for example, SYN Flood, IP mocking DoS attack, UDP flood attack, the PING flood attack, Teardrop attack, Land attack, Smurf attack, Fraggle attacks, and so on are clarified. Discovery techniquesfor above attacks are recorded with their organization area. Certain methodologies are recommended for improving barrier techniques.

S. Yu, et al. [8], proposed a strategy for recognizing flash crowds from DDoS attacks in view of stream connection coefficient. The attackers utilize the movement design fundamentally the same as blaze swarm which cripples the recognition of attack. This poses a test for the individuals who endeavor to safe defense the DDoS attacks. By distinguishing genuine DDoS attack utilizing this technique applies fitting resistance component to protect against DDoS attacks. They made overlay organize on switches that was under their control. The approaching stream was observed and number of packets in each stream was recorded. This recorded data helps in isolating glimmer swarm from real movement. They assessed the created component utilizing 1998 FIFA World CUP genuine informational indexes of blaze group and genuine attack tools, Mstream.

B. S. K. Devi, et al. [9] proposed Interface Based Rate Limiting (IBRL) algorithm for moderating recognized DDoS attacks in the system. It ensures that enough data transmission is accessible for honest to goodness activity amid attack. System checking framework sent in exploratory testbed gather the movement follows in organizes. This movement is investigated for measuring its effect amid attack utilizing host and system based measurements, for example, packet loss, latency, connect use and throughput took after by rate-restricting on the attack movement in order to permit genuine clients. Trial comes about show increment in throughput of honest to goodness traffic.

In [10], A. Mishra, et al. nearly portrayed different defense systems, diverse attacking instruments and preferences impediments of these strategies. The procedures for interruption location and moderation are grouped on the premise of blame resistance and nature of administrations gave.

In [11], Z. Chao-yang, et al. given a definite investigation of existing refusal of administration attack counteractive action standards. Four sorts of barrier procedures are clarified. In the first place strategy is protecting utilizing switch utilizing reverse way sending. Second strategy includes utilizing TCP catch for TCP obstructing for constraining SYN attack. Third technique is creating trusted stage in which a chain of trust and validation is shaped in view of confided in root. Fourth strategy utilizes confirmation framework for giving validation.

In [12], J. Mirkovic, et al. displayed examination between resistance instruments that channel parodied attack movement in light of some execution measurements. The accessible resistances are either conveyed at end organize or require joint effort of center switch for sifting or parcel checking. Every resistance is assessed in its controlled condition; henceforth, they played out a near examination to discover the execution of every component as a rule organize setting with no topology changes.

In [13], X. Bi, et al. proposed an idea to fabricate and ensure security declaration framework for avoiding DDoS attacks. This strategy depends on a Service Oriented Architecture (SOA). Servers and different supplies shape overlay arrange concealing the genuine area of server. The overlay network has two arrangement of nodes, steering nodes that allots distinctive transfer speed to various streams and serving nodes. Customer needs to first get to direct declaration toward access the server, however it requires parcel of CPU time. As cost of propelling effective attack is high it is valuable in anticipating attacks.

M. S. Fallah [14], proposed amusement theoretic approach for controlling customer baffle based resource utilization. Four protection strategies were created, two for single source attack and two for appropriated attacks. In customer bewilder based technique for safe defending the flooding attacks, asked for resources of the server are dispensed if the customer gives amend answer for the baffle sent by server. Confuse explaining expends the resources of assailant; henceforth, the aggressor is debilitated from making attack more than once. The amusement hypothesis approach keeps up ideal level of riddles in order to serve effectively to honest to goodness customers.

B. Krishna Kumar, et al. [15] proposed a bounce tally based parcel handling approach for recognizing aggressors utilizing mock source IP address. In this technique the bundles from the frameworks at a similar jump tally going through a similar switch are set apart with a similar recognizable proof number which is the mix of 32 bits IP address of the switch way and the scrambled estimation of the bounce check. This esteem is coordinated with as of now put away an incentive at getting switch. In this way, attack packets are recognized early and caricaturing dangers are diminished.

J. Atoum, et al. [16], introduced two methodologies of defense systems for upgrading the productivity of resistance against DDoS. The principal procedure called Distributed discovery/parcel Reflector utilizes bundle reflecting system and the second Graveyard methodology drops malignant packets in the wake of playing out a few levels of testing on them. This strategy joins information mining, learning sharing and is sent at numerous areas in the system.

III. 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] - [9]. Similarly, IP traceback-based methods trace attacks back through the network toward the attacking sources [10] - [14]. In addition, the approaches proposed in [19] - [20] 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

[4] 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 a node 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 [15], 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 spirt, 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 IP-prefixes of end hosts change frequently. On the other hand, the PIDs are kept secret and change dynamically in D-PID. While this incurs cost since destinations need to re-send GET messages, the results presented in Sec. V show that the cost is fairly small.

The capability-based designs [16] - [17] also share the same spirt with "off by default" and D-PID. In these approaches, a sender first obtains the permission from the destination in order to send data packets to it. The destination provides the capabil-ities to the sender if it wants to receive packets from the sender. The sender then embeds the obtained capabilities into packets. Routers along the path from the sender to the destination verify the capabilities in order to check whether or not the destination wants to receive the packets. If not, the routers simply discard the packets. D-PID differentiates from the capability-based approaches in two aspects. On one hand, communications are initiated by receivers in D-PID but by senders in capability-based approaches. On the other hand, as pointed out in [43], 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. By contrast, D-PID effectively mitigates such attacks because of three reasons. First, the GET messages carry the PIDs along the paths from the compromised computers to the victim. Second, the PIDs are negotiated by neighboring domains that can verify the authenticity of PIDs when they forward GET messages. These two reasons makes it convenient to trace back the attackers. Third, the ubiquitous in-network caching in CoLoR reduces the GET messages sent to the target victim.

Named data networking (NDN) [25] is another approach closely related to our work. In NDN, a content consumer sends out an Interest packet when it wants a piece of content. The Interest is routed (by the content name) to the content provider by routers in the Internet. When a router forwards the Interest toward the content provider, it inserts an entry into its pending Interest table (PIT) that stores the content name and the incoming interface of the Interest packet. When the content provider receives the Interest packet, it sends the corresponding Data packet back to the subscriber. The routers then forward the Data packet back to the content consumer according to the PIT entries stored by them. Unfortunately, maintaining a PIT table at routers makes NDN vulnerable to Interest flooding attacks [44]. By contrast, routers in D-PID do not maintain any forwarding state. In addition, as stated in the previous paragraph, carrying PIDs along the path from attackers to the victim makes it convenient to trace back the attackers, thus help preventing them from launching attacks by sending plenty of GET messages.

IV. EXISTING SYSTEM

Many approaches have been proposed in order to prevent DDoS flooding attacks, including network ingress filtering, IP traceback, capability-based designs, and shut-up messages. Godfrey et al. proposed pathlet routing, inwhich networks advertise the PIDs of pathlets throughout theInternet and a sender in the network constructs its selectedpathlets into an end-to-end source route. Koponen et al.further argued in their insightful architectural paper that usingpathlets for inter-domain routing can allow networks to deploydifferent routing architectures, thus encouraging the innovationand adoption of novel routing architectures. Jokela etal. proposed in LIPSIN to assign identifiers to links ina network and to encode the link identifiers along the pathfrom a content provider to a content consumer into a zFilter(i.e., a PID), which is then encapsulated into the packetheader and used by routers to forward packets. Luo et al.proposed an information-centric internet architecture calledCoLoRthat also uses PIDs as inter-domain routingobjects in order to enable the innovation and adoption of newrouting architectures.

V. PROPOSE SYSTEM

In this paper, we present the design,implementation and evaluation of a dynamic PID (D-PID)mechanism. In D-PID, two adjacent domains periodicallyupdate the PIDs between them and install the new PIDsinto the data plane for packet forwarding. Even if the attackerobtains the PIDs to its target and sends the malicious packetssuccessfully, these PIDs will become invalid after a certainperiod and the subsequent attacking packets will be discardedby the network. Moreover, if the attacker tries to obtain thenew PIDs and keep a DDoS flooding attack going, it not onlysignificantly increases the attacking cost, but alsomakes it easy to detect the attacker. In particular,our main contributions are two fold. On one hand, we propose the D-PID design by addressingthe following challenges. First, how and how often shouldPIDs change while respecting local policies of autonomoussystems (ASes)? Second, since inter-domain packet forwarding is based onPIDs that change dynamically, it is necessary to maintainlegitimate communications while preventing illegal communicationswhen the PIDs change. To address this challenge,D-PID lets every domain distribute its PIDs to the routersin 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.

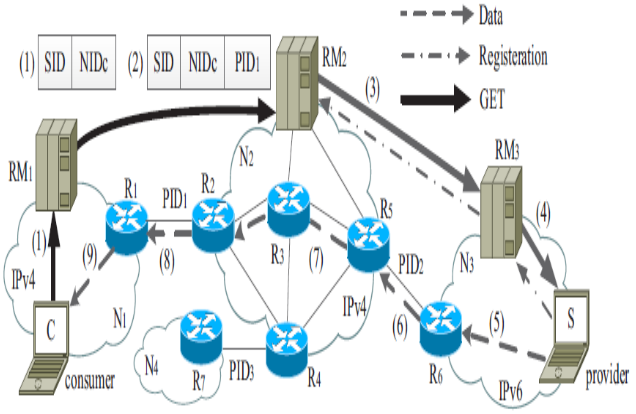

Figure 1

Brief Introduction to CoLoR

CoLoR is a receiver-driven information centric network architecture that assigns unique and persistent content names (or service identifiers, SIDs) to content chunks. As in [20] and [27], CoLoR assigns intrinsic secure self-certifying node identifiers (NIDs) to network nodes and ASes so that authenti-cating a node/AS does not require an external authority such as ICANN, thus improving security and privacy. In addition, two neighboring domains negotiate a PID for every inter-domain path 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 inter-domain routing table, which records the PID of each inter-domain path and the border router that the PID originates, as illustrated at the upper right corner in Fig. 1. For instance, the border router in domain N_2 connecting

PID_2 in Fig. 1 is R_5 . 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 log-ically centralized (but may be physically distributed) resource manager (RM) used to propagate the reachability information of SIDs. Particularly, when a content provider wants to provide a content chunk to consumers, he registers the SID of the content chunk to its local RM . The local RM then registers the SID to its providers or peers, by using an approach similar to the one used in [26].

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 neighboring 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.

Figure 1 illustrates the basic content retrieval process in CoL-oR, assuming that the content provider holds a piece of desired content with name SID and the content consumer wants to obtain the content. Firstly, the content provider registers the SID to its local RM (i.e., RM_3), which then registers the SID to its provider RM (i.e., RM_2), as illustrated by the dashed lines in Fig. 1. When the content consumer wants to obtain the content, it sends a GET message to its local RM (i.e., RM_1), as illustrated by (1) in Fig. 1. Since no local host can provide the desired content, RM_1 appends PID_1 at the end of the GET message and forwards the GET message to its neighbor RM_2 , as illustrated by (2) in Fig. 1. Similarly, RM_2 appends PID_2 at the end of the GET message and forwards the GET message to RM_3 , as illustrated by (3) in Fig. 1. RM_3 then forwards the GET message to the content provider who holds the desired content, as illustrated by (4) in Figure 1.

The ingress border router encapsulates them an outer header corresponding to the routing protocol used by the domain. By contrast, when a data packet leaves a domain, the egress border router removes the outer header. For instance, when router R_5 receives a data packet carrying PID_1 , it encapsulates the data packet with an IPv4 header (as illustrated by (7) in Fig. 1), if domain N_2 uses IPv4 for intra-domain routing. When router R_2 receives the data packet, it removes the outer IPv4 header, as illustrated by (8) in Fig. 1. Note that the outermost PID is popped out by the ingress border router of each domain in order to prevent content consumers from knowing the PIDs toward a content provider and launching DDoS attacks. For example, when the border router R_5 receives the data packet, it removes PID_2 since it is the outermost PID, as can be seen by comparing (6) and (7) in Figure 1.

CoLoR offers several interesting features. First, as an information-centric network architecture, routers in the net-work can locally cache the popular contents so as to serve nearby users, thus reducing redundant transmission and con-tent retrieval delay. To achieve this, the router caching a piece of content simply needs to register the SID of the content to its local RM . Second, it is easy to accurately, timely estimate the traffic matrices of a network since an ingress border router of a domain can know the egress border router of a packet by looking up the inter-domain routing table [29]. Third, CoLoR makes it easy to efficiently integrate information-centric networking and software-defined networking [30]. In addition, the data plane in CoLoR is scalable because the sizes of inter-domain routing tables maintained by border routers depend on the number of neighbors of a domain, which is limited to be several thousands in the Internet today. While RMs needs to deal with SIDs whose number is quite large, the RM in a large domain can be realized by using a distributed system (e.g., a data center) and the content names could be aggregated by using appropriate name formats such as P : L

Furthermore, to upgrade from the current Internet to CoLoR, the intra-domain routers of domains do not need to be upgraded, thus reducing deployment cost. Finally, CoLoR offers some security benefits [31] while avoiding Interest flooding attacks suffered by NDN [44] because 1) both routers and RMs in CoLoR do not maintain pending Interest tables; and 2) the PIDs carried in GET messages can be used to trace back attackers.

CoLoR also has some drawbacks that need to be addressed before its real deployment in the future. First, carrying the NID of the content consumer and the desired SID in packet headers reveals user privacy. Second, border routers need to encapsulate/decapsulate outer packet headers (e.g., IPv4 headers), which makes it challenging to realize line-speed packet forwarding. Third, as it is shown below, attackers can learn PIDs in the network and launch DDoS attacks in the data plane, if PIDs are static. As an attempt to address these drawbacks, in this paper we propose D-PID to prevent DDoS attacks in the data plane.

Why Dynamically Changing PIDs

In this subsection, we explain why it is necessary to dynam-ically 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.

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. We call such a process as the PID learning stage in the rest of this paper. Figure 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, AS J 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 Figure 2. The attacker then learns the corresponding PIDs in the network, which are represented by solid lines in Figure 2.

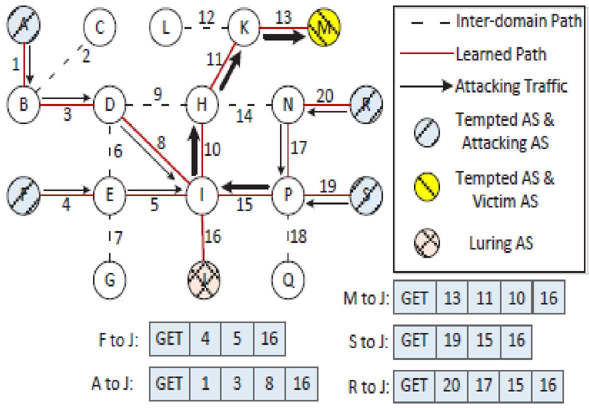

Fig. 2. Illustration for the GET luring.

As mentioned before, the communications that last long enough (e.g., longer than $2T_{PID}$) may be interrupted when D-PID is applied in CoLoR. As a result, we need to design a mechanism to timely update the PIDs that are used by the active communication between a content provider and a content consumer. To achieve this, we let the subscriber periodically retransmit the same GET message for an active communication. When the source receives the PID sequence contained by the retransmitted GET message, it updates the old PID sequence to the fresh one and uses the updated PID sequence to send subsequent data packets. Accordingly, the data packets from the source can be correctly forwarded to the subscriber and the ongoing communication will not be interrupted. In the rest of this paper, we denote the GET retransmission period as T_{GET} .

We use the example in Fig. 4 to further illustrate such a process. Assuming that, before time t_1 , the subscriber in domain C sends a GET message to the source in domain

F to request a content. Thus, the source initially sends the corresponding data packets back to the subscriber based on the old PID 0x1A01010A. After t_1 , the PID is updated to 0x1AF12C03, while the data plane can still forward data packets based on the old PID 0x1A01010A. Accordingly, the communication will not be interrupted before $t_1 +T_1$, but will be interrupted after $t_1 + T_1$. However, if the subscriber timely retransmits the same GET message to the source (an enough time period before $t_1 + T_1$), the new PID 0x1AF12C03 will be encapsulated in the GET message and sent to the source.

This way, the source knows the new PID 0x1AF12C03 and uses it to forward the subsequent data packets to the subscriber. Therefore, if the subscriber periodically retransmits the GET message and the period is appropriately set, the source can know the valid PIDs toward the subscriber, thus maintaining a legal communication.

It is worthy of noting that the retransmission of a GET message is initiated by a content consumer based on its GET retransmission period T_{GET} and the initial time the consumer sends out the first GET message for a piece of content. Therefore, when the PID of an inter-domain path from the content provider to the consumer changes, the content consumer does not resend a GET message immediately when the itP ID changes. This way, we can efficiently avoid the sudden increase in the number of GET messages received by RMs in the Internet.

Setting T_{PID} and T_{GET}

In D-PID, T_{PID} and T_{GET} should be carefully set so that a legal communication will not be interrupted when PIDs change dynamically. To achieve this, we build a mathematical model to calculate the appropriate values of T_{PID} and T_{GET} , with the help of Figure 5.

Without loss of generality, we assume that there are N inter-domain paths along the path from a server to a client. In particular, we consider the i − th (1 ≤ i ≤ N) inter-domain path and assume that it connects two domains A and B. Specifically, we call the timeout period in which a GET message arrives at domain A as the present timeout period and the GET message will be forwarded to domain B. We also assume that the present timeout period begins at time zero, as illustrated by Fig. 5. In addition, we assume that the GET message arrives at domain A at time t^i_0 and the round-trip time from domain A to the content provider is $RT^i T$. For ease of presentation, we denote the timeout period of the i–th path be $T_P{}^i{}_{ID}$.

With these assumptions, one can observe from Fig. 5(a) that, if (t^i_0 + $RT^i T$) is less than $T_P{}^i{}_{ID}$, the corresponding data packets for the GET message will arrive at domain B in the same timeout period in which the GET message arrives at domain A. In this case, the data packets can be correctly forwarded to domain A. Similarly, if (t^i_0 + $RT^i T$) is less than $2T_P{}^i{}_{ID}$, the corresponding data packets will arrive at domain B in the next timeout period, as shown by Fig. 5(b). In this case, the data packets also can be correctly forwarded to domain A because domain B is able to forward data packets based on the PIDs chosen for the present and the previous timeout periods.

However, when (t^{i}_{0} + RT^{i} T) is larger than 2T$_{\text{P}}^{i}$ ID, as shown in Figure 5(c), data packets will be dropped by domain B. There- fore, in order to guarantee correct data packet forwarding, it must hold that:

$$t^{i}_{0} + \text{RT}^{i}\ \text{T} < 2\text{T}_{\text{P}}^{i}\ \text{ID}:$$

Note that t^{i}_{0} should be evenly distributed in [0; T$_{\text{P}}^{i}$ ID). To guarantee correct packet forwarding, it requires that:

$$\text{T}_{\text{P}}^{i}\ \text{ID} > \sup(\ \text{RT}^{i}\ \text{T}\ );$$

where sup(x) represents the supremum of x. This formula indicates that when we set T$_{\text{P ID}}$ for an inter-domain path, the value should be greater than the supremum value of the network's round-trip time. From Figure 13 in [35], we know that with probability higher than 99.9%, the round-trip time is less than 1.5 seconds. Therefore, we can treat the value of sup(RT^{i} T) as 2 seconds in practice.

We now describe how to set the value of the GET re-transmission period T$_{\text{GET}}$ for an active session. Obviously, the second GET message (i.e., the first retransmitted GET message) arrives at domain A at the time (t^{i}_{0} + T$_{\text{GET}}$). When the data source receives the second GET message, it will renew the PID sequence and then sends subsequent data packets to the client by using the new PID sequence. We assume that the first one of these subsequent packets arrives at domain B at time (t^{i}_{0} + T$_{\text{GET}}$ + RT^{i} T).

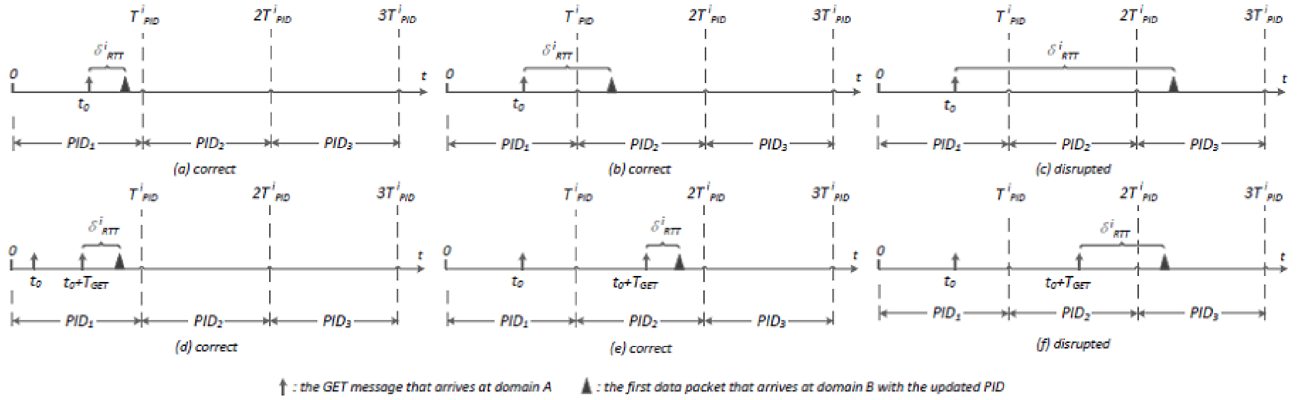

Figure 3. The mathematical model for determining T$_{\text{PID}}$ and T$_{\text{GET}}$

Similar to our discussions on setting T$_{\text{P}}^{i}$ ID, there are also three cases. In the first case, t^{i}_{0} and (t^{i}_{0} + T$_{\text{GET}}$ + RT^{i} T) are in the same timeout period, In this case, when domain B receives the data packets sent by the server, it can correctly forward these data packets to domain A. In the second case, (t^{i}_{0} + T$_{\text{GET}}$ + RT^{i} T) is in the next timeout period to t^{i}_{0}, as illustrated by Fig. 3(e). In this case, domain B also can correctly forward data packets to domain A because now domain B can forward packets based on both

PID$_1$ and PID$_2$. In the third case, (t^{i}_{0} + T$_{\text{GET}}$ + RT^{i} T) is larger than 2T$_{\text{P}}^{i}$ ID, as illustrated by Fig. 5(f). In this case, some data packets will be discarded during the period (2T$_{\text{P}}^{i}$ ID, t^{i}_{0} + T$_{\text{GET}}$ + RT^{i} T). Therefore, to guarantee the correct data forwarding, it must hold that:

$$t^{i}_{0} + \text{T}_{\text{GET}} + \text{RT}^{i}\ \text{T} < 2\text{T}_{\text{P}}^{i}\ \text{ID}:$$

As discussed before, t^{i}_{0} may be very close to T$_{\text{P}}^{i}$ ID, so the above inequation can be rewritten as:

$$\text{T}_{\text{GET}} < \text{T}_{\text{P}}^{i}\ \text{ID} - \sup(\ \text{RT}^{i}\ \text{T}\ ):$$

Note that the above discussions are focused on a given inter-domain path. Since there are N inter-domain paths between the client and the server, to ensure that the data packets could be correctly forwarded along all intermediate paths, the T_{GET} should be given by:

$$T_{GET} < \min_{i=1;2;:::;N} (T^{i}_{PID} - \sup(RT^{i}T)):$$

Without loss of generality, we assume that $\sup(RT^{i}T)$ is the same for all inter-domain paths. In addition, as discussed before, we can set $\sup(RT^{i}T)$ to be 2 seconds in practice. Accordingly, the above inequation can be rewritten as:

$$T_{GET} < \min_{i=1;2;:::;N} (T^{i}_{PID}) - 2: \qquad (1)$$

Collecting Minimum T_{PID}^{i}

From Equation (1), we know that the subscriber needs to be aware of $\min_{i=1;2;\cdots;N} (T^{i}_{PID})$ in order to appropriately set T_{GET} . To achieve this, we add a field called MINI-MUM PERIOD in the GET message and the data packet to collect the PID update period T_{PID}^{i} of PIDs along the path from the subscriber to the source. When the subscriber sends out a GET message, the value of MINIMUM PERIOD is set to be infinite. When a RM receives a GET message and chooses

VI. CONCLUSION

In this paper, we've got presented the design, implementa-tion and analysis of D-PID, a framework that dynamically changes path identifiers (PIDs) of inter-domain methods so as to stop DDoS flooding attacks, once PIDs area unit used as inter-domain routing objects. we've got represented the look details of D-PID and enforced it in a very 42-node paradigm to verify its practicableness and effectiveness. we've got bestowed numerical results from running experiments on the paradigm. The results show that the time spent in negotiating and distributing PIDs area unit quite little (in the order of ms) and D-PID is effective in preventing DDoS attacks. we've got conjointly conducted in depth simulations to judge the value in launching DDoS attacks in D-PID and also the overheads caused by D-PID. The results show that D-PID considerably will increase the value in launching DDoS attacks whereas incurs very little overheads, since the additional variety of GET messages is trivial (only one.4% or 2.2%) once the retransmission amount is three hundred seconds, and also the inflammatory disease update rate is considerably but the update rate of informatics prefixes within the current web.To the simplest of our data, this work is that the commencement toward victimization dynamic PIDs to defend against DDoS flooding attacks. we have a tendency to hope it'll stimulate a lot of researches during this space.

VII. REFERENCES

[1]. S. Yu, Y. Tian, S. Guo, D. Wu, "Can We Beat DDoS Attacks in Clouds", IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2245-2254, Sept. 2014.

[2]. V. A. Foroushani, A. N. Zincir-Heywood, "TDFA: Trace back based Defense against DDoS Flooding Attacks", IEEE 28th International Conference on Advanced Information Networking and Applications, pp. 597-604, May 2014.

[3]. B. Liu, J. Bi, A. V. Vasilakos, "Toward Incentivizing Anti Spoofing Deployment", IEEE Transactions on Information Forensics and Security, vol. 9, no. 3, pp. 436-450, March 2014.

[4]. A. Compagno, M. Conti, P. Gasti, G. Tsudik, "Poseidon:Mitigating Interest Flooding DDoS Attacks in Named Data Networking", IEEE 38th Conference on Local Computer Networks, pp. 630-638, Oct. 2013.

[5]. C. Chung, P. Khatkar, T. Xing, J. Lee, D. Huang, "NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems", IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 4, pp. 198-211, July/Aug. 2013.

[6]. S. Rastegari, P. Hingston, C. Lam, M. Brand, "Testing A Distributed Denial of Service Defense Mechanism Using Red Teaming", IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), pp. 23-29, April 2013.

[7]. L. Jingna, "An Analysis on DOS Attack and Defense Technology", IEEE 7th International Conference on Computer Science & Education (ICCSE), pp. 1102-1105, July 2012.

[8]. S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, F. Tang, "Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient", IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 6, pp. 1073-1080, June 2012.

[9]. B. S. K. Devi, G. Preetha, S. M. Shalinie, "DDoS Detection using Host-Network based Metrics and Mitigation in Experimental Testbed", IEEE International Conference on Recent Trends In Information Technology (ICRTIT), pp. 423-427, April 2012.

[10]. A. Mishra, B. B. Gupta, R. C. Joshi, "A Comparative study of Distributed Denial of Service Attacks, Intrusion Tolerance and mitigation Techniques", European Intelligence and Security Informatics Conference (EISIC), pp. 286-289, Sept. 2011.

[11]. Z. Chao-yang, "DOS attack analysis and study of new measures to prevent", IEEE International Conference on Intelligence Science and Information Engineering, pp. 426-429, Aug. 2011.

[12]. J. Mirkovic, E. Kissel, "Comparative Evaluation of Spoofing Defenses", IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 2, pp. 218-232, March-April 2011.

[13]. X. Bi, Q. Zheng, "Study on Network Safety Strategy against DDoS Attack", IEEE International Conference on Advanced Management Science (ICAMS), pp. 623-627, July 2010.

[14]. M. S. Fallah, "A Puzzle-Based Defense Strategy Against Flooding Attacks Using Game Theory", IEEE Transactions on Dependable and Secure Computing, vol. 7, no. 1, pp. 5-19, Jan.-March 2010.

[15]. B. Krishna Kumar, P. K. Kumar, R. Sukanesh, "Hop Count Based Packet Processing Approach to Counter DDoS Attacks", IEEE International Conference on Recent Trends in Information, Telecommunication and Computing, pp. 271-273, March 2010.

[16]. J. Atoum, O. Faisal, "Distributed Black Box and Graveyards Defense Strategies against Distributed Denial of Services", 2nd International Conference on Computer Engineering and Applications, pp. 87-91, March 2010.

[17]. Z. Xiao-hui, P. Xuan-ge, L. Man-hua, X. Hong-qi, J. Shi-yao, "Research on An Effective Approach against DDoS Attacks", IEEE International Conference on Research Challenges in Computer Science, pp. 21-23, Dec. 2009.

[18]. G. Jin, F. Zhang, Y. Li, H. Zhang, J. Qian, "A Hash-based Path Identification Scheme for DDoS Attacks Defense", IEEE 9th International Conference on Computer and Information Technology, pp. 219-224, Oct. 2009.

[19]. J. Mirkovic, A. Hussain, S. Fahmy, P. Reiher, R. K. Thomas, "Accurately Measuring Denial of Service in Simulation and Testbed Experiments", IEEE Transactions on Dependable and Secure Computing, vol. 6, no. 2, pp. 81-95, April-June 2009.

[20]. R. Kumar, R. Karanam, R. C. Bobba, Raghunath S., "DDoS Defense Mechanism", IEEE International Conference on Future Networks, pp.254-257, March 2009.

[21]. Y. Xie, Shun-Zheng Yu, "A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviours", IEEE/ACM Transactions on Networking, vol. 17, no. 1, pp. 54-65, Feb. 2009.

[22]. X. Wang, "Mitigation of DDoS Attacks through Pushback and Resource Regulation", IEEE International Conference on Multimedia and Information Technology, pp. 225-228, Dec. 2008.

[23]. P. Jayashree, K. S. Easwarakumar, Anandharaman V., Aswin K., Raja Vijay S, "A Proactive Statistical Defense Solution for DDOS Attacks in Active Networks", IEEE 1st International Conference on Emerging Trends in Engineering and Technology, pp. 878- 881, July 2008.

[24]. M. Muthuprasanna, G. Manimaran, "Distributed divide-and conquer techniques for effective DDoS attack defenses", IEEE 28th International Conference on Distributed Computing Systems, pp. 93-102, June 2008.

[25]. S. Malliga, A. Tamilarasi, "A defensive mechanism to defend against DoS/DDoS attacks by IP trace back with DPM", IEEE International Conference on Computational Intelligence and Multimedia Applications, pp. 115-119, Dec. 2007.

[26]. C. Chae, S-H. Lee, J-S. Lee, J-K. Lee, "A Study of Defense DDoS Attacks using IP Trace back", IEEE International Conference on Intelligent Pervasive Computing, pp. 402-408, Oct. 2007.

AUTHOR DETAILS

Y.SRINIVASA RAO Is Working An Assistant Professor In Vignan's Lara Institute Of Technology & Science..Vadlamudi-522213 Guntur Dist.He Has Experience In The Teaching Field For 6 Years And His Interested In Research Areas Networking Security And Subject Expect In ,C, And Java

CHANDU TIRUPATHAMMA she is Currently pursuing MCA in MCA Department,Vignan's Lara Institute Of Technology & Science, Vadlamudi, Guntur(Dt), Andhra Pradesh, India. she received his Bachelor of science from ANU