Distributed Defence of Service (DiDoS): A Network-layer Reputation-based DDoS Mitigation Architecture

Andikan Otung[a] and Andrew Martin[b]
Computer Science Department, University of Oxford, Parks Road, Oxford, U.K.

Keywords: DDoS, Defence, Mitigation, Security Architecture, Security Framework, Design Analysis, IoT, Reputation, Internet, Accountability.

Abstract: The predominant strategy for DDoS mitigation involves resource enlargement so that victim services can handle larger demands, however, with growing attack strengths, this approach alone is unsustainable. This paper proposes DiDoS (Distributed Defence of Service), a collaborative DDoS defence architecture that leverages victim feedback to build network-level sender reputations that are applied to identify and thwart attack traffic – thus alleviating the need for resource enlargement. Since attack traffic is dropped at points of contention in the Internet, (rather than rote blocking at source) DiDoS reduces the impact of false positives and enables the traversal of legitimate traffic from said devices across the Internet. Through anti-spoofing protection and preferential treatment of DiDoS-compliant devices, DiDoS offers adoption incentives that help offset the Tragedy of the Commons effect of DDoS mitigation, which commonly sees non-victim intermediary entities benefit little from DDoS defence expenditure. In this paper, the tenets and fundamentals of the architecture are described, before being analysed against the presented threat model. Simulation results, demonstrating the effectiveness of the reputation convergence of the scheme, in the use-case of a local access network, are also presented and discussed.

1 INTRODUCTION

Denial of service (DoS) attacks are attempts by attackers to prevent legitimate users from accessing connected services through the disconnection, corruption or malicious consumption of the resources upon which the victim service depends. Distributed denial of service attacks (DDoS) are a form of DoS attacks that are executed by multiple distributed agents. Since their first documentation more than two decades ago (Michael Aaron Dennis, 2012; Raghavan & Dawson, 2011), DDoS attacks have consistently grown, year-on-year, in magnitude and prevalence to become recognized by Internet service providers (ISPs) as the top operational threat to customers (Arbor Networks, 2015; Behal, Kumar, & Sachdeva, 2018; Cisco, 2018; Labovitz, 2010). This threat, however, is compounded by the phenomenon of the Internet of things (IoT), which has inadvertently contributed to growing botnet sizes and resulting attack strengths through an abundant supply of connected yet unsecured devices.

Flood attacks, such as the notorious 1.2 Tbps attack against DNS provider DYN (David Homes, 2016; Nicky Woolf, n.d.; Scott Hilton, 2016) in which DoS was achieved through the scale of data sent, are particularly challenging to deal with. This is because, even if the victim server is able to process requests at its Internet access line rate, the finite resources (routers) along the attack-path eventually reach their capacity to forward received data and, once that limit is exceeded, are no longer able to forward all incoming legitimate requests.

The state of the art in handling such attacks, involves enlarging the victim resources (server and router processing power and bandwidths) to increase the capacity of the victim to serve clients. However, the cost of doing so, combined with the rate of increasing attack strengths, has led researchers to conclude that bolstering victim resources as a means to defending against DDoS attacks is not sustainable (Osterweil, Stavrou, & Zhang, 2019; Zeijlemaker & Rouwette, 2017).

[a] https://orcid.org/0000-0001-7517-7159
[b] https://orcid.org/0000-0002-8236-980X

Otung, A. and Martin, A.
Distributed Defence of Service (DiDoS): A Network-layer Reputation-based DDoS Mitigation Architecture.
DOI: 10.5220/0009091206190630
In *Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP 2020)*, pages 619-630
ISBN: 978-989-758-399-5; ISSN: 2184-4356

If attack traffic were to be intercepted further away from its destination, the victim would be far less susceptible to DoS. However, earlier interception requires the ability of intermediary network devices (routers) to reliably identify and eliminate such traffic. This is a challenge since the traditional classification of traffic as malicious is difficult to achieve outside the context of victim-centric information (such as what constitutes "wanted" or "normal" to the victim) and especially if the malicious entity mimics legitimate traffic patterns.

Reputation-based defences address this by associating traffic legitimacy with the past behaviour of its sender. For the reputation-based defence to be effective, the basis on which reputations are built and destroyed must capture features that are intrinsic (and predominantly exclusive) to an attacker's observable behaviour.

This paper describes one such defence, DiDoS (Distributed Defence of Service – pronounced "Die-DOS"): an anti-spoofing DDoS defence architecture that applies collaboratively-maintained reputations to discriminate between malicious and benign traffic closer to its source.

Beginning with a brief review of notable pieces of related work (section 2), a threat model is then presented (section 3) before the DiDoS architecture is articulated (section 4). Various design considerations with respect to the threat model are discussed throughout section 4, with a more focused analysis presented in section 5. Simulation results demonstrating the viability of a reputation-based approach and the effectiveness of the reputation update algorithm under various DiDoS adoption percentages and attacker strategies are also presented. The paper concludes with a summary of key findings (section 6) and a brief discussion of possible future work in section 7.

2 RELATED WORK

Passport (Liu, Li, & Yang, 2008) is a DDoS defence that mitigates attacks that use spoofed source addresses. In Passport, source autonomous systems (ASs) append lightweight message authentication codes (MACs) to their sent packet that enable transit and recipient ASs to verify their origin as from the source AS. MAC's are checked at the borders between AS's and packets with invalid MACs are dropped. Packets are also dropped if they possess no MACs yet originate from an AS that participates in the scheme. AS's are able to verify MACs using symmetric shared secrets between them that are obtained during a setup phase that piggybanks over the BGP protocol and uses a Diffie Helman key exchange. Passport is able to eliminate DDoS attacks that attempt inter-AS source IP address spoofing, however it cannot eliminate attacks that leverage intra-AS source IP address spoofing, or those that do not apply source IP address spoofing at all.

The Forwarding Accountability for Internet Reputability (FAIR) architecture (Pappas, Reischuk, & Perrig, 2015), like Passport, leverages lightweight MACs, however, in FAIR, AS's embed the cryptographic markings to packets as they traverse from source to destination. This allows recipient AS's to later prove (via feedback reports) to sending AS's that they did send the packet. Once the proof is accepted by the AS that sent the packets, the recipient AS deprioritizes traffic from that sender. FAIR implements reputation-based accountability by leveraging an evil bit to penalize AS's that forward traffic from misbehaving customers.

Operating at AS granularity, FAIR is prone to false positives since legitimate packets originating from AS's deemed to have misbehaved are all deprioritized and marked with a suspicious-bit. Furthermore, FAIR uses policies that are shared beforehand between AS's as a metric of distinction to determine misbehaviour. This metric however does not closely correlate with denial of service attacks, attackers or the denial of service effect, since AS's forwarding malicious traffic may simultaneously comply with the pre-agreed committed information rate (CIR) and committed burst size (CBS) policies, and so would evade consequences under FAIR.

FR-WARD (Mergendahl, Sisodia, Li, & Cam, 2017) is a source network end defence based on D-WARD (Mirkovic & Reiher, 2005) that targets IoT environments. FR-WARD monitors the source network to identify the egress of suspicious traffic. Suspicious senders are challenged with TCP congestion control signals that, if ignored, will result in the throttling of traffic flow from the sender. In this way legitimate traffic from compromised nodes is still allowed to egress the network – albeit at a reduced rate. However, attacks where attacking agents comply with "normal" sender rates and respond to TCP congestion control signals are unstoppable for this defence.

A defence architecture that 1) provides granular packet-spoofing protection, 2) leverages a metric[1]

[1] See section 4 for details of how the frequency of an entity's involvement in attacks is leveraged.

that is closely aligned with attacker behaviour, and 3) incorporates verifiable victim feedback to hold malicious senders to account, forms the main contributions of this work.

3 THREAT MODEL AND ASSUMPTIONS

The threat model considers an attacker as an entity that has amassed a large botnet and leases out its use for financial gain. As a result, fundamental characteristics of said attacker involve attacking as frequently and severely as possible. However, underlying motivations for attack vary according to the attacker's clients and may include extortion, hacktivism or sabotage – to name a few.

Clients of the *commercial attacker* consist of any kind of organisation motivated by a plethora of reasons, however, attacking objectives are considered to be independent of individual motivations (such as revenge, entertainment or financially motivated sabotage) and include:

- Rendering a service inaccessible by flooding the victim server with requests
- Degrading the quality of service experienced by the victim server's legitimate clients through the same means

These attacker objectives are also independent of the defence mechanism employed.

Since DiDoS is reputation based, in order to anticipate emergent threats to a DiDoS adherent Internet infrastructure, the following reputation-based objectives – which apply to reputation systems, in general – also form part of the threat model:

- **Slander** – attackers falsely diminish the reputation of other entities
- **Self-promotion** – attackers falsely increase their own reputation
- **Evasion** – attackers escape reputation penalties for misbehaviour
- **Self-destruction** – an attacker works to excessively diminish the reputation associated with its own identifier in a system. This attack makes sense when the attacking agent shares an identifier with a victim entity.
- **Flattery** – attackers act to falsely inflate the reputation of other entities
- **Sabotage** – an attacker takes steps to prevent the reputation system from operating correctly; for example, by preventing nodes from receiving or disseminating reputation values.

However, it is assumed that the attack avenues introduced by a reputation-based defence, such as Sybil attacks, multiple identity attacks or spoofed identity attacks, are used to achieve the original general attacker objectives first listed and do not create new attacker objectives.

The attacker capabilities describe the actions an attacker can undertake in order to achieve its goals. The DiDoS threat model considers not just the actions an attacker can undertake on a compromised entity, but also the kind of entity under the attacker's control.

Each host under an attacker's control is assumed to have full control of the device to perform functions such as: monitoring, processing, data manipulation, data fabrication and device actuation.

The impact of these capabilities varies greatly depending on the kind of entity compromised. Of the three layers of compromise considered in the DiDoS threat model – which include: host, private network and autonomous system – the main focus is on compromised hosts, which captures the conventional threat of numerous compromised end devices that form a large botnet.

The DiDoS threat model encapsulates the actions an attacker can instigate in cyberspace to subvert DiDoS and achieve any of its listed objectives. This includes specific kinds of active attacks such as replay, spoofing, reflection, collusion and slander attacks. However, passive non-participation in the DiDoS architecture is also a threat.

Despite the possibility of compromised devices being able to effect changes in their environment through actuation, the potential denial of service attacks resulting from this lie outside the scope of this research. Other explicit exclusions include duplicate packets, Economic Denial of Service Attacks (EDoS) (Naresh Kumar et al., 2012), application layer attacks and attacks that exploit particular hardware, software or protocol vulnerabilities.

We assume that DDoS attacks are detected with low false-positives such that generated reports correspond closely to attacks.

4 SYSTEM DESCRIPTION

4.1 Overview

This section describes the components of DiDoS – including the roles, mechanisms and cryptographic schemes used to achieve its goal of DDoS attack mitigation.

DiDoS introduces reputations to the network layer under the principle that each entity is responsible for the data it sends and is held to account by the entities that forward its traffic. Hence routers and routing networks store and maintain reputation levels for each device or network directly connected to them. These reputation-levels are embedded into packets by the entities that forward them. Packets with higher reputation are prioritized in transit, whereas packets with lower reputations are deprioritized and dropped at points of network contention, such as that arising from DDoS attacks.

Reputations are maintained by a feedback process in which victims send samples of attack packet headers, via cryptographically verifiable reports, to the entities that forwarded them.

DiDoS can be conceptualized as consisting of three *functional layers* that work together to mitigate attacks, with each layer being dependent on that below it. These are:

- A **Hierarchical Identification Scheme** – defines the way in which *packet forwarding accounting* (PTA) is attained, including specification of the identifiers against which reputations from the second layer are bound.
- A **Collaborative Reputation-management System** – encompasses the method by which reputations are maintained,
- And a **Distributed Mitigation Mechanism** – stipulates the method by which malicious traffic is deprioritized and dropped in transit.

These three layers are outlined in subsequent sections. Each functional layer of DiDoS consists of networked entities playing different roles to fulfil said functions; and the actions, undertaken by said entities to fulfil said functions, varies according to the phase of operation. Roles and phases are briefly described in the next subsection (4.2), however the main purpose of section 4 is to articulate DiDoS with respect to its functional layers.

4.2 Hierarchical ID Scheme and Roles

Beginning with the *hierarchical identification scheme*, which is the foundation of the functional layers of DiDoS, this subsection shall describe its organisation and how it relates to DiDoS roles. The hierarchical identification scheme consists of three layers: the autonomous system (AS) layer – at the top, followed by the private network layer and then the host layer. These are illustrated in Figure 1.

The goal of the identification scheme is to persistently bind entities to identifiers, such that entity actions can be linked to their identifiers and previous actions.

DiDoS achieves this hierarchically through roles played within and between layers, which include: *reputor*, *reputee* and *reputation domain manager*.

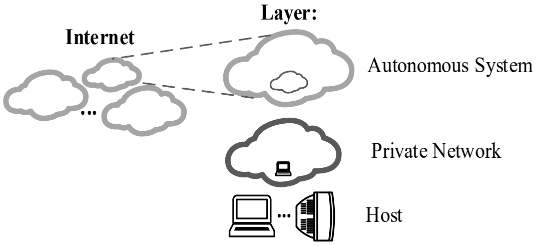

Figure 1: DiDoS hierarchical layer organisation.

The role of a reputor includes the functions required to manage the reputations of connected entities in the layer below said reputor. The reputee role, on the other hand, consists of the functions that facilitate the management of an entity’s own reputation by a reputor, such as the self-assessment and marking of the reputee’s own packets in order to facilitate the acquisition of a better reputation, which is described later in section 4.3. These reputor-reputee relationships are defined by the devices that forward traffic on behalf of other devices. For example, an access router providing Internet access to desktops in a private corporate network would be playing the reputor role, to its reputee desktops. Autonomous systems, that forward data on each other’s behalf, play the roles of both reputor and reputee and are considered *peers*. This is also the case for connected private networks that have been appropriately provisioned into separate reputation domains.

The last role in DiDoS is **reputation domain manager**, which can be a centralized server that manages the reputations of groups of entities within a particular administrative domain, facilitating functions such as the normalisation of reputations projected out by domain members, and the receipt and dissemination of attack-feedback reports.

The phases, mentioned in the previous section, include *setup*, *operational* and post-attack *feedback*. Setup occurs each time a reputor enlists a reputee and results in shared secrets between the two entities. It is assumed that reputees can be mutually authenticated to reputors with identification mechanisms of different strengths. These varying strengths of authentication are leveraged in the collaborative reputation management system to reflect the relative certainty of the persistent binding between all of a reputees actions (sent packets) and its identifier. As shall be shown, reputees with stronger authentication incur less reputational penalty when involved in an attack.

The characteristics that distinguish between the different levels of identification scheme strengths include: 1) the thing to which an identity is bound, 2) the presence of third party authentication and the cost of acquiring said 3rd party identity authentication (such as a registration or manual enrolment process – indicating a limited number of identities per entity), and 3) non-repudiation of an entities actions.

As an example, an entity with an identity that is 1) bound to *hardware* via a platform such as a Trusted Platform Module (TPM) (Martin & Others, 2008), whose 2) endorsement key is authenticated by a trusted certificate authority to which the TPM manufacturer registered with in advance, and 3) whose communications are protected by message authentication codes – constructed with secrets known only between that entity and its reputor, would be considered at the top level of identification scheme mechanisms.

4.3 Reputation Management System

At the centre of the DiDoS architecture is the collaborative reputation management system. This system consists of the methods and processes by which reputors manage the reputation of reputees and reputation domain managers (RDMs) coordinate the reputation management of entities within administrative domains.

Reputations are collaboratively maintained by a feedback process in which reports, containing a sample of the traffic (packet headers) to the victim at the time of attack – are generated by the victim and sent upwards for processing. In order to preserve the reputations of known benign users, such as paying customers, packets from said users are excluded from attack feedback reports. Reports are sent from the reputee to the reputor in the above layer, where, after an exchange between that reputor and its reputation domain manager, in which the reputation domain manager agrees to receive the report, the report is passed from the reputee to the reputation domain manager via the reputor. At the reputation domain manager, the report is verified by checking the message authentication codes (MACs) of the contained packets to confirm whether they were indeed forwarded by the reputor. Following that, packet provenance information – contained in each packet sent by a DiDoS-compliant source – is inspected and reputation scores (R_n) of reputees sending the packets contained in the report are adjusted according to 1) the strength of the reputee's identification I, 2) its volumetric contribution to the attack $\mathcal{V}$, and 3) the appropriateness $\mathcal{A}$ of the reputation score marked on the packet by the reputee. The equation combining these update variables is shown below:

$$R_n[x] = R_n[x-1] - \beta\left\{a_1 + a_2 \frac{\mathcal{V}}{I \cdot \mathcal{A}} + \right\} \tag{1}$$

β is the convergence factor that controls the rate at which R_n generalizes to a long-term value and x is a discrete time event variable representing the points at which R_n is updated. The coefficients a_1 and a_2 provide weighting to the components of the update amount and can be tailored according to the preferences of the RDM. The reputation penalty meted is proportional to the volumetric contribution of that reputee to the attack. However, the reputations of reputees with stronger identity authentication (as described in section 4.2) receive less penalty as shown. Insufficiently authenticated reputees share a single channel reputation, such that penalties incurred as a result of one of these grouped reputees are meted out on all of them.

The reputation scores lie within a range governed by: $R_{min} \leq R_n < R_{max}$; and, after calculation, the reputation scores are converted into reputation levels L_n by a process of quantisation: $L_n[x] = Q(R_n[x])$.

This allows for the flexibility of a granular range of reputation scores for grading reputees to be combined with a small set of reputation levels for faster in-transit packet processing.

The convergence factor β, could simply be set to equal a fixed base penalty amount P_B. However, since low adoption is a threat to the DiDoS architecture, it is essential, for the update algorithm to be resilient to low rates of DiDoS adoption, which would be reflected in lower numbers of attack feedback reports sent. In such a case (fewer sent feedback reports), the reputation penalty for an attack should automatically increase to compensate for unreported attacks. Thus:

$$\beta = a[x] \times P_B \tag{2}$$

Where,

$$a[x] = \frac{Z_A N((1-m) + km)}{Z[x]} \tag{3}$$

The numerator represents an estimate of the number of attack reports a reputor would expect to receive during full DiDoS adoption, which is calculated from: its estimate of the percentage of its own reputees that are malicious m, the number of its reputees N, an acceptable[2] attack rate Z_A, and a factor (k>1) representing how many more times a malicious reputee is expected to attack than a benign reputee. $Z[x]$ represents the actual number of attack reports received by the reputor over a given recent period.

Once new reputations are calculated and updated, the reputation domain manager creates new reports, each consisting of a subset of the packet headers from the original report, which are grouped according to the reputees[3] from which they originated. These new smaller reports are then disseminated to these reputees, which then proceed to process the reputations of their constituent reputees accordingly.

The observant reader may notice that the reputation update equation (Equation 1) exclusively permits negative adjustments to reputee reputations, about which the inquisitive reader may question the need for a process that allows the rebuilding of tarnished reputations or a mechanism that prevents the permanent flatlining of all reputations to a single constant bottom value. These issues are addressed in DiDoS through *passive periodic accumulation*.

In passive periodic accumulation, reputors increment the reputations of all their reputees equally by a small amount every fixed period of time (say daily). This allows reputee reputations to be strengthened over incident-free periods, which allows benign entities that have been circumstantially involved in an attack, and entities that have been purged of their malicious components, to recover their reputation.

In this way, reputees passively recover their reputation, by not participating in DDoS attacks. This mechanism, intentionally excludes the recovery of reputation through active means, such as sending benign data, because enabling legitimate entities to improve their reputation through specific actions would inherently also enable malicious entities to do the same whilst participating in attacks, and, hence, subvert the system. Granted, an attacker may elect to improve the reputations of its botnet entities by simply waiting, however, this strategy bears an economic cost to the attacker that is proportional to the waiting period.

4.4 Distributed Mitigation Mechanism

The DiDoS distributed mitigation mechanism describes the way in which reputations, from the previous DiDoS layer, are leveraged in order to alleviate the damage of DDoS flood attacks.

Senders (including routers and hosts) mark sent packets with the reputation level of their reputees. In the case of a router forwarding a packet, the reputee is the host that sent the packet, and in the case of a host, the reputee is the software originating the packet. The process of marking and checking packets does not occur on every device in the packet path, but is reduced to occurring at the borders of administrative and reputation domains in order to reduce the number of operations on packets in transit and hence added latency.

In order to satisfy the requirements of a large range of reputation scores – needed for Equation 1 – and a small number of reputation levels that are marked on each packet for faster in-transit processing, reputations scores, stored by reputors, are converted into reputation levels, by a process of quantisation, which are then marked on each packet. These conversions need not be in real time but must occur each time a reputee reputation is updated.

Packets marked with higher reputations are forwarded in higher priority queues and packets with lower reputation scores are forwarded in lower-priority queues and are therefore more likely to be dropped. By implementing a proportional relationship, between a packet's reputation and its likelihood of being dropped, scaled against the amount of network congestion, the following benefit occurs. The impact of false negatives is reduced in comparison to the automatic-dropping of packets below a certain reputation since packets are given opportunity to traverse if resources are available, and are only dropped to serve packets of higher reputation. This is pertinent in the case of compromised devices with poor reputations, such as compromised IP cameras, which may still send legitimate packets – as part of their original design specification – that should not be dropped in the absence of attack or congestion. A consequence of this approach is the possibility of attack packets still reaching the victim, although not necessarily at the expense of higher reputed packets.

Details of the header markings inscribed in DiDoS packet headers are described in the next section.

[2] The acceptable attack rate Z_A is defined as the rate of attack that would see an attacker's reputation remain unchanged over the long term and is defined by the base attack penalty P_B and the periodic accumulation amount (see Equation 6).

[3] The direct reputees of the reputor entity processing the report

4.5 Header Specifications

The Internet header length (IHL) field of IPv4 packets indicate the length of the IPv4 packet header in 32-bit words. The minimum value of this 4-bit field is 5 – indicating no packet options present – and its maximum value is 15 (Tanenbaum, 2011). Therefore, IPv4 offers ten 32-bit words of available header-space to accommodate packet options. Through the mechanism of extension headers, available header space is even less of a constraint in IPv6, as IPv6 supports multiple extension headers on each packet and a single extension header can be up to 256 octets (or 64 32-bit words) long (Deering & Hinden, 1998). DiDoS leverages this available option space in order to ensure compatibility with existing Internet infrastructure, as illustrated in Figure. By using Internet options, “non-DiDoS-compliant” routers are able to ignore DiDoS header options in received IP packets whilst forwarding them intact.

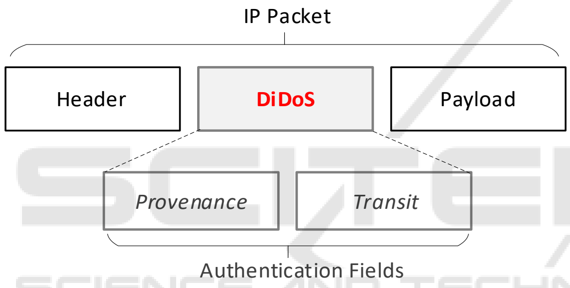

Figure 2: A DiDoS IP option in IP packet header.

As illustrated in Figure, the DiDoS packet option consists of two authentication field sections: *provenance* and *transit*. The provenance section contains fields that allow a source autonomous system to authenticate and process received feedback reports and to identify its reputees to which it must disseminate portions of said report. The transit section contains information that enables a transit AS to verify a packet as coming from a particular AS.

The cryptographic mechanism used to determine packet integrity is Message Authentication Codes (MACs), which are hashes created over various input fields combined with a secret known only to select entities. Each reputor in the DiDoS layer adds its own identifier (that its reputor knows it by), a reputation level for the packet, and two MACs. The first MAC is created with a secret shared between itself and its reputor, and the second MAC is created with a secret known only to itself. The former allows the entity reputor to verify the reputee against the identifier provided, and the latter enables the reputee to verify, during the feedback process, the validity of packet headers contained in received reports.

Hosts may additionally include an identifier that, upon receipt and analysis of feedback reports, enables the identification of the particular piece of – software running on the host – that generated the packet.

A sequence number is added by the first DiDoS entity in the transit path and a timestamp is added by the first Network Time Protocol-compliant DiDoS entity on the path. These fields help to mitigate against feedback report manipulation and repetition attacks, since packets deviating beyond appropriate thresholds from the current time at the receiving entity are disregarded – in the case of reports – and dropped – in the case of network traffic.

Additionally, the source AS also adds a MAC with a secret shared between the destination AS, which prevents the sources packets from being spoofed and holds it accountable to the destination AS.

It is assumed that enrolment processes (in the setup phase) enable the sharing of identifiers and secrets between reputor and reputees. These values can be periodically changed (during the operational phase) to preserve the privacy of reputees and mitigate against cryptanalysis attacks.

5 RESULTS & EVALUATION

5.1 Design Analysis

This section analyzes the DiDoS design against the specified threat model and extends the analysis already included in section 4.

Self-promotion, where an attacker falsely boosts its own reputation is avoided in DiDoS by having a strong correlation between reputation score increases and the ideal behaviour sought, which in this case is not being involved in an attack over periods of time. *Flattery*, where an attacker falsely boosts the reputations of other senders, is possible for an intermediary reputor forwarding the traffic of that sender, since it has opportunity to embed its own reputation level. However, during the post-attack feedback phase, in which reports are disseminated to reputees, entities that incorrectly rated their traffic highly are penalized more than entities that accurately rated the traffic they sent.

Slander attacks that cause a reputor to falsely rate its reputee poorly are achievable if the attacker can somehow get the slander target to send a lot of data, during an attack, since attack feedback reports (the input by which reputee reputations are diminished) contain proofs that demonstrate a sender's contribution to an attack.

Non-participation in the DiDoS architecture can be an attacker's attempt at *evasion* to escape repercussions for misbehaviour. However, since reputors (routers) are responsible for the traffic they forward and, as a result, penalize unauthenticated traffic, little advantage is gained by non-participation.

Another attack avenue for evasion arises from the adaptation mechanism of the reputation update algorithm, in which the reputation attack-penalty is reduced as the attack report rate increases (see section 4.3). An attacker, with multiple (reputee) agents accountable to the same reputor, may attempt to reduce the attack penalty meted by that reputor by initiating extremely large numbers of attacks to solicit similarly high numbers of feedback reports and thus cause the attack penalty to be reduced. Theoretically, if the attack reports are high enough by the attacker sacrificing a small number of agents that are accountable to the reputor in question, then the remaining attacking agents could end up attacking with impunity. However this attack is easily mitigated by capping the amount a single reputee can contribute to the total attack frequency number that is input to the reputation penalty calculations (Equation 3).

The above attack can be described as evasion via collusive *self-destruction*, since an agent destroys its own reputation in order to execute the attack.

Opportunities for *sabotage*, where an attacker hampers the operational ability or integrity of the system, are mitigated by various design features, such as the distributed nature of the architecture and the cryptographic protections that facilitate packet forwarding accounting. For example, a distributed DDoS defence helps to avoid a single point of failure, which, if attacked, could disrupt the entire system.

The practical adoption of DiDoS has associated costs, such as time, equipment and human resources costs. The architecture, however, does offer adoption incentives, the value of which grows geometrically with increasing adoption – via the network effect – since an organization adopting DiDoS not only benefits itself (via spoofing protection against DDoS attacks and increased prioritization of its packets over the internet), but also benefits other entities through 1) the provision of attack feedback reports that help identify malicious actors, and 2) the granular marking of its sent packets, that helps other entities filter malicious traffic.

Another consideration of the architecture is the processing overheads, which are of two types: *in-transit* and *background*. In-transit processing occurs as packets traverse the Internet and contributes to transit latency. The addition and in-transit verification of the message authentication codes that are added to packet headers to facilitate anti-spoofing protection and verifiable attack feedback, are examples of in-transit processing.

However, it is important to highlight that such processing (described in section 4) is not required at every router in transit, but only at the boundaries of reputation domains, such as between autonomous systems. Despite the presence of tens of thousands of autonomous systems (ASs) in the Internet, research has shown that packets, on average, only traverse 3.9 autonomous systems (AS's) for IPv4 and 3.5 AS's for IPv6 (Pappas et al., 2015). Additionally prior work has demonstrated the feasibility of such in-transit MAC processing (Liu et al., 2008).

5.2 Use-case Experiment Setup

An experiment to investigate the effectiveness of the reputation convergence of the DiDoS architecture was constructed in C++. The particular use-case simulated was a local access network (LAN) in which multiple devices access the Internet via a single access router – illustrated in figure 3.

As such, each access device is considered as a reputee to the reputor access router. A proportion of said access devices were considered to be malicious and the rest benign. This disposition was reflected by the differing instance values of the (reputee-) class attributes that determined the data rates that a reputee exhibited during attacks and its attack involvement frequency [4] (AIF), both of which were normally distributed – with malicious devices generating greater in-attack data rates and higher frequencies of attack involvements.

The simulation process worked by iteration over the set of reputees per specified period, to determine, from the aforementioned attributes of each reputee, the number and content of reports passed on to the reputor access router. The number of attacks incidental in each iteration of the simulation, did not directly correspond to the number of attack reports received by the reputor access router, but each provisional report generated was passed through a function incorporating DiDoS adoption rate as a probability of whether said report would reach the access router.

[4] The number of times a reputee is involved in an attack in a given period of time.

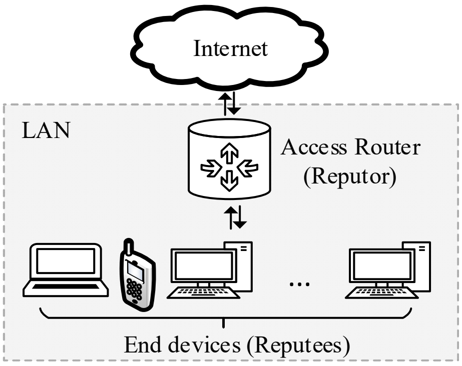

Figure 3: Illustration of use-case simulation scenario of reputor and reputees in LAN.

For the simulation, the coefficients a_1 and a_2 from Equation 1, were set to 1 and 0 respectively, causing the reputee identification strength I, the volumetric attack contribution $\mathcal{V}$, and the appropriateness $\mathcal{A}$ to have no effect on the reputation score penalty size, thus highlighting the effectiveness of the fundamental characteristic of attack-involvement as a means of reputation maintenance. As such, the reputation score update was given by the equation[7]:

$$R_n[x] = R_n[x-1] - \left(\frac{Z_A N(1-m+km)}{Z[x]}\right) P_B \qquad (4)$$

Since, $Z[x]$ was implemented by averaging the number of attack reports received over the last 3 periodic accumulation increments, a method was needed to prevent infinite reputation decrement amounts, arising should $Z[x]$ equal zero.

By choosing the maximum attack penalty to be one third of half of the total range of reputation score values, the minimum value allowed for the actual attack report rate $Z[x]$ was given by the equation[5]:

$$Z_{min} = \frac{2 \cdot Z_{Exp} \cdot P_B}{3 \cdot (|R_{min}| + R_{max})} \qquad (5)$$

Where the expected attack rate Z_{Exp} is the numerator of equation (3).

Since the short data type was used to store the reputation scores, the possible reputation score values ranged from -32768 to 32767.

The variables of the rep update equation were chosen to be as follows: The periodic accumulation amount was chosen such that half of the entire range of reputation scores could be traversed in 200 periodic increments.

The value of k in equation (3) was chosen to be 10, and the expected malicious coefficient m was selected to be 0.5. Based on the period of time T_R required for a reputee to recover from a single reputation penalty decrement of P_B, the allowed attack rate Z_A was defined by:

$$Z_A = \frac{1}{T_R} = \frac{\delta_{PA}}{P_B} \qquad (6)$$

Where T_R, the "base-penalty recovery time"[6], was chosen to be 30 days and δ_{PA} represents the periodic accumulation amount, and was administered, in the simulation environment, daily.

100 devices were considered in the LAN, with half of them possessing the traits of malicious senders and the other half possessing that of benign. The reputee AIFs were random and normally distributed and the simulations were repeated up to 100 times for generalized inferences.

5.3 Results & Discussion

Figure 4(**a**) illustrates the progression of the malicious and benign reputee reputation scores as the simulation progressed. As expected, the reputations of malicious senders quickly converge for a global adoption of 100% to settle at bottom values within 30 days.

For malicious devices, the frequency of attack involvement (AIF) averaged approximately 1.08 per day and ranged between 0 and 3 attacks per day. For benign devices, the average was approximately 0.027 attacks per day or approximately 1 attack every 37 days, ranging between 0 and one attack per day.

The reputations of the benign devices, on the other hand, steadily increase to eventually reach top reputation levels. The y-axis error-bars on the graph represent the standard deviations of the sets of reputation scores for the respective simulation iteration (or day).

It should be noted that the starting value of the reputation scores corresponded to a "Low" reputation level. By penalizing fresh reputees, attacks in which attackers leverage new reputee identifiers are disincentivized.

The effectiveness of the reputation convergence algorithm persists for global adoption percentages as low as 5%. However, as illustrated in Figure 4(**b**), between an adoption rate of between 3% and 4%, the mean reputation scores of malicious reputees no longer progress below their starting reputation scores.

[5] All equation terms are previously defined in section 4.3

[6] The time taken for a reputee to recover its reputation after a single reputation penalty of the base-penalty amount.

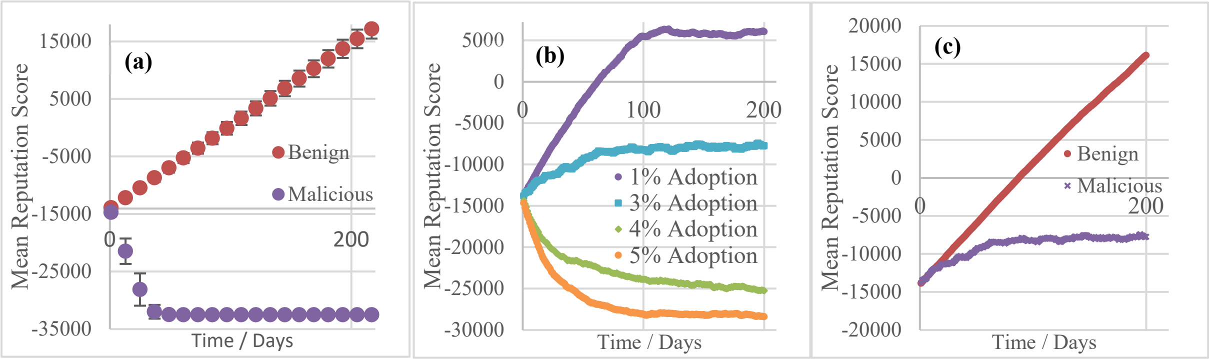

Figure 4: Mean reputation scores a) at 100% DiDoS adoption, b) [of malicious device reputations] at various adoption percentages; c) at 3% DiDoS adoption.

This is because the number of attack feedback reports received becomes insufficient to sustain diminishing reputation scores, despite the low-report-rate penalty-boosting coefficient introduced (equation 3).

The standard deviations of the reputation scores of the malicious devices, increase significantly as global adoption percentages traverse 5% to 3% - from approximately 9,000 to 19,000 respectively. Thus we can glean a critical global adoption threshold for reasonably reliable convergence of malicious device reputations of 5%.

Despite the mean reputation scores of malicious devices increasing above their starting value, for a global adoption percentage of 3%, the mean of reputation scores stabilizes over time to a value below that of the benign reputees, thus enabling, on average, the reputations of the malicious devices to be distinguishable from those attributed to the benign. This is illustrated in figure 4 (**c**).

In order to more widely capture the performance of the reputation update algorithm, the devices in the LAN were divided into two equally sized groups (A & B) and a set of simulations was carried out to observe the effectiveness of the reputation update algorithm with respect to different frequencies of attack involvements.

The mean AIF of group B was held constant at 1.08 per day, whereas the mean AIF of group A was varied (between simulations) ranging from 0.027 to 13 per day in 300 constant increments. We can gather from the literature that such AIF's are not unreasonably high for DDoS botnets, since a study of one such botnet discovered evidence of 300 attacks being launched in just 12 days(Joven & Ananin, 2018). Another study monitoring a collection of botnets was able to detect 406 DDoS attacks, launched in a period of 151 days (Freiling, Holz, & Wicherski, 2005).

For the lower AIFs of group A, the mean reputation scores of group A trended upwards, whilst those of group B trended downwards (see figure 4 (**a**)). However, as the AIF of group A approached and crossed 0.24 per day, the mean reputation scores of the group began to trend downwards (Figure 5), whilst the mean AIF of group B continued to trend downwards.

Further increases in the frequency of attack involvement of group A devices saw the mean reputation scores of the group (A) match the pace of downward trend of group B when their AIF's were almost the same at ~1.08 per day (Figure 6).

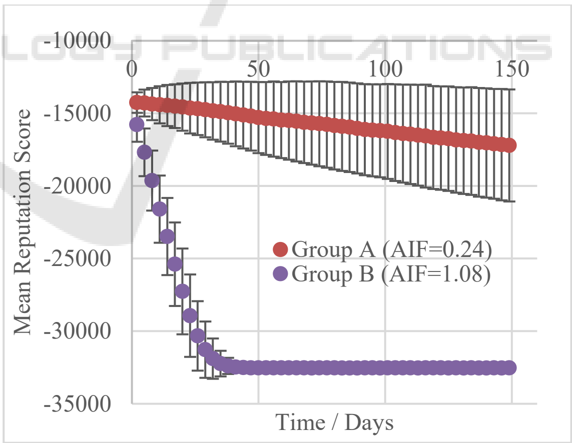

Figure 5: Group reputation scores of devices in a simulated LAN with specified AIFs.

As the benign AIF increased further the mean reputation score of group B devices began to decline slower than that of group A until, the group A AIF reached ~6.75, at which point the mean reputation scores of group B devices began increasing, whilst that of group A continued to decrease. This reversal of the trending direction of the mean reputation score of both groups persisted (becoming more exaggerated) as the AIF of group A increased further.

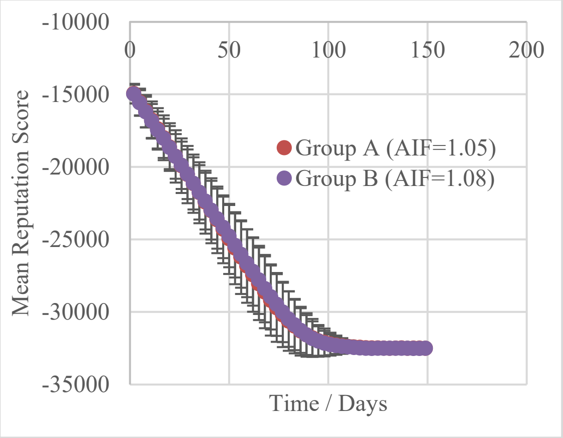

Figure 6: Group reputation scores of devices in a simulated LAN with almost matching AIFs.

This reversal is a significant aspect of the reputation update algorithm as it demonstrates the ability of a reputor employing said algorithm to adapt to the environment of reputees in which it is situated. This is important since, in reality, the AIF may vary drastically across LANs in the Internet. However, the aforementioned reversal shows that even if benign devices have high AIF's, the update algorithm is able to distinguish them from malicious devices, provided that the AIF of the malicious devices is sufficiently greater.

6 CONCLUSIONS

During a DDoS attack, a victim receives competing traffic from an increased number of sources that are unknown to the victim, making it challenging to prioritize legitimate requests. However, though the sources are unknown to the victim, they need not necessarily be unknown to the network. DiDoS leverages this concept to address the challenge of prioritizing legitimate requests, by the downstream[8] signalling of sender (reputee[9]) reputations by network reputors. These reputation levels are embedded, by reputors, into the headers of received reputee packets before forwarding, in order to facilitate legitimate packet prioritization at points of network contention. Therefore, using the information embedded in packet headers, DDoS victims are able to better identify and deprioritize malicious traffic, thus freeing resources to sustain service to legitimate clients.

Since reputations are embedded in packet headers and applied at points of network contention, DiDoS minimizes the impact of false positives and enables compromised devices to continue to send legitimate traffic across the Internet. Furthermore, because packets from entities that participate in the DiDoS architecture are prioritized at points of network contention, DiDoS provides incentives for its adoption.

The DiDoS reputation update algorithm was demonstrated to be able to distinguish between malicious and benign devices over a flexible range of attack involvement frequencies. Additionally, simulation of the reputation update algorithm found a minimum DiDoS adoption rate threshold for reasonably reliable reputation convergence of approximately 5%.

7 FUTURE WORK

Future work on DiDoS involves the detailed specification of the architecture – to the number of bytes for each field in the packet header. An important next step would be to simulate the operation of the DiDoS scheme against realistic traffic patterns and network designs.

Future directions include investigating the possibility of persistently binding pieces of software to an operating system such that software running on a device would be the reputee of the reputor host operating system. In this way, using the post-attack feedback, malware could potentially be discovered on hosts. Another direction is the investigation of the interactions of DiDoS with higher-layer defences and the potential benefits of cooperation to thwart application layer attacks such as Slow Loris attacks.

REFERENCES

Arbor Networks. (2015). *Worldwide Infrastructure Security Report*. Retrieved from https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf

Behal, S., Kumar, K., & Sachdeva, M. (2018). D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events. *Journal of Network and Computer Applications*. https://doi.org/10.1016/j.jnca.2018.03.024

[8] Downstream indicates a flow towards a destination address, whereas upstream indicates towards the source.

[9] Term defined in section 4.2

Cisco, P. (2018). *Cisco Visual Networking Index: Forecast and Trends 2017-2022*. Retrieved from https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf

David Homes, F. N. I. (2016). 2016 DDoS Attack Trends. Retrieved from https://f5.com/Portals/1/PDF/security/2016_DDoS_Attack-Trends.pdf

Deering, S., & Hinden, R. (1998). *RFC 2460: Internet Protocol, Version 6 (IPv6) Specification. Request for Comments (IETF)*. Retrieved from https://tools.ietf.org/pdf/rfc2460.pdf

Freiling, F. C., Holz, T., & Wicherski, G. (2005). Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/11555827_19

Joven, R., & Ananin, E. (2018). DDoS-for-Hire Service Powered by Bushido Botnet. Retrieved October 14, 2019, from https://www.fortinet.com/blog/threat-research/ddos-for-hire-service-powered-by-bushido-botnet-.html

Labovitz, C. (2010). The Internet Goes to War. Retrieved December 11, 2018, from https://asert.arbornetworks.com/the-internet-goes-to-war/

Liu, X., Li, A., & Yang, X. (2008). Passport : Secure and Adoptable Source Authentication. *Nsdi*.

Martin, A., & Others. (2008). The Ten-Page Introduction to Trusted Computing. *Computing Laboratory, Oxford University Oxford*, 1–8.

Mergendahl, S., Sisodia, D., Li, J., & Cam, H. (2017). Source-End DDoS Defense in IoT Environments, 63–64. https://doi.org/10.1145/3139937.3139954

Michael Aaron Dennis. (2012). Denial of Service Attack (DoS attack). Retrieved April 18, 2017, from https://www.britannica.com/topic/denial-of-service-attack

Mirkovic, J., & Reiher, P. (2005). D-WARD: A source-end defense against flooding denial-of-service attacks. *IEEE Transactions on Dependable and Secure Computing*, *2*(3), 216–232. https://doi.org/10.1109/TDSC.2005.35

Naresh Kumar, M., Sujatha, P., Kalva, V., Nagori, R., Katukojwala, A. K., & Kumar, M. (2012). Mitigating Economic Denial of Sustainability (EDoS) in Cloud Computing Using In-cloud Scrubber Service. In *2012 Fourth International Conference on Computational Intelligence and Communication Networks* (pp. 535–539). IEEE. https://doi.org/10.1109/CICN.2012.149

Nicky Woolf, T. G. (n.d.). DDoS attack that disrupted internet was largest of its kind in history, experts say. Retrieved May 29, 2017, from https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

Osterweil, E., Stavrou, A., & Zhang, L. (2019). 20 Years of DDoS : a Call to Action. Retrieved from https://arxiv.org/pdf/1904.02739.pdf

Pappas, C., Reischuk, R. M., & Perrig, A. (2015). FAIR: Forwarding Accountability for Internet Reputability. In *2015 IEEE 23rd International Conference on Network Protocols (ICNP)* (pp. 189–200). IEEE. https://doi.org/10.1109/ICNP.2015.22

Raghavan, S. V., & Dawson, E. (Edward). (2011). *An investigation into the detection and mitigation of denial of service (DoS) attacks : critical information infrastructure protection*. Springer India Pvt. Ltd.

Scott Hilton. (2016). Dyn Analysis Summary Of Friday October 21 Attack. Retrieved May 29, 2017, from https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

Tanenbaum, A. S. (2011). *Computer Networks* (V). Prentice Hall.

Zeijlemaker, S., & Rouwette, E. (2017). A financial evaluation of DDOS defences dynamics from an organisational perspective: how long will these defences hold? In *Proceedings of the 35th International Conference of the System Dynamics Society*. Retrieved from https://www.systemdynamics.org/assets/conferences/2017/proceed/papers/P1171.pdf