

Towards a Distributed Reputation-based DDoS Mitigation Architecture

Andikan Otung

Linacre College

UNIVERSITY OF OXFORD

A thesis submitted for the Degree of

Doctor of Philosophy

December 2023

Abstract

Distributed denial of service (DDoS) attacks are network attacks that leverage distributed attacking agents to cause excessive resource consumption such that legitimate users are denied access. Since their first documented occurrence in 1998, DDoS attacks have grown in magnitude and prevalence to being named the biggest threat to communication service provider (CSP) customers. Exponential growth of the Internet of things (IoT) has compounded the issue and has led researchers to conclude that the current mitigation strategies are unsustainable. We address this problem through the proposition of a **di**stributed **d**efence **o**f **s**ervice (DiDoS) architecture that harnesses the power of collaboratively maintained sender reputations to enable victim networks to make quick informed prioritization decisions on a per-packet granularity to alleviate the impact of DDoS attacks. The DiDoS architecture is evaluated both theoretically and empirically, via simulation in NS3, and is shown to eliminate the in-attack packet loss experienced by adopting clients even for low overall DiDoS adoption percentages.

Since the DDoS problem is vast, spanning many networks and organisations, much research agrees that a cooperative distributed defence is needed. The realization of such solutions requires collaboration between corporate, government, academic and Internet organisations; and such collaboration would be greatly aided by a method to commensurably compare the effectiveness of different defences. However, prior work evaluated defences in isolation, which is inconducive to commensurable comparison, or used benchmarking, which produces evaluation conclusions of limited value and longevity due to the limitations of the attacker strategies and capabilities captured in the static benchmarks.

We address this challenge by the proposition of a framework for the **c**omparative **e**valuation of **DD**oS **d**efences (CED3). CED3 introduces the notion of true effectiveness, which changes the paradigm of defence evaluation from asking whether a defence can withstand a particular attack to asking what it would take to overcome that defence, which boosts longevity of evaluation conclusion. Furthermore, CED3 leverages a defence map, which enables visual comparison between defences in a way that highlights both strengths and weaknesses. We apply the CED3 method to the comparative evaluation of four DDoS defences including the DiDoS framework, which is shown to possess the greatest true effectiveness and the second greatest ranking of adoptability (second to the defence of increasing the network capacity of a victim service).

Acknowledgements

First and foremost, I would like to thank God for blessing and sustaining me through this journey. For ordering my steps, my inspiration and for keeping me physically and mentally able through the COVID-19 pandemic, I am truly grateful.

I would like to thank my supervisor Prof Andrew Martin for his brilliant support, guidance and encouragement throughout this process. I am particularly grateful to him for making himself available for meetings throughout the journey and for his timely feedback, despite many other commitments.

I would like to thank my family for their amazing goodwill and support manifested by their keen interest and kind words. I am particularly grateful to my father Prof Ifiok E. Otung, not only for his encouragement and advice concerning my DPhil, but also for entertaining my myriad of random unrelated and often highly technical ideas.

I would like to thank my college advisors Dr Cezar Ionescu and Dr Fergus Cooper for the advice and encouragement throughout the process. I am also grateful to the staff at Linacre college for providing a warm and welcoming environment.

I am grateful to my examiners, particularly Ivan Martinovic and Chris Hargreaves for the advice and guidance offered in the later stages.

I received invaluable support from so many individuals, from my colleagues and teachers at Oxford CDT in Cybersecurity, from my Church family, past lecturers, and my friends. There are a few names I'd like to mention in particular: John Durodola, Chris Neil, Seb Savory, Kenneth Tong, Jonathan Landau and Jason Nurse.

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) [EP/P00881X/1]. This project made use of the University of Oxford Advanced Research Computing (ARC) facility [http://dx.doi.org/10.5281/zenodo.22558].

Contents

Abstract i
Acknowledgements iii
Contents v
Acronyms ix
Chapter 1: Introduction 1
1.1 Motivation 1
1.2 Mitigation Challenges & Approaches 2
1.3 Objectives & Contributions 4
1.4 Thesis Structure 6
Chapter 2: Literature Review 8
2.1 Background 8
2.1.1 DDoS Attacks 8
2.1.2 DDoS Defences 13
2.1.3 Discussion 17
2.2 Related Work 20
2.2.1 Accountability-based DDoS Defence 20
2.2.2 Existing Commercial Technologies 25
2.2.3 Collaborative Architectures and Core-based DDoS Defences 28
2.2.4 Other DDoS Defence Options 30
2.2.5 DDoS Defence Evaluation and Benchmarking 34
2.2.6 Discussion 41
Chapter 3: DiDoS Design 45
3.1 Introduction 45
3.2 Goals and Methodology 46

3.3 Threat Model and Assumptions 49
3.4 System Description 52
3.4.1 Overview 52
3.4.2 Hierarchical Identification Scheme and Roles 54
3.4.3 Collaborative Reputation Management System 56
3.4.4 Distributed Mitigation Mechanism 66
3.4.5 DiDoS Header Specifications and Cryptographic Protection 67
Chapter 4: DiDoS Evaluation 72
4.1 Overview 72
4.2 Design Assessment 72
4.2.1 Review 72
4.2.2 Threat Analysis 78
4.3 Empirical Validation 82
4.3.1 Reputation Convergence 82
4.3.2 Mitigation Performance 88
4.4 Discussion 99
Chapter 5: CED3 Design 105
5.1 Introduction 105
5.2 Goals and Methodology 106
5.3 Foundations 109
5.3.1 Inevitable Theoretical Subversion 109
5.3.2 True Effectiveness 111
5.3.3 Scopes 112
5.3.4 Representative Evaluation 112
5.4 Framework 114
5.4.1 Overview 114
5.4.2 Evaluation Context 116
5.4.3 True Effectiveness 119
5.4.4 Adoptability 127
5.4.5 Comparison 136
Chapter 6: CED3 Evaluation 139

6.1 Overview 139
6.2 Environment 139
6.2.1 Defences 139
6.2.2 Network 141
6.2.3 Legitimate Traffic 143
6.3 Threat Model and Attack Scenarios 143
6.4 Results 144
6.4.1 Metrics 144
6.4.2 Defence Effectiveness 148
6.4.3 Adoptability 154
6.5 Discussion 157
Chapter 7: Conclusions 160
Chapter 8: References 167
Appendix A. Further Information i
A.1 NS-3 i
A.2 Alternative Metrics & Combining Formulae ii
Appendix B. Deeper Dive iv
B.1 DDoS Perspectives iv
B.2 Structured Theoretical Analysis vii
B.3 Choosing the TrustGuard Threshold xi
Appendix C. Result Set Data xii

Acronyms

AFE	Attack Feedback Endorser
AFO	Attack Feedback Owner
AFR	Attack Feedback Report
ARC	Advanced Research Computing
AS	Autonomous System
ASIC	Application-Specific Integrated Circuit
ASHM	Attack Strategy Heat Map
BASH	Bourne-again SHell
BGP	Border Gateway Protocol
CED3	Comparative Evaluation of DDoS Defences
CDN	Content Delivery Network
CPU	Computer Processing Unit
CSP	Communication Service Provider
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Protocol
DiDoS	Distributed Defence of Service
EDOS	Economic Denial of Service
FAIR	Forwarding Accountability of Internet Reputability
FGCD	Fine-Grained Capabilities for Flooding DDoS Defence
GRE	Generic Routing Encapsulation [1]
HTTP	HyperText Transfer Protocol [2]

ICMP	Internet Control Message Protocol [3]
IoT	Internet of Things
IP	Internet Protocol
MAC	Message Authentication Code
ML	Machine Learning
NS3	Network Simulator 3
NTE	Normalized True Effectiveness
PoP	Point of Presence
PPA	Passive Periodic Accumulation
RAM	Random Access Memory
RMC	Reputation Management Center
RoQ	Reduction of Quality
SDN	Software-Defined Networking
SLA	Service Level Agreement
SLURM	Simple Linux Utility for Resource Management
Tbps	Terabits Per Second
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UEQ	User-experienced Quality of Service
VIP	Virtual IP
VLAN	Virtual Local Area Networking [4]

Chapter 1: Introduction

1.1 Motivation

Denial of service (DoS) attacks are attempts by attackers to prevent legitimate users[1] 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. Several sources place the first DDoS attacks as occurring between 1998 and 2000 [5]–[9]. However, two decades on, despite significant academic, industry and military research efforts [10][11], DDoS remains a significant threat to Internet security today [12]–[14]. In fact, far from being a solved problem, the number and scale of DDoS attacks continues to grow [15]–[17].

Largescale DDoS attacks such as the 1.2 Tbps attack against Domain Name Systems (DNS) provider DYN [18]–[20], and the staggering 125,000 weekly attack-launch rate helped tout 2016 as the "year of DDoS attacks" [21]. Other than the popularity of the multiple websites that were rendered inaccessible by the DYN attack, of note was the use of Internet of things (IoT) devices to carry out the attack, which comprised of 100,000 compromised IP camera's, baby monitors and other kinds of devices. An estimated six billion IoT devices were in use at the time of the DYN attack mentioned and, that number is expected to increase by a factor of 5 to reach ~30 billion by 2030 [19], [22], [23]. The increasing phenomenon of the IoT, therefore, would seem only to compound the threat of DDoS.

An investigation on the impact of DDoS by Incapsula – involving hundreds of North American organisations that ranged in size from hundreds of employees to thousands – estimated the cost of DDoS attacks to be an average of $500,000 per attack [24]. Two thirds of attacks were reported to have lasted for more than six hours; with most attacks (86%) lasting less than 24 hours. The reported costs incurred, were not limited to the IT group but included impacts on security and risk management, customer service and sales. Affected companies reportedly had to replace hardware or software.

[1] Legitimate users of a service are clients that use the service without the intent of causing its denial to other users.

Another development in the evolving threat of DDoS is the "democratization of censorship", which is a phrase coined by independent investigative journalist Brian Krebs [25]. It refers to the increasing inescapability of censorship in the Internet and how that censorship can be implemented by lone individuals who have amassed powerful weapons in cyberspace, such as DDoS-capable botnets. Krebs's own journalist website was hit with a historically large 620 Gbps DDoS attack, which led to it being dropped from the services of DDoS protection provider, Akamai, to preserve their paying clients.

The types of perpetrators and reasons behind attacks vary. Common motives include ideological hacktivism, cyber vandalism or political or financial gain [26]. Perpetrators can be lone individuals, members of organised criminal gangs, members of activist groups like Anonymous [27], or even nations states participating in cyberwarfare [28]. Since DDoS attacks are a viable method to satisfy the diverse motivations of these pervasive societal groups, it is without a doubt that, if unaddressed, attacks will continue to cost organisations [29], undermine online freedoms and threaten the increasingly Internet-dependent way of modern societal life. However, there are many obstacles to resolution.

1.2 Mitigation Challenges & Approaches

Part of the challenge in DDoS mitigation is the "arms race" that exists between defenders and attackers, which sees defensive measures continually bolstered and subverted. Other fundamental challenges, from which enormous complexity in the field derives, include:

- Distinguishing between malicious and benign[1] activity, behaviour and entities, in an environment with evolving attackers, norms and services
- The elimination of attacks as close to the source as possible, whilst minimizing false positives
- The provision (by the defence) of benefits in partial deployment with support for incremental deployment and benefits on the way to full deployment[2]
- Maximizing defence effectiveness, security properties and benefits whilst maintaining the quality of user experience provided and compatibility with existing Internet infrastructure
- Presenting deployment incentives that overcome obstacles – including costs

Note that this is not intended to be a comprehensive list of all the factors influencing successful DDoS defence design and adoption, but, rather, it is an indication of certain factors that make addressing DDoS particularly complex.

[1] Benign refers to the intent of the initiating entity
[2] Full deployment here refers the scale of the deployment of a defence that would provide the maximum benefit.

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, and traffic scrubbing [30], [31]. However, the cost of doing so, combined with the rate of increasing attack strengths, has led researchers to conclude that continuing with state-of-the-art methods to defend against DDoS attacks is not sustainable [32], [33].

If attack traffic were to be intercepted further away from its destination, the victim would be far less susceptible to DoS as the elimination of attacks close to the source would inhibit the cascade resource consumption, characteristic of flooding DDoS attacks and thus reduce bottlenecks. However, this is not practicable unless traffic can be identified as malicious away from the victim, which is difficult to achieve outside the context of victim-centric information (such as what constitutes "wanted" or "normal" to the victim) and if the malicious entity mimics legitimate traffic.

Reputation-based defences address this by associating traffic quality with the past behaviour of its sender and thus attempt to prevent agents from attacking with impunity. For such defences 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. These defences must tackle further sets of challenges associated with the defensive technique. One being the use of an appropriate identifier to which reputations and entities are bound. IP addresses, for example, do not make for appropriate identifiers due to the possibility of IP address spoofing, Dynamic Host Control Protocol (DHCP), Network Address Translation (NAT) and mobile clients; which would all compromise the integrity of the reputations bound to the respective addresses.

Fine-Grained Capabilities for Flooding DDoS Defence Using Client Reputations [34], for example, proposes a reputation-based DDoS defence that is shown to effectively respond to attacks yet the authors acknowledge the use of IP addresses as identifiers as a weakness to be addressed in future work. More on reputation-based defences can be found in section 2.2.1 of the Related Work chapter.

Other than technical solutions operating in cyberspace, approaches to preventing DDoS include law enforcement intervention to arrest attackers and seize perpetrating assets like domain names and web servers. A recent example of this is the cyber-attack website *webstresser.org*, which allowed individuals with little to no technical expertise, to execute powerful DDoS attacks for as little as $15 (~£11) [35], [36]. One week after the website was shut down at the start of the 2nd quarter of 2018, DDoS protection provider Link11 reported a 60% decrease in DDoS attacks in Europe. However, the number of attacks for the whole quarter was later reported to have decreased by only 36% [37], [38], which could be as a result of perpetrators transitioning to alternative attack avenues.

Another approach is the improving of device security in order to avoid infections that could lead to the creation of powerful attack botnets in the first place. An example of this is the Secure by Design initiative by the UK government Department of Digital, Culture, Media & Sport (DCMS), which spearheaded an investigation that led to recommendations in protecting IoT devices from compromise [39].

1.3 Objectives & Contributions

We address the DDoS problem through a novel reputation-based DDoS mitigation architecture called DiDoS (for Distributed Defence of Service, pronounced "dye-dos"). DiDoS leverages collaboratively maintained sender reputations by embedding reputation levels to packets to allow distinction between malicious and benign traffic closer to their source. An overview of DiDoS is illustrated in Figure 1. Although the steps 1 to 5 can be considered as a cycle that repeats continuously, in an operational DiDoS implementation, each of the steps can be happening concurrently at various points in the network.

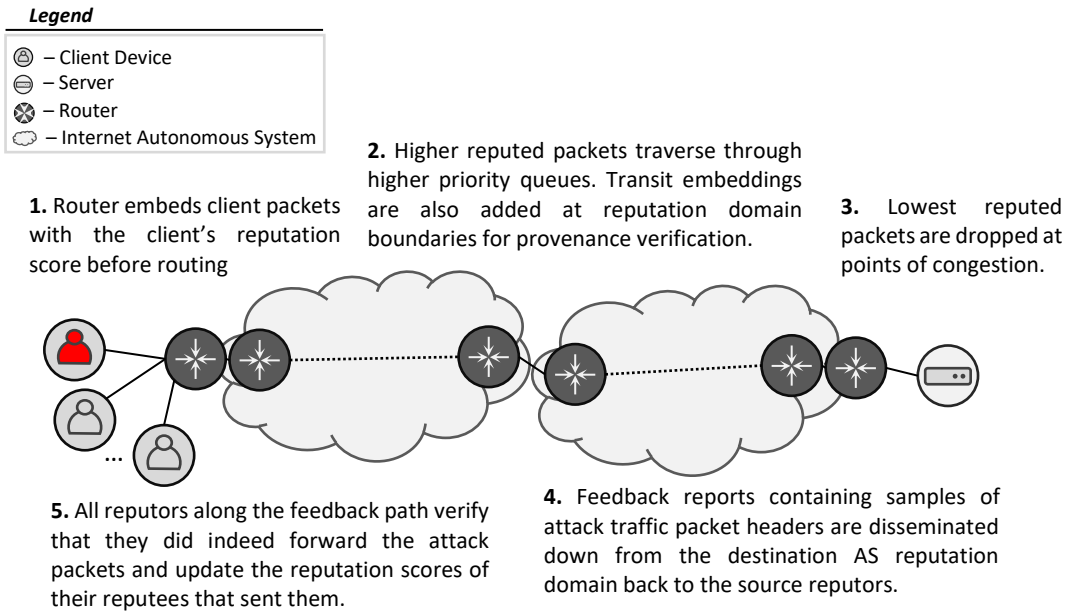


Figure 1: Overview of the DiDoS Defence Proposition

The novelties of this proposition lie in the collaborative way in which the source routers are equipped by the destination reputation domains to facilitate upstream DDoS attack mitigation on a per-packet granularity. As part of this, the headers of packets sampled during an attack are placed in a feedback report that is disseminated to the downstream reputors. These headers contain cryptographic mechanisms that allow each reputor to verify whether the reputees indicated by the packets in the report did indeed send the packet and whether the reputor itself did forward the packet. There are many nuanced mechanisms that characterize the novelty of DiDoS, the details of which are described in section 3.4.

Another novelty, compared to the state-of-the-art commercial DDoS mitigation solutions, is the distributed way in which DiDoS mitigates attack traffic. The state-of-the-art processes traffic at scrubbing centers and at satellite points of presence. However, this approach is vulnerable to regional outages since due to the limited network access and mitigation capacity at the edges. Furthermore, the DDoS mitigation servers of the state-of-the-art do not receive per-packet information to facilitate granular mitigation of attacks. DiDoS improves on the state-of-the-art by enabling informed mitigation of attack packets right up to the client device that's originating the traffic. Additionally attack mitigation does not require an attack to be detected and analysed before mitigation occurs, as is required for the state-of-the-art, since the lowest reputed packets are automatically dropped at any points of congestion. See section 2.2.2 for more information on the state-of-the-art.

This research project is by no means the first to identify that a distributed approach is necessary to effective DDoS mitigation. In fact, numerous publications are in agreement that a cooperative, distributed approach is necessary in overcoming the DDoS threat [40]–[44]. However, due to interdependence between global Internet stakeholders, deployment of such approaches would require acceptance by, and collaboration between many organizations [42], including government agencies, corporations and Internet organizations; such as the IETF [45], Cisco, BT or the NCSC [46]– to name a few.

Underpinning any such collaboration would be agreement on which, of the many possible defences, are deployed. The realization of any such consensus would be greatly aided by a method to systematically assess performance, impact and suitability of possible defensive solutions in a way that allows them to be compared. This method would ultimately supersede the numerous heterogenous methods of DDoS defence evaluation to date and would help stakeholders navigate the myriad of DDoS defence solutions.

As shall be shown in the literature review (section 2.2), prior work on DDoS defence evaluation has addressed a number of factors including metrics and representative test scenarios but does not solve the following problems of longevity of evaluation results, commensurable comparison of evaluated defences and visual assessment and comparison that enables both a defences strengths and weaknesses to be highlighted. The literature contains numerous examples of defences that are shown to be effective under favourable conditions but are either not tested or are shown to be ineffective under unfavourable conditions. An example is TrustGuard [47], which monitors the lack of packet-size variation as a metric to drop malicious flows but is not tested under conditions where an attacker varies the size of its packets.

Therefore, another goal of this research project is to construct a method through which DDoS defences can be effectively and commensurably compared. The creation of this approach, called CED3 (Comparative Evaluation of DDoS Defences), leveraged inductive reasoning to build on prior DDoS evaluation work in the literature. Spurring from an induced principle of

inevitable theoretical subversion (see section 5.3.1), CED3 proposes a technique that enables systematic evaluation and granular comparison of DDoS defences in an attacker agnostic manner. In order to assess the usefulness and viability of CED3 as an option for comparative DDoS defence evaluation, CED3 was applied to four defences including DiDoS.

The overarching aim of this research is to investigate and develop solutions that could contribute to bringing closer an effective DDoS-mitigated Internet eventuality by addressing key problems identified in the literature: 1) the unsustainable arms-race between DDoS attack and mitigation strategies which is being compounded by continuous growth in the IoT and 2) the lack of suitable methods to evaluate DDoS defences in a way that captures their intrinsic value whilst allowing them to be commensurably compared.

The research goals of this project can be captured in the research questions below:

Research question 1: *How can metrics strongly correlated to DDoS attackers or the DDoS effect be leveraged to disrupt the current unsustainable trend of DDoS attacks and mitigation strategies?*

Research question 2: *How can attacks be mitigated closer to the source in a way that minimizes false positives and negatives?*

Research question 3: *How can the effectiveness of DDoS defences be captured in a way that is less dependent on nominal test conditions whilst enabling commensurable comparison between defences?*

Research question 4: *How can DDoS defence evaluation and comparison be facilitated such that both a defence's strengths and weaknesses can be visually highlighted and compared?*

The related publications of this thesis are detailed in Table 1.

Title	Type	Venue	Related Chapters
Distributed Defence of Service (DiDoS): A Network-layer Reputation-based DDoS Mitigation Architecture [48]	12 Page Paper	ICISSP	3 & 4
Towards Comparative Evaluation of DDoS Defences [49]	10 Page Paper	IEEE SINCONF	5 & 6
CED3: Comparative Evaluation of DDoS Defences [50]	Poster	DSDS	5 & 6

Table 1: A table of related thesis publications

1.4 Thesis Structure

The rest of the thesis is structured as follows:

Chapter 2: The *Literature Review* chapter is divided into two sections: *Background* and *Related Work*. The background surveys the field of DDoS through the exploration of attack

and defence taxonomies. The development of the particular taxonomy used to structure the CED3 defence map (in 5.4.5) is found in section 2.1.3.1. The related work section looks at prior work on accountability-based DDoS defences and DDoS defence evaluation methods. It also provides a summary of the state of the field (in section 2.2.6), from which the research objectives were derived.

Chapter 3: The *DiDoS Design* chapter details the design of the distributed defence of service architecture (DiDoS). Beginning with the goals of the design, the chapter proceeds to specify the threat model against which the defence should be resilient. The design process was recursive, involving numerous iterations, however for brevity, only the final design is specified in section 3.4.

Chapter 4: The *DiDoS Evaluation* chapter presents the evaluation of the DiDoS architecture. The evaluation is comprised of two parts: *Design Assessment* and *Empirical Validation*. The design assessment section theoretically assesses the design against other related defences and also against the previously specified design goals. The section also discursively assesses how the design mitigates against threats specified in the threat model. The empirical validation section describes and presents the results of the experiments that were designed to 1) demonstrate the ability of DiDoS to differentiate between malicious and non-malicious senders, and 2) discover the effectiveness of the architecture at thwarting DDoS attacks, when the *collaborative reputation management system* (3.4.3) is applied to the *distributed mitigation mechanism* (3.4.4). The chapter concludes with a discussion on the merits and shortcomings of DiDoS and the evaluation process, including assumptions, and proposes areas for future work.

Chapter 5: The *CED3 Design* chapter presents the framework for the Comparative Evaluation of DDoS Defences (CED3). Beginning with the goals, the chapter details the inductive process by which the tenets of the framework were derived. Said tenets are described in section 5.3. The CED3 process, including assessing true effectiveness, adoptability and comparing defences, is presented in section 5.4.

Chapter 6: The *CED3 Evaluation* chapter details the application of the CED3 framework to the comparative evaluation of four DDoS defences, including DiDoS. The comparative evaluation comprises assessments of both true effectiveness and adoptability of the candidate defences. The chapter concludes with a discussion on the outcome of the evaluation and the suitability of the CED3 framework, including potential avenues for future work (section 6.5).

Chapter 7: Concludes the thesis with a summarization of the main contributions and a brief exploration of possible avenues for future work. Also included is a list of some key insights that shaped the research and a brief discussion of the main limitations of the research identified in the aforementioned discussion sections (4.4 and 6.5).

Chapter 2: Literature Review

2.1 Background

2.1.1 DDoS Attacks

There exists multiple definitions of Distributed Denial of Service (DDoS) attacks [51]–[53]. Generally, DDoS is a form of Denial of Service (DoS) that is effected using multiple systems.

This project defines a DDoS attack as: *an attacker's attempt to deny normal users access to online services by consuming available resources, using multiple, usually compromised, systems.*

From this definition, the existence of Parties, Resources and Connectivity can be gleaned, where Resources are connected by the internet and are controlled (wholly or partially) by respective Parties. During a DDoS attack, the four parties include: the victim, the attacker, Communications Service Providers (CSP) and other internet users. Furthermore, resources are the processing power, memory and bandwidth capacities possessed by routers, servers, personal computing devices and switches that are connected to the internet.

It can also be gleaned from the definition that the distinct attribute that differentiates DDoS attack agents[1] from all other communicating parties is the consumption of victim resources *with* the intent of denying legitimate users. This distinction holds true across all types of DDoS attacks, such as UDP Flood or Syn Flood attacks [54]. However, the extent to which intent can be digitally ascertained so as to prevent service disruption by DDoS attack agents is a key, evolving and challenging issue.

This section discusses and summarizes the different kinds of DDoS attacks in the literature. As defined previously, distributed denial of service (DDoS) is a form of denial of service[2] (DoS) that is effected by multiple distributed attack-agents. This research project considers DDoS in the Internet, which is accomplished by malicious communications rather than through other means such as cable-cutting. There are other forms of DDoS attack that lie

[1] Where an agent can be considered as an autonomous piece of code that is reactive and performs some function on behalf of an end user or another program

[2] Denial of service (DoS) attacks are attempts by attackers to prevent legitimate users from accessing connected services

outside the scope of the traditional definition. These include attacks that aim to reduce the quality of service experienced by users of the victim service and economic denial of sustainability attacks (EDoS) [55]. In both cases, denial of service can be achieved indirectly, through inadequate user experience or economic coercion – respectively.

An example of the former is a reduction of quality (RoQ) attacks [56], which exploits the transient behaviour of a system in order to reduce its capacity, cause delays and create jitter. EDoS, on the other hand, occurs in hosted cloud environments where victims pay providers to host services on their behalf, and are charged by the provider based on the amount of resources used in hosting that service. In this case, EDoS is achieved when an attacker launches an economic Distributed Denial of Service (eDDoS) attack, consisting of legitimate requests to the hosted service, that causes more provider resources to be used and, in turn, higher financial charges to the victim, to the point where the victim is no longer able to afford the costs of hosting. EDoS differs from DDoS in that the goal is not to deny legitimate clients from accessing the service directly, but to achieve this means indirectly through financial means via the process described. However, the focus of this literature review is on DDoS.

In [57], Mirkovic and Reiher identify four stages of a DDoS attack: *recruit*, *exploit*, *infect* and *use*. Recruit refers to the scanning of remote machines to identify candidate hosts with security vulnerabilities. Once these are identified the vulnerability is exploited and the host is infected with malware. The last stage refers to the participation of the infected host in the execution of DDoS attacks. An example of the recruit-exploit-infect strategy is the infection of a host machine by a trojan horse such as a malicious email attachment.

On DDoS attacks, the main contribution of [57] is a taxonomy of DDoS attacks that classifies attacks based on eight top-level classification criteria; each with a unique classification code – as illustrated in Figure 2. Not illustrated, are the subcategories possessed by each criterion. These inherit the classification code of their parent classification criterion and are assigned a number to differentiate them from fellow sibling categories. For example, the category, *host*, of the top-level criterion, *victim type (VT)*, is given the classification code VT-2. When multiple layers of parents exist, the classification code of the subcategory incorporates the classification codes of all parents in the lineage via concatenation of each parent code with the colon character, ":", as shall be seen later. It is also possible for DDoS attacks to adopt the classification codes of multiple subcategories within the same parent category, if it exhibits the features of those subcategories. In this way any DDoS attack can be classified within the presented taxonomy using the classification codes.

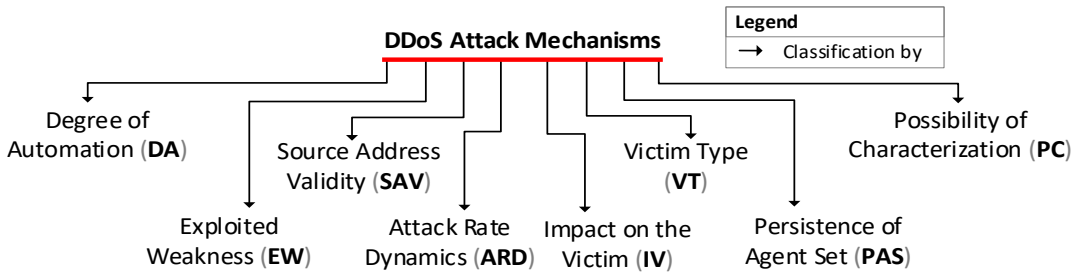


Figure 2: Eight top-level classification criteria of DDoS attacks [57]

Beginning from the left of Figure 2, **degree of automation** (DA), refers to the level of automation applied across the four stages of attack. The categories of DA are *manual* (DA-1), *semi-automatic* (DA-2) and *automatic* (DA-3). In manual attacks, the attacker individually recruits exploits and infects hosts to create a botnet. Afterwards, attacks using the botnet are also initiated by himself. Mirkovic and Reiher acknowledge, in the paper, that modern DDoS attacks no longer belong to the manual category. Semi-automatic attacks automate the recruit, exploit and infect phases, whilst *use* remains under the manual operation of the attacker. Automatic attacks automate all phases including *use*. This requires attack target, time, and type to be preprogrammed into the attack code. However, changes can be enabled through a backdoor mechanism to each infected host. Another layer of subcategories that differentiate attacks by their communication mechanism and host scanning strategies, is included in the taxonomy under the DA classification criterion.

Exploited weaknesses (EW) refers to the vulnerability exploited to effect denial of service. This classification criterion possesses just two subcategories: *semantic* and *brute force*. Semantic attacks exploit a specific design flaw or implemented (application or protocol) bug. For example, the TCP SYN attack exploits a flaw in the TCP handshake that allocates substantial memory upon the immediate receipt of a TCP SYN request. Brute force attacks on the other hand, achieve denial of service by overwhelming a victim's resources with transactions such as an HTTP flood attack. Since no vulnerabilities are exploited to exacerbate resource consumption, brute force attacks generally require a greater volume of attack traffic.

Source address validity (SAV) classifies attacks by whether spoofing is used or not into two subcategories: *spoofed source address* (SAV-1) and *valid source address* (SAV-2). The spoofed source address subcategory contains further classifications that differentiate between the type of spoofing employed such as randomly spoofed addresses or spoofed addresses comprising of non-routable source addresses.

Attack rate dynamics (ARD) distinguishes between attacks that exhibit a constant rate of sent packets and those that vary via two categories: *constant rate* (ARD-1) and *variable rate* (ARD-2). Within the variable rate category, differentiation is made between attacks that

exhibit *increasing rate* (ARD-2:RCM[1]-1) of packets and those that exhibit a *fluctuating rate* (ARD-2:RCM-2) of attack traffic.

Impact on the victim (IV) is subdivided into two categories, disruptive (IV-1) and degrading (IV-2), based on whether service is disrupted or degraded. The disruptive category is subdivided further into three categories, based on how easily the victim can recover from attack, named: *self-recoverable* (IV-1:PDR-1), *human-recoverable* (IV-1:PDR-2) and non-recoverable (IV-1:PDR-3). It is noted by Mirkovic and Reiher that, whilst non-recoverable attacks are theoretically conceivable, no reliable information was available – at the time of writing – on their practical occurrence.

Victim type (VT) classification criteria is split into four categories that describe the attacked victim: *application* (VT-1), *host* (VT-2), *resource attacks* (VT-3), *network attacks* (VT-4) and *infrastructure* (VT-5). Application attacks degrade or disrupt functionality or service provided by a targeted application. Host attacks render a target machine unavailable by subverting some aspect of the host machine that is necessary for the provision of service, such as available dynamic memory. Resource attacks target crucial resources in a victim's network, such as important routers, links or DNS servers. Network attacks consume the bandwidth of the victim. Infrastructure attacks target some distributed service critical to the operation of the global Internet. In isolation, an attack on a core router could be considered a host attack, but if it is a victim in an attack on multiple instances of a critical service, then it would be considered an infrastructure attack. Example targets include domain name servers, routing protocols and certificate servers.

The classification criteria, **persistence of agent set** (PAS), possess only two categories: *constant agent set* (PAS-1) and *variable agent set* (PAS-2). The criteria differentiates between attacks whose attackers behave in a similar way for the entirety of the attack (PAS-1) and attacks with varying attacking agents at different points in the attack (PAS-2).

The last classification criterion is **possibility of characterization** (PAC). This possesses two categories *characterizable* (PC-1) and *non-characterizable* (PC-2), which describe whether the attack can be identified and characterized by the data contents of its header and payload (PC-1) or not (PC-2). Characterizable attacks are further differentiated by the subcategories filterable (PC-1:RAVS:-1) and non-filterable (PC-1:RAVS-2). Filterable attacks are able to be filtered by a firewall – based on their characterization – without causing significant harm to legitimate clients because the filterable packets contain malformed content, or they are destined for non-critical services. Non-filterable attacks are characterized in a way that cannot sufficiently differentiate their packets from those of legitimate clients; thus, filtering

[1] RCM – Rate Change Mechanism

according to their characterization alone could cause significant disruption to legitimate clients.

Mirkovic and Reiher acknowledged room for extension of their taxonomy and other publications have been written that organise DDoS attacks differently [26], [58], [59] and in more niche areas.

Understanding DDoS Attack & Its Effect In Cloud Environment [60], by Deshmukh and Devadkar, is one such example that focuses on DDoS in the cloud environment. This is illustrated in Figure 3. In it, a taxonomy is presented that excludes the recruit exploit and infect stages previously considered by Mirkovic and Reiher.

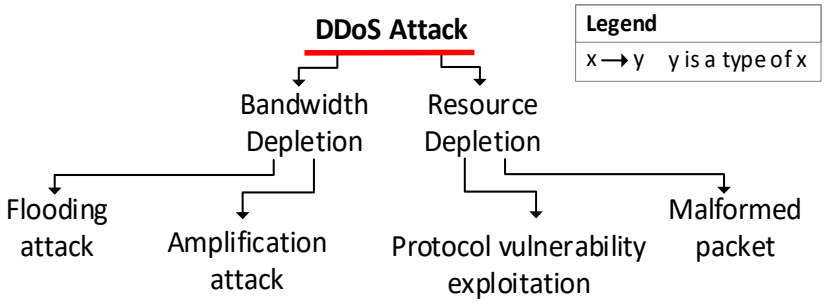


Figure 3: A DDoS taxonomy [60]

Despite the focus on a cloud environment, this taxonomy is more compatible with industry nomenclature of DDoS attacks [54], [61], [62] because both ignore attacker provenance and focus on distinct features of attacks themselves. Simple examples include TCP flood and UDP flood attacks, which are named after the overwhelming nature of the attacks and the type of packets employed.

Deshmukh and Devadkar, define bandwidth attacks as attacks that overwhelm a defences bandwidth directly or indirectly (flood vs amplification attack) and the only alternative, resource depletion attacks, as those that exploit protocol vulnerability or employ malformed packets. Therefore, the taxonomy does not accommodate for resource depletion attacks that are neither executed by malformed packets nor by exploiting protocol vulnerabilities. An example of this is an attack that uses relatively small bandwidth to instigate numerous legitimate content searches on a website. This application layer attack is neither a bandwidth depletion or a resource depletion attack - as resource depletion is defined in the taxonomy.

Presented in [63] is a classification of DDoS attacks based on the network layer of impact. The classification hierarchy possess just five categories: *network device level*, *OS level*, *application level*, *data flood*, *protocol feature level.*

The below table lists various DDoS attacks and their classifications according to the taxonomy specified in [60].

Attack Type	Flooding	Amplification	PVE[1]	Malformed Packet	Comments
Coremelt [64]	✓	-	-	-	Traffic only between attackers
Fraggle [65]	-	✓	-	-	Spoofed UDP packets
NTP Amplification [66]	-	✓	-	-	Spoofed NTP requests
Ping Flood [67]	✓	-	-	-	ICMP ping messages
Ping of Death [68], [69]	-	-	-	✓	Exploits IP packet fragmentation
Shrew [70]	-	-	✓	-	Exploits TCP's time-out
Slowloris [71]	-	-	-	-	Extends connection time slowly
Smurf attack [72]	-	✓	-	-	Spoofed ICMP packets
SYN ACK Flood [73]	-	-	✓	-	Exploits TCP handshake

Table 2: Table illustrating classification of various attacks

As shown, SlowLoris attacks do not fit into the definition of flooding attacks provided by [60], unless the definition is extended to include flooding of resources other than bandwidth, such as CPU and memory. Any protocol that can be leveraged in an amplification attack, where the source address is spoofed to that of the victim, can be used in a flood attack, despite its potentially diminished attack-potency.

2.1.2 DDoS Defences

2.1.2.1 Taxonomy

In [57], Mirkovic and Reiher present a taxonomy of DDoS defence mechanisms based on three top-level classification criteria – as illustrated in Figure 4.

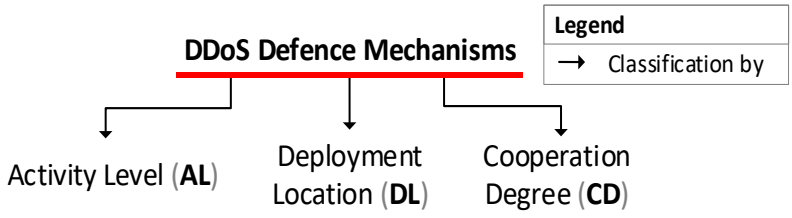


Figure 4: Taxonomy of DDoS defence mechanisms [57]

As with their taxonomy of DDoS attacks, presented in the previous section, each classification criteria possesses multiple categories and each category may possess subcategories, and so on, thus forming a hierarchical classification structure.

Beginning with the left of Figure 4, the classification criterion *activity level* possesses two categories: *preventative* (AL-1) and *reactive* (AL-2). The differences specified between preventative and reactive are that reactive defences need to detect the attack and that preventative defences may eliminate the possibility of DDoS attacks altogether.

[1] Protocol Vulnerability Exploit

Preventative mechanisms are further distinguished, by their prevention goal, into two subcategories: *attack prevention* (AL-1:PG-1) and *DoS prevention* (AL-1:PG-2). Attack prevention has 2 subcategories: *system security* (AL-1:PG-1:ST-1) and *protocol security* (AL-1:PG-1:ST-2). System security is an attack prevention mechanism that involves improving the security of connected systems via updates to machines to remove bugs and harden security, whereas protocol security prevents by addressing the design phase to fix flawed protocol designs.

Resource accounting (AL-1:PG-2:PM-1) and *resource multiplication* (AL-1:PG-2:PM-2) are two subcategories of the DoS prevention category. Resource accounting defence mechanisms use access privileges to allow access to only legitimate and well-behaved users. Resource multiplication defence mechanisms, on the other hand, counteract denial of service by increasing the resources available to both legitimate and non-legitimate users.

Returning to reactive defence mechanisms, these are classified further into three subcategories based on their attack detection strategy. These subcategories are *pattern detection* (AL-2:ADS-1), *anomaly detection* (AL-2:ADS-2) and *third-party detection* (AL-2:ADS-3). Pattern detection mechanisms work by monitoring communication activity to match against a database of known malicious attack patterns. Whereas with anomaly detection, the communication activities are monitored for deviations against known normal behaviours. With third party detection, the victim receives notification of an attack from a third party.

The second major DDoS defence classification criterion of Figure 4, *deployment location*, is divided into three categories, *victim network* (DL-1), *intermediate network* (DL-2) and *source network* (DL-3). These classification locations appear to be based on where the mitigative actions of the defence are executed and not necessarily where the attack is detected. Examples given for the victim network category include resource accounting and protocol security defence mechanisms. Examples given for the intermediate network include pushback and traceback defence mechanisms. Source network defences, however, are deployed on the networks of client hosts with the purpose of preventing potential clients from generating a DDoS attack.

Cooperation degree (CD) is the last major classification criterion considered in the DDoS defence taxonomy by Mirkovic and Reiher. CD refers to the amount of cooperation involved or required in the defence mechanism in question. The criterion consists of three categories: *autonomous* (CD-1), *cooperative* (CD-2), *interdependent* (CD-3). Autonomous mechanisms require no cooperation outside of the victim network where they are deployed. Even if a defence mechanism operates in a distributed manner, it is considered autonomous if it can be completely deployed in the victim network. Examples of this include intrusion detection systems.

Cooperative mechanisms, despite being able to operate completely autonomously, are able to cooperate with entities outside of the victim network. Interdependent mechanisms are unable to operate autonomously and require distributed deployment or cooperation from other entities. Examples of these are traceback mechanisms.

Other taxonomies of DDoS defence mechanisms exist [9], [74], [75], and have been produced in response to the evolving threat landscape.

In [75], Keshariya and Foukia present a DDoS defence framework illustrated in Figure 5.

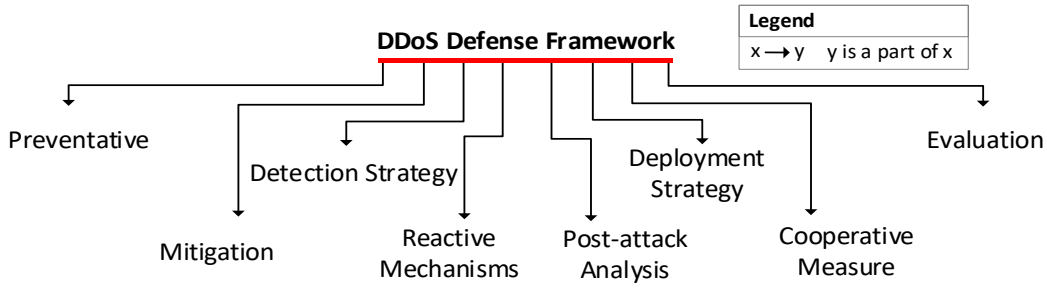


Figure 5: Excerpt of a DDoS defence framework [75]

The first seven nodes of the illustrated framework hierarchy, act as criteria for the classification of defences. The last node, *Evaluation*, encompasses the goals of evaluating a DDoS defence, which Keshariya and Foukia identify as: accuracy, effectiveness, transparency, compatibility, performance, vulnerability, deployability and scalability. Other than brief descriptions, of what each evaluation goal is, the paper doesn't delve into the steps required in achieving said goals.

Beginning form the left of Figure 5, *preventative*, refers to defence measures that are applicable in the absence of attack. Preventative defences are divided further into three subclasses: *zombie prevention*, *protocol* security and *system security*. Where each of these classes deals with ways of eliminating various vulnerabilities that make DDoS possible; such as securing connected hosts, ameliorating protocol design and "securing resources like buffers, file descriptors address space, disk space, CPU cycles, and bandwidth".

The next node, *mitigation measures*, describes defences that regulate the probability of attack. Keshariya and Foukia list the following as mitigation measures: resource accounting, resource multiplication, resource pricing, DoS aware algorithms, traffic flow monitoring, traffic volume monitoring, source IP address monitoring, the monitoring of other features, IP hopping, load balancing, TCP-migration, and mutable services. DoS aware algorithms are algorithms that are deployed in an operating system and reduce the effects of a DDoS attack. An example of one is a scheme, to drop all half-open connections.

Detection strategy possesses two categories, *pattern detection* and *anomaly detection*. Pattern detection mechanisms, alert to the presence of an attack when a known attack-pattern is

detected whereas anomaly detection mechanisms are triggered by the presence of abnormal patterns.

Keshariya and Foukia describe *reactive mechanisms*, as those that cease the impact of an attack during attack. Nine sub-classes of reactive mechanisms are given as: *filtering*, *rate limiting*, *reconfiguration*, *active traceback*, *agent identification*, *data recording*, *alert/reporting*, *deflection* (honeypots) and *congestion control*. Reconfiguration refers to the dynamic reallocation of resources during attack to bolster the capacity of the victim.

Keshariya and Foukia define *post-attack analysis* as the process of analysing data recorded during an attack to help update defences. Examples of post traffic analysis include forensics of event and system logs, passive traceback, traffic pattern analysis, attacker identification, attack reconstruction and, lastly, system updates to implement findings.

The *deployment strategy* classification is divided into three subclass areas: *victim-end defence*, *source-end defence* and *distributed defence*. This classification node describes where the defensive activity occurs. Source-end defences act to stop an attack from leaving an attack source, such as a device or a network. Distributed defences are those that employ different collaborative agents that work together to mitigate an attack.

Cooperative degree possesses seven non-mutually exclusive sub-classifications that describe the degree of cooperation that a defence mechanism employs. These are: autonomous, intermediate, distributed, distance-based, global and interdependent. Interdependent defences require deployment at multiple locations or rely on other mechanisms for detection or response.

2.1.2.2 Malice Distinction

At the heart of reactive DDoS defence is the task of discrimination. Whether considering the phases of reaction, detection or prevention (as defined in [3]), the malicious must be distinguished from the benign, the normal from the abnormal and the acceptable from the unacceptable. There exists a multitude of properties possessed or exhibited by communicating entities that can be examined to distinguish between malicious and legitimate traffic or resource consumption. This is the process of malice distinction and is used to identify malicious: activity, flows, actors, and resource consumption.

These properties include attributes of sent data and the behaviour and capabilities of the sending entities. Their quantification produces ranges of values that can be used to distinguish between malicious and legitimate traffic, if the quantified property-values of the malicious and legitimate entities lie within different sections of the scale. However, this is not always the case.

Figure 6 illustrates probability distributions of malicious and legitimate entities over a metric of distinction, which is any quantifiable attribute, behaviour or capability that a defence architecture could employ to distinguish between malicious and legitimate.

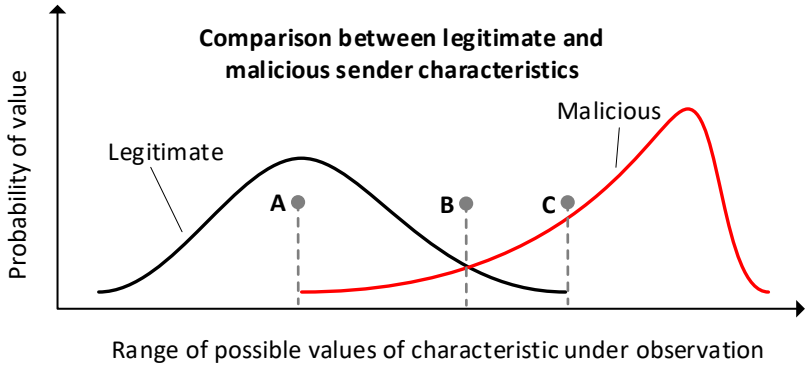


Figure 6: Graph showing the how characteristics of legitimate and malicious senders may vary over the range of values of that characteristic

As can be seen by the example probability distribution plot of Figure 6, values across sections of the metric scale may be shared by both malicious and legitimate traffic (from points A to C). This would cause uncertainty in decisions about malice made over that range of values, leading to false positives, if legitimate traffic is incorrectly identified as malicious, and false negatives, if malicious traffic is incorrectly treated as legitimate.

Nevertheless, for all incoming traffic, a decision *is* made and at the point of equal probability (illustrated in Figure 6) the likelihood of producing a false positive and a false negative is the same (assuming a normalized illustration of both curves). This raises the important question of which is preferable. Additionally, if minimizing false positives is preferable, then at what point along the scale of the metric of distinction does the risk of producing a false positive become acceptable? These are important questions for designers of discriminative defence architectures to consider.

Using metrics of distinction to identify malicious traffic, requires knowledge of how legitimate traffic varies across said metric. This knowledge is important in the production of realistic environments in which to test DDoS defences for evaluation.

2.1.3 Discussion

2.1.3.1 Attack

In a DDoS attack taxonomy, there is merit if attacks are grouped only by features that manifest during the attack itself and not by aspects that don't, as is the case in the taxonomy presented in [57]. This is because for practical purposes, one should be able to classify an attack without dedicating additional resources to attempt to determine, for example, the details of how the botnet that launched the attack was built. The impact of the attack is independent of such information and thus, we argue, should be irrelevant to its classification.

Building on the DDoS attack classification structure presented in [30], which considered the targeted resource and whether it was targeted directly, we propose the following aspects for classification:

- The resource targeted
- The protocol used
- The destination directness
- The path directness

Notable about this simple taxonomy is that each aspect is applicable to all DDoS attacks but still provides relevant information able to meaningfully distinguish between types of attacks. The *resources targeted* aspect refers to the resources the attacker attempted to excessively consume and contains the following classes: network bandwidth, memory, and CPU cycles. The *protocol used* aspect specifies which protocol the attacker leveraged for the attack. All connectivity Internet protocols may be considered classes of this category; however, the main groups are whether it was IPv4 or IPv6 traffic and whether it was UDP, TCP or ICMP traffic. The *destination directness* of the attack refers to whether the destination of the attack was direct or indirect, signifying whether traffic was sent to the host of the victim service directly or, conversely, whether it was sent indirectly to another destination of resources (e.g. network links) upon which the victim service depends. The two classes in this category are direct or indirect. *Path directness* refers to whether the attacker sends reflected traffic or not. The two classes in this category are reflected and un-reflected. Since path and destination directness are independent of each other, any combination of the two sets of classes could represent an attack. For example, an indirect attack could be launched using un-reflected packets.

The astute reader may question how such a taxonomy can accommodate hybrid attacks, which use a mixture of methods within the same attack. We address this through an *attack strategy heat map* (ASHM) illustrated in Figure 7.

As shown, the type of DDoS attack can be depicted within the formulated taxonomy. Attacks leveraging different strategies are accommodated by illustrating the different strategies used in the appropriate areas of the map. The shading in the map is indicative of the proportion of the overall attack that the particular technique in question contributed, where the brighter the shading the greater the contribution. The exact percentage contribution is also labelled for clarity. There are several possible ways to calculate this percentage contribution for each method used in the attack. One could base it on: the number of bots used, the percentage increase in consumption of the targeted resource, the data rate sent by the attacking botnet, or the data rate received at the attack destination. Using the percentage increase in consumption would be the most informative since it would provide some measure of the impact of the attack. However, we reject this method since it would corrupt attack classifications by making them dependent on the victim and the variable state of the legitimate traffic at the time. The

number of bots would be difficult to determine, since source-address spoofing may be used, and using the number of bots as a basis would present added complications should the same bot be used for different attack methods within the same attack. The rate of data sent by the attacker would be difficult to ascertain since the attack is distributed and some attack packets may be dropped on the way. We settle for the rate of data inbound (RBI) to its attack destination, as the basis for calculating the percentages of each attack method contribution, since it can be more easily measured and is independent of the nominal attack victim and legitimate traffic. The RBI includes the data of packets dropped by legitimate defences, to the extent that it is known. For attacks that use amplification, the RBI considers the data rate of the attack traffic *after* it is reflected and not before.

		Direct		Indirect	
		Reflected	Un-reflected	Reflected	Un-reflected
Targeted Resource	Bandwidth			IPv4:UDP:DNS 70%	
	Memory		IPv4:TCP:HTTP 30%		
	CPU Cycles				

Figure 7: Attack Strategy Heat Map depicting example attack

In cases were multiple protocols are used for the same DDoS method, the protocols can be added, along with their percentage mix to the square. For example in the above example attack, added to the area representing an indirect reflected bandwidth attack, could be the protocol IPv4:ICMP, with the appropriate percentage value and corresponding shading. However, the sum of the percentages of all shaded areas must equal 100%.

2.1.3.2 Defence

The DDoS defence taxonomies presented, propose classes of DDoS defences that exclude other DDoS mitigation strategies from accommodation in the classification schemes. We propose a high-level DDoS mitigation taxonomy in which the prior defence taxonomies could be placed. Our proposition is illustrated in Figure 8 and the prior taxonomies presented would be extensions of the *alleviation* class of DDoS mitigation strategies.

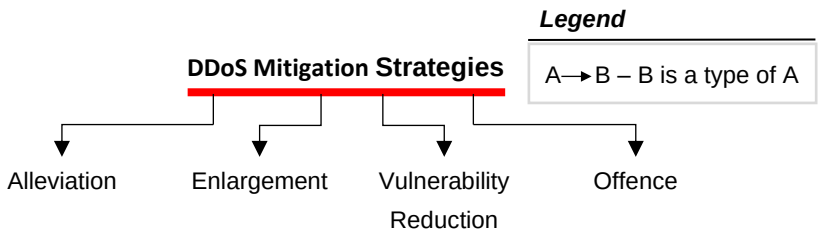


Figure 8: A high-level DDoS mitigation taxonomy

Alleviation reduces the amount of victim (and intermediary) resource consumption caused by the activities of malicious actors. This involves distinguishing between malicious and legitimate activity, by monitoring packet attributes, sender behaviour or sender capabilities, and thwarting such activity through means such as dropping packets and denying requests. To be explicit, the mitigation mechanisms of alleviation DDoS defences are designed to initiate after an attack has already begun, which excludes mechanisms that stop attacks before the begin. In this regard, the beginning of an attack is defined by when the attacker initiates the attack and not when the alleviation defence detects it. Therefore, malicious resource request are handled by such defences after they are formulated and not before.

Enlargement defences increase the capacity of the victim to service requests. This encompasses efficiency increases in both hardware and software, as well as resource capacity enlargements. Using faster disk drives and purchasing larger servers are examples of these. Additionally, increasing the connection data rate or *degree centrality* of victim nodes also falls under the category of enlargement defences.

Vulnerability reduction defence techniques limit exposure to DDoS attacks by minimizing attack-avenues. This involves, securing connected devices to prevent infection and assimilation into botnets, removing bugs in victim software that can be exploited during attack, geographically distributing servers to reduce single points of failure, and applying secure configurations and settings such as blocking unnecessary ports, and specifying hard limits on connection timeouts and the number and type of connections supported by a victim server.

Offence involves the targeting of attackers and their assets to prevent or thwart attacks. This can occur before an attach is launched, during an attack, or after an attack has ended (i.e. before the next attack is launched). If one considers attackers as ultimately human agents utilizing processes, systems and devices, the scope of possible defence by offence strategies is vast. Examples include, thwarting command and control centers used by botnets for communication, introducing deterrence measures and targeting botnets by removing bots from individual machines using antivirus software or otherwise.

2.2 Related Work

2.2.1 Accountability-based DDoS Defence

2.2.1.1 Accountability Mechanisms

Since accountability refers to the holding of an entity to account for its actions, accountability mechanisms form the basis of reputation-based DDoS defence architectures. This section discusses various accountability mechanisms designed to be incrementally deployable and compatible with today's Internet architecture. It is important to note that although numerous futuristic accountability mechanisms have been documented (e.g. [76], [77]), their

incompatibility with incremental deployment, and today's Internet, renders them unsuitable to the goals of this project; which seeks to satisfy adoptability in today's Internet.

Passport [78] is an anti-spoofing mechanism operating at the AS level, that prevents malicious AS's from passing off forged packets to participant AS's as if they were from another participant AS. Passport utilizes asymmetric cryptography to generate and maintain secrets between AS's that are used in the MAC's appended to packet headers originating from a Passport compliant AS. The source AS router leverages BGP [79] to obtain the path information of the destination address. The source AS then appends one MAC for each AS on the source-destination path using the respective keys it shares between them. On traversal to its destination each MAC is verified and (the MAC is) discarded by its corresponding AS – to prevent cryptanalysis. If the MAC fails verification the packet is dropped. If an AS receives a packet purporting to be from a passport participating AS, yet that packet possesses no appended Passport, that packet is dropped. In this way Passport prevents inter-AS spoofing between participating AS's.

However, Passport, unlike DiDoS, does not consider any mechanisms that combats intra-AS spoofing within AS's and proposes no mechanisms to combat it. Evidence is provided to show that the lightweight approach of verifying packets using MAC's, with keys obtained with asymmetric cryptography, is feasible. However, passport provides no mechanisms for maintaining and using sender reputations, which is a key component of our proposition. Passport is also ineffective against flooding attacks from sources that do not spoof IP addresses belonging to hosts lying outside their own autonomous systems. This is because in Passport the IP-address spoofing detection occurs *between* autonomous systems only, and not *within*. DiDoS on the other hand leverages the granular cryptographic mechanisms of the Hierarchical Identification Scheme to implement intra-AS IP spoofing mitigation.

Forwarding Accountability for Internet Reputability (FAIR) is an architecture proposed in [80] that addresses the incentives for Autonomous Systems (AS's) to regulate their traffic by the introduction of automated punitive action against misbehaving source AS's. Misbehaviour is defined as the violation of destination AS policies by source AS's. These policies are characterized by committed information rate (CIR) and committed burst size (CBS) agreements that are specified beforehand to the source AS by the destination AS. The punitive action involves the marking of traffic sent from violating AS's by a suspicion bit, which enables transit and destination AS's to deprioritize said traffic. The key feature of FAIR is the ability for destination AS's to cryptographically prove misbehaviour to sources AS's by storing and returning offending packet headers. The returned packet-headers include message authentication codes (MAC) that enable the source AS to verify that it indeed sent the traffic. However, the input fields to the MAC include factors that enable the source to verify policy violation only, rather than additionally verify packet-provenance, which is a crucial factor in enabling the source AS to apply corrective measures. Although the architecture implements a

"suspicion" bit to mark suspicious traffic, the basis upon which suspicious traffic is marked does not distinguish between DDoS-attack and legitimate traffic, but rather only between traffic forwarded by AS's that are deemed by a destination AS to have violated sending policy agreements. In this way, the architecture is fundamentally prone to false positives and negatives as the circumstances required for policy violation (CIR and CBS violations between AS's) bear weak correlation with individual DDoS-attack packets. For example, under this method, no DDoS attack traffic from a source AS will be marked as suspicious if the source AS never violates destination AS sending policies, which is possible if the source AS applies traffic policing to fulfil policy agreements. Similarly, all legitimate packets from AS's that violate policy agreements will be marked as suspicious regardless of their involvement in any DDoS attack. In contrast, DiDoS leverages metrics, to distinguish between malicious and benign traffic, that are strongly correlated to the attacker behaviour and the denial-of-service effect (sections 3.2 and 3.4).

The paper *Building Accountability into the Future Internet* [81] proposes a comprehensive accountability architecture consisting of three layers that leverage packet integrity, server authorisation and sender reputation to combat spoofing, thwart unwanted traffic and hold senders to account. The first layer encompasses the attachment of *source signatures* to each packet that a host creates and sends. These signatures enable packets to be verified as originating from that host *using trapdoor hash functions*. Provers (the senders of packets) use a secret key TK to generate a signature *r*, such that the hash of the message *m* with *r,* matches the concatenation of a verification key V and the sequence number of the packet SEQ_p (i.e. $h(m,r) = V : SEQ_p$). The verifier, defined as any receiver of the packet, then checks the validity of the packet by computing the hash of the variables on both sides of the aforementioned equation to ensure they match. Note that this process requires the sender to have already disclosed at hash key HK and a verification token V with all receivers of the packet.

The second layer involves *packet tickets* that are issued by servers to clients and act as capabilities that enable servers to recognize previous clients and deny or grant them access for specified periods of time. The purpose of these capabilities is to enable the elimination of DDoS attacks closer to the source by providing a mechanism that can indicate to routers close to the source that a destination has allowed communication from that client. The third aspect of the accountability architecture proposed is a reputation system that can be used by servers to prioritize the issuance of ticket requests based on client reputations. The architecture proposes local reputation centers operating within AS's that collect reports on clients from servers within that AS. These local reputation centers also peer with each other to propagate reputation scores.

It is not clear how unwanted packets can be filtered close to their source based on server capabilities (server authorisation tickets attached to sender packets). If packets without

attached capabilities are dropped, then nodes compliant to this architecture would hinder legacy Internet communications and hence would make the architecture incompatible with partial or incremental deployment. In contrast, DiDoS is designed to be compatible with legacy infrastructure by not summarily dropping legacy traffic and by specifying the DiDoS header in a way that allows it to be ignored by legacy devices (section 3.4.5). Furthermore, in [81], since ticket requests are prioritized by the architecture, how the architecture responds to malicious ticket-request floods remains an unresolved question. This source of vulnerability is not present in DiDoS which prioritizes traffic based on merited reputations of the traffic sender, and not the kind of traffic itself. Thus, senders of historical poor behaviour cannot have their traffic prioritized, unlike in [81], which prioritizes all ticket request.

2.2.1.2 Reputation-based Defences

In [34] the authors propose a ticket-based mechanism, that grants tickets to prospective clients, based on their accrued reputation. The assigned tickets encapsulate a credit score commensurate to the client's overall traffic contribution during an attack, which is assumed to be higher for malicious clients. Thus, historic traffic contribution is the metric of distinction of the defence. The main contribution of the work involves the proposition of more ticket reputation levels, which facilitate the differentiation between clients to a finer granularity. Furthermore, the tickets are entirely generated at the destination, which reduces defence operational cost and makes the tickets route-independent. Under the parameters tested, the defence is able to reduce the percentage of attack-traffic reaching the victim to 5% (down from 100%) in less than one minute after the beginning of the attack. Nevertheless, the defence does not address the issue posed by multiple clients behind the same public IP address, as is the case with NAT-shielded devices. Since the defence uses the received source IP address as the identifier against which client reputations are maintained, it is unable to distinguish between the reputations of clients that share the same public IP address. This causes the reputations of legitimate clients to be tarnished by those of attacking clients if they utilize the same public IP address. This issue is addressed by the Hierarchical Identification Scheme of our DiDoS proposition, which uses finer granularity reputee identifiers identify and authenticate clients.

Additionally, in [34], since the reputations of clients are held at the victim, scalability issues arise with increasing number of clients, as the victim must store the reputations of each client. Furthermore, multiple victims would independently maintain the reputation of every client, which presents the inefficiency of overlapping records in the cases where clients communicate with multiple (potential victim) destinations. DiDoS avoids this problem via the Collaborative Reputation System, in which the reputations of senders are distributed in the reputors of all DiDoS compliant networks.

In [82], a reputation-based defence called Trust Management Helmet (TMH) that combats session flooding attacks is proposed. TMH identifies clients by a combination of the client IP

address and a unique identifier assigned by the victim. This helps to address the issue of distinguishing between clients behind a NAT device. The main metric of distinction used by TMH to distinguish between malicious and legitimate clients is the interval between accesses of a client to the server at various levels of utilized server access-capacity. TMH additionally provides support for collaboration between servers to share client blacklists and trust scores. However, how this mechanism fairs against malicious servers or Sybil attacks is not fully explored. Additionally, TMH provides no mechanisms for the elimination of attack traffic closer to the attack source. Therefore, regardless of the robustness of the claim of the 'lightweight-ness' of the defence enabling the victim to process received licenses at the maximum rate at which the network can send them, the defence is still susceptible to flooding attacks that produce contention at routers further upstream from the (victim) destination. On the contrary, the Distributed Mitigation Mechanism of the DiDoS architecture enables the mitigation of traffic at any point of contention in the network.

In the paper *TrustGuard: A Flow-level Reputation-based DDoS Defense System* [47], a flow-level reputation-based DDoS defence system that combines macro and micro level features to combat attacks is proposed. TrustGuard is deployed at the border router of an organisation and monitors the macro and micro-level features of flows to detect and mitigate attacks. Of the possible, macro-level features mentioned, such as flow symmetry features, packet-to-flow ratios and the ratio of SYN to SYN/ACK, TrustGuard adopts packet size for attack-detection on the basis that a higher concentration of packets at lower sizes is indicative of an attack. Once an attack is detected at the macro level, TrustGuard assesses traffic based on its chosen *micro-level anomaly indicator*, which is the correlation of packet-sizes of individual flows. TrustGuard uses this feature to build short-term reputations for each flow via credit accumulation on the basis that malicious flows are more highly correlated such that they are hypothesized to accrue less credit than legitimate ones. A probabilistic drop filter is implemented by the defence to drop more packets of flows that are more likely to be malicious (possessing low credit) and less packets of flows that are more likely to be benign (possessing high credit).

Many of the assumptions upon which TrustGuard is designed are based on two historic datasets depicting particular scenarios: the *Anonymized 2008 Internet Traces Dataset* and the *CAIDA DDoS Attack 2007 Dataset*. These include the assumptions of the basis upon which attacks are detected on a macro-level and filtered at the micro-level. The former being that the presence of small payloads is indicative of an attack and the latter being that attack flows are more likely to be correlated in packet-size than legitimate flows. The paper assumes that an attacker of the future would not vary its attack packet size due to the cost to its attack strength of doing so. However, this cost is not specified, and neither are the potential benefits to an attacker in doing so in the face of such a defence as TrustGuard. Furthermore, the

premise is contradicted in [30], which states that network bandwidth consuming attacks are "generated by crafting network traffic packets that hold the maximum possible payload data".

The paper reports TrustGuard to have identified 98% of malicious traffic, with a false positive rate of 1%, under empirical evaluation. However, due to the same data, from which assumptions were derived, being used for testing, it is not clear to what extent overfitting occurred in the results.

The TrustGuard mechanism is compatible with being distributed at any point in the network that can receive packets from multiple sources with different IP addresses. However, although TrustGuard can be distributed, unlike DiDoS, it is not collaborative. Another big distinction between DiDoS and TrustGuard is the metric used to distinguish between malicious and benign traffic. TrustGuard uses packet size, which is cheap for an attacker to change and thus evade the defence, whereas DiDoS uses reputations built on attack involvement frequency and in-attack volume, which are costly for the attacker to evade.

2.2.2 Existing Commercial Technologies

This section explores the commercial state-of-the-art in DDoS defence technologies, compares them, and, at the end, summarizes their defensive mechanisms and deficiencies, with respect to our contribution.

Commercial DDoS defensive options consists of dedicated DDoS protection providers, Cloud Service Providers (CSPs) (including Content Delivery Network [CDNs] providers), and Internet Service Providers (ISPs). Despite being popular in industry, very few [83]CDN-based DDoS defences were found in the academic literature.

Akamai is one of the leading DDoS defence providers and was recognized by Forrester as a leader in web application firewalls [84]. Akamai's DDoS defence solution comprises three products: The App and API Protector, Prolexic, and Edge DNS [85]. The App and API Protector works by rerouting client traffic through a distributed high-capacity scrubbing center, which enables it to mitigate attacks of greater bandwidth and closer to the source. The solution leverages automation through hands-off updates and self-tuning, API discovery to prevent new vulnerabilities, and an Adaptive Security Engine which automatically learns attacks patterns and adapts to future threats.

Prolexic is a cloud-based DDoS defence solution built on a platform of 20 Tbps dedicated DDoS defence capacity. Instead of originating their prefix advertisements from their datacenters, Prolexic customers make either a border gateway protocol route advertisement change or DNS redirection (A record or CNAME record) to originate from one of many Akamai global scrubbing centers. The cleaned traffic is then directed to the customer using GRE tunnels [1], Layer 2 VLAN [4] connections, and/or VIP-to-origin back-end mapping. [86]

Edge DNS [87] provides DDoS resilient DNS services that connects the websites and applications of Akamai customers with their end users. DDoS resilience is achieved through scale and distribution and the ability leverage other Akamai services [88]. For example, Edge DNS comprises thousands of name servers deployed in over 1,000 points of presence spread over more than 150 cities, and leverages an IP Anycast model to respond to DNS queries. Despite this Akamai acknowledge the risk of regional outages from DDoS attacks which can exceed 1Tbps.[89]

AWS Shield [90], [91] is the DDoS protection solution provided by Amazon, which customers of Amazon Web Services (AWS) can utilize for free. Shield employs distributed DDoS mitigation infrastructure located at the AWS network border and AWS edge locations providing 100 Tbps of mitigation capacity [92]. At the AWS network border DDoS mitigation systems activate mitigations when signalled by Shield detection or a Shield Response Team engineer, whereas at AWS edge locations DDoS mitigation systems inspect every packet for continuous mitigation.

AWS Shield detects infrastructure attacks by monitoring capacity utilization of resources (e.g. Elastic Compute Cloud [EC2] instances, Elastic IP addresses [EIPs], Classic and Application load balancers) across multiple dimensions (e.g. port and protocol) to deploy mitigations for a specified time to live (TTL) once thresholds are exceeded. Amazon Shield Advanced minimizes false positives by raising the threshold for mitigation deployment if the health check of Shield Advanced protected application suggests the application is in good health. Additionally, DDoS mitigation deployments for Shield Advance protected resources follow an exponentially increasing TTL, which reduces the time to protect resources that are attacked frequently or in bursts. [93]

AWS Shield detects application layer attacks by identifying anomalies in monitored volume and composition of traffic when compared to historic baselines that account for timing windows in order to reduce false positives [94]. AWS Shield Advanced automatically attempts to identify the attack signature and create DDoS mitigation rules that exclusively target the attack traffic. These automatically generated rules are compared to historic application baselines in order to determine if they exclusively target the attack traffic. If so, the rules are deployed in the Web Application Firewall (WAF). However, if not, none are deployed. [95]

Table 3 Summarizes the aspects of various commercial DDoS defences including those discussed. As shown, scale and distribution are important components for all commercial providers of DDoS mitigation.

#	Commercial DDoS Defence	Mitigation Capacity	Number of PoPs	Geographical Distribution	Advertised Market Differentiators
1	Akamai [85]	20 Tbps	>1,000	>150 Cities	0-second SLA. Flexible On-Prem Options
2	AWS Shield [90]	100 Tbps	>550 [96]	>100 Cities	Inline, low-latency mitigation
3	Cloudflare [97]	228 Tbps [1]	197 [98][2]	>310 Cities	Accolades, 100% Uptime guarantee with 25x reimbursement
4	F5 Networks [99]	>12 Tbps [100]	23	<24 Cities [2]	Multilayer Protection, On-demand scalability
5	Imperva [101]	10 Tbps [102]	44 [103]	<45	SDN automated tuning
6	Azure DDoS Protection [104]	50 Tbps [105]	>190 [106]	109 Cities [106]	Always on adaptive real-time tuning. Detailed in-attack reporting.
7	7. Neustar UltraDDoS Protect [107]	>12 Tbps	14	<15	Monitoring detection and response from Neustar security experts
8	Netscout Abor DDoS Protection [108]	15 Tbps [109]	15 [109]	<15	Adaptive DDoS protection leveraging global visibility and threat intelligence
9	Radware [110]	12 Tbps	19	<20	Behaviour-based algorithm for zero-day protection
10	NexusGuard [111]	1.4 Tbps [112]	10	9 Cities	Lower total cost of ownership, Granular visibility & control through portal

Table 3: Commercial DDoS Defence Comparison Table

Understandably, for security reasons, all commercial providers analysed withheld great detail concerning the algorithms and metrics used to mitigate malicious traffic. Also, despite providing details on the mitigation capacity of their networks and scrubbing centers, these figures were not accompanied by information on the efficiency and effectiveness of these centers. For example, a scrubbing center capable of receiving 10 Tbps traffic to clean, does not necessarily have 0% false positive and false negative rates. This means that although a scrubbing center can process and filter large amount of traffic, malicious packets may still be served (false negatives) and benign traffic may still be dropped (false positives). This issue is especially exacerbated when the malicious traffic is identical to the benign traffic. This highlights one of the benefits of our contribution, DiDoS, which enables the identification of malicious traffic, despite the ability of the attacking agent to perfectly replicate the characteristics of benign traffic. DiDoS achieve this through a hierarchical chain of persistently bound reputations that are embedded into the traffic, which is described in more detail in section 3.4

[1] Cloudfare did not clearly define network capacity, nor did they clearly distinguish between DDoS mitigation capacity and network capacity, but appeared to use the term synonymously over multiple information resources.

[2] 197 is the tally of all the individual datacenter locations listed on that page including those marked as "AI inference location planned by end of 2023"

All commercial DDoS defence providers emphasize the importance of scale and distribution of scrubbing centers and points of presence. Despite this, there still exist the threat of regional outages, for attacks that may not exceed the total capacity of the defence provider but are concentrated on particular regions. For example, raw traffic destined to a DDoS-protected resource is routed to an edge datacenter (as is the case for Cloudflare) where filtering begins. However, this traffic must pass between multiple router hops before it even reaches the network of the network of the DDoS protection provider. It is at this point that significant vulnerability exists for regional volumetric attacks, since said routers will drop excess packets that may be benign. Our contribution, DiDoS, extends this through its distributed mitigation mechanism (described in section 3.4.4). Essentially, DiDoS extends the mitigation points of the commercial state by enabling routers that are closer to the source to make intelligent decisions on which packets to drop. Importantly, these decisions are enabled without the requirement of additional wider context of the attack itself, thus eliminating the time-to-mitigate.

2.2.3 Collaborative Architectures and Core-based DDoS Defences

Collaborative DDoS defences are those that leverage the cooperation of distributed otherwise-independent components in attempt to achieve some advantage for the purpose of mitigating DDoS defences. Numerous collaborative architectures have been proposed [40]–[44], [124], [139]–[143], and there is overlap with other domains of DDoS defence, such as core-based, SDN-based and ML-based. In this section we assess one such architecture and compare it to our work. We then consider core-based DDoS defences.

Bungee-ML, proposed in [124] is a technique that leverages the fast-processing speed of the data plane and the high-intelligence-capability of the control plane to detect and mitigate distributed denial of service (DDoS) attacks in networks. Bungee-ML achieves this by a process called *vertical cooperation*, in which traffic is continuously monitoring at the data plane to detect traffic anomalies and supplies machine learning (ML) models, that run in the control plane, with inputs that are used to perform deep traffic analysis. The findings of this analysis are applied to progressively push back malicious traffic farther away from the victim via process called *horizontal cooperation* between forwarding devices. This is achieved by the controller progressively configuring the rules of forwarding devices, in response to the attack. Bungee-ML each forwarding devices uses an application-specific integrated circuit (ASIC) to continuously perform entropy analysis. Additionally, a general-purpose CPU, required on each forwarding device, is used to store local statistics and compare them with global statistics during an attack – as well as maintain a suspect list. Suspicious packets are diverted to the control plane, which performs a packet feature analysis to determine whether the source address is removed from the suspect list or added to the blacklist. This decision is passed on to the data plane, where packets from blacklist sources are then dropped. The detecting forwarding device, since it is likely close to the victim, performs push back in which

it shares the blacklist source addresses progressively upstream, so they can be filtered earlier on.

The Bungee-ML solution provides an approach to collaborative DDoS mitigation with a very useful push-back mechanism that is able to instruct nodes further away from the victim to begin mitigation despite there being little or no DoS impact at those nodes. Such a push-back mechanism would be complimentary to our DiDoS proposition; however, the value is in the compatibility with downstream nodes. A limitation of Bungee-ML is that it is collaborative within a single administrative domain, and thus, does not address the inter-domain collaboration challenges required of a real-world solution. Since deep packet analysis is required at the control plane for every suspicious packet, this is a vulnerability of a centralized system which could cause DoS of the control plane by forcing more inspections that the control plane can handle. This is a problem especially for a control plane servicing many (thousands) of forwarding devices. Their experiment only tested for the scenario of 4 forwarding nodes.

Furthermore, the metric that Bungee-ML uses to distinguish between malicious and benign packets is entropy, which is shown to be effective, but Bungee-ML was only tested against a single static historical dataset. The viability of Bungee-ML is not shown for the case where an attacker is able to mask its entropy by for example mimicking the behaviour of legitimate senders and increasing the number of attacking agents slowly. Therefore, the reliability of their entropy-based detection method remains an open question. Due to the continuous monitoring of entropy required at every forwarding node, Bungee-ML requires greater processing power than traditional networks, SDN networks, and the DiDoS proposition.

Core-based DDoS defences are DDoS defences that leverage the network core to detect or mitigate DDoS attacks. Numerous core-based DDoS defences have been proposed [134]–[139].

In [124], Horde, an autonomous DDoS defence framework to mitigate against dynamic Infrastructural DDoS attacks (I-DDoS) is proposed. Horde employs reinforcement learning agents that continuously learn attack strategies and predict attack actions. These agents also learn the optimal composition of mitigation strategies such as filtering, rate limiting and rerouting flows. Horde inputs information such as link state and leverages reinforcement learning on multiple fronts making the precise way in which it distinguishes between malicious and benign traffic obscure.

However, when it comes to mitigation, Horde is deployed on upstream service provider or ISP networks whereas the deployment of DiDoS extends all the way to the client, enabling the mitigation of DDoS attacks closer to their source – reducing upstream congestion. This distributed design additionally allows DiDoS to be able to mitigate against larger volumetric attacks since congestion occurring outside of the upstream ISP can be mitigated.

2.2.4 Other DDoS Defence Options

2.2.4.1 SDN-based

Software Defined Networking (SDN) is a networking paradigm that decouples the control plane and data plane of networked devices, such that the logic of the entire network is dictated by a, usually centralized, SDN controller [125]. Since the onset of the SDN paradigm much of the DDoS research carried out has explored the use of SDN to mitigate DDoS attacks [125]–[129], but also to address the DDoS vulnerabilities inherent in SDN architectures [125], [130]–[135]. This section explores a few of these propositions that leverage SDN to mitigate against DDoS and compares them to our DiDoS proposition.

RADAR[1] proposed in [127] leverages commercial off-the-shelf (COTS) SDN switches to provide mitigation against flooding-based DDoS attacks. Unlike other defences, RADAR is able to work on COTS SDN switches because it instructs switches, via the controller, on how to collect partial information.

It achieves this using three interoperating architectural components: the *RADAR Collector*, the *RADAR Detector* and the *RADAR Locator*. The Collector receives collection rules from the Detector and instructs switches to collect particular flows of interest from network-wide traffic. The Detector receives statistics from the Collector and performs correlation analysis of suspicious flows to create collection rules and detect attacks. For example, by analysing the rate of link utilization changes and the correlation between links and paths composed of these links, RADAR is able to crossfire attacks if the following exceed various thresholds:

- the total number of link utilization changes,
- the number of congested links in each congested path
- the congestion duration of a path equals the total congestion duration of links composing said path

The Locator adaptatively generates collection rules using signals received from the Detector. It performs correlation analysis between individual suspicious flows and aggregate flows at the victim, in order to locate and throttle attack traffic. The attack traffic is mitigated according to an iterative process by the Locator that identifies attacking nodes using statistics to check if the leaf nodes generate flows that exhibit the same pattern of the attack.

The authors identify the risk that benign traffic is dropped by RADAR if it matches the features of attack traffic, and they the address this by a port-based max-min fairness technique. This technique essentially rate limits flows on a per OpenFlow port basis, with the logic that the excess dropped traffic is more likely to be attack-traffic.

The RADAR scheme is able to address sophisticated attacks like crossfire. However, although its able detect attacks in real time, it relies on statistical correlative attributes to distinguish

[1] Reinforcing Anti-DDoS Actions in Realtime

between malicious and benign flows. This is a limitation for a sophisticated attacker that is able to mimic benign flows. In comparison, DiDoS does not need to detect an attack in order to perform mitigation as the lower reputed malicious traffic is simply dropped at points of contention. Furthermore, since DiDoS monitors the reputation of malicious devices themselves, after a learning phase, it is able to distinguish between malicious and benign flows, even if the attacker successfully mimics the flows of benign devices.

In [126] an SDN-assisted method is proposed to mitigate against Slow HTTP DDoS attacks, which are difficult to detect in a network due to traffic patterns being similar to that of legitimate devices. Instead of the defence being hosted at the webserver, [126] proposes a Slow HTTP DDoS Defense Application (SHDA) that runs on the SDN controller. The webserver sends suspicious requests to the SHDA which verifies if its malicious based on the time out values. The SHDA then works with the SDN controller to update rules on the control plane nodes to mitigate against subsequent request from the sender. Compared to the DiDoS architecture, this limited in scope, in that it mitigates against only a specific kind of attack, and operates on the application layer, which DiDoS does not. Furthermore, the scalability of this defence is in question for an SDN environment that hosts many servers that are under attack, since the suspicious request sent to the controller would become a form of attack on the controller itself. Furthermore, this attack is vulnerable to malicious webservers, that may feign attack in order to flood the SHDA rendering ineffective.

In summary because of the good programmability of the data plane, SDN is a good tool for DDoS mitigation, since attack-mitigation rules can be propagated to any node. Additionally, as shown in [127] the data plane nodes can also be leveraged for intelligent attack detection due to its ability to dynamically adopt monitoring and collection rules specified by the controller. However, although SDN has seen significant attention and even implementation since its inception, most of the Internet is not SDN and whether that will quickly change is uncertain, making most SDN DDoS defences limited in present value or an uncertain long-term investment.

SDN networks themselves could be fit into the DiDoS architecture as reputation domains, where the controller would be the reputation management center and would receive attack feedback reports and specify the reputation levels of reputees and instruct the data plane nodes on which reputations levels to assign to which ports. However, a challenge is whether COTS SDN switches are able to perform MAC verification at line rate or whether a workaround would be needed like increasing processing power in [124].

2.2.4.2 ML-based including Graph-based

Machine Learning (ML) DDoS defences are those that leverage machine learning techniques to facilitate DDoS defence, in ways that could include, detecting attacks, identifying malicious packets or optimizing DDoS attack response. Whereas Graph-based DDoS defences are a type

of ML-based DDoS defence that leverage graph neural networks (GNNs) which is designed to process data that can be represented as graphs. The literature includes numerous ML DDoS defence propositions [13], [136]–[143], many of which apply machine learning for attack detection.

A DDoS defence based on a Spatial-Temporal Graph Convolutional Network (ST-GCN) is presented in [144]. Unlike typical applications of ML that merely detect whether an attack is occurring, this proposition also detects the paths from which the attack is occurring. This is achieved by mapping the network nodes into a graph and leveraging the data plane to feed samples of the current network status into the ST-GCN, thus allowing it to "sense" attacks. In this way, ST-GCN finds both temporal and spatial features of network state and uses that to find the attack path.

The proposition mitigates against DDoS attacks using an enhanced whitelist and precise dropping strategy. The enhanced whitelist is achieved by adding the IP addresses of commonly used business hosts to whitelists that are propagated to the rest of the network. To reduce the impact of an attacker spoofing the IP addresses of these trusted hosts, an additional dropping rule is added that allows flows to the victim to be dropped if the traffic is unidirectional. The effectiveness of this strategy is predicated on the assumption that attack traffic is unidirectional.

The precise dropping strategy leverages the insight, provided by the ST-GCN, of which links the attack traffic originates from, in order to reduce the collateral damage on benign traffic by focussing mitigation on attack-sourced links only.

This approach can be effective in theory, but in practice, DDoS attack traffic converges from multiple distributed sources until it reaches the single destination. This means that the probability that attack traffic will not traverse a public link connected to the destination is almost zero. Consider a server with 2 links connecting it to the Internet, being attacked by 1000 distributed malicious devices. The probability that all of the attack traffic will traverse through just one of those links is very low ($\frac{1}{2^{1000}}$), especially if both links are equally used by distributed legitimate clients.

Other than the requirement of SDN infrastructure for this DDoS defence proposition to work, the key difference between this defence and our DiDoS architecture is the metrics used to distinguish between malicious and benign traffic. They use entropy, link provenance and whitelists all of which can be gamed by an intelligent attacker. For example, IP addresses of sources that have prior benign communications with business hosts are added to the whitelist. This avenue can be exploited by an attacker that performs the activities needed for it to join the network-wide whitelist. DiDoS, on the other hand, deliberately provides no active means for an attacking device to boost its own reputation.

2.2.5 DDoS Defence Evaluation and Benchmarking

2.2.5.1 Empirical

In [57], a lack of any "benchmark suite of attack scenarios" or established evaluation methodologies that would enable the comparison between defence systems, was highlighted as one of the several serious factors that hinder DDoS defence research. Since then, numerous efforts have been made in the research community to address the problem including a Defence Advanced Research Projects Agency (DARPA) sponsored project, from which the paper [145] was produced.

The paper presents some contributions of the project towards the development of a common evaluation methodology for DDoS defences. The approach taken by the project involved the division of the task into three output components: a *benchmark suite*, a set of *performance metrics* and a *testing methodology*.

The purpose of the benchmark suite was for the specification of everything needed to create an adequate set of DDoS scenarios in a testbed environment. Automated tools were employed to gather samples of attack and normal user traffic, and network topology examples. The purpose behind the automation was to facilitate the adaptation of the benchmark suite to future internet trends.

The paper considers performance metrics as ways to assess the effectiveness, cost and security of a defence. Effectiveness indicates whether or not a defence removes the denial-of-service effect, cost refers to the financial deployment and operational costs of the defence, and security captures a defence's susceptibility to insider and outsider threats. The main aspect of performance focussed on by the paper is effectiveness, whereas cost and security are only briefly described with acknowledgement of the unique challenges in objectively assessing a defence mechanism on these aspects.

Thus, objective means of measuring the costs or security of a DDoS defence remains an open research question.

After a brief review of several possible metrics, the paper settles on service quality as the proposed metric; where the measured variables constituting service-quality would answer the question: "did the legitimate clients receive acceptable service or not?" However, recognized, is the application-dependency of the term "acceptable", as certain conditions acceptable for one application may be unacceptable for another.

Measurement methodology is briefly discussed in the paper. It encompasses the guidelines to be used for the aggregation of multiple measured results to produce a few meaningful numbers. The rules for this process are not specified in the paper, however, the authors anticipate it to be a "challenging and controversial" endeavour.

The paper concludes with the specification of four future research directions that are likely to pose serious technical challenges: 1) the collection of sufficient topology data to construct typical test suites, 2) the understanding of how traffic, topology and resources interact and using that understanding to create comprehensive but practicable test sets, 3) the determination of success criteria for each application, and lastly 4) the definition of a concise results-aggregation strategy. All four directions remain open research questions.

The strong contribution of this paper is the consideration of transaction type as part of the metrics for defence performance evaluation. We have adopted this in our CED3 evaluation framework. However, the use of benchmarking via fixed DDoS-attack tests and environments, is a limited way to compare DDoS defences. Firstly, the benchmarks may not be able to capture the strengths or weakness of a defence, leading to unreliable conclusions. Secondly, attackers in the real-world may exceed the attack capabilities of the benchmark test scenarios, causing defences previously deemed as effective to be exposed as ineffective in actual deployment. CED3 addresses by the introduction of the metric of true effectiveness, which changes the paradigm of DDoS defence evaluation from "whether a defence withstands a particular attack" to "what would it take to overcome this defence".

In [146], 6 notable publications that evaluate DDoS defence architectures and analyses their employed evaluation methodologies for shortcomings, which are found in all. The paper discusses the goals of evaluation and then analyses approaches to testing, the metrics required and intricacies of acquiring said metrics.

Seven evaluation goals that a DDoS defence evaluation methodology should consider, are identified as assessing the:

1. **effectiveness** of the defence,
2. **collateral damage** experienced by legitimate traffic in the presence or absence of attack,
3. **delay** of the defences response to attack,
4. deployment computational **costs** – such as memory,
5. effectiveness of **partial deployment** and the scale needed to meet a certain effectiveness,
6. **scalability** of how costs vary with effectiveness, and
7. **resilience** of the defence to attacks and subversion.

However, the paper focussed primarily on the goals of effectiveness and collateral damage. Not much is said in the way of techniques for the analysis of the resilience of a DDoS defence to attacks and subversion. This area, in general, remains an open research question.

The paper explores three aspects of test scenarios that can be modelled in theory, simulation and emulation, including: legitimate traffic, attack type and topology. The potential that each aspect may have in morphing the results of experimentation in various ways is discussed in

detail and recommendations for the purpose of avoiding unrepresentative biases are made. A notable example of unrepresentative bias is the use of replay traffic from public traces in a non-congestion-responsive manner, which does not account for the dynamic responsive nature of TCP[1] to congestion.

Furthermore, in order to promote more robust defence evaluation, specific kinds of test scenarios are also recommended for each kind of defence architecture. For example, *privileged customer* approaches, where legitimate users obtain *passes* that give them prioritized access to resources are recommended to be tested in scenarios where, 1) the attacker behaves as required to obtain a pass but then turns bad, 2) attackers flood the system with pass requests, 3) attackers attempt to bypass checking.

On metrics, the paper argues that since DoS is a "subjective phenomenon perceived by human users", selecting the right metrics is a challenging problem. Five metrics are then analysed, discussing their various advantages and disadvantages. These are: loss, throughput or goodput, request/response delay, transaction duration and allocation of resources. However, two challenges with existing metrics are identified as defects that are common to all metrics. The first is that they may not directly correspond to DoS, and secondly, existing metrics do not scientifically define acceptable ranges for each, which is dependent on the application or task.

Furthermore, the paper recommends that metrics be only reported with error-bars or low and high percentile values, in order to sufficiently communicate the variability in performance. This reporting thus requires the numerous testing of a defence in a particular scenario.

Our CED3 framework adopts many of these concepts presented in [146], such as the tailored testing of each defence and the multiple testing of scenarios to produce error bars. We, however, extend the presented capabilities by adding the capability of commensurable comparison of evaluated DDoS defences. CED3 achieves this by the use of the metric of *true effectiveness* to characterize the defences (tested under the same *context*) within particular *scopes*, thus allowing their strengths and weaknesses to be commensurably highlighted.

Despite the great research effort into ways of evaluating DDoS defences, inconsistent methods are still being used to validate defence proposals. In [147], Behal and Kumar recognize this persisting diversity, and analyse different evaluation techniques. Four methods of evaluation are discussed: *mathematical models*, *simulation*, *emulation* and *real systems*; and are qualitatively compared against each other. Mathematical models refer to theoretical evaluation techniques where defences are modelled symbolically for validation. Simulation refers to the virtual implementation of a defence system in software, where it is validated. Emulation mixes elements of simulation and real-world systems such as operating systems

[1] TCP – Transport Control Protocol

and applications with virtual components such as soft routers or network links. Emulation is executed in real-time, unlike simulation which executes in virtual time. Real systems are made from real networked devices processing real traffic.

Behal and Kumar put forward a "new dimension" of evaluation that consists of the use of real datasets. The argument is that certain properties of real datasets would be "more appropriate for the accurate validation of DDoS research". Whilst this may be the case for the validation of defences in the context of specific historical scenarios, the paper does not clarify how such appropriateness would uphold in the context of new attacks.

Additionally, in [146], Mirkovic et al. explicitly warn against the oversimplification of legitimate traffic in testing DDoS defences. Specifically, that the failure to capture the dynamic nature of TCP traffic can lead to the overestimation of a defence's capability, which is a possibility when dealing with real static datasets.

Behal and Kumar, however, identify other limitations in the use of publicly available datasets for validation, such as the presence of asymmetric traffic, pseudonymised IP addresses and the brevity of captured network traces. Most of their other concerns are due to poor capture quality and missing details. However, Behal and Kumar conclude that the following properties are necessary for the ideal real dataset:

- Real source and destination IP addresses
- A wide range of random source IP Addresses
- Actual packets with valid headers
- Appropriate mixture of normal and attack traffic

Brief explanations of the above criteria can be found in the paper, however Behal and Kumar offer no explanation on the basis by which "appropriate" is determined in the mixture of normal and attack traffic, which appears to be another open research question.

Table 4 summarises the qualitative comparison, by Behal and Kumar, of four evaluation techniques against six attributes.

Attributes	**Simulation**	**Emulation**	**Real Systems**	**Real Dataset**
Fidelity	Lowest	Moderate	Highest	Highest
Repeatability	Highest	Moderate	Lowest	Highest
Programmability	Highest	Moderate	Highest	Lowest
Extensibility	Highest	Moderate	Lowest	Highest
Research Functionality	Lowest	Moderate	Highest	Highest
Abstraction	Highest	Moderate	Lowest	Lowest
Type of Network Elements	Virtual Nodes	Mixture of real and virtual nodes	Real nodes	Real nodes

Table 4: Comparison of validation techniques used for DDoS Research [147]

Fidelity refers to how well the experiment represents real networks in various topological characteristics such as number of nodes, bandwidths of links and delays, and mixture of hardware and software. Repeatability refers to the ease with which an experiment can be reproduced and repeated in the same conditions as before. Programmability deals with the ability of the experiment to implement new and customized mechanisms. Extensibility refers to the ability to topologically scale the experiment and remotely access it. Research functionality refers to the realism of the evaluation technique, which captures the social aspects. Level of abstraction describes how much complexity is included in the experiment. Lastly, type of network elements describes the types of nodes of each validation method.

The conclusion of the paper is that real datasets are the best method for the validation of DDoS defences, however it is difficult to affirm its objectivity due to the lack of arguments put forward in corroboration of the chosen (Lowest, Moderate and Highest) attribute ratings of each validation technique.

Our CED3 proposition differs from the approach of [147], by using dynamically generated traffic to test defences, thus aligning with that of [146]. Our deviation is necessary, as, to find the true effectiveness of a defence, we must consider the most powerful attacks tailored to that defence, not generic historical attack packet captures.

2.2.5.2 Qualitative

In [58], Beitollahi and Deconinck, analyse various DDoS defence mechanisms classified as *survival*, *proactive* or *reactive*. Survival techniques are things that the victim does to increase its ability to withstand attack such as enlargement of resources like bandwidth, CPU power or memory. Proactive techniques are preventative measures that mitigate attacks before they can damage the victim. In these techniques routers usually play key parts in their operation. Examples of these techniques include ingress and egress filtering, various firewalls upstream from the victim, secure overlay architectures and more. In reactive techniques, the attack response begins with detection of the attack by the victim. It is noted that most DDoS defence techniques belong to that category. Examples of this include pushback – where rate limit requests for specific downstream flows are sent upstream and implemented by compliant routers - and traceback techniques, which attempt to identify the true location of attacking agents.

Beitollahi and Deconinck discuss five questions that must be answered when analysing the design of DDoS defence methods: 1) where the attack is detected, 2) how it is detected, 3) what the response mechanism is, 4) where the response mechanism should be applied, and 5) where the control (decision) center is. For each DDoS countermeasure, the paper provides a brief description of its method of operation and then proceeds to discuss the challenges

associated with the mechanism. In this way, a qualitative analysis is performed on the selected DDoS defence mechanisms.

Assertions are made on the effectiveness of certain defence mechanisms; however, no empirical evidence is contributed by the paper. Additionally, conditions for the effectiveness of certain mechanisms are specified, such as a requirement for universal deployment of ingress and egress filtering in order to be effective, yet no adequate definition of effectiveness is provided. In short, the paper provides a theoretical analysis of defence mechanisms with qualitative arguments, and largely excludes empirical analysis with quantitative data.

Whilst enabling theoretical strengths and weaknesses to be compared, the method of analysis carried out by Beitollahi and Deconinck does not facilitate the ranking of DDoS defence mechanisms that would enable a decision maker to objectively determine which mechanisms are better. Nevertheless, the challenges discussed for each defence mechanism, provides some guidance for the development of tests as part of possible future research efforts.

A similar qualitative evaluation approach is undertaken in the paper, *Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems*, by Peng et al [59], which compares different defence techniques based on qualitative arguments.

In order to foster objectivity, our CED3 approach differs from that of [58] and [59] by prioritizing empirical means to measure meaningful differences between defences. However, when it comes to the CED3 assessment of defence adoptability, we have employed qualitative means to analyse defences, but have transformed it into *rank data* and applied weighted aggregation to combine multiple aspects into a single ordinal metric for each defence, thus allowing the defences to be compared by ranking. (See section 5.4.5 form more information)

In the paper *Vulnerability Analysis of Existing Distributed Denial of Service (DDoS) Defense Frameworks* [148], published in 2018, an assortment of 12 DDoS defence architectures are qualitatively evaluated for the purpose of identifying how to better create one. The paper presents an operational comparison of the chosen defences via a table that specifies characteristics of each defence in the three phases of *detection*, *response* and *mitigation*. Under the detection phase, sub-categories include, its timings (passive, on-time or proactive), the *parameter* monitored and the *detection metric* used. Two subcategories exist under the response phase classification table heading that describe the trace-back technique and the response-action (such as filtering, rate limiting and reconfiguration). For defences that did not employ aspects of a particular subcategory, such as defences that did not employ any trace-back techniques, the corresponding table cell was marked N/A.

The paper presents a second table that then evaluates each of the defences by identifying vulnerabilities – describing them qualitatively. Albeit a concise indicator, much of the contained statements on vulnerability are unsubstantiated and appear subjective - such as “less scalable” or “less accuracy”. Furthermore, although aspects of various defences are presented,

which can help guide further investigations, the method of evaluation does not allow for objective comparison between them, unlike CED3.

2.2.5.3 Adoptability

Adoptability refers to the ease with which a solution can be adopted into the real world Internet or target deployment environment. The concept of adoptability is a concept that is rarely discussed in the literature with direct regard to DDoS defence implementation. However, attempts at defining and evaluating adoptability exist in neighbouring areas such as improving BGP security.

The method in [149] attempts to assess and compare the adoptability between five classes of secure BGP protocols. Chan et al. define adoptability as a property of a security protocol that describes how much it contributes to its own adoption. They measure this metric by simulation using an *incentives-compatible adoption scenario,* where a set of Autonomous Systems (AS) containing a subset of early adopters are iterated over continually until either all AS's have adopted the protocol or no incremental adoptions occur. The AS are modelled to be greedily rational and selfishly motivated and thus make decisions to adopt the protocol once the immediate *security benefit* of adoption exceeds the *switching threshold,* which represents the cost of adopting and operating the protocol.

Chan et al. consider the security benefit to be a derivative of the incremental reduction in probability that any random bit passing through said AS cannot be maliciously diverted by a single malicious AS elsewhere on the Internet. The value of the switching threshold is considered a variable input to the simulations and is varied over a range to observe how the adoption of the protocol over the AS's converges as a result.

The results of the simulations are a range of switching thresholds for which adoption of the protocol reached 100%. Chan et al. conclude that this range is the measure of adoptability arguing that it closely represents how well a protocol contributes to its own adoption.

The paper presents some strong contributions, especially its definition of adoptability and the empirical iterative method by which a measure of adoptability is obtained.

This method works well for systems where the burden of adoption rests on a single stakeholder (autonomous systems). However, for a system where multiple types of stakeholders are required to take action in order for the protocol to function, the iterative method used becomes significantly more complex. This is because the security benefit of adoption, to be received by one kind of stakeholder, could be intricately dependent on the various states of adoption of all the other kinds of stakeholders, hence the solution becomes a function of many variables. Thus requiring even more assumptions to produce output that would need to accommodate potentially multiple ranges of switching thresholds (since each kind of stakeholder may possess different switching thresholds).

Limitations of the acquisition method include its failure to distinguish between implementation deployment and adoption (as defined in [150]) and specifically the time taken for entities in question to achieve each stage and its impact on the overall process. Also not output by the acquisition method is the anticipated total time taken for adoption to reach 100%. This is important because it enables the distinction between the adoptability of two protocols, whose adoptability properties may both drive 100% adoption over exactly the same range of switching thresholds, however one protocol may consistently reach 100% in a fraction of the time (years) that it takes the other to do so. Hence adoption velocity is not considered in this method.

We have leveraged the insight gained from [149], which is an adoptability evaluation method strictly for protocols, to propose a discrete simulation-based adoptability assessment method applicable to DDoS defences (see section 5.4.4.4). Our approach links each simulation iteration, which the true effectiveness (and *global benefit*) of the defence in question, which then determines how the adoption progresses in the next iteration.

2.2.6 Discussion

Numerous publications are in agreement that a cooperative and distributed approach is necessary in overcoming the DDoS threat [40]–[44]. However, due to the interdependent nature of the Internet, the deployment of such mechanisms requires concerted collaboration between organisations. This is a challenge because, with numerous defence mechanisms available, and a myriad of heterogeneous methods of evaluating said defences, it is difficult to assess which collection of defence mechanisms and architectures should be pursued for global deployment.

Difficulties in selecting the best defence solutions for adoption, lie not only in the inconsistency of the methods used to validate propositions, but also in the suitability of evaluation outputs for conclusive comparisons between solutions. Additionally, no consensus currently exists on the definition of what constitutes “the best” and how to go about determining it. The complexity of the problem space, diversity of solution mechanisms and heterogeneity in the fundamentals of investigative efforts are further key contributing factors.

However, the introduction of a widely-accepted evaluation methodology that can assess solutions in a practicable, objective and representative manner, could help facilitate the decision-making and agreements required for the adoption of effective cooperative solutions. This postulate has precedent within cybersecurity, which has seen the use of structured evaluation lead to advancements in international standards in the past [151]–[153].

Table 5 illustrates the diversity of methods used to evaluate DDoS defences. Such diversity makes comparisons between solutions difficult due to the differences in approach, input variables and output metrics.

Nevertheless, the objective comparison of solutions is important for the selection of "best" solutions, since best cannot be determined without differentiation by some method of comparison. One may argue that neither selection nor comparison are necessary as one could deploy *all* solutions. This proposition, however, is not possible, practical or beneficial for three reasons. These include the incompatibility of defences between each other, the cost of deploying multiple defences and the collateral damage (to legitimate users) incurred as a result of deploying many additional algorithms, processes and protocols. An example of functional incompatibility between two defences is that between [34] and [154]. In [34] flows that do not reduce their sending rate during attack congestion are dropped as malicious but in [154], legitimate flows are encouraged to increase their rate of sending during attack to obtain a larger share of resources. Deploying both schemes would result in the dropping of legitimate clients by [34] due to the sending rate increases imposed by [154].

Defence	Testing Approach	Test Cases			Metris
		Legitimate Traffic	Attack Scenarios	Topology	
TVA	Simulation with NS-2	Fixed RTT = 60 ms, FTP of 20 KB file serially, abort if duration > 64 s or more than 10 same packet retransmissions	Bandwidth flood, colluding attacker at the destination side, capability request flood, imprecise authorization	Dumbbell	Transfer time and fraction of completed transfers, average values only
	Emulation on one machine	Kernel packet flood			
SOS	Theory		Attack on beacons, servlets and SOAPs with and without dynamic recovery, attack on underlying network. Attacks arrive with Poisson distribution.		Probability of communication failure
	Emulation on Planetlab	Contact 3 SSL servers		Varied number of overlay hops	Time to complete requests, average only. Healing time, single value.
DEFCOM	Emulation on DETER	Serial, fixed length, telnet sessions.	Bandwidth flood, various degrees of path sharing and defence deployment. Insider attackers.	Tree topology	Bottleneck link usage over time, legitimate traffic goodput, operating cost in time and no. of messages.
Speak-Up	Theory		Bandwidth flood		
	Emulation on Emulab	HTTP requests w Poisson arrivals, and limit on outstanding requests. Backlog of queued requests and abort if > 10 s in backlog.	Attack traffic as legitimate but at higher rate and with concurrent requests, different cost of serving requests, different levels of path sharing.	One shared LAN and multiple shared LANs with different bandwidth	Fraction of server resources to good clients, fraction of good requests served single number. Byte overhead and delay, average and 90%.

PSP	Simulation with NS-2	Traffic matrices from measurements, simulated as UDP with constant packet size.	Bandwidth flood. Distributed attack on US topology with distribution inferred from real attacks. Targeted attack on EU topology, synthetic.	Two large ISP networks, one US and one EU	Crossfire OD pairs suffering loss, total packet loss rate of crossfire traffic, loss frequency, averages, 10% and 90%.
StopIt	Emulation on DETER		Bandwidth flood, multiplexed Ips to act as multiple attackers. Target the victim and target destinations in victim's network.	Line with two AS, two end hosts and a router with limited memory.	Time to stop attack, average and error bars. Probability that an attacker is caught after certain number of attacks.
	Simulation with NS-2	Serial, fixed length, FTP transfers, abort if > 25 s	Bandwidth flood, with varied number of attackers. Flood the destination, flood its network with and without colluding attackers	Random portion of AS map.	Fraction of completed transfers, transfer time single value.

Table 5: A Survey of evaluation approaches [146]

Moreover, the literature contains examples of defences that are shown to be effective under favourable conditions but are either not tested or are shown to be ineffective under unfavourable conditions. An example is TrustGuard [47], which monitors the lack of packet-size variation as a metric to drop malicious flows but is not tested under conditions where an attacker varies the size of its packets.

Therefore, we see a clear benefit to a comparative evaluation framework, since it would assist researchers in being able to demonstrate just how well their solutions perform and where their solutions fit in the attack landscape – relative to other solutions.

Chapter 3: DiDoS Design

3.1 Introduction

For any kind of defence against any kind of DDoS attack, increasing the rate of sent packets, will eventually lead to denial of service for legitimate users. This is because the finite resources along the attack-path eventually reach their capacity to process the received data and, once that limit is exceeded, are no longer able to process all incoming legitimate data. These are known as a flooding DDoS attacks, as DoS is achieved through the scale of data sent. Once this request-processing limit is reached, further increases in sent malicious data, will further exacerbate the severity of the attack, by increasing the probability of not processing legitimate service requests.

From this perspective, defences that aim to distinguish malicious flows from benign flows, are subject to a request-processing limit and can become overwhelmed with data requests. This is the case for defences that: discriminate between malicious and legitimate traffic, reduce protocol vulnerabilities, increase request processing efficiencies or enlarge resources to increase the capacity of victims to provide service. The only defences, unaffected by request processing limits are those that prevent attacks *before* they occur, such as criminal justice proceedings, botnet elimination and deterrence.

Of the various classes of technical DDoS mitigation techniques mentioned (discriminative, efficiency-increasing, vulnerability-mitigating and capacity enlargement), all are applicable to all kinds of DDoS attacks regardless of attack layer in the OSI model, with the exception of vulnerability-mitigation techniques, which only apply to attacks that exploit particular vulnerabilities. However, within these broad categories, defences become more attack-specific. For example, within the category of discriminative DDoS defences, the defence mechanism of monitoring HTTP [155] browsing behaviour of clients as a distinction feature (between malicious and benign), will not be applicable to an ICMP [156] flood as HTTP browsing data will not be available.

Discriminative DDoS defences, work by observing the content and characteristics of received packets and use them to determine whether or not a request, flow or network event is legitimate or malicious. The observable characteristics depend on the point of observation:

source, destination or intermediate point. For example, information on the payload may only be observable at the destination if the communication flow employs end-to-end encryption. Examples of observable characteristics are data rate, packet contents, timing, attributes, sender capabilities and behaviour. Certain aspects of behaviour are only observable on the source machine, such as browsing behaviours in reCAPTCHA's [157], and the observable attributes of sent data are layer specific.

If one were to draw an "attack-map" of all possible known DDoS attacks, only a few defences would be able to mitigate against a large area of possible attacks. These defences are protocol agnostic and address features that are fundamental to the denial-of-service effect or features that are intrinsic to the attacker.

These include, increasing the capacity of the victims to serve – whether by resource enlargement or by improved efficiency – and reputation-based discriminative defences. The assumption with the latter is that, over time, attacking agents are more likely to be involved in a DDoS attack, since botnets are used repeatedly to execute attacks [158], [159].

This chapter proposes one such defence, DiDoS (**Di**stributed **D**efence **o**f **S**ervice): a hierarchical distributed DDoS defence architecture that applies granular collaboratively-maintained reputations to discriminate between malicious and legitimate clients closer to their source. DiDoS is designed to provide robust security whilst maintaining compatibility with existing Internet architecture and partial deployments. Furthermore, the benefits, gained by the entities deploying DiDoS, provide incentives for increased adoption.

3.2 Goals and Methodology

This section articulates the goals of the DiDoS architecture and describes the approach taken in its design. The idea for designing a new DDoS defence stemmed from the primary research goal of addressing the problem of DDoS attacks, which continue to grow in strength and prevalence. Figure 9 illustrates a process flow diagram of the methodology used to create DiDoS. The first steps were to research the problem of DDoS to gain a deeper understanding of the field and identify strengths and weaknesses in the current state of the art mitigation techniques. Summaries of this process are presented in Chapter 2: .

Leveraging the understanding and insights gained from the first steps, suitable design goals were chosen in accordance with identified avenues for contribution. The threat model was also constructed to provide a crucial basis upon which to assess the security of any design proposed.

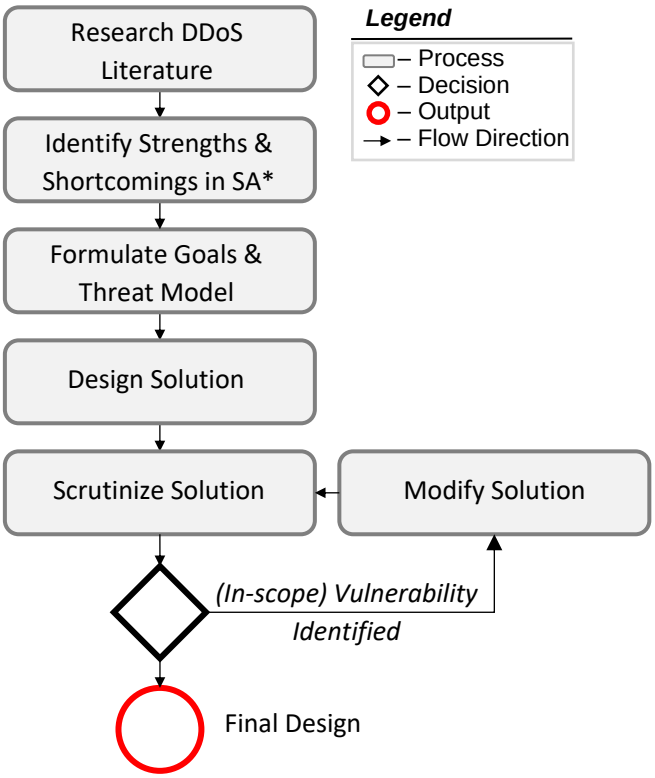


Figure 9: Design Methodology Process Flow Diagram *[1]

An initial solution was then designed. It would be inaccurate to assert that no design work occurred prior to the formulation of the threat model, since numerous design ideas arose whilst researching the DDoS problem and the state-of-the-art solutions. However, the diagram is ordered in this way since the majority of the design work occurred after the initial research phase. Similarly, references to the literature also occurred during the design phase, despite the research phase being listed first. In fact, by exploring the literature and re-examining aspects of existing solutions, new ideas were generated. This process was assisted by the use of creativity techniques such as SCAMPER, which is a technique used to generate new ideas [160]–[162].

As illustrated in Figure 9, the design approach was an iterative process, however this chapter does not attempt to document each iteration but rather provide a description of the output design with justification of various design decisions where particularly relevant to the design goals. Following each iteration, the design was scrutinized against the threat model and goals, and possible trade-offs were analysed between security properties, overheads, and anticipated performance. Any in-scope deficiencies detected were addressed in the next design iteration, until an adequate architecture was attained. The empirical evaluation of this final design is presented in Chapter 4: .

[1]SA – State of the Art

Investigations in the initial research phase revealed that, DDoS defences that distinguish between legitimate and attack traffic, enforce asymmetric treatment of benign and malicious traffic based on decisions using various metrics. These metrics must be able to be known in real-time by the mitigating device and must be available at the level of granularity with which the mitigating device wishes to mitigate the traffic, whether on a per-packet or per-flow basis or otherwise. The better a metric is able to separate between benign and malicious traffic along its spectrum, the more effective the defence, due to the reduction in false positives and false negatives, which would cause malicious traffic to avoid mitigation and, conversely, benign traffic to be obstructed, respectively. Common metrics used to distinguish between the traffic sent by malicious and benign senders include: data rate, packet size, protocol or packet integrity. [47], [67], [78], [163]

A key insight obtained as part of the preliminary investigation pertained to the relationship between the nature of the metric used for asymmetric facilitation and the resulting efficacy of the defence. It was discovered that metrics that can identify features that are largely exclusive to the attacker can be effective, however if the attacker is able to alter its attributes, then such a defence would lose its efficacy. However, metrics that additionally identify features that are largely exclusive to the denial-of-service effect are the most effective. This is because, although an attacker may have the ability to adjust their own characteristics, adjusting features that are intrinsic to the denial-of-service effect could lead to undesirable outcomes for the attacker such as reduced attack capability or attack failure.

Two examples of such metrics are in-attack volume and attack involvement frequency (AIF), since a malicious agent may be able to be identified by its higher in-attack volume or AIF. However, reducing attack volume would lead to lower attack effectiveness due to the resulting reduced resource consumption in the victim network, as a result of the reduced volume of packets sent. Lastly, reducing how often a botnet attacks could lead to reduced revenue or impact, since the botnet would attack less often and thus may not be able to charge clients or cause DDoS attack disruption as often.

From such insights, the vast scope of possibilities was refined to include the following primary design goals:

1. **Flood Resistant** – Due to the growth in unsecured IoT infrastructure enabling growing flood attacks that are forecast to exceed current defensive capabilities, resistance to flood attacks was decided as an important design goal.
2. **Distributed attack-mitigation** – To assimilate the recurrent theme, in the literature, of the necessity of mitigating DDoS attacks closer to their source, it was decided to include the means to accurately identify and eliminate malicious traffic closer to its source as a design goal.
3. **Strong Metric for Distinction –** As discussed above, the stronger the correlation between the metric a defence uses for asymmetric facilitation and the denial-of-

service effect, the more difficult it is for a DDoS attacker to evade said defence (Ceteris Paribus).

4. **Compatibility with existing Internet infrastructure** – despite the plethora of literature on hypothetical futuristic defences that are incompatible with Today's Internet, this research aims to be compatible with existing deployments, in order to maximise adoptability and, hence, usefulness.

It is important to note that this is not intended to be a comprehensive list of all of the goals of the defence design but rather a sample of main goals derived from the literature. Other goals include, fast response time and mitigation effectiveness.

The explicit exclusion of goals and research avenues was also an important part of scope refinement. Specific exclusions include:

- **Application layer attacks and defences** – Data and effects that are only visible at the application layer, lie outside the scope of this project.
- **DiDoS functionality extensions** – Potential extensions to DiDoS, such as its application to attacks other than DDoS, cooperation with application layer defences or implementation of entity granularity beyond host-level, lie outside the focus of this research project.

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

3.3 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.

Attacker capabilities refers to what an attacker can do in order to achieve its goals of defence subversion and causing denial of service. These are specified in terms of the functions an attacker can execute on compromised devices and the kind of device on which said functions are executed, which would likely affect the attack effectiveness – depending on the role of the compromised device and the nature of defence employed. Possible attacker functionalities on compromised devices include:

- **Monitoring -** the ability to capture any information stored on, generated from or passing through a device – including security secrets. Also, the ability to use a device's sensors to capture more information.
- **Manipulation** – the ability to alter any information received, stored on or sent from a device. This includes deletion.
- **Fabrication** – the ability to create any information to stored or sent from a device. This includes the ability to generate traffic based on information gathered from other parts of the Internet via other members of a botnet.
- **Actuation** –the ability to control device actuators and components that interact with a device's environment. This could result in data exfiltration through side channels.

Further delineations are possible within the above listed categories. For example, the capability of monitoring may be in part only due to a malware's ability to monitor some but not all data ingressing, egressing or persisting on a device. Similarly, it is not necessarily correct to assume that a piece of malware will have all the above capabilities by virtue of its

existence on a compromised device. For example, a Trojan malware with user (non-administrative) privileges may be able to fabricate information, but, may not be privy to information stored or have access to controllers controlled by the device operating system. However, for a more thorough assessment this threat model considers bots to possess full capabilities on the devices in which they operate, with said capabilities only being used for the attacker goals listed earlier.

Device roles can be described in terms of standard Internet roles, or those present within a particular scheme. Since this threat model is to challenge the DiDoS defence architecture, the device roles are defined accordingly.

- Host
- Private network access router
- Private network edge router
- Autonomous system access router
- Autonomous system border router

Generally, the further down the list of DiDoS roles the more severe the consequences of compromise.

In the paper Forwarding Accountability for Internet Reputability (FAIR) [80], two threat models are considered. One in which the source, transit and destination entities could be malicious and the other in which all transit AS's are not malicious. Particular threats considered include defence against replay attacks, collusion between AS's and packet injection attacks in which AS's attempt to inject fraudulent packets into the network. In Passport [78], three types of attackers with different capabilities are considered. The first is the *compromised host*, which has the ability to inject packets into the network but cannot eavesdrop on traffic sent by legitimate entities. The second is the *compromised host and monitor*, which can both inject packets into the network and eavesdrop on traffic sent by legitimate hosts. The compromised host and monitor also has the ability to replay traffic received at its location. The last kind of attacker considered is the *compromised router and host*, which can do what the compromised host and monitor can do, but additionally has the ability to modify the packets it obtains by eavesdropping. Like Passport, the threat of duplicate packets created in transit during an attack lies outside the scope of this project. We consider three cumulative threat scenarios. The first is a set of coordinated compromised hosts or botnet with no compromised routers. The second builds on the first to add compromised routers at the private network layer and the third adds compromised autonomous systems.

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 in the DiDoS threat model is

on compromised hosts, which captures the conventional threat of many compromised (likely IoT) devices that form a botnet.

Included in attacker capabilities is attacker knowledge. In the DiDoS threat model, secrets stored on uncompromised machines are considered unknown to the attacker. Transmitted data is assumed to be as confidential as the cryptographic mechanisms protecting the data and is not necessarily known by the attacker. In both cases the confidentiality of the secret is proportional to the difficulty with which an attacker can guess the secret, which can be denoted in probability over a unit of time for a select amount of attack hardware attempting to deduce the secret by brute force.

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 attacks such as replay, spoofing, reflection, collusion and slander attacks. Despite the possibility of compromised devices being able to effect changes in its environment through actuation, the potential denial of service attacks resulting from this lie outside the scope of this research. Likewise, attacks where an attacker prevents legitimate data from leaving its source device, lies outside scope. Other explicit exclusions include duplicate packets, Economic Denial of Service Attacks (EDoS) [55], application layer attacks and attacks that exploit particular hardware, software or protocol vulnerabilities.

Lastly, certain defences mitigate attacks by distinguishing between the characteristics of attack and legitimate traffic. Unlike the Trust Management Helmet proposition in [164], which is validated against static historic data in which the attacker cannot evolve their approach according to the defence, the DiDoS threat model assumes that an attacker 1) knows the characteristics of legitimate entities and 2) is able to emulate said characteristics to a reasonable degree. This allows for a much more powerful, yet realistic, attacker, which allows for a more illuminating assessment of the defensive proposal.

3.4 System Description

3.4.1 Overview

DiDoS is a DDoS mitigation architecture, operating primarily at the TCP/IP network layer, that mitigates the effects of DDoS attacks using collaboratively maintained reputations. DiDoS implements a hierarchical persistent-binding approach, in which each host is responsible for the data it sends[1] on the network and thus each host maintains the reputations of the entities from which it directly receives packets. Internet-wide participants (in DiDoS) are leveraged to provide feedback to other participating entities that enables them to maintain the reputations of the entities from which they receive traffic.

[1] Including both the data that it generates and that which it forwards

The layers of accountability include the *inter-autonomous system layer* at the top, in which AS's hold each other to account. The next layer is the *intra-autonomous system layer*, in which AS access points hold hosts and private networks connected to them to account. The next layer is the *private network layer*, in which private routers hold the hosts that connect them to account. The final layer is the *host layer* in which the software running on an operating system can be held to account by a piece of software running on that host. Such functionality could lie within the operating system or may be added as an extension to antivirus software on the machine.

A lightweight cryptographic scheme, involving message authentication codes, is used to prevent the spoofing of the actions of participating entities. This includes the protection of the integrity of the feedback accepted by participating entities.

A setup phase allows the sharing of symmetric keys via Diffie Helman exchanges between accountable nodes and the nodes to which they are accountable. Once set up, each participant entity attaches a trustworthiness rating to each packet that it forwards. This rating is based on the reputation of the accountee that the accountor in question collaboratively maintains, as described. This approach provides per-packet information that is used to make better decisions (e.g. which packets to drop) at moments of contention along the transit path.

Though it is possible for servers to maintain whitelists of clients, it is not practicable or efficient for them to store and maintain effective blacklists of all malicious clients. This solution distributes the reputation management responsibilities by enabling each sender to be held accountable for the data it forwards, thus providing incentive to mitigate malicious traffic by properly managing the reputations of the entities from which it receives traffic.

Much of the literature is agreed on the necessity of global cooperation for effective DDoS mitigation, but the personal incentives, for all kinds of stakeholders (e.g. end users, service providers, manufacturers etc.) to invest in any chosen collaborative solution, remains an issue. This is likened to *the tragedy of the commons* in [59], which is a term used in social science that refers to situations where individuals attempt to maximise their own personal benefit more than that of the common public. This solution helps address this by providing personal incentives to each adopting entity, which is the ability to maintain a good reputation by preventing the spoofing of one's own identity for malicious purposes and by providing the means to reduce participation in DDoS attacks.

The DiDoS architecture consists of three **functional layers**, with each layer being dependent on that below it. At the top, is a *distributed mitigation mechanism*, responsible for the elimination of malicious traffic at points of contention in the network. Below that, is a *collaborative reputation-management system* that defines and maintains the reputations used by the distributed mitigation mechanism, and the last layer is a *hierarchical identification*

scheme that specifies the identifiers against which reputations from the second layer are bound.

These three layers are outlined in subsequent sections. However, in orthogonality to a functional hierarchy, DiDoS also consists of network entities playing different **roles** to achieve said functions. The third dimension of DiDoS consists of the **phases** of operation, which vary according to the functional layer in question. In this way, at any functional layer of DiDoS, there are networked entities that play different roles which vary depending on the phase of operation of that layer. This is illustrated in Figure 10.

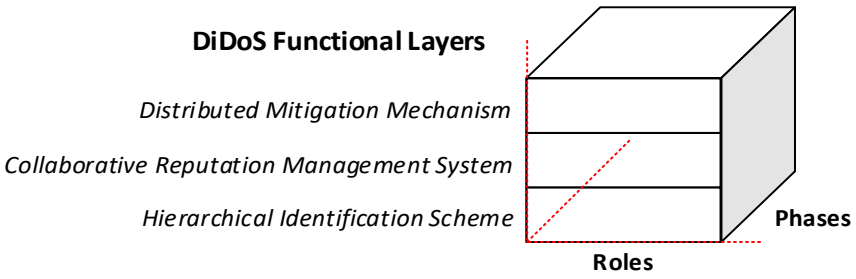


Figure 10: The Three Dimensions of the DiDoS Architecture

The elements of the three-dimensional matrix in Figure 10 depicting the DiDoS architecture are explored in subsequent sections.

3.4.2 Hierarchical Identification Scheme and Roles

Beginning with the *hierarchical identification scheme* (HIS), which is the foundation of the *functional layers* dimension, this section describes its organisation and how it relates to DiDoS *roles*. As indicated in its name, the hierarchical identification scheme is also hierarchically structured, like the functional layer axis in which it lies, and consists of three layers: the autonomous system (AS) layer – at the top, followed by the private network layer and then the host layer. This is illustrated in Figure 11.

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 the creation of entity identifiers and the persistent binding of entity actions to their identifiers being carried out by devices within the layer above. The exception, however, is the top (AS) layer in which identifiers – autonomous system numbers (ASN's) – are assigned and maintained by the Internet Assigned Numbers Authority [165], and persistent binding is carried out on a peer-to-peer basis.

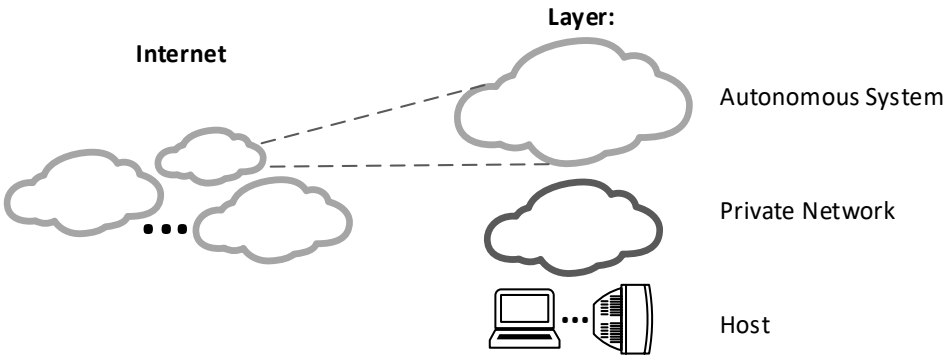


Figure 11: DiDoS Hierarchical Layer Organization

Thus, the identification scheme provides the basis of accountability upon which reputation-based mitigation can occur.

In alignment with the hierarchical identification scheme, DiDoS consists of three roles that are carried out by participating devices within each layer: reputor, reputee and reputation management center.

The role of a *reputor* includes the functions required to manage the reputations of connected entities in the layer below said reputor. An access router managing the reputations of connected desktops in a private corporate network would be playing the reputor role, for example.

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 3.4.3.

The last role in DiDoS is a *reputation management center*, 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 reputation self-assessment and reputations projected out by members of the administrative group.

As part of the DiDoS identification scheme, reputees are authenticated to reputors with different strengths of identification mechanism. These varying strengths of authentication are leveraged in the DiDoS collaborative reputation management system to reflect the ability to hold entities to account for their actions. As shall be shown, reputees with stronger authentication incur smaller 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 entity's 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) [166], 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.

Details of the cryptographic components that under pin the HIS are presented in 3.4.5. Due to the authentication between reputor and reputee, DiDoS is able to prevent IP address spoofing by checking that the IP address of the packet lies within the set of expected IP addresses in the reputation domain, and then verifying the *Transit MAC* on the packet to ensure that it was indeed sent by the authenticated client assigned to that IP address.

3.4.3 Collaborative Reputation Management System

3.4.3.1 Summary

At the center 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 management centers (RMCs) coordinate the reputation management of entities within administrative domains.

During DDoS attacks, 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[1] signalling of sender (reputee) 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. The cryptographic protection of reputation scores embedded into the headers of reputee packets is detailed in section 3.4.5.

Reputations are collaboratively maintained by a feedback process in which reports, containing a sample of the traffic (packet headers) sent to the victim at the time of attack, are generated by the victim's network 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 the reputation management center, in

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

which the reputation management center agrees to receive the report, the report is passed from the reputee to the reputation management center via the reputor. At the reputation management center, 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 DiDoS compliant packet – is inspected and reputations of reputees R_n 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 in Equation 1:

Equation 1

$$R_n[x] = R_n[x-1] - \beta\left\{a_1 P_b + \frac{a_2 \mathcal{V}}{a_3 I + a_4 \mathcal{A}}\right\}$$

β is the convergence factor that controls the rate at which R_n generalizes to a long-term value, P_b is the base penalty amount, and x is a discrete time event variable representing the points at which R_n is updated. The coefficients a_1, a_2, a_3 and a_4 provide weighting to the components of the update amount and can be tailored according to the preferences of the RMC. 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 3.4.2) 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:

Equation 2

$$\beta = a[x] \times \beta_0$$

Where,

Equation 3

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

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 fraction of its own reputees that are malicious m, the number of its reputees N, an equilibrium[1] attack rate Z_E, 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 management center creates new reports containing the packet headers of the original report, grouped according to the autonomous system (AS) from which they originated. These new smaller reports are then disseminated to their source AS's 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 existence of a process that allows the rebuilding of tarnished reputations or mechanism that prevents the permanent flatlining of all reputations to an equal bottom value. These issues are addressed in DiDoS through the mechanism of *passive periodic accumulation* (PPA).

In PPA, reputors increment the reputations of all their reputees equally by a small amount every fixed period of time (e.g., daily). This allows reputee reputations to be strengthened over incident-free periods, which allows benign entities that have been circumstantially involved in an attack, or 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, which shall be captured in the evaluation of the defence.

[1] The equilibrium attack rate Z_E 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 11).

3.4.3.2 Reputation Updates

This section delves into the dynamics concerning the reputation levels that reputors ascribe to their reputees, considering aspects like reputations chains and reputation management goals.

The reputation update is a crucial part of the DiDoS architecture. It supports the classification of packets in hierarchical levels that enable victims of DDoS flooding attacks to quickly determine which packets are more likely to be malicious. This section outlines the reputor update algorithm design goals and presents the resulting algorithm by positioning it as a solution that achieves said design goals whilst satisfying identified design constraints.

The ultimate goal of the reputation update algorithm is to enable a reputor to be able to identify malicious reputees and thus producing an environment where the reputations marked on packets can be used by victim entities under attack to quickly distinguish between malicious and benign packets from unknown senders. In order for this to happen, the reputation levels marked on the packets reaching the victim should provide a fair and granular reflection of the reputations of the packet senders. For example, if each reputor, provided reputations on packets that are based only on observations of that reputor's reputees, then the reputation levels, on the packets arriving at the victim, would only reflect high reputation granularity derived from observations by the final reputor (in the packet forwarding chain) of its direct reputees.

Due to hierarchical and peer to peer accountability of the DiDoS architecture, two kinds of reputation level assignments must occur as packets traverses the Internet from source to destination: one where the received packet already has a reputation level assigned to it, and one where it does not.

If we consider a DiDoS-compliant packet-forwarding chain (Figure 12), the first reputor in the chain observes its reputees and independently marks their resulting reputations on each packet they send. These markings are described as independent since no prior reputation level exists on the packet. A reputor receiving packets from another reputor, however, would receive packets that have already been assigned a reputation level by the sending reputor. In order to prevent the receiving reputor from blindly forwarding packets from a sending reputor that maliciously overrates its sent packets, a method is needed for the receiving reputor to integrate its own reputational observations of the sending reputor into received packets.

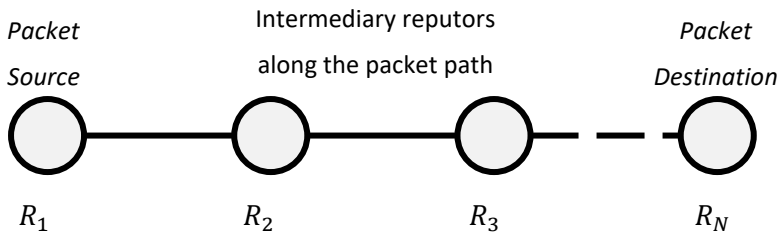


Figure 12: A chain of reputors along a packet's path

A simple solution would be for the receiving reputor, to independently mark the reputation score it has of the sending reputor into the header of the received packet. However, this approach discards the variation in the packet reputation levels embedded by the sending reputor which could help downstream victims more accurately determine the trustworthiness of the sent packet, due to the finer granularity of the identities upon which the marked reputation levels would be based. Ideally the algorithm employed by the receiving reputor would account for the reputation sent by the sending reputor, but also employ a mechanism to detect, and penalize dishonest reputation level assignments by the sending reputor. This requires the receiving reputor to input the received packet reputation level into a function and apply its own assessment of the sending reputor into that function in order to produce an output reputation that is added to the received packet. Such a function would need to be lightweight so as to avoid introduction of excessive transit delays. Thus, the reputation level marked to packets sent from the r'th reputee to the n'th reputor can be expressed as follows:

$$L_n^{r_n} = f(S_n^{r_n}, L_{n-1}^{r_{n-1}})$$

Where L_n^r denotes the reputation level marked to packets sent from the r'th reputee to the n'th reputor, S_n^r denotes the reputation score maintained by the n'th reputor to characterize its r'th reputee, and f is a function of the two variables. The function f can be a weighted average function such that:

Equation 4

$$L_n^{r_n} = \frac{a \cdot T\left(S_n^{r_n}\right) + b^{r_n} \cdot L_{n-1}^{r_{n-1}}}{a + b^{r_n}}$$

Where $T\left(S_n^{r_n}\right)$ is a function $T()$ that converts $S_n^{r_n}$ from a score to a level, b^{r_n} is a coefficient for the r'th reputee of the n'th reputor, that ranges from 0 to b_{max} and a is defined by b such that:

$$a = b_{max} - b^{r_n}, \qquad 0 \le b^{r_n} \le b_{max} \therefore\ 0 \le a \le b_{max}$$

The variable $S_n^{r_n}$ is the reputation score that the n'th reputor in the packet forwarding chain maintains concerning the r'th reputee accountable to that reputor. It is decremented in

response to verified attack feedback reports that contain packet headers sent from the reputee in question. Similar to $S_n^{r_n}$, b^{r_n} is decremented in response to attack feedback reports, however, instead of being decremented by the presence of reputee attack packets, b^{r_n} is decremented in accordance with the proportion and severity[1] of overrated packets sent by the reputee in question. In this way, b^{r_n} tracks the level of trust that a reputor has in the accuracy with which its reputee rates the packets sent by said reputee.

$S_n^{r_n}$ is maintained for each reputee r of the nth reputor in question. $S_n^{r_n}$ begins on a starting value for all reputees and is decremented in response to attack feedback reports. Since reputors further down the reputor chain are more likely to receive feedback reports (due to the network branches and increasing number of devices indirectly connected to the reputor in question), a challenge exists in the reputation update process. Suppose an edge reputor (R_1) monitoring its reputees employs an algorithm that sufficiently decrements misbehaving reputees such that their reputation becomes distinguishable from legitimate entities. The next reputor in the chain, (R_2) for which R_1 is a reputee would receive all the reports that were received by R_1 but these reports would be attributable to R_1 by R_2, instead of the reputational penalties being divided by a number of reputees. In this situation, if the exact same algorithm that R_1 applied to its reputees were applied by R_2 concerning R_1, the reputation of R_1 would most certainly be negative. However, suppose that R_1 was not the only reputee of R_2 but was in fact least involved in attacks, we can quickly deduce that if the same algorithm were applied (by R_1 & R_2) all the reputations of the reputees of R_2 would be negative. In fact, we can more generally see that each reputor (R_n) cannot have a higher rating with its own reputors (R_{n+1}) than the lowest rated reputee that (R_n) is responsible for.

This is the case when an absolute attack involvement frequency is considered, which does not scale well in the presence of traffic aggregation and reputation chains. This is an issue because it would inhibit the granular distinction between malicious and benign traffic at points of network contention (where ill-reputed traffic is dropped).

There are multiple possible ways to address this problem (where an inflexible reputation update algorithm could cause the reputations of reputors to be no higher than that of the reputee with the worst reputation amongst all reputees downstream of any attack feedback path). One such solution is to reduce the penalty that a reputor levies on its own reputees in proportion to 1) the mean number of reputees contained in attack feedback reports received by the reputor, and 2) the ratio between the number of reports received an the number of reports expected; such that the greater number of reputees the less the attack feedback penalty, and the more the reports exceed expectation the less the attack feedback penalty.

[1] The disparity between the reputation level recorded by the reputee on the packet header and the level expected by the reputor, where the expected level is given by $T\left(S_n^{r_n}\right)$

In DiDoS, the main goal of a reputor is to mark the packets it sends with appropriate reputation levels. The DiDoS architecture incentivizes this goal in two ways. Firstly, marking packets from legitimate senders with a higher reputation enables them to experience a higher quality of service, which is in alignment with the goals of a service provider to its clients. Secondly, marking the packets of malicious senders with low reputations enables a reputor to preserve its own reputation with its own reputors by avoiding reputation penalties associated with forwarding malicious traffic marked as benign. Ultimately it is the prerogative of the reputor to decide which algorithm to use to accomplish these goals.

The passive periodic accumulation (PPA) amount A_p can be chosen based on the time period T_r that a reputor deems it should take for a reputee to recover its reputation after being cleansed of malice. The total reputation points traversed by this recovery R_r would range from the minimum reputation to score S_{min} to a reputation score that represents a level of good reputation S_g, such that:

Equation 5

$$R_r = S_g - S_{min}$$

Thus A_p can be calculated from:

Equation 6

$$A_p = \frac{R_r T_A}{T_r}$$

Where T_A is the time interval between increments of PPA. Using value of the PPA amount, the base reputation penalty amount P_b can be calculated from:

Equation 7

$$P_b = r\, A_p$$

Where r is a value chosen by the reputor that stipulates the number of PPA increments that a reputee would need to receive in order to recover from a single base reputation penalty decrement. Multiplying the value of r by the period of time T_A between PPA increments yields the time it takes for a reputee to recover from a single attack penalty. This time period is significant since it is the amount of time that a malicious reputee would need to wait after attacking to prevent its reputation from trending downwards and hence its attacks becoming ineffective. For example, if T_A is 1 day and r is 30, then it would take one month for a reputee to fully recover from an attack reputation penalty provided that it receives no further attack penalty during that period.

The convergence factor β tapers the rate of convergence of reputee reputations and can be made to be dynamic:

Equation 8

$$\beta = \frac{Z_E}{Z_A}\beta_0$$

Where Z_E is the expected rate of attack reports to be received by the reputor, Z_A is the actual number of attack reports and β_0 is a coefficient used to limit the range of β. In operations, the value of Z_A can simply be measured obtained whereas the value of Z_E can be calculated from the number of reputees that a reputor has and estimates of the proportion of reputees that are malicious and the estimated attack rates of malicious and non-malicious reputees. A brief analysis of Equation 8 reveals that if the actual rate of received reports is less than the expected rate of received reports, then the convergence factor is increased. This helps address the reality that not all attacks, in which the reputees of a reputor participate, may be reported. This phenomenon can occur if the packets from the attacking agent are not captured as part of the sample of the attack traffic contained in the attack feedback report and would likely be more prevalent at lower levels of adoption of the DiDoS architecture.

3.4.3.3 Attack Feedback

This section describes, in more detail, the way in which attack reports are generated and fed back to the reputors involved in routing the packets of the attack to its destination. For the purpose of brevity, it is to be assumed that all entities mentioned in this section are DiDoS compliant, unless otherwise stated.

The process of attack feedback begins with attack detection. A victim of a DDoS attack notifies its reputor that it is under attack. This notification is passed upwards, from reputor to reputor, until it reaches the autonomous system layer, where the autonomous system enlists the support of a trusted third party to corroborate that the specified victim is indeed under attack. The autonomous system, in which the victim is located, is the *attack feedback owner* (AFO) and the trusted third party is the *attack feedback endorser* (AFE).

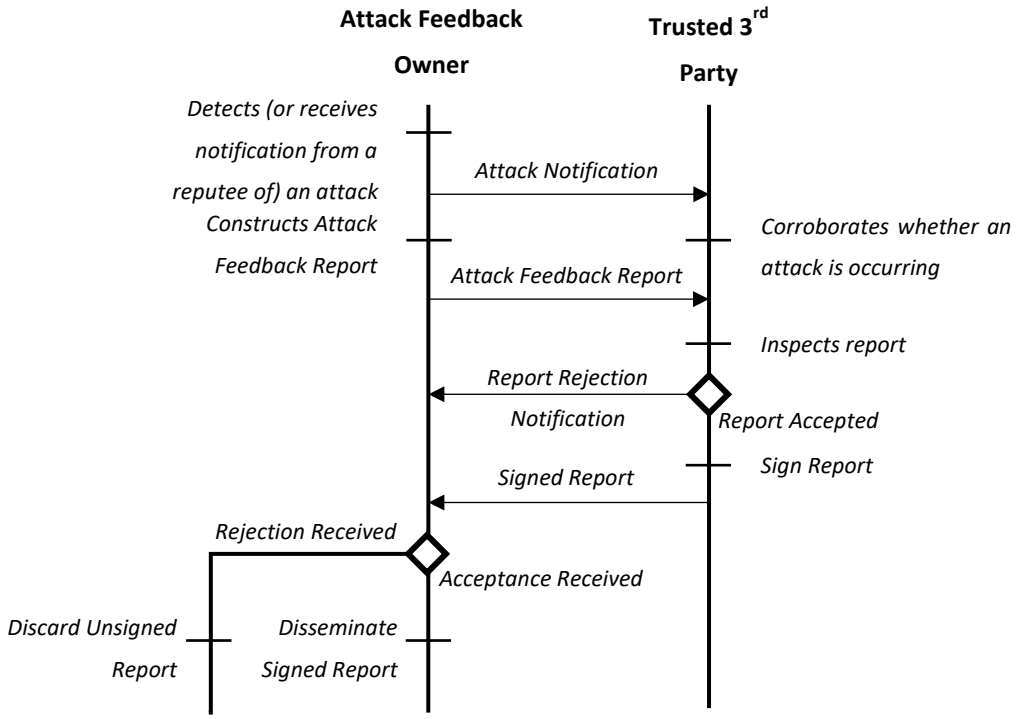


Figure 13: Protocol Overview of Attack Feedback Report Authorization

After notifying the AFE, the AFO then generates an attack feedback report (AFR) containing information about the attack including:

- The time of the attack
- The attack destination
- A sample of the DiDoS compliant packet headers involved in the attack
- The start and end time between which the attack sample was collected.
- The total number of packets collected in the attack sample

The report contains a header section and a body section, with the header section containing information about the attack and the body section containing the sampled DiDoS packet headers. The AFO signs the report header which prevents intermediate AS's from editing the header without the possibility of detection. The sample of packets in the AFR body are grouped together according to the first feedback hop, and the packets within each first-hop group are further grouped according to the second feedback hop. This is illustrated in Figure 14. The AFO adds a hash of the AFR header to each AFR group and signs all of the groups. Doing so binds each part of the AFR body to the AFR header such that when the AFR body is divided up and disseminated along the feedback path, Malicious ASs cannot swap-out the report body without detection. Furthermore, signing the constituent AFR body groups prevents intermediate malicious ASs from adding packet samples to the report that may have been headed to the destination and may have been sent by the source AS but were not included in the original report that was signed by the trusted third party. The division of the AFR during

attack feedback dissemination, helps reduce the amount of data to transfer between each feedback hop, since an AS is only concerned with the packets it interacted with in the AFR.

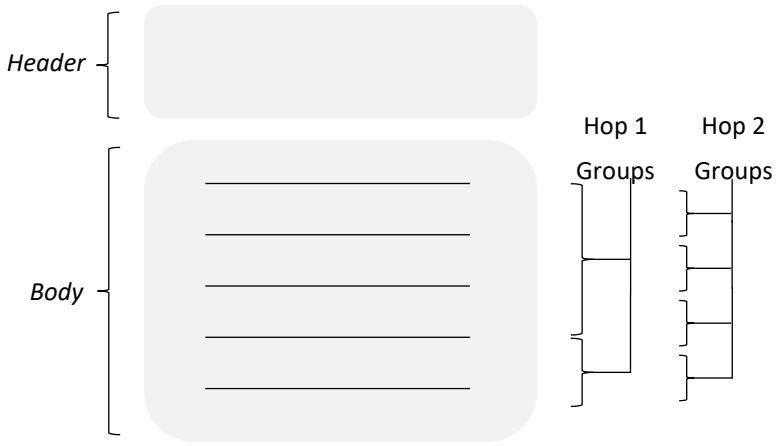


Figure 14: Attack Feedback Report (AFR) Structure

After receiving the notification from the AFO, the AFE monitors attack traffic to corroborate that an attack is occurring. Once the AFR is created and sent by the AFO to the AFE, the AFE reviews it and signs the AFR header and also signs each group of the sample packets contained in the AFR body. This allows ASs to verify the integrity of the AFR's constituent parts as it is divided up and disseminated along the feedback path. Figure 15 illustrates the dissemination of the AFR along the feedback path. At each AS, the report is divided such that the next AS along the feedback path is sent the AFR header and the portion of AFR body pertaining to the sample of attack traffic that passed through the 'next AS' to AS that is sending the AFR. The AFE only approves (and thus signs) the AFR if no malfeasance is detected in the report and if the AFE concurs that an attack did occur at the destination and time specified in the report. Once approved and signed the AFE returns the AFR to the AFO, which then disseminates the signed report along the feedback path (Figure 15).

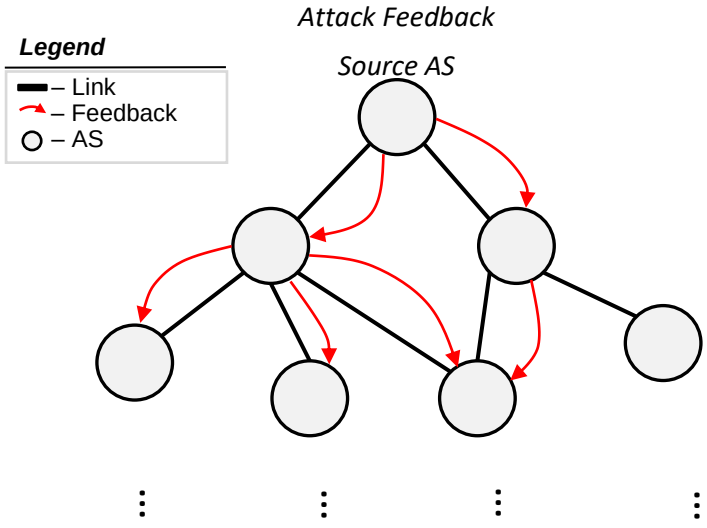


Figure 15: Illustration of Attack Feedback Report Dissemination

The purpose of digital signatures contained within the report is to enable subsequent recipients of the report to cryptographically verify its authenticity as it is divided up and disseminated along the feedback paths. Each AS along the feedback path verifies that an attack did occur by verifying the report header signed by the trusted party and the AFO. By verifying the signatures of the AFR body, each AS also verifies that each packet within the AFR body did reach the destination AS, since the AFR body was signed by the AFO and the subsequent addition of any packets would invalidate the AFR body signature. Each AS then updates the reputations of its reputees based on the packets contained in the AFR body. Packets in the AFR body that the AS cannot verify that it did forward (based on the MAC, created with a secret known only to the AS in question, that would have been added to the packet as it traversed the AS) are rejected.

Cases of feedback replay attacks are limited by the signed and MAC-protected timestamps contained within the report header and packet headers of the sample AFR body. A reputor will not accept an AFR, that has already recently been processed by the reputor or whose attack occurred outside of a certain timeframe from the time of receipt of the report. Furthermore, the packets contained within a report's body will be rejected if the timestamp on the packet occurs outside an acceptable window of the timestamps in the report's header.

3.4.4 Distributed Mitigation Mechanism

The DiDoS distributed mitigation mechanism describes the way in which reputations, formed in 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 the reputee. 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 DiDoS architectural layers and administrative 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, marked on each packet, by a process of linear quantisation. 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 low reputation scores are forwarded in low-priority queues and are therefore more likely to be dropped. By implementing a proportional relationship, between a packets 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 at the expense of higher reputed packets.

The direction of packets into higher priority queues based on their reputation may incur some additional forwarding delay, however, since lower reputed packets are continuously being deprioritized, the response time of DiDoS to an attack can be described as instantaneous.

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

3.4.5 DiDoS Header Specifications and Cryptographic Protection

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 [167]. Therefore, IPv4 offers ten 32-bit words of available header-space to accommodate packet options. DiDoS leverages this available option space in order to ensure compatibility with existing Internet infrastructure, as illustrated in Figure 16. By using Internet options, non-DiDoS-compliant routers are able to receive DiDoS compliant packets and ignore their DiDoS header option whilst forwarding them intact.

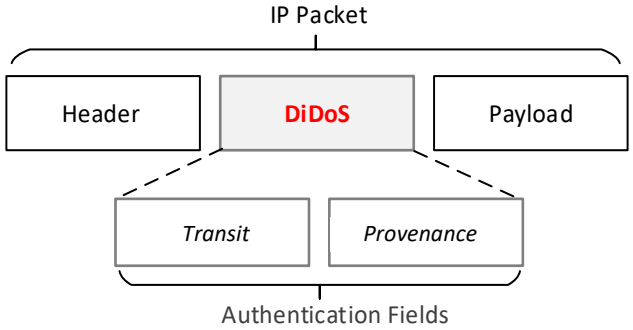


Figure 16: Illustration of a DiDoS Compliant IP Packet

As illustrated in Figure 16, the DiDoS packet option consists of two authentication field sections: *provenance* and *transit*. The provenance section contains fields that allow a reputor to authenticate packets in feedback reports by verifying (using the feedback message authentication code [MAC]) that it indeed sent the packet, and to identify the reputee responsible for sending the attack packet and appropriately (using the reputation score that the reputee ascribed to its sent packet) hold it to account. The reputor ID field in the

provenance section of the DiDoS header also identifies the reputee to which a reputor must disseminate portions of its attack feedback report. Additionally, the first DiDoS-compliant autonomous system (AS) in the route of a packet adds a *destination MAC* which prevents source address spoofing and covert tampering. If the number of additional DiDoS provenance header section bytes required by the number of DiDoS hops between the source AS and the destination AS would cause the maximum transmission unit (MTU)[1] to be exceeded, then the packet is fragmented before the destination MAC is added to each fragment. This protects against false positive errors resulting from destination MAC verification, should the packet be subsequently fragmented after the destination MAC is added. The destination AS uses the source IP address of the received packet to look up the secret key shared with the source AS. If a shared key exists, the MAC is checked and, if its verification fails, then the packet is dropped. If a shared key exists yet no destination MAC is present on the packet, the packet is also dropped. Otherwise, the packet is routed. The destination AS removes all provenance fields from the packet as it routes the packet towards the end node or private network. The header evolution of a packet traversing a DiDoS internetwork is illustrated in Figure 17 with the coverage of the MAC fields illustrated in Figure 18. Table 6 specifies the field names and numbers of the DiDoS header, which these figures reference.

The transit section contains transit MACs that enable the authentication between reputor and reputee. In the cases of partial deployment, where intermediate AS's may not be DiDoS compliant, the transit MAC added uses a secret key shared with the current hop and the next DiDoS compliant hop.

DiDoS headers are only added by reputees that are sending their packets to a reputor with which they are authenticated. This ensures that the provenance section of the DiDoS header will be appropriately removed by the time the packet reaches the destination node. This requirement is due to *payload encapsulation* (illustrated in Figure 17), which leverages a portion of the IPv4 payload to store DiDoS provenance fields that may not be reliably processed by non-DiDoS-compliant end-nodes. Payload encapsulation significantly extends the number of possible DiDoS feedback-enabled hops to over 25 reputors without needing to sacrifice the security and scalability of DiDoS by reducing the MAC sizes and the ID field lengths in order to squeeze multiple feedback-enabled hops into the conventional 320 bits of IPv4 options header space. No encapsulation is required to support the same number of reputor hops in IPv6 since both the *Hop-by-Hop Options* header and the *Destination Options* header are specified to be up to 256 octets in length [168].

[1] The MTU can be discovered using the standardized Path MTU Discovery protocol [208]

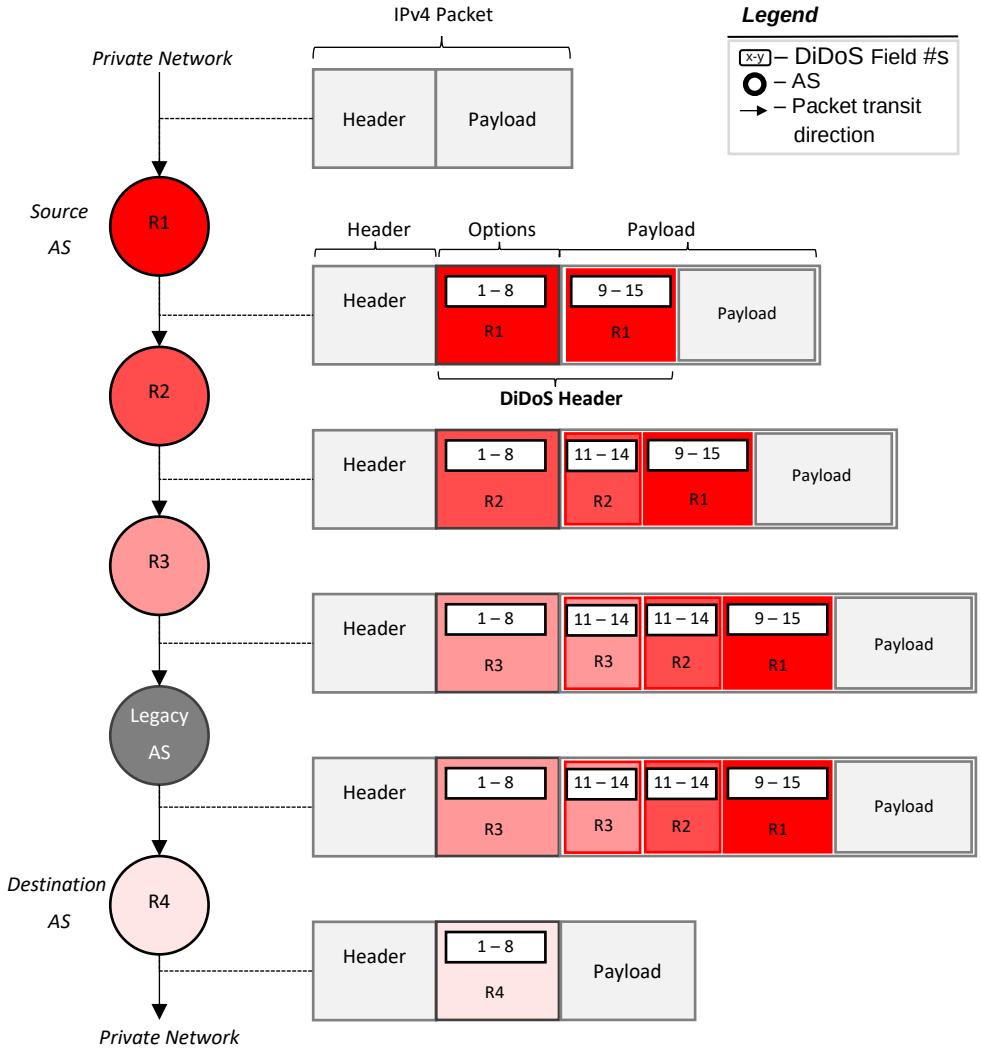


Figure 17: Diagram illustrating payload encapsulation and the evolution of the DiDoS header along an example route

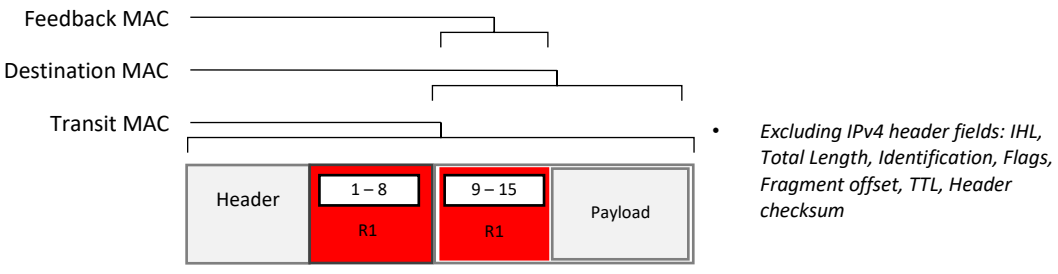


Figure 18: Illustration of the field coverage of the DiDoS MACs

Field #	Type	Name	# Bytes	Description	Added By
1	Transit	Option Type	8	Standard IPv4 Options field [169]. Specifies the type of the option.	1st DiDoS entity on path
2	Transit	Option Length	8	Standard IPv4 Options field [169]. Specifies the length of the Transit portion of the DiDoS header.	1st DiDoS entity on path
3	Transit	Transit MAC	32	Constructed using shared secret between reputor and reputee. Enables Reputee-reputor authentication.	Each DiDoS reputee
4	Transit	Reputee ID	32	Identifies the adding entity to its reputor.	Each DiDoS reputee
5	Transit	Reputee Reputation Level	3	Provides a rating, of packet that is to be routed, to the reputor of the entity that is setting this field.	Each DiDoS reputee
6	Transit	Transit Padding	5	Ensures the DiDoS Option Header is a multiple of 8.	Each DiDoS reputee
7	Transit	Provenance Fields Length	8	Specifies the provenance portion of the	Each DiDoS reputee
8	Transit	First AS Feedback Byte	8	Specifies the byte from which the provenance fields of the source AS begins. (Is used by the destination AS to locate field #15)	Each DiDoS reputee
9	Provenance	Nonce	16	Random generated number.	1st DiDoS entity on path
10	Provenance	NTP Timestamp	64	Indicates the time at which the packet traversed the first DiDoS reputor.	1st DiDoS entity on path
11	Provenance	Reputor ID	32	Identifies the reputee that sent this packet to the reputor that adds this field.	Each DiDoS reputor
12	Provenance	Reputor Reputation Level	3	Records the packet reputation level rated by the reputee that sent the packet to this (adding) reputor.	Each DiDoS reputor
13	Provenance	Provenance Padding	5	Ensures the DiDoS provenance data is a multiple of 8.	Each DiDoS reputor
14	Provenance	Feedback-MAC	32	Uses a secret known only to the adding reputor. Enables authentication of packets in feedback reports.	Each DiDoS reputor
15	Provenance	Destination MAC	64	Constructed using secret between source & destination AS. Enables Reputee-reputor authentication.	1st DiDoS AS on path

Table 6: DiDoS Fields Specification

Due to payload encapsulation reducing the available IPv4 payload size by up to 256 bytes, fragmentation of received packets may be required. DiDoS compliant packet originators, ensure payload sizes are small enough to accommodate packet length growth. The exact amount of DiDoS hops accommodated is at the discretion of the packet originator, but with a recommended minimum of 7 hops (83 bytes). For perspective, the default maximum TCP segment size is 536 bytes [170] which is transported inside IPv4 packets which support a maximum size of 65,535 bytes (although packet sizes can be limited by data-link layer protocols such as Ethernet which has a standard maximum transmission unit of 1,500 bytes).

Reputors receiving packets, will only fragment if the available payload space is insufficient for the reputor to add its own provenance header fields. DiDoS fragmentation preserves the provenance section of the DiDoS header on each resultant fragment (packet). This preserves the accountability of all fragments during the attack feedback process.

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 DiDoS reputee 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 nonce is added by the first DiDoS entity in the transit path and a timestamp is added by the first NTP[1]-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.

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.

[1] Network Time Protocol.

Chapter 4: DiDoS Evaluation

4.1 Overview

Following the design of DiDoS, a reputation-based DDoS mitigation architecture, this chapter is dedicated to its evaluation of how and how well the architecture meets its design goals.

The chapter is divided into two parts: Design Assessment, and Empirical Validation. The former reviews how DiDoS met the design goals and compares the DiDoS architecture to other DDoS mitigation schemes. The Design Assessment also scrutinizes the design against threats identified in the threat model (section 3.3).

The Empirical Validation section, on the other hand, addresses how well the DiDoS architecture performs. This performance refers to the capability of the collaborative reputation management system to identify malicious entities and includes to the mitigation effect of the DiDoS architecture.

The chapter includes the results of the various experiments conducted and concludes with a discussion of the evaluation and the DiDoS architecture.

4.2 Design Assessment

4.2.1 Review

This section reviews DiDoS in the context of other related DDoS defences, highlighting the differences between DiDoS before summarizing with a comparison table (Table 7). The section concludes with a review of how the design goals presented in section 3.2 are addressed by the DiDoS architecture.

4.2.1.1 Against Related DDoS Defences

Passport [78] defends against DDoS attacks that used spoofed IP addresses. Operating at the autonomous system (AS) level, Passport prevents malicious AS's from passing off forged packets to Passport-participant AS's as if they were from another participant AS. Like DiDoS, Passport requires a setup phase in which secrets are generated between AS's that are used in the message authentication codes (MAC's) appended to packet headers originating from a Passport compliant AS.

In this process, Passport piggy-backs off border gateway protocol (BGP) [79] in order for the source AS to obtain the path information of the destination address. The source AS then appends one MAC for each AS on the source-destination path using the respective secrets it shares between them as keys. On traversal to its destination, each MAC is verified and discarded by its corresponding AS – to prevent cryptanalysis. If the MAC fails verification the packet is dropped. If an AS receives a packet purporting to be from a passport participating AS, yet that packet possesses no appended Passport, that packet is dropped. In this way Passport prevents inter-AS spoofing between participating AS's.

DiDoS also prevents inter-AS spoofing but differs in approach since the source AS in DiDoS does not add one transit MAC for each AS on the packet path but only one using a secret shared with the destination AS. This design choice reduces header size requirements and the amount of packet processing required, to allow for the placement of feedback MACs by each reputor along the packet path. The result is that intermediate AS's in DiDoS still have the capability to detect and drop spoofed packets and packets with invalid MACs since all the information required is still present. Similar to DiDoS, the number of MACs associated with a packet grows with the number of autonomous system hops.

Fundamental differences between DiDoS and Passport arise in regard to the scope of the attack mitigation of each. Passport cannot mitigate against attacks where attackers do spoof IP addresses but, instead of spoofing an IP address from another AS (i.e. AS's other than the one that the attacking device uses to Access the Internet), spoof the IP addresses of other entities within the same AS domain as the attacker (intra-AS spoofing). By combining a hierarchical identification scheme, a collaborative reputation system and a distributed mitigation mechanism, as described in Chapter 3: , DiDoS offers more scope of mitigation.

Forwarding Accountability for Internet Reputability (FAIR) is an architecture proposed in [80] that addresses the incentives for Autonomous Systems (AS's) to regulate their traffic by the introduction of automated punitive action against misbehaving source AS's. Misbehaviour is defined as the violation of destination AS policies by source AS's. These policies are characterized by committed information rate (CIR) and committed burst size (CBS) agreements that are specified beforehand to the source AS by the destination AS. The punitive action involves the marking of traffic sent from violating AS's by a suspicion bit, which enables transit and destination AS's to deprioritize said traffic. The key feature of FAIR is the ability for destination AS's to cryptographically prove misbehaviour to sources AS's by storing and returning offending packet headers.

Like DiDoS, in FAIR the returned packet-headers include message authentication codes (MAC) that enable the source AS to verify that it indeed sent the traffic. However, contrary to DiDoS the input fields to the MAC include factors that enable the source to verify policy violation only, rather than additionally verify packet-provenance, which is a crucial factor in enabling the source AS to apply corrective measures.

Another fundamental difference between DiDoS is the way in which traffic deemed excess is identified and handled. Although FAIR implements a "suspicion" bit to mark suspicious traffic, the basis upon which suspicious traffic is marked does not distinguish between DDoS-attack and legitimate traffic, but rather only between traffic forwarded by AS's that are deemed by a destination AS to have violated sending policy agreements. In this way, the architecture is fundamentally prone to false positives and negatives as the circumstances required for policy violation (CIR and CBS violations between AS's) bear weak correlation with individual DDoS-attack packets. For example, under this method, no DDoS attack traffic from a source AS will be marked as suspicious if the source AS never violates destination AS sending policies, which is possible if the source AS applies traffic policing to fulfil policy agreements. Similarly, all legitimate packets from AS's that violate policy agreements will be marked as suspicious regardless of their involvement in any DDoS attack. DiDoS, on the other hand, leverages the frequency of entity involvement in attacks, the honesty of sender ratings and the voluminous contribution of involved entities to said attacks as metrics to distinguish between malicious and benign packets in the form of reputations.

In the paper *Fine-Grained Capabilities for Flooding DDoS Defense Using Client Reputations (FGCD)* [34] the authors propose a ticket-based mechanism that grants tickets to prospective clients, based on their accrued reputation, that enable them to communicate with the server. The assigned tickets encapsulate a credit score commensurate to the client's overall traffic contribution during an attack, which is assumed to be higher for malicious clients. Additionally, a penalty score applied to the packets of clients that don't reduce their traffic in response to congestion is used for policing and ticket-request prioritization. Thus, historic attack-traffic contribution and response to congestion are the metrics of distinction of the defence.

The main contribution of the work involves the proposition of more ticket reputation levels, which facilitate the differentiation between clients to a finer granularity. Furthermore, the tickets are entirely generated at the destination, which reduces defence operational cost and makes the tickets route-independent. Tickets are assigned to sending entities Under the parameters tested, the defence is able to reduce the percentage of attack-traffic reaching the victim to 5% (down from 100%) in less than one minute after the beginning of the attack. Nevertheless, the defence does not address the issue posed by multiple clients behind the same public IP address, as is the case with NAT-shielded devices. Since the defence uses the received source IP address as the identifier against which client reputations are maintained, it is unable to distinguish between the reputations of clients that share the same public IP address. This causes the reputations of legitimate clients to be tarnished by those of attacking clients if they utilize the same public IP address. DiDoS entities, on the other hand, are organised into reputor-reputee relationships and are identifiable beyond NAT due to the identifiers reputees embed in packets sent to reputors.

Additionally, since the reputations of clients in FGCD are held at the victim, scalability issues arise with increasing number of clients, as the victim must store the reputations of each client. Furthermore, multiple victims would independently maintain the reputation of every client, which presents the inefficiency of overlapping records in the cases where clients visit multiple victims. This effect is alleviated in DiDoS since, in DiDoS, it is the network where the sender is located that bears responsibility for reputation management and not the multitude of possible downstream destinations as in FGCD.

Lastly, since tickets must be requested and are required before a client can communicate with a server, FGCD is vulnerable to ticket request floods which incur a costly Diffie-Hellman exchange. This is not an issue for DiDoS since tickets are neither required nor requested.

In the paper *Mitigating Application Layer Distributed Denial of Service Attacks via Effective Trust Management* [164], a reputation-based defence called Trust Management Helmet (TMH) that combats session flooding attacks is proposed. TMH identifies clients by a combination of the client IP address and a unique identifier assigned by the victim. This helps to address the issue of distinguishing between clients behind a NAT device. The main metric of distinction used by TMH to distinguish between malicious and legitimate clients is the interval between accesses of a client to the server at various levels of utilized server access-capacity. TMH additionally provides support for collaboration between servers to share client blacklists and trust scores. However, how this mechanism fairs against malicious servers or Sybil attacks is not fully explored. Additionally, TMH provides no mechanisms for the elimination of attack traffic closer to the attack source. Therefore, regardless of the robustness of the claim of the 'lightweight-ness' of the defence enabling the victim to process received licenses at the maximum rate at which the network can send them, the defence is still susceptible to flooding attacks that produce contention at routers further upstream from the (victim) destination. Like DiDoS, TMH offers the ability to distinguish multiple clients from behind a NAT IP address. However, unlike DiDoS, TMH does not offer distributed defence capability meaning attack traffic is only mitigated at the destination host, which is ineffective in mitigating large flooding attacks. Differences between DiDoS and related work from the literature are summarized in Table 7.

	Defence			
Aspect	**Passport** [78]	**FAIR** [80]	**FGCD** [34]	**DiDoS**
Metric/s of Distinction Between Malicious & Legitimate	Invalid or missing MACs and spoofed global IP address	Breached CIR and CBS policies between AS's	Voluminous contribution to attack and response of attacker to congestion	Frequency of involvement in attacks, voluminous contribution to attack, packet rating honesty and spoofed IP address
Distinction granularity	AS	AS	Host	Software
Distributed mitigation	Y	N	N	Y
Inter-AS Spoofing Mitigation	Y	N	Y – except un-ticketed traffic	Y
Intra-AS Spoofing Mitigation	N	N	Y – except un-ticketed traffic	Y
Feedback Process Using Packet Provenance Headers to Identify Malicious Entities	N	N	N	Y
Instantaneous Mitigation[1]	Y	Y	N	Y
Identification Scheme-aware Reputation Maintenance	N	N	N	Y

Table 7: Summary comparison of DiDoS against relevant work

4.2.1.2 Against Design Goals

This section discusses how the DiDoS architecture addresses the design goals presented in section 3.2. The main design goals, which were derived from the literature, were as follows:

1. Flood resistance
2. Distributed attack mitigation
3. Strong metric for distinction
4. Compatibility with existing Internet infrastructure

DiDoS meets the first design goal of flood resistance by implementing the second design goal of distributed attack mitigation. By enabling packets from malicious senders to be dropped at routers, via lower priority queues, DiDoS provides a second line of defence, which allows for mitigation before legitimate packets are dropped in the network at points of contention resulting from the DDoS attack. The first line of defence being the protections on the victim network such as a network firewall. Additionally, since the reputation levels of packets are embedded within each packet header, the possibility of well-informed distributed mitigation is enabled by the architecture. This capability is complementary to the DDoS Open Threat Signalling project commissioned by the IETF [171], which aimed to enable the coordination

[1] An attack need not be detected before mitigation occurs

of DDoS mitigation measures between entities across the Internet using a standard API. The benefits of using DiDoS reputations with DOTS remains an open research avenue.

The primary method DiDoS uses to differentiate between malicious and benign activity is reputation levels embedded on packets in transit. As discussed in section 3.1, two factors make for a strong metric of distinction: a strong correlation with the denial-of-service effect, and a strong correlation with attacker behaviour or activity – as opposed to the behaviour and activity of legitimate senders. In order to maintain the reputation levels, which are used to differentiate between malicious and legitimate senders, DiDoS employs three metrics. The main metric used is the frequency with which a sender is involved in attacks. We believe this to be a strong metric since DDoS attacking agents are more likely to be involved in a DDoS attack than legitimate senders free of bot malware. This is self-evident since attacking agents *cause* the attacks. Furthermore, the fact that an attack happened is strongly correlated with the denial-of-service effect hence the frequency of involvement in attack is a strong metric.

The second metric used to augment the convergence of reputee reputations, is the voluminous contribution to an attack. The metric is proportional to the sending data-rate, which is strongly correlated to the denial-of-service effect. Additionally, since bots initiate requests at a much higher rate than humans, and since flooding DDoS attacks leverage high sending rates, there is some correlation between voluminous contribution to an attack and malice.

The last metric used is the appropriateness with which a sender rates the packets it sends. This metric springs from the DiDoS architecture itself and is used to identify and hold malicious reputees to account. The metric also enables intermediate routers of malicious packets to preserve a good reputation by appropriately marking these packets with a low reputation, thus allowing them to be dropped at points of contention. In this way the metric is somewhat correlated with the denial-of-service effect and it's able to distinguish between malicious reputors who overrate the malicious packets they send.

The last DiDoS design goal of "compatibility with existing Internet infrastructure" was addressed by DiDoS making use of IP header options in order to allow non-DiDoS-compliant routers to forward DiDoS-compliant packets without issue. However, in order to avoid hop limitations in IPv4, due to the per-packet header information necessary to facilitate DiDoS functionality and cryptographic protections, a technique of payload encapsulation is leveraged to extend the number of feedback-enabled hops whilst sustaining the strength of the cryptographic protections offered. This requires that DiDoS provenance section headers only be added by DiDoS reputees with authenticated connection to the reputors, such that non-DiDoS-compliant end nodes do not receive payloads they cannot appropriately unpack. However, these limitations are lifted in IPv6 due to the increased size of optional header space available in IPv6. Both IPv4 and IPv6 DiDoS implementations can offer more than 25 reputation domain hops, which is more than enough, considering the average number of autonomous system (AS) hops is less than 4.

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 3.4.5) 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 ASs in the Internet, research has shown that packets, on average, only traverse 3.9 ASs for IPv4 and 3.5 ASs for IPv6 [80]. Additionally prior work has demonstrated the viability of such in-transit MAC processing [78].

4.2.2 Threat Analysis

This section scrutinizes the DiDoS architecture against the threats arising from the threat model specified in section 3.3. The purpose is to assess how DiDoS addresses the specified threats and to outline any vulnerabilities of the architecture along with their possible impact. The focus is on *marginal vulnerabilities*, which are new vulnerabilities introduced as a result of the architecture. For example, incumbent vulnerabilities, such as the possibility of a compromised router contributing to an attack by dropping all ingress traffic, are not the focus of this threat analysis. This section begins by analysing the threats pertaining to the reputation management system before proceeding to evaluate DiDoS against the wider threats to the wider architecture.

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. DiDoS deliberately introduces no mechanism for a non-malicious entity to actively increase its reputation, which prevents such a mechanism being co-opted by malicious entities. An attacker trying to maximise its attack impact would need to maximise its reputation relative to legitimate senders, which can be done by using less bots for each attack or by attacking less often. However, both of these strategies would incur a cost to the attacker that is proportional to the length of time required for a reputee to recover the reputation score penalty resulting from an attack.

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 packets they sent. In other words, each reputor is made accountable for the reputation level it marked on the packets it transmits. Another

manifestation of flattery would be if a malicious entity could somehow unfairly raise the reputation score that a reputor has for one of its reputees. As with the threat of self-promotion, described earlier, since no mechanism exists for an entity to actively improve reputations, there is no mechanism to exploit in attempts to achieve flattery. The reputation score that a reputor holds for a reputee is only increased by a process call passive periodic accumulation (PPA), described in section 3.4.3.1, which is independent of reputee or third-party actions.

Slander is an attack where the attacker falsely diminishes the reputations of other entities. This can occur for any router or autonomous system lying on the route of a packet. The router may assign a bad reputation level to the packet. However, the impact of this act is not as severe as what else a router along the path of a packet can do without DiDoS deployed (e.g. dropping the packet altogether). As described in section 3.4.4 the impact of assigning a low reputation level to a packet, is an increased likelihood that the packet will be dropped at points of network contention. Furthermore, for this attack to work the malicious router would need to be positioned exactly in the path that a packet takes to reach its destination. If the attacker's goal were to cause the packets of the sender to be dropped, then reducing the reputation level of its packets in transit would be an ineffective way of achieving it.

Lying in the attack feedback process would be another avenue to for attackers to harness in order to effect slander. If an attacker were to fabricate an attack feedback report placing in it packets of clients whose reputation the attacker wishes to tarnish, then the reputors – of the targeted clients – receiving the reports would reduce the targeted client's reputations, hypothetically. However, the protections built into the attack feedback process mitigates against this. Firstly, attacks must be corroborated by and subsequent attack reports signed by a trusted third party, such that fabricated attack reports would be rejected, by all recipient reputors. Secondly, fabricated packets in attack feedback reports would be rejected by recipient reputors since, due to the feedback MAC each reputor embeds in the packet header of every packet it sends, each recipient reputor would be able to verify that it did not send the packets contained in the received feedback report.

An alternative avenue for slander is that a malicious intermediate reputor, situated along the packet-path between the victim of an attack and the legitimate clients of said victim, stores packets sent by the legitimate clients and attempts to maliciously insert them into attack feedback reports such that reputors situated downstream in the feedback path would unduly penalize the legitimate client's reputation. This is mitigated in DiDoS in several ways. Firstly, each reputor checks that the destination address of each packet matches the attack destination specified in the report header, which is signed by a trusted third party. Packets with mismatching destinations are discarded. Secondly, the timestamp on each constituent packet header allows the reputor receiving the attack report to verify that each constituent packet lies within the bounds specified in the report header. Lastly, each report body is signed by a trusted third party where the signature includes the hash value of the report header such that

previously signed report bodies cannot be attached to a new attack report header without detection.

Despite these protections, there exists a possibility for an attacker to maliciously cause a reputor to falsely rate its reputees poorly, if the attacker can manipulate the reputee into sending a lot of data, during an attack, since attack feedback reports (the input by which reputee reputations are diminished) would contain proofs that demonstrate the reputees contribution to an attack. This would be the case for reflection attacks; however, attack spoofing is mitigated by the DiDoS architecture.

Non-participation in the DiDoS architecture can be an attacker's attempt at *evasion* – to escape the repercussions of 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. Non-participation stymies the extent to which DiDoS entities can be held to account. For example, a non-participating AS will not process AFRs destined for its neighbouring AS's. Similarly, a malicious entity may refuse to pass on AFRs in the feedback chain, thus limiting the learning rate of downstream reputees and hampering its own ability to granularly distinguish between malicious and benign packets. Despite the numerous compromised end devices, we believe it to be reasonable to assume that only a few routing reputors, particularly autonomous systems, will be malicious; thus, allowing reputation convergence to still occur and entities to still be held to account for attacks.

Nevertheless, the threat of non-participation does not only exist through malice but is also a consequence of entities not adopting the DiDoS architecture without malice. 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 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 3.4.3.2). 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 (achieved 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 8 and Equation 1, respectively).

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. Another scenario for self-destruction, is an attacker sending attacks via a particular reputor such that the reputor repeatedly appears in the attack feedback path and so has its reputation penalized by its reputors. This is self-destruction since the perpetrating bots would also have their reputation penalized. DiDoS mitigates against this by enabling reputors to receive lesser reputation penalties by appropriately labelling the attack traffic with low reputation levels.

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. Another avenue for sabotage of the DiDoS architecture would be the flooding of the collaborative reputation systems with indiscriminate falsified attack feedback reports, which would cause the reputation levels placed on packets to become unreliable. DiDoS however mitigates against this through the requirement of attack feedback reports to be verified by a trusted third party before dissemination. This is described in section 3.4.3.3.

Violations against the integrity of the hierarchical identification scheme can lead to vulnerabilities in the collaborative reputation management system, such as slander, evasion and sabotage. For example, if a compromised device in a local access network (LAN) is able to impersonate other devices in a LAN, it could tarnish their reputations by launching attacks under their identity. This threat is mitigated against in DiDoS by the transit MAC[1] placed in each packet header sent between the LAN devices (reputees) and the LAN access router (reputor). This MAC enables the reputor to authenticate the sender of a packet by verifying the MAC value, which was created with a secret shared only between the reputor and the reputee. Thus, a compromised device attempting to impersonate another device that is known to the reputor via a different IP address and identifier value, will likely fail since it does not know the secret shared between the other device and the reputor. Another attack vector of the identification scheme is the possibility of a single malicious entity fabricating multiple reputee identities with its reputor. This would enable the malicious entity to attack with impunity (evasion) since, once one of its identifiers is held to account, it could simply continue attacking with the next one, allowing its used identifiers to recover the reputation. This is possible if the malicious reputee uses no or weak authentication mechanisms to create its new identities with its reputor, such as self-signed certificates [172], [173]. This is counteracted in

[1] Message Authentication Code

DiDoS by grouping the reputations of all identifiers created with weak authentication mechanisms. For example, all reputees that have authenticated to the reputor using self-signed certificates would share a single reputation score, such that attacks enacted by one reputee in the group would incur a reputation penalty that is shared by all in the group. However, the identifiers of reputees that were registered with a reputor using means of strong authentication that limits fabricated identifiers, (e.g. manual enrolment or TPM-based authentication in which identities are bound to hardware) would not be placed in reputation groups. This approach eliminates much of the gain a malicious entity can achieve by fabricating multiple identifiers for itself. It also provides incentive for manufacturers and users of end devices to implement and activate means of stronger device authentication to prevent the reputation of their devices from being tarnished by malicious entities and thus foster greater quality of service over the Internet.

An attacker may attempt to leverage passive periodic accumulation to make the reputation levels of their botnet devices similar to that of benign devices. The attacker may do this to charge premium pricing per attack (since they would be more likely to be more effective) and can achieve this by not attacking for a period of time whilst the reputation scores of its own botnet devices rise. This approach in of itself is a cost to the attacker that is proportional to the period of wait time required for the botnet devices to recover their reputation from their previous attack. This cost could be an increase of several orders of magnitude for an attacker that used to attack several times a day, and a required reputation recovery time of multiple days or months. The argument could be made that the attacker charges higher prices, however, the price of attack would need to increase by at least that same cost-increase factor, which would itself act as a disincentive to attack.

The main threat that DiDoS aims to mitigate against is that of DDoS attacks. This is achieved by the transmission of sender reputations along with each packet sent, such that the packets of identified malicious senders can be deprioritized at points of contention in the network. The performance of the DiDoS architecture via its capability to identify malicious senders and its effectiveness in attack mitigation is evaluated in the next section.

4.3 Empirical Validation

4.3.1 Reputation Convergence

4.3.1.1 Setup

This section describes a use-case experiment to investigate the effectiveness of the reputation convergence of the DiDoS architecture. The experiment, which represents a simple scenario for a proof of concept of the DiDoS reputation convergence capabilities, was constructed in C++ and the results are presented in the next section. 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 19.

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[1] (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.

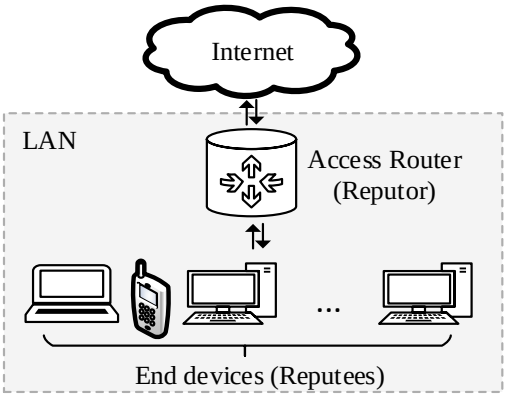


Figure 19: Illustration of use-case simulation scenario of reputor and reputees in LAN

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[1]:

Equation 9

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

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

Since, $Z[x]$ was implemented by averaging the number of attack reports received over the last 3 passive periodic accumulation (PPA) 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[1]:

Equation 10

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

Where the expected number of attacks in a period 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 PPA amount was chosen such that half of the entire range of reputation scores could be traversed in 200 periodic increments (163).

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 equilibrium attack rate Z_E was defined by:

Equation 11

$$Z_E = \frac{1}{T_R} = \frac{\delta_{PA}}{P_B}$$

Where T_R, the "base-penalty recovery time"[2], was chosen to be 30 days and δ_{PA} represents the PPA amount, and was administered, in the simulation environment, daily. Thus, the values for P_B and Z_E are 4890 and 0.0333 respectively. The selection of the values for T_R and PPA affects how quickly reputees lose and recover their reputation. The combination of these speeds creates a trade-off between the mitigatory impact that DiDoS has on DDoS attackers and the ability for previously compromised devices to recover their reputation after being purged. We further discuss this tradeoff and justify the values chosen in section 4.4, whilst the mitigatory impact of the magnitudes of these values is addressed in section 4.2.2.

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

[1] All equation terms are previously defined in section 3.4.3.
[2] The time taken for a reputee to recover its reputation after a single reputation penalty of the base-penalty amount.

normally distributed and the simulations were repeated up to 100 times for generalized inferences.

4.3.1.2 Results

This section presents the results obtained and various key findings. However, further discussion on the experiment and results can be found in section 4.4.

Figure 20 (**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 20 (**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. 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 20 (**c**).

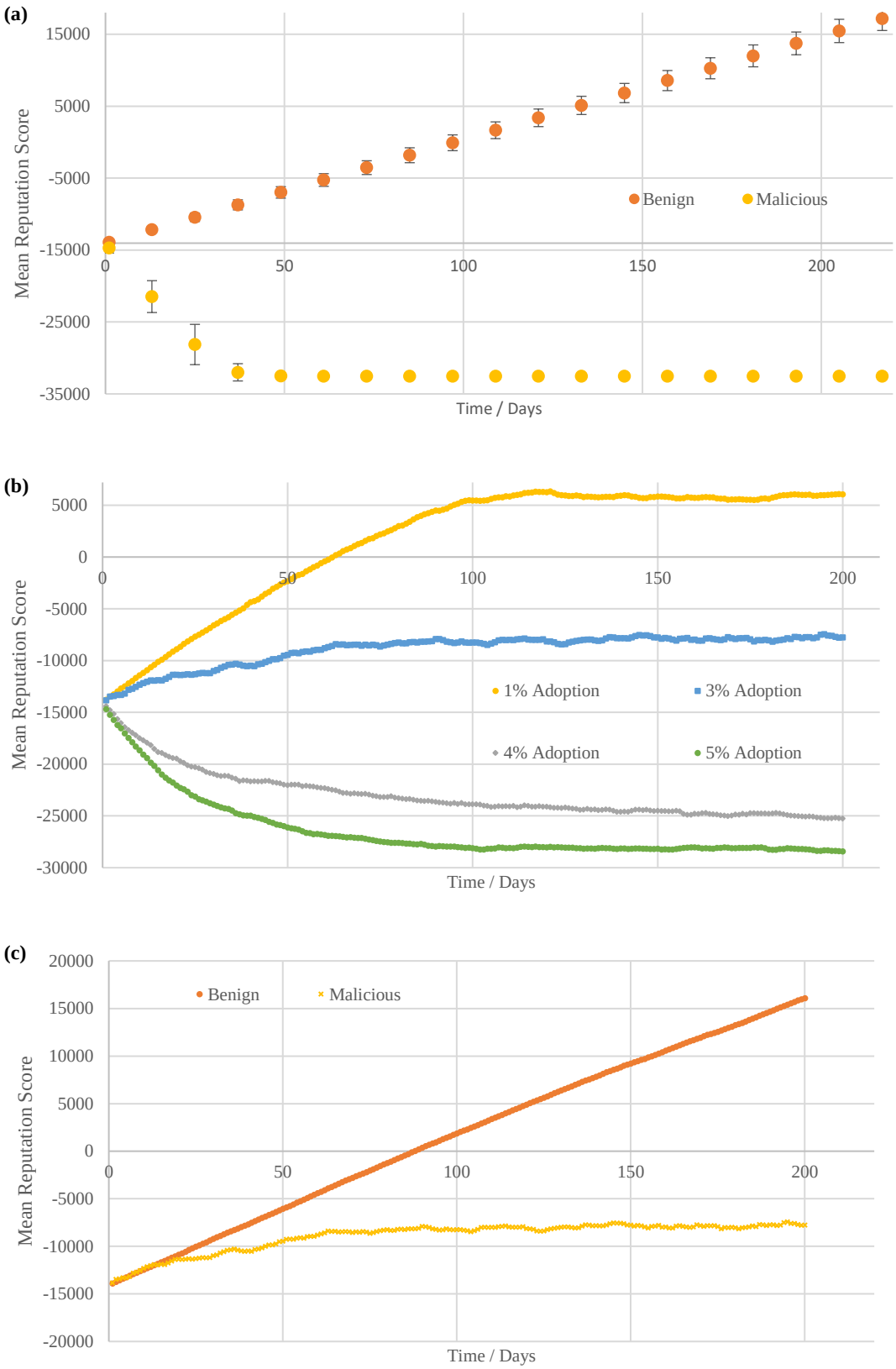


Figure 20: Mean reputation scores a) at 100% DiDoS adoption, b) [of malicious device reputations] at various adoption percentages; c) at 3% DiDoS adoption

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 20 (**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 21), 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 AIFs were almost the same at ~1.08 per day (Figure 22).

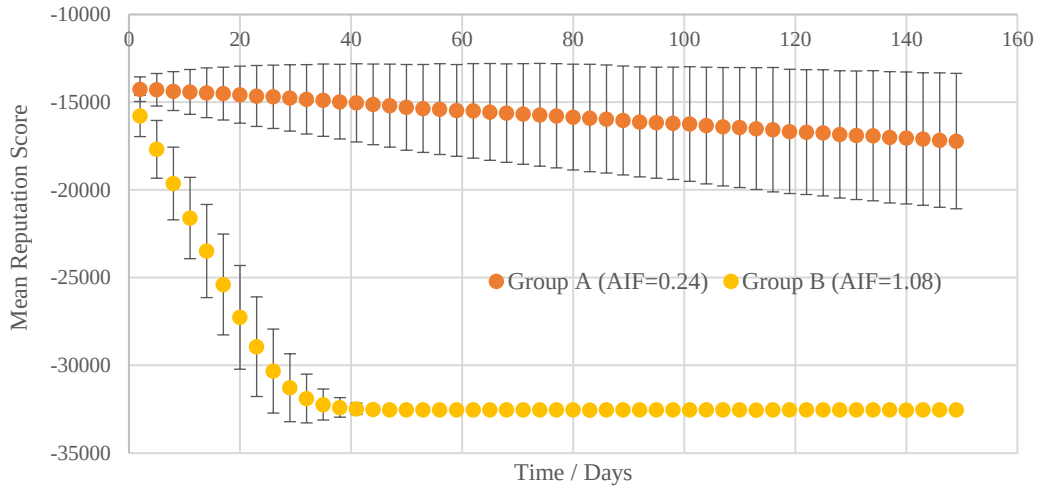


Figure 21: Group reputation scores of devices in a simulated LAN with specified AIFs

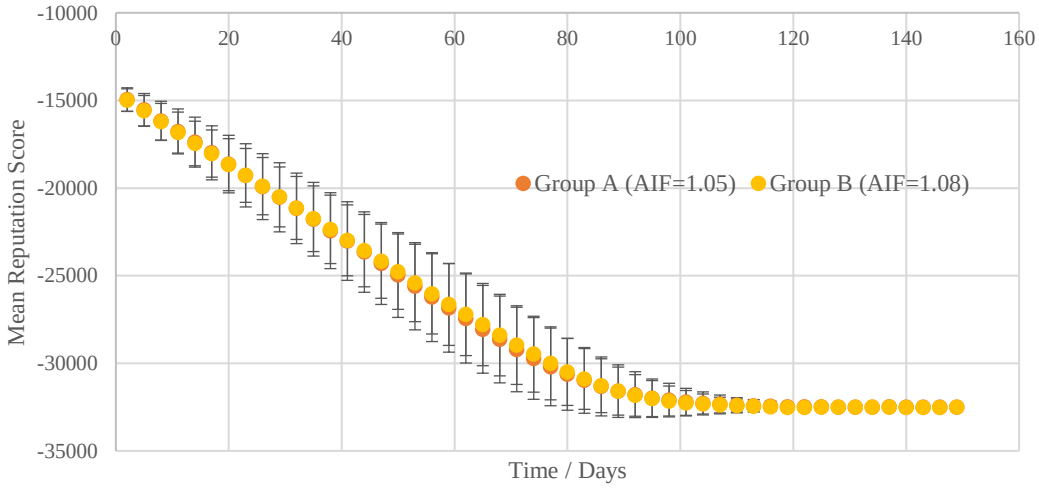


Figure 22: Group reputation scores of devices in a simulated LAN with almost matching AIFs

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.

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.

4.3.2 Mitigation Performance

4.3.2.1 Setup

This section describes an experiment to validate the effectiveness of the DiDoS architecture in mitigating DDoS flood attacks. Also explored is how the effectiveness of the defence varies with the time the reputations have had to converge and the adoption percentage of the architecture. A summary of the simulation parameters and environment with various justifications is presented in Table 8, and further discussion of assumptions in included in section 4.3.2.1.

These experiments extend the reputation convergence simulations of the previous section to observe the mitigatory impact of the application of the DiDoS reputations. As such, rather than a single LAN, in which DiDoS is deployed, the experiment environment consists of a mesh network to which numerous LANs are connected such that distributed denial of service attacks can be effected. Both the client-side and server-side networks were connected o the core nodes by 100 Mbps links. This is illustrated in Figure 23, where the depicted client-side network consists of numerous LANs.

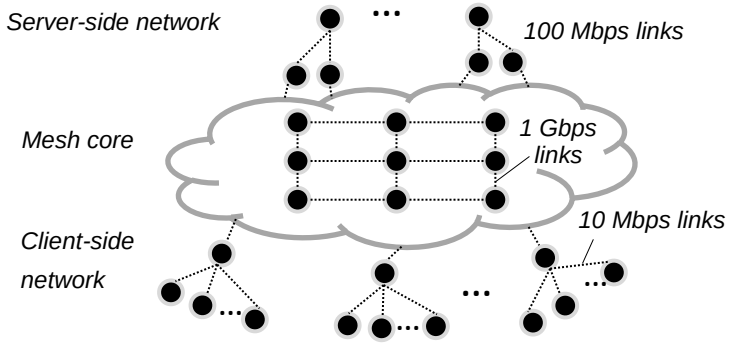


Figure 23: Illustration of the experiment network environment

The network environment consisted of distributed legitimate traffic communicating with several servers and consuming 15% of their network bandwidth. The 15% utilization value was chosen since it lies in the middle of the range of typical real-world utilization levels [174]. An echo application was installed on each server to respond to each received packet with an identical packet sent to its source. The senders of the legitimate traffic were distributed around the network and so were the bots responsible for launching the DDoS attacks.

Although experiments running on real hardware testbeds can offer results most representative of real-life network environments – should the testbeds accurately replicate said environments – simulation was chosen as the means for validating the DiDoS architecture for several reasons. Firstly, simulation enables speedy and low-cost exploration of the effects of DDoS attacks over a several hundred node internetwork, compared to using physical real hardware. Secondly, simulation affords a great amount of versatility and control over the experiment environment. For example, entire topologies, link capacities and queue sizes can be altered by the execution of a script. This mode and level of control enabled the automation of the thousands of simulations carried out to produce the results of this section.

The software chosen for the simulations was NS-3 [175], a discrete-event open-source network simulator that is developed and actively maintained by the research community. One of the main advantages of NS-3 is its extensibility, which allowed the implementation of the custom DiDoS functionality. Further justification of the use of NS-3 can be found in Appendix A.1.

Multiple NS-3 simulations were orchestrated via BASH [176] scripts submitted to a SLURM [177] hypervisor. These ran on the ARC computing cluster [178], taking thousands of CPU hours to complete.

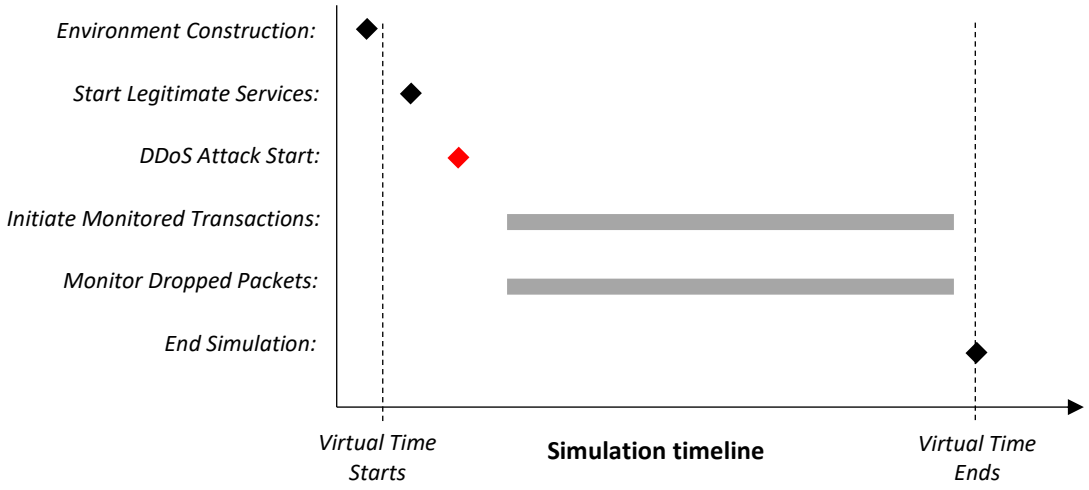


Figure 24: Gantt Chart of Simulation Timeline

Each simulation consisted of a DDoS attack executed on a victim server. The sequence of events in each simulation is illustrated in Figure 24.

Part of the environment construction phase involves the creation and connection of the 3 network segments (core, client-side networks, and server-side networks), which results in multiple LANs and nodes within each LAN. Legitimate clients are then added, as custom NS-3 *Apps* to these nodes as needed, in order to generate the background benign traffic. NS-3 apps representing malicious agents are then installed on nodes of these LANs in a globally

random and distributed manner, as needed – according to the specified simulation attack size. The mean data rate and packet size of each app was randomly generate – per app – according to the normal distributions specified in , and assigned to the apps themselves. Thus the total number of apps required to achieve 15% server utilizations or the required attack strength varied across simulations.

DiDoS compliance (of reputees and reputors) was probabilistically assigned according to the adoption percentage specification of the particular simulation. For each LAN with access router reputors that were designated as DiDoS compliant, a pre-simulation was executed to determine the reputation scores of each reputee. These pre-simulations used the same DiDoS parameters as specified in section 4.3.1. including that of the base-penalty recovery time.

The greater the base-penalty recovery time, the longer it will take both benign and malicious devices to recover from a reputation. This is a tradeoff and is addressed in the discussion of section 4.4. For the purposes of this simulation 30 days was chosen since it is a factor large enough to cause meaningful financial impact[1] to commercial botnets, whilst being small enough to allow benign devices, that were incidentally involved in an attack, to recover their reputation within a reasonable time frame. It is important to note that the reputation algorithm tapers the actual reputation penalty based on other factors (like volumetric contribution and frequency of attacks), so that malicious devices would take longer to recover their reputation benign devices would take relatively shorter.

After the attack is launched in the simulation, a series of transactions of random start time and finite duration (30s) was launched and the number of dropped packets (belonging to the monitored transactions) were tracked. The simulation ended at the earlier of the virtual times when all the finite transactions completed their transactions or after 5 minutes of virtual time. Performance of the defence was measured by the reduction in the number of transaction packets dropped from when the defence was present compared to that when no defence was present.

A three-dimensional matrix of simulations was carried out, where each matrix axis represented a range of values of a particular variable, such that each element of the matrix denoted a unique combination of three variable values input into the particular simulation of that element. The matrix rows (the first axis) determined values of a *convergence time* variable which specified the length of time each reputor had to maintain the reputation of its reputees. The matrix columns (the second axis) varied the adoption percentage of the DiDoS architecture. Despite the variation in adoption percentage, the victim network was always assumed to be DiDoS compliant. The last axis of matrix input values was the simulation seed,

[1] As discussed in section 4.2.2, an attacker may attempt to game the system by wating between attacks, so that its attacking agents can recover their reputation levels. However, the longer the wait, the higher the cost.

which enabled the repetition of the simulation under different random number generation sequences.

The reputation scores S stored by the reputors were converted into reputation levels l embedded in the packets using the below equation:

Equation 12

$$l = 1 + \left\lfloor \frac{S - S_{min}}{((S_{max} - S_{min})/L_N)} \right\rfloor - \left\lfloor \frac{S}{S_{max}} \right\rfloor$$

Where $S_{min} \leq S \leq S_{max}$ such that S_{max} and S_{min} represent the maximum and the minimum reputation scores, which were 32767 and -32768, respectively. L_N indicates the number of intervals of reputation levels, which was 7 in these experiments. The levels in the above equation range from 1 to L_N in increments of 1. The final term in Equation 12 ensures that the level value does not exceed number of levels L_N when $S = S_{max}$.

The DiDoS compliant routers in the test environment implemented priority queuing [179], [180] in which packets are placed into queues of different priority, according to their reptation level, such that higher reputed packets are placed in higher priority queues and are routed with higher priority. This queuing algorithm was chosen since it supported the DiDoS mitigation strategy of preventing a lesser-reputed packet from being routed at the expense of a higher-reputed packet, whilst also allowing lesser-reputed packets to be routed. Three levels of priority queues were used: low, medium and high. Packets whose reputation level was less than or equal to 2 were routed through the low priority queue2. Packets whose reputation level was between three and four (inclusive) were routed through the medium priority queues. Lastly, packets whose reputation levels were five and above were routed through the high priority queues.

The test environment topology consisted of 100 LANs and over 700 nodes, with 1 Gbps core links, 100 Mbps server access links and 10 Mbps client access links. Simulations (of just 5 minutes of virtual time) where the server access links were 1Gbps took up to 4 days to complete, whereas with 100 Mbps server access links, the simulation time was less than 4 hours. Hence the decision was made to proceed with 100 Mbps server links since numerous simulations were required due to the many elements of the simulation matrix (exploring a multidimensional space of parameters) described earlier. A summary of the simulation parameters and environment with justification is presented in Table 8.

Aspect	Specification	Reason
Topology	Mixed: (mesh, star, ring) of >700 nodes	A mesh core, ring server-side, and star client access segments were used to represent different aspects of a scaled-down internetwork with their relative degrees of interconnectivity. The network segmentation allowed the modelling of DiDoS in a more representative way.
# of client access networks	100	Multiple servers were required to simulate background legitimate traffic and even more clients were required to generate it and also host attacker agents. Grid-mesh core nodes quadratically increased simulation time, although 3x3 was sufficient to generate diverse routes between nodes. There was a trade-off to make these values as large as possible whilst achieving feasible simulation times.
# of server access networks	10	
# of core nodes	9	
Client-side link bandwidth	10 Mbps	To a achieve a (timewise) feasible simulation these bandwidths were chosen at a scale smaller many real deployments but relative to each other such that the core bandwidth was the largest and the client-side bandwidth the smallest, to mimic real-world internetworks.
Server-side link bandwidth	100 Mbps	
Core link bandwidth	1 Gbps	
Client-side link delays	3 ms	Link delays, in conjunction with link bandwidths and number of nodes, were chosen to balance increasing scale with feasible simulation times. The delays were relatively chosen to reflect a higher delay at the core links versus a lowest delay at the client-side links. This was to reflect the relative physical distances between nodes at these points in the network.
Server-side link delays	5 ms	
Core link delays	7 ms	
Client-side queue size	10 packets	Queue sizes were chosen according to the rule of thumb and formula $B = (RTT_{overline} \times C)\sqrt{n}$ described in [181], where $RTT_{overline}$ is the average round-trip time of a flow passing across the link, C is the data rate of the link, and n is the number of flows on the link. The resulting queue-size was converted into packets using the mean packet size and adjusted in consideration of the relative link rates and expected router sizes of the different network segments.
Server-side queue size	20 packets	
Core queue size	80 packets	
Server utilization	15%	Server utilization was chosen according to typical real world utilization percentages. [174]
Legitimate client UDP data rate	N[1]{0.3, 0.03, 5, 0.1} Mbps	The mean data rates and packet sizes of the UDP and TCP traffic were chosen to align with that gleaned from the literature [174], [182], [183]. Standard deviations were chosen to avoid frequent significant deviations from the mean values. The lower the client data-rates, the more clients required to achieve 15% server utilizations, so the minimum client data rate was chosen to avoid multiple legitimate clients (that communicate to the same server) having to reside on the same node. The maximum client data rate was chosen to reduce the chances of legitimate-traffic-caused congestion on the 10 Mbps client-side links.
Legitimate client UDP packet size	N{613, 10, 1500, 200} B[2]	
Legitimate client TCP data rate	N{0.5, 0.1, 5, 0.1} Mbps	
Legitimate client TCP packet size	N{613, 10, 1500, 200} B	

[1] The notation N{a, b, c, d} represents a normal distribution with specified mean, minimum, maximum and standard deviation, respectively.
[2] B – Bytes

DDoS attack client data rate	N[1]{0.4, 0.1, 5, 0.2} Mbps	Attack traffic was set to have a higher mean, and minimum data rate to achieve significant attack strengths using fewer nodes and to mimic reality. The attack traffic was set to mimic that of the benign traffic. All attack traffic was UDP to maximize the impact by not responding to congestion.
DDoS attack client packet size	N{613, 10, 1500, 200} B	

Table 8: Summary of simulation parameters and environment with reasons

4.3.2.2 Results

This section describes the results obtained and key findings. Further discussion on the experiment, results and assumptions can be found in section 4.4.

As illustrated in Figure 25, with sufficient learning time and full deployment, DiDoS is able to leverage the learned reputations to effectively reduce the number of packets of legitimate clients that were lost to zero. The y-axis represents the proportion of the legitimate clients that were dropped during an attack. This is expressed as a percentage of the total number of packets sent per transaction. In each simulation the victim was subjected to a UDP flood DDoS attack corresponding to 300% of victim capacity. When no defence was deployed, the proportion of packets of legitimate transactions lost was 85%. When DiDoS was deployed and the number days for the reputors to learn the reputation of reputees was low, this value remained unchanged. This is because in the DiDoS architecture, the reputation scores of all reputees begin at one corresponding to a low reputation level (-14,045). Thus, in the implemented priority queuing algorithm consisting of 3 queues (of priority ranked low medium and high) per implementing device degree, all packets, both benign and malicious, passed through the same lowest priority queues – since the reputors did not receive enough attack feedback reports to distinguish between benign and malicious senders. This initial trend, of unchanging proportion of dropped transaction packets, ceases once the number of days of reputation learning reaches 20. This is because passive periodic accumulation (PPA) over 20 days is the minimum required to elevate the reputation scores of non-malicious clients such that they can be distinguished from the malicious clients, whose reputation scores typically get worse (see Figure 20 a).

We observe that between 20 and 42 days of reputation learning, the mean of the number of packets trends downwards until it reaches zero. During this same period the standard deviation of each measurement is particularly high. This is because the performance is limited by the number of benign devices whose reputation scores have not exceeded the threshold score[2] (required to classify them in a separate queue to the malicious clients) and reaching this

[1] The notation N{a, b, c, d} represents a normal distribution with specified mean, minimum, maximum and standard deviation, respectively.

[2] This threshold score value is -4,682. The reputors of reputees with reputation scores above this value, mark the packets of these reputees with a *medium* or higher reputation level, which in turn causes them to be routed in medium or high priority queues. a

threshold is limited by a random function specifying whether a benign device was involved in an attack.

From 42 days of reputation learning time and upwards, we observe that the proportion of packets of non-malicious clients that were dropped remains at zero with no standard deviation. This is because from this point, enough PPA has occurred boosting the reputation scores of the non-malicious reputees such that the maximum number of reputation penalties meted on any non-malicious reputee does not reduce its reputation score to below the threshold score that separates packets between the low priority queues and the medium priority queues. Therefore, reputation level separation is achieved between the packet of malicious and non-malicious senders.

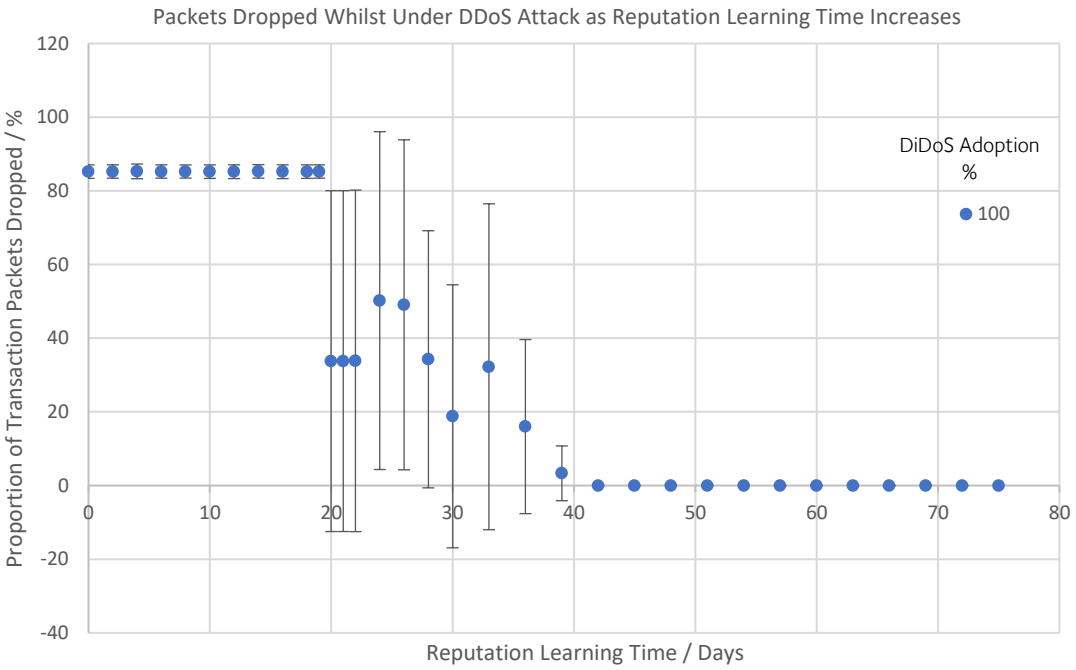


Figure 25: Graph depicting reduction in packet loss as reputation learning time increases

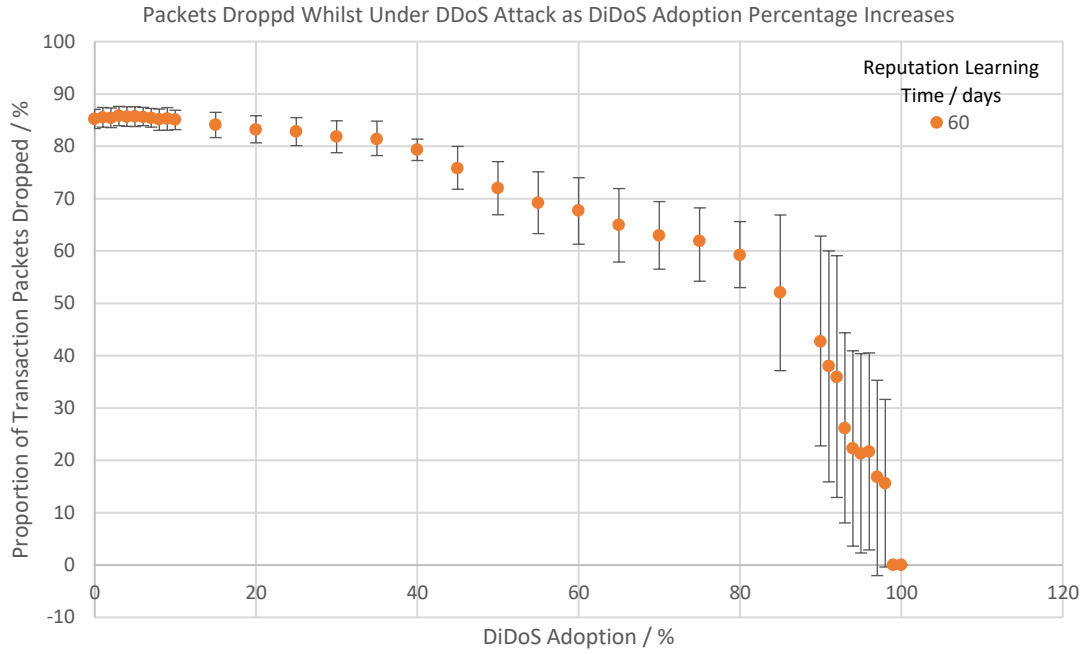


Figure 26: Graph depicting reduction in packet loss as DiDoS adoption percentage increases

The increase in attack mitigation of DiDoS as the adoption percentage increases is illustrated in Figure 26. The reputation learning time depicted is 60 days, meaning that the reputations of malicious and non-malicious senders have already diverged to the point that their packets travel via different priority queues, as explored previously. As shown, the improvement in mitigation is non-linear, with similar adoption increments producing increasing packet-loss mitigations as the overall adoption percentage nears 100. This can be broadly understood by the recognition of two interacting phenomena. The first is considering that the adoption percentage allows the malicious packets, sent from LANs within the adoption percentage, to be known and appropriately filtered at points of contention. At a certain adoption percentage, the amount of traffic generated by attackers outside of the adoption percentage is insufficient to cause contention. This would suggest that at a certain adoption percentage (well before 100%) the proportions of legitimate transaction packets dropped under the attack would equal zero. However, this is not the case due to the second phenomenon. As the malicious senders to the victim server are distributed across the simulation environment, so is the case for the non-malicious clients of the victim server. Should any of these clients fall within a LAN that has not adopted the DiDoS defence, the reputation level of the packets of these benign clients will be low and hence these packets will traverse low priority queues. As a result, the packets of these clients will still need to contend with the attack packets, hence the proportion of transaction packets dropped does not reach zero before 100% adoption of DiDoS.

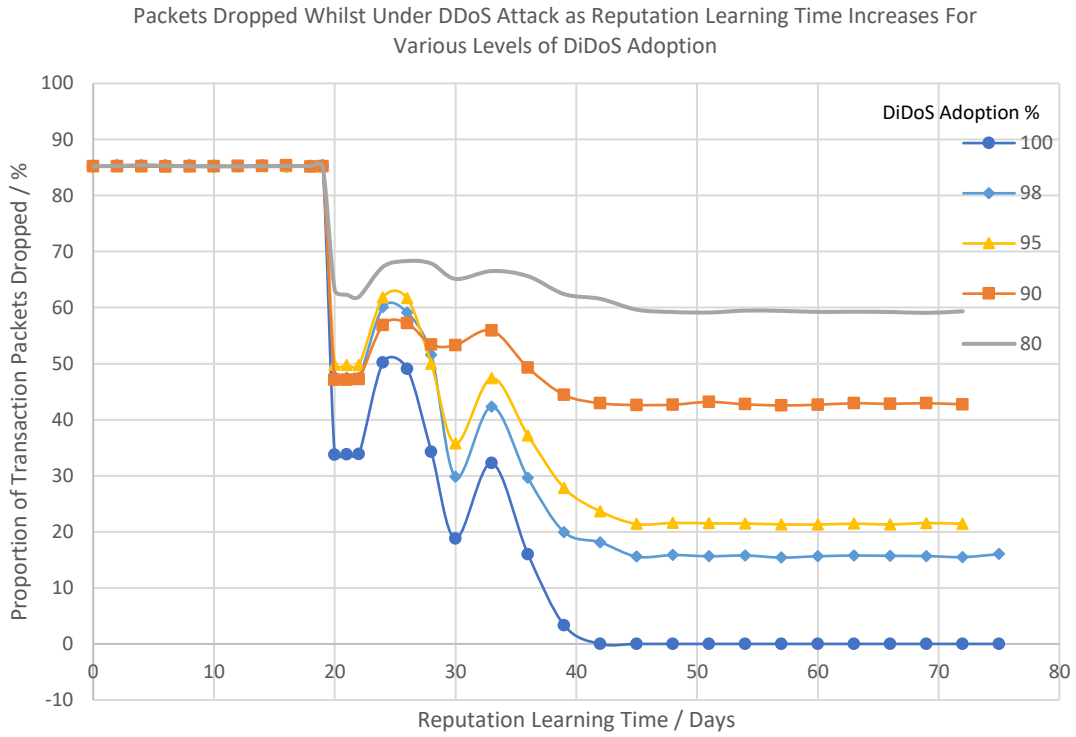


Figure 27: Graph depicting reduction in packet loss as reputation learning time increases for various percentages of DiDoS Adoption

The results shown in Figure 27 extend that of Figure 25 by illustrating additional curves representing different adoption percentages. As shown, the defence performance (measured by the reduction in dropped packets during attack) reduces as the adoption percentage reduces. An adoption percentage of 80% is able to reduce the proportion of non-malicious packets dropped during attack by 30%[1]. This means a change 80% increase in adoption produces just a 30% increase in performance, which can be characterized as an adoption performance change rate (APCR) [2] of 0.379 per additional adoption percentage (30/80). Subsequently, increasing the adoption percentage by 10% (to 90%) produces an additional 28% reduction in dropped packets (from 59.3% to 42.7%) representing an APCR of 2.8. A marginal 5% increase in adoption percentage (to 95%) produces a significant return on performance with an APCR of 4.3.

The distinct oscillation of the gradients of the curve between 20 and 42 days learning of Figure 25 initially appears as an anomalous pattern. However, this phenomenon is consistently repeated in the curves of Figure 27. It is important to note the large standard deviations of each point in this range (a complete set of these results and their standard deviations can be

[1] The fall from 85.2% lost packets to 59.3% is a reduction of 30.4%
[2] APCR is calculated by dividing the change in performance by the change in adoption percentage over a specified adoption range – in this case ((30-0)/(80-0)).

found in Appendix C). This phenomenon can be explained as the probabilistic manifestation[1] of a finite[2] bounded random walk[3] within a one-dimensional plane[4] with directional asymmetry of step probability[5] and size[6]. The different probabilities manifest themselves through different mean values of dropped transaction packets depicted by each plot point on the graph.

Figure 28 illustrates the wider trend of Figure 26 by depicting multiple curves representing different reputation learning times. The plot exhibits a distinct pattern of divergence between curves corresponding to learning times of 19 days and under and the curves corresponding to learning times of 20 and above. Since 20 days was the minimum learning time from which the packets of non-malicious senders were able to be prioritized over those of malicious senders, no attack mitigation occurred for reputation learning times of less than 20 days. Thus, all the curves corresponding to reputation learning times of less than 20 days (not all of which are illustrated in Figure 28), exhibited a straight line, indicative of absent attack mitigation. Once the reputation learning time exceeded 19 days, the mitigation effect of the DiDoS architecture increased (indicated by reductions in packet loss of the transactions of non-malicious clients). The trends are similar, and the standard deviations are high, however, the lowest mean packet loss exhibited for each curve (occurring at an adoption percentage of 100%) ranges from 0% to 33%.

[1] A probabilistic manifestation since it is the manifestation of a probability distribution (of the likelihood of a single reputee exceeding the reputation score threshold) that is realized and averaged over multiple reputees.
[2] Finite because the number of days of reputation learning is finite.
[3] Where "steps" of the walk refers to passive periodic accumulation (forward steps) and reputation penalty (backward steps) of reputee reputations.
[4] The single dimension is the reputation score of a reputee, whose possible values can be perceived on a line.
[5] Directional asymmetry of probability means that the chances of a reputation increase is different from the chances of a reputation decrease.
[6] Asymmetric step size refers to the value of passive periodic accumulation being smaller than the reputation penalty on a reputee over a single day.

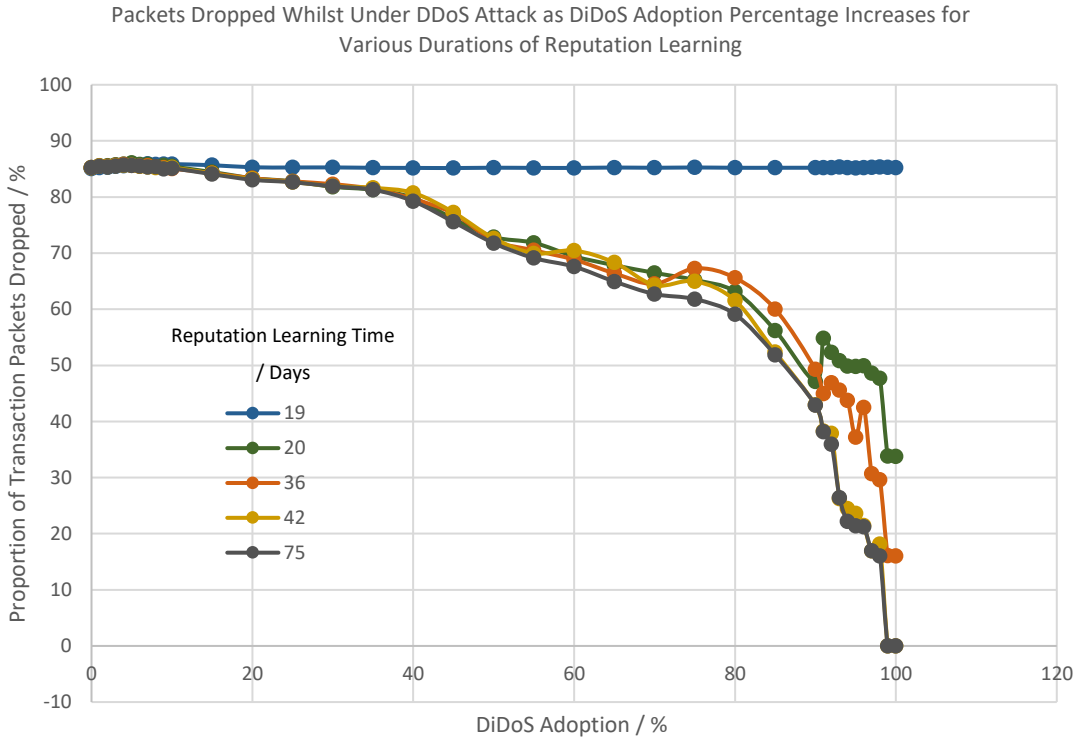


Figure 28: Graph depicting reduction in packet loss as DiDoS adoption percentage increases for various durations of reputation learning

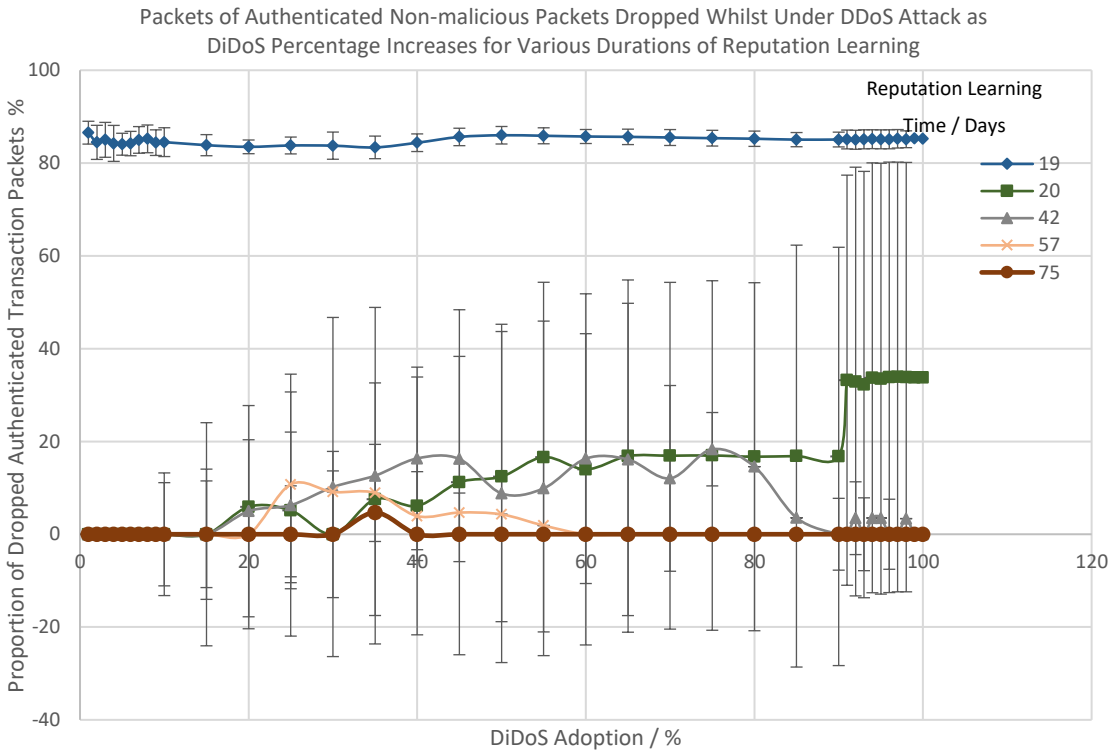


Figure 29: Graph depicting reduction in authenticated non-malicious packet loss as DiDoS adoption percentage increases for various durations of reputation learning

The results detailed so far have presented the performance of DiDoS by the proportion of dropped packets of all clients of the victim server, whether they are authenticated or not. However, Figure 29 presents the proportion of dropped packets belonging to authenticated clients only. This gives an indication of the benefit the client of a source network can receive by adopting DiDoS despite the lack of adoption globally. We observe that as the when the number of days learning is 19 or below, no reduction in packets dropped under DDoS attack occurs. This is because a minimum of 20 days is required for the PPA to cause the reputation scores of clients to exceed a threshold score that would cause their packets to be placed in a higher priority queue. Thus, when the number of days learning reaches 20 an immediate elimination of the packets dropped is seen for adoption percentages as low as 1%. This is because the packets of the few authenticated clients have their reputations passed on as reputation levels which enable them to travers higher priority queues at the victim network. However, once the adoption percentage reaches 20%, the proportion of dropped packets begins to rise, suggesting the onset of another factor for consideration. This phenomenon is due to the fact that as the adoption percentage begins to rise, the probability that a non-malicious device will incidentally be involved in an attack also rises to the point that the non-malicious clients incur reputation penalties which drag their reputation score below the threshold for differentiation between malicious and non-malicious reputees. This effect increases as the adoption percentage increases. However, by increasing the time that the reputors spend developing the reputations of their reputees, the reduction in packet loss increases, since the reputations of the non-malicious clients have a longer period of time to trend upwards. When the reputation learning time is 75 days, the curve remains largely at zero packets of authenticated non-malicious packets dropped, for the adoption range plotted. We observe that at 100% adoption and the same reputation learning time the mean proportion of packets dropped (the point on the curve) is the same values in both Figure 28 and Figure 29. This is because at 100% DiDoS adoption the proportion of authenticated non-malicious clients and unauthenticated non-malicious clients are the same.

4.4 Discussion

The design assessment section of this evaluation chapter compared the defence to some prior work that inspired the design of this defence. Also included in the assessment was the analysis of various threats against the DiDoS architecture.

These results of the empirical validation have shown that the convergence of the reputation algorithm is demonstrated to produce reputation differentiation between malicious and authenticated non-malicious senders at single-digit adoption percentages (see section 4.3.1.2). However, to produce significant overall DDoS mitigation[1], DiDoS adoption must be above

[1] "Overall DDoS mitigation" here encompasses the mitigation effects the defence has on the packets of both authenticated and unauthenticated non-malicious reputees.

80% and for this mitigation to be relatively stable[1] the reputors must have had over 40 days to learn to distinguish the reputations between benign and malicious reputees (Figure 27). However, when considering the packets dropped from authenticated non-malicious clients only, significant reductions in dropped packets occurred for adoption percentages as low as 1%, which we believe is due to the high standard deviation of the reputations scores of malicious clients.

The period of time required (or "differentiation period") for DiDoS reputors to be able to distinguish between malicious and benign reputations is dependent on:

- the adoption percentage of the DiDoS architecture,
- the magnitude of the difference between the attack involvement frequency (AIF) of the benign and malicious devices, and
- the values of the parameters of the DiDoS algorithm such as the Passive Period Accumulation amount and the base-penalty recovery time.

These factors dictate the differentiation period by affecting the rates of change of the reputee reputation scores, which we can observe (for changes in the adoption percentage and AIF factors) in Figure 20 b), Figure 21, and Figure 22.

Although the deviation between the mean reputation scores of malicious and benign devices begins from day 1, the reason that 40 days was specified as the time period required to distinguish between benign and malicious reputees was that that was the time it took the benign reputees to move up a reputation **level**, thus allowing them to be routed through higher priority queues. Thus, the differentiation time would have been longer if the Passive Periodic Accumulation amount had been smaller or applied less frequently, or if the base-penalty recovery time had been much smaller.

For a global DDoS defence deployed on the Internet that would be expected to last decades, we believe that 40 days is a reasonable and relatively small timeframe to achieve distinction between malicious and benign devices. Furthermore, for new benign devices that would be added as reputees to existing reputors, we believe that 40 days is not an unreasonable amount of time to wait before the newly added device increases one reputation level to become more trusted. This is because, if the newly added device was not benign but was malicious, its attacks would be less impactful for that period, which we believe is sufficiently long to cause significant financial impact to DDoS botmaster who would need to wait before attacking effectively. However, a drawback is that for this period, a newly added benign device would have a higher chance of being drop during an attack. Hence this is a trade-off that can be influenced by the selection of the DiDoS parameter values.

[1] Stable, here, is used with respect to the standard deviation

In a similar vein, a trade-off also exists in the size of the base-penalty recovery time, since the greater this value, the smaller the longer it will take malicious reputees to recover their reputations but also the longer it will take benign reputees to recover their reputation. Increasing the time taken for a reputee to recover from an attack penalty helps increase the mitigatory impact of DiDoS since attackers would need to wait longer to attack effectively, which we describe in section 4.2.2. However, it would also mean that benign devices would take longer to distinguish their reputation levels, should they be incidentally caught in an attack. We believe the used base-penalty recovery time value of 30 is a reasonably large enough factor of cost increase to disrupt commercial DDoS botnets that try to game the system.

Further advanced economic analysis could be undertaken to explore the optimum values for these trade-offs, however, such work is out of the scope of this research project. We believe that the values used here are sufficient to demonstrate the effectiveness of the DiDoS architecture and have shown that the optimal trade-off points (once known) can be targeted by adjusting the DiDoS parameters.

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[184]. Another study monitoring a collection of botnets was able to detect 406 DDoS attacks, launched in a period of 151 days [185].

Various assumptions underpin the results of simulation experiments, such as the adequacy of the scale of the simulated NS-3 network, and the assumption of normal distribution about a mean value for various simulation aspects such as client sender data rates and packet sizes. The scale of the NS-3 simulations was a limitation of this experimentation because the target deployment environment is the Internet, which is many orders of magnitude larger than what was simulated (with respect to nodes, connections, flows and bandwidths etc.). Simulating the entire Internet is infeasible due to lack of knowledge of all aspects of the Internet and lack of time and computing resources to capture and execute such a large-scale simulation. Therefore, simplifications must be made. The challenge, from a research perspective is knowing how adequate the representativeness of the model is, such that any conclusions derived from the simulation would likely hold, if carried out in the target real-world environment. Discussion and justification of the simulation approach and the simulation

environment are already presented in sections 4.3.2.1. In this section, however, we address the question of adequacy and consider possibilities for improvement.

From the perspective of adequacy, despite being smaller in scale, we believe the NS-3 environment sufficiently captured the essential elements of the in-scope flooding DDoS attacks. Perhaps the most crucial of these elements is the dropping of excess packets at points of network congestion. The environment didn't, for example, simulate server RAM or CPU usage, as application layer attacks were out of scope (see sections 3.2 and 3.3). However, topology, packets, link delays, and router queues, were all modelled in the NS-3 environment.

Concerning the assumptions of normal distribution, we believe this was reasonable since it captured a variation in packet sizes and data rates that are present in real-world networks and helped produce more sophisticated network jitter. Two alternatives to this assumption would have been to have no distribution at all (e.g. the same packet size for all packets) or uniform random distribution across a range. The former would have been less representative of real traffic and the latter would, undesirably, produce values that lie close to the extremes of the ranges more often. Future work could explore the representability of the normal distributions, by analysing real-world packet traces from multiple networks, multiple services and combinations, and over multiple times of day. However, we believe the most important aspect of the background traffic, (the mean data rates, packet sizes and resulting bandwidth utilizations on the simulated servers) were captured and enough to produce sufficiently representative results.

Another assumption of this experiment is the operational impact of the DiDoS overhead, which stems from the message authentication code (MAC) checks occurring on DiDoS packets in transit. The practical viability of real-time MAC verification on routing hardware has already been carried out in [78], [186], [187]. A full implementation of DiDoS processing on actual router hardware would be required to determine whether the overheads are acceptable or not, however, we believe the data and processing overheads of DiDoS would not largely deviate from [78], which is architecturally similar to DiDoS. Hardware implementation of the DiDoS architecture is a direction for future work.

DiDoS is a granular reputation-based DDoS mitigation architecture, where the reputations of clients are transported within the packets they send. As such, DiDoS may be complementary to other DDoS mitigation initiatives such as DDoS Open Threat Signalling (DOTS) [171], which is an IETF initiative aimed at enabling different organizations to coordinate DDoS attack mitigation responses through upstream signalling to request throttling of various flows, for example. In such a system, the reputations produced by the DiDoS architecture could be leveraged to effect distributed mitigation with fewer false positives, since benign clients would be more likely to have a higher reputation and thus their packets would be less likely to be dropped. Additionally, the value of DiDoS reputations could also be leveraged by application-layer defences that attempt to reduce the memory and processing resources

consumed by an attack [188]. For example, reputation levels on the network layer could be applied to prioritize requests or free up memory allocations. Additionally, the reputation levels on DiDoS packets could form inputs to various schemes, for the potential detection of DDoS attacks.

There are many potential avenues for future work concerning DiDoS, this includes the exploration of new design features and the improvement or investigation of existing features. It may be of value to explore other faster algorithms for the reputation level adjustment of packets from peer reputors, possibly using a lookup table instead.

The body of work so far has modelled reputors as a single logical entity, however in practice private network or autonomous system (AS) reputors may typically be a collection of nodes. Therefore, a valuable avenue for future work is a distributed algorithm for coordinating the different exits (edge nodes) of a reputation domain, such that a reputee having access to different degrees could be marked with a consistent reputation level despite the degree it chooses to forward traffic.

A related direction is the development of a way for the reputors, of reputees that have multiple reputors, to share feedback concerning their reputees to avoid them leveraging multiple egress reputors as a way of reducing reputation score degradation. Such a functionality would need to be designed in such a way as to prevent malicious or dishonest reputors from exploiting the ability to influence the reputations of the reputees of neighbouring reputors.

In DiDoS reputors estimate the number of malicious reputees that they have in order to direct the degradation of their reputee reputations accordingly. But what if these estimates could be dynamically and collaboratively maintained across reputors? For example, the reputation scores of reputees could be set to target an average reputation value, and possible standard deviation, such that actual scores experienced are a tapered curve being defined by a target average. This target average value could be maintained by a gossip mechanism in which reputor peers share information[1] about themselves and their neighbours so reputors can collaboratively determine what their target average should be. The exploration of greater coordination between peer reputors is a possible avenue for future work.

The DiDoS attack feedback process has the potential to direct feedback across a chain of reputors from AS to the end client, whose reputees could be pieces of software or services running on the device. Future work could explore the possibility of techniques to cryptographically bind a source application to the operating system of a device to achieve properties such as non-repudiation, such that it may be possible to identify malware or trojan horses through the DiDoS attack feedback process.

[1] Information such as the number of attacks they've experienced and the number of reputees they have and other information.

Chapter 5: CED3 Design

5.1 Introduction

As the literature indicates, since the first DDoS attack tools were discovered and documented in 1998 [9], [60], a multitude of papers have been written identifying new attacks and describing all manners of defensive techniques. However, the threat of DDoS remains prevalent today.

This can be attributed to many reasons observed in the literature, such as the arms race that exists between DDoS defenders and attackers [65], in which mechanisms of increasing sophistication are created for both mitigation and subversion. Furthermore, the DDoS threat is compounded by the growing number of unsecured Internet of things (IoT) devices that inadvertently facilitate the growth of botnet sizes and attack strengths [39].

Another reason is that despite the plethora of defensive techniques designed, no framework existed that enabled them to be objectively evaluated in a way that would allow them to be compared to determine which would be best suitable for deployment. Without such a framework, it is difficult for researchers to demonstrate just how well their solutions perform relative to others, and where their solutions fit in the attack landscape, relative to others.

In the global field of Information Technology, evidence-based agreement within and between relevant decision-making bodies and organisations drives action to improve standards, legal requirements and best practice guidelines. An example of this is the improvements made to the TLS protocol by the IETF resulting from formal analysis carried out by the academic community [151]. For such collaborations, or those between industry, governing bodies and the research community, demonstrable and widely accepted methods of assessment are needed, and this is no different for collaborations to improve the DDoS resilience of Internet infrastructure. Such methods would help facilitate the objective derivation of common conclusions that are acceptable across stakeholders, and are resistant to invalidation from subsequent independent investigative efforts. Depending on the DDoS defence in question, examples of stakeholders include ICANN [189], the ITU [190], DARPA [191], the IETF [45], Cisco [192], NIST [193] and the NCSC [46].

Numerous evaluation methodologies have been implemented or proposed (see section 2.2.5 of the literature review), yet none has offered comparative empirical outputs nor attained global recognition and adoption amongst industry, government and Internet organisations. The CED3 framework builds on proposals published in the literature to progress towards the possibility of a widely accepted evaluation method. through the development of a novel methodology encompassing a comprehensive set of evaluation goals that encapsulate the defence performance, deployment-implications and the evaluation process itself.

The CED3 framework, presented in this chapter, offers a starting point for comprehensive DDoS evaluation that could, help individual organisations evaluate defences for their own purposes or help global organisations to collaborate on next-generation Internet infrastructure. It was created by first analysing key concepts and organizing important aspects before creating ways, and guidelines in which these aspects could be captured in evaluations. The framework was then applied to four DDoS defences in demonstration of its usefulness and capability.

This chapter begins by outlining the goals of the CED3 framework before discussing, from first principles, the key concepts that give rise to the CED3 framework. CED3 is then detailed before CED3 itself is evaluated

5.2 Goals and Methodology

Outlined in this section are goals indicating what a useful and comprehensive evaluation methodology should achieve and how it should do so. This is illustrated in Figure 30. The detailed implementation of how CED3 fulfils these goals is addressed and discussed in subsequent sections. Also discussed in this section is the methodology by which the CED3 framework was constructed.

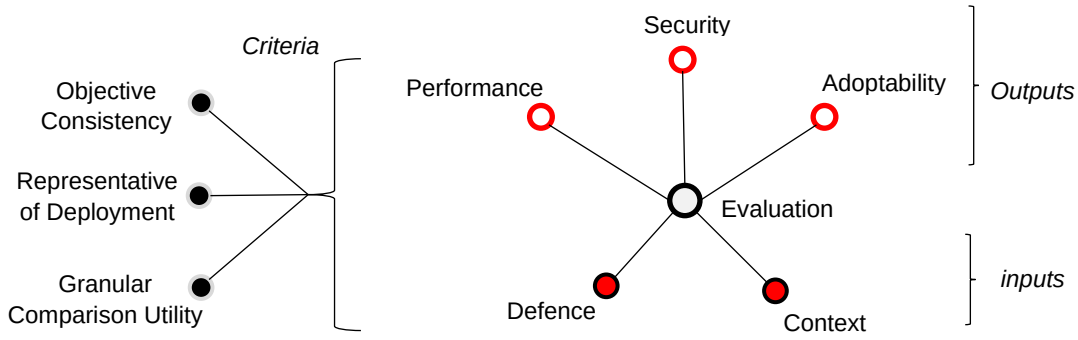


Figure 30: Evaluation Goals of a wide-scale Internet solution

Figure 30 illustrates the inputs to the evaluation process, the outputs of the evaluation process and the criteria that the evaluation outputs should meet.

The evaluation of a defence mechanism must capture how well the defence achieves its purpose – its **performance** – and must, not only determine whether a defence would work, but must also measure the extent to which it would work, with sufficient granularity to allow it to be **compared** with other defence mechanisms. In general evaluation should attempt to produce quantitative outputs as much as it makes sense to do so.

Another important criterion, **objective consistency**, is the idea that independent investigations of DDoS defences should produce the same comparative outcomes regardless of the individual investigators, provided that the same evaluation methodology and underlying assumptions are used. The way in which we can prevent a defence from appearing to be effective, only to later be shown to be ineffective, when tested under different circumstances, also falls under objective consistency.

Furthermore, the conclusions of the evaluation methodology should be as **representative of** actual **deployment** as possible (i.e. consistent assumptions must be used within the test scenarios between defences and these assumptions must be representative of the anticipated deployment environment). An example would be the characteristics of the legitimate traffic such as the mean packet size, data rate and the proportion of the total victim resources consumed by the legitimate traffic. In this regard, for commensurate comparison between evaluated defences to be achievable, the same contextual assumptions must be used when testing each defence.

The **security** aspect of evaluation output goals encompasses the possibility of a defence itself being subverted by an attacker, or introducing security vulnerabilities to the victim, stakeholders and the Internet in general, such as privacy leaks or single points of failure.

Although, producing a single scalar value as the evaluation output would make the process of ranking defences simpler, the process of converting multiple diverse outputs into a single scalar quantity would add complexity and would likely introduce unnecessary subjectivity. Additionally, due to the differing nature, and likely units, of the specified output goals (performance, security and adoptability), it doesn't make sense for these to be quantitatively combined, like comparing, or in this case, summing, apples and oranges.

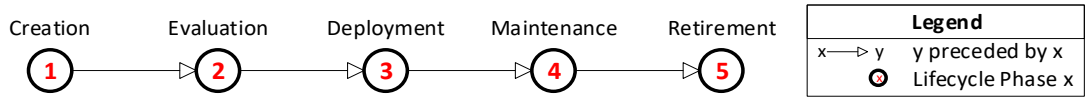


Figure 31: Internet solution lifecycle

The potential lifecycle of an Internet solution is illustrated in Figure 31. Evaluation occurs in the second phase of the Internet solution lifecycle, after creation of the evaluation object, and attempts to assess the effectiveness and collateral impact of the evaluation object upon

deployment. This has largely been the traditional focus of DDoS evaluation methodologies. However, also of great importance is the consideration of the suitability of the evaluation object at all subsequent lifecycle stages, which describes its **adoptability**, and takes into account factors that may not necessarily affect the performance of the defence but would influence its successful deployment, maintenance and retirement.

The fundamental concepts explored in constructing the CED3 framework are outlined in the next section.

It is not lost on the author that this section describes the methodology of creating a methodology. Nevertheless, the process began with an extensive literature review on existing evaluation methodologies and consideration of their shortcomings. Strengths and weaknesses were identified, and a new methodology was developed that attempted to build on identified strengths, whilst addressing weaknesses. For more details of the evaluation techniques reviewed, see section 2.2.5 of the related work.

The methodology combines theoretical and simulation procedures to evaluate and score various defences. It also combines a plethora of variables and other relevant considerations in order to produce outputs to the user that useful for the comparison of defences.

The purpose of the evaluation methodology is usefulness, via output reliability and practicability in meeting the needs of evaluation. An ancillary goal of the evaluation method is its widespread adoption to promote consistent objective evaluation of DDoS defence techniques. A more comprehensive list of evaluation methodology goals can be found in section 5.2.

On objectivity, it is not necessary for the underlying assumptions to be absolute in order to achieve objectivity in evaluation, however, they must be representative, justifiable and consistently applied in all evaluations. (An example of this is the estimated cost to an attacker of sending 1 Mbps of attack traffic to the victim). These three notions form the guiding principles from which much of the analysis and discussion in the creation of CED3 arises.

The challenge during the creation process is the assessment of how well the methodology achieves its purpose. The primary method of assessment is logical reasoning throughout the creation of CED3 to justify the approach and any assumptions made. This has been achieved by using completeness diagrams[1] and past solutions as frames of references to place the utility of various aspects of the methodology into perspective.

Application of the evaluation methodology is another way to test how well it achieves its purpose. By applying the methodology to several defences, various aspects of its performance

[1] A completeness diagram is a graph of delineated spaces that illustrates all the possible outcomes from the combination of a selection of independent variables. This may be done to highlight the scope of something.

can be observed. For example, one goal of the methodology is to provide outputs useful for the granular comparison of defences. This can be verified by the application of the methodology to numerous defences to see, for example, that all scores do not lie between 1001-1003 (as an arbitrary selection of numbers to depict a very small range of scores despite a seemingly large number of possible values).

Lastly, the reliability of the evaluation methodology can be evaluated by regression analyses of constituent parts to assess the dependency of the evaluation score on any variables. The change in the actual output score of the evaluation methodology due to variations in a variable is not so important as an indication of the robustness of the methodology. What is important, is the *relative* change of scores, where a defence that was scored lower, scores higher due to a change in a variable, such as legitimate traffic mix or the type of router queuing algorithm. Due to the variations in the way defences operate, this scenario is conceivable and does not necessarily invalidate the robustness of an evaluation methodology. However, this scenario serves to highlight the key variables of consideration and thus those that demand greater rigour in the justification of associated assumptions.

Concerning scope, the focus of the research defined by the attack-type, the victim and the source of the attack, is on flood attacks on ecommerce websites orchestrated by IoT devices that may or may not be concealed by Network Address Translation. The primary motive for attack considered in this project shall be financial gain.

Of the four methods of evaluating DDoS defences identified in [146] (theoretical, simulation, emulation, deployment), particular focus is to be placed on theoretical and simulation. Simulation allows for a flexible yet controlled environment in which sets of experiments can be specified and automatically executed, which allows for the easier assessment of the impact of various parameters. The disadvantage of simulation is the lack of realism when compared to running in real time (emulation) and on actual hardware (deployment), however this also means that simulation results are less hardware specific, which allows for more hardware-independent inferences in findings. The use of theoretical means enables the mitigation of the common phenomenon in DDoS literature of previously validated defences being deemed ineffective at a later time, when exposed to different attack patterns or test conditions.

5.3 Foundations

5.3.1 Inevitable Theoretical Subversion

The postulates constructed in this section, underpin the developed evaluation methodology.

As mentioned previously in section 2.1.3, *alleviation* DDoS defences harness observable and interrogatable characteristics of the attack landscape to make distinctions between malicious and legitimate activity. The obvious objective being the thwarting of malicious activity, whilst minimizing false positives and false negatives. However, since these characteristics are

digitally expressed in cyberspace, it is possible for an attacker to manipulate their exhibited characteristics to evade a defence by the emulation of legitimate actors and activity. However, it is important to note that some attributes of legitimate actors are more difficult or costly to emulate than others. For example, an attacker that reduces the individual data rate of its attacking bot agents in order to match that of the victim's legitimate clients, would require more bots to compensate and maintain its overall attack strength.

We know that even non-malicious activity can cause denial of service for users, since overwhelming flash crowds can cause DoS [26]. This is due to the finite capacity of victim devices to service requests because of limited processing power, memory and bandwidth at both the victim server and the access devices to which it is connected. In such circumstances, any defence relying solely on asymmetric facilitation would prove ineffective as the scale of service demanded by the legitimate actors would mean that even perfect distinction (with no false positives or false negatives) in the defensive measure, would still be unable to prevent DoS to legitimate actors.

Since DoS can be triggered by events that are void of malice (flash crowds) due to finite victim resources, and since it is possible for an attacker to emulate such events, then any attacker capable of sufficiently mimicking an overwhelming flash crowd is able to effect DoS. This is because victim resources are finite, and increasing attack, strengths, which increase malicious resource consumption, eventually deprive the victim's clients of necessary resources to receive adequate service; thus we form the postulate of **Inevitable Theoretical Subversion**:

Any DDoS defence technique is theoretically surmountable by an attacker.

This postulate extends the paradigm of defence effectiveness evaluation from assessment of how well a defence would work, to consideration of what it would take to overcome it. This extension of consideration in evaluation is important as it solves a problem identified in the Introduction (section 5.1), where a change in attack pattern would prove a defensive technique, that had initially been considered effective, to be ineffective. Therefore, evaluation longevity can be realized by considering and capturing, as part of the evaluation process, the ways in which a defence can be surmounted and the costs to do so, respectively.

The concept of associating the strength of a security mechanism with what it would take to subvert it has precedent in information security [194]. Passwords, for example, facilitate privileged access, yet some can be guessed or brute-forced more easily than others.

Therefore, if any attribute, used by a defence system to distinguish between malicious and non-malicious activity, can theoretically be emulated by attackers in order to subvert the defence, then

any truly representative measure of the effectiveness of a DDoS defence must incorporate the difficulty and or cost to the attacker in overcoming the defence.

This consideration helps avoid an unstable theoretical measure of effectiveness that is dependent on particular attack patterns or the capability of the attacker.

5.3.2 True Effectiveness

Effectiveness describes how well something achieves its purpose and, in DDoS defence literature, has been portrayed in different ways [146]. In order to clearly define effectiveness concerning DDoS defences, a clear understanding of the purpose of DDoS defence is needed, since effectiveness describes how well a defence achieves its purpose.

The purpose of a DDoS defence is to alleviate the negative effects of a DDoS attack as well as possible, whilst minimizing its own contribution to negative effects. Therefore, the effectiveness of DDoS defences shall describe how well defences alleviate the negative effects of DDoS attacks; where "negative" is defined from the perspective of attack victims and legitimate network users.

In addition to a clear definition of effectiveness, suitable metrics are needed that should sufficiently capture the negative effects of DDoS attacks in a way that reflects the quality of the experience of legitimate users (UEQ). This is discussed further in section 5.4.3.

In the consideration of the effectiveness of a defence, it is necessary for completeness to incorporate the resilience of said defence against attack and subversion. This is because, if an attacker can easily neutralize or evade a defence, the merit of the defence is disputable, and its effectiveness is nullified. Effectiveness nullification of previously certified defences, following exposure to different attack patterns, is a recognized phenomenon in DDoS literature [2].

Therefore, we introduce the notion of **true effectiveness**, which is a measure of the effectiveness of a defence that incorporates the resilience of a defence to attempts by an attacker to evade or subvert it. True effectiveness considers the effectiveness of a defence, not as its performance under a given attack but as the cost it would take an attacker to overcome the defence and cause denial of service as a result. Specifically, the true effectiveness of a defence is the increase in the *minimum* cost it would take for an attacker to overcome it. This is in alignment with the postulate of inevitable theoretical subversion and helps tackle the phenomenon of subsequent invalidation of previously validated defences in the literature.

The metric of true effectiveness is an important addition to the DDoS defence evaluation literature, due to its offering of longevity of evaluation conclusions. However, this metric alone is limited in its ability to highlight both a defences strength and weaknesses when comparing it to other defences. This is because true effectivenes, applied alone, provides a single value of effectiveness that does not distinguish between different attacker strategies,

despite the reality that different defences perfom differently to different attacker strategies. This brings us to the notion of *scopes of effectiveness*, discussed in the next section.

5.3.3 Scopes

Scopes of the defence landscape are remits describing a particular set of possible DDoS attacks, within which a defence subjected to said remit will exhibit a particular value of true effectiveness. This value of true effectiveness exhibited may vary with the true effectiveness exhibited should the defence be subjected to the remit of attacks of a different scope.

Thus, we may strategically divide the total attack landscape (specified by the threat model) into scopes in order to reveal different values of true effectiveness of a defence. The consignment of the true effectiveness of a defence to the scope of that defence is essential, since niche defences that target particular kinds of attacks would otherwise be deemed generally ineffective, despite being particularly effective for a given scope of attacks.

This notion, of non-uniform effectiveness of a defence over an attack landscape, led us to create the concept of *DDoS defence maps*. Defence maps enable DDoS defences evaluated under the CED3 framework to be visually compared, in a way that contrasts the strengths and weaknesses of the compared defences. each defence. allows them to be visualized -like a map.

The construction of these maps, as well as guidelines governing the delineation of effectiveness scopes contained within these maps, are discussed in section 5.4.5. Furthermore, the way in which true effectiveness incorporates costs for a given scope is described in sections 5.4.3 and 5.4.5.

5.3.4 Representative Evaluation

Complexity in comparative DDoS defence evaluation springs from the combinatorial explosion of possibilities arising from three input areas: the attack, the network and the legitimate user activity. The previous section posited principles to assist navigation of the domain of the first core input variable, the attack. This section addresses navigation of the core input variables of network and legitimate user activity. The principle underpinning this navigation is that of representative evaluation.

Representative evaluation refers to the notion that the output of an evaluation process should reflect that which would be observed had the object of evaluation actually been deployed, such that the evaluation outputs represent the reality. This requires that the evaluation conditions – parameters, experimentation method and assumptions – reflect that of the anticipated real-world deployment environment, as closely as possible.

However, since it is not possible to replicate, or even know, the entire Internet for testing, simplifications must be made. These simplifications apply to the network upon which a defence is tested and the legitimate user activity present within the test environment. The

network and the legitimate user activity together form the *evaluation context* – depicted in Figure 30.

For a defence to be robustly evaluated, the network of the testing environment must be able to sufficiently accommodate all of the features of that defence. For example, an environment for testing a defence that operates at the borders of autonomous systems, must contain borders between autonomous systems. In addition to this requirement for robust individual evaluation of defences, for defences to be commensurably compared, they must be evaluated under the same context. The evaluation context would then be seen as a caveat of any comparative evaluation conclusions.

The papers [146] and [147] discuss DDoS defence testing by theory, simulation, emulation and deployment. From the CED3 standpoint any method is suitable provided it sufficiently captures the mechanics of the anticipated real-world deployment environment. However, the "new dimension" of evaluation put forward in [147], comprising the use of real datasets, is unsuitable for CED3 since it doesn't allow for the realization of true effectiveness due to the use of a single historical static attacker strategy.

DDoS attacks operate at different layers of the TCP/IP Model and their study has largely been divided into the areas of network and application layer attacks.

Different properties of IT communication are observable on different layers. For example, the application layer payload of a delivered packet is not visible at the network layer, although IP addresses are. Therefore, in order to evaluate discriminative defences that utilize metrics on any layer, the evaluation methodology must not be limited by the layer at which the defence operates, and must continue to provide granular, objective and representative outputs[1] provided the defence can be sufficiently implemented in a suitable test environment.

[1] See the criteria outlined in the goals section

5.4 Framework

5.4.1 Overview

Figure 32 depicts an overview of the inputs and outputs of the CED3 evaluation process. As shown, CED3 realizes the DDoS defence evaluation goal of assessing performance (discussed in section 5.2), via the notion of *true effectiveness*. The constraints and issues regarding adoptability are too vast and diverse to be meaningfully simplified into a single metric, however CED3 collates a set of considerations that can each be qualitatively combined to provide an indication of the suitability of a defence scheme for adoption. We integrate the defence evaluation goal of security assessment (discussed in section 5.2) into both the evaluation outputs of true effectiveness and adoptability. This is because, when it comes to DDoS defence deployment, we consider the significance of security vulnerabilities (identified by any security assessment process) are only important with respect to how they impede the capabilities of a defence to reliably mitigate attacks or how they diminish a defence's suitability for adoption.

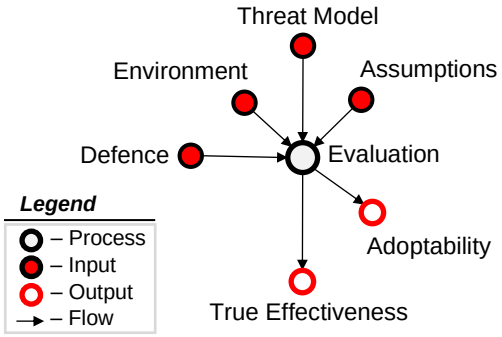


Figure 32: CED3 Output Overview

Figure 33 illustrates the steps of the CED3 evaluation process, in the form of a flow diagram. As shown, the process begins with a *structured theoretical analysis* (STA) before forking into two evaluation pathways: empirical and non-empirical. The separate threads of execution are largely independent of each other and hence can for the most part be executed in parallel. The exception is certain empirical data that would affect adoptability of a defence, such as collateral damage (see section 5.4.3.2). The findings of the STA allow a suitable test environment to be designed, that can accommodate all the defences to be evaluated. The STA also enables the identification of the weaknesses of each defence in order to design more effective attack scenarios for each defence. This phase is structured in order to enhance scrutinization and promote objective consistent evaluations. It is important to note that findings of the STA needn't be exclusively effectiveness or adoptability related and may lie in both. For example, a vulnerability allowing an attacker to evade detection may also enable a user to impersonate another user.

Following the STA, a suitable test environment that can sufficiently support all the defences is designed. Following the effectiveness pathway, suitable attack scenarios are designed and an experiment plan to identify the lowest-cost effective attack is formulated for each defence. The defences and test environment are then implemented, and the experiment plan is then executed. Data output from the experiments are then processed and analyzed. If the data is sufficient, as determined from its analysis, then the true effectiveness of each defence is output. However, if it is evident, after analysis, that more data is needed then a new experiment plan is made to acquire more data. An example situation requiring more data is, if there is a sudden change in UEQ between adjacent experiments, it would be informative to acquire more data points between the parameter values that produced the sudden change in performance. This may be necessary in order to determine, more precisely, the value of the parameter that is required in order to cause a defence to be subverted.

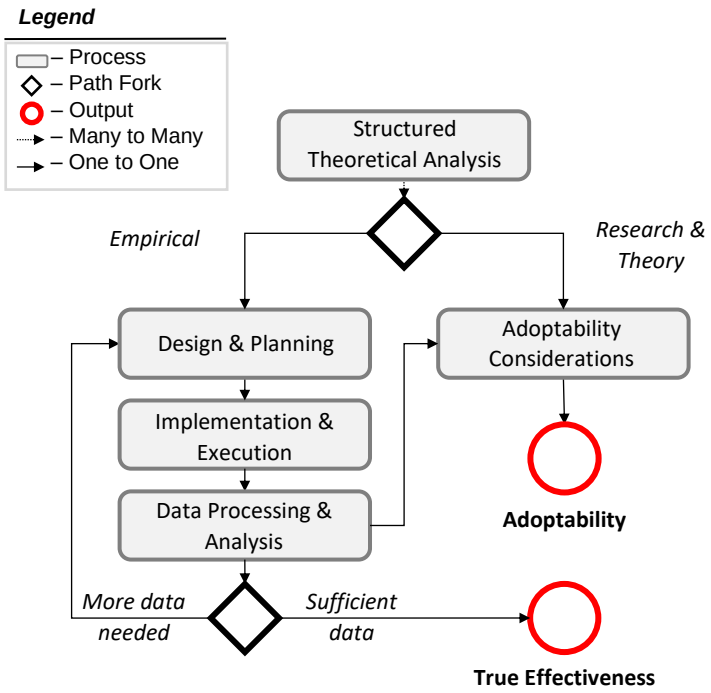


Figure 33: Flow diagram illustrating the steps of the CED3 evaluation process

Outcomes of the STA that relate to adoptability include a detailed understanding of how a defence is put together, which will assist in evaluating adoptability considerations such as costs. As discussed in section 5.4.4, certain adoptability requirements are specified in the STA in order to explicitly draw findings that are relevant for the adoptability considerations. The outputs of the CED3 evaluation process, true effectiveness and adoptability, can be presented in ways that allow easier visual comparison between defences. This is in the form of *defence maps*, to compare true effectiveness, and *adoptability graphs* to qualitatively compare the adoptability of different defences. These are discussed in section 5.4.5.

The rest of this section (5.4) is dedicated to exploring the components of Figure 32 and the steps outlined in Figure 33. For completeness, details of the STA can be found in appendix B.2.

5.4.2 Evaluation Context

As indicated in Figure 32, the entire evaluation process is shaped by the evaluation *context*. We define this context of evaluation as the collection of factors that will influence how representative an evaluation outcome is to the anticipated deployment environment of the evaluated defences. Such factors include:

- The network
- The legitimate traffic
- The threat model

And all assumptions about the circumstances of deployment envisioned for the candidate defences to be evaluated. It is important to note that for a fair comparative analysis, the anticipated deployment environment of all defences selected for the comparative evaluation, must be the same. For example, it is easy to see how one defence tested in conditions where the victim server has a higher bandwidth capacity, could produce more favourable results that do not reflect the capabilities of the defence itself.

CED3 doesn't specify a particular network topology, however the framework recommends the following. When choosing a suitable topology for the test environment, it must be able to accommodate the kind of attacks possible within the attack landscape governed by the threat model. The topology must also be able to support the deployment of all the chosen defences for the comparative evaluations; allowing them to be correctly placed in the portion of the network for which they were designed. For example, if two defences were to be evaluated under CED3 for the purpose of being compared, and one was designed to operate in the network core, whilst the other was designed for deployment at the access layer, then the topology to be used for the empirical testing of the defences must possess both the core and access layers. Needless to say, the same topology must be used for all experiments in order to uphold ceteris paribus and eliminate topological changes as cause for observed differences in results. If the intended deployment environment is clear, then CED3 recommends the test environment topology matches it as much as is relevant. This includes factors such as link latencies, node divergence (the number of degrees on each node), link capacities, degree of mesh connectivity, and so on.

The legitimate traffic of the test environment, similar to topology, should try and match the anticipated deployment environment as much as practicable. Aspects to consider matching are the link capacities consumed by the legitimate traffic, the protocols used and the proportionate mix of protocols. Statistical aspects such as the mean, and standard deviations of the mean data rate, the packet size, and number of active clients should also be observed.

By mimicking the statistical attributes of the legitimate traffic and not the legitimate traffic itself, one is able to repeat experiments to obtain a generalized measure of performance, rather than repetitions producing results that are exactly the same. Additionally, by using dynamically generated data, that adheres to the statistical attributes of the anticipated deployment environment, we are able to capture the effects of client responses to congestion, such as TCP congestion control means.

The threat-model defines the landscape of all possible attacks available for use in a CED3 evaluation. The closer the threat-model is to the reality of the anticipated environment in which the defences will be deployed, the more representative and meaningful the evaluation process. For fair evaluation, the threat model must not unduly exclude attacks that may be effective against one of the defences. As we've seen in section 5.3, the purpose of CED3 employing the notion of true effectiveness is to enable confidence in the reliability and longevity of the comparative evaluation findings, and a strong threat model is required as a result. However, for attack aspects deemed to be insignificant to all candidate defences under evaluation, it is reasonable to limit the threat model accordingly, especially when the means of testing does not readily support said aspects. Examples of these include Economic Denial of Service attacks and application layer attacks, the likes of which are excluded from the CED3 framework proposed.

The context of evaluation also encompasses any assumptions required for the operation of any of the candidate defences. For an example, if a defence uses PKI authentication in it's set-up phase, an assumption in the testing of said defence would be the availability of PKI infrastructure necessary for the defence to operate, even if said infrastructure is not necessary after the set-up phase.

However, any assumptions prerequisite for the defence to operate, whose stipulation differs widely from the anticipated deployment environment, would need to be considered in the adoptability evaluation phase. The questions that would need to be answered concerning such an assumption are how impactful is said assumption to the reliability of the evaluation conclusions and what it would take to transform the anticipated deployment environment so that it satisfies the assumption. Two obvious such considerations are the feasibility of transformation and the financial cost in doing so. We recommend that assumptions that deviate widely from the anticipated deployment environment should be avoided as much as possible so as to foster applicable and reliable evaluation conclusions. Needless to say, for a fair evaluation, the same context must be used in the evaluation of all defences chosen for the comparative evaluation process.

It is recommended that preliminary baseline experiments and sensitivity analyses be performed before the evaluation context is finalized, so that reasonable values of parameters can be selected in order to satisfy various trade-offs. For example, replicating the number of nodes of the anticipated deployment environment may be costly (simulation time, or hardware

cost) if the number is very high, therefore preliminary experimentation where the number of nodes is adjusted to observe its effect on attack impact is recommended so that a reasonable number may be chosen for the test network. Such preliminary experiments may provide corroborating data when discussing the veracity of any evaluation conclusions later on.

5.4.3 True Effectiveness

5.4.3.1 Metrics

As discussed in section 5.3.2, the notion of true effectiveness extends the conventional paradigm of effectiveness by considering not just how well a defence performs, but, following from the postulate of inevitable theoretical subversion, what it would take to overcome said defence. However, before we can determine what it takes to overcome a defence, we must first define what it means for a defence to be overcome. This definition must then be accompanied by practicable metrics that are able to accurately identify when a defence has been overcome and when it hasn't. This section identifies these metrics and discusses the process of how true effectiveness can be obtained in more detail including how to measure it and the specifics of which metrics to use.

Since the purpose of a DDoS defence is to prevent denial of service (DoS), the effectiveness of such a defence is characterized by how well it can prevent DoS. Therefore, by defining the conditions for denial of service and finding suitable metrics that can identify those conditions, we construct a way to determine if a defence has indeed been successful or not.

By definition, denial of service occurs when legitimate users of an online service are made to no longer be able to access that service. This includes cases where the quality of service provided to the legitimate users is so far diminished that the service is no longer usable, despite the service still being technically accessible. An example of this is a VoIP call, in which the words spoken are unintelligible despite the connection between the two callers remaining intact. We name the scale upon which this threshold of unsatisfactory quality of service sits, the *user-experienced quality of service* or UEQ. Thus we define UEQ as the fundamental metric by which denial of service shall be determined.

Now that UEQ has been identified as the fundamental quality that determines whether a DDoS attack was successful, how can it be empirically ascertained?

We identify several aspects of connectivity that can indicate conditions of sufficiently diminished UEQ to be considered as DoS. These aspects describe of the amount and timeliness of data transfer between server and client and can capture the perceived UEQ since they characterize the connectivity of legitimate clients and their changing may be symptomatic of both network and application layer DDoS attacks. We choose the following metrics to represent these aspects pertaining to perceived UEQ:

- Packet loss
- Delay
- Jitter and
- Transaction time

Packet loss is a metric that describes the number of packets sent that fail to reach their intended destination. In the context of DDoS defence evaluation, we define the packet loss metric as the number of packets sent, as part of a transaction between hosts, that fail to reach their destination. Where in most cases one host is the server and the other the client and the particulars of the transaction is defined by the service provided by the server. An intuitive example of the impact hat packet loss can have to the UEQ is a UDP remote desktop service, in which a individual, say, a home-worker, connects to and operates a machine in a remote location. Under good operating conditions instructions from the user flow as packets across the Internet to the remote machine and the response, including video and audio flow from the remote machine back to the user. Packet-loss in such a scenario, could mean that instructions, such as mouse-clicks sent by the user, could fail to be registered by the remote machine and responses to received instructions, such as screen changes, may fail to be received in full. Too much of this and the service would be unusable.

The metric *delay* captures the transit time between source and destination. Delay is important since it would mean unsatisfactory UEQ for some services if increased sufficiently. An example of such as service is real-time online gaming, in which increased delays could significantly diminish a user's gaming performance in and enjoyment of the game.

Jitter is a metric that captures the periodicity (or "jumpiness") of packet arrivals. Increased levels of jitter can affect certain real-time applications, such as VoIP by causing words to sound garbled.

Transaction time describes the time it takes for particular steps of a service to complete. This metric varies based on the service in question and the steps of the service considered as one transaction is also flexible. This flexibility is not an issue, provided the steps are able to be completed in a reasonable timeframe, when no attack is present, and provided that a consistent definition is used throughout the comparative evaluation process. For example, the steps of a transaction could be downloading and uploading two sets of files from and to an FTP server; in which case the transaction time would be the total time taken for the transfers to complete. Since transaction time correlates to how long it takes to do things online, it is clear to see how increased transaction time reduces the UEQ.

The significance of these four metrics is such that denial of service cannot be achieved without their degradation. The exception to this includes compromises in the integrity of data processing and transmission, both of which are symptoms of application-layer DDoS attacks and lie out of scope of this evaluation methodology.

Now that we have the means to objectively determine if a defence has prevented DoS or not, how do we choose attack scenarios to test each defence in order to ascertain the lowest cost for an attacker to overcome the defence (i.e. ascertain the true effectiveness of the defence)?. This question is answered in the following subsections (5.4.3.2 & 5.4.3.3).

5.4.3.2 Measurement

5.4.3.2.1 What

As illustrated in Figure 34, it is not just the effectiveness of a DDoS defence that is worth capturing, when attempting to measure its impact.

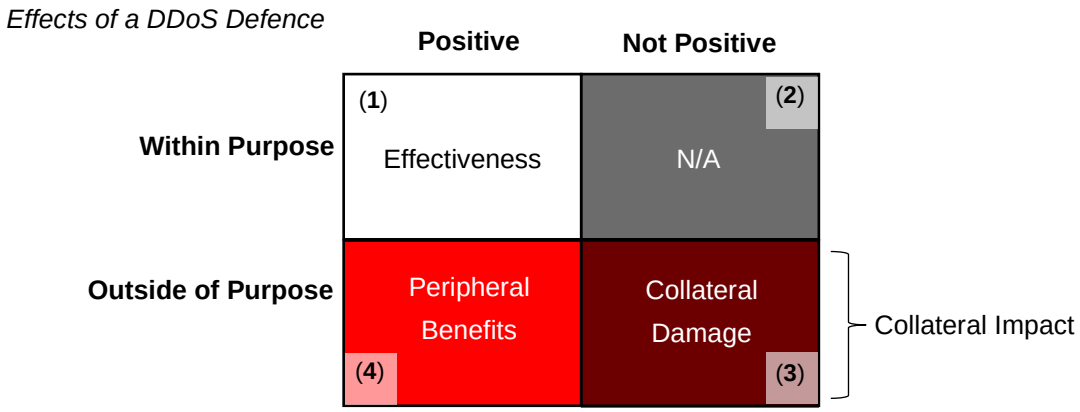


Figure 34: A completeness diagram delineating evaluation outputs pertaining to defence effects

Collateral damage is also a significant factor and refers to the reduction a defence may cause in the UEQ of legitimate network users and service providers. Such effects include anything that could degrade the quality of service that legitimate network users experience – such as jitter, delay or packet loss. Similarly, if a defence requires additional user interaction, such as CAPTCHA's then this would also be considered collateral damage.

However, not all collateral effects may be negative. It is possible that some secondary effects of a defence may produce positive benefits to legitimate internet users. We name this category of collateral impact *peripheral benefits*.

The purpose of DDoS defences is intended to be exclusively positive, therefore no upper right quadrant exists in Figure 34 as all possible negative effects of a defence lie outside of its purpose.

For the purpose of clarifying alignment between the goals described in section 5.2, "how well" a defence alleviates the negative effects of DDoS attacks shall exclude its effects outside of its purpose, however positive said effects are.

Both collateral damage and peripheral benefits are factors that could influence the adoptability of a defence and so are captured in the analysis and measurement process.

5.4.3.2.2 How

The UEQ-based effectiveness and *collateral impact* can be found by testing the defence under various scenarios. These are illustrated in the completeness diagram of Figure 35.

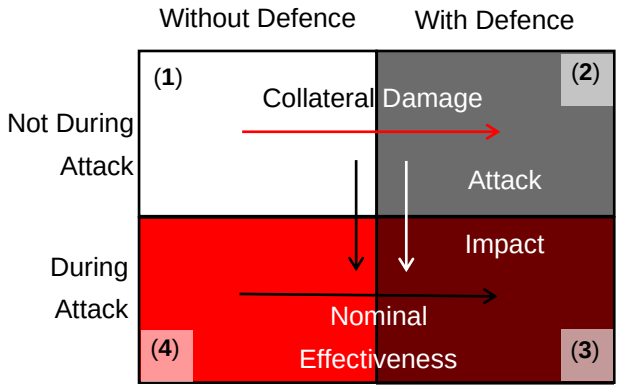


Figure 35: Attack and peacetime scenarios for defence testing

By measuring the change in UEQ upon the introduction of the defence when there is no attack, the collateral impact of the defence can be deduced. We define the UEQ-based collateral impact as the positive or negative effects a defence can have on the UEQ of users outside of an attack. If this impact is a reduction in UEQ, we call this collateral damage, if this impact is an improvement in UEQ, we classify this as a peripheral benefit (see section 5.4.4.3 for more on peripheral benefits).

Measuring the change in UEQ from during an attack *without* the defence, to during the same attack *with* the defence deployed, provides an indication of the *nominal effectiveness* of the defence. Nominal effectiveness is an indication of the impact of a defence under a given attack scenario and is measured by the change in UEQ, when the defence is present in the scenario (ceteris paribus).

Since nominal effectiveness is derived in this way, by measuring the alleviating impact provided by a defence in the presence of an attack, the nominal effectiveness of a defence is attack specific. However, since the possible variations of attacks are potentially unlimited, some method is needed in order to organize the measurement of effectiveness for a defence in a way that objectively and meaningfully characterizes the effectiveness of a defence. CED3 achieves through the notion of true effectiveness, which is obtained by measuring the UEQ degradation caused by attacks formulated from applying a chosen threat model to the weaknesses of a defence, which are discovered by a theoretical analysis process.

Figure 36 illustrates the four test scenarios marked along a UEQ axis, which decreases in the downwards direction.

The horizontal arrow along the UEQ axis, marks the degradation threshold, which is a UEQ value, beyond which user service is considered to be suitably disrupted for an attack to be deemed successful. The true effectiveness then, is the increase in minimum cost necessary for

an attacker, within a given scope[1] of attacker strategies, to degrade the UEQ to below the degradation threshold. The first cost is the minimum cost necessary to diminish the UEQ of clients of a victim service to below the degradation threshold when there is no DDoS defence deployed. This cost (to the attacker) is measured in scenario 4, for an attack that is successful at the lowest cost. The second cost is that required for an attacker to successfully degrade the UEQ beyond the degradation threshold with the defence deployed, and is the measure of the lowest-cost successful attack in scenario 3.

We note then that the level of UEQ in scenario 3 is dependent on the strength of the attack applied in that scenario. Hence in order to discover exactly which cost is required for an attacker to overcome a defence, we can increase the attack strength until the UEQ of the scenario crosses the degradation threshold. We also note that, despite the degradation threshold, remaining the same for all defences under test, the cost required for an attacker to cross that threshold, will vary depending on the defence deployed

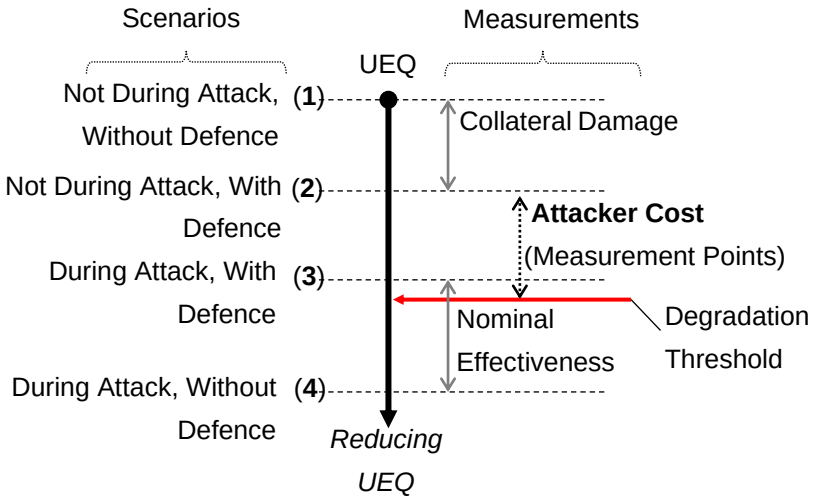


Figure 36: Output measurements and scenarios depicted along UEQ axis

Consideration of collateral damage is a crucial part of evaluating the performance of a defence [195]. In CED3, the collateral damage of a defence consists of two parts: that caused by a defence when there is no attack, and that inflicted on servers that are not under attack, during an attack on a victim. The former, indicated in Figure 35, is obtained from the difference between the UEQ measurements taken when no defence is deployed and no attack is present, and when the defence is added (ceteris paribus) and no attack is present - quadrants 1 and 2, respectively. The latter is measured between quadrants 3 and 4 and captures the damage a defence may inflict, on legitimate entities that are not under attack, in response to an attack.

[1] The scope of an attacker strategy refers to the set of techniques an attacker may use to achieve DoS, which may have explicit exclusions. This is described in section 5.3.3.

For a given test scenario, indicated by the n'th quadrant in Figure 35, the UEQ of users of the victim server, $\varphi_V^n(\eta^n)$ is given by:

Equation 13

$$\varphi_V^n(\eta^n) = \frac{K_1}{a_L\overline{\boldsymbol{\mu}_L^n}(\eta^n) + a_D\overline{\boldsymbol{\mu}_D^n}(\eta^n) + a_J\overline{\boldsymbol{\mu}_J^n}(\eta^n) + a_T\overline{\boldsymbol{\mu}_T^n}(\eta^n) + K_2}$$

The multiplicand $\overline{\boldsymbol{\mu}_X^n}(\eta^n)$ represents the mean of the measured metric x, in the n'th quadrant, where x is either L, D, J or T; denoting: loss, delay, jitter or transaction time. It is a function of η, which represents the network conditions in the n'th quadrant. The multipliers of the form a_X are coefficients to balance the contributions of each metric to the UEQ. The dividend K_1 and the summand K_2 are constants to restrict the general magnitude of the UEQ variable and prevent division by values close to zero.

Equation 13 presents a simple function that is adequate enough to capture UEQ on a scale. It holds the property of inverse proportionality between the combined UEQ and the individual measurable metrics such that increases in the individual metric results in decreases in the overall UEQ. This is exactly what is needed since, as described in sections 5.4.3.1, increases in the identified metrics loss, delay, jitter and transaction time could lead to unusable service. The possibility for combining metrics using different formula is discussed in section 6.5.

The peacetime collateral damage $X_V(\eta^n)$ is given by:

$$X_P(\eta^n) = \varphi_P^1(\eta^n) - \varphi_P^2(\eta^n)$$

And is measured on the users in general that may be impacted by the defence deployed. The in-attack collateral damage $X_A(\eta^n)$ is given by:

$$X_A(\eta^n) = \varphi_A^4(\eta^n) - \varphi_A^3(\eta^n)$$

And is measured on users on the network that are not in communication with servers that are under attack. The nominal effectiveness is given by:

$$X_E(\eta^n) = \varphi_V^4(\eta^n) - \varphi_V^3(\eta^n)$$

And is measured on the quality of the connection between the victim server and users of the victim server.

If we consider the network conditions in the n'th quadrant η^n as a function of the network topology N_T, the legitimate traffic L_T and the attack traffic A_T^n:

$$\boldsymbol{\eta^n} = F(N_T, L_T, A_T^n)$$

We find the true effectiveness by finding the lowest cost attack A_T^n such that the UEQ with the defence deployed crosses the degradation threshold τ_D:

$$\varphi_V^3\left(\eta^3\left(A_{T=\tau_D}^3\right)\right) < \tau_D$$

Where

$$\varphi_V^3\left(\eta^3\left(A_{T=\tau_D}^3\right)\right) = \frac{K_1}{a_L\overline{\boldsymbol{\mu}_L^3}\left(\eta^3\left(A_{T=\tau_D}^3\right)\right) + a_D\overline{\boldsymbol{\mu}_D^3}\left(\eta^3\left(A_{T=\tau_D}^3\right)\right) + a_J\overline{\boldsymbol{\mu}_J^3}\left(\eta^3\left(A_{T=\tau_D}^3\right)\right) + a_T\overline{\boldsymbol{\mu}_T^3}\left(\eta^3\left(A_{T=\tau_D}^3\right)\right) + K_2}$$

And subtracting from it the lowest-cost successful attack where the defence is deployed. Hence the true effectiveness ε of a defence is given by:

$$\varepsilon = C\left(A_{T=\tau_D}^3\right) - C\left(A_{T=\tau_D}^4\right)$$

Where $C(A_Z^\eta)$ is a cost function that outputs the cost of the input attack. If, when the defence is deployed, the victim service can be overcome by an attack of the same cost as the minimum cost required when no defence is deployed, then the true effectiveness of the defence is zero. If, somehow, after the defence is deployed, the victim service can be denied by an even cheaper attack, then the true effectiveness is negative, and the users of the victim service are likely better-off *without* said defence in question deployed.

5.4.3.3 Experiment Planning

As illustrated in Figure 33, theoretical analysis underpins the evaluation process. With respect to determining true effectiveness, the theoretical analysis of the chosen defences for evaluation is necessary in order to plan the testing of each defence. Since true effectiveness requires the discovery of the minimum cost to an attacker in overcoming a defence, we must identify the weaknesses of the defence in order to execute a targeted approach to discovering the lowest-cost attack able to overcome it.

Since DDoS attack strength is generally boosted by increasing the number of attackers or their mean data rate, what we're searching for is other aspects of the attack that an attacker can manipulate, within the bounds of the given threat model, in order to maximise the impact of the attack against a particular deployed defence.

One method to accomplish this is via the identification of *significant defence factors*. These are aspects that a defence uses to distinguish between malicious and legitimate activity for the purposes of *asymmetric facilitation*. Asymmetric facilitation refers to the action taken by a defence in order to mitigate a DDoS attack. Examples of this include the dropping of suspected attack packets, the allocation of such packets to low priority queues or the marking of suspicious flows for de-prioritization upstream. Conversely asymmetric facilitation also includes the favourable treatment that may be afforded to traffic that is not identified as malicious or suspicious, such as various resource allocation prioritizations. So how are significant defence factors used to shape the experiment design?

Let us consider a defence for which the data rate of client flows is identified as a significant defence factor, such that the defence uses it to identify malicious packets. The process of discovering the minimum cost of attack to overcome the aforementioned defence would need to attempt to find a flow rate that attacking clients can use in order to maximise the impact of the attack for as little cost possible. This would be the highest flow data rate for which the defence in question is unable to accurately distinguish between that malicious flow and the legitimate flows running on the network.

Another point of consideration is the structure of the defence itself to identify weaknesses such as single points of failure or unsecured channels of communication. In this regard, we note that findings from the recursive threat analysis process may contribute to the discovery of the lowest cost attack able to surmount the defence in question.

5.4.4 Adoptability

5.4.4.1 Overview

We define a*doptability* as an attribute of a DDoS defence that describes how conducive a defence is to its own adoption in a given environment. This encompasses aspects that may increase conduciveness as well as those that may reduce it. We recognize the distinctions between *implementation*, *deployment* and *adoption*, specified by Eardley et al. [150] but extend the definition of implementation to accommodate the diversity of DDoS mitigation solutions.

Implementation refers to the coding and manufacture of the software and hardware components of the DDoS defence solution. Whereas deployment refers to the implemented solution being put into the network. Adoption refers to the additional step that the deployed solution is actually used. Hence deployment is prerequisite to adoption and implementation is prerequisite to deployment. The benefit of these distinctions is revealed later on in this section, when considering different stakeholders that may be responsible for executing these different phases.

Therefore, adoptability describes how conducive a defence is towards its being used in a given network environment, which encompasses its conduciveness towards it being both implemented and deployed. The consideration of adoptability is important when comparing or merely evaluating defences since, adoptability covers aspects, outside of effectiveness, that would affect the viability and success of defence should it be selected for deployment.

CED3 addresses the question of adoptability by providing guidance on significant factors worthy of consideration in assessment of a defence's adoptability. The CED3 framework does not stipulate that all factors be evaluated by parties applying the CED3 framework in order for an evaluation to be completed, but rather CED3 suggests certain factors for consideration where individual researchers can choose which aspects are considered based on need and availability of resources. Similarly, although the list of recommended factors to consider is broad, CED3 does not prohibit the use and consideration of other potential factors when evaluating the adoptability of defences. However, when performing a comparative evaluation of multiple defences, it is important that the same adoption factors be considered for each candidate defence.

In addition to identifying adoptability factors for consideration, this section also presents some guidance on making comparative adoptability evaluations, discussing the complexity and various challenges involved.

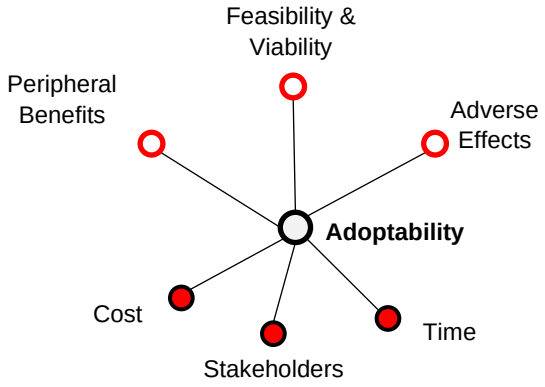


Figure 37: Adoptability Aspects for Consideration

Figure 37 illustrates three categories of adoptability aspects for consideration (peripheral benefits, feasibility & viability, and adverse effects) and three factors to consider (cost, stakeholders and time) when evaluating said categories.

Both *peripheral benefits* and *adverse effects* address the impact a defence would have outside of its intended purpose, which is to mitigate DDoS attacks. The former addresses any positive effects and the latter addresses the negatives. The category *Feasibility and Viability* addresses the questions of how realizable a defence is in practice and how likely it is to succeed.

These categories are explored in further detail in subsequent subsections of this section. As shall be shown, CED3 adoptability evaluation focusses on aspects that are caused by the DDoS defence itself, rather than other aspects that may indeed affect the success of the defence but are not caused by the defence itself – such as marketing, branding, general awareness of the defence and geopolitical nuance.

5.4.4.2 Feasibility and Viability

The category 'feasibility and viability' encompasses aspects of DDoS defences that indicate how likely defences are to succeed in an anticipated deployment environment. It not only considers how viable a defence would be within the anticipated deployment environment, but also considers its suitability within expected future versions of the anticipated deployment environment. This section discusses the following aspects of feasibility and viability:

- Interoperability
- Adoption incentives
- Scalability
- Accessibility

The aspect of *interoperability* considers whether a defence and its components would be compatible with the anticipated deployment environment such that both the defence and the anticipated deployment environment can function as required, without disruption.

Interoperability concerns can be highlighted in the analytic phase of DDoS defence evaluation and tested in the experimental phase.

Adoption incentives consider the rational motivations an adopting entity would have to adopt a DDoS defence. The entities in question are any entities required to take action for the success of the defence, and the actions include implementation, adoption and actual use of the defence (per the prior definition of adoption). The adoption incentives category weighs the benefits an entity receives from taking action in the adoption process against the cost incurred in taking said action. An example would be a service provider entity choosing to take the action of deploying a particular defence for the benefit of protecting existing customers or creating new revenue streams.

The reason adoption incentives are so important is that they can strongly indicate how likely it is that a defence will actually be used, without outside intervention, since rational selfishly-motivated players won't take action without perceived personal benefit. Analysing adoption incentives can even help determine the type, level and cost of any outside intervention required to make the DDoS defence achieve widespread use.

CED3 stipulates no specific process to evaluate adoption incentives but rather offers some recommendations: The same process should be used to evaluate the adoption incentives of each defence of a comparative evaluation. This includes the same methods for obtaining costs. Not all defences are constructed with the same terminology and delineation of entities, so for ease of comparison, a common language and delineation of entities should be derived and used. For defences whose benefits vary with adoption scale (via, perhaps, the network effect), this should be considered, and common deployment scales should be used when making comparisons between the adoption incentives of defences.

Scalability considers the technical practicalities of the process of growth of the defence and the technical limitations that may affect the anticipated deployment environment and its future variants. Part of that is *incremental deployment analysis,* which considers the technical aspects of actually deploying (and using) the defence until deployment and use reach saturation. It also addresses questions on service disruption duration and scale. In this regard, to facilitate comparison between different defences, it is recommended that incremental deployment analysis consider units of cost and time; where the costs would include opportunity cost resulting from service outage.

Assessment of any hard limits of the defence, also falls under the remit of scalability. For example a defence may assign a global unique unsigned integer identifier to network entities. In such a case, assuming no values are off-limits, the number of entities able to receive a global unique identifier would be limited by the capacity of the data type used to store each identifier. For example, a 32-bit unsigned integer, could store 2^{32} (or 4,294,967,296) unique

identifiers, which would impose a limit on the number of entities able to possess said unique identifier.

Accessibility takes into account the availability of any information and legal rights of use necessary for the implementation and adoption of the defence. The consideration of accessibility provides two main outputs: one binary and one quantitative. The binary output specifies whether the accessibility of the defence is sufficient or insufficient for its anticipated use, whereas the quantitative output specifies any costs required to provide sufficient access for adoption. These costs may refer to royalties or licencing fees which may be fixed or recurring.

Complications may arise from the timing or duration of legally binding accessibility contracts, and these should be considered with respect to how they may affect the feasibility and viability of the overall defence.

5.4.4.3 Peripheral Benefits and Adverse Effects

Peripheral benefits of a DDoS defence include any positives of adopting the defence that lie outside of the purpose of DDoS defences, which is to mitigate DDoS attacks. An example of such a benefit could be other security benefits offered by the defence architecture, such as non-repudiation. The defence may depend on additional infrastructure to successfully operate in the anticipated deployment environment. The presence of that additional equipment, in the anticipated employment environment, could enable or improve other services, outside of DDoS mitigation. An example of this type of equipment could be PKI infrastructure.

Adverse effects refer to the negative impacts a defence may have on its environment and stakeholders. Collateral damage (introduced in section 5.4.3.2.1) refers to the degradation of the user-experienced quality of service (UEQ) that a defence inflicts on benign network users. Illustrated in Figure 35 is the *peacetime collateral damage*. It is the damage inflicted to the UEQ of clients in the network when no attack is present and is obtained by subtracting the UEQ of network clients before the defence is deployed (quadrant 1) from the UEQ of those clients after the defence is deployed (quadrant 2). A negative value would not be considered as collateral damage, since it would actually indicate an increase in the peacetime UEQ as a result of the deployment of the defence, which would be a peripheral benefit of the defence.

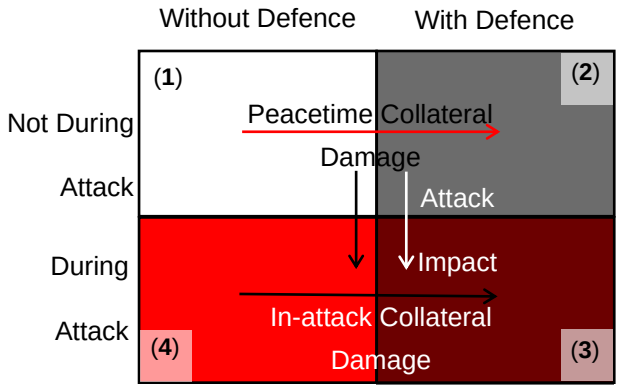


Figure 38: Collateral damage illustrated on attack and peacetime scenarios

Figure 38 illustrates the measurement of both types of collateral damage on the same diagram. The *in-attack collateral damage* is the reduction in UEQ caused by the defence when an attack is in-progress. It is obtained by measuring the reduction in UEQ from when the defence is not deployed and there's an attack (quadrant 4) to when the defence is deployed and there's an attack (quadrant 3). Unlike the peacetime collateral damage, the in-attack collateral damage excludes in its measurements, the clients of the victim service under attack. This is because the attack-effects on these clients are already captured by the true effectiveness measurement process. Similar to the peacetime collateral damage, negative (reductions in UEQ) values are not considered collateral damage but rather as peripheral benefits.

Other forms of adverse effects other an collateral damage exist, such as adverse security or compatibility implications. For example a DDoS defence may hi-jack an option of an IPv4 header and thus prevent the user of services for which that options was dedicated. Also, a defence may introduce a security vulnerability into the target deployment environment that would leave the security of the target deployment environment in a less secure state after deployment.

Since the peripheral benefits of a DDoS defence refer to benefits that lie outside the direct purpose of mitigating DDoS attacks, their contribution to the comparative evaluation between DDoS defences lies in their ability to incentivize the adoption of their defence due to the peripheral benefits offered, and also to foster a reduced DDoS environment perhaps due to improved device security or accountability which would prevent infections or disincentivise attacks in the first place.

The consideration of adverse effects contributes to the comparative evaluation process by identifying defences that may have performed well but whose adverse effects are unacceptable for the anticipated deployment environment (ADE). We do not see sufficient value in ranking adverse effects of a defence if they are unacceptable for users of the ADE. Therefore, consideration of adverse effects serves primarily to determine which defences are

unsuitable for the ADE, then, secondarily, to provide input into a selection process amongst suitable defences only, which is carried out by weighing pro's and con's.

We recognize the possibility for subjectivity in the assessment of whether certain impacts of an evaluated defence would be considered positive or negative. Furthermore, it is possible that a single outcome could be considered as both in the same evaluation, when analysed from the perspective of different stakeholders. For example, a defence that introduces the ability for service providers to distinguish between devices behind a network address translation (NAT) gateway [196], could be perceived as a positive for government security services but as a negative privacy violation for end users. We recommend that in classifying such effects, the researchers should consider the perspectives of all stakeholders in the anticipated deployment environment.

5.4.4.4 Assessment Methods

Progressing from the previous sections, which discussed the various aspects that may affect the successful adoption of a DDoS defence, this section discusses tying them together in a way that promotes objective comparison. Since there are numerous factors affecting adoptability, several of which are qualitative in nature, a strategic approach is required to enable objective comparison between defences. The simplest approach is qualitative discourse, in which aspects are discussed and the differences between defences are highlighted discursively. The advantage to this approach is its simplicity, however it does not offer much in the way of ease of comparison. Building on such an approach, we can leverage tools of multiple attribute decision-making [197], such as transforming qualitative data into *pairwise comparison data* or *rank data*, and applying weighted aggregation to combine the aspects into a single ordinal metric for each defence, thus allowing the defences to be compared by ranking.

Although more advanced in promoting ease of comparison, this method is still lacking in that the ordinal metric produced for comparison, does not intuitively denote the meaning of adoptability. Furthermore, various aspects of adoptability are interrelated and so issues arise converting them to reliable quantitative data. For example, the adoption incentives are dependent on the deployment scale (since some defences increase their protection with scale of deployment), which in turn is dependent on the adoption incentives, which are both functions of time. Thus, a static method of analysis is not well-suited to a system of dynamic dependent variables, since the selection of which combination of interdependent values (system state) to apply is problematic.

We propose an alternative method for the assessment of adoptability: *adoptability by simulation* (ABS). By characterizing the essence of adoptability into a selection of meaningful metrics and constructing a detailed model of the adoption process for each stakeholder, we can observe, as the simulation progresses, how these metrics of interest evolve over time and

record their values at notable points in the simulation (such as the point of adoption saturation). As with the experiments carried out to obtain the true effectiveness of a defence, the closer the model aligns with the reality of the anticipated deployment environment, the more representative the results of the experimentation. The rest of this section is dedicated to detailing how the relevant adoptability aspects are incorporated in a simulation model, and how the simulation model can be used to extract measurements of adoptability.

Let us begin by deriving an adoptability simulation model itself. As mentioned before, part of the limitations of discursive and weighted aggregation methods of adoptability assessment is their inability to handle non-static and interdependent adoptability variables in a manner conducive to fair comparison. The key problematic variable in the case of adoptability was adoption incentives, which determines progression of adoption scale, but is also itself determined by the scale of adoption itself, for certain defences. To model adoption incentives, we must take into consideration the type of stakeholder, since adoption criteria may vary between stakeholder types.

We consider *participant stakeholders* of a defence as stakeholders who need to take some form of action in order for the defence to be adopted. We assume these stakeholders are rational and selfishly motivated, such that they act in their perceived best interests. Thus, the widespread adoption of a defence is dependent on the individual decision matrices of participant stakeholders at any given point in time. The decision of the rational stakeholder would answer the following questions:

1. *Net Incentive* – Does the perceived benefit of adoption justify the resources required. These resources include time, effort and costs (including opportunity costs).
2. *Means* – Does the stakeholder in question possess the resources required to take the action towards adoptability.

Thus, a decision to adopt is taken by the participant stakeholder if and only if said participant stakeholder has sufficient incentive *and* sufficient means, otherwise a decision is taken not to adopt. The net incentive can be derived from an equation that weighs the benefits of adoption to that particular stakeholder (since we assume selfish motivation) against the costs of adoption. This benefit can be quantized from assessments of the true effectiveness and any peripheral benefits offered by the defence to the participant stakeholder in question at the time (and level of adoption) of consideration.

This model of individual stakeholder decision matrices governing discrete adoption events over time lends itself to discrete event simulation (DES) [198] which has the ability to represent complex behaviour of individual agents and their interactions between each other and their environment [199]. A flow diagram of DES is illustrated in Figure 39.

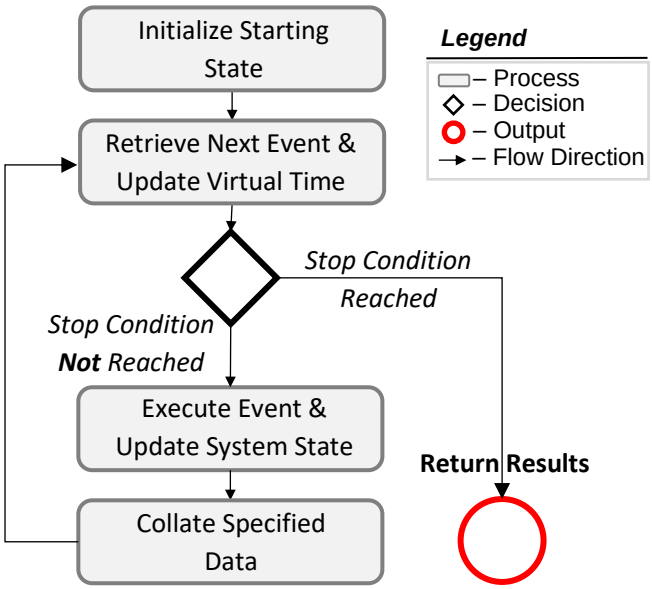


Figure 39: Discrete Event Simulation Flow Diagram

The *initialize starting state* process involves setting the simulation parameters, including the simulation (virtual) time, and scheduling initial and recurring events. Once initialized, the soonest event following the current system time is retrieved and the simulation time is updated to that of the retrieved event. Checks on whether a stop condition for the simulation has been reached. Such conditions include the simulation time reaching a specified limit or a part of the simulation environment reaching a specified state, or the null event object being retrieved – indicating no further events scheduled. If no stop condition returns true, the event is then executed and any changes it causes to the system are then applied – such changes also including scheduling of new events where required. Simulation data designated for extraction is then collated and the process repeats until a stop condition returns true.

In this model, we can observe that the decision a participant stakeholder makes, on whether or not to adopt a defence, can be modelled as a scheduled event in which the inputs to the decision matrix are variables forming part of the state of the system and following a decision to adopt, the state of the system modified accordingly. This does not necessarily mean that following a decision to adopt, the completion of the adoption process is instantaneous, but rather that a *completion of adoption* event is scheduled for that stakeholder which causes the modification of any relevant state variables – such as the defence's true effectiveness – at the appropriate virtual time.

We introduce the notion of *outside investment* as a measure of adoptability. Outside investment refers to the monetary amount required to progress and sustain adoption, where adoption would otherwise have stagnated or receded, due to insufficient net incentives or means of participant stakeholders. For an individual participant stakeholder, the outside investment is the monetary value required to either 1) produce a positive net incentive such that the decision is taken by the stakeholder to initiate or sustain adoption (of a defence under

evaluation), or 2) bolster the means of said stakeholder such that the initiation or sustenance of adoption is possible.

For example, in an adoptability simulation, the defence under assessment may be expensive enough that it satisfies the decision matrices of only a portion of all the participant stakeholders in the simulation, such that after that proportion has adopted the defence, the growth in adoption of the defence stagnates. This is where the notion of outside investment comes in. The outside investment is applied to progress the defence's adoption, and its cumulative amount tracked and stored,

Thus, outside investment is able to provide a comparative indication of the artificial resources required to roll-out a defence. There's flexibility in the way in which outside investment can be applied in a simulation but it is recommended that the strategy of application promotes lowest cost, firstly, and secondly lowest (adoption) time.

The other metrics worth noting and tracking for the purpose of assessing adoptability are total costs and global benefit; where total costs include the additional costs incurred by participants stakeholders as a consequence of decision to adopt, and any outside investment applied to facilitate adoption.

We define the *benefit* a defence provides as the weighted sum of the true effectiveness afforded to each server, where each server is weighted according to the typical resources consumed by the clients of the service. We consider the resources consumed by the clients of a server instead of the number of clients, since it provides a more representative accommodation of the beneficial effect of a defence, which is to preserve legitimate resource consumption whilst averting malicious resource consumption. Should the number of clients of a service be used as a weight instead, it would cause the metric to exaggerate the benefit of a defence for services that require less resource consumption per user. To promote objectivity, since the true effectiveness is a function of resources consumed rather than the number of clients that consumed it, we consider that the measure of the benefit of a defence should consistently align with the metric of true effectiveness.

Therefore, as the simulation progresses and adoption increases, the global benefit should be part of the specified data that's collated to formulate the simulation output.

Assessing adoptability through simulation is fitting since the other aspects of adoptability (feasibility, viability, peripheral benefits and adverse effects) will be reflected in the specified outputs of the simulation. For example, scalability limitations of a defence will limit the global benefit output and the peripheral benefits of a defence may act to reduce the outside investment required for that defence.

The outputs of the adoptability simulations (total costs, outside costs, global benefit and adoption percentage) are plotted over in order to facilitate comparison of the adoptability

between various defences. From the output data, assessors can calculate the values of further metrics that could be formulated according to the assessor's needs. For example, global benefit divided by total costs.

Despite its many strengths there are also many weaknesses to the assessment of adoptability by simulation. These limitations derive mostly from the complexity and unknown variables governing the real environment to be simulated. These result in simplifications, assumptions and or estimations that must be made, which erode representative reliability in the simulation output. We shall now discuss several.

A key area of complexity lies in the economic aspects of the simulation. Assumptions concerning, discounted cash flow models, going concern status, market entrants would need to be made. Furthermore, the means and model perceived benefit of participant stakeholders would need to be estimated.

One recommended assumption is that all necessary technology required for participant stakeholders to take action, is available for purchase at the start of the simulation. This assumption skips the design and implementation phase of equipment suppliers and thus simplifies the simulation. We believe that this is appropriate however, since any adoptability differences between defences that are introduced by such a phase, can be appropriately reflected in the costs of the technology required.

Another assumption made of the simulation model, is that of monotonically increasing adoption percentage. These are sustained in the simulation by increases in outside investment, where the adoption percentage may otherwise have fallen.

5.4.5 Comparison

The concepts and methods described in this section are intended to facilitate the objective comparison of the data collated from the empirical phase of the CED3 process, such that the strengths and weaknesses of the evaluated defences can be meaningfully highlighted and contrasted. Additionally, this section discusses the linkage between true effectiveness and adoptability when comparing the evaluated defences.

The primary tool proposed for comparing the performance of DDoS defences is the *DDoS Defence Map*. The idea of a DDoS defence map is to provide a way to present the effectiveness of a defence in a common context, highlighting the landscape of possible attacks that said defence can and, crucially, cannot mitigate against. The CED3 defence map achieves this by mapping defence performance to scopes of attacker strategies to allow the distinction between scopes of differing effectiveness within a visual 2D plane.

The defence map is a grid of areas, with each grid area denoting a remit of possible attacks that derive their common properties from the column and row headers within which they are situated (shown in black and white rectangles in Figure 40). These headers are independent

variables and should be selected according to the defences under evaluation, to highlight differences in their scopes of performance. In the example map on Figure 40, the row headers (to the left of the grid) indicate the type of protocol used in the attack, whereas the column headers at the top represent the attack strategy employed by the attacker. “Direct” in the column header refers to attacks that directly target a victim, whereas “indirect” attacks target the resources upon which a victim depends, such as link-flooding attacks [200], [201]. Reflection attacks are those where the attacking agents recruit unwitting accomplices to attack the target, whereas in un-reflected attacks, the attacking agents attack the target without mobilizing assistance from unwitting accomplices.

The value entered in each rectangular scope is the normalized true effectiveness (η_x) for that scope. The value of the normalized true effectiveness is obtained by dividing the true effectiveness within the attack scope when the defence is deployed by the value of the true effectiveness within the same scope when **no** defence is deployed. The greater this value, the greater the true effectiveness of the defence within the given scope. To facilitate ease of visual comparison, greater values of effectiveness are also depicted by brighter shades of red in the defence map.

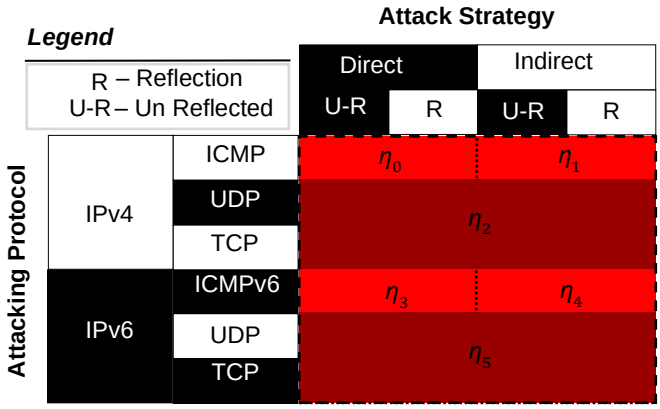


Figure 40: An (hypothetical) example CED3 defence map

It is possible for the scope of each major grid area to be further divided, where each sub-scope would represent a remit of possible attacks that are mutually exclusive between scopes. This may make sense to highlight nuances in the defence performance. However, a general rule is that no scope or sub-scope should restrict the attacker’s capability such that the attacker, within the defined scope, cannot overcome the defence in question. For example, the total attack data rate must not be bounded, the maximum botnet size must not be curtailed, and the per-bot-data-rate should only be restricted by network and device limitations that are common to all test scenarios.

The normalized true effectiveness values depicted on the defence maps are predicated on the context of the evaluations from which the values derived. As discussed in section 5.4.2, the

context of evaluation comprises of the models and assumptions that govern the anticipated deployment environment. Since the assumption of the scale to which a defence is deployed can affect the true effectiveness of a defence, deployment scale also forms part of the context of evaluation. It is recommended that the normalized true effectiveness values depicted in the defence maps, represent a context with the assumption of full deployment of a defence. This is so that the comparisons of true effectiveness between defences can be made the to capture the differences between defences at their full potential. However, the assumed scale of deployment is the prerogative of the evaluating entity and should be decided according to their evaluation requirements, provided the selected scale is made to be consistent between defences.

The evolution in the true effectiveness of a defence as the scale of deployment progresses is captured by the adoptability assessment process, described earlier. Since the output of the adoptability assessment by simulation (total cost, cumulative outside investment and global benefit) can be plotted against time, the comparative analysis of the adoptability of evaluated defences can be facilitated by plotted graphs. In these graphs the independent variable is time (on the x-axis), the dependent variables are the output metrics (on the y-axis), and each graph features the resulting curves of the respective defences.

Chapter 6: CED3 Evaluation

6.1 Overview

This chapter presents the application of the CED3 Framework, proposed in Chapter 5: , to the evaluation and comparison of a selection of diverse defences. The purpose of this chapter is to demonstrate the practical application of the CED3 framework in achieving its goals of evaluating different types of DDoS defences in a way that enables them to be commensurably compared. Additionally, one of the defences included in this evaluation is the DiDoS architecture presented in Chapter 3: Thus, by including DiDoS among the candidate defences, the CED3 evaluation can additionally serve to further evaluate the DiDoS contribution and subsequently position it in a wider context.

This section is organized to reflect the process of the CED3 evaluation. The chosen defences and the consequent design of the simulation environment, following a structured analysis of the chosen defences, are presented in section 6.2. The threat model, including the experiment plan are discussed in section 6.3.

The results discussed in section 6.4, begin with an assessment of the UEQ metrics of the CED3 framework, followed by a presentation of the simulated results of the planned test scenarios, and conclude with a discursive evaluation on adoptability.

The chapter concludes with a discussion of the findings of the CED3 evaluation, including an assessment of the utility of CED3 and its limitations.

6.2 Environment

6.2.1 Defences

This section details the defences chosen for the CED3 evaluation. Practical considerations, such as the adequacy of available information about the solution to enable its sufficient reproduction and evaluation, were taken into account. Additionally, the reproduction complexity of candidate defences was also considered to ensure enough time for implementation and evaluation of all chosen candidates.

We chose the following defences to be comparatively evaluated using the CED3 framework:

- Increasing the Victim Capacity
- Passport [78]
- TrustGuard [47]
- and DiDoS [48]

Aside from DiDoS, which is a contribution of this research project, all were sourced from the literature and were chosen based on their diversity, mitigation completeness and relative simplicity; since they each apply different mechanisms, satisfy all defence phases from detection to mitigation and could each be implemented in reasonable timeframes. Mitigation completeness refers to each defence being an integrated DDoS solution and not just a partial detection or mitigation mechanism. This criterion helps reduce evaluation assumptions and, thus, avoid misleading or inaccurate results.

Passport is a proposed network scheme that eradicates inter-AS spoofing by identifying spoofed packets through failed verification of message authentication codes (MAC) appended to the packet headers. Identified spoofed packets are then demoted and discarded by intermediate and end routers respectively.

TrustGuard, the most complex of the three, employs a dual macro and micro level analysis that trigger a probabilistic drop filter to remove attack traffic. Once the *macro-level anomaly indicator* is triggered for a particular destination address, *micro-level anomaly indicators* are used to direct the probabilistic drop filter to target malicious packets. The equation for the macro-level anomaly indicator is:

Equation 14

$$H(A) = -\sqrt{i_{max}} \cdot \sum_{i=1}^{12} p(x_i) \log_2 p(x_i), \qquad (1 \leq i \leq 12)$$

Where x_i refers to the i'th packet-size level, $p(x_i)$ refers to the probability that a packet has a size that places it in said packet-size level; and i_{max} is the packet-size level in which most packets fall for a given destination address. Further explanation of TrustGuard, including equations for the micro-level credit accumulation and the probabilistic drop filter can be found in [47]. Certain assumptions were necessary for the implementation of TrustGuard due to ambiguities in the paper. Firstly, since it was unspecified, we calculated the probability function $p(x_i)$ based on the absolute probability of packets incident on the TrustGuard device. This probability distribution was seeded with an even distribution of probabilities and dynamically updated for each incident packet, during the simulation. The updates continued for the duration of the simulation regardless of whether an attack was occurring or not. A second assumption necessary for the implementation of TrustGuard was the positioning of the TrustGuard devices. Following clues in the paper that each TrustGuard device handles packets of multiple source addresses heading to multiple destination addresses, we placed the

TrustGuard functionality between the core and the client-side networks. We chose the value of the macro-anomaly indicator threshold TH_H to be 1.3 based on a preliminary assessment (see Appendix B.3 Deeper Divefor details).

Increasing the victim capacity, whether by recruiting the resources of a third-party network security service provider, is identified in the literature as part of the state of the art in DDoS defence [30], [32]. We have chosen a factor of 2 to increase the victim's capacity to service clients and shall subsequently refer to this defence as "doubling the victim capacity". When owners of potential victim servers implement this defence the factor of increase is determined by capacity before the upgrade and that after the upgrade, which, naturally, varies between capacity owners and incidents of upgrades. We are considering the scenario where the victim upgrades its capacity (including bandwidth, memory and CPU capacity) but not its topology (including the number of degrees of connection to the wider network). The factor of 2 was arbitrarily chosen, but, as shall be shown, the pattern of results is indicative factor increases in general.

6.2.2 Network

Figure 41 illustrates the test-environment topology produced which consisted of client-side networks, server-side networks and a mesh core connecting the two. This design accommodated the deployment requirements of the four defences in that: Passport mitigation was deployed in the core, TrustGuard was deployed at the client side, doubling of the victim capacity occurred at the server-side links, and DiDoS was present on all three network segments. Full deployment of each defence was implemented for the appropriate test scenarios.

Due to the speed, control, scale and low-cost, simulation was chosen as the means of empirical evaluation. Further considerations on the choice of simulation are detailed in Chapter 4: . NS-3 [175], a discrete-event open-source network simulator that is developed and actively maintained by the research community, was chosen as the software for the simulations due to its extensibility, which allowed the custom implementation of the candidate DDoS defences. Further justification of the use of NS-3 can be found in Appendix A.1.

Multiple NS-3 simulations, written in C++, were orchestrated via BASH [176] scripts submitted to a SLURM [177] hypervisor. These ran on the ARC computing cluster [178], taking hundreds of thousands of CPU hours to complete.

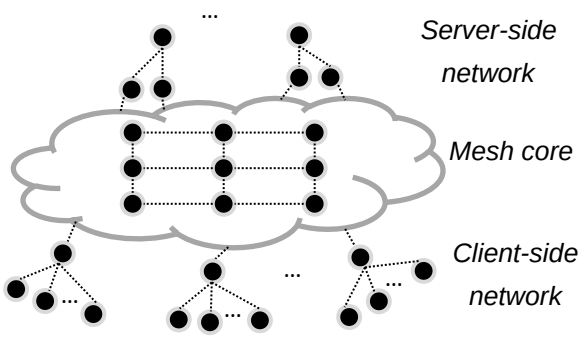


Figure 41: Test-environment topology

The test environment topology consisted of approximately 500 nodes with 1 Gbps core links, 100 Mbps server access links and 10 Mbps client access links. Following preliminary baseline simulations, we determined 100 Mbps server bandwidths provided an acceptable balance between server bandwidth realism and overall simulation time. As shown in Figure 42, the pattern of UEQ degradation remained the same as the server bandwidth varied over 3 orders of magnitude, with each line crossing the UEQ threshold at the same attack strength (expressed as a percentage of the victim bandwidth).

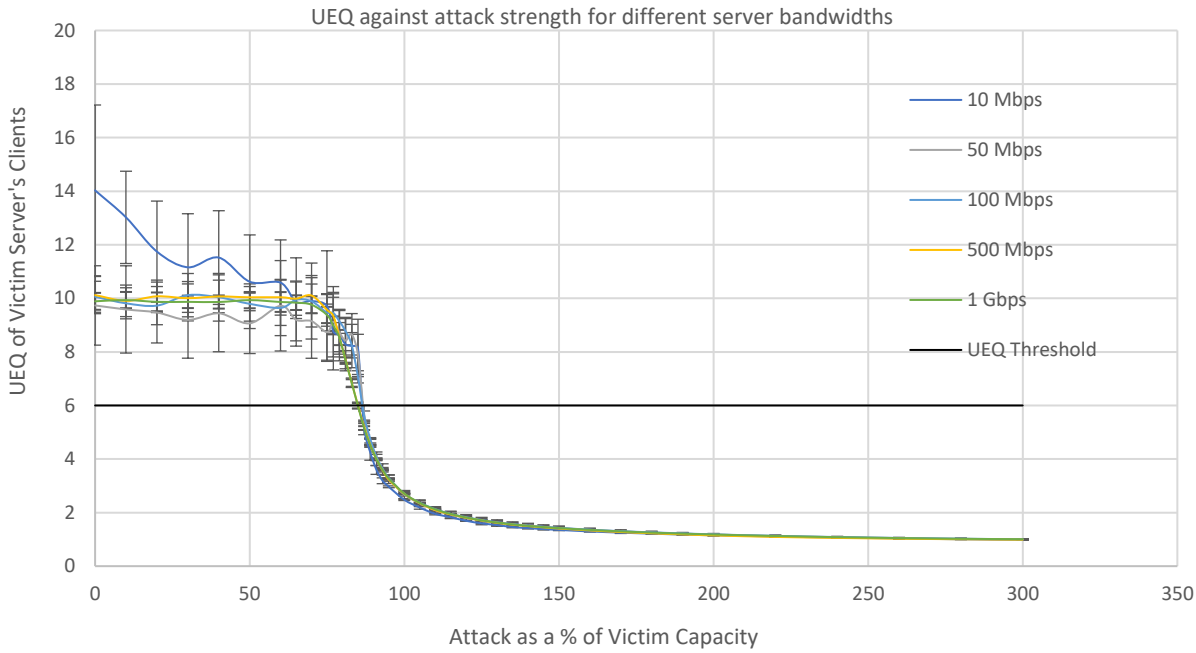


Figure 42: Preliminary baseline simulations with no defence deployed

However, the simulation time for each scenario with 1Gbps server access links was up to 4 days for a virtual attack duration of 5 minutes, whereas with 100 Mbps server access links, the simulation time was less than 4 hours. Hence, we decided to proceed with 100 Mbps server links since each scenario would need to be repeated (in this case, 5 times) to produce a standard deviation (as plotted).

Since device implementation weaknesses, and application layer attacks lie outside of scope, it was assumed that each end server node could process all of its received data.

6.2.3 Legitimate Traffic

The legitimate traffic is an important part of the evaluation context and, as with topology, should ideally match the anticipated deployment environment as closely as possible. For this evaluation, we derived aspects (such as the mean packet size and data rates) of the legitimate traffic from the literature [174], [182], [183], since there was no particular anticipated deployment environment. Such aspects included mean packet-size and mean data-rates, which were randomly generated and normally distributed. Legitimate traffic consisted of both UDP and TCP traffic of equal split. The legitimate clients were distributed across the nodes of the test environment and supplied data to servers, causing 15% utilization of the server bandwidth. The mean packet sizes for UDP and TCP packets were 623 bytes and 924 bytes, respectively; and the mean data rates of UDP and TCP clients were 30 Kbps and 50 Kbps, respectively. The attack traffic consisted of malicious UDP senders exhibiting a mean data rate of 1 Mbps.

Echo applications running on each server reflected the received payloads back to the sending clients. A transaction in this case was defined by the receipt of a certain number of packets at the receiving server. The specified number of packets varied between clients but was chosen based on the sending rate of the client to enable the completion of said transaction in 30s (assuming no dropped packets). In order to generalize the measurement of transaction times over the simulation, the transactions were randomly staggered from the start of the simulation.

6.3 Threat Model and Attack Scenarios

The threat model employed for this evaluation was an attacker in control of a distributed botnet of attacking agents capable of launching coordinated UDP flood attacks. The attacker was able to increase its attack strength and manipulate its attack characteristics in ways that could undermine the defensive mechanisms of each defence in order to cause denial of service as cheaply as possible. The attacking agents were implemented as custom NS3 applications running on distributed nodes and the attack strength was increased by increasing the number of attacking agents, which were randomly distributed across the test network. Out of scope of the attacker capability was the execution of intra-AS-domain spoofing and the compromising of autonomous systems and network routers.

From the structured theoretical analysis carried out on the three defences, two metrics of distinction were identified: packet size, for TrustGuard and packet validity for Passport. Thus, in order to characterize the performance of these defences, a matrix of attack scenarios was simulated for each of the defences. For Passport, the axes of variation were attack strength and packet validity, and for TrustGuard they were attack strength and packet size. Since

DiDoS is reputation-based, the metrics of distinction were the scale and quality of the reputations being applied, which were dependent on the time spent learning the reputations and the adoption percentage of the architecture. Since an attacker has no control over the time a reputor spends learning about its reputees, the axes of variation for DiDoS were attack strength and adoption percentage. Each scenario was repeated several times with differing attack and legitimate traffic that upheld the same chosen characteristics (of bandwidth utilization, mean packet-size and mean sender rate) thus enabling a standard deviation to be obtained for each matrix element. In this way, tens of thousands of attack-scenario simulations were executed on the ARC cluster [202], the results of which are presented in the next section.

6.4 Results

6.4.1 Metrics

The metrics chosen to represent the user-experienced quality of service (UEQ) are observed in this section to verify their correlation with DDoS attacks, and thus their suitability for inclusion in the UEQ calculation (Equation 13). The test scenario producing these results was the network described in section 6.2 without any DDoS defence deployed. Attacks of increasing intensity were applied and the resulting impact on the UEQ metrics: Loss, Delay, Jitter and Transaction Time, were monitored. Figure 43 presents the resulting variations of these metrics. The graph plots these variations as percentages of their original values when there was no DDoS attack.

Packet loss was calculated as a percentage on a per transaction basis by recording the number of packets belonging to a transaction that were dropped in any part of the network and dividing the tally by the total number of packets sent as part of the transaction. The control offered by the NS-3 simulation software allowed the flexibility to capture packet drops anywhere in the network and correlate them to particular transactions, using the tracing and tagging functionality. Delay was measured by time taken for a packet of a legitimate transaction to reach the victim server from the moment of its sending. Since each transaction sent many packets over its duration, a weighted exponential moving average function was used to capture a generalized value for delay. Measurements for the delay of a transaction ended once the transaction completed.

Jitter was calculated by, for each incident packet of a flow of a legitimate transaction, measuring the time between the latest packet arrival and the previous packet arrival, and subtracting this value by the time duration between the previous packet arrival and the packet before the previous; and applying the absolute value of this subtraction to a weighted exponential moving average function in order capture a generalized value for jitter. Transaction time was measured by the time taken for server to finish receiving a certain number of packets sent from a legitimate client, where the timing begins from the moment the first packet of a particular transaction is received. The sending rate of and packet size of

each client was random and normally distributed. However, the number of packets required to be received to complete a transaction was chosen such that at the specified data rate and mean packet size, the transaction would complete in approximately 30 seconds, under non-attack conditions. Percentage variations in transaction time were calculated by dividing the deviation of the new transaction time from original transaction time when no attack is present.

All metrics were captured on a per transaction basis, which was then collated for all transactions of a single simulation, producing an average value for each metric of the simulation. These averages were then combined with the averages of repeat simulations – consisting of the same inputs but different random generator seed values – to produce a group average and a standard deviation.

As we can see the metric that varied the most within the range of attacks strengths simulated was jitter, followed by packet loss. However, the higher rate of increase of packet loss towards the end of the range suggests the possibility of an intersection between the curves for some greater strength of DDoS attack. The metrics begin to change at roughly the same point as expected, however jitter and delay are the first to be affected by increased attacks followed by packet loss and transaction time. The ordering of responsiveness to increasing attack can be more clearly seen in Figure 43 (c). The explanation for this ordering is that, as the attack strength increases, the router queues begin to fill up which increases the delay due to increased queueing also increases the jitter since the queuing combined with the increasing number of random (attack) packets disrupt the cadence of each legitimate flow. The percentage variation in delay was so small relative to the other metrics that we plotted it separately in Figure 43 (b) to more clearly illustrate its pattern. The reason the delay variation was so small when expressed as a percentage of the delay when no attack was present was that the link latency was much larger than the queuing delays added as a result of DDoS attack.

Figure 43 (c) illustrates how the UEQ value itself varies with its input metrics. Each value of each curve has been normalized against the largest value occurring in the curve, so that the relative pattern of each curve can be more easily perceived and compared due to the large differences in range of the UEQ and metric values. As expected, the UEQ is inversely proportional to each metric. We also note that the UEQ is highly sensitive to large variations in jitter, when the UEQ is close to its maximum. We observe the early randomness in the jitter values being expressed in the UEQ, where an increase in jitter results in a decrease in UEQ and a decrease in jitter results in an increase in UEQ. This phenomenon is due to little offsetting variation in the other metrics and the mathematics of dividing a constant K_1 by a variable. When a divisor is small, small variations produce large changes in the output value. However, when the divisor is large, those same variations produce smaller variations in the output value.

The section demonstrates the impact a DDoS attack has on the chosen metrics packet loss, delay, jitter and transaction time. One may query why all are needed to formulate the UEQ since they each follow a similar trend. However, as discussed in section 5.4.3, the impact of degradation in each of these metrics varies depending on the services concerned and these metrics can vary independently of each other. As an example, a defence that reduces packet loss by rerouting packets instead of dropping them, may adversely affect jitter and delay, which would be damaging to voice of IP services.

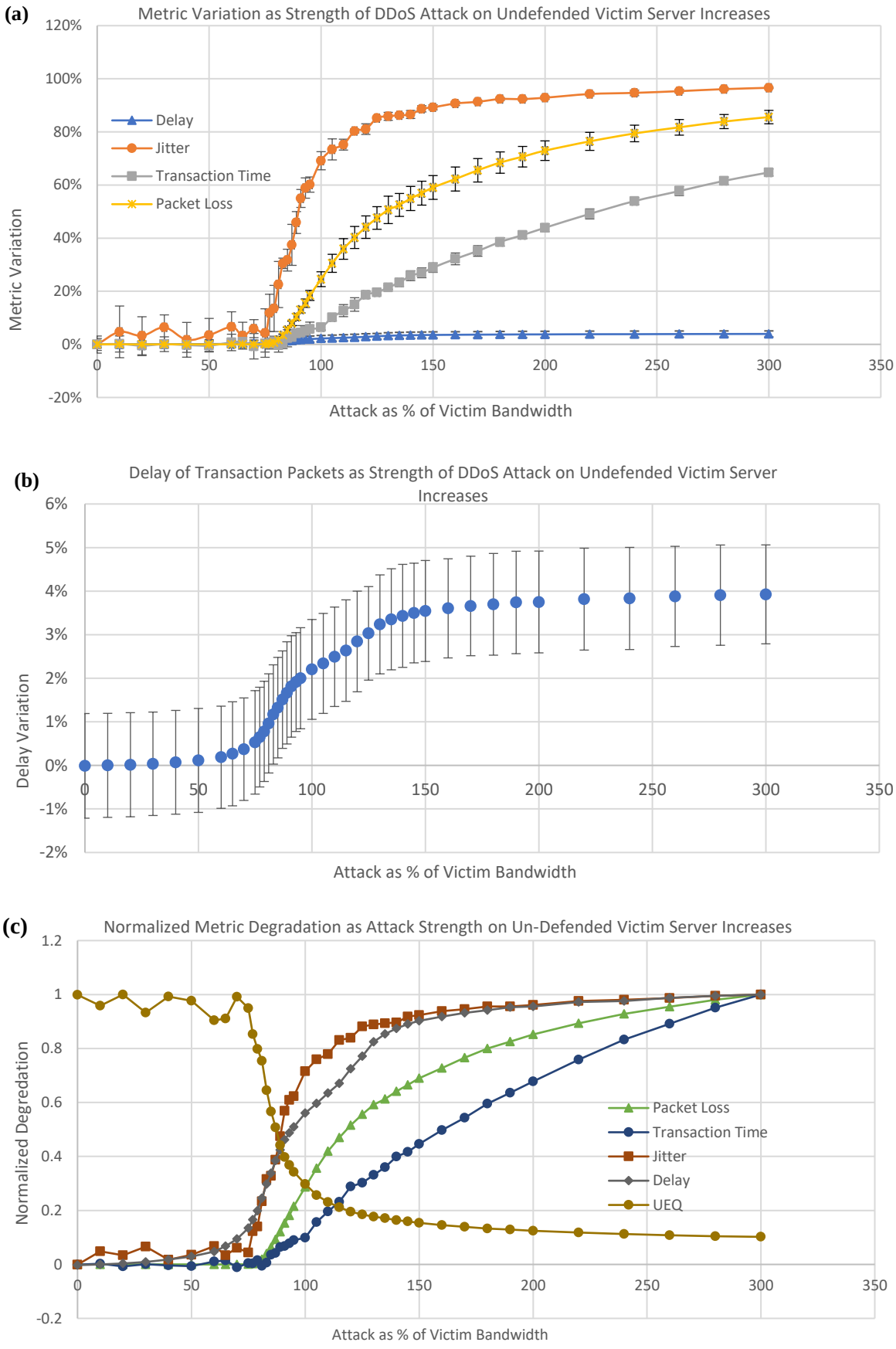


Figure 43: Graphs plotting, as DDoS attack intensity increases, the variation in metrics (a), the variation in delay (b) and the normalized degradation (c)

6.4.2 Defence Effectiveness

The results illustrated in Figure 44, Figure 45, Figure 46 are graphs showing how the user-experienced quality of service (UEQ) varied under differing attacks for the four candidate defences: doubling victim capacity, Passport and TrustGuard and DiDoS. Each plot point represents the average UEQ measured from a set of repeated simulations of a particular attack scenario (which represents one element of the simulation matrix discussed in section 6.3). The error bars indicate the standard deviation of the UEQ in the repeated simulations. Figure 46 illustrates a defence map depicting the normalized true effectiveness (NTE) of each defence over an attack-landscape, where the brighter the shade of red, the higher the value of the NTE depicted and the more effective the defence over the specified scope.

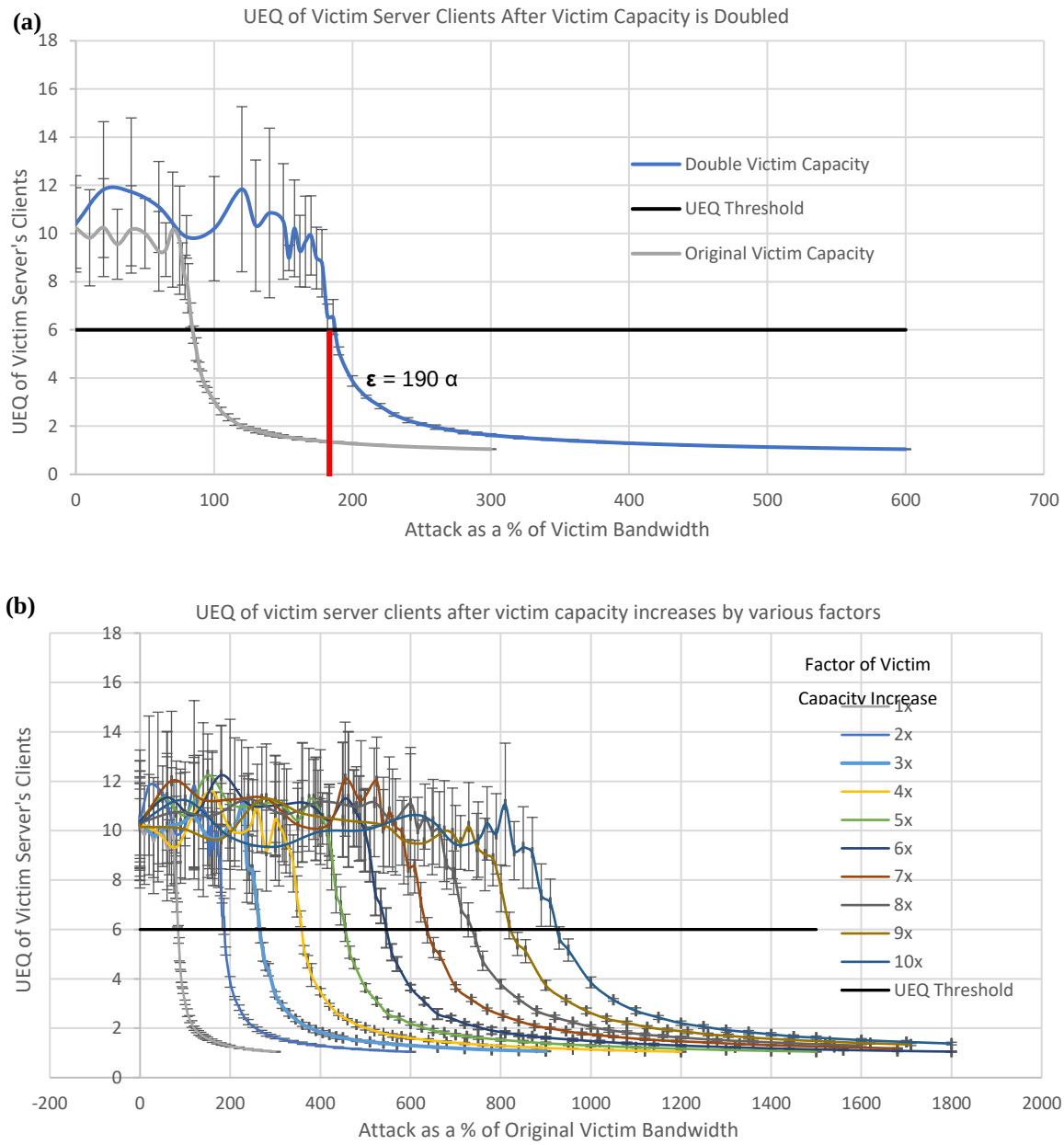


Figure 44: Graphs plotting the UEQ of victim server clients as the DDoS attack intensity increases for the defence of increased victim capacity at different factors of capacity increases

Figure 44 (a) illustrates two curves of UEQ as the attack strength increases. The first curve is in the test environment when the defence of doubling the victim capacity is deployed. Marked on this curve is the point at which the measured UEQ crosses the UEQ threshold, indicating the attack successfully caused denial of service. This point is the lowest strength of attack required to subvert the defence. It directly correlates to the true effectiveness ($\boldsymbol{\varepsilon}$) of the defence, in that it is the cost of producing this attack that is the true effectiveness of the defence. We measure this cost in *attack units* (α), where one attacking unit corresponds to the financial cost in inflicting an attack of total data rate equal to 1% of the original victim capacity, which in this case is 1% of 100 Mbps – 1 Mbps.

The second curve illustrates, for comparison purposes, the cost of inflicting denial of service on the victim server when no defence is deployed. We observe that the cost of effecting denial of service increases from 87 α, when no defence is deployed to 190 α, when doubling the victim capacity is deployed. There are no other curves on this graph because, according to the structured theoretical analysis, no metric of distinction was identified for this defence that could reduce the cost of inflicting denial of service.

Figure 44 (b) illustrates the effect on performance of increasing the victim capacity by various percentages. We observer an unsurprising linear effect. This graph was constructed to illustrate that the pattern of UEQ deterioration remains the same regardless of which factor of increase of victim capacity is arbitrarily chosen.

Figure 45 (a) illustrates the variation of UEQ as the attack strength increased, when Passport was deployed. The five curves plotted represent five mixes of spoofed and non-spoofed attack traffic. We see that when the attack consists of 100% spoofed traffic, Passport is extremely effective, able to withstand attacks greater than three times the victim bandwidth with no degradation of UEQ. However, as the portion of non-spoofed packets increases, the effectiveness of the defence reduces until DoS is able to be achieved at just 87 α, which is the same cost as that required to cause denial of service to the victim server when no defence is deployed. The true effectiveness of Passport is marked on the graph as 87 α, despite the defence being more effective in the presence of greater proportions of spoofed attack traffic. This is because we consider the cost difference, to an attacker, between using spoofed traffic rather than non-spoofed traffic to be negligible.

Figure 45 (b) illustrates the curve when TrustGuard is deployed. We observe that when the mean attack payload size is low, the UEQ drops to a level of approximately 4, and remains steady at that level, despite increasing attack strengths. However, as the mean attack payload size increases, the initial drop in UEQ lessens and increasing attack strengths begin to have an effect on the UEQ. The reason for this phenomenon is that when the mean payload size of the attack packets is small, it activates TrustGuard's probabilistic drop filter by causing the macro-level anomaly detection indicator to cross the activation threshold. The probabilistic

drop filter then drops packets from all flows towards the victim, with flows of greater average packet size experiencing less probability of being dropped. As the mean payload size of the attack packets increased, the magnitude of the initial UEQ drop reduced because fewer TrustGuard filters were activated by the macro-level anomaly indicator.

TrustGuard was the only defence to produce significant levels of collateral damage to the clients of the victim server. However, none of the defences, including TrustGuard produced significant collateral damage to the UEQ of clients communicating with servers-nodes that were *not* under attack.

The Effectiveness of DiDoS in sustaining UEQ under attack as the adoption percentage varies is illustrated in Figure 45 (c). As we can see the greater the adoption percentage the greater the strength of attack required to degrade the UEQ beyond the threshold. However, when DiDoS is at 100% adoption, no attack of strength within the range tested was able to degrade the UEQ of clients of the victim server. No true effectiveness limitation has been marked for DiDoS because, according to the threat model in section 6.3, adoption percentage is metric lying outside of attacker's control. Thus, should DiDoS be fully deployed by non-malicious network routers, in the test environment specified, DiDoS is able to mitigate flood attacks of intensities up to 3 times the victim bandwidth, irrespective of attacker sending strategies (such as packet size, data rate, or packet validity).

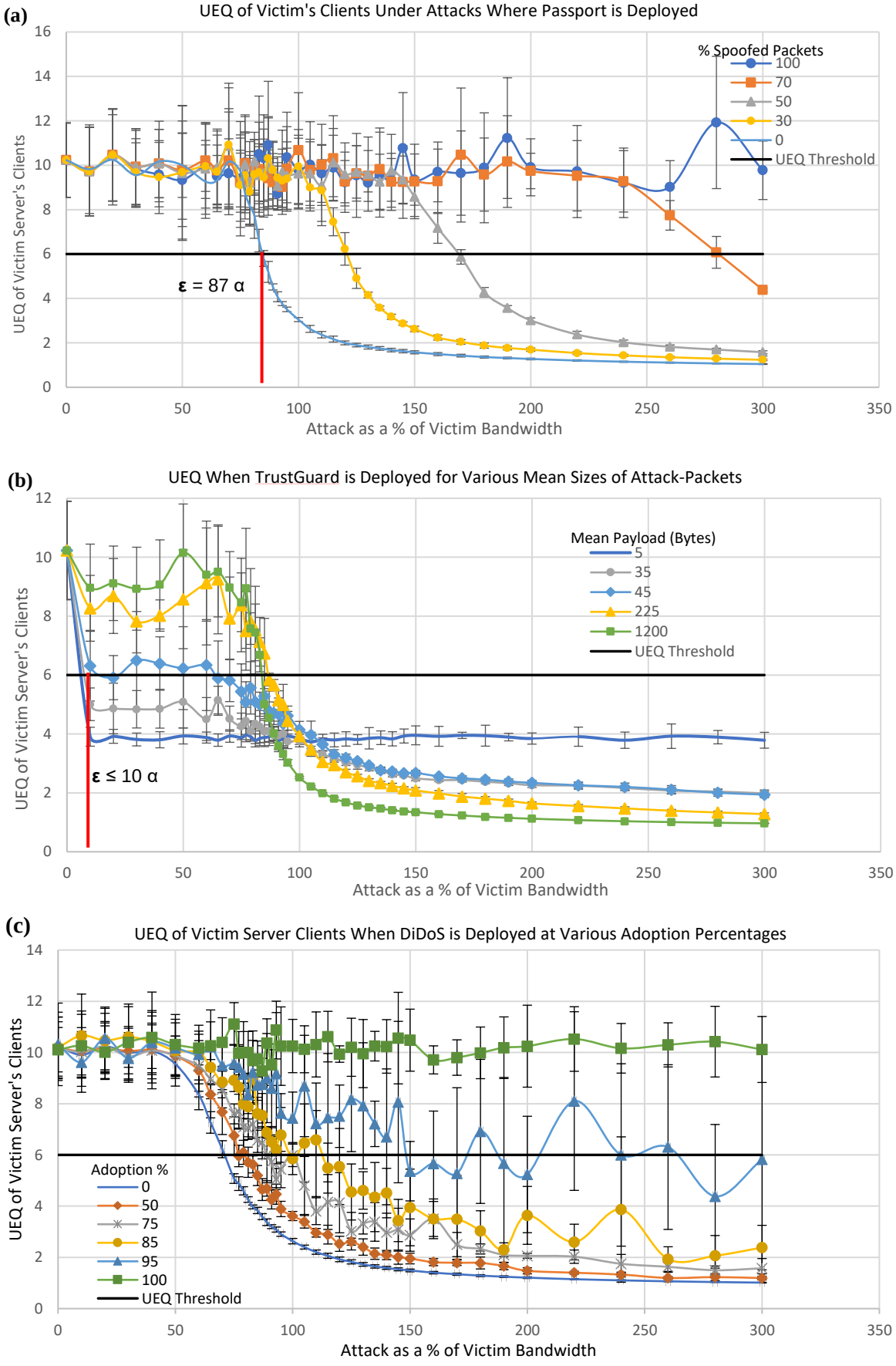


Figure 45: Graphs plotting the performances of the defences Passport (a), TrustGuard (b) and DiDoS (c) for different values of metrics of distinction as the DDoS attack intensity increased

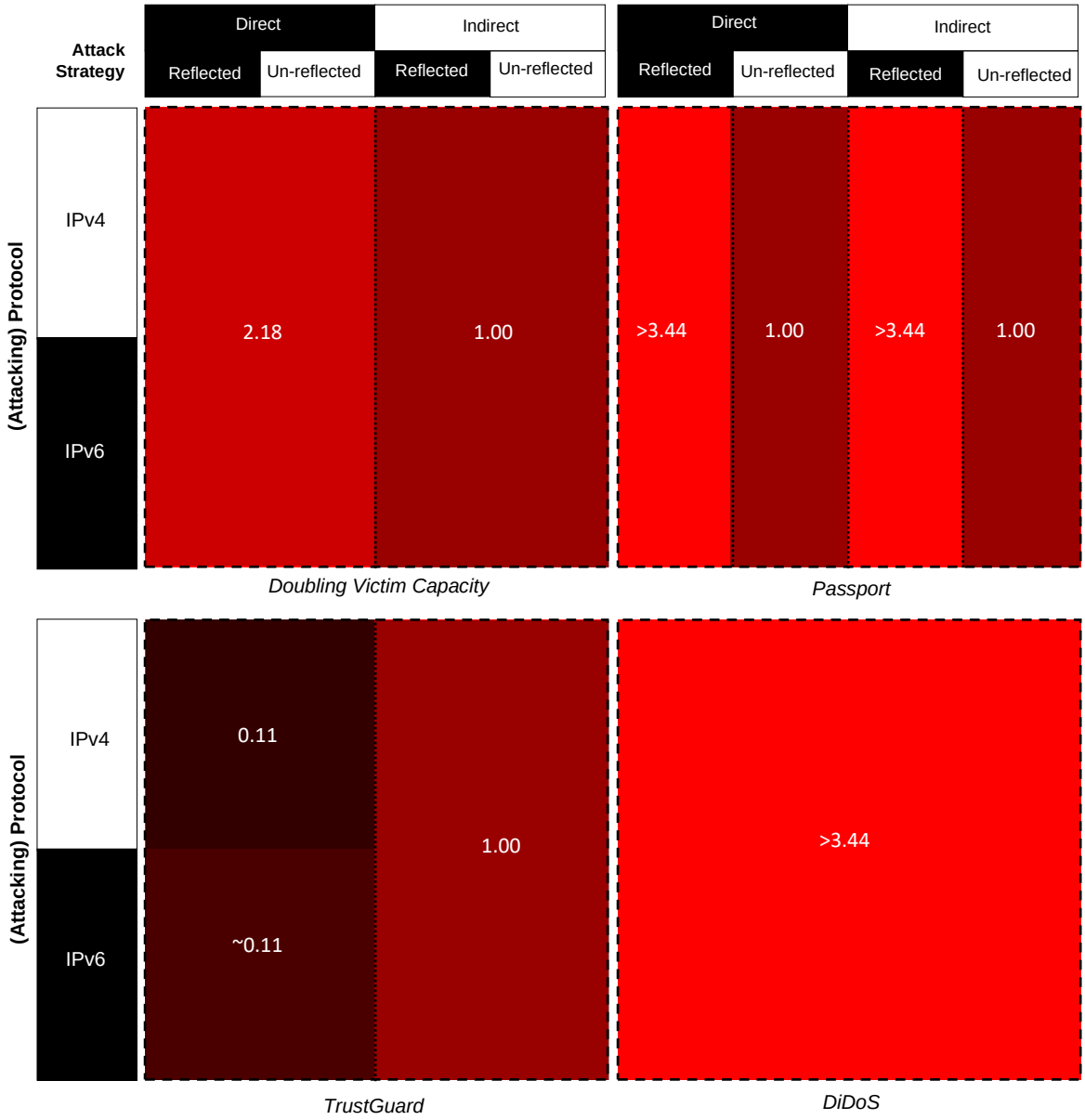


Figure 46: CED3 Defence Maps for Doubling Victim Capacity, Passport, TrustGuard and DiDoS

Figure 46 illustrates the defence map, which is a tool designed to aid the visual comparison of the true effectiveness of DDoS defences. The areas of the map correspond to the scope of attacks employed by the attacker and the brighter the shade of red of the area, the higher the true effectiveness of the defence within that scope. The values of normalized true effectiveness shown within the defence map were inferred from the effectiveness values obtained by simulation coupled with an understanding of the defences gained from the structured theoretical analysis process. Because the constituent values are normalized, inherent amplification effects of reflection attacks and the inherent ineffectiveness of indirect attacks are nullified, allowing the relative differences between the values to come from the defence performance itself.

The true effectiveness of Passport is independent of the attack protocol used and whether the attack is direct or indirect. However, as shown, since Passport mitigates against spoofed packets, it is most effective against reflection attacks - since reflection attacks require the use of spoofed packets.

For direct un-reflected attacks, TrustGuard has low attack cost regardless of protocol, since the attacker may choose to use low packets in its attack which would activate the TrustGuard filter. We hypothesise that this is also the case for direct reflection attacks assuming that spoofed packets can be crafted to generate reflection packets of low packet size, hence the areas share the same shade. For indirect un-reflected attacks, the attack packets are sent to destinations other than the victim but with the attempt to overload a certain link on the path used by the victims' clients. Therefore, if the attacker were to use small packets in this scenario, the TrustGuard probabilistic drop filters may activate causing both legitimate and attack packets to be dropped. However, since the filtering occurs based on the destination address of the attacking packets, no packets of the clients of the intended victim server would be dropped as a result since the attack is indirect, yet attack packets would still be being dropped. In this case we can see it would be more effective for an attacker to avoid activating the probabilistic drop filters by using larger attack packets. Hence for indirect attacks, a NTE of 1 that matches the true effectiveness when no defence is deployed is shown, since the most effective indirect attack is to avoid activating the TrustGuard queues. Since TrustGuard is sensitive to packet-size, we estimate that its true effectiveness for direct attacks will be higher since IPv6 have larger packets, thus requiring a greater flow of smaller packets (greater attack strength) to activate the TrustGuard queues. The greater size of IPv6 packets does not affect indirect attacks on TrustGuard since the lowest-cost attack strategy is to avoid activating the TrustGuard queues, as previously discussed.

The lower right quadrant of Figure 46 depicts the defence map for DiDoS. As shown DiDoS for the threat model and assumptions considered, the NTE is extrapolated to be consistently high irrespective of the attacker strategy employed. For un-reflected direct attacks, the reputations marked on each packet sent enable malicious and legitimate traffic to be separated and prioritized accordingly at the victim network thus allowing the UEQ of victim server clients to be preserved. The effectiveness is extremely high due to the use of priority queuing which prevents lower-reputed (attack packets) from being routed at the expense of higher-reputed (non-malicious) packets. For un-reflected direct attacks, since DiDoS mitigation occurs at any point of contention in the network, no advantage is realized by the attacker attacking indirectly. For reflection attacks, both direct and indirect, no advantage is gained by the attacker since spoofed packets are dropped before being reflected due to anti-spoofing mechanisms provided by the DiDoS architecture.

The reason the performance of DiDoS is so high is because 100% adoption is assumed, as we have shown in Figure 45 (c) that the ability for DiDoS to sustain the UEQ of clients of the

victim server whilst under attacks diminishes as the adoption percentage diminishes. This raises the question of whether 100% adoption is realistic, which in turn raises the valid question of whether these results are meaningful, if based on an unlikely assumption. Firstly, this evaluation, as well to demonstrate the evaluation of CED3, also intended to assess the defences at their full potential. Such analysis has merit when considering the long-term potential of a defence. In the cases of industrial applications of the CED3 method to practical decision making, we expect that practitioners could absolutely assess their defence options at different levels of adoptability – the output of which may be a series of defence maps representing the evolving mitigation landscape as the adoptability levels increase. However, we believe that the consideration of just one adoption percentage, indicative of the full potential of a defence, is sufficient for demonstrating the merit of the CED3 framework. Lastly, as has already been shown in the empirical evaluation of section 4.3.2, DiDoS is able to provide strong attack mitigation for authenticated clients of the victim server even when the adoption percentage is as low as 1%.

6.4.3 Adoptability

This section discursively evaluates the adoptability of the four defences. The purpose is to demonstrate how the criteria identified in section 5.4.4 may be considered in practice. The first step of assessment of adoptability is to identify any disqualifying factors, that may deem the defence unsuitable for actual deployment, due to feasibility constraints. The next step is to comparatively assess the defences against important non-binary criteria. These assessments are carried out under the same assumptions and threat model used to evaluate true effectiveness – specified in section 6.3. The final step involved the quantification of the assessment such that adoptability evaluation scores could be derived.

Firstly, no feasibility disqualifications for doubling the victim capacity (labelled *2x capacity* in Table 9) were found. It was the only defence in question that did not require a distributed deployment to be effective, since it operated on the victim network.

Despite the performance of Passport, TrustGuard and DiDoS all being affected by the scale of their deployment, we do not consider this a feasibility constraint for disqualification since their ability to be operational in the anticipated deployment environment is not itself prevented.

Table 9 presents the quantitative scoring of the adoptability of the four defences. As shown, doubling the victim capacity emerged with the top score followed closely by DiDoS. Passport scored 3rd and TrustGuard scored the lowest. The table was constructed by distributing 100 points over 5 variables and four defences such that the sum of all the individual defence scores is 100. Exactly 10 points (excluding the weight factor) was allocated per adoptability criteria, and the sum of the wights was fixed at exactly 10. The total score of each defence was produced by a sum of products operation between the column of the defence and the column

of weights. The function of the weight in this assessment method is to allow for the emphasis of particular adoptability aspects in the overall scores, which may be exercised to reflect the preferences and priorities of the evaluating party.

The global benefit refers to the strength of the DDoS mitigation benefit provided the defence and is described in section 6.4.3. The individual row scores were derived from the defence map (Figure 46), which depicts DiDoS has possessing a strong mitigation coverage of double that of Passport whilst, doubling the victim capacity offers a much weaker normalized true effectiveness than that of Passport. The global benefit of TrustGuard was specified as 0 since no area was identified on the defence map where TrustGuard offered superior true effectiveness.

Concerning peripheral benefits, TrustGuard and Passport are tied since Passport offers an anti-spoofing architecture, which has application outside of DDoS mitigation, whilst TrustGuard offers a means for attack detection. DiDoS was scored higher than both since it offers anti-spoofing protections at greater granularity, furthermore, the reputations formed by reputors of the DiDoS architecture may be useful in identifying compromised hosts and botnets, with use outside of DDoS attacks. Doubling the victim capacity shares the same score as DiDoS since, the additional capacity can be helpful to the server network owners in cases outside of a DDoS attack, for example by facilitating sustained service during flash crowds, without dropping connections.

Adverse effects encapsulates the negative effects and risks caused by a defence including collateral damage. Doubling the victim capacity scored top in this category since it offered no adverse effects, whilst TrustGuard scored bottom since it actually exposed the victim of attack to increased vulnerability of denial of service. Passport scored in-between the range since it introduces processing overheads and may drop benign traffic during configuration mismatches. DiDoS is also rated within the range, but slightly lower than Passport, for the same reasons as Passport, except that DiDoS requires additional packet processing due to the greater protections offered. Additionally, DiDoS may raise privacy concerns due to the possibility of the activity of an authenticated DiDoS device being monitored through the ID field in the DiDoS packet header. Adverse effects did not include costs.

A detailed cost evaluation, stemming from vendor cost data applied to a complex cost model – as in [203] – is outside the scope of this research project. We, alternatively, summarize costs by considering the number of segments of the anticipated deployment environment (Figure 41) that the defence is needed in order to achieve full deployment. Doubling the victim capacity, Passport and TrustGuard score equally in this regard, since only one network segment is affected for their deployment. For doubling the victim capacity, this affects the server-side network, for Passport, it’s the core network and for TrustGuard, it’s the client-side network. DiDoS, on the other hand, requires deployments in all network segments for the attainment of 100% adoption, resulting in its lowest score.

	Weight	**2x Capacity**	**Passport**	**TrustGuard**	**DiDoS**
Global Benefit	2	1	3	0	6
Peripheral Benefits	2	3	2	2	3
Adverse Effects	2	5	3	0	2
Cost	2	3	3	3	1
Adoption Incentives	2	4	2	1	3
Total:	10	32	26	12	30

Table 9: Quantitative Adoptability Comparison

Adoption incentives aimed to capture how closely the benefit of deployment of a defence was correlated with the deploying entity. Doubling the victim capacity scored the highest since it is implemented by owners of the victim network who directly benefit with improved resilience to DDoS. The lowest scoring was TrustGuard, since, when deployed at the source network, the mitigation effects are directed at other networks largely owned by other entities. Both DiDoS and Passport both offer anti-spoofing incentives for deploying entities – meaning that the deploying entity benefits because attackers will be less likely to spoof its address. However, DiDoS offers more granular anti-spoofing methods. Furthermore, benign client entities deploying DiDoS directly benefit from improved reputation scores which translates to reduced negative effects of DDoS attacks. Hence, DiDoS is rated higher than Passport. The reason why DiDoS is rated lower than doubling the victim capacity, is because for any benefit to be received in deploying DiDoS, at least one other network must also deploy DiDoS to enable report feedback for reputation learning and also to enact the prioritization of reputation levels embedded in compliant packet headers.

From this adoptability assessment, we can gain some insight into why increasing capacity is a favored method of DDoS mitigation in practice, despite not offering as much mitigation protection as other methods. The following section discusses the findings of the CED3 evaluation including the discursive adoptability evaluation.

6.5 Discussion

The results demonstrate how the CED3 method can be practically applied to evaluate and compare the *true effectiveness* of multiple defences. Despite this work being an early step towards commensurable comparative defence evaluation, the results demonstrate how the performance of different types of DDoS defences can be compared in a way that highlights their strengths, whilst also exposing their weaknesses. This aspect is enabled by the concept of defensive scopes, depicted in the defence maps of Figure 46. These scopes highlight differences in defense effectiveness to different scopes of attack, which in turn could help a practitioner select complementary defences. Because CED3 actively seeks effective lowest-cost attacks, the comparative output of this evaluation is expected to have longevity, caveated by the evaluation context (see section 5.4.2), which we believe is sufficient for proof of concept.

The process, however, is not without its limitations. Despite the many strengths offered by simulation, including environment control and repeatability, there is always the question of whether the simulation would be sufficiently representative of an actual deployment environment. One aspect that deviated from reality was that only the logical functionality of each defence was modeled so that possible transit delays introduced by each defence were not considered. However, we believe the aspects modelled using the NS3 software (including link delays, routing, queueing etc.) were sufficient to approximate the response of each defence to increasing UDP flood attacks and consequently demonstrate the usefulness of true effectiveness as a sustainable evaluation output.

For complex defences, possessing multiple metrices of distinction, the number of experiments could increase – potentially quadratically – for each additional metric of distinction of the defence. However, this increase can be mitigated by employing more efficient ways to find the lowest cost attack. For example, by using algorithms that find global minima of multivariate functions, where applicable.

The value of the degradation threshold of the user-experienced quality of service (UEQ), used to determine whether denial of service was achieved, was chosen arbitrarily at a UEQ level corresponding to packet losses of approximately 7%. The veracity of this threshold could be augmented by deriving it from empirical data sourced from experiments where participants are subjected to different degradation levels and are asked to note the point at which they deem the service which they are using to be untenable. Likewise, the structure of the equation by which the UEQ is calculated (Equation 13), could also be modified so that its exhibited degradation mimics empirical data. This is an open research direction. However, we believe that the equation used and the threshold chosen, were sufficient to enable the determination of whether an attack succeeded, as part of the process of demonstrating how to find the lowest-cost attack.

The financial cost of executing an attack is expressed in *attacking units*, which are units linked to the total data rate of the attack. We consider this to be appropriate in determining differences between attack costs, under the assumption that generating attacks with greater total bandwidth generally costs more since more bots would generally be required, which have a financial cost to acquire, operate and maintain [158]. Future work could involve the conversion of attack units into actual dollar values, which would enable more evaluation metrics such as *cost-effectiveness* or *value for money* to be obtained.

By the presentation of the normalized true effectiveness in scopes on a defence map, we have shown how the strengths and weaknesses of a defence can be highlighted. Doing so may help to facilitate the selection of complementary defences, since, by knowing the weaknesses of one defence, we may seek to combine said defence with another defence that happens to be strong in at least one scope where the former defence is weak. The process of selecting complementary defences, however, would also require some empirical verification to ensure that the combination of the selected defences is actually feasible and does not produce unwanted phenomena such as collateral damage.

We have shown how the adoptability of a defence can be assessed discursively against the CED3 adoptability criteria. We have also shown how the qualitative findings can be quantified in a table for comparisons. However, quantitative evaluation comparisons are only as objective as the objectivity of the processes that led to the quantification. By adjusting the weight values corresponding to each adoptability aspect in Table 9, we can obtain different adoptability rankings, such that Passport or DiDoS could overtake doubling the victim capacity with the highest adoptability score. This enables the evaluating entity to adjust the weights according to their priorities, which allows the accommodation of the requirements of different evaluating entities, however it also implies subjectivity in scoring. This subjectivity could be counteracted by formulating a more objective mechanism by which the weights are selected (perhaps by mass surveys of the participant stakeholders to determine or rank which aspects are most impactful to the decision on whether to adopt). However, such investigations are a potential avenue for future work. The subjectivity of the relative scores within the adoptability table could also be addressed by further research on the commensurable quantifications of adoptability aspects, using a common metric, such as cost, for relative comparison.

Concerning the CED3 defence map, it has been shown that using a DDoS attack taxonomy to construct a grid over the map can enable different areas of defensive strength and weaknesses to be highlighted and contrasted. For complex defences and analyses, producing a greater number of areas of strengths and or weaknesses, this can be accommodated by adding further scopes within each grid area or adding multiple maps as appropriate. For example, should a defence evaluation consider multiple adoption percentages of the evaluated defences, this could be illustrated by several defences. Furthermore, should a defence consider an additional

metric of distinction, such as flow data rate, the performance implications could be depicted by further scopes within the defence map. Although, handling increased defence or evaluation complexity is feasible, it is not necessarily aesthetic, as the number of scopes and or maps could soar.

The fact that the CED3 experiments were executed on a large high performance computing cluster [178] consisting of thousands of nodes and took hundreds of thousands of CPU hours to complete, may make the CED3 method appear outside the economic reaches of many researchers without access to such vast infrastructure. However, the number of CPU hours required to complete simulations can be reduced in a few ways, such as: by reducing the number of nodes tested, the bandwidth of the links in the network, the virtual time duration of each simulated scenario; and the number of scenarios tested. The last option is possibly the most impactful since by using strategic computational algorithms (to more quickly hone in on the attack scenario that produces denial of service at the lowest cost to the attacker) we can avoid excess simulations.

Chapter 7: Conclusions

Despite decades of research into its mitigation, the threat of Distributed Denial of Service (DDoS) remains significant and is growing, costing up to hundreds of thousands of dollars per attack. Conventional mitigation strategies, which involve increasing capacity and traffic scrubbing, have been shown by researchers to be unsustainable as DDoS attackers harness the growth in IoT for increased attack strengths.

During DDoS attacks, a victim network 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[1] signaling of sender (reputee[2]) 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. Reputors maintain the reputations of reputees through feedback sent from autonomous systems when their reputees have been involved in an attack. Therefore, the more reputees are involved in and contribute voluminously to attacks, the worse their reputation score. Using the information embedded in packet headers, DDoS attack victims are able to better identify and deprioritize malicious traffic, thus freeing resources to sustain service to legitimate clients.

In this way DiDoS bridges the gap between DDoS attack *destination* networks, which are better positioned to detect DDoS attacks but not track their globally distributed attacking agents, and DDoS attack *source* networks, which are better positioned to monitor malicious agents and embed downstream warnings but are not well-positioned to detect attacks and relay feedback accordingly. This is a novel contribution of this work and is further detailed in section 3.4.

The evaluation of DiDoS revealed that the reputation update algorithm is able to differentiate between malicious and non-malicious reputees and eliminate the in-attack packet loss of

[1] Downstream indicates a flow towards a destination address, whereas upstream indicates towards the source.
[2] Reputees are network entities whose reputations are managed by reputors through whom reputees communicate. (See section 3.4.2 for more information)

authenticated legitimate clients for global adoption percentages as low as 1%. However, when considering the packet loss of all legitimate clients (both authenticated and unauthenticated), an adoption percentage of 90% was required to achieve a significant (50%) reduction of in-attack packet loss. This dichotomy, between the in-attack packet loss experienced by DiDoS compliant and non-DiDoS compliant clients, served to indicate the benefit of DiDoS adoption to beneficiary stakeholders. At full adoption, DiDoS was shown to eliminate all in-attack packet loss for all clients.

In addressing research questions 3 and 4 (listed in section 1.3) concerning the evaluation of DDoS defences, we built on prior work by proposing a novel framework by which DDoS defences could be evaluated in a way that allows their strengths and weaknesses to both be highlighted and commensurably compared.

Starting from the postulate of *inevitable theoretical subversion*, which states that all DDoS defences are theoretically surmountable, we deduce that any meaningful evaluation of a defence must consider the cost to the attacker in surmounting the defence. From this we formulate the notion of *true effectiveness*, which is the minimum cost to an attacker required to surmount a defence within a given scope of attacker strategies.

The resulting framework called CED3 (comparative evaluation of DDoS defences), directs the evaluation process through a *structured theoretical analysis* followed by empirical data acquisition such that the strengths and weaknesses of defences can be identified and measured. CED3 leverages true effectiveness as a metric for commensurable comparison between defences and the highlighting of differences between defences when plotted on a *defence map*. In addition to providing a map for visualizing outputs, CED3 introduces a paradigm shift in DDoS defence evaluation from ascertaining "how well" a defence would work to discovering "what it would take to overcome it". This is achieved through the notion of true effectiveness of a defence, which promotes evaluation longevity, thus addressing the issue identified in the literature of defences being validated under favourable conditions, only later to be shown to be ineffective under different conditions.

CED3 also guides the consideration of important aspects to the adoptability of defences and proposes various methods by which adoptability aspects can be comparatively assessed, according to the requirements of the assessing party.

We applied CED3 to the evaluation of four diverse DDoS defences, including DiDoS. The theoretical and empirical findings were extrapolated onto a defence map, thus allowing them to be commensurably compared. DiDoS emerged as the defence with the greatest[1] true effectiveness followed by Passport, doubling the victim capacity and TrustGuard, respectively. However, when comparatively evaluating the adoptability of the defences,

[1] Greatest was calculated by the weighted summation of the normalized true effectiveness values on the defence map – weighted according to the map area each value covers.

doubling the victim capacity scored the greatest followed by DiDoS, Passport and TrustGuard, respectively. Increasing the victim capacity scored so highly as, despite offering the third lowest mitigation benefit, it offered the greatest adoption incentives (since the adopting party making the investment wholly benefits from adoption) and least adverse effects. In this way, the viability of a CED3-like process to evaluate the performance and adoptability of DDoS defences and commensurably compare them, was demonstrated.

Many insights obtained as part of this research also helped steer the direction and contributions of this work. Below is a list of several of these key insights that helped to characterize the research produced:

- *The most effective metrics that a DDoS defence can use for asymmetric facilitation are those that are difficult or costly for the attacker to manipulate and are linked to the attacker behaviour or the denial-of-service effect* – This insight among many others shaped the DiDoS architecture, leading to a flooding DDoS defence who's normalized true effectiveness remain consistent across diverse attack strategies. See sections 3.2 and 3.4.3 for more information.
- *Any DDoS defence is theoretically surmountable, and so it is not enough to consider whether a defence stops a particular attack, given that attacks are limitless, but rather we should identify what it takes to overcome the defence for more meaningful evaluations of defence effectiveness* – This insight led to the creation of the metric of true effectiveness, a fundamental component of the CED3 framework that allows commensurable comparison between diverse DDoS defences. See sections 5.3.1 and 5.3.2 for further explanation.
- *A single value is insufficient to capture the nuance of the effectiveness of many DDoS defences due to variability in performance under different contexts and attacks* – This insight led to the creation of the DDoS defence map, which allows the visual capture of this variation of DDoS defence effectiveness under the different contexts. See sections 5.3.3 and 5.4.2, and Figure 46 of section 6.4.2 for more information.
- Reputations should not be able to be increased by an action of a reputee otherwise malicious reputees could exploit that action to offset the penalties they receive for attacking. This insight, coupled with the need to prevent the flat-lining of reputee reputations, led to the idea of passive periodic accumulation described in section. See section 3.4.3.1for more details.

Limitations: Despite the numerous insights and contributions of this work, the research was not without its limitations. For example, the primary method of evaluating the DiDoS architecture, was with simulations using the NS-3 software. The limitation here is twofold. Firstly, due to the breadth of simulations required to test the various aspects of DiDoS, and the square relationship between simulation environment scale and time to complete the

simulation, the scale of the simulations had to be limited with respect to number of nodes, link bandwidths and DDoS attack size. This reduced scale is a deviation from the intended deployment environment of DiDoS. However, we believe that the scale carried out was sufficient to demonstrate the DDoS mitigatory impact of the DiDoS architecture as well as its performance variation under differing DiDoS adoption percentages and attack scales. Secondly, the use of simulated network environments instead of real-world networks is an inherent deviation from the intended deployment environment of DiDoS and thus a limitation of this research. However, as discussed in section 4.4, we believe that the simulated environment modelled enough detail and real-world behaviour to produce reliable conclusions on the mitigatory performance of DiDoS. Both these limitations (of scale and representativeness) can only truly be addressed by testing the performance of DiDoS over real-world infrastructure. One approach for attaining such an eventuality would be for a pioneering service provider to adopt DiDoS on its own live autonomous systems and observe the reputation convergence and attack mitigation impact over time. The learning process could also be intelligently accelerated by hiring real-world botnets to launch attacks on honeypots located within the DiDoS compliant autonomous systems.

Although DiDoS has been designed for incremental adoption, another limitation of the DiDoS architecture is the software and possibly hardware upgrades required for the existing networks. These upgrades are necessary to implement the DiDoS functionality, such as the cryptographic protection mechanisms of the DiDoS reputors and reputees or the reputation-based priority queues of the distributed mitigation mechanism. One avenue to navigate this limitation is to standardize the DiDoS architecture, as an ISO [204] or IETF [45]standard. Following this, vendors could implement the standards at the demand of Service Providers, which could be specified in the form of request for proposals (RFPs) or public tenders.

The combination of the concepts of true effectiveness and scopes in the CED3 framework is a significant step towards commensurable DDoS comparison because it captures the worst-case performance of a defence under various contexts, thus allowing relative strengths and weaknesses to be highlighted. However, this evaluation approach presents a limitation when it comes to comparing complex DDoS defences. This is because a significant number of tests may be required to identify the worst-case performance of each defence in multiple scopes. We believe this limitation fundamentally unavoidable, since it is a reflection of the reality of the innovative attacker that will attempt to overcome the defence. However, the strategy and efficiency of the search (to find the lowest-cost attack that can surmount the defence) may be able to be improved. Ways in which this can be achieved is a topic for future work.

The notion of structured theoretical analysis (STA) as part of CED3 is an attempt to provide objectivity in DDoS defence evaluations such that independent CED3 evaluations following the CED3 framework will reach the same evaluation conclusions every time. However, a current limitation of this analysis, is the room for ambiguity, which could potentially lead to

different comparative evaluation outcomes. A possible avenue to address this issue would be the strict formal construction of the STA, such that the rules of formal construction and the rules of formal analysis of DDoS defences are robustly specified. This possibility is an avenue for future work.

Future work: In addition to addressing the aforementioned limitations, the project presents many potential interesting and valuable avenues for future work. Firstly, is the implementation of the DiDoS architecture in hardware for the exploration of its application to live systems. The detailed specification of supporting functions of DiDoS is another avenue. For example, the protocols used to set up and maintain secret keys between reputors and reputes and between DiDoS peers could be specified and formally assessed in detail. Another possible direction could be the formal verification of components of the DiDoS architecture such as the attack feedback process and reputation management system.

DiDoS offers the possibility of a globally distributed network of evidence-based reputations able to identify malicious senders. This capability is currently focused on the IP network layer, but explorations could be made to see how such an asset could be leveraged. One possibility is for the per-packet reputation levels to be used to better inform application layer defences to help mitigate against application layer DDoS attacks or otherwise. Furthermore, there is the possibility to explore operating systems (OSs) as reputors and thus methods to cryptographically bind reputee software agents to the OS, in a non-repudiatory manner, so that trojan horse or malicious software can be identified.

The evidence-based reputations issued on DiDoS compliant packets enables attacks to be mitigated closer to the source. This feature is complementary to other DDoS mitigation constructs such as the IETF DOTS [171] initiative, which facilitates coordination between entities in response to DDoS attacks. In such a construct, a victim autonomous system would be able to signal to a source autonomous system (from which packets are arriving) for attack mitigation assistance. The source the source autonomous-stems would then be empowered by the DiDoS per-packet reputations to insightfully throttle, divert or drop the packets of attacking agents. For practical adoption value, this research direction would also require the creation of mechanisms to prevent or disincentivise the abuse of this construct by malicious autonomous systems.

Potential areas for future work also span the CED3 framework. The capability of CED3 to evaluate and commensurably compare more sophisticated defences, and perhaps combinations of defences could be explored. Furthermore, in order to reduce the number of required experiments in finding the true effectiveness of defences, future work could involve the creation of algorithms to efficiently discover the lowest cost attack capable of surmounting diverse kinds of defences. Another research avenue is the exploration of the viability and utility of *adoptability by simulation* (introduced in section 5.4.4.4) as a method of comparatively assessing the adoptability of candidate defences. Furthermore, future work

may involve applying CED3 in the context of a real (non-arbitrary) network, where the evaluation context would reflect the environment of chosen network.

Chapter 8: References

[1] D. Farinacci, S. Hanks, D. Meyer, and P. Traina, "RFC: 2784 Generic Routing Encapsulation (GRE)," 2000. Accessed: Sep. 14, 2023. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2784.html

[2] R. Fielding and J. Reschke, "RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content," Request for Comments (RFC). Accessed: Dec. 31, 2018. [Online]. Available: https://tools.ietf.org/html/rfc7231

[3] J. Postel, "RFC 792: Internet Control Message Protocol," Request for Comments (RFC). Accessed: Dec. 31, 2018. [Online]. Available: https://tools.ietf.org/html/rfc792

[4] "IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks," *IEEE Std 802.1Q-2014 (Revision of IEEE Std 802.1Q-2011)*, pp. 1–1832, 2014, doi: 10.1109/IEEESTD.2014.6991462.

[5] Dave Dittrich, "DDoS attack tool timeline." Accessed: May 29, 2017. [Online]. Available: https://staff.washington.edu/dittrich/misc/ddos/timeline.html

[6] P. C. I. Mark Schuchter, "Distributed Denial of Service (DDoS) Timeline." Accessed: May 29, 2017. [Online]. Available: https://www.parabon.com/faqs/ddos-timeline.html

[7] F. Stefanie Hoffman, "DDoS: A Brief History." Accessed: May 29, 2017. [Online]. Available: https://blog.fortinet.com/2013/03/25/ddos-a-brief-history

[8] Michael Aaron Dennis, "Denial of Service Attack (DoS attack)." Accessed: Apr. 18, 2017. [Online]. Available: https://www.britannica.com/topic/denial-of-service-attack

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

[10] Aliya Sternstein, "The Pentagon Wants To Wage War on Denial-of-Service Cyber Attacks," Defense One. Accessed: Feb. 14, 2018. [Online]. Available:

http://www.defenseone.com/ideas/2015/08/pentagon-wants-wage-war-denial-service-cyber-attacks/119196/

[11] M. Cooney, “DARPA extreme DDOS project transforming network attack mitigation | Network World.” Accessed: Feb. 14, 2018. [Online]. Available: https://www.networkworld.com/article/3073481/security/darpa-extreme-ddos-project-transforming-network-attack-mitigation.html

[12] S. Behal, K. Kumar, and M. Sachdeva, “D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events,” *Journal of Network and Computer Applications*, 2018, doi: 10.1016/j.jnca.2018.03.024.

[13] A. Agarwal, M. Khari, and R. Singh, “Detection of DDOS Attack using Deep Learning Model in Cloud Storage Application,” *Wirel Pers Commun*, vol. 127, no. 1, pp. 419–439, 2022, doi: 10.1007/s11277-021-08271-z.

[14] Akshay Joshi, “Biggest-ever DDoS attack threatens companies worldwide, and other cybersecurity news to know this month,” World Economic Forum. Accessed: Dec. 19, 2023. [Online]. Available: https://www.weforum.org/agenda/2023/11/biggest-ddos-attack-cybersecurity-news-to-know-november-2023/

[15] P. Cisco, “Cisco Visual Networking Index: Forecast and Trends 2017-2022,” 2018. Accessed: Dec. 27, 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf

[16] Arbor Networks, “Worldwide Infrastructure Security Report,” 2015. [Online]. Available: https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf

[17] Steven Vaughan-Nichols, “Google Cloud, AWS, and Cloudflare report largest DDoS attacks ever,” ZDNET. Accessed: Dec. 19, 2023. [Online]. Available: https://www.zdnet.com/article/google-cloud-aws-and-cloudflare-report-largest-ddos-attacks-ever/

[18] T. G. Nicky Woolf, “DDoS attack that disrupted internet was largest of its kind in history, experts say.” Accessed: May 29, 2017. [Online]. Available: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

[19] F. N. I. David Homes, “2016 DDoS Attack Trends,” 2016, Accessed: May 29, 2017. [Online]. Available: https://f5.com/Portals/1/PDF/security/2016_DDoS_Attack-Trends.pdf

[20] Scott Hilton, “Dyn Analysis Summary Of Friday October 21 Attack,” Dyn Blog. Accessed: May 29, 2017. [Online]. Available: https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

[21] F. I. B. Blog Author, “2016: The Year of DDoS Attacks.” Accessed: May 16, 2017. [Online]. Available: https://blog.fraudwatchinternational.com/expert-explanations/2016-year-ddos-attacks

[22] Statista, “Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2023, with forecasts from 2022 to 2030.” Accessed: Dec. 27, 2023. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/#:~:text=The%20number%20of%20Internet%20of,around%208%20billion%20consumer%20devices.

[23] R. van der Meulen, “Gartner Says 8.4 Billion Connected ‘Things’ Will Be in Use in 2017, Up 31 Percent From 2016,” Gartner Press Release. [Online]. Available: http://www.gartner.com/newsroom/id/3598917

[24] T. Matthews, “Incapsula Survey: What DDoS Attacks Really Cost Businesses,” 2014, [Online]. Available: http://lp.incapsula.com/rs/804-TEY-921/images/DDoS Report Q2 2015.pdf

[25] B. Krebs, “The Democratization of Censorship — Krebs on Security,” Krebsonsecurity.com. Accessed: Feb. 28, 2018. [Online]. Available: https://krebsonsecurity.com/2016/09/the-democratization-of-censorship/

[26] S. Bhatia, “Detecting Distributed Denial-of-Service Attacks and Flash Events,” The LNM Institute of Information Technology, 2013. [Online]. Available: https://eprints.qut.edu.au/62031/1/Sajal_Bhatia_Thesis.pdf

[27] W. S. PENDERGRASS, “What is Anonymous?: A case study of an information systems hacker activist collective movement,” *ProQuest Dissertations and Theses*, 2013.

[28] S. Khattak, N. R. Ramay, K. R. Khan, A. A. Syed, and S. A. Khayam, “A Taxonomy of botnet behavior, detection, and defense,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 898–924, 2014, doi: 10.1109/SURV.2013.091213.00134.

[29] D. Kobialka, “Kaspersky Lab Study: Average Cost of Enterprise DDoS Attack Totals $2M - MSSP Alert,” MSSP Alert. Accessed: Sep. 02, 2018. [Online]. Available: https://www.msspalert.com/cybersecurity-research/kaspersky-lab-study-average-cost-of-enterprise-ddos-attack-totals-2m/

[30] G. Pandya, “Preparing to withstand a DDoS Attack,” 2021. Accessed: Jul. 25, 2022. [Online]. Available: https://sansorg.egnyte.com/dl/OzcvT28KCw

[31] E. Osterweil, A. Stavrou, and L. Zhang, “21 Years of Distributed Denial-of-Service: A Call to Action,” *Computer (Long Beach Calif)*, vol. 53, no. 8, pp. 94–99, Aug. 2020, doi: 10.1109/MC.2020.2993330.

[32] E. Osterweil, A. Stavrou, and L. Zhang, “20 Years of DDoS : a Call to Action,” 2019. [Online]. Available: https://arxiv.org/pdf/1904.02739.pdf

[33] S. Zeijlemaker and E. Rouwette, “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*, 2017. Accessed: Sep. 02, 2018. [Online]. Available: https://www.systemdynamics.org/assets/conferences/2017/proceed/papers/P1171.pdf

[34] M. Natu and J. Mirkovic, “Fine-grained capabilities for flooding DDoS defense using client reputations,” in *Proceedings of the 2007 workshop on Large scale attack defense - LSAD ’07*, New York, New York, USA: ACM Press, 2007, p. 105. doi: 10.1145/1352664.1352668.

[35] National Crime Agency, “International operation shuts down notorious cyber crime website.” Accessed: Sep. 06, 2018. [Online]. Available: http://www.nationalcrimeagency.gov.uk/1335-international-action-against-ddos-tool

[36] D. Casciani, “Cyber-attack website Webstresser taken down - BBC News,” BBC News. Accessed: Apr. 25, 2018. [Online]. Available: http://www.bbc.co.uk/news/uk-43893420

[37] T. Pohle, “Latest Link11 DDoS Report shows that DDoS attack volumes increased by 50% in Q2 2018 - Link11,” Link11. Accessed: Sep. 06, 2018. [Online]. Available: https://www.link11.com/en/blog/latest-link11-ddos-report-shows-that-ddos-attack-volumes-increased-by-50-in-q2-2018/

[38] F. Truta, “DDoS Attack Volume Rose 50% in Q2 2018,” Security Boulevard. Accessed: Sep. 06, 2018. [Online]. Available: https://securityboulevard.com/2018/08/ddos-attack-volume-rose-50-in-q2-2018/

[39] Departement for Digital Culture Media & Sport (DCMS), “Secure by Design: Improving the cyber security of consumer Internet of Things Report,” London, 2018. Accessed: Sep. 06, 2018. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/686089/Secure_by_Design_Report_.pdf

[40] G. Oikonomou, J. Mirkovic, P. Reiher, and M. Robinson, “A framework for a collaborative DDoS defense,” *Proceedings - Annual Computer Security Applications Conference, ACSAC*, pp. 33–42, 2006, doi: 10.1109/ACSAC.2006.5.

[41] K. Hoffman, D. Zage, and C. Nita-Rotaru, “A Survey of Attack and Defense Techniques for Reputation Systems,” 2007.

[42] S. Hin Khor, N. Christin, T. Wong, and A. Nakao, “Power to the People: Securing the Internet One Edge at a Time *”, Accessed: Feb. 06, 2018. [Online]. Available: https://www.andrew.cmu.edu/user/nicolasc/publications/KCWN-LSAD07.pdf

[43] J. Steinberger, B. Kuhnert, A. Sperotto, H. Baier, and A. Pras, “Collaborative DDoS defense using flow-based security event information,” *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, no. Noms, pp. 516–522, 2016, doi: 10.1109/NOMS.2016.7502852.

[44] B. Rodrigues, T. Bocek, A. Lareida, D. Hausheer, S. Rafati, and B. Stiller, “A blockchain-based architecture for collaborative DDoS mitigation with smart contracts,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017. doi: 10.1007/978-3-319-60774-0_2.

[45] IETF, “IETF | Internet Engineering Task Force.” Accessed: Jan. 06, 2019. [Online]. Available: https://www.ietf.org/

[46] NCSC, “National Cyber Security Centre (NCSC).” Accessed: Jan. 06, 2019. [Online]. Available: https://www.ncsc.gov.uk/

[47] H. Liu, Y. Sun, V. C. Valgenti, and M. S. Kim, “TrustGuard: A flow-level reputation-based DDoS defense system,” *2011 IEEE Consumer Communications and Networking Conference, CCNC’2011*, no. PerNets, pp. 287–291, 2011, doi: 10.1109/CCNC.2011.5766474.

[48] A. Otung and A. Martin, “Distributed Defence of Service (DiDoS): A Network-layer Reputation-based DDoS Mitigation Architecture,” 2020, pp. 619–630. doi: 10.5220/0009091206190630.

[49] A. Otung and A. Martin, “Towards Comparative Evaluation of DDoS Defences,” in *2022 15th International Conference on Security of Information and Networks (SIN)*, 2022, pp. 1–10. doi: 10.1109/SIN56466.2022.9970527.

[50] A. Otung, “CED3: Comparative Evaluation of DDoS Defences,” Sep. 2022, doi: 10.17862/cranfield.rd.18279839.v1.

[51] Q. Y. B and W. Huang, “A DDoS Detection and Mitigation System Framework Based on Spark and SDN,” vol. 10135, pp. 350–358, 2017, doi: 10.1007/978-3-319-52015-5.

[52] T. Peng, L. C., and K. Ramamohanarao, “Detecting Distributed Denial of Service Attacks by Sharing Distributed Beliefs,” in *ACISP: Information Security and Privacy*, R. Safavi-Naini and J. Seberry, Eds., Springer, Berlin, Heidelberg, 2003, pp. 214–225. doi: https://doi.org/10.1007/3-540-45067-X_19.

[53] Vangie Beal, “What is DDoS - Distributed Denial of Service? Webopedia.” Accessed: May 31, 2017. [Online]. Available: http://www.webopedia.com/TERM/D/DDoS_attack.html

[54] Incapsula, “DDoS Attacks: Specific DDoS Attacks Types.” Accessed: May 01, 2017. [Online]. Available: https://www.incapsula.com/ddos/ddos-attacks/

[55] M. Naresh Kumar, P. Sujatha, V. Kalva, R. Nagori, A. K. Katukojwala, and M. Kumar, “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*, IEEE, Nov. 2012, pp. 535–539. doi: 10.1109/CICN.2012.149.

[56] M. Guirguis, A. Bestavros, and I. Matta, “Exploiting the transients of adaptation for RoQ attacks on internet resources,” *Proceedings - International Conference on Network Protocols, ICNP*, pp. 184–195, 2004, doi: 10.1109/ICNP.2004.1348109.

[57] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, p. 39, 2004, doi: 10.1145/997150.997156.

[58] H. Beitollahi and G. Deconinck, “Analyzing well-known countermeasures against distributed denial of service attacks,” 2012, doi: 10.1016/j.comcom.2012.04.008.

[59] T. Peng, C. Leckie, and K. Ramamohanarao, “Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems,” *ACM Comput Surv*, vol. 39, no. 1, pp. 3-es, 2007, doi: 10.1145/1216370.1216373.

[60] R. v Deshmukh and K. K. Devadkar, “Understanding DDoS Attack & Its Effect In Cloud Environment,” *Procedia - Procedia Computer Science*, vol. 49, pp. 202–210, 2015, doi: 10.1016/j.procs.2015.04.245.

[61] Canton David, “Classification of DoS Attacks | CERTSI,” certsi Blog. Accessed: Aug. 16, 2018. [Online]. Available: https://www.certsi.es/en/blog/classification-dos-attacks

[62] Arbor Networks, “DDoS Attack Types,” 2017. Accessed: Aug. 16, 2018. [Online]. Available: http://resources.arbornetworks.com/wp-content/uploads/INFO_DDoSAttackTypes_EN.pdf

[63] M. Aamir and M. Arif, “Study and Performance Evaluation on Recent DDoS Trends of Attack & Defense,” *Information Technology and Computer Science*, vol. 08, pp. 54–65, 2013, doi: 10.5815/ijitcs.2013.08.06.

[64] M. Backes and P. Ning, “The Coremelt Attack,” *Computer Security – ESORICS 2009*, vol. 6879, no. August 2003, pp. 37–52, 2011, doi: 10.1007/978-3-642-23822-2.

[65] S. Specht and R. Lee, “Taxonomies of Distributed Denial of Service Networks, Attacks, Tools, and Countermeasures,” 2003. Accessed: Aug. 19, 2018. [Online]. Available: www.princeton.edu

[66] Cloudflare, “NTP Amplification DDoS Attack | Cloudflare.” Accessed: Sep. 20, 2018. [Online]. Available: https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/

[67] V. K. Yadav, M. C. Trivedi, and B. M. Mehtre, “DDA: An approach to handle DDoS (Ping flood) attack,” in *Advances in Intelligent Systems and Computing*, 2016, pp. 11–23. doi: 10.1007/978-981-10-0129-1_2.

[68] K. M. Elleithy, D. Blagovic, W. Cheng, and P. Sideleau, “Denial of Service Attack Techniques: Analysis, Implementation and Comparison,” *J Syst Cybern Inf*, vol. 3, no. 1, pp. 66–71, 2005, Accessed: Sep. 19, 2018. [Online]. Available: http://www.iiisci.org/journal/cv$/sci/pdfs/p129065.pdf

[69] K. Heu, “Vulnerability Exploits and Countermeasures: The Ping of Death,” 2009, Accessed: Jun. 25, 2017. [Online]. Available: http://pearl.ics.hawaii.edu/~sugihara/courses/ics426/AssignmentReports/assign4_report2_PingOfDeath.pdf

[70] A. Kuzmanovic and E. W. Knightly, “Low-rate TCP-targeted denial of service attacks,” *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM ’03*, p. 75, 2003, doi: 10.1145/863965.863966.

[71] V. L. Nguyen, P.-C. Lin, and R.-H. Hwang, “MECPASS: Distributed Denial of Service Defense Architecture for Mobile Networks,” *IEEE Netw*, vol. 32, no. 1, pp. 118–124, Jan. 2018, doi: 10.1109/MNET.2018.1700140.

[72] F. Sajjad, “Denial of Service – The Smurf Attack”.

[73] M. Narita, T. Katoh, B. B. Bista, and T. Takata, “A Distributed Detecting Method for SYN Flood Attacks and Its Implementation Using Mobile Agents,” in *Multiagent System Technologies*, Springer, Berlin, Heidelberg, 2009, pp. 91–102. doi: 10.1007/978-3-642-04143-3_9.

[74] R. K. Deka, D. K. Bhattacharyya, and J. Kalita, “DDoS Attacks: Tools, Mitigation Approaches, and Probable Impact on Private Cloud Environment,” 2017. Accessed: Aug. 07, 2018. [Online]. Available: http://www.arbornetworks.com,

[75] A. Keshariya and N. Foukia, “DDoS Defense Mechanisms: A New Taxonomy,” in *Data Privacy Management and Autonomous Spontaneous Security*, J. Garcia-Alfaro, G. Navarro-Arribas, N. Cuppens-Boulahia, and Y. Roudier, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 222–236. Accessed: Feb. 06, 2018. [Online]. Available: https://link.springer.com/content/pdf/10.1007%2F978-3-642-11207-2_17.pdf

[76] G. Hu, W. Chen, Q. Li, Y. Jiang, and K. Xu, “TrueID: A practical solution to enhance Internet accountability by assigning packets with creditable user identity code,” *Future Generation Computer Systems*, vol. 72, pp. 219–226, 2017, doi: 10.1016/j.future.2016.09.005.

[77] Y. Mallios, S. Modi, A. Agarwala, and C. Johns, “Persona: Network Layer Anonymity and Accountability for Next Generation Internet,” Springer, Berlin, Heidelberg, 2009, pp. 410–420. doi: 10.1007/978-3-642-01244-0_36.

[78] X. Liu, A. Li, and X. Yang, “Passport : Secure and Adoptable Source Authentication,” *Nsdi*, 2008.

[79] S. Hares, Y. Rekhter, and T. Li, “RFC 4271: A Border Gateway Protocol 4 (BGP-4),” Request for Comments (IETF). Accessed: Jan. 01, 2019. [Online]. Available: https://tools.ietf.org/html/rfc4271

[80] C. Pappas, R. M. Reischuk, and A. Perrig, “FAIR: Forwarding Accountability for Internet Reputability,” in *2015 IEEE 23rd International Conference on Network Protocols (ICNP)*, IEEE, Nov. 2015, pp. 189–200. doi: 10.1109/ICNP.2015.22.

[81] J. Mirkovic and P. Reiher, “Building accountability into the future internet,” *4th IEEE Workshop on Secure Network Protocols, NPSec’08*, pp. 45–51, 2008, doi: 10.1109/NPSEC.2008.4664880.

[82] J. Yu, C. Fang, L. Lu, and Z. Li, “Mitigating application layer distributed denial of service attacks via effective trust management,” *IET Communications*, vol. 4, no. 16, p. 1952, 2010, doi: 10.1049/iet-com.2009.0809.

[83] N. I. Mowla, Inshil Doh, and Kijoon Chae, “Multi-defense Mechanism against DDoS in SDN Based CDNi,” in *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, IEEE, 2014, pp. 447–451. doi: 10.1109/IMIS.2014.64.

[84] Danielle Walter, “Akamai Named a Leader in Forrester Wave Web Application Firewalls,” Akamai Blog. Accessed: Aug. 02, 2023. [Online]. Available: https://www.akamai.com/blog/security/akamai-named-a-leader-in-the-forrester-wave#:~:text=Forrester%E2%80%99s%202022%20evaluation%20of%20web%20application%20firewalls%20ranks,across%20current%20offering%2C%20strategy%2C%20and%20market%20presence%20categories.

[85] Akamai Technologies, “Akamai DDoS Protection.” Accessed: Aug. 02, 2023. [Online]. Available: https://www.akamai.com/solutions/security/ddos-protection

[86] Akamai Technologies, “Prolexic Solutions.” Accessed: Sep. 05, 2023. [Online]. Available: https://www.akamai.com/products/prolexic-solutions

[87] Akamai, “Edge DNS.” Accessed: Oct. 17, 2023. [Online]. Available: https://www.akamai.com/products/edge-dns

[88] Akamai, “Edge DNS: Product Brief,” 2020. Accessed: Oct. 17, 2023. [Online]. Available: https://www.akamai.com/resources/product-brief/edge-dns

[89] Akamai, “Designing DNS for Availability and Resilience Against DDoS Attacks,” Mar. 2020. Accessed: Oct. 17, 2023. [Online]. Available: https://www.akamai.com/resources/white-paper/designing-dns-for-availability-and-resilience-against-ddos-attacks

[90] Amazon Web Services Inc., “AWS Shield.” Accessed: Oct. 29, 2023. [Online]. Available: https://docs.aws.amazon.com/waf/latest/developerguide/shield-chapter.html

[91] Amazon Web Services Inc., “AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide,” 2023. Accessed: Nov. 01, 2023. [Online]. Available: https://docs.aws.amazon.com/pdfs/waf/latest/developerguide/waf-dg.pdf

[92] Amazon Web Services Inc., “How AWS Shield Mitigates Events.” Accessed: Oct. 29, 2023. [Online]. Available: https://docs.aws.amazon.com/waf/latest/developerguide/ddos-event-mitigation.html

[93] Amazon Web Services Inc., “Detection Logic for Infrastructure Layer Threats.” Accessed: Nov. 01, 2023. [Online]. Available: https://docs.aws.amazon.com/waf/latest/developerguide/ddos-event-detection-infrastructure.html

[94] Amazon Web Services Inc, “Detection Logic for Application Layer Threats.” Accessed: Nov. 03, 2023. [Online]. Available: https://docs.aws.amazon.com/waf/latest/developerguide/ddos-event-detection-application.html

[95] Amazon Web Services Inc, “How Shield Advanced Manages Automatic Mitigation.” Accessed: Nov. 03, 2023. [Online]. Available: https://docs.aws.amazon.com/waf/latest/developerguide/ddos-automatic-app-layer-response-behavior.html

[96] Amazon Web Services Inc, “Global Edge Network.” Accessed: Nov. 06, 2023. [Online]. Available: https://aws.amazon.com/cloudfront/features/?whats-new-cloudfront.sort-by=item.additionalFields.postDateTime&whats-new-cloudfront.sort-order=desc

[97] Cloudfare Inc., “Industry Leading DDoS Protection.” Accessed: Nov. 12, 2023. [Online]. Available: https://www.cloudflare.com/en-gb/ddos/

[98] Cloudflare Inc., “The Cloudflare Global Network.” Accessed: Nov. 22, 2023. [Online]. Available: https://www.cloudflare.com/en-gb/network/

[99] F5 Inc., “DDoS Attack Protection.” Accessed: Nov. 12, 2023. [Online]. Available: https://www.f5.com/solutions/use-cases/ddos-attack-protection

[100] F5 Inc., “Distributed Cloud DDoS Mitigation,” 2023. Accessed: Nov. 13, 2023. [Online]. Available: https://www.f5.com/pdf/solution-overview/f5-distributed-cloud-ddos-mitigation.pdf

[101] Imperva, “DDoS Protection.” Accessed: Nov. 15, 2023. [Online]. Available: https://www.imperva.com/products/ddos-protection-services/

[102] Imperva Inc, “DDoS Protection for Networks.” Accessed: Nov. 15, 2023. [Online]. Available: https://www.imperva.com/products/infrastructure-ddos-protection-services/

[103] S. Sharma, “Imperva DDoS: How cloud based DDoS mitigation works.” Accessed: Nov. 18, 2023. [Online]. Available: https://community.imperva.com/blogs/shikhar-sharma1/2020/08/17/imperva-ddos-how-cloud-based-ddos-mitigation-works#:~:text=Imperva%E2%80%99s%20cloud-based%20DDoS%20mitigation%20service%20relies%20on%2044,purpose-built%20data%20centers%20located%20strategically%20around%20the%20world.

[104] Microsoft, “What is Azure DDoS Protection?” Accessed: Nov. 22, 2023. [Online]. Available: https://learn.microsoft.com/en-us/azure/ddos-protection/ddos-protection-overview

[105] Avery Kim, “Maximizing Effectiveness: Best Practices for Azure DDoS Protection and Application Resilience.” Accessed: Nov. 22, 2023. [Online]. Available: https://techcommunity.microsoft.com/t5/azure-network-security-blog/maximizing-effectiveness-best-practices-for-azure-ddos/ba-p/3914324

[106] A. Duong, H. Al Kazwini, M. B. Harikrishnan, and P. Ovhal, “Azure Front Door POP locations by metro,” May 2023, Accessed: Nov. 22, 2023. [Online]. Available: https://learn.microsoft.com/en-us/azure/frontdoor/edge-locations-by-region

[107] Neustar Inc., “Neustar Security Services Overview,” 2020. Accessed: Nov. 22, 2023. [Online]. Available: https://www.cdn.neustar/resources/product-literature/security/neustar-security-overview.pdf

[108] Netscout, “Arbor DDoS Protection.” Accessed: Nov. 22, 2023. [Online]. Available: https://www.netscout.com/arbor

[109] R. Sharma, “NETSCOUT Expands its Arbor Cloud Attack Mitigation Scrubbing Centers to 15 Worldwide.” Accessed: Nov. 22, 2023. [Online]. Available: https://www.thefastmode.com/technology-solutions/30287-netscout-expands-its-arbor-cloud-attack-mitigation-scrubbing-centers-to-15-worldwide

[110] Radware, “DDoS Protection.” Accessed: Nov. 22, 2023. [Online]. Available: https://www.radware.com/solutions/ddos-protection/

[111] NexusGuard, “360 DDoS Protection.” Accessed: Nov. 26, 2023. [Online]. Available: https://www.nexusguard.com/360-ddos-protection

[112] NexusGuard, “DDoS Mitigation Whitepaper,” 2020. Accessed: Nov. 26, 2023. [Online]. Available: https://www.nexusguard.com/hubfs/Nexusguard_Whitepaper_DDoS_Mitigation_EN_A4.pdf

[113] S. Chen and Q. Song, “Perimeter-based defense against high bandwidth DDoS attacks,” *IEEE transactions on parallel and distributed systems*, vol. 16, no. 6, pp. 526–537, 2005, doi: 10.1109/TPDS.2005.74.

[114] L. A. Quintero González, L. Castanheira, J. A. Marques, A. E. Schaeffer-Filho, and L. Paschoal Gaspary, “Bungee-ML: A Cross-Plane Approach for a Collaborative Defense Against DDoS Attacks,” *Journal of network and systems management*, vol. 31, no. 4, p. 77, 2023, doi: 10.1007/s10922-023-09769-6.

[115] Huifen Feng, Ying Liu, Xincheng Yan, Na Zhou, and Zhihong Jiang, “A Blockchain-enabled Multi-domain DDoS Collaborative Defense Mechanism,” *KSII transactions on Internet and information systems*, vol. 17, no. 3, pp. 916–937, 2023.

[116] H. Huang, P. Ye, M. Hu, and J. Wu, “A multi-point collaborative DDoS defense mechanism for IIoT environment,” *Digital Communications and Networks*, vol. 9, no. 2, pp. 590–601, 2023, doi: 10.1016/j.dcan.2022.04.008.

[117] C. Fung and Y. Pillai, “A Privacy-Aware Collaborative DDoS Defence Network,” in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, IEEE, 2020, pp. 1–5. doi: 10.1109/NOMS47738.2020.9110330.

[118] B. Rashidi, C. Fung, and E. Bertino, “A Collaborative DDoS Defence Framework Using Network Function Virtualization,” *IEEE transactions on information forensics and security*, vol. 12, no. 10, pp. 2483–2497, 2017, doi: 10.1109/TIFS.2017.2708693.

[119] D. Wang, Z. Yufu, and J. Jie, “A multi-core based DDoS detection method,” in *2010 3rd International Conference on Computer Science and Information Technology*, IEEE, 2010, pp. 115–118. doi: 10.1109/ICCSIT.2010.5564969.

[120] M. Mehta, K. Thapar, G. Oikonomou, and J. Mirkovic, “Combining speak-up with DefCOM for improved DDoS defense,” in *IEEE International Conference on Communications*, 2008, pp. 1708–1714. doi: 10.1109/ICC.2008.329.

[121] M. S. Fallah and N. Kahani, “TDPF: a traceback-based distributed packet filter to mitigate spoofed DDoS attacks,” *Security and communication networks*, vol. 7, no. 2, pp. 245–264, 2014, doi: 10.1002/sec.725.

[122] A. Dutta, E. Al-Shaer, S. Chatterjee, and Q. Duan, “Autonomous Cyber Defense Against Dynamic Multi-strategy Infrastructural DDoS Attacks,” in *2023 IEEE Conference on Communications and Network Security (CNS)*, IEEE, 2023, pp. 1–9. doi: 10.1109/CNS59707.2023.10288937.

[123] X. Yong and L. Gao, “Research on DDoS attack and defense strategy for border gateway routing protocol,” in *Proceedings of SPIE - The International Society for Optical Engineering*, SPIE, 2022, pp. 1233214-1233214–6. doi: 10.1117/12.2652500.

[124] A. Dutta, E. Al-Shaer, S. Chatterjee, and Q. Duan, “Autonomous Cyber Defense Against Dynamic Multi-strategy Infrastructural DDoS Attacks,” in *2023 IEEE Conference on Communications and Network Security (CNS)*, IEEE, 2023, pp. 1–9. doi: 10.1109/CNS59707.2023.10288937.

[125] I. A. Valdovinos, J. A. Pérez-Díaz, K. K. R. Choo, and J. F. Botero, “Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions,” *Journal of Network and Computer Applications*, vol. 187. Academic Press, Aug. 01, 2021. doi: 10.1016/j.jnca.2021.103093.

[126] K. Hong, Y. Kim, H. Choi, and J. Park, “SDN-Assisted Slow HTTP DDoS Attack Defense Method,” *IEEE communications letters*, vol. 22, no. 4, pp. 688–691, 2018, doi: 10.1109/LCOMM.2017.2766636.

[127] Jing Zheng, Qi Li, Guofei Gu, Jiahao Cao, D. K. Y. Yau, and Jianping Wu, “Realtime DDoS Defense Using COTS SDN Switches via Adaptive Correlation Analysis,” *IEEE*

transactions on information forensics and security, vol. 13, no. 7, pp. 1838–1853, 2018, doi: 10.1109/TIFS.2018.2805600.

[128] Y. Zhou, G. Cheng, and S. Yu, "An SDN-Enabled Proactive Defense Framework for DDoS Mitigation in IoT Networks," *IEEE transactions on information forensics and security*, vol. 16, pp. 5366–5380, 2021, doi: 10.1109/TIFS.2021.3127009.

[129] M. Wang, Y. Lu, and J. Qin, "Source-Based Defense Against DDoS Attacks in SDN Based on sFlow and SOM," *IEEE access*, vol. 10, pp. 2097–2116, 2022, doi: 10.1109/ACCESS.2021.3139511.

[130] S. Li, Y. Cui, Y. Ni, and L. Yan, "An Effective SDN Controller Scheduling Method to Defence DDoS Attacks," *CHINESE JOURNAL OF ELECTRONICS*, vol. 28, no. 2, pp. 404–407, 2019, doi: 10.1049/cje.2019.01.017.

[131] K.-Y. Chen *et al.*, "SDNShield: NFV-Based Defense Framework Against DDoS Attacks on SDN Control Plane," *IEEE/ACM transactions on networking*, vol. 30, no. 1, pp. 1–17, 2022, doi: 10.1109/TNET.2021.3105187.

[132] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions," *Comput Sci Rev*, vol. 37, p. 100279, 2020, doi: 10.1016/j.cosrev.2020.100279.

[133] R. Swami, M. Dave, and V. Ranga, "Mitigation of DDoS Attack Using Moving Target Defense in SDN," *Wirel Pers Commun*, vol. 131, no. 4, pp. 2429–2443, 2023, doi: 10.1007/s11277-023-10544-8.

[134] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, and Y. Deng, "A New Framework for DDoS Attack Detection and Defense in SDN Environment," *IEEE access*, vol. 8, pp. 161908–161919, 2020, doi: 10.1109/ACCESS.2020.3021435.

[135] J. Cui, M. Wang, Y. Luo, and H. Zhong, "DDoS detection and defense mechanism based on cognitive-inspired computing in SDN," *Future generation computer systems*, vol. 97, pp. 275–283, 2019, doi: 10.1016/j.future.2019.02.037.

[136] N. M. Yungaicela-Naula, C. Vargas-Rosales, J. A. Perez-Diaz, and D. F. Carrera, "A flexible SDN-based framework for slow-rate DDoS attack mitigation by using deep reinforcement learning," *Journal of network and computer applications*, vol. 205, p. 103444, 2022, doi: 10.1016/j.jnca.2022.103444.

[137] A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and I. H. Hasbullah, "A Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-Defined Networking," *Sensors (Basel)*, vol. 23, no. 9, p. 4441, 2023, doi: 10.3390/s23094441.

[138] X. Yuan, C. Li, and X. Li, “DeepDefense: Identifying DDoS Attack via Deep Learning,” in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, IEEE, May 2017, pp. 1–8. doi: 10.1109/SMARTCOMP.2017.7946998.

[139] B. Naveen Bindra and S. Manu Sood, “Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset,” *Automatic control and computer sciences*, vol. 53, no. 5, pp. 419–428, 2019, doi: 10.3103/S0146411619050043.

[140] K. Malialis and D. Kudenko, “Distributed response to network intrusions using multiagent reinforcement learning,” *Eng Appl Artif Intell*, vol. 41, pp. 270–284, 2015, doi: 10.1016/j.engappai.2015.01.013.

[141] H. Li, Y. Zhao, W. Yao, K. Xu, and Q. Li, “Towards real-time ML-based DDoS detection via cost-efficient window-based feature extraction,” *Science China. Information sciences*, vol. 66, no. 5, 2023, doi: 10.1007/s11432-021-3545-0.

[142] R. K. Deka, D. K. Bhattacharyya, and J. K. Kalita, “Active learning to detect DDoS attack using ranked features,” *Comput Commun*, vol. 145, pp. 203–222, 2019, doi: 10.1016/j.comcom.2019.06.010.

[143] L. Yang and H. Zhao, “DDoS Attack Identification and Defense Using SDN Based on Machine Learning Method,” in *2018 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN)*, IEEE, 2018, pp. 174–178. doi: 10.1109/I-SPAN.2018.00036.

[144] Y. Cao, H. Jiang, Y. Deng, J. Wu, P. Zhou, and W. Luo, “Detecting and Mitigating DDoS Attacks in SDN Using Spatial-Temporal Graph Convolutional Network,” *IEEE Trans Dependable Secure Comput*, vol. 19, no. 6, pp. 1–1, 2022, doi: 10.1109/TDSC.2021.3108782.

[145] UCLA, “Benchmarks for DDoS Defense Evaluation: A University of Delaware Subcontract to UCLA.” Accessed: Feb. 14, 2018. [Online]. Available: https://lasr.cs.ucla.edu/Benchmarks_DDoS_Def_Eval.html

[146] J. Mirkovic, S. Fahmy, P. Reiher, and R. K. Thomas, “How to Test DoS Defenses,” in *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, IEEE, Mar. 2009, pp. 103–117. doi: 10.1109/CATCH.2009.23.

[147] S. Behal and K. Kumar, “Trends in Validation of DDoS Research,” in *Procedia - Procedia Computer Science*, ScienceDirect, 2016, pp. 7–15. doi: 10.1016/j.procs.2016.05.170.

[148] S. Gupta, D. Grover, and A. Bhandari, “Vulnerability Analysis of Existing Distributed Denial of Service (DDoS) Defense Frameworks,” 2018. Accessed: Dec. 13, 2018. [Online]. Available: www.trp.org.in

[149] H. Chan, D. Dash, A. Perrig, and H. Zhang, “Modeling Adoptability of Secure BGP Protocols *,” 2006. Accessed: Nov. 23, 2021. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1159913.1159946

[150] P. Eardley, M. Kanakakis, A. Kostopoulos, T. Levä, K. Richardson, and H. Warma, “Deployment and Adoption of Future Internet Protocols,” in *The Future Internet*, J. Domingue, A. Galis, A. Gavras, T. Zahariadis, D. Lambert, F. Cleary, P. Daras, S. Krco, H. Müller, M.-S. Li, H. Schaffers, V. Lotz, F. Alvarez, B. Stiller, S. Karnouskos, S. Avessta, and M. Nilsson, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 133–144.

[151] C. Cremers, M. Horvat, J. Hoyland, R. Holloway, and S. Scott, “A Comprehensive Symbolic Analysis of TLS 1.3 Thyla van der Merwe,” 2017, doi: 10.1145/3133956.3134063.

[152] D. Basin, C. Cremers, and S. Meier, “Provably Repairing the ISO/IEC 9798 Standard for Entity Authentication *.” Accessed: Sep. 03, 2018. [Online]. Available: https://www.cs.ox.ac.uk/people/cas.cremers/downloads/papers/BCM2013-iso9798-JCS.pdf

[153] D. Basin, C. Cremers, K. Miyazaki, S. Radomirovic, and D. Watanabe, “SECURITY ON TAP,” 2015. Accessed: Sep. 03, 2018. [Online]. Available: www.ieee802.org/16/published.html

[154] M. Acm Reference Format: Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker, “DDoS Defense by Offense,” *ACM Transactions on Computer Systems*, vol. 28, no. 3, 2010, doi: 10.1145/1731060.1731063.

[155] R. Fielding, M. Nottingham, and J. Reschke, “RFC 9110 HTTP Semantics,” Jun. 2022, doi: 10.17487/RFC9110.

[156] J. Postel, “Internet Control Message Protocol,” Sep. 1981, doi: 10.17487/RFC0792.

[157] Sarah Gooding, “Google Launches Invisible reCAPTCHA – WordPress Tavern.” Accessed: Jun. 25, 2017. [Online]. Available: https://wptavern.com/google-launches-invisible-recaptcha

[158] C. G. J. Putman, A. Abhishta, and L. J. M. Nieuwenhuis, “Business Model of a Botnet,” *Proceedings - 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2018*, pp. 441–445, 2018, doi: 10.1109/PDP2018.2018.00077.

[159] A. Wang, W. Chang, S. Chen, and A. Mohaisen, “Delving Into Internet DDoS Attacks by Botnets: Characterization and Analysis,” *IEEE/ACM transactions on networking*, vol. 26, no. 6, pp. 2843–2855, doi: 10.1109/TNET.2018.2874896.

[160] O. Serrat, “The SCAMPER Technique,” in *Knowledge Solutions*, Singapore: Springer Singapore, 2017, pp. 311–314. doi: 10.1007/978-981-10-0983-9_33.

[161] P. Mulder, “SCAMPER Technique by Bob Eberle, a Creativity Tool | ToolsHero,” ToolsHero. Accessed: Aug. 30, 2018. [Online]. Available: https://www.toolshero.com/creativity/scamper-technique-bob-eberle/

[162] Luciano Passuello, “Creative Problem Solving with SCAMPER - Litemind.” Accessed: Aug. 30, 2018. [Online]. Available: https://litemind.com/scamper/

[163] J. Mirkovic and P. Reiher, “D-WARD: A source-end defense against flooding denial-of-service attacks,” *IEEE Trans Dependable Secure Comput*, vol. 2, no. 3, pp. 216–232, 2005, doi: 10.1109/TDSC.2005.35.

[164] M. Jhansi, M. Radha, and B. Simmi, “Mechanism to Mitigate Application Layer DDoS Attack with a Light Weight Trust Approach,” *Special Issue of International Journal of Computer Science & Informatics (IJCSI)*, vol. 2, no. 1,2, pp. 173–180, 2012, Accessed: Nov. 15, 2018. [Online]. Available: https://pdfs.semanticscholar.org/2ae7/55a8dc3ace18aa0843e1dc339fdcd170d398.pdf

[165] IANA, “Internet Assigned Numbers Authority.” Accessed: Jan. 24, 2019. [Online]. Available: https://www.iana.org/

[166] A. Martin and Others, “The Ten-Page Introduction to Trusted Computing,” *Computing Laboratory, Oxford University Oxford*, pp. 1–8, 2008.

[167] A. S. Tanenbaum, *Computer Networks*, V. Prentice Hall, 2011.

[168] S. Deering and R. Hinden, “RFC 2460: Internet Protocol, Version 6 (IPv6) Specification,” 1998. [Online]. Available: https://tools.ietf.org/pdf/rfc2460.pdf

[169] “RFC 791 - Internet Protocol.” Accessed: Sep. 25, 2022. [Online]. Available: https://datatracker.ietf.org/doc/rfc791/

[170] “RFC 9293: Transmission Control Protocol (TCP).” Accessed: Sep. 28, 2022. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9293

[171] N. T. A. Mortensen, F. Andreasen, T. Reddy, C. Gray, R. Compton, “Distributed-Denial-of-Service Open Threat Signaling (DOTS) Architecture - draft-ietf-dots-architecture-03,” 2017, Accessed: Jun. 22, 2017. [Online]. Available: https://tools.ietf.org/pdf/draft-ietf-dots-architecture-03.pdf

[172] D. Bisson, “The Risks of Self-Signed Certificates Are Often Mistaken for Benefits | Venafi.” Accessed: Aug. 17, 2022. [Online]. Available: https://www.venafi.com/blog/self-signed-certificates-cyber-criminals-can-quickly-turn-strength-vulnerability

[173] D. Cooper *et al.*, “RFC5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.” Accessed: Aug. 17, 2022. [Online]. Available: https://www.ietf.org/rfc/rfc5280.txt

[174] J. Barr, “Cloud Computing, Server Utilization, & the Environment,” AWS News Blog. Accessed: Mar. 22, 2022. [Online]. Available: https://aws.amazon.com/blogs/aws/cloud-computing-server-utilization-the-environment/#:~:text=Servers and related IT resources,levels are often under 20%25

[175] The NS-3 Research Community, “NS-3 Introduction — Tutorial.” Accessed: Jul. 25, 2018. [Online]. Available: https://www.nsnam.org/docs/tutorial/html/introduction.html#about-ns3

[176] Free Software Foundation, “Bash - GNU Project - Free Software Foundation.” Accessed: Aug. 08, 2022. [Online]. Available: https://www.gnu.org/software/bash/

[177] SLURM Team, “Slurm Workload Manager - Overview.” Accessed: Aug. 09, 2022. [Online]. Available: https://slurm.schedmd.com/overview.html

[178] A. Richards, “University of Oxford Advanced Research Computing,” Aug. 2015, doi: 10.5281/ZENODO.22558.

[179] J. Walrand and P. Varaiya, “The Internet and TCP/IP Networks,” *High-Performance Communication Networks*, pp. 155–203, Jan. 2000, doi: 10.1016/B978-0-08-050803-0.50009-5.

[180] J. Xie, Q. M. He, and X. Zhao, “On the stationary distribution of queue lengths in a multi-class priority queueing system with customer transfers,” *Queueing Syst*, vol. 62, no. 3, pp. 255–277, 2009, doi: 10.1007/S11134-009-9130-0.

[181] G. Appenzeller, I. Keslassy, and N. McKeown, “Sizing Router Buffers,” *ACM SIGCOMM Computer Communication Review* , vol. 34, no. 4, pp. 281–292, Aug. 2004.

[182] P. Velan, J. Medková, T. Jirsík, and P. Čeleda, “Network traffic characterisation using flow-based statistics,” in *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, Institute of Electrical and Electronics Engineers Inc., Jun. 2016, pp. 907–912. doi: 10.1109/NOMS.2016.7502924.

[183] P. Jurkiewicz, G. Rzym, and P. Boryło, “Flow length and size distributions in campus Internet traffic,” *Comput Commun*, vol. 167, pp. 15–30, Sep. 2018, doi: 10.1016/j.comcom.2020.12.016.

[184] R. Joven and E. Ananin, “DDoS-for-Hire Service Powered by Bushido Botnet.” Accessed: Oct. 14, 2019. [Online]. Available: https://www.fortinet.com/blog/threat-research/ddos-for-hire-service-powered-by-bushido-botnet-.html

[185] F. C. Freiling, T. Holz, and G. Wicherski, “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)*, 2005. doi: 10.1007/11555827_19.

[186] T. Hyun-Jin Kim CyLab, C. Basescu ETH Zürich, S. Bum Lee Qualcomm, Y.-C. Hu, and A. Perrig ETH Zürich, “Lightweight Source Authentication and Path Validation,” 2014, doi: 10.1145/2619239.2626323.

[187] H. Li, V. Kumar, J.-M. Park, and Y. Yang, “Cumulative Message Authentication Codes for Resource-Constrained IoT Networks,” *IEEE Internet Things J*, vol. 8, no. 15, pp. 11847–11859, doi: 10.1109/JIOT.2021.3074054.

[188] S. Nair M. P., M. Cherian, and M. K. Preetha, “Application Layer DDoS Attack Defense Methods with a New Method against Flooding,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, 2020, doi: 10.35940/ijrte.F6986.038620.

[189] “ICANN: Internet Corporation for Assigned Names and Numbers,” 2019. Accessed: Jan. 06, 2019. [Online]. Available: https://www.icann.org/

[190] ITU, “International Telecommunications Union (ITU).” Accessed: Jan. 06, 2019. [Online]. Available: https://www.itu.int/en/Pages/default.aspx

[191] DARPA, “Defense Advanced Research Projects Agency (DARPA).” Accessed: Jan. 06, 2019. [Online]. Available: https://www.darpa.mil/

[192] Cisco, “Cisco – UK & Ireland.” Accessed: Jan. 06, 2019. [Online]. Available: https://www.cisco.com/c/en_uk/index.html

[193] NIST, “National Institute of Standards and Technology | NIST.” Accessed: Jan. 06, 2019. [Online]. Available: https://www.nist.gov/

[194] E. Stinson and J. C. Mitchell, “Towards systematic evaluation of the evadability of bot/botnet detection methods,” *WOOT’08 Proceedings of the 2nd conference on USENIX Workshop on offensive technologies*, 2008.

[195] K. Singh, S. Dhindsa, B. Bhushan, K. Singh Dhindsa, and G. N. Khalsa, “PERFORMANCE ANALYSIS OF AGENT BASED DISTRIBUTED DEFENSE MECHANISMS AGAINST DDOS ATTACKS,” 2018. Accessed: Aug. 31, 2018. [Online]. Available: www.computingonline.net

[196] P. Srisuresh and M. Holdrege, “RFC 2663 - IP Network Address Translator (NAT) Terminology and Considerations.” Accessed: Feb. 08, 2022. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2663

[197] K. Yoon and C.-L. Hwang, *Multiple Attribute Decision Making*. SAGE Publications, Inc., 2011. doi: 10.4135/9781412985161.

[198] J. Karnon, J. Stahl, A. Brennan, J. J. Caro, J. Mar, and J. Möller, “Modeling Using Discrete Event Simulation,” *Medical decision making*, vol. 32, no. 5, pp. 701–711, doi: 10.1177/0272989X12455462.

[199] S. Brailsford, L. Churilov, B. T. Dangerfield, and S. Bayer, *Discrete-event simulation and system dynamics for management decision making*. in Wiley series in operations research and management science. Chichester, England, 2014.

[200] L. Wang, Q. Li, Y. Jiang, X. Jia, and J. Wu, “Woodpecker: Detecting and mitigating link-flooding attacks via SDN,” *Computer Networks*, 2018, doi: 10.1016/j.comnet.2018.09.021.

[201] M. Rezazad, M. R. Brust, M. Akbari, P. Bouvry, and N. M. Cheung, “Detecting target-area link-flooding DDoS attacks using traffic analysis and supervised learning,” *Advances in Intelligent Systems and Computing*, vol. 887, pp. 180–202, 2019, doi: 10.1007/978-3-030-03405-4_12.

[202] A. Richards, “University of Oxford Advanced Research Computing,” Aug. 2015, doi: 10.5281/ZENODO.22558.

[203] F. Rambach *et al.*, “A multilayer cost model for metro/core networks,” *Journal of Optical Communications and Networking*, vol. 5, no. 3, pp. 210–225, 2013, doi: 10.1364/JOCN.5.000210.

[204] International Organisation for Standardization - ISO, “Benefits of International Standards.” Accessed: Feb. 09, 2017. [Online]. Available: http://www.iso.org/iso/home/standards/benefitsofstandards.htm

[205] C. R. Group, “NS2 versus NS3 Comparison between NS2 and NS3: NS2 NS3.” Accessed: Jul. 27, 2018. [Online]. Available: http://wrc-ejust.org/crn/images/Tutorials/ns2vsns3.pdf

[206] W. M. D. Chia, S. L. Keoh, A. L. Michala, and C. Goh, *Real-time Recursive Risk Assessment Framework for Autonomous Vehicle Operations*. 2021. Accessed: Mar. 29, 2022. [Online]. Available: http://eprints.gla.ac.uk/234649/http://eprints.gla.ac.uk

[207] R. Society and B. Sciences, "The Contribution of Latent Human Failures to the Breakdown of Complex Systems," *Philos Trans R Soc Lond B Biol Sci*, vol. 327, no. 1241, pp. 475–484, 1990, [Online]. Available: https://www.jstor.org/stable/55319

[208] Dr. S. E. Deering and J. Mogul, "RFC 1191 - Path MTU discovery," IETF RFC . Accessed: Oct. 02, 2022. [Online]. Available: https://datatracker.ietf.org/doc/rfc1191/

Appendix A. Further Information

A.1 NS-3

Although experiments running on a real hardware testbed can offer the most representative empirical results – should the testbed accurately replicate the anticipated deployment environment, simulation was chosen as the means for validating the CED3 method. As such, the empirical tools detailed in this section comprise of the simulation software used and the hardware on which the simulations were executed. Simulation was chosen for the following reasons.

Firstly, the cost of using hardware to build a test environment comprising of several hundred nodes, lay far outside the budgetary constraints of this research project, whereas doings so within a virtual environment was feasible. Secondly, simulation affords a great amount of versatility and control over the experiment environment. For example, entire topologies, link capacities and queue sizes can be altered by the execution of a script. This mode and level of control enabled the automation of many diverse experiments at low incremental cost.

The software chosen for the simulations was NS-3 [175]. NS-3 is a discrete-event open-source network simulator that is developed and actively maintained by the research community. One of the main advantages of NS-3 is its extensibility. As a collection of libraries written entirely in C++, it can be combined with external software libraries. One main disadvantage of NS-3 is the lack of a native graphical user interface, however various visualization tools can be used to view networks and simulations.

Many network simulators were considered, but NS-3 was chosen due to its ability to meet all the needs of the project:

- Ready-built internet protocols
- Ability to create and deploy custom protocols
- Ability to create and deploy custom nodes
- Custom traffic pattern generation
- Flexible data collection
- Scalable topology
- Ability to automate the creation and execution of multiple simulations

- Available free documentation and tutorials
- Free use of software
- Active tool support
- Extensibility of the tool itself

Despite NS-2 being able to meet the above requirements and also having more community-contributed code (hence ready-made customizations and protocols) NS-3 was chosen above NS-2 due to the simplicity and efficiency benefits in design and simulation [205]. NS-3 simulations can be created and executed entirely in C++, however NS-2 requires an additional obscure language, oTCL, in order to perform scripting [205].

Other simulation tools considered, such as GNS-3, NetSim and Cisco Packet Tracer, were discounted for various reasons including inabilities to meet the project requirements. For example, despite being free, possessing a graphical user interface and being associated with organisations such as AT&T, Google and VISA, GNS3 was discounted from consideration due to its lack of free training and documentation, which also made assessment of its capabilities and suitability more difficult.

The simulations were run on the Advanced Resource Computing (ARC) cluster [178], which consisted of thousands of computing nodes coordinated by the SLURM management software.

A.2 Alternative Metrics & Combining Formulae

This section details alternative metrics for indicating denial of service as well as briefly presents alternative methods of outputting an indication of denial of service from potential input metric values.

Goodput and packet loss are both indicators of successful data transfer. However, although goodput specifically monitors the useful data rate (by excluding overhead and retransmissions) packet loss was chosen instead, since the delay that the end user sees, caused by retransmissions, would have been redundantly captured by the transaction time metric.

Other possible metrics for gauging the denial-of-service effect or the effectiveness of evaluated defences are listed below:

- Ram usage
- Bandwidth usage
- CPU[1] usage
- Percentage of malicious traffic reaching the server
- packet content integrity

[1] CPU – Computer Processing Unit

- Packet loss (but of attack packets)
- Fraction of completed transfers
- Probability of communication failure
- Healing time
- Time to stop an attack
- Operating cost
- Probability that an attacker is caught after a certain number of attacks

A possible alternative to the determination of denial of service by the value of Equation 13 crossing a threshold is through a mix of combinational logic such that limiting thresholds are found for each UEQ metric and when any measurement exceeds the predetermined thresholds, DoS is declared, regardless of the values of the other measured metrics. The thresholds for the metrics would be service dependent and would thus depend on the evaluation context employed.

To objectively find the ideal formula, could require a study of users making use of different services under different network conditions for jitter, delay, packet loss and transaction time, and each user must rate if they considered the service usable. This would help to determine the thresholds for these metrics, then further experiments combining variations of each metric would enable a deeper understanding of how say, increased jitter and delay combined could make the service more unusable for lower thresholds of each.

Appendix B. Deeper Dive

B.1 DDoS Perspectives

Beginning with the victim, the rest of this discussion analyses different perspectives of DDoS in order to highlight the problem of open connectivity in the presence of malice and conclude the need for targeted interaction and collaboration between parties and devices to mitigate DDoS.

Consider a server hosting an online website-based store for "AB Company", using which, customers of AB Company can browse and purchase goods. Being online, the service provided by AB Company is intrinsically open and accessible by all. However, as not all traffic is welcome, hence the inherent notion of wanted and unwanted traffic.

From the perspective of AB Company, wanted traffic may be broadly defined as any traffic facilitating the company's objectives – such as sales & marketing[1] – with all other traffic being considered as unwanted. Furthermore, the hosts in communication with the AB Company server will either be known to the server or unknown.

The point being, if a communicating host is known to the server, knowledge of the communicating entity can be used to decide whether to service or ignore its requests depending on whether it is deemed to be wanted or unwanted. However, not all wanted traffic originates from known entities, therefore the server **must** provide services to unknown entities to avoid denying service to users of wanted traffic.

Figure 47. illustrates these different kinds of traffic from the Victim's perspective via four quadrants.

[1] For simplicity, other objectives such as investor relations, recruitment and branding have been omitted.

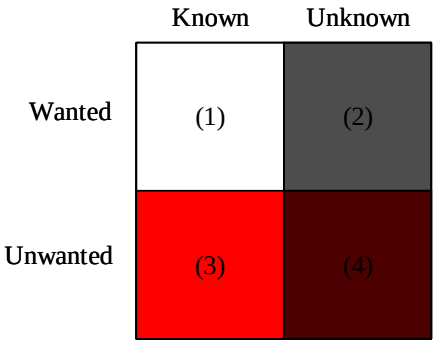


Figure 47: Categories of (Inbound and Outbound) Traffic from the Perspective of a Party

During a DDoS attack, traffic can fall into any of the four categories illustrated.

Returning to the case of AB Company that sells goods via a website, specific examples of wanted traffic could be that generated from new or existing customers searching for products on the website, which are depicted by quadrants (2) and (1) respectively. Examples of unwanted traffic include DDoS attack traffic from known and unknown sources – quadrants (3) and (4).

DDoS attack traffic remains unwanted for AB Company regardless of whether AB Company is the victim[1] of the attack or is an unwitting participant in a reflection-based amplification attack. Due to collateral damage and higher costs of operation, it is clear that DDoS attack traffic is also unwanted from the perspective of other internet users and CSP's.

Therefore, an alignment of interests exists between the three non-malicious parties – the victim, the CSP and other internet users – to minimize the resource consumption of unwanted DDoS traffic.

Traffic received from unknown sources would be more difficult to identify as wanted or unwanted – quadrants (2) and (4). However, even if known sources are determined to have sent unwanted traffic, blocking IP Address of the source outright, could prevent the servicing of legitimate users sending wanted traffic.

Lastly, the process of distinguishing between *wanted* and *unwanted* traffic at the victim-side of an attack before servicing each request is, in itself, resource consuming. Therefore, even if perfect distinctions are made between *wanted* and *unwanted* traffic and the victim's resources are large enough to process requests at the maximum possible rate of receipt, Denial of Service is still possible with increasing volumes of attack traffic, due to the finite resources of routers connected between the victim server and sources of the attack traffic.

From this, it is clear that some form of collaborative approach would be needed for a DDoS-free internet, since victims are better placed to detect attacks against them yet the network is

[1] The victim's perspective encompasses DDoS-attack defence providers acting in defence of said victim

better placed to sinkhole DDoS traffic, near its source, to reduce downstream resource consumption. Such collaborative approaches have previously been considered in [40], [43], [44], [51].

B.2 Structured Theoretical Analysis

The goal of the *structured theoretical analysis* (STA) is to acquire insights into the defences for evaluation, in order to generate an appropriate test environment and attack plan. Firstly, the defences are reviewed in order to understand test-network requirements and to identify metrics of distinction relevant to each defence, which can be leveraged for more effective testing. Metrics of distinction are parameters that a defence uses to distinguish between malicious and benign activity. Next, a *recursive threat analysis* (RTA) is carried out in order to identify defence weaknesses.

STA is an improvement of the pre-evaluation step of [146], which promotes assessment of the kind of defence under consideration before deciding which types of attack scenarios to deploy. STA extends this with RTA, which additionally discovers vulnerabilities of a defence that an attacker can exploit.

RTA, illustrated in Figure 48, is a systematic method we created for discovering the vulnerabilities of DDoS defences. Recursion itself is a widespread algorithmic tool that has been applied to solve many problems, including the task of risk assessment [206]. However, we are not aware of its use in this manner for evaluating DDoS defences.

RTA begins at a high level and successive iterations end up considering the low-level features of a defence. This enables RTA to be flexible enough to accommodate a broad range of defence types, whilst being able to meaningfully consider security aspects in detail.

Of the different kinds of methods used to evaluate system security in the literature (such as semantic-based, refinement-based or simulation-based methods) RTA sits on a higher layer and is more flexible and able to assess defences of varying form and complexity by recursive iteration (as shall be shown). RTA differs from sematic-based methods like symbolic analysis, which require a system and properties to first be formally specified. RTA, on the other hand, starts from top layer goals of the system and recursively assesses the mechanisms used to achieve said goals. As an example of the higher-layer nature of the RTA method, a DDoS defence proposition could rely on a custom protocol that is identified as a mechanism by the RTA method and assessed by existing formal methods, as part of the RTA method as well. That same defence may leverage a custom software, which itself can also be assessed using a different formal method, all part of the same execution of the RTA method. The formal verification of the capabilities of the proposed RTA process itself is a topic for future work.

RTA begins by identifying the goals of a DDoS defence. These goals are then broken down into the fundamental *mechanisms*, processes or functionality required of the architecture to achieve them. The *security requirements* of these components are then identified and the possible ways in which an attacker may compromise said security requirements are analyzed.

These are the *threats*. If a security requirement is identified for which the architecture cannot prevent an attacker from compromising, this is noted as a *vulnerability* of the architecture; and is an output of RTA process. However, if further mechanisms address that would-be vulnerability (threat), these mechanisms are input into the next iteration of recursive analysis. The security requirements of these mechanisms are then identified and possible ways in which an attacker may compromise these requirements are considered, and so on.

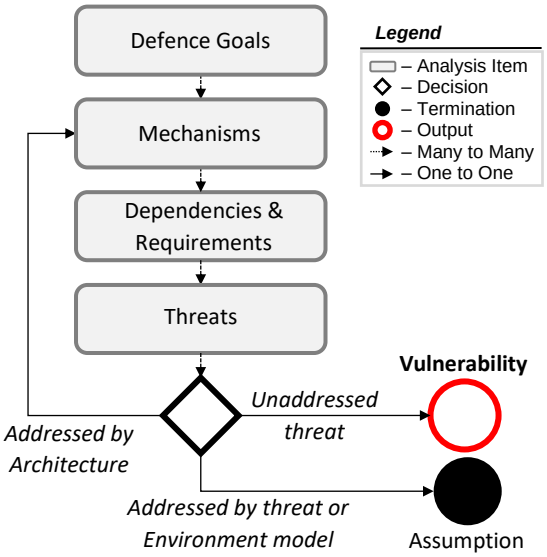


Figure 48: Recursive threat analysis flow chart

In consideration of how an attacker could compromise architectural tenets, a threat model must be applied that defines the attacker's capabilities. It is important, for evaluation consistency, that this same threat model is used for each successive layer of analysis. If the identified threats are addressed by architectural mechanisms of the defence, then these defence mechanisms that address the identified threats are fed back into the RTA process and the cycle of analysis repeats. Further to the threat model determining threats posed by an attacker, the recursive threat analysis process is also dependent on deployment assumptions of the architecture, such as participation rates in various schemes, which may affect the effectiveness of attacks executed by an attacker.

Security related adoptability considerations feed into each level of the RTA process as security requirements despite not being directly related to the main goal of the mechanism under consideration but whose inclusion enables the discovery of pertinent threats that may compromise the adoptability of the defence. For example, privacy requirements like the prohibition of the defence from exposing confidential information of network users or, goals like the defence not enabling the tracking of user activity, would then be considered when evaluating the threats posed to the security requirements of the mechanism under consideration.

This inclusion of adoptability related security concerns as part of the threat analysis process is important since it enables the capture of threats to seemingly innocuous attributes of a defence that could expose clients to increased risk of, for example, activity tracking and hence impact on the adoptability of the defence. One such attribute could be the presence of an identifier in the defence-compliant packets, which, despite not being a vulnerability of the security requirements of a given security mechanism itself, could be harnessed to compromise adoptability-related security properties. Therefore, to enable the RTA process to be able to determine whether the defence in question would weaken the security properties offered by the anticipated deployment environment, we must ensure that enough adoptability-related security properties are specified and included in the RTA process.

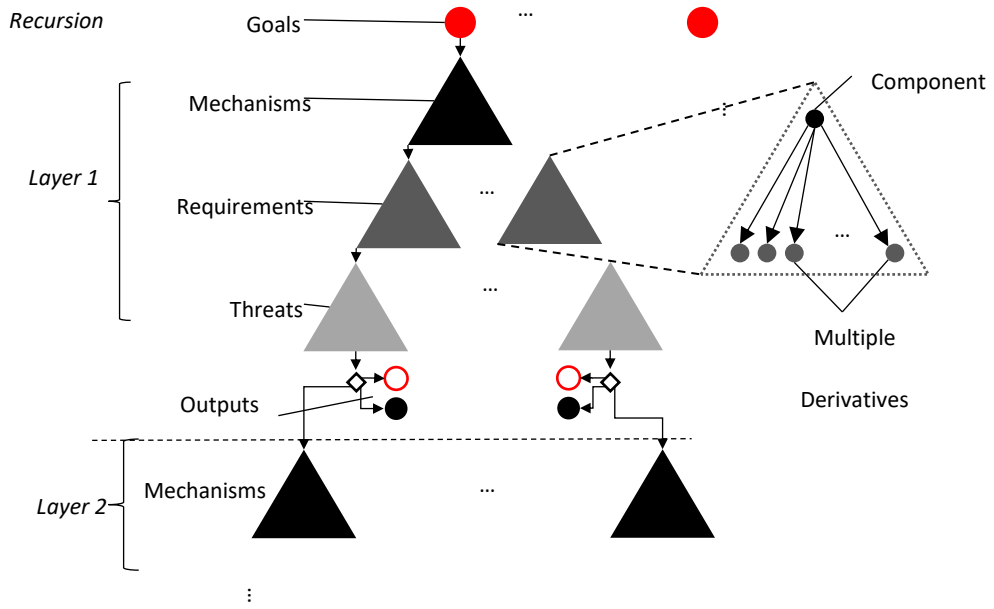


Figure 49: Illustration of the multiplicative progression of the recursive threat analysis process

Figure 49 illustrates the multiplicative nature of the threads of the RTA process. The factor of increase between each layer, depends on the requirements of each mechanism and the resulting number of threats identified that are addressed by the defence architecture (NB: Threats that are not addressed by the defence itself do not produce new RTA threads but terminate as vulnerabilities or assumptions).

These recursive threads of RTA end when no further architectural mechanisms for mitigating discovered vulnerabilities exist and all that remain are a set of vulnerabilities and or assumptions. It is important to note at this stage that, without assumptions, we consider that no DDoS defence is left without vulnerabilities, even if the final mechanisms that address identified threats are cryptographic primitives. By default, we consider these cryptographic primitives as vulnerabilities, since they possess some inherent level of risk that they could be

compromised, and their security property broken. For example, there is a possibility, albeit a very low one, that an attacker could produce the correct hash value for a message authentication code (MAC) verification scheme, and so compromise the security property upheld by this scheme. Therefore, an appropriate assumption may be to consider the 128-bit shared secret of a MAC scheme as unguessable.

We see then that how representative and valid the conclusions of the recursive threat analysis are, is dependent on the realism of the assumptions, the threat model and the completeness with which security requirements, defence mechanisms and new threats are identified.

Aggregate vulnerabilities are vulnerabilities are formed from multiple minor vulnerabilities such that when combined, they can lead to exploits able to compromise the security objectives of the architecture. Hence, in the analogy of the Swiss Cheese Model [207], aggregate vulnerabilities are the tunnels formed from the individual (vulnerability) holes in the layers of a system that may align to form a path for system subversion. Therefore, as part of recursive threat analysis, it is important to consider the implications presented by minor vulnerabilities when combined with weaknesses in other layers.

Since it is possible for the vulnerabilities discovered from the RTA process to limit the performance of the defence in question, we consider them in the design of low-cost attack scenarios as part of the discovery of the true effectiveness of the defence (see section 5.4.3).

This systematic approach enables a more thorough evaluation that is as replete as the depth of the considerations of each successive level of analysis. Documenting the execution of this approach also aides communication and scrutiny, as third parties may gauge for themselves the robustness of the process by assessing the completeness of each recursive step.

B.3 Choosing the TrustGuard Threshold

In this section we choose a value for the threshold of the macro-level anomaly indicator (MAI) [47] that would trigger the activation of the TrustGuard probabilistic filters. Therefore, we probed the MAI function by passing data representing packets of different sizes through it and chose a threshold based on the criteria indicated in [47]. Streams of normally distributed packet-sizes of increasing means were passed to the function and the results of this probe are illustrated in Figure 50.

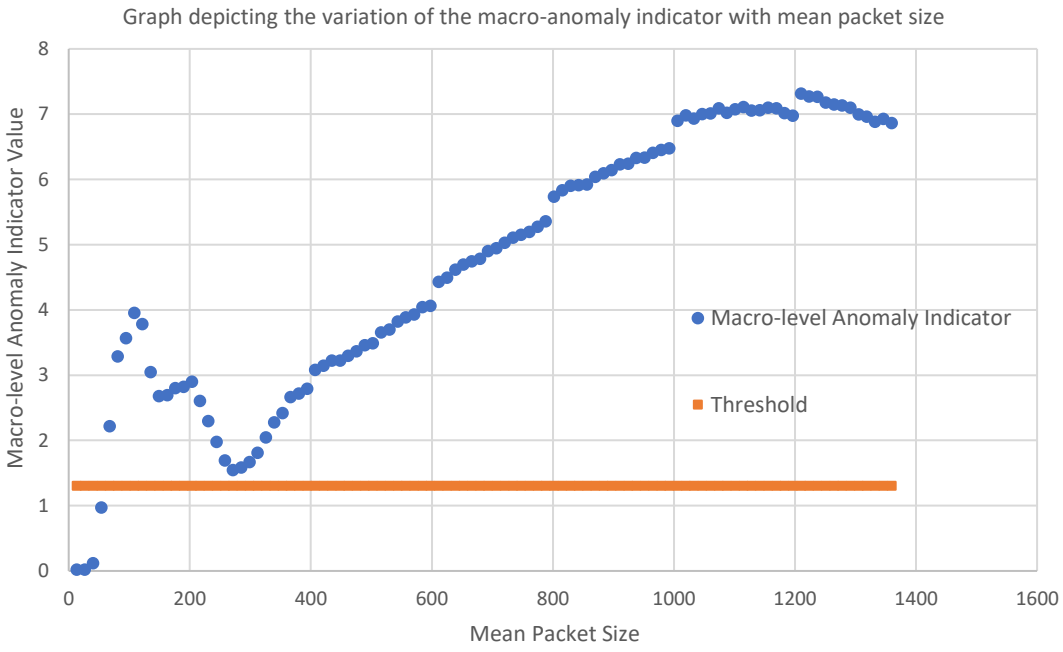


Figure 50: Plot of variation of macro-anomaly indicator with mean packet size

As shown, the MAI function is non-linear. According to the criteria indicated, the TrustGuard queues were only to be triggered by packets of low size, specifically of sizes falling within size-level 1 (i.e., between 40 and 50 bytes). Therefore, we chose a value of 1.3, since that fulfilled the criteria, whilst also avoiding the triggering of the MAI module at higher mean packet sizes.

Appendix C. Result Set Data

	1	2	3	4	5	6	7	8	9	10	15	20	25	30	35	40	45	50
0	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22
2	85.22	85.22	85.19	85.19	85.24	85.19	85.22	85.22	85.19	85.18	85.24	85.23	85.17	85.22	85.26	85.22	85.21	85.21
4	85.22	85.24	85.23	85.28	85.25	85.31	85.26	85.27	85.50	85.28	85.43	85.29	85.28	85.28	85.19	85.21	85.41	85.20
6	85.22	85.21	85.18	85.34	85.33	85.39	85.39	85.30	85.36	85.46	85.47	85.40	85.48	85.33	85.19	85.26	85.22	85.21
8	85.22	85.21	85.25	85.41	85.32	85.47	85.51	85.59	85.50	85.43	85.55	85.39	85.41	85.27	85.24	85.32	85.20	85.23
10	85.22	85.24	85.25	85.39	85.56	85.53	85.49	85.69	85.69	85.62	85.52	85.56	85.40	85.32	85.23	85.20	85.21	85.20
12	85.22	85.24	85.24	85.43	85.50	85.58	85.62	85.54	85.56	85.63	85.71	85.52	85.36	85.31	85.22	85.22	85.25	85.22
14	85.22	85.24	85.29	85.49	85.53	85.64	85.71	85.63	85.80	85.76	85.89	85.51	85.34	85.31	85.23	85.31	85.22	85.19
16	85.22	85.22	85.29	85.46	85.44	85.67	85.66	85.83	85.67	85.94	85.86	85.76	85.31	85.42	85.20	85.24	85.22	85.22
18	85.22	85.23	85.28	85.47	85.49	85.72	85.65	85.89	85.87	85.72	85.83	85.86	85.42	85.34	85.25	85.27	85.19	85.24
19	85.05	85.25	85.29	85.47	85.59	85.63	85.72	85.91	85.82	85.85	85.87	85.67	85.30	85.26	85.27	85.21	85.19	85.16
20	85.22	85.57	85.55	85.75	85.88	86.10	85.80	85.80	85.40	85.63	85.44	84.45	83.37	82.80	81.82	81.28	79.85	76.10
22	85.05	85.56	85.50	85.64	85.87	85.98	85.86	85.57	85.50	85.23	85.31	84.64	83.32	82.68	82.24	81.38	79.63	77.20
24	85.22	85.57	85.51	85.60	85.87	85.94	85.74	85.52	85.63	85.49	85.52	84.47	83.31	82.80	82.03	81.46	79.67	77.26
26	85.22	85.52	85.51	85.67	85.78	85.79	85.78	85.58	85.48	85.26	85.15	84.39	83.25	82.68	81.92	81.38	79.84	77.37
28	85.22	85.56	85.60	85.61	85.76	85.87	85.59	85.38	85.45	85.21	85.26	84.48	83.36	82.89	81.93	81.40	79.45	76.33
30	85.22	85.54	85.58	85.62	85.76	85.75	85.69	85.75	85.25	85.31	85.33	84.34	83.31	82.75	81.93	81.42	79.90	76.00
33	85.22	85.59	85.51	85.51	85.56	85.80	85.57	85.70	85.28	85.32	85.43	84.31	83.29	82.73	82.12	81.34	79.66	75.96
36	85.22	85.49	85.44	85.75	85.62	85.70	85.57	85.47	85.30	85.27	85.34	84.61	83.38	82.80	81.97	81.55	80.85	76.11
39	85.22	85.53	85.44	85.65	85.81	85.62	85.52	85.55	85.34	85.21	85.05	84.27	83.34	82.78	82.29	81.44	79.66	77.08
42	85.22	85.48	85.50	85.53	85.76	85.55	85.44	85.39	85.15	85.28	85.29	84.18	83.16	82.76	82.12	81.45	79.77	75.75
45	85.22	85.50	85.41	85.56	85.62	85.62	85.47	85.35	85.24	85.09	85.26	84.19	83.22	82.67	81.96	81.63	80.74	77.26
48	85.22	85.47	85.44	85.60	85.69	85.53	85.60	85.34	85.19	84.97	85.13	84.17	83.14	82.91	82.01	81.40	79.57	75.82
51	85.22	85.50	85.29	85.49	85.51	85.57	85.50	85.63	85.09	85.21	85.10	84.17	83.20	82.85	81.95	81.29	80.65	75.66
54	85.22	85.51	85.37	85.71	85.75	85.79	85.33	85.49	85.23	85.21	85.20	84.24	83.10	82.71	81.98	81.26	79.61	76.07
57	85.22	85.49	85.51	85.55	85.74	85.62	85.64	85.60	85.27	85.22	85.24	84.22	83.13	82.75	81.91	81.34	79.59	75.58
60	85.22	85.52	85.42	85.49	85.71	85.60	85.54	85.34	85.08	85.06	84.89	84.30	83.10	82.77	81.96	81.35	79.56	75.67
63	85.22	85.51	85.42	85.82	85.63	85.67	85.59	85.39	85.12	85.25	85.07	84.13	83.18	82.83	81.87	81.37	79.32	75.77
66	85.22	85.47	85.39	85.50	85.59	85.57	85.56	85.46	85.16	85.18	85.15	84.31	83.34	82.72	81.86	81.23	79.31	75.48
69	85.22	85.54	85.37	85.48	85.58	85.77	85.72	85.48	85.31	85.16	85.16	84.09	83.10	82.73	81.94	81.32	79.33	75.68
72	85.22	85.42	85.33	85.46	85.58	85.74	85.85	85.51	85.07	85.12	85.17	84.14	83.15	82.72	81.96	81.29	79.40	75.56
75	85.22	85.43	85.42	85.45	85.62	85.74	85.53	85.33	85.48	85.17	85.00	84.14	83.20	82.76	81.90	81.31	79.36	75.54

Table 10: Mean percentage packet loss of all benign transactions for various reputation learning times (rows / days) and DiDoS adoption percentages (columns / %) – Part 1 of 2

	55	60	65	70	75	80	85	90	91	92	93	94	95	96	97	98	99	100
0	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22	85.22
2	85.22	85.29	85.18	85.19	85.21	85.27	85.22	85.19	85.16	85.20	85.22	85.25	85.21	85.18	85.30	85.25	85.22	85.25
4	85.28	85.25	85.19	85.25	85.19	85.40	85.17	85.22	85.27	85.20	85.19	85.19	85.23	85.25	85.21	85.17	85.23	85.28
6	85.17	85.31	85.21	85.22	85.29	85.30	85.24	85.17	85.29	85.19	85.21	85.18	85.20	85.22	85.26	85.30	85.24	85.25
8	85.22	85.21	85.18	85.16	85.25	85.21	85.18	85.22	85.29	85.24	85.19	85.27	85.18	85.20	85.23	85.28	85.23	85.24
10	85.23	85.21	85.21	85.18	85.20	85.19	85.34	85.22	85.22	85.21	85.25	85.17	85.22	85.21	85.22	85.30	85.29	85.21
12	85.24	85.27	85.20	85.16	85.21	85.19	85.21	85.26	85.19	85.23	85.21	85.25	85.22	85.20	85.19	85.19	85.24	85.20
14	85.20	85.21	85.18	85.19	85.29	85.23	85.23	85.28	85.19	85.21	85.21	85.20	85.20	85.28	85.21	85.30	85.20	85.28
16	85.27	85.18	85.19	85.25	85.22	85.22	85.21	85.36	85.23	85.25	85.22	85.18	85.20	85.20	85.19	85.17	85.19	85.18
18	85.20	85.20	85.18	85.22	85.21	85.26	85.19	85.18	85.21	85.22	85.27	85.27	85.19	85.25	85.20	85.25	85.21	85.21
19	85.19	85.19	85.25	85.21	85.26	85.21	85.21	85.22	85.22	85.22	85.34	85.24	85.20	85.21	85.29	85.35	85.29	85.25
20	71.84	69.44	67.85	66.49	65.21	63.15	56.22	47.14	54.84	52.33	50.84	49.87	49.82	49.94	48.62	47.70	33.81	33.78
22	71.82	69.38	67.75	66.40	65.18	62.32	56.24	47.12	54.86	52.91	51.96	49.27	49.73	49.89	48.52	47.46	33.15	33.78
24	70.36	70.68	67.88	66.00	65.17	61.93	56.24	47.25	54.48	52.54	52.12	49.90	49.72	49.73	48.61	47.46	33.52	33.84
26	70.45	70.44	67.87	66.00	69.73	67.28	60.30	56.87	64.70	64.45	64.20	62.39	61.78	62.57	61.03	60.06	50.20	50.20
28	71.81	70.85	68.02	66.01	68.69	68.31	61.84	57.20	64.83	64.44	63.16	61.11	61.66	61.21	60.26	59.11	49.17	49.06
30	70.37	70.49	67.92	65.83	69.46	67.86	61.05	53.41	60.95	57.27	52.96	47.92	49.94	54.92	47.24	51.60	33.18	34.27
33	71.66	70.43	67.76	65.63	66.43	65.11	58.48	53.30	53.40	50.07	41.88	37.39	35.74	36.22	31.25	29.79	18.70	18.80
36	71.94	70.60	67.80	65.40	67.83	66.52	60.79	55.91	51.73	49.72	50.18	46.92	47.40	46.88	43.32	42.32	32.37	32.25
39	70.49	68.85	66.41	64.52	67.27	65.60	60.01	49.28	44.93	46.89	45.56	43.78	37.18	42.48	30.69	29.61	16.11	16.01
42	70.40	68.93	66.22	65.93	66.74	62.40	54.58	44.42	40.05	39.51	30.81	29.40	27.81	24.46	19.21	19.95	3.94	3.32
45	69.97	70.45	68.33	64.29	65.00	61.55	52.35	42.94	38.26	37.91	26.24	24.49	23.63	21.44	16.95	18.12	0.00	0.00
48	70.18	70.15	65.46	63.46	62.16	59.63	51.89	42.62	38.12	36.18	26.20	22.10	21.40	21.45	16.99	15.57	0.00	0.00
51	69.85	68.06	65.15	62.70	61.72	59.21	52.00	42.68	38.37	36.04	26.16	22.00	21.59	21.43	17.01	15.85	0.00	0.00
54	69.50	67.72	64.82	62.73	62.11	59.13	52.07	43.18	38.08	35.83	27.25	22.33	21.51	21.33	17.00	15.64	0.00	0.00
57	69.25	67.75	64.87	62.72	61.87	59.46	51.85	42.75	38.32	35.95	26.20	22.26	21.48	21.27	16.91	15.77	0.00	0.00
60	69.33	68.01	64.89	62.83	61.67	59.41	51.72	42.54	38.16	35.98	26.20	21.94	21.32	21.55	16.97	15.41	0.00	0.00
63	69.21	67.71	64.98	62.95	61.88	59.23	52.09	42.68	37.99	35.91	26.16	22.30	21.30	21.65	16.82	15.64	0.00	0.00
66	69.24	67.63	65.00	63.00	61.79	59.27	52.17	42.93	38.14	35.79	26.36	22.23	21.44	21.42	16.95	15.76	0.00	0.00
69	69.40	67.71	64.96	62.87	61.83	59.22	52.05	42.83	38.14	36.00	26.26	22.23	21.32	21.42	16.91	15.72	0.00	0.00
72	69.30	67.70	64.99	62.87	62.90	59.07	51.79	42.94	38.12	35.83	26.27	22.09	21.56	21.30	17.09	15.67	0.00	0.00
75	69.28	67.81	64.95	62.76	61.92	59.34	51.96	42.74	38.18	35.97	26.29	22.06	21.43	21.44	16.95	15.49	0.00	0.00

Table 11: Mean percentage packet loss of all benign transactions for various reputation learning times (rows / days) and DiDoS adoption percentages (columns / %) – Part 2 of 2

	1	2	3	4	5	6	7	8	9	10	15	20	25	30	35	40	45	50
0	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83
2	1.83	1.87	1.85	1.81	1.84	1.83	1.82	1.86	1.87	1.85	1.88	1.79	1.88	1.81	1.82	1.84	1.79	1.81
4	1.83	1.81	1.81	1.84	1.81	1.82	1.86	1.92	2.00	1.82	1.85	1.83	1.95	1.84	1.84	1.84	1.90	1.85
6	1.83	1.86	1.86	1.84	1.86	1.82	1.93	1.77	1.85	1.88	1.80	1.77	1.88	1.90	1.85	1.84	1.85	1.80
8	1.83	1.85	1.85	1.77	1.86	1.83	1.72	1.72	1.78	1.80	1.72	1.75	1.79	1.97	1.82	2.04	1.86	1.87
10	1.83	1.85	1.90	1.80	1.74	1.78	1.87	1.79	1.67	1.88	1.77	1.76	1.69	1.92	1.85	1.85	1.86	1.88
12	1.83	1.87	1.92	2.00	1.79	1.75	1.75	1.77	1.81	1.85	1.66	1.65	1.77	1.89	1.86	1.85	1.90	1.80
14	1.83	1.86	1.85	1.80	1.83	1.73	1.69	1.76	1.76	1.78	1.62	1.73	1.79	1.93	1.84	1.95	1.84	1.83
16	1.83	1.79	1.86	1.83	1.87	1.78	1.79	1.74	1.95	1.66	1.65	1.62	1.79	1.83	1.84	1.87	1.85	1.86
18	1.83	1.83	1.85	1.93	1.88	1.75	1.68	1.89	1.95	1.75	1.65	1.52	1.85	1.80	1.81	1.84	1.86	1.85
19	1.69	1.86	1.92	2.04	1.85	1.85	1.72	1.96	1.86	1.79	1.79	1.77	1.86	1.88	1.84	1.87	1.85	1.82
20	1.83	1.97	1.93	1.77	1.92	1.84	1.87	1.74	2.23	2.07	1.94	2.27	2.29	2.76	3.00	3.35	2.29	4.58
22	1.69	2.03	1.92	1.74	1.85	1.76	1.80	1.69	2.04	1.96	2.02	2.02	2.47	2.74	3.08	3.43	2.09	6.30
24	1.83	2.03	1.91	1.92	2.07	1.85	1.74	1.81	1.93	2.04	2.03	2.22	2.48	2.72	3.07	3.52	2.10	6.30
26	1.83	1.98	1.99	1.83	1.98	1.81	1.85	1.82	1.88	1.92	1.87	2.18	2.45	2.73	3.12	3.43	2.24	6.43
28	1.83	1.96	2.00	1.84	1.89	2.02	1.67	1.93	1.95	2.18	1.91	2.27	2.70	2.83	2.95	3.47	2.05	4.86
30	1.83	1.98	1.97	1.84	1.83	1.73	1.76	1.89	2.17	2.04	1.90	2.45	2.56	2.78	3.01	3.54	2.28	4.49
33	1.83	1.84	1.91	1.90	1.91	1.90	1.85	1.72	2.10	2.00	1.85	2.10	2.44	2.80	3.39	3.30	2.03	4.44
36	1.83	1.95	1.94	1.82	1.81	1.86	1.74	1.85	2.09	2.13	2.14	2.13	2.44	2.77	3.04	3.78	3.76	4.77
39	1.83	2.01	1.98	1.87	1.93	1.88	1.77	1.81	2.06	1.93	1.95	2.31	2.52	2.81	3.31	3.64	2.08	6.16
42	1.83	1.95	1.97	1.88	1.74	1.89	1.82	1.53	2.12	1.99	1.78	2.29	2.55	2.68	3.10	3.66	2.14	4.36
45	1.83	1.91	1.93	1.83	1.74	1.82	1.66	1.77	2.04	1.97	2.00	2.20	2.50	2.62	3.18	3.57	3.32	5.71
48	1.83	1.94	1.89	1.81	1.86	1.88	1.69	1.80	1.94	2.11	1.90	2.29	2.67	2.75	3.08	3.51	2.15	4.19
51	1.83	1.93	1.90	1.83	1.88	1.99	1.63	1.75	2.03	1.90	2.03	2.43	2.46	2.75	3.26	3.42	3.28	4.24
54	1.83	1.97	1.90	1.75	1.87	1.71	1.78	1.85	1.92	1.99	1.96	2.26	2.60	2.61	3.02	3.19	2.01	4.46
57	1.83	1.90	1.93	1.79	1.83	1.82	1.62	1.54	2.00	1.92	1.82	2.48	2.54	2.75	3.11	3.35	2.08	4.14
60	1.83	1.97	1.94	1.85	1.85	1.77	1.86	1.76	2.01	1.86	1.94	2.44	2.54	2.80	3.16	3.40	2.22	4.21
63	1.83	1.90	1.90	1.80	1.90	1.90	1.83	1.83	1.99	2.13	1.82	2.35	2.65	2.65	3.00	3.43	2.04	4.21
66	1.83	1.90	1.76	1.91	1.94	1.80	1.77	1.74	2.01	2.05	1.98	2.41	2.67	2.63	2.99	3.23	2.00	4.09
69	1.83	1.99	1.90	1.98	1.90	1.83	1.65	1.63	2.12	2.00	1.98	2.35	2.60	2.79	2.94	3.34	2.07	4.29
72	1.83	1.86	1.86	1.89	1.83	1.90	1.66	1.74	2.09	2.19	1.88	2.48	2.52	2.71	3.11	3.15	2.05	3.98
75	1.83	1.87	1.84	1.78	1.97	1.80	1.63	1.78	1.97	2.22	2.00	2.22	2.60	2.76	2.98	3.31	2.04	4.15

Table 12**: Standard deviation of mean percentage packet loss of all benign transactions for various reputation learning times (rows / days) and DiDoS adoption percentages (columns / %) – Part 1 of 2**

	55	60	65	70	75	80	85	90	91	92	93	94	95	96	97	98	99	100
0	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83
2	1.84	1.84	1.84	1.82	1.88	1.86	1.83	1.86	1.88	1.88	1.86	1.87	1.84	1.86	1.84	1.85	1.83	1.82
4	1.81	1.89	1.83	1.90	1.84	1.88	1.84	1.89	1.85	1.81	1.89	1.87	1.93	1.88	1.80	1.86	1.80	2.00
6	1.82	1.85	1.85	1.84	1.93	1.79	1.91	1.84	1.84	1.90	1.87	1.84	1.84	1.83	1.79	1.82	1.85	1.81
8	1.85	1.85	1.88	1.82	1.84	1.82	1.82	1.84	1.81	1.86	1.84	1.86	1.85	1.85	1.83	1.82	1.84	1.78
10	1.81	1.86	1.86	1.84	1.81	1.80	1.76	1.83	1.84	1.82	1.91	1.87	1.82	1.84	1.83	1.84	2.06	1.84
12	1.85	1.84	1.83	1.88	1.84	1.88	1.84	1.86	1.87	1.87	1.88	1.81	1.88	1.84	1.83	1.82	1.80	1.87
14	1.89	1.89	1.82	1.81	1.74	1.84	1.84	1.77	1.84	1.85	1.86	1.87	1.80	1.81	1.81	1.77	1.83	1.83
16	1.81	1.87	1.86	2.02	1.82	1.97	1.83	1.91	1.87	1.84	1.83	1.87	1.86	1.81	1.82	1.82	1.84	1.87
18	1.85	1.84	1.91	1.84	1.85	1.85	1.81	1.84	1.85	1.82	2.03	1.83	1.85	1.81	1.86	1.91	1.88	1.84
19	1.86	1.85	1.82	1.85	1.87	1.81	1.84	1.86	1.84	1.86	1.88	1.81	1.83	1.83	1.84	1.80	1.97	1.81
20	9.46	8.46	11.82	12.25	12.92	13.54	21.23	27.01	30.88	32.90	32.18	35.80	35.49	35.78	35.88	35.89	46.32	46.27
22	9.49	8.28	11.49	11.93	12.75	11.29	21.29	26.57	31.07	33.37	34.16	34.79	35.63	35.86	35.87	36.02	45.39	46.27
24	7.04	10.64	11.96	12.23	12.55	10.48	21.32	26.79	30.48	33.17	33.50	35.44	35.78	35.70	35.56	35.94	45.93	46.35
26	7.40	10.21	11.37	11.94	13.00	14.50	22.41	29.28	29.82	29.77	29.64	33.34	32.89	33.28	33.90	34.40	45.85	45.85
28	9.46	10.73	11.74	12.38	12.09	15.23	23.95	29.21	29.76	30.09	28.67	32.19	32.82	32.26	33.00	33.53	44.89	44.79
30	7.30	10.06	11.62	11.96	13.32	14.51	23.12	24.81	27.15	25.19	23.77	25.44	27.05	28.94	26.29	29.60	34.51	34.90
33	9.26	10.03	11.51	11.38	9.93	11.26	20.34	25.38	24.78	25.98	27.50	30.34	31.28	30.46	33.31	32.78	35.76	35.70
36	9.55	10.17	11.48	9.72	11.96	12.74	22.85	27.92	32.48	29.98	33.21	33.92	34.45	34.08	38.61	38.81	44.40	44.24
39	7.18	7.23	9.02	9.34	10.51	11.43	22.28	21.47	25.33	27.01	29.63	30.30	24.26	29.22	25.38	25.30	23.75	23.62
42	7.27	7.52	8.78	11.54	11.16	8.26	16.96	20.01	21.65	22.16	17.85	19.07	18.26	19.48	19.60	16.79	8.82	7.43
45	6.68	10.05	11.26	8.72	11.03	7.39	15.55	19.83	22.03	21.67	18.18	18.90	19.37	19.06	18.64	17.68	0.00	0.00
48	7.02	9.56	7.58	7.07	5.97	6.07	14.80	19.94	21.85	23.22	18.30	18.15	18.94	19.16	18.50	16.00	0.00	0.00
51	6.30	6.54	7.23	6.32	5.46	6.29	14.84	19.89	22.00	23.08	18.37	18.36	18.89	18.83	18.61	16.43	0.00	0.00
54	6.18	6.47	7.19	6.34	6.31	6.37	14.90	20.38	22.30	22.96	20.62	18.68	19.17	19.27	18.63	15.79	0.00	0.00
57	5.83	6.29	7.00	6.42	6.25	6.37	14.93	19.88	21.98	22.96	18.25	18.38	19.07	18.98	18.64	16.53	0.00	0.00
60	5.99	6.19	6.97	6.44	6.32	6.26	14.71	19.96	22.31	23.12	18.21	18.59	18.85	18.95	18.66	15.86	0.00	0.00
63	5.91	6.29	6.94	6.48	6.35	6.38	14.78	20.17	22.03	23.18	18.20	18.63	19.09	18.88	18.48	16.01	0.00	0.00
66	5.81	6.56	6.85	6.41	6.44	6.27	14.85	19.86	22.05	23.16	18.08	18.57	18.91	19.01	18.52	16.04	0.00	0.00
69	6.14	6.53	7.05	6.37	6.21	6.44	14.94	19.82	21.75	23.22	18.42	18.46	19.00	18.92	18.73	16.12	0.00	0.00
72	5.89	6.43	7.10	6.43	7.66	6.24	14.96	19.92	22.10	23.00	18.12	18.69	19.00	18.76	18.84	16.01	0.00	0.00
75	6.06	6.38	6.82	6.54	6.50	6.23	14.83	19.93	22.13	23.17	17.92	18.31	19.15	19.17	18.79	16.04	0.00	0.00

Table 13: Standard deviation of mean percentage packet loss of all benign transactions for various reputation learning times (rows / days) and DiDoS adoption percentages (columns / %) – Part 2 of 2

	1	2	3	4	5	6	7	8	9	10	15	20	25	30	35	40	45	50
0	86.84	84.54	84.84	83.93	83.59	83.56	84.26	84.47	83.65	83.66	83.38	83.42	83.60	83.76	83.36	84.26	85.53	85.95
2	86.69	84.28	84.68	83.87	83.58	83.49	84.06	84.54	83.76	83.64	83.48	83.36	83.58	83.75	83.66	84.47	85.56	85.89
4	86.66	84.07	84.66	84.05	83.78	84.80	84.32	84.56	83.92	83.97	83.94	83.54	83.69	83.67	83.26	84.60	85.69	85.91
6	86.80	84.33	84.91	84.03	83.98	83.86	84.25	84.76	84.10	83.94	83.55	83.70	83.73	83.67	83.42	84.28	85.59	85.94
8	86.85	84.41	84.48	83.57	84.03	83.86	84.60	84.89	83.87	84.10	83.66	84.60	83.76	83.80	83.32	84.38	85.68	85.93
10	86.67	84.31	84.69	84.25	83.97	83.75	84.61	85.05	84.28	84.00	83.83	83.54	83.81	83.72	83.36	84.28	85.54	85.91
12	86.70	84.36	84.73	84.10	84.07	84.03	84.63	84.85	84.27	84.35	83.75	83.48	83.69	83.83	83.36	84.32	85.62	86.07
14	86.79	84.21	84.78	84.29	84.04	83.94	84.64	84.97	84.42	84.51	83.83	83.46	83.79	84.11	83.38	84.26	85.56	85.94
16	86.45	84.57	84.64	84.11	84.20	84.17	84.68	85.05	84.63	84.50	84.01	83.48	83.75	83.68	83.38	84.30	85.66	85.94
18	86.61	84.69	84.85	84.16	84.96	84.00	84.91	85.16	84.42	84.32	84.12	83.66	83.68	83.78	83.29	84.20	85.58	85.92
19	86.55	84.46	85.00	84.24	84.06	84.20	84.99	85.21	84.40	84.51	83.85	83.49	83.79	83.75	83.38	84.38	85.63	85.99
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.91	5.14	0.00	7.55	6.11	11.21	12.42
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.14	9.62	9.26	11.37	7.96	16.77	10.74
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.14	3.99	7.48	9.18	12.01	16.93	16.73
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.96	11.70	0.00	11.07	6.36	12.55	16.66
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.05	10.72	0.00	11.58	12.74	10.43	13.10
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.05	8.65	11.25	4.97	10.21	9.89	11.22
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.58	12.99	5.46	14.25	17.04	11.87	13.38
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.47	10.33	11.92	12.48	9.50	16.50	8.43
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.86	7.76	12.73	12.93	11.62	7.29	10.12
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.97	6.28	10.18	12.62	16.34	16.22	8.80
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.10	11.61	9.65	10.62	9.03	6.82	16.24
48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.91	7.45	12.07	6.02	16.20	5.06	5.56
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.92	3.76	4.70	0.00	8.31	7.54	9.62
54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.60	8.04	8.16	7.96	1.96	5.00
57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.75	9.11	8.91	3.91	4.68	4.28
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.55	7.82	0.00	8.50	2.18	5.87	3.25
63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.78	8.20	3.78	0.00	0.00	0.00	0.00
66	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.87	9.60	0.00	4.17	0.00	3.72	4.15
69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.11	8.54	0.00	0.00	0.00	0.00
72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.24	0.00	4.78	3.82	0.00	0.00
75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.67	0.00	0.00	0.00

Table 14: Mean percentage packet loss of authenticated benign transactions for various reputation learning times (rows / days) and DiDoS adoption percentages (columns / %) – Part 1 of 2

	55	60	65	70	75	80	85	90	91	92	93	94	95	96	97	98	99	100
0	85.83	85.74	85.61	85.50	85.27	85.24	85.05	85.03	84.95	84.97	85.12	85.12	85.11	85.09	85.13	85.12	85.23	85.22
2	85.87	85.66	85.59	85.46	85.29	85.32	85.13	85.00	84.93	85.00	85.13	85.15	85.10	85.12	85.19	85.15	85.24	85.25
4	85.86	85.71	85.59	85.53	85.27	85.21	85.13	85.05	84.98	84.96	85.16	85.11	85.14	85.16	85.13	85.19	85.21	85.28
6	85.86	85.75	85.62	85.58	85.33	85.21	85.11	85.17	85.08	85.02	85.13	85.26	85.14	85.16	85.17	85.11	85.28	85.25
8	85.85	85.68	85.60	85.47	85.26	85.26	85.08	85.07	84.98	84.97	85.10	85.12	85.14	85.11	85.14	85.14	85.21	85.24
10	86.15	85.80	85.64	85.52	85.28	85.28	85.10	85.11	84.97	84.95	85.12	85.08	85.12	85.11	85.11	85.15	85.25	85.21
12	85.82	85.67	85.57	85.48	85.33	85.29	85.06	85.07	85.02	85.01	85.13	85.19	85.10	85.09	85.14	85.11	85.22	85.20
14	85.86	85.68	85.62	85.64	85.33	85.24	85.13	85.04	84.96	85.07	85.16	85.11	85.14	85.20	85.14	85.09	85.17	85.28
16	85.83	85.67	85.62	85.47	85.30	85.22	85.17	85.11	84.99	85.06	85.09	85.16	85.08	85.14	85.13	85.14	85.19	85.18
18	85.89	85.72	85.61	85.45	85.31	85.26	85.09	85.12	84.95	85.07	85.08	85.21	85.13	85.13	85.13	85.07	85.23	85.21
19	85.88	85.72	85.64	85.50	85.36	85.24	85.05	85.07	85.08	85.02	85.12	85.14	85.11	85.12	85.23	85.13	85.28	85.25
20	16.63	13.98	16.85	16.92	16.97	16.71	16.84	16.77	33.20	32.91	32.26	33.72	33.52	33.82	33.89	33.86	33.81	33.78
22	12.53	16.94	16.94	16.87	16.73	14.06	16.98	16.71	32.81	33.28	33.54	33.67	33.68	33.62	33.69	33.85	33.64	33.84
24	13.08	16.43	16.47	16.68	32.18	31.46	30.35	33.12	49.47	49.47	49.52	50.05	49.47	50.04	50.11	50.14	50.21	50.20
26	16.58	17.08	16.80	16.93	30.21	32.93	33.01	33.13	49.50	49.52	48.32	48.75	49.31	48.85	49.14	49.13	49.15	49.06
28	12.11	16.45	16.73	16.58	32.15	32.07	31.55	25.75	44.71	37.50	36.42	34.23	36.46	41.95	35.68	41.65	33.16	34.27
30	16.48	16.35	16.37	15.99	21.96	25.27	24.96	23.02	29.63	23.01	21.09	19.85	18.80	19.42	18.63	18.66	18.68	18.80
33	16.68	16.57	16.53	13.65	28.37	29.50	31.31	31.76	32.23	28.94	32.36	31.35	31.51	31.20	32.08	32.48	32.38	32.25
36	12.53	10.12	12.54	13.20	27.21	27.36	30.05	19.44	20.12	24.08	27.05	27.76	20.61	26.27	18.42	18.65	16.07	16.01
39	12.42	11.36	11.91	16.28	25.84	18.18	14.98	7.57	3.94	8.04	7.51	10.35	8.85	4.74	3.43	6.02	4.10	3.32
42	9.89	16.30	16.12	12.04	18.33	14.57	3.54	0.00	0.00	3.46	0.00	3.51	3.51	0.00	0.00	3.38	0.00	0.00
45	11.25	15.66	6.63	5.59	2.21	1.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
48	7.75	4.21	3.90	0.00	3.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
51	5.16	0.00	0.00	0.00	1.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
54	1.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
57	1.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
66	3.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
69	0.00	0.00	0.00	0.00	8.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 15: Mean percentage packet loss of authenticated benign transactions for various reputation learning times (rows / days) and DiDoS adoption percentages (columns / %) – Part 2 of 2

	1	2	3	4	5	6	7	8	9	10	15	20	25	30	35	40	45	50
0	3.31	3.65	2.37	3.81	4.06	4.12	2.68	2.81	3.02	3.03	2.61	3.34	2.24	1.47	1.70	2.93	2.52	1.87
2	3.73	3.32	2.44	3.40	3.76	4.02	2.57	2.90	3.20	3.05	2.59	3.26	2.23	1.53	2.37	3.01	2.41	1.90
4	3.74	3.79	2.40	3.49	3.79	4.30	2.39	2.88	2.96	2.96	2.60	3.37	2.29	1.57	1.78	2.92	2.45	1.86
6	3.88	3.65	2.47	3.80	3.61	3.64	2.56	2.78	3.01	3.02	2.65	3.18	2.27	1.46	1.68	2.91	2.44	1.88
8	3.64	3.45	2.53	3.86	3.75	3.73	2.43	3.04	2.92	3.10	2.45	1.71	2.34	1.35	1.74	2.97	2.43	1.91
10	3.47	3.56	2.62	3.78	3.88	4.25	2.60	3.00	2.90	3.03	2.35	3.24	2.25	1.42	1.86	2.95	2.51	1.88
12	3.48	3.40	2.39	3.50	3.60	3.76	2.44	3.00	2.92	2.84	2.28	3.19	2.29	1.46	1.72	2.97	2.50	2.12
14	3.50	3.81	2.11	3.88	3.58	4.01	2.66	2.91	3.09	3.07	2.37	3.22	2.32	2.15	1.73	2.98	2.48	1.87
16	3.65	3.53	2.18	3.84	3.21	3.61	2.78	2.59	2.84	2.80	2.22	3.13	2.15	1.43	1.71	2.86	2.46	1.81
18	3.65	3.39	2.46	3.63	4.53	4.02	2.45	3.00	2.89	3.08	2.40	3.28	2.35	1.39	1.79	2.99	2.53	1.93
19	3.57	3.53	2.46	3.67	3.76	3.88	2.36	2.63	2.87	2.99	2.76	3.10	2.28	1.48	1.82	2.93	2.43	1.90
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	13.22	11.49	0.00	16.88	13.65	25.07	27.78
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.48	21.52	20.70	25.42	17.79	37.49	24.01
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.48	8.92	16.72	20.53	26.85	37.85	37.41
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	26.75	26.16	0.00	24.75	14.21	28.07	37.25
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.30	23.98	0.00	25.89	28.49	23.33	29.29
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.82	19.35	25.16	11.12	22.82	22.12	25.10
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	25.89	29.04	12.20	31.86	38.09	26.53	29.93
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.95	23.11	26.65	27.92	21.24	36.89	18.84
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	22.04	17.36	28.47	28.91	25.99	16.29	22.62
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.12	14.03	22.76	28.23	36.55	36.27	19.67
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	20.35	25.97	21.58	23.75	20.19	15.25	36.31
48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.28	16.66	27.00	13.45	36.22	11.32	12.44
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	17.70	8.40	10.51	0.00	18.59	16.86	21.52
54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.53	17.98	18.24	17.79	4.38	11.17
57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.04	20.37	19.92	8.74	10.46	9.58
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	23.58	17.50	0.00	19.01	4.88	13.12	7.27
63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.11	18.34	8.45	0.00	0.00	0.00	0.00
66	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	17.61	21.46	0.00	9.32	0.00	8.31	9.29
69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.13	19.09	0.00	0.00	0.00	0.00
72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	20.67	0.00	10.69	8.54	0.00	0.00
75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.43	0.00	0.00	0.00

Table 16: Standard deviation of mean percentage packet loss of authenticated benign transactions for various reputation learning times (rows / days) and DiDoS adoption percentages (columns / %) – Part 1 of 2

	1	2	3	4	5	6	7	8	9	10	15	20	25	30	35	40	45	50
0	3.31	3.65	2.37	3.81	4.06	4.12	2.68	2.81	3.02	3.03	2.61	3.34	2.24	1.47	1.70	2.93	2.52	1.87
2	3.73	3.32	2.44	3.40	3.76	4.02	2.57	2.90	3.20	3.05	2.59	3.26	2.23	1.53	2.37	3.01	2.41	1.90
4	3.74	3.79	2.40	3.49	3.79	4.30	2.39	2.88	2.96	2.96	2.60	3.37	2.29	1.57	1.78	2.92	2.45	1.86
6	3.88	3.65	2.47	3.80	3.61	3.64	2.56	2.78	3.01	3.02	2.65	3.18	2.27	1.46	1.68	2.91	2.44	1.88
8	3.64	3.45	2.53	3.86	3.75	3.73	2.43	3.04	2.92	3.10	2.45	1.71	2.34	1.35	1.74	2.97	2.43	1.91
10	3.47	3.56	2.62	3.78	3.88	4.25	2.60	3.00	2.90	3.03	2.35	3.24	2.25	1.42	1.86	2.95	2.51	1.88
12	3.48	3.40	2.39	3.50	3.60	3.76	2.44	3.00	2.92	2.84	2.28	3.19	2.29	1.46	1.72	2.97	2.50	2.12
14	3.50	3.81	2.11	3.88	3.58	4.01	2.66	2.91	3.09	3.07	2.37	3.22	2.32	2.15	1.73	2.98	2.48	1.87
16	3.65	3.53	2.18	3.84	3.21	3.61	2.78	2.59	2.84	2.80	2.22	3.13	2.15	1.43	1.71	2.86	2.46	1.81
18	3.65	3.39	2.46	3.63	4.53	4.02	2.45	3.00	2.89	3.08	2.40	3.28	2.35	1.39	1.79	2.99	2.53	1.93
19	3.57	3.53	2.46	3.67	3.76	3.88	2.36	2.63	2.87	2.99	2.76	3.10	2.28	1.48	1.82	2.93	2.43	1.90
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	13.22	11.49	0.00	16.88	13.65	25.07	27.78
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.48	21.52	20.70	25.42	17.79	37.49	24.01
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.48	8.92	16.72	20.53	26.85	37.85	37.41
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	26.75	26.16	0.00	24.75	14.21	28.07	37.25
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.30	23.98	0.00	25.89	28.49	23.33	29.29
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.82	19.35	25.16	11.12	22.82	22.12	25.10
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	25.89	29.04	12.20	31.86	38.09	26.53	29.93
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.95	23.11	26.65	27.92	21.24	36.89	18.84
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	22.04	17.36	28.47	28.91	25.99	16.29	22.62
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.12	14.03	22.76	28.23	36.55	36.27	19.67
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	20.35	25.97	21.58	23.75	20.19	15.25	36.31
48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.28	16.66	27.00	13.45	36.22	11.32	12.44
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	17.70	8.40	10.51	0.00	18.59	16.86	21.52
54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.53	17.98	18.24	17.79	4.38	11.17
57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.04	20.37	19.92	8.74	10.46	9.58
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	23.58	17.50	0.00	19.01	4.88	13.12	7.27
63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	24.11	18.34	8.45	0.00	0.00	0.00	0.00
66	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	17.61	21.46	0.00	9.32	0.00	8.31	9.29
69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.13	19.09	0.00	0.00	0.00	0.00
72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	20.67	0.00	10.69	8.54	0.00	0.00
75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.43	0.00	0.00	0.00

Table 17: Standard deviation of mean percentage packet loss of authenticated benign transactions for various reputation learning times (rows / days) and DiDoS adoption percentages (columns / %) – Part 2 of 2