

# **USING GAME THEORY TO PROTECT THE DOMAIN NAME SYSTEM (DNS) AGAINST DISTRIBUTED DENIAL OF SERVICE ATTACKS (DDoS)**

**By**

**Phillimon Mwape Mumba**

**(609784)**

**September, 2012**

Project Dissertation Submitted to Swansea University in Partial fulfilment of the  
requirements for the Degree of Master of Science in Computing and Software  
Technology

**Department of Computer Science, Swansea University**

Project Supervisor

Prof. Tom Chen

# ABSTRACT

The Domain Name System (DNS) is a critical component in the functioning of the Internet. However, it is vulnerable to distributed denial of service (DDoS) attacks. There are a number of defence mechanisms that have been proposed to reduce the chances of a DDoS attack successfully occurring. However, network administrators and Internet Service Providers (ISPs) find it challenging in selecting the best defence mechanism for DDoS attacks on the DNS. Two reasons have made this task challenging. The first one is that most of the risk management frameworks being used to analyse cyber-attacks are not adequate for intentional and intelligent attacks. The second reason is that some of these risk management frameworks do not consider quantitative data in analysing the risks. In this research an application was written to model DDoS attacks on DNS servers as Bayesian games between an intelligent attacker and a DNS server. The use of this model allows risk managers to understand the impact of DDoS attacks on the DNS server. The model also overcomes the two limitations that most risk management frameworks face. From the results obtained during the experimentation with the application, it was observed that random dropping of packets is an effective defence mechanism against amplification DDoS attacks. Over-provision of bandwidth is an effective defence mechanism for weak DDoS attacks. Therefore in this research, it is recommended that for weak DDoS attacks where the incoming traffic is just slightly higher than the bandwidth, more bandwidth should be provided to ensure that a denial of service does not occur. In a strong DDoS attack where the incoming network traffic is more than twice the available bandwidth, random dropping of packets should be used. A system can be implemented that monitors the incoming traffic to the DNS server and switch between the two strategies as the incoming traffic changes.

# DECLARATIONS AND STATEMENTS

## DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ..... Date.....

## STATEMENT 1

This dissertation is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by giving explicit references. A bibliography is appended.

Signed ..... Date .....

## STATEMENT 2

I hereby give my consent for my dissertation, if relevant and accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... Date .....

# ACKNOWLEDGEMENT

As I reach the end of the journey towards my masters, I would like to express gratitude to all who have supported me.

First of all I would like to thank God for giving me this opportunity to further my education. Secondly I would like to thank the Staff Development office, the School of Mathematics administration and lecturers in Computer Science department at the Copperbelt University for sponsoring me for this course.

I would like to thank the supervisor for the valuable guidance he gave me throughout the project. I would also like to thank the lecturers in Computer Science department at Swansea University for the valuable knowledge they have impacted in me.

I would like to thank my family for being there for me. This one year has not been easy for me but the encouragements you gave me help me throughout my stay here. To my fiancée and best friend, Jane Whitney Nakazwe, thank you very much honey for being there for me. You are an amazing lady. This one year has not been easy for us. It has been lonely but you were there for me. You are the shoulder I can lean on always.

I would like to thank the members of Swansea SDA church for the warm fellowship I enjoyed in this one year I was away from home. I will greatly miss your company. To the Hacuma family, you have been my family here in Swansea. Words are inadequate to express how grateful I am. To my housemates and friends at Ty Beck house, thank for the moments we shared. I enjoyed the time we shared together. To my classmates it was great learning with you guys. I have learnt a lot from you. To my friends back home, thank you for the support you rendered to me.

## Table of Contents

SUMMARY.....	i
DECLARATIONS AND STATEMENTS .....	ii
ACKNOWLEDGEMENT .....	iii
LIST OF TABLES .....	vii
LIST OF FIGURES.....	viii
List of symbols used.....	viii
CHAPTER ONE - INTRODUCTION .....	1
1.0 BACKGROUND .....	2
1.0.1 Reasons why cybercrime has been successful .....	5
1.0.2 Cyber Security Risk management.....	6
1.0.3 Importance of the Domain Name System (DNS) for the functioning of the Internet .....	6
1.1 PROBLEM STATEMENT .....	7
1.2 AIMS AND OBJECTIVES.....	7
1.3 BENEFITS OF THE RESEARCH .....	8
1.4 SCOPE AND LIMITATION OF THE RESEARCH .....	8
1.5 METHODOLOGY .....	8
1.5.1 Literature review and research .....	8
1.5.2 Programming.....	8
1.5.3 Testing.....	9
1.5.4 Limitation of the methodology used .....	9
1.6 Outline of the dissertation .....	9
CHAPTER 2 - LITERATURE REVIEW .....	11
2.0 CHAPTER OVERVIEW .....	12
2.1 GAME THEORY CONCEPTS .....	12
2.1.1 Bayesian games Concepts .....	14
2.2 vulnerabilities of the Domain Name System.....	14
2.3 Distributed Denial of Service Attacks on the DNS Server.....	14
2.3.1 Defence proposals for DDoS attacks.....	16
2.3.2 Attempts to solve Direct DDoS attacks.....	16
2.3.3 Countermeasures against Amplification DDoS attacks.....	17
2.4 Cyber-Security Risk Management.....	19

2.4.1	Importance of risk management.....	19
2.4.1	Criteria for risk management framework .....	20
2.4.2	Existing Risk Management frameworks.....	21
2.4.3	Cyber Security Risk management model .....	22
2.5	Use of game theory in Computer Networks and Risk Management.....	23
2.6	challenges in cyber security risk management .....	24
2.7	evaluation and reflection .....	24
CHAPTER 3 – REQUIREMENTS SPECIFICATION.....		25
3.0	Chapter Overview.....	26
3.1	limitation of current cyber security risk management methods.....	26
3.2	DNS DDoS attack simulation.....	27
3.3	Benefits of the research.....	27
3.3	Requirements of the simulation application .....	27
3.3.1	Functional requirements .....	28
3.3.2	Non- functional requirements .....	28
CHAPTER 4 - DESIGN SPECIFICATION .....		30
4.0	Chapter Overview .....	31
4.1	Modelling DNS DDoS as a Bayesian Game .....	31
4.1.1	Direct DDoS Attack Game .....	32
4.1.2	Reflected Attack (Amplification Attack) Game .....	37
4.1.1.8	Payoff when there is an attack and the firewall drops packet at random.....	41
4.1.3	Normal form representation of the game .....	42
4.1.3	Preferences representation in Bayesian games.....	43
4.1.4	Nash Equilibrium .....	44
4.1.5	Updating the defender's belief about the attacker .....	44
4.2	Architectural design of the program .....	45
4.2.1	Application's Architecture Design flowchart .....	46
4.3	Interface Design.....	47
4.3.1	Initial Prototype .....	47
4.3.2	Final Design.....	48
CHAPTER 5 - IMPLEMENTATION .....		50
5.0	Chapter Overview .....	51

5.1	Tools Used.....	51
5.1.1	Java Programming language .....	51
5.1.2	Netbeans Integrated Development Environment (IDE) 7.1.2 .....	51
5.1.3	Bob's Concise Coding Convention .....	51
5.1.4	Doxygen Documentation tool .....	52
5.2	Snapshots of the application.....	52
CHAPTER 6 - TESTING .....		54
6.0	Chapter Overview .....	55
6.1	Equivalence classes of the inputs.....	55
6.2	Test Suite.....	57
6.3	Test Results .....	60
6.4	Observations from the testing.....	62
6.5	Proposed Solution for defending the DNS against DDoS attacks .....	63
Chapter 7 - Evaluation.....		64
7.0	Chapter overview .....	65
7.1	Attack to simulate .....	65
7.2	Methodology to use.....	65
7.2.1	Use of Game Theory software .....	65
7.2.2	Writing an application for use in the research.....	66
7.3	Type of game to use for the research .....	67
7.3.1	Strategy versus Extensive games .....	67
7.3.2	Cooperative versus Non-cooperative Game.....	68
7.4	Determination of the payoffs for the game.....	68
7.5	Ways of carrying out the Distributed Denial of Service attacks.....	68
7.6	Formulation of the game .....	69
7.7	Evaluation of the research process.....	69
	The Project Plan .....	70
	Project implementation .....	71
7.8	Reflection on the project duration .....	72
7.9	Limitation of the Research .....	72
7.10	Future enhancements to the research .....	72
CHAPTER 8 – CONCLUSION .....		74

8.0 Chapter Overview.....	75
8.1 Recap on the objectives.....	75
8.2 Work Done .....	75
8.3 Summary of the findings.....	76
8.4 Recommendation for Defending the DNS against DDoS attacks .....	76
8.5 Limitation of the research .....	76
8.6 Future enhancements to the research .....	77
8.7 Self-Reflection.....	77
REFERENCES.....	78
REFERENCES.....	79
APPENDIX 1 – USER MANUAL .....	84
Overview.....	85
Running the application .....	85

## LIST OF TABLES

Table 1 Direct Attack game .....	42
Table 2 Amplification attack game .....	43
Table 3 Equivalence classes of the inputs.....	56
Table 4 Test Suite (Test cases 1-12).....	58
Table 5 Test Suite (Test Cases 13-16).....	59
Table 6 Test Results for test cases 1 - 10.....	60
Table 7 Test results for test cases 11-16.....	61



# LIST OF FIGURES

Figure 1 Internet Users as percentage of population[2].....	2
Figure 2 Proportion of companies reporting security incidents with financial impact[7] .....	4
Figure 3 evolution of cyber-attacks[8] .....	5
Figure 4 Direct DDoS attack on DNS Server [47].....	32
Figure 5 Amplification DDoS Attack on DNS Server .....	38
Figure 6 Application's Logical Design .....	46
Figure 7 Initial Interface Design .....	47
Figure 8 Final design for Network Setting interface .....	48
Figure 9 Final design for the attacker settings interface .....	49
Figure 10 DNS Settings Form.....	52
Figure 11 Attacker Settings Form .....	53
Figure 12 Project Plan .....	70
Figure 13 Project Implementation chart .....	71
Figure 14 Legitimate Users Settings .....	85
Figure 15 Attacker Settings .....	86
Figure 16 Game Results illustration .....	87

## List of symbols used

Symbol	Description
A	Set of actions for the players
N	Set of players
$\theta_i$	Specific type of a player
$\Theta$	Set of the types of a player
F	Joint probability distribution according to the players drawn
U	Utility/payoff function of a player
$\lambda$	Probability of nature picking a direct attacker

# **CHAPTER ONE - INTRODUCTION**

# 1.0 BACKGROUND

A computer network is an interconnection of computers. An interconnection of these computer networks forms an internet. In today's world, the global Internet is a very vital part in people's lives. Some areas in which the Internet is being used include carrying out business transactions, communication, social networking and education. Singh argues that "in today's world, the fortunes of most organisations are tied with the information they possess and the sophistication with which they are able to manage it" [1]. The statistics from the World Bank report [2] on the Internet usage supports the fact that many people are now using the Internet. As it can be seen in Figure 1, there is an increase in the usage of the Internet all around the world.

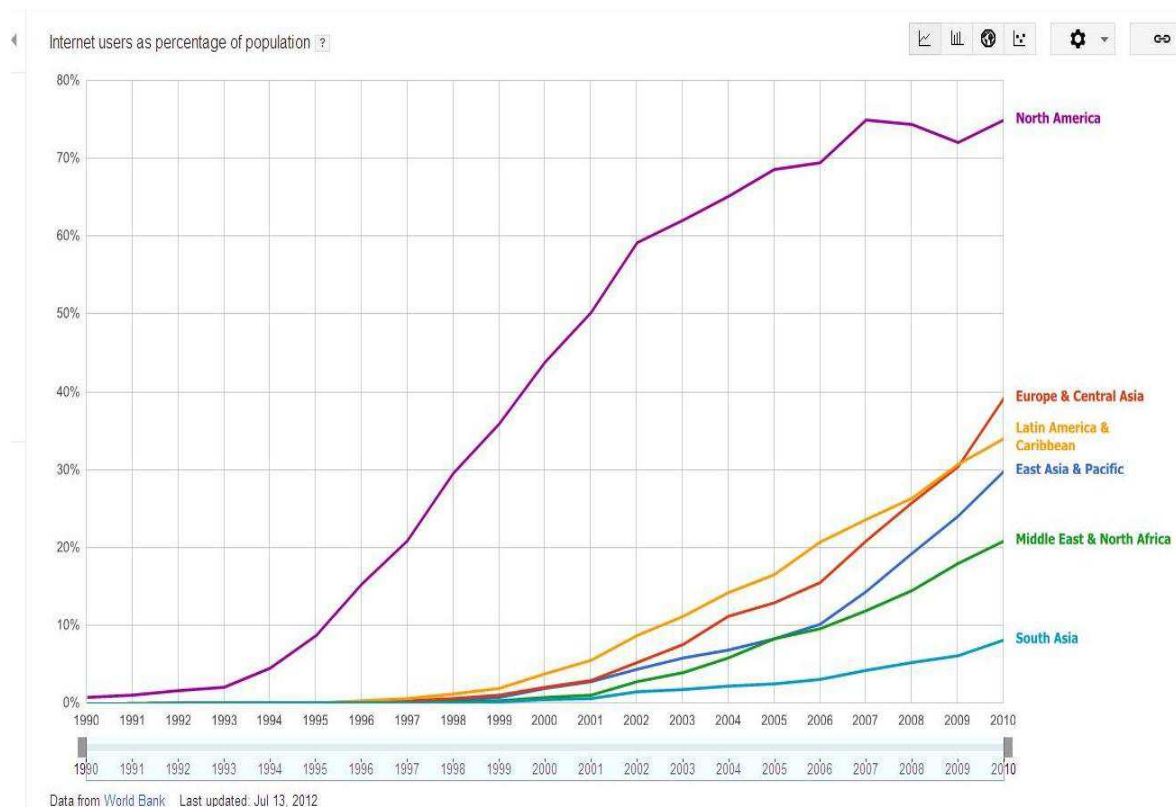


Figure 1 Internet Users as percentage of population[2]

As the world activities become more dependent on the Internet, concerns have arisen over the increase in cybercrimes. Cybercrimes is defined by the Australian-New Zealand Policing Advisory Agency in two ways. Firstly it is defined as "crimes directed at computing and communication technology technologies", and secondly, as "crimes where the use of the internet

or information technology is integral to the commission of the offence” [3]. Examples of cybercrimes include among others, credit card fraud, malware, social networking profile hacking, phishing, and denial of service.

There has been numerous cybercrimes targeting different organisations, governments and individuals. The seriousness of these attacks has been increasing over the years. For example, in the year 2004, businesses around the globe incurred costs between US\$169 billion and US\$204 billion due to cybercrime [4]. The Norton Cybercrime report of 2011 estimates that in the year 2010, the cost businesses incurred due to cybercrimes was more than \$388 billion [3]. In 2008, a cyber-attack was carried out in about 280 cities worldwide simultaneously. The attackers managed to get away with over \$9 million in cash [5]. It was estimated in 2010 that everyday about one million people become victims of cybercrimes [6]. The diagram in Figure 2 [7] shows that cyber-attacks against companies are increasing. Figure 3 shows how cyber-attacks evolved between the period 2000 and 2009 [8]. As it can be seen in this Figure, the motivation for attacks and the size of the attacks is changing. Ponemon Institute argue that “the most costly cybercrimes are commonly caused by malicious code, denial of service, stolen or hijacked devices and malicious insiders” [9]. Denial of service attacks are attacks aimed at causing legitimate users of a service not to have access to it. The domain name system (DNS) is often targeted. Additionally, even in cases where a DNS server was not the target of the Denial of service attacks, the DNS suffers indirectly from this attack since it is responsible for translating the computer names to IP addresses.

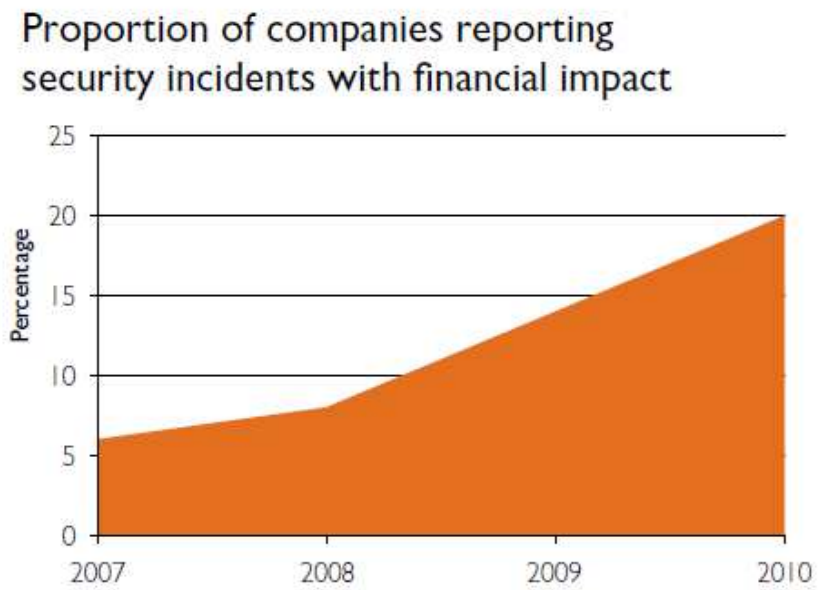


Figure 2 Proportion of companies reporting security incidents with financial impact[7]

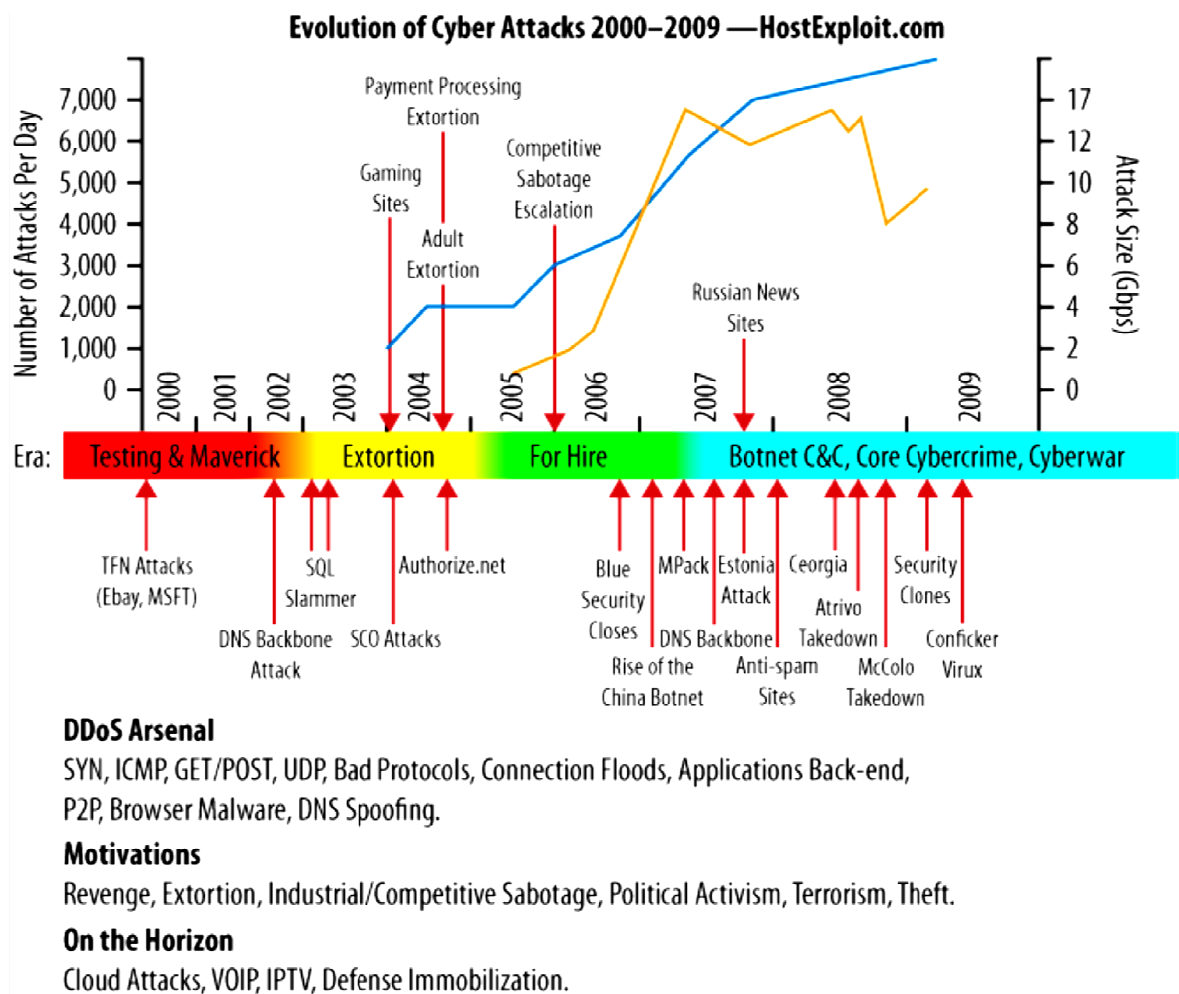


Figure 3 evolution of cyber-attacks[8]

### 1.0.1 Reasons why cybercrime has been successful

Several reasons have contributed to the success of the cybercrimes. The first reason has been that, the design of some protocols used in computer networks were not designed with security in mind. An example to this effect is the domain name system (DNS). The second reason has been the reluctance by some Internet users to use the available defence measures. These measures include anti-viruses and software patches to handle know vulnerabilities and malware. The third reason has been people's ignorance about cybercrime and ways of protecting against cybercrime.

### **1.0.2 Cyber Security Risk management**

There are concerns of the security of important resources and infrastructure owing to the vulnerabilities of the Internet to cyber-attacks. Internet service providers (ISP) and network administrators carry out risk management to determine the best way to protect the infrastructure on the Internet. Infrastructures which are most vital to the functioning of the Internet and most vulnerable to attacks are given higher priority when it comes to resource allocation.

There are several techniques used in carrying out risk management. To effectively analyse the effectiveness of these risk management, the criteria put together by Singh[1] is very helpful. According to these criteria, an effective risk management technique must meet the following:

1. *It must manage risk to an acceptable level.*
2. *It must provide decision-support.*
3. *It must be a continual process.*
4. *It must be aligned with an organisation's business objective.*
5. *It must be adaptive*
6. *It must be scalable*
7. *It must be compliant with Government and industry mandates*
8. *It must produce consistent results irrespective of who conducts the responsibilities associated with risk management*

### **1.0.3 Importance of the Domain Name System (DNS) for the functioning of the Internet**

A Domain name system (DNS) could be described as a hierarchical distributed naming system responsible for the translation of uniform resource locator (URL) and computer names of computers and services on a network into Internet Protocol (IP) addresses so that computers are able to contact other computers and access services on a network. This role played by the DNS is very vital to ensure the effective working of the Internet. Without presence of the DNS Internet users would have to remember the IP addresses of the all the services they want to use on the Internet and the Internet would not have expanded to the extent it has. Because of this vital role played by the DNS servers, they are being targeted by cyber-criminals. One common cyber-attack on the DNS has been the distributed denial of service attacks. In this attack, the aim is to exhaust the DNS server's resources so that legitimate users cannot use it. Additional information on the functioning of the DNS can be found in [10].

## **1.1 PROBLEM STATEMENT**

There are several risk management frameworks that have been proposed for different purposes and environments [11-13]. Most of these risk management frameworks assume that accidents are random events. These frameworks cannot be used in understanding and protecting the domain name system against distributed denial of service (DDoS) attacks and other cyber-attacks since attacks on the DNS are intentional. Furthermore, many risk management frameworks do not meet the criteria stated in section 1.2. The Internet Corporation for Assigned Names and Numbers [14] also recognise this fact and are in the process of revising the current framework. Bayesian games are useful in understanding risks posed by an intelligent attacker trying to attack a system. This research demonstrates how dynamic Bayesian game theory can be used in understanding DDoS attacks on the DNS.

## **1.2 AIMS AND OBJECTIVES**

The aims of this project are:

- 1) To model Distributed Denial of Service (DDoS) attacks on the DNS servers as a Bayesian game between the attacker and the defender (that is, the DNS server).
- 2) To determine the Nash Equilibrium for the Bayesian game model for the DDoS attack on DNS server.

The Nash Equilibrium in a game shows quantitatively the best strategies for the players in that game. If any player chooses a strategy that is not in the Nash equilibrium, they will get a poor result from their action.

- 3) To demonstrate how dynamically evolving games can be used to better understand risk management of critical infrastructure (that is, the Domain Name system).



## **1.3 BENEFITS OF THE RESEARCH**

This research offers two main benefits. The first one is that it demonstrates how the Internet can adapt to the changing risk environment around it. By using this game model, network managers can select the best defence strategies based on the quantitative effects of these strategies. The second benefit is that the quantitative aspect of risk management that is lacking in most frameworks has been incorporated through the use of Game Theory. Risk management decision can be made more effectively.

## **1.4 SCOPE AND LIMITATION OF THE RESEARCH**

This research only looks at bandwidth exhaustion DDoS attacks on DNS servers. It does not look at the DDoS attacks that exhaust the server's buffer or any other kind of attack of the DNS. The research also does not consider the effects that using multiple defence mechanism has on protecting the DNS against distributed denial of service attacks. It also does not look at the financial costs of the different defence mechanism.

## **1.5 METHODOLOGY**

In order to carry out a successful research on how Bayesian games can be used in protecting the DNS against DDoS attacks the methodology used comprised of literature review and implementation of the findings of the research in a computer application. A description of methodology is given below.

### **1.5.1 Literature review and research**

An extensive literature review was carried out to understand the limitations of existing risk management frameworks, ways of defending DNS servers against DDoS attacks and how game theory can be used to improve risk management of DNS servers and the Internet.

### **1.5.2 Programming**

The information gathered from the literature reviews was used in coming up with the program to simulate the interaction between a DNS server and an attacker during a distributed denial of service attack. The programming language that was used in developing the simulation is Java. Java programming language was chosen for a number of reasons. The first reason is that, it is

platform independent; hence an application can run on any computer that has the Java Virtual Machine (JMV) irrespective of the operating system being used. The second reason is that, Java is an open- source programming language. The third reason is that Java is a widely used programming language. This makes it easier to find materials on the programming language.

### **1.5.3 Testing**

To ensure that the simulation is working as expected it was tested using strong robust Equivalence Class testing. Strong Robust Equivalence Class testing is a Black Box testing technique. Black Box testing techniques are those which rely only on the requirement specification of an application. Black Box testing ensures that all the functionality specified in the requirements specification document is tested. In Strong Robust Equivalence Class testing, inputs are put in different sets known as equivalence classes [15]. Inputs in equivalence classes have the same properties. Therefore, to reduce on the testing effort, one has to only test one member of each equivalence class.

### **1.5.4 Limitation of the methodology used**

The best option for gathering the data for determining the players' benefits and probabilities of attacks would have been to carrying out actual experiment and use the raw data to determine the payoffs and the probabilities. However, this was not possible due to the fact that the amount of time given for the project is very limited for one to conduct experimental research and develop the solution.

## **1.6 Outline of the dissertation**

The rest of the dissertation is as follows. Chapter two summarises the researches that have been done in risk management, game theory and cyber-attacks. Chapter three gives the requirements for the application that was developed to demonstrate how game theory can be used in protecting DNS servers against distributed denial of service attacks. Chapter four describes the design of the game, how the solution of the game is found and the design of the application for this research. Chapter five describes the tools that were used in the implementation of the research. Chapter six describes the testing that was carried out and analyses the results that were obtained from running the application. Chapter seven gives an evaluation of the research process. Chapter eight gives the limitations of the research and future enhancements that can be done on the

research. Chapter nine gives the conclusion. A bibliography and Appendices are appended at the end.

## **CHAPTER 2 - LITERATURE REVIEW**

## 2.0 CHAPTER OVERVIEW

This literature review examines the main issues that surround the use of game theory in understanding how distributed denial of service (DDoS) attacks on the domain name system (DNS) and how an effective defence mechanism can be decided on by using a dynamic Bayesian game. This review focuses on the three research objectives set out in sub-section 1.4 of the introductory chapter. As stated earlier, the objectives of the project are:

- 1) To model Distributed Denial of Service (DDoS) attacks on the DNS servers as Bayesian game between the attacker and the defender.
- 2) To determine the Nash Equilibrium for the Bayesian game model for the DDoS attack on DNS server.
- 3) To demonstrate how dynamically evolving games can be used to better understand risk management of critical infrastructure (that is, the Domain Name system).

## 2.1 GAME THEORY CONCEPTS

Mayerson describes game theory as “the study of mathematical models of conflict and cooperation between intelligent decision-makers” [16]. Morgenstern and Neumann defines a game as “any interaction between agents that is governed by a set of rules specifying the possible moves for each participant and a set of outcomes for each possible combination of moves” [17]. Hargreaves-Heap and Varoufakis [18] argue that “game theory can be applied in any social interaction where individuals have some understanding of how outcome for one is affected not only by his or her actions but also by the actions of others”. The main concepts in game theory are pure strategy, mixed strategy, dominant strategy, dominated strategy, payoffs (or utilities) and Nash Equilibrium.

Polak defines a strategy as “one of the options a player can choose in a setting where the outcome depends not only on his own actions but on the action of others” [19]. Each strategy of a player is known as a pure strategy. Each player has a set of strategies known as a strategy set. If each strategy in the strategy set of a player is assigned a probability it is called a mixed strategy. The probability shows the chance of the player using a particular pure strategy in the strategy set. A dominated strategy is defined as “a strategy that guarantees a player lower outcomes than he

would receive [if he was] playing some other strategy, against all possible strategies of the opposition” [18]. They also define a dominant strategy as the exact opposite of the dominated strategy. Gibbons describes best response strategy as “the strategy (or strategies) which produces the most favourable outcome for a player, taking other players’ strategies as given” [20].

The most important concept in game theory is the Nash equilibrium, which [18] describes as “The outcomes of strategies  $R_i$  for player  $R$  and  $C_j$  for player(s)  $C$  is a Nash equilibrium of the game, and thus a potential ‘solution’, if  $R_i$  is the best response to  $C_j$  and, at once,  $C_j$  is the best reply to  $R_i$ ”. The Nash equilibrium is important in game theory because it represents the potential solution of a game. If any player uses a strategy other than the one specified in the Nash equilibrium they will obtain a lower outcome. There are variations of Nash equilibrium for different game types but they are all based on this definition of the Nash equilibrium.

There are several ways of classifying games. The first way of classifying games is either as strategic games or extensive games. A strategic game is one in which all the players select their strategies at the start of the game and they are not allowed to change the strategies as the game proceeds. That is the strategy remains unchanged throughout the game. In extensive games the players select their strategies at the end of the game but they can also change their strategies as the game proceeded. Strategies in an extensive game can be changed whenever the player has to make a decision [21].

The second way of classify is by the information that the player has [20]. If the players have all the information about their environment and the history of the moves each player made to get to the present stage, the game is known as a game with complete information. If the players do not know the exact action of the other player(s) to get to the present stage then the game is called a game with incomplete information [21]. An example of a game with complete information is a dynamic game while Bayesian game is an example of a game with incomplete information. Both kinds of games have been extensively used to study the interactions in different areas. These areas include economics and network security among others.

### 2.1.1 Bayesian games Concepts

In addition to the game theory concepts described earlier, Bayesian games introduce an additional concept of belief system. As already stated in the previous paragraph, in games with incomplete information, the players do not have full information about the game. Therefore, for players to make rational decisions they must have some belief on the actions that their opponent(s) made up to that particular stage of the game. This belief is given as a conditional probability known as the Bayes rule [18]. The players update their belief when observes the results of an interaction. The initial belief a player has is known as the prior belief. The updated belief is known as Posterior belief. The result observed by the player is known as the evidence. The chance of observing the result if a certain action is done is known as the likelihood. The expression of the Bayes rule is given in Eq. (1). A full description of Bayesian games can be found in the following references [18, 22, 23]

$$\text{Posterior belief} = \frac{\text{Likelihood} * \text{Prior belief}}{\text{evidence}} \quad (1)$$

## 2.2 vulnerabilities of the Domain Name System

Chatzis point out that the Domain name system is the largest distributed system. This makes it a very complex system [24]. When software is complex, it is liable to configuration errors and makes it vulnerable to security attacks. Rastegari et al. argue that the DNS is also vulnerable to denial of service (DoS) and distributed denial of service (DDoS) attacks because no authentication on the network layer takes place in the communication between a DNS server and a client host [25]. Microsoft Corporation also identified that their Windows DNS servers were vulnerable to denial of service attacks [26].

## 2.3 Distributed Denial of Service Attacks on the DNS Server

A Denial of service (DoS) attack is an attacked aimed at denying legitimate users of a service access to it. A DoS attack is carried out using a single source of packets. Distributed Denial of Service (DDoS) attack is a DoS attack using multiple sources of packets [27]. There are two kinds of DDoS attacks. The first type exploits vulnerabilities in the protocol being used by a website or web service. An example of this game type is the TCP SYN attack which utilises the

vulnerabilities in the Transport Control Protocol (TCP) to carry out a DDoS attack [28]. The second type of DDoS attack is the flooding attack. This is aimed at using up the network resources (usually the bandwidth) of the web service so that it cannot process legitimate requests from users [29]. The attacks are typically carried out using compromised hosts on the Internet known as bots. These bots form a botnet. A single botnet can have as many as ten million bots [8]. These bots send large number of packets to the target website or service using up its network resources. In recent years, the level of automation of DDoS attacks has increased causing the attacks to be more severe [29]. DDoS attacks on DNS servers are consumption attacks that disrupt services by limiting access to a DNS service [24].

In [29, 30] DDoS attacks are identified as being one of the most severe assaults on high-profile websites and other services on the Internet. DNS servers are also targets of DNS attacks. Furthermore, Chatzis argued that, even in cases where the DNS is not the target in DDoS attacks, it is always affected by the attack because it is responsible for name translation [24]. Brownlee et. al identified that, in January 2001 that about eighty-five percent (85%) of the traffic sent root DNS servers was bogus [31]. These bogus packets were aimed at causing the performance of the DNS to reduce. The biggest DDoS attack on the DNS was in 2002 when eight of the thirteen root servers were severely attacked [32, 33]. Kambouraskis estimated that in the year 2006, there nearly 4000 DDoS attacks taking place on the Internet every week [34].

Rastegari, et al. identified that there are mainly two common types of DDoS attacks that are carried out on the DNS [35]. The first one is known as the direct attack and the second one is the reflected amplification attack. Reflected amplification attacks are also called as reflection distributed denial of service (RDDoS) attacks or simply as the amplification attacks. In the direct DDoS attack, a large number of bots send query packets directly to the DNS server consuming the available network resources. In the amplification attack, a large number of bots send query packets to recursive DNS servers with the spoofed address of the victim DNS servers. The recursive servers send the response packets to the victim DNS server consuming the available network resources [33, 36]. Rastegari, et al. argues that amplification attacks have more adverse effects than direct DDoS attacks because the size of the response packets is much larger than the query packet [35].



Ahlawat and Sharma argue that denial of service attacks on DNS servers have been successful because it is very difficult to differentiate between genuine traffic and traffic aimed at causing a denial of service [29]. DDoS attacks on the DNS are carried out at the network and transport layers of the OSI communication model [30]. No authentication takes place at this layer, hence all packets are forwarded (Cisco System Inc., 2005).

### **2.3.1 Defence proposals for DDoS attacks.**

Peng et al. identifies that there are four categories of defence mechanisms against DDoS attacks [27]. The first category of defence mechanism is the attack prevention. Attack prevention mechanisms try to stop DDoS attacks before they reach the target. An example of this defence mechanism is the filtering of spoofed packets. Packet filtering should be done as close to the source of attack as possible. The second category of defence mechanisms is the attack source detection mechanism, which aims to identify the source of the attack. The limitation of this kind of defence mechanism is that, the attacker mostly uses spoofed addresses hence it is difficult to trace the source of the attacks. The third mechanism is the attack detection mechanism which aims to detect a DDoS attack when it occurs. The final kind of defence mechanism is the attack reaction mechanism that tries to eliminate or reduce the impact of an attack.

### **2.3.2 Attempts to solve Direct DDoS attacks**

There have been several researches on direct DDoS attacks. Hal and Bill suggested the use of IP traceback as a way of reducing DDoS attacks [37]. IP Traceback attempts to trace the origins of the attackers. When the true identity has been established, it is “taken out” by some administrative means, for example, by manually shutting down by a network manager. Such an approach is useful for reducing future occurrences of the attack but Sravani et al argues that such most traceback schemes consume huge amount of resources hence it is not an economical solution (2011). Furthermore some of the messages sent during the traceback process get filtered by the routers. This option may not be an attractive one for defending the DNS system during an attack since traceback can only be done after the attack. Wang et al. suggest that to identify a potential attack, you must monitor the flow of packet traffic between the client and the DNS server and analysing it using a statistical model [38]. However this scheme does not point out the cause of the anomaly.

Kotenko identifies that prevention of cyber-attacks (DDoS attacks included) is more effective if each device on the network is protected [39]. He argues that in protecting against DDoS attacks, the identification of an attack is best done near the victim while the separation of legitimate and bogus traffic is best done near the source [40]. Mansfield-Devine identify that many firms fell victims to DDoS attacks because of misconfigurations in the firewall [30]. This fact reduces the chances of Kotenko's suggestion succeeding.

Another defence mechanism that has been used is the over-provision of network bandwidth when a DDoS attack occurs. Changhua et al. argue that over-provision of the network bandwidth is the most effective means of defending against direct DDoS attacks on a DNS server [33]. Over-provision of network resources can be implemented using technologies such as anycast addressing [41].

### **2.3.3 Countermeasures against Amplification DDoS attacks**

Several countermeasures have been proposed to solve the amplification DDoS attack problem. Generally, these countermeasures involves using several layers of protection [32]. The first protection layer is to detect or prevent spoofing. Examples of these spoofing detection mechanisms can be found in [42-44]. Additional security mechanisms are included to ensure the integrity. Recursive servers are supposed to be configured to prevent open recursion on the name server from outside a network. These countermeasures face two challenges in reality. The first one is that most DNS servers are misconfigured [32]. The second challenge is that including security extension to DNS packets increases their size thus increasing the chances of a denial of service attack [45].

Kambourakis proposes a mechanism that is based on a one-to-one mapping of DNS requests and responses for detecting amplification attacks [34]. The protective mechanism maintains a list of requests sent out to authority DNS servers. If a response is received that has no matching request, it is identified as potentially being part of an amplification attack. Detection of these bogus responses triggers a defence mechanism that drops incoming responses from other Servers. Although this approach is useful, it has a few drawbacks. The first drawback is that the size of the memory containing the list of requests increases as the number of requests increases.

Secondly, in the early stages of the attack before the defence mechanism is in place, a denial of service is possible.

The ICANN Security and Stability Advisory committee [46] gave a solution that involved the verification of the source IP address, including advanced security for open recursion on name servers and securing the operations of the DNS server applications. It argues that this method will completely eliminate DNS amplification attacks. However Changhua et al. argue that most Internet Service Providers (ISPs) are not motivated to implement the solution because this mechanism protects other ISPs from attacks and not the implementer[33].

Deshpande et al. identifies random packet dropping, aggressive retries and packet filtering as the three commonly used countermeasures for protecting against amplification distributed denial of service attacks [45]. The random packet dropping mechanism uses either the perimeter router or a stateful firewall to randomly drop packets as the incoming traffic increases. The rate at which packets are dropped increases as the traffic to the server increases. The packet drop rate can be given by a sigmoid function. An example of the sigmoid function that can be used to specify the drop rate of packets can be found in [47]. This countermeasure ensures that the available network resources are not used up. The limitation of this countermeasure is that legitimate packets are also dropped together with the bogus or attack packets.

Packet filtering is the second defence mechanism identified by [45]. It attempts to identify the source of the attack. Once the attack source is found, packets from it are blocked. Deshpande et al. argue that filtering is comparatively very accurate with an error rate of less than 10% [45]. The computational cost of this defence mechanism depends on the filtering strategy and the strength of the attack. The firewall can be used for filtering incoming packets [48].

The aggressive retries countermeasure encourages legitimate users to increase the number of retries sent to the DNS servers if they do not get a response from the DNS server [49, 50]. the theory behind this strategy is that, if the attacker's rate of sending packets is constant, more legitimate traffic will be processed by increasing the number of retries [45]. It can be argued that using this mechanism will cause a prolonged DDoS attack since the network bandwidth is already overwhelmed with the incoming traffic.

## 2.4 Cyber-Security Risk Management

The terms cyber security risk management and information risk management are closely related. Cyber security risk management is aimed at “protecting against criminal or unauthorised use data” [51]. Singh defines information risk management as “a discipline that is focused on assessing, mitigating, monitoring and optimizing risks to information” [1]. a risk can be described as an event that causes unwanted changes in the cost, schedule or technical performance of a system if it occurs [52]. From Singh’s definition of information risk management, the first step in risk management is the identification of the risk. This is done by studying the system or the infrastructure for weaknesses [52]. Once the risk has been identified, the next stage is to analyse and prioritise the risks. The prioritisation of risks is according to the impact that a risky event would have on an infrastructure or a system. After prioritising the risks, resources are allocated to mitigate the risk or the consequence of the risk. Decision analysis is used in determining how resources should be allocated. According to Garvey [52], “Decision analysis involves choosing the best or most-preferred options among a set of competing options”. After the mitigation of the risk, the next step is to monitor the effectiveness of the protection measures used against the risk.

### 2.4.1 Importance of risk management

Garvey (2009, p.2) points out the following as the importance of risk management

1. *Risk management enables the early and continuous identification of risks.*
2. *Risk management enables program management to be risk-based. That is, the decisions made in program management take into consideration the risks involved.*
3. *Risk management justify the funds that are set aside for the managing of risks.*
4. *Risk management helps in resource allocation to different sections of an institution. And,*
5. *Risk management allows management to study the risk trends.*

Hopkin [53] identifies the following reasons as the importance of risk management

1. *Since the events that cause disruption are identified and action has been taken to reduce the chances of these events occurring, the operations of an organisation or infrastructure will be more efficient.*

2. *In a project, the effectiveness of processes is improved since the selection of the processes takes into consideration the risks involved in the available alternatives.*
3. *The risk involved in different strategies is fully analysed and better strategic decisions can therefore be made.*

#### **2.4.1 Criteria for risk management framework**

Singh identifies that there is a lack of definition of characteristics that a risk management system (framework) should have [1]. He therefore suggests the following criteria for evaluating risk management methods.

1. *A risk management system “must manage risk to an acceptable level based on enterprise’s risk appetite” [54]. This means that every institution has a specific level of risks they can tolerate.*
2. *A risk management system must be able to support decision-making.*
3. *Risk management should not be a continual process [55].*
4. *Risk management should be in line with the business objectives of the organisation.*
5. *The risks, threats and vulnerabilities of the organisation change, therefore risk management systems must be adaptable.*
6. *The complexity of risks changes, therefore a risk management system must be scalable to take into account these changing complexities.*
7. *Risk management systems must comply with government and industry mandate.*
8. *Risk management systems must produce consistent results regardless of the individual conducting the risk management.*

Singh further argues that most risk methods do not meet these outlined criteria[1]. He argues that, most risk management frameworks concentrate on the qualitative data. Sajjan et al. also support this argument [56]. They further add that the lack of quantitative data reduces the value of the risk management to decision makers.

## 2.4.2 Existing Risk Management frameworks

Modern risk management is based on the equation

$$\text{Risk} = \text{Threat} \times \text{Vulnerability} \times \text{Consequence} \quad (2)$$

2. According to this equation, a system's risk is the product of its vulnerability to attacks, the threats on it, and the consequences if it is attacked [12]. Vulnerability represents a weakness in the system. Threat is an event that can cause loss or damage to an asset if it occurred. The result of the threat occurring is known as the consequence. Young support the use of Eq. (2) in security risk management [57].

The Risk Analysis and Management of Critical Asset Protection (RAMCAP™) framework used by the United States Department of Homeland Security is based this equation. RAMCAP™ argues that the advantage of using Eq.(2) for risk analysis is that “risk associated with one asset can be added to others to obtain the aggregate risk of the entire facility” [12]. However Cox argues that this summation of risks is mostly incorrect mathematically [12]. In [13] it is argued that the department of homeland security must be careful when using Eq. (2) in risk assessment. It is not always appropriate.

The Risk management model proposed by the department of Homeland Security for international supply chain security uses the layered defence concept [58]. Layered defence aims to protect every node in a supply chain network in a bid to reduce risks. The European Central Bank supports this layered defence model [59]. The idea of a layered defence is difficult to achieve on the Internet owing to the fact that Internet is made up of different computer networks managed by different Internet service providers. Wheeler [60] identifies inconsistencies in the implementation of information security policies as one of the challenges faced in information security risk management.

Cox [12] further argues that equation (2) has a number of limitations that makes it unsuitable for use in protecting critical infrastructure against intelligent adversaries. The first limitation he identifies is that, the equation does not properly define a threat. The equation assumes the existence of a threat probability. In an intelligent attack, the adversaries base their attacks on what they suspect to be the defence strategy being used to protect an asset. In such a case, basing an estimate on the threat, as indicated in the equation is self-defeating. The second limitation is

that, calculating vulnerability using Event Trees can be difficult. Yao et al. [61] identify that even Fault Trees Analysis (EFA) which could be used as an alternative to Event Tree analysis suffers the same problem. The third limitation is that, the concept of consequences can be subjective. That is, in the case of a purposeful attack, an adversary has several ways of attacking an asset or infrastructure. The consequence depends on the choices made by the adversary and the protection strategy in use.

### **2.4.3 Cyber Security Risk management model**

Bier et al. [62] argue that risk analysis approaches based on Eq. (2) are inadequate for information security risk management. This is because these models assume that the threat is static. Assmuth and Hilden [63] identify that the concept of risk has different meanings to different users and context. This means that it would be wrong to use risk management models meant for managing random accidents to manage purposeful attacks on an infrastructure. In the case of security, the threat is not static. The equation does not take into consideration this fact. Taquechel proposes a Model Based Risk Assessment (MBRA) for carrying out risk assessment of transfer threat networks [64]. This model could be used in information security risk management since it takes into account vulnerabilities of individual components and those weaknesses that occur in subsystems. The challenge that would be faced in using this model would be in resource allocation. In this model, resources are allocated based on emergent behaviour (that is, unexpected behaviour), until the risk is minimised. Allocation of resources in this way can cause them to be mismanaged.

The International Standards organisation has provided guidelines for carrying out information security risk management in organisations [55]. However, they have not specified a methodology for carrying out the risk management. Therefore, the success of the risk management process depends on the methodology employed in the risk management. RAMCAP™ is an example of a framework that has implemented risk assessment guidelines wrongly [65] .

## **2.5 Use of game theory in Computer Networks and Risk Management**

Bier et al. [62] suggests that in such a case, the best way of modelling cyber security risk management is to model it as a game between rational players. the use of a game-theoretic approach is supported in [66], [67] and [68]. This is because game theory takes into account that the adversary is a rational person who is making intelligent decisions for his benefit [69]. The second reason why game-theoretic approaches are attractive for cyber security risk management is because of their rigor and mathematic depth. The risk estimation obtained by the use of game theory are better than those obtained by other means that do not take into consideration the action of the adversary. Furthermore, game theoretic solutions respond better in dynamic situations than other solutions [56].

Sevillano et al. [70] models the Somali pirates' case as a sequential game between the ship owner and the pirates. The sequential game used in this case is a three-step game. The ship owner makes the first move, followed by the pirates and finally the ship owner makes the last move. The payoffs of this game model the consequences of the ship owner's action and the pirate's actions. Merrick et al. argue that this approach allows the ship owner to model uncertainty about what the pirates' actions will be [67].

Wu et al. (2010) models denial of service and distributed denial of service attacks using static and dynamic game models. The use of the game models enables the defender to determine the best countermeasure against attacks. Lye and Wing [71] use a stochastic game model to study the interaction of the attacker and the defender in cyber security. Their model assumes that both players have perfect information about each other. Sajjan et al. [56] argue that this does not represent how the interactions happen in reality. In reality, the attacker and the defender do not know exactly what the other player's actions are. Shiva et al.[72] improves on Lye and Wing's game model using a game model that uses the min-max principle. In the min-max principle, one player is attempting to maximise their chances of winning a game while the other player is trying to minimising the first player's chances of winning [73]. Xu and Wooyang [74] also proposed the use of the "MinMax" principle to sustain websites against distributed denial of service



attacks. In the context of Network Security, the attacker is trying to maximise the impact of his attacks on an infrastructure while the defender is trying to reduce the impact of an attack.

Game-theoretic models however also have some drawbacks. The first one is that, the model may not easy to develop, quantify, apply or validate. The second limitation is that some assumptions made about the game may be unrealistic. These assumptions are mostly about initial beliefs of the players and the use of mixed strategies. The third limitation is that game-theoretic models ignore important psychological and behavioural factors that drive real-world behaviour.

## **2.6 challenges in cyber security risk management**

One challenge that risk management faces is finding the balance between the security needs of an infrastructure and the quality of service provided to the users [75]. If the security needs of the infrastructure are too tight, the quality of service rendered to users will be poor. In the case of the DNS system, this means that, if the security requirements are too tight, most of the user queries may not get to be processed. For example, if the protection measure used in protecting the DNS server only allows one query per second from a user, a user who accidentally sends two queries per second may have their queries dropped on the suspicion of being bogus.

## **2.7 evaluation and reflection**

Based on the surveyed literature, this project uses Bayesian games to demonstrate how an effective solution to the distributed denial of service of attacks on the domain name system (DNS). The DNS server's defence mechanisms used in this project are random dropping of packets, suspending hosts who are sending packets at a rate higher than the specified rate, and providing more bandwidth in the case of an attack. The two ways of carrying out distributed denial of service attacks on a DNS server is treated as two kinds of attackers, each with a dominant strategy of attacking and a dominated strategy of not attacking. The Bayesian Nash equilibrium represents the optimal defence strategy for the DNS server given the strategy of the attacker. The Bayesian belief gives the value of the risk of a particular type of DDoS attack on a DNS server.

# **CHAPTER 3 – REQUIREMENTS SPECIFICATION**

## **3.0 Chapter Overview**

This chapter gives a detailed description of the work that was involved in coming up with a simulation of how game theory can be used in protecting the domain naming system (DNS) against distributed denial of service (DDoS) attacks. It describes the requirements of the simulation, that is, what the simulation is expected to do, the inputs it will require and the outputs it will give, and the constraints under which the simulation can work

### **3.1 limitation of current cyber security risk management methods**

Most Current risk management methods are based on Eq. (2). These methods are inadequate for defending against Cyber Attacks. The following are the limitations that most of these risk management methods have.

1. Most risk management methods assume that the value of the threat, vulnerability and consequence is static. This assumption works only in environments where disasters are as a result of random works of nature. In environments where disasters happen due purposeful attacks such as cyber-crimes, the assumptions do not hold. The value of the threat, vulnerability and consequences in purposeful attacks depends on what the attacker does and the available defence mechanisms. In other words, the model is only useful for unintelligent accidents.
2. In risk management methods based on the equation risks are ranked and resources are allocated based on the risk ranking. However, this is not an effective way of allocating resources. The effective way is to allocate resources so as to reduce the overall risks.
3. Risks in cyber security change. A risk management model must be scalable to be able to accommodate the complexity of these changes. However, most risk management models are not scalable.
4. Risk management is supposed to be a continual process but most risk management methods consider risk management as a one-time activity.

## **3.2 DNS DDoS attack simulation**

This simulation has the following objectives:

- 1) To model Distributed Denial of Service (DDoS) attacks on the DNS servers as Bayesian game between the attacker and the defender.

The simulation aims to show how the DDoS attacks are carried out in reality. The model shows how the defender forms their belief on which kind of DDoS attack the attacker is likely to launch on a DNS server.

- 2) To determine the Bayesian Nash Equilibrium for the Bayesian game model for the DDoS attack on DNS server.
- 3) To demonstrate how dynamically evolving games can be used to better understand risk management of the Domain Name system.

## **3.3 Benefits of the research**

The simulation demonstrates how the use of dynamic Bayesian games can be useful in risk management of domain name system (DNS). It shows how the intentions of an adversary can be predicted with greater accuracy and hence reduces or eradicate the impact of the attack on a DNS.

## **3.3 Requirements of the simulation application**

A requirement is a feature that must be included in an application. There are two types of requirements in this simulation. These are functional and non-functional requirements. Functional requirements describe what the simulation is supposed to do. That is, it describes the inputs, the processes and the output that is expected from the simulation application. Non-functional requirements specify the constraints under which the simulation application is expected to work [76].

### **3.3.1 Functional requirements**

The functional requirements of this simulation application are data entry, data processing and output.

#### **3.3.1.1 Data Entry**

The simulation application allows the user to enter data about a distributed denial of service attack on a DNS server. The data that is entered include the bandwidth of the network on which the DNS server is, the kind of DDoS attack, the number of bots used during an attack and the rate at which the bots send packet to the DNS server.

#### **3.3.1.2 Data Processing**

Data processing specifies what is done on and with the entered data. In this simulation, the following are the data processing performed.

1. Verify that the entered data is of the right data type.
2. Determine the payoff for each player in the Bayesian game.
3. Update the DNS' belief on the kind of attack to expect from the attacker based on Bayes rule.
4. Determine the Nash equilibrium for a particular game. The Nash equilibrium in a Bayesian game shows the best strategies for all the players in a game.

#### **3.3.1.3 Output**

The simulation gives the following output:

1. It displays the payoff for all the possible strategies of the attacker and the DNS.
2. It displays the Nash Equilibrium in the game.
3. It displays the belief system of the DNS

### **3.3.2 Non- functional requirements**

Non-functional requirements of this simulation specify the minimal software and hardware requirements for the simulation application to run.

### **3.3.2.1 Software Requirement**

Software requirements specify the software that must be available on a device running the simulation. For this simulation to work, it requires that the computer should have the Java Virtual Machine (JVM). All applications made in Java require the JVM for them to run. A full description of the JVC can be found in [77]. The Java Development Kit (JDK) comes with the JVM. JDK version 5 or should be used for running this application.

### **3.3.2.2 Hardware requirements**

The computer running this simulation must have the following hardware requirements.

1. It must have a keyboard and a mouse for entering values into the application.
2. Random access memory of at least 256 megabytes.
3. Hard disk space of 110 megabytes.
4. Processing speed of 866 MHz.

# **CHAPTER 4 - DESIGN SPECIFICATION**

## 4.0 Chapter Overview

This chapter describes the design of the application for the simulation of distributed denial of service attacks on the domain name system. The design of the application is divided into four sections. The first section gives the design of the game, outlining the payoff functions for the players. The second section describes the functions for determining the Bayesian Nash equilibrium for the game. The third section gives the logical design of the application. The final section gives the interface design of the application.

### 4.1 Modelling DNS DDoS as a Bayesian Game

In this application, DDoS attacks on a DNS server is modelled as a Bayesian game between the defender (DNS server) and the attacker. There are two kinds of DDoS attack on the DNS server. These are direct DDoS attacks and amplification attacks. In this simulation, each of these attacks represents an attacker. That is, we have a direct attacker (that is, one who carries out direct DDoS attacks on the DNS) or an Amplification attacker (that is, one who carries out amplification attacks). Each attacker has two strategies, that is, “attack” and “No Attack”. The DNS server (the defender) has three strategies, namely, over-provision of network bandwidth, random dropping of packets by the firewall, and limiting the rate at which a host can send queries to the DNS server.

Formally, the Bayesian game for the DDoS attack on the DNS is the tuple  $(N, A, \Theta, F, U)$ , where

$N = \{\text{Attacker, Defender}\}$  is the set of players,

$A = \{A_{\text{attacker-i}}, A_{\text{defender}}\}$  is the set of actions for the players.

$A_{\text{attacker-i}} = \{\text{Attack with specified rate and bots, Not}\}$  is the action set of the attacker.

$A_{\text{defender}} = \{\text{Firewall drop rate, Over-provision of Network Bandwidth, specify rate at which a host can send requests to the DNS}\}$

$\Theta = \{\Theta_{\text{attacker}}, \Theta_{\text{defender}}\}$  is the set of types for each player.  $\theta_i \in \Theta_i$  represents a specific type of each player.

$\Theta_{\text{attacker}} = \{\text{Direct, Amplification}\}$

$\Theta_{\text{defender}} = \{\text{defender}\}$



$F: \Theta \rightarrow [0, 1]$  represents a joint probability distribution according to the type of players drawn.

$$p(\Theta_{\text{attacker}} = \text{Direct}) = \lambda$$

$$p(\Theta_{\text{attacker}} = \text{Amplification}) = 1 - \lambda.$$

$U = \{U_{\text{attacker}}, U_{\text{defender}}\}$  where  $U_i: A \times \Theta \rightarrow \mathbb{R}$  is the payoff function for the player  $i$ . that is, the payoff for a player  $i$  depends on the action and type of that player. The payoff will be expressed as a percentage.

The assumptions made in this game are:

1. All the bots are being controlled by one attacker.
2. The defender uses only one defence mechanism at a time.
3. The defender has an initial (prior) belief on which kind of attacker would be drawn.
4. The payoffs for the players are the bandwidth used by each player scaled by the bandwidth.

#### 4.1.1 Direct DDoS Attack Game

In this game the attacker directly attacks the DNS server by sending many requests for name resolutions at the same time using bots. Figure 4 [47] illustrates the direct attack type of game.

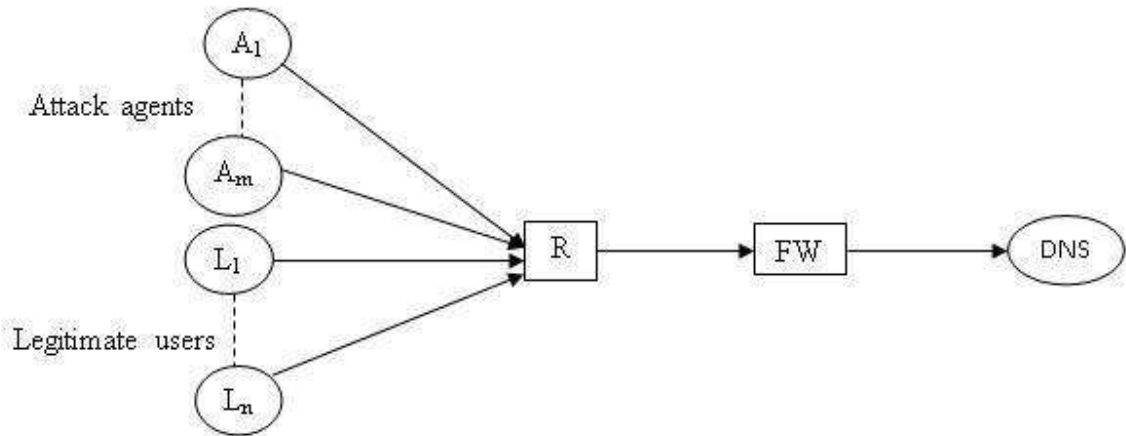


Figure 4 Direct DDoS attack on DNS Server [47]

In Figure 4, the letters  $A_1$  to  $A_m$  represents the attack agents (botnet) of the attacker. in this case, the attacker has  $m$  bots under his control. The letters  $L_1$  to  $L_n$  represents network traffic coming

from legitimate users. R represents a perimeter router that routes the network traffic to the DNS server. FW is the firewall of the DNS server. DNS represents the target DNS server.

#### **4.1.1.1 Amount of traffic generated by the attacker's packets**

Assuming that  $m$  is the number of bots used by the attacker and that each bot generates packets at the rate of  $r_a$  packets per second. If the size of each packet is  $x$  bytes, the total network traffic ( $F_a$ ) from the attacker's packets is given by Eq. (3).

$$F_a = r_a * x * m \quad (3)$$

#### **4.1.1.2 Amount of traffic generated by the legitimate users' packets**

Assuming that  $n$  is the number of legitimate user and that each legitimate user generates name resolution at the rate of  $r_d$  packets per second. The total network traffic ( $F_d$ ) from the legitimate user's packets is given by Eq.(4).

$$F_d = r_d * x * n \quad (4)$$

The total amount of bandwidth used is given by Eq. (5).

$$F_{ad} = F_a + F_d \quad (5)$$

#### **4.1.1.3 Fraction of the traffic that reaches the DNS server when the incoming traffic size exceeds the bandwidth**

If  $F_{ad}$  is greater than the bandwidth ( $B$ ), the DDoS attack is considered as successful. In such a case only a fraction of the incoming network traffic gets to reach the DNS server. The fraction of the network traffic reaching the server is given by Eq. (6).

$$\delta = B / F_{ad} \quad (6)$$

#### **4.1.1.4 Payoff when the DNS Server provides more bandwidth when there is a Direct DDoS attack (over-provision of network bandwidth)**

If there is an attack and the DNS provides more bandwidth to handle the incoming traffic, the players' payoffs are determined by checking whether packets are still dropped after providing more bandwidth.

#### 4.1.1.5 Payoffs when the incoming traffic does not exceed the provided bandwidth

The payoff for the attacker is the amount of bandwidth used by his packets if the incoming network traffic is less than the provided bandwidth ( $B^{\sim}$ ). The payoff for the defender is the difference between the available bandwidth and the bandwidth used the attacker. These payoff functions are given in equations 7 and 8.

$$U_{attacker1-1}: \text{Attack} * \text{Direct} = F_a / B^{\sim} \quad (7)$$

$$U_{defender1-1}: \text{Over} - \text{provision} * \text{Defender} = (B^{\sim} - F_a) / B^{\sim} \quad (8)$$

Where  $F_a$  is determined using Eq. (3).

#### 4.1.1.6 Payoffs when bandwidth used exceeds the provided bandwidth

When the incoming network traffic ( $F_{ad}$ ) exceeds the available bandwidth ( $B^{\sim}$ ) only a fraction of the incoming traffic reaches the DNS server. This fraction is given by equation 6.

The bandwidth used by the attacker's requests is therefore calculated as

$$B_{attacker} = F_a * \delta \quad (9)$$

The total amount of legitimate flow that is lost is given by Eq. (10).

$$B_{lost} = (1 - \delta)F_d \quad (10)$$

Where  $F_d$  is determined using Eq. (4). The aim of the attacker is to increase the  $B_{attacker}$  and  $B_{lost}$ .

The total amount of legitimate network traffic that gets to the DNS server is given by

$$B_{defender} = F_d * \delta \quad (11)$$

The payoff functions for the players when the bandwidth has been exceeded are given in equations 12 and 13.

$$U_{attacker1-1}: \text{Attack} * \text{Direct} = (B_{attacker} + B_{lost}) / B * 100\% \quad (12)$$

$$U_{defender1-1}: \text{OverProvision} * \text{Defender} = B_{defender} / B * 100\% \quad (13)$$

#### 4.1.1.7 Payoff for the Attacker when there is no attack

When there is no attack, the payoff for the attacker is 0 because no bandwidth is wasted. The payoff for the defender when using the over-provision of bandwidth strategy is equal to the provided bandwidth. The function representing these payoffs is given in equations 14 and 15. This is also applicable when there is no attack and the defender is using the rate-limiting strategy.

$$U_{\text{attacker1-2: Direct * No Attack}} = 0 \quad (14)$$

$$U_{\text{defender1-2: Overprovision * Defender}} = B/B * 100\% \quad (15)$$

#### 4.1.1.8 Payoffs for an Attack in the presence of a firewall

The firewall on the DNS server helps in protecting against DDoS attacks in two (2) ways. Firstly, the firewall on the DNS server is used to specify the maximum number of request that can be accepted from a single source per second [36]. Secondly, the firewall is used to specify the rate at which requests should be dropped to ensure that the DNS server's resources are not exhausted. Wu et al. (2010), suggests sigmoid function in Eq. (16) for determining the rate at which requests will be dropped by the firewall.

$$f(x) = \frac{1}{(1 + e^{-\beta(\frac{x-M}{B})})} \quad (16)$$

#### 4.1.1.10 Payoff when there is an attack and the firewall specifies the maximum rate of sending queries

Let  $\pi$  be the specified maximum rate at which hosts can send requests to the DNS server and  $r_a$  is the Bot's packet generating rate.

##### Payoff for the attacker when $r_a$ is less or equal to $\pi$

If the bot's rate of sending packets ( $r_a$ ) is less or equal to  $\pi$ , Eq. (3) is used to determine the bandwidth used by the attacker's packets. Equation 4 is used to determine the flow from legitimate users and Eq. (5) is used to determine the bandwidth of the incoming network traffic. Since all the incoming traffic is allowed to pass through the defence mechanism, the payoff

functions will depend on whether the bandwidth is exceeded or not. If the bandwidth is not exceeded the payoff functions is given by equations 17 and 18.

$$U_{\text{attacker1-3: Attack * Direct}} = F_a/B \sim * 100\% \quad (17)$$

$$U_{\text{defender1-3: Attack}_{(\text{attack rate} \leq \text{Limit Rate})} * \text{Defender}} = (B - F_a)/B * 100\% \quad (18)$$

If the bandwidth is exceeded, the payoff functions are given by the equations 19 and 20.

$$U_{\text{attacker1-3: Attack}_{(\text{attack rate} \leq \text{Limit Rate})} * \text{Direct}} = \left( (B_{\text{attacker}} + B_{\text{lost}}) / B * 100\% \right) \quad (19)$$

$$U_{\text{defender1-3: RateLimit * Defender}} = B_{\text{defender}}/B * 100\% \quad (20)$$

#### **When $r_a$ is greater than $\pi$**

if the rate at which each bot is sending name translation requests ( $r_a$ ) is greater than  $\pi$ , the firewall blocks all incoming requests from the botnet. The payoff for the attacker will be zero while the payoff for the defender is equal to the provided bandwidth. This is represented in equations 21 and 22.

$$U_{\text{attacker1-3: Attack}_{(\text{attack rate} > \text{Rate})}} = 0 \quad (21)$$

$$U_{\text{defender1-3: RateLimit * Defender}} = B/B * 100\% \quad (22)$$

#### **4.1.1.11 Payoff when there is no attack and the firewall has specified the rate limit for incoming requests**

When there is no attack, the payoff for the attacker is zero since no bandwidth is used by the attacker as indicated in Eq. (14). The payoff for the defender is equal to the network bandwidth. This is given in Eq. (23).

$$U_{\text{defender1-4 : RateLimit * Defender}} = B/B * 100\% \quad (23)$$

#### 4.1.1.12 Payoff when there is an attack and the firewall drops packets at random

In determining the drop rate for specific network traffic the sigmoid function in Eq. (16) is used. Given that the total incoming traffic is  $F_{ad}$ . The drop rate is given by  $F(F_{ad})$ . The payoff for the defender is the available bandwidth less the bandwidth that is lost by the legitimate or wasted by the bogus packets. The payoff for the attacker is the amount of legitimate traffic that is lost and the bandwidth used by the bogus packets. These payoff functions are represented in equations 24 and 25.

$$U_{attacker1-5}: \text{Attack} * \text{Direct} = ((f(F_{ad}) * F_d) + (1 - f(F_{ad})) * F_a) / B \quad (24)$$

$$U_{defender1-5}: \text{dropRate} * \text{Defender} = ((B - ((f(F_{ad}) * F_d) + (1 - f(F_{ad})) * F_a)) / B * 100\% \quad (25)$$

Where  $F_a$  is the traffic from the bots,  $F_d$  is the traffic from legitimate users and  $F_{ad}$  is the total incoming traffic.

#### 4.1.1.12 Payoff when there is no attack and the firewall has set a packet drop rate

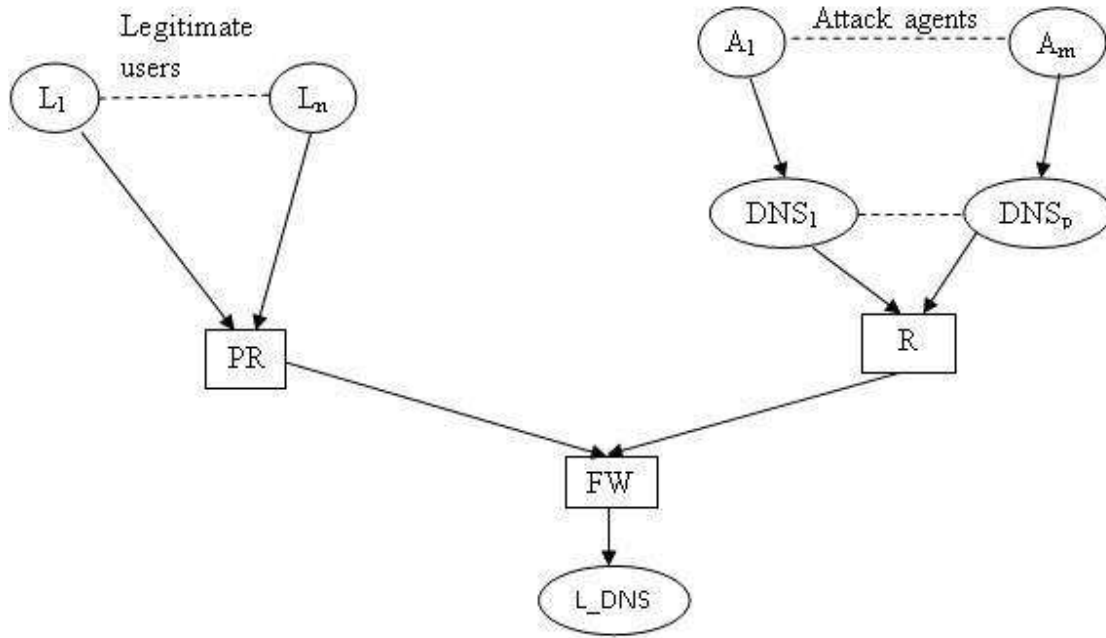
The payoff for the attacker when there is no attack is zero as indicated in Eq. (14). The payoff for the defender is the available bandwidth less the packets that are dropped at the firewall or router. The defender's payoff function is shown in Eq. 26.

$$U_{defender1-6}: \text{Firewall Drop rate} * \text{Defender} = (B - (f(F_d) * F_d)) / B \quad (26)$$

### 4.1.2 Reflected Attack (Amplification Attack) Game

In this game, the attacker indirectly causes exhaustion of a DNS server's resources. The attacker sends many name resolution requests to one or more recursive servers with the spoofed IP address of the target DNS server. The recursive DNS servers after receiving the responses from authority servers send them to the target DNS server. The response packets from the Authority DNS servers are much larger than the request packets. This causes the performance of the target DNS server to reduce. Figure 5 illustrates a reflected DDoS attack topology. In the Figure,  $L\_DNS$  represents the targeted DNS server.  $DNS_1$  to  $DNS_p$  represents the recursive DNS server(s) to which, the attacker sends spoofed requests using their attack agents  $A_1$  to  $A_m$ . R

represents the router that routes server-to-server traffic while PR represents the router that routes traffic from the users to the local DNS server and vice versa.



**Figure 5 Amplification DDoS Attack on DNS Server**

#### 4.1. 2.1 Payoffs for the DNS Amplification Attack Game

Like the direct DDoS attacks, the aim of the attacker is to use up the DNS' network bandwidth so that legitimate name resolution requests cannot be handled. In this game type, the attacker specifies the rate at which the botnet sends the requests to the recursive DNS servers. The recursive DNS servers are the ones that will send their responses to the targeted DNS server. The defender specifies his defence strategy. The payoffs are determined by considering the strategies of both players. The strategies of the defender are either to specify a rate at which incoming requests are dropped by the firewall or to increase the provision of bandwidth when an attack occurs.

#### 4.2.1.1 Calculating the bandwidth used in an attack

The amount of incoming traffic ( $F_a$ ) from the amplification attacker is also determined using the Eq. (3). The only difference in this case is that the size of the packet is set to that of the response size packet. The incoming traffic from the legitimate users ( $F_d$ ) is the same in both kinds of games. The total size of the incoming traffic ( $F_{ad}$ ) is determined using Eq. (5).

#### 4.1.2.2 Payoff when there is an attack and the defender provides more bandwidth

If there is an attack and the DNS provides more bandwidth to handle the incoming traffic, the payoffs are determined by checking whether packets are still dropped after providing more bandwidth.

#### 4.1.2.3 Payoffs when bandwidth used does not exceed the provided bandwidth

When the total incoming traffic ( $F_{ad}$ ) is less than the provided bandwidth ( $B^{\sim}$ ), the payoff for the attacker is equal to the percentage of the bandwidth used by his packets. The payoff for the defender is the difference between the provided bandwidth and the bandwidth used by the attacker's traffic. These payoff functions are represented using equations 27 and 28.

$$U_{\text{attacker2-1}} : \text{Attack} * \text{Amplification} = F_a / B^{\sim} * 100\% \quad (27)$$

$$U_{\text{defender2-1}} : \text{Over} - \text{provision} * \text{Defender} = (B^{\sim} - F_a) / B^{\sim} \quad (28)$$

#### 4.1.2.4 Payoffs when bandwidth used exceeds the provided bandwidth

When the incoming network traffic ( $F$ ) exceeds the available bandwidth ( $B^{\sim}$ ) only a fraction ( $\delta$ ) of the incoming traffic reaches the DNS server. This fraction is calculated using the Eq. (6). Each player will lose  $(1 - \delta)$  of their incoming traffic. Equation 9 is used to determine  $B_{\text{attacker}}$  which is the bandwidth used by the attack packets. Equation 10 is used to determine  $B_{\text{lost}}$  which is the amount of legitimate traffic that is lost in the attack. The payoff for the defender ( $B_{\text{defender}}$ ) is determined using Eq. (11). These payoff functions are represented using equations 29 and 30.

$$U_{\text{attacker2-1}} : \text{Attack} * \text{Amplification} = (B_{\text{attacker}} + B_{\text{lost}}) / B * 100\% \quad (29)$$

$$U_{\text{defender2-1}} : \text{Over} - \text{provision} * \text{Defender} = B_{\text{defender}} / B * 100\% \quad (30)$$



#### 4.1.2.5 Payoff when there is no attack

When there is no attack, the payoff for the attacker is zero. The payoff for the defender is the available bandwidth. These payoff functions are represented in equations 31 and 32.

$$U_{\text{attacker2-2: No Attack} * \text{Amplification}} = 0 \quad (31)$$

$$U_{\text{defender2-2: OverProvision} * \text{Defender}} = B/B * 100\% \quad (32)$$

#### 4.1.2.6 Payoff when there is an attack and the firewall specifies the maximum rate of sending queries

This defence mechanism is not effective for defending against amplification DDoS attack on the DNS server since all the packets from the recursive DNS servers will get passed this defence mechanism. In this case, the payoffs for the players depend on whether the bandwidth has been exceeded or not. If the bandwidth is exceeded only a fraction of the total incoming traffic reaches the server. This fraction is determined using Eq. (6). The payoff for the defender is the amount of the legitimate traffic that reaches the server. The payoff for the attacker is the sum of the amount of illegitimate traffic that reaches the server and the legitimate traffic that is dropped due to exhaustion of the bandwidth. Equations 33 and 34 represent this

$$U_{\text{attacker2-3: Attack} * \text{Amplification}} = (B_{\text{attacker}} + B_{\text{lost}})/B * 100\% \quad (33)$$

$$U_{\text{defender2-3: Ratelimit} * \text{Defender}} = B_{\text{defender}}/B * 100\% \quad (34)$$

Where  $B_{\text{attacker}}$  is determined using EQ. (9),  $B_{\text{lost}}$  is determined using Eq. (10), and  $B_{\text{defender}}$  is determined using Eq. (11).

#### 4.1.2.7 Payoff when there is no attack and the firewall specifies the maximum rate of sending queries

When there is no attack, all the available bandwidth can be used by the legitimate traffic. The payoff to the defender is the bandwidth size. The payoff to the attacker is zero as shown in Eq. (31). Equation 35 gives the defender's payoff.

$$U_{\text{defender2-4}}: \text{RateLimit} * \text{Defender} = B/B * 100\% \quad (35)$$

#### 4.1.1.8 Payoff when there is an attack and the firewall drops packet at random

The payoffs for the attacker and the defender when this defence strategy is used are calculated the same way as in the case of a direct attack. The payoff functions for the attacker and defender are given in the Eq. (36) and Eq. (37) respectively.

$$U_{\text{attacker2-5}}: \text{Attack} * \text{Amplification} = [(f(F_{\text{ad}}) * F_{\text{d}}) + (1 - f(F_{\text{ad}})) * F_{\text{a}}] / B \quad (36)$$

$$U_{\text{defender2-5}}: \text{drop rate} * \text{Defender} = (B - ((f(F_{\text{ad}}) * F_{\text{d}}) + (1 - f(F_{\text{ad}})) * F_{\text{a}})) / B * 100\% \quad (37)$$

Where  $F_{\text{a}}$  is the traffic from the bots,  $F_{\text{d}}$  is the traffic from legitimate users and  $F_{\text{ad}}$  is the total incoming bandwidth. These have been determined using equations 2, 3 and 4 respectively.

#### 4.1.2.9 Payoff when there is no attack but the firewall has set a packet drop rate

The payoff for the attacker when there is no attack is zero as shown in Eq. (31). The payoff for the defender is the difference between the available bandwidth (B) and the amount of legitimate packets that are dropped at the firewall. The defender's payoff function is given by the Eq. (38).

$$U_{\text{defender2-6}}: \text{Firewall drop rate} * \text{Defender} = (B - (f(F_{\text{ad}}) * F_{\text{d}})) / B * 100\% \quad (38)$$

### 4.1.3 Normal form representation of the game

In the normal form representation of the, the column player is the attacker and the row player is the defender. The first payoff in each cell is for the defender and the second one is for the attacker. A third player called nature is included. The role of this third player is to specify which game is being played, that is, either, the direct attack game or the amplification attack. Nature will select a direct attack game with the probability ( $\lambda$ )

#### 4.1.3.1 Direct DDoS Attack Game Representation

Table 1 shows the normal form representation of the direct DDoS attack game.

Table 1 Direct Attack game

Defender/Attacker	Attack	No attack
Over-provision of Network bandwidth	$U_{\text{defender1-1}}, U_{\text{attacker1-1}}$	$U_{\text{defender1-2}}, U_{\text{attacker1-2}}$
Rate of sending request limit specified	$U_{\text{defender1-3}}, U_{\text{attacker1-3}}$	$U_{\text{defender1-2}}, U_{\text{attacker1-4}}$
Firewall specifies the Drop rate (M)	$U_{\text{defender1-5}}, U_{\text{attacker1-5}}$	$U_{\text{defender2-2}}, U_{\text{attacker2-6}}$

The probability of nature choosing this game is ( $\lambda$ )

#### 4.1.3.2 Amplification DDoS Attack Game Representation

Table 2 shows the normal form of the Amplification DDoS attack game.

Table 2 Amplification attack game

Defender/Attacker	Attack	No attack
Over-provision of Network bandwidth	$U_{\text{defender2-1}}, U_{\text{attacker2-1}}$	$U_{\text{defender2-2}}, U_{\text{attacker2-2}}$
Rate of sending request limit specified	$U_{\text{defender2-3}}, U_{\text{attacker2-3}}$	$U_{\text{defender2-2}}, U_{\text{attacker2-2}}$
Rate of sending request limit specified	$U_{\text{defender2-5}}, U_{\text{attacker2-5}}$	$U_{\text{defender2-6}}, U_{\text{attacker2-2}}$

The probability of nature choosing this game is  $(1 - \lambda)$ .

#### 4.1.3 Preferences representation in Bayesian games

The preferences for the players (attacker and the defender) in this simulation takes into account the belief that the players have about which game is being played, whether a direct DDoS attack or an amplification attack game. This belief is represented by the probability  $\Pr(\theta|p_i)$ . This is interpreted as the belief that a player  $i$  of type  $p_i$  has that the state  $\theta$  has occurred. As earlier mentioned there are two types of attackers, the direct attacker and the amplification attacker. These two types of attackers are what are referred to in this context as the state. The expected payoff for a player  $i$  given the player type  $p_i$  is given in Eq. 39.

$$\sum_{\theta \in \Theta} \Pr(\theta|p_i) u_i((a_i, a_{-i}(\theta)), \theta). \quad (39)$$

This is the sum of what each player gets given the action of the other player. In the equation,  $a_i$  is the action played by player  $i$ ,  $a_{-i}$  is the action of the other player.  $\Theta$  is the set of types for each player.

#### 4.1.4 Nash Equilibrium

The Nash equilibrium for this game is the simulation is given in Eq. (40).

$$\sum_{\theta \in \Theta} \Pr(\theta|p_i) u_i \left( (a_i^*, a_{-i}^*(\theta)), \theta \right) \geq \sum_{\theta \in \Theta} \Pr(\theta|p_i) u_i \left( (a_i, a_{-i}^*(\theta)), \theta \right), \text{ for all } a_i. \quad (40)$$

This expression simply state that, an action profile is a Nash Equilibrium if the action chosen is the best given the choices of the other players. Where  $a_i^*(\theta)$  is the profile of action for player i and  $a_{-i}^*(\theta)$  is the action profile for the other players besides i.

#### 4.1.5 Updating the defender's belief about the attacker

In order for the defender to update their belief on the kind of attacker they are facing, the defender needs to keep an attacker's history profile. The histories profile the shows the kind of attackers that have been observed as the game evolves. In this simulation, aggregated values on the attacker types are given as the history. This is given in Eq. (41).

$$h_{att}^{def}(\theta_{att}, t_k) = (\sum \theta_{amp}, \sum \theta_{dir}) \quad (41)$$

Where  $t_k$ , represents the current stage in the game.  $\theta_{att}$  represents the attacker type, that is either amplification attacker or direct attacker.  $h_{att}^{def}$  is the history profile of the attacker with respect to the defender. The Bayes' rule has been used to update the defender's belief on the kind of attacker they are facing. Since this game has more than one stage, the defender needs to maintain a history of the attackers faced up to the current stage. Every stage of the game is represented by  $t_k$ . The belief is updated using Eq.(42). the equation has been adapted from the works of [78].

$$\mu_{att}(\theta_{att}|h_{att}^{def}(t_k)) = \frac{\mu_{att}(\theta_{att}|h_{att}^{def}(t_k)) * P(a_{att}(t_k)|\theta_{att}, h_{att}^{def}(t_k))}{\sum_{\theta_{att}} \mu_{att}(\theta_{att}|h_{att}^{def}(t_k)) P(a_{att}(t_k)|\theta_{att}, h_{att}^{def}(t_k))} \quad (42)$$

Where

$\mu_{att}(\theta_{att}|h_{att}^{def}(t_k))$  represents the conditional probability of the attacker being in a particular state (that is, either in direct attacker or amplification attacker type).

$P(a_{att}(t_k)|\theta_{att}, h_{att}^{def}(t_k))$  Gives the likelihood of the attacker being of a particular type given the history profile up to that stage.

The denominator gives the probability of seeing the state for both types of attackers.

## **4.2 Architectural design of the program**

This section describes the logic that has been used for the computer program that simulates the DDoS on the server as a dynamically evolving game between the attacker and the defender. Figure 6 gives the flowchart that has been used for representing the logic of the simulation. The reasons for using a flowchart in showing the logic for simulation are as follows:

- 1) Flowcharts visually display the logic of a program. Flowcharts use a combination of symbols and text to describe what is going on in a process or program.
- 2) Flowcharts are easier to translate into a computer program than a description in a natural language such as English.
- 3) Object-oriented design concepts are not useful in this case since the entire game is identified as a single class

#### 4.2.1 Application's Architecture Design flowchart

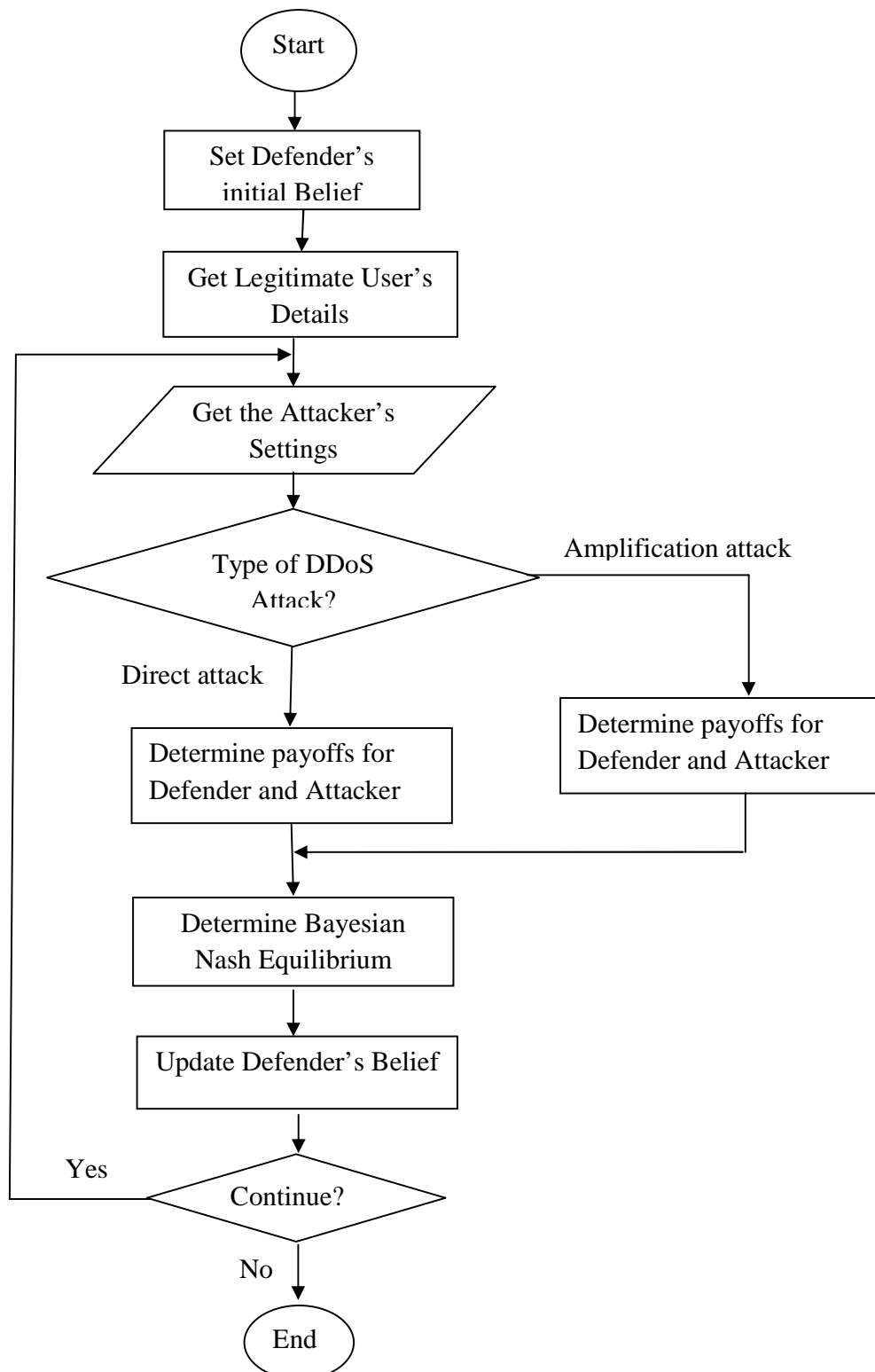


Figure 6 Application's Logical Design

## 4.3 Interface Design

The interface design shows the layout of the interface for the application that was developed for simulating DDoS attacks on DNS servers. The initial prototype of the design is shown in Figure 7. The final interface design that has been used is given in Figures 8 and 9 shows the final design. The initial prototype was modified to come up with the final design. The limitation that the initial design had was that the layout had too many items in a small space.

### 4.3.1 Initial Prototype

<p><u>Network Settings</u></p> <p>Network Bandwidth: <input type="text"/></p> <p>Legitimate Users: <input type="text"/></p> <p>Legitimate Sending Rate: <input type="text"/></p>	<p>Direct Attack Game</p> <table border="1"> <thead> <tr> <th></th> <th>Attack</th> <th>No Attack</th> </tr> </thead> <tbody> <tr> <td>Drop Rate</td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td>Over-Provision</td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td>Rate-Limit</td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> </tbody> </table>		Attack	No Attack	Drop Rate	<input type="text"/>	<input type="text"/>	Over-Provision	<input type="text"/>	<input type="text"/>	Rate-Limit	<input type="text"/>	<input type="text"/>	<p>Nash Equilibrium</p> <p>Direct Attack:</p> <p>Amplification Attack:</p>
	Attack	No Attack												
Drop Rate	<input type="text"/>	<input type="text"/>												
Over-Provision	<input type="text"/>	<input type="text"/>												
Rate-Limit	<input type="text"/>	<input type="text"/>												
<p><u>Attacker Settings</u></p> <p>Number of Bots <input type="text"/></p> <p>Bot's Sending Rate: <input type="text"/></p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><input type="radio"/> Direct Attack</p> <p><input checked="" type="radio"/> Amplification Attack</p> </div> <p style="text-align: center; margin-top: 10px;"><input type="button" value="Simulate"/></p>	<p>Amplification Attack Game</p> <table border="1"> <thead> <tr> <th></th> <th>Attack</th> <th>No Attack</th> </tr> </thead> <tbody> <tr> <td>Drop Rate</td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td>Over-Provision</td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td>Rate-Limit</td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> </tbody> </table>		Attack	No Attack	Drop Rate	<input type="text"/>	<input type="text"/>	Over-Provision	<input type="text"/>	<input type="text"/>	Rate-Limit	<input type="text"/>	<input type="text"/>	<p>Game History</p>
	Attack	No Attack												
Drop Rate	<input type="text"/>	<input type="text"/>												
Over-Provision	<input type="text"/>	<input type="text"/>												
Rate-Limit	<input type="text"/>	<input type="text"/>												

**Figure 7 Initial Interface Design**



### 4.3.2 Final Design

#### Interface Design for Networking Setting

Specify DNS Settings

Network Connectivity to DNS

Average number of legitimate users:

Legitimate User's rate of sending rate:

**Figure 8 Final design for Network Setting interface**

## Interface design for the attacker settings and results

Network Settings	Direct Attack Game	Nash Equilibrium
Number of Bots:	AttackNo Attack	Direct Attack:
Bot's Sending Rate:	Drop Rate <input type="text"/> <input type="text"/>	Amplification Attack:
<input type="radio"/> Direct Attack <input checked="" type="radio"/> Amplification Attack	Over-Provision <input type="text"/> <input type="text"/>	
	Rate-Limit <input type="text"/> <input type="text"/>	
<input type="button" value="Simulate"/>	<div>Amplification Attack Game</div> <div>AttackNo Attack</div> <div>Drop Rate<input type="text"/><input type="text"/></div> <div>Over-Provision<input type="text"/><input type="text"/></div> <div>Rate-Limit<input type="text"/><input type="text"/></div>	<div>Game History</div>
	<input type="button" value="Back"/>	

Figure 9 Final design for the attacker settings interface

# **CHAPTER 5 - IMPLEMENTATION**

## **5.0 Chapter Overview**

This chapter describes the implementation of the computer application to simulation how Bayesian games can be used to understand DDoS attacks on the DNS. The chapter describes the tools that were used in carrying out the research. Snapshots have been shown to show how the design of the game was translated into running application.

## **5.1 Tools Used**

### **5.1.1 Java Programming language**

Java is the programming language that was used for the implementation of the application to simulate distributed denial of service attacks on the domain name system. Java is an open-source, Object-Oriented programming language. The phrase “Open-Source” in software engineering refers to software that is available in source code form. The rights to the source code are held under an open-source license. It allows users to analyse, modify, improve and redistribute the software[79]. Object-oriented programming is a programming paradigm that looks at programs as a set of interacting objects. Applications developed in Object-Oriented programming languages are more easily understood by people since objects gives a more natural way of modelling real-world entities [80].

### **5.1.2 Netbeans Integrated Development Environment (IDE) 7.1.2**

The Netbeans IDE is a development environment for writing Java applications. It provides an easier way of creating the graphical user interface (GUI) than other IDEs such as Eclipse IDE and JCreator. For this reason the Netbeans IDE was used for implementing the application.

### **5.1.3 Bob’s Concise Coding Convention**

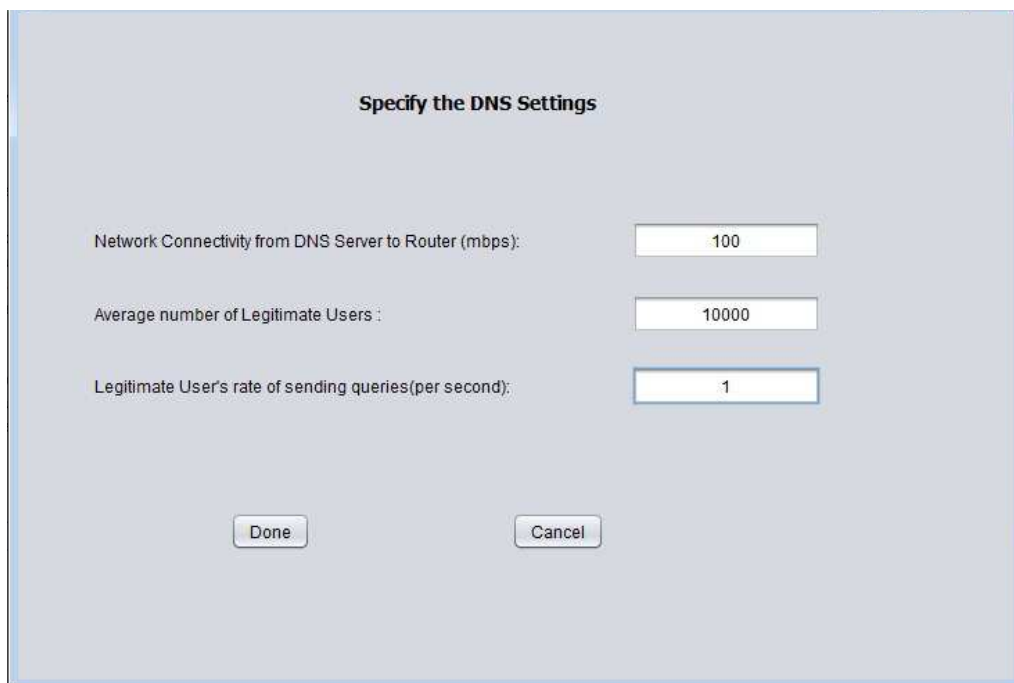
Bob’s Concise Coding Conventions are suggested programming guidelines to ensure that an application that is developed is concise, easy to understand and modify in future[81]. The conventions have been used to ensure that the source code for the application is concise and easy to understand.

### 5.1.4 Doxygen Documentation tool

In Software development, it is very important to document the source code. This is done to ensure that the software can be improved on in future. Doxygen is a source code documentation tool for programs written in C++, C, Java and a few other programming languages. Doxygen has been used to document the application.

## 5.2 Snapshots of the application

Figure 9 and 10 gives the snapshots of the application that was developed. The values indicated in the application are just for demonstration purposes.



**Specify the DNS Settings**

Network Connectivity from DNS Server to Router (mbps):	100
Average number of Legitimate Users :	10000
Legitimate User's rate of sending queries(per second):	1

Done Cancel

**Figure 10 DNS Settings Form**

Attacker's Settings

Number Of Bots: 1500000

Bot's sending rate(/ second): 1

Type of Attack

☐ Direct Attack

☒ Amplification Attack

Simulate

Simulation Results

Direct Attack Results

Defender/Attacker	Attack: Bots = 1500000, Rate = 1	No Attack
Firewall Drop Rate $F(r)$ :	94.21, -12.19	98.24, 0.0
provision of more Bandwidth:	0.66, 100.38	100.00, 0.0
Limit Requests From Hosts:	0.66, 103.80	100.00, 0.0
Defender's Belief in this game...	0.33	

Amplification Results

Defender/Attacker	Attack: Bots = 1500000, Rate = 1	No Attack
Firewall Drop Rate $F(r)$ :	94.21, -0.20	98.24, 0.0
Provision of more Bandwidth:	0.01, 101.69	100.00, 0.0
Limit Requests From Hosts:	0.01, 105.10	100.00, 0.0
Defender's Belief in this game ...	0.67	

Bayesian Nash Equilibrium

Defender's Payoff: 94.21

Defender's Strategy for Direct Attack:

Drop Rate

Defender Strategy for Amplification Attack:

Drop Rate

Attacker's Payoff: -0.2

Attacker Strategy:

attack

Game History

1. amplification attack  
Best Direct Attack Strategy: Drop Rate  
Best Amplification Attack Strategy: Drop Rate

Back

Figure 11 Attacker Settings Form

# **CHAPTER 6 - TESTING**

## 6.0 Chapter Overview

This chapter describes the testing that was done to ensure the application is working as expected. The aim of testing is to identify errors in an application. A successful test is one that detects errors in an application. The testing technique that was used was strong robust equivalence class testing. This testing technique puts in equivalence classes. All members of the equivalence class are treated in the same way. Strong Robust Equivalence class testing also tests values out of the normal expected range of inputs to insure that the application is robust. For each defence strategy used in this Bayesian game between the DNS and DDoS attackers, test suites have been provided to ensure the accuracy of the application that has been developed. A test suite is a set of test cases to test the accuracy of the application. The test cases have been selected to ensure that low-impact and high-impact DDoS attacks are tested. Low impact attacks are those attacks in which only a few thousand bots are used and the impact of the attack is not severe, whereas, high-impact attacks are those attacks in which hundreds of thousands bots are used in the attack and the impact is very severe.

### 6.1 Equivalence classes of the inputs

Table 3 summarises the equivalence classes for the individual inputs to the application and the action to be taken for each class of the inputs.



**Table 3 Equivalence classes of the inputs**

Input	Equivalence Class	Description of the Class	Action on the input
Network Bandwidth	Bandwidth $\leq 0$	Inputs that are 0 or less	Ask user to enter a correct number
	Bandwidth $> 0$	Inputs that are greater than 0	Accept the value
	Bandwidth not a number	Inputs that are outside the specified data types	Ask user to enter a correct number
Number of Bots	Bots $< 0$	Inputs that are less than 0	Ask user to enter a correct number
	Bots $\geq 0$	Inputs that are 0 or more	Accept the value
	Bots not a number	Inputs that are outside the specified data types	Ask user to enter a correct number
Bots requesting rate	Bots Rate $< 0$	Inputs that are less than 0	Ask user to enter a correct number
	Bots Rate $\geq 0$	Inputs that are 0 or more	Accept the value
	Bots rate not a number	Inputs that are outside the specified data types	Ask user to enter a correct number
Number of Legitimate Users	Users $< 0$	Inputs that are less than 0	Ask user to enter a correct number
	Users $\geq 0$	Inputs that are 0 or more	Accept the value
	Users not a number	Inputs that are outside the specified data types	Ask user to enter a correct number
Bots requesting rate	Bots Rate $< 0$	Inputs that are less than 0	Ask user to enter a correct number
	Bots Rate $\geq 0$	Inputs that are 0 or more	Accept the value
	Bots rate not a number	Inputs that are outside the specified data types	Ask user to enter a correct number

## 6.2 Test Suite

Tables 4 and 5 give the test suite that was used in testing the application. The hyphens (-) in the Table indicate that a particular input or output is not required since the earlier one is wrong. The abbreviations Def is used for Defender and Att for attacker respectively. In the test cases, it is assumed that in each test case, the initial belief of the defender is 0.5. If the initial belief is different, the output of the belief would be different. The bandwidth for the network is assumed to be 100Mbps. the same tests can be carried out using any value for the bandwidth.

**Table 4 Test Suite (Test cases 1-12)**

Inputs							Output						
Tes t ID	Strategy Tested	Bandwidth (mbps)	Legitimate Users	Legitimate Rate	Number of Bots	Bot Rate	Attack Type	Direct Attack Payoff		Amplification attack Payoff		Direct attack Belief	Message/Nash Equilibrium
								Att	Def	Att	Def		
1	All	0	-	-	-	-	-	-	-	-	-	-	Enter correct value
2	All	A	-	-	-	-	-	-	-	-	-	-	Enter correct value
3	All	1	-1	-	-	-	-	-	-	-	-	-	Enter correct value
4	All	1	B	-	-	-	-	-	-	-	-	-	Enter correct value
5	All	1	1	-2	-	-	-	-	-	-	-	-	Enter correct value
6	All	1	1	A	-	-	-	-	-	-	-	-	Enter correct value
7	All	1	1	1	-1	-	-	-	-	-	-	-	Enter correct value
8	All	1	1	1	V	-	-	-	-	-	-	-	Enter correct value
9	All	1	1	1	1	-1	-	-	-	-	-	-	Enter correct value
10	All	1	1	1	1	B	-	-	-	-	-	-	Enter correct value
11	Rate- Limiting (limit = 1)	100	15000	2	45000	4	Direct	0	0	100	0	0.67	Over-provision of bandwidth for Direct attacks and Packet Dropping for amplification attacks
12	Rate- Limiting (limit = 1)	100	15000	1	45000	1	Amplificatio n	23.0	77	106.6	0.5	0.33	Rate-limiting or over-provision of bandwidth for direct attacks and rate-limiting for amplification attacks

**Table 5 Test Suite (Test Cases 13-16)**

Test ID	Strategy Tested	Bandwidth (mbps)	Legitimate Users	Legitimate Rate	Number of Bots	Bot's Rate	Attack Type	Direct Attack Payoff		Amplification attack Payoff		Direct attack Belief	Nash Equilibrium Strategy
								Att	Def	Att	Def		
13	Random Packet Dropping	100	1000	3	6000	5	Amplification	60.6	89.8	89.8	0.2	0.33	Dropping packets for both attacks
14	Random Packet Dropping	100	15000	3	45000	4	Direct	43.3	50.1	0.4	50.1	0.33	Provision of Bandwidth for direct attack and dropping packets for amplification attacks
15	Over-provision of bandwidth	100	15000	2	1000	3	Direct	1.54	98.46	32	68	0.67	over-provision of bandwidth for direct attacks and random packet dropping for amplification attacks
16	Over-Provision of Bandwidth	100	10000	3	180000	3	Amplification	92.2	7.8	0.1	104.9	0.33	Dropping packets for both attacks

## 6.3 Test Results

After running the test cases, errors were observed in the application. These errors were corrected and the test cases run again. The application produced the results as expected. Tables 6 and 7 show the results that were obtained during the testing of the application after all the errors were corrected.

**Table 6 Test Results for test cases 1 - 10**

Test ID	Strategy	Expected Results				Actual Results				Message/ Nash Equilibrium
		Direct Attack		Amplification Attack		Direct Attack		Amplification Attack		
		Attacker	Defender	Attacker	Defender	Attacker	Defender	Attacker	Defender	
1	All	-	-	-	-	-	-	-	-	Enter correct value
2	All	-	-	-	-	-	-	-	-	Enter correct value
3	All	-	-	-	-	-	-	-	-	Enter correct value
4	All	-	-	-	-	-	-	-	-	Enter correct value
5	All	-	-	-	-	-	-	-	-	Enter correct value
6	All	-	-	-	-	-	-	-	-	Enter correct value
7	All	-	-	-	-	-	-	-	-	Enter correct value
8	All	-	-	-	-	-	-	-	-	Enter correct value
9	All	-	-	-	-	-	-	-	-	Enter correct value
10	All	-	-	-	-	-	-	-	-	Enter correct value

**Table 7 Test results for test cases 11-16**

Test ID	Strategy	Expected Results				Actual Results				Message/ Nash Equilibrium
		Direct Attack		Amplification Attack		Direct Attack		Amplification Attack		
		Att	Def	Att	Def	Att	Def	Att	Def	
11	Rate-Limiting(rate = 1)	0	0	0	0.3	0	0	100	0	Over-provision for direct attacks and dropping packets for amplification attacks
12	Rate-Limiting	23.0	77.0	106.6	0.5	23.04	76.96	106.62	0.53	Dropping packets for both direct attacks and for amplification attacks
13	Random Packet Dropping	60.6	89.8	0.2	89.8	60.61	89.76	0.17	89.76	Dropping packets for both attacks
14	Random Packet dropping	43.4	53.4	0.3	53.4	43.38	53.36	0.27	53.36	Over-provision of bandwidth for direct attack and dropping packets for amplification attacks
15	Over-provision of bandwidth	1.5	98.5	32	68	1.54	98.46	32.00	68.00	over-provision of bandwidth for direct attacks and dropping packets for amplification attacks
16	Over-provision of Bandwidth	92.2	7.8	115.2	0.1	92.16	7.84	115.18	0.09	Dropping packets for both attacks

The following should be noted on Tables 7 and 8.

1. The first ten test cases have been omitted since they are just being used to test the robustness of the application.
2. The network bandwidth used in these test cases was 100 Mbps.
3. The incoming traffic in amplification attacks is likely to exceed the network bandwidth even in cases where only a few thousand bots are used.
4. In this project, any attack in which the incoming bandwidth is less than or just slightly more than the available bandwidth is a low-impact attack while any attack in which the incoming traffic is more than 120% of the available bandwidth is a high-impact attack.

## **6.4 Observations from the testing**

From the results obtained in testing of the simulation, it is clear that there no single defence mechanism that can perfectly protect a DNS server from both high-impact and low-impact DDoS attacks. For example, the random dropping of packets is effective for high-impact DDoS attacks. However, using this defence strategy for low-impact DDoS attacks will cause some legitimate requests to be unnecessarily dropped.

The provision of more bandwidth when there is a low-impact DDoS attack is more effective than the other strategies. The strategy of limiting the rate at which requests are sent is not very effective since an attacker can circumvent this defence mechanism by using more bots and a low sending rate.

The rate-limiting strategy is not an effective defence strategy since the rate of sending requests differs. For example, in July, 2009 following the death of Michael Jackson, the rate of sending queries increased such that websites such as Google, Wikipedia and BBC experienced an accidental denial of service attack [82]. If the rate-limiting strategy was used, all the incoming queries to the DNS server would be treated as bogus and thus they would be dropped.

## **6.5 Proposed Solution for defending the DNS against DDoS attacks**

A solution that could be more effective would be to use a combination of providing more bandwidth for low-impact DDoS attacks and random dropping of packets when the attack impact increases. By providing more bandwidth when there is a low-impact DDoS attack all the incoming requests could reach the DNS server. When there is a high-impact DDoS attack, the best response for the DNS server is to drop the incoming traffic. In this case, most of the legitimate requests will be dropped but this ensures that the DNS server will not be damaged.



## **Chapter 7 - Evaluation**

## **7.0 Chapter overview**

This chapter describes the alternative approaches that were considered in coming up with a model to demonstrate how game theory can be used in protecting the domain name system. It highlights both the weaknesses and the strengths of the methodology used in comparison to other available alternatives that were considered. The chapter also evaluates the development process.

### **7.1 Attack to simulate**

The attacks that I considered were cache poisoning and distributed denial of service. In cache poisoning attack, the attacker attempts to poison the cache of the DNS server with bogus records. With the implementation of the DNS security extension (DNSSEC), cache poisoning attacks are almost impossible to carry out. For this reason this attack was not researched further [83] [84]. Distributed denial of service attacks on the other hand continues to be a threat to the functioning of the Domain Name System because there is no way of differentiating legitimate packets and bogus packets. For this reason DDoS attacks were selected for this research.

### **7.2 Methodology to use**

To model distributed denial of service attacks on the Domain name system (DNS) as a Bayesian game, I had two options. The first option was to use available game theory software or to write an application to simulate the game.

#### **7.2.1 Use of Game Theory software**

Gambit software for game theory was the open-source software that was considered for this research. Using this software would have reduced the time that would be required to carry out the project. However, the downside of using Gambit is that one has to compute the payoffs manually. The game only computes the Nash equilibrium. Calculating the payoffs manually is prone to error.

## **7.2.2 Writing an application for use in the research**

The second option that was considered was to write an application for use in the research. This takes more time than using the available game theory software but it has the advantage in that once the program is written, there is no need to manually calculate the payoffs. Furthermore, the results obtained with the computer program are more accurate than those calculated manually. Because of the limitations of GamBit, I decided to write a simple application for the research.

### **7.2.2.1 Programming language to use for the application**

The programming languages that were considered when developing the application were MATLAB and Java. MATLAB was an attractive option because it has tools that make easier to develop the graphical user interface for applications. In addition, MATLAB does not require additional libraries to produce graphical output. However MATLAB is a proprietary programming language. One is required to have a licence to use it. Due to this fact, MATLAB is not as widely used as Java for Object-Oriented programming. MATLAB has a further limitation in that one does not explicitly declare variables. These are implicitly declared at the application's runtime. This makes the task of testing an application at the time of compiling it difficult.

Java on the other hand is an open-source programming language. This means that it is free for anyone to use. Because of this Java is more widely used than MATLAB. Furthermore, Java programming documentation is much easier to find than documentation for MATLAB. Furthermore, Java has the unit testing framework called JUnit which can be used to automatically generation of test cases hence reducing the work of testing applications. These reasons prompted me to use Java for development of the program.

### **7.2.2.2 Use of Object-Oriented diagrams in the design**

Object-Oriented programming has a rich set of diagrams that could be used in the design of the simulation application. These include class diagrams, use case diagrams and activity diagrams. These diagrams help in the design of complex systems. Complex systems have many interacting objects and each object keeps its own information private. However, in an application such as this one which has just one object of interest, Object-Oriented designs do not offer any advantage. A flowchart is very helpful in this case since there are no interacting objects. Flowcharts show the sequence of steps involved in carrying out a process, therefore it is easier to

understand than a description in a natural language such as English. For this reason, a flowchart was used.

#### **7.2.2.3 Testing Technique to use**

There are several testing techniques that could be used for testing this application. The two techniques that were considered were decision table-based testing and equivalence class testing. Decision Table-based testing ensures that all conditions in the application are tested. Equivalence class testing ensures that all possible combinations of inputs are tested. In this research strong robust Equivalence class testing has been used to ensure that the application is working as expected. Strong robust Equivalence class testing is just as good as Decision Table-based testing since by testing all possible inputs to the application, all the decisions in an application will be checked.

### **7.3 Type of game to use for the research**

There are several classes of game types that could have been used in coming up with the simulation. As described in the literature review, there are several ways of classifying games. One way is to classify a game either as a strategy game or an extensive form game. The second way is to classify the game either as a game with complete information or a game with incomplete information. This simulation has been modelled as a non-cooperative, extensive-form game with incomplete information.

#### **7.3.1 Strategy versus Extensive games**

In a Strategy game, the players select their strategy at the start of the game and do not change the strategy. In an extensive form game, the players can change their strategies at any time. This game is not effective for carrying out risk adversarial assessment. This is because, in adversarial risk management, the attacker and the defender can change a strategy to ensure that they get a better result. An extensive –form game with complete information is a better option than the use of a strategy game for this simulation than a strategy game. However, this is not the best option for the simulation because it assumes that each player is aware of what the exact action of the other player. In real-life situations, the attacker and the defender do not have full information about each other's actions or payoffs.

### **7.3.2 Cooperative versus Non-cooperative Game**

In cooperative games, players cooperate to get good results. However, in adversarial risk management, each player is trying to gain a better result than the other. For this reason, cooperative games are not useful for risk management of DNS servers. A cooperative game would be useful if the research was about ways in which a number of defenders can cooperate to defend a DNS server.

In this research non-cooperative Extensive-form Games with incomplete information was used in this research because they take into account that players in a risk game do not know what the other player has done.

## **7.4 Determination of the payoffs for the game**

The payoffs for the attacker and the defender could either be given in terms of the bandwidth or the memory that the packets of each player occupies. Using bandwidth usage as a measure of the payoff for the players is a better option since DDoS attacks on DNS server target consumption of the bandwidth. Furthermore, in a DDoS attack where the bandwidth is consumed, only a few packets, if any get to reach the server. This implies that even when a successful attack has occurred, the memory usage will still appear as normal.

## **7.5 Ways of carrying out the Distributed Denial of Service attacks**

There are two main ways in which DDoS attacks are carried out. The first way of carrying out a DDoS attack is by targeting weaknesses on the protocols being used by the web server. The second way of carrying out the DDoS attack is by flooding the web server with many packets hence using up its resources. The DNS server is mostly vulnerable to flooding attacks. Therefore a study of DDoS attacks should focus on flooding attacks.

## 7.6 Formulation of the game

There are two ways in which this game could have been formulated. The first way is having one attacker whose strategies are direct attack and amplification attack. This approach eliminates the need for the “No Attack” strategy making the game simpler. However, in this approach, amplification attack would be a dominant strategy. Therefore the effects of how a direct attack affects a DNS would not be seen.

The second way is to have two kinds of attackers, namely, direct attacker and amplification attacker. Each of these attackers has a dominant strategy of attacking and a dominated strategy of not attacking. This game is a bit complex than the first one since the defender must now have a belief system. However, in this game, the effects of both kinds of attacks would be examined. For this reason, this game was used for the research.

## 7.7 Evaluation of the research process

Figure 12 shows the project plan that was made for the project. There are significant differences between the project plan and the actual implementation of the project. The initial intention for the project was to model distributed denial of service and cache poisoning attacks on DNS servers. However, Cache-poisoning attacks were removed from the project after literature suggested that the implementation of DNS Security extensions (DNSSEC) has made it almost impossible to carry out this attack [83]-[84]. Following the removal of Cache-Poisoning attacks from the scope of the project, the literature review and the requirement specification that was written for the project prior to June, 2012 was to be reworked to reflect this change in the research scope. As a result, the entire project plan was revised to adapt to this change. The other major change that was made was in the programming language to use for the project. After spending a number of weeks learning MATLAB 7.0, I decided not to use MATLAB since there were limited resources on Object-Oriented programming in MATLAB. These two major changes caused the completion of the project to be delayed for about seven days. The Gantt chart in Figure 13 shows how the project was actually implemented.

## The Project Plan

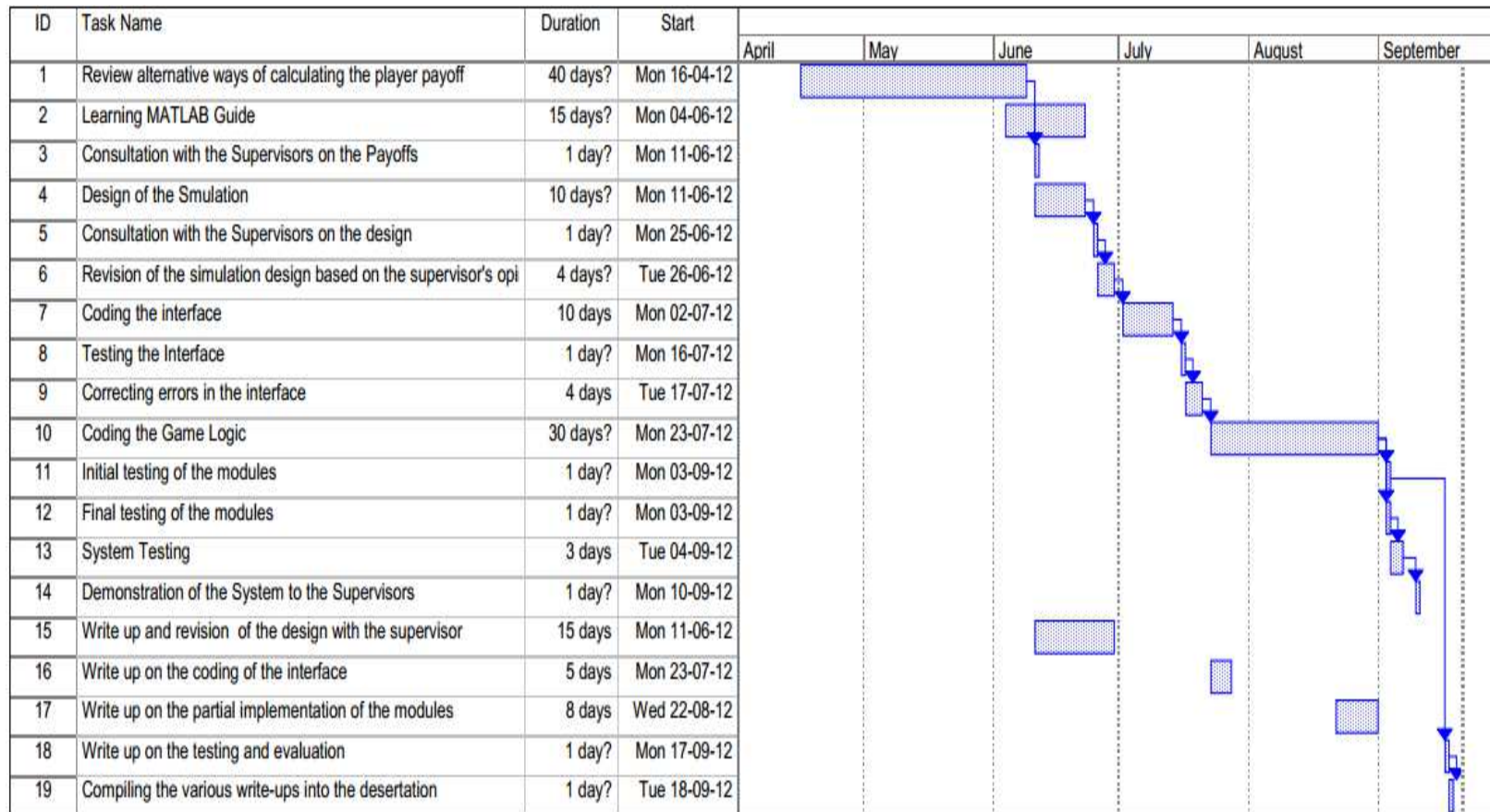


Figure 12 Project Plan

## Project implementation

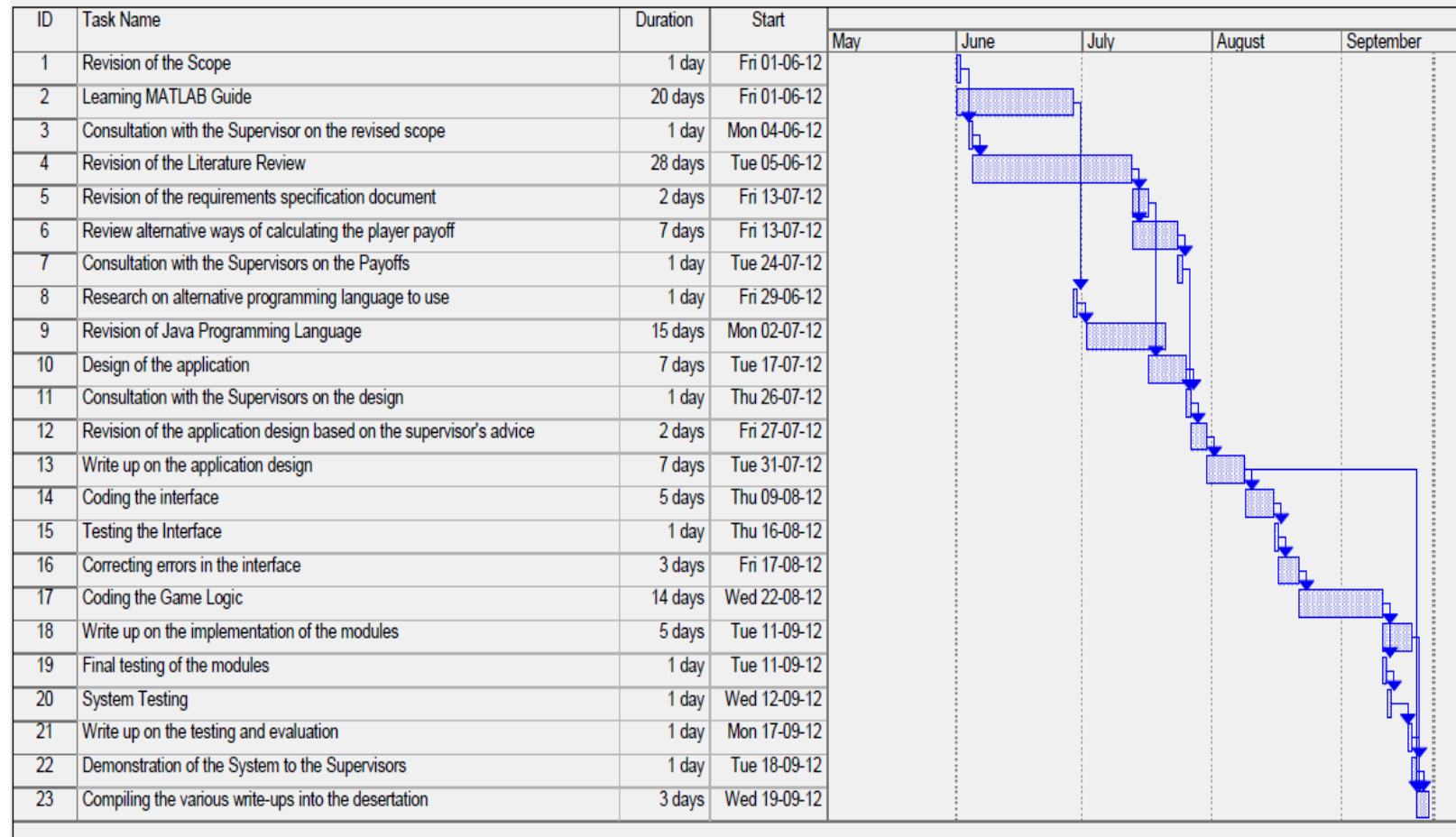


Figure 13 Project Implementation chart



## **7.8 Reflection on the project duration**

The duration that the implementation of the project took could have been reduced in two ways. The first way of reducing the duration is by using a programming language that is widely used. If I had concentrated on using Java from the onset of the project, the time that was spent on learning MATLAB could have been used for other project tasks. The second way I could have reduced on the time was if I concentrated only on DDoS attacks from the onset of the project.

## **7.9 Limitation of the Research**

This research has the following limitations;

1. This research only considered distributed denial of service (DDoS) attacks on DNS servers. There are other attacks that can affect DNS servers. These attacks include SQL injection attacks and malware [85].
2. The research only considers the players' payoffs in terms of the bandwidth only. During DDoS attacks, it is not only the bandwidth that is affected by the attack. The memory and processor is also affected.
3. The research simulation assumed that the DNS implements only one countermeasure against DDoS attacks. In reality DNS server can implement several defence mechanisms.
4. The financial cost of defending DNS servers has not been considered in this research. Defence mechanisms usually have a cost. For example, the provision of more bandwidth has a financial cost bearing.
5. The research has not consider the contribution that security of individual network device devices makes on the security of the DNS

## **7.10 Future enhancements to the research**

The following are the improvements that can be made on this research.

1. Future enhancements should consider more cyber-attacks that affect DNS servers.
2. Future enhancements should incorporate memory usage and processor time in the players' payoff functions.
3. Future research should allow the DNS to use several defence mechanisms at the same time.

4. The cost of defence mechanisms must be incorporated into future enhancements on the research.
5. Future enhancements should consider the contributions that other devices on a network make in protecting the DNS against attacks.

## **CHAPTER 8 – CONCLUSION**

## 8.0 Chapter Overview

This chapter gives a conclusion on the research that was carried out on how game theory can be used to protect critical infrastructure. To be able to do this, distributed denial of service (DDoS) attacks on the DNS were modelled as a Bayesian game between an attacker and a defender. This chapter summarises the objectives of the research, the work that has been done, the findings and recommendation, the limitations of the research and the author's self-reflection on the research.

### 8.1 Recap on the objectives

The aims of this research were:

- 1) To model Distributed Denial of Service (DDoS) attacks on the DNS servers as a Bayesian game between the attacker and the defender.

The simulation aims to show how the DDoS attacks are carried out in reality. The model also shows how the defender forms their belief on which kind of DDoS attack the attacker is likely to launch on a DNS server.

- 2) To determine the Nash Equilibrium for the Bayesian game model for the DDoS attack on DNS server.

The Nash Equilibrium in a game shows quantitatively the best strategies for the players in that game. If any player chooses a strategy that is not in the Nash equilibrium, they will get a poor result from their action.

- 3) To demonstrate how dynamically evolving games can be used to better understand risk management of critical infrastructure (that is, the Domain Name system).

### 8.2 Work Done

In order to achieve the objectives a review of the available literature on the domain name system, distributed denial of service attacks, game theory and risk management was made. The findings obtained from this review were used in coming up with an application in Java to simulate how DDoS attacks on the DNS can be modelled as a Bayesian game and how the best defence mechanism can be chosen using the concept of the Nash Equilibrium.

### **8.3 Summary of the findings**

The research has demonstrated that when protecting a DNS server against DDoS attacks, there is no single defence mechanism that would work best in all the situations. a strategy that is effective for defending against DDoS with very large amounts of incoming traffic might be too restrictive for defending against DDoS attacks with low amounts of incoming traffic.

### **8.4 Recommendation for Defending the DNS against DDoS attacks**

The DNS should use a combination of defence mechanisms to protect against DDoS attacks. In this research, it has been shown that, over-provision of network bandwidth could be used to prevent low-impact DDoS attacks while random dropping of packets can be used to protect against high-impact DDoS attacks. By providing more bandwidth when there is a mild attack, all the legitimate queries gets to be processed and the DNS server continues to function properly. By randomly dropping incoming packets during a high-impact DDoS attack, it ensures that the DNS does not go down due to heavy incoming traffic. An automated system should be implemented that monitors the incoming traffic to the DNS server. The system must be able to switch between the over-provision of bandwidth and randomly dropping of packets as the incoming traffic changes.

### **8.5 Limitation of the research**

This research has the following limitations;

1. This research only considered distributed denial of service (DDoS) attacks on DNS servers. There are other attacks that can affect DNS servers. These attacks include SQL injection attacks and malware.
2. The research only considers the players' payoffs in terms of the bandwidth only. During DDoS attacks, it is not only the bandwidth that is affected by the attack. The memory and processor is also affected.
3. The research simulation assumed that the DNS implements only one countermeasure against DDoS attacks. In reality DNS server can implement several defence mechanisms.

4. The financial cost of defending DNS servers has not been considered in this research. Defence mechanisms usually have a cost. For example, the provision of more bandwidth has a financial cost bearing.
5. The research has not consider the contribution that security of individual network device devices makes on the security of the DNS

## **8.6 Future enhancements to the research**

The following are the improvements that can be made on this research.

1. Future enhancements should consider more cyber-attacks that affect DNS servers.
2. Future enhancements should incorporate memory usage and processor time in the players' payoff functions.
3. Future research should allow the DNS to use several defence mechanisms at the same time.
4. The cost of defence mechanisms must be incorporated into future enhancements on the research.
5. Future enhancements should consider the contributions that other devices on a network make in protecting the DNS against attacks.

## **8.7 Self-Reflection**

I enjoyed doing this research. I would advise future students interested in cyber security to explore game theory and how it can be used in cyber security. This research has helped me understand how attackers exploit weaknesses in an infrastructure in a network to weaken the entire network. I have learnt the challenges that risk managers face when selecting the defence mechanisms to use in protecting critical infrastructure against intelligent attackers. Usually there is a trade-off between security measures used and the quality of service offered to the users. I have been able to understand the usefulness of game theory in cyber security and in risk management in general.

## REFERENCES

## REFERENCES

1. Singh, A., *Improving Information Security Risk Management*, in *FACULTY OF THE GRADUATE SCHOOL* 2009, UNIVERSITY OF MINNESOTA. p. 108.
2. World Bank. *Internet Usage as Percentage of population*. 2012; Available from: <http://data.worldbank.org/indicator/IT.NET.USER.P2>.
3. Australian Crime Commission, *The Cyber White Paper: Connecting with Confidence*, November 2011.
4. British-North American Committee, *CYBER ATTACK: A RISK MANAGEMENT PRIMER FOR CEOs AND DIRECTORS*, 2007, British-North American Committee.
5. Federal Bureau of Investigation. *High-Tech Heist 2,100 ATMs Worldwide hit at once* 2009 [accessed on 26/07/2012]; Available from: [http://www.fbi.gov/news/stories/2009/november/atm\\_111609](http://www.fbi.gov/news/stories/2009/november/atm_111609).
6. Australian Bureau of Statistics, *Australian Social Trends June 2011*, A.B.o. Statistics, Editor 2011.
7. United Kingdom Cabinet Office, *Cyber Security Strategy*, Cabinet Office, Editor November 2011.
8. Carr, J., *Inside Cyber Warfare: Mapping the Cyber Underworld*. O'REILLY. 2010, 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media Inc. 234.
9. Ponemon Institute, *Second Annual Cost of Cyber Crime Study - Benchmark study of U.S. Companies*, 2011.
10. Liu, C. and P. Albitz, *DNS and BIND*, 2006, O'Reilly Media: California, USA.
11. Alberts, C.J., et al., *Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) Framework, Version 1.0 (CMU/SEI-99-TR-017)*. in *Networked Systems Survivability Program* June 1999.
12. Cox, L.A., *Game Theory and Risk Analysis*. Risk Analysis, 2009. **29**(8): p. 1062-1068.
13. National Research Council and Committee to Review the Department of Homeland Security's Approach to Risk Analysis, *Review of the Department of Homeland Security's Approach to Risk Analysis*, D.o.H. Security, Editor 2010, National Academies Press: Washington DC.
14. ICANN, *Request for Proposals for DNS Risk Management Framework Consultant*, 2012, Internet Corporation for Assigned Names and Numbers.
15. Jorgensen, P.C., *Software Testing: A Craftsman's Approach*. Third Edition ed. 2008, New York, USA: Auerbach Publications Taylor and Francis Group.
16. Myerson, R.B., *Game Theory: Analysis of Conflict*. 1991: Harvard University Press.
17. Morgenstern, O. and J.V. Neumann, *Theory of Games and Economic Behaviour*. 60th-anniversary Edition ed. 2004, UK: Princeton University Press.
18. Hargreaves-Heap, S.P. and Y. Varoufakis, *Game Theory: A critica introduction*. Second Edition ed. 2004, USA and Canada: Taylor and Francis e-Library.
19. Polak, B. *Game Theory:Lecture 1 Transcript ECON 159*. 2012 [accessed on 28/08/2012]; Available from: <http://oyc.yale.edu/economics/econ-159/lecture-1>.
20. Gibbons, R., *A Primer in Game theory*. 1992: Prentice Hall.
21. Osborne, M.J., *An introduction to Game theory*. 2004, New York, USA: Oxford University Press.
22. Peters, H., *Game Theory: A Multi-Leveled Approach*. 2008, Berlin, Germany: Springer. 11,59-67.



23. Dixit, A. and S. Skeath, *Games of strategy*. Second Edition ed. 2004, New York, USA: W.W. Norton and company, Inc. 263-295.
24. Chatzis, N. *Motivation for Behaviour-Based Security: A taxonomy of DNS-Related Threats*. in *International Conference on Emerging Information, Systems and Technologies* 2007.
25. Rastegari, S., M. Saripan, and M. Rasid, *Detection of Denial of Service Attacks against Domain Name System Using Neural Networks*. *International Journal of Computer Science Issues*, 2009. **6**(1): p. 23-27.
26. Microsoft Corporation. **Microsoft Security Bulletin MS12-017 –Important Vulnerability in DNS Server Could Allow Denial of Service (2647170)**. 2012 [cited 2012 11/08]; Available from: <http://support.microsoft.com/kb/2647170>.
27. Peng, T., C. Leckie, and K. Ramamohanarao, *Survey of network-based defense mechanisms countering the DoS and DDoS problems*. *ACM Computing Surveys (CSUR)*, 2007. **39**(1).
28. Thing, V.L., M. Sloman, and N. Dulay, *Enhanced TCP SYN Attack Detection*, in *IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM)* November 2007: Toulouse, France.
29. Ahlawat, N. and C. Sharma, *Classification and Prevention of Distributed Denial of Service attacks*. *International Journal of Advanced Engineering Science and Technologies (IJAEEST)*, 2011. **3**(1): p. 52-60.
30. Mansfield-Devine, S., *DDoS: threats and mitigation*. *Network Security*, 2011. **2011**(12): p. 5-12.
31. Brownlee, N., K.C. Claffy, and E. Nemeth. *DNS measurements at a root server*. in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*. 2001.
32. Kambourakis, G., et al. *A Fair Solution to DNS Amplification Attacks*. in *Digital Forensics and Incident Analysis, 2007. WDFIA 2007. Second International Workshop on*. 2007.
33. Changhua, S., L. Bin, and S. Lei. *Efficient and Low-Cost Hardware Defense Against DNS Amplification Attacks*. in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. 2008.
34. Kambourakis, G., et al., ***Detecting DNS Amplification Attacks***. *Critical Information Infrastructures Security*, 2008. **5141**: p. 185-196.
35. Rastegari, S., I.M. Sarapan, and M.F.A. Rasid. *Detection of Denial of Service Attacks against Domain Name System Using Machine Learning Classifiers*. in *World Congress on Engineering*. 2010. London, United Kingdom.
36. Liu, S., *Surviving Distributed Denial-of-Service Attacks*. *IT Professional*, 2009. **11**(5): p. 51-53.
37. Hal, B. and C. Bill. *Tracing Anonymous Packets to Their Approximate Source*. in *14th System Administration Conference, Usenix LISA 2000*. 2000.
38. Wang, Y., et al., *Tracking Anomalous Behaviors of Name Servers by Mining DNS Traffic*, in *Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops*, B. Di Martino, et al., Editors. 2006, Springer Berlin / Heidelberg: Berlin. p. 351-357.
39. Kotenko, I., *Multi-agent Modelling and Simulation of Cyber-Attacks and Cyber-Defense for Homeland Security*. *4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 2007: p. 614-619.

40. Kottenko, I. and A. Ulanov, *Agent-based simulation of DDOS attacks and defense mechanisms*. Journal of Computing, 2005. 4(2).
41. Bu-Sung, L., et al., *Availability and effectiveness of root DNS servers: A long term study*. 2010 IEEE/IFIP Network Operations and Management Symposium - NOMS 2010, 2010: p. 862-865.
42. Arends, R., et al., *DNS Security Introduction and Requirements* in RFC 4033 March 2005, Internet Engineering Task Force.
43. Arends, R., et al., *Protocol Modifications for the DNS Security Extensions*, in RFC 4035 March 2005, Internet Engineering Task force.
44. Arends, R., et al., *Resource Records for the DNS Security Extensions*, in RFC 4034 March, 2005, Internet Engineering Task Force.
45. Deshpande, T., et al. *Formal Analysis of the DNS Bandwidth Amplification Attack and Its Countermeasures Using Probabilistic Model Checking*. in *High-Assurance Systems Engineering (HASE), 2011 IEEE 13th International Symposium on*. 2011.
46. ICANN SSAC, *SSAC Advisory SAC008 DNS Distributed Denial of Service (DDoS) Attacks*, 2006, ICANN Security and Stability Advisory Committee.
47. Wu, Q., et al., *On Modeling and Simulation of Game Theory-Based Defense Mechanisms against DoS and DDoS Attacks*. SpringSim, 2010.
48. Guanhua, Y., C. Songqing, and S. Eidenbenz. *Dynamic Balancing of Packet Filtering Workloads on Distributed Firewalls*. in *16th International Workshop on Quality of Service, 2008. IWQoS 2008*. 2008.
49. Walfish, M., et al., *DDoS Defence by Offense*. ACM Transactions on Computer Systems, March 2010. 28(1): p. 3-54.
50. Khanna, S., et al., *Adaptive Selective Verification*, in *The 27th Conference on Computer Communications* April 2008. p. 529-537.
51. Oxford Dictionaries, *Cybersecurity Definition*, April 2010, Oxford University Press.
52. Garvey, P.R., *Analytical methods for risk management: a systems engineering perspective*. 2009: Chapman & Hall /CRC Press.
53. Hopkin, P., *Fundamentals of Risk Management: understanding, evaluating and implementing effective risk management* 2010, Britain: Kogan Page Limited.
54. Singh, A. and D. Lilja, *Criteria and Methodology for GRC Platform Selection*. ISACA, 2010. 1: p. 32-37.
55. International Standards Organisation, *Information Technology - Security Techniques - Information Security Risk Management*, in ISO/IEC 27005:2008.
56. Sajjan, S., R. Sankardas, and D. Dipankar, *Game theory for cyber security*, in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research* 2010: Oak Ridge, Tennessee. p. 1-4.
57. Young, C.S., *Metrics and Methods for Security Risk Management*, ed. P. Chester and G. Chalson. 2010, USA: Elsevier. 272.
58. Department of Homeland Security, *Strategy to Enhance International Supply Chain Security*, D.o.H. Security, Editor 2007.
59. European Central Bank, *Recent Advances in Modelling Systemic Risk using Network Analysis*, January 2010: Munich, Germany.
60. Wheeler, E., *Security Risk Management : Building an Information Security Risk Management Program from the Ground Up*, ed. A. Ward. 2011, USA: Elsevier Inc.

61. Yao, C., L. Zhengjiang, and W. Zhaolin. *Improvement of Fault Tree Analysis in Formal Safety Assessment Using Binary Decision Diagram*. in *Information Science and Engineering (ICISE)*, 2009 1st International Conference on. 2009.
62. Bier, V.M., L.A. Cox, and M.N. Azaiez, *Why Both Game Theory and Reliability Theory are Important in Defending Infrastructure Against Intelligent Attacks*, in *Game Theoretic Risk Analysis of Security Threats*, V.M. Bier and M.N. Azaiez, Editors. 2009. p. 1-11.
63. Assmuth, T. and M. Hilden, *The significance of information frameworks in integrated risk assessment and management*. *Environmental Science and Policy*, 2008: p. 71-86.
64. Taquechel, E., *Layered defense: modeling terrorist transfer threat networks and optimizing network risk reduction*. *IEEE Network*, November/December 2010. **24**(6): p. 30-35.
65. Cox, L.A.J., *Some Limitations of "Risk = Threat \* Vulnerability \* Consequence" for Risk Analysis of Terrorist Attacks*. *Risk analysis*, 2008. **28**(6): p. 1749 -1761.
66. Maille, P., P. Reichl, and T. Bruno, *Of Threats and Costs: A Game-Theoretic Approach to Security Risk Management*, in *Performance Models and Risk Management in communication systems*, N. Gulpinar, P. Harrison, and B. Rustem, Editors. 2010, Springer.
67. Merrick, J.R.W., et al., *Games and Decisions in Reliability and Risk*. *Decision Analysis*, June 2012. **9**(2): p. 81-85.
68. Golany, B., et al., *Nature plays with dice - Terrorists do not: Allocating resources to counter strategic vs probabilistic risks*. *European Journal of Operational Research*, 2007.
69. Krzysztof, R.A., *A Primer on Strategic Games*, in *Lectures in Game Theory for Computer Scientist*, R.A. Krzysztof and E. Gradel, Editors. 2011, Cambridge University Press: New York, USA. p. 1-37.
70. Sevillano, J.C., I.D. Rios, and J. Rios, *Adversarial Risk Analysis: The Somali Piracy Case*. *Decision Analysis*, June 2012. **9**(2): p. 86-95.
71. Lye, K. and J. Wing, *Game strategies in network security*. *International Journal of International Security*, 2005. **4**(2): p. 71-86.
72. Roy, S., et al. *A Survey of Game Theory as Applied to Network Security*. in *System Sciences (HICSS)*, 2010 43rd Hawaii International Conference on. 2010.
73. Poole, D.L. and A.K. Mackworth, *Artificial Intelligence: Foundations of Computational agents*. 2010, New York, USA: Cambridge University Press.
74. Xu, J. and L. Wooyong, *Sustaining availability of Web Services under Distributed Denial Of Service Attacks*. *IEEE Transactions on Computers*, 2003. **52**(2): p. 195-208.
75. Sanquist, T.F., H. Mahy, and F. Morris, *An exploratory Risk Perception Study of Attitudes Toward Homeland Security Systems*. *Risk Analysis*, 2008. **28**(4): p. 1125-1133.
76. Sommerville, I., *Software Engineering*. 8th Edition ed. 2007: Addison-Wesley Publishers Limited.
77. Lindholm, T., et al., *The Java<sup>TM</sup> Virtual Machine Specification Java SE 7 Edition*. 2011, California, USA: Oracle.
78. Liu, Y., C. Comaniciu, and H. Man, *A Bayesian game approach for intrusion detection in wireless ad hoc networks*, in *Proceeding from the 2006 workshop on Game theory for communications and networks2006*, ACM: Pisa, Italy. p. 4.
79. Free Software Foundation. *GNU GENERAL PUBLIC LICENSE Version 2*. 1991 [accessed on 10/09/2012]; Available from: <http://www.gnu.org/licenses/gpl-2.0.txt>.

80. Watt, D. and W. Findlay, *Programming Languages Design Concepts*. 2004, West Sussex, England: John Wiley and Sons Limited.
81. Laramee, R.S., *Bob's Concise Coding Conventions*, 2009, Swansea University.
82. gigenetcloud.com. *History of DDoS - Famous Attacks*. 2012 [accessed on 20/08/2012]; Available from:  
[http://www.gigenetcloud.com/history\\_of\\_ddos.html#Accidental\\_DD0S\\_attack\\_on\\_Google](http://www.gigenetcloud.com/history_of_ddos.html#Accidental_DD0S_attack_on_Google).
83. Trostle, J., B. Van Besien, and A. Pujari, *Protecting Against DNS Cache Poisoning Attacks*. *6th IEEE Workshop on Secure Network Protocols (NPSec)*, 2010: p. 25 - 30.
84. Yu, X., X. Chen, and F. Xu, *Recovering and Protecting against DNS Cache Poisoning Attacks*. 2011 International Conference of Information Technology, Computer Engineering and Management Sciences, 2011. **2**: p. 120 - 123.
85. Cisco Systems Inc, *A Beginner's Guide to Network Security*, 2011, Cisco Systems Inc.

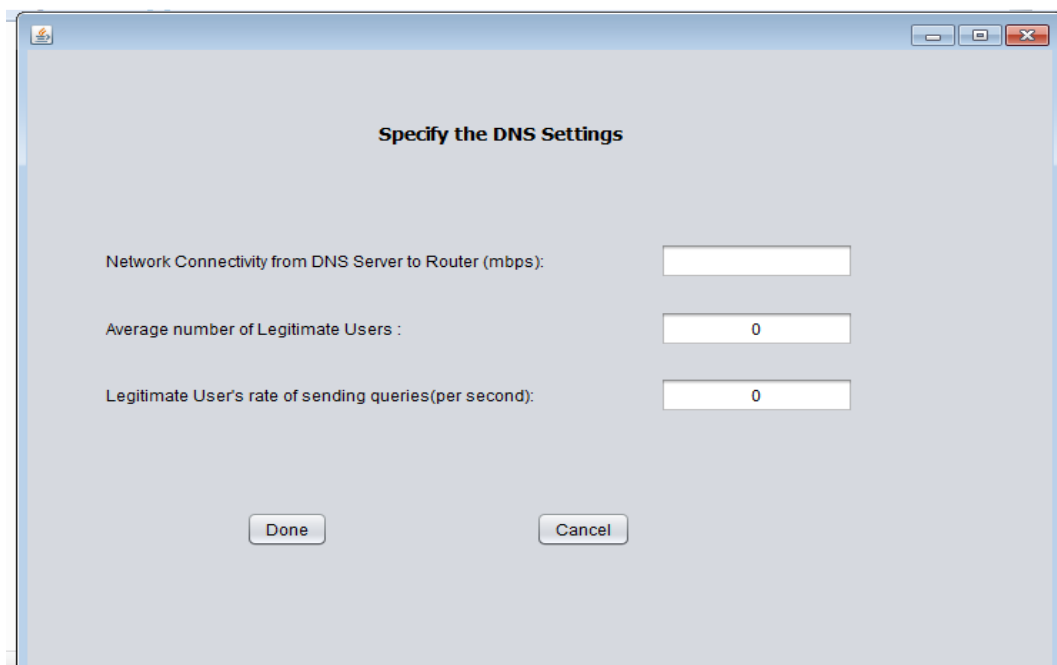
# **APPENDIX 1 – USER MANUAL**

## Overview

This appendix describes how the application for modelling Distributed denial of service attacks on DNS servers as Bayesian games will be operated by the users.

## Running the application

To start the application, double-click on the ProjectSimulation.Jar icon on the compact disk or wherever it has been placed. When this is done, the form in figure 14 will appear.



**Specify the DNS Settings**

Network Connectivity from DNS Server to Router (mbps):

Average number of Legitimate Users :

Legitimate User's rate of sending queries(per second):

Figure 14 Legitimate Users Settings

On this form, the user must enter the network bandwidth, the number of legitimate users and the number of queries a legitimate user can send per second. After entering these values, the user must the “Done” button. Once this is done, the form in figure 15 appears.

Figure 15 Attacker Settings

On this form, the user enters the number of Bots used in an attack, the rate of sending queries for each bot and the kind of attack to be launched. After this the user presses the “Simulate” button. The results will now be shown on the form as illustrated in Figure 16. To run the simulation the second time with the same network settings, the user just need to change the number of bots, the sending rate or attack type and click Simulate. To run a simulation with different settings, the user must click on back and change the settings when the network settings form appears.

Attacker's Settings

Number Of Bots: 1500000

Bot's sending rate(/ second): 1

Type of Attack

☐ Direct Attack

☒ Amplification Attack

Simulate

Simulation Results

Direct Attack Results

Defender/Attacker	Attack: Bots = 1500000, Rate = 1	No Attack
Firewall Drop Rate F(r):	94.21 , -12.19	98.24 , 0.0
provision of more Bandwidth:	0.66 , 100.38	100.00 , 0.0
Limit Requests From Hosts:	0.66 , 103.80	100.00 , 0.0

Defender's Belief in this game... 0.33

Amplification Results

Defender/Attacker	Attack: Bots = 1500000, Rate = 1	No Attack
Firewall Drop Rate F(r):	94.21 , -0.20	98.24 , 0.0
Provision of more Bandwidth:	0.01 , 101.69	100.00 , 0.0
Limit Requests From Hosts:	0.01 , 105.10	100.00 , 0.0

Defender's Belief in this game ... 0.67

Back

Bayesian Nash Equilibrium

Defender's Payoff: 94.21

Defender's Strategy for Direct Attack:

Drop Rate

Defender Strategy for Amplification Attack:

Drop Rate

Attacker's Payoff: -0.2

Attacker Strategy

attack

Game History

1. amplification attack  
Best Direct Attack Strategy: Drop Rate  
Best Amplification Attack Strategy: Drop Rate

**Figure 16 Game Results illustration**