ATTACKER TRACING USING PACKET MARKING

By

ASC SHAHEEN IQBAL

ASC SULMAN NAEEM

ASC M SAMI IQBAL

PC AHMED NAWAZ

Submitted to the Faculty of Computer Software EngineeringNational University of Sciences and Technology, Islamabad in partial fulfillmentFor the requirements of a B.E. Degree in Computer Software Engineering

ABSTRACT

With the advent of the modern information age, cyber warfare or cyber war is characterized as 5th generation warfare where adversaries fight for the secret sensitive information and want to make the services provided unreachable. Many techniques have been used since the ancient times to keep data available and is top priority in government organizations therefore secure media is very important.

Modern day security concepts stand on three primary pillars known as Confidentiality, Integrity and Availability a.k.a. The CIA Triad. The product in development (Attacker Tracing Using Packet Marking) will focus mainly on the availability of data. Packet marking will be used to achieve availability. An Algorithm "Flexible Deterministic Packet Marking" is used to mark the packets.

Attacker Tracing Using Packet Marking will be responsible to mark packets deterministically and ensure that data is available all the time. Interactive interface will be provided to ensure ease of usage to system users. Software will be compatible with the already hardware installed at the government organization.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis – Attacker Tracing Using Packet Marking carried out by Shaheen Iqbal, Sulman Naeem, M Sami Iqbal and Ahmed Nawaz in supervision of Asst. Prof M.M.Waseem Iqbal for partial fulfillment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Asst. Prof M.M. Waseem Iqbal

CSE DEPARTMENT

MCS

DATED:

DECLARATION

No portion of the work presented in this dissertation has been submitted inSupport of another award or qualification either at this institution or elsewhere

DEDICATION

In the name of Allah, the Most Merciful, the Most BeneficentTo our parents, without whose unflinching support and cooperation,a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

We would like to thank Allah Almighty for His incessant blessings which have been bestowed upon us. Whatever we have achieved, we owe it to Him, in totality. We are also thankful to our families for their continuous moral support which makes us what we are.

We are extremely grateful to our project supervisor Asst. Prof M.M. Waseem Iqbal from MCS who in addition to providing valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project.

We are highly thankful to all of our teachers and staff of MCS who supported and guided us throughout our course work. Their knowledge, guidance and training enabled us to carry out this whole work.

Finally we are grateful to the faculty of Computer Software Department of the Military College of Signals, NUST.

In the end we would like to acknowledge the support provided by all our friends, colleagues and a long list of well-wishers whose prayers and faith in us propelled us towards our goal.

TABLE OF CONTENTS

CHAPTER:1

INTRODUCTION

1 INTRODUCTION

1.1 Purpose

The purpose of the Software Requirements Specification (SRS) is to give the user a clear and precise description of the functionality of this project which would be built for NESCOM.

This document is aimed to eliminate ambiguities and misunderstandings that may exist. For the user, the SRS will explain all functions that the software should perform. For the developer, it will be a reference point during software design, implementation and maintenance.

This document encompasses the requirements for version-1 of this project. The main focus of the project is to provide a network based implementation for tracing origin of DDOS attacks.

1.2 Problem Statement

A DDoS attack is an availability attack, for it is characterized by an explicit attempt from an attacker to prevent legitimate users from using the desired resources. Many institutes have reported DDoS attacks cause a huge amount of financial loss every year. DDoS attacks can cause the inability of information infrastructure to function, total stoppage or severe impairment of activity; therefore it can result in threatening for life etc. The difficulty in solving the problem is first the DDoS tools are easy to get and use, thus even an inexperienced hacker can launch an attack effortlessly. The second reason is that it is difficult to separate unambiguously the attack traffic from legitimate traffic, and then filter out the attack traffic. Accurately separating and filtering attack traffic can provide maximum possible resources to legitimate users of the information infrastructure.

1.3 Document Conventions

The document is based on a standard IEEE format. It follows a convention that involves bold-facing of headings, the use of indentions and numbering for major parts and subparts. This SRS is divided into sections detailing an overall description, the external interface requirements, system features, and other nonfunctional requirements.

The following pairs of words are used interchangeably in the SRS.

“Application” and “Software”

“Product” and “Project”

“Feature” and “Function”

1.4 Intended Audience and Reading Suggestions

1.4.1 Intended Audience

The intended readers of the SRS are the NESCOM employees who will have the system implemented as well as development team. This document serves as an agreement between both parties (Development team and the NESCOM authorities) regarding the product to be developed.

- **Project Supervisor (Asst. Prof. M.M. WaseemIqbal)**

This document will assist in supervision and guiding the team. Further, document will act as a reference to ensure completion of all requirements and proper implementation.

- **Developers and Testers (Project group, NESCOM Team)**

This document provides the guideline for developers to code during implementation phase and testers to create test cases during testing phase.

- **Project Evaluation Team (MCS NUST, NESCOM Team)**

It will help the evaluation team at NESCOM to evaluate the progress of FYP project. The document will provide the evaluators with the scope, requirements and details of the software to be developed. It will also be used as basis for the evaluation of the implementation of the project.

1.4.2 Reading suggestions

It would be suggested to the users to go through the requirement section thoroughly.

For the developers it is suggested that they read and understand the product scope, overall description and system features thoroughly.

Testers should go through the operating environment, constraints, and the non-functional requirements before developing the test scenarios for the system.

1.5 Scope

The scope of this project is to trace back the attacker in the network who is threating the availability. It is developed for private networks within the organizations. We are using java simulations to show the working of the project. This project will deal with all type of DDOS attacks. Packet marking technique is used to trace back the attacker.

CHAPTER:2

OVERALL DISCRIPTION

2 OVERALL DESCRIPTION

2.1 Product Perspective

The product is java simulation to trace the attacker. This project will be built for sensitive organizations providing a mechanism to trace the original source of the attacker. In market products to detect and prevent availability attacks are available but no such products are available in market for this purpose. It is built for organizations to insure the availability of data in the organization. With this product they can not only detect but also trace back the attacker.

2.2 Product Function:

The main features of the Attacker Tracing using Packet Marking are highlighted below:

- System will enable user to enter the data and send it on network to a specific IP address.
- System will enable attacker to send malicious traffic on network.
- System will mark packets using FDPM(Flexible Deterministic Packet Marking) technique.
- This feature checks the purity of packet and if the packet is found to be defected, it traces back attacker's IP address.
- Creating a java simulation

2.3 User Classes and Characteristics

Following are user classes and their brief description.

2.3.1 Operating environment

Required operating environment for the application is listed below.

2.3.2 Hardware Requirements

The product is a java simulation simulated on a pc.

2.3.3 Software Requirements

- Windows: 7,8,8.1,10
- Java simulator
- Eclips Java IDE
- Packet Analyzer

2.3.4 Design and Implementation Constraints

Constraints of the product are given below:

- The system will handle multiple requests at a time.
- It can be used by multiple users at a time.
- The computing of classification algorithms is dependent on the performance of simulation.

2.3.5 User Documentation

A user manual will be provided to the users in which separate instructions will be given according to the particular user i.e. Regular user and the admin, developers and testers. It will include the details of the system's working. Help documents will also be a part of the system.

The project report will also be available for the users which will highlight the system features, working and procedures.

2.3.6 Assumptions and Dependencies

- User must know about the Interface for the better performance of the product.
- Limitations of the product must be kept in mind by the user.

3 EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interfaces

The homepage will contain buttons for all the important functionalities provided by the software. The simulation will run that shows the routers and pcs and how they are connected to each other Simulation will be explanatory regarding the options and functionality provided by the software.

3.2 Hardware Interfaces

- 1 PC will be required as it is based on simulation.

3.3 Software Interfaces

- This project will require Windows operating systems
- Java based Simulation will be used

4 System Features

4.1 System Feature 1:

System will enable user to enter the data and send it on network to a specific IP address.

4.2 System Feature 2:

System will enable attacker to send malicious traffic on network.

4.3 System Feature 3:

System will mark packets using FDPM(Flexible Deterministic Packet Marking) technique.

4.1 System Feature 4:

This feature checks the purity of packet and if the packet is found to be defected, it traces back attacker's IP address.

4.2 System Feature 5:

The simulation will be made and pcs and routers will be attached to it using java

5 OTHER NON FUNCTIONAL REQUIREMENTS:

5.1 Usability

The user interface will be easy to understand and navigate for both the Server and Client.

5.2 Accuracy

To ensure reliability and correctness, there will be zero tolerance for errors in the algorithm that computation results.

5.3 Availability

The application will be available from boot to shutdown, provided server is in working state and the internet is available and configured properly.

5.4 Flexibility

The design and architecture of the application will be flexible so that in later stages, more features can easily be added based on user feedback.

5.5 Data Integrity

If the application crashes during addition, deletion or editing there will be no changes.

5.6 Scalability

The application is expected to handle multiple user at a time. Multiple instances of the application could be opened on a different clients connected to server at a time.

CHAPTER:3

DESIGN AND DEVELOPMENT

3 Introduction

3.1 Purpose of the document:

This software design specification (SDS) document describes the architecture and system design of Attacker Tracing Using Packet Marking. It mostly contains different design diagrams and their explanation. The document is intended to inform stakeholders, developers and support team at organization of the details of the design and the design process. This document will help the developer(s) in implementation and maintenance of the Software.

3.2 Scope of the Development Project:

The scope of this project is to trace back the attacker in the network who is threating the availability. It is developed for private networks within the organizations. We are using java simulations to show the working of the project. This project will deal with all type of DDOS attacks. Packet marking technique is used to trace back the attacker.

3.3 Overview of the document

This document shows the design and working of Attacker Tracing Using Packet Marking. It starts from higher level details for a non-technical reader to understand just by seeing the diagrams to the lower level details that aid the developer to code and understand other technical details of the application.

In Section 2, the **System Architecture Description** gives a detailed overview of the application. Section 2.1 Overview of Modules/Components shows the main component of the application and their inter-relationships. Section 2.2 Structure and Relationships shows the higher level details system working by the means of System Block, Activity, State Transition, and Use Case diagrams. Lower level details are described using the Class, Sequence diagrams and Structure Chart. Section 2.3 describes how the application is designed to curb the tendency of User Interface Issues and problems during User Interaction.

In Section 3, **Detailed Description of Component** is given to show the working of modules with low level details. It shows the purpose, function, subordinates, dependencies, interfaces, resources, processing and data of the components and their relationships with each other.

Section 4 shows the **Reuse and Relationship to other Products** i.e.; information about work done in the same project before and any reuse of the same work. The section also provides a key to reuse this system for further upgrades.

3 System Architecture Description

This section provides detailed system architecture of Attacker Tracing Using Packet Marking. Overview of system modules, their structure and relationships are described in this section. User interfaces and related issues are also discussed.

3.1 Overview of Modules/Components

This project has following required modules. Here we give a brief overview of all these modules. Detailed descriptions of these modules are presented in section 3.

Normal Traffic Generation

One of the basic functionalities of the system is to select a file from the destination ad send them to the desired network .

Marking and Encoding:

The second basic functionalities of the system is to marks all the incoming packets at engress router and using flexible deterministic packet marking (FDPM) and encode the marking information in the identification field of ip header.

Malicious Traffic Generation:

Attacker will generate a large amount DDoS traffic and send it to the desired server.

Ip Traceback:

The victim will detect the malicious traffic when the traffic will exceed the threshold limit and start ip traceback procedure and trace the ip of the attackers' ingress router.

3.3 Structure and Relationships

This section covers the technical description of Attacker tracing using Packet marking. It shows relationships between different components and how system modules are connected. This section also covers working with respect to different point-of-views. This also covers its higher and lower levels details, user interfaces, and system architecture and design pattern

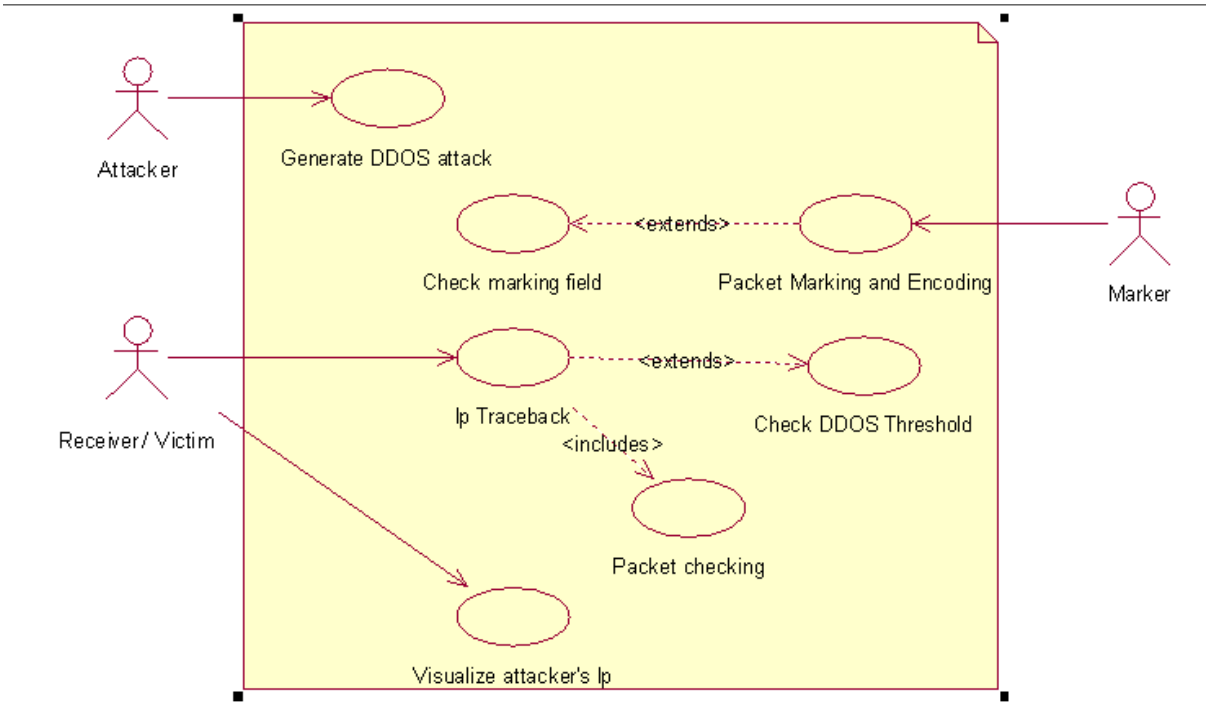
3.3.1 System Block Diagram

This diagram shows the higher-level description of the application. It shows all the modules of the system and their associations and flow of data between modules.

3.4 UML Diagrams:

3.4.1 User View (Use case diagram)

Following diagram shows course of events that take place when an actor (user and other allowed interactions) interact with the system.



3.4.2 Use Case Descriptions

Generate Attack traffic:

Use Case Requirement: Attacker will generate DDoS attack traffic and send it to desired port.
Use Case Paths • Normal: – Attacker will generate attack – Attacker will send the traffic to desired server. • Exceptional: – Attacker will not able to send attack to desired server

Preconditions
• Availability of DDoS tool .

Packet Marking and Encoding

Use Case Requirement: Ingress router will mark all packet deterministically and encode routers' ip address in identification field of ip address.
Use Case Paths • Normal: – Ingress router will mark all packets. – Mark packets with router ip address. – Any router other than ingress will not mark packets. • Exceptional: – Router is not able to mark packets.
Preconditions • Packets should be captured at routers.

Ip Traceback

Use Case Requirement: System should trace detected malicious packets using marking information.
Use Case Paths • Normal: – System detects malicious packets by checking DDoS threshold limit. – System will trace the attackers' ingress routers' ip address. • Exceptional: -- System may be able to detect malicious traffic.
Preconditions Packet should be capture marked and detected.

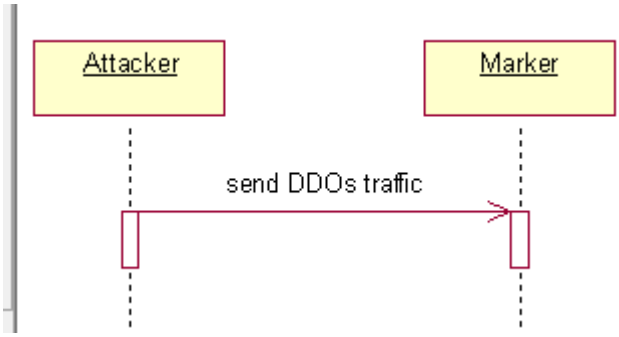
Visualization

Use Case Requirement: System must show the attackers' ingress routers' ip address to victim .
Use Case Paths • Normal: – System will show the attackers' ingress routers' ip address to the victim. • Exceptional: – System may not be able to show the trace back ip address to user.
Preconditions • Attackers' ingress routers' ip address should be obtained.

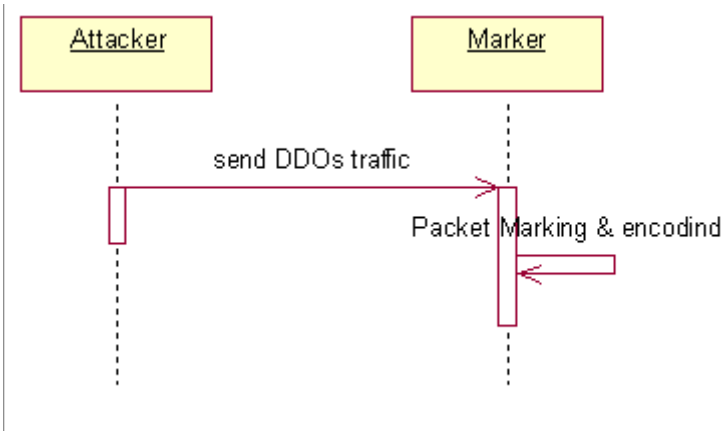
3.4.3 Sequence Diagrams

Following sequence diagrams show the sequence of activities performed in application.

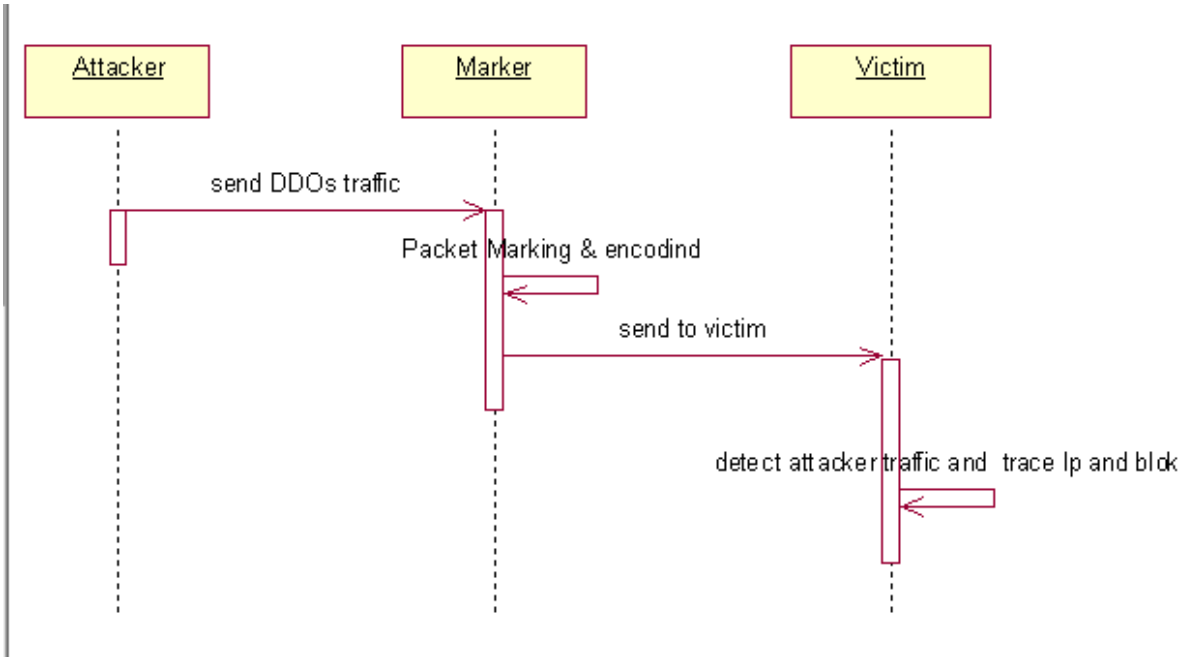
3.4.3.1 *Initialize Attack:*



3.4.3.2 *Marking and Encoding:*

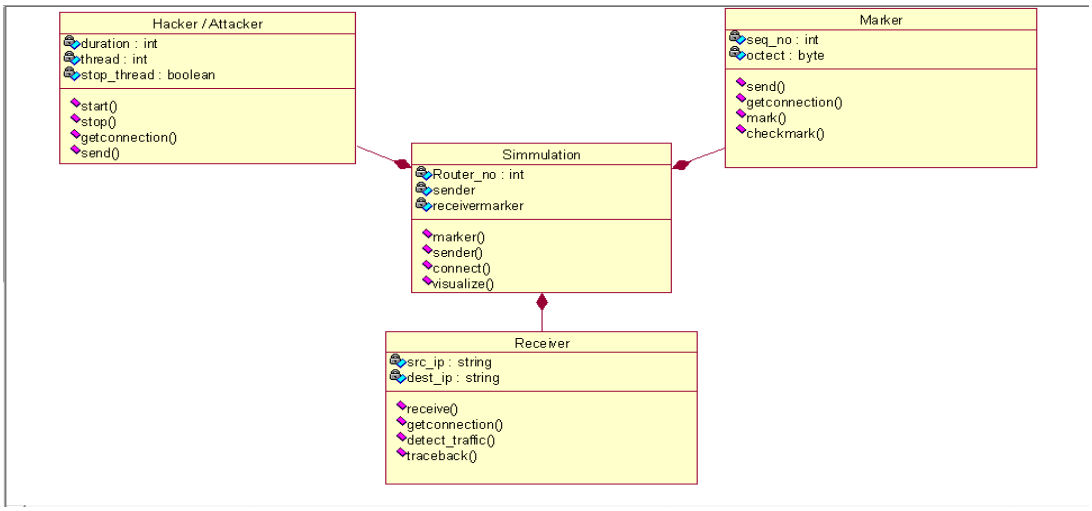


3.4.3.3 *Traceback and Visiualize:*



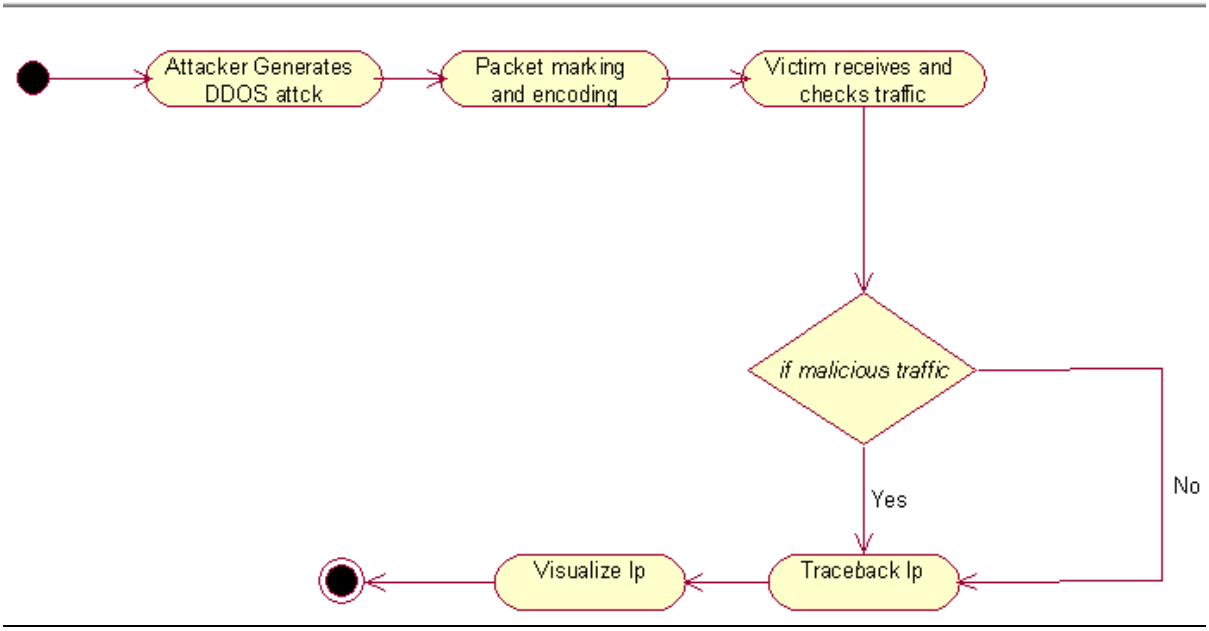
3.4.4 Implementation View (Class Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently with their respective start and end states.



3.4.5 Dynamic View (Activity Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown.



3.5 Detailed Description of Components

This section describes in detail all components of Attacker tracing using Packet Marking.

Attacker

Identification	**Attacker**
Type	Class
Purpose	It will send malicious traffic to desired server
Function	• Generate DDoS traffic. • Send DDoS traffic.
Subordinates	None
Dependencies	• Network connectivity
Interfaces	Java
Resources	• Java simulator • Eclipse IDE
Processing	This components will allow attacker to send DDoS traffic.
Data	

Marker

Identification	**Marker**
Type	Class
Purpose	Mark incoming packets from attacker by giving them unique router id.

Function	• Capture incoming packets • Mark Incoming packets by giving them unique id
Subordinates	Packet Source(Attacker, User etc)
Dependencies	• Packet Marking algorithm
Interfaces	Java
Resources	• Java simulator • Eclipse IDE
Processing	This component will mark incoming packet by flexible deterministic packet marking algorithm and encode it in identification field of ip header.
Data	• Incoming packets from attacker or user.

Victim

Identification	**Victim**
Type	Class/Component
Purpose	It will block malicious traffic and trace back the attacker's ingress router ip and visualize it to victim.
Function	• Block malicious traffic • Trace back the attacker's ingress router ip • Visualize the ip.
Subordinates	None

Dependencies	• Threshold condition • Marking of all incoming packets
Interfaces	Interface to visualize the attacker's path.
Resources	• Java simulator • Eclipse IDE
Processing	This component will block malicious traffic and trace back the attacker by reconstructing the attacker's ingress router ip using reconstructing algorithms and visualize it to victim
Data	Incoming packets from router.

3.6 Reuse and Relationships to other Products

Attacker Tracing using Packet Marking is not an extension of any other project at any level but For Now, it is limited to simulated environment. The primary motive of this application is to provide NESCOM with an application which is not proprietary and source code is available to re-use, maintain and extend in any way desired.

3.7 Design Decisions and Tradeoffs

We have kept the user interface simple and friendly so even any member relating to law enforcement with only basic knowledge of windows and computer can use it effectively.

Since this system is a sensitive application off its category, our focus is to develop it very perfectly.

CHAPTER:4

ANALYSIS AND EVUALUATION

4 Analysis and Evaluation

4.1 INTRODUCTION

This test plan document describes the appropriate strategies, process and methodologies used to plan, execute and manage testing of the "Attacker tracing using packet marking". The test plan will ensure that “Attacker tracing using packet marking” meets the customer requirements at an accredited level.

Manual testing will be followed which includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. Each Unit will be tested separately and then will be integrated with other units; therefore, Unit Testing and Integration testing will be followed. For each unit, Black box Testing is done and for combined units Acceptance Testing is done. The test scope includes the Testing of all functional, application performance and use cases requirements listed in the *requirement document.*

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed.

This document includes the plan, scope, approach and procedure of testing the “Attacker tracing using packet marking”. The pass/fail criteria of the test items are also defined. The document tracks the necessary information required to effectively define the approach to be used in the testing of the product.

4.2 Approach

Acceptance test will be executed based on this acceptance test plan. And after all test cases are executed, a test report will be summarized to show the quality of “Attacker tracing using packet marking”. Following test approaches will be used in test execution:

- **Unit test**. Developers are responsible for unit testing. The implementation of each module and individual component will be verified separately.
- **Integration test**. After the unit test is passed above the defined quality threshold, testers will execute the integration test cases. After all the modules are integrated, it is crucial to test the product as a black-box.
- **Positive and negative testing design technique**. This approach will be combined with unit test and integration test. Test cases are designed in obvious scenarios, which ensure that all functional requirements are satisfied. What’s more, different test cases will also be covered to show how the system reacts with invalid operations.

4.3 Features

Following Features are tested:

- System will enable user to enter the data and send it on network to a specific IP address.
- System will enable attacker to send malicious traffic on network.
- System will mark packets using FDPM(Flexible Deterministic Packet Marking) technique.
- This feature checks the purity of packet and if the packet is found to be defected, it traces back attacker’s IP address.
- The victim requires the fewest packets for the ip reconstruction.

4.4 Item Pass/Fail Criteria

Details of the test cases are specified in section Test Deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

Preconditions are met

- Inputs are carried ou0t as specified.
- The result works as what specified in output => Pass
- The system doesn't work or not the same as output specification => Fail.

4.5 Testing tasks

- Develop test cases.
- Execute tests based on the developed test cases for the software.
- Report defects from the executed test cases if any.
- Provide complete test report.
- Incorporate or manage changes later in the stage of the project development.

4.6 Test Deliverables

- Test cases
- Output from tools

4.7 Responsibilities

All developers of the project are responsible for the completion of all components testing and integration testing tasks.

4.8 Staffing and training needs

Basics knowledge of testing strategies and techniques is needed for the testing of the project. Techniques such as Black Box testing, integration testing should be known to developers.

All the developers will be testing each other's work and will be actively participating in the development and testing of the project simultaneously.

4.9 Risks and contingencies

4.9.1 Schedule Risk:

The project might get behind schedule so in order to complete the project in time we will need to increase the hours/day that the project is being worked on.

4.9.2 Operational Risks:

Operational risks will be eliminated by Scheduling daily meetings and regular deadlines to meet the goals of the project as well as provide proper communication within the group.

4.9.3 Technical risks:

Technical risks will be eliminated by keeping the once defined requirements constant.

4.9.4 Programmatic Risks:

In case of a programmatic risk the scope of the project will be limited in order to stay inside the constraints of the project.

4.10 Test cases:

Test Case Number	01
Test Case Name	Normal traffic generation
Description	Testing whether normal traffic is sent or not.
Testing Technique	Black Box Testing
Preconditions	N/A
Steps	• Generate normal traffic. • Send traffic on network.
Expected output	Traffic is sent on the network.
Actual output	Traffic is sent to the network
Status	Test case passed successfully.

Test Case Number	02
Test Case Name	Malicious traffic generation
Description	Testing whether malicious traffic is sent or not.
Testing Technique	Black Box Testing
Preconditions	N/A
Steps	• Generate malicious traffic. • Send traffic on network.
Expected output	Traffic is sent on the network.
Actual output	Traffic is sent on the network

Status	Test case passed successfully.
Test Case Number	03
Test Case Name	Marking and encoding on routers.
Description	Testing whether traffic is marked and encode on routers or not with the FDPM marking algorithm.
Testing Technique	Black Box Testing
Preconditions	• Traffic should be on the same network. • Router is enabled to mark packets.
Steps	• Router captures traffic. • Router marks traffic based on FDPM algorithm. • Send traffic on network.
Expected output	Traffic is marked by router with the FDPM marking algorithm.
Actual output	Traffic is marked by router with the FDPM marking algorithm.
Status	Test case passed successfully.

Test Case Number	04
Test Case Name	Traffic received by the server.
Description	Testing whether traffic is received by server or not.
Testing Technique	Black Box Testing
Preconditions	• Traffic should be on the same network. • Server is in listening mode.
Steps	• Same port number is open as in sender's machine. • Router marks traffic based on FDPM algorithm. • Send traffic on network.
Expected output	Traffic is received by server.
Actual output	Traffic is received by server.
Status	Test case passed successfully.

Test Case Number	05
Test Case Name	Reconstruction of attacker's ingress routers' ip.
Description	Actual attacker's ingress routers' ip will be reconstructed.
Testing Technique	Black Box Testing
Preconditions	• Traffic is received by the server is malicious.
Steps	• Malicious packets are sent to reconstruction procedure. • Reconstruction procedure will analyze packets and reconstruct the attacker's ingress routers' ip.
Expected output	Attacker's ingress routers' ip is reconstructed.

Actual output	Attacker’s ingress routers' ip is reconstructed.
Status	Test case passed.

Test Case Number	06
Test Case Name	Visualization of attacker’s ingress routers' ip.
Description	Actual attacker’s ingress routers' ip will be visualized.
Testing Technique	Black Box Testing
Preconditions	• Traffic is received by the server is malicious. • Attacker’s ingress routers' ip is be reconstructed.
Steps	• Reconstructed ip is sent to visualization procedure. • visualization procedure will show the attacker’s ingress routers' ip.
Expected output	Attacker’s ingress routers' ip is visualized.
Actual output	Attacker’s ingress routers' ip is visualized.
Status	Test case passed.

5 Future Work

This project can be used as a basis to understand and add features to make it into an even bigger and complex system, which can be modified per an organization’s needs or it could continue growing as a generic software by Cyber Security Organizations and for critical systems protection. Additional features can be added in the future to make it more useful.

6 BIBLIOGRAPHY

1. Practical network support for IP Trace back schemes by Savage, Wetherall, Karlin, Anderson
2. Advanced and Authenticated marking schemes for IP Trace back by Song, Perrig
3. A Precise Termination Condition of the Probabilistic Packet Marking Algorithm by Wong Tsz-Yeung, Wong Man-Hon, Lui Chi-Shing
4. IP Trace back with Deterministic Packet Marking Andrey Belenky and Nirwan Ansari
5. Flexible Deterministic Packet Marking: An IP Traceback System to Find the Real Source of Attacks Yang Xiang, Member, IEEE, Wanlei Zhou, Member, IEEE, and Minyi Guo, Senior Member, IEEE

Appendix A. Glossary

FDPM: Flexible Deterministic Packet Marking

IDE: Integrated Development Environment

OS: Operating System

7 Appendix B: Issues/Limitations

All possible issues have already been mentioned where required in the SRS. Any remaining ones are listed below:

1. The group shall try to match the features and NFRs as best as possible, however, like all software projects, any discrepancies are apologized for at this stage.
2. Feedback on requirements is expected from the users to help the group in improving the design and implementation of the project.