THE DESIGN OF A LOGICAL TRAFFIC ISOLATION FORENSIC MODEL

[1]Innocentia Dlamini, [2]Martin Olivier

Information and Computer Security Architectures Research Group (ICSA)

Department of Computer Science, University of Pretoria

[1]idlamini@csir.co.za, [2]molivier@cs.up.ac.za

ABSTRACT

The network evidence currently presented in a court of law is often insufficient for prosecution purposes due to a loss of packets during the network transmission. Such packet loss may be caused by the congestion of data transmitted over the network, which only serves to further compound the delay in data transmission. The paper in hand extends the earlier work done on a forensic model for traffic isolation based on Differentiated Services (DiffServ). The logical traffic isolation (LTI) forensic model intends to solve the packet loss problem that may cause evidence to be insufficient. It isolates suspicious traffic from the normal flow by placing it on a dedicated route using DiffServ prioritising characteristics that avoid congestion of the suspicious traffic. The LTI model further includes a preservation station that serves to record all suspicious traffic before it is forwarded to its destination. This paper focuses on the analysis and design of the LTI model. An attempt is made to design a more flexible and reliable system – with a minimal loss of evidence – by incorporating some of the design algorithms.

KEY WORDS

Differentiated services, preservation station, network forensics, suspicious traffic, unified modelling language, Network Intrusion Detection System.

THE DESIGN OF A LOGICAL TRAFFIC ISOLATION FORENSIC MODEL

1 INTRODUCTION

This paper presents the design of the concept of a forensic model for Logical Traffic Isolation (LTI) based on Differentiated Services (DiffServ), as proposed by Strauss et al. [1]. Whenever network forensic investigations need to be performed, it is to the advantage of investigators if the crime is still in progress [2]. Seeing that they do not have to shut down the communication, they can often succeed in gathering enough evidence. The LTI model intends to solve the packet loss problem that can be the cause of insufficient evidence. It isolates suspicious traffic from the normal flow and places it on a dedicated route using DiffServ prioritising characteristics, thus avoiding a congestion of the suspicious traffic. The LTI model also includes a preservation station that serves to record all suspicious traffic before it is forwarded to its destination.

The LTI model utilises the DiffServ approach to isolate malicious traffic from normal traffic [1]. This could well reduce cost, since DiffServ is a standard technique. If a DiffServ infrastructure is already in place where an investigation needs to be performed, evidence collection could be facilitated with minimal changes to the network. The DiffServ approach allows Network Forensic investigators to attach both their marking station (ingress router) and preservation station to a cyber victim's network. The purpose of the marking station is to isolate the suspicious traffic and that of the preservation station is to investigate the situation at hand. The advantage of this approach is that it requires minimal network downtime and, most importantly, minimal network reconfiguration. This DiffServ-based scheme makes provision for a preservation station to store records of the isolated traffic with a view to its later analysis [1].

However, in order to minimise network transmission problems such as transmission delays and high network traffic, the preservation station proposes to store only records related to malicious network traffic. While the proposal seems plausible, it has not been tested yet to prove the LTI system’s viability. In order to ensure a successful and reliable implementation, this paper uses various design techniques in modelling the LTI model. A Unified Modelling Language (UML) technique is favoured in most of the cases. It provides abundant diagrams that can explicitly depict most of the processes and the interaction between the components of the LTI model. The rest of the paper is structured as follows: Section 2 discusses the architecture of the LTI model. Section 3 presents a design of the LTI model using the system design technique, while Section 4 serves to conclude the paper.

2 THE LTI ARCHITECTURAL MODEL

The design stage started with a careful revision of the requirements of the LTI system that were defined by Strauss et al. [1] after adding further elements. Some of the requirements included the type of network setup already in place. The system is intended to solve the problem of inefficient and inadequate evidence by introducing a preservation station for capturing identified packets. This station can be easily plugged into the network whenever an intrusion has been detected, thus allowing the system to immediately conduct an investigation while the suspected cyber-crime is being committed, i.e. live-network forensics. [2]

For experimentation reasons, the LTI system should have seven nodes: two nodes on a traffic generator that act as users and generate normal and suspicious traffic randomly; three nodes or routers on the DiffServ network, in other words the ingress, immediate and egress routers; and a sixth node that is the preservation station for recording the traffic that has been detected as suspicious. The last node is the sink server that receives and processes the requests generated. (Both the traffic generator and the sink server are additional nodes.) The LTI system should be able to isolate the two types of generated traffics within the DiffServ network and record the suspicious packets at the preservation station. The system is designed on the basis of three assumptions: (1) The network has its intrusion detection system in place; (2) There are various users transmitting data (represented by a traffic generator for experimentation purposes); and (3) The receiver or the destination node is represented by the sink server.

The requirements of the system served as the foundation for this study and resulted in the following implementation infrastructure of the LTI model (see Figure 1). It provides a conceptual view of the LTI model based on DiffServ for isolating suspicious traffic. The model consists of two traffic generators on the client side to initiate suspicious and normal traffic and of the DiffServ network with three routers (ingress, interior and egress) for experimental purposes. The preservation station ensures forensic soundness and system reliability [3] [4], while the sink server receives and responds to all the requests generated by the traffic generator. This nodal setup is, however, for experimentation purposes only.

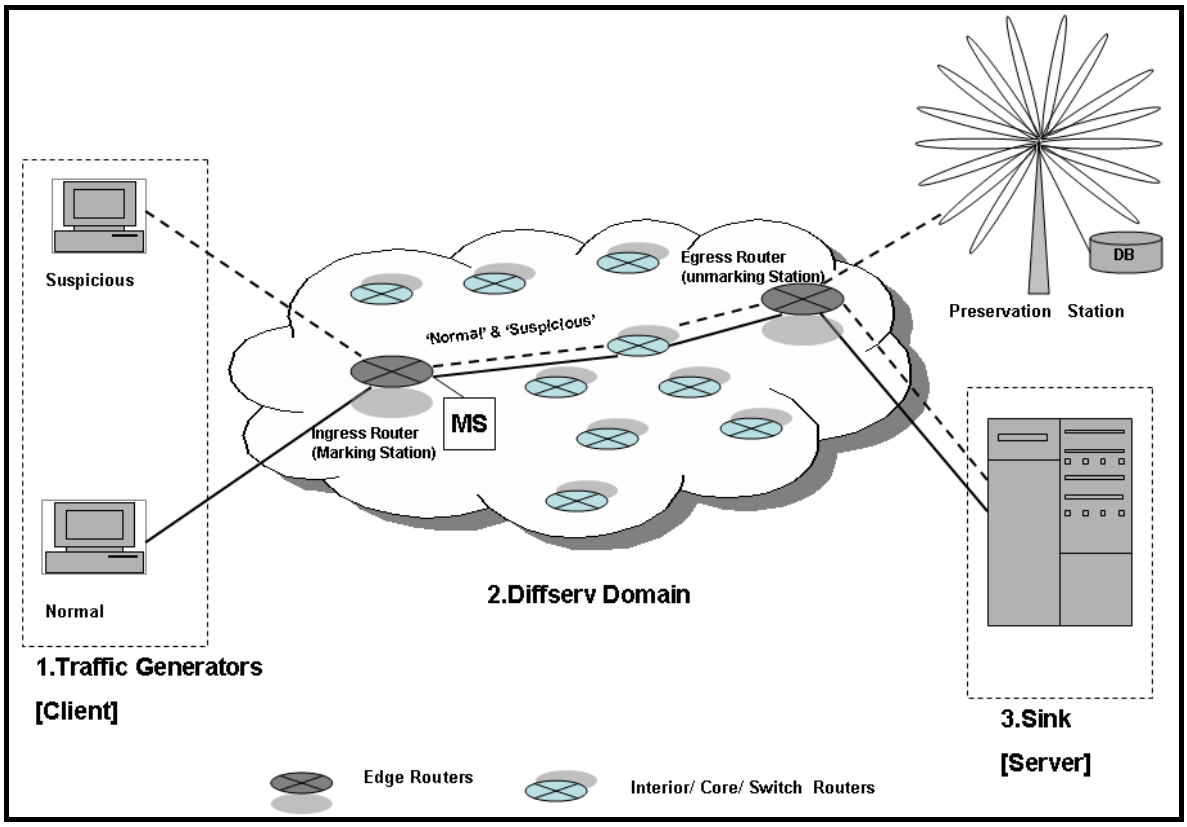

Figure 1: The Implementation Infrastructure of the LTI model using DiffServ

The two clients generate normal and suspicious traffic and forward these packets to the DiffServ domain. The ingress edge router at the entrance boundary of the DiffServ domain is the first domain recipient and serves as a marking station. This router is responsible for *packet*

classification and has *marking*, *shaping* and *dropping* capabilities. The ingress router marks any suspicious traffic by using the packet classifier and forwards them to the nearest core router. The core routers are found within the centre of the DiffServ domain, and they simply forward traffic towards the egress router. The egress router is found at the exit boundary of the DiffServ domain. It unmarks the traffic and decides the destination of each network packet according to its behaviour: compromised traffic is forwarded to the preservation station and then to the sink server, while normal traffic is sent directly to the sink server. In a network-related cyber incident, the investigator searches the preservation station when conducting his/her investigation and captures all recorded suspicious network packets as evidence. The LTI model is further formalised in various UML diagrams. This includes the sequence diagram and activity diagram. The following section discusses the design of the LTI model in detail.

3 THE LTI SYSTEM DESIGN

The second step in the design of the LTI model is its representation using the UML design technique. Although the UML is not a cure-all, it does simplify our work. These diagrams do not include too much detailed information; they simply depict the applicability and functionality of the LTI model and the involvement of the requirements of the system.

3.1 The LTI System Scenarios

In Figure 2, the Network Investigator (the actor) interacts with the system by performing different processes.

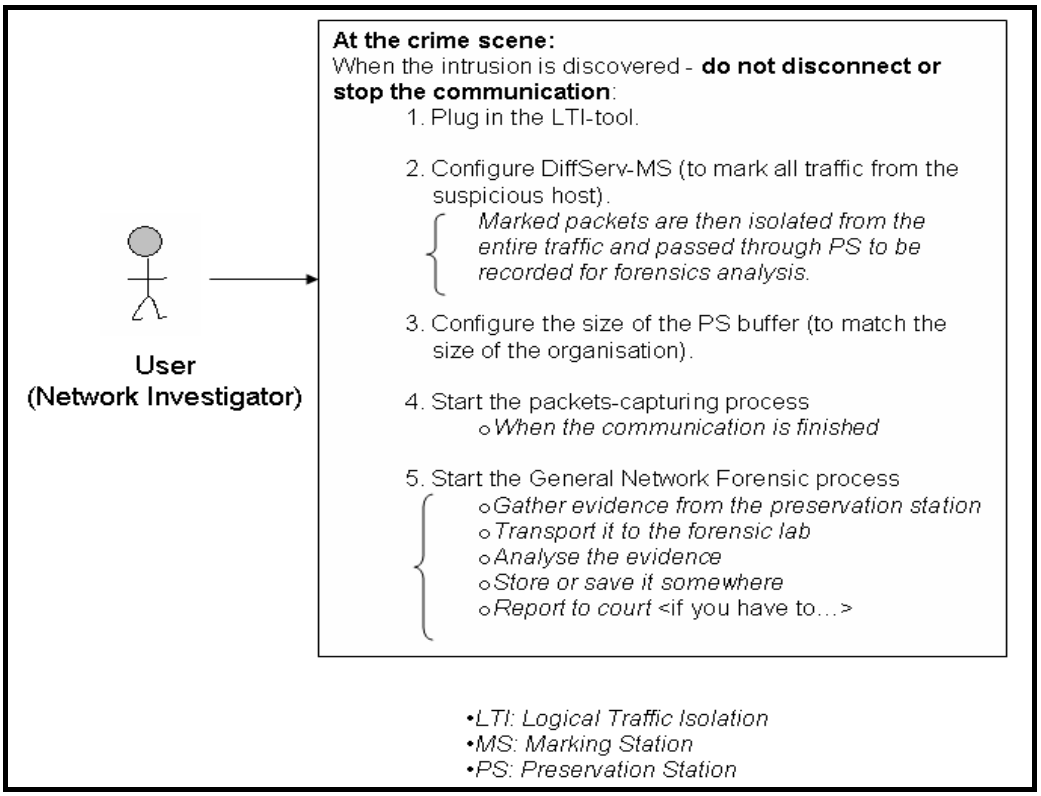

Figure 2: The Scenarios Involved in the Logical Traffic Isolation system

These processes are:

- Plug in the LTI tool
- Configure DiffServ-Marking Station (MS)
- Configure the buffer size of the Preservation Station (PS)
- Start the packets-capturing process
- Start the Network Forensics [2] investigation into the system

The LTI model involves all of the above processes during the course of an intrusion; in other words, it starts immediately when the intrusion has been detected.

When an investigator arrives at the crime scene, the suspicious communication has to be left up and running in order to capture and record the detected packets. The LTI tool should easily plug into the affected network and require only minor configuration to suit the size of the network at hand. Such configuration includes enabling the marking station to mark the packets from the suspicious host, considering the buffer size of the preservation station and ensuring correspondence with the size of the organisation. When the necessary configuration has been completed, the investigator can start the tool to capture suspicious packets. As soon as the suspicious communication is over, the normal network forensics processes can be initiated (see Figure 2, note 5). The processes involved in the LTI model can be arranged into different sequences, as is discussed in the following subsection.

3.2 The Sequence Diagram of the LTI Model

A sequence diagram is also part of the UML. It is used to show how processes operate with one another and in what order. Figure 3 depicts five components.

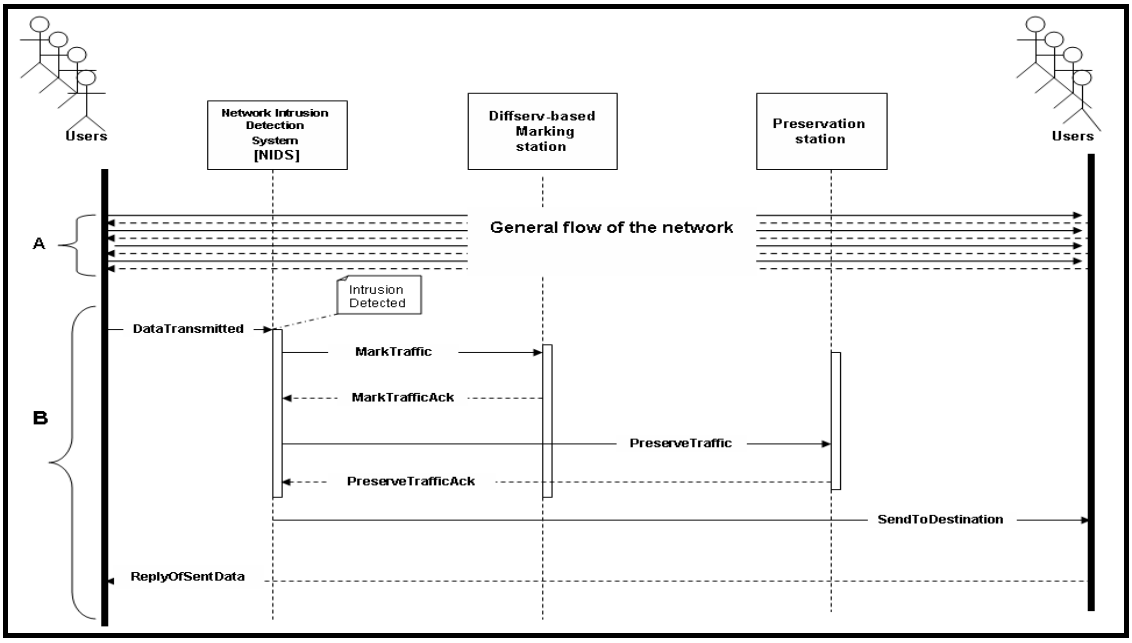

Figure 3: Sequence Diagram for the Logical Traffic Isolation System

These components include the users, a Network Intrusion Detection System (NIDS) (this can be any detection system; it differs from one organisation to the next), DiffServ network, a preservation station and the sink server. As mentioned above, the NIDS and sink server are part of the model. Two categories of traffic can be distinguished – suspicious and normal traffic. These are randomly generated. In Figure 3 above, Scenario A represents the normal network traffic flowing from the users to their destinations (which is represented by the sink server for supporting the experimentation of the LTI model).

This network passes through all the nodes, except the preservation station. Scenario B represents the sequence when suspicious traffic has been detected. The NIDS system reports to the DiffServ module to mark this traffic and give it higher priority. It also provides proper routing methods to find it, as suspicious traffic is the type of traffic that is special and significant to the cyber investigator. Suspicious traffic is first routed to the preservation station to be recorded, after which it is allowed to be routed to its destination. The ReplyOfSentData shown by dotted lines depicts the destination user's reply to the initiation user's request. The same procedure as in B can also be applied to show the response of the targeted system. As part of the UML diagrams, the activity diagram is used in the following subsection to show the activities performed by the actors involved in the LTI system.

3.3 The Activity Diagram of the LTI Model

Activity diagrams provide another means for clarifying which actor carries out which activity. Consider the activity diagram in Figure 4, which provides a breakdown of main activities into different subactivities. This diagram starts with normal flow of the network, assuming that a detection system is already in place. The latter serves as a deciding device as it informs the network administrator of any detected incident.

The network administrator easily plugs in the LTI tool and then configures the marking station to mark the packets of detected traffic. The LTI system checks whether the packets have been marked, and if not, sends them back. Once they have been marked, they are forwarded to the preservation station to be recorded. The system again checks to ensure that packets have indeed been recorded and sends them back if not. Once they have been recorded, they are sent on to their destination (sink server is used in our model as a supporting node). The system continuously checks whether more packets should be detected and, if this is the case, returns them to the marking station to start the process again.

The Network Forensics [2] [5] investigation process commences at this stage and its activities are included in a type of activity diagram. The investigation process is initiated and the approved Network Forensic tool is used to collect the evidence recorded by the preservation station. This step continues until all evidence has been gathered. The next activity that is performed is the transportation of forensic evidence to the forensic lab; followed by the analysis of all evidence gathered.

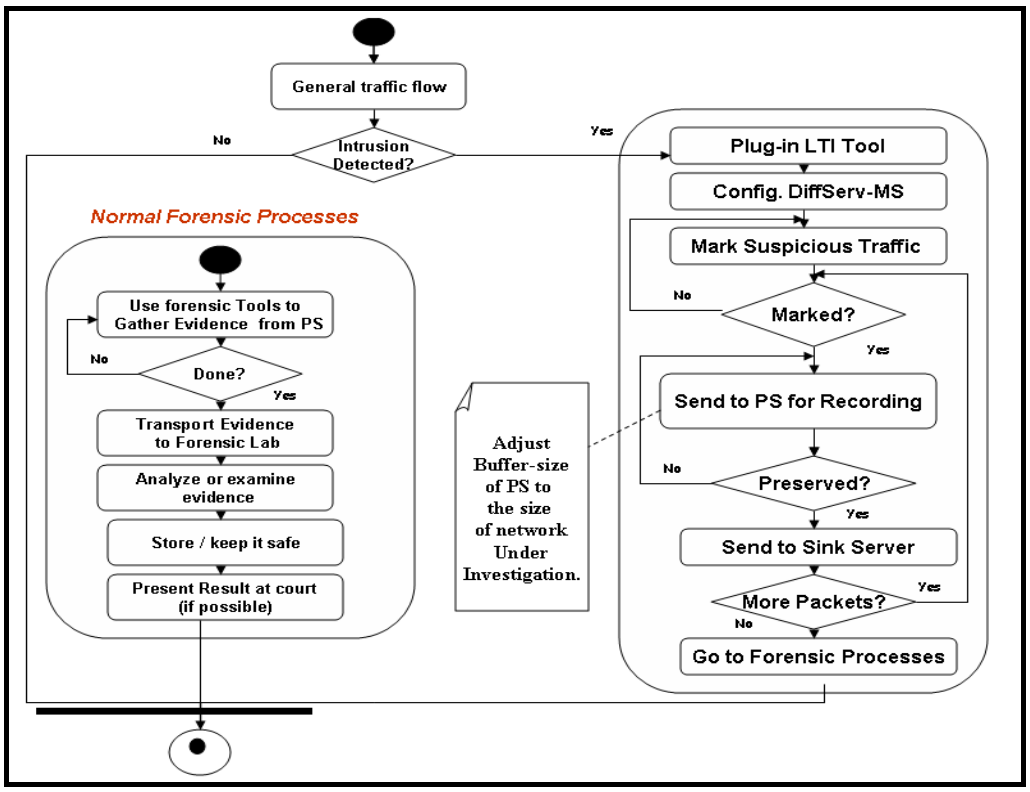

Figure 4: Activity Diagram for the Logical Traffic Isolation system

The findings of such analyses are stored in a place that is adequately safe while waiting for the court date. The final activity is the presentation of evidence in the form of a report to a court of law if this is deemed necessary. The next subsection discusses the class diagram of the LTI model.

3.4 The Class Diagram of the LTI Model

A class diagram is formulated from the components mentioned above. Figure 5 depicts the class diagram of the LTI system. Some Object-Oriented design principles [6] that were considered during the modelling of this class diagram are as follows:

"... strive for loosely coupled design between objects that interact (p. 53); …open-close principles (p. 86); … favour composition over inheritance (p. 75)" [6].

The relationship between the subject and the observers in the observer pattern complies with the design principle for favouring composition over inheritance, while the communication between the subject and the observers is kept loosely coupled. The open-close principle is implemented by the decorator pattern through allowing the behaviour of the traffic generated to be extended without any modification to the entire code. The traffic generator and the sink server objects use the DiffServ object for communication. This reduces the number of messages sent between the objects in the system and DiffServ therefore acts as a mediator.

Three design patterns are used in modelling the LTI architecture, namely the Decorator, Observer and Mediator patterns. The decorator pattern [6] is used to randomly wrap the behaviour

of the traffic generated. The observer pattern [6] [7] is interchangeably used in most of the components of the LTI model, including the traffic generator, DiffServ, preservation station and sink server. The mediator pattern [8] [9] is used to coordinate the traffic generator with the preservation station or sink server components.

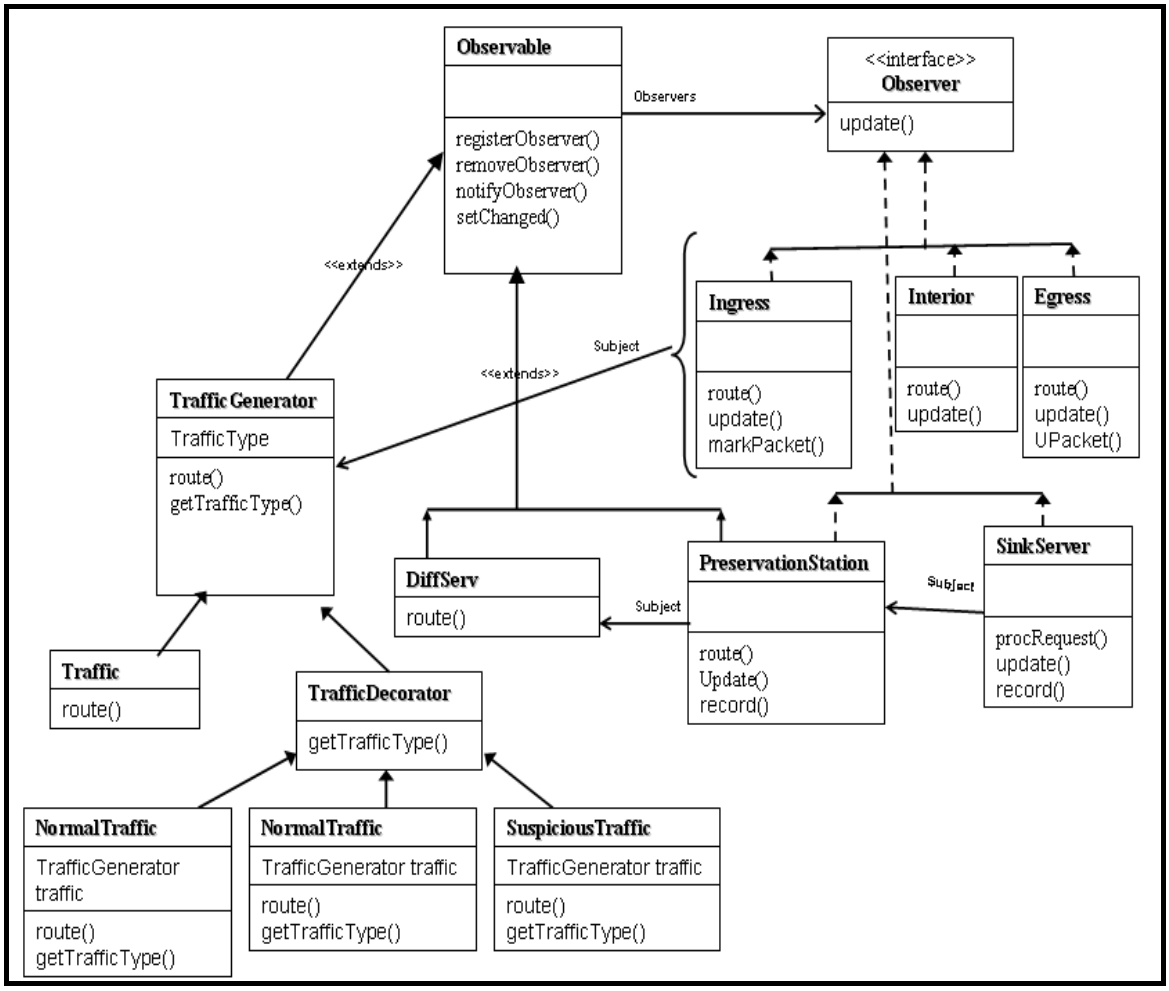

Figure 5: The Class Diagram of the LTI model using DiffServ

4 CONCLUSION

This paper uses the UML design technique in most of the cases to present the LTI model. According to the specified requirements of the system, various design diagrams (the sequence diagram, activity diagram and class diagram) are used to represent the model. These diagrams specify in detail the role of each component of the system and the interaction between investigator and system, as well as the precautions that must be kept in mind when handling each piece of evidence. The use of these diagrams simplifies the LTI model and facilitates its easy implementation. The LTI model is currently in the process of being implemented, and the necessary performance evaluation and tests should therefore still be carried out in future.

5 REFERENCES

[1] Strauss, T., Olivier, M.S. & Kourie, D.G. 2006, Differentiated Services for Logical Traffic Isolation, in M.S. Olivier and S. Shenoi (Eds), Advances in Digital Forensics II, pp. 229-237, Springer.
[2] Corey, V., Peterman, C., Shearin, S., Greenberg, M.S. & Van Bokkelen, J. 2002, Network Forensics Analysis, Internet Computing, Volume 6, pp. 60- 66, IEEE.
[3] Solomon, M.G., Barrett, D. & Broom, N. 2005, The Need for Computer Forensics, in L. Newman and W.G. Kruse (Eds), Computer Forensics Jump Start, pp. 01-20, SYBEX Inc.
[4] Kohn, M., Eloff, J. & Olivier, M.S. 2006, Framework for a Digital Forensic Investigation, in H.S. Venter, J.H.P. Eloff, L. Labuschagne and M.M. Eloff (Eds), Proceedings of the ISSA 2006 from Insight to Foresight Conference, Sandton, South Africa (published electronically).
[5] Zantyko, K. 2007, Commentary: Defining Digital Forensics, Forensic Magazine, 20, Vicon Publishing, Feb-March 2007 issue, [Online] Available at: http://www.forensicmag.com/articles.asp? pid=130, as on 12 April 2008.
[6] Freeman, E. & Sierra, K. 2004, Head First Design Patterns, Volume 1, O'Reilly Media, Sebastopol (CA), USA.
[7] Shalloway, A. & Trott, J. 2001, Design Patterns Explained: A New Perspective on Object-Oriented Design, Addison-Wesley.
[8] Bains, K. & Lau, E. 2002, Mediator Design Pattern. Available at: http://sern.ucalgary.ca/courses/SENG/443/W02/assignments/Mediator/, University of Calgary.
[9] Black, S. 2004, Mediator Design Pattern. Available at: http://stevenblack.com/PTN-Mediator. ASP. Steven Black Consulting.
[10] [GoF] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. 1996, Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2.