ISSN:0975-9646

Efficient way of Detecting Malicious Packet Losses

K Bala murali Krishna[#1],K Prasada Rao*[2]
[#]*(M.Tech),Dept. of CSE,AITAM,Tekkali,*
Andhrapradesh,India-532201

[*]*Asst.Prof,Dept. of CSE,AITAM,Tekkali,*
Andhrapradesh,India-532201

Abstract— **We consider the problem of detecting whether a compromised router is maliciously manipulating its stream of packets. In particular, we are concerned with a simple yet effective attack in which a router selectively drops packets destined for some Victim. Unfortunately, it is quite challenging to attribute a missing packet to a malicious action because normal network congestion can produce the same effect. Modern networks routinely drop packets when the load temporarily exceeds their buffering capacities. Previous detection protocols have tried to address this problem with a user-defined threshold: too many dropped packets imply malicious intent. However, this heuristic is fundamentally unsound; setting this threshold is, at best, an art and will certainly create unnecessary false positives or mask highly focused attacks. We have designed, developed, and implemented a compromised router detection protocol that dynamically infers, based on measured traffic rates and buffer sizes, the number of congestive packet losses that will occur. Once the ambiguity from congestion is removed, subsequent packet losses can be attributed to malicious actions. We have tested our protocol in Emulab and have studied its effectiveness in differentiating attacks from legitimate network behavior.**

Keywords-Denial of service,Emulab, Object Modeling Technique (OMT), Class Responsibility Collaborator (CRC)

I. INTRODUCTION

The Internet is not a safe place. Unsecured hosts can expect to be compromised within minutes of connecting to the Internet and even well-protected hosts may be crippled with denial-of-service (DoS) attacks. However, while such threats to host systems are widely understood, it is less well appreciated that the network infrastructure itself is subject to constant attack as well. Indeed, through combinations of social engineering and weak passwords, attackers have seized control over thousands of Internet routers. Even more troubling is Mike Lynn’s controversial presentation at the 2005 Black Hat Briefings, which demonstrated how Cisco routers can be compromised via simple software vulnerabilities. Once a router has been compromised in such a fashion, an attacker may interpose on the traffic stream and manipulate it maliciously to attack others—selectively dropping, modifying, or rerouting packets.
Several researchers have developed distributed protocols to detect such traffic manipulations, typically by validating that traffic transmitted by one router is received unmodified by another. However, all of these schemes—including our own—struggle in interpreting the absence of traffic. While a packet that has been modified in transit represents clear evidence of tampering, a missing packet is inherently ambiguous: it may have been explicitly blocked by a compromised router or it may have been dropped benignly due to network congestion. In fact, modern routers routinely drop packets due to bursts in traffic that exceed their buffering capacities, and the widely used Transmission Control Protocol (TCP) is designed to cause such losses as part of its normal congestion control behavior. Thus, existing traffic validation systems must inevitably produce false positives for benign events and/or produce false negatives by failing to report real malicious packet dropping. We offer two definitions for packet loss rate
There are inherently two threats posed by a compromised router. The attacker may subvert the network control plane (e.g., by manipulating the routing protocol into false route updates) or may subvert the network data plane and forward individual packets incorrectly.

II. PROBLEM FORMULATION

There are inherently two threats posed by a compromised router. The attacker may subvert the network control plane (e.g., by manipulating the routing protocol into false route updates) or may subvert the network data plane and forward individual packets incorrectly. The first set of attacks have seen the widest interest and the most activity—largely due to their catastrophic potential. By violating the routing protocol itself, an attacker may cause large portions of the network to become inoperable. Thus, there have been a variety of efforts to impart authenticity and consistency guarantees on route update messages with varying levels of cost and protection.We do not consider this class of attacks in this paper. Instead, we have focused on the less well-appreciated threat of an attacker subverting the packet forwarding process on a compromised router. Such an attack presents a wide set of opportunities including DoS, surveillance, man-in-the-middle attacks, replay and insertion attacks, and so on. Moreover, most of these attacks can be trivially implemented via the existing command shell languages in commodity routers.The earliest work on fault-tolerant forwarding is due to Perlman who developed a robust routing system based on source routing, digitally signed route-setup packets, and reserved buffers. While groundbreaking, Perlman’s work required significant commitments of router resources and high levels of network participation to detect anomalies. Since then, a variety of researchers have proposed lighter weight protocols for actively probing the network to test whether packets are forwarded in a manner consistent with the advertised global topology. Conversely, the 1997 WATCHERS system detects disruptive routers passively via a distributed monitoring algorithm that detects deviations from a "conservation of flow" invariant. However, work on WATCHERS was abandoned, in part due to limitations in its distributed detection protocol, its overhead, and the problem of ambiguity stemming from congestion Finally, our own work broke the problem into three pieces: a traffic validation mechanism, a distributed detection protocol, and a rerouting countermeasure. In and, we focused on the detection protocol, provided a formal framework for evaluating the accuracy and precision of any such protocol, and described several practical protocols that allow scalable implementations. However, we also assumed that the problem of congestion ambiguity could be solved, without providing a solution. This paper presents a protocol that removes this assumption.

III. SYSTEM DESIGN

Design involves identification of classes, their relationships as well as their collaboration. In objectory ,classes were divided into Entity classes ,interface classes and the control classes. The Computer Aided Software Engineering tools that are available commercially do not provide any assistance in this transition. Even research CASE tools take advantage of meta modeling are helpful only after the construction of class diagram is completed. In the Fusion method ,it used some object-oriented approaches like Object Modeling Technique (OMT), Class Responsibility Collaborator (CRC) and Objectory, used the term Agents to represent some of the hardware and software systems .In Fusion method, there was no requirement phase ,where in a user will supply the initial requirement document. Any software project is worked out by both analyst and designer. The analyst creates the Use case diagram. The designer creates the Class diagram. But the designer can do this only after the analyst has created the Use case diagram. Once the design is over it is need to decide which software is suitable for the application

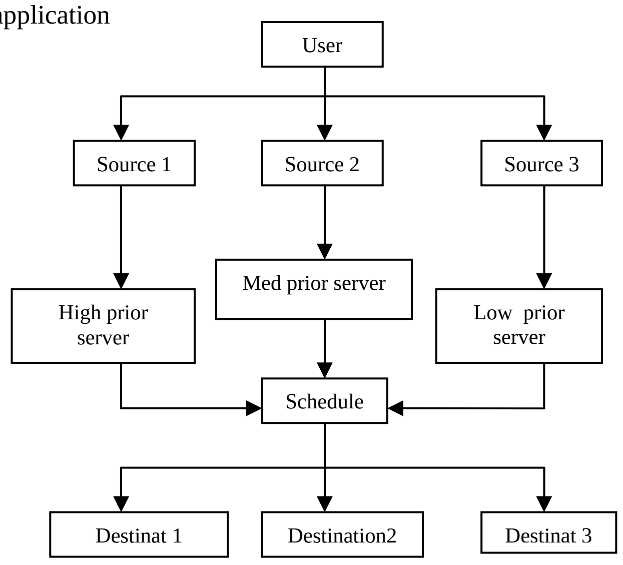

Fig.1: Architecture Diagram

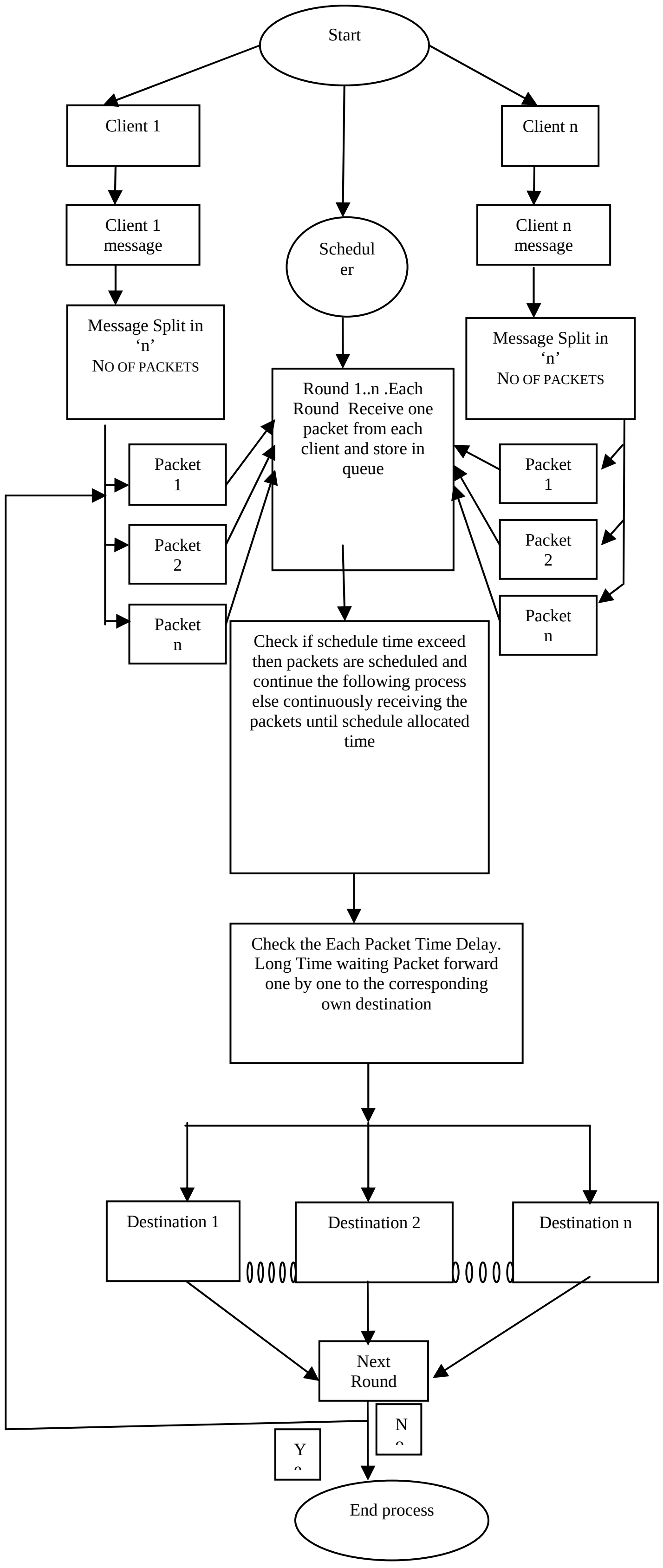
Start
Client 1
Client n
Client 1 message
Scheduler
Client n message
Message Split in 'n' NO OF PACKETS
Round 1..n .Each Round Receive one packet from each client and store in queue
Message Split in 'n' NO OF PACKETS
Packet 1
Packet 2
Packet n
Packet 1
Packet 2
Packet n
Check if schedule time exceed then packets are scheduled and continue the following process else continuously receiving the packets until schedule allocated time
Check the Each Packet Time Delay. Long Time waiting Packet forward one by one to the corresponding own destination
Destination 1
Destination 2
Destination n
Next Round
N
Y
End process

Fig.2:Data Flow Diagram

Phase Description :

TABLE.1: PHASE DESCRIPTION

Phase	**Task**	**Description**
Phase 1	Analysis	Analyze the information given in the IEEE paper.
Phase 2	Literature survey	Collect raw data and elaborate on literature surveys.
Phase 3	Design	Assign the module and design the process flow control.
Phase 4	Implementation	Implement the code for all the modules and integrate all the modules.
Phase 5	Testing	Test the code and overall process weather the process works properly.
Phase 6	Documentation	Prepare the document for this project with conclusion and future enhancement.

IV.IMPLEMENTATION AND RESULTS

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization.

The implementation can be preceded through Socket in java but it will be considered as peer to peer communication .For proactive routing we need dynamic routing. So java will be more suitable for platform independence and networking concepts. For maintaining route information we go for MS-SQL as database back end.

Module Description

1.Login Module

Used to check the user authorization.

2.Source Module

Used to maintain and perform the client input data and also the source data's are packet wise forwarded to the server in one by one.Each round only one packet send to scheduler . Give the chance to each class frames.

3. Packet Separation:

In this module we have to separate the input data into packets. These packets are then sent to the Queue.

measured traffic rates and buffer sizes, the number of congestive packet losses that will occur. Subsequent packet losses can be attributed to malicious actions. Because of non

Used to performing the following 3 operations…

That is Receiving the message, Queuing the message, Forwarding the Message.

Here Maintain one queue .That Queue based all the above 3 process are performed. EnQueue and Dequeue Process are performed. At last after receiving and scheduling packets are forwarded to the corresponding own destination.

4. Packet Receiver:

The Packet Receiver is used to receive the packets from the Queue after the packet loss. Then the receiver displays the received packets from the Queue.

5. User Interface Design:

In this module we design the user interface for Sender, Queue, Receiver and Result displaying window. These windows are designed in order to display all the processes in this project.

6. Packet Loss Calculation:

The calculations to find the packet loss are done in this module. Thus we are developing the tool to find the packet loss.

Future Enhancements

There are a number of avenues for future work, Our future research is to study end-to-end QoS in a Different domain by using Packet Loss Measurement at the routers. This is a challenging and interesting issue as it would require the schedulers in routers along the path to cooperate with each other to provide a desired QoS.

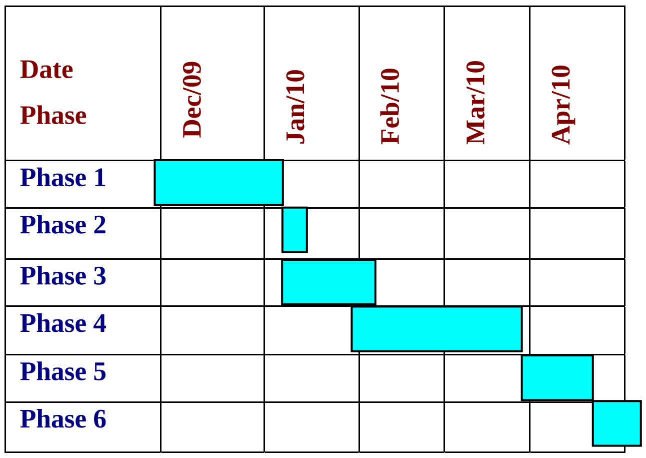

Fig.3:Survey of phase diagram

V . CONCLUSION

To the best of our knowledge, this paper is the first serious attempt to distinguish between a router dropping packets maliciously and a router dropping packets due to congestion. Previous work has approached this issue using a static user-defined threshold, which is fundamentally limiting. Using the same framework as our earlier work (which is based on a static user-defined threshold), we developed a compromised router detection protocol _ that dynamically infers, based on determinism introduced by imperfectly synchronized clocks and scheduling delays, protocol _ uses user-defined significance levels, but these levels are independent of the properties of the traffic. Hence, protocol _ does not suffer from the limitations of static thresholds.We evaluated the effectiveness of protocol _ through an implementation and deployment in a small network. We show that even fine-grained attacks, such as stopping a host from opening a connection by discarding the SYN packet, can be detected..

REFERENCES

- S. Blake et al., An Architecture for Differentiated Services, RFC 2475, Dec. 1998.
- V. Jacobson, K. Nichols, and K. Poduri, An Expedited Forwarding PHB, RFC 2598, June 1999.
- J. Heinanen et al., Assured Forwarding PHB Group, RFC 2597, June 1999.
- A.G.P. Rahbar and O. Yang, “The Output-Controlled Round Robin Scheduling in Differentiated Services Edge Switches,” Proc. IEEE BROADNETS ’05, Oct. 2005.
- D. Bertsekas and R. Gallager, Data Networks. Prentice Hall, 1992.
- S. Kanhere, A. Parekh, and H. Sethu, “Fair and Efficient Packet Scheduling Using Elastic Round Robin,” IEEE Trans. Parallel and Distributed Systems, vol. 13, no. 3, pp. 324-336, Mar. 2002.
- Y. Ito, S. Tasaka, and Y. Ishibashi, “Variably Weighted Round Robin Queueing for Core IP Routers,” Proc. IEEE Int’l Performance, Computing, and Comm. Conf. (IPCCC ’02), Apr. 2002.
-]M. Shreedhar and G. Varghese, “Efficient Fair Queuing Using Deficit Round Robin,” IEEE/ACM Trans. Networking, vol. 4, no. 3, June 1996.
- Y. Jiang, C.-K. Tham, and C.-C. Ko, “A Probabilistic Priority Scheduling Discipline for Multi-Service Networks,” Elsevier Computer Comm., vol. 25, no. 13, pp. 1243-1254, 2002.

BIBLIOGRAPHY NOTES

Author1:Bala Murali Krishna Kondagorri, is pursuing his MTech at Aditya Institute of Technology And Management , Tekkali,Srikakulam,Andhra Pradesh, affliated to JNTU Kakinada, Andhrapradesh. He received his BTech from JNTUniversity, Andhrapradesh, India, in 2008. He Worked as Asst. Prof for one year after his B.Tech.

Author 2 : K Prasad Rao, received B. Tech in Computer Science from Jawaharlal Technological University, Kakinada, A.P, India, M.Tech in computer science from Jawaharlal Technological University, Kakinada, A.P, India. He is having seven years of total experience in teaching. He guided many projects for B.Tech, M.Tech. His area of research include Computer Communications, Network Security