ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2025, Volume 6 Issue 2
www.ijisea.org

A Policy Enforcing Mechanism for Trusted Ad Hoc Networks

Mutyam Lakshmi Prasanna[1], P.Chiranjeevi[2]
PG Student[1], Professor[2]

Department of Master of Computer Applications
Amrita Sai Institute of Science and Technology
ParitalaNTR(Dist) Andhra Pradesh 521180
mutyamsrilakshmi87@gmail.com

ABSTRACT

This project presents the design and implementation of a policy enforcement mechanism using a Trusted Execution Monitor (TEM) built on top of the Trusted Platform Module (TPM). The goal is to enhance the security and trustworthiness of systems by enforcing security policies in a tamper-resistant environment. To demonstrate the effectiveness of the proposed approach, we developed a SATEM-based method to implement network access control in ad hoc networks. Ad hoc networks, due to their decentralized and dynamic nature, are particularly vulnerable to unauthorized access and malicious activity. By integrating TPM capabilities with a TEM framework, our system ensures that only trusted devices adhering to predefined security policies can participate in the network. The solution enables secure communication, dynamic policy updates, and real-time monitoring of node behavior. Experimental results indicate that the proposed mechanism provides robust access control while maintaining performance suitable for resource-constrained ad hoc environment(Arial size 10 normal)

I.INTRODUCTION

1. With the maturity of short-range wireless technologies and proliferation of mobile computing devices, building real-life applications over mobile ad hoc networks (MANET) becomes feasible. For instance, two potential applications are traffic monitoring in vehicular networks and peer-to-peer file sharing in ad hoc networks of smart phones.

2 .A key to the success of such applications is a mechanism assuring secure communication and proper collaboration among all participant entities. To achieve this goal, communication policies that govern the interactions between entities must be defined and enforced. For instance, in a traffic monitoring application, the policy can guarantee that a car always forwards accident alerts to cars coming behind it. Similarly, in a peer-to-peer application, the policy can guarantee that a smart phone can post a query only if it has made several contributions such as publishing files or forwarding other queries. While these methods provide sufficient expressive power to represent policies for MANET applications, the challenge is how to enforce such policies in MANETs. Most of the existing policy enforcement solutions have focused on Internet-based systems . Unfortunately, these solutions are not fit for MANET for two reasons. First, they enforce policies on trusted “choke points” (e.g., firewall or proxy), which do not exist in MANETs due to the lack of infrastructure.

.

II.RELATED WORK

1 . Secure Routing

Consider a group of nodes supporting Ad hoc On Demand Distance Vector (AODV) routing protocol. AODV is known to be vulnerable attacks, in which an attacker exploits a fast tunnel to attract all network traffic through it. One way to defeat this attack is to implement Packet. The node has to rely on roundtrip delay to estimate the time needed for an AODV message to reach the other

2 .Unselfish Sharing:

Each node simultaneously posts queries, answers queries, receives responses, and forwards queries for others. To benefit all nodes in the network, it is vital to ensure that enough nodes respond to and relay the queries Posted by others. Similar concerns exist in other applications such as a P2P file sharing network, where sufficient file providers are desired. To achieve these goals, each node must abide by a policy, like the following before joining the network:

Every node has to serve or relay at least 1 request from others after posting 3 queries to the network

3 .Policy Enforcement:

Each policy is enforced at its associated trusted tier independently. Each trusted tier Ti ensures both compliance and authenticity of the messages in Ti.S as follows:

4 .Trusted Multi-tier Network*:*

A tier is created step-by-step. First, a node begins to enforce the tier policy. It creates the tier key, which is used to authenticate in tier communications. By doing so, it becomes the first member of the tier, called originator of the tier, e.g. node 1. The originator then broadcasts an invitation to its neighbors, e.g. node 2 and 3, to join the newly created tier. Assume node 2 and 3 choose to join this tier. Since node 3 enforces PR, it succeeds in joining the tier and receives the tier key from node 1, but node 2 fails because it does not enforce PR. Next, node 3 extends the tier one step further by inviting nodes 4 and 5. Similarly, node 4 joins and continues the process to include node 6 in the tier.

III.SYSTEM ANALYSIS

The maturity of short-range wireless technologies and proliferation of mobile computing devices, building real-life applications over mobile ad hoc networks (MANET) becomes feasible. For instance, two potential applications are traffic monitoring in vehicular networks and peer-to-peer file sharing in ad hoc networks of smart phones. A key to the success of such applications is a mechanism assuring secure communication and proper collaboration among all participant entities.

IV .EVENT LISTENERS

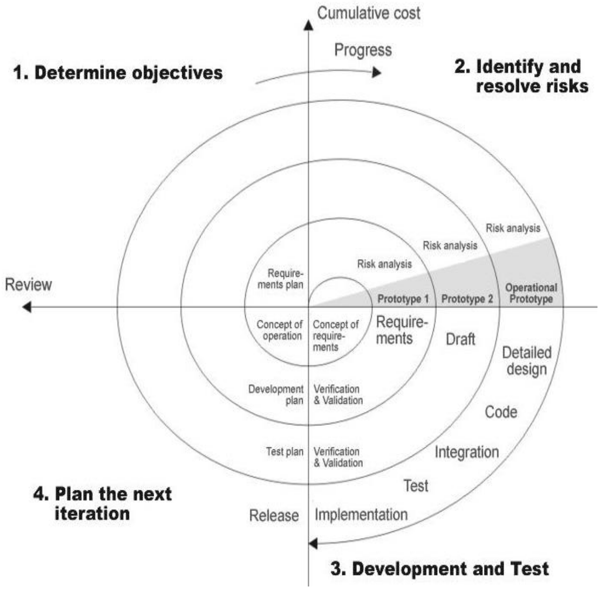
A listener is an object that is notified when an event occurs. It has two major requirements.

First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications. The methods that receive and process events are defined in a set of interfaces found in java.awt.event.

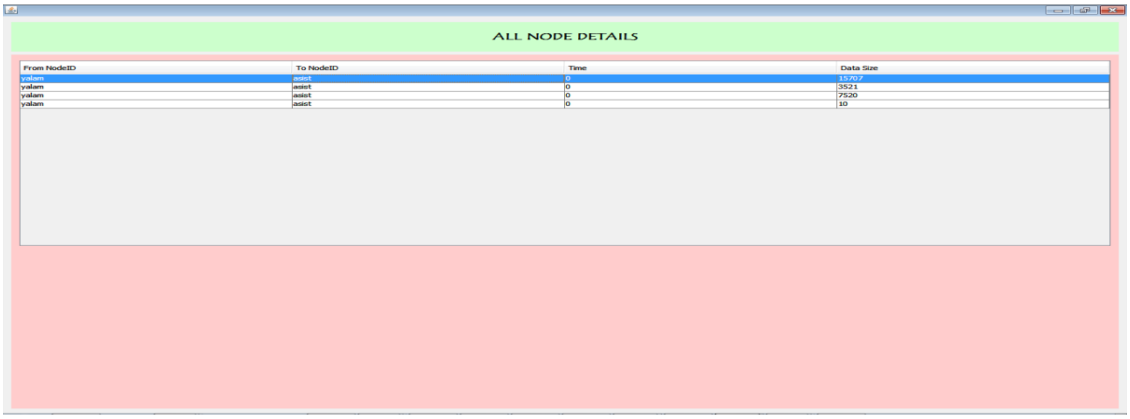
V .RESULTS

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

Class Diagram



VI.RESULT


ALL NODE DETAILS

From NodeID	To NodeID	Time	Data Size
yalam	asist	0	15707
yalam	asist	0	3521
yalam	asist	0	7520
yalam	asist	0	10

VII .CONCLUSIONS

We evaluated the method through a prototype based on an IEEE 802.11 ad hoc network and through network simulations.The results demonstrate the feasibility of the proposed method as well as its low overhead.

.

VIII.DISCUSSIONS

The tier manager is an application that allows the node to create, join and merge into a tier. Then, the tier manager communicates with the tier managers on other nodes through the JOIN or MERGES protocol. Before creating or joining a tier, the user first registers the tier enforcer with the tier manager.

ACKNOWLEDGEMENT(S) :

The authors wish to thank Dr.P.ChiranjeeviPh.D,Dr.M.SasidharDr.P.Chiranjeevi. This work was supported in part by a grant from Mr. D.VENKATESWARLUREFERENCES :Dr.P.Chiranjeevi, Department of CSE, for their steadfast support, direction, and use full advice. which played a crucial role in project and were essential in the creation of this thesis.