Software Requirements Specification

for

Network Intrusion Detection System

Version 1.0 approved

Prepared by Nijesh kumar

Snr college

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

To create a Network Intrusion Detection System (version 1), that detects the intruding activities by capturing the data and inspecting it. The system responds the user by an alarm. This project is applied to single host machine which is networked in an organization.

1.2 Document Conventions

The Document uses Times New Roman font, with a 14 pixel size for heading and 12 for others and the sub headings are marked as bold and main features in italics, for quick reference to the clients.

1.3 Intended Audience and Reading Suggestions

Network Intrusion Detection System (NIDS) is based on detection the attacks by intruders. The project is intended to develop for network administrator, who can control the system. The project may also useful for programmers to add more features than the existing one.

1.4 Project Scope

The Objective of the project is to create a Network Intrusion Detection System (NIDS) that detect the attacks of the intruders which is already defined. The NIDS sniffs the incoming and outgoing packets and compares the IP address to check whether it is to be block or not. It also maps what all packets have transferred. The benefit of this project is that administrator can add new IP s to be block to the NIDS.

1.5 References

a) *www.java2s.com*
b) *William stallings " network and security"*
c) *Network Security complete reference.*

2. Overall Description

2.1 Product Perspective

The product here we are creating is a newer version of Intrusion Detection System that can be applied to a single system in a distributed environment. The present system checks the current log details and also store the past details.

2.2 Product Features

a) ***Login*** *class: Login class will have a administrator and user.*
Administrator is a user provided with a user name and password. He has the privilege to add IP address to the database which is to be blocked. He will be provided with a log screen which maps all the details of incoming and outgoing packets.
User only can go through mapping details. He doesn't need a username and password to login.
b) ***IP management****: Steps to be carried out in managing IP address. Gives the details of what all IP address to be blocked. This is done by the Administrator.*
c) ***Packet Analyzer****: This part analyze all incoming and outgoing packet that pass through the system. Destination and source details are also listed in this part.*
d) ***Alarm Generator****: Alarm generator compares the incoming and outgoing IP address with the existing one in the database. If both matches it produces an alarm.*

2.3 User Classes and Characteristics

Admin class*: That provides all the functionality like add IP to the system which to be blocked. Details of the Incoming and outgoing IP address.*

User class*: user can map the details of incoming and outgoing IP to that particular system.*

2.4 Operating Environment

The IDE is written completely in java
Operating system is windows XP with a 1 GB ram and 80GB hard disc.
To run the project we need a java runtime environment (JRE) and a Mysql server.

2.5 Design and Implementation Constraints

NA

2.6 User Documentation

User documentation of the project is provided after the project is completed.

2.7 Assumptions and Dependencies

NA

3. System Features

The NIDS provides security against intruder attack. The packet Sniffer catches all incoming and outgoing packets. Intrusion Unit analyze them whether to block it or not. If the incoming or outgoing IP is to be blocked it will response through an alarm (warning). A mapping screen gives details of all incoming and outgoing data to a system.

Features includes

a) ***Login*** *class: Login class will have a administrator and user.*
Administrator is a user provided with a user name and password. He has the privilege to add IP address to the database which is to be blocked. He will be provided with a log screen which maps all the details of incoming and outgoing packets. The administrator privilege includes insertion, update and deletion of IP address in the database. He can also trace where the alarm is generated.
User only can go through mapping details. He doesn't need a username and password to login. His functions include start the scanning, pause the scanning and stop the scanning.

b) ***IP management****: Steps to be carried out in managing IP address. Gives the details of what all IP address to be blocked. The IP management includes adding, deleting and editing the IP address in the database. This is done by the Administrator.*

c) ***Packet Analyzer****: This part analyzes all incoming and outgoing packets that passes through the system. Destination and source details are also listed in this part.*

d) ***Alarm Generator****: Alarm generator compares the incoming and outgoing IP address with the existing one in the database. If both matches it produces an alarm.*

4. External Interface Requirements

4.1 User Interfaces

NIDS have an user interface screen where user get all the details of incoming and outgoing packets,An administrator screen where he can add IP address (block) to the database.

4.2 Hardware Interfaces

The System must be in distributed environment. NIDS works on TCP protocol. The IP address which is to be blocked should be well known for the administrator.

4.3 Software Interfaces

The system has a back end of MySql server where the IP address to be blocked is kept.

4.4 Communications Interfaces

The system requires a distributed environment to work with. The System uses TCP protocol to communicate.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>