SMARTECH

NetCheck Project

Requirements Analysis Report

Neslihan Bulut

Kezban Demirtaş

Hande Çelikkanat

Gülşah Karaduman

Filiz Alaca

Department of Computer Engineering

METU

November 2005

TABLE OF CONTENTS

1. PROJECT SCOPE AND DEFINITION

1.1 Project Title

Our project title is “NetCheck”.

1.2 Problem Definition

In today’s world, Internet has become the key tool in every aspect of life. However, it has the misfortune of being vulnerable to abuse. Being dependent on the Internet, organizations have to take precautions for abuse. At this point, software tools appear to act as the protector against malicious usage.

1.3 Project Scope

NetCheck will be a web-based application level gateway which will mainly provide the following facilities:

- Real-time network monitoring
- Content filtering
- Download restriction
- Access restriction
- Statistical data about network traffic
- Caching
- Web interface for the control of the program

The following sections of this report intend to explain these facilities and the requirements that should be met by our product.

2. MARKET OBSERVATION

Market observation is one of the key concepts in order to understand what the system is. With this aim in mind, we made literature survey and arranged meetings with the customers.

2.1 Literature Survey

We have searched the existing programs having similar features with our system in order to improve our knowledge about the current marketplace. With the help of the survey, we have determined the main properties of our application and we have also added some new features on it. In addition, literature survey showed us the missing points in the area. For instance, there are a wide range of features implemented in different software systems but a program supporting the whole features does not exist in the market. A comparative way of thinking enabled us to recognize most necessary features for an application level gateway.

Main properties of the searched programs are summarized below, together with the comparison table.

- ***Snort***

Snort is one of the best network intrusion detection and prevention systems, besides being free and open source. It makes use of a rule-driven language, which allows inspections based on signatures, protocols and anomalies detected.

Snort has four different running modes: Sniffer mode, Packet Logger mode, Network Intrusion Detection System (NIDS) mode and Inline Mode.

- In Sniffer mode, packets are read off the network and their details are displayed on the console.
- Packet Logger mode logs the packets to the disk. Users can change the default settings for logging to a convenient extend. For instance, there is an option which records incoming packets into subdirectories of the log directory, with the directory names being based on the address of the remote host. Another option, which is especially useful for a high speed network, is logging in binary mode, which is a more compact

form than ASCII format. In this mode, the packets are logged in tcpdump format to a single binary file. This file can later be easily read by either Snort or any other sniffer that supports tcpdump format (e.g. tcpdump or Ethereal).

- The NIDS mode, checks the packets against the rules configured in the configuration file, and decides if any action against an intrusion attempt is to be taken. Users can specify the format of logging and type of alerts. The default mode, which is the full alert mode, prints the alert message in addition to the full packet headers. It is also possible to run Snort with the “fast” option, for example to keep up with a 1000 Mbps connection.
- The Inline mode, obtains packets from iptables instead of libpcap and then uses new rule types to help iptables take actions regarding packets according to the Snort rules.

What makes Snort so powerful is that rules are powerful, flexible and relatively easy to write. For example, implementing a policy-based intrusion detection system is almost always too tricky for many users. However, Snort rules are “readable”, which allows a user to comprehend existing rules and modify them, and provides an opportunity to manipulate the whole system in a well-guided manner.

- ***DansGuardian***

DansGuardian [2] is a web content filtering proxy for Linux, FreeBSD, OpenBSD, NetBSD, Mac OS X, HP-UX, and Solaris systems. It supports the following features:

- Blocking advertisements by the use of an advertisement URL block list,
- Filtering text and HTML pages for pornographic content,
- Filtering according to MIME type and file extension,
- Filtering according to URLs,
- Filtering https requests,
- Working in a 'white list' mode where all sites except those listed are blocked,
- Blocking all IP based URLs,
- Blocking sites when users try using the IP address of the site instead,
- Logging in a human readable format,
- Optional logging in CSV for easy import into databases,

- Ability to switch off filtering for specified sites, parts of sites, browser IP’s and usernames,
- Blocking specified source IP’s and usernames,
- Blocking or limiting web uploading (e.g. attachments in e-mails),
- Ability to work in a stealth mode where it logs sites that would have been blocked, but does not block them allowing the administrator to monitor users without them knowing.

- ***WebSense***

Websense [3] is web filtering and web security software with a wide range of features;

- It works at the internet gateway, on the network and at the desktop.
- Web access should be managed with the options; allow, block, quota, continue, block by bandwidth and block by file type.
- Websense protects from spyware, key logging, malicious code and phishing scams.
- It offers the control of IM and P2P network access for the control of the corporation’s bandwidth. Unmanaged use of instant message attachments will also be controlled to preserve from risks.
- It blocks the access to spyware websites, and spyware backchannel communication at the gateway and doesn’t allow the spyware application to access at the desktop.
- Websense has policies for Windows NT, Active Directory, LDAP, eDirectory and RADIUS.
- For security threats it provides real time updates.
- Use policies can be defined for either groups or individual users.
- MSN messenger is allowed for only Professional Services organizations.
- Playing games is allowed for the employees but they are warned that the activity is not appropriate.
- Reporting tools allow the administrator to view application use and network access attempts.
- Removable devices such as CD/DVD burners, floppy drivers, flash drivers and external hard drives can be controlled by the administrator to supply policy definition flexibility.

  - Websense Security Labs are for improving Websense by analyzing web. For gathering feedback from the customer they use AppCatcher and WebCatcher technologies; WebCatcher lets the customers send uncategorized URLs to WebSense for analyzing, AppCatcher is the technology to ensure that latest executables and applications have been categorized and normalized at the customer’s side.

- ***NetNanny***

NetNanny [4] is a security tool especially designed for families who want to protect their children from dangers that they can encounter on the web. For this purpose, it stops illicit material from invading child’s computer by filtering and blocking web content while they surf, stopping illegal file sharing, protecting personal info and limiting time spent online.

Main features of NetNanny are;

- **Web Filtering ;**

  - NetNanny filters out inappropriate terms on web pages, in chat rooms and in e-mails by enabling user to enter keyword lists.
  - It enables user to create a "whitelist" of certain web sites so that only those sites, and no others, may be visited or viewed.
  - It uses pop-up blockers to keep the screen from filling up with advertisements.
  - It has an updatable block list of inappropriate web sites.

- **Logging and Monitoring;**

  - By monitoring activity, it gives the user the chance of observing the network traffic.
  - It gives alerts in case of both intentional and unintentional violations.
  - It sends activity reports by email.

- **Privacy Protection;**

  - It protects user’s private information by filtering it out of the data that leaves the computer in email, chat rooms and on the web.

- **Time Management;**

  - It limits the total amount of time spent online per day.
  - It has the option of setting up separate limits for different users.

- **Internet Application Blocking**

  - It can block access to chat and instant messaging.
  - It can block access to Internet games and newsgroups.
  - It can prevent illegal downloading of copyrighted or obscene material.

- ***Liss II Secure Gateway***

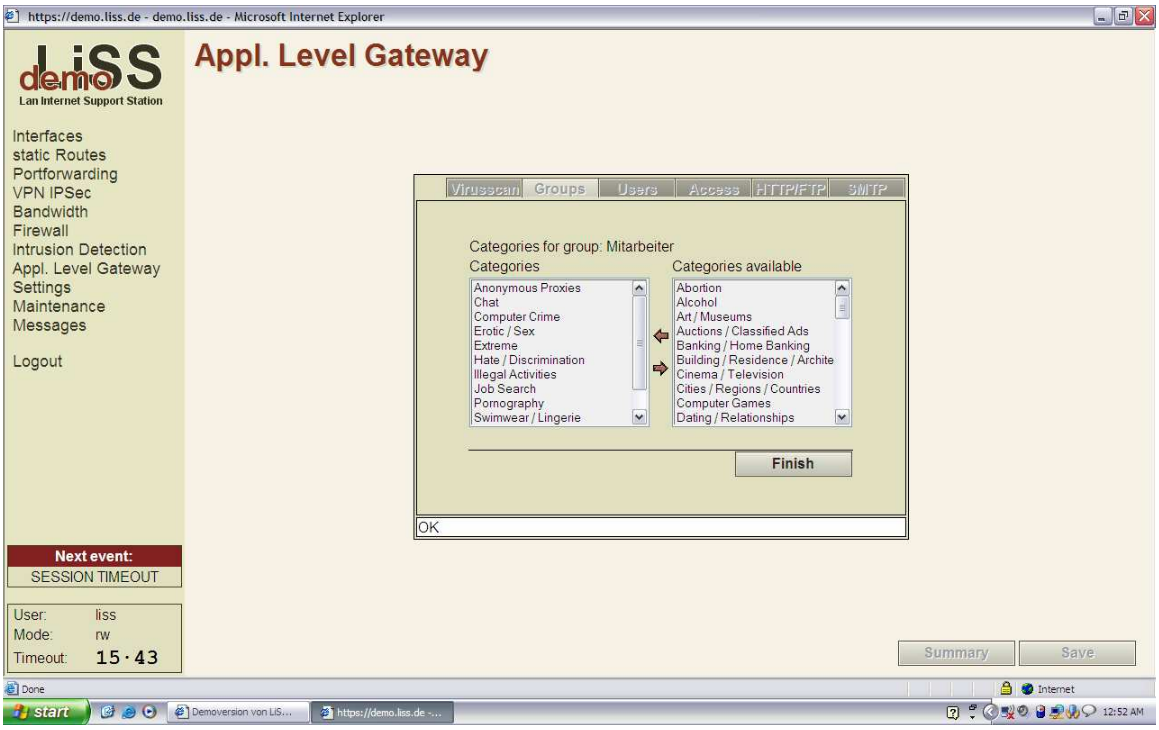
Liss II Secure Gateway [5] is a tool for companies which controls and regulates the traffic between the local network and the Internet and also prevents unauthorized access.

Main features of Liss II Secure Gateway are:

  - User identification
  - Proxies for HTTP, HTTPS and FTP
  - URL-blocking (Whitelists / Blacklists)
  - HTML-filtering (JavaScript, ActiveX, etc.)
  - Filtering of HTTP-headers
  - Active filtering of contents
  - Status-orientated packet filter (Stateful inspection)
  - Filtering according to IP- and/or MAC address
  - Filtering of IP packets according to source, target, port (service) and Interface
  - Static NAT

- Intrusion detection system
- Real-time data flow analysis
- Mail alarm
- Limitation of data downloads according to file type und volume
- Comprehensive record-keeping
- Formation of user groups
- Administered via a multilingual web interface

We have gained a better insight into the systems' working principles by examining the Liss' demo. Below, you can find some of the screenshots of the demo. This demo gives us an idea about the user interface part of our program.



Categories for group: Mitarbeiter

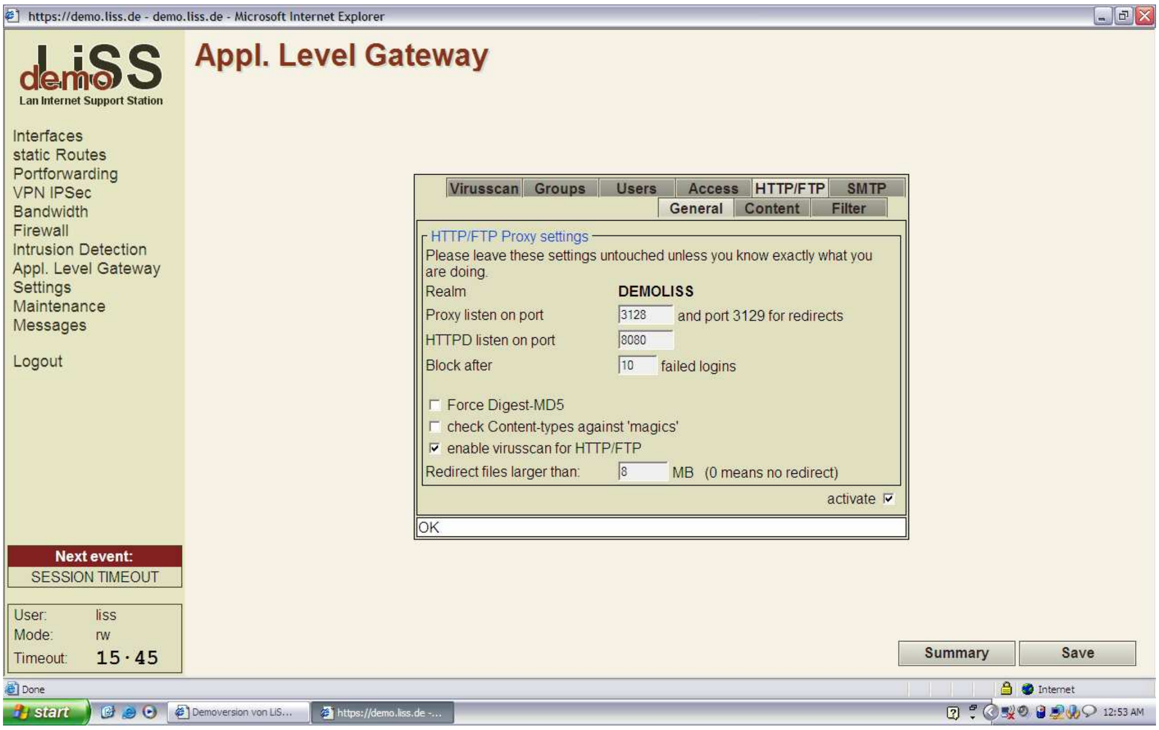
Categories
Anonymous Proxies
Chat
Computer Crime
Erotic / Sex
Extreme
Hate / Discrimination
Illegal Activities
Job Search
Pornography
Swimwear / Lingerie

Categories available
Abortion
Alcohol
Art / Museums
Auctions / Classified Ads
Banking / Home Banking
Building / Residence / Archite
Cinema / Television
Cities / Regions / Countries
Computer Games
Dating / Relationships

OK

Next event:
SESSION TIMEOUT

User: liss
Mode: rw
Timeout: 15·43



Appl. Level Gateway

HTTP/FTP Proxy settings

Please leave these settings untouched unless you know exactly what you are doing.

Realm **DEMOLISS**

Proxy listen on port 3128 and port 3129 for redirects

HTTPD listen on port 8080

Block after 10 failed logins

□ Force Digest-MD5

□ check Content-types against 'magics'

☑ enable virusscan for HTTP/FTP

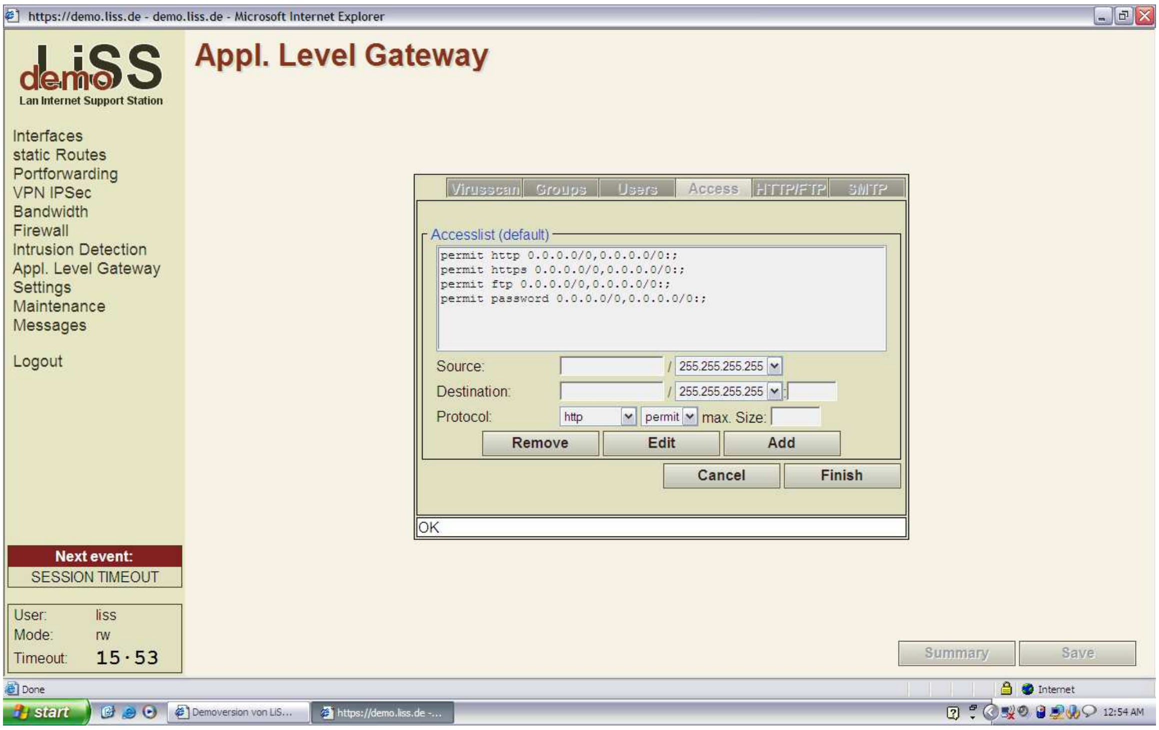
Redirect files larger than: 8 MB (0 means no redirect)

activate ☑

OK

Next event:
SESSION TIMEOUT

User: liss
Mode: rw
Timeout: 15:45



Appl. Level Gateway

Accesslist (default)

```
permit http 0.0.0.0/0,0.0.0.0/0:;
permit https 0.0.0.0/0,0.0.0.0/0:;
permit ftp 0.0.0.0/0,0.0.0.0/0:;
permit password 0.0.0.0/0,0.0.0.0/0:;
```

Source: / 255.255.255.255

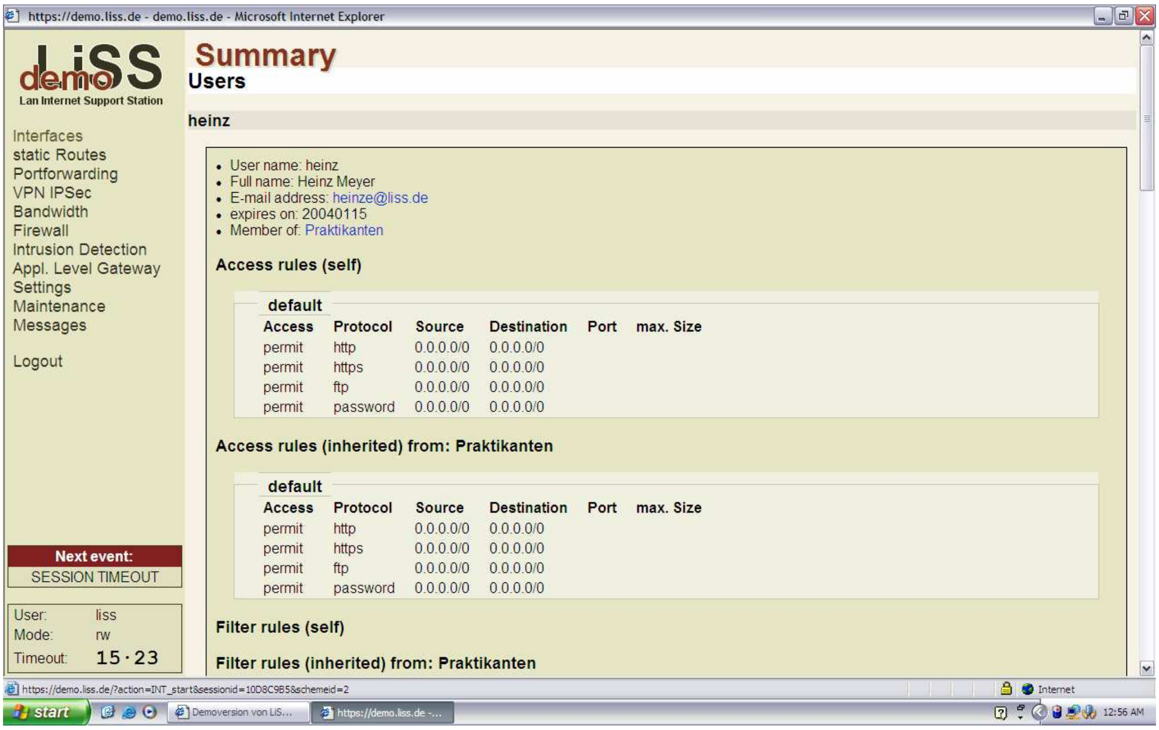
Destination: / 255.255.255.255 :

Protocol: http permit max. Size:

OK

Next event:
SESSION TIMEOUT

User: liss
Mode: rw
Timeout: 15:53



User: liss
Mode: rw
Timeout: 15·23

Summary

Users

heinz

- User name: heinz
- Full name: Heinz Meyer
- E-mail address: heinze@liss.de
- expires on: 20040115
- Member of: Praktikanten

Access rules (self)

default

Access	Protocol	Source	Destination	Port	max. Size
permit	http	0.0.0.0/0	0.0.0.0/0		
permit	https	0.0.0.0/0	0.0.0.0/0		
permit	ftp	0.0.0.0/0	0.0.0.0/0		
permit	password	0.0.0.0/0	0.0.0.0/0		

Access rules (inherited) from: Praktikanten

default

Access	Protocol	Source	Destination	Port	max. Size
permit	http	0.0.0.0/0	0.0.0.0/0		
permit	https	0.0.0.0/0	0.0.0.0/0		
permit	ftp	0.0.0.0/0	0.0.0.0/0		
permit	password	0.0.0.0/0	0.0.0.0/0		

Filter rules (self)

Filter rules (inherited) from: Praktikanten

https://demo.liss.de/?action=INT_start&sessionid=10D8C9B5&schemeid=2

	Snort	**DansGuardian**	**Liss II Secure Gateway**	**WebSense**	**NetNanny**
Network Traffic Monitoring					
Real Time Monitoring	✓		✓	✓	✓
Access Restriction & Content Filtering					
Black List Blocking		✓	✓	✓	✓
White List Blocking		✓	✓	✓	✓
Time-based Blocking					✓
Dynamic Filtering		✓	✓	✓	✓
File Extension Filtering		✓		✓	
MIME Filtering		✓			
PICS Filtering		✓			
Advertisement Filtering		✓			✓
Download Restriction		✓	✓	✓	✓
Post Limiting (Upload Restriction)		✓			
Logging					
Network Traffic Logging	✓	✓	✓	✓	✓
Administrative Action Logging					
Logging Format					
CSV Format		✓			
Tcpdump Format	✓				
Management					
Window-based Interface			✓	✓	✓
Web-based Interface			✓	✓	
Rule-based Management	✓	✓		✓	
Statistics					✓
Administrator Authentication	✓	✓	✓	✓	✓
Automatic Invocation	✓				
Logging in Stealth Mode		✓			
Administrator Reporting (Alerts)	✓		✓	✓	✓

	Snort	**DansGuardian**	**Liss II Secure Gateway**	**WebSense**	**NetNanny**
Caching	✓				
Multiple-mode Design **(Ability to Run in Different Modes)**	✓	✓			
Network Intrusion					
Detection	✓		✓	✓	
Prevention	✓			✓	
Spyware Control				✓	

2.2 Meetings with Customers

- *Have you ever used an application level gateway?*

ETC-IS:	Yes.
TR.NET:	Yes.
Aydın Yazılım:	Yes.

- *If so, what is the name of the program?*

ETC-IS:

We have used Fort Knox Policy, but due to significant decrease in system performance, we have switched to Cisco PIX 515E (reference) Syslog Server.

TR.NET:

We used to use WebSense, but now we do not continue with such a program.

Aydın Yazılım:

We would rather not mention the name of our program due to security concerns.

- *What are the features of the program that you use? What are the hardware and software specifications required by the program?*

ETC-IS:

Cisco PIX has the following facilities:

- A web interface
- IP logging
- Log-based monitoring
- Content filtering
- Network address translation (NAT)
- Blocking specified Instant Messaging applications
- Console based user interface
- Protocol / port control
- Visual alerts notifying attacks

For Fort Knox Policy, which was an NT-based, self-standing hardware firewall, we can list:

- It provides instant monitoring
- It comes with an easy-to-use graphical user interface
- It can act as a server, since the DMC port is available.
- It requires regular hardware updates.
- It decreases the performance of the system because it uses its integrated RAM.
- It is vulnerable to deadlocks (which happens quite frequently).
- It notifies the administrator via audio alerts in case of intrusion detection.

TR.NET:

We used WebSense in one of our departments, because:

- It has network traffic monitoring capability.
- It controls web access, providing “allow”, “block” and “continue” options.
- It provides prevention from spyware and malicious content.
- It has network address translation facility.
- It supports real-time updates for blocking list.

Aydın Yazılım:

We are using a program that is specialized on intrusion detection and prevention, that has:

- Text-based logging feature
- Packet filtering
- Detailed and statistical analysis of intrusions
- User-based control on download bandwidth
- Caching

Also we are running another program which supports content filtering only.

- *Which features are essential in such a program?*

ETC-IS:

- It should not slow the connection while it is providing security.
- It must have a logging feature.
- It must have a reporting mechanism to the administrator.
- Unblocking must be provided as well as blocking.
- It must provide web-based control.
- It must send automated alerts to the administrator in case of attacks.
- It must be monitoring download and upload statistics.

TR.NET:

- Spam mail filtering is very crucial. (However, that should not affect normal mail traffic.) For instance, this system can make use of a blocking list.
- The program may rely mainly on its own predefined rules. It is not necessary for the users to manipulate it more than a certain extend.
- Statistical data must be kept track of.
- It should be maintained and updated automatically through Internet.
- Speed is an important issue.

Aydın Yazılım:

- Fallacies in TCP/UDP must be detected and compensated.
- Analyzing the attacks, to separate marginal attacks from persistent ones is essential.
- Intrusion prevention must be added.
- Caching mechanism should be supported for keeping up with the connection speed.

- *What kind of difficulties are encountered during installation and management?*

ETC-IS:

- We have not encountered any type of difficulties.

TR.NET:

➢ We have not encountered any type of difficulties.

Aydın Yazılım:

➢ We have not encountered any type of difficulties.

➢ *Is the program maintainable?*

ETC-IS:

It is maintainable.

TR.NET:

Yes, it is.

➢ *Do you have the chance of extending the program according to your needs? Which features can be specified by the administrator?*

ETC-IS:

We are satisfied with the default settings of the program. Nevertheless, the program supports extensions such as adding URL's to the blocking list, and specifying keywords for content filtering.

TR.NET:

We have not had such a need.

Aydın Yazılım:

We have used the SDK of the program to extend its intrusion prevention facilities; and added virus protection capabilities.

➢ *How did you learn about the program?*

ETC-IS:

We are using the same program with our leading company in America.

TR.NET:

We were informed about the program by a demonstration made by its developers.

Aydın Yazılım:

We have carried out market research for the program that meets our needs to the greatest extend.

- *Is there any blocking mechanism in the system? If so, what are the advantages of this?*

ETC-IS:

There is a blocking mechanism. We can control the internet access of our employees with the help of this feature.

TR.NET:

Yes, there is. It is useful for web access control.

Aydın Yazılım:

We use blocking for download bandwidth restrictions.

- *Does the program support a web-based interface for the administrator? If not, do you think that it is necessary?*

ETC-IS:

Yes, it supports.

TR.NET:

Yes, it does.

Aydın Yazılım:

We do not control the program through the Internet.

- *Is there any reporting mechanism for the administrator? If not, do you think it would be necessary for the ease of usage?*

ETC-IS:

Although the administrator can view the logs, a separate reporting mechanism is not supplied. This would be of great use.

TR.NET:

There is not such a mechanism. We have not felt such a need thus far.

Aydın Yazılım:

There is such a mechanism. It provides statistical data.

- *Does it have a virus blocking module?*

ETC-IS:

It does not include such a module. We do not find it necessary because there are anti-virus programs that are dedicated to this task.

TR.NET:

No, it does not. We are using anti-virus programs instead. We would not expect such a facility since viruses are not packet specific.

Aydın Yazılım:

Sure. This module is extended by us.

- *Is there an administration authentication support?*

ETC-IS:

Yes, the administrator should make an NT-based password change per forty days. The program alerts the user for password change near the time of change. The password should not be less than six characters.

TR.NET:

Yes, administrator authentication should be developed with an encryption/decryption algorithm.

Aydın Yazılım:

Yes.

3. PROJECT SCHEDULE

3.1 Work Breakdown Structure (WBS)

1.0 NetCheck Project Proposal
- 1.1 Analysis
  - 1.1.1 Requirement Analysis
    - 1.1.1.1 Literature Survey
    - 1.1.1.2 Communication
      - 1.1.1.2.1 Communication with Developers
      - 1.1.1.2.2 Communication with Users
  - 1.1.2 Risk Analysis
  - 1.1.3 Project Scheduling and Tracking
  - 1.1.4 Project Quality Plan
- 1.2 Gathering Background Information
  - 1.2.1 Network Architecture
  - 1.2.2 TCP/IP Protocols
  - 1.2.3 Network Security
  - 1.2.4 Linux Networking
  - 1.2.5 Web Programming
  - 1.2.6 Data Mining
- 1.3 Design
  - 1.3.1 Design of Filtering Module
  - 1.3.2 Design of Monitoring Module
  - 1.3.3 Design of Blocking & Restriction Module
  - 1.3.4 Design of Logging Module
  - 1.3.5 Design of Management Module
    - 1.3.5.1 Design of Web Page Interface
    - 1.3.5.2 Design of Statistics Interface
  - 1.3.6 Design of Database
  - 1.3.7 Design of Different Running Modes
- 1.4 Prototype Development
  - 1.4.1 Coding Prototype
- 1.5 Implementation
  - 1.5.1 Implementation of Filtering Module
  - 1.5.2 Implementation of Monitoring Module
  - 1.5.3 Implementation of Blocking & Restriction Module
  - 1.5.4 Implementation of Logging Module
  - 1.5.5 Implementation of Management Module
    - 1.5.5.1 Implementation of Web Page Interface
    - 1.5.5.2 Implementation of Statistics Interface
  - 1.5.6 Implementation of Database
  - 1.5.7 Implementation of Different Running Modes
- 1.6 Testing
  - 1.6.1 Testing of Filtering Module
  - 1.6.2 Testing of Monitoring Module
  - 1.6.3 Testing of Blocking & Restriction Module
  - 1.6.4 Testing of Logging Module
  - 1.6.5 Testing of Management Module
  - 1.6.6 Testing of Different Running Modes
- 1.7 Project Finalization
  - 1.7.1 Application Setup Development
  - 1.7.2 User Manual Preparation

3.2 Gannt Chart

Our Gannt chart can be seen under Appendix-A.

4. TEAM ORGANIZATION

4.1 Team Structure

We have decided our team structure to be Controlled Decentralized (CD). This is due to the following facts:

- We need a team leader who will be responsible for the coordination of team meetings and tasks to be performed.
- We will be solving our problems as a group, not as individuals
- We need horizontal communication among team members.

4.2 Role of Team Members

The following table shows the responsibilities of all team members:

	Gülşah	**Neslihan**	**Hande**	**Filiz**	**Kezban**
Project Manager	√				
Project Schedule Coordinator	√	√	√		
Project Archive Keeper			√	√	√
Contact People		√		√	√

4.3 Ground Rules

The followings are the ground rules to be obeyed by all of the members of our team:

- Members who have an excuse for a weekly meeting will inform the project manager until 12:00 on the day of the meeting.
- Member’s late arrivals for the meetings are prohibited if there is no excuse for the lateness.
- All members will be responsible for checking the project’s mail group every day.
- All actions and tasks performed for the project by the team members will be posted to the project mail group.
- All members are responsible for meeting the deadlines that are determined by the whole team.

5. PROCESS MODEL

Due to the structure of our project we have chosen linear sequential model as our main methodology and the need for better risk management forced us to use a hybrid of spiral model.

For the application of the linear sequential model firstly we have to define the requirements of the project absolutely, which we believe we have completed in an appropriate manner. We have arranged meetings with the software companies that are potential customers for our project. In addition we have analyzed many software products aiming the similar functionality as our project. Our model proceeds with design, implementation and testing phases.

Employing spiral model through the linear sequential model will allow us to identify and implement higher priority features of the project first and then add more features into it. By this method we should get more insight into the project and analyze risks more appropriately.

6. REQUIREMENT ANALYSIS

6.1 System Requirements

Requirements for users who will purchase our final product are explained in sections Software Requirements and Hardware Requirements. The tools we are planning to use in development are explained in the Tools part.

6.1.1 Software Requirements

The machine to be used should provide the following facilities on a Linux operating system:

- A web server (e.g. Apache)
- PostgreSQL as the Database Management System
- A firewall (Iptables)
- A web browser for the administrative purposes
- GNU C++

6.1.2 Hardware Requirements

For the convenience of users, the following properties are expected:

- Minimum 512 MB RAM
- Minimum 5 GB of free disk space, for database storage
- A Pentium IV processor
- Minimum two network interface cards

6.1.3 Tools

We will make use of the following tools for developing our project:

- Linux operating system as the development platform
- C++ programming language
- GNU C++
- PostgreSQL Database Management System
- PHP and Apache web server or Java Virtual Environment, JSP and Tomcat

6.2 Implementation-Related Requirements

In this section, we are intending to mention the requirements related to internal structure of the system. We have aimed to provide facilities that would enhance our system in terms of functional, interface, security and performance measures.

Functional Requirements

We have used the outcomes of our literature survey and customer meetings to decide on the functional requirements of our system. Our system consists of basically two modules: the gateway module that regulates the network traffic, and the configuration module by which the administrator changes the system's default settings.

Main functionalities that are expected from such a system are;

- Connecting the local network of the organization to outside
- Filtering content of incoming packets from black words
- Restricting access to some predefined sites
- Preventing, detecting and stopping intrusion attempts aiming to take charge of internal system
- Taking logs of network traffic for later inspections of the administrators
- Calculating statistics for administrators to gain insight about the network
- Caching most frequently visited URLs
- Monitoring the current status of the network, and enabling instant interruption of the administrator
- Ability to manage access and download rights for groups of users and/or individual users

We are also intending to provide the following extra features for the convenience of users:

- Different running modes for users with different needs
- Alerts via e-mail or SMS to the administrator, in case of emergency, such as an intrusion detected
- Reporting mechanism that informs the administrator about the latest changes made to the system
- A trash deleting mechanism, that saves the deleted functionalities on the system for further usage, and restores them back if needed later.

Interface Requirements

The interface of a system is the main facility that decides the usability of the system. With this vision in mind, we intend to design a web interface so that the system will be accessible from anywhere. Our interface will be user-friendly and easy to understand. The details of interface requirements are explored in our Use Case Diagrams.

Security Requirements

Security being the main purpose of our system, we must try for highest security features in the implementation of it. These features include:

- Administrator details should be protected through an efficient encryption / decryption algorithm.
- Local servers should be defended from outside attacks via a through inspection of HTML forms, which should be implemented by validating the compatibility of incoming and outgoing forms.
- In order to preserve our local network from intrusions coming from outside, we will log the requests from a non-local network.
- Intrusions like SQL-injection or cross-site scripting should be avoided by parsing the data in the incoming packets and inspecting attack patterns.
- We are aiming to protect our log files by defining special protection rules for them.
- We will also inspect outgoing data for confidential information that should not leave local network.
- We are going to use Linux Operating System as the platform of development, and the running platform of the final product. This is due to security issues, because of the more sophisticated security facilities of Linux.

Performance Requirements

Since real-time tracking of all incoming and outgoing packets is the main issue of our program, speed efficiency is the primary concern. For a fast implementation, the programming language should be selected accordingly. A language closer to the machine level should be preferred.

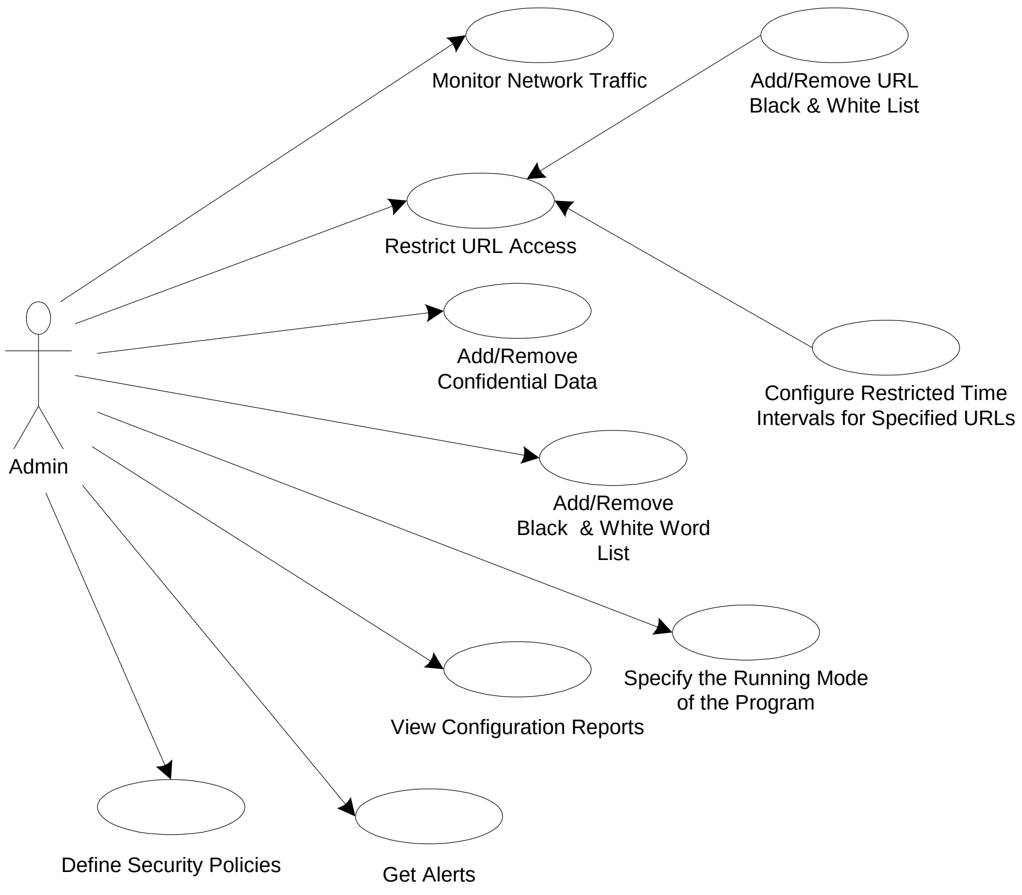
Another issue is the implementation of rule definitions. We are considering using regular expressions for defining the rules, because we think parsing the packets and detecting malicious content would be faster by this way. However, this issue is due to further consideration.

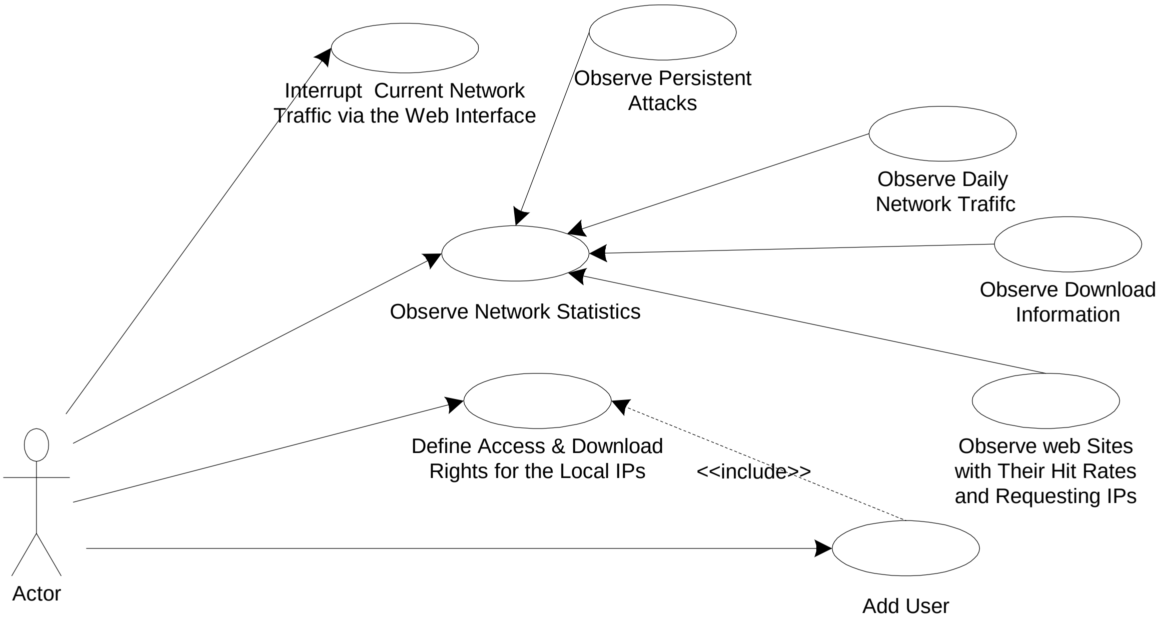
We have also considered how to make our database access more efficient. We will keep three tables for tables that are subject to frequent insertions, such as the Log table. These tables will hold entries for respectively, daily, weekly and monthly insertion. At the end of each time period, the table on the upper level will be dumped into lower level. This will speed up database access, because it is highly dependent on the size of tables.

6.3 Use-Case Analysis

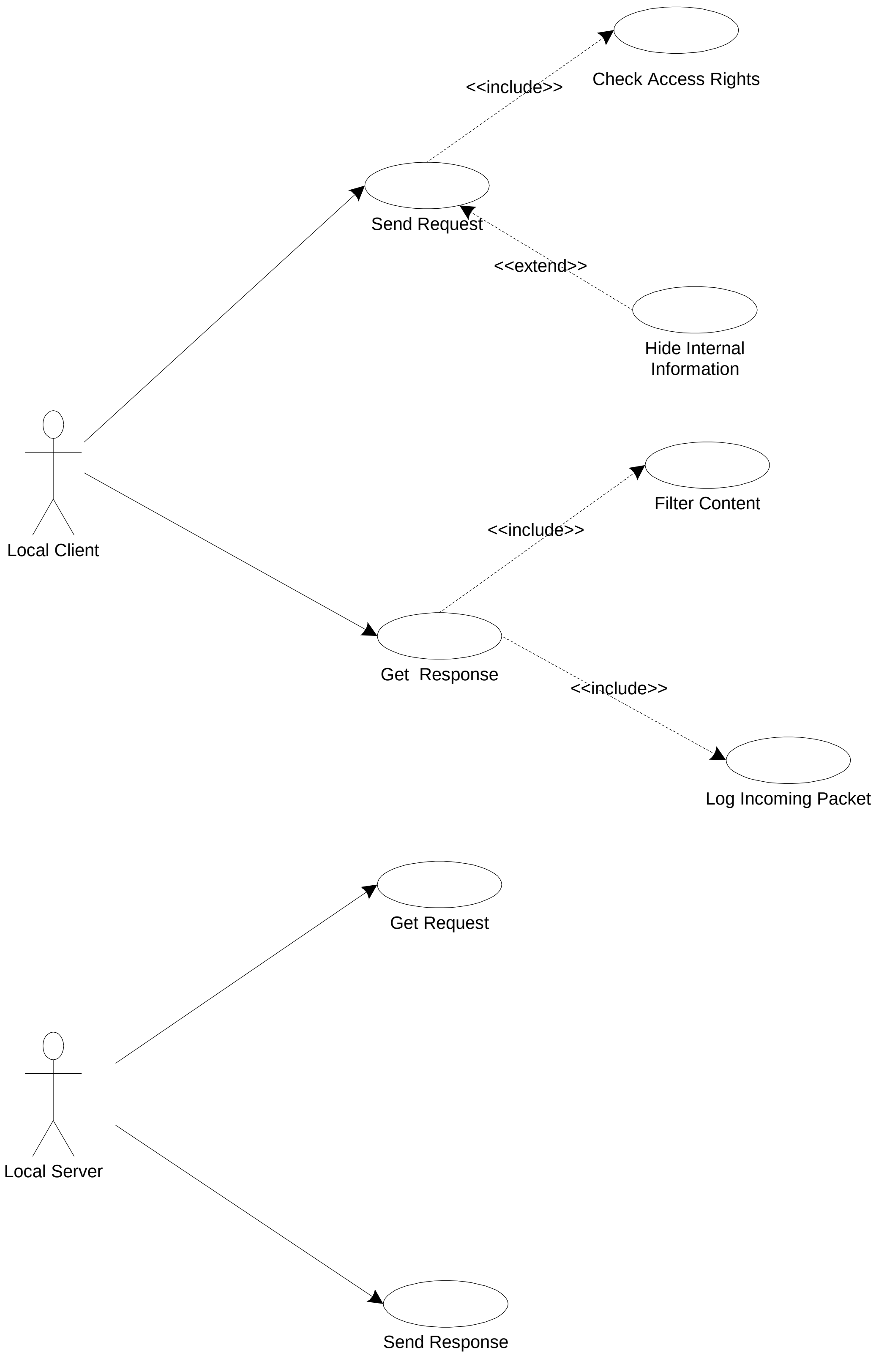
6.3.1 Use-Case Diagrams

Use Case for Administrator

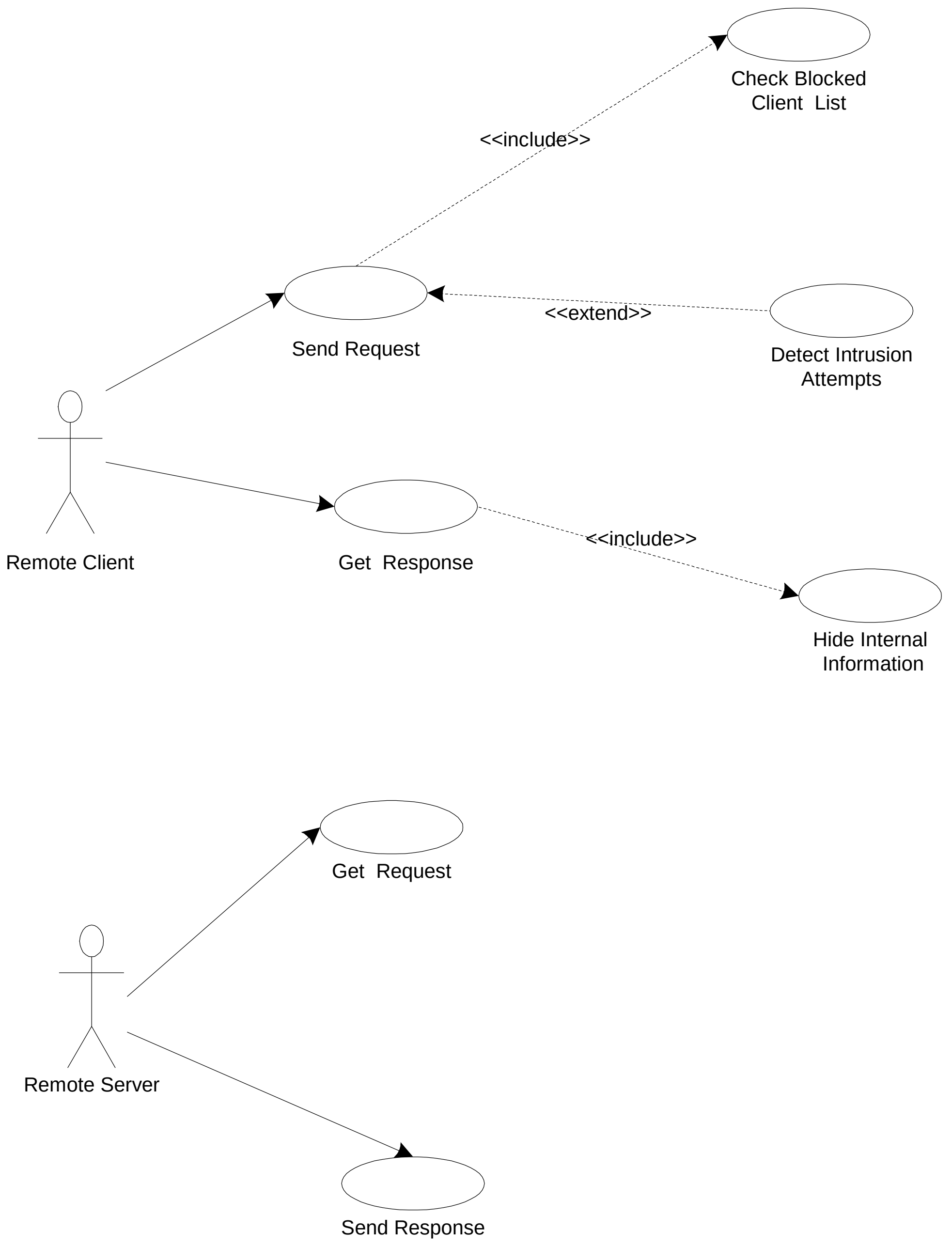

Monitor Network Traffic
Add/Remove URL Black & White List
Restrict URL Access
Add/Remove Confidential Data
Configure Restricted Time Intervals for Specified URLs
Admin
Add/Remove Black & White Word List
Specify the Running Mode of the Program
View Configuration Reports
Define Security Policies
Get Alerts


Interrupt Current Network
Traffic via the Web Interface
Observe Persistent
Attacks
Observe Daily
Network Trafifc
Observe Download
Information
Observe Network Statistics
Define Access & Download
Rights for the Local IPs
<<include>>
Observe web Sites
with Their Hit Rates
and Requesting IPs
Actor
Add User

Use Case for Local Client and Server



Use Case for Remote Client and Server



6.3.2 Use-Case Scenarios

Admin scenarios

Monitor Network Traffic

Basic Flow of Control: Administrator can view the current network traffic through a web interface. Details such as Source IP, Destination IP, packet size and time information will be displayed.

Restrict URL Access

Basic Flow of Control: Administrator may *Add/Remove URL Black and White List*, *Configure Restricted Time Intervals for Specified URLs*.

Add/Remove URL Black and White List

Basic Flow of Control: Administrator can configure URL Black List Table, which contains the URLs that are forbidden to be accessed by local clients, or the URL White List Table, which contains URLs that will not be blocked in any case.

Configure Restricted Time Intervals for Specified URLs

Basic Flow of Control: Administrator may specify time limitations for accessing some sites. For instance, newspaper sites may be forbidden during the morning, when network traffic is especially busy.

Define Access and Download Rights for Local IPs

Basic Flow of Control: Download and access rights can be granted to user groups and/or individual users. Download rights indicate the maximum packet size limit that can be downloaded in a day. Access rights define which entries of black URL / word list apply to the user / user group.

Add/Remove Black and White Word List

Basic Flow of Control: Administrator can configure Black Word List Table, which contains the word that will be used in content filtering, or the White Word List Table, which contains words that will not be filtered in any case.

Observe Network Statistics

Basic Flow of Control: Administrator may *Observe Web Sites with Their Hit Rates and Requesting IPs*, *Observe Download Information*, *Observe Daily Network Traffic* or *Observe Persistent Attacks.*

Define Security Policies

Basic Flow of Control: Administrator may define rules to help detection of intrusions.

View Configuration Reports

Basic Flow of Control: Administrator can view the reports about latest configurations made to the system.

Get Alerts

Basic Flow of Control: Administrator will be alerted in case of an intrusion attempt.

Interrupt Current Network Traffic via the Web Interface

Basic Flow of Control: When viewing the network traffic, the administrator may want to interrupt certain connections, if network seems to be too busy.

Add User

Basic Flow of Control: New users are added via web interface. *include(Define Access and Download Rights for Local IPs).*

Specify the Running Mode

Basic Flow of Control: Administrator may choose between different modes of the program, which offer different functionalities.

Local Client Scenarios

Send Request

Basic Flow of Control: Local client sends a request then *include(Check Access Rights).* If the request does not violate the access rights then *Hide Internal Information.*

Alternative Flow of Control: If the local client's request fail to satisfy *include(Check Access Rights)* then an error message will be displayed to the client.

Check Access Rights

Basic Flow of Control: Request packet will be first inspected to control if the destination address is in the white list. If the destination IP is in the white list, request will be served without any further considerations. However in case when the destination address does not exist in the white list, source address will be taken into account to check for restrictions on the black list table. If no violations are detected permission will be granted to the user.

Hide Internal Information

Basic Flow of Control: Company oriented security policies will be detected in order to prevent the private data from being sent out and NAT (Network Address Translation) will be implemented to local client IPs.

Get Response

Basic Flow of Control: Local client gets the response from the remote server. *include(Filter Content) and include (Log Incoming Packet).*

Alternative Flow of Control: If the incoming packet fail to satisfy *include(Filter Content)* then an error message will be displayed to the client

Remote Client Scenarios

Filter Content

Basic Flow of Control: Incoming packet will be first inspected to control if it contains any of the words in the Black Word List. If a suspicious word also exists in the White Word List, it will be allowed.

Alternative Flow of Control: If the packet contains malicious words, and these words do not exist in the White Word List, the packet will not be allowed. An error message will be displayed to the client.

Log Incoming Packet

Basic Flow of Control: Content of the incoming packet is logged for further usage.

Check Blocked Client List

Basic Flow of Control: When a remote client sends a request to the local server, the client will be checked whether it has been blocked due to an intrusion attempt before. If not, it will be allowed to get a response.

Alternative Flow of Control: If the client has been blocked, it will not be able to get a response from the local server.

Detect Intrusion Attempts

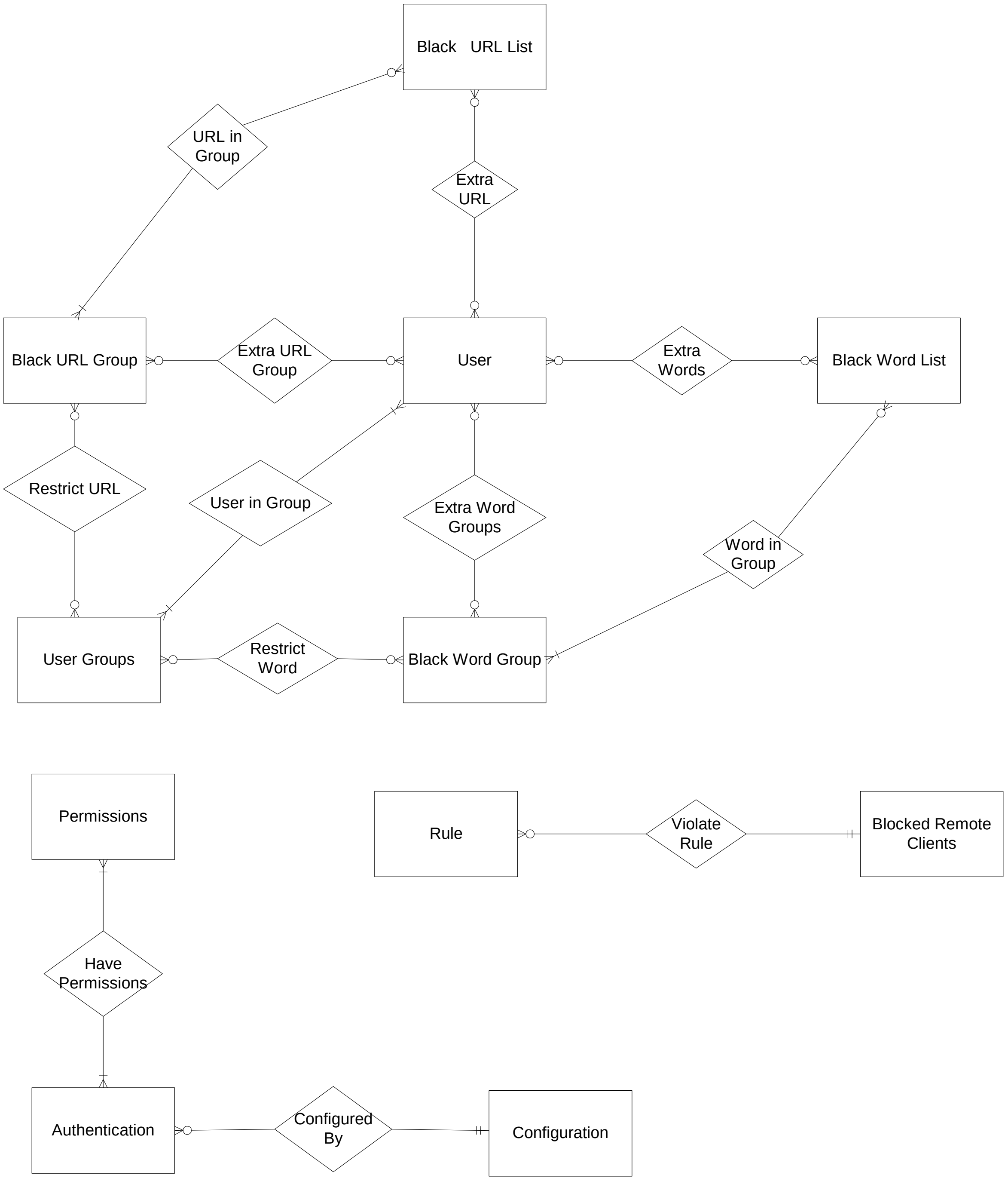
Basic Flow of Control: Incoming request packet will be inspected to control if it violates the rules specifying possible intrusion patterns. If not, the client will be allowed to get a response from the local server.

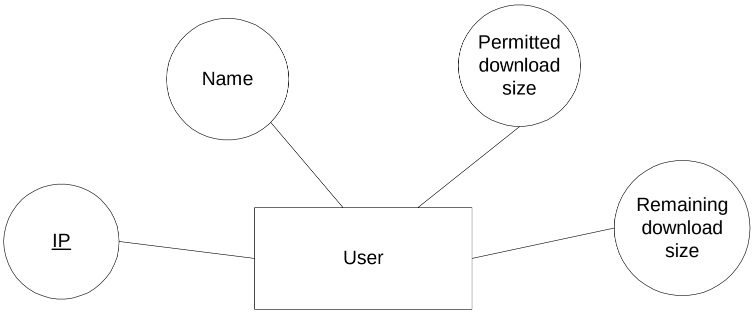
Alternative Flow of Control: If the packet violates with one of the rules, then the client will not be allowed, and it will be blocked for further access.

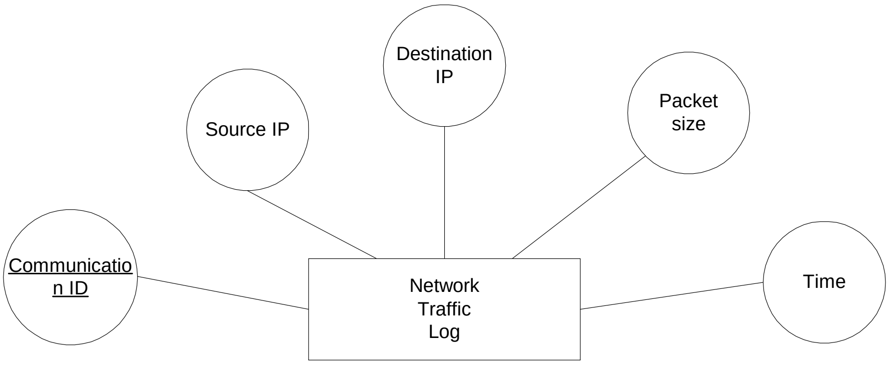
7. MODELING

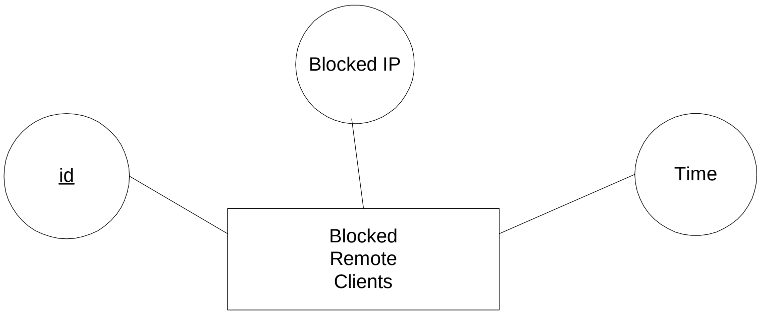
7.1 Data Modeling

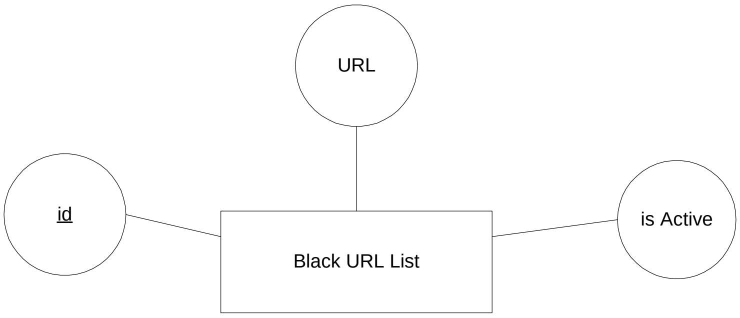
7.1.1 Entity-Relationship (ER) Diagrams

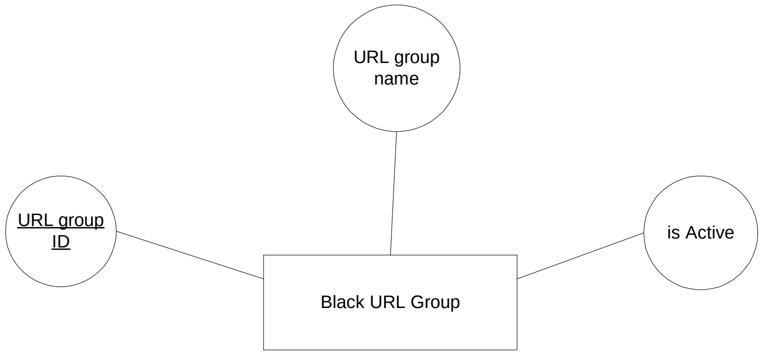


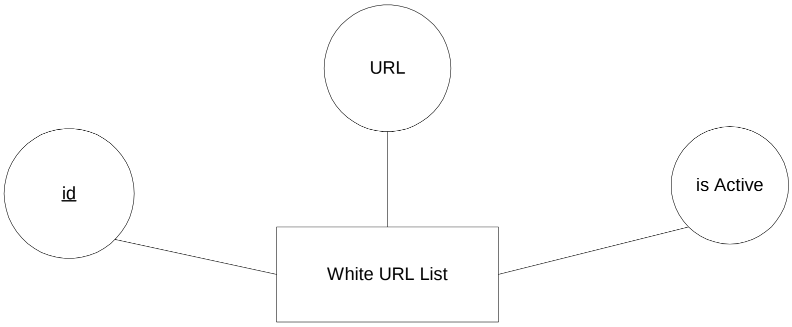

Name
Permitted download size
IP
User
Remaining download size

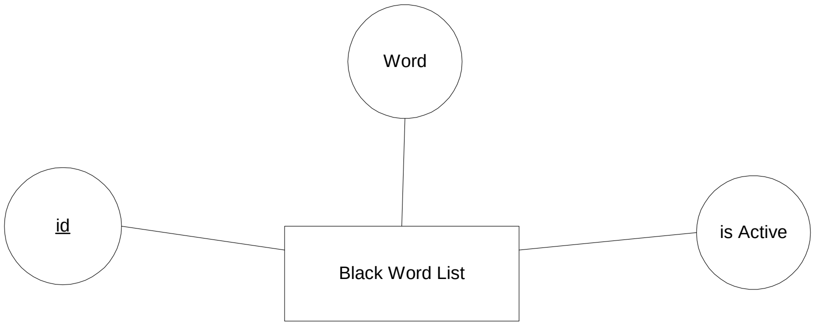

Destination IP
Source IP
Packet size
Communication ID
Network Traffic Log
Time

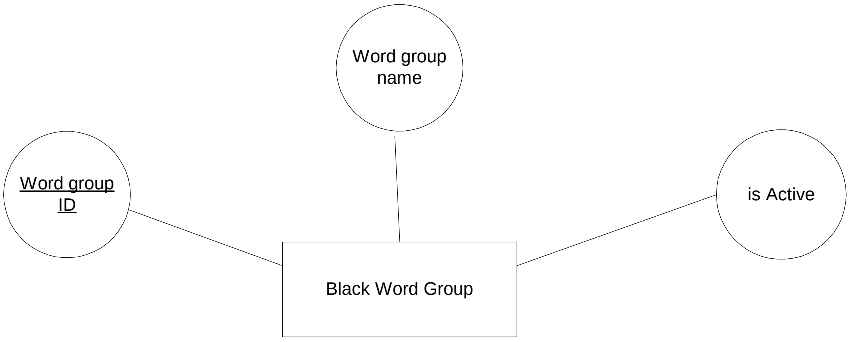

Blocked IP
id
Time
Blocked Remote Clients

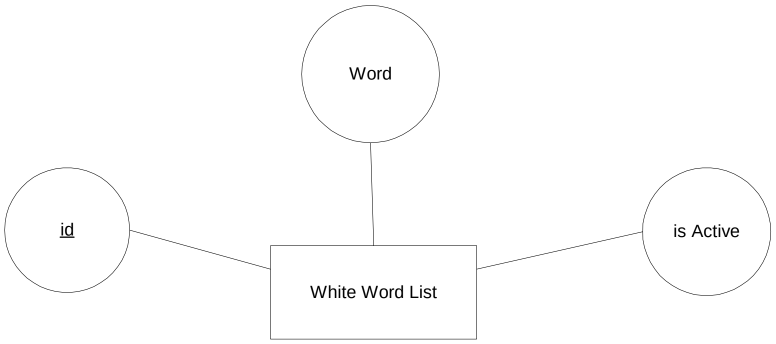

URL
id
is Active
Black URL List

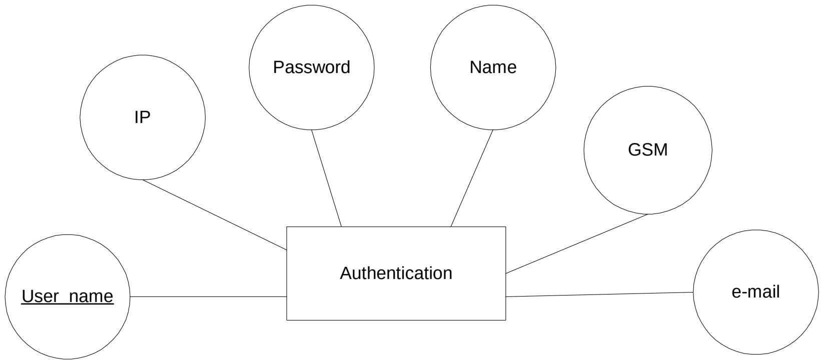

URL group name
URL group ID
is Active
Black URL Group

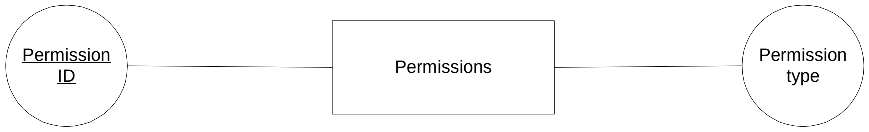

URL
id
is Active
White URL List

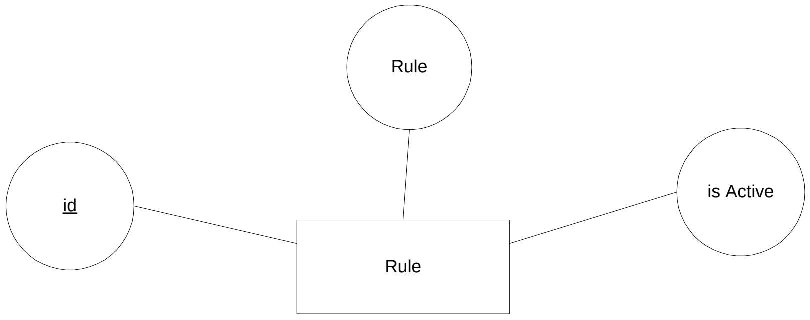

Word
id
is Active
Black Word List

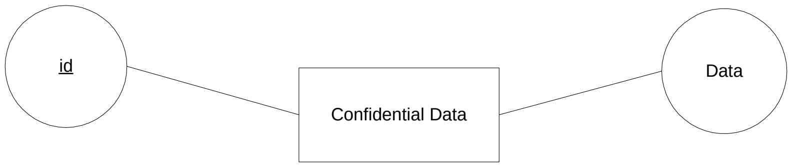

Word group name
Word group ID
is Active
Black Word Group

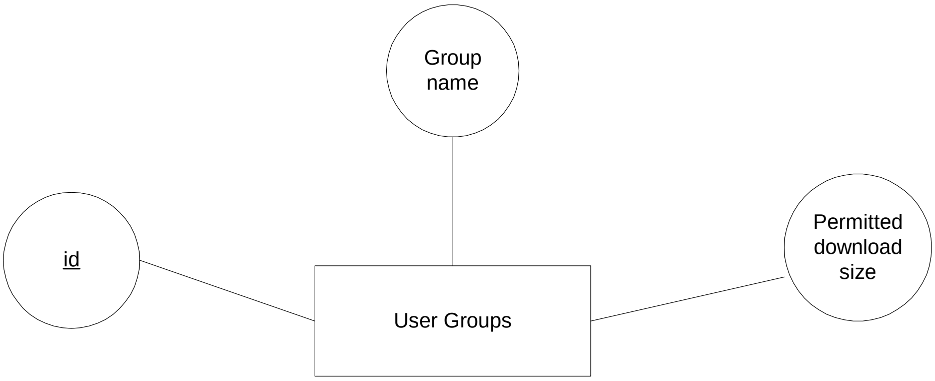

Word
id
is Active
White Word List

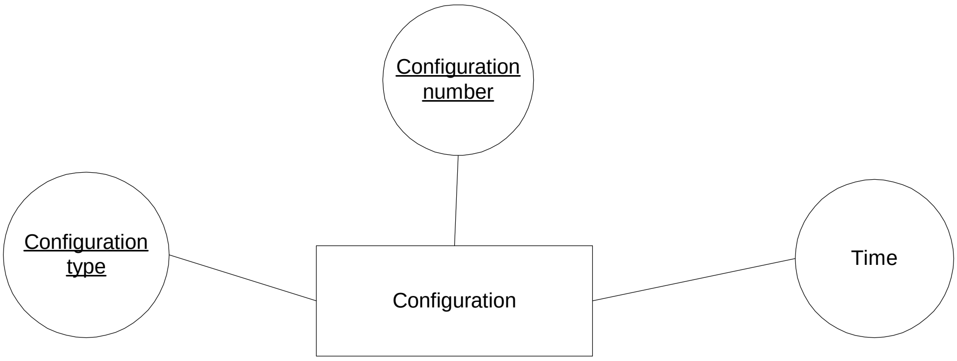

Password
Name
IP
GSM
Authentication
User_name
e-mail

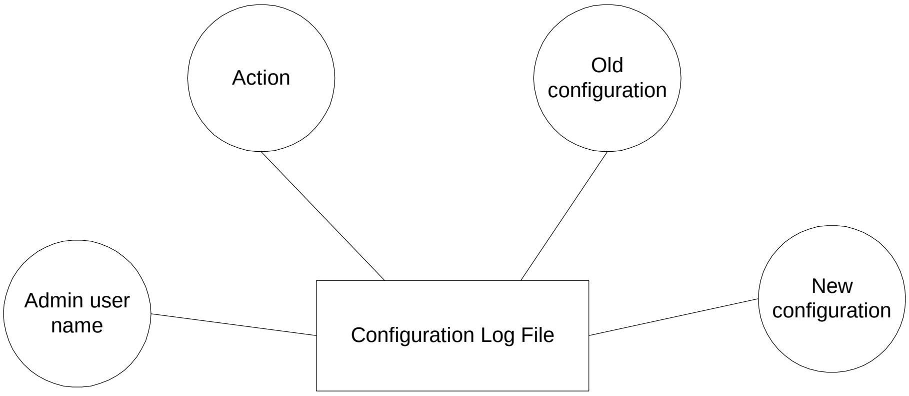

Permission ID
Permissions
Permission type


Rule
id
is Active
Rule


id
Data
Confidential Data


Group name
id
Permitted download size
User Groups


Configuration number
Configuration type
Time
Configuration


Action
Old configuration
Admin user name
New configuration
Configuration Log File

7.1.2 Data Dictionary

Name	Black Word List
Where used / How used	Content Filtering Module 3.1(input) Restriction Configuration Module 1.4 (output)
Description	This table will hold the black words that should be filtered from a packet before it is served to client. *Field names:* *id* : AutoNumber *word* : string *is active* : integer (*is active* is 1 if the entry is in use for content filtering at the moment, and 0 if the administrator has temporarily deactivated it.)
Format	A table in database

Name	Black Word Group
Where used / How used	Content Filtering Module 3.1(input) Restriction Configuration Module 1.4 (output)
Description	Black Word Group holds the words that represent a group of black words. *Field names:* *word-group id:* AutoNumber *is active:* integer *word-group name:* string (*word-group name* is the definitive label that defines the group, such as “gambling”.)
Format	A table in database

Name	White Word List
Where used / How used	Content Filtering Module 3.1 (input) Restriction Configuration Module 1.4 (output)
Description	White words are the words that should not be filtered in any case for any type of user. *Field names:* *id :* AutoNumber *word :* string *is active :* integer
Format	A table in database

Name	Black URL List
Where used / How used	Blacklist and Whitelist Control Module 2.1(input) Restriction Configuration Module 1.4 (output)
Description	Black URL List keeps the black URLs that should not be served to client. *Field names:* *id :* AutoNumber *URL :* string *is active :* integer
Format	A table in database

Name	Black URL Group
Where used / How used	Blacklist and Whitelist Control Module 2.1(input) Restriction Configuration Module 1.4 (output)
Description	Black URL Group holds the URLs that represent a group of black URLs. *Field names:* *URL-group id:* AutoNumber *URL-group name:* string *is active:* integer
Format	A table in database

Name	White URL List
Where used / How used	Blacklist and Whitelist Control Module 2.1(input) Restriction Configuration Module 1.4 (output)
Description	White URL List keeps the URLs that should not be blocked to any type of user in any case. *Field names:* *id :* AutoNumber *URL :* string *is active :* integer
Format	A table in database

Name	Rule
Where used / How used	Check for Intrusions 2.6 (input) Intrusion Detection Configuration Module 1.6 (output) Check for Intrusions 2.6 (output)
Description	Rule table holds rules that are used for detecting intrusions. *Field names:* *id :* AutoNumber *rule :* string *is active :* integer
Format	A table in database

Name	Confidential Data
Where used / How used	Modification of Internal Client Packets 2.4 (input) Administrative Facilities Module 1.5 (output)
Description	This table holds company related private data that should not be allowed to leak outside. *Field names:* *id :* AutoNumber *data :* string
Format	A table in database

Name	Permissions
Where used / How used	Interact Through Web Interface 1.1
Description	Permissions table lists permissions given to administrator about system configuration. *Field names:* *permission id :* AutoNumber *permission type:* string
Format	A table in database

Name	Authentication
Where used / How used	Interact Through Web Interface 1.1
Description	Authentication table holds information about administrators of the system. *Field names:* *user name :* string *password :* string *IP :* string *full name :* string *e-mail :* string *GSM :* string
Format	A table in database

Name	User
Where used / How used	Restriction Configuration Module 1.4 (output) Blacklist & Whitelist Control Module 2.1 (input)
Description	This table holds information about the users of the system *Field names:* *IP:* string *name:* string *permitted download size:* Numerical data type *remaining download size:* Numerical data type
Format	A table in database

Name	Network Traffic Log
Where used / How used	Monitoring Module 1.2(input)
Description	This table holds data for monitoring network traffic. *Field names:* *communication id:* AutoNumber *source IP:* string *destination IP:* string *packet size:* Numerical data type *time:* Date/Time
Format	A table in database

Name	Blocked Remote Clients
Where used / How used	Check for Blocked Clients 2.5 (input)
Description	This table lists the remote clients that were blocked due to their intend to attack to the system. *Field names:* *id:* AutoNumber *blocked IP:* string *time:* Date/Time
Format	A table in database

Name	User Groups
Where used / How used	Administrative Facilities 1.5 (output)
Description	This table groups users according to their rights in the system. *Field names:* *id :* AutoNumber *group name:* string *permitted download size:* Numerical Data Type
Format	A table in database

Name	Configuration
Where used / How used	Restriction Configuration Module1.4 (output) Administrative Facilities Module 1.5 (output) Intrusion Detection Configuration Module 1.6 (output) Check for Intrusions 2.6 (output)
Description	Configuration table lists all kinds of changes made to system by the administrator. *Field names:* *configuration number:* AutoNumber *configuration type:* string *time:* Date/Time
Format	A table in database

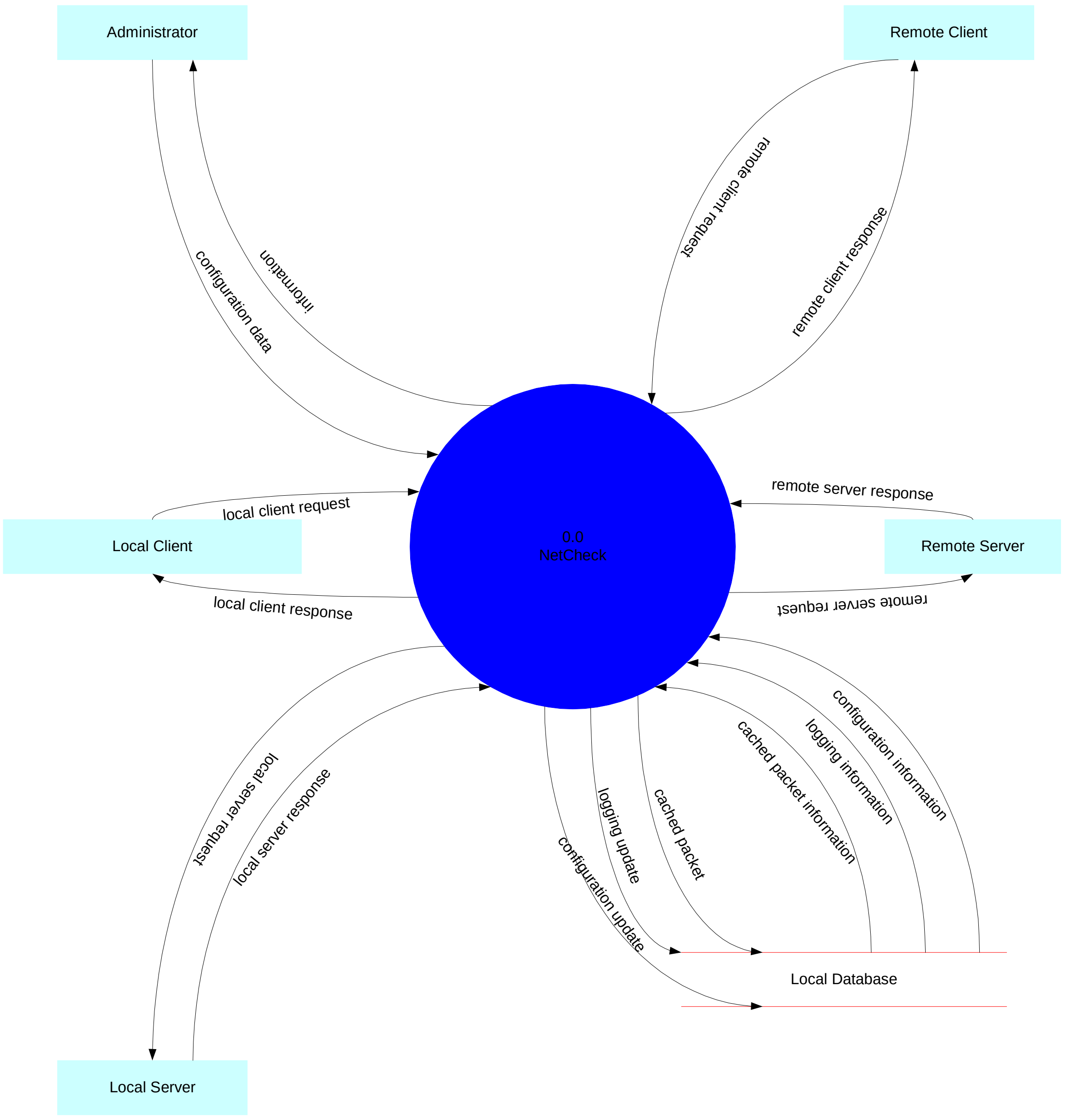
Configuration Log File is a file kept in local disk. Administrator can view this file to see the latest changes made to the system. This file also holds information about deleted entries, which is not kept in the Configuration table in the database.

Name	Configuration Log File
Where used / How used	Restriction Configuration Module1.4 (output) Administrative Facilities Module 1.5 (output) Intrusion Detection Configuration Module 1.6 (output) Check for Intrusions 2.6 (output)
Description	Configuration log file is saved in local disk and it holds the recent configurations made to the system by the administrator. *Field names:* *admin user name:* string *action:* string *old configuration:* string *new configuration:* string
Format	A file in Local Disk

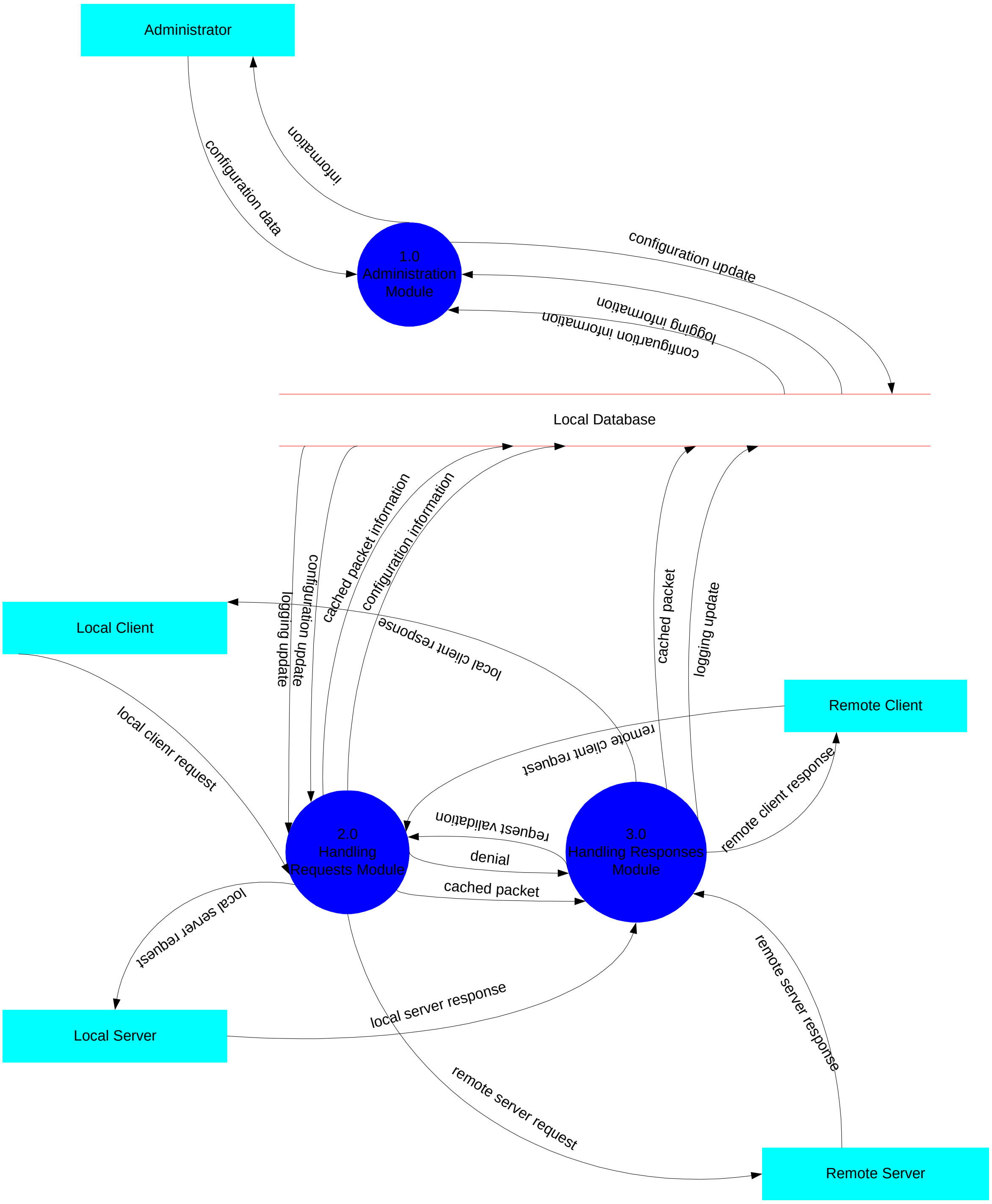
7.2 Functional Modeling

7.2.1 Data Flow Diagrams (DFD)

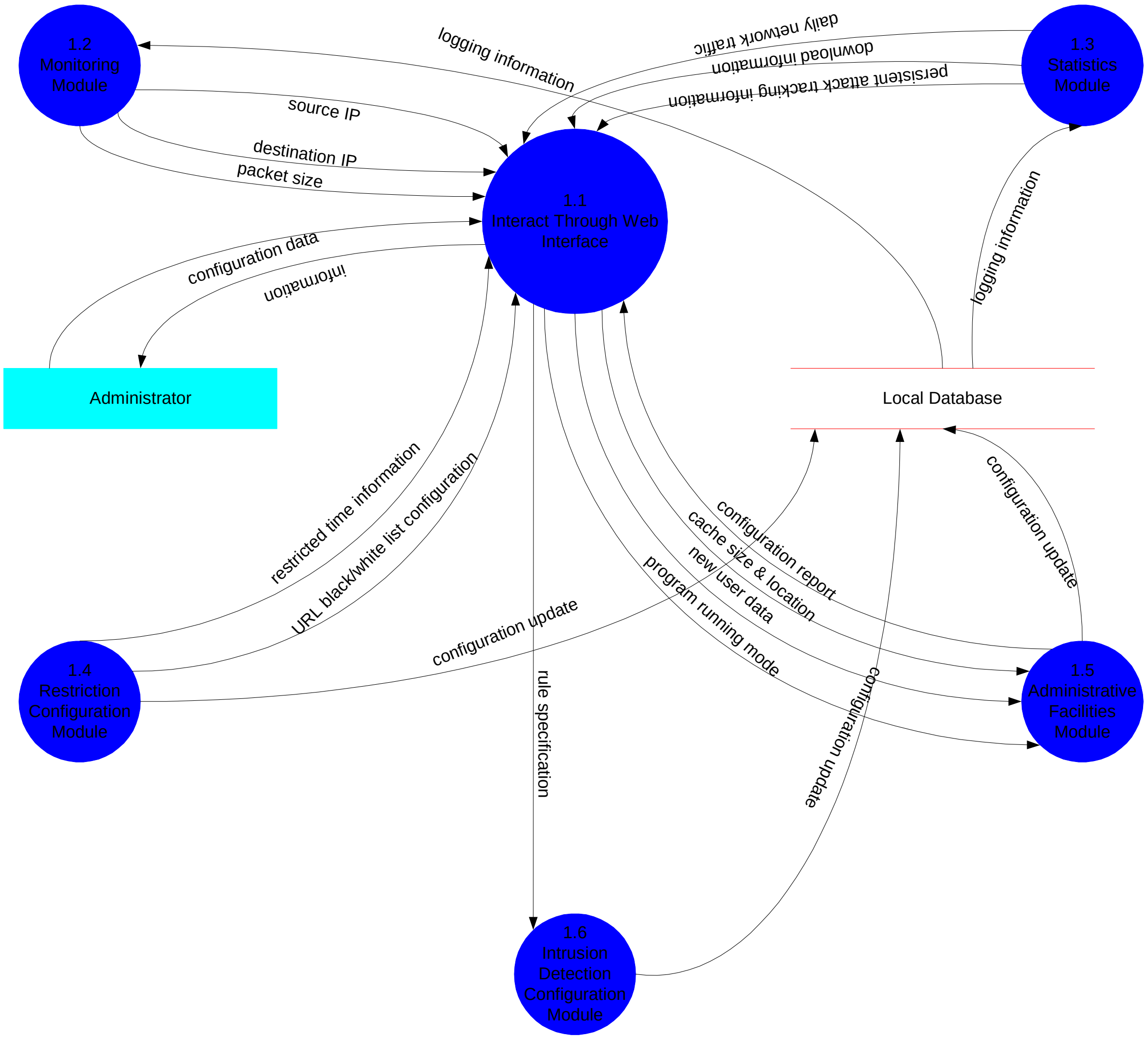
Level 0



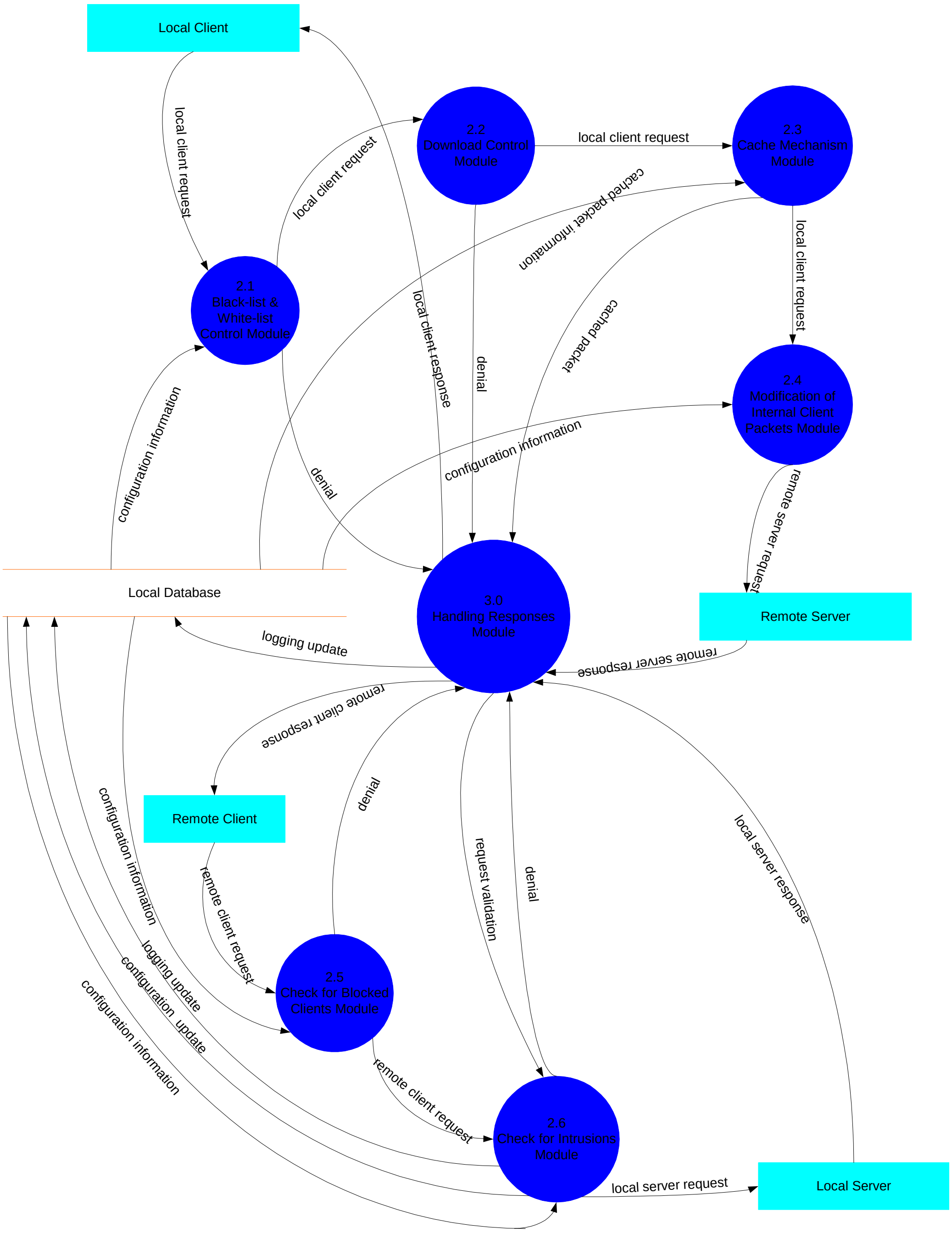
Level 1


Administrator
configuration data
information
1.0
Administration
Module
configuration update
logging information
configuartion information
Local Database
logging update
configuration update
cached packet informatio
configuration information
cached packet
logging update
Local Client
local client response
local clienr request
Remote Client
remote client request
remote client response
2.0
Handling
Requests Module
request validation
denial
cached packet
3.0
Handling Responses
Module
local server request
Local Server
local server response
remote server request
remote server response
Remote Server

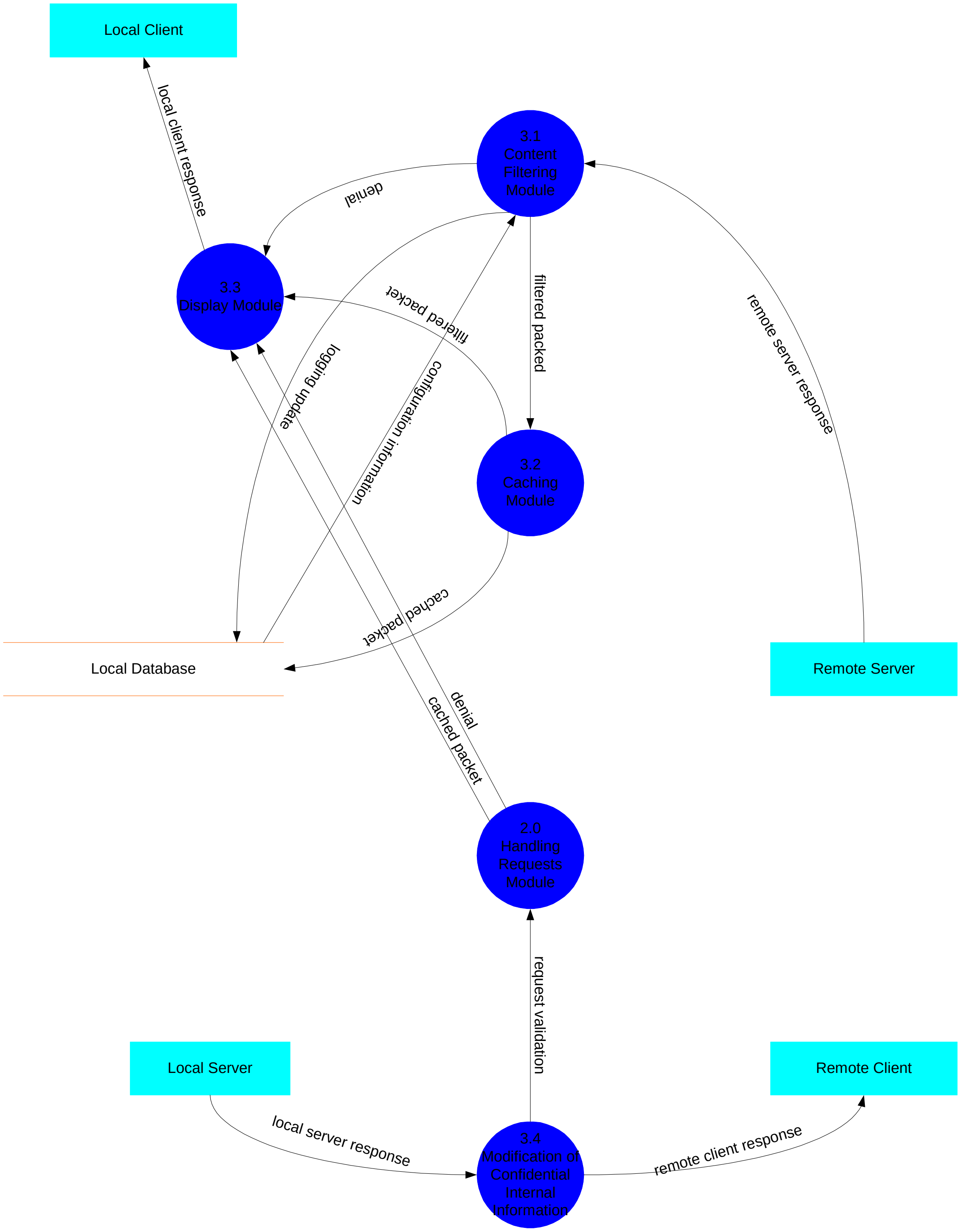
Level 2 – Administration Module



Level 2 – Handling Requests Module



Level 2 – Handling Responses Module



7.2.2 Data Dictionary

Name	Configuration Data
Input to	Interact Through Web Interface 1.1
Output from	Administrator
Description	This data is all configurations that can be made to the system by the administrator, such as additions to the blacklist.
Format	String

Name	Information
Input to	Administrator
Output from	Interact Through Web Interface 1.1
Description	This data is all the feedback that returns from the system to the administrator, such as statistical reports.
Format	String displayed through a graphical user interface

Name	Local Client Request
Input to	Blacklist and Whitelist Control Module 2.1 Download Control 2.2 Cache Mechanism 2.3 Modification of Internal Client Packets 2.4
Output from	Local Client Blacklist and Whitelist Control Module 2.1 Download Control 2.2 Cache Mechanism 2.3
Description	Any packet that is sent by the local client to the system.
Format	TCP/IP packet

Name	Local Client Response
Input to	Local Client
Output from	Display Module 3.3
Description	Any packet that is sent to the local client by the system.
Format	TCP/IP packet

Name	Local Server Request
Input to	Local Server
Output from	Check for Intrusions 2.5
Description	Any “remote client request” packet that is forwarded to the local server by the system.
Format	TCP/IP packet

Name	Local Server Response
Input to	Modification of Confidential Internal Information 3.4
Output from	Local Server
Description	Any packet that is sent by the Local Server to the system, in response to a “Local Server Request”.
Format	TCP/IP packet

Name	Remote Client Request
Input to	Check for Blocked Clients 2.5 Check for Intrusions 2.6
Output from	Remote Client Check for Blocked Clients 2.5
Description	Any packet that is sent by the remote client to the system.
Format	TCP/IP packet

Name	Remote Client Response
Input to	Remote Client
Output from	Modification of Confidential Internal Information 3.4
Description	Any packet that is sent to the remote client by the system.
Format	TCP/IP packet

Name	Remote Server Request
Input to	Remote Server
Output from	Modification of Internal Client Packets 2.4
Description	Any “local client request” packet that is forwarded to the remote server by the system.
Format	TCP/IP packet

Name	Remote Server Response
Input to	Content Filtering Module 3.1
Output from	Remote Server
Description	Any packet that is sent by the Remote Server to the system, in response to a “Remote Server Request”.
Format	TCP/IP packet

Name	Configuration Update
Input to	Local Database
Output from	Check for Intrusions 2.5 Administrative Facilities 1.5 Intrusion Detection Configuration Module 1.6 Restriction Configuration Module 1.4
Description	This data is the all the updates on the Local Database.
Format	SQL query

Name	Configuration Information
Input to	Content Filtering Module 3.1 Blacklist and Whitelist Control Module 2.1 Check for Blocked Clients 2.5 Check for Intrusion 2.6 Modification of Internal Client Packets 2.4
Output from	Local Database
Description	Any data that is requested from the database.
Format	SQL query

Name	Logging Update
Input to	Local Database
Output from	Check for Intrusions 2.5 Content Filtering Module 3.1 Display Module 3.3
Description	Information about packets coming to the system.
Format	Logging format

Name	Logging Information
Input to	Monitoring Module 1.2 Statistics Module 1.3
Output from	Local Database
Description	Feedback to the administrator about the logs.
Format	Logging format

Name	Cached Packet
Input to	Local Database Display Module 3.3
Output from	Caching Module 3.2 Cache Mechanism 2.3
Description	A cached packet can be sent from Caching Module to the Local Database, in case a packet is acquired from the network and decided to be cached by the caching algorithm. Or else, if a packet is requested from the network, and Cache Mechanism decides that is has been cached before, it may send it to the Display Module.
Format	TCP/IP packet

Name	Cache Packet Information
Input to	Cache Mechanism 2.3
Output from	Local Database
Description	The cached packet, if it has been cached before or information about it is not being in the database.
Format	TCP/IP packet or a string

Name	Restricted Time Information
Input to	Restriction Configuration Module 1.4
Output from	Interact Through Web Interface 1.1
Description	The update made by the administrator, about users time restrictions.
Format	String

Name	URL Black/ White List Configuration
Input to	Restriction Configuration Module 1.4
Output from	Interact Through Web Interface 1.1
Description	The update made to the blacklist or whitelist by the administrator.
Format	String

Name	New User Data
Input to	Administrative Facilities Module 1.5
Output from	Interact Through Web Interface 1.1
Description	Information about new users added by the administrator.
Format	String

Name	Program Running Mode
Input to	Administrative Facilities Module 1.5
Output from	Interact Through Web Interface 1.1
Description	The decision about which mode to run the program in, made by the administrator.
Format	String

Name	Configuration Report
Input to	Interact Through Web Interface 1.1
Output from	Administrative Facilities Module 1.5
Description	The feedback to the administrator about the latest changes made by the administrator.
Format	String

Name	Download Information
Input to	Interact Through Web Interface 1.1
Output from	Statistics Module 1.3
Description	Statistics about download of users.
Format	String

Name	Daily Network Traffic
Input to	Interact Through Web Interface 1.1
Output from	Statistics Module 1.3
Description	Statistics about daily network traffic.
Format	String

Name	Persistent Attack Tracking Information
Input to	Interact Through Web Interface 1.1
Output from	Statistics Module 1.3
Description	Statistics about persistent intrusion efforts.
Format	String

Name	Source IP
Input to	Interact Through Web Interface 1.1
Output from	Monitoring Module 1.2
Description	Source IP’s of all packets, monitored by the system.
Format	String

Name	Destination IP
Input to	Interact Through Web Interface 1.1
Output from	Monitoring Module 1.2
Description	Destination IP’s of all packets, monitored by the system.
Format	String

Name	Packet Size
Input to	Interact Through Web Interface 1.1
Output from	Monitoring Module 1.2
Description	Sizes of all packets, monitored by the system.
Format	Numerical data type

Name	Cache Size and Location
Input to	Administrative Facilities Module 1.5
Output from	Interact Through Web Interface 1.1
Description	Update about the size of disk and disk location that is decided by the administrator.
Format	String

Name	Rule Specification
Input to	Interact Though Web Interface 1.1
Output from	Intrusion Detection Configuration Module 1.6
Description	Update of the rules that are used to detect and prevent intrusions, made by the administrator.
Format	String (a regular expression)

Name	Denial
Input to	Display Module 3.3
Output from	Blacklist and Whitelist Control Module 2.1 Download Control 2.2 Check for Blocked Clients 2.5 Check for Intrusions 2.6 Content Filtering Module 3.1
Description	A warning indicating the packet will not be admitted.
Format	String

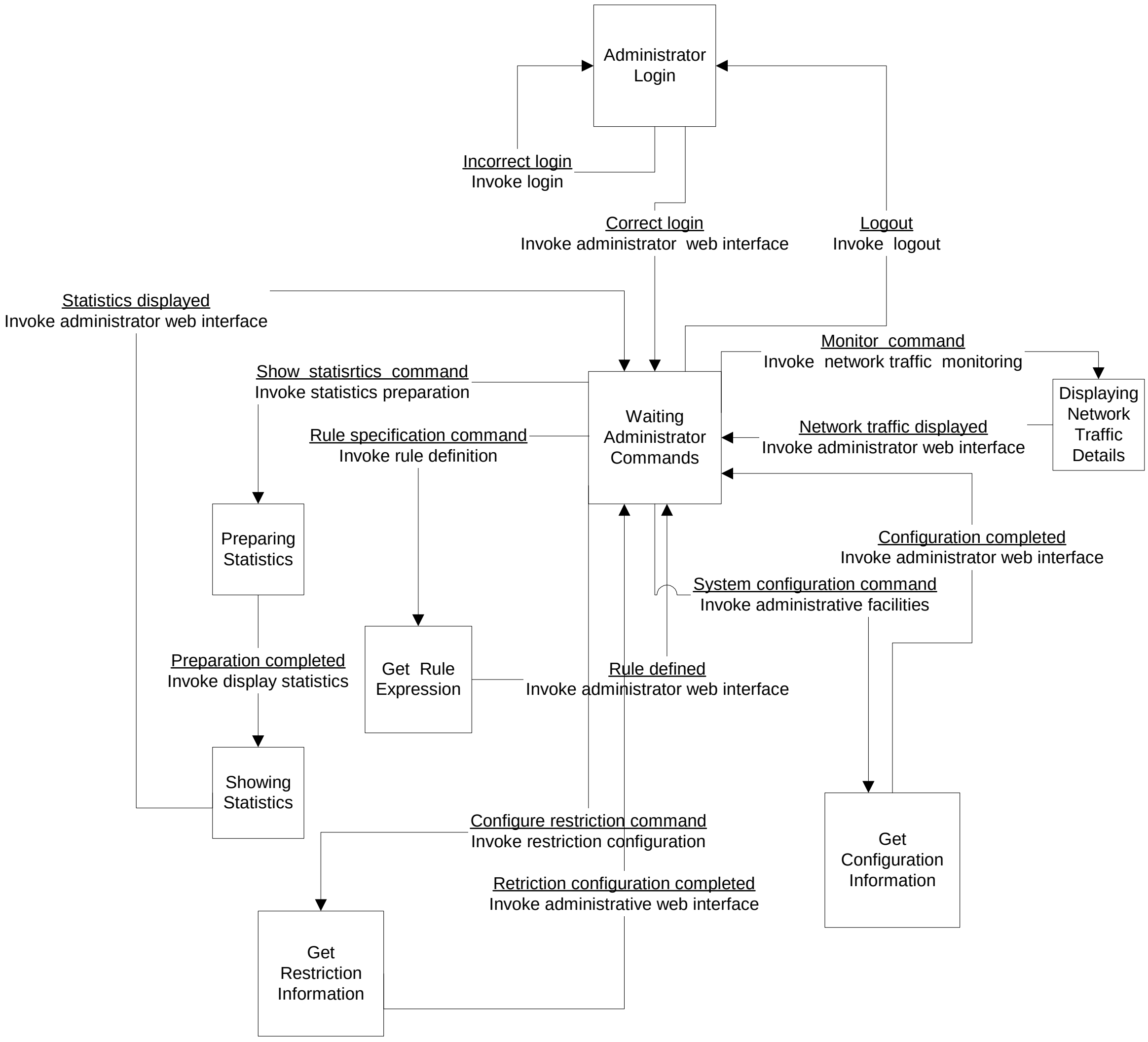
Name	Filtered Packet
Input to	Caching Module 3.2 Display Module 3.3
Output from	Content Filtering Module 3.1 Caching Module 3.2
Description	A packet from a Remote Server is filtered in the Content Filtering Module. Then it is sent to Caching Module to be cached and forwarded from the Caching Module to the Display Module.
Format	TCP/IP packet

Name	Request Validation
Input to	Check for Intrusions 2.5
Output from	Modification of Confidential Internal Information 3.4
Description	A packet from a Remote Server is filtered in the Content Filtering Module. Then it is sent to Caching Module to be cached and forwarded from the Caching Module to the Display Module.
Format	TCP/IP packet

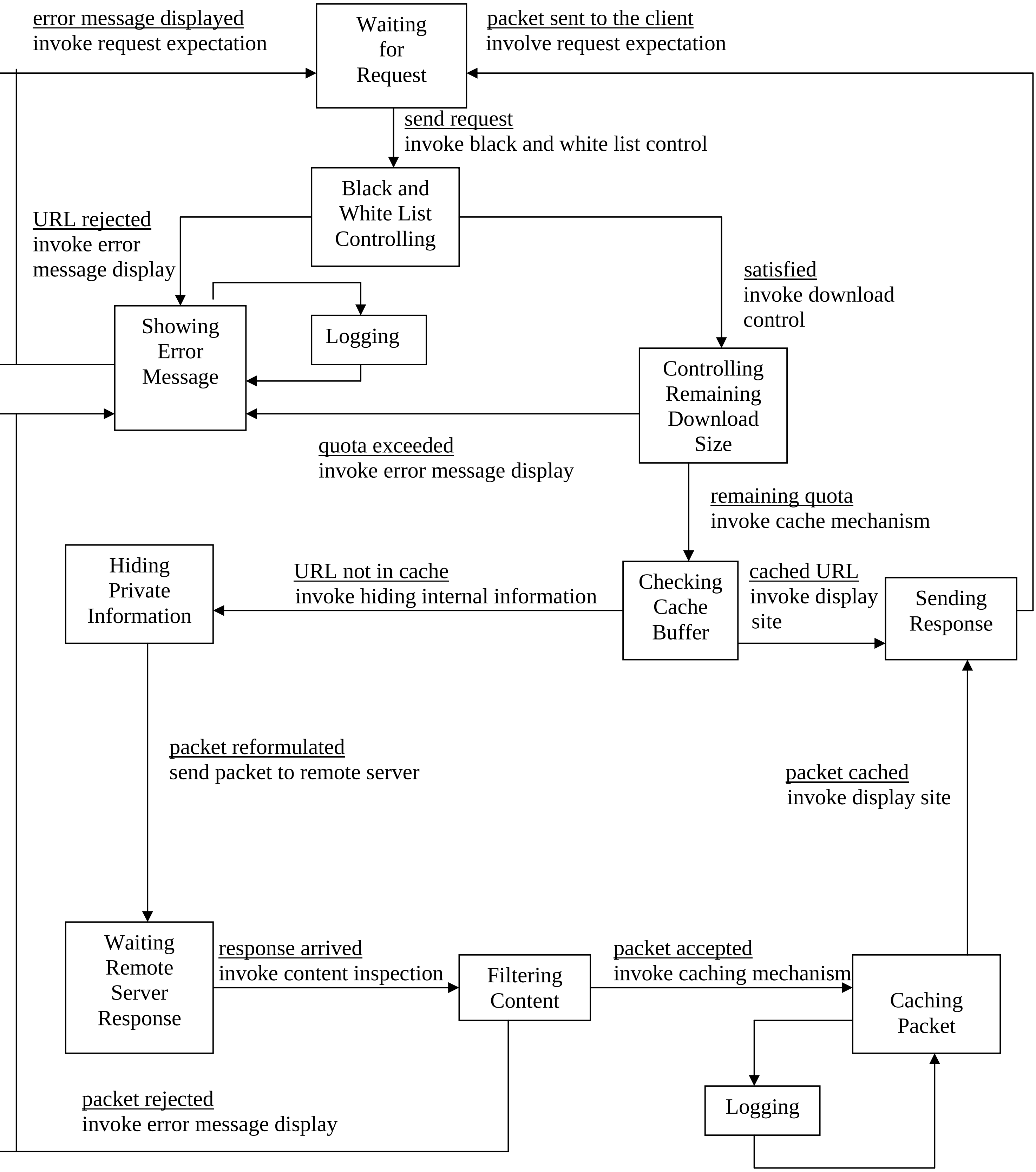
7.3 Behavioral Modeling

7.3.1 State Transition Diagrams (STD)

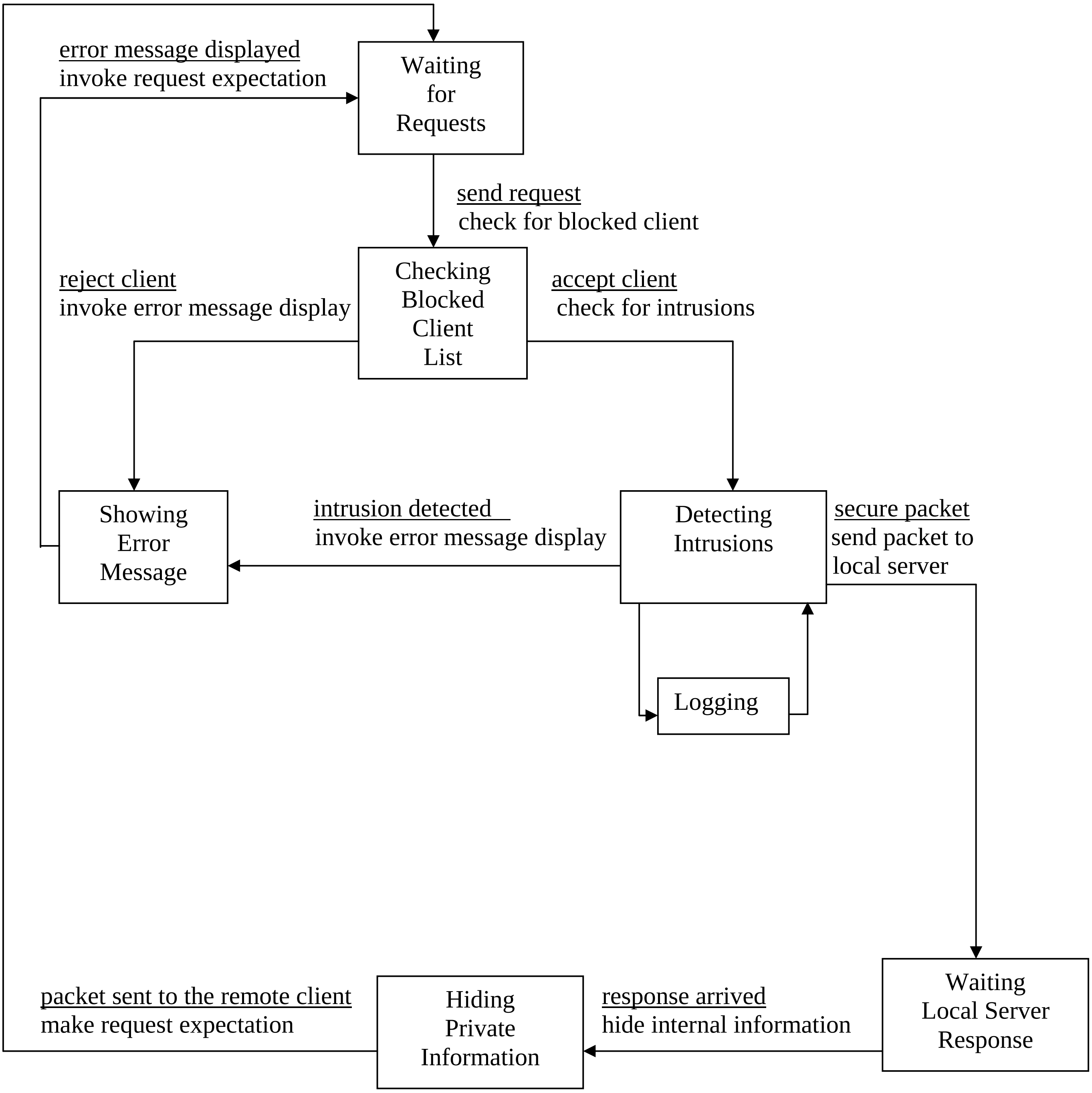
Web Administration Interface


Administrator Login
Incorrect login
Invoke login
Correct login
Invoke administrator web interface
Logout
Invoke logout
Statistics displayed
Invoke administrator web interface
Monitor command
Invoke network traffic monitoring
Show statisrtics command
Invoke statistics preparation
Waiting Administrator Commands
Displaying Network Traffic Details
Rule specification command
Invoke rule definition
Network traffic displayed
Invoke administrator web interface
Preparing Statistics
Configuration completed
Invoke administrator web interface
System configuration command
Invoke administrative facilities
Preparation completed
Invoke display statistics
Get Rule Expression
Rule defined
Invoke administrator web interface
Showing Statistics
Configure restriction command
Invoke restriction configuration
Get Configuration Information
Retriction configuration completed
Invoke administrative web interface
Get Restriction Information

State Transition Diagram for Local Client Actions



State Transition Diagram for Remote Client Actions


error message displayed
invoke request expectation
Waiting for Requests
send request
check for blocked client
reject client
invoke error message display
Checking Blocked Client List
accept client
check for intrusions
Showing Error Message
intrusion detected
invoke error message display
Detecting Intrusions
secure packet
send packet to local server
Logging
packet sent to the remote client
make request expectation
Hiding Private Information
response arrived
hide internal information
Waiting Local Server Response

8. RISK MANAGEMENT

8.1 Scope

Risk management is one of the facilities that should be performed in every project. Facing risks is unavoidable while doing projects and potential risks will affect the overall performance of the project if they occur, so having a risk management plan and coping with risks leads to more successful projects. As a result, we decided that we should perform risk management activities through the life cycle of our project.

8.2 Risk Table

Risk	**Probability**	**Impact**
Lack of knowledge on subject	%30	2
Misunderstanding requirements of customers	%20	2
Members' failure in performing due responsibilities	%30	3
Lack of experience in holding a project management process	%20	3
Falling behind schedule	%10	3
Hardware / Software failure on development platform	%5	1
Withdraw of a team member	%3	2
Disagreements among team members	%15	4

Impact Values:

1- Catastrophic management process

2- Critical

3- Marginal

4- Negligible

8.3 Risk Mitigation, Monitoring and Management Plan

- ***Lack of Knowledge on Subject***

Since none of our team members have a background on security or network areas, it is highly possible that our project will be affected. This is likely to have a critical impact on the project; so to avoid being blocked by missing knowledge, we have decided to track the progress of each member in weekly reviews. In case we realize we are in a risky situation, we will try to cooperate in research and consult experienced developers studying in this area.

- ***Misunderstanding Requirements of Customers***

Misunderstanding of customer requirements is also a common failure in software engineering. To avoid this risk, we are meeting with people from as many companies as possible, and trying to lower the risk that specifications may escape unnoticed. These companies are both developer companies and end users, so that we are exposed to different points of view. We are trying to be as clear as we can in terms of language and jargon, and repeat the information we get, asking for confirmative feedback. As the project progresses, we will develop prototypes incrementally and ask for feedback at regular intervals, so that any misleading can be noticed as early as possible.

- ***Members' failure in performing due responsibilities***

Being senior students and attending various technical elective courses with highly changeable demands makes it a high possibility that we may fall behind our personal deadlines. This is another case where we rely on weekly reviews and our mail group to monitor. Also to avoid the risk of failures due to personal factors, we are trying to plan personal tasks in terms of weeks so that the members are able to plan their weekly schedule with respect to project progress. For the management process, we have decided that if a member is not likely to fulfill necessary tasks in due course, others must be notified beforehand by the mail group, so that appropriate steps may be taken before too late, such as a redistribution of tasks.

- ***Lack of Experience in Holding a Project Management Process***

Another shortcoming of our team is that since this is our first major-scaled project, we are not experienced in coordination and management of a project. We are trying to foresee the problems of working in a 5-member team and take decisions in collaboration. In case we detect any fallacies in project management, we will try to overcome it by scheduling more meetings.

- ***Falling Behind Schedule***

Besides failing in individual tasks, it is also possible that we may altogether overestimate our joined capacity, and fail in scheduling the overall process. We are trying to prepare and follow our schedules as early as possible, which will lead to more time to regain control if any planning fallacy occurs. We are always considering that extra time may be required due to natural tendencies to overestimate personal capabilities and underestimate the time that will be required to fulfill tasks.

- ***Hardware / Software Failure on Development Platform***

Hardware and software failures must be taken care of in all kinds of projects, since although its probability is not so high, its impacts will be great. We have chosen two team members, who will be responsible with weekly backup of work done. This responsibility is also assigned to all members in terms of their private work. We are also intending to use the group cvs that will not be hosted in our personal computers. In case of a failure, every member will be responsible by the restoration of the project from the backups, in terms of restoring one's own work to the group cvs.

- ***Withdraw of a Team Member***

Withdrawal of a team member, although not that common, is a great risk in terms of its effects. We are intending to view the process as professionally as we can, and this realization will avoid serious misbehavior which follow from personalizing of situations. In case that these efforts are useless, and a member decides to withdraw from the team, all members are required to document the work they are doing individually, and store both this documentation and the source codes in the group CVS. Also we have come to an agreement that the member will spend one more week with the rest of team, helping and instructing them about the work that must be later developed by the team.

- ***Disagreements Among Team Members***

Disagreements among team members are not rare in groups larger than a few people, because there is a great deal of subjects to be decided, varying from the times scheduled for meetings to the prospective properties of the final product. Therefore, we have decided to always follow democratic rules in decision process, and try to convince opposed people before moving to other decisions. Members are asked not to be conserved about their feelings, to avoid serious disagreements that might lead to greater problems later. In case of serious disagreements, we will all take time for more research and discuss the subject at the following meeting. Since our team consists of five members, we will always follow the majority and the members have agreed to accept the decision of the majority as their own.

9. SOFTWARE QUALITY MANAGEMENT PLAN

9.1 Introduction

Developing a project as concerning project quality and continuity does stand as an undeniable property leading to the constructed projects perfection. According to that idea, we make analysis for what we can do in order to maximize the project that we plan to construct and publish in future. To aim at highest-level quality, we shall detect possible aspects which are potentially against product development and product itself, and avoid those aspects throughout the software process. Especially, we will care about the following points;

➢ Security is the most important quality measure for our application level gateway because it is mainly designed for network security. So throughout the project process, we will mostly care about whether our tool provides enough security and reliability.

➢ Network access speed is another important issue for our tool due to its indispensable necessity for providing acceptable network connection access.

➢ Ease of use and user friendliness is unavoidable for almost all projects . So, we will try to provide the user with carefully designed graphical user interfaces so that the user will not have to read loads of manuals.

The design of the software should be flexible enough to be compatible with modules that can be added in the later development phases This is another quality issue that we will consider.

9.2 Quality Assurance Tasks

For ensuring software quality, we determined some tasks that will be conducted throughout the software process by all of the members of the group.

- Schedule checking will be performed to cope with the risk of falling behind the timetable.
- A coding standard document will be prepared to make the code understandable for all of the group members and to ease debugging in case of error.
- Umbrella activities will be conducted regularly.
- Unit tests will be held regularly.

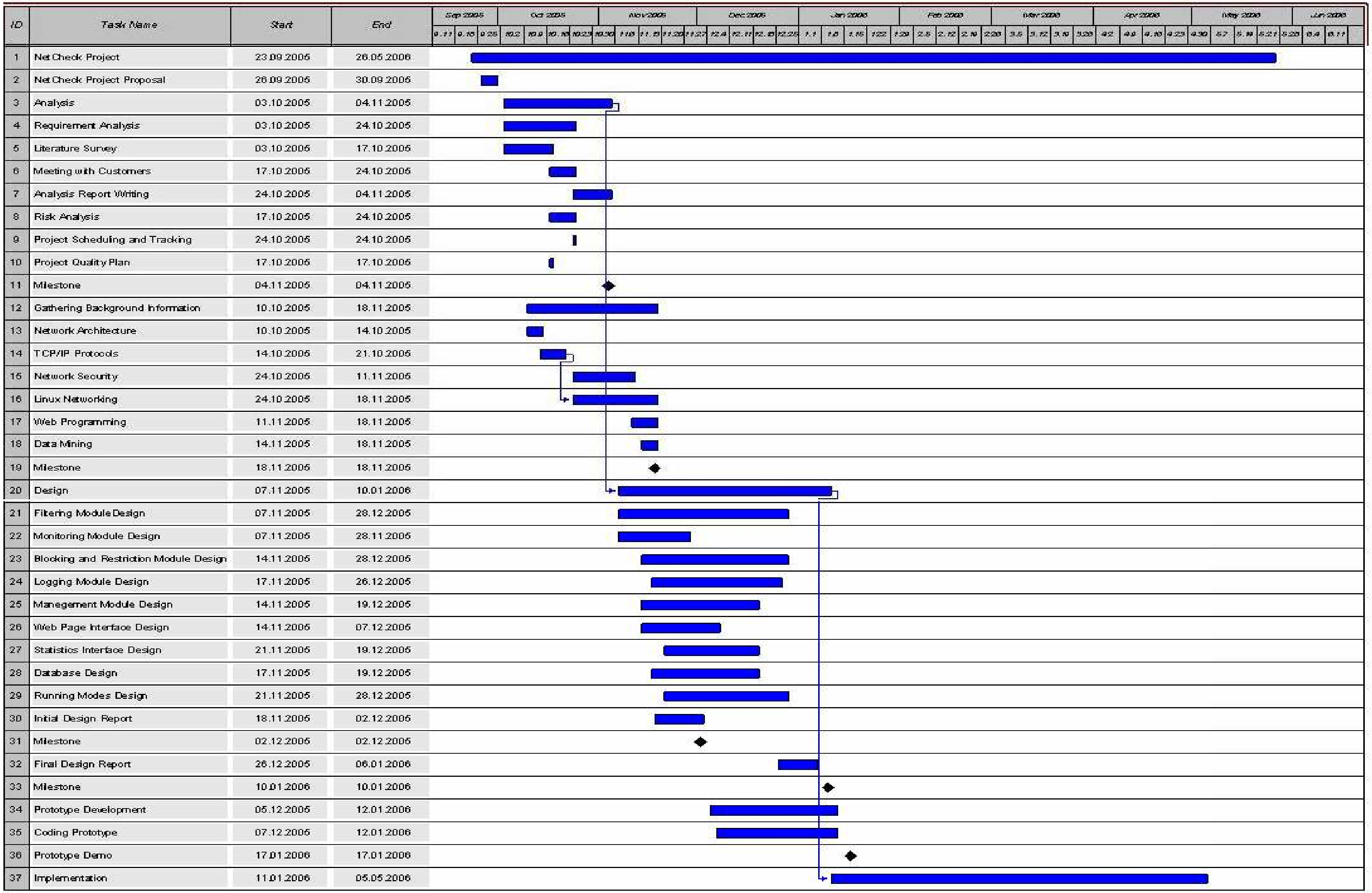
9.3 Reviews

Since formal technical reviews are an important software quality activity, we will perform frequent and regular meetings for performing these formal technical reviews. By these meetings we intend to trace the progress of the project and assign weekly works to members of the group.

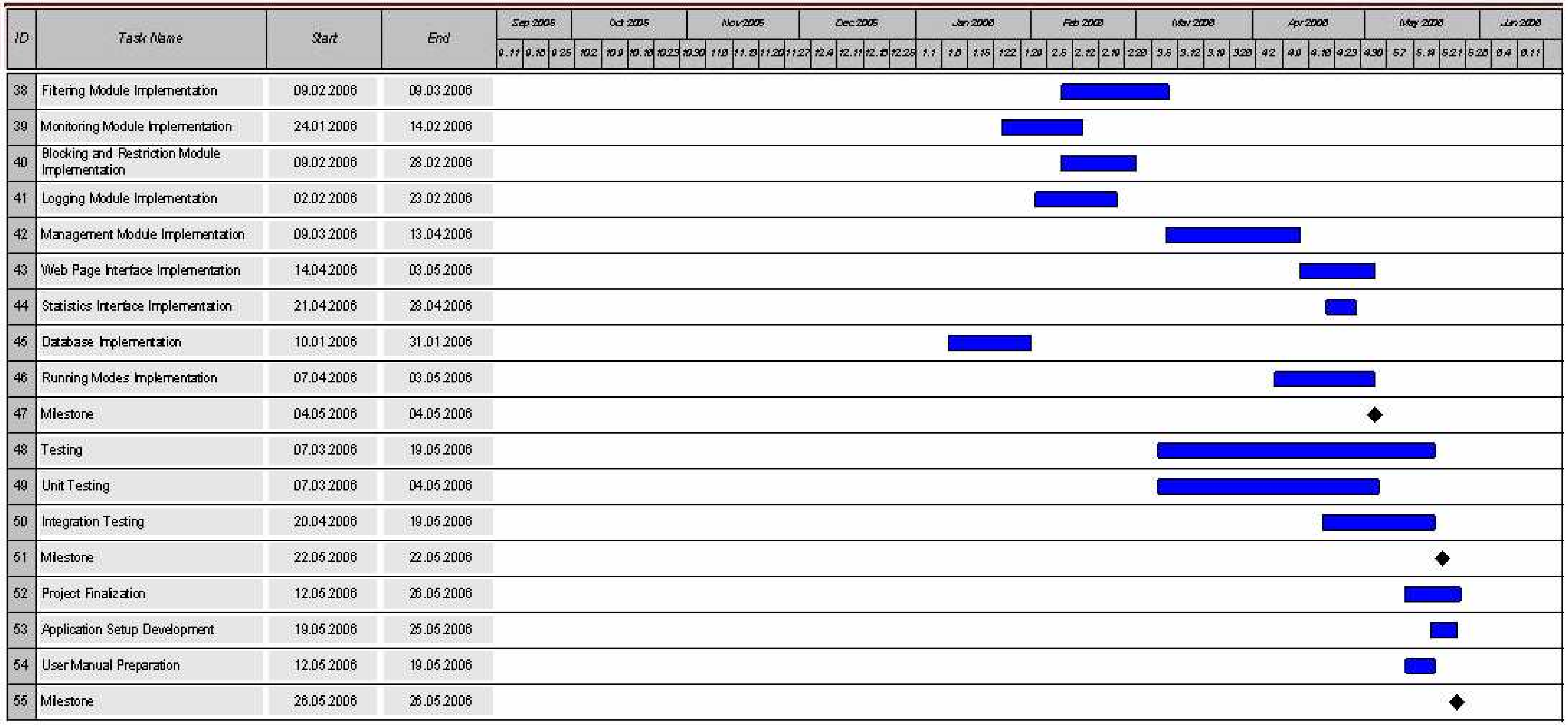
Our main objectives by performing formal technical reviews are;

- To make our project more manageable and traceable.
- To confirm that our project meets its requirements.
- To uncover errors in an early phase of the project.
- To ensure that our project obeys the predefined standards.

10. APPENDIX-A



ID	Task Name	Start	End
1	Net Check Project	23.09.2005	26.05.2006
2	Net Check Project Proposal	26.09.2005	30.09.2005
3	Analysis	03.10.2005	04.11.2005
4	Requirement Analysis	03.10.2005	24.10.2005
5	Literature Survey	03.10.2005	17.10.2005
6	Meeting with Customers	17.10.2005	24.10.2005
7	Analysis Report Writing	24.10.2005	04.11.2005
8	Risk Analysis	17.10.2005	24.10.2005
9	Project Scheduling and Tracking	24.10.2005	24.10.2005
10	Project Quality Plan	17.10.2005	17.10.2005
11	Milestone	04.11.2005	04.11.2005
12	Gathering Background Information	10.10.2005	18.11.2005
13	Network Architecture	10.10.2005	14.10.2005
14	TCP/IP Protocols	14.10.2005	21.10.2005
15	Network Security	24.10.2005	11.11.2005
16	Linux Networking	24.10.2005	18.11.2005
17	Web Programming	11.11.2005	18.11.2005
18	Data Mining	14.11.2005	18.11.2005
19	Milestone	18.11.2005	18.11.2005
20	Design	07.11.2005	10.01.2006
21	Filtering Module Design	07.11.2005	28.12.2005
22	Monitoring Module Design	07.11.2005	28.11.2005
23	Blocking and Restriction Module Design	14.11.2005	28.12.2005
24	Logging Module Design	17.11.2005	26.12.2005
25	Manegement Module Design	14.11.2005	19.12.2005
26	Web Page Interface Design	14.11.2005	07.12.2005
27	Statistics Interface Design	21.11.2005	19.12.2005
28	Database Design	17.11.2005	19.12.2005
29	Running Modes Design	21.11.2005	28.12.2005
30	Initial Design Report	18.11.2005	02.12.2005
31	Milestone	02.12.2005	02.12.2005
32	Final Design Report	26.12.2005	06.01.2006
33	Milestone	10.01.2006	10.01.2006
34	Prototype Development	05.12.2005	12.01.2006
35	Coding Prototype	07.12.2005	12.01.2006
36	Prototype Demo	17.01.2006	17.01.2006
37	Implementation	11.01.2006	05.05.2006



ID	Task Name	Start	End
38	Filtering Module Implementation	09.02.2006	09.03.2006
39	Monitoring Module Implementation	24.01.2006	14.02.2006
40	Blocking and Restriction Module Implementation	09.02.2006	28.02.2006
41	Logging Module Implementation	02.02.2006	23.02.2006
42	Management Module Implementation	09.03.2006	13.04.2006
43	Web Page Interface Implementation	14.04.2006	03.05.2006
44	Statistics Interface Implementation	21.04.2006	28.04.2006
45	Database Implementation	10.01.2006	31.01.2006
46	Running Modes Implementation	07.04.2006	03.05.2006
47	Milestone	04.05.2006	04.05.2006
48	Testing	07.03.2006	19.05.2006
49	Unit Testing	07.03.2006	04.05.2006
50	Integration Testing	20.04.2006	19.05.2006
51	Milestone	22.05.2006	22.05.2006
52	Project Finalization	12.05.2006	26.05.2006
53	Application Setup Development	19.05.2006	25.05.2006
54	User Manual Preparation	12.05.2006	19.05.2006
55	Milestone	26.05.2006	26.05.2006

11. REFERENCES

[1] Snort, The De Facto Standard for Intrusion Detection/Prevention
http://www.snort.org

[2] DansGuardian, Web Content Filtering for All,
http://www.dansguardian.org

[3] WebSense, Web Filtering and Web Security- Products and Services,
http://www.websense.com/

[4] NetNanny, Keeping Your Kids Safe on the Internet,
http://www.netnanny.com

[5] Liss II Secure Gateway, Telco Tech Security Systems
http://www.techotech.de