Title: Development of a firewall monitoring application

Volume: 1/1

Student: Mayra del Río López

Supervisor: Jose María Barceló Ordinas

Co-supervisor: Sandra Hernández Marsa

Department: Computer Architecture

INDEX

Chapter

1

INTRODUCTION

This document is the MFP (Master Final Project) final report of the Master's Degree in Information Technology, it bears the title of: "Development of a firewall monitoring application".

The introduction of this document is intended to give an overview of the project, as well as serve like a guide to the reader through the document, so that once you read this chapter you have a clear idea of the work done on the project.

For that, first are exposed the reasons that are carry out to do the project and the goals that are expected to be accomplished. Next are described the motivations because of it the student choose to do it and the work environment in which the project has been developed. Finally the organization of all the final report is detailed.

1.1. Reason and opportunity of the project

When the time of searching a MFP to finalize my studies in the FIB arrived, the first step I had to do was thinking about what field I would carry out my master thesis, an important decision in order to have an interesting experience.

In my case, it was clear to me that I would liked to do any job related to systems and/or networks, although I neither ruled out to relate it to some new field to discover or in which I could increase my knowledge.

So I decided to look regularly at the list of available projects in the FIB and I was looking for any offer that attracted my attention. I found some eye-catching offers but I was not sure about them. I was working in ASAC (Àrea de sistemes, aules informàtiques i serveis de comunicacions) area of the LCFIB (Laboratori de Càlcul de la FIB) so I addressed to its staff with the hope that they could give me some advice or idea for a master thesis. Finally speaking with Sandra Hernández we concluded that implementing a web application for monitoring firewalls could be an interesting tool. LCFIB needed such a tool since the current system they were using did not satisfied their user requirements. Sandra Hernández accepted to supervise the master thesis and I consulted the whole idea with Jose Maria Barceló, who is a lecturer at the faculty in the networking area, to be the tutor of the master thesis.

At the end I decided to develop the application for the LCFIB, because it was the closest project from what I was looking for. In my opinion, this project would give me the opportunity to apply the knowledge acquired during my studies and the possibility of acquiring new knowledge.

1.2. Project goals

The main goal of this project consists of the development of a via web application that offers to the LCFIB staff an easy and comfortable way to do the monitoring of their firewall logs.

In terms of functionality, the application must provide:

- Basic monitoring of firewall logs that the user wants to view, so the user can see what is happening in real time or logs from another time prior to the current date.
- Logs are filtered to have a better view of what the user wants to monitor. The application offers the possibility of managing the own filters that are created by the user.
- Compatibility with different log formats, in order to have in a single tool a centralized monitoring of the different firewalls.
- Completion of reports using statistics of the data that the user previously selects for this purpose, with their respective graphical representation. These graphs are necessary to facilitate the understanding of all the data that the firewall logs can contribute.
- "Strange" packet detection in logs. It can help to detect any anomalies in the system.
- Incongruities detection in the configuration files of the monitored firewalls to prevent a possible security failure.

1.3. Motivation

Several reasons has taken me to develop this project, the most significant are detailed below.

When doing a task is helpful to know that it can be useful, so that the performance of an application that will be used and the direct contact with the future users are an encouragement to do it better.

The fact that the project is related to networks and systems was the reason that taken me to decide for this because it is the part of informatics which I have more knowledge and personal interest. On the other hand, the web development brought me new knowledge about this topic and I was able to learn further technologies that I haven’t known much.

Another reason is the chance to start a project from the beginning, taking care of the problems that might appear daily, and having the ability to choose the best choice in each case.

Finally this project has been carried out in LCFIB, which has been an advantage as for tools, documentation and support.

1.4. Work environment and tools: LCFIB

Like was quoted, this project has been done in LCFIB. The hardware and the software were provided by the department.

Related to hardware a server located in LCFIB's machine room had been available for exclusive use for the project. I also have had at my disposal a personal computer in which I have been developing my everyday work.

Regarding software I was able to benefit from licenses held by LCFIB for some products like Windows XP, Office 2003, SSH Secure Shell and other programs that have been used occasionally. The rest of tools were free software.

The information was extracted the major part from internet, from degree notes all about specification and design, and especially from the LCFIB colleagues so I would like to mention the support that I have received from them. They helped me on any of the topics related to the project, because without them carry out this project possibly would have been more difficult.

Follow-up meetings

During the development different kind of meetings were done:

- Initials meetings with the final users to collect the real needs and establish the application requirements.
- Meetings with the LCFIB staff to consult the application design.
- Follow-up meetings with the co-supervisor. The communication also was done frequently via mail.
- Meetings with the responsible of the ASAC area to present the phases of the application and correct the possible mistakes.
- Meetings with the supervisor. He controls the part more formal of the project like the redaction of this final report.

Finally, it is remarkable to say that on June 18, 2008 an internal presentation to the LCFIB staff was done. The main idea was to let the LCFIB people know the application that was doing developed and the current state of the project.

1.5. Final report organization

The final report is organized in a way that reflects the chronological order followed by the working process to carry out the project.

- CHAPTER 1

  In the present chapter, an introduction to the project is done with the reasons that are carried out to do it, which are the goals, the motivation for doing it, the work environment and the structure of this final report.

- CHAPTER 2

  In chapter 2 is described everything related to the previous study done which contains an introduction to firewalls and their environment, a brief technological study and finally the initial planning.

- CHAPTER 3

  In chapter 3 is presented an analysis of requirements showing the initial situation, it refers to other firewall monitoring applications. The system requirements are specified, both functional and non-functional.

- CHAPTER 4

  The specification phase is showed in chapter 4, it includes the use case model, the conceptual model and the behaviour model.

- CHAPTER 5

  The design phase is in chapter 5. In this phase the decisions that have been taken and the design of the different layers in which the system is based are exposed.

- CHAPTER 6

  The implementation phase is in chapter 6. The first step is to choose the technologies to be used. Also some implementation details and the tests are explained.

- CHAPTER 7

In chapter 7 the accomplished goals are reviewed together with a review of the initial planning, an economic study is done, the future prospects to follow are quoted, and moreover the personal evaluation and the project license are in this chapter.

- CHAPTER 8

Chapter 8 is a collection of bibliography that has been getting help for the project development. A distinction is made between books and websites.

- APPENDIX

User manual and installation manual are in the appendix.

Chapter

2

PREVIOUS STUDY

This chapter presents the previous study of the technologies that can be accessible for the development of the application and a planning of the project. The goal of this study is to get knowledge of the current technologies related to the project. This knowledge will be used as a starting point to choose the technologies to develop the web application.

To provide an overview, first an introduction to the firewalls is done together with an indication of the project field. It is also important to decide among the wide variety of technological solutions which of them are the most appropriated to the project, so that a brief study of each of these solutions is done. In the next chapter will be explained in more detail which are chosen to carry out part of design and implementation. Finally a planning of tasks done in the beginning of this project is showed.

2.1. Introduction to the firewalls

2.1.1. Basic networking concepts

Before describing what a firewall is, it is considered significant to describe some concepts that are essential in networking context. These concepts will make easy to the reader the understanding of this chapter of this final report.

Network protocol:

Network protocol, or communications protocol, is the set of rules that specify the exchange of data or orders during a communication among the equipment that is a part of a network.

In OSI classification, the communication can be divided into 7 layers. The protocols of each layer have a well defined interface. A layer communicates generally with the next lower one, the next upper one, and the layer at the same in other equipment in the network. This protocols division offers abstraction in the communication. An example of technology and network protocols according to OSI model:

Layer 7: Application layer	SNMP, SMTP, NNTP, FTP, SSH, HTTP, SMB/CIFS, NFS, Telnet, IRC, ICQ, POP3, IMAP
Layer 6: Presentation layer	ASN.1
Layer 5: Session layer	NetBIOS, RPC, SSL
Layer 4: Transport layer	TCP, UDP, SPX
Layer 3: Network layer	ARP, RARP, IP (IPv4, IPv6), X.25, ICMP, IGMP, NetBEUI, IPX, Appletalk
Layer 2: Data link layer	Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, FDDI, ATM, HDLC

Layer 1: Physical layer	Coaxial cable, fibre optic cable, twisted pair cable, microwaves, radio, RS-232

Table 1. OSI model

LAN:

The LANs, Local Area Network, allow applying computer technology at companies for share locally files and prints efficiently, and make possible for them the internal communications. This is done connecting data, local communications and computer equipment.

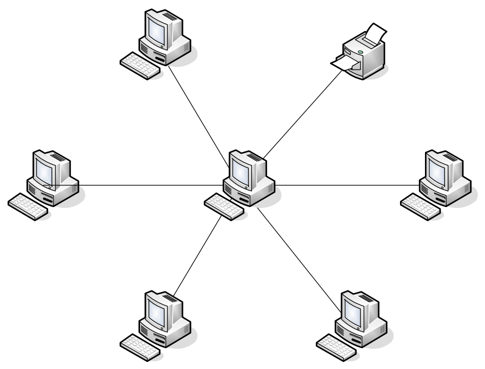


Illustration 1. Local network with star topology

VPN:

A VPN, Virtual Private Network, is a private network that is built inside a public or not controlled network infrastructure, like internet.

With a virtual private network, a distance employee can access to the company head office network through internet, a safe tunnel is formed between the employee computer and a VPN router in the head office.

The standard protocol in virtual private network is IPSec, but also exists PPTP, L2F, L2TP, SSL/TLS, SSH, etc. each of them with its characteristics as for safety and the supported type of clients.

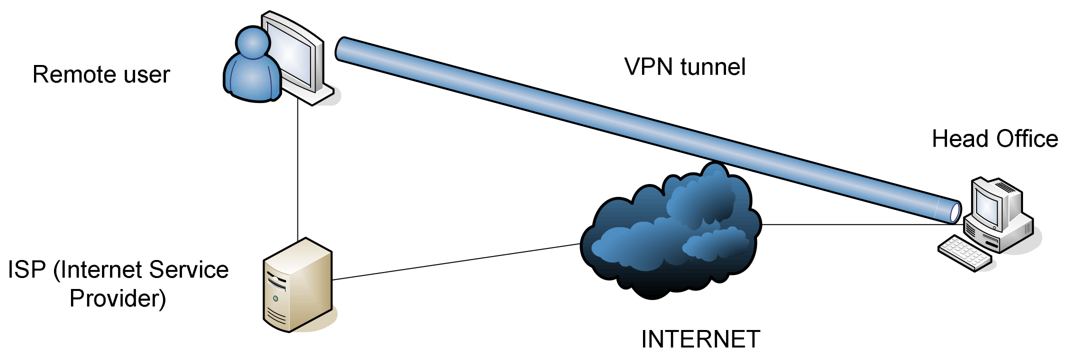


Illustration 2. VPN network example

DMZ:

A DMZ, Demilitarized Zone, is a local network located between the internal network of an organization and the external network, usually the internet.

The DMZ goal is that the connections both from the external network and the internal network to the DMZ are allowed, however the connections from the DMZ only are allowed to the external network. This allows that DMZ machines can give services to the external network, while they protect the internal network in the case of some intruder could get in any DMZ equipment. The DMZ usually is used to house web, mail or DNS servers.

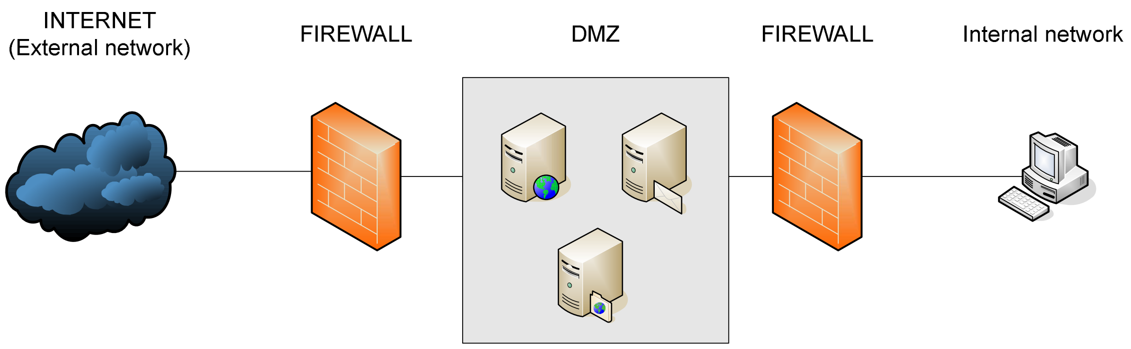


Illustration 3. Double firewall DMZ example

IDS:

An IDS, Intrusion Detection System, is a program used to detect unauthorized requests to a network or a specific machine. These requests can be accesses made by hackers or automatic tools.

The operation of IDS is based in the network traffic analysis which is analyzed comparing it with well known attacks or suspicious behaviours, like a port scanning, bad formed packets, etc. Moreover the type of traffic is being analyzed, the IDS also check the content and its behaviour.

An IDS usually integrates with a firewall, because if it was alone it is unable to stop the attacks and so the IDS intelligence joins with the power of firewall blocking, because it is the point where the packets pass before enter to the network.

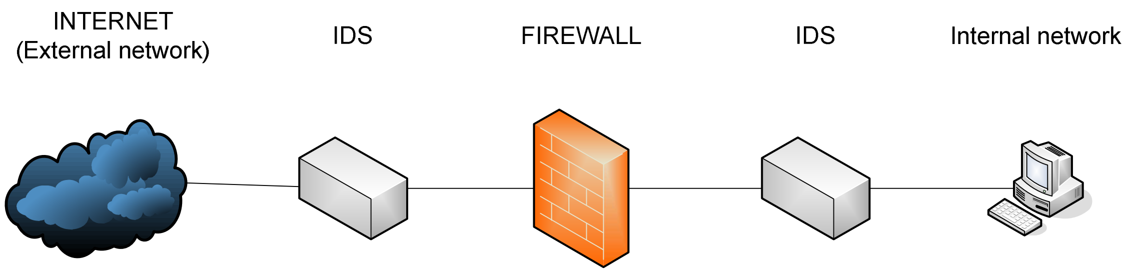


Illustration 4. IDS example

2.1.2. Firewall

A firewall is a hardware or software element used in a computers network to control the communications. These communications can be allowed or forbidden depending on the network policies that the organization responsible of the network have defined, that is, a firewall is useful to filter the traffic among networks.

In general, a firewall should be seen like a system with two or more network interfaces, in which are established filter rules. These rules examine the packet header and it is decided if a packet is passed, modified, changed or dismissed.

Next illustration shows the typical scheme of a firewall to protect a local network (LAN) connected through a router to the external network which is normally the internet. In this way the internal network is protected against unauthorized access attempt from the external network, these accesses can take advantage of vulnerabilities of the internal network systems. The firewall must be placed between the router and the internal network.

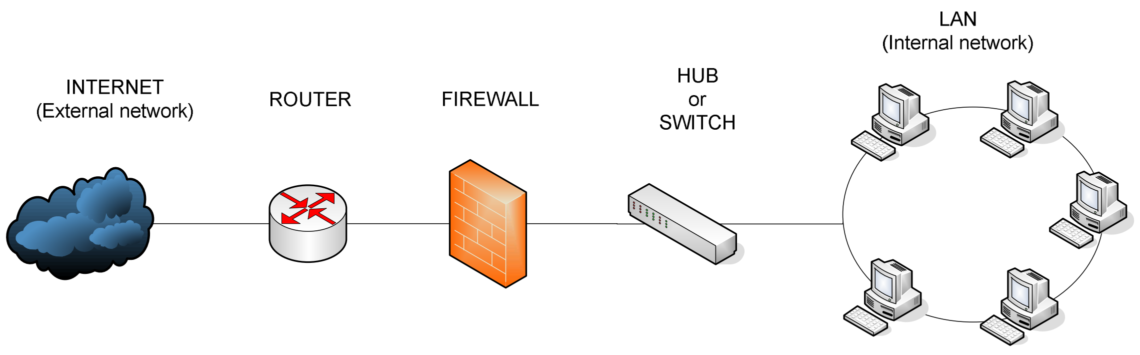


Illustration 5. Firewall classic typology

From the firewall basic concept, depending on the needs of each network the firewalls can be structured using different methods, the most typical are:

- One or more firewall can be placed for establish different security perimeter.
- Some server are necessarily exposed to the internet, for example a web server or a mail server, in this case any connection to them must be accepted. In this situation locate the server in another network zone is recommended, this is called DMZ or demilitarized zone as seen in previous sections. The firewall has three entries.

- A demilitarized zone with a double firewall can be done.
- In companies is usual to have a firewall like an internet protection with a double function: control the external accesses inwards and the internal accesses outwards.
- To have one or more firewalls filtering all the installation or a part of it is normal in hosting companies with many allocated servers.

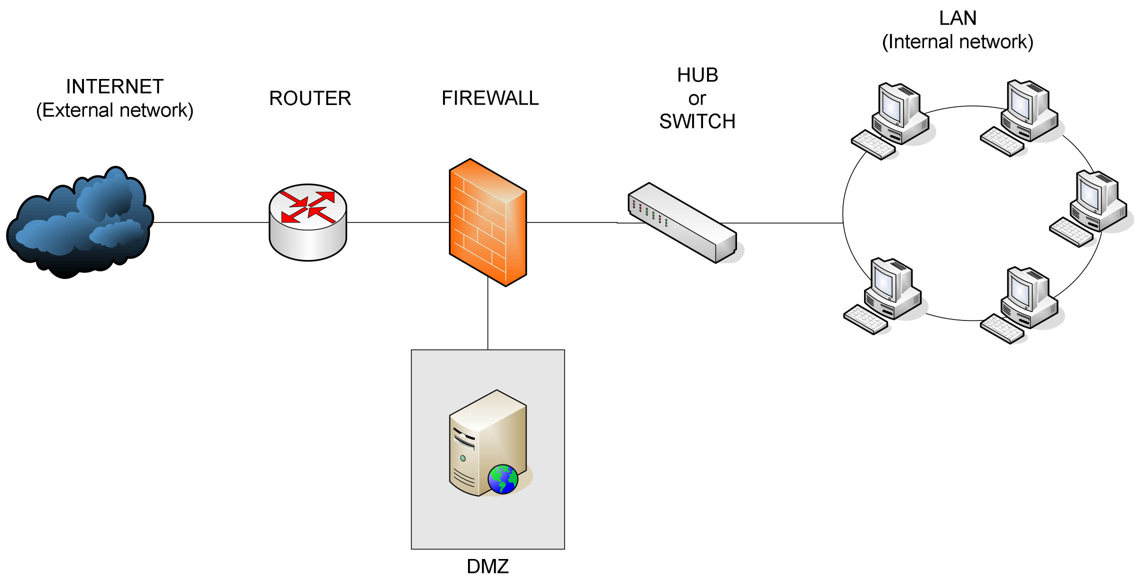


Illustration 6. DMZ with a firewall

All this different firewall structures have in common that they generally only have a set of rules. These rules examines the source and destination of TCP/IP protocol packets, or another protocol that could be filtered like UDP, ICMP, IPSec and another protocols linked to virtual private network (VPN).

Firewall configuration

It is important the order of setting the rules to the firewall. When it is decided what to do with a packet, it is compared with every firewall rule until it reaches one that affect it, and this rule is applied, next nothing else is watched for this packet. Then it can be inferred that if very permissive rules are placed at the beginning of the firewall rule set, they will be applied to many packets and the rear rules that could be to concrete situations won't be applied, so we are risking the network that is trying to be protected.

There are two ways of implementing a firewall:

→ Accept default policy.

All what pass through the firewall is accepted and only what is directly specified will be denied.

This policy makes easy the firewall management, because only ports or addresses of interest are protected. For that the points opened must be known and the rules for protect must be setted. The problem is that what is opened is not controlled or the dangerous ICMP packets aren't examined.

→ Deny default policy.

All is denied and only what is accepted explicitly will be able to pass the firewall.

A really inaccessible network can be obtained with this policy. The problem is that is more difficult to get ready a firewall, the operation of the firewall system (IPTables, IPFilter, etc) must be clear and what is accepted. This is always the recommended policy.

A correctly configured firewall adds big protection to the network that is being to try protecting, but in any case it should be considered sufficient, because the computer security covers a lot of fields and protection levels.

Firewall systems

There are a lot of firewall systems, on one hand there are systems that are centred in an operating system and on the other there are the systems developed by the manufacturers of computer components, Cisco for example. Below some firewalls of the first kind are described, they are the most common.

IPTables is a firewall system linked to Linux kernel. IPTables is the replacement of the previous systems IPfwadm and IPChains. This firewall system is a part of the operating system and it applies the defined rules. The `iptables` command that is offered in Linux is used for manage the rules.

First it's done when a firewall packet arrives is to watch if the destination is the own firewall or another machine. The INPUT rules are applied for the packets destined to the own firewall. The OUTPUT rules are applied for the packets destined outside the firewall. And the FORWARD rules are applied for the packets destined to other networks or machines.

The packet can be ruled out (and any more rule is being evaluated) or accepted. In the case of the packet would be accepted it arrives at a local process or it is sent back to the network like is shown in the next diagram:

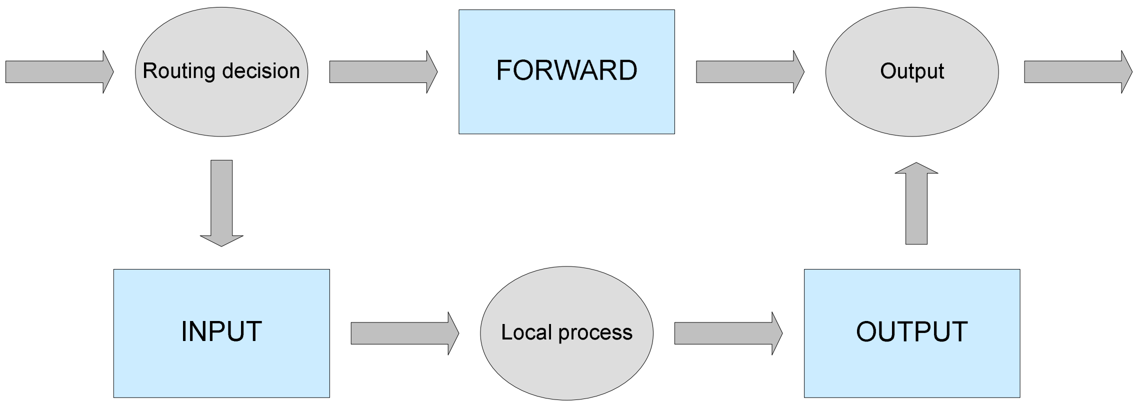


Illustration 7. IPtables scheme

IPTables also allow applying NAT (Network Address Translation), this is useful in the case that private addresses have to be converted or ports are readdressed. The NAT rules are applied before processing the INPUT chain, and after processing the OUTPUT chain.

Furthermore, we can have MANGLE rules before the NAT rules, the MANGLE rules are useful to modify the content of packets, but they are not very usual.

IPFilter is a firewall system based on free software (like IPTables) but it offers greater portability to different operating systems, some of them have included it in its kernel, for instance: Solaris, Open Solaris, FreeBSD and NetBSD. IPFilter stand out because of it portability, so other operating systems supported it: OpenBSD, Sun OS, IRIX, AIX, etc.

IPFilter operation is different from IPTables operation, the rules are stated in a configuration file instead of running a command every time, and moreover the rules are continued to be examined although there are several that affect at the same packet that are being processed. It is the last rule the one that will be applied in this case.

For increase efficiency, IPFilter allow giving priority to rules, so that in case the packet that is being processed would be affected by a rule with priority, this rule will be applied and the rest of policy would be leaved from examining.

PktFilter is a firewall system based on free software, it configures the packet filtering of Windows 2000/XP/Server2003. PktFilter only configures the filtered, it doesn’t implement other operations that other firewalls implements like traffic redirection or network address translation (NAT).

The filter rules are specified in a text file that follows the same syntax as IPFilter.

2.1.3. Field of project

The main concept of this project is the firewall monitoring through the analysis of the logs that have been generated by the own firewall system. It is important to know well the system and the traffic type that we are interested to analyze in a concrete machine to have a good view of what is happening, so that the adequate rules for what we want to register can be defined.

With respect to firewall systems, this project supports both IPTables and IPFilter, partly because they are the most extended type of firewall and partly because the firewalls that LCFIB are interested to analyze use one of these systems.

It is important to emphasize that this project does not develop an IDS, because this kind of tools analyze all the network traffic, whereas this project only will analyze the logs left by the firewall. A real-time network analysis is a very complex task and it isn’t inside the goals planned at the beginning.

2.2. Technological study

This section is centred in the technological study of the different elements that are needed for the construction of the application. In this section it is explained the current technologies. In the chapter 6, *Implementation,* is showed and justified which of those technologies have been chosen. Below is a list of these elements:

- Operating system: The operating system in which the application will work.
- Web server: The application will be a web application so a web server is needed to accept the requests from clients and provide an HTTP response to them.
- Server language: The application will be created with a server language that the server will process.
- Database management system: The database management system will be the manager of data between the application and the database.
- Web page design: The web page will be the only way the user can interact with the application, so it is important to know the languages we can use to interact with the web browser.

2.2.1. Operating systems

Microsoft Windows

Microsoft Windows is an operating systems family developed by Microsoft. There are versions for home, business, servers and mobile devices such as handled computers and smart phones. All of them are based on graphical user interface based on window paradigm, hence his name in English.

Linux

Linux is an operating system that has two characteristics which differentiate it from other operating systems. The first it is that is free software, it means you don't have to pay any type of license to anybody. The second is that is open source, the source code can be freely modified, used and redistributed by anyone.

2.2.2. Web servers

Apache

The Apache HTTP Server is a free software and open source web server. The application is available for Unix, Solaris, Windows and other operating systems.

The Apache HTTP Server is a project of the Apache Software Foundation. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

Apache has been the most popular web server on the internet since April 1996.

Tomcat

Tomcat (also called Jakarta Tomcat or Apache Tomcat) is a web container developed at the Apache Software Foundation. Tomcat implements the Java Servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. Tomcat includes its own internal HTTP server.

IIS

Microsoft IIS (Internet Information Server) is a set of internet based services for servers using Microsoft Windows. These services provide a web server.

Web developers can use Microsoft ASP technology, which means that applications can be embedded in web pages that modify the content sent back to users.

2.2.3. Server languages

PHP

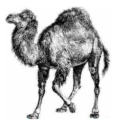
PHP is a computer scripting language that is specially suited for web development and can be embedded into HTML. It generally runs on a web server, taking PHP code as an input and creating web pages as an output. With PHP anything that would be done can be made with a CGI script, like process form information, create web pages, or send and receive cookies.

There are three fields in which PHP scripts can be used:

- Server side scripts.
- Command line scripts.
- Graphic interface applications.

The most powerful and notable feature of PHP is the support to a great quantity of databases, like Informix, MySQL and Oracle.

Perl



Perl is a general purpose dynamic programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more.

Perl borrows features from a variety of other languages including C, shell scripting (sh), AWK, sed and Lisp. In its features are included lists, associative arrays, complex data structures, object oriented programming model, automatic data typing or memory management, among other things.

Java

- JSP (Java Server Pages)

JSP is a Java technology that allows dynamically generating a response to a web client request. The JSP allows to specify Java code inside the HTML code.

- Servlets

A servlet is a program written in Java that is executed as a network service. A servlet is an object that receives a request and generates a response based on that request.

ASP

ASP (Active Server Pages) is Microsoft server side script engine for dynamically generated web pages. Most ASP pages are written in Visual Basic Script or Jscript (Microsoft's JavaScript). ASP runs inside IIS.

Many different applications can be created with ASP. It allows database access, server filesystem access and in general access to all the resources of the server.

In 2002, classic ASP was substituted with ASP.NET which includes improvements to language possibilities and work speed.

2.2.4. Database management systems

MySQL

MySQL is a database management system. The project source code is available under de GNU General Public License, but they also have MySQL Enterprise subscription offered for business users. MySQL was owned and sponsored by MySQL AB company, a subsidiary of Sun Microsystems since February 2008.

Next there are some features of MySQL:

- Connectivity: clients can connect to MySQL server using several protocols.
- Internals and portability.
- Localization: the server can provide error messages to clients in many languages.
- Security: all password traffic is encrypted when you connect to a server.
- Scalability and limits: handles large databases.

PostgreSQL

PostgreSQL is an object relational database management system. It is released under a BSD style license and is thus free software. PostgreSQL is not controlled by any single company, but relies on a global community of developers and companies to develop it.

Some features of PostgreSQL are high concurrency, foreign keys, triggers, views, inheritance, etc.

Oracle

Oracle database is a relational database management system that has become a major presence in database computing. Oracle Corporation produces and markets this software.

In its features the most notable are that support transactions, scalability, stability and it’s multi-platform.

2.2.5. Web page design

HTML

HTML (HyperText Markup Language) is the predominant markup languages for web pages, so it is the language used by browsers to show web pages to users. HTML was designed to display data.

HTML provides a means to describe the structure of text based information in a document and to supplement that text with interactive forms, embedded images and other objects. HTML can also describe the appearance and semantics of a document, and can include embedded scripting language code which can affect the behaviour of web browsers and other HTML processors.

HTML is an easy language, but at the same time it is very limited, so it is used accompanied by other languages like JavaScript or CSS, that offers multiple functionalities.

XML

XML (Extensible Markup Language) is a general purpose specification for creating custom markup languages. It is classified as an extensible language because it allows its users to define their own elements. XML was designed to transport and store data.

A DTD (Document Type Definition) is one of several SGML and XML schema languages. The purpose of a DTD is to define the legal building blocks of an XML document, it defines the document structure with a list of legal elements and attributes.

There are two levels of correctness of an XML document:

- Well formed. A well formed document conforms to all of XML's syntax rules.
- Valid. A valid document additionally conforms to some semantic rules. These rules are user defined, or included as an XML schema or DTD.

CSS

CSS (Cascade Style Sheets) is a style sheet language used to describe the presentation of a document written in a markup language. The most common is to style web pages written in HTML and XHTML, but the language can be applied to any kind of XML document.

CSS is used to help readers of web pages to define colours, fonts, layout, and other aspects of document presentation. It is designed primarily to enable the separation of document content from document presentation.

JavaScript

Javascript is a scripting language most often used for client side web development. The syntax is similar to Java and C languages.

Javascript is an object oriented programming language. All the modern browsers have incorporated it. Next is a list of things that is able to do this language: display information based on the time of the day, identify the visitors browse, control browsers, validate forms data, create cookies, add interactivity to your website, and change page contents dynamically.

2.3. Initial planning

The initial task planning consists of doing an estimation of time in which each phase of the project must be developed. This is done to have an approximated view of project length.

The first step consists of identifying the set of necessary phases to develop the project. The planning of this project is based in the basic software engineering life-cycle, although it is a little bit modified according to our particular needs.

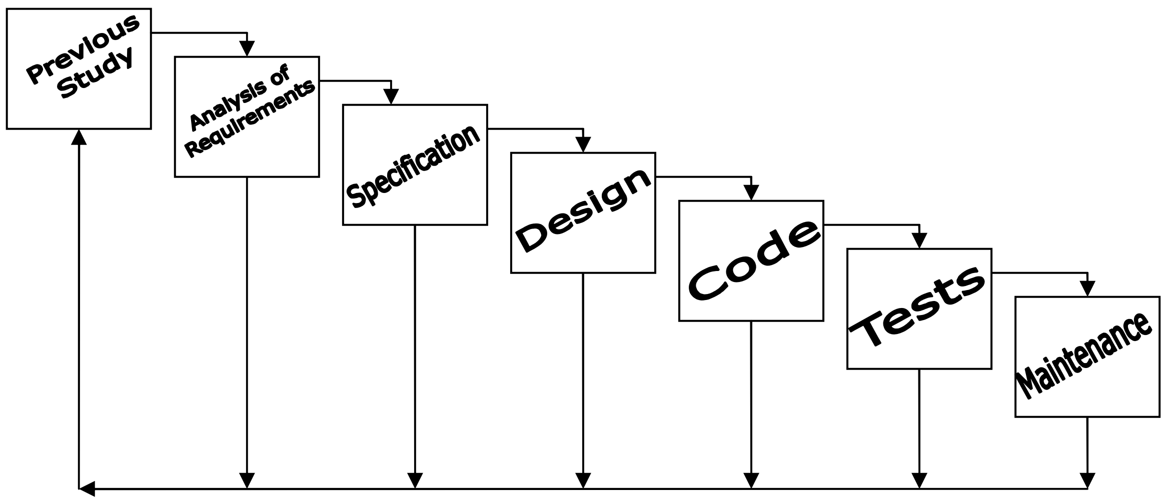


Illustration 8. Basic software engineering life-cycle

Project phases

Phase	Description	Hours
Previous study	In this phase is included the previous study and the analysis of requirements.	**70**
Programming languages	Study what different technologies could be used to develop the project, especially the programming languages.	14
IPTables, IPFilter	Study the different firewall systems.	36
Collect information of another products	Define which are the different existing products that do similar functionality to the project that is going to be developed.	20
Application development		**700**
Specification / Design of application	In this sub-phase it is done a brief specification and a general design. Later they are token up again because the design can be divided in different modules.	20
Main engine	Since here, the code phase begins and it is divided in modules.	190
Filtered		60

IPTables / IPFilter		100
Reports		100
Check the configuration file		90
Detect attacks		140
Tests	This is the tests phase. It is also included the maintenance phase because it’s considered that they are strongly linked.	**60**
Documentation	The documentation will be done along the length of the entire project, but at the end of the project it is expected to dedicate more time to do a complete revision of the entire document and complete it.	**60**
TOTAL		**890**

Table 2. Project phases of initial planning

The carrying out of this master final project represents 30 ECTS (European Credit Transfer System) and every credit entail among 25-30 hours. The length of this project was estimated in 890 hours, so it is in the range of expected hours.

Gantt diagram

The Gantt diagram is showed with the aim of have a clear overview of the content described in this section.

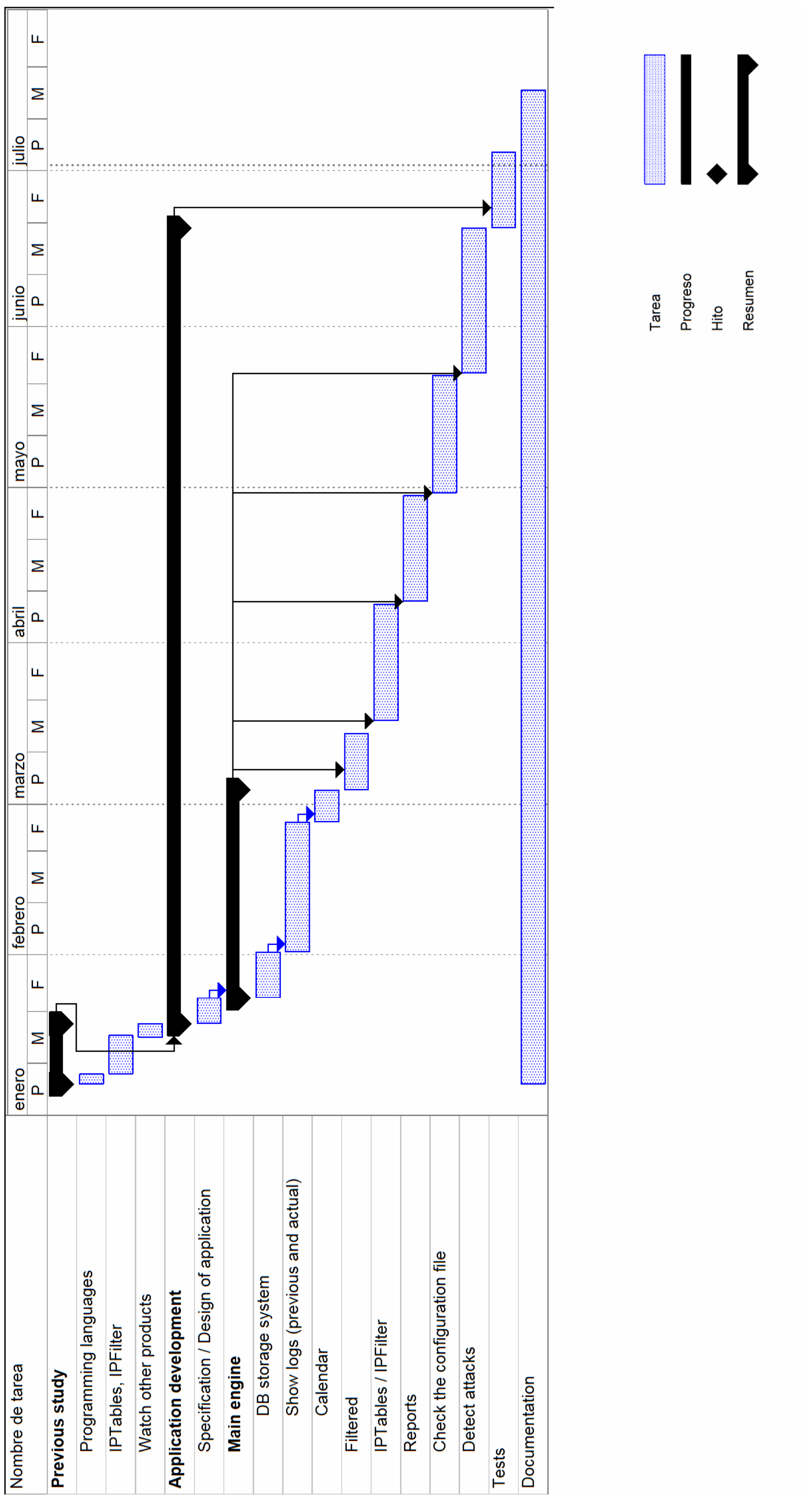


Illustration 9. Gannt diagram of initial planning

Chapter

3

ANALYSIS OF REQUIREMENTS

The analysis of requirements is the stage of the software engineering process assigned to discover, organize and document the real needs that brings the client forth request the system development, so a project planning of the necessary work to create this system in the next phases could be done.

What is expected with this analysis is to show the scope of the project, describe the functionalities and enumerate the characteristics that it has to accomplish.

3.1. Initial situation

In the following section a description of some applications/products that have the main functionality of firewall monitoring is done.

There is no need to examine exhaustively the whole products to realize the current solutions, so the study is only centred in some of them. Specifically, the reason why these applications have been chosen is simply because they have been used some time or a previous knowledge of them was acquired in LCFIB.

There are some applications that have been installed in order to test their operation and in order to obtain a better knowledge of how they perform. However there are others that can't have been accessible by the fact that they are commercials, in spite of it there are references by the LCFIB's staff that have been tested it time ago.

It is also important to quote the inclination for the free software, by the fact that can be shared with other universities. Otherwise, with closed software we have a big dependence on the manufacturer, which means certain limitations that in the *Conclusions* section will be explained.

For every one of the examined applications the following points will be considered:

- A brief summary about their operation, general characteristics, license type, etc.
- In the case of the installed and tested ones a list of positive and negative characteristics are added, this classification is elaborate having in mind the final product we want obtain for this project.

With that, a general view of the existing software in this field can be obtained, this way our application will be developed taking into account only the necessary characteristics and forgetting the dispensable ones in this case.

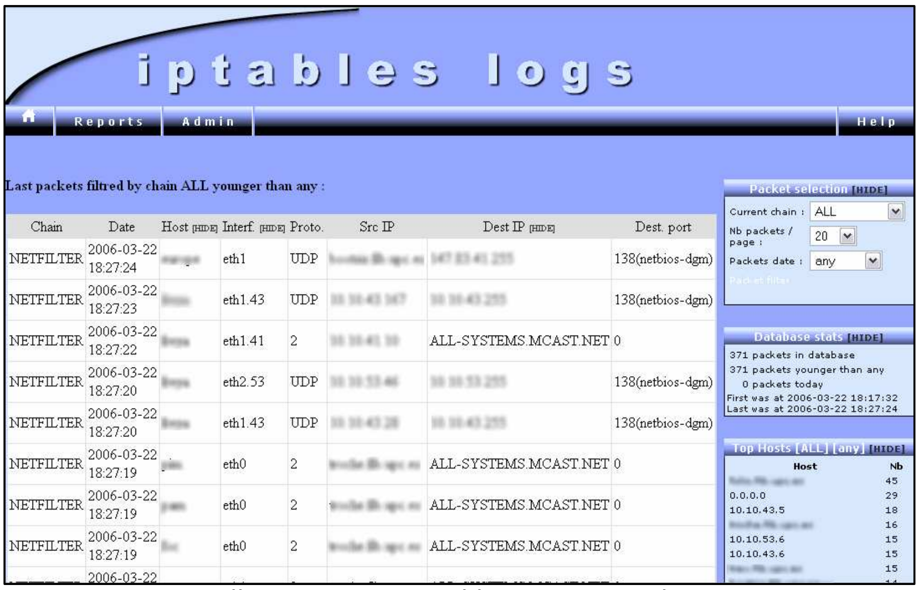
3.1.1. IPTables Log

http://www.gege.org/iptables/

"IPTables Log Analizer" shows in a HTML page format the logs of IPTables, all of them: rejected packets, accepted packets, masked packets, and so on.

The showed page must be easy to read and to understand, with the intention that log analysis time can be reduced. So that it contains statistics on packets and it has links to more detailed information on a given host, port, domain and the rest of information.

License: This software is free software (Open Source), distributed under the terms of GNU GPL. All source code is freely available for everyone.



Last packets filtred by chain ALL younger than any :

Chain	Date	Host [HIDE]	Interf. [HIDE]	Proto.	Src IP	Dest IP [HIDE]	Dest. port
NETFILTER	2006-03-22 18:27:24	[redacted]	eth1	UDP	[redacted]	[redacted]	138(netbios-dgm)
NETFILTER	2006-03-22 18:27:23	[redacted]	eth1.43	UDP	[redacted]	[redacted]	138(netbios-dgm)
NETFILTER	2006-03-22 18:27:22	[redacted]	eth1.41	2	[redacted]	ALL-SYSTEMS.MCAST.NET	0
NETFILTER	2006-03-22 18:27:20	[redacted]	eth2.53	UDP	[redacted]	[redacted]	138(netbios-dgm)
NETFILTER	2006-03-22 18:27:20	[redacted]	eth1.43	UDP	[redacted]	[redacted]	138(netbios-dgm)
NETFILTER	2006-03-22 18:27:19	[redacted]	eth0	2	[redacted]	ALL-SYSTEMS.MCAST.NET	0
NETFILTER	2006-03-22 18:27:19	[redacted]	eth0	2	[redacted]	ALL-SYSTEMS.MCAST.NET	0
NETFILTER	2006-03-22 18:27:19	[redacted]	eth0	2	[redacted]	ALL-SYSTEMS.MCAST.NET	0
	2006-03-22						

Packet selection [HIDE]

Current chain : ALL

Nb packets / page : 20

Packets date : any

Database stats [HIDE]

371 packets in database
371 packets younger than any
0 packets today
First was at 2006-03-22 18:17:32
Last was at 2006-03-22 18:27:24

Top Hosts [ALL] [any] [HIDE]

Host	Nb
[redacted]	45
0.0.0.0	29
10.10.43.5	18
[redacted]	16
10.10.53.6	15
10.10.43.6	15
[redacted]	15

Illustration 10. IPTables Log screenshot

Positive characteristics:

- ✓ The interface has a pleasant tonality colours combined with the use of tables, this makes easier the reading of so much log lines.
- ✓ The application allows seeing the lines grouped by any characteristic, for example, the packets which come in by port 22. This is made with direct links, so it's enough intuitive.

Negative characteristics:

- It hardly has functionalities. It only shows the log file but with a pleasant aspect, the project application requires to be more complete.
- There are some links to some functionalities that aren’t implemented, we supposed that it is because they are in development, however neither in the website nor the manual says nothing about this topic.

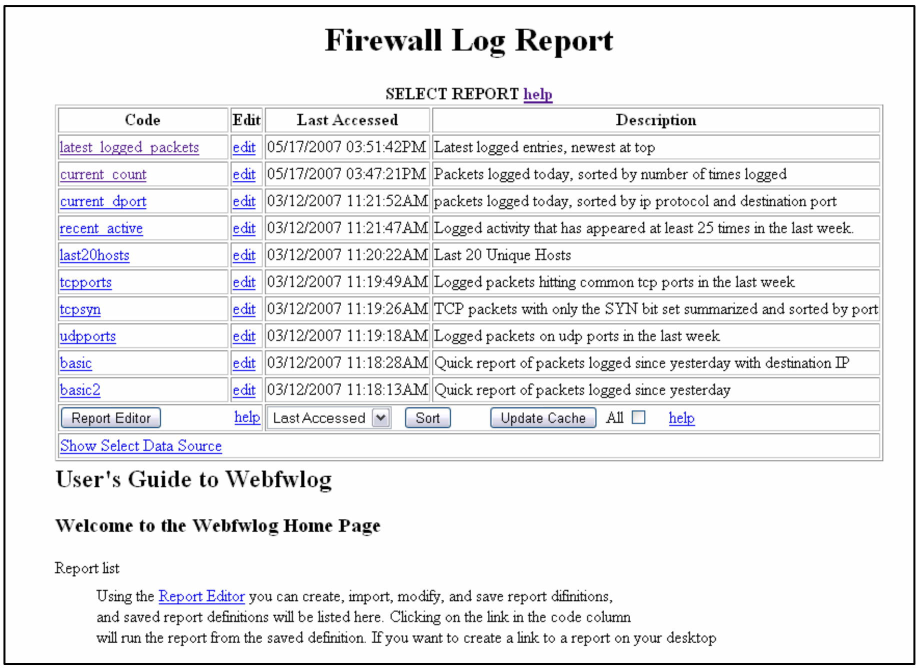
3.1.2. Webfwlog (Firewall Log Report)

http://webfwlog.sourceforge.net/

"Webfwlog" is a firewall log analyzer and a reporting tool, all this is showed in a web page. It supports standard system logs for several operating systems, supporting log file formats. "Webfwlog" also supports the logs saved in a database using the ULOGD option of the Linux Netfilter Project (IPTables).

With "Webfwlog" can be designed reports to use with the logged data in any desired configuration. The reports can be ordered easily and saved it for later use. Direct links to any saved report can be also created.

License: This software is free software (Open Source), distributed under the termns of GNU GPL.



Firewall Log Report

SELECT REPORT help

Code	Edit	Last Accessed	Description
latest_logged_packets	edit	05/17/2007 03:51:42PM	Latest logged entries, newest at top
current_count	edit	05/17/2007 03:47:21PM	Packets logged today, sorted by number of times logged
current_dport	edit	03/12/2007 11:21:52AM	packets logged today, sorted by ip protocol and destination port
recent_active	edit	03/12/2007 11:21:47AM	Logged activity that has appeared at least 25 times in the last week.
last20hosts	edit	03/12/2007 11:20:22AM	Last 20 Unique Hosts
tcpports	edit	03/12/2007 11:19:49AM	Logged packets hitting common tcp ports in the last week
tcpsyn	edit	03/12/2007 11:19:26AM	TCP packets with only the SYN bit set summarized and sorted by port
udpports	edit	03/12/2007 11:19:18AM	Logged packets on udp ports in the last week
basic	edit	03/12/2007 11:18:28AM	Quick report of packets logged since yesterday with destination IP
basic2	edit	03/12/2007 11:18:13AM	Quick report of packets logged since yesterday

User's Guide to Webfwlog

Welcome to the Webfwlog Home Page

Report list

Using the Report Editor you can create, import, modify, and save report difinitions, and saved report definitions will be listed here. Clicking on the link in the code column will run the report from the saved definition. If you want to create a link to a report on your desktop

Illustration 11. Webfwlog screenshot

Positive characteristics:

- ✓ The strong point of this software is considered the fact that can filter the reports with the desired data. For example if only the source and destination port of connections want to be seen.

Negative characteristics:

- The interface usability can be improved, because it is few intuitive and a user that enter for first time rarely found what he is looking for.

- All the information are showed in plain text, with any type of colour, this makes the analysis log a task so hard like the log firewall file was seen directly.

- About the help section, it isn’t coherent that the section was at the final of each page and with a large explanation of each section. Simplicity is necessary, in the case they want include help in the same page it must be briefer, but it would be enough having a link to a help page.

3.1.3. FWAnalog

http://tud.at/programm/fwanalog/

"FWAnalog" is a shell script that parses and summarizes firewall log files. Its actual version is able to analyze logs from ipf, OpenBSD 3.x pf, Linux 2.2 ipchains, Linux 2.4 IPTables, some ZyXEL/NetGear routers and Cisco PIX, Watchguard Firebox, Firewall-One, FreeBSD ipfw and Sonicwall firewalls.

It is easily extensible to other log file formats with just editing some regular expression.

"FWAnalog" uses the free software Analog[1] to create its reports. It converts the firewall logs in a false web server log and calls Analog with a modified configuration.

License: This software is free software (Open Source), distributed under the terms of GNU GPL.

Positive characteristics:

- ✓ The big difference compared with the rest of the free software analyzed is that can generate graphs with more significant data. This is important because not only statistics and numbers are given, also the graphic representation of these results is given. This is helpful because the user don't have to create this graphs with other software.

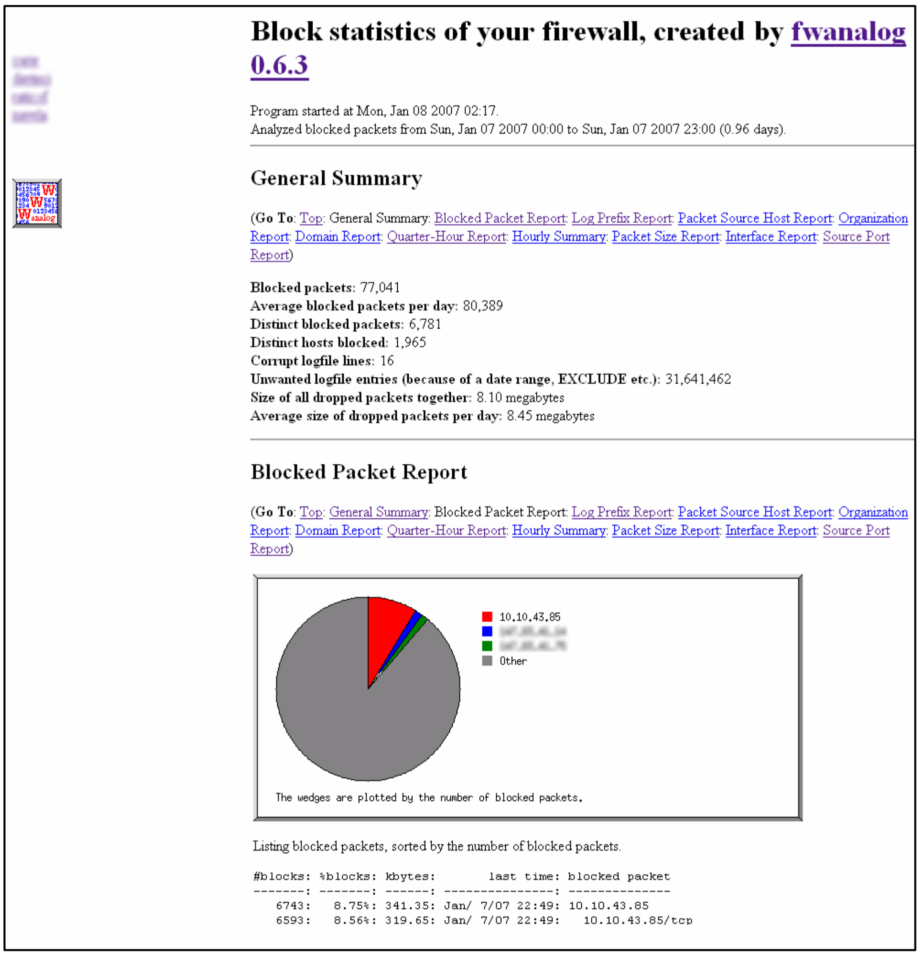
Negative characteristics:

- ✗ In each page there are a lot of information, it makes the user feel lost and looks like all this information are out of order, however what is happening is that it's excessive.
- ✗ All the information is showed in plain text with the exception of graphs, without any type of colour, this makes the log analysis a hard task.

[1] Analog is free software that analyzes web server logs to create statistics, for example of more visited pages, the countries of which the visitants are connected and all type of useful information.

- The application doesn't allow interaction with anything, so it gives the impression that restricts the user.

- Analog is a software used to make statistics but doesn't analyze software in real time. Fwanalog is using this software so it allows having information about the packets but it doesn't allow analysing what is happening at a given time. In the case of a firewall logs analyzer this characteristic is very important, because the system administrator wants to know what is happening if someone is attacking the network and it's very useful to analyze what is happening inside the firewall in real time.



Block statistics of your firewall, created by fwanalog 0.6.3

Program started at Mon, Jan 08 2007 02:17.
Analyzed blocked packets from Sun, Jan 07 2007 00:00 to Sun, Jan 07 2007 23:00 (0.96 days).

General Summary

(**Go To**: Top: General Summary: Blocked Packet Report: Log Prefix Report: Packet Source Host Report: Organization Report: Domain Report: Quarter-Hour Report: Hourly Summary: Packet Size Report: Interface Report: Source Port Report)

Blocked packets: 77,041
Average blocked packets per day: 80,389
Distinct blocked packets: 6,781
Distinct hosts blocked: 1,965
Corrupt logfile lines: 16
Unwanted logfile entries (because of a date range, EXCLUDE etc.): 31,641,462
Size of all dropped packets together: 8.10 megabytes
Average size of dropped packets per day: 8.45 megabytes

Blocked Packet Report

(**Go To**: Top: General Summary: Blocked Packet Report: Log Prefix Report: Packet Source Host Report: Organization Report: Domain Report: Quarter-Hour Report: Hourly Summary: Packet Size Report: Interface Report: Source Port Report)

Listing blocked packets, sorted by the number of blocked packets.

```
#blocks: %blocks: kbytes:        last time: blocked packet
-------: -------: ------: ---------------: --------------
   6743:   8.75%: 341.35: Jan/ 7/07 22:49: 10.10.43.85
   6593:   8.56%: 319.65: Jan/ 7/07 22:49:   10.10.43.85/tcp
```

Illustration 12. FWAnalog screenshot

3.1.4. Check Point Firewall/VPN

http://www.checkpoint.com/products/firewall-vpn.html

The product "Check Point Firewall/VPN" allows organizations to protect the entire network infrastructure and information, this is possible because of a unified security architecture that simplifies management and certifies certain level of security.

As follows, a group of characteristics of Firewall/VPN are shown:

- Alerts: Alerts contain control identifiers to automate reporting and give priority to the alarms for the analysis.
- Reports: These reports provide summary detail on control performance.
- Meta-controls: Meta-controls automate control related alerts and offers pre-defined reports on control performance.
- Using controls to manage security.
- Control alerts: There are categorized alerts that enable reporting on exceptions and violations of controls, this enforces the security policy.
- Control reports: There are reports pre-defined mapped to the specific controls.

References: LCFIB's staff.

- According to the references this application has a lot of options and configurable parameters.
- It also includes a tool to create firewall configuration files based on rules that are introduced with an easier syntax than the own firewall one.
- It allows correlating different Check Point firewall logs, so that you can know if an attack was made in several network segments.

Disadvantage:

- The fact that this product is commercial is a big disadvantage, because it isn't open software and so it isn't alterable by the own LCFIB to adjust it at its needs.
- It only works with Check Point products (firewalls and VPN) and it can't be linked to other type of firewalls that are in the organization.

3.1.5. SunScreen Secure Net

http://www.sun.com/software/securenet

"SunScreen Secure Net" is a firewall system designed for access control, authentication, and network data encryption. It consists of a rules based in dynamic packet filtering for network access control, just as an encryption and authentication engine that enable companies to create secure virtual private network (VPN) by integrating public key encryption technology.

"SunScreen Secure Net" characteristics:

- Centralized management: All the firewalls of the company can be managed from a central location.
- Graphic interface management system: This allows the administrator to manage remotely the application.
- High availability: The primary "layer" is the layer that manages all, but if it fails there is a second "layer" in passive mode that would be the active one, so that it assumes the control.

Reference: LCFIB's staff

- A graphic interface that shows and filters the logs.
- It has two operation modes, normal mode in which the stored logs are shown and interactive mode in which what is happening at this moment can be sawn.

Disadvantage:

- × As in the previous application, the fact that this product is commercial is a big disadvantage, because it isn't open software and so it isn't alterable by the own LCFIB to adjust it at its needs.
- × Also is a disadvantage that it only works if you use the own application firewall, when it would be more useful if it was adaptable to other type of firewall that are in the organization.

3.1.6. Intellitactics Security Manager

http://www.intellitactics.com

"Intellitactics Security Manager" is a security management application with benefits for security operations. "Security Manager" combines security event management with security information management to increase real time security effectiveness to stop attacks, resolve incidents and enforce policy.

The features of "Security Manager" are:

- Enable strategic business initiatives while protecting the infrastructure, applications and intellectual property of the organization.
- Increasing the efficiency of security operations.
- Enforcing policy by implementing best practices.
- Developing metrics, security performance indicators, communicating them in business terms to management, auditors and other stakeholders.

Reference: Security company staff.

Disadvantage:

- × As in the two previous, the fact that was commercial is a big disadvantage, because it isn't open software and so it isn't alterable by the own LCFIB to adjust it at its needs.

3.1.7. Conclusions

It is worth noting the remarkable difference between the commercial products and the “open source” ones, especially for their functionalities.

The open source software products usually are designed to satisfy specific user requirements. On the other hand, they do not have many usability options. However, it is an interesting choice if they can meet the user requirements without the need of a professional solution. Furthermore, the cost also is a issue to take into account. Open source usually is more flexible than commercial software and the code is available for further modifications.

Commercial products are more professional tools that in general cover much more functionalities than open source tools. However at the cost of price and depending on user requirements not all the functionalities are usually needed or used. Another disadvantage is how to integrate some of these products with current software at the lab. Dependency on the commercial product integration can be difficult. For example, if you have a Sun firewall and a Check Point firewall a different tool must be used for everyone to monitoring them instead to integrate all in one, like the open source products try to do.

The desired application is more complete than the opensource software we tested. For its making will be into consideration the advantages of everyone and adding the needed characteristics for the people who will use it. But it is not necessary a tool so large as the commercials ones, because a lot of functionalities wouldn’t be used by the reason that other applications cover them and the intention of this project is not replace them.

3.2. System requirements

The system requirements are determined by the needs or conditions to meet for a new or altered product, taking into account the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users.

There are two types of requirements:

- Functional requirements: a functional requirement defines a function of a software system or its component, so the set of functional requirements specify particular behaviours of a system.

- Non-functional requirements: non-functional requirements are requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviours. Generally they are related to performance requirements, security or reliability. They are often called qualities of a system.

To determine these requirements the principal ways have been the ideas contributed by the final users (LCFIB staff) and the ones extracted from other applications.

3.2.1. Functional requirements

These are the functional requirements:

- Firewall logs monitoring with IPTables and IPFilter compatibility.

  The base of the application is to monitor the firewall logs, it must support both IPTables and IPFilter format.

  All the logs will be centralized, to do that the logs must be passed through a parser that catch the data, convert it and finally store it in the storage system that we decide to use. This system should permit to make easy queries. These queries should guarantee the access to the logs of any day.

  Starting with this base, the extensions of the application can be added.

- Log filtered

  The filtering of the logs is the first complement to be added. The logs that were obtained in the previous requirement are filtered to show only the important data.

  The user can view the fields that he wants, so he doesn't need to read all the data of the same event, only the data that interests him. The filtered is also concerning content, it means that the logs that satisfy one or more conditions, for example the connections done from a concrete IP address.

  There is a management of these filters so that can be saved to be used later. This management includes: creation of a new filter, creation from an existing filter, modify the filter or delete it.

- Reports

  The need to have reports is a very important requirement. These reports aren't automated, because the main idea is that the user can create them dynamically when he needs them.

  The reports include graphs build on a field in a range of determined dates. In order to complete this graphs there are also a basic statistics.

- Firewall configuration

  This requirement is a checker of the rules that are been applied in the firewall, it is compatible with IPTables and IPFilter.

  This is useful for checking easily and quickly the valid rules of a set of rules. Currently the own firewall systems include this type of "checkers" so that this topic does not been studied in length.

- Anomalies detector

  The anomalies detector module works with the stored logs and its task is to detect the anomalies that can be produced the day before. It makes vertical and horizontal scans to detect the greatest number of suspicious connections. The user can consult the detections of any past day if it was scanned.

- System configuration

  The user can configure some settings of the application. The actual parameters are showed to the user and then he can change them according to his preferences.

3.2.2. Non-functional requirements

These are the non-functional requirements grouped by categories:

- Documentation: the application must be documented, there will be a user manual and an installation manual for the administrators. The code of the application also will be commented with the goal of helping somebody who wants to change it.

- Efficiency: the efficiency is the measure of which time and/or space is used for a concrete task, the compliance also must be considered.

- Reliability: reliability is the ability of a system to perform and maintain its functions in routine circumstances, as well as hostile or unexpected circumstances. So that the developed system will have a minimal time to be maintained and doing database backup will be easy.

- Usability: usability is the ease with which people can employ the application, it means that the application must be elegant and clear to make easy the interaction with it. Some things to take into account to do it are:

    - The set of pages must have a concrete structure consisting of header, body of the page, menu with options and footer.

    - The colour tonality must be pale and friendly to the human eye.

    - The application must work with any browser.

    - Satisfy the W3C standards, as far as possible.

Chapter

4

SPECIFICATION

The specification is a complete description of the behaviour of the system to be developed. It provides the necessary details about the specific requirements in a clear and not ambiguous manner.

The UML (Unified Model Language) notation has been used for the specification of this project. UML is a standardized visual specification language for object modeling.

4.1. Use case model

Use case model is a model that describes the functional requirements of a system in terms of use case. Use case modeling allows future users of a software system introduce as possible into its design. It uses the vocabulary of the users, not programmers.

There are two main entities in the use case model: actors and use cases.

An actor is something or someone which will use the system, they may be end users, other systems or hardware devices. In a system can be as many actors as roles can be taken by the users. In this application there are some types of roles, below it will be detailed.

A use case is a technique for documenting the potential requirements of the application. The use cases describe all the interactions that the actors will have with software to accomplish a goal. Use cases treat the system as a black box, and the interactions with the system are perceived as from outside the system, this is done because it forces to focus on what the system must do and not how is to be done.

4.1.1. Use case diagram

The use case diagram have the purpose to present a graphical overview of the functionality provided by a system in terms of actors, use cases and any dependence between this use cases.

In this application there are three actors: the User who connects to the web for interact with the application, the Clock who do a task in a determinate moment and the Update Process who is the manager of the logs.

The use cases correspond with the requirements quoted in the functional requirements, for some requirement has been necessary to do more than one use case, because more than one actor take part in it and it isn't necessary the intervention of all the actors to do their respective task. The use cases are: manage logs, monitor logs, filter logs, generate reports, check firewall rules, detect anomalies, monitor anomalies and configure system.

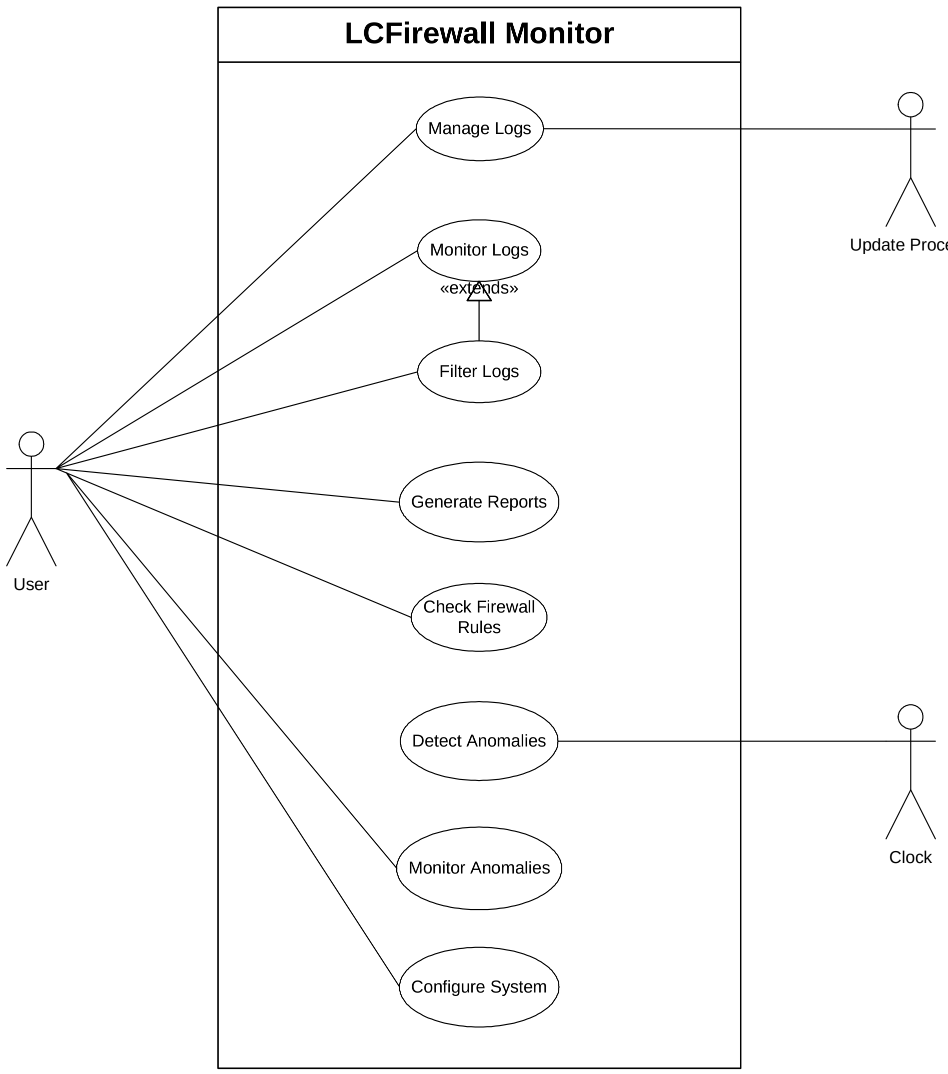


Illustration 13. Use case specification

4.1.2. Use case specification

The use case specification contains a description of events describing the interaction between the actors and the system. The specification also contains other information such as preconditions, postconditions, requirements and key scenarios.

Next is the specification of the use cases.

Use case	Manage Logs
Brief description	Use case which is the manager of store the logs in the system. Type: Primary and essential. Actor: User, Update Process.
Event flow	Basic flow: 1. The actor wants to save the logs. If the actor is the User the storage will be done at the moment and if the actor is the Update Process the storage will be done in real time. 2. The actor indicates to the system the file which contains the logs. 3. The system parses the file and stores the logs.
Preconditions	-
Postconditions	The logs are stored in the system.

Use case	Monitor Logs
Brief description	Use case which is the manager of show all the logs of the server selected by the actor. Type: Primary and essential. Actor: User.
Event flow	Basic flow: 1. The actor begins the use case. 2. The system displays configured servers. 3. The actor selects a server. 4. The system displays the logs of this server. Alternative flows: *Select date*: The actor selects a concrete date. (4) 1. The actor selects a date in the calendar. 2. The system displays the logs of this server for this date.

Log filtered: The actor decides to filter the showed logs. (3)
(Use case: Filter Logs)

Preconditions The actor must configure at least one server, otherwise there will be no logs to show.

Postconditions -

Use case Filter Logs

Brief description Use case which is the manager of filter the showed content by the use case *Monitor logs.* The criteria of filtered will be select by the actor.

Type: Primary and essential.

Actor: User.

Event flow Basic flow:

1. The system displays all the possible filter options to the actor.
2. The actor selects the logs characteristics that want to view.
3. The actor confirms his decision.
4. The system displays the logs of this server, discriminating against the criteria selected by the actor.

Alternative flows:

Load filter: The actor decides to load the filter. (2-3)

1. The actor selects a filter and gives to load the filter.
2. The system display the filter selected.
3. The actor puts a name to the filter and save it.

Save filter: The actor decides to save the filter. (3)

1. The actor shows that wants to save the filter.
2. The system displays a text box to put a name.
3. The actor puts a name to the filter and save it.

Modify filter: The actor decides modify an existing filter. (2-3)

1. The actor shows that want to modify the filter.
2. The system asks for confirmation.
3. The actor confirms.

Delete filter: The actor decides delete an existing filter. (2-3)

1. The actor shows that want to delete the filter.
2. The system asks for confirmation.
3. The actor confirms.

	Turn back: The actor decides to turn back. (1-3) 1. The system displays again the logs without having into account the new filtered.
Preconditions	The actor must configure at least one server, otherwise there will be no logs to show.
Postconditions	The logs are showed filtered according to the actor criteria.

Use case	Generate Reports
Brief description	Use case which is the manager of show the reports which the actor wants to view. Type: Primary and essential. Actor: User.
Event flow	Basic flow: 1. The actor begins the use case. 2. The system displays configured servers, fields, dates, kind of graphs, etc. 3. The actor selects the options he wants to the graph. 4. The system displays the report with a graph and statistics.
Preconditions	The actor must configure at least one server and select a field and a date range with logs, otherwise there will be no data to show.
Postconditions	-

Use case	Check Firewall Rules
Brief description	Use case which the firewall rules that the actor introduce are tested. Type: Primary and essential. Actor: User.
Event flow	Basic flow: 1. The actor begins the use case. 2. The system displays an area to introduce the rules. 3. The actor introduces the firewall rules and selects the type of firewall system that they belong to (IPTables or IPFilter).

	4. The system displays if the rules are correct or if there are any failure.
Preconditions	-
Postconditions	-

Use case	Detect Anomalies
Brief description	Use case which is the manager to detect the anomalies with the stored logs. Type: Primary and essential. Actor: Clock.
Event flow	Basic flow: 1. The actor consults the logs. 2. The system returns the requested logs. 3. The actor checks if there are any kind of anomaly and save them.
Preconditions	-
Postconditions	If there are anomalies, they are stored in the system.

Use case	Monitor Anomalies
Brief description	Use case which is the manager to show the anomalies detected by the application. Type: Primary and essential. Actor: User.
Event flow	Basic flow: 1. The actor begins the use case. 2. The actor selects the date he wants to view the anomalies. 3. The system displays the anomalies. Alternative flows: *Select date*: The actor selects a concrete date. (3) 1. The actor selects a date in the calendar. 2. The system displays the anomalies for this date. *Monitor logs*: The actor decides to monitor the logs of a concrete anomaly. (3)
Preconditions	-

Postconditions	-

Use case	Configure System
Brief description	Use case which is the manager to configure the parameters of the system. Type: Primary and essential. Actor: User.
Event flow	Basic flow: 1. The actor begins the use case. 2. The actor changes the parameters that he wants. 3. The actor finishes his changes. Alternative flows: *Invalid values*: The system displays that changes are not valid. (3)
Preconditions	-
Postconditions	The parameters of the system will be changed.

4.2. Conceptual data model

A conceptual data model is a map of concepts and their relationships. It describes what is significant to an application (object classes), the properties of these objects (attributes) and the association between pairs of those things of significance (relationships).

In addition to defining and organizing the data, data modeling will impose constraints or limitations on the data.

In short, the conceptual data mode consists of:

- Object classes.
- Associations between object classes.
- Attributes of object classes.
- Integrity constraints.

4.2.1. Class diagrams

In the UML, a class diagram is a type of static structure diagram that describes the structure of a system by showing the classes of the system, their attributes and the relationships between the classes.

Class diagrams are used for a wide variety of purposes, including both conceptual modeling and detailed design modeling.

Next is the class diagram for the specification of the application:

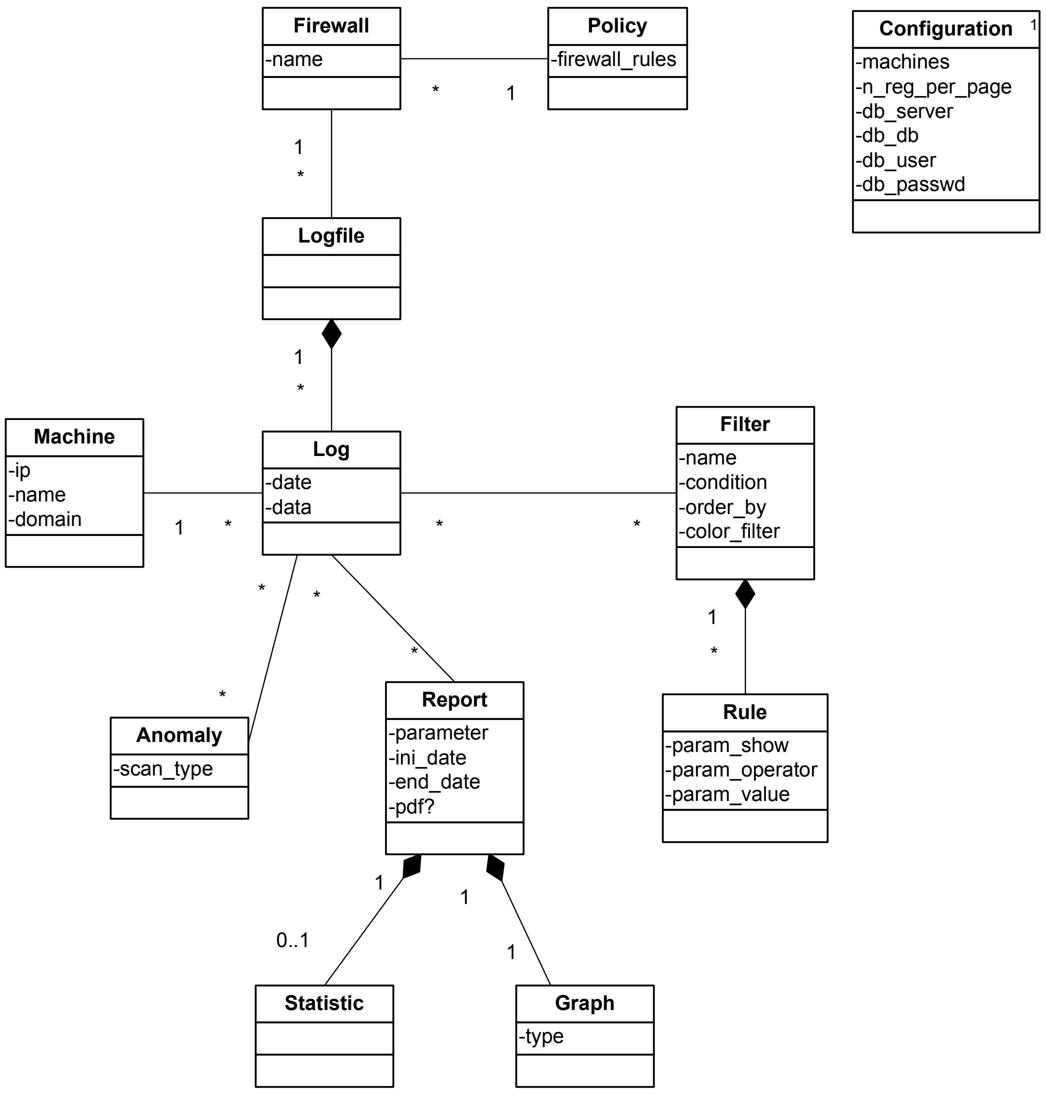


Illustration 14. Class diagram

4.2.2. Integrity constraints

The graphic models are usually not sufficient to make a good specification, so it is necessary to impose constraints.

The textual constraints of the conceptual data model are:

- ❖ A firewall could have many policies but only has one active policy. For example, a firewall by the day could have one policy and by the night another one, but only one active at the moment.
- ❖ Depends on the parameter of the report, it could have statistics or not.

4.3. Behaviour model

The objects communicated between them invocating operations from other objects. The “system” is a special kind which includes all the objects. The specification of the behaviour is done with the behaviour model of the system.

The behaviour model is composed of:

- Sequence diagrams.

  For each use case there are a sequence diagram, they show the event sequence between the actors and the system. They help us to identify the system operations.

- Operation contracts.

  The contracts describe the effects of operations in the system.

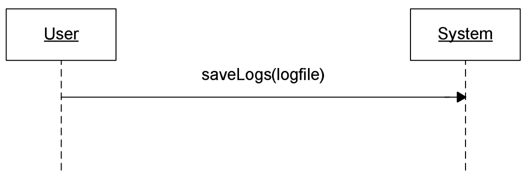
4.3.1. Sequence diagram

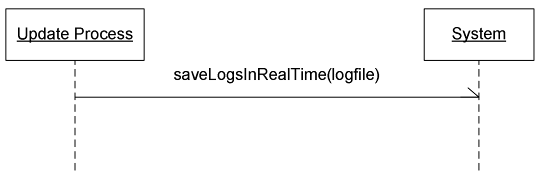
The purpose of sequence diagrams is to identify the events and the operations of the system. For doing that use case model is the point of departure. The use cases describe the interactions between actors and software system.

For a particular scene, a sequence diagram shows the external events generated by the actors, the order of them and the internal events of the system (operations) that are produced because of the external events.

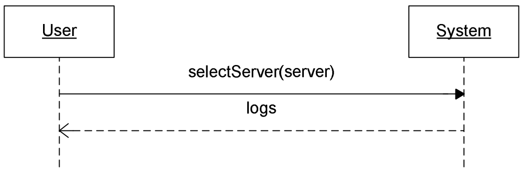
For every important flow of events in a use case, there is a sequence diagram. With the purpose of not increase this chapter, there a not included all the diagrams, only the basic flows and some alternative flow.

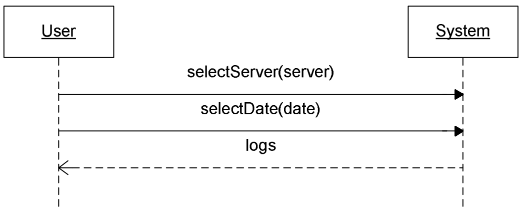
Manage Logs



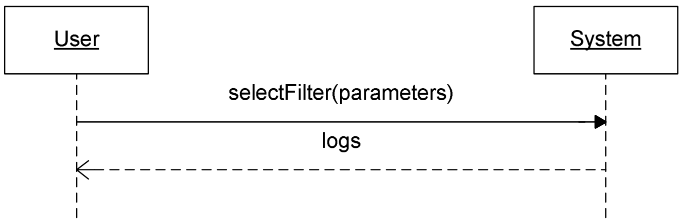


Monitor Logs

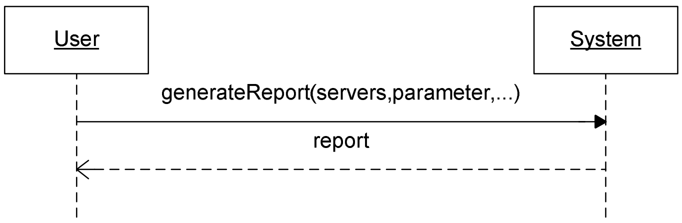




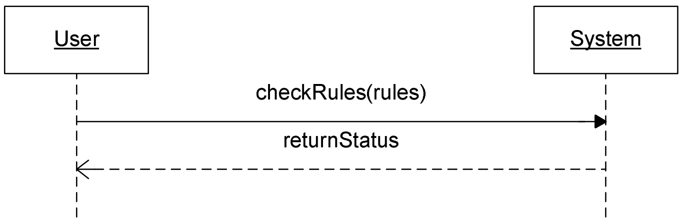
Filter Logs



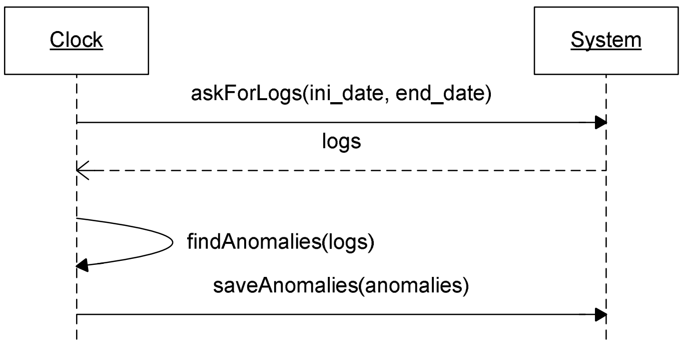
Generate Reports



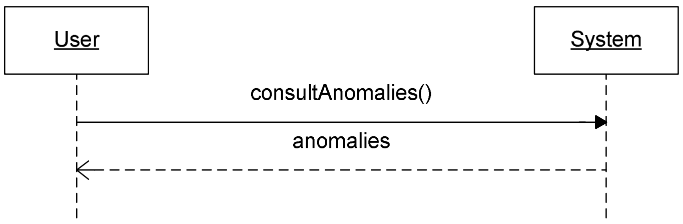
Check Firewall Rules



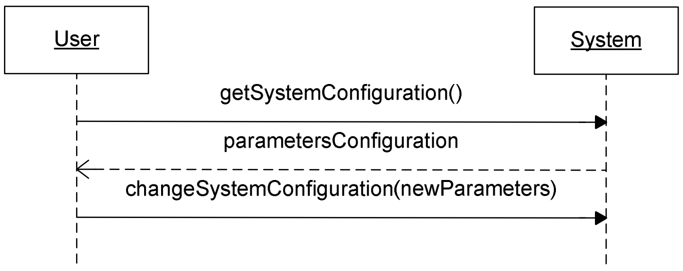
Detect anomalies



Monitor anomalies



System Configuration



4.3.2. Operation contracts

Operation contracts describe system behaviour in terms of which are the state changes of information and which are the outputs that the system provides when an operation is called.

The type of description is declarative, it means that the important is what the operation do and not how it works.

The operation contracts include:

- Operation: name and parameters of the operation.
- Semantic: informal description of the operation purpose.
- Preconditions: system state before calling the operation.
- Postconditions: changes in the system state after calling the operation.
- Output: description of the output that provides the operation.

Manage Logs

Operation:	saveLogs(logfile)
Semantic:	Parses the logfile and save the logs.
Preconditions:	-
Postconditions:	The logs are stored in the system.
Output:	-

Operation:	saveLogsInRealTime(logfile)
Semantic:	Parses in real time the logfile and save the logs.
Preconditions:	-
Postconditions:	The logs are stored in the system.
Output:	-

Monitor Logs

Operation:	selectServer(server)
Semantic:	Select the server which the user wants to view the logs.
Preconditions:	The server should be a server previously configured.
Postconditions:	-
Output:	Logs associated to this server.

Operation:	selectDate(date)
Semantic:	Select the date which the user wants to view the logs.
Preconditions:	selectServer(server).
Postconditions:	-
Output:	Logs associated to this date.

Filter Logs

Operation:	selectFilter(parameters)
Semantic:	Select the parameters (rules) which the user wants to filter the logs.
Preconditions:	-
Postconditions:	-
Output:	Logs filtered by the criteria of parameters.

Generate Reports

Operation:	generateReport(servers, parameter, ini_date, end_date, grap_type)
Semantic:	Generate a report with the data the user wants to monitor.
Preconditions:	The servers should be previously configured. The parameter should be a valid parameter.
Postconditions:	-
Output:	Report from 'ini_date' to 'end_date' with a graph of kind 'type' and statistics dependant on the parameter.

<u>Check firewall rules</u>

Operation:	checkRules(rules)
Semantic:	Check if a set of firewall rules are correct or not.
Preconditions:	-
Postconditions:	-
Output:	Return if the rules are correct or the fails they contain.

<u>Detect anomalies</u>

Operation:	askForLogs(ini_date,end_date)
Semantic:	Gets the logs stored in the system for a concrete range of time.
Preconditions:	-
Postconditions:	-
Output:	Logs stored in the system.

Operation:	findAnomalies(logs)
Semantic:	Find anomalies in the logs with different kind of scans.
Preconditions:	-
Postconditions:	-
Output:	-

Operation:	saveAnomalies(anomalies)
Semantic:	Save the anomalies in the system.
Preconditions:	-
Postconditions:	The anomalies are stored in the system.
Output:	-

Monitor anomalies

Operation:	consultAnomalies()
Semantic:	Consult the anomalies of the system.
Preconditions:	-
Postconditions:	-
Output:	Anomalies detected.

System Configuration

Operation:	getSystemConfiguration()
Semantic:	Get the parameters of the system configuration.
Preconditions:	-
Postconditions:	-
Output:	Parameters of the system configuration.

Operation:	changeSystemConfiguration(newParameters)
Semantic:	Change the parameters of the system configuration.
Preconditions:	-
Postconditions:	The parameters of the system are changed.
Output:	Only if the parameters are wrong the system warns the user.

Chapter

5

DESIGN

Software design is the application of different techniques and principles with the aim of defining a detailed system to allow their physical construction (implementation).

In the specification the intention was to know what the system should do. Now, in the design the goal is to know how the system should do the specified work. This is achieved describing subsystems and components of the system and their relations.

The proposed solution given in the next chapters is not unique, it implies that the application could have different designs and implementations. So that in this chapter the aim is to expose and to justify the taken decisions for developing the system that solves the problem.

5.1. Design decisions

A big system should be split in subtasks groups, each group of subtasks is in a determinate abstraction level.

The problem is that the system should be designed with the feature of including aspects of high and low abstraction level. The high level tasks are based in the low level ones because they can't be implemented using directly the platform services due to their complexity, so intermediate services are needed. The system also requires a horizontal structure.

The next aspects need to be balanced in design phase:

- Extensibility. New capabilities can be added to the software without major changes to the underlying architecture.
- Maintainability. Changes in the code shouldn't be spread in all the system.
- Portability. Portability to other platforms is wanted.
- Reliability. The software should be able to perform a required function under stated conditions for a specified period of time
- Reusability. The components should be reused and replaced to alternative implementations.
- Security. The software is able to withstand hostile acts and influences.

The solution is to structure the system in an appropriated number of layers[2]. The layers are disposed vertically. All the components of a layer should work at the same abstraction level. The services provided by a layer use the services provided by the layer below, and they can depend from other services at the same layer. In software engineering this is called the multi-layer architecture. The most widespread use of multi-layer architecture refers to **three-layer architecture**.

[2] The terms tier and layer are usually used interchangeably, but there is a difference them. Tiers indicate a physical separation of components, but layers refer to a logical separation of components. A multi tier project is more suitable to large size projects, whereas a multi layer design is suitable to small size projects.

Three-layer architecture is an architecture in which the user interface, functional process logic, computer data storage and data access are developed and maintained as independent modules.

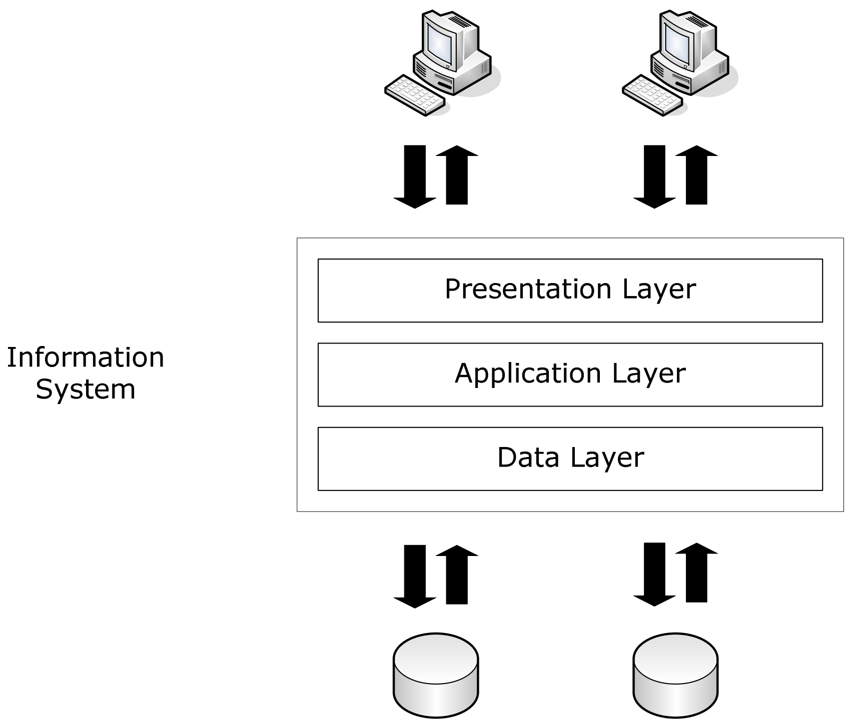


Illustration 15. Overview of a three-layer application

The **presentation layer** is the top level of the application. This layer knows the way to show the data to the users, but it doesn't know the response to user requests. The presentation layer is composed of all the components that manage the interaction with the user:

- Management of non-persistent data.
- Processing of data sent by the user.
- Transformation of data to be presented to the user.
- Validation of data entered.

The **domain layer** knows the way to satisfy the user requests, but it doesn't know where the data are stored and how they are showed to the user. The domain layer contains all the application's business components:

- Logical management of the application.
- Security monitoring.
- Transactional support when necessary.

The **data layer** knows where and how are the data stored, but it doesn't know how treat it. Its job is to provide the database systems with access components by mapping each abstraction to reality of persistent data (relational databases, file systems, etc).

In this application the use of another layer should be necessary. This layer is called **scripts**, and as its names implies in it there are scripts used to do some particular application tasks. The presentation layer communicates with scripts which treat the data and communicate with the storage system. The upper diagram is modified to include this new component.

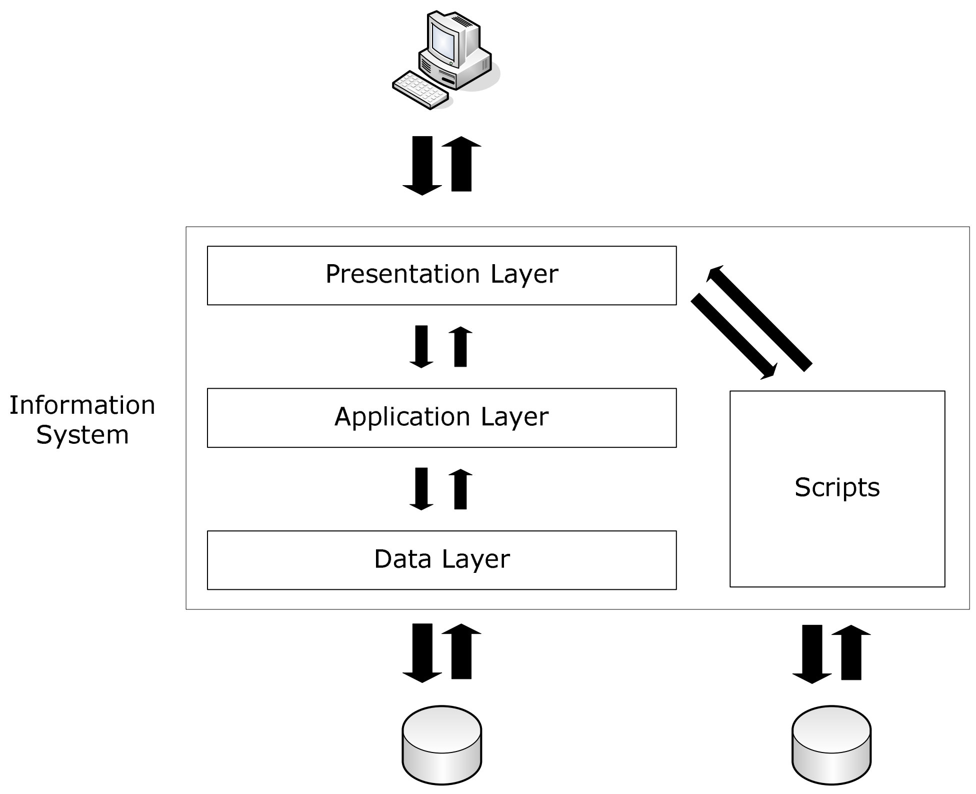


Illustration 16. Modification of a three-layer application adding a scripts layer

5.2. Presentation layer

The presentation layer is the layer that the user views, also is called user layer. It presents the system to the user, communicates him the information and captures the user information processing it in a brief mode, only do a previous filtered to check format errors. It is also known as the graphic interface and should be friendly for the user.

This layer is only communicated with the domain layer.

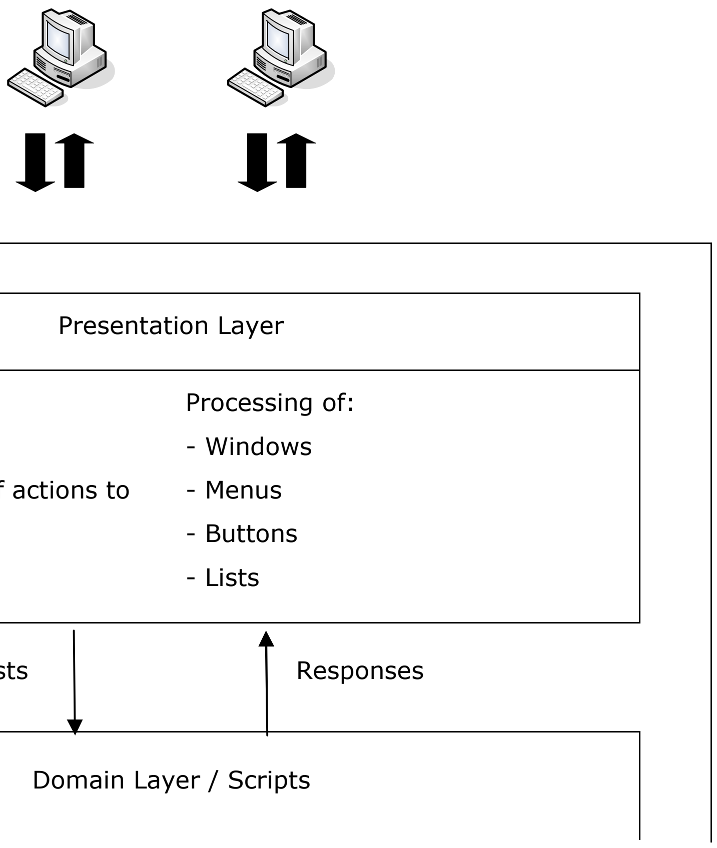


Illustration 17. Presentation layer

The presentation layer design is a process included in design phase and it is in charge to defining:

- The interaction between the user and the software system (external design).
- The interaction between the presentation layer and the domain layer and between the presentation layer and the scripts (internal design).

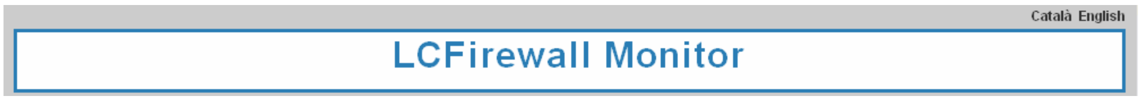
5.2.1. External design

The goal of external design is to design the elements viewed by the user when he interacts with the system. External design consists in the next mechanisms:

- Interaction mechanisms: these are the mechanisms which the user can make requests to the system.
- Presentation of information mechanisms: these are the ways to show the responses to the user requests.

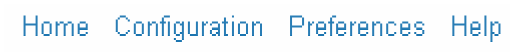
<u>Header</u>

The header is the element placed at the beginning of each application screen. It shows the name of the application and the language which is preferred.



<u>Fix Menu</u>

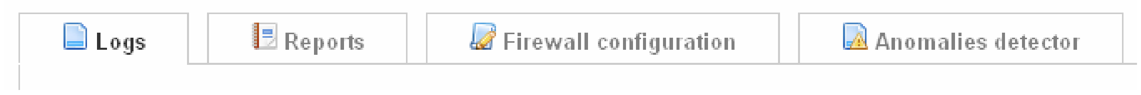
The menu is placed at the right side, over the tabs. The user can access to application complements like configuration, preferences or help.



<u>Tabs</u>

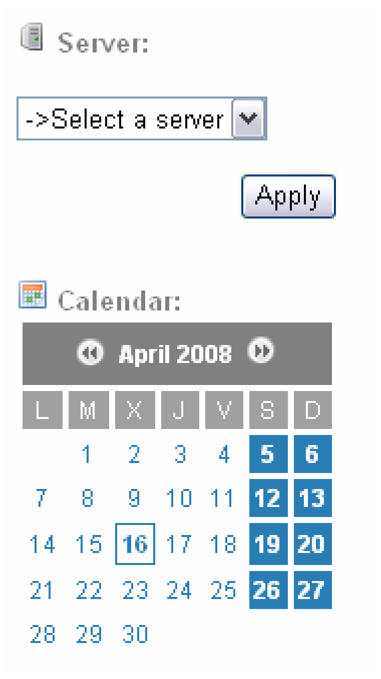
The tabs help the user to have an overview of the application because they suggest a physical space. The use of tabs is self evident, they are hard to miss. If done correctly, tabs can add polish and serve a useful purpose.

The tabs are placed below the fix menu when the user is in home. Every time the user clicks in a tab the main content and the options are changed according to the selected tab.



Options

Depending on the section the user chooses with tabs, in the left side of the page could be different options. For example in the logs screen the user can choose the server and the date:


Server:

->Select a server

Apply

Calendar:

April 2008

L	M	X	J	V	S	D
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Icons

Icons are used to make more intuitive the application.

LCFirewall Monitor logo

Logs tab

Filter option

Server option

Calendar option

First page

First page (deactivated)

Previous page

Previous page (deactivated)

Next page

Next page (deactivated)

Last page

Last page (deactivated)

Reports tab

Firewall configuration tab

Anomalies detector tab

Messages

The system shows messages to the user. Here there are some of them:

There are no results to show

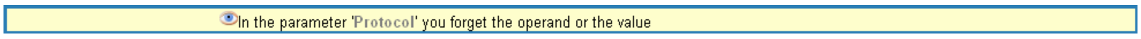
There is no chart to show

✓ Correct rules

✗ Incorrect rules

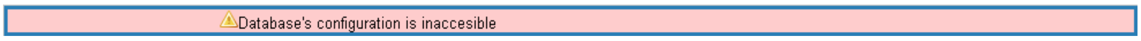
Warning messages

These messages are showed when the user does something that is incorrect, the application informs to user.


In the parameter 'Protocol' you forget the operand or the value

Error messages

These messages are showed when an error occurs in the application. The red colour favours to user to detect that an error happens.


Database's configuration is inaccesible

5.2.2. Internal design

The internal design of presentation layer consists in the design of mechanisms which caught, process and gives response to the user requests.

The objectives of the presentation layer can be divided in four parts:

- The event receiver of the presentation

  The event receiver of the presentation is a set of graphic interfaces based in events which allow the communication between the user and the server. The graphic interfaces of this application are HTML pages viewed with the web browser. So the own browser is the manager of receive all the events generated by the user and communicate them to the presentation layer.

- Management interaction with the user

  The management interaction with the user controls the presentation events reception, processes these events and generates system events to the elements which have to process it.

- Presentation information

  The presentation information receives the data to be presented to the user and transform it in the specific user formats.

- Communication with the domain layer and the scripts

  With the chosen architecture the presentation layer should communicates with the domain layer and the scripts in order to call the corresponding events and receive the responses.

5.3. Domain layer

The domain layer is the manager of execute the actions, controls the data validity, changes the state of application, makes requests to the data layer and communicates the results to the presentation layer.

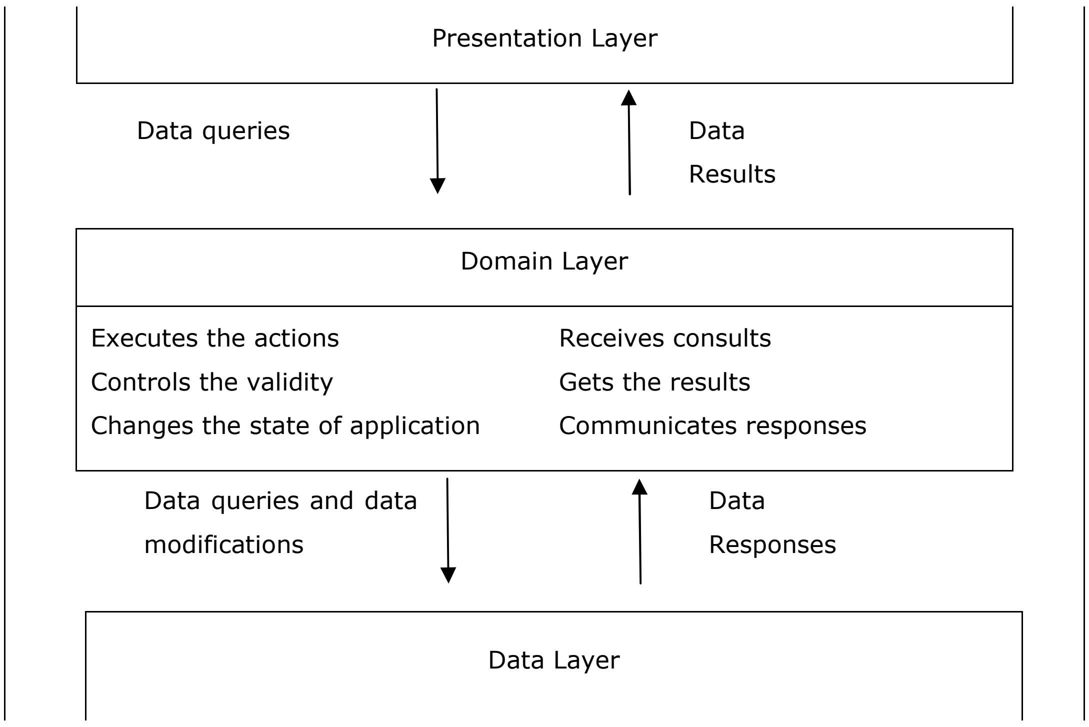


Illustration 18. Domain layer

The design of this layer depends on the design pattern used. A design pattern is a description or template for how to solve a problem that can be used in many different situations.

A design pattern provides a schema for polish the subsystems or the components of a software system and the relations between them. It describes the recurrent structures to communicate components that solve a design problem in a particular context. They are patterns with an abstraction level lower than the architectonic patterns, so they are closest to the final code. Their usage is not reflected in the global system structure.

In this section is explained how to solve the first problem found in the domain layer. The software systems receive events and when they are intercepted some component of the system should receive them and execute the correspondent actions. The problem is to know the responsible of receive this event. The solution is to assign this responsibility to a **controller**.

A controller is an object of a certain class. It delegates to one or more objects the event processing. The object who processes the event doesn't know the kind of controller. There are different controller types:

- Front Controller: an object represents all the system.
- Use Case: an object represents a use case instance.
- Command: an object represents an event instance.

In this system the chosen type is the use case controller. For every use case defined in the system is associated a use case controller. In the static aspect, there are as many classes as use cases the system has. As in the architecture design there are the domain layer and the scripts layer, the use cases are divided into these layers so that not all the use cases are in this layer.

The uses cases associated with this layer are the next:

⊙ Monitor Logs

When the user selects a configure server (or all the servers) he can see the logs stored in the system. The consult is delegated directly to the Log class and it passes the consult to the data layer.

The logs can be filtered by date, so that the application needs a special controller for the calendar functions. This calendar will be used in another use cases like "Generate Reports" or "Monitor Anomalies".

⊙ Filter Logs

The filtered of the logs is an extension from the use case "Monitor Logs", the logs the user see can be filtered to show only the data that the user considers relevant.

To do this, the domain layer needs to know the fields to show and the restrictions of the filtered. These restrictions can be from field, or general restrictions like the order or condition operator. All these data is passed to data layer and the results return another to the presentation layer.

The filters can be saved, this is an important feature. The user can recover, modify or delete a filter after save it. The filters are distinguished by its name, two filters can't exist with the same name.

⊙ Generate Reports

The reports are generated dynamically, the user generates it when he is interested in some concrete data. It means that they aren't stored by the system, so if the user wants to store them he has the possibility to create them in PDF format and then save them.

The user can choose if he wants to make the report from one or more configured servers, then he choose the field about the report is approach. He also selects a range of dates, and the kind of graph he wants to see: pie, vertical bars, horizontal bars or lines.

The report is the graph with the parameters which has been chosen by the user and a serial of statistic data like arithmetic mean, median, mode, variance, standard deviation and coefficient of variation.

⊙ Monitor Anomalies

The anomalies previously detected by the use case "Detect anomalies" are stored and they can be viewed by the user when he wants. The anomalies can be filtered by date.

The user also can see exactly which logs are involved in a concrete anomaly, the logs data are filtered to see exactly the logs the user wants to see. For example for a specific day, scan type (they are explained below in the section *Scripts*, in the use case "Detect anomalies"), source address and destination port.

- Configure System

  To change the configuration of the system, first the system has to check if the data provided by the user is correct. This is made with a simple connection to the database with the user and password given. If the data are correct they are stored to future use, if not the user is advised about it.

  Another parameter that the user can configure is the number of logs per page he wants to see. There are no restrictions to configure this value.

Then the domain layer will have six controllers, one for every use case described above and another for the calendar. These controllers receive the events from presentation layer and as soon as be possible they delegate responsibility to other objects.

5.4. Data layer

The data layer knows where and how is the data stored. So the functions of this layer depend on the management of the data.

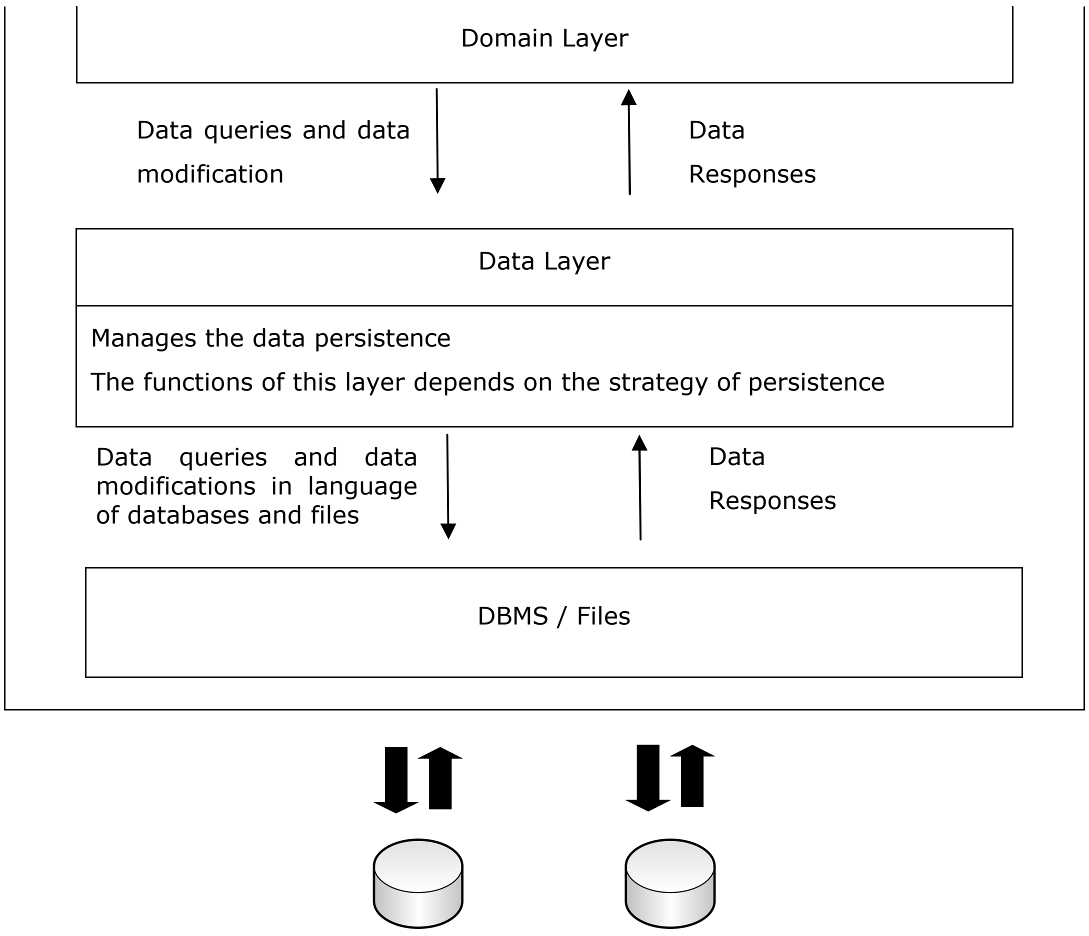

Domain Layer
Data queries and data modification
Data Responses
Data Layer
Manages the data persistence
The functions of this layer depends on the strategy of persistence
Data queries and data modifications in language of databases and files
Data Responses
DBMS / Files

Illustration 19. Data layer

The persistence is the ability that many applications require to store and obtain data using a permanent storage system. For this reason the systems which need persistence in its data have a data layer.

In data layer the storage technology is important, because is not the same to have an object oriented database, a relational database or other kind of database. In this project the DBMS is a relational database.

The benefits to use a relational database are:

- Avoid errors.

  Data updates are simple with no need to change same info in several different files. Relational databases avoid data-typing problems. Data are validated on entry to filter impossible values.

- Manipulate data.

  Combine different datasets easily and efficiently. Data manipulation is performed using queries. These queries use SQL (Structured Query Language) to combine, update and manage data.

  The data are introduced only one time in a table, thanks to the relations these data can appear in any table the user wants. Any modification only should be done one time and automatically it was done in the rest of tables related with this one.

5.4.1. Database design

<u>Firewall</u>

For each firewall monitored there is a table like that to save all the logs.

Field	Type	PK	Extra	Description
id	int(10) unsigned	*	Auto increment	Log identification
raw_mac	varchar(80)			MAC identification
oob_time_sec	int(10) unsigned			Time and date (seconds)
oob_time_usec	int(10) unsigned			Time and date (milliseconds)
oob_prefix	varchar(32)			Prefix
oob_mark	int(10) unsigned			Mark

oob_in	varchar(32)			Input interface
oob_out	varchar(32)			Output interface
ip_saddr	int(10) unsigned			Source IP address
ip_daddr	int(10) unsigned			Destination IP address
ip_protocol	tinyint(3) unsigned			Protocol
ip_tos	tinyint(3) unsigned			TOS (Type Of Service)
ip_ttl	tinyint(3) unsigned			TTL (Time To Live)
ip_totlen	smallint(5) unsigned			TOTLEN
ip_ihl	tinyint(3) unsigned			IHL
ip_csum	smallint(5) unsigned			Checksum
ip_id	smallint(5) unsigned			Id
ip_fragoff	smallint(5) unsigned			Fragment offset
tcp_sport	smallint(5) unsigned			Source TCP port
tcp_dport	smallint(5) unsigned			Destination TCP port
tcp_seq	int(10) unsigned			Sequence number
tcp_ackseq	int(10) unsigned			ACK sequence number
tcp_window	smallint(5) unsigned			TCP receive window flag
tcp_urg	tinyint(4)			Urgent flag
tcp_urgp	smallint(5) unsigned			Urgent pointer
tcp_ack	tinyint(4)			ACK
tcp_psh	tinyint(4)			Push flag
tcp_rst	tinyint(4)			RST (reset) flag
tcp_syn	tinyint(4)			SYN flag
tcp_fin	tinyint(4)			FIN flag
udp_sport	smallint(5) unsigned			Source UDP port
udp_dport	smallint(5) unsigned			Destination UDP port
udp_len	smallint(5) unsigned			LEN
icmp_type	tinyint(3) unsigned			ICMP type

icmp_code	tinyint(3) unsigned			ICMP code
icmp_echoid	smallint(5) unsigned			Echo id
icmp_echoseq	smallint(5) unsigned			Echo sequence
icmp_gateway	int(10) unsigned			Gateway
icmp_fragmtu	smallint(5) unsigned			Fragmentation MTU
pwsniff_user	varchar(30)			IPSec user
pwsniff_pass	varchar(30)			IPSec password
ahesp_spi	int(10) unsigned			IPSec AH/ESP SPI

Table 3. Firewall class persistence

Filters

Contains information about the filters the user decides to save.

Field	Type	PK	Description
id_name	varchar(32)	*	Filter name
valores	varchar(500)		Values to show
param	varchar(1000)		Parameters restrictions
orden	varchar(50)		Order ascend or descend
color	varchar(50)		Colour restriction

Table 4. Filters class persistence

Anomalies detector

Contains information about he anomalies detected by the system.

Field	Type	PK	Description
time_sec	int(10) unsigned		Date
servidor	varchar(50)		Affected server

tipo	varchar(50)		Type of anomaly
ip_orig	int(10) unsigned		Source IP address
ip_dest	int(10) unsigned		Destination IP address
p_dest	smallint(5) unsigned		Destination port

Table 5. Anomalies class persistence

5.4.2. Configuration file

A configuration file must be necessary to save the system data. The data about the parameters of the application will be saved in this file. The stored data in the file should be organized to recover it in an easy and fast way.

Why does the configuration be stored in a file instead of a database as the rest of the application persistence? Because in this configuration there is the own data of the database, and it would be redundant and cyclic if this data would be contained in the database.

This configuration will be easy to change directly. This application is thought for system administrators, so it is usually that they edit configuration files. Anyway, the application works via web, so that they can change this parameters via web.

The parameters that are stored in the file are:

Server of database
Name of database
User of database
Password of database
Other parameters (They can be stored in the database, but it is considered that create a table to store a few values isn't necessary, and they can be stored with the rest of the system parameters)

Table 6. Configuration file persistence

5.5. Scripts

The scripts are a special layer to unify the presentation layer with the systems job associated to this project. They have functionalities from domain layer and data layer because they are the manager of execute the actions, controls the data validity, changes the state of application, makes requests to the DBMS, manages the data and communicates the results to the presentation layer.

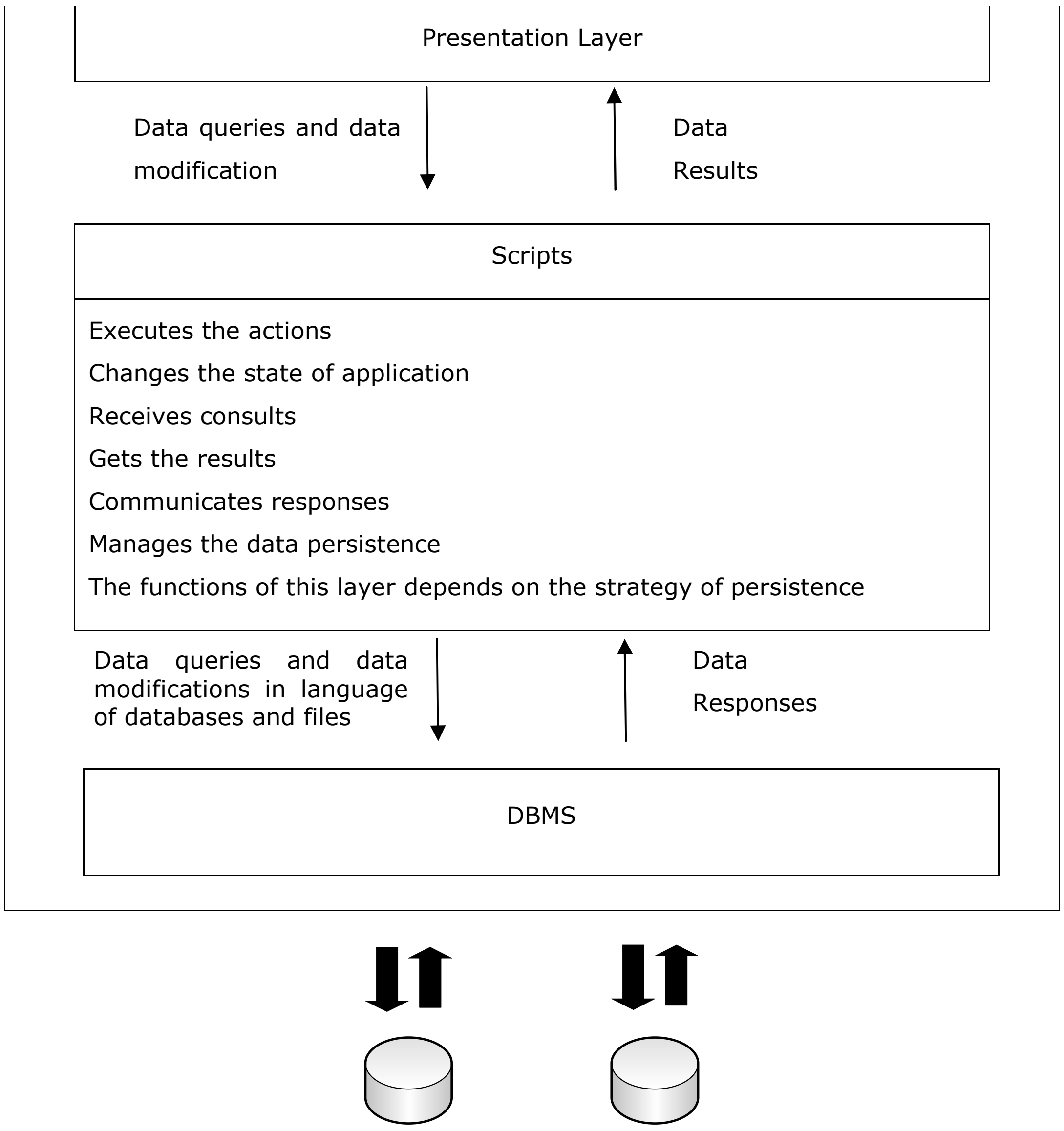

Presentation Layer
Data queries and data modification
Data Results
Scripts
Executes the actions
Changes the state of application
Receives consults
Gets the results
Communicates responses
Manages the data persistence
The functions of this layer depends on the strategy of persistence
Data queries and data modifications in language of databases and files
Data Responses
DBMS

Illustration 20. Scripts layer

The decision to make scripts instead of follow the typical three layer architecture was taken because it is the typical way to work in systems tasks. The use of scripts is more intuitive and easier to implement.

In scripts the storage technology is important, like in data layer, because they interact directly with the DBMS. So when the decision of choose a DBMS will be taken it is important to think that the scripts layer will have to interact with the database too.

The tasks to do with scripts are the associated to the next use cases, they are the ones which don't go to domain layer:

- Manage logs

  The management of the logs is done in this layer because the action of catch the logs and store them is a task that manages files, process the information and access to database. The task of both actors is distinguished for the reason that there is a little difference in the time they process the file.

  The actor User wants to save the logs that he has saved in a file. With this script the file is parsed, the logs are extracted and finally they are stored in the database.

  The actor Update Process is constantly monitoring a file in which the firewall writes the logs. These logs are parsed and stored in the database, with the same criteria that in the previous case.

- Check firewall rules

  The actor can check the firewall rules when he wants. The rules are passed through the presentation layer and in this layer they are parsed to check if there is any syntax error.

  The script that makes this task should work with regular expressions. In this case is not necessary to access to the database like in the other tasks of the scripts layer.

- Detect anomalies

  This is a periodical task, so the typical way to do this kind of task is to do a script and then it will be executed by the actor when he considers it.

  The actor Clock periodically checks the stored logs and analyzes them to detect the possible anomalies in the monitored logs. The type of scans will be horizontal scans, vertical scans and other personalized scans.

  The horizontal ones scan for a concrete source address that access to specific port or ports across a range of hosts in a certain time.

  The vertical ones scan a range of ports on a specific host in a certain time.

  The personalized scans are for specific situations, for example when some source address is making ping across a range of hosts, or other anomalous situations.

Chapter

6

IMPLEMENTATION

The main goal of this phase is to implement the design which was obtained in the previous phase. In the implementation all the development done in the previous phases is put into practice by codifying the system with specific technologies.

First the technologies to be used will be chosen and the reasons which to select this technologies will be explained. Then some details of the implementation that has been considered relevant for understanding the operation of the system will be commented, is useful to do this explanation so that the reader can understand better the application. Finally the system is testes to check whether its requirements are accomplished.

6.1. Election of technologies

In chapter 2 a detailed description of technologies and software was given. Now in this chapter is showed which of those technologies have been chosen and why they have been elected to develop this application.

6.1.1. Operating system

The operating system used for the development of the application has been **Linux openSUSE** 10.2. The development has been done in a LCFIB's server that was exclusive for carrying out the project.

The reason why this operating system has been chosen is because we want to simulate an environment the most similar possible to the server where the application will be installed in production. The most of the servers that the LCFIB staff use for monitoring applications are Linux, specifically openSUSE.

Anyway the programming languages and the web server are available for multiple operating systems. This means that the application can be used in different operating systems, so that there is flexibility about the choice of the operating system in which the application can run.

6.1.2. Web server

Apache was the web server elected between the three proposed. The version of the web server is Apache 2.2.4.

As the server is Linux, the selection of the web server is reduced to decide between Apache or Tomcat, because they are most famous web servers which runs on most Linux based servers. The main feature of Tomcat is that can run JSP pages, and that isn't the case of this application, so that finally Apache has been chosen.

Apache offers various advantages to users, developers and web administrators:

- Features. Apache has various useful features, including implementation of the latest protocols.
- Customizable. Apache’s modular architecture allows you to build a server that is “made to measure”.
- Administration. Apache configuration files are in ASCII. They are transferable, so one can effectively clone a server.
- Extensible. Apache server and API source code are open to public.
- Efficient. Apache’s C code is optimized for performance, as a result, it runs faster and consumes less systems resources than many other servers.
- Portability. Apache runs on a wide variety of operating systems.
- Stability/Reliability. Apache’s source code is open to public. When any bug is found, they are quickly communicated, and rapidly fixed. Updates are made and announced thereafter.
- Support. Apache is supported by the Apache Group, a large number of dedicated users.

6.1.3. Server languages

The majority of the development is coded in **PHP5**, this is one of the server languages described above. The version installed in the development server is PHP 5.2.2.

ASP was ruled out because it runs in the IIS. Java at the begin was for running client side applications, the problems associated with this applications has led Sun and many other Java developers to use the language in other ways, so that now can allow web sites to connect to databases and produce other server-side applications, or "servlets." A beginner with no programming background will find it difficult to begin working with servlets, because of the complexity of the language as well as the complicated JSP system design. Knowing that, Java technologies were not elected and PHP is the candidate to do the development. A little bit of knowledge about PHP are also taken in account to make this decision.

PHP uses are widespread, and can include any kind of server functionality that takes user's input and displays or manipulates the input. PHP can run on both UNIX and Windows servers. PHP5 a fully object oriented language and its platform independence and speed on Linux server helps to build large and complex web applications. PHP is a particularly useful programming language because is easy to integrate with web pages.

A disadvantage is that the server is the manager of do everything, don't delegate to client anything, so that it can be more inefficient insofar the requests number grow. Other disadvantage is that the mix of HTML and PHP sentences could affect to the code legibility. Both disadvantages aren't taken in account because the number and the importance of advantages are sufficient to decide use PHP.

Another server language used for develop the application is **Perl** in his version Perl 5.8.8. It is used specifically for scripts, they do some functionalities related directly with regular expressions and the use Perl is an easy way to implement them.

The main features used of Perl are the portability, string processing and especially regular expression support, the CPAN has a huge collection of free and reusable Perl code for many purposes.

6.1.4. Database management system

The database management system is a strong point of application because all the logs will be stored in the database. These logs will be the source of most of the requirements. The chosen system was **MySQL5**, the version installed for the development is MySQL 5.0.41.

Oracle seems to be the best option but the highest economical cost is the main reason to suppress it for the list of available technologies. Now the decision is between PostgreSQL and MySQL. Some reasons for using MySQL over PostgreSQL are that MySQL are relatively faster than PostgreSQL, database design will be simpler and it is designed to work well with web based servers. Although PostgreSQL has more features than MySQL, it isn't an obstacle to prefer MySQL.

6.1.5. Web page design

The web page design is done with the traditional **HTML** language. HTML has a fixed set of tags to work with, and can define content and structure all together.

Thanks to the **CSS** design patterns, the content and the presentation are separated. Probably the mostly useful feature of CSS is that all of the style and layout is removed from the HTML, so the html page size is very much smaller. A disadvantage of CSS is that does not work consistently in different browsers, the application was tested with Mozilla Firefox and Internet Explorer and works fine.

Moreover, **Javascript** has been used to define some necessary functions to the correct application performance that can’t be done with another technique. Javascript has problems of security because users can disable it and it can be exploited for malicious purposes. So that Javascript code has been used the minimum possible, the idea was don’t use it very much.

6.2. Implementation details

In this section the most important implementation details are described. These details are referred to the implementation phase.

6.2.1. Internationalization

The internationalization is the characteristic of adapt to various languages the application. The changes to make in the application don't entail engineering changes. So according to the decisions taken by the moment, a system to do the internationalization should be thought.

Most of the application is done in PHP language. It's logical that the methods to do the change of language has been done is PHP too. The used method is that when the application needs to insert a text, it has been done inserting a variable. This variable will change depending on the language chosen.

For example, inside of type in the page code:

```
echo "There are no results to show"
```

In our application is typed:

```
echo $LANG['error']['noresult']
```

And the value of this variable will be in a file with the value:

```
$LANG['error'] = Array(
      'noresult' => "There are no results to show",
      ...
```

The $LANG variable is in a separated file and it contains all the texts that appear in the application. For every language there is a distinct file. They are called lang_en,php, lang_es.php, lang_ca.php, and they correspond for everyone of the languages of the application.

Initially the application is available in English, Spanish and Catalan. But it is easy to extend it to allow other languages. Only it's necessary to:

1) Create a file called lang_CODE.php (CODE is the code for the representation of names of languages) and put in the directory with the other language files.

2) Add the new language in a couple of variables of the conf.php file, the configuration file of the application.

This is made possible thanks to the features of PHP that allow include files, use variables in the name of files or other characteristics.

6.2.2. PHP Libraries

Some external libraries to PHP are used for the development of certain functionalities. These libraries are made with the PHP basic libraries and it is preferred to use these libraries because they are easier and more intuitive to use. Moreover they spare to use the PHP basic libraries to do these tasks.

Libchart

Libchart is a free chart creation PHP library. It can generate bar, line diagrams or pie charts. It is compatible with PHP5, the PHP should be compiled with GD and FreeType support to work fine. Libchart doesn't require other external dependency.

Libchart is used in the use case of "Generate Reports". It is used for draw the graphs of the reports. These graphs are created dynamically, so they aren't stored in any place of the application because it is considered that if the user needs any report in concrete they can create it.

License: Libchart is free software (Open Source), distributed under the terms of GNU GPL. All source code is freely available for everyone.

FPDF

FPDF is a free PHP class which allows to generate PDF files with pure PHP, without using the PDFlib library which is for payment. It is compatible with PHP4 and PHP5, only requires the zlib extension to activate compression.

The advantages of FPDF are the high level functions. Next is a list of it main features:

- Choice of measure unit, page format and margins
- Page header and footer management
- Automatic page break
- Automatic line break and text justification
- Image support (JPEG and PNG)
- Colours
- Links
- TrueType, Type1 and encoding support
- Page compression

FPDF is used in the use case of "Generate Reports". It is used for generate the reports in PDF format. It is an optional feature of the reports, with it the user can save this report for another future usage.

License: FPDF is free software, there is no usage restriction. It may be embedded freely in the application, with or without modification. It may be redistributed, too.

6.2.3. Perl modules

The modules of Perl are downloaded from CPAN (Comprehensive Perl Archive Network). CPAN is an archive of modules of software written in Perl, as well as documentation for it. Modules are the Perl mechanisms to use external libraries of code, allowing a single file to contain common routines used by several programs.

The Perl modules are used in the scripts layer and are the next:

DBI

DBI is the database independent interface for Perl. It defines methods, variables and conventions that provide a consistent database interface, independent of the actual database being used.

This module is used to interact with the database. In the use cases of "Manage Logs" and "Detect anomalies", the application should access to the database to store the compiled data.

File::Tail

File::Tail is a Perl extension for reading from continuously updated files. The purpose is reading and analysing log files while they are being written, this is very useful for monitoring the logging process.

This module is used in the use case "Manage Logs", it is the manager to wait the logs in the file specified by the user. When a new log is generated then the script parses it and save it in the database. It is only used by the actor Process Update that treats the logs at the moment, the actor User analyzes a file in the moment like was explained in the sections *Specification* and *Design*.

Date::Parse

Date::Parse is a module which parses date strings into time values. It only provides two routines. The first parses a date and returns a Unix time value. The second parses a date and returns an array of values (second, minute, hour, day, …).

This module is used in the use case "Manage Logs" to extract the date from the logs and then store in the database with the rest of data.

6.2.4. Ulogd

In the use case of “Manage Logs”, with the actor Process Update, if IPTables is used there are two ways of store the logs in the database: with the application script (explained in the chapter *Design*) or with ulogd. Both do the same, the user can choose the method to carry out this task.

Ulogd is a userspace logging daemon for IPTables related logging. This includes per-packet logging of security violations, per-packet logging for accounting purpose as well as per-flow logging.

To activate ulogd, in the IPTables firewall rule the target ULOG should be assigned. The possible options are --ulog-nlgroup, --ulog-prefix, --ulog-cprange, --ulog-qthreshold.

It is important to understand that ulogd without plugins does nothing. It will receive packets, and do nothing with them. The plugin used for this application is the ulogd_MYSQL.so. It is an output plugin for logging into a mysql database. This is only compiled if you have the mysql libraries installed, and the configure script was able to detect them. The plugin automatically inserts the data into the configured table. The module defines the following configuration directives:

table	Name of the table to which ulogd should log.
db	Name of the mysql database.
host	Name of the mysql database host.
port	TCP port number of mysql database server.
user	Name of the mysql user.
pass	Password for mysql.

Table 7. Configuration directives of ulogd_MYSQL plugin

The parameters are changed in the configuration file ulogd.conf. And finally the ulogd daemon should be started.

6.3. Tests

Software testing is an important part of the software engineering to obtain a software system that works fine. The goal of the tests is to determine if the implementation of the application satisfy the functionalities, the restrictions and the requirements established in the specification phase. A good known of the application should exist to do this job.

A problem with software testing is that testing all combinations of inputs and preconditions is not feasible. The number of errors in an application can be very large and the errors that occur infrequently are difficult to find in testing. Furthermore, more dimensions of quality (usability, reliability, etc) can be very subjective, something that constitutes sufficient value for a person may be intolerable to another. For all this tests will be done carefully.

The tests can’t be limited to the last phase of the system development once the implementation is finished because this can put at risk the project success. So the tests have been done during the implementation, checking that they work as was planned. If the result wasn’t the expected, the implementation was modified correcting the error and the test has been done another time. This way, the task of check the application has been distributed along the implementation phase.

The features checked in the tests are the detection of incorrect functions, interface errors, access to database errors, performance errors, execution errors and any kind of anomaly. All these errors were tested following all the possible paths.

The tests were done by the application developer. These tests were more technical thanks to the knowledge of the system. They can be deeper in the system architecture and can detect several errors that maybe a normal user never arrives to this situation.

Also the LCFIB staff, who will be the future users, was testing the application. They prove mainly the presentation aspects and the functionalities without go deep into the internal operation. They make load charge about the capacity of process the logs and then this data will be contained in the database and will be accessed by the application. This kind of testing is called black box testing, it treats the software as a black-box without any understanding of internal behaviour.

However, the tests can't assure the absence of errors, they only can detect the presence of them. A test is the execution of a system functionality to detect an error. If this error is executed, in certain situations the system will produce wrong results, causing a failure. Not all defects will necessarily result in failures, for example, defects in a dead code never result in failures.

For all the explained above the conclusion about the test phase is that the tests done try to examine all the possible failures, but how it's described along this section is not possible to make a perfect application.

6.3.1. Set of tests

Next is a list with the set of tests that has been done to check the features of the application.

- Usability.

  The usability is the clarity in the interaction with the application. This feature was tested by the final users, because the opinion of the developer is not objective in this field.

  The access to the different tasks of the application is direct and unambiguous. The functionalities are easy to learn so the use of the application is satisfactory to the user. In this aspect the users does not have many complaints because from beginning these was one of the non-functional requirements.

- Interface errors.

  The interface errors should be checked because this is the face of the application. These errors can be wrong format in the page, bad translations, unexpected errors or any error related to the interface. In all the development phase this kind of errors was being checked.

o Access to database errors.

The database errors can occur if there is an invalid request from the application to the database. It can occur if the user introduces incorrect data from delivered or undelivered way. The application should be used by the system administrators so it is not logical that they want to hack the application, but anyway these errors should be solved.

Anyway, the application checks any entry before pass to another layer of the application, so the errors can not be propagated. If the data is not valid the application inform to the user so he can correct the mistake and then the petition to the database will be correct.

o Performance errors, execution errors and any kind of anomaly.

Performance testing can serve different purposes. It can demonstrate that the system meets performance criteria. It also can detect the execution errors or any kind of anomaly in the application.

These tests are very useful in order to polish some aspects of the application and to correct errors that can be detected before because in the development phase some aspects can be overlooked.

o Data load testing.

The load testing is the process of creating demand on a system or device and measuring its response. Specifically the data load is tested, because the volume of data can be high in this type of application. This kind of test also is called volume testing.

First was tested to load a big file with the functionality of load file. The load of the file it is limited by the upload speed of the client, because in the server side works quickly. It is important that the server was configured first to support the file uploading of large file size.

Then it was tried to monitor a firewall with high activity. These firewalls generate a lot of logs and it is good way to test the volume testing dynamically. In the set of the tested firewalls, the ones with most traffic have between 600,000 to 1,000,000 logs per day.

The application supports a big number of logs, but it seems to be slowly when it is charged a very big quantity, the order to 600,000 logs in the development server. It is important take into account that the development server does not have the same good features as the production server that can support more data load. The response time with the same charge is around three times lower than in the development environment.

The user should be responsible of the logs that his firewall writes, it is not necessary to log everything, only the important data to be monitored, so the application won’t be charged in excess. The bottleneck is the MySQL database, but with a conscious log firewall policy there is no problem with the overload.

Chapter

7

CONCLUSIONS

When the project is finished it’s time to draw conclusions about it. In this chapter are explained in detail all these final conclusions.

First are checked if the goals proposed at the beginning of the project were accomplished. Next, the initial planning was reviewed with the aim of compare the time stipulated at the start with the real time. Also an economical study of the developed system has been done.

The future lines of the project are showed, these lines can be followed in future projects to add functionalities to he actual application.

Last, the knowledge acquired by the student in the development of the project and the license under is distributed this application are showed.

7.1. Accomplished goals

A balance between the initials project goals and the accomplished ones can be done once the project is finished having in account the goals planned at the beginning. All the goals have been accomplished like it has been reflecting along the report:

- *Basic monitoring of firewall logs that the user wants to view, so the user can see what is happening in real time or logs from another time prior to the current date.*

  This goal was the base of the project so it was the first step done. The user can select the configured servers and a date of what logs he wants to monitor. The goal is accomplished and it is the main functionality.

- *Logs are filtered to have a better view of what the user wants to monitor. The application offers the possibility of managing the own filters that were created by the user.*

  With the option of filter the logs the monitoring can be more selective. This aids to the user in the task of watch what is happening in the monitored network. The user can create, modify, delete or restore filters. He also can order the logs by any of their fields.

- *Compatibility with different log formats, in order to have in a single tool a centralized monitoring of the different firewalls.*

  This is a feature implemented from beginning. The application is compatible with IPTables and IPFilter log format.

- *Completion of reports using statistics of the data that the user previously selects for this purpose, with their respective graphical representation. These graphs are necessary to facilitate the understanding of all the data that the firewall logs can contribute.*

  The reports are created on demand, it allows create personalized reports of a concrete date range with the parameters that the user are interested in. There are the option of create the report in PDF format, so it can be saved for a future use.

- *“Strange” package detection in logs. It can help to detect any anomalies in the system.*

  The detection is done every day. There are detected the anomalies from the day before. Then the user can view these anomalies and filter the concrete logs of this anomaly to find more information about it.

- *Incongruities detection in the configuration files of the monitored firewalls to prevent a possible security failure.*

  The method to detect failures in the configuration file is adding the configuration rules and then the application will check it. It is obvious that the application couldn’t access to the configuration files of each configured firewall.

Besides this goals also are implemented the way to set the configuration of the system from a simple interface. This is not directly a goal but it is a thing to have into account, so it is a use case like was seen in the corresponding chapters.

During the implementation there are some things that were finished and work correctly. But a best solution was found and then they were changed. The main idea was to make a good application with facilities to modify it if will be necessary in the future.

7.2. Planning revision

At this point of the project it is the time to analyze the initial task planning about the estimation of time in which each phase must be developed. The possible deviations respect the initial planning done at the start of the project are determined and analyzed in this section.

First is necessary to calculate the real hours spent to carry out the project and then compare it with the planned hours. In the next table is showed this comparative:

Phase	Planned hours	Real hours
Previous study	**70**	**70**
Programming languages	14	14
IPTables, IPFilter	36	36
Collect information of another products	20	20
Application development	**700**	**720**
Specification / Design of application	20	50
Main engine	190	190
Filtered	60	70
IPTables / IPFilter	100	100
Reports	100	120
Check the configuration file	90	90
Detect attacks	140	100
Tests	**60**	**60**
Documentation	**60**	**140**
TOTAL	**890**	**990**

Table 8. Project phases of planning revision

A deviation around 100 hours can be observed analyzing the obtained results. This means that the work was delayed 3 weeks approximately. This time will be added at the initial planning.

The causes of this deviation are mainly the delay in the phases of specification/design and documentation. There are other deviations in implementation but there are complemented between them, so these two phases are the most significant.

On one hand, the specification and the analysis was a little confusing because I never do a similar application with some part with web design and some one with scripts design. Finally a satisfactory job was done. The intention of this phase wasn’t an exhaustive analysis and design because also was taken some decisions during the implementation of each functionality, so it was taken fewer hours than in a normal project.

On the other hand, the documentation was the main cause of delay. The initial planned days have proved short. The fact of writing this document in English may have contributed on slowing progress down.

7.3. Economic evaluation

The total costs of the project are divided in three parts, in order to realize a detailed economic evaluation of it. These parts are:

- Human resource costs
- Hardware costs
- Software costs

Human resources costs

The human resource costs are estimated depending on the different profiles of the workers involved in the development of the project. The profiles that have been participated in this project are:

- System analyst. It is the responsible of the previous study, the analysis of requirements, the specification and the design.
- Developer. It is the responsible of the implementation and the tests.

The following table displays human resource costs detailing the professional profile of the worker, accumulated work time and estimated per hour salary (source: LCFIB):

Worker type	Hours	Price / hour	Cost
System Analyst	290	35 €	10150€
Developer	700	25 €	17500€
Total			27650€

Table 9. Human resource costs

Hardware costs

Hardware costs include a server where the application is installed, and an additional computer for development tasks:

Hardware	Cost
1 Personal computer	1100€
1 Server	4500€
Total	5600€

Table 10. Hardware costs

The network infrastructure used is the network existent in LCFIB. The connection to the external network is done through the Scientific Ring (Anella Científica), so it does not entail supplementary cost.

Software costs

The software used in the work environment (Windows XP, Office 2003, SSH Secure Shell) was used because the LCFIB provided it, but it is possible to carry out the project using alternative opensource software, so this cost was not included in the software costs.

All the rest of software used in the application (Eclipse) is opensource software, so no additional software cost needs to be added.

TOTAL COSTS

Resource	Cost
Human resources	27650€
Hardware	5600€
Software	0€
Total	33250€

Table 11. Total costs

7.4. Future prospects

The immediate future plans for this project are probably the addition of new features or functionalities to the application.

Some ideas that have been emerged since the end of the development of the application are:

- Add a new feature to the functionality of filter logs.

  This feature will allow apply a filter to a concrete logs (from the last day, the last month, etc) and send the logs that are the result of the filter appliance by mail to the user in the frequency he considers appropriate.

- Add a new feature to the functionality of monitor logs.

  The data that are considered deprecated by the user, because are too old for example, can be dumped in a file. In this way the database will not be so loaded and will work better.

  And vice versa, the data that are dumped in a file can be stored in the database again to use the application with these logs.

Also is important to take into account the integration with the rest of system administration tools of the LCFIB. This objective can be done easily implanting the authentication system CAS (Central Authentication Service). CAS is a Single Sign On[3] authentication protocol designed to allow untrusted web applications to authenticate users against a trusted central server. When the client visits a protected application, it will be automatically redirected by the application to CAS.

[3] Single Sign On (SSO) is a method of access control that enables a user to authenticate once and access to all the applications he has been authorized to access.

7.5. Personal evaluation

After several months spending a lot of time working on this project, it is time to think over the things that has brought it to me and to extract the personal conclusions.

Regarding to the technological evaluation, it is remarkable the learning of the basic web technologies during the development of the application. This kind of technologies is a field in which I have not the opportunity of know in depth along the master's degree. Moreover I extend my knowledge about the firewalls and network monitoring. Finally I see that a tool for monitoring the firewall logs is really important to manage correctly all the information but it is also important to know what information add in the log file in order to don't introduce data that produce noise in the monitoring.

My personal evaluation about this master final project is very positive, I consider that it has been a profitable experience. The fact of carry out a project from beginning to end by itself is a valuable fact, the support of the LCFIB colleagues in any arisen problem is always taken into account.

7.6. Project license

In the regulations of the Master Final Project in the FIB are stipulated that the intellectual property of the projects carried out in companies is determined by the agreement established with the company.

The LCFIB and the student are agree with the idea that this project would be distributed under the terms of the GNU General Public License (GPL) as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. The decision was taken because in the application there are libraries from third parties, these libraries are distributed under the terms of GPL, so if they want to be included in the code of the application it can not impose any further restrictions on the exercise of the rights granted or affirmed under the GPL license. So the best way is to distribute the application under the GPL license too.

Next is a brief summary of the conditions of the GPL license:

GPL is a free software license so it allows the recipients of a computer program the rights of use, study, modify and distribute it. The characteristics of GPL are the two added terms:

- All the software derived should be distributed under the GPL license, this is called Copyleft. Copyleft describes the practice of using copyright law to remove restrictions on distributing copies and modified versions of a work for others and requiring that the same freedoms be preserved in modified versions.

- It can not impose any further restrictions on the exercise of the rights granted or affirmed under the GPL license.

In the file `gpl.txt` of the application there are all the terms of the GNU GPL license.

Chapter

8

BIBLIOGRAPHY

8.1. Books

- 🕮 Dolors Costal; Xavier Franch; M.Ribera Sancho; Ernest Teniente *Enginyeria del software. Especificació.* 3rd edition. Barcelona: Edicions UPC, 2005. ISBN 84-8301-799-7

- 🕮 Enginyeria del Software 2 notes.

8.2. Web pages

- **Netfilter: IPTables**

  http://www.netfilter.org/documentation/index.html#documentation-howto

  http://logi.cc/linux/netfilter-log-format.php3

  http://iptables-tutorial.frozentux.net

  http://www.madeinshell.net/IPTables%20-%20Manual%20Practico.pdf

- **IPFilter**

  http://coombs.anu.edu.au/~avalon/

  http://www.obfuscation.org/ipf/ipf-howto.pdf

  http://www.sawmill.net/formats/ipmon.html

  http://blogs.sun.com/mgil/category/Seguridad

- **Other products**

  http://www.gege.org/iptables/

  http://webfwlog.sourceforge.net/

  http://tud.at/programm/fwanalog/

  http://www.checkpoint.com/products/firewall-vpn.html

  http://www.sun.com/software/securenet

  http://www.intellitactics.com

- **Web servers**

  http://www.apache.org

  http://tomcat.apache.org

  http://www.microsoft.com/spain/technet/productos/iis/default.mspx

- **Server languages**

  http://www.php.net

  http://www.desarrolloweb.com

  http://www.webtaller.com

  http://www.perl.org

  http://www.gratisweb.com/disidents/ascii/codex/perl_a_dolor_01.html

  http://www.sunsite.ualberta.ca/Documentation/Misc/perl-5.6.1/pod/perlfunc.html

  http://java.sun.com

  http://www.asp.net

- **Database management system**

  http://www.mysql.com

  http://www.postgresql.org

  http://www.oracle.com/global/es/index.html

- **Web page design**

  http://www.w3schools.com/tags

  http://www.w3schools.com/css/css_reference.asp

  http://javascript-reference.info

- **Icons**

  http://www.famfamfam.com

- **Regular expressions**

  http://www.troubleshooters.com/codecorn/littperl/perlreg.htm

  http://www.cs.tut.fi/~jkorpela/perl/regexp.html

- **Statistics**

  http://www.fisterra.com/mbe/investiga/10descriptiva/10descriptiva.asp

- **Firewall forensics**

  http://www.linuxsecurity.com/resource_files/firewalls/firewall-seen.html

  http://www.cs.ucsd.edu/~clbailey/PortScans.pdf

  http://www.rediris.es/cert/doc/unixsec/unixsec.pdf

- **PHP Libraries**

  http://naku.dohcrew.com/libchart/pages/introduction

  http://www.fpdf.org

- **Perl modules**

  http://www.cpan.org

- **Ulogd**

  http://www.netfilter.org/projects/ulogd/index.html

- **Language resources**

  http://www.wordreference.com

  http://www.upc.edu/slt/recursos/english_resources

  http://thesaurus.reference.com

Licenses

http://www.gnu.org/licenses/gpl.txt

http://www.fsf.org/licensing/licenses/gpl-howto.html

Appendix

A

USER MANUAL

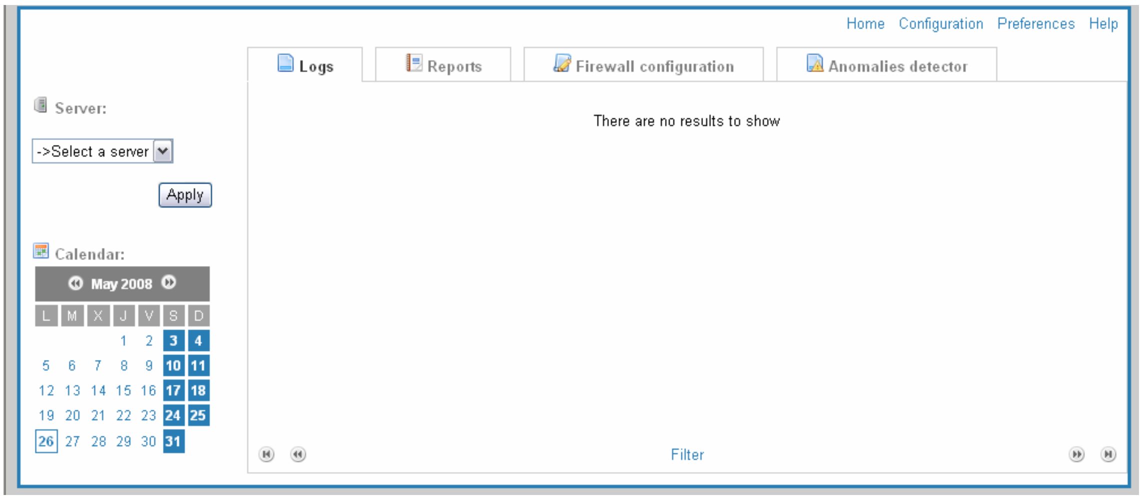
This manual is intended to the final user of the application. It is recommended that the user read this manual before the use of the application for first time to have an overview of all the functionalities and then he can use it appropriately.

The entire user manual is explained in the next sections of the chapter. They are classified according to the functionalities of the application. It is done this way to provide an easy reference guide to the users. As was quoted in the previous chapters the functionalities are:

- Firewall logs monitoring
- Log filtered
- Reports
- Firewall configuration
- Anomalies detector
- System configuration

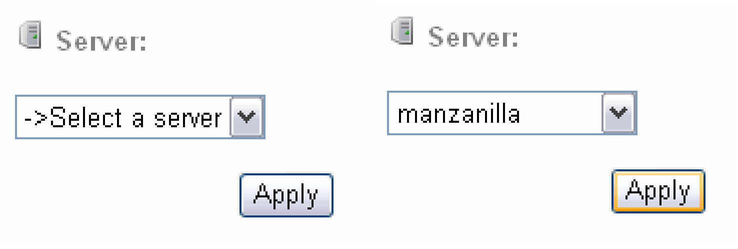
A.1. Firewall logs monitoring

The home page of the application is the firewall logs monitoring view that is the main functionality. If the user connects for first time he sees the next page:

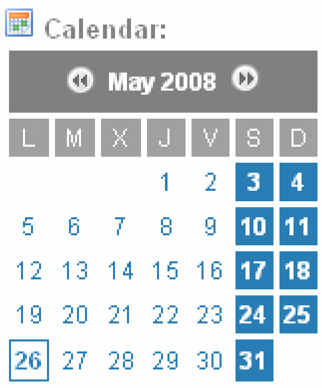


There are no results to show

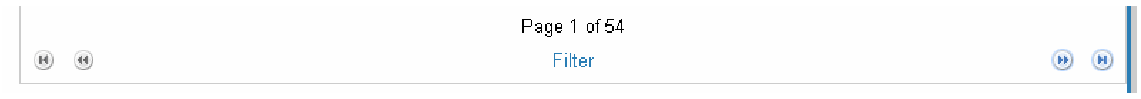
Supposing that the user has configured at least one server (see below in the section *System configuration - Configuration*), to show the data he first should select the server from the Server pull-down menu and click the Apply button.



He also can select a concrete date in the Calendar. In it the days with any log are underlined to know in which of them are available data. The user can go to any month with the left and right arrows, and select a concrete day clicking in it.

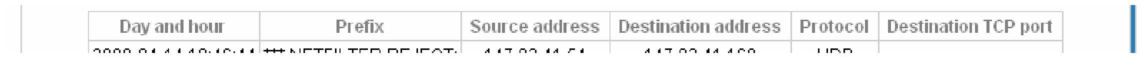


Once the logs are showed in the page, they are paginated (the number of logs per page can be configured, see the section *System configuration - Preferences*) and the user can surf by the pages with the correspondent arrows which allow going to the first, previous, next or last page.


Page 1 of 54

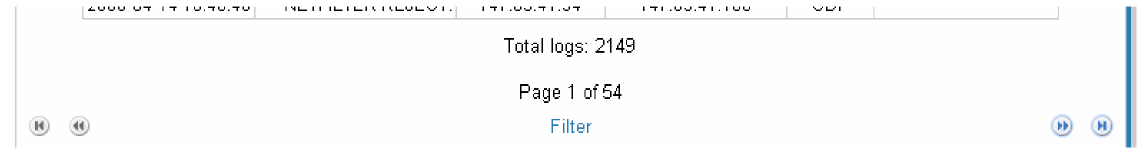
Filter

The logs also can be ordered by any of the columns showed by clicking in the title of the column. In the first click they are ordered in ascendant order, then in descendant and so successively.



Day and hour	Prefix	Source address	Destination address	Protocol	Destination TCP port

In the page there is information about the total number of logs that are selected to view currently and the actual page respect the total number of pages.

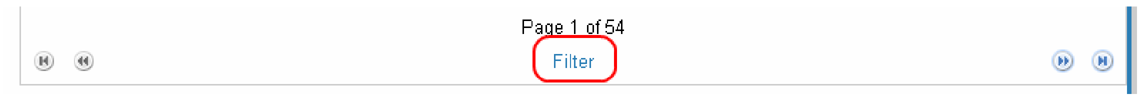

Total logs: 2149

Page 1 of 54

Filter

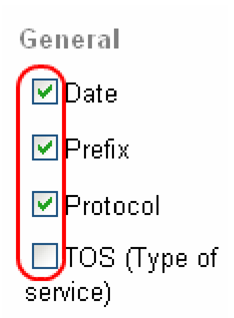
A.2. Log filtered

The user can access to the log filtered functionality from the main page.

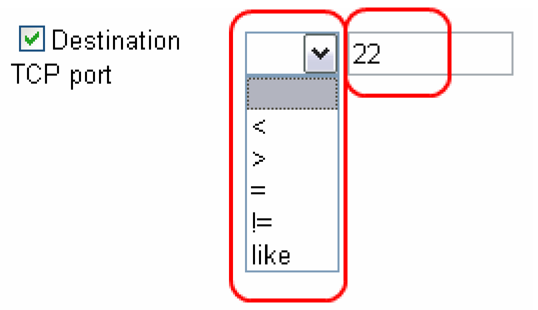


There are different concepts that are part of a filter:

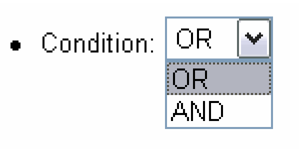
- Fields to show. The fields are the columns showed of every log. The user can select all the fields he wants, in the basic filter mode are only showed the most important and in the advanced one are showed all the fields. To select a field only the checkbox must be marked.



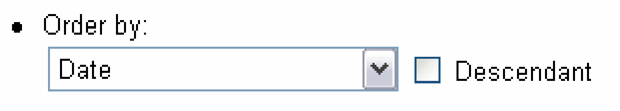
- Restriction of fields. The user can filter with restrictions of every field. This characteristic is independent of the fields showed. Depending on the type of data of the field the restriction can be: leaser than, greater than, equal to, different to, like. Then the user introduces a valid value to be compared.



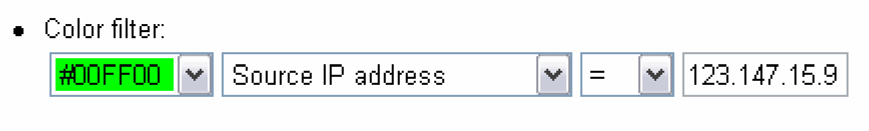
- Condition of restrictions. The restrictions can be joined with the AND operator or with the OR operator. All the restrictions must be carried out with the AND operator and one of the restrictions must be carried out with the OR operator.



- Order by. This is the order which have the showed logs, this option is the same that if the user ordered the logs clicking in the title of the field.



- Colour filter. The colour filter helps the user to emphasize the logs which are agreed with the restriction of the field selected to this purpose.

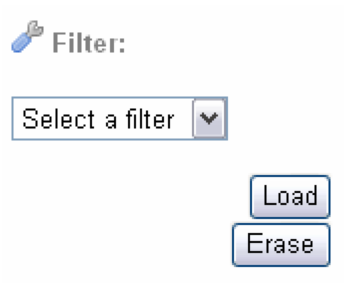


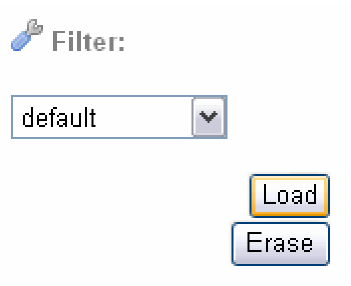
If the user wants to view all the options to select in a filter he should select the advanced filter, to do it he should click the Advanced filter link. If he wants to back to the basic view he should click the Basic filter link.

Once the user has selected the filter he wants to apply, he should click the Apply button to make effective the filter in the showed logs. To go back without the appliance of any filter the user just should click the Return link.

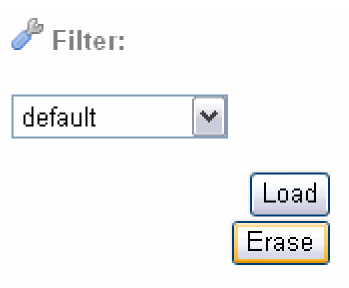
The filters the user creates can be saved for use them later. When a filter is defined for first time, the user should click the Save as… button to save it. Then the system will ask for the name of the filter, this name will be the identification of the filter.

The stored filters are available to load it. In the page there is a Filter pull-down menu with all the stored filters. The user should select the filter he wants to charge and click the Load button.





Under the Load button it is the Erase button, it erases the actual loaded filter.

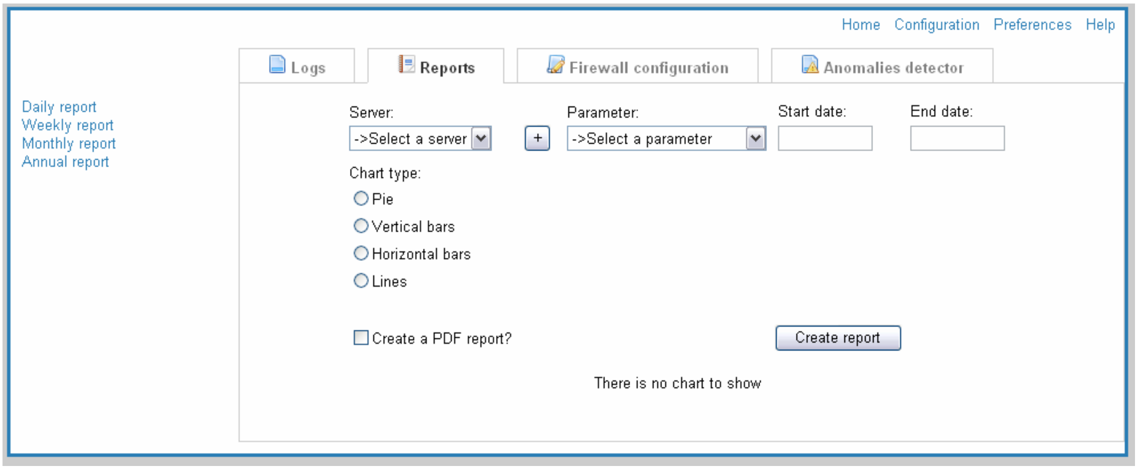


A filter can be modified, the user selects the new options he wants to apply in the filter and then save the changes with the Save button.

If the user change the filter options and then he wants to restore the previous values he can do it with the Restore button. It undoes the last changes he did.

A.3. Reports

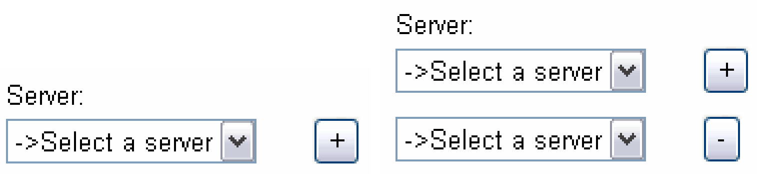
From the main page the user can go to the reports page by clicking on the Reports tab. There are different ways to make the reports. The standard way is from the principal page of the Reports functionality.



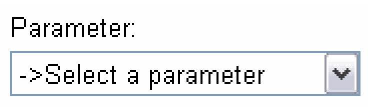
There are other modes: daily report, weekly report, monthly report, annual report. All of them are the same but with facility to the user of choose the range of dates in which are included the requested report. For example in the monthly report case there is a pull-down menu with all the months in which are available data from the configured servers, and the same occurs with the weekly report and the annual report.

The different options to configure in the reports are:

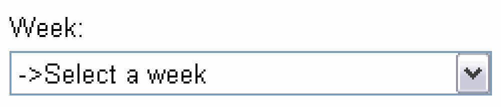
- Server. The server that will be the source of the data will be chosen from a Server pull-down menu. This option can be more than one value, to add server values just click the '+' button. When there is more than one Server pull-down menu appears the '-' button that is to subtract servers.



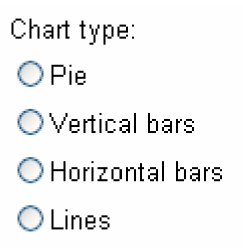
- Parameter. The parameter of the selected servers that the user wants to monitor in the report. With the Parameter pull-down server can be selected any of the parameters stored for logs.



- Date. The range of time that the user wants to observe in the report. In the basic and in the daily mode he introduces the date manually, but in the weekly, monthly and annual mode he chooses the available dates from a pull-down menu.



- Chart type. There are different char types to choose: pie, vertical bars, horizontal bars, lines. The user will choose the type which is more convenient to the report he wants to generate.

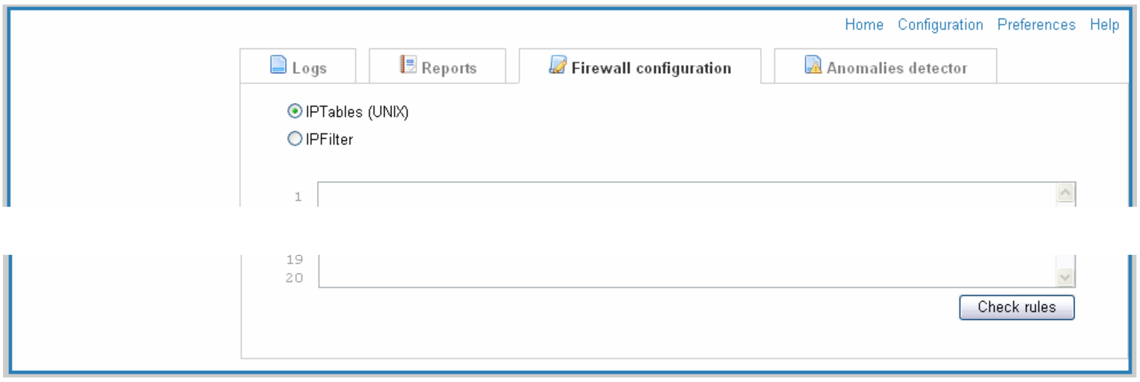


- PDF. The user can decide if he wants the report in PDF format, else the report will be created at the same web page. If the report is created in PDF format then he can save it for a later use.

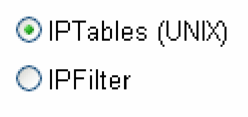
Once the user has selected the options of the report, he should click the Create report button to generate the report.

A.4. Firewall configuration

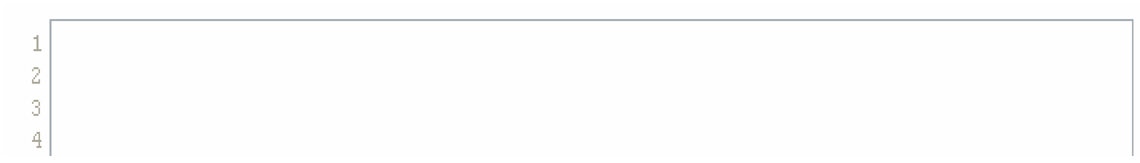
From the main page the user can go to the firewall configuration page by clicking on the Firewall configuration tab.



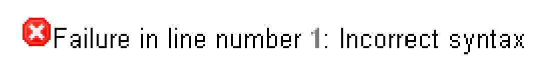
This functionality provides a firewall rules checker. The introduced rules can be in IPTables or IPFilter format. The user can select it before he introduces the rules.

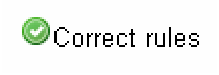


The user introduces the rules in the text area and then clicks the Check rules button.



The checker will verify the correct syntax of the rules and when it finishes it will show a message with the result of the process.


Failure in line number 1: Incorrect syntax

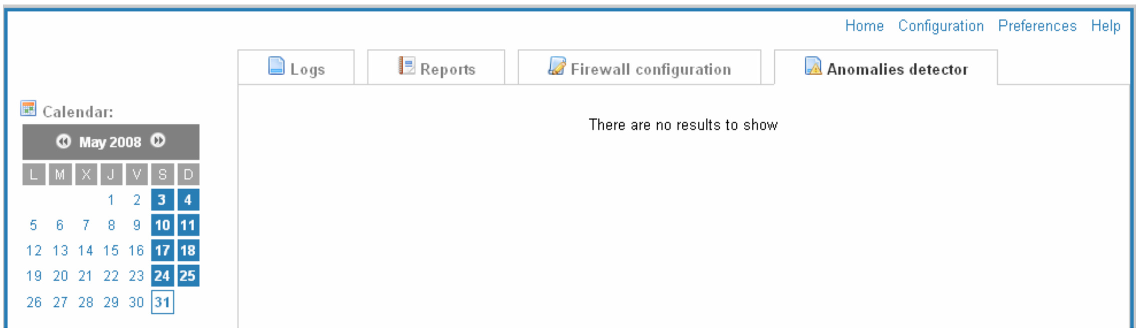

Correct rules

The format of the rules can be consulted in the respective manuals:

- IPTables: http://www.netfilter.org/documentation/index.html
- IPFilter: http://coombs.anu.edu.au/~avalon

A.5. Anomalies detector

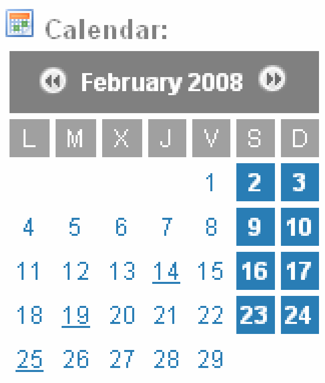
From the main page the user can go to the anomalies detector page by clicking on the Anomalies detector tab.



There are no results to show

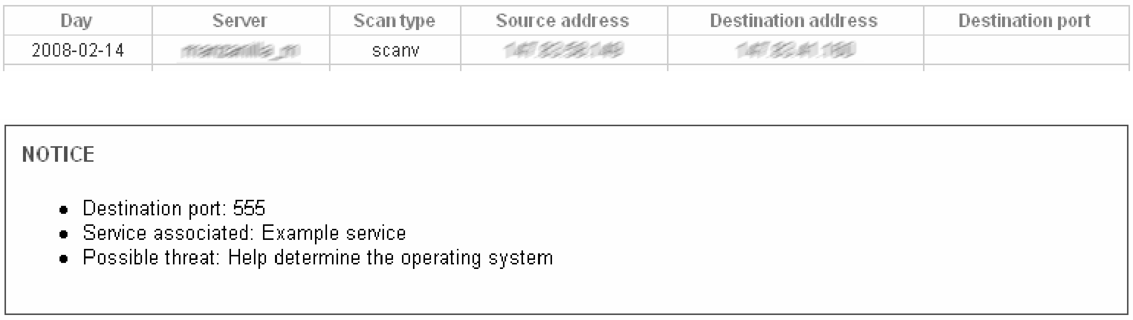
The user should add a scheduled task on the system where is installed the application. This scheduled task will run a script that every day will analyze the stored logs. The script `anomalias.pl` is located in the `scripts` directory.

The user can select a concrete date in the Calendar. In it the days with any anomaly detected are underlined to know in which of them are available data. The user can go to any month with the left and right arrows, and select a concrete day clicking in it.



L	M	X	J	V	S	D
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29		

The user can click in the line of a concrete anomaly in order to see with more detail the logs which are related to it. If the anomaly is a known attack in the screen there will be an alert with the possible attack, this is only an advice and it couldn't be the cause of the attack.



Day	Server	Scan type	Source address	Destination address	Destination port
2008-02-14	[redacted]	scanv	[redacted]	[redacted]	

NOTICE

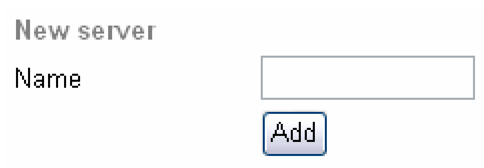
- Destination port: 555
- Service associated: Example service
- Possible threat: Help determine the operating system

A.6. System configuration

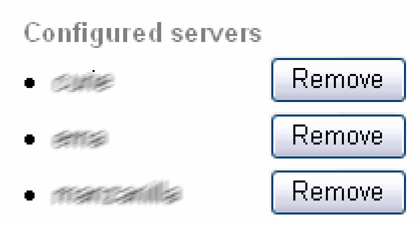
Configuration

The configuration of the system can be changed by the user. In the fix menu of the top right there is the Configuration link that shows the page.

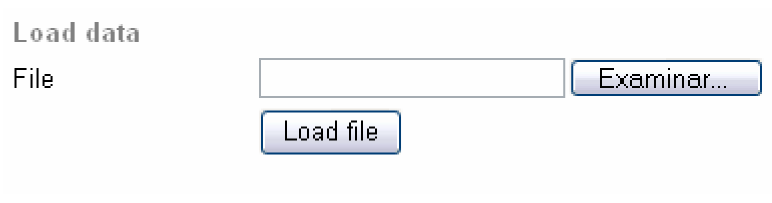
The user can introduce any new server to be monitored. He introduces the server name in the text box and then clicks the Add button. But it is not all, in the server he must run the script `mi_ulogd.pl` (for IPTables) or `mi_ulogd_solaris.pl` (for IPFilter) located in the folder `scripts`. There will be a copy of the script running for every different file in which there are logs to be monitored. For example if in the same log file there are logs from several servers and the server to add is in a file that now is being monitored, just adding the server in the web application should be sufficient.



A list of the configured servers is showed. Any server can be deleted from the configuration, it implies that this server won't be monitored and all the stored logs will be deleted. The user just should click the Remove button correspondent to the server he doesn't want to monitor. If there was a specific script running only for this server it's also advisable that the user stop it.



If the user has logs from previous date he can load the data. He should select the file with the browser Examine button and when the file which contains the logs will be selected he clicks the Load file button.



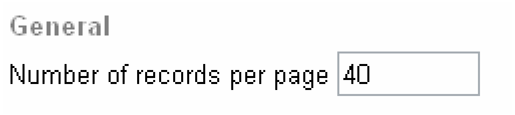
Preferences

The user can change some values of the application preferences in the Preferences link located at the fix menu of the top right.

The preferences are classified in general preferences and database preferences.

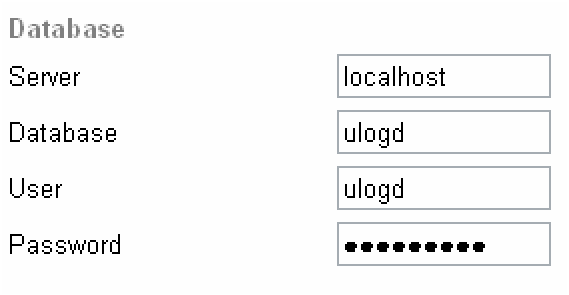
General preferences	Number of records per page
Database preferences	Server Database User Password

When all the preferences has been introduced, the user clicks the Save changes button. If there is any problem with the database preferences the system will advice to the user of it and don't save these values.



General

Number of records per page	40



Database

Server	localhost
Database	ulogd
User	ulogd
Password	●●●●●●●●●●

In both cases, configuration and preferences, if in any moment the user wants to go back he should click the Return link.

A.7. Miscellaneous

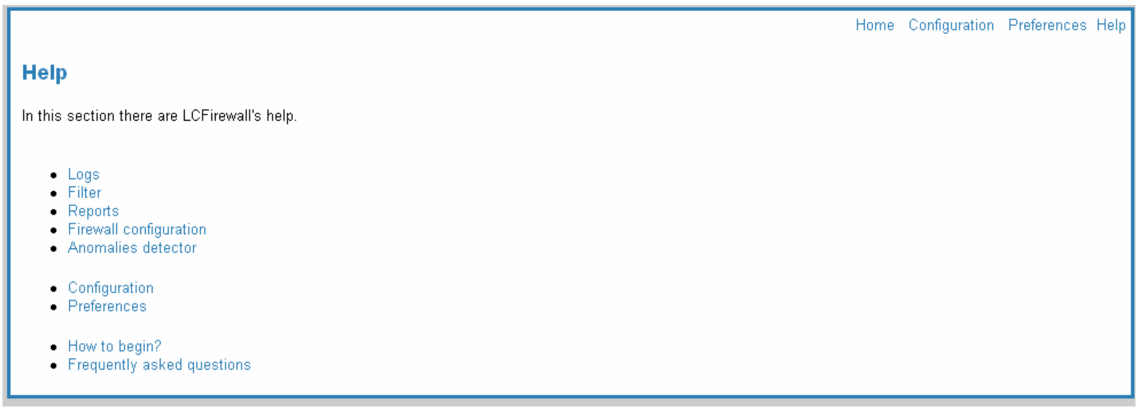
Internationalization

The language can be changed by clicking of the correspondent language at the top right corner of any page.

The available languages are English, Spanish and Catalan. An advanced user can add more languages like was explained at the section *Internationalization* in chapter *Implementation*.

Help

There is a help section for every page of the application. In the fix menu of the top right there is the Help link.



Help

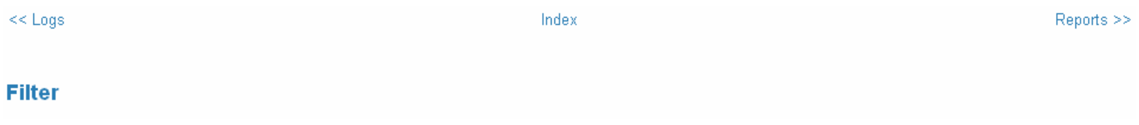
In this section there are LCFirewall's help.

- Logs
- Filter
- Reports
- Firewall configuration
- Anomalies detector

- Configuration
- Preferences

- How to begin?
- Frequently asked questions

When the user is in a concrete section of the application and goes to the help, the page correspondent to this section will be showed. He can go to the index and go to another section of the help. He also can go to the previous or the next section of the actual one.



Appendix

B

INSTALLATION MANUAL

The installation manual explains the necessary steps to have the application ready for use it.

The application was carried out with the purpose that the installation was the simpler possible for the user. With the next instructions the user can easily and quickly get the application prepared.

B.1. Requirements

The application needs some additional software to work correctly, the installation requirements are the next:

⌘ Apache Server or any other web server.

The tests have been done with Apache server, but the application can run with any other web server. The reasons for what the Apache Server has been chosen were exposed in the *Election of technologies* section, on *Implementation* chapter.

⌘ MySQL5

The database management system is MySQL, in concrete since MySQL5 any version can be installed because are used functionalities that are included since this version.

⌘ PHP5 compiled with support to GD2, FreeType2, libpng and MySQL.

The application is developed in PHP, so the PHP is needed. Specifically the PHP5 version is needed because there are used functions that are included since this version.

The support to MySQL is because is the database management system used by the application. Finally, the support to GD2, FreeType2 and libpng is needed because the libraries used to make the graphs need these libraries.

⌘ Perl with modules File::Tail, Date::Parse and DBI.

The scripts layer is implemented in Perl, so an installation of Perl is the last requisite. The installed version to carry out the tests was the Perl 5.8.8, so it's preferable install this version or greater.

The additional modules can be downloaded from CPAN like was explained in the *Implementation details* section, on *Implementation* chapter.

B.2. Installation

The application is available in the tar file `LCFirewall.tar`. This file contains all the files and folder structures that compose the code application. To install it just is necessary to follow the next steps:

1) Extract the file and put the extracted folder in the desired place of the system.

2) Configure the Apache (or the installed web server) with the location of the application. For example if the user extracts the application in /home/soft then his Apache configuration could be:

```
Alias /LCFirewall /home/soft/LCFirewall/
<Directory /home/soft/LCFirewall>
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

3) Create the database and the user in MySQL. Then with the file `inicio.mysql` located on the `scripts` folder, the necessary tables will be created.

```
mysql -u <user> -p <bbdd> < inicio.mysql
```

4) Change the preferences in the web application in the section *Preferences*.

5) For each server the user wants to monitor, he should do:

5.1) Add the server in the web application in the section Configuration (see the *Configuration* section on the chapter *User manual*).

5.2) If the user wants to use the `ulogd` (IPTables feature) he should add the correspondent lines in the firewall configuration, else if he wants to use the scripts `mi_ulogd.pl` (for IPTables firewall logs) or `mi_ulogd_solaris.pl` (for IPFilter firewall logs) he should check the configuration in the first lines and then run it.

6) Check the configuration of the script `anomalias.pl` in the first lines. Then run it daily, for example in the Unix servers should be added in the `cron`.

Now the application is ready to begin use it.