POLITECNICO DI TORINO

III Facoltà di Ingegneria
Corso di Laurea Specialistica in Ingegneria delle Telecomunicazioni

Master of Science Thesis

Split Management of TR069 Enabled CPE Devices

Author: Juan Pablo Martinez Rojas

Supervisor: Prof. Marco Mellia
Dipl. Ing. Max Riegel
Dipl. Ing. Klaus Wich

June, 2011
Summary

Everyday the world witnesses the overwhelming telecommunications’ revolution that has been unleashed over the past few decades. The new ways of communications have changed and touched most of the inhabitants of our planet, reaching even the most remote places, transforming the way how people communicate, interact, work, study and more. All these aspects of life have been affected for a new generation of amazing devices and technologies that give the chance to people to stay connected any time and anywhere. Nonetheless, due to the incredibly growth of the mobile industry and its evolution, an evolution that indeed is happening at rates that even 5 years ago would have been really difficult to foreseen, is leading the mobile networks into a stage of data overload. Fuelled by the new smart-phone devices, USB dongles, tablet PCs and all the new generation of mobile devices, it is estimated that the mobile networks have increased their data load in 159% from 2009 to 2010[10], and the market is expected to grow even more in the years to come. The exponential growth of this huge wave of data bytes going through the networks is also result of the big competition between mobile companies with all their flat rates of unlimited data consume in order to attract more customers. This had become a major concern for the service providers all over the world, overflown of data is already a reality and adding more capacity to their mobile networks has shown to be neither sufficient nor effective due to its high cost.

As a result of this critical trend for the mobile service providers, the need to look for innovative solutions to maximize the available bandwidth for their customers as the cost per bit is lowered has become essential for their survival in the market. Therefore, to look for alternatives to offload the traffic on their networks to second networks that liberate them of this wave of data can result into a very convenient an affordable solution. Wi-Fi, due to its characteristics is a good alternative as an offload mechanism, it grants to the companies thousands of access points already distributed through all the cities, with low cost and a free spectrum that provide internet access to every user within the access point’s radio range, saving the Mobile Operators a lot of traffic going through their mobile networks, and giving the customers a better internet experience in a less congested network with a better bandwidth, in most cases.

For granting the access to those customers that want to use Wi-Fi in their mobile devices, a split of the management of the device needs to be done, splitting the responsibilities between the customer who uses the CPE at home, and the Network Operator who wants to ensure that the off-load can be possible using the same
device. In this thesis we will investigate and create a model that enables a split of the management of a CPE device, management that will be done remotely by the network operator using TR-069 and by the Private User managing the device locally for his personal interests. Additionally, important aspects like security and privacy will be analysed, giving a detailed explanation of future risks and how to prevent them and what privacy policies should be enforced to create a successful division in the administration rights. This new model will be implemented and tested using two different CPEs using a customized firmware based in OpenWrt.
Acknowledgement

First of all, I would like to express my appreciation and gratitude to my friends and colleagues in Nokia Siemens Networks. Their encouragement and support was essential for me and the development of this master thesis.

I would like to give especial thanks to Max Riegel and Klaus Wich, who gave me the opportunity to be at Nokia Siemens Networks working on the project and since the beginning have supported me with their patience, ideas and guidance throughout all the process. Thanks for taking your time to review my work and the often discussions and meetings, they helped me out on doing a better job.

Last but not least, I would like to thank my family, my girlfriend and my friends who have been always there for me, giving me their support in the good times and the strength to overcome the hard times, without them this would not have been possible.
## Contents

1 Introduction .......................................................... 1  
  1.1 Network Architecture for Cellular Offload ................. 4  

2 LAN and WLAN Architecture ..................................... 7  

3 TR-069 Overview ..................................................... 10  
  3.1 CWMP Features .................................................. 10  
  3.2 CWMP Protocol Stack and Operation ......................... 12  
    3.2.1 Operation .................................................. 13  
  3.3 Other Technical Reports ....................................... 13  

4 Security Background ............................................... 15  
  4.1 Wireless Network Security ................................... 15  
    4.1.1 Wired Equivalent Privacy (Wired Equivalent Privacy) . 15  
    4.1.2 Wi-Fi Protected Access (WPA) .......................... 15  
    4.1.3 IEEE-802.11i (WPA2) .................................... 16  
    4.1.4 Wireless Network Security Threats ....................... 17  
    4.1.5 Security Threats In the Web Interface ................. 20  
  4.2 CWMP Security .................................................. 22  
    4.2.1 Security Mechanisms .................................... 22  
    4.2.2 TR-069 Security Risks .................................. 23  
  4.3 Botnets ........................................................ 24  
    4.3.1 Botnets in embedded devices ......................... 25  

5 Privacy ............................................................. 26  
  5.1 Private User’s Privacy ....................................... 27  
    5.1.1 When will the client access the CPE configuration interface? 28  
    5.1.2 Using additional functions ............................. 29  
  5.2 Privacy on the Public Network ............................... 29  

6 OpenWrt Overview .................................................. 31  
  6.1 OpenWrt Features ............................................. 32  
  6.2 UCI - Unified Configuration Interface ....................... 32  
  6.3 LUCI - Lua Configuration Interface ......................... 33  

7 The CPE Devices ..................................................... 34
CONTENTS

8 Implementation of the Management Split 37
  8.1 The Web Interface ........................................ 38
    8.1.1 The Authentication and Accounting .................. 39
    8.1.2 The Discarded Options ............................... 45
    8.1.3 The Added Options .................................... 46
    8.1.4 The Modified Options .................................. 52
    8.1.5 The Design ............................................. 61
    8.1.6 Operation On The OpenWrt enabled CPEs .............. 62
  8.2 Groups and Access Rights ................................. 63
  8.3 Packages in the OpenWrt Firmware ........................ 63

9 Security Threats 67
  9.1 Attacks Coming From The Home Network .................. 67
    9.1.1 Attacks to the Web Interface ......................... 68
  9.2 Attacks Coming from the Broadband network ............. 69
  9.3 Attacks Coming from the Public Network ................ 69

10 Recommendations 70
  10.1 WEP, WPA and WPA2 ..................................... 70
  10.2 802.11w .................................................. 70
    10.2.1 802.11w in OpenWrt ................................. 71
  10.3 HTTPS Over LUCI ......................................... 71
  10.4 Secure Shell, How to Prevent Attacks .................. 72
  10.5 Intrusion Detection System and Intrusion Prevention System 74
    10.5.1 Snort ............................................... 74
  10.6 Using CWMP for security purposes ....................... 74

11 Conclusion 76

Appendix:

A Tables of The Privacy Policy 82
B Modifications in Folder /model/cbi/admin/network/ 83
C Modifications in Folder /model/cbi/luci/fw/ 84
D Source Code of the New Web Interface 85
List of Figures

1.1 Expected growth of mobile services subscribers on the next years.[4] 1
1.2 Growth of mobile data during the first half of 2010. [10] . . . . . . . 2
1.3 Network Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . 4
2.1 LAN Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
3.1 CWMP Protocol Stack . . . . . . . . . . . . . . . . . . . . . . . . . 12
3.2 CWMP communication example . . . . . . . . . . . . . . . . . . . . 13
3.3 Overall specification structure for the TR-069 family of standards. [40] 14
5.1 Reading permits of the different sections of the CPE . . . . . . . . . 27
5.2 Writing permits of the different sections of the CPE . . . . . . . . . 28
7.1 Router, Linksys WRT54GL . . . . . . . . . . . . . . . . . . . . . . 35
7.2 Router, D-Link DIR-825 . . . . . . . . . . . . . . . . . . . . . . . . 35
8.1 Example of the redesigned web interface . . . . . . . . . . . . . . . 62
9.1 Threat Model for the CPE . . . . . . . . . . . . . . . . . . . . . . . 67
List of Tables

8.1 Files removed from the LUCI configuration . . . . . . . . . . . . . . 46
8.2 List of Packages Installed in OpenWrt . . . . . . . . . . . . . . . . . 65
8.3 Drivers Required for the LINKSYS WRT54GL . . . . . . . . . . . . . 66
10.1 Packages required to enable HTTPS in OpenWrt . . . . . . . . . . . 72
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACS</td>
<td>Auto Configuration Server</td>
</tr>
<tr>
<td>ADSL</td>
<td>ADSL Asymmetric Digital Subscriber Line</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>CCMP</td>
<td>Comprehensive Conservation and Management Plan</td>
</tr>
<tr>
<td>CPE</td>
<td>Customer Premises Equipment</td>
</tr>
<tr>
<td>CSRF</td>
<td>Cross-Site Request Forgery</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>CWMP</td>
<td>CPE WAN Management Protocol</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed Denial of Service</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service</td>
</tr>
<tr>
<td>DSL</td>
<td>Digital Subscriber Line</td>
</tr>
<tr>
<td>DSLAM</td>
<td>Digital Subscriber Line Access Multiplexer</td>
</tr>
<tr>
<td>EAP</td>
<td>Extensible Authentication Protocol</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GB</td>
<td>Giga Bytes</td>
</tr>
<tr>
<td>GHz</td>
<td>Giga Hertz</td>
</tr>
<tr>
<td>GPL</td>
<td>General Public License</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure</td>
</tr>
<tr>
<td>IDPS</td>
<td>Intrusion Detection and Prevention System</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion Detection System</td>
</tr>
<tr>
<td>IE</td>
<td>Internet Explorer</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPS</td>
<td>Intrusion Prevention System</td>
</tr>
<tr>
<td>IPV4</td>
<td>Internet Protocol Version 4</td>
</tr>
<tr>
<td>IPV6</td>
<td>Internet Protocol Version 6</td>
</tr>
<tr>
<td>IRC</td>
<td>Internet Relay Chat</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>kB</td>
<td>Kilo Bytes</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LED</td>
<td>Light Emitting Diode</td>
</tr>
<tr>
<td>LMO</td>
<td>Lua Machine Objects</td>
</tr>
<tr>
<td>LUCI</td>
<td>Lua Configuration Interface</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MAC</td>
<td>Mandatory Access Control</td>
</tr>
<tr>
<td>MD5</td>
<td>Message-Digest algorithm 5</td>
</tr>
<tr>
<td>MITM</td>
<td>Man-In-The-Middle attack</td>
</tr>
<tr>
<td>PPPoE</td>
<td>Point to Point Protocol over Ethernet</td>
</tr>
<tr>
<td>SCP</td>
<td>Secure Copy Protocol</td>
</tr>
<tr>
<td>session ID</td>
<td>Session Identifier</td>
</tr>
<tr>
<td>SFTP</td>
<td>SSH File Transfer Protocol</td>
</tr>
<tr>
<td>SIM</td>
<td>Subscriber Identity Module</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>SSID</td>
<td>Service Set Identifier</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>UCI</td>
<td>User Configuration Interface</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>VLAN</td>
<td>Virtual Local Area Network</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WEP</td>
<td>Wired Equivalent Privacy</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WPA</td>
<td>Wi-Fi Protected Access</td>
</tr>
<tr>
<td>WPA2</td>
<td>Wi-Fi Protected Access Version 2</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XSS</td>
<td>Cross-site scripting</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In 1858 after the success of the first transatlantic telegraph message the world started to dream in a planet interconnected thanks to the telecommunications, 150 years later after a lot of outstanding innovations and hard work, the mankind have achieved its purpose and in an exceptional way. Communications are a reality in the whole globe, internet, telephones, cellphones have made this a reality and it gets better and better, everyday new and better phones come out on the market, mobile networks that provide faster navigation on the internet, better quality calls, video calls, conference calls, etc... The spread of telecommunications have reached places where even the electricity is not yet arrived, and this trend is expected to continue. By 2014 the amount of mobile broadband subscribers using a smart-phone will grow by more than 4 times of the actual number of users worldwide, exceeding 1.6 billion subscribers around the world.

![Figure 1.1: Expected growth of mobile services subscribers on the next years.[4]](image)

In addition, the coming wave of data overflow will be enhanced by the usage the customers are giving to their devices, is not any more only about browsing the internet, is about using apps, video streaming, VoIP, instant messaging services and downloading contents and files. This is what really gives the added value to their smart-phones and mobile devices and this will contribute considerably the growth
CHAPTER 1. INTRODUCTION

on mobile bandwidth usage. A regular cell-phone generates on average 3.3 MB of data traffic monthly, smart-phones generated 24 times the same amount on 2010, and laptops connected using USB-dongles even more than 500 times, approximately 1.7 GB per month on average\[10\]. The stream is leading that the main type of data transferred in a mobile network will not be any more voice data, e.g. as shown in figure 2, in the first half of 2010 video streaming presented a growth of more than 90\%. If we consider that smart-phones only represent 13\%[10] of the total of handsets in use today, in 5 years the amount of mobile data only of video streaming could exceed 2010’s consumed mobile traffic by far. The service operators are already witnessing these big increases of data in their mobile networks but the real data avalanche is yet to come.

Figure 1.2: Growth of mobile data during the first half of 2010.[10]

This ciphers only tell the inevitable and therefore the question now is what should be done?, solving a problem have many solutions, optimization of data compression and transfer, new mobile networks with more transfer capacity, data offload, etc..., there is not a unique solution but they all differ in cost, velocity of implementation and effectiveness.

The so called data offload is a technique that involves using alternative wireless networks in order to transfer data from and to a mobile device instead of using the operator’s mobile network, in this case using DSL internet connection through Wi-Fi. Data offload presents itself as a very efficient way to ease off the traffic of the mobile networks, since most of the new handsets are capable of using Wi-Fi, a normal user can connect its device to its private or office internet access points, this give to the customer access to a larger band and much less congested network in most of cases. Data offload is already being used by many customers, and while the global mobile data traffic was incremented by 159\% in 2010[10], this number could have been larger if it was not for data offload. It is estimated that 31\% of smart-phone’s traffic was offloaded using Wi-Fi or femtocells[10]. Nonetheless of its moderate success, this cipher could be a lot larger. Data offload has not reached its
full capacity as alternative mean of data transfer, in most of large and medium cities there are thousands of Wi-Fi access points already widely distributed, so no extra hardware is needed, that could be used by the customers. Therefore, reusing common Wi-Fi routers by enabling a secure access through separate SSIDs and virtual networks in the same hardware, having transparent migration into different spots and keeping privacy and security of the owners of each access points would grant the service for the home user and the public user and thus a major data offload of the mobile devices.

Hence, to be able to grant this features, it is essential that the network operator has a way to manage each one of these devices, a way to remotely access a Device sitting far away from the central server and interact with it. The CPE Wan Management Protocol, also known as CWMP by its initials, is a protocol designed as a mechanism of communication between a CPE (Customer Premise Equipment) and an auto-configuration server in a secure way, encompassing also other CPE management features. Defined in the TR-069 suite of documents by the Broadband Forum, CWMP is a protocol that gives to the service provider the ability of performing installations, updating, retrieving relevant data and configuring thousands of CPE’s distributed through the whole city.

On the other hand, the CPE will still be a device sitting at the customer’s home, with the special purpose of granting access to internet to its wireless and wired handsets and devices. So, if the CPE is managed only by the network operator the customer would loose the possibility to configure basic parameters that it expects to administrate, even if it does not care about some specific settings of the device, e.g. the up-link configuration or some DHCP settings, the customer needs a mean to change some basic configuration features for its private use, e.g. when a user wants to create a wireless network to connect his computer and his wife’s, then, he needs to choose a name for the network, a password, the type of encryption, etc...

For this reason, the management of the device needs to be split, one part managed by the network operator, using TR-069, and who will have most of the control on the device, taking special care on keeping up the internet connection, management of the Up-Link, and management of the wireless network created for the access of the users who want to offload their mobile handset’s traffic, and the second part will be managed by the customer, being the main administrator of its own private network, and giving internet access to the wired and wireless devices connected to it.

However, not every CPE out in the market has a CWMP client installed within its firmware, moreover, using the configuration provided by the vendors does not provide much configuration flexibility nor the possibility to add packages for special purposes. Not having flexibility on the configuration of the firmware would make the split of management practically impossible, and if many different brands would like to be supported, a different CWMP client for each one would be needed. Instead, to change the original vendor’s firmware into a more flexible one that would give the desired freedom to be customized and to add the characteristic’s needed. OpenWrt is a Linux based firmware distribution, specially designed for embedded devices,
that allows full customization of the CPE, this means that special security and other features configurations could be done, as well as the installation of additional packages to the CPE and the split of rights for the shared administration of the device. Then, the effort would be focused in designing the desired packages and pre-configuring important features in order to compile the OpenWrt firmware for the selected routers.

Additionally, OpenWrt offers the possibility to configure the router through a web interface. Using it, the customer have a simple and easy way to administrate its own private network. This web-interface is itself configurable, so it can be designed to satisfy the home customer’s needs and, at the same time, keeping the device safe of miss configurations, accessing important information stored in the router or other security threads for the device itself or any other connected to it.

1.1 Network Architecture for Cellular Offload

In the next figure, it is shown the expected configuration of the broadband access network, in which, the Network Operator uses the ACS to access the TR-069 enabled CPE going through a BRAS and a DSLAM.

- CPE: Customer-premises equipment or Customer-provided equipment, refers to any customer owned or purchased router or residential gateway[6] and in this case, it must be compliant to TR-069 and TR-098, where is specified on detail the characteristics required by the CPE. In addition, the CPE should include the DSL modem, routing and traffic handling features and to be able to handle multiple SSIDs. It is administered by both the customer and the network operator, however the up-link and most of the configuration and update is handled by the network operator remotely using CWMP.
CHAPTER 1. INTRODUCTION

• DSLAM: Digital Subscriber Line Access Multiplexer, is the device in charge of controlling and routing the DSL traffic between the CPEs and the internet service provider (ISP) network[18]. Working on layer 2, this unit permits to split and combine traffic from different end-user connection points into a single high-capacity uplink (Uplink or Gigabit Ethernet) to the internet access and the service provider network, enabling on this way an efficient deployment of broadband networks for high speed internet access. The equipment is usually located in a range of 1 - 5 km from the customer, since higher distances would not guarantee a high speed internet connection. Most of DSLAM used today support additional features like[41]:

  – Multiple DSL Transmission Types (e.g. ADSL, SDSL, IDSL)
  – Traffic Management
  – Quality of Service (QoS)
  – Security Functionalities

• BRAS: Broadband Remote Access Server, is a key component in the network since is the responsible for routing the traffic between the DSLAMs and the ISP’s network. In addition the BRAS manages subscribers functions like authentication, authorization and accounting (AAA), using Radius, DHCP or others; handles also session termination usually done by PPPoE; and IP address assignment. However, this devices have evolved in both functionality and popularity in the last years and now include more advanced features, some of the most common ones are:

  – Session termination – ATM PVC, PPP
  – AAA using PAP/CHAP, RADIUS, DHCP Option 82
  – Comprehensive IP routing – BGP, OSPF, RIP
  – IP address management – DHCP server, relay, proxy services
  – Integrated layer 2 switching – ATM, Ethernet, MPLS
  – Policy management and dynamic per session QoS
  – Firewall and security capabilities
  – IP multicast routing – PIM, MBGP, IGM [41]

• ACS: Auto-Configuration Server[6], is the device which is in charge for remote management of upgrades, configuration data files, optional features and subscriptions on each of the TR-069 capable CPE. The ACS usually is connected to a ISP’s subscribers data base from where it retrieves information about configuration data or special services directed to a specific customer, which is then used to install or upgrade a device. The scalability of its nature let the ACS to centrally manage a large number of CPEs in a reliable and secure mode.
Using this architecture and the equipment and features described before, it would be possible to enable the different SSIDs on the CPE, separate the traffic and run AAA from the BRAS for the users that log in the network for mobile access. In this way if each customer using the Wi-Fi for mobile service network is authenticated and authorized, the traffic from this specific customer could be forwarded to another service provider, in this way, the network could be rented for their interaction and service support. Consequently, for those cellular service operators who do not own a DSL[18] infrastructure it would be possible to use a external DSL network for the off-loading of the mobile traffic when the service is available in the area. This is important because interoperability between companies would be possible and, not only for mobile service but other services coming from third parties could be added, like IP-TV, in which case TR-069 could also have a major management importance.
Chapter 2

LAN and WLAN Architecture

After introducing the architecture of a network used for offload of cellular traffic on the section 1.1, in this chapter it will be made a deeper approach on the local networks that will be needed to make the offload of the traffic possible, as well as keeping the benefits and services granted to the customer.

First of all, it is important to define who will have access to the service using a certain CPE and what role will they play in the network. The customers can be categorized in two different classes, Private User and Public User:

- **Private User:**
  Is the person who acquired the CPE and has an agreement with the ISP for getting DSL connectivity through it at his place, therefore is the person who maintains the device at his home. He is the main user of the device and is willing to connect all his equipment using the router in a private and secure way. Because of this he will need to play a role in the management of the CPE.

- **Public User:**
  Is any other customer who has made an agreement with the ISP to get connected to internet in his mobile device using the Wi-Fi, and connecting to the WLANs enabled for this purpose. He will not be able to manage anything in the CPE’s configuration, and his access will be limited just to get internet service. His connection is expected as well, to be secure and to be able to keep his privacy safe.

On the LAN side of the CPE the architecture will be characterized by two different networks, as shown in the next figure:

- **The Home Network:**
  This is the network where all the private devices from the end-customer (Private User) will be connected to. This term in this case include both wired and wireless networks that are managed and created from the CPE device or the Home router, with the purpose of giving internet access to the Private User’s
equipment, as well as communication between themselves, e.g. can be used to connect the computers of the house to the printer using the wireless network, or to enable the VoIP service on the IP Phone.

Most likely to be administered by the private user. Even if the home user is supposed to be the administrator of the network, using TR069 the ISP will be able, as well, to remotely manage different devices connected in the private network if is required, e.g. IPTV set-top boxes.

The security has been enhanced by only activating the access to the router configuration by a especially designed web interface powered by LUCI, the official web interface for OpenWrt, this interface has been developed to fulfil the security requirements, as well as the user will have a friendly environment for the configuration of its router like basic routing settings, Wi-Fi security, firewall settings and others, in order to assure the communication and the service.

- **The Public Network:** This network’s purpose is to provide internet service and offload the data from the mobile network of the subscribers within the radio range of the CPE Wi-Fi signal, Public Users, granting major traffic offload from the mobile networks (Smartphones, Tablets, PDAs, Laptops). No further permits in the router should be given to the users connected using this network, for security and privacy reasons.

  The service provider will fulfil the role as manager of this network and he will be in charge only for the functionalities regarding the Public Network, in order to guarantee the internet access to the private and public customers, update of the firmware and to keep security on the device and the network.

It is important for these networks to be shielded from each other, this is done in
order to avoid security intrusions into the device from the persons that gain access through the Public Network and for keeping the privacy of the home user. For this, further specifications will be given in chapter 5.

For this thesis, the names of Private User, Public User, Home Network and Public Network will always refer to what has been exposed in this chapter.
Chapter 3

TR-069 Overview

In this chapter it will be introduced the CWMP protocol or TR-069, this protocol has been chosen to be used to remotely manage the CPE device, and it is important to understand its main characteristics, security description and other features of it, in order to comprehend its operation and how it will be used by the network operator, and how it will affect the split of the management.

The Broadband Forum[6] is a non-profit consortium formed by approximately 200 world leading technology companies working in the sectors of networking, computing, telecommunication equipment manufacturing and service providers. Established in 1994 with the scope to create guidelines for broadband network system development and deployment. Originally known as the ADSL Forum and later the DSL Forum, the Broadband forum got its actual name after the join of the IP/MPLS forum, becoming after this union the main body for upcoming generations of IP network specifications.

On may 2004 the Broadband Forum released the first version of their technical report number 069 (TR-069)[39] where they described the CPE WAN Management Protocol or CWMP. This protocol was created to establish communication between a CPE and its ACS (Auto-Configuration Server), covering secure auto-configuration of the CPE among other functionalities in a common framework.

3.1 CWMP Features

The CPE WAN Management Protocol is intended to support a variety of types of CPE, including routers and broadband access equipment, as well as different functionalities to manage a large group of CPE, including the following main capabilities[39]:

- Auto-configuration and dynamic service provisioning:
  CWMP was designed as a remote management protocol that allowed an ACS to attend a CPE or collection of CPE parallel contacting the CPEs based on a variety of criteria. This criteria based on the identification mechanism that has been included in TR-069 can be individual criteria such as specific service
support required from the owner of a CPE or collective criteria, that can go from hardware characteristics like vendor or the model of the CPE, to software characteristics like firmware image version.

Moreover, the provisioning mechanism allows a CPE to be provisioned even from the time of the initial connection to the broadband access network, and the possibility and ability of further attendance for re-configure or re-provisioning at any time after. This mechanism includes, by the way, support for a synchronous ACS-initiated re-provisioning of a CPE, as well as, additional tools to manage the CPE-specific components of optional applications or services, specifically those requiring a special care for the security, like those involving payments. Nonetheless, it is known that the technology is still evolving and that some capabilities or services are not yet included in the latest version of CWMP, for this, the provisioning mechanism allows straightforward future extension to allow those missing features.

- **Software and firmware image and module management:** Software and firmware tools management are well implemented in CWMP. This tools include version verification, download of the updating file (ACS initiated downloads and optional CPE initiated downloads) and update success or failure notification. With this tools the ISP can assure that all the CPE will always be updated, avoiding security problems for outdated CPEs and it is guaranteed to the customers that they are always getting the best service, possible. Additionally, the CPE WAN Management Protocol allows the ACS to manage, in an effective way, additional services activation for those clients that require so, inasmuch as, it enables modular software management (install, update and uninstall modules, start and stop applications on the CPE, inventory of the software modules available on the device), success/failure implementation notifications, and execution environments on a CPE(enable and disable of the environments).

- **Status and performance monitoring:** ACS periodical reports and monitoring of the CPEs are supported and encouraged in the CPE WAN Management Protocol, tools are available to gather valuable information from the status and performance of the CPEs, giving to the service providers essential information and further availability of statistics. Furthermore, some mechanisms are defined so the CPE can inform of any changes in its state or configuration to the ACS, keeping like this a record of configuration and avoid security threats.

- **Diagnostics:** The CPE WAN Management Protocol provides support for a CPE to make available information that the ACS may use to diagnose and resolve connectivity or service issues as well as the ability to execute defined diagnostic tests[39].
3.2 CWMP Protocol Stack and Operation

The interaction between the ACS and the CPE is based on the implementation of RPC methods. These methods define a mechanism on how the ACS reads or write data from the CPE, for configuring or monitoring. Between the functions defined by the RPC methods it is implemented the download and upload of files on the CPE, the main purpose of this is the firmware update of one or several CPEs, therefore, unicast and multicast transfer protocols can be used, as HTTP/HTTPS, FTP, SFTP, and TFTP for unicast and multicast protocols include FLUTE and DSM-CC.

RPC Methods are encoded using standard XML-based syntax called SOAP. CWMP recommends using the SOAP 1.1 protocol, Simple Object Access Protocol. The reason to use SOAP in CWMP is because SOAP is platform independent and runs over HTTP 1.1 protocol. The use of HTTP makes CWMP more flexible compared to other distributed object technology such as DCOM, which requires communication through a specific communication port (port 135). A corporate firewall will typically block this port and other ports to prevent malicious access to the web server in the demilitarized zone (DMZ); however, firewalls frequently opens port 80, the default port for HTTP. SOAP’s security is enabled using SSL 3.0 and TLS 1.0 specifications, which run over standard TCP/IP protocol.

The CPE device acts as the HTTP client and the ACS acts as the HTTP server, although this role can be changed sometimes during communication, this switch of role has forced the developers of CWMP into developing their own SOAP implementations since SOAP was intended to be a unidirectional protocol with clearly specified roles of client-server, as a result, some interoperability problems have arisen due to miss-interpretations of the SOAP messages. HTTP can also be run directly over TCP/IP if security is not a major concern.
3.2.1 Operation

The sessions in CWMP are intended to be always initiated by the CPE, this sends a “Connection Request”, which is a simple HTTPGet made on the CPE at an arbitrary URL/port set by the CPE, although the CPE should start the session, the ACS can stimulate the CPE to do so. TR-069 requires the use of HTTP basic, HTTP digest, or Certificate based authentication. Authentication occurs in both directions, the CPE authenticates the ACS’s Connection Requests and the ACS authenticates the CPE’s session initiation. Once the underlying protocols have established the connection (over SSL), the CPE sends first an inform request RPC method, where the CPE includes the reason of establishing the session and some basic parameters that are required by the ACS, then, the ACS replies with an inform response and complete the inform RPC. After completing the RPC if the CPE requests no other RPC methods an empty message is sent, and then the ACS can start any of the RPC methods needed, after finishing all the procedures, the ACS will send an empty HTTP response as well, indicating that no longer communication is needed meanwhile, and the session will be closed, the underlaying protocols will terminate the connection and the session will be declared “successfully terminated”.

![CWMP communication example](image)

Figure 3.2: CWMP communication example

3.3 Other Technical Reports

Besides what has been specified in TR-069, there are other reports related to CWMP that complement the operation of the protocol and other enhancements for the support of new types of devices that have come out in the market in the past few years. The next figure, taken from the technical report TR-106[40], by the broadband
CHAPTER 3. TR-069 OVERVIEW

From the figure it can be seen that the whole integration is based on what is specified in TR-069, basic requirements, and additionally TR-106, which defines the rules for how the data model of CWMP must be designed. Behind these two reports three other reports come out, these reports are called root describe the data models that should be used in CWMP, it is encouraged to use these ones, but not mandatory. These three reports are: TR-098, which defines the Internet Gateway Device Data Model for TR-069; and TR-182 issues 1 and 2, which define the device data model for TR-069, divided in two different issues because issues 1 and 2 are not compatible. On top of them TR-143 and TR-157, described as component objects for the root data models and intended for performance test, monitoring and general purposes. Finally, it is introduced the service objects, and these reports refer to the data model description of additional services, like Femtocell support, voice services, storage services and STB devices for digital television services, the reports are:

- TR-104: Provisioning Parameters for VoIP CPE
- TR-135: Data Model for a TR-069 Enabled STB
- TR-140: TR-069 Data Model for Storage Service Enabled Devices
- TR-196: Femto Access Point Service Data Model
Chapter 4

Security Background

Embedded devices are likely to be target for attacks because they do not use malware detection, they are always on-line and they are not as monitored as general purposes servers. These CPEs are in constant risk not only for intrusions that may come over the DSL line, but also coming from the wireless networks and the private LAN.

4.1 Wireless Network Security

4.1.1 Wired Equivalent Privacy (Wired Equivalent Privacy)

WEP or Wired Equivalent Privacy, was defined by the IEEE in the 802.11 standard. It was the first wireless security mechanism, intended to provide to the WLAN with confidentiality, access control and data integrity, with an implementation based on shared key secrets and the RC4 stream cipher[12]. Even though its good intentions, some flaws in the WEP design were quickly found and exploited. WEP has been demonstrated to be vulnerable due to its small Initialization Vector, poor stream cipher implementation, lack of prevention on forgery packets and replay attacks and lack of key management and poorly update[2],[45]. After the release of WEP version 1 other versions came out like WEP 2 or WEP+, this new protocols were released trying to solve the most known flaws on the first version, but not all the hardware support this versions, moreover, these versions are still not secure enough. Nowadays, an attacker would be able to gain access to a WLAN within just a few minutes, granting internet access through the CPE and the attacker would have the possibility to sniff important information from the user, or many other stronger and more dangerous attacks. The usage of WEP for both the Private and the Public network is highly discouraged.

4.1.2 Wi-Fi Protected Access (WPA)

WAP was designed as an interim solution of the security problem in wireless networks[2][12]. WPA is a security protocol similar to WEP, this means that WPA was designed to be compatible with WEP equipment, and it can be enabled with an update of the drivers of the devices. One of the improvements of WPA is the introduction of TKIP
as its security protocol, TKIP is a more advanced and secured version of what was implemented by WEP. TKIP introduced four major changes:

1. A cryptographic message integrity code, or MIC, called Michael, to defeat forgeries.
2. A new IV sequencing discipline, to remove replay attacks from the attacker’s arsenal.
3. A per-packet key mixing function, to de-correlate the public IVs from weak keys.
4. A re-keying mechanism, to provide fresh encryption and integrity keys, undoing the threat of attacks stemming from key reuse.\[2\]

Another big change of WPA was the possibility of being operable in two different modes, the Personal WPA or WPA-PSK mode, created for home or small office use and the Enterprise WPA mode, created for commercial use. There is a main difference between them, and it is that the Enterprise version uses a different user authentication, so no pre-shared key is used, instead, 802.1X+EAP is used, therefore the user will be authenticated using a certificate that will be validated by a radius server. This authentication method generates an excellent control and security in the traffic between the user and the wireless network, plus some other benefits depending on the EAP methods that are supported, like EAP-TLS or EAP-MD5.

Nevertheless of the improvements of WPA, some flaws were found in the protocol. It has been proven that PSK is vulnerable when weak passwords are used, and it can be easily cracked using a brute force or a dictionary attack, due to a flaw in the method in which the encryption scheme is initialized.\[2\]. Moreover, WPA did not solve DoS attacks vulnerabilities due to unprotected management and control frames. \[1\]

4.1.3 IEEE-802.11i (WPA2)

This security protocol was released in 2004 in the 802.11i amendment of the IEEE. WPA2 was design to override the problems of its predecessors and to be a durable standard. Like WPA, WPA2 supports two modes of operation, Home and Enterprise, functioning in similar ways. However, the main difference grounds on the security protocol used by WPA2, CCMP[12]. CCMP is based on AES, Advanced Encryption Standard, which is a publicly accessible, well documented and open, which has even been approved by the National Security Agency of U.S. for cyphering top secret information. Over the years, CCMP has demonstrated to be a more refined and secure protocol than TKIP and has claimed the position of the most robust and secure protocol for wireless networks, currently. CCMP can be enabled in modern devices with WPA, but not every equipment is compatible with it.

\[1\]Temporal Key Integrity Protocol
CHAPTER 4. SECURITY BACKGROUND

EAP-SIM

EAP-SIM is an EAP mechanism for authentication, authorization and accounting created based on the subscriber identity module or SIM, for GSM communications. EAP runs over data link layers, so it can be used with PPP, PPPoE, etc. EAP-SIM is worth to outline, because it can be implemented as the authentication method and distribution of keys in the public network, which also supports privacy protection of subscribers identity, using temporary identifiers and mutual authentication, which means that, using EAP-SIM would prevent evil twin radius attacks. The mutual authentication is first done based on a pre-shared secret of 128 bits, and followed by a peer challenge sent by the client to the server.

EAP-SIM has other benefits as well:

- Compatibility with 3G infrastructure and database.
- No redundant provisioning, preventing two persons trying to log in with the same data.
- Laptops and other devices can be supported, as they can integrate the SIM card using a smartcard reader or a USB dongle.\[22\]

EAP-AKA

Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement created by 3GPP. EAP-AKA is a similar authentication mechanism to EAP-SIM, with similar benefits but uses the 3G authentication algorithm instead of the 2G(A3/A8)\[8\]\[22\].

4.1.4 Wireless Network Security Threats

Brute Force Attack:

Is a method of gaining unauthorized access to the network or to information of it by trying a large number of possibilities in order to break a cryptographic scheme, for example, trying out every possible key in order to decrypt a message or trying with several different user names and passwords in order to get into a network. Since this technique test all the password space, this process can take from some minutes to many years, depending in many factors like the length of the key or the complexity of the characters used in it.\[30\]

Dictionary Attack:

Is a method to break password based security systems, where the attacker tries to gain access while using all possible passwords contained in a dictionary. This dictionary is usually formed by a large list of the passwords with higher chance of being used, specially those that contain places or names. This is a similar attack to the brute-force attack but in the case that the home user is the responsible on setting the password, it is considerably more effective, most of the people choose poor passwords for their networks or devices, so if the set password is ”router” there is a very high chance of success of the attack, but if the password is set as
It is practically impossible that the attacker could gain access to the WLAN.[47] There are many tools out there that can be used to perform dictionary attacks, like coWPAtty or KisMAC and many more free downloadable dictionaries.

**Man In The Middle Attack:**
This kind of attack is performed when the attacker intercepts the communication between two nodes, making each of the nodes believe that the attacker is the other party involve in the communication, getting in this way, access to all the communication between the two parties. With this attack the person performing it can steal passwords and secret private data.

A man in the middle attack is fairly easy to launch when data is unprotected, although, software such as LANjack or AirJack, include scripts that make the job very easy for attackers, because they automate many states of the process. Therefore, a hacker can intercept the messages over the air and change the content to mislead parties that are communicating, in this way a hacker can establish communication with both parties and have a steady communication without the other parties having any knowledge that they are under an attack, making it seem as if the hacker is actually one of the parties[46]

**MAC Spoofing:**
The MAC spoofing attack consist in changing the MAC address of the attacker device, making the others believe he is an authorized person, or hiding his real "identity". Wireless client computers can be configured to behave like legitimate participants in the network. In this manner, a hacker can mimic an authorized user or even act as a honeypot² AP.[46]

It can be thought that by employing MAC filters, would help to avoid MAC spoofing attacks, but this is very easy to bypass, since there are programs that help to detect the information about the users in the area of the Wi-Fi range, sniffers, not even special hardware is required, with any laptop with Wi-Fi capabilities is enough. And therefore, it would be easy for the attacker to spoof the MAC of one of the authorized users and trick the MAC filter, who in the end would only give a false sense of security, instead of adding any value to the security of the system.

**Evil Twin Attack**
An Evil Twin attack is performed when an attacker creates a wireless network using the same SSID as the attacked person’s WLAN. Doing this, the person might connect to the attackers AP instead of his own, giving to the attacker the freedom to

²Honeypot is a term referring to a computer or network, that is specially set to attract people to connect to it and then get information of that user, or set malicious software in the person’s computer, such as trojan horses.
sniff over his personal private information, getting access to passwords, information about bank accounts and more. In order to launch this attack, the attacker should know the password of the twin WLAN, otherwise, the device trying to connect will always choose the user’s AP, because is the one that matches the pre-saved configuration. Much easier and effective is when the WLAN is not configured with any kind of security. This attack is very effective, because there is no easy way for the users to tell to which AP is going to connect if the configurations are identical[19], the only way is to connect and check in the router’s configuration if it is the one that he owns.

**Evil Twin Radius**

Performing an Evil Twin RADIUS attack takes things a step further. When employing WPA2-Entreprise, and a RADIUS is in charge of the authentication, the attacker have the possibility of creating a twin AP with its own RADIUS authentication, doing this, the attacker has a good chance that the user connects to the twin AP. For this, it is recommended that the client always specifies the authentication server when configuring the RADIUS client.[47]

**DoS Attack**

A Denial-of-Service attack (DoS) is an attack with the scope of disrupting a network’s operation. Unlike other attacks, it is not addressed into stealing confidential data or gaining access to a device. A DoS attack is performed by an attacker or many attackers that send to a targeted CPE continuous amounts of bogus requests, which overwhelm the CPU and the memory resources of the device and thus, legitimate users would not be able to connect to the network, or the network could crash[31]. Attacks directed into jamming the signal of the device are also considered DoS attacks.

Bogus request can be premature successful connection messages, failure messages, or other commands using different protocols, DNS, TCP, IP, EAP. DoS attacks can be very effective, and there have been many famous DoS attack to big companies in the internet, like 2010 DDoS attacks to VISA and Mastercard, by WikiLeaks sympathizers[42]. Its effectiveness comes on how easy an attack can be launched and the difficulty of the servers on preventing them. In the case of wireless networks, the most common DoS attacks are based on authorization requests and association requests[37]. Even if DoS attacks does not give any important information to the attacker, it can be performed to get information about the network, after a successful attack and getting the network to crash, it is expected that the users attempt to connect again to this network, with no suspects that the network might have been attacked. In this moment, the malicious attacker can sniff and record the 4-way handshake used in re-establishing the connection between the CPE and the subscribers, and then, analysing all the information cached to try to crack the network security, i.e. retrieve the pre-shared key.
Spoofing and DoS in 802.11 DoS attacks are easy to launch and they are often used in combination with other kinds of attacks. For example, an attacker can spoof the MAC address of clients in the area using the WLANs, and masquerading as the spoofed client, he would be able to send DoS attacks to deassociate or deauthenticate this client, and break this person's communication. This attack can even be done when the communication is using 802.11i, because even in it, this frames are not protected.[23]

Caffe Latte Attack

In October 2007, two researchers from the security area of AirTight Networks, showed to the world a way of cracking into WEP protected networks, that do not even require the attacker to be in the range of the WLAN to be attacked. This attack consists of sending a flood of ARP requests, after which, the client will send the ARP responses, to whom the attacker will start a trace of these responses and further launch a PTW attack. The result is a way to get WEP secret keys in less than 10 minutes[5][7].

4.1.5 Security Threats In the Web Interface

Most of the routers in the market uses a web interface for user/device interaction and configuration. Therefore, web interfaces can be perfect tools for engaging an attack, by this mean attackers have an additional way to theft the router service, or even worse gain root/admin privileges and connect to other devices that are connected in the LAN or WLAN. For this is very important to analyse the possible threats in the web interface, and design the most secure possible web interface.

Authorization Bypass

Authorization bypass refers to a way of exploiting bugs in the implementation of the web interface that would allow an attacker to avoid the authorization process and gain access to the configuration system, partially or even completely in some cases. The most common attack is to avoid calling the default authorization website, and directly call a known address. Another variation of the attack, is to make the interface believe that he has been already authorized to enter the web interface, and therefore, no further authentication is needed. This can be easily done when the authentication is checked over a fixed parameter of the web page's script.[30]

For avoiding this attacks the web interface implement the use of tokens or session IDs, which is given to each authenticated user, and is required in every attempt to access the web interface. Techniques about how to generate the token need to be studied, since is very important that they are generated in a random way and can

---

3Pyckine-Tews-Weinmann, in 2007, these researchers found a way of exploiting the flaws of RC4, to retrieve the key of a WEP protected network.
not be predicted, otherwise, an attacker could predict the token number, and use it to gain access.

File retrieval or Directory Traversal

This kind of attack is done trying to trick the web server with scripts in order to retrieve important information from a file that was not intended to be accessible or available for the public. This kind of attack bases on that the application do not plan any kind of input validation, and it can be launch without help of any special tools, usually attackers retrieve the web interface structure using a spider crawler, and then send the malicious code through the web browser.

Cross-Site Scripting (XSS)

A Cross-Site Scripting or XSS attack consist in the injection of malicious scripts into an input parameter, in order to generate malicious output, exploiting the trust between the user and the web application and hitting the outputs where the web application do not validate or encode the output. “Cross-Site” refers to the security restrictions that the client browser usually places on data (i.e. cookies, dynamic content attributes, etc.) associated with a web site[27][30].

Cross-site Scripting is not based in just one code language and although is usually written in HTML/JavaScript, it can exploit other languages like VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

Once an attacker has succeeded on seeding the malicious code, as soon as the user execute it, it will run within the security context (or zone) of the web interface. Within this framework, the attacker would have the ability to retrieve, modify and transmit any important data accessible by the browser. A Cross-site Scripted user could have his account hijacked, their browser redirected to another location, or possibly to load content from the file system that may execute code under the local machine zone allowing for system compromise[16].

Cross-Site Request Forgery (CSRF)

The main idea behind this attack is to forge a user to perform an action by injecting HTTP requests to the trusted application without the user’s knowledge, so, for a user that has been already authenticated to a trusted web site, is tricked to visit the malicious web site. In this way an HTTP request can be injected by the malicious web site, making the trusted web site to accept the request since it has already been authenticated. The attack can be performed in predictable web sites, were URL and commands are launched in a repeatable way[30]. In this way the CSRF attack exploits the trust that a site has for a user, in contrast to, XSS where the attacker exploits the trust that a user has for a web site.[14]
There are several ways of performing the attack, for example, a user that logs in the web interface, after the authentication process becomes a trusted user, then the attacker knowing that most of the web interfaces in routers are fixed to a number (usually 192.168.1.1) can launch the attack injecting an HTTP request to change the user password, trusting that most of the web interfaces do not request again for the old password because the authentication has been already done. If the request is set correctly, then the password would be changed and the attacker would gain control over the device.

Web sites with rigid static structures, allowing to perform functions through static URLs or HTTP POST and GET request are the most vulnerable.[15]

Privilege Escalation

A user with authorized restricted access, could try to escalate his rights, and gain access to other sections of the configuration of the device, that he is not supposed to. This can happen when a user tries to escalate from basic to administrative or from administrative to root rights. This kind of attack or security breach, can be avoided doing a constant check of the specific rights of the user and the session created, using random session IDs.

4.2 CWMP Security

Security has always being one of the biggest concerns on the remote management protocols, because of this, the broadband forum has taken major attention in it and have designed a protocol with major security improvements, giving to CWMP a big advantage over its remote management protocol predecessors. The security of CWMP has been designed following the next security goals:

1. Prevent tampering with the management functions of a CPE or ACS, or the transactions that take place between a CPE and ACS.
2. Provide confidentiality for the transactions that take place between a CPE and ACS.
3. Allow appropriate authentication for each type of transaction.
4. Prevent theft of service. [39]

The security model is also designed to be scalable, this means that more basic devices will get less robust CPE security implementations and on the other hand, the better and more advance equipment will get fancier security mechanisms. So, the best security will be provided to each device according to each device’s capabilities.

4.2.1 Security Mechanisms

In order to fulfil the goals stated before, the CPE WAN Management Protocol has been designed to use two main security mechanisms:
The protocol supports the use of SSL/TLS for communications transport between CPE and ACS. This provides transaction confidentiality, data integrity, and allows certificate based authentication between the CPE and ACS.

The HTTP layer provides an alternative means of CPE and ACS authentication based on shared secrets. Note that the protocol does not specify how the shared secrets are learned by the CPE and ACS.[39]

TR-069 brings out the best of the services provided by these security mechanisms preventing tampering with the management functions of a CPE or ACS, as well as it prevents theft of service, ensure confidentiality of the transactions, at the same time that asserts appropriate authentication for them.

For the support of devices that are located in the LAN, behind a firewall which would prevent the communication between the ACS and the CPE, the broadband forum has specified in TR-111 a way to resolve this problem, using encapsulation of connection request in a UDP datagram and the STUN protocol, with which the UDP datagram would be able to go through the gateway.

TLS and SSL

TLS (Transport Layer Security Protocol) and its predecessor SSL (Secure Socket Layer Protocol) were created with the scope of providing communications security over the internet. SSL, was created by Netscape, its version 1 never made it out to deployment and version 2 was released in 1995, and then replaced by its version 3 due to flaws in the security. Having three security protocols which were similar to each other, but incompatible was bad for the industry. Due to this the IETF (International Engineering Task Force) introduced a fourth similar protocol called Transport Layer Security (TLS) This new protocol improved the cryptographic algorithm for key expansion and authentication and today nearly all web browsers and web servers support TLS.

The use of TLS in CWMP is highly recommended although is known that not all the CPEs today support TLS, and the ACS should still be able to manage CPS with or without TLS support. When TLS is being used, the authentication of the ACS is done using a pre-loaded certificate or the CPE can be provided with it securely. Considering that TLS provides confidentiality, data integrity, and authentication based on a certificate and shared secret, not using TLS means sacrificing essential aspects of security.

4.2.2 TR-069 Security Risks

The main security risk, is to communicate with CPEs using plain HTTP, instead of HTTPS with SSL3.0/TLS 1.0, then the communication would risk on being eavesdropped and has higher chances of suffering a man in the middle attack, and then the

---

4Simple Traversal of UDP through NAT
CHAPTER 4. SECURITY BACKGROUND

integrity of the device would be compromised. Also, standard means of strong au-
thentication, such as Digest or Certificate-based authentication should be used[28].

It is important to outline that the ACS can not initiate sessions, only send a
connection request to the specific CPE, in which request no data for configuration
can be sent. This means that malicious ACS can not change important data of
the CPE, taking control of the device. Instead, the CPE will only accept configu-
ration changes from the pre-destined servers when the CPE has initiated the session.

On the ACS side, since it is connected and accessible from the internet always,
the device would be in constant risk of attack, if important information about the
clients is stored in it, it would be extremely dangerous if somebody successfully
breaks the security of the ACS, and gains access to all the stored information. Be-
cause of this, as a prevention, this information should be stored in a server from the
ISP, from where the ACS can retrieve the information needed, at the time needed.

4.3 Botnets

Botnet is a term that refers to a network formed of bots. These bots are infected
devices like computers, DSL modems, routers, or any other device that can belongs
to the network, whose actions are controlled by the owner of the malware\textsuperscript{5}. This
person is able as well to trick the devices to perform criminal actions or give away
important information about the owner of the device, and the users connected to
it. The name bot, was given because the devices act as robots (bots), following the
will of his programmer(s).

One of the characteristics os botnets is their capability to reproduce the malware
and therefore spread very quickly infecting other computers and devices[20], travel-
ing through networks, both broadband and private. The botnets are programmed
to hide themselves in the system, working quietly, generating small amounts of traf-
fee or working only when the connection is not being used to not awake any suspects,
going unnoticed from the owner of the equipment. Once a device has been compro-
mised, the malware activates a specific IRC (Internet Relay Chat) that it will use
to get the commands of its owner, to perform different attacks.

Between the most known usages of the botnets it can be identified:

- Monitoring Network Traffic
- Key Logging
- DDoS attacks
- Spam Generation
- Automatic Spreading\textsuperscript{[30]}

\textsuperscript{5}Also known as malicious software or robot software
4.3.1 Botnets in embedded devices

In the last year two types of botnets were identified to have attacked embedded devices using Linux distributions. The first botnet known to have infected home routers and DSL modems was PSYB0T, this was discovered in early 2009, and was shut down by its creator shortly after the detection. It is estimated that the botnet had already infected more than 80000 devices at the moment that the botnet was shutdown\[20\]. This worm targeted routers using MIPS based processors\[6\] with low security configurations, to whom could access by breaking security using brute force attacks over telnet.

Later in 2009 the Chuck Norris botnet was identified\[20\] by researchers in Check Republic. This malware exploits similar vulnerabilities as the PSYB0T. Targeting embedded devices with Linux based firmwares, and accessing the system using telnet launching brute force attacks. The Chuck Norris botnet has been packed using UPX packer and it propagates in the form of packed binary files and shell scripts. The binary is split in two parts, one called (m) which scope is to propagate using the broadband internet providers, and one called m-ram that targets devices in the LAN area. Both scripts work in similar ways, but the chances of the m binary of infecting other devices is much higher, since there are more potential victims to the attack. After the success of the botnet to infect the device, it will initialize the IRC, and continue spreading over the networks.

The main problem about these botnets is the difficulty to detect them in the embedded devices, which due to its lack of resources do not count with proper software to detect these attacks, like anti-virus or anti-malware programs. Although, for these botnets a reboot of the device was enough to remove the malware\[9\], improved botnets might come in the future, and stronger security would be required to prevent and counter their attacks.

\[6\] Microprocessor without Interlocked Pipeline Stages, their latest generation widely implemented in embedded devices because of its low power consumption and heat characteristics.
Chapter 5

Privacy

Due to the configuration of the CPE, granting access to internet to persons on the public network, that may be not even trusted by the Private User, gives a sense of privacy vulnerability that needs to be taken with a lot of care, so the user will not feel threaten by a possible leak of information on the other side of the network.

In addition the fact that remote management is being used, one of the main concerns from the user is how to maintain its privacy, and the protocol objective should be to give the user the certainty of having its privacy safe but at the same time being able to successfully provide a good service and support to the customer.

The passwords stored in the CPE, in CWMP are well protected and cannot be read even by the ACS, so users do not have to worry about it, however, the passwords can be changed by the ACS, the problem with this is that sometimes security rights could be affected by the ACS erasing a password set by the user and granting insecure access to unwanted users or depriving the user of accessing the CPEs configuration.

It is important to identify properly the main role of the private user and the network operator as managers, this, in order to be able to define which data should remain private and which data can be accessed by the network operator.

So, as mentioned before, the two administrators of the device are:

- The Network Operator,
  have to maintain a good service to the client, and is interested in accessing the CPE in order to update, re-configure, monitor or/and troubleshoot the router. The Network Operator main tasks are:
    1. To manage the wide area network connection (Uplink).
    2. To configure basic parameters of the required protocols.
    3. To create and maintain the Public Network and its full service.

- The Private User,
  Needs internet connectivity for all his devices in a secure and safe way. Some of the customers prefer to administrate its own network but most of them simply require to have an access gate to the internet.
5.1 Private User’s Privacy

As said before, the main role of the user is to be the private network administrator, as well as, is the main user and beneficiary of the device. However, it is known that most of the users only want to plug in their routers and be able to connect to internet, not taking much care to the configuration of the network itself but this depends on his personality and on how far he wants to manage his own network. On the other hand, this should not set as reference to limit on how far the ACS will administrate inside the private network and regardless the customer’s personality there are some requirements for the management of the CPE that can only be managed by the service provider, like basic firewall statements, management of the uplink, basic protocol configuration to assure the connectivity between the CPE and the ACS, etc. and some parameters that can only be configured by the private user, like the wireless security, password and encryption.

The privacy policy would be based on: First, that the private user should only access what he really needs in order to fulfil his task as network administrator and the network operator should respect the configurations done by him. Everything that is not configurable by the private user will be responsibility of the Network Operator. And information about the private user devices, concerning LAN configuration and data traffic confidentiality should remain private and should not be accessible by the ACS.

In the next figures, we present a schema of the rights split of the different areas of the CPE, being yellow for the private user, and red for the Network Operator. In the case of shared sections (red and yellow) for both parties, more detailed information about the parameters rights can be found in appendix A.

![Figure 5.1: Reading permits of the different sections of the CPE](image-url)

The first figure refers to the read rights, and the second to the write rights, and as it can be seen only two sections differ between each other, and this is because
some reading rights need to be given to the private user, in order to check that he is receiving the service expected. For example, a private user should not configure any parameters on the WAN of the CPE, but it is of interest to him to check that the WAN is enabled and he is getting proper internet service.

5.1.1 When will the client access the CPE configuration interface?

General Information Retrieve

This section refers to general data of the CPE, its hardware and software, this information should always be available to the Private User for his knowledge and good administration.

Set-up a wireless or wired network

There are 4 main configurations that the user should be able to access, these are: the network configuration, the firewall configuration, the DHCP configuration and the wireless configuration. It is important to emphasize that in these configuration files the user SHOULD NOT be able to reconfigure or delete the parameters corresponding to the network configuration set up by the ACS Manager, this includes other LANS and specially MUST NOT change the Up-Link interface configuration, otherwise the user may lose connectivity to the Service Provider.

To grant access to the network to a new device (Configuration of authentication and authorization)

This refers especially to create temporary wireless networks for temporary guests, if the user do not like to share his private network, he can create a restricted network for granting access to the new device to the internet using his connection.
When a change of the configuration is needed (Firewall update, routing, etc.)

The Forwarding, Redirect and Rule sections the firewall can be preconfigured by the Service Provider according to the needs and the services established to be supported on the CPE, but there are some cases where the Private User may want to create new firewall rules or change the configuration of it. For example, to open a specific port for some service it wants to enable. The choice to create new entries in this section should be done with extreme care. This rules should not overrun the configurations done by the Service Provider, or the CPE could be present some problems and be potentially exposed to new threats.

Trouble-shooting

In the case that one of the devices looses its connection, the private user should be able to check the state of the connection and the function of the CPE in order to make the relevant corrections.

5.1.2 Using additional functions

This item refers to the additional functionalities that may be found in a CPE device, options like using a USB stick in the router for memory enhancement or file sharing, VoIP services, etc. All this functions should be taken into account to be included in the final solution and to leave them out would decrease the quality of the equipment and service that the customer gets.

5.2 Privacy on the Public Network

In a network especially created for providing public internet, it is very important to maintain the privacy and confidentiality of the users connecting to the public network willing to offload the traffic of their mobile devices. It is important to protect the information about the customer and about his handset from other users connected to the public network and also from the private user, including, personal information and account information regarding his service with the ISP or the mobile service provider. On the other hand, the users on the public network, should not be able to connect anyhow to the router configuration, or get any access to the information about the private user or his devices.

Using authentication methods that allow to maintain data confidentiality and create a secure connection, like those mentioned in section 4.1, would guarantee to those users a complete protection of the privacy. Otherwise, if the security is not configured properly any user in the network could try to get information about the others, eavesdropping traffic or performing other attacks.

The use of WPA2 in combination with the authentication of EAP-SIM or EAP-AKA (as introduced in section 4.1.3), would grant the necessary security/confidentiality that the user needs, this can be done enabling the anonymity option on the AAA.
server, who will use the International Mobile Subscriber Identity (IMSI) of each customer, to create a map of random pseudonyms that will be used by the network to identify the users[21], in this way it would be possible to keep the user’s privacy; and on the other hand, the AAA will handle the translation to the IMSI, with which the Mobile Operator can identify the customers.

Additionally, the use of a AAA server in communication to the mobile network operator, would override the need of any other kind of authentication of the user, e.g. setting a website, where the user needs to authenticate; moreover it would make it practically impossible that malicious people try to set rogue APs or evil radius twins.
Chapter 6

OpenWrt Overview

OpenWrt has been chosen to be used for the firmware in the CPE devices selected for the project purposes. In this chapter it will be explained its main characteristics and features, and deeper introduction in two of its most important tools like UCI and LUCI will be done as well, these tools will be very important later in the implementation of the management split.

OpenWrt\cite{35} is a Gnu/Linux open source distribution specially created for embedded devices. It is a free software and it runs under the GPL\textsuperscript{1} license. It first started as a project based on another firmware called Linksys GPL, and it was designed to work only with the Linksys WRT54G, after gaining some popularity and demonstrating to be an stable firmware, some other developers joined the project and in 2005, the first stable release extensible to other embedded systems was launched under the name of White Russian. The latest version was released in April 2010, and is called Backfire. This has been chosen to be the one that will be installed in the CPE devices, since many bugs have been corrected since it predecessors, and many new packages are coming to be available for it.

OpenWrt was designed to be a very powerful and highly customizable operating system, so it provides a framework to build the desired firmware by the addition of additional packages, more than 2000 packages are available on the official site. Over the years OpenWrt has been established as the best firmware solution for embedded devices, towering over its competition in performance, stability, extensibility, robustness, and design. The continuous expansion and the entrance of new collaborators guarantee the future and improvement of OpenWrt on the next years.

\footnote{GNU General Public License, The GPL is the first copy left license for general use, which means that derived works can only be distributed under the same license terms. Under this philosophy, the GPL grants the recipients of a computer program the rights of the free software definition and uses copy left to ensure the freedoms are preserved, even when the work is changed or added to}
6.1 OpenWrt Features

Besides the features of OpenWrt mentioned above, OpenWrt supports many other that are worth of being outlined:

- Wi-Fi services and security using WEP and the more advanced WPA and WPA2, in both home and enterprise versions.
- A centralized configuration interface, called UCI. This interface is intended to make life easier to the users, and bypass problems of syntax and skip troubles on finding the files and libraries that are needed to change.
- Extensible through its package manager called opkg. The software repositories contain about 2000 packages chiefly ones suited for environment with limited resources.
- Network and routing configuration, involving VLAN configuration and use of PPP.
- Filter, manipulate, delay and rearrange of network packets, using firewall, port forwarding, quality of service, IP tunneling and monitoring.
- Full support of DHCP and DHCP leases.
- Support of external devices plugged in USB ports.
- A binary can be compiled using the needed and also self-compiled packages for commercial solutions.
- Dynamic DNS, UPnP and many more protocols are supported.
- An extensible web configuration interface, LUCI[33].
- As a Linux based system, the division of permits and management can be done using different user accounts and groups.

6.2 UCI - Unified Configuration Interface

The configuration of OpenWrt has been all centralized in UCI[35], as mentioned before, this has been done in order to create an easy and straight forward way to change configurations on the devices. Since many programs have their own config-files lying around somewhere, like /etc/network/interfaces or /etc/exports or /etc/dnsmasq.conf and sometimes they use slightly different syntaxes, UCI makes the life easier for the users and has overcome this problems providing a human readable/writable solution, with a simple configuration format and designed to support most of the routers.

UCI, provides also an easy way to the user to retrieve the data of the configuration files, displacing the use of other commands like grep or awk, which is discouraged to use in OpenWrt in these cases. Moreover, UCI employs a special
way to change the configuration files based on submit and commit changes. So, first a change or series of changes are submitted by introducing them in the command line, after that, UCI will expect a commit or revert, if the commit is performed, UCI will carry out the relevant changes and then a complete or partial reboot of the device will be fulfilled, avoiding unnecessary reboot of the whole device when not needed. On the other hand, if a revert is performed, the changes will be deleted and the old configuration will remain intact, avoiding unwanted changes and lost of configuration.

UCI supports some basic configuration files in all the routers, these are:

- DHCP
- Dropbear
- Firewall
- Network
- System
- Time Server
- Wireless

In addition, UCI already supports many other programs created for OpenWrt or supported by it, like, LUCI for web remote configuration, httpd for web server options, QoS for quality of service support, Samba for print services, Mountd for mounting external devices and many others more.

### 6.3 LUCI - Lua Configuration Interface

LuCI was created in 2008 to be a free, clean, extensible and easily maintainable web user interface for embedded devices[33]. Specially created to be compatible with OpenWrt and its UCI centralized system, and designed to use Lua, a programming language designed to be fast, lightweight and embeddable.\(^2\) The use of Lua, gives LuCI and strategic advantage towards its competitors, which being based on heavier script shell programming, lack on performance, installation size and maintainability.

LuCI is an open source project and it is in constant evolution. It is licensed mostly under the Apache License, however some parts are licensed under different licenses like GNU LGPL and GNU GPL. Its extensible nature, allows the users to modify it for special purposes, by adding packages to OpenWrt and day by day more services are configurable using LUCI. LUCI’s default configuration is set to use HTTP, although it can be enhanced to use HTTPS.

\(^2\)Lua is a powerful, fast, lightweight, embeddable scripting language. It combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics, in addition, it has memory management with incremental garbage collection, making it ideal for configuration, scripting, and rapid prototyping.
The CPE Devices

The CPE is the device in charge to provide with internet access to the private customer, as well as, to the customers using their mobile devices accessing internet through the public network. Its main functions are the management of the wireless and wired networks, routing and switching functionalities, and the DSL connection with the ISP. The routers currently in the market are known to be limited in their resources, specially, small memory capacity and limited processing power. This is critical for the security of the router, since no antivirus or antimalware software can be installed in it without overloading the physical memory or the processor, plus, its limitations leave it open for threats based on attacking these limited resources like the DoS attack.

For the purpose of the project two different routers have been chosen for development and testing, the WRT54GL from Linksys and the DIR825 from D-LINK, both routers have been selected taking care to be capable of handling multiple SSIDs and latest wireless security, encryption with WPA/WPA2. Security based on WPA or WPA2 is important because it uses dynamic key encryption, so the information would be protected as it passes over the airwaves, keeping the information protected from other unwanted parties sneaking. The next important requirement is that the CPE must have a compatible processor with the latest OpenWrt releases, since it is an OpenSource project and not all the vendors open their chip’s information for public use, OpenWrt does not support every hardware in the market, although, some of the chips which are not public have been reversed engineered and so most of hardware out there is yet compatible. In addition, VLAN tagging is needed for keeping privacy, security enhancement and routing and separation of the different wireless networks.

**Linksys WRT54GL**

The WRT54GL is a residential gateway with Wi-Fi capable, created specially to support Linux based firmwares. This CPE complies with 802.11g and 802.11b (2.4 GHz) standards and multiple SSIDs capable, moreover, it has an unsurpassed wireless security with Wi-Fi protected due to the WPA2 and MAC filtering capabilities. On the other hand, the WRT54GL has a limited flash memory of 4 MB plus a RAM memory of 16 MB, but it is still enough to load the OpenWrt firmware with
the needed packages. The processor used in it is the Broadcom BCM5352 a 32-bit MIPS architecture processor, compatible with all versions of OpenWrt[43].

D-Link DIR825

The DIR825 is a dual band router supporting both 2.4 GHz and 5 GHz signals at the same time. This CPE complies with 802.11a, 802.11b, 802.11g and 802.11n. In addition, it includes a USB port enabled for using devices like printers or for storage. WPA and WPA2 are available for security over the multiple SSIDs. The DIR825 uses an Atheros AR7161 CPU, which is compatible with all versions of OpenWrt and its 64 MB RAM memory represents no space problem on loading that firmware[44].

This two devices will be set for testing and although the D-Link is a superior device, due to its bigger storage capacity, dual band and better processor, both devices are sufficiently good for the purposes settled.

It is important to identify which are the roles of each party in the management of the network, and the roles of both type of clients accessing internet through the CPE. This home network will be managed remotely through the auto-configuration server (ACS) operated by the internet service provider (ISP). For the private user it is also important to have the freedom to configure and manage his own network, keeping the privacy of his information on the network and not feeling threaten about having his home CPE accessed by unknown persons to him, but this freedom has to
be limited as the operator still needs to provide the service and update, re-configure and trouble-shoot the router, whenever is needed.
Chapter 8

Implementation of the Management Split

In this chapter it will be described the steps that were taken to split the management of the device between the two already mentioned parties, the Network Operator and the Private User. As said in chapter 6, the base component of our split management is OpenWrt, this will be customized to the needs of the project and exposed in the precedent chapters. The customization will be done in two different levels, the first level is to make a division of rights and access to the different areas in the firmware and the next level is to customize the management tool decided to be used for the configuration of the CPE.

Since OpenWrt is a Linux based OS different groups will be created, where each group will have different rights over the configuration files. Creating in this way a low level division of what can each party access and what kind of information is available for them.

The main and only configuration tool for the Private User will be through the web interface. This will be the mean of controlling the device, and this interface has been specially designed to only support changes over what has been decided to be accessible to the private user, according to the group that the user belongs to. For the Public User, no management options will be given, and it is imperative that this user do not crack or break the CPE’s security, for this additional recommendations will be given in chapters 9 and 10, where an additional security analysis of the CPE and OpenWrt will be done.

The network administrator using the ACS, will only access the CPE configuration and management systems using TR-069 which features has been already introduced in chapter 3. For this is important to mention that the client of TR-069 should respect and be designed according to what has been defined in chapter 5 and in the tables in annex A, if this is not honoured the management split would not be successful. Although the privacy and access rights of what can be accessed using CWMP has been studied and described, the creation of the CWMP client is out of the scope of the study of this thesis.
CHAPTER 8. IMPLEMENTATION OF THE MANAGEMENT SPLIT

8.1 The Web Interface

The web interface has been created based on LUCI V.0.9, one of the projects developing web interface for remote configuration over OpenWrt[33]. The basic LUCI is a web interface offering two levels of interaction with the manager, one called essentials, which is intended to support basic changes of the configuration of the CPE, as well as to inform the user on general statistics and overview of the device. The next level is the administration level, and in this the user finds an environment where more advance and complex configurations can be done, as well as, the inclusion of important log files access. From the administration level a user can practically manage all the basic OpenWrt system and even some supported additional packages, many of this of great importance like the firewall, routes, switch, wireless security, etc. On the other hand, LUCI was designed, as well as OpenWrt, as a one administrator system, this means, that they are not supporting or directed to a multi-user operative system, because of this, additional modifications on the accounting and authentication method of LUCI and its interaction with OpenWrt will be needed. Additionally, the LUCI design will be modified in accordance to a more commercial design and using the model of privacy described in chapter 5.

For the function of LUCI it is needed that the OpenWrt firmware includes the next packages:

- luci-admin-core - 0.9+svn6239-1
- luci-admin-full - 0.9+svn6239-1
- luci-admin-mini - 0.9+svn6239-1
- luci-app-firewall - 0.9+svn6239-1
- luci-app-initmgr - 0.9+svn6239-1
- luci-cbi - 0.9+svn6239-1
- luci-core - 0.9+svn6239-1
- luci-http - 0.9+svn6239-1
- luci-i18n-english - 0.9+svn6239-1
- luci-ipkg - 0.9+svn6239-1
- luci-lmo - 0.9+svn6239-1
- luci-nixio - 0.9+svn6239-1
- luci-sgi-cgi - 0.9+svn6239-1
- luci-ssl - 0.9.0-1
- luci-sys - 0.9+svn6239-1
- luci-theme-base - 0.9+svn6239-1
• luci-theme-openwrt - 0.9+svn6239-1
• luci-uci - 0.9+svn6239-1
• luci-uvl - 0.9+svn6239-1
• luci-web

The main luci files will be installed under the folder of lua, located in: /lib/lua/luci, here, are located the files that control the LUCI interface, its models, its interaction with OpenWrt, the information shown and the authentication. On the other hand, there is another folder important for LUCI, /www. In this folder, LUCI saves the files regarding the visual design of the web page that uses a CSS\textsuperscript{1} design style for it.

Mainly the focus of the changes will be directed to the first folder mentioned, because is the folder with the most information of interest to us. In this folder we will focus over three main parts:

• \texttt{./controller}: This is where the files in charge of generating the templates are located. They have the function to call the functions to create the web pages and links, as well as, the options on the menu bar of the interface. Additionally, some functions are called from this files, some of them are: Log Out, Change Password, Reboot the device, Update and Removal of packets.

• \texttt{./model}: In this section LUCI retrieves the information to be displayed on the web interface, this section uses largely the UCI interface to get, set and remove the information and options selected by the user. This section will be of special interest for the changes to be done, in order to, modify the availability of certain config files and its features.

• \texttt{./dispatcher.lua}: Is the manager of the web interface, and the one in charge of initiating it, as well as, authentication, call of the http headers, and creating the tree that connects all the sections of the web interface between themselves.

Even if these two folder and the dispatch file will be of main interest, there are other files that will be analysed in the next sections, depending on their importance and functionalities.

8.1.1 The Authentication and Accounting

This is one of the main changes that need to be done in the LUCI configuration. LUCI, as well as OpenWrt were designed to be uni-user systems, this means that their designs were based on the fact that only a user with root rights would access the

\textsuperscript{1}Cascading style sheets is a language for describing the presentation of Web pages.
\textsuperscript{2}The (.) refers to /usr/lib/lua/luci
\textsuperscript{3}The (.) refers to /usr/lib/lua/luci
\textsuperscript{4}The (.) refers to /usr/lib/lua/luci
web interface, and this user would be able to change everything what he wants on the router’s configuration. So, there are two main problems identified, authentication and authorization of one or more users different from root, and avoid on giving root rights to this new user(s).

**Changes in dispatcher.lua**

For accomplishing this, some changes had to be done in the file `dispatcher.lua`, which is the one in charge of the authorization. Moreover, there will be two levels of security according to the user needs, this has been done in order to separate the administrator of the router from potential users that would only need basic information of the router and are not very familiar with network configuration.

First of all, a new function has been created `get_permit()`, this function is in charge of retrieving the level of permission of the user that wants to access the web interface.

```lua
function get_permit(self)
    if self.authsession then
        if require "luci.fs".isfile("/tmp/luci-sessions/authorization..self.authsession") then
            local auth = require "luci.fs".readfile("/tmp/luci-sessions/authorization..self.authsession")
            ..value = string.find(auth, 'authorized=(%a*)%n')
            self.permit = true and value ~= ""
        end
    end
end
```

The first check that is done in `get_permit` is whether or not an authorized session has been created for the specific user, if that is true, then a file that contains all the authorized users and their level of access is read, this file is created every time a new user logs in LUCI, and is created in accordance to the group that the user belongs to. Then the level of authorization is retrieved from the file and copied to the global array of values called `context`\(^5\).

The variable `accs`, is a variable that contains the username of the authorized users to log in into the system, but since it was intended that only a root user would log in, the way it is written had to be changed as well. It will be written not as a string but as a hashmap of strings, where the variable corresponds to the username and the key corresponds to the group from where the user belongs to.

```lua
local def = (type(track.sysauth) == "string") and track.sysauth
```

\(^5\)In LUCI, `context` is an array of values with important information about the session, like the token, path and permit values, among others. This is very important and will be mentioned in further explanations.
After retrieving the list of authorized users, it is checked whether the username trying to log into the system exists in that list. Then the password is checked if it matches with the password stored in the device to the specific user. The password is supposed to be unique for each user for security reasons. If at least one of those conditions is not verified, the program return an empty response, and it is informed to the user through the web interface.

Otherwise, if a session ID has not already been created, so if is the first log in, a new ID is given to the session using a generator random function of 16 bits, in hexadecimal numbers called uniqueid. This function uses the urandom function. This function uses data and devices available to the kernel and estimates the amount of entropy collected from these sources to generate the numbers, making it astronomically difficult to predict, since this environment entropy can hardly be replicated. With the session ID generated, now it is possible to create a file that belongs only to that session, stored under /tmp/luci-sessions/ and always beginning with the prefix authorization plus the session id number, with the information of the username that owns it, and his permit level.

```
local accs = {}
local function salva(x,y,z) return z end
local users = util.split(require "luic/fs".readfile("/etc/passwd"), ",")
for i,str in pairs(users) do
  if str ~= "" then
    key = salva(string.find(str, '::*:([^d+]*):')
    accs[string.sub(str,1,string.find(str, ':')-1)] = key
  end
end

local sess = ctx.authsession
local verifytoken = false
if not sess then

  if not util.contains(util.keys(accs), user) then
    if authen then
      ctx.urltoken.stok = nil
      local user, sess = authen(luci.sys.user.checkpasswd,
        accs, def)
      if not user or not util.contains(util.keys(accs), user)
        then
        return

    else
      sid = sess or luci.sys.uniqueid(16)
      if accs[user] == "0" or accs[user] == "1" then
        ctx.permit = true
        require "luic/fs".writefile("/tmp/luci-sessions/authorization.
          "..sid , "authorized=true\username="..user)

 16 bits have been set, so there are 65536 different numbers that can be generated. For an attacker is almost impossible to predict a session, although this number can be set higher if is required.
```
CHAPTER 8. IMPLEMENTATION OF THE MANAGEMENT SPLIT

Finally, the session is created, and the session id is used as token of itself, this is important for avoiding CSRF attacks, because sessions’ url can not be predicted.

Changes in the general header of the web pages

LUCI uses a general header that is common to all the pages in the web interface, and it is stored under: /view/themes/openwrt.org/header.htm. The header will be used to avoid XSS attacks. The way to avoid it, is to check if the host trying to access had already created a session with the web interface, and if the token corresponds to the specific user. For this, random and not consecutively tokens are generated for each session. The next use that will be given to the header, is to check if an authorized user has the required permits to access the requested web page. So, the first part is done, reading the context variable, if this contains a valid permit characteristic, is because the user has been previously authenticated using the actual session id. Unfortunately this is not always true, and some links between pages do not send all the information of the session in its context variable and for this cases a especial check must be done afterwards. The second checking is done over the parameter authsession of context, this parameter is created when the user logs into the web interface, and even if an unauthorized user tries to steal the token of another authorized session, the attacker will not be able to by pass this check. So, using this parameter it can be checked if a session has been created for this user, and verify the username and its permits.
CHAPTER 8. IMPLEMENTATION OF THE MANAGEMENT SPLIT

```lua
value = string.find(auth, 'authorized=(%a*)
permit = true and value == ""
end
end
```

With the information of the user and its given permits, then the block of unauthorized and not allowed users is performed. This is done checking if the address required is valid and the permits are given, moreover, if the required address correspond to one directed only for administrator users and the user is not allowed an error message is returned, otherwise the web page is loaded and shown to the user.

```lua
if node.title and permit ~= nil then
  if string.find(node.path[1], "admin") and not permit then
    return luci.dispatcher.error404("User not authorized. Please log in as administrator")
  end
end
```

Finally, a line has to be modified in order to show or not the change of mode to administrator mode in the menu bar, this is done by simply adding the condition of `permit`, in line 152, as shown next:

```lua
if node.title and not node.hidden and permit then
```

**Creation of New Accounts**

For the creation of new accounts in LUCI, a whole new module had to be created, this new module has been called `newuser.lua` and has been set under `./model/cbi/admin_system/`. This module will use the `simpleform` of the LUCI environment, this is a form very simple and especially used when no data needs to be shown.

```lua
f = SimpleForm("newuser", translate("a_s_newuser"),
              translate("a_s_newuser1"))
```

This module will count with four options, the username, the password, confirmation of the password, and the level of access the new user will have, two options has been set, basic and administrator.

```lua
nuser = f:field(Value, "nuser", translate("a_i_user"))
nuser.username = true
plevel = f:field(ListValue, "plevel", translate("a_i_plevel"))
plevel:value("", translate("cbi_select"))
plevel:value("Basic")
plevel:value("Administrator")
```
The next step is to define the function the web page will execute after the validation or create button is clicked. First, the password needs to be checked, as both texts written should match. Here an additional check could be placed to check the password’s complexity in order to make the system more secure. It can be asked to the user to choose passwords with at least 8 characters and using symbols, numbers and both upper and lower case. This can be easily checked using the patterns of default of Lua, %d for digits, %l for lowercase %u for uppercase and %x for not alphanumerical.

```lua
function pw2.validate(self, value, section)
    return pw1:formvalue(section) == value and value
end
```

After checking that the password is valid, we proceed to create the account. Another check should be done first with the scope to avoid double identities, this means that two accounts under the same username should not exists. Then, the new user is assigned to a group whose rights has been previously set, in this case the number 76 has been use for the group with basic rights, and the number 1 has been use for the group of the administrators. Finally, the user is created and the password is set. The user is inform of the results using the two LUCI messages methods, *.message for successful results and *.errmessage when errors occur.

```lua
function f.handle(self, state, data)
    if state == FORM.VALID then
        levelopt = tostring(data.plevel)
        local accs = {}
        local stat
        local function salva(x,y,z) return z end
        local users = util.split(require "luci.fs".readfile("/etc/passwd"), "\n")
        for i,str in pairs(users) do
            if str == "" then
                key = string.sub(str,1,string.find(str, ':')-1)
            else
                if key == "root" then
                    tmp = str
                end
            end
            if util.contains(util.keys(accs), data.nuser) then
                f.errmessage = translate("user_exists")
            else
```

if levelopt == "Basic" then group = "76"
else if levelopt == "Administrator" then group = "1"
else
    f.errmessage = translate(tostring(data.plevel))
    return f
end

password = string.find(tmp,'(:.*):0:')
local newuser = data.nuser.."":""..password.."":"
..group.."":""..group.."":""..data.nuser.."":"
.data.nuser.."":/bin/ash"

local writescss = require "luci.fs".addline("/etc/passwd",newuser)
local mkdircss = fs.mkdir("/"..data.nuser)
if writescss and mkdircss then
    stat = sys.user.setpasswd(data.nuser, data.pw1) == 0
    if stat then
        f.message = translate("a_s_user_created ")
    else
        f.errmessage = translate("unknownerror" )
    end
end

data.nuser = nil
data.pw1 = nil
data.pw2 = nil
end

return true

8.1.2 The Discarded Options

LUCI is a web interface that allows to control most of the router configuration remotely, our new interface has been changed with a view of following three important principles, the first one is, to avoid giving the private user the option of changing many parameters for the reasons that has been exposed already. The second principle is to protect important information that could be used by attackers or malicious users to crack the router’s security, information about packages installed, processes, hosts, and configuration. The third principle is to simplify, make a simple interface for the users. Because most of users have no knowledge about how to configure the router, restricting the options from where the user can choose and showing just what is necessary, will make the configuration process much simpler. On the other hand, many of the modules could not be completely removed because some features had to be preserved, in this section are presented only those that had been completely removed. The partially modified files will be presented in the section 8.1.4.

A total of eight (8) modules have been discarded from the new interface configuration. All these files could be found in the folder ./model/cbi/ and they are
catalogued in the next table.

Table 8.1: Files removed from the LUCI configuration

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/admin_network/vlan.lua</td>
<td>This module is in charge of controlling the switch. An important piece in order to split the traffic in the router and therefore will be only manage by the Network Operator.</td>
</tr>
<tr>
<td>/admin_system/sshkeys.lua</td>
<td>The function of this module is to create new SSH keys, this is deleted because the SSH port of the router will be strongly blocked of being accessed by any private user and with more reason to the attackers.</td>
</tr>
<tr>
<td>/admin_system/processes.lua</td>
<td>This module can be deleted or left with modifications for information of the processes being held in the device, but the functions of start, stop and restart need to be deactivated. The recommendation is to delete it completely.</td>
</tr>
<tr>
<td>/admin_system/leds.lua</td>
<td>This is in charge of configuration of the led lights of the CPE. This feature is of no interest to the user and it should be left to default configuration.</td>
</tr>
<tr>
<td>/admin_status/routes.lua</td>
<td>This module has been deleted because the static routing has been all centralize under the section of the network, where statics routes can be created, plus important information could be eavesdrop by the private user about the routing table for the public network.</td>
</tr>
<tr>
<td>/admin_services/dropbear.lua</td>
<td>In this module can be found options about the configuration of SSH, like the port of use and if users are allowed to connect or not, as mentioned before, this should be deactivated for security issues.</td>
</tr>
<tr>
<td>/admin_services/crontab.lua</td>
<td>This module’s function is to schedule tasks in the CPE; If a private user with the require knowledge uses this function, the CPE could be compromised and easily cracked.</td>
</tr>
<tr>
<td>/admin_services/httpd.lua</td>
<td>This modules were design so the http server could be modified by the user, for security reasons and avoid miss configurations this will be deleted. Moreover, these are functions with no interest to the private user.</td>
</tr>
</tbody>
</table>

8.1.3 The Added Options

As well as some modules had to be deleted, new modules have been added to the new interface for enabling options that were not considered by the LUCI developers, one of the modules has been already explained in section 8.1.1, where the module of creating new users was exposed. Besides this one, three other modules have been
added to the web interface. The first file is related to CWMP, and is intended to show relative information to the user about the CWMP service status.

For this module a simpleform structure has been chosen, and in this module the user will be able to read information about the version of the CWMP client, the packet installed, the data model and if the CWMP client has been able to connect to the ACS and is up and running, plus an additional option of transferring management responsibilities to the network operator.

```python
f = SimpleForm("cwmp", translate("cwmp"), translate("cwmp1"))
s = f:section(SimpleSection, translate("networks"))
cwmpinfo = fs.readFile("/etc/cwmp.conf")
newstate = "."
dm = s:option(DummyValue, "datamodel", translate("cwmpdm"))
function dm.cfgvalue(self, section)
    datamodel = string.find(cwmpinfo, 'TR(%d+)')
datam = "TR−" . . datamodel
    return datam or "−"
end
cv = s:option(DummyValue, "version", translate("cwmpcv"))
function cv.cfgvalue(self, section)
    cwmpversion = string.find(cwmpinfo, 'cwmpversion.+(%d%p%d
∗\n')
    return cwmpversion or "−"
end
ca = s:option(DummyValue, "activated", translate("cwmpca"))
function ca.cfgvalue(self, section)
    running = false
    for i,j in pairs(ps) do
        if string.find(j.COMMAND, 'cwmp') then
            running = true
        end
    end
    if running then
        show = "CWMP Running"
    else
        show = "CWMP NOT RUNNING! Please contact us immediately."
    end
    return show
end
ci = s:option(DummyValue, "installed", translate("cwmpci"))
function ci.cfgvalue(self, section)
    return util.exec("opkg list_installed | grep cwmp")
end
```

This additional option has been created for those users that have little or none knowledge of the CPE configuration and would like that the Network Operator manages
the private network as well.

```
cla = s:option(ListValue, "lanactive", translate("cwmpcla"))
cla:value("1","Yes, Please")
cla:value("0","No, Thanks")

function cla.cfgvalue(self, section)
   lanactive = string.find(cwmpinfo, 'lanactive%s=\"%s\"(%a+)'
   n-')
   if lanactive == "true" then
      up = "1"
   elseif lanactive == "false" then
      up = "0"
   end
   return up
end
```

Although it has been only set as an optional flag, the real adjustment would need to be done in the area of the CWMP configuration. The value of this flag will be stored in the CWMP configuration file, under /etc/cwmp.conf.

```
function cla.write(self, section, value)
   if value == "0" then
      newstate = "lanactive=\"false\"
      f.message = "Now you are managing your LAN network, remember you can always reactivate the remote-management service."
   elseif value == "1" then
      newstate = "lanactive=\"true\"
      f.message = "Your network now is being remotely-managed by us, seat and relax."
   end
   if newstate then
      newinfo = string.gsub(cwmpinfo, 'lanactive%s=\"%s\"(%a+)\n-1', newstate)
      fs.writeFile("/etc/cwmp.conf",newinfo)
      cwmpinfo = newinfo
   end
   return newstate
end
```

The next module added is /model/cbi/mini/routes.lua, this module is a modified copy of the module in /model/cbi/admin_network/routes.lua, which has been changed to be an only readable section, differing from the administrative one, which is used to write new static routes as well. Some lines needed to be removed, and some of them modified, like changing the type of variable from Value to DummyValue, which is the only readable variable employed by LUCI.
require("luci.tools.webadmin")
m = Map("network", translate("a_n_routes"), translate("a_n_routes1"))
local routes6 = luci.sys.net.routes6()
local bit = require "bit"

s = m:section(TypedSection, "route", translate("a_n_routes_static4"))
s.anonymous = true
s.template = "cbi/tblsection"
s:option(DummyValue, "interface", translate("interface"))
s:option(DummyValue, "target", translate("target"), translate("a_n_r_target1"))
s:option(DummyValue, "netmask", translate("netmask"), translate("a_n_r_netmask1"))
.s:option(DummyValue, "gateway", translate("gateway"))

The next part is for support of the upcoming IPV6 networks. Although, IPV6 is the future of the networks, not every option is supported for the protocol, and if it wants to be fully supported, future modifications will be needed.

if routes6 then
    s = m:section(TypedSection, "route6", translate("a_n_routes_static6"))
    s.addremove = false
    s.anonymous = true
    s.template = "cbi/tblsection"
    s:option(DummyValue, "interface", translate("interface"))
    s:option(DummyValue, "target", translate("target"), translate("a_n_r_target6"))
    s:option(DummyValue, "gateway", translate("gateway6"))
end

return m

Finally, the last module added has been named network_calculations.lua, saved under the folder /controller/mymod/, this one was created to contain important functions that will help to identify if the information is related to one of the forbidden networks to the private user or not. The first function has been created for checking to which network a host belongs. This is an important addition to LUCI because this function will allow to the server to block information related to the hosts connected to the Public Network, guarding their privacy.

function check_inside_network(ipaddr)
    olwinet = get_network_name("olwinet")
    wan = get_network_name("wannet")
    is = false
    uci:foreach("network", "interface",

2Internet Protocol Version 6
The following functions are just auxiliary functions that have been created to support the `check_inside_network` function. The first one returns the maximum IP address possible given an IP address of the network and the netmask. The second one is a function that transform an IP address, into a table of four sections, where every section is one of the parts of the IP address.

Finally, the `compare_ips` function is the one that determines given an IP address and its netmask if a host belongs to a specific network one or not.
After these auxiliary functions, three more functions have been added to the module, the first function called `get_network_name` retrieves the names of the networks created by the network operator, so they can be taken out of LUCI’s shown information, this function gets the information from `/etc/cwmp.conf`, so it is important that the Network Operator uses this file to inform the names chosen for the networks that need to be kept in private, for the uplink the variable name "wannet" has been chosen, and for the Public Network the name “olwinet”.

```lua
function get_network_name(network)
  net = require "nixio.fs".readfile("/etc/cwmp.conf")
  if net and network then
    z = string.find(net, network.."=")
  end
  return z
end
```

The next function is related to the ports, if port forwarding wants to be enabled for Private User’s configuration, there will be a need to protect some special ports, in this case it has been implemented a function that actually protects the CWMP port, but this can be enhance to protect a list of ports that may be important for security, privacy and quality of service. This function uses as well, the `/etc/cwmp.conf` file to retrieve the number of the port that has been chosen for the CWMP service, since, this port is not intended to be fixed to any number. In the same way, a list of port could be saved in that file, from where it would be read and proceed to block any changes related to them.

```lua
function check_port(value)
  local range = string.find(value, "−")
  local cwmp = fs.readfile("/etc/cwmp.conf")
  local cwmp_port = string.find(cwmp, "localport=")
  cwmp_port = tonumber(cwmp_port)
  if range then
    if cwmp_port > tonumber(string.sub(value, 1, range-1))
      and cwmp_port < tonumber(string.sub(value, range+1))
    then
      return ""
    else
      return value
    end
  else
    value = tonumber(value)
    if value ~= cwmp_port then
      return value
    else
      return value
    end
  end
end
```
The last function of this module is called `check_interface`, and this is a simple function created to check the respective name of an interface given the name of the network, this is important for the interoperability of the interface, because not all the devices uses the same code for its interfaces, i.e the DLINK CPE uses the name `eth1` for the WAN, on the other hand, the LINKSYS CPE uses `eth0.1` for the same one.

```lua
function check_interface(intname)
    ifname = {}
    uci:foreach("network", "interface",
        function(section)
            if section.ifname == intname then
                ifname[#ifname+1]=section[".name"]
            end
        end
    )
    return ifname
end
```

8.1.4 The Modified Options

Many modules that had not been completely deleted had to be modified somehow, in other basic modules some functions had been modified to present limited information and avoid configurations of the public network and the up-link. The changes will be presented in four sections, in accordance to their location: files in the main folder, files in the controller folder, files in the model folder and others.

Modifications in the Main LUCI Folder

In the main LUCI folder there are three modules that need to be modified, the module `dispatcher.lua` has been discussed previously in section 8.1.1, and the others are `fs.lua` and `sauth.lua`.

The first module to be discuss here is `fs.lua`, in this module a new function has been created, a function needed to add lines to files stored in OpenWrt. This one is specially used for the creation of the new users.

```lua
function addline(path, data)
    olddata = fs:readfile(path)
    if olddata then
        newdata = olddata .. data
    end
end
```
The second module is `sauth.lua`, in this file only a line has been added, this line is intended to close and delete the file of a given session when the user logs out, this file needs to be deleted to avoid attackers intentionally using closed sessions to gain unauthorized access to the system. This is also important so the CPE’s memory does not get full of these files, even if this unlikely due to that these files are usually just a few bytes big.

```lua
fs.unlink(sessionpath .. "/authorization." .. id)
```

**Modifications in the Controller Folder**

Controller folder is divided in three sub-folders, one for the administration level called `admin`, one for the basic level called `mini`, and one for the firewall called `luci-fw`. There is another sub-folder called `mymod`, but it contains only new modules especially created for this application and is not of concern in this section. First, we will focus on the `admin` sub-folder. The first modified file is `index.lua`, this file has been change to support authorization to all the users in the administration group, before it was set to be accessible only by `root`.

```lua
local accs = {}
local function salva(x, y, z) return z end
local users = require "luci.util".split(require "luci.fs".readfile("/etc/passwd"), ",\n")
for i, str in pairs(users) do
    if str == "" and salva(string.find(str, ":%d+":)) == "0" then
        table.insert(accs, string.sub(str, 1, string.find(str, ":"))
    end
end
page.sysauth = accs
```

Some additional lines have been deleted in this module in order to take out the function `luci components`, this function, intended to show the luci files installed and modify them, has been considered to be only manageable by the network operator and this should not be modified anyhow, unless a special update package comes out.

```lua
page.title = i18n("a.i.ui", "Oberfläche")
page.order = 10
entry({"admin", "index", "components"}, call("redir_components"),i18n("luci_components", "LuCI_Components"), 20)
```
The next modified file is services.lua, in this file the modifications has been limited to remove the code that creates the links to some of the modules already described as unnecessary or threatening to the security and privacy, and that had been deleted, pointed out in section 8.1.2, these correspond to the modules crontab, httpd and lucittpd.

```lua
function index()
    luci.i18n.locale("admin−core")
    local i18n = luci.i18n.translate 1
    entry({"admin", "index", "components"}, call("redirectcomponents"), i18n("luci_components", "LuCI Components"), 20)

local page = node("admin", "services", "crontab")
page.target = form("admin_services/crontab")
page.title = i18n("admin_services/crontab")
page.order = 50
if nixio.fs.access("/etc/config/lucittpd") then
    local page = node("admin", "services", "lucittpd")
    page.target = cbi("admin_services/lucittpd")
    page.title = "LuCItpd"
    page.order = 10
end

if nixio.fs.access("/etc/config/httpd") then
    local page = node("admin", "services", "httpd")
    page.target = cbi("admin_services/httpd")
    page.title = "Busybox HTTPd"
    page.order = 11
end

local page = node("admin", "services", "dnsmasq")
page.target = cbi("admin_services/dnsmasq")
```

The same analysis has been done in the file system.lua, where many options have been taken out as well. This correspond to the options of software, SSH keys, processes and LED configuration.

```lua
entry({"admin", "index", "system"}, alias("admin", "system", "system"), i18n("system"), 30).index = true
entry({"admin", "system", "system"}, cbi("admin_system/system"), i18n("system"), 1)

entry({"admin", "system", "packages"}, call("action_packages"), i18n("a_s_packages"), 10)
entry({"admin", "system", "packages", "ipkg"}, form("admin_system/ipkg"))
```
In addition, the function named `action_packages()` will be deleted as it is of no use any more. On the other hand, an addition has been done in order to include the module for managing new users:

```plaintext
1 entry({"admin", "system", "newuser"}, form("admin_system/newuser"), i18n("a_s_newuser"), 25)
```

Additional lines have been deleted in other two files, in `network.lua` the entrance to the switch configuration has been deleted, so no reconfiguration of it will be supported, and in `status.lua` it has been considered that the information shown by this modules, besides the CWMP one, is not relevant to the users, or is already shown in other parts, or is considered as privacy and security threat, so these entries should be erased, and the next line should be added, for the inclusion of CWMP:

```plaintext
1 entry({"admin", "status", "cwmp"}, cbi("admin_status/cwmp"), i18n("cwmp"), 7)
```

However, the CWMP module might be considered to be transferred to the system menu and then completely delete this menu out of the bar. In addition the functions `action_syslog`, `action_dmesg` and `action_iptables` need to be deleted, because they are not of use any more since they refer to the system log and other options that as said before will not be used in the web interface.

In the mini subfolder only two files needed modification, the first one is `index.lua`, in this file as the one mentioned in the `admin` section the changes has been done in function to authorize multi-users log in.

```plaintext
local accs = {}
```
local permit = {}
local function salva(x, y, z) return z end
local users = require "luci.util".split(require "luci.fs".readfile("/
    etc/passwd"), "\n")
for i, str in pairs(users) do
    if str == "" and salva(string.find(str, ":.+:(%d+):'")) == 0 then
        table.insert(accs, string.sub(str, 1, string.find(str, ":'")
            )-1))
    end
end
page.sysauth = accs

The next file is network.lua, and it has just a simple modification adding an en-
trance over the routes.lua, which addition has been explained previously in section
8.1.3.

```
entry({"mini", "network", "routes"}, cbi("mini/routes", {autoapply=true
}), "Routes")
```

The firewall folder did not require any modification on the controller area, but it
did in the model area, this changes will be mentioned in the next section.

Modifications in the Model Folder

The model folder unlike the controller folder, is subdivided in more parts, a total of
ten (10) sub-folders can be found inside it and most of them belong to the adminis-
trative area, where most of the changes take place. Starting with the admin_index
sub-folder, only one file has been modified. This file, named luci.lua, has been
changed to take out the option to choose the files to keep when the new firmware is
updated since this should be only choice of the network operator.

```
for k, v in pairs(luci.config.themes) do
    if k:sub(1, 1) == "." then
        t:value(v, k)
    end
end
```

```
u = m:section(NamedSection, "uci_oncommit", "event", translate("a_i_ucicommit"),
    translate("a_i_ucicommit1"))
u.dynamic = true
f = m:section(NamedSection, "flash_keep", "extern", translate("a_i_keepflash"),
    translate("a_i_keepflash1"))
f.dynamic = true
```

```
return m
```
It is important to highlight the importance on keeping at most the private users configuration to avoid privacy infringement and disagreements between customers and operators.

The second sub-folder is `admin_network`, most of the files here located had to be modified because here most of the changes of the CPE configuration take place. The adjustments made here were done thinking on isolating the private network configuration and leaving out the public network and the up-link configuration, making invisible the other networks to the private user. However, some information has been left for informative purposes, due to that some information may be relevant to him, i.e. a user would like to know if the up-link of the CPE is actually connected to the broadband network, and the service is working properly. The changes here use widely the created module `network_calculations.lua` located in `/controller/mymod`, and explained previously in section 8.1.3. The specific modifications of the files in `admin_network`, has been documented and they can be found in Appendix B.

The third sub-folder is `/admin_system/` and here only the file `passwd.lua` was modified. The other files were already analysed in previous sections. The changes here were done because of the new multi-user functionality of the interface. `passwd.lua`, as other LUCI files, supposes that `root` is the only user of the interface, because of this, the only password that it was able to change was `root`’s. A part of code needed to be added to check first to which user the session belongs to, so a certain user "user1" can not change the password of a certain "user2", and then a code line was modified changing "root" to a variable, that in this case it has been use the name "user".

```
var = require "luci.sys".getenv(var)
if var then
    ...token = string.find(var.REQUEST_URI, 'stok=(%w*)/')
    local userinfo = require "luci.fs".readfile("/tmp/luci-sessions/authorization..
                                          user = string.find(userinfo,'username=(%w*)')
end

local stat = luci.sys.user.setpasswd(user, data.pw1) == 0
```

The exact same changes have been done in `/mini/passwd.lua`.

The other sub-folders directed to the administrative level, except for the firewall configuration files, have been analysed already, or were not changed.

Now, on the mini level, or for basic users, many of the changes have been done also as for the administrative level, to isolate the private network configuration, but the philosophy of this level is also to provide a basic and simple configuration interface, giving to the user, the necessary tools to configure a network and provide internet service and routing between his devices. In the file `/mini/network.lua`, some lines were added in order to restrict the information shown and the configurable parameters:
CHAPTER 8. IMPLEMENTATION OF THE MANAGEMENT SPLIT

The next line is a modification of the previous LUCI file:

```lua
if v[".type"] == "interface" and k ~= "loopback" and k ~= olwinet then
```

And the next lines were added to support the inclusion of the IP network broadband parameter:

```lua
bcast = s:option (Value, "bcast", translate("broadcast") .. translate("cbi_optional"))
bcast.rmempty = true
```

Everything else written in this module has been discarded; the options supported in those lines were deleted because in most changes the default option would be enough for the good operation of the network, other options were decided to be not supported, because it is not important for home or SOHO networks configuration.

In the file `wifi.lua`, the changes have been a little more complex, under the same assumption of simplifying the interface. Information not relevant as power transmission level, signal level, or noise level and options like wifi disablement, hiding and isolating the network or changing the Wi-Fi mode and encryption method were deleted, and left to be changed only by the user with administrator rights. Additionally, some lines needed to be modified or added. The next lines were modified to restrict the shown networks:

```lua
if not uci:get("network", wan) then
   uci:section("network", "interface", wan, {proto="none", ifname=""})
```

```lua
if v[".type"] == "wifi-iface" and v["network"] ~= olwinet then
```

This lines have been added to support devices with dualband functionalities, like the DLINK-DIR825 and some other lines were added to make the interface more friendly to the user, adding more specified options, so he can make a better choice towards his configuration.

```lua
for i=1,#wifidevs do
   s = m:section (NamedSection, wifidevs[i], "wifi-device", translate("radio_devices..i"))
   s.addremove = false
```

*Small Office/Home Office*
The next lines are included in the same file, and we would like to point out that the devices supported are expected to be the ones taken into consideration and exposed in section 7, since not every chip set give the same names to its features it would require more effort to be able to support all of them, and it is not the scope of the investigation, so the chips supported are broadcom and atheros families, and similar ones to these.

```
if hwtype == "atheros" then
  mode = s:option(Value, "hwmode", translate("mode"))
  mode:override_values = true
  mode:value("", "auto")
  mode:value("11b", "802.11b")
  mode:value("11g", "802.11g")
  mode:value("11na", "802.11n+a")
  mode:value("11bg", "802.11b+g")
  mode:rmempty = true
else
  protocol = s:option(DummyValue, "hwmode", translate("mode"))
  function protocol.cfgvalue(self, section)
    local mode = m:get(wifidevs[i], "hwmode")
    if string.len(mode)==4 then
      hwmode = "802." .. string.sub(mode, 1, 3) .. "+" .. string.sub(mode, 4)
    else
      hwmode = "802." .. mode
    end
    return mode and hwmode
  end
  return
end
```

Finally, in the DHCP configuration of the basic level, mini.lua, a function has been created for checking that the user do not write an entry whose value is not permitted, in this case, the function checks that the number of leases do not surpass a given maximum number of hosts that the network can support, this time 255 has been chosen.

```
function limit.write(self, section, value)
```
value = tonumber(value) - 1

if value < 255 then
    return Value.write(self, section, value)
else
    return ""
end

Additional lines were added for security and privacy of the Public Network:

info = luci.util.split(lease, ",")
if nc.check_inside_network(info[3]) then
    table.insert(leases, info)
end

Further changes were done in one of the sub-folders in model, and they are related to the firewall, but as well as the commented before for the administrative level in /model/cbi/admin_network, they are all directed to protecting the privacy and configurations of the networks that are not configurable by the Private User. The files are located in /model/cbi/luci_fw/ and the specific changes can be found in appendix C.

Additional Commentaries and Modifications

It is important to point out two special sections managed by the module /controller/admin/system.lua, these are backup/restore and flash firmware. Both options can be enabled in the new firmware but further security checks need to be done for these to be enabled, if the security is not enhanced to check the originality of the file a person could easily change the firmware or/and the CPE configuration, for its own advantage. The file created when a backup is performed is required to be encrypted using, additionally, a secret and unique key to the operative system, so it could be only decrypted by an authorized OpenWrt system and it can not be read by a human. Important information is saved in the configuration files produced when a backup is performed and it is extremely important to maintain them secret to attackers. A similar way of recognition of authenticity should be use with the firmware files, OpenWrt and LUCI already have a way of validating images using MD5SUM hash function, however, MD5 has been proven to have flaws and there are known collisions and preimage attacks that have been documented[13]. There are
other hashing algorithms like SHA-2\textsuperscript{9} or Whirlpool\textsuperscript{10}, that have shown to be very secure and can be taken into consideration.

There is another file that is worth to mention an is /tools/webadmin.lua, in this file an important function has been included, this function returns a reference field to the networks that are only related to the Private User. This can be useful specially if the Private User, requires more than one network for his private use. For applying it, some lines needed to be added previously:

```lua
local nc = require "luci.controller.mymod.network_calculations"
o winet = nc.get_network_name("olwinet")
w an = nc.get_network_name("wannet")

function cbi_add_networks_restricted(field)
    uci.cursor():foreach("network", "interface",
        function (section)
            if section[".name"] ~= "loopback" and section[".name"] ~= olwinet
                and section[".name"] ~= wannet then
                field:value(section[".name"])
            end
        end
    )
    field.titleref = luci.dispatcher.build_url("admin", "network",
        "network")
end
```

8.1.5 The Design

Now that we have presented the changes about the content, it is opportune to talk about the changes about the design. The design of the web interface, lies in the folder /www in the root folder. LUCI uses a CSS\textsuperscript{11} style for the design of the interface, the special style of CSS allows to be used on multiple pages without being reloaded, since it can be stored in the browser cache, reducing times and data transfer. The main file for the design is cascade.css, and in this file all the changes for the interface were done, this one is located in /www/luci-static/openwrt.org/. Additional resources like images for background and special modifications for specific browsers may be store in this folder as well.

\textsuperscript{9}Stands for Secure Hash Algorithm version 2, is a widely use cryptographic function created by the National Security Agency (NSA), there are not known collisions showing a good security level.

\textsuperscript{10}The Whirlpool algorithm, is based on a dedicated block cipher W that operates on 512-bit data blocks using a 512-bit key and the hash function is constructed from a compression function using the Merkle-Damgård method\textsuperscript{[11]}

\textsuperscript{11}CSS, Cascading style sheets is the language for describing the presentation of Web pages, including colours, layout, and fonts. It allows to adapt the presentation to different types of devices, such as large screens, small screens, or printers. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation.\textsuperscript{[26]}
There is another folder important for the design of LUCI and it is /www/luci-static/resources/, in this folder are stored some Java Script files necessary for the operation of the interface, and a sub-folder containing all the icons used by the interface.

In the next figure an example of how the Web Interface has been changed can be seen:

Figure 8.1: Example of the redesigned web interface

The texts displayed in the Web Interface that are not retrieved from the CPE configuration, are all loaded from a dictionary stored in /luci/i18n/. This dictionary in LUCI use a format called LMO\(^{12}\) (Lua Machine Object), which is a binary format to pack language strings in a more efficient form\(^{[33]}\). This, eases the task on creating versions of the software in other languages for multi-lingual support.

The complete code for the design of the interface can be found in Appendix D.

8.1.6 Operation On The OpenWrt enabled CPEs

The new web interface has been implemented, and set in the two CPEs chosen for trials. Both devices have been tested using a computer running Windows 7, with a processor Intel Core i5 2.40 GHz and 4.00 GB of memory. The browsers used were Firefox 3.0, Google Chrome 10.0 and Internet Explorer 9.0.

\(^{12}\text{LMO can be generated from PO files, another multilingual format created by the GNU project, using the binaries that can be found in LUCI's website.}\)
In the DLINK DIR825 as expected, navigation was faster compared to the LINKSYS, since it counts with better physical resources. Using Google Chrome and IE, the experience about the design of the web page was good, and the interaction with the web interface trying different configurations changes did not report any identified bug, or any problem of miss configuration. With Firefox the only problem identified was a change on the size of the pre-loaded images of the web interface, this happened in both devices.

8.2 Groups and Access Rights

In OpenWrt three different groups were created. The first group is exclusive for root, and has been left to number 0, as default, no other user should be placed in this group or share the root rights. The second group has been identified with the number 1, and it correspond to the users with administrative rights. The third group was created for the users with basic rights and the number 72 has been designated.

Dropbear should be configured to enable only access to root, and none of the other accounts should be able to use SSH or telnet to connect. The rights of the files and folders, should be changed only giving access to those strictly necessary. For the administrator and the basics cases the permits should be changed in the next files:

- etc/passwd
- etc/config/
- etc/init.d/boot
- etc/init.d/dnsmasq
- etc/init.d/firewall
- etc/init.d/network
- uci and luci

As seen in section 8.1, the limitation of the configuration tool of the router is done mainly through the web interface, on the Private User side, but what has been introduced in this section is important for the good function of the configuration methods and to provide access to both working groups.

8.3 Packages in the OpenWrt Firmware

Next is shown a list of the packages that have been installed in OpenWrt, for the purpose of the project and the function of it.
This list have been taken from the configuration of the DLINK, but this matches as well with the configuration of the LINKSYS in most of the packages, to achieve the interoperability of the firmware with the variety of CPEs. The difference of packages are mainly on the drivers which can be identified for the prefix `kmod`. A full list of these files installed in the LINKSYS is shown next:
### Table 8.2: List of Packages Installed in OpenWrt

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>base-files - 51-r22614</td>
<td></td>
</tr>
<tr>
<td>busybox - 1.16.2-1</td>
<td></td>
</tr>
<tr>
<td>crda - 1.1.0-2</td>
<td></td>
</tr>
<tr>
<td>cwnmp - 0.4-5</td>
<td></td>
</tr>
<tr>
<td>dnsmasq - 2.55-4</td>
<td></td>
</tr>
<tr>
<td>dropbear - 0.52-5</td>
<td></td>
</tr>
<tr>
<td>firewall - 2-9</td>
<td></td>
</tr>
<tr>
<td>hotplug2 - 1.0-beta-2</td>
<td></td>
</tr>
<tr>
<td>iptables - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>iptables-mod-contrack - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>iptables-mod-nat - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>iw - 0.9.20-1</td>
<td></td>
</tr>
<tr>
<td>kernel - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-ath - 2.6.32.16+2010-07-29-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-ath9k - 2.6.32.16+2010-07-29-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-button-hotplug - 2.6.32.16-2</td>
<td></td>
</tr>
<tr>
<td>kmmod-cfg80211 - 2.6.32.16+2010-07-29-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-crc-ccit - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-crypto-aes - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-crypto-arc4 - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-crypto-core - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-input-core - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-input-gpio-buttons - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-input-polldev - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-ipt-contrack - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-ipt-core - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-ipt-nat - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-ipt-nathelper - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-leds-gpio - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-mac80211 - 2.6.32.16+2010-07-29-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-nls-base - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-ppp - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>kmmod-pppoe - 2.6.32.16-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libnn - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
<tr>
<td>libcyassl - 1.4.0-2</td>
<td></td>
</tr>
<tr>
<td>libgcc - 4.3.3+cs-51</td>
<td></td>
</tr>
<tr>
<td>libiptc - 1.4.9.1-1</td>
<td></td>
</tr>
<tr>
<td>liblua - 5.1.4-6</td>
<td></td>
</tr>
<tr>
<td>libm -tiny - 0.1-1</td>
<td></td>
</tr>
<tr>
<td>libc - 0.9.30.1-51</td>
<td></td>
</tr>
</tbody>
</table>
Table 8.3: Drivers Required for the LINKSYS WRT54GL

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>kmod-brcm-wl</td>
<td>2.4.37.9+4.150.10.5.3-6</td>
</tr>
<tr>
<td>kmod-crypto-aes</td>
<td>2.4.37.9-1</td>
</tr>
<tr>
<td>kmod-crypto-core</td>
<td>2.4.37.9-1</td>
</tr>
<tr>
<td>kmod-diag</td>
<td>2.4.37.9-6</td>
</tr>
<tr>
<td>kmod-ipt-conntrack</td>
<td>2.4.37.9-1</td>
</tr>
<tr>
<td>kmod-ipt-core</td>
<td>2.4.37.9-1</td>
</tr>
<tr>
<td>kmod-ipt-nat</td>
<td>2.4.37.9-1</td>
</tr>
<tr>
<td>kmod-ipt-nathelper</td>
<td>2.4.37.9-1</td>
</tr>
<tr>
<td>kmod-ppp</td>
<td>2.4.37.9-1</td>
</tr>
<tr>
<td>kmod-pppoe</td>
<td>2.4.37.9-1</td>
</tr>
<tr>
<td>kmod-switch</td>
<td>2.4.37.9-3</td>
</tr>
<tr>
<td>kmod-wlcompat</td>
<td>2.4.37.9+4.150.10.5.3-6</td>
</tr>
</tbody>
</table>
Chapter 9

Security Threats

In this chapter it will be analysed the security issues of the system proposed based on what has been already introduced in chapter 4. Security threat refers to the dangers of the confidentiality, integrity, availability, or the legitimate use by a person to an event or a resource[3]. The security approach that will be done next, will be based on the perspective of the source from where the attack may be generated, having three scenarios, as we can see in Figure 9.1:

![Figure 9.1: Threat Model for the CPE](image)

9.1 Attacks Coming From The Home Network

The possibility of an attack to the Home Network is strictly related to how well the user implements the security of it. So, the first line of defence should be the education of users, and present them an easy way to set up a connection with the best security possible. In the best case the user will set up a network using WPA2 and CCMP avoiding sniffing over the data, man in the middle attacks, MAC spoofing, and caffe latte attacks. With a very strong password, the user can counter back dictionary and brute force attacks as well. In the worst scenario the user could implement WEP in the WLAN, and be open to all the attacks that have been men-
tioned before, being exposed to a lot of active and passive attacks.

Even if the security of the Home Network is compromised, the CPE should be able to keep giving a good service to the users in the Public Network, as a regular user has no access to the router’s configuration, but by the web interface. If the security of the Web Interface is cracked also, for example, if an attacker that has cracked the Home Network, eavesdrop the password and username used in the web interface, the attacker should not be able to change any configuration besides to what is available for the Home Network, so the integrity of the Public Network, the up-link and other zones of the CPE should still be secure.

Telnet and SSH may be as well source of problems to the CPE, as was presented in the case of the Chuck Norris botnet, additional security precautions have to be taken, for avoiding unauthorized users to be able to access this services.

9.1.1 Attacks to the Web Interface

The web interface has been designed to override authorization bypass attacks, file retrieval, and privilege escalation, further enhancement of the security of the web interface depends also on whether HTTPS is used or not. If used, then the clients would have a secure encrypted channel to the web interface, preventing eavesdropping of account information. the analysis about how a cross site scripting and a cross site request forgery attack could affect the integrity of the application can be a lot more complicated.

In the past, flaws of the LUCI interface, concerning CSRF and XSS, has been exposed, but they have been corrected from v.9.0. Moreover, free tools that test the web sites against cross site scripting are available, and three of them have been used to test our implementation. The way to operate of UCI, is also a barrier for these kinds of attacks, because UCI takes the job on checking the validity of the input. Other parts where UCI is not used, like the password change section, needed other precautions for avoiding XSS attacks. Additionally, a tool for web security scan called Securify, was used to check the vulnerability to XSS attacks, with which four possible attacks were identified:

```plaintext
1 request:
2 GET http://192.168.1.1/luci-static/resources/cbi/?<wslite> HTTP/1.1
3
4 request:
5 GET http://192.168.1.1/luci-static/resources/?<wslite> HTTP/1.1
6
7 request:
8 GET http://192.168.1.1/luci-static/openwrt.org/?<wslite> HTTP/1.1
9
10 request:
11 GET http://192.168.1.1/luci-static/?<wslite> HTTP/1.1
```
This attacks can retrieve the basic CSS and JS files, that are used to generate the web pages. Even if an attacker could retrieve this files, they would not contain important information about the users and their accounts nor configuration files of the CPE. On the other hand, the user can gain knowledge of how the LUCI works, but these files are already public, and can be found in the LUCI repositories, and they do not represent a real threat to the integrity of the device.

Even if now LUCI seems to be a robust interface against CSRF and XSS attacks, there may be flaws which are still not identified, and the best option is to keep the LUCI files updated.

9.2 Attacks Coming from the Broadband network

If the web interface is enable to the up-link as well, this would present similar risks as stated for the Home Network, but, with more potential attackers on the web, the use of HTTPS over LUCI would not be an option any more.

A well configured firewall is always important to prevent DoS attack and to block the access to the SSH port on the WAN. The security in this case is also responsibility of the ISP, and how is designed the authorization and authentication of the CPE, is important the use of PPPoE, a protocol very secure with strong encryption mechanisms, and the fact that CWMP can make periodical checks of the equipment and its status, gives an added value to the security, and would help to prevent MITM attacks.

9.3 Attacks Coming from the Public Network

Using WPA2 with EAP-SIM or AKA, firewall rules and separate VLANs, to only enable internet access through the interface used by the Public Network (Web Interface Configuration, SSH) and keep traffic separated from the traffic from the Home Network, should keep away malicious users from launching attacks to the CPE.

The risk of DoS attacks in this network is a reality, and even unintended DoS attacks can be experienced. For example, if many customers try to connect to the same CPE at the same time, there might be a crash of the network due to the high flow of authorization requests. For avoiding these situations, a limit of incoming authorization requests and on the retry attempts should be set.
Chapter 10

Recommendations

In this chapter a series of recommendations that could be used to make a better and more secure device have been given. Configuration recommendations, as well as additional software and the relation between different features that could be installed in the device have been described.

10.1 WEP, WPA and WPA2

As exposed in section 4.1, wireless networks using WEP are not secure and instead can be cracked in a matter of minutes, giving to the customer a false sense of security. It is recommended to discourage the use of WEP or even discard it as an option of configuration by blocking it as a choice in the web interface.

Even if WPA or WPA2 are used, the network would not be 100% secure, and attackers can exploit other vulnerabilities, performing dictionary attacks, DoS attacks, or by employing rogue APs and client miss-association to gain access to private users data.

Practices like hiding the SSID, do not give any added value to the security of the network, since many different tools have been created, and are free to download, that can easily discover hidden networks, in the area of the attacker. Between the most used tools are AirMagnet and Airjack.

10.2 802.11w

802.11w is a standard for WPA and WPA2, created to prevent DoS attacks launched by spoofing of deauthentication, disassociation, association and authentication request in existing connections. In order to prevent this attacks, 802.11w adds encryption to the deauthorization and disassociation frames, so an attacker employing MAC spoofing could not launch an attack.

For the association and authentication request in existing connections, 802.11w uses a mechanism called Security Association (SA) to prevent disconnecting the already connected client[25]. When an AP receives an association or authorization
request from an already associated client, instead of terminating the connection immediately, as is intended in 802.11, the AP sends an encrypted probe to the client, that the client needs to response, and if this sends back the encrypted response the AP considers that the request was a spoofed and malicious request, and then rejects it, otherwise the request is considered valid and the connection is terminated.

Although 802.11w successfully prevents some kinds of DoS attacks, there are still other DoS attacks class, like RF jamming, EAP spoofing, connection request flooding, etc. that are out of its scope.

10.2.1 802.11w in OpenWrt

The support of 802.11w in OpenWrt is limited to drivers of the ath9k kind, this means that the routers that have been chosen for testing which use the brcm2.4 (LINKSYS) and the ar7xxx(DLINK), are not able to use it. For the routers using that chipset, the 802.11w driver is included in hostapd and wpa_supplicant, in their versions 0.7.2 or later.

Its configuration is done using the wireless configuration files, and setting the rule:

1. uci set wireless.radio0.ieee80211w=1 (optional)
2. uci set wireless.radio0.ieee80211w=2 (required)

10.3 HTTPS Over LUCI

A good way to add another security layer over the web interface is to enable HTTPS over LUCI, this can be done easily adding the next packages to OpenWrt:

These files can be downloaded from the file repository of OpenWrt, and they do not represent a major memory usage for any of the CPEs taken into account, 104.77 kb. Nevertheless, HTTPS does not require major configuration efforts after installation, the http port should be closed for avoiding users on using it and forcing the use of HTTPS always, this can be done easily using UCI:

1. # uci delete uhttpd.main.listen\_http
2. # uci commit
3. # /etc/init.d/uhttpd restart

The first two lines tell to the http daemon to stop listening to the http port, and the third line restart the daemon for the application of the new change.

It is important to mention that the web interface reduces considerably its speed of navigation, tested with Internet Explorer 9.0 and Google Chrome 10.0. With Google Chrome, there were some problems loading of the images on the background of the web page itself. Therefore, an additional care on the LUCI files should be
Table 10.1: Packages required to enable HTTPS in OpenWrt

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (bytes)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>luci-ssl</td>
<td>782</td>
<td>Meta package. Standard OpenWrt set including full and mini admin, the standard theme + HTTPS support.</td>
</tr>
<tr>
<td>uhttp-mod-tls</td>
<td>5825</td>
<td>The TLS plugin adds HTTPS support to uHTTPd.</td>
</tr>
<tr>
<td>libcyassl</td>
<td>69682</td>
<td>CyaSSL is an SSL library optimized for small footprint, both on disk and for memory use.</td>
</tr>
<tr>
<td>px5g</td>
<td>28480</td>
<td>Px5g is a tiny standalone X.509 certificate generator. It’s suitable to create key files and certificates in DER(^1) and PEM(^2) format for use with stunnel, uhttpd and others.</td>
</tr>
</tbody>
</table>

...taken and these might need to be optimized accordingly. One of the most discussed points on the source of delay of HTTPS is the SSL handshake, this, usually takes at least 4 times the time of the TCP handshake, used by HTTP. Therefore, optimization of the session lengths and catching behaviour, could considerably result in a better performance. Nonetheless, this is not the only factor and some other factors need to be taken into account.

10.4 Secure Shell, How to Prevent Attacks

Secure Shell or SSH is a protocol created for data exchange between two devices using a secure channel. Using this, the shell account of the OpenWrt firmware can be access for troubleshooting, configuration changes, manual firmware upgrades or any other change on the device, basically it grants full access to the files in the device with root rights. A break on this port would represent a major threat and the attacker would gain full access and control of the device. Therefore, to prevent attacks incoming to it is of extreme importance.

- Using an IP filter,
  SSH in this routers should be only used for troubleshooting by authorized people, by limiting the IP address or the interface with access to the SSH port. Attempts to access the CPE files using SSH coming from unauthorized people can be limited. Then, e.g. all the incoming attempts from the public Wi-Fi to the CPE could be blocked. This can be done using IP tables.

Example of an IP table. In this case, only the requests coming from the interface eth0 and the IP address 192.168.1.10 would have access to the port 22, which is the default SSH port.
• Using IP tables,
It is also possible to limit the amount of failed attempts incoming to the SSH port, by doing so, the probability of a successful dictionary or brute force attack, would be minimized, since only a few attempts per minute would be allowed.
Example of an IP table. In this case, if there are more than four failed attempts of logging in to the CPE SSH port the CPE will drop any other future coming retry from anyone.

```
1 iptables -A INPUT -i eth0 -s 192.168.1.10 -p tcp --dport 22 -j ACCEPT
```

• Port Knocking,
This is a mechanism that permits to request to open a port or series of ports that are only needed in special cases. The request is done using a sequence of authentication TCP packets directed to specific closed ports on the CPE[36], whereupon the port knocking daemon reads the log files and successfully identifies the sequence, the port would be opened. A port knocking client is already available for OpenWrt for the Backfire version under the name of knock, and should not represent a memory problem since is only 12 kbytes big and is also low on processor consumption. Port knocking has also the characteristic of stealth, since the attacker would not be able to detect if the CPE is using port knocking or not. Moreover, port knocking would give an extra security layer to protect the CPE. Some knowns problems of port knocking like[29], delay on arrival of the packets, or trespassing a NAT, are not of concern in this case where the port would only be opened for access in trouble shooting scenarios, and then the person would access the CPE through a direct LAN connection.

OpenWrt uses a Linux implementation of SSH called Dropbear, this is an open source implementation of SSH version 2, and supports SCP\(^3\) and SFTP\(^4\) is optional. OpenSSH is also available for OpenWrt but Dropbear is a much lighter implementation (110 kB on an x86 processor), reason why is preferred in embedded systems. In this solution the first line of defence would be represented by the port knocking client, hiding the SSH port to unauthorized users, but only as a complementary service to the authentication done for the next layers. Port knocking biggest flaw is that is a one way authorization mechanism, being vulnerable to attacks in which the attacker masquerades itself as a server[29]. The next line in security would be the firewall, and the last one the user authentication done by Dropbear.

---

\(^3\)Secure Copy Protocol
\(^4\)SSH File Transfer Protocol
10.5 Intrusion Detection System and Intrusion Prevention System

Intrusion detection systems or IDS enhances the network security by monitoring system activities for signs of attack and then, based on the system settings, responds to the attack as well as generates an alarm\cite{12}. IDS are considered to be security reactive, this means that the attack is launched, and based on how the attack affects the, the IDS generates a response to it. An Intrusion prevention system or IPS, on the other hand, analyses the traffic and takes actions before letting it pass to the network\cite{34}. In the networks security field an IPS is considered a much powerful tool, because of its fast response to the attacks before the attack can make any damage to the network.

The problem of using these techniques in embedded devices, is the lack of resources, getting as result a low performance of the device, or overload of the memory. Because of this, is usual to keep the log files in a separate data base, but for the usual home administrator this is not an effective solution. Additionally, many administrators had problems with IDS in the past, specially caused by overloads of generated warnings, because of miss configurations and the so called false positives\cite{5}, considering a network like the one proposed in section 1.1, where it is expected that the warnings of the IDS were sent to the ACS, so the network operator could take care of the threat, it would be potentially easy to perform DoS attacks on the ACS by attacking those CPEs.

10.5.1 Snort

Snort is an open source network intrusion prevention system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching, and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and more\cite{38}.

A version for OpenWrt is available in the official repositories. Although the limitation about employing this technology in embedded devices is still there, limitation of memory and processor, the capability of customization of Snort, gives to the system a window on using IDS for very specific purposes, that may not overload the CPE.

10.6 Using CWMP for security purposes

One of the major problems of actual routers and DSL modems, is the lack of updated firmwares, this is one of the strong points of the usage of CWMP. With CWMP it is guaranteed that the CPEs will work using the best firmware available, and security

\footnote{Generated warnings to none real threats.}
updates and checks can be done regularly for keeping the integrity of the device. Regular updates can be used as well, to prevent attacks that have been identified in other CPEs. For example, if a botnet has been detected to had infected some CPEs from the Network of the ISP, with the help of an IDS and specific rules created analysing the behaviour of the botnet, CWMP could be used to take precautions on discovered ongoing attacks, plus, quick actions and updates would prevent other DSL modems to get infected. Certainly this subject would require a deep study and is not the purpose of this thesis.

Another way to integrate the CWMP to the security using an IDS, is when used to prevent jamming and optimizing the use of channels between neighbours CPEs. If two CPEs are interfering with each other using the same channel, then warnings could be sent to the ACS, who could take actions against this.
Chapter 11

Conclusion

Many CPE management systems have been created and are being used nowadays, nonetheless all these systems assume a unidirectional management and configuration of the device. For the purpose of the project presented by Nokia Siemens Networks, the need of a split of the management system was imperative due to the characteristics of the off-load network architecture and the need of cooperation and relation to the customer.

Based on the research done, a new model of management system has been created. The model has been developed using a primary theoretical approach to the division and share of access rights and to the configuration tools of a CPE device, continued by the customization of different tools available in the selected firmware, OpenWrt. This model includes as well a detailed explanation of the different customer roles, Private and Public User, and who represent the two different parties in the administration of the device, the Network Operator as a remote manager of the CPE, and the Private User as local manager. Additionally, a specific split of the rights have been elaborated to give access and authority to the Private User over the management of certain areas of the device at the same time that a remote management is done by the Network Operator.

It has been chosen two different mechanisms of management for the CPE, the web interface for private management and TR-069 for the remote management of the Network Operator. A customized web interface has been created based in the LUCI environment and a series of detailed changes have been established for it. After the model was developed and the modifications concreted, the system has been taken to a real implementation and tested using two different CPEs, D-LINK DIR825 and LINKSYS 54WRTGL, obtaining positive results in both performance and integration between the management systems.

Finally, an analysis of the equipment’s security and privacy and confidentiality of the users, in a frame of a shared network have been done and how to enhance it in OpenWrt. It has been shown that a CPE using OpenWrt has the capability to enable state of the art security mechanisms in the devices, using WPA2, SSL,
TLS, VLAN, HTTPS, 802.11w and iptables. Depending on the CPE resources, security can be enhanced, with the integration of IDPS technologies. Another of its strengths is the faculty on upgrading/updating the firmware, using CWMP, which will be very helpful in securing the device against future coming attacks, and the evolution of the firmware to keep up to modern technologies.
Bibliography


Appendix A

Tables of The Privacy Policy
Appendix B

Modifications in Folder
/model/cbi/admin_network/
Appendix C

Modifications in Folder
/model/cbi/luci_fw/
Appendix D

Source Code of the New Web Interface