

02.02.07

# Deliverable DJ5.1.5,1: Inter-NREN Roaming Infrastructure and Service Support Cookbook - First Edition



## Deliverable DJ5.1.5,1

Contractual Date: 31/01/2007  
Actual Date: 02/02/2007  
Contract Number: 511082  
Instrument type: Integrated Infrastructure Initiative (I3)  
Activity: JRA5  
Work Item: 1 - Roaming  
Nature of Deliverable: R (Report)  
Dissemination Level: PU (Public)  
Lead Partner: SURFnet  
Document Code: GN2-06-258v8

**Authors:** K. Wierenga (SURFnet, main author), P. Dekkers (SURFnet), L. Guido (FCCN), T. Kersting (DFN), S. Papageorgiou (NTUA/GRNET), Janos Mohacsi (NIIF/HUNGARNET), R. Papez (ARNES), M. Milinovic (CARNet/Srce), D. Penezic (CARNet/Srce), J. Rauschenbach (DFN), S. Winter (RESTENA), T. Wolniewicz (Nicolaus Copernicus University, Torun), JRA5 group

## Abstract

This Deliverable provides a collection of guidelines and installation instructions ("cookbook") for implementing the eduroam service.

# Table of Contents

0	Executive Summary	v
1	Introduction	1
2	eduroam in a Nutshell	2
2.1	General overview	2
2.2	Elements of the eduroam infrastructure	3
2.2.1	Confederation top-level RADIUS Server (TLR)	3
2.2.2	Federation TLR	4
2.2.3	Institutional RADIUS	4
2.2.4	Suplicants	4
2.2.5	Access Points	4
2.2.6	Switches	5
3	Example of eduroam Setup	6
3.1	Reference Federation RADIUS Proxy	6
3.1.1	Common configuration	6
3.1.2	Client definition	7
3.1.3	Handling unknown realms under the own TLD	8
3.1.4	Handling empty realms	8
3.1.5	Proxying to an organisation with one RADIUS Server	8
3.1.6	Proxying to multiple RADIUS servers	9
3.2	Reference Campus Setup	11
3.2.1	Introduction	11
3.2.2	Configuring the Ethernet switch for eduroam	13
3.2.3	Setting up the RADIUS server	14
3.2.4	Configuring the Access Point for eduroam	19
3.2.5	Suppliant	22
4	Conclusions	29
5	References	30

6	Acronyms	31
Appendix A	RADIUS servers	33
A.1	Radiator institutional server	34
A.2	FreeRADIUS institutional server	35
A.2.1	Setting up FreeRADIUS	35
A.2.2	Defining clients - Access Points and RADIUS servers	36
A.2.3	Configure realm handling and proxying	37
A.2.4	Users authentication and realm handling	38
A.2.5	Setting up accounting in the SQL database	39
A.2.6	The master RADIUS configuration	41
A.2.7	Logging the client IP address (Optionally!)	44
A.2.8	More information	44
A.3	Navis institutional and national server	45
A.3.1	Institutional Navis RADIUS server (ver. 4.5.8)	45
A.3.2	National Navis RADIUS server (ver. 4.5.8)	50
A.4	VitalAAA Institutional and national server	54
A.4.1	Institutional VitalAAA Server (ver. 5.0.10)	54
A.4.2	National VitalAAA Server (ver. 5.0.10)	61
A.5	Microsoft Internet Authentication Service server as institutional server	67
A.5.1	Installing IAS	67
A.5.2	Configuring IAS to act as a university RADIUS server in the eduroam hierarchy	71
A.5.3	Configuring remote RADIUS servers	76
A.5.4	Configuring Domain Users to be able to use eduroam with their credentials to Windows Domain	78
A.5.5	Configuration of Authentication methods	79
A.5.6	Troubleshooting	81
A.5.7	References	82
Appendix B	Access Points	83
B.1	Cisco Aironet 1200 Series example setup	83
B.2	LANCOM L-54 Series Access Points	88
B.2.1	NTP setup (confederation requirement: reliable timing source)	88
B.2.2	Logging	89
B.2.3	Configuring the SSID	90
B.2.4	WPA Enterprise security	90
B.2.5	RADIUS accounting server (optional)	92

Appendix C	Suplicants	94
C.1	SecureW2	94
C.2	MacOS	100
C.3	WPA_Suppliant	102

## Table of Figures

Figure 2.1:	Layers of the eduroam RADIUS hierarchy	3
Figure 3.1:	Network Topology	12
Figure A.1:	Message flow in RADIUS server Identity Management System	34

## 0 Executive Summary

This document provides installation instructions for administrators to make an eduroam implementation as seamless as possible. The cookbook idea is to provide a general model first which is then amended by a variety of specific implementation examples. There are plenty of options to build an eduroam infrastructure, especially at the campus level. Devices and software solutions both of commercial vendors as well as open source products are available. This deliverable presents the various elements needed by a typical institution to participate in the eduroam network and provides guidelines to configure those. It is intended as a technical guideline "cookbook" to help administrators of institutions willing to join the eduroam network. It is not meant to describe eduroam and the underlying architecture in detail, these descriptions can be found in the deliverable DJ5.1.4 („Inter-NREN Roaming Architecture: Description and Development Items“).

The deliverable describes the main functional principle and introduces the components needed in an eduroam infrastructure in the general case. A reference campus set-up based on products frequently used in the research and education area is provided as an example with detailed configuration and installation instructions. However, this is not a recommendation which technical equipment to buy or to use. Other solutions might provide the same or at least the same level of functionality. An essential part of the document is therefore dedicated to the configuration of alternative components, for a better readability these instructions are provided as an appendix to the deliverable. It must be added that this again, is not an exhaustive list. The products described are used by one or more JRA5 participants ensuring a certain level of practical experience.

This is the first version of the "cookbook". A second version will follow and provide an option to integrate new developments both in the eduroam architecture and in the spectrum of products and solutions considered.

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

# 1 Introduction

The current eduroam architecture was described in more detail in the deliverable "Inter-NREN Roaming Architecture: Description and Development Items" [GN2DJ514]. It did not address the question how this generally depicted architecture including the components could be implemented in a real environment. This "cookbook" document addresses exactly this question. The purpose is not to promote any product, but to select popular solutions with a broader usage rate in the research and education area as examples to describe the installation procedures. The idea is to enable the campus administrator or the eduroam operator to adapt these configuration principles for the devices operated in the home environment.

After a very brief description of eduroam, chapter 2 provides an overview of the necessary, basic elements needed as well as their specific functions within the eduroam network, whereas the second section (chapter 3) shows the configuration of a typical example setup ranging from national RADIUS servers down to user supplicants. Configuration examples for multiple variations of the elements described are shown in the Appendices A to C, covering RADIUS servers, switches, access points and supplicants of different vendors or available as open source solutions.

The list of products for which guidelines are provided is not exhaustive and might also depend on the version of the software used. Changes in the context of technical progress are unavoidable, so that the deliverable has an inherent dynamic aspect and the reader should be aware of this. A second version of the "cookbook" is part of the project plan and will provide an option to reflect new developments and necessary corrections to the material provided in version 1 as well.

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

## 2 eduroam in a Nutshell

### 2.1 General overview

Please refer to deliverable DJ5.1.4 “Inter-NREN Roaming Architecture: Description and Development Items” for an in-depth description of eduroam and the underlying architecture.

Eduroam stands for EDUcation ROAMing, it offers users from participating academic institutions secure Internet access at any other eduroam-enabled institution. ~~participant~~The eduroam architecture making this possible is based on a number of technologies and agreements, which together provide the eduroam user experience: open your laptop and be online.

The crucial agreement laying the foundation of eduroam is that the authentication of a user is done at his home institution using their specific authentication method, whereas the authorisation decision allowing access to the network resources upon proper authentication is done by the owner of the visited network.

In order to transport the authentication request of a user from the visited institution to his home institution and the authentication response back, a hierarchical system of RADIUS servers is created. Typically every institution deploys a RADIUS server, which is connected to a local user database, this RADIUS server is connected to a central national RADIUS server which in turn is connected to a European (or global) RADIUS server. Because users are using usernames of the format ‘user@realm’, where realm is the institution’s DNS domain name, often of the form institution.tld (tld=country code top-level domain), the RADIUS servers can use this information to route the request to the appropriate next hop in the hierarchy, until the home institution is reached. An example of the RADIUS hierarchy is shown in Figure 2.1.

To transfer the user’s authentication information securely across the RADIUS-infrastructure to his home institution and to prevent other users from hijacking the connection after successful authentication, the access points or switches use the IEEE 802.1X standard which encompasses the use of EAP, the Extensible Authentication Protocol. Using the appropriate EAP-method either a secure tunnel will be established from the user’s computer to his home institution through which the actual authentication information

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

(username/password etc.) will be carried (EAP-TTLS or PEAP) or mutual authentication by public X.509 certificates, which is not vulnerable to eavesdropping, will be used (EAP-TLS).

After successful authentication by the home institution and authorisation by the visiting institution, this visited institution grants network access to the user, possibly by placing the user in a specific VLAN intended for guests.

In the next chapter the various elements of this architecture and their functions will be shortly described.

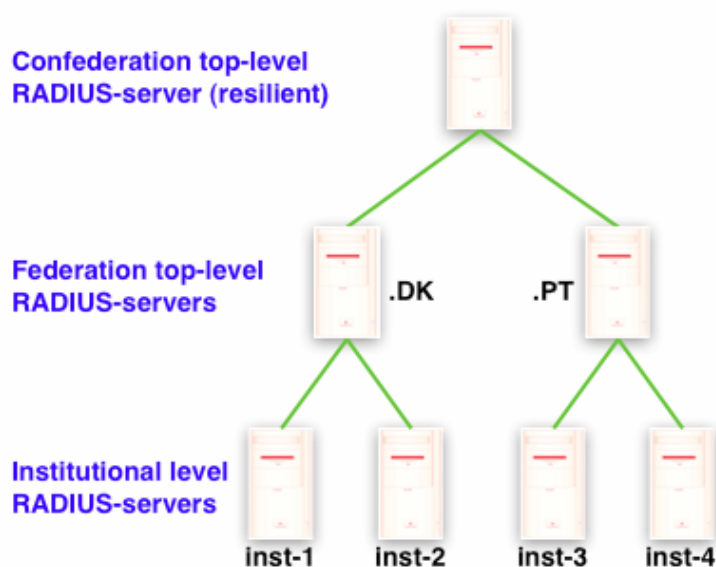


Figure 2.1: Layers of the eduroam RADIUS hierarchy

## 2.2 Elements of the eduroam infrastructure

### 2.2.1 Confederation top-level RADIUS Server (TLR)

The confederation top-level RADIUS Servers, at the time of writing located in the Netherlands and Denmark for the European confederation and Australia and Hongkong for the Asian and Pacific region each have a list of connected country domains (.nl, .dk, .au, .cn etc.) serving the appropriate NRENs. They accept requests for federation domains for which they are authoritative and subsequently forward them to the associated RADIUS



server for that federation (and transport the result of the authentication request back). Requests for federation domains they are not authoritative for are forwarded to the proper confederation TLR.

## 2.2.2 Federation TLR

A federation RADIUS server has a list of connected institutional servers and the associated realm. It receives requests from the confederation servers and institutions it is connected to and forwards them to the proper institution or in case of a request for a confederation destination to a confederation server.

## 2.2.3 Institutional RADIUS

The Institutional RADIUS server is responsible for authenticating its own users (at home or visiting another institution) by checking the credentials against a local identity management system and for forwarding requests from visiting users to the respective federation RADIUS server. Upon proper authentication of a user the local institutional RADIUS server may assign a VLAN to the user.

Note that the institutional RADIUS server is the most complex of all, whereas the other RADIUS servers merely proxy requests, the institutional server also needs to handle the requests, and therefore needs to be able to terminate EAP requests and perform identity management system lookups.

The Identity Management System contains the information of the end users, for instance usernames and passwords. They must be kept up-to-date by the responsible institution.

## 2.2.4 Supplicants

A supplicant is a piece of software, often built into the Operating System but as well available as a separate program, that uses the 802.1X protocol to send authentication request information using EAP.

## 2.2.5 Access Points

Access Points need to be 802.1X capable and able to forward access requests coming from a supplicant to the institutional RADIUS server, to give network access upon proper authentication and to possibly assign users to specific VLANs based on information received from the RADIUS server. Furthermore Access Points exchange keying material (initialization vectors, public and session keys, etc.) with client systems to prevent session hijacking.

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

## 2.2.6 Switches

Switches need to be able to forward access requests coming from a supplicant to the institutional RADIUS server, to grant network access upon proper authentication and to possibly assign users to specific VLANs based on information received from the RADIUS server.

## 3 Example of eduroam Setup

The following sections provide a sample setup of the eduroam network from top to bottom. Initially, the configuration for a national RADIUS Server is presented, followed by the various components needed at an institutional level right down to the supplicants of the end-users.

### 3.1 Reference Federation RADIUS Proxy

This chapter covers the configuration of the Federation RADIUS Proxy server assuming that the Radiator (<http://www.open.com.au/radiator/>) RADIUS server has been installed and basic configuration performed.

Examples in this chapter can be used as a repository of configuration snippets for building complex proxy servers, or when used as is, for simple proxy relations between one organisation and two top-level servers.

Radiator expects the configuration to be in file `/etc/radiator/radius.cfg`

#### 3.1.1 Common configuration

```
LogDir          /var/log/radiator
DbDir           /usr/share/radiator
```

LogDir defines the directory in which start-up logs and PID file reside, DbDir defines the path to Radiator's data files such as dictionaries.

Trace 3 logs will be sent to the system syslog, and Trace 4 logs will be stored in the directory `/var/log/arch/radiator/`

Here the location of logfiles (referred in the configuration as %L) and databases (%D) can be defined as well as the amount of log output is given (with Trace).

The log files will be split on a daily basis.

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```

<Log SYSLOG>
    Facility    local7
    LogIdent    log-syslog
    Trace       3
</Log>

<Log FILE>
    Filename    /var/log/arch/radiator/radiator.%Y_%m_%d.log
    LogIdent    log-file
    Trace       4
</Log>

```

SNMP allows remote monitoring of activity on a RADIUS server with tools such as RADAR from OSC (<http://www.open.com.au/radar/index.html>) or drawing simple graphs of activity by rgraph from CESNET (<http://www.eduroam.cz/rgraph/>).

```

<SNMPAgent>
    ROCommunity xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    Managers    localhost 127.0.0.1
</SNMPAgent>

```

The ports on which Radiator will listen for Authentication and Accounting requests need to be defined. The port numbers 1812 and 1813 were assigned to the RADIUS protocol by IANA, unfortunately some RADIUS servers (such as CiscoACS 3.x) are still using old the numbers 1645, 1646. For this reason it is suggested to use both numbers.

```

AuthPort      1645,1812
AcctPort      1646,1813

```

### 3.1.2 Client definition

In the client section all possible peers (the 'institutional' and the confederation RADIUS servers) have to be listed and a secret has to be assigned to them. As this secret is the only thing, which protects the communication between the RADIUS servers from eavesdropping, it has to be cryptographically strong and well protected.

```

<Client localhost>
    Secret      mysecret
    DupInterval 0
</Client>

<Client radius.orgA.tld>
    Secret      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    Identifier  radius.orgA.tld
</Client>

<Client etlrl.eduroam.org>
    Secret      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
        Identifier  etlr1.eduroam.org
</Client>

<Client etlr2.eduroam.org>
    Secret          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    Identifier      etlr2.eduroam.org
</Client>
```

It is necessary to mark each host by a unique identifier, it is later used to prevent loops in the hierarchy. etlr1.eduroam.org and etlr2.eduroam.org are the European top-level RADIUS servers.

### 3.1.3 Handling unknown realms under the own TLD

Known realms under the TLD for which the RADIUS server is responsible are explicitly listed, so if the RADIUS server cannot match an Access-Request under his TLD to an entry in the Handler section, the realm is considered unknown. Any Access-Request with unknown realm under the TLD for which this RADIUS server is responsible is rejected with an Access-Reject message. Possible Accounting requests are acknowledged and then dropped.

```
<Handler Realm=/.*\.tld$/i>
    AccountingHandled
</Handler>
```

### 3.1.4 Handling empty realms

Theoretically empty realms should not reach a National RADIUS Proxy, but if they do, this code will prevent the National RADIUS Proxy from sending these empty realms to the international top-level proxy servers.

```
<Handler Realm=/^$/>
    AccountingHandled
</Handler>
```

### 3.1.5 Proxying to an organisation with one RADIUS Server

In the case in which an organisation has only one server, configuration is relatively easy:

```
<Handler Realm=/^orgA.tld$/i,
    Client-Identifier=/^(?!radius.orgA.tld)/>
  <AuthBy RADIUS>
    RetryTimeout          3
    Retries                1
    FailureBackoffTime    0
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
UseExtendedIds

<Host radius.orgA.tld>
  AuthPort      1812
  AcctPort      1813
  Secret        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
</Host>
</AuthBy>
</Handler>
```

Realms should be pattern matched case-insensitive, because users sometimes type their realms using upper case letters.

The configuration directive “Client-Identifier=/<sup>^</sup>(?!radius.orgA.tld\$)/” blocks RADIUS packets from radius.orgA.tld with realm orgA.tld from entering this Handler.

If such a packet will be received it will be rejected by the handler matching realm=/.\*\.tld\$/i. This prevents loops between the national proxy and the server of orgA.tld.

Servers to proxy should never be marked as dead, because in doing so, radiator will not try to communicate with them until BackoffTime expires, even if that server is already up again.

### 3.1.6 Proxying to multiple RADIUS servers

Top-level RADIUS servers (confederation or federation) are duplicated for better reliability of the eduroam service as required in the policy document [GN2DJ513,2], also some organisations have multiple RADIUS servers for the same reason. To use multiple servers in eduroam, a special setup has to be used. Standard detection of dead hosts based on timeout will not work here, timeouts might be caused by the infrastructure behind the direct peer of RADIUS server – in this case the server cannot assume that his peer is dead. But in the case it is really dead, a switch to the backup server must be done. Unfortunately the RADIUS protocol does not provide a clear way to do that.

The problem can be solved by using a special configuration developed at CESNET and described in the article “Dead-realm marking feature for Radiator RADIUS servers” (<http://www.eduroam.cz/dead-realm/docs/dead-realm.html>).

The idea behind the dead-realm marking is quite simple. Instead of marking the server dead for all realms (as dead-host marking is doing) just the particular realm on the host is marked dead and requests are routed to another configured host.

The variable dr\_timeout controls for how long a realm will be marked dead on a particular host. 3600 seconds is the suggested value. Much lower values will cause the system to check the dead primary server too often. Higher values can be considered if the backup server is equally powerful as the primary server.

```
DefineFormattedGlobalVar      dr_timeout 3600
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

Following the two `AuthBy` sections are definitions of the servers, which will be used as a pool for destinations defined later. `NoReplyHook` and `ReplyHook` are necessary for the dead-realm marking feature, a Perl implementation is available online at <http://www.eduroam.cz/dead-realm/>.

`NoReplyHook` is called when no reply is received from the RADIUS server after 2 attempts within 6 seconds. In this case the processed packet is ignored, the realm is marked as dead on this server and when a subsequent packet with the same realm is received it will be forwarded to the other defined server.

`ReplyHook` is called when a response from the server is received, in case the realm is marked dead the realm will be unmarked dead.

```
<AuthBy RADIUS>
  Identifier                etlrl.eduroam.org

  RetryTimeout              3
  Retries                   1
  FailureBackoffTime       0

  UseExtendedIds

  <Host etlrl.eduroam.org>
    AuthPort                1812
    AcctPort                1813
    Secret                  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
  </Host>

  NoReplyHook file: "/etc/radiator/dr_no-reply-hook.pl"
  ReplyHook file: "/etc/radiator/dr_reply-hook.pl"
</AuthBy>
```

```
<AuthBy RADIUS>
  Identifier                etlr2.eduroam.org

  RetryTimeout              3
  Retries                   1
  FailureBackoffTime       0

  UseExtendedIds

  <Host etlr2.eduroam.org>
    AuthPort                1812
    AcctPort                1813
    Secret                  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
  </Host>

  NoReplyHook file: "/etc/radiator/dr_no-reply-hook.pl"
  ReplyHook file: "/etc/radiator/dr_reply-hook.pl"
</AuthBy>
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

The following directive (DefineFormattedGlobalVar) defines the variable `dr_TOPLEVEL_server_list` and assigns a list of servers responsible for the handler marked as `TOPLEVEL` to this variable.

```
DefineFormattedGlobalVar dr_TOPLEVEL_server_list \  
    etlr1.eduroam.org,etlr2.eduroam.org
```

The handler with the filter `Realm=/^.+$/` will receive any request which will not be caught by more specific filters before (`Realm=/^orgA.tld$/i`, `Realm=/.*\.tld$/`, `Realm=/^$/`). In this example it will be all requests, which belong to other TLD's than the ones defined on this server – requests, which have to be forwarded to top-level servers.

`RequestHook` will use the value of `Identifier (= TOPLEVEL)` defined in this handler to search the variable `dr_TOPLEVEL_server_list` holding the list of servers. It will forward the request to a server not marked dead or if all are marked dead the one with the oldest dead-realm mark.

```
<Handler Realm=/^.+$/>  
    Client-Identifier=/^(?!etlr1.eduroam.org$)/>  
    Client-Identifier=/^(?!etlr2.eduroam.org$)/>  
  
    Identifier TOPLEVEL  
  
    <AuthBy INTERNAL>  
        RequestHook file: "/etc/radiator/dr_choose-server.pl"  
    </AuthBy>  
</Handler>
```

## 3.2 Reference Campus Setup

### 3.2.1 Introduction

Campus networks vary widely, both in topology and used equipment, software etc. In order to assist a campus administrator in setting up eduroam on their campus we present the implementation of a typical setup. It is hoped that this will allow even users of different topologies and/or equipment to understand the necessary steps to take. Furthermore, in the appendices the same setup will be worked out for a number of other common types of equipment and software. Lastly, we are planning to provide these and future example configurations at the website <http://www.eduroam.org>.

For the reference network we use a typical set of network equipment consisting of:

- a Cisco Catalyst 3550 (or similar) switch,
- a Cisco Aironet AP-1200 Access Point,
- a laptop with Windows XP and

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8



- a RADIUS server

The network topology is as follows:

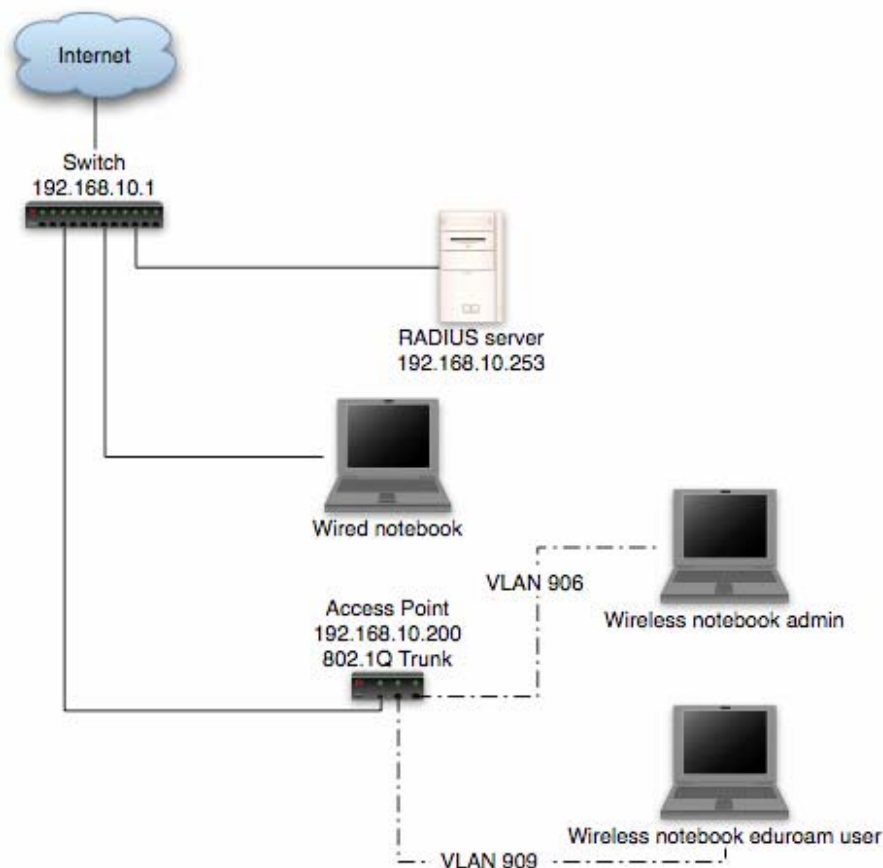


Figure 3.1: Network Topology

In this setup wireless users are separated in different VLANs: VLAN906 for ‘administrative’ users and VLAN909 for normal eduroam users. The next table describes each VLAN used in this document:

VLAN ID	Propose
901	VLAN for internet access – access to core routers
902	The Administrative VLAN of the hotspot (AP’s; RADIUS; etc.)
903	VLAN with open SSID for giving information about the institute
906	VLAN reserved for administrative users

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

909	VLAN reserved for 'normal' eduroam users
-----	--

Table 3.1: VLAN description

The next table describes the IP configuration for the router sub-interfaces and what networks are configured for each VLAN:

Interface	802.1Q Tag	Interface IP Address	DHCP Pool	What is accessible in this network
FE0.901	901	Some public IP address	N/A	
FE0.902	902	192.168.10.254	N/A	AP's; RADIUS Server
FE0.906	906	10.9.6.254	10.9.6.0/24	administrators
FE0.909	909	10.9.9.254	10.9.9.0/24	eduroam clients

Table 3.2: Router Configuration

### 3.2.2 Configuring the Ethernet switch for eduroam

In order to gain access to the Internet the configuration of the Ethernet switch needs to be changed. You have to create a VLAN in which the Access Points will be placed, and provide it with the correct IP-address and gateway information. This can be done with the commands described below.

The next table describes the VLAN associated with each Port of the switch and what equipment will be connected to that specific port

Port	VLAN configuration (T – Tagged; U – Untagged)	What is connected to it
1	U (902)	RADIUS Server
2-47	U (902) T (909)	Access Points
48	U (901) T (902; 909)	Central Ethernet Switch

Table 3.3: Ethernet Switch Configuration

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

First we will configure the port where the RADIUS Server will be connected and put it on the Administrative VLAN:

```
switch(config)#interface fastethernet0/1
switch(config-if)#description RADIUS Server
switch(config-if)#switchport mode access
switch(config-if)#switchport access vlan 902
switch(config-if)#spanning-tree portfast
```

Then configure all switch-ports that will connect Access Points for the VLAN's that users and Access Points can have access to (in trunk mode) – at least the administrative VLAN and the VLAN where authenticated users will be placed:

```
switch(config)#interface range fastethernet0/2 - 47
switch(config-if)#description eduroam Access Points
switch(config-if)#switchport trunk encapsulation dot1q
switch(config-if)#switchport trunk native vlan 902
switch(config-if)#switchport trunk allowed vlan 902, 909
switch(config-if)#switchport mode trunk
```

The uplink can be defined with:

```
switch(config)#interface fastethernet0/48
switch(config-if)#switchport trunk encapsulation dot1q
switch(config-if)#switchport trunk native vlan 901
switch(config-if)#switchport trunk allowed vlan 901, 902, 909
switch(config-if)#switchport mode trunk
```

### 3.2.3 Setting up the RADIUS server

Now the RADIUS server will be configured.

Because of the EAP authentication within RADIUS, a (small) PKI is required. If there is no PKI available one could create the required key and certificate with for instance TinyCA. TinyCA (<http://tinyca.sm-zone.net/>) is a simple graphical interface on top of OpenSSL. It is possible to use OpenSSL directly (but instructions to do so are outside the scope of this document).

There is also a bootable CD available based on Knoppix that runs TinyCA, the roCA (read-only CA) that can be found at <http://www.intrusion-lab.net/roca/>.

Depending on the EAP-type used, client certificates may also be needed.

Within the Radiator distribution there are also simple scripts available to create certificates for testing purposes.

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

The Radiator RADIUS server needs the configuration file `/etc/radiator/radius.cfg`.

This configuration file can be created with the editor of choice, e.g.

```
vi /etc/radiator/radius.cfg  
or  
pico /etc/radiator/radius.cfg
```

In the following examples there are two kinds of EAP which are configured at 'institution'. Firstly, EAP-TLS based on client-certificates, secondly, EAP-TTLS and EAP-PEAP that do not require client certificates but the traditional mechanism username/password authentication instead.

### 3.2.3.1 Clients

RADIUS is based on a client-server model. The NAS-devices (Access Points, switches etc.) forward credentials to a RADIUS server, i.e. act as a client, and therefore need to be defined on the RADIUS server. Other RADIUS servers can act as a client as well, so every kind of RADIUS-request can be forwarded to another server.

The clients are configured within Radiator using the `<Client>`-clause.

```
<Client 192.168.10.200>  
Secret 6.6obaFkm&RNs666  
    Identifier ACCESSPOINT1  
    IdenticalClients 192.168.10.201  
</Client>
```

In this example there is a client definition for 192.168.10.200, an Access-Point. The secret is a series of (at best 16) characters that are used to encrypt the credentials sent in the RADIUS-request.

It is of course recommended to create a secret that cannot be easily guessed since otherwise the RADIUS-message can be decrypted. This is not a big problem with EAP-authentication using 802.1X since the credentials are also transmitted over a SSL-encrypted tunnel between the client and the final authentication server, but especially with regular credentials like the ones used with Web-based redirection authentication this is sensitive information that might be captured, therefore a reasonably complex secret and an SSL tunnel is recommended.

The Identifier in the Client-definition can be used further on in the Radiator-configuration to filter on a specific request.

If there is more than one Client to use this same secret and identifier definition, the `IdenticalClients` statement can be used. If there are really many clients with different IP-addresses, there is also the possibility for a "catch-all" client. This is the default client that is used after all other client definitions didn't match. Define this client as:

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
<Client DEFAULT>
```

If this kind of configuration is used, it is worth filtering with firewall-rules on RADIUS packets. There are only a few places from where a RADIUS-request should come: the management VLAN with the NAS-devices (switches and access-points), and the RADIUS infrastructure where unknown requests can be sent on.

### 3.2.3.2 Realms and VLAN assignment

The processing of authentication and accounting requests is done by linear processing of the present `<Realm>`- or `<Handler>`-clauses in the Radiator configuration file. Handler-clauses are more potent than Realm clauses in terms of filtering anything besides realms and are therefore the preferred method. A *realm* is the part behind a username to indicate the origin of a user. With RADIUS, the username is usually separated from the realm with a “@” so the complete username looks like a regular e-mail address.

A `<Handler>`-clause is terminated with a `</Handler>`.

Within a Handler many mechanisms can be configured that define what to do with the RADIUS request.

#### *PROXY example*

The simplest Handler for proxy-ing the request to another server uses the “AuthBy RADIUS” definition within this Handler.

In this example a proxy-configuration is shown. First we have a Handler that matches on any request, as long as it does not come from the client with the identifier “Proxy-Identifier”. This is to prevent a proxy loop. When a request comes from a proxy-server, it should never be forwarded back to that proxy-server.

Another important part is the hostname to which the request should be forwarded. Multiple hostnames can be defined here for redundancy reasons: if the first host does not respond within 3 seconds, the second one is tried instead. The UDP ports to which the RADIUS-request should be forwarded, can be defined in this “AuthBy RADIUS” clause as well.

```
<Handler Client-Identifier=/^(?!Proxy-Identifier$)/>
  <AuthBy RADIUS>
    Host 192.87.36.3
    Secret super_secret!
    AuthPort 1812
    AcctPort 1813
    StripFromReply Tunnel-Type, Tunnel-Medium-Type, \
      Tunnel-Private-Group-ID
    AddToReply Tunnel-Type=1:VLAN, Tunnel-Medium-Type=1:Ether_802, \
      Tunnel-Private-Group-ID=1:909
  </AuthBy>
</Handler>
```

For a “Host” both the IP-address and FQDN can be used. The choice is more or less a personal preference of the RADIUS administrator, but one must be aware that the hostnames are only looked up once at the Radiator (re)start. If the lookup fails, the Host cannot be used until the next restart. This can represent a problem at a power outage, where for instance the DNS server is not yet available when Radiator already is.

While by using hostnames one benefits from the administrative ease when an IP-address is changed, in such case it is still needed to restart the RADIUS server.

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

The last part in this <AuthBy RADIUS>-definition shows the addition of RADIUS-attributes to the RADIUS-response. These attributes can be used to define a VLAN that will be assigned to users that are authenticated using this Handler. With StripFromReply the attributes that came from the proxy-server are stripped first to prevent malicious VLAN-assignments, afterwards the attributes are added with the proper values for the local network design. In this case, VLAN 909 is used for guests.

### Secure authentication with EAP-TLS

If it is an option to issue end user certificates the EAP-TLS mechanism can be used.

In this example the AuthBy-definition is outside the Handler, and is referred to using the Identifier. (This is useful if the AuthBy-definition is reused in another Handler for instance.)

```
<AuthBy FILE>
  Identifier ID4-TLS
  Filename %D/TLS-users
  EAPType TLS
  EAPTLS_CAFfile %D/cert/institution-ca-chain.pem
  EAPTLS_CertificateFile %D/cert/radius-server-cert.pem
  EAPTLS_CertificateType PEM
  EAPTLS_PrivateKeyFile %D/cert/radius-server-key.pem
  EAPTLS_PrivateKeyPassword (the secret that secures the private-key)
  EAPTLS_MaxFragmentSize 1024
  AutoMPPEKeys
  SSLeayTrace 1
  StripFromReply Tunnel-Type,Tunnel-Medium-Type,Tunnel-Private-Group-ID
  AddToReply Tunnel-Type=1:VLAN,Tunnel-Medium-Type=1:Ether_802,Tunnel-Private-Group-
ID=1:909,User-Name=%u
</AuthBy>
```

In this AuthBy-clause there is an EAPTLS file defined, in which every employee is listed. In this way, we control the users that can access the infrastructure using EAP-TLS.

The next definitions determine what to do with the EAP-request. First the “EAPType TLS” limits the use of this AuthBy-definition for TLS-only. Here, we do not want regular password authentication, just certificates. Next, the certificate files are configured and the secret that secures the private-key file can be provided. If there is no secret for the private key, this can be omitted.

The next part defines in what size chunks the EAP-messages should be fragmented. Since some parts of the EAP-TLS challenge are too big to fit in a RADIUS request the packets should be fragmented.

The MPPE-keys (Microsoft Point to Point Encryption, the protocol for encrypting the data across links) portion is important for wireless access. With 802.1X, encryption is done at the edge of the network, between the Access-Point and the client. To provide this secure encryption, a unique key is created and encrypted using the MPPE-keys that are derived from the SSL-challenge. This can both be done at the Access-Point and the Client end so that there is no need to transfer the WEP-key in plain text over the air. This, and the fact that the key can be rotated within a period defined by either the Access-Point or the RADIUS server provides 802.1x users with a good level of security.

The last part of the AuthBy-definition shows again how to assign a proper VLAN.

```
<Handler Realm=instituion.cc, EAP-Message=/.+/>
  AuthBy ID4-TLS
</Handler>
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

The Handler above shows the referral to the AuthBy-definition and some filtering mechanisms to filter out the proper requests. If more things need to be filtered on, they can be added to this handler, as follows:

```
NAS-Port-Type=/^(Wireless-IEEE-802-11|Ethernet)$/
```

In this way, only requests with the proper NAS-Port-Types are allowed.

For Accounting purposes, a new handler should be defined in this case, that filters on

```
"Request-Type=Accounting-Request"
```

since the request does not match the Handler that filters on the EAP-Message.

### *EAP-TTLS or EAP-PEAP*

When issuing end user certificates is not an option, the EAP-mechanisms PEAP and TTLS can be used.

These two mechanisms look the same, they both set up an SSL tunnel on which the credentials can be transported. In the case of PEAP this is another EAP-method in an encrypted tunnel, while when using TTLS the inner payload consists of RADIUS or Diameter attributes, which is more flexible because this enables the use of PAP-authentication based on a plain-text username and password, transported over a SSL tunnel.

The choice whether to use PEAP or TTLS depends mostly on the already existing identity management backend on campus. If an ActiveDirectory infrastructure is present, using PEAP has security-related advantages, since at no point in the authentication chain plain text passwords are visible. PEAP can also be used with any other backend as long as the backend provides NT-style password hashes (which can be mapped into FreeRADIUS' NT-Password attribute).

In all other cases, the use of PAP is encouraged, since it can be used with virtually every backend.

When the backend contains plaintext usernames and passwords (for guests for instance) both PEAP and TTLS can be used simultaneously with this backend.

Instead of a flat file a more flexible backend for user accounts is a database like MySQL, or LDAP.

```
<Handler TunnelledByPEAP=1, Realm=tunnelled.institution.cc>
  <AuthBy FILE>
    Filename %D/peap-users
    EAPType MSCHAP-V2
  </AuthBy>
</Handler>

<Handler TunnelledByTTLS=1, Realm=tunneled.institution.cc>
  <AuthBy FILE>
    Filename %D/ttls-users
  </AuthBy>
</Handler>
```

In these Handlers with the filtering option "TunneledByPEAP" and "TunnelledByTTLS" define that the tunneled authentication (with the username and password in it) is handled here.

The "outer authentication", where the SSL tunnel is set up, looks like the TLS handler.

```
<Handler Realm=group_1>
  <AuthBy FILE>
    # the %D/users file can be empty, it's there for normal PAP
    # authentication. This can however be used for the WEB captive
    # portals.
    Filename %D/users
    EAPType TTLS, PEAP
    EAPTLS_CAFfile %D/root.pem
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
EAPTLS_CertificateFile %D/server.pem
EAPTLS_CertificateType PEM
EAPTLS_PrivateKeyFile %D/server.pem
EAPTLS_PrivateKeyPassword serverkey
EAPTLS_MaxFragmentSize 1024
EAPAnonymous anonymous@group1
AutoMPPEKeys
</AuthBy>
</Handler>
```

### 3.2.4 Configuring the Access Point for eduroam

Cisco AP 1200 Series (802.11g Radio)

The configuration examples used in this document were tested and made on a Cisco Series 1200 with a 802.11g Radio Module and with the following Cisco software:

**IOS Version:**

Cisco IOS Software, C1200 Software (C1200-K9W7-M), Version 12.3(8)JA2, RELEASE SOFTWARE (fcl)

**Bootloader:**

C1200 Boot Loader (C1200-BOOT-M) Version 12.2(8)JA, EARLY DEPLOYMENT RELEASE SOFTWARE (fcl)

#### 3.2.4.1 The multiple (dynamic) VLAN assignment

If multiple VLANs are configured on the Cisco AP, it is mandatory to associate each SSID to at least one VLAN otherwise the Access Point will not activate the SSID's. It is possible however, to put different users who are connected to the same SSID e.g. eduroam - on different VLANs based for instance on the user profile. To activate this feature it is necessary to enter 'aaa authorization network default group **radiusgroup**' in the Access Point's configuration. The AP will then give priority to the VLANs returned by RADIUS over the ones statically associated with the SSID. This feature is often called dynamic VLAN assignment.

Cisco's Access Points require that 2 virtual interfaces (a radio and an Ethernet port interface) are configured for each VLAN. If e.g. 4 VLANs are to be used for eduroam users (for students, admin staff, teachers and visiting eduroam users from other institutions for example) then it is necessary to define one Dot11Radio0.vlanID, and one FastEthernet0.vlanID, and ensure that both have the same encapsulation dot1Q **vlanID** and the same bridge-group for each VLAN.

Two commands that are also needed are: 'encryption vlan **vlanID** mode ciphers aes-ccm tkip wep128' and 'broadcast-key vlan **vlanID** change 600 membership-termination capability-change' otherwise the access point will not change the user to the received VLAN.

In the example configuration below there will be no dynamic VLAN assignment.

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8



### 3.2.4.2 The eduroam SSID – logical setup

In the following example the eduroam SSID is configured to accept all cipher modes used today on eduroam enabled institutions: 802.1X/WEP; WPA/TKIP; WPA2/AES.

### 3.2.4.3 Cisco Aironet 1200 Series basic configuration

First some basic eduroam configuration parameters for the Cisco Aironet Series 1200 (similar for other Cisco Access Points Series) are presented.

#### Setting the Name and IP address

First, an IP address on the BVI interface (the IP address that this Access Point will have for accessing resources like the RADIUS server) needs to be configured. Also a unique name for this Access Point (ap1200) will be configured.

```
ap#configure terminal
ap1200(config)#hostname ap1200
ap1200(config)#interface BVI 1
ap1200(config-if)# ip address 192.168.10.200 255.255.255.0
```

#### RADIUS/AAA section

In the authentication, authorisation and accounting configuration parameters (AAA) at least one group needs to be defined (radsrv), that will be assigned later for the several AAA operations. More groups can be defined if needed for various purposes – one for authentication, another for accounting, etc. In this example the RADIUS server has the IP address 192.168.10.253.

```
ap1200(config)#aaa new-model
ap1200(config)#radius-server host 192.168.10.253 auth-port 1812 acct-port 1813 key <secret>
ap1200(config)#aaa group server radius radsrv
ap1200(config-sg-radius)#server 192.168.10.253 auth-port 1812 acct-port 1813
ap1200(config-sg-radius)#!
ap1200(config-sg-radius)#aaa authentication login eap_methods group radsrv
ap1200(config)#aaa authorization network default group radsrv
ap1200(config)#aaa accounting send stop-record authentication failure
ap1200(config)#aaa accounting session-duration ntp-adjusted
ap1200(config)#aaa accounting update newinfo periodic 15
ap1200(config)#aaa accounting network default start-stop group radsrv
ap1200(config)#aaa accounting network acct_methods start-stop group radsrv
```

#### Configuring the SSID's

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

For each SSID one dot11 ssid <SSID NAME> must be configured. In this section the default VLAN for the SSID will be configured as well as the authentication framework, the accounting and, if desired, the SSID to be broadcast (guest-mode).

```
ap1200(config)#dot11 ssid eduroam
ap1200(config-ssid)#vlan 909
ap1200(config-ssid)#authentication open eap eap_methods
ap1200(config-ssid)#authentication network-eap eap_methods
ap1200(config-ssid)#authentication key-management wpa optional
ap1200(config-ssid)#accounting acct_methods
ap1200(config-ssid)#guest-mode
```

More SSID's can be configured. An open SSID for giving information about the institution and/or how to connect to the eduroam SSID

```
ap1200(config)#dot11 ssid guest
ap1200(config-ssid)#vlan 903
ap1200(config-ssid)#authentication open
ap1200(config-ssid)#accounting acct_methods
```

## The Radio Interface

Now the configured SSID's will be mapped to the radio interface and it will be specified what ciphers will be used/allowed on each VLAN. If dynamic VLANs are planned the ciphers for those VLANs must also be configured even if there's no direct mapping on any SSID (this example shows the usage of the VLANs 906 and 909 for eduroam users)

```
ap1200(config)#interface Dot11Radio 0
ap1200(config-if)# encryption vlan 906 mode ciphers aes-ccm tkip wep128
ap1200(config-if)# encryption vlan 909 mode ciphers aes-ccm tkip wep128
ap1200(config-if)#ssid eduroam
```

To bind extra SSID's the previous command, for each SSID to be bound, needs to be repeated.

The following command sets the maximum time (e.g. 300 seconds, which is recommended) for rekeying/reauthentication:

```
dot1x reauth-period 300
```

## VLAN interfaces

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

For each VLAN to be used for wireless clients 2 virtual interfaces need to be defined: one on “the air” (DotRadio) and another on the “wire” (FastEthernet) then they need to be bridged together with the same bridge group. These VLANs are always tagged with the proper VLAN identifier.

An administrative VLAN needs to be configured as well (for maintenance/management and authentication/accounting traffic). This VLAN is usually untagged (the command defining the VLAN has to be suffixed with the keyword “native”) and belongs to bridge-group 1. The Radio virtual interface for this VLAN does not need to be defined since default will keep the physical interface (Dot Radio 0) in bridge-group 1.

Since VLANs can be from 1 to 4094 and bridge-groups from 1 to 255, it is not necessary to have the same bridge-group id as the vlan id.

```
ap1200(config)#interface fastEthernet 0.902
ap1200(config-subif)#encapsulation dot1Q 902 native
ap1200(config-subif)#bridge-group 1

ap1200(config)#interface dot11Radio 0.903
ap1200(config-subif)#encapsulation dot1Q 903
ap1200(config-subif)#bridge-group 3

ap1200(config)#interface fastEthernet 0.903
ap1200(config-subif)#encapsulation dot1Q 903
ap1200(config-subif)#bridge-group 3

ap1200(config)#interface dot11Radio 0.906
ap1200(config-subif)#encapsulation dot1Q 906
ap1200(config-subif)#bridge-group 6

ap1200(config)#interface fastEthernet 0.906
ap1200(config-subif)#encapsulation dot1Q 906
ap1200(config-subif)#bridge-group 6

ap1200(config)#interface dot11Radio 0.909
ap1200(config-subif)#encapsulation dot1Q 909
ap1200(config-subif)#bridge-group 9

ap1200(config)#interface fastEthernet 0.909
ap1200(config-subif)#encapsulation dot1Q 909
ap1200(config-subif)#bridge-group 9
```

## 3.2.5 Supplicant

### 3.2.5.1 SecureW2

The usage of EAP-TTLS has been considered the easiest way to implement eduroam in a large (especially student) community. MS Windows has no built-in support for EAP-TTLS, but it can be added by installing SecureW2, a product from Alfa&Ariss Network Security Solution, and thus enable a large user community for

EAP-TTLS application. Installing SecureW2 can pose some security issues, which can be addressed by using a preconfigured distribution.

The main security problems are:

1. Allowing users to set up new connections – this is one of the advanced options of SecureW2; when disabled it will prevent users from carelessly accepting a rogue RADIUS server, and thus will prevent the users from sending their credentials to a fake server; however running SecureW2 this way requires the necessary certificates for the home RADIUS server to be preinstalled;
2. Installation of certificates – SecureW2 requires the certificates necessary to confirm server authenticity to be installed in a specific certificate store; which is different from the default certificate store used by Windows. The result is that even though the user has the main certificate of their home institution installed, SecureW2 will not be able to use it properly to establish the connection.

The proposed procedure is completely secure (under the assumption that users have previously installed their home institution certificate in the standard Windows certificate store). It also vastly simplifies the configuration of SecureW2.

Note: Installation instructions for a single SecureW2 client can be found in the appendices.

In order to prepare the preconfigured exe file, the administrator has to take the following steps:

1. Prepare the SecureW2.INF file
2. Prepare the NSIS configuration file
3. Create the exe file with NSIS
4. Digitally sign the exe file.

Detailed instructions of these administrative steps as well as an illustration of the installation process from a user perspective are provided in the following sections.

### 3.2.5.2 Preparation of the SecureW2 installer by the administrator

#### Prerequisites

1. Nullsoft Scriptable Install System (NSIS) – Download NSIS from <http://nsis.sourceforge.net/> and install it;
2. Code signing
  - a) Download the .NET Framework SDK from Microsoft and install it;
  - b) Prepare a code-signing key using openssl
    - Generate the key  
`openssl genrsa -des3 -out test_sign.key 1024`
    - and certification request  
`openssl req -new -key test_sign.key -out test_sign.csr`
  - c) The CA needs to certify the key for code signing. Using OpenSSL this is done by  
`openssl ca -out test_sign.crt -in test_sign.csr -extensions \ id_kp_codeSigning -extfile codesigning`
    - where the codesigning file contains:  
`[ id_kp_codeSigning ]  
extendedKeyUsage = 1.3.6.1.5.5.7.3.3`

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

- d) Generate the PFX file  
`openssl pkcs12 -export -out test_sign.pfx -in test_sign.crt -inkey \`  
`test_sign.key -clcerts`
- e) Install the PFX in Windows by double-clicking and accepting the default values except for the key protection where strong protection should be used

**Step 1.** Prepare the SecureW2.INF file containing:

```
[Version]
Signature = "$Windows NT$"
Provider = "Alfa & Ariss"
Config = 7

[Certificates]
Certificate.1 = your_ca_cert.der

[WZCSVC]
Enable = AUTO

[SSID.1]
Name = "eduroam"
Profile = "DEFAULT"
[Profile.1]
AuthenticationMethod = PAP
EAPType = 0
Name = "DEFAULT"
Description = "Enter your login credentials:"
UseAlternateIdentity = FALSE
VerifyServerCertificate = TRUE
PromptUserForCredentials = FALSE
TrustedRootCA.0 = your_ca_cert_fingerprint
UserName = PROMPTUSER
```

**Notes:**

1. Under the Certificates section you need to provide the entire certification path starting from the CA directly certifying your RADIUS server and ending with the root CA. Numbering goes from 1 up. As Certificate.0 you may (but do not have to) install the certificate of your RADIUS server.
2. As TrustedRootCA.0 you should provide a fingerprint of one of the CAs in the chain (the most natural choice will be the fingerprint of your organisation's CA)
3. Section WZCSVC causes the automatic start of Window Zero-Configuration service which is needed for SecureW2
4. There are many other options that you may put into the INI file (see SecureW2 administrators guide), but the provided example is sufficient for our configuration. One feature that one might want to add is the use of anonymous outer identity. The configuration above makes the outer identity equal to the real username.

**Step 2.** Prepare the NSIS configuration directory containing:

1. The configuration file – SecureW2.NSI:

```
!-----
!define APPLICATION "SecureW2 installer"
!define VERSION "1.0.0"
!-----
!include "MUI.nsh"
;General
;Name and file
Name "${APPLICATION} ${VERSION}"
OutFile "SecureW2_312_Test.exe"
!-----
;Interface Settings
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
!define MUI_ICON "your_icon.ico"
!define MUI_UNICON "your_icon.ico"
!define MUI_ABORTWARNING
Section "${APPLICATION}" SecInstall
SectionIn RO
; Extract all file to the temp dir
SetOutPath $TEMPDIR
; Define all the files required for the installation here:
File "SecureW2_312.exe"
File "SecureW2.INF"
File "your_ca_cert.der"
ExecWait "SecureW2_312.exe"
; If an error occurs then goto Error label else goto Continue label
IfErrors Error
Goto Continue
; Error Label, show error box and then quit
Error:
MessageBox MB_OK|MB_ICONEXCLAMATION "SecureW2 installation problem, please report
to ..."
; Continue Label
Continue:
; Remove temporary files
Delete "$TEMPDIR\SecureW2_312.exe"
Delete "$TEMPDIR\SecureW2.INF"
Delete "$TEMPDIR\your_ca_cert.der"
Quit
SectionEnd
```

**Note:** you should customise the OutFile name, the text in the MessageBox, the icon filename and the certificate filename(s)

2. The original SecureW2 exe file – SecureW2\_312.exe
3. The SecureW2.INF file
4. The icon file (your\_icon.ico) – it is advisable to have your own Windows icon to mark your customised installer;
5. The certificate file “your\_ca\_cert.der” (if your certificate chain consists of more files then you need to provide all files listed in the SecureW2.INF file and list them in the SecureW2.NSI file, remember to add them to the Delete section as well);

### Step 3. Prepare your customised installer

Right click on SecureW2.NSI and chose “Compile NSI script”. If all went well you should now have your exe file. If there were some errors, NSI will report them.

### Step 4. Sign your customised installer

```
C:\Program Files\Microsoft.NET\SDK\v2.0\Bin\signtool sign /a your_installer.exe
```

you can also use the GUI to signtool:

```
C:\Program Files\Microsoft.NET\SDK\v2.0\Bin\signtool signwizard
```

## 3.2.5.3 User installing SecureW2

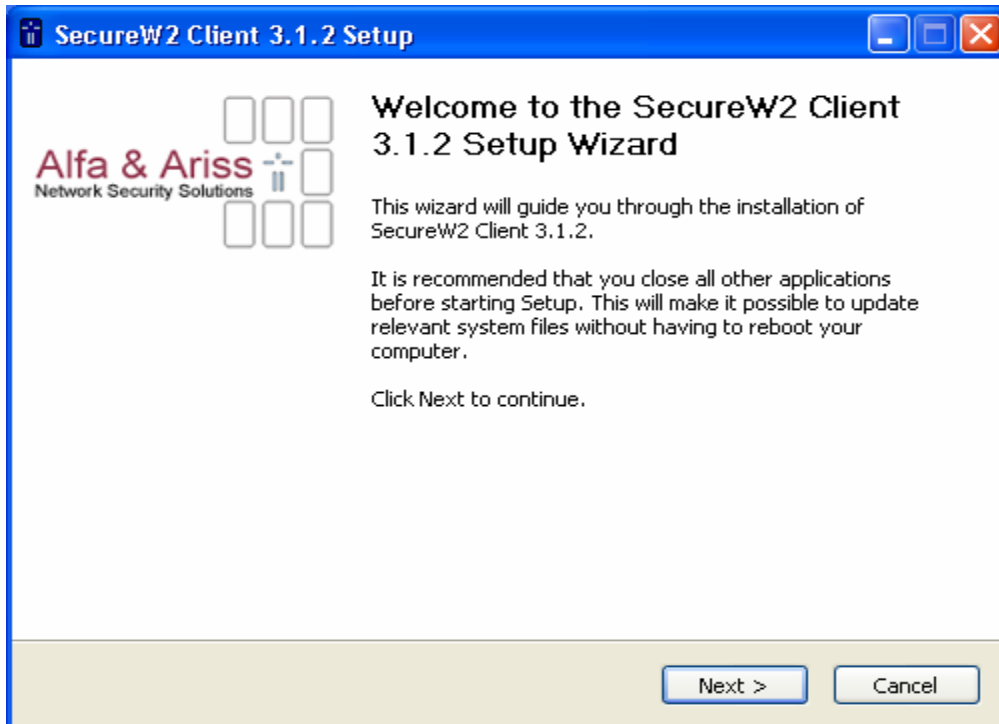
The steps for the user are:

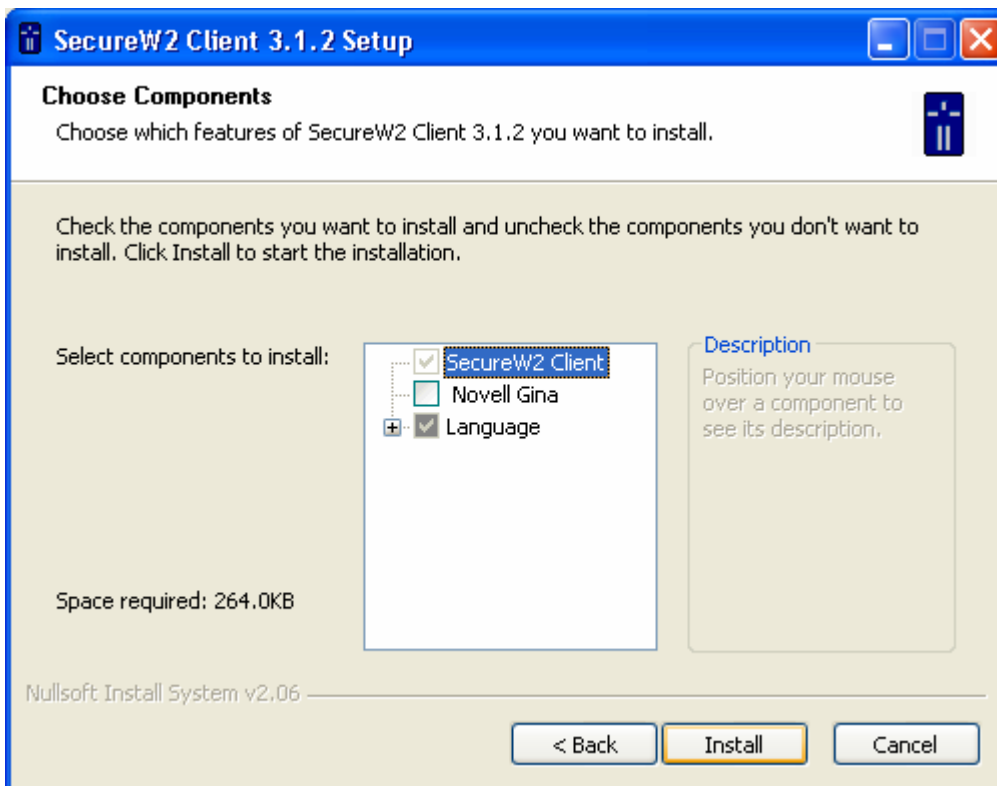
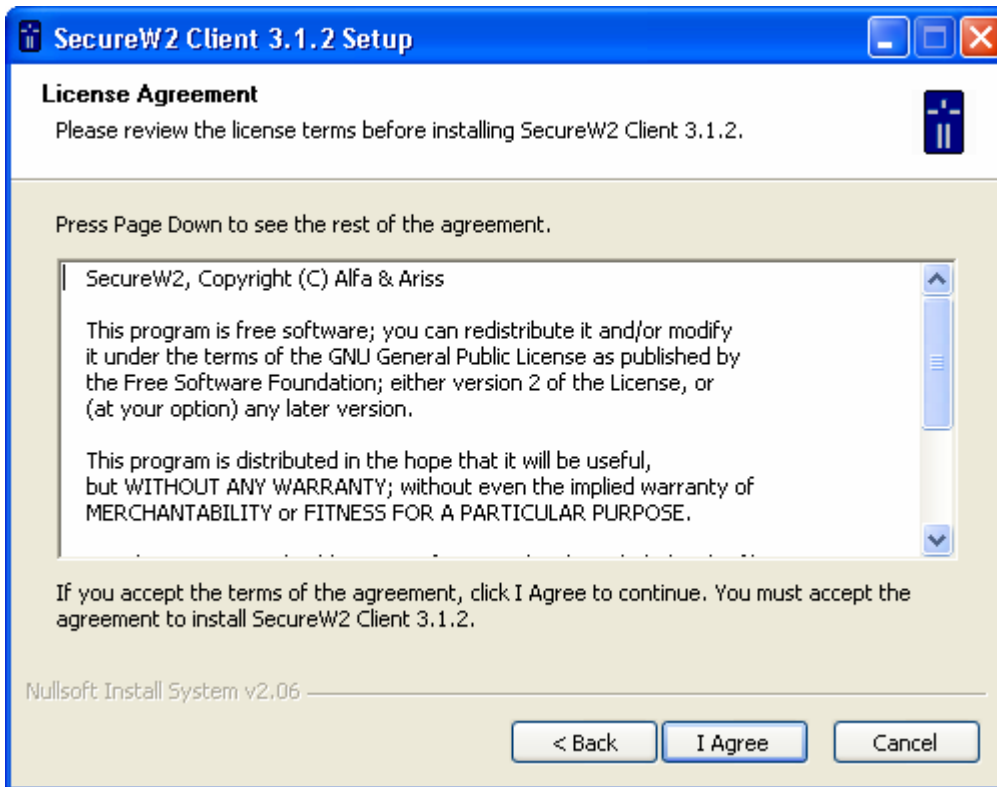
1. Download the preconfigured SecureW2 exe file

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

2. Confirm the signature of the exe file
3. Start the exe file and enter his/her credentials into the program prompt window
4. Reboot the computer
5. Choose SecureW2 as the authentication method for the eduroam network
6. Connect to eduroam

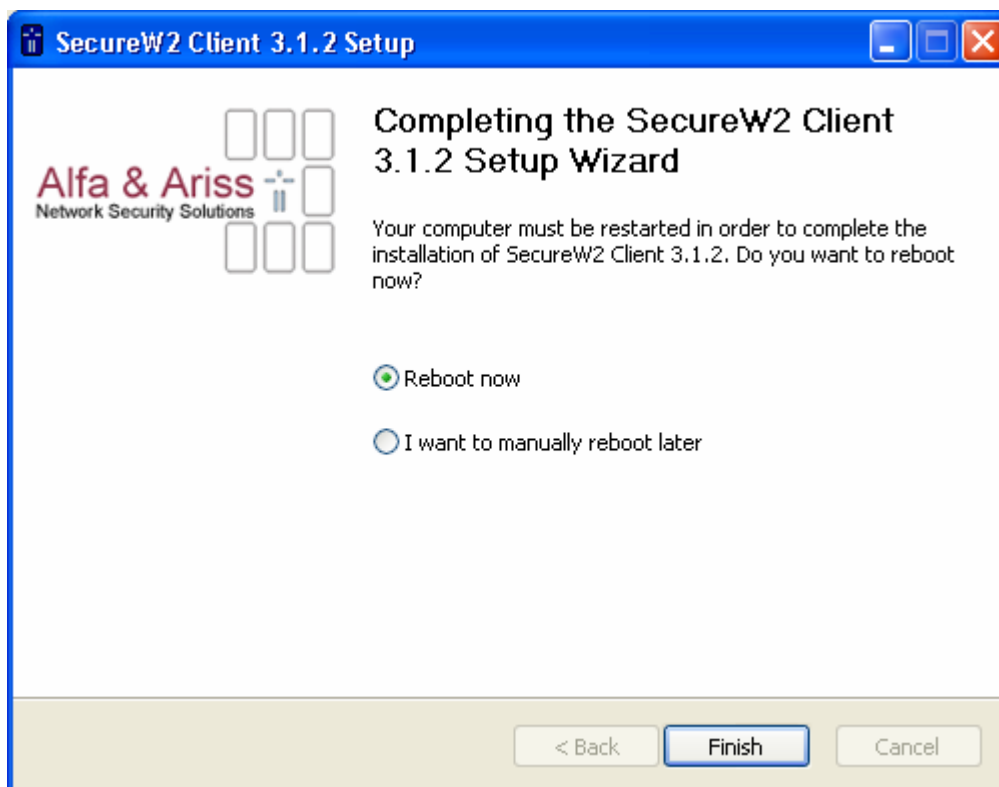
After downloading the installer, the user should right-click the filename and check the properties for the correct digital signature. After confirming that the file is genuine, the user should start the installer, that immediately starts the SecureW2 installer and the user has to go through a few self-explanatory steps:





Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8





Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

## 4 Conclusions

This document aims at giving administrators sufficient information to make it relatively straightforward to implement eduroam. It is obvious that campus architectures vary widely and it is far from us to suggest one specific vendor. However, the document provides both a cookbook for a typical setup with equipment that is found at many campuses in the research and education networking community in Europe as well as giving sufficient information for those using different architectures or vendors to understand the necessary steps to become 'eduroam-enabled'.

To further assist these administrators the appendices contain example configurations for many products as assembled by those in the NREN community in Europe. Furthermore, this document is meant to live on as a living document that will be constantly adapted and augmented.

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

## 5 References

- [GN2DJ511]** JRA5 Glossary of terms  
<http://intranet.geant2.net/server/show/conMediaFile.6254>
- [GN2DJ512]** Documentation on GEANT2 Roaming Requirements  
<http://www.geant2.net/upload/pdf/GN2-05-71v6.pdf>
- [GN2DJ513,2]** D.Simonsen, J.Rauschenbach, J.Howlett, R.Papez, R.Castro, T.Wilberg, K.Wierenga, S.Winter, et al. Roaming Policy and Legal Framework Document – Part2. GEANT2 Deliverable DJ5.1.3,2. November 2006. [http://www.geant2.net/upload/pdf/GN2-06-080v5-Deliverable\\_DJ5-1-3\\_2\\_Roaming\\_Policy\\_and\\_Legal\\_Framework-Part2\\_20061108094807.pdf](http://www.geant2.net/upload/pdf/GN2-06-080v5-Deliverable_DJ5-1-3_2_Roaming_Policy_and_Legal_Framework-Part2_20061108094807.pdf)
- [GN2DJ514]** K.Wierenga, S.Winter, R.Arends, R.Castro, P.Dekkers, H.Eertink, L.Guido, J.Leira, M.Linden, M.Milinovic, R.Papez, A.Peddemors, R.Poortinga, J.Rauschenbach, D.Simonsen, M.Sova, M.Stanica et al. Inter-NREN Roaming Architecture: Description and Development Items. GEANT2 Deliverable DJ5.1.4. September 2006. [http://www.geant2.net/upload/pdf/GN2-06-137v5-Deliverable\\_DJ5-1-4\\_Inter-NREN\\_Roaming\\_Technical\\_Specification\\_20060908164149.pdf](http://www.geant2.net/upload/pdf/GN2-06-137v5-Deliverable_DJ5-1-4_Inter-NREN_Roaming_Technical_Specification_20060908164149.pdf)
- [RFC2865]** Remote Authentication Dial In User Service (RADIUS)  
<http://www.ietf.org/rfc/rfc2865.txt>

## 6 Acronyms

In JRA5 used acronyms can be found in the JRA5 Glossary of Terms [GN2DJ511]. Additional terms are listed below.

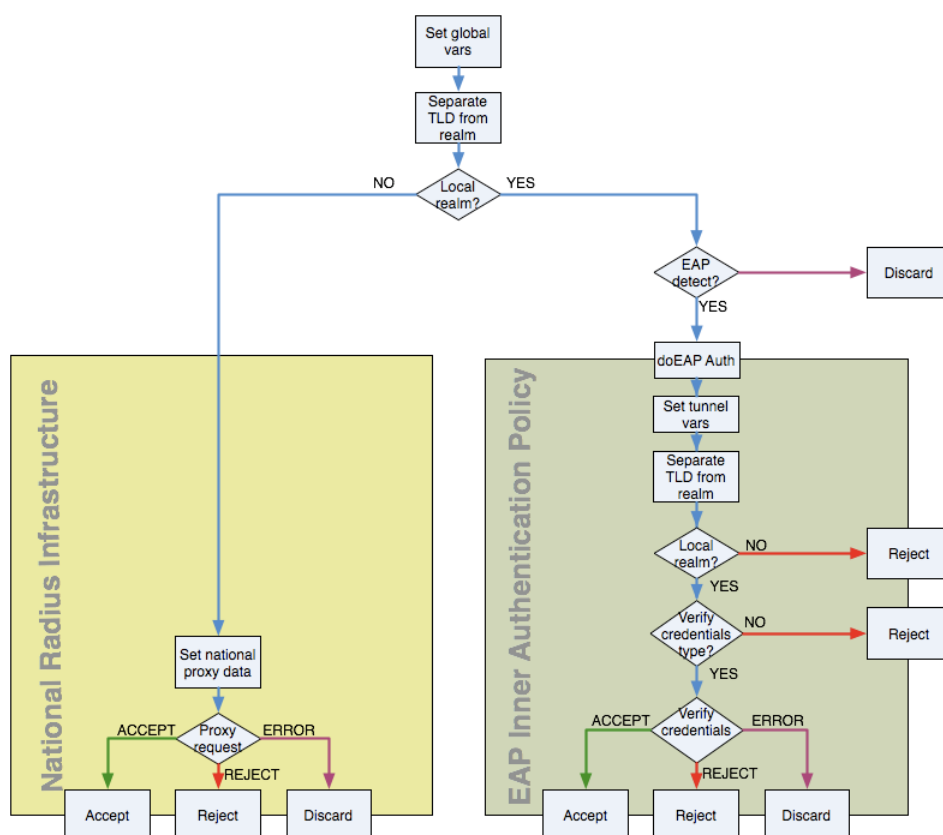
- [ARP]** [The Address Resolution Protocol (ARP) is the method for finding a host's hardware address when only its network layer address is known. Due to the overwhelming prevalence of IPv4 and Ethernet, ARP is primarily used to translate IP addresses to Ethernet MAC addresses. It is also used for IP over other LAN technologies, such as Token Ring, FDDI, or IEEE 802.11, and for IP over ATM]
- [BVI]** [A BVI (Bridge Virtual Interface) is a virtual interface within the campus switch router that acts like a normal *routed* interface. A BVI does not support bridging, but it actually represents the corresponding bridge group to routed interfaces within the switch router. The interface number is the link between the BVI and the bridge group.]
- [Credentials]** [A credential is a proof of qualification, competence, or clearance that is attached to a person, and often considered an attribute of that person. In this document it refers to username/password pairs.]
- [FQDN]** [A fully qualified domain name (or FQDN) is an unambiguous domain name that specifies the node's position in the DNS tree hierarchy absolutely. To distinguish an FQDN from a regular domain name, a trailing period is added. Ex: somehost.example.com. An FQDN differs from a regular domain name by its absoluteness; a suffix will not be added]
- [GUI]** [A graphical user interface (or GUI), is a particular case of user interface for interacting with a computer which employs graphical images and widgets in addition to text to represent the information and actions to the user. Usually the actions are performed through direct manipulation of the graphical elements.]
- [IANA]** [Internet Assigned Numbers Authority. IANA is broadly responsible for the allocation of globally-unique names and numbers that are used in Internet protocols that are published as RFC documents. It maintains a close liaison with the IETF and RFC Editor in fulfilling this function.]
- [MPPE]** [Microsoft Point-to-Point Encryption (MPPE) is a protocol for encrypting data across Point-to-Point Protocol and Virtual Private Network links. It uses the RSA RC4 encryption algorithm. MPPE supports 40-bit, 56-bit and 128-bit session keys, which are changed frequently to improve security. The exact frequency at that the keys are changed is negotiated, but may as frequent as every packet. MPPE alone does not compress or expand data, but the protocol is often used in conjunction with Microsoft Point-to-Point Compression, which compresses data across PPP or VPN links.]
- [MS-IAS]** [Internet Authentication Service (IAS) is the Microsoft Implementation of a Remote Authentication Dial-in User Service (RADIUS) server and proxy. As a RADIUS server, IAS performs centralized connection

authentication, authorization, and accounting for many types of network access, including wireless and virtual private network (VPN) connections. As a RADIUS proxy, IAS forwards authentication and accounting messages to other RADIUS servers.]

- [.NET]** [Microsoft .NET is an umbrella term that applies to a collection of products and technologies from Microsoft. Most have in common a dependence on the Microsoft .NET Framework, a component of the Windows operating system]
- [PFX]** [pfx (Personal inFormation eXchange) is a common file extension for X.509 certificates. A .pfx File may contain certificate(s) (public) and private keys (password protected)]
- [PID]** [The process identifier (normally referred to as the process ID or just PID) is a number used by some operating system kernels (such as that of UNIX or Windows NT) to uniquely identify a process. Under Unix, the PID of a newly created child process is returned by the fork() system call to the parent. The PID can be passed to process control functions like waitpid() or kill() to perform actions on the given process, and if the operating system as procfs support the files in /proc/pid/ can be queried for information about the process.]
- [SDK]** [A software development kit (SDK or “devkit”) is typically a set of development tools that allows a software engineer to create applications for a certain software package, software framework, hardware platform, operating system or similar. It may be something as simple as an “application programming interface” in the form of some files to interface to a particular programming language, or include sophisticated hardware to communicate with a certain embedded system. Common tools include debugging aids such as an IDE and other utilities. SDKs also frequently include sample code and supporting technical notes or other supporting documentation to help clarify points from the primary reference material.]
- [SNMP]** [The simple network management protocol (SNMP) forms part of the Internet protocol suite as defined by the Internet Engineering Task Force (IETF). More specifically, it is a Layer 7 or Application Layer protocol that is used by network management systems for monitoring network-attached devices for conditions that warrant administrative attention]

## Appendix A RADIUS servers

The main part of the document covered the configuration of the Radiator RADIUS implementation for the various servers in the eduroam hierarchy. As this cannot be understood as a recommendation to use this specific implementation, the following parts of Appendix A present the configuration of alternative RADIUS implementations. Starting with Radiator as institutional server only to complement the main part, later on the respective configurations for Freeradius, Navis, Vital AA and MS-IAS are presented. As the eduroam infrastructure is built upon RADIUS servers, the following picture depicting the message flow inside an institutional RADIUS server might be helpful.



Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

Figure A.1: Message flow in RADIUS server Identity Management System

## A.1 Radiator institutional server

Note : more explanation is given in the main document

Trace 4

```
LogDir /var/log/radius
DbDir /etc/radiator

AuthPort 1812
AcctPort 1813

<Client 192.168.10.200>
Secret 6.6obaFkm&RNs666
    Identifier ACCESSPOINT1
    IdenticalClients 192.168.10.201
</Client>

<Handler Client-Identifer=/^(?!Proxy-Identifer$)/>
    <AuthBy RADIUS>
        Host 192.87.36.3
        Secret super_secret!
        AuthPort 1812
        AcctPort 1813
        StripFromReply Tunnel-Type, Tunnel-Medium-Type, \
            Tunnel-Private-Group-ID
        AddToReply Tunnel-Type=1:VLAN, Tunnel-Medium-Type=1:Ether_802, \
            Tunnel-Private-Group-ID=1:909
    </AuthBy>
</Handler>

<AuthBy FILE>
    Identifier ID4-TLS
    Filename %D/TLS-users
    EAPType TLS
    EAPTLS_CAFfile %D/cert/institution-ca-chain.pem
    EAPTLS_CertificateFile %D/cert/radius-server-cert.pem
    EAPTLS_CertificateType PEM
    EAPTLS_PrivateKeyFile %D/cert/radius-server-key.pem
    EAPTLS_PrivateKeyPassword (the secret that secures the private-key)
    EAPTLS_MaxFragmentSize 1024
    AutoMPPEKeys
    SSLeayTrace 1
    StripFromReply Tunnel-Type, Tunnel-Medium-Type, Tunnel-Private-Group-ID
    AddToReply Tunnel-Type=1:VLAN, Tunnel-Medium-Type=1:Ether_802, Tunnel-Private-Group-
ID=1:909, User-Name=%u
</AuthBy>

<Handler Realm=instituion.cc, EAP-Message=/.+/>
    AuthBy ID4-TLS
</Handler>

<Handler TunnelledByPEAP=1, Realm=tunneled.institution.cc>
    <AuthBy FILE>
        Filename %D/peap-users
        EAPType MSCHAP-V2
    </AuthBy>
</Handler>
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
<Handler TunnelledByTTLS=1, Realm=tunneled.institution.cc>
  <AuthBy FILE>
    Filename %D/ttts-users
  </AuthBy>
</Handler>

<Handler Realm=group_1>
  <AuthBy FILE>
    # the %D/users file can be empty, it's there for normal PAP
    # authentication. This can however be used for the WEB captive
    # portals.
    Filename %D/users
    EAPType TTLS, PEAP
    EAPTLS_CAFfile %D/root.pem
    EAPTLS_CertificateFile %D/server.pem
    EAPTLS_CertificateType PEM
    EAPTLS_PrivateKeyFile %D/server.pem
    EAPTLS_PrivateKeyPassword serverkey
    EAPTLS_MaxFragmentSize 1024
    EAPAnonymous anonymous@group1
    AutoMPPEKeys
  </AuthBy>
</Handler>
```

## A.2 FreeRADIUS institutional server

### A.2.1 Setting up FreeRADIUS

FreeRADIUS (<http://www.freeradius.org>) is a very powerful, configurable, fast and freely available open-source RADIUS server. The sample configurations are based on FreeRADIUS 1.1.3 and require the regex patch for easy filtering of realms using regular expressions. The RPM installation packages for Fedora Core Linux and a standalone regex patch are available for download from the <http://www.pingo.org/eduroam>.

All the configuration files for FreeRADIUS are stored in a `/etc/raddb` directory. Configuration is broken down into several files by default.

Server is configured as follows:

- EAP-TTLS + PAP protocol for authenticating the clients
  - Outer identity is [anonymous@domain.tld](#)
- A few test user accounts are statically entered in a file
- Regular users are authenticated from an LDAP directory, using the eduPerson schema. Usernames are stored in eduPersonPrincipalName attribute and the plaintext password in userPassword attribute.
- Accounting is stored in a MySQL database



- Optionally a script can be set-up to update accounting SQL table with information of assigned IP addresses.

## A.2.2 Defining clients - Access Points and RADIUS servers

Access points, RADIUS servers and other RADIUS clients (NAS devices, RADIUS test scripts, ...) are defined in the file `/etc/raddb/clients.conf`. This file lists devices that may send requests to the server. First add the access points:

```
#####  
# APs and other devices  
#####  
client ap<name 1> {  
    secret = <shared secret 1>  
    shortname = ap<name 1>  
    nastype = cisco  
}  
[...]  
client ap<name N> {  
    secret = <shared secret N>  
    shortname = ap<name N>  
    nastype = cisco  
}
```

For Cisco APs the *nastype* is set to *cisco*, for other NAS devices the value should be set differently. It is recommended to use a different shared secret for every access point. You can use *mkpasswd* to randomly generate shared secrets. Since a linked RADIUS server is viewed as a RADIUS client device, they also have to be added here:

```
# TLD radius - 1  
client 10.1.2.3 {  
    secret = <TLD shared secret 1>  
    shortname = tld1  
    nastype = other  
}  
  
# TLD radius - 2  
client 10.1.2.4 {  
    secret = <TLD shared secret 2>  
    shortname = tld2  
    nastype = other  
}
```

A loopback client is useful for running testing scripts and even mandatory for tunnelled authentication methods like TTLS and PEAP, so we make sure it is set correctly:

```
client 127.0.0.1 {
    secret = <shared secret for Loopback>
    shortname = loopback
    nastype = other
}
```

### A.2.3 Configure realm handling and proxying

RADIUS requests from local users must be handled locally, while requests from roaming users must be proxied to the national TLD RADIUS server. If organisation has a domain *domain.tld* all requests for *\*.domain.tld* are forwarded from the national RADIUS server to their organisational RADIUS server and it is their responsibility to filter out invalid domains via the blackhole rule. Proxying is configured in the file */etc/raddb/proxy.conf*, the first rule with a match is used.

```
# Proxy servers configuration
proxy server {
    synchronous = no
    retry_delay = 5
    retry_count = 2
    dead_time = 120
    default_fallback = no
    post_proxy_authorize = yes
}

realm domain.tld {
}
realm student.domain.tld {
}

# blackhole and NULL are handled locally (later denied in users file)
realm blackhole.domain.tld {
    regex = "^.*\.domain\.tld$"
}
realm NULL {
}

# For unknown realms, forward requests to TLD1 or TLD2, round robin scheduling
realm DEFAULT {
    type = radius
    authhost = 10.1.2.3:1812
    accthost = 10.1.2.3:1813
    secret = <TLD shared secret 1>
    ldflag = round_robin
    nostrip
}
realm DEFAULT {
    type = radius
    authhost = 10.1.2.4:1812
    accthost = 10.1.2.4:1813
    secret = <TLD shared secret 2>
    ldflag = round_robin
}
```

```
        nostrip
    }
```

Value *ldflag* is set to *round\_robin* in order that requests to TLD servers are divided equally. The *nostrip* command specifies that no domain stripping is allowed (username must always be a Network Access Identifier in the form [short\\_username@domain.tld](#), for both local and roaming users

## A.2.4 Users authentication and realm handling

The configuration file */etc/raddb/users* is very important. It contains statically configured users, specifies how requests are handled and how different modules should authenticate and authorize the users. A `\` char shows an unintentional line-wrap. In the actual configuration file, the character must be removed and lines concatenated together:

```
#####
# Users with a blackholed or NULL realm should be rejected
#####
DEFAULT Realm == NULL, Auth-Type := Reject
DEFAULT Realm == blackhole.domain.tld, Auth-Type := Reject

#####
# User anonymous must be denied
# User anonymous@realm should be allowed, activate EAP
# (Caution! Watch the line wrap)
#####
DEFAULT User-Name =~ "^[Aa][Nn][Oo][Nn][Yy][Mm][Oo][Uu][Ss]@.*$", \
Auth-Type := EAP

#####
# Accounting fix for AP;
# We send a real user-name in the outer Access-Accept
#####
DEFAULT Realm == student.domain.tld, Freeradius-Proxied-To == 127.0.0.1
    User-Name = `%{User-Name}`,
    Fall-Through = yes
DEFAULT Realm == domain.tld, Freeradius-Proxied-To == 127.0.0.1
    User-Name = `%{User-Name}`,
    Fall-Through = yes
```

All NAS's used with this RADIUS configuration are required to follow RFC2865 recommendation for using the *User-Name* in *Access-Accept* for accounting information. With this entry the outer anonymous username is replaced by the real users username so NAS can send correct accounting for the correct user.

```
#####
# Static test account for the NREN
#####
nren_radius_test Realm == domain.tld, User-Password == "<test password>"
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

RADIUS links are established with UDP and are connectionless. Every organisation is required to provide a test username to the NREN (more exactly: TLD RADIUS maintainers) for testing of the RADIUS link. This way we can also verify all RADIUS links when the TLD RADIUS is reconfigured. Activate LDAP authorization for the students:

```
#####  
# Activate LDAP for the students, ID: LDAP2  
# Linewrap caution! This MUST also be a single line  
#####  
DEFAULT Realm == student.domain.tld, Autz-Type := LDAP2,\  
Freeradius-Proxied-To == 127.0.0.1
```

Activate LDAP authorization for the staff and assign them to a VLAN 313.

```
#####  
# Activate LDAP for the staff, ID: LDAP1  
# Linewrap caution! Only Tunnel-* attributes are on new lines  
#####  
DEFAULT Realm == domain.tld, Autz-Type := LDAP1,\  
Freeradius-Proxied-To == 127.0.0.1  
    Tunnel-Type = VLAN,  
    Tunnel-Medium-Type = IEEE-802,  
    Tunnel-Private-Group-Id = 313
```

## A.2.5 Setting up accounting in the SQL database

We log all accounting information into the SQL database. The configuration is in the file

*/etc/raddb/sql.conf*:

```
#####  
# MySQL settings for accounting  
#####  
sql {  
    driver = "rlm_sql_mysql"  
    server = "localhost"  
    login = "<user_for_mysql>"  
    password = "<password_for_mysql>"  
    radius_db = "radius"  
  
    accounting_start_query = "INSERT into ACCOUNTING SET\  
        `User-Name` = '%{User-Name}',\  
        `Calling-Station-Id` = '%{Calling-Station-Id}',\  
        `Called-Station-Id` = '%{Called-Station-Id}',\  
        `NAS-IP-Address` = '%{NAS-IP-Address}',\  
        `NAS-Port` = %{NAS-Port},\  
        `Timestamp Start` = NOW(),\  
"
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```

        `Acct-Unique-Session-Id` = '%{Acct-Unique-Session-Id}'
    "

    accounting_update_query = "UPDATE ACCOUNTING SET\
        `Acct-Session-Time` = '%{Acct-Session-Time}',\
        `Acct-Input-Octets` = '%{Acct-Input-Octets}',\
        `Acct-Output-Octets` = '%{Acct-Output-Octets}',\
        `Acct-Input-Packets` = '%{Acct-Input-Packets}',\
        `Acct-Output-Packets` = '%{Acct-Output-Packets}'\
    WHERE `Acct-Unique-Session-Id` = '%{Acct-Unique-Session-Id}'\
    LIMIT 1
    "

    accounting_stop_query = "UPDATE ACCOUNTING SET\
        `Timestamp Stop` = NOW(),\
        `Acct-Session-Time` = '%{Acct-Session-Time}',\
        `Acct-Input-Octets` = '%{Acct-Input-Octets}',\
        `Acct-Output-Octets` = '%{Acct-Output-Octets}',\
        `Acct-Input-Packets` = '%{Acct-Input-Packets}',\
        `Acct-Output-Packets` = '%{Acct-Output-Packets}',\
        `Acct-Terminate-Cause` = '%{Acct-Terminate-Cause}'\
    WHERE `Acct-Unique-Session-Id` = '%{Acct-Unique-Session-Id}'\
    LIMIT 1
    "
}

```

The format of the SQL 'RADIUS' database is:

```
mysql> describe ACCOUNTING;
```

Field	Type	Null	Key	Default	Extra
User-Name	varchar(100)				
Calling-Station-Id	varchar(100)				
Client-IP-Address	varchar(100)				
Called-Station-Id	varchar(100)				
NAS-IP-Address	varchar(100)				
NAS-Port	int(10) unsigned			0	
Timestamp Start	datetime			0000-00-00 00:00:00	
Timestamp Dhcp	datetime			0000-00-00 00:00:00	
Timestamp Stop	datetime			0000-00-00 00:00:00	
Acct-Unique-Session-Id	varchar(100)				
Acct-Session-Time	int(10) unsigned			0	
Acct-Input-Octets	int(10) unsigned			0	
Acct-Output-Octets	int(10) unsigned			0	
Acct-Input-Packets	int(10) unsigned			0	
Acct-Output-Packets	int(10) unsigned			0	
Acct-Terminate-Cause	varchar(100)				

How to create the database:

```

CREATE DATABASE radius;
USE radius;
CREATE TABLE ACCOUNTING (
    `User-Name` varchar(100) NOT NULL default '',
    `Calling-Station-Id` varchar(100) NOT NULL default '',

```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
`Client-IP-Address` varchar(100) NOT NULL default '',
`Called-Station-Id` varchar(100) NOT NULL default '',
`NAS-IP-Address` varchar(100) NOT NULL default '',
`NAS-Port` int(10) unsigned NOT NULL default '0',
`Timestamp Start` datetime NOT NULL default '1970-01-01 01:00:00',
`Timestamp Dhcp` datetime NOT NULL default '1970-01-01 01:00:00',
`Timestamp Stop` datetime NOT NULL default '1970-01-01 01:00:00',
`Acct-Unique-Session-Id` varchar(100) NOT NULL default '',
`Acct-Session-Time` int(10) unsigned NOT NULL default '0',
`Acct-Input-Octets` int(10) unsigned NOT NULL default '0',
`Acct-Output-Octets` int(10) unsigned NOT NULL default '0',
`Acct-Input-Packets` int(10) unsigned NOT NULL default '0',
`Acct-Output-Packets` int(10) unsigned NOT NULL default '0',
`Acct-Terminate-Cause` varchar(100) NOT NULL default ''
) TYPE=MyISAM;
```

## A.2.6 The master RADIUS configuration

The master FreeRADIUS configuration file `/etc/raddb/radiusd.conf` specifies which modules are to be used, and defines the actual authentication. This is the module configuration:

```
modules {
    # We use PAP authentication, passwords are in the clear
    pap {
        encryption_scheme = clear
    }

    $INCLUDE ${confdir}/eap.conf
    # If LDAP is not on localhost, you must use TLS
    ldap ldap_student {
        server = "localhost"
        identity = "cn=RADIUS,dc=domain,dc=tld"
        password = "<secret for identity dn>"
        basedn = "ou=student,dc=domain,dc=tld"
        filter = "(eduPersonPrincipalName=%{User-Name})"
        start_tls = no
    }
    ldap ldap_staff {
        server = "localhost"
        identity = "cn=RADIUS,dc=domain,dc=tld"
        password = "<secret for identity dn>"
        basedn = "ou=staff,dc=domain,dc=tld"
        filter = "(eduPersonPrincipalName=%{User-Name})"
        start_tls = no
    }
}
# usernames are in format: short_username@realm
realm suffix {
    format = suffix
    delimiter = "@"
    ignore_default = no
}
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
        ignore_null = no
    }

    # Mind the Linewrap!
    # This specifies which fields in accounting packet are used to compute the
    # unique accounting ID. This value is inserted into the MySQL database.
    acct_unique {
        key = "User-Name, Acct-Session-Id, Calling-Station-Id,\
        Called-Station-Id, NAS-IP-Address, NAS-Port"
    }

    $INCLUDE ${confdir}/sql.conf
}
```

The section “authorize” specifies how a request gets handled before the credential verification. The packet traverses modules in the following list:

- preprocess – cleans up some attributes from weird NAS's
- auth\_log – logs the packet
- suffix – determines the correct realm
- group – specifies to send packet to the files and ldap modules as follows. First process the files (file */etc/raddb/files*). If user is found, return result and if user isn't found, check the ldap module.

```
authorize {
    preprocess
    auth_log
    suffix
    group {
        # first check ldap and if isn't found see the files
        # the files also activates EAP for user anonymous

        # retrieve password from LDAP
        Autz-Type LDAP1 {
            ldap_staff
        }
        Autz-Type LDAP2 {
            ldap_student
        }

        files {
            notfound = 1
            ok = return
        }
    }
}
```

It makes sense to uncomment the sections concerning `pre_proxy_log` and `post_proxy_log` in the sample configuration file that's provided with the FreeRADIUS installation and to enable these sections in the `pre-proxy { }` and `post-proxy { }` stanza, respectively. It generates valuable log data that can assist in debugging and abuse tracking.

Here we disable all the automatic detection of authentication protocols and use only the ones that are manually set (via the users file and other modules). By default FreeRADIUS is configured to auto-detect the authentication protocol, which is bad when you strictly mandate only one (EAP-TTLS + PAP in our case):

```
authenticate {
    Auth-Type EAP {
        eap
    }
    # password was retrieved during authorization from LDAP or from file
    # pap module just checks that LDAP and user supplied passwords match
    Auth-Type PAP {
        pap
    }
}
```

Here we specify how accounting requests are handled. We preprocess, create a unique accounting ID, check the realm (send accounting to the home institution as well) and process through files:

```
preacct {
    preprocess
    acct_unique
    suffix
    files
}
```

Accounting details are logged to the file and then to the sql database:

```
accounting {
    detail
    sql
}
```

The replies that are sent by the server itself can also be logged (except Access-Challenge packets), which is useful to keep track of the actual outcome of the authentication. It is needed to define a detail instance `reply_log` (analogous to the instance `pre_proxy_detail` configuration in the sample config file) and configure the `post-auth { }` stanza in the server as follows:

```
post-auth {
    reply_log
    Post-Auth-Type REJECT {
        reply_log
    }
}
```



## A.2.7 Logging the client IP address (Optionally!)

Client IP address is logged in the DHCP log file. However this information can also be stored in the ACCOUNTING table with other accounting data and thus provide easy access to that data. A Sysadmin needs to set-up a script to update the SQL database information with the assigned IP address. The client IP address is determined by tailing the DHCP log file and monitoring all IP assignments. The script also monitors active connections and cleans up accounting (closes accounting) for stale connections. SNMP access to Access Points is required.

The script is available from here:

[http://www.eduroam.si/uploads/CN/eC/CNeCC3Uc7XI9Tw\\_dLtwYZg/eduroam\\_monitor-20060809.tar.bz2](http://www.eduroam.si/uploads/CN/eC/CNeCC3Uc7XI9Tw_dLtwYZg/eduroam_monitor-20060809.tar.bz2)

The access points need to be registered with the script. This is done by entering the needed access point data into the database.

```
USE radius;
create table access_points (
  `IP address` varchar(100) PRIMARY KEY NOT NULL,
  `snmp secret` varchar(100) NOT NULL default '',
  `radius secret` varchar(100) NOT NULL default '',
  `root username` varchar(100) NOT NULL default '',
  `root password` varchar(100) NOT NULL default ''
) TYPE=MyISAM;
```

## A.2.8 More information

The original Slovenian Eduroam technical specifications and sample configuration site:

<http://www.eduroam.si>

ARNES AAI technical support e-mail address: [aaa-podpora@arnes.si](mailto:aaa-podpora@arnes.si)

FreeRADIUS files:

<http://www.pingo.org/eduroam>

Eduroam-in-a-box web configuration tool:

<http://eduroam.sourceforge.net>

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

## A.3 Navis institutional and national server

### A.3.1 Institutional Navis RADIUS server (ver. 4.5.8)

Set-up of a local RADIUS server.

Following configuration files have to be changed:

- `server_properties`
- `method_select`
- `clients`

In addition the following files have to be replaced:

- `acct_methods`
- `auth_methods`

Changes in the `server_properties` file:

```
Radius-Acct-Address = "*:1813"  
Radius-Auth-Address = "*:1812"  
Database-Address = "0"  
Radius-CharSet = UTF8  
Delimiter-Precedence = "@"  
Suffix-Delimiters = "@"
```

Add the following line in the `method_select` file:

```
*      ClientClass setWorkingVars      doAcct
```

Add the lines with the national proxy server information in the `clients` file:

```
<192.168.1.10 national_server_secret>  
<192.168.1.20 national_server_secret>  
...
```

Create the file `acct_methods` with the following content:

```
#  
# This file specifies the processing steps (methods) to take for  
# each RADIUS accounting request. The initial method is selected  
# by rules in the method_select file.  
# -----  
# Drop any request Recieved  
# -----  
doAcct      Method-Type=WriteDebug Method-Disabled=FALSE  
            WriteDebug-Map = <<  
${Request Variable Group}=${request.*};  
${Packet Variable Group}=${packet.*};  
${User Variable Group}=${user.*};  
${Check Variable Group}=${check.*};  
${Reply Variable Group}=${reply.*};  
>>
```

Create the file `auth_methods` with the following content:

```
setWorkingVars      Method-Type=ReadWrite      Method-Next=checklocal      Method-On-
Fail=discardPolicyError Method-On-Error=discardPolicyError
    ReadWrite-Map = "${user.nrplip} := \"192.168.10.10\";\n${user.nrplport} :=
\"1812\";\n${user.nrplsecret} := \"some-secret\";\n${user.nrplretry} :=
\"1\";\n${user.nrpltimeout} := \"7000\";\n${user.nrp2ip} :=
\"192.168.10.20\";\n${user.nrp2port} := \"1812\";\n${user.nrp2secret} := \"some-
secret\";\n${user.nrp2retry} := \"1\";\n${user.nrp2timeout} :=
\"7000\";\n${user.localrealm} := \"lorealm.tld\";"
    ReadWrite-NewUser = "FALSE"
```

```
checklocal Method-Type=Compare      Method-On-Error=discardPolicyError      Method-On-
Fail=setProxyVars Method-Next=checkEAP
    Compare-Input1 = "${packet.User-Realm[trim,toLower]}"
    Compare-Input2 = "${user.user.localrealm[trim,toLower]}"
    Compare-Operator = "=="
```

```
setProxyVars      Method-Type=ReadWrite      Method-Next=doProxy1      Method-On-
Fail=discardPolicyError Method-On-Error=discardPolicyError
    ReadWrite-Map = "${user.proxy1ip} := ${user.nrplip};\n${user.proxy1port} :=
${user.nrplport};\n${user.proxy1secret} :=
${user.nrplsecret};\n${user.proxy1retry} :=
${user.nrplretry};\n${user.proxy1timeout} :=
${user.nrpltimeout};\n${user.proxy2ip} := ${user.nrp2ip};\n${user.proxy2port} :=
${user.nrp2port};\n${user.proxy2secret} :=
${user.nrp2secret};\n${user.proxy2retry} :=
${user.nrp2retry};\n${user.proxy2timeout} :=
${user.nrp2timeout};\n${user.proxyname} := \"Proxy RADIUS\";"
    ReadWrite-NewUser = "FALSE"
```

```
doProxy1      Method-Type=Radius      Method-On-Error=doProxy2      Method-
Next=acceptAuthentication Method-On-Fail=rejectAuthentication
    Radius-ServerAddress = "${user.proxy1ip}:${user.proxy1port}"
    Radius-Secret = "${user.proxy1secret}"
    Radius-Dictionary = "#default"
    Radius-Timeout = "${user.proxy1timeout}"
    Radius-Retries = "${user.proxy1retry}"
    Radius-RequestMap = "${*}=${request.*};"
    Radius-PacketType = "${packet.Packet-Type}"
    Radius-ClientAddress = "*"
    Radius-CharSet = "UTF8"
    Radius-InauthenticFailure = "FALSE"
    Radius-CheckAuthenticator = "TRUE"
    Radius-RemoveTrailingNul = "TRUE"
    Radius-AppendTrailingNul = "FALSE"
    Radius-StrictEncoding = "FALSE"
    Radius-CopyMode = "FALSE"
    Radius-Mib = "AUTO"
    Radius-RecvBufferSize = "8192"
    Radius-SendBufferSize = "8192"
```

```
doProxy2      Method-Type=Radius      Method-Next=acceptAuthentication      Method-On-
Fail=rejectAuthentication Method-On-Error=discardProxiesError Method-Disabled=FALSE
    Radius-ServerAddress = "${user.proxy2ip}:${user.proxy2port}"
    Radius-Secret = "${user.proxy2secret}"
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```

Radius-Dictionary = "#default"
Radius-Timeout = "${user.proxy2timeout}"
Radius-Retries = "${user.proxy2retry}"
Radius-RequestMap = "${*}=${request.*};"
Radius-ReplyMap = "${reply.*}=${*};"
Radius-PacketType = "${packet.Packet-Type}"
Radius-ClientAddress = "*"
Radius-CharSet = "UTF8"
Radius-InauthenticFailure = "FALSE"
Radius-CheckAuthenticator = "TRUE"
Radius-RemoveTrailingNul = "TRUE"
Radius-AppendTrailingNul = "FALSE"
Radius-StrictEncoding = "FALSE"
Radius-CopyMode = "FALSE"
Radius-Mib = "AUTO"
Radius-RecvBufferSize = "8192"
Radius-SendBufferSize = "8192"

acceptAuthentiation      Method-Type=Return
    Return-Disposition = "ACCEPT"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH:  ACCEPT  for user  ${packet.Base-User-
Name}@${packet.User-Realm} authenticated on ${user.proxyname}"

rejectAuthentiation      Method-Type=Return
    Return-Disposition = "REJECT"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH:  REJECT  for user  ${packet.Base-User-
Name}@${packet.User-Realm} authenticated on ${user.proxyname}"

checkEAP      Method-Type=Compare      Method-On-Error=discardPolicyError      Method-On-
Fail=discardEAPError      Method-Next=doEAPTTLS
    Compare-Input1 = "${packet.EAP-Identity}"
    Compare-Input2 = ""
    Compare-Operator = "!="

doEAPTTLS      Method-Type=AuthEapTtls      Method-On-Error=discardPolicyError
    AuthEapTtls-TunnelMethod = "setLocalVars"
    AuthEapTtls-TunnelWriteMap =
"${request.*}:=${tunnel.*};\n${user.localrealm} := ${user.localrealm};"
    AuthEapTtls-RsaCertFile = "server.pem"
    AuthEapTtls-RsaKeyPassword = "secret"
    AuthEapTtls-DsaCertFile = ""
    AuthEapTtls-DsaKeyPassword = ""
    AuthEapTtls-TrustedFile = "trust.pem"
    AuthEapTtls-FragmentSize = "253"
    AuthEapTtls-CertificateMap = "${packet.X509-Serial-Number} = ${Serial-
Number};\n${packet.X509-Subject-DN} = ${subject-DN};\n${packet.X509-Issuer-DN} =
${Issuer-DN};"
    AuthEapTtls-KeyMap = "${reply.MS-MPPE-Recv-Key} := ${1-
32[isRadius]};\n${reply.MS-MPPE-Send-Key} := ${33-64[isRadius]};\n${reply.EAP-
Master-Session-Key} := ${1-64[isDiameter]};"

setLocalVars      Method-Type=ReadWrite      Method-Next=checkRealm      Method-On-
Fail=discardPolicyError      Method-On-Error=discardPolicyError

```

```

ReadWrite-Map = "${user.LDAPIP} := \"192.168.100.1\";\n${user.LDApport} :=
\"389\";\n${user.uid} := ${packet.Base-User-Name[trim,toLower]};\n"
ReadWrite-NewUser = "FALSE"

checkRealm Method-Type=Compare Method-On-Error=discardPolicyError Method-On-
Fail=rejectRealm Method-Next=getRealmCountry
Compare-Input1 = "${packet.User-Realm[trim,toLower]}"
Compare-Input2 = "${user.localrealm[trim,toLower]}"
Compare-Operator = "=="

getRealmCountry Method-Type=PatternMatch Method-Next=checkPassword Method-On-
Fail=rejectRealm Method-On-Error=discardPolicyError
PatternMatch-SearchKey = "${packet.User-Realm[toLower]}"
PatternMatch-Mode = "REGEX"
PatternMatch-Operation = "MATCHES"
PatternMatch-Case = "(.+)\.\.(.+)$ ${user.realm}=${1};${user.country}=${2};"
PatternMatch-IgnoreCase = "TRUE"
PatternMatch-SingleLine = "FALSE"
PatternMatch-MultiLine = "FALSE"
PatternMatch-Extended = "FALSE"

checkPassword Method-Type=Compare Method-On-Fail=rejectPassword Method-On-
Error=discardPolicyError Method-Next=doLDAP
Compare-Input1 = "${request.password}"
Compare-Input2 = ""
Compare-Operator = "!="

doLDAP Method-Type=Ldap Method-On-Error=discardLDAPError Method-
Next=acceptLocalAuthentication Method-On-Fail=rejectLocalAuthentication
Ldap-ServerAddress = "${user.LDAPIP}:${user.LDApport}"
Ldap-Version = "3"
Ldap-BindDn = "uid=${user.uid},dc=${user.realm},dc=${user.country}"
Ldap-BindPassword = "${request.password}"
Ldap-Operation = "BIND"
Ldap-SearchBase = "ou=People, o=${packet.User-Realm[escape]}"
Ldap-SearchScope = "SCOPE_ONE"
Ldap-SearchFilter = "(mail=${packet.Base-User-Name[escape]}@*)"
Ldap-ConnectionLimit = "2147483647"
Ldap-BindTimeout = "600000"
Ldap-ConnectionTimeout = "10000"
Ldap-OperationTimeout = "10000"
Ldap-CacheConnections = "TRUE"
Ldap-AuthFailureIsError = "FALSE"
Ldap-NewUser = "FALSE"
Ldap-StartTls = "FALSE"

rejectPassword Method-Type=Return
Return-Disposition = "REJECT"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: REJECT for user ${packet.Base-User-
Name}@${packet.User-Realm} no valid password"

rejectRealm Method-Type=Return
Return-Disposition = "REJECT"

```

```
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: REJECT for user ${packet.Base-User-Name}@${packet.User-Realm} no valid realm"

acceptLocalAuthentication Method-Type=Return
Return-Disposition = "ACCEPT"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: ACCEPT for user ${packet.Base-User-Name}@${packet.User-Realm} authenticated on LDAP"

rejectLocalAuthentication Method-Type=Return
Return-Disposition = "REJECT"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: REJECT for user ${packet.Base-User-Name}@${packet.User-Realm} authenticated on LDAP"

discardUnknownCountry Method-Type=Return
Return-Disposition = "DISCARD"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: DISCARD unknown country ${user.country}"

discardEmptyCountry Method-Type=Return
Return-Disposition = "DISCARD"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: DISCARD empty country ${packet.User-Realm}"

discardPolicyError Method-Type=Return
Return-Disposition = "DISCARD"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: DISCARD policy error ${packet.Last-Disposition-Message}"

discardProxiesError Method-Type=Return
Return-Disposition = "DISCARD"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: DISCARD proxies error ${user.proxyname}"

discardEAPError Method-Type=Return
Return-Disposition = "DISCARD"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: DISCARD not EAP ${request.User-Name}"

discardLDAPError Method-Type=Return
Return-Disposition = "DISCARD"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: DISCARD LDAP not respond"
```

In the above file enter the correct info about the proxy servers, (for IdP case ONLY) the local realm and the IdM system:

```
${user.nrp*} – enter correct info about the proxy servers (IP addresses, ports, ...)
${user.localrealm} – enter correct info about the local realm (IdP ONLY)
${user.LDAP*} enter correct info about the local LDAP server (IdM system)
```

Additional files needed (due to the use of EAP):

server.pem - server certificate

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

trust.pem – root certificate

### A.3.2 National Navis RADIUS server (ver. 4.5.8)

Set-up of a (national) RADIUS proxy server.

The following configuration files have to be changed:

- server\_properties
- method\_select
- clients

In addition the following files have to be replaced:

- acct\_methods
- auth\_methods

Changes in the server\_properties file:

```
Radius-Acct-Address = "*:1813"  
Radius-Auth-Address = "*:1812"  
Database-Address = "0"  
Radius-CharSet = UTF8  
Delimiter-Precedence = "@"  
Suffix-Delimiters = "@"
```

Add the following line in the method\_select file:

```
* ClientClass setWorkingVars doDropRequest
```

Add the lines with the eduroam proxy server and local RADIUS servers information to the clients file:

```
192.87.106.34 <eduroam_secret>  
130.225.242.109 <eduroam_secret>  
<local_server_1_IP> <local_server_secret>  
<local_server_2_IP> <local_server_secret>  
...
```

Create the file acct\_methods with the following content:

```
#  
# This file specifies the processing steps (methods) to take for  
# each RADIUS accounting request. The initial method is selected  
# by rules in the method_select file.  
# -----  
-----  
# Drop any request Recieved  
# -----  
-----
```

```
doDropRequest Method-Type=Return Method-Disabled=FALSE
    Return-Disposition = "DISCARD"
    Return-LogLevel = "INFO"
    Return-LogMessage = "ACCN: DISCARD Client ${packet.Source-Address}
sent accounting"
```

Create the file `auth_methods` with the following content:

```
setWorkingVars Method-Type=ReadWrite Method-Disabled=FALSE Method-
Next=getCountry Method-On-Fail=discardPolicyError Method-On-
Error=discardPolicyError
    ReadWrite-Map = <<
${user.erplip} := "192.87.106.34";
${user.erplport} := "1812";
${user.erplsecret} := "some-secret";
${user.erplretry} := "1";
${user.erpltimeout} := "7000";
${user.erp2ip} := "130.225.242.109";
${user.erp2port} := "1812";
${user.erp2secret} := "some-secret";
${user.erp2retry} := "1";
${user.erp2timeout} := "7000";
${user.loplip} := "192.168.200.1";
${user.loplport} := "1812";
${user.loplsecret} := "local-some-secret";
${user.loplretry} := "1";
${user.lopltimeout} := "7000";
${user.lop2ip} := "192.168.200.2";
${user.lop2port} := "1812";
${user.lop2secret} := "local-some-secret";
${user.lop2retry} := "1";
${user.lop2timeout} := "7000";
${user.eduroamcountry} := "at*bg*hr*cz*dk*ee*fr*fi*de*gr*hu*it*ir*lv*lt*lu*mt*nl*no*pl*pt*ro*si*es*uk*ch";
${user.localcountry} := "lo";
>>
    ReadWrite-NewUser = "FALSE"
```

```
getCountry Method-Type=PatternMatch Method-Disabled=FALSE Method-
Next=setUserCountry Method-On-Fail=discardEmptyCountry Method-On-
Error=discardPolicyError
    PatternMatch-SearchKey = "${packet.User-Realm}"
    PatternMatch-Mode = "REGEX"
    PatternMatch-Operation = "MATCHES"
    PatternMatch-Case = "(+)\.\.\.(+)\$ ${user.country} = ${2};"
    PatternMatch-IgnoreCase = "TRUE"
    PatternMatch-SingleLine = "FALSE"
    PatternMatch-MultiLine = "FALSE"
    PatternMatch-Extended = "FALSE"
```

```
setUserCountry Method-Type=ReadWrite Method-Disabled=FALSE Method-
Next=checkeduroam Method-On-Fail=discardPolicyError Method-On-
Error=discardPolicyError
    ReadWrite-Map = "${user.countryf} := \"*${user.country[toLowerCase]}\"";
    ReadWrite-NewUser = "FALSE"
```



```

checkeduroam      Method-Type=Compare      Method-Disabled=FALSE      Method-On-
Fail=checklocal  Method-On-Error=discardPolicyError  Method-Next=setProxyVarseduroam
    Compare-Input1 = "${user.eduroamcountry}"
    Compare-Input2 = "${user.countryf}"
    Compare-Operator = "contains"

checklocal  Method-Type=Compare      Method-Disabled=FALSE      Method-On-
Fail=discardUnknownCountry  Method-On-Error=discardPolicyError  Method-
Next=setProxyVarslocal
    Compare-Input1 = "${user.localcountry}"
    Compare-Input2 = "${user.countryf}"
    Compare-Operator = "=="

setProxyVarseduroam  Method-Type=ReadWrite  Method-On-Fail=discardPolicyError
Method-On-Error=discardPolicyError  Method-Disabled=FALSE  Method-Next=doProxy1
    ReadWrite-Map = <<
${user.proxy1ip} := ${user.erplip};
${user.proxy1port} := ${user.erplport};
${user.proxy1secret} := ${user.erplsecret};
${user.proxy1retry} := ${user.erplretry};
${user.proxy1timeout} := ${user.erpltimeout};
${user.proxy2ip} := ${user.erp2ip};
${user.proxy2port} := ${user.erp2port};
${user.proxy2secret} := ${user.erp2secret};
${user.proxy2retry} := ${user.erp2retry};
${user.proxy2timeout} := ${user.erp2timeout};
${user.proxyname} := "eduroam";
>>
    ReadWrite-NewUser = "FALSE"

setProxyVarslocal  Method-Type=ReadWrite  Method-On-Fail=discardPolicyError  Method-
On-Error=discardPolicyError  Method-Disabled=FALSE  Method-Next=doProxy1
    ReadWrite-Map = <<
${user.proxy1ip} := ${user.loplip};
${user.proxy1port} := ${user.loplport};
${user.proxy1secret} := ${user.loplsecret};
${user.proxy1retry} := ${user.loplretry};
${user.proxy1timeout} := ${user.lopltimeout};
${user.proxy2ip} := ${user.lop2ip};
${user.proxy2port} := ${user.lop2port};
${user.proxy2secret} := ${user.lop2secret};
${user.proxy2retry} := ${user.lop2retry};
${user.proxy2timeout} := ${user.lop2timeout};
${user.proxyname} := "local";
>>
    ReadWrite-NewUser = "FALSE"

doProxy1  Method-Type=Radius  Method-Disabled=FALSE  Method-On-Error=doProxy2
Method-Next=acceptAuthentication  Method-On-Fail=rejectAuthentication
    Radius-ServerAddress = "${user.proxy1ip}:${user.proxy1port}"
    Radius-Secret = "${user.proxy1secret}"
    Radius-Dictionary = "#default"
    Radius-Timeout = "${user.proxy1timeout}"
    Radius-Retries = "${user.proxy1retry}"
    Radius-RequestMap = "${*}=${request.*}";

```

```
Radius-ReplyMap = "${reply.*}=${*};"  
Radius-PacketType = "${packet.Packet-Type}"  
Radius-ClientAddress = "*"   
Radius-CharSet = "UTF8"   
Radius-InauthenticFailure = "FALSE"   
Radius-CheckAuthenticator = "TRUE"   
Radius-RemoveTrailingNul = "TRUE"   
Radius-AppendTrailingNul = "FALSE"   
Radius-StrictEncoding = "FALSE"   
Radius-CopyMode = "FALSE"   
Radius-Mib = "AUTO"   
Radius-RecvBufferSize = "8192"   
Radius-SendBufferSize = "8192"
```

```
doProxy2      Method-Type=Radius      Method-Disabled=FALSE      Method-  
Next=acceptAuthentication      Method-On-Fail=rejectAuthentication      Method-On-  
Error=discardProxiesError
```

```
Radius-ServerAddress = "${user.proxy2ip}:${user.proxy2port}"  
Radius-Secret = "${user.proxy2secret}"  
Radius-Dictionary = "#default"  
Radius-Timeout = "${user.proxy2timeout}"  
Radius-Retries = "${user.proxy2retry}"  
Radius-RequestMap = "${*}=${request.*};"  
Radius-ReplyMap = "${reply.*}=${*};"  
Radius-PacketType = "${packet.Packet-Type}"  
Radius-ClientAddress = "*"   
Radius-CharSet = "UTF8"   
Radius-InauthenticFailure = "FALSE"   
Radius-CheckAuthenticator = "TRUE"   
Radius-RemoveTrailingNul = "TRUE"   
Radius-AppendTrailingNul = "FALSE"   
Radius-StrictEncoding = "FALSE"   
Radius-CopyMode = "FALSE"   
Radius-Mib = "AUTO"   
Radius-RecvBufferSize = "8192"   
Radius-SendBufferSize = "8192"
```

```
discardUnknownCountry      Method-Type=Return      Method-Disabled=FALSE  
Return-Disposition = "DISCARD"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: DISCARD unknown country ${user.country}"
```

```
discardEmptyCountry      Method-Type=Return      Method-Disabled=FALSE  
Return-Disposition = "DISCARD"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: DISCARD empty country ${packet.User-Realm}"
```

```
discardPolicyError      Method-Type=Return      Method-Disabled=FALSE  
Return-Disposition = "DISCARD"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: DISCARD policy error ${packet.Last-Disposition-  
Message}"
```

```
discardProxiesError      Method-Type=Return      Method-Disabled=FALSE  
Return-Disposition = "DISCARD"
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: DISCARD proxies error ${user.proxyname}"

acceptAuthentication Method-Type=Return Method-Disabled=FALSE
Return-Disposition = "ACCEPT"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: ACCEPT for user ${packet.Base-User-Name}@${packet.User-Realm} authenticated on ${user.proxyname}"

rejectAuthentication Method-Type=Return Method-Disabled=FALSE
Return-Disposition = "REJECT"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: REJECT for user ${packet.Base-User-Name}@${packet.User-Realm} authenticated on ${user.proxyname}"
```

In the above file enter the correct info about the eduroam and local servers. Also enter the list of TLDs participating in eduroam:

```
${user.erp*}- enter correct info about the eduroam servers (IP addresses, ports, ...)
${user.lop*}- enter correct info about the local RADIUS servers (IP addresses, ports, ...)
${user.eduroamcountry} - enter the list of TLDs participating in eduroam
${user.localcountry} - enter local TLD
```

## A.4 VitalAAA Institutional and national server

### A.4.1 Institutional VitalAAA Server (ver. 5.0.10)

Set-up of a local RADIUS server:

The following configuration files have to be changed:

- server\_properties
- method\_dispatch
- clients

In addition the following files have to be created.

- error.pf
- proxy.pf
- acct.pf
- prepare.pf
- local.pf

Changes in the server\_properties file:

```
Radius-Acct-Address = "*:1813"
Radius-Auth-Address = "*:1812"
Database-Address = "0"
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
Radius-CharSet = UTF8
Delimiter-Precedence = "@"
Suffix-Delimiters = "@"
```

Changes in the `method_dispatch` file:

```
radius      Auth 1      prepare      setWorkingVars
radius      acct 4      acct         dumpAcct
```

Add the lines with the national proxy server information in the `clients` file:

```
<192.168.1.10 national_server_secret>
<192.168.1.20 national_server_secret>
...
```

Create the file `error.pf` with the following content:

```
discardUnknownCountry
    Method-Type = "Return"
    Method-Disabled = "FALSE"
    Return-Disposition = "DISCARD"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH: DISCARD unknown country ${user.country}"

discardEmptyCountry
    Method-Type = "Return"
    Method-Disabled = "FALSE"
    Return-Disposition = "DISCARD"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH: DISCARD empty country ${packet.User-Realm}"

discardPolicyError
    Method-Type = "Return"
    Method-Disabled = "FALSE"
    Return-Disposition = "DISCARD"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH: DISCARD policy error ${packet.Last-Disposition-
Message}"

discardProxiesError
    Method-Type = "Return"
    Method-Disabled = "FALSE"
    Return-Disposition = "DISCARD"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH: DISCARD proxies error ${user.proxyname}"

discardEAPError
    Method-Type = "Return"
    Method-Disabled = "FALSE"
    Level-On-Success = "INFO"
    Level-On-Failure = "INFO"
    Level-On-Error = "INFO"
    Return-Disposition = "DISCARD"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH: DISCARD not EAP ${request.User-Name}"

discardLDAPError
```

```
Method-Type = "Return"  
Method-Disabled = "FALSE"  
Level-On-Success = "INFO"  
Level-On-Failure = "INFO"  
Level-On-Error = "INFO"  
Return-Disposition = "DISCARD"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: DISCARD LDAP not respond"
```

Create the file `proxy.pf` with the following content:

```
doProxy1  
    Method-Type = "Radius"  
    Method-Disabled = "FALSE"  
    Method-On-Error = "doProxy2"  
    Method-On-Success = "acceptAuthentication"  
    Method-On-Failure = "rejectAuthentication"  
    Radius-ServerAddress = "${user.proxy1ip}:${user.proxy1port}"  
    Radius-Secret = "${user.proxy1secret}"  
    Radius-Dictionary = "#default"  
    Radius-Timeout = "${user.proxy1timeout}"  
    Radius-Retries = "${user.proxy1retry}"  
    Radius-RequestMap = "${*}=${request.*};"  
    Radius-PacketType = "${packet.Packet-Type}"  
    Radius-ClientAddress = "*"   
    Radius-CharSet = "UTF8"  
    Radius-InauthenticFailure = "FALSE"  
    Radius-CheckAuthenticator = "TRUE"  
    Radius-RemoveTrailingNul = "TRUE"  
    Radius-AppendTrailingNul = "FALSE"  
    Radius-StrictEncoding = "FALSE"  
    Radius-CopyMode = "FALSE"  
    Radius-Mib = "AUTO"  
    Radius-RecvBufferSize = "8192"  
    Radius-SendBufferSize = "8192"  
    Radius-SuccessMap = "${reply.*}=${*};"  
  
doProxy2  
    Method-Type = "Radius"  
    Method-Disabled = "FALSE"  
    Method-On-Success = "acceptAuthentication"  
    Method-On-Failure = "rejectAuthentication"  
    Radius-ServerAddress = "${user.proxy2ip}:${user.proxy2port}"  
    Radius-Secret = "${user.proxy2secret}"  
    Radius-Dictionary = "#default"  
    Radius-Timeout = "${user.proxy2timeout}"  
    Radius-Retries = "${user.proxy2retry}"  
    Radius-RequestMap = "${*}=${request.*};"  
    Radius-PacketType = "${packet.Packet-Type}"  
    Radius-ClientAddress = "*"   
    Radius-CharSet = "UTF8"  
    Radius-InauthenticFailure = "FALSE"  
    Radius-CheckAuthenticator = "TRUE"  
    Radius-RemoveTrailingNul = "TRUE"  
    Radius-AppendTrailingNul = "FALSE"  
    Radius-StrictEncoding = "FALSE"
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
Radius-CopyMode = "FALSE"
Radius-Mib = "AUTO"
Radius-RecvBufferSize = "8192"
Radius-SendBufferSize = "8192"
Radius-SuccessMap = "${reply.*}=${*};"
Method-On-Error = "error:discardProxiesError"
```

```
acceptAuthentication
Method-Type = "Return"
Method-Disabled = "FALSE"
Return-Disposition = "ACCEPT"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: ACCEPT for user ${packet.Base-User-
Name}@${packet.User-Realm} authenticated on ${user.proxyname}"
```

```
rejectAuthentication
Method-Type = "Return"
Method-Disabled = "FALSE"
Return-Disposition = "REJECT"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: REJECT for user ${packet.Base-
```

Create the file `acct.pf` with the following content :

```
dumpAcct
Method-Type = "WriteDebug"
Method-Disabled = "FALSE"
Level-On-Success = "INFO"
Level-On-Failure = "INFO"
Level-On-Error = "INFO"
WriteDebug-Map = "${Request Variable Group}=${request.*};\n${Packet Variable
Group}=${packet.*};\n${User Variable Group}=${user.*};\n${Check Variable
Group}=${check.*};\n${Reply Variable Group}=${reply.*};"
```

Create the file `prepare.pf` with the following content :

```
setWorkingVars
Method-Type = "ReadWrite"
Method-On-Success = "checklocal"
Method-On-Failure = "error:discardPolicyError"
Method-On-Error = "error:discardPolicyError"
Method-Disabled = "FALSE"
Level-On-Success = "INFO"
Level-On-Failure = "INFO"
Level-On-Error = "INFO"
ReadWrite-Map = "${user.nrplip} := \192.168.10.10\";\n${user.nrplport} :=
\1812\";\n${user.nrplsecret} := \some-secret\";\n${user.nrplretry} :=
\1\";\n${user.nrpltimeout} := \7000\";\n${user.nrplip} :=
\192.168.10.20\";\n${user.nrpl2port} := \1812\";\n${user.nrpl2secret} := \some-
secret\";\n${user.nrpl2retry} := \1\";\n${user.nrpl2timeout} :=
\7000\";\n${user.localrealm} := \loreal.tld\";"
ReadWrite-NewUser = "FALSE"

checklocal
Method-Type = "Compare"
Method-On-Error = "error:discardPolicyError"
Method-Disabled = "FALSE"
```

```

Level-On-Success = "INFO"
Level-On-Failure = "INFO"
Level-On-Error = "INFO"
Compare-Input1 = "${packet.User-Realm[trim,toLower]}"
Compare-Input2 = "${user.user.localrealm[trim,toLower]}"
Compare-Operator = "=="
Method-On-Failure = "setProxyVars"
Method-On-Success = "local:checkEAP"

```

setProxyVars

```

Method-Type = "ReadWrite"
Method-On-Success = "proxy:doProxy1"
Method-On-Failure = "error:discardPolicyError"
Method-On-Error = "error:discardPolicyError"
Method-Disabled = "FALSE"
Level-On-Success = "INFO"
Level-On-Failure = "INFO"
Level-On-Error = "INFO"
ReadWrite-Map = "${user.proxy1ip} := ${user.nrp1ip};\n${user.proxy1port} :=
${user.nrp1port};\n${user.proxy1secret} :=
${user.nrp1secret};\n${user.proxy1retry} :=
${user.nrp1retry};\n${user.proxy1timeout} :=
${user.nrp1timeout};\n${user.proxy2ip} := ${user.nrp2ip};\n${user.proxy2port} :=
${user.nrp2port};\n${user.proxy2secret} :=
${user.nrp2secret};\n${user.proxy2retry} :=
${user.nrp2retry};\n${user.proxy2timeout} :=
${user.nrp2timeout};\n${user.proxyname} := \"Proxy RADIUS\";";
ReadWrite-NewUser = "FALSE"

```

In the file above enter the correct info about the proxy servers and (for IdP case ONLY) for the local realm:

- \${user.nrp\*} – enter the correct info about the proxy servers (IP addresses, ports, ...)
- \${user.localrealm} – enter the correct info about the local realm (IdP ONLY)

Create the file local.pf with the following content :

checkEAP

```

Method-Type = "Compare"
Method-Disabled = "FALSE"
Level-On-Success = "INFO"
Level-On-Failure = "INFO"
Level-On-Error = "INFO"
Compare-Input1 = "${packet.EAP-Identity}"
Compare-Input2 = ""
Compare-Operator = "!="
Method-On-Error = "error:discardPolicyError"
Method-On-Failure = "error:discardeAPError"
Method-On-Success = "doEAPTTLS"

```

doEAPTTLS

```

Method-Type = "AuthEapTtls"
Method-On-Error = "error:discardPolicyError"
Method-Disabled = "FALSE"
Level-On-Success = "INFO"
Level-On-Failure = "INFO"
Level-On-Error = "INFO"
AuthEapTtls-TunnelMethod = "setLocalVars"

```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```

AuthEapTtls-TunnelWriteMap                                     =
"${request.*}:=${tunnel.*};\n${user.localrealm} := ${user.localrealm};"
AuthEapTtls-RsaCertFile = "server.pem"
AuthEapTtls-RsaKeyPassword = "secret"
AuthEapTtls-DsaCertFile = ""
AuthEapTtls-DsaKeyPassword = ""
AuthEapTtls-TrustedFile = "trust.pem"
AuthEapTtls-FragmentSize = "253"
AuthEapTtls-CertificateMap = "${packet.X509-Serial-Number} = ${Serial-
Number};\n${packet.X509-Subject-DN} = ${subject-DN};\n${packet.X509-Issuer-DN} =
${Issuer-DN};"
AuthEapTtls-KeyMap = "${reply.MS-MPPE-Recv-Key} := ${1-
32[isRadius]};\n${reply.MS-MPPE-Send-Key} := ${33-64[isRadius]};\n${reply.EAP-
Master-Session-Key} := ${1-64[isDiameter]};"

setLocalVars
Method-Type = "ReadWrite"
Method-Disabled = "FALSE"
Level-On-Success = "INFO"
Level-On-Failure = "INFO"
Level-On-Error = "INFO"
ReadWrite-Map = "${user.LDAPIP} := \"192.168.100.1\";\n${user.LDAPport} :=
\"389\";\n${user.uid} := ${packet.Base-User-Name[trim,toLower]};\n"
ReadWrite-NewUser = "FALSE"
Method-On-Success = "checkRealm"
Method-On-Failure = "error:discardPolicyError"
Method-On-Error = "error:discardPolicyError"

checkRealm
Method-Type = "Compare"
Method-On-Error = "error:discardPolicyError"
Method-Disabled = "FALSE"
Level-On-Success = "INFO"
Level-On-Failure = "INFO"
Level-On-Error = "INFO"
Compare-Input1 = "${packet.User-Realm[trim,toLower]}"
Compare-Input2 = "${user.localrealm[trim,toLower]}"
Compare-Operator = "=="
Method-On-Failure = "rejectRealm"
Method-On-Success = "getRealmCountry"

getRealmCountry
Method-Type = "PatternMatch"
Method-Disabled = "FALSE"
Level-On-Success = "INFO"
Level-On-Failure = "INFO"
Level-On-Error = "INFO"
PatternMatch-SearchKey = "${packet.User-Realm[toLower]}"
PatternMatch-Mode = "REGEX"
PatternMatch-Operation = "MATCHES"
PatternMatch-Case = "(.+)\.\.(+)\$ ${user.realm}=${1};${user.country}=${2};"
PatternMatch-IgnoreCase = "TRUE"
PatternMatch-SingleLine = "FALSE"
PatternMatch-MultiLine = "FALSE"
PatternMatch-Extended = "FALSE"

```



```
Method-On-Success = "checkPassword"  
Method-On-Failure = "rejectRealm"  
Method-On-Error = "erorr:discardPolicyError"
```

#### checkPassword

```
Method-Type = "Compare"  
Method-On-Failure = "rejectPassword"  
Method-On-Error = "erorr:discardPolicyError"  
Method-Disabled = "FALSE"  
Level-On-Success = "INFO"  
Level-On-Failure = "INFO"  
Level-On-Error = "INFO"  
Compare-Input1 = "${request.password}"  
Compare-Input2 = ""  
Compare-Operator = "!="  
Method-On-Success = "doLDAP"
```

#### doLDAP

```
Method-Type = "Ldap"  
Method-Disabled = "FALSE"  
Level-On-Success = "INFO"  
Level-On-Failure = "INFO"  
Level-On-Error = "INFO"  
Ldap-ServerAddress = "${user.LDAPIP}:${user.LDAPport}"  
Ldap-Version = "3"  
Ldap-BindDn = "uid=${user.uid},dc=${user.realm},dc=${user.country}"  
Ldap-BindPassword = "${request.password}"  
Ldap-Operation = "BIND"  
Ldap-SearchBase = "ou=People, o=${packet.User-Realm[escape]}"  
Ldap-SearchScope = "SCOPE_ONE"  
Ldap-SearchFilter = "(mail=${packet.Base-User-Name[escape]}@*)" "  
Ldap-MaxSearchResults = "1"  
Ldap-ConnectionLimit = "2147483647"  
Ldap-BindTimeout = "600000"  
Ldap-ConnectionTimeout = "10000"  
Ldap-OperationTimeout = "10000"  
Ldap-CacheConnections = "TRUE"  
Ldap-AuthFailureIsError = "FALSE"  
Ldap-NewUser = "FALSE"  
Ldap-StartTls = "FALSE"  
Method-On-Error = "erorr:discardLDAPError"  
Method-On-Success = "acceptAuthentication"  
Method-On-Failure = "rejectAuthentication"
```

#### rejectPassword

```
Method-Type = "Return"  
Method-Disabled = "FALSE"  
Level-On-Success = "INFO"  
Level-On-Failure = "INFO"  
Level-On-Error = "INFO"  
Return-Disposition = "REJECT"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: REJECT for user ${packet.Base-User-Name}@${packet.User-Realm} no valid password"
```

```
rejectRealm
    Method-Type = "Return"
    Method-Disabled = "FALSE"
    Level-On-Success = "INFO"
    Level-On-Failure = "INFO"
    Level-On-Error = "INFO"
    Return-Disposition = "REJECT"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH: REJECT for user ${packet.Base-User-
Name}@${packet.User-Realm} no valid realm"

acceptAuthentication
    Method-Type = "Return"
    Method-Disabled = "FALSE"
    Level-On-Success = "INFO"
    Level-On-Failure = "INFO"
    Level-On-Error = "INFO"
    Return-Disposition = "ACCEPT"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH: ACCEPT for user ${packet.Base-User-
Name}@${packet.User-Realm} authenticated on LDAP"

rejectAuthentication
    Method-Type = "Return"
    Method-Disabled = "FALSE"
    Level-On-Success = "INFO"
    Level-On-Failure = "INFO"
    Level-On-Error = "INFO"
    Return-Disposition = "REJECT"
    Return-LogLevel = "INFO"
    Return-LogMessage = "AUTH: REJECT for user ${packet.Base-User-
Name}@${packet.User-Realm} authenticated on LDAP"
```

In the file above enter (for IdP case ONLY) the correct info about the local IdM system:  
\${user.LDAP\*} enter the correct info about the local LDAP server (IdM system)

Additional files needed (due to the use of EAP):

server.pem - server certificate  
trust.pem – root certificate

## A.4.2 National VitalAAA Server (ver. 5.0.10)

Set-up of a (national) RADIUS proxy server.

The following configuration files have to be changed:

- server\_properties
- method\_dispatch
- clients

In addition the following files have to be created.

- aaa.pf
- error.pf

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

- proxy.pf
- prepare.pf

Changes in the `server_properties` file:

```
Radius-Acct-Address = "*:1813"  
Radius-Auth-Address = "*:1812"  
Database-Address = "0"  
Radius-CharSet = UTF8  
Delimiter-Precedence = "@"  
Suffix-Delimiters = "@"
```

Changes in the `method_dispatch` file:

```
radius      Auth 1      prepare      setWorkingVars  
radius      acct 4      aaa      dropRadiusAcct
```

Add the the lines with the eduroam proxy server and the local RADIUS servers to the `clients` file:

```
192.87.106.34      <eduroam_secret>  
130.225.242.109   <eduroam_secret>  
<192.168.1.10>    <local_server_secret>  
<192.168.1.20>    <local_server_secret>  
...
```

Create the file `aaa.pf` with the following content:

```
#-----  
# This file specifies the processing steps (methods) to take for  
# each AAA request.  The initial method is selected  
# by rules in the method_dispatch file.  
#-----  
# Revision $Id: aaa.pf,v 1.1 2006/03/12 02:40:42 glenn Exp $  
#-----  
dropStateServerAuth  
    Method-Type = "Return"  
    Return-Disposition = "DISCARD"  
    Return-LogLevel = "INFO"  
    Return-LogMessage = "Discarding StateServer Auth Request"  
  
dropStateServerAcct  
    Method-Type = "Return"  
    Return-Disposition = "DISCARD"  
    Return-LogLevel = "INFO"  
    Return-LogMessage = "Discarding StateServer Acct Request"  
  
dropRadiusAuth  
    Method-Type = "Return"  
    Return-Disposition = "DISCARD"  
    Return-LogLevel = "INFO"  
    Return-LogMessage = "Discarding RADIUS Auth Request"  
  
dropRadiusAcct  
    Method-Type = "Return"  
    Method-Disabled = "FALSE"  
    Level-On-Success = "INFO"  
    Level-On-Failure = "INFO"  
    Level-On-Error = "INFO"
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
Return-Disposition = "DISCARD"  
Return-LogLevel = "INFO"  
Return-LogMessage = "ACCN: DISCARD Client ${packet.Source-Address} sent  
accounting"
```

```
dropDiameterAuth  
Method-Type = "Return"  
Return-Disposition = "ERROR"  
Return-LogLevel = "INFO"  
Return-LogMessage = "Discarding Diameter Auth Request"
```

```
dropDiameterAcct  
Method-Type = "Return"  
Return-Disposition = "ERROR"  
Return-LogLevel = "INFO"  
Return-LogMessage = "Discarding Diameter Acct Request"
```

Create the file `error.pf` with the following content:

```
discardUnknownCountry  
Method-Type = "Return"  
Method-Disabled = "FALSE"  
Return-Disposition = "DISCARD"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: DISCARD unknown country ${user.country}"
```

```
discardEmptyCountry  
Method-Type = "Return"  
Method-Disabled = "FALSE"  
Return-Disposition = "DISCARD"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: DISCARD empty country ${packet.User-Realm}"
```

```
discardPolicyError  
Method-Type = "Return"  
Method-Disabled = "FALSE"  
Return-Disposition = "DISCARD"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: DISCARD policy error ${packet.Last-Disposition-  
Message}"
```

```
discardProxiesError  
Method-Type = "Return"  
Method-Disabled = "FALSE"  
Return-Disposition = "DISCARD"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: DISCARD proxies error ${user.proxyname}"
```

Create the file `proxy.pf` with the following content:

```
doProxy1  
Method-Type = "Radius"  
Method-Disabled = "FALSE"  
Method-On-Error = "doProxy2"  
Method-On-Success = "acceptAuthentication"  
Method-On-Failure = "rejectAuthentication"  
Radius-ServerAddress = "${user.proxy1ip}:${user.proxy1port}"
```

```
Radius-Secret = "${user.proxy1secret}"  
Radius-Dictionary = "#default"  
Radius-Timeout = "${user.proxy1timeout}"  
Radius-Retries = "${user.proxy1retry}"  
Radius-RequestMap = "${*}=${request.*};"  
Radius-PacketType = "${packet.Packet-Type}"  
Radius-ClientAddress = "*"   
Radius-CharSet = "UTF8"  
Radius-InauthenticFailure = "FALSE"  
Radius-CheckAuthenticator = "TRUE"  
Radius-RemoveTrailingNul = "TRUE"  
Radius-AppendTrailingNul = "FALSE"  
Radius-StrictEncoding = "FALSE"  
Radius-CopyMode = "FALSE"  
Radius-Mib = "AUTO"  
Radius-RecvBufferSize = "8192"  
Radius-SendBufferSize = "8192"  
Radius-SuccessMap = "${reply.*}=${*};"
```

#### doProxy2

```
Method-Type = "Radius"  
Method-Disabled = "FALSE"  
Method-On-Success = "acceptAuhentication"  
Method-On-Failure = "rejectAuhentication"  
Radius-ServerAddress = "${user.proxy2ip}:${user.proxy2port}"  
Radius-Secret = "${user.proxy2secret}"  
Radius-Dictionary = "#default"  
Radius-Timeout = "${user.proxy2timeout}"  
Radius-Retries = "${user.proxy2retry}"  
Radius-RequestMap = "${*}=${request.*};"  
Radius-PacketType = "${packet.Packet-Type}"  
Radius-ClientAddress = "*"   
Radius-CharSet = "UTF8"  
Radius-InauthenticFailure = "FALSE"  
Radius-CheckAuthenticator = "TRUE"  
Radius-RemoveTrailingNul = "TRUE"  
Radius-AppendTrailingNul = "FALSE"  
Radius-StrictEncoding = "FALSE"  
Radius-CopyMode = "FALSE"  
Radius-Mib = "AUTO"  
Radius-RecvBufferSize = "8192"  
Radius-SendBufferSize = "8192"  
Radius-SuccessMap = "${reply.*}=${*};"  
Method-On-Error = "error:discardProxiesError"
```

#### acceptAuhentication

```
Method-Type = "Return"  
Method-Disabled = "FALSE"  
Return-Disposition = "ACCEPT"  
Return-LogLevel = "INFO"  
Return-LogMessage = "AUTH: ACCEPT for user ${packet.Base-User-  
Name}@${packet.User-Realm} authenticated on ${user.proxyname}"
```

#### rejectAuhentication

```
Method-Type = "Return"
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```

Method-Disabled = "FALSE"
Return-Disposition = "REJECT"
Return-LogLevel = "INFO"
Return-LogMessage = "AUTH: REJECT for user ${packet.Base-User-Name}@${packet.User-Realm} authenticated on ${user.proxynome}"

```

Create the file `prepare.pf` with the following content:

```

setWorkingVars
    Method-Type = "ReadWrite"
    Method-Disabled = "FALSE"
    Method-On-Success = "getCountry"
    ReadWrite-Map = <<
${user.erplip} := "192.87.106.34";
${user.erplport} := "1812";
${user.erplsecret} := "some-secret";
${user.erplretry} := "1";
${user.erpltimeout} := "7000";
${user.erp2ip} := "130.225.242.109";
${user.erp2port} := "1812";
${user.erp2secret} := "some-secret";
${user.erp2retry} := "1";
${user.erp2timeout} := "7000";
${user.loplip} := "192.168.200.1";
${user.loplport} := "1812";
${user.loplsecret} := "local-some-secret";
${user.loplretry} := "1";
${user.lopltimeout} := "7000";
${user.lop2ip} := "192.168.200.2";
${user.lop2port} := "1812";
${user.lop2secret} := "local-some-secret";
${user.lop2retry} := "1";
${user.lop2timeout} := "7000";
${user.eduroamcountry} :=
"at*bg*hr*cz*dk*ee*fr*fi*de*gr*hu*it*ir*lv*lt*lu*mt*nl*no*pl*pt*ro*si*es*uk*ch";
${user.localcountry} := "*lo";
>>

    ReadWrite-NewUser = "FALSE"
    Method-On-Failure = "erorr:discardPolicyError"
    Method-On-Error = "erorr:discardPolicyError"

```

`getCountry`

```

Method-Type = "PatternMatch"
Method-Disabled = "FALSE"
Method-On-Success = "setUserCountry"
PatternMatch-SearchKey = "${packet.User-Realm}"
PatternMatch-Mode = "REGEX"
PatternMatch-Operation = "MATCHES"
PatternMatch-Case = "(+)\.\.(+)\$ ${user.country} = ${2};"
PatternMatch-IgnoreCase = "TRUE"
PatternMatch-SingleLine = "FALSE"
PatternMatch-MultiLine = "FALSE"
PatternMatch-Extended = "FALSE"
Method-On-Failure = "erorr:discardEmptyCountry"
Method-On-Error = "erorr:discardPolicyError"

```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
setUserCountry
    Method-Type = "ReadWrite"
    Method-Disabled = "FALSE"
    Method-On-Success = "checkeduroam"
    ReadWrite-Map = "${user.countryf} := \"*${user.country[toLowerCase]}\";"
    ReadWrite-NewUser = "FALSE"
    Method-On-Failure = "error:discardPolicyError"
    Method-On-Error = "error:discardPolicyError"

checkeduroam
    Method-Type = "Compare"
    Method-Disabled = "FALSE"
    Method-On-Failure = "checklocal"
    Method-On-Success = "setProxyVarseduroam"
    Compare-Input1 = "${user.eduroamcountry}"
    Compare-Input2 = "${user.countryf}"
    Compare-Operator = "contains"
    Method-On-Error = "error:discardPolicyError"

checklocal
    Method-Type = "Compare"
    Method-Disabled = "FALSE"
    Method-On-Success = "setProxyVarslocal"
    Compare-Input1 = "${user.localcountry}"
    Compare-Input2 = "${user.countryf}"
    Compare-Operator = "=="
    Method-On-Failure = "error:discardUnknownCountry"
    Method-On-Error = "error:discardPolicyError"

setProxyVarseduroam
    Method-Type = "ReadWrite"
    Method-Disabled = "FALSE"
    ReadWrite-Map = <<
${user.proxy1ip} := ${user.erplip};
${user.proxy1port} := ${user.erplport};
${user.proxy1secret} := ${user.erplsecret};
${user.proxy1retry} := ${user.erplretry};
${user.proxy1timeout} := ${user.erpltimeout};
${user.proxy2ip} := ${user.erp2ip};
${user.proxy2port} := ${user.erp2port};
${user.proxy2secret} := ${user.erp2secret};
${user.proxy2retry} := ${user.erp2retry};
${user.proxy2timeout} := ${user.erp2timeout};
${user.proxyname} := "eduroam";
>>
    ReadWrite-NewUser = "FALSE"
    Method-On-Failure = "error:discardPolicyError"
    Method-On-Error = "error:discardPolicyError"
    Method-On-Success = "proxy:doProxy1"

setProxyVarslocal
    Method-Type = "ReadWrite"
    Method-Disabled = "FALSE"
    ReadWrite-Map = <<
${user.proxy1ip} := ${user.loplip};
```

```
{user.proxy1port} := {user.lop1port};
{user.proxy1secret} := {user.lop1secret};
{user.proxy1retry} := {user.lop1retry};
{user.proxy1timeout} := {user.lop1timeout};
{user.proxy2ip} := {user.lop2ip};
{user.proxy2port} := {user.lop2port};
{user.proxy2secret} := {user.lop2secret};
{user.proxy2retry} := {user.lop2retry};
{user.proxy2timeout} := {user.lop2timeout};
{user.proxyname} := "local";
>>

ReadWrite-NewUser = "FALSE"
Method-On-Failure = "error:discardPolicyError"
Method-On-Error = "error:discardPolicyError"
Method-On-Success = "proxy:doProxy1"
```

In the file above enter the correct info about the eduroam and local servers. Also enter the list of TLDs participating in eduroam:

```
{user.erp*}- enter the correct info about the eduroam servers (IP addresses, ports, ...)
{user.lop*}- enter the correct info about the local RADIUS servers (IP addresses, ports, ...)
{user.eduroamcountry} - enter the list of TLDs participating in eduroam
{user.localcountry} - enter the local TLD
```

## A.5 Microsoft Internet Authentication Service server as institutional server

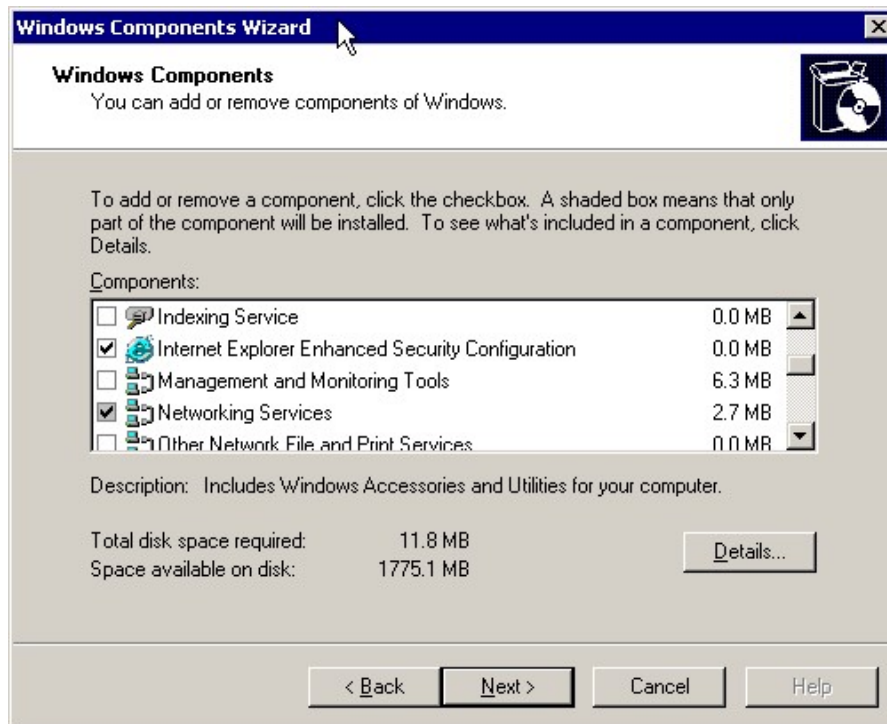
Internet Authentication Service (IAS) in Microsoft Windows Server 2003, Standard Edition; Windows Server 2003, Enterprise Edition; and Windows Server 2003, Datacenter Edition is the Microsoft implementation of a Remote Authentication Dial-in User Service (RADIUS) server and proxy. You can configure IAS in Windows Server 2003, Standard Edition, with a maximum of 50 RADIUS clients and a maximum of 2 remote RADIUS server groups. You can define a RADIUS client using a fully qualified domain name or an IP address, but you cannot define groups of RADIUS clients by specifying an IP address range. In the Enterprise and Datacenter Edition of Windows Server 2003 these limitations do not exist.

### A.5.1 Installing IAS

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

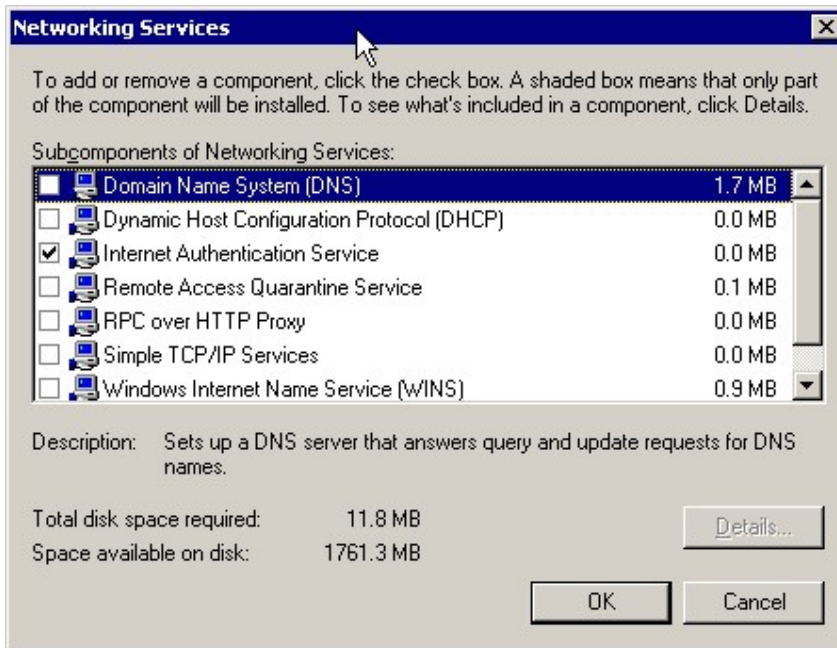


Windows Server 2003 does not install IAS in the default installation. The IAS must be installed separately later



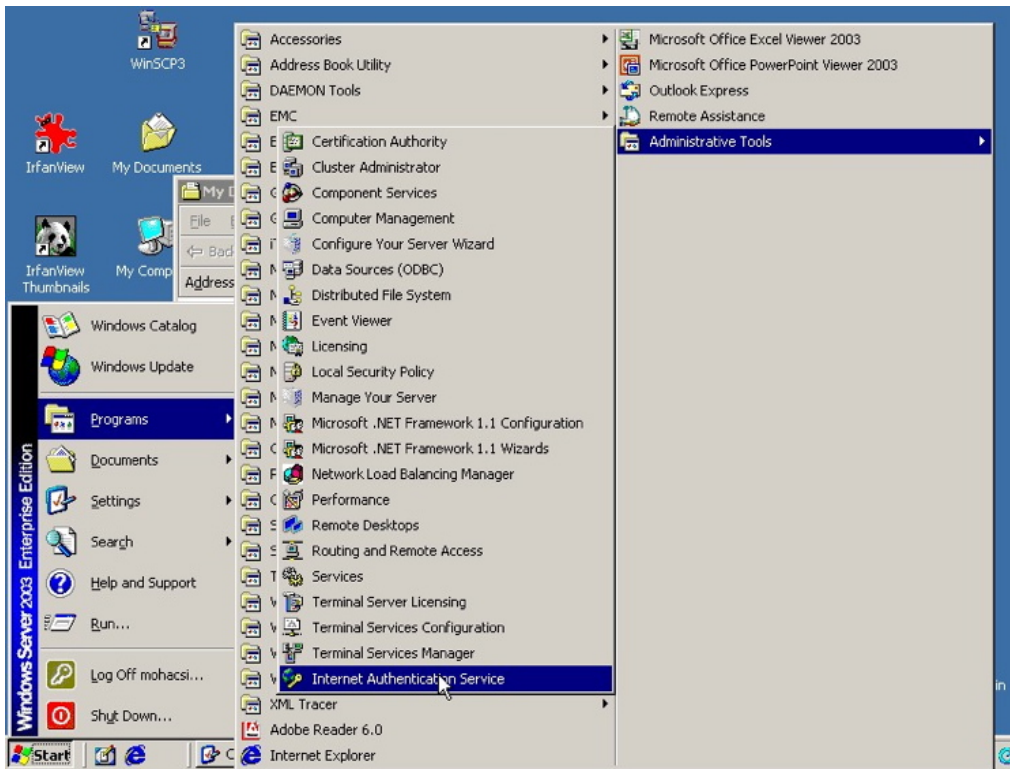
from windows components under the **Networking Services**:

The **Internet Authentication Service** must be selected:

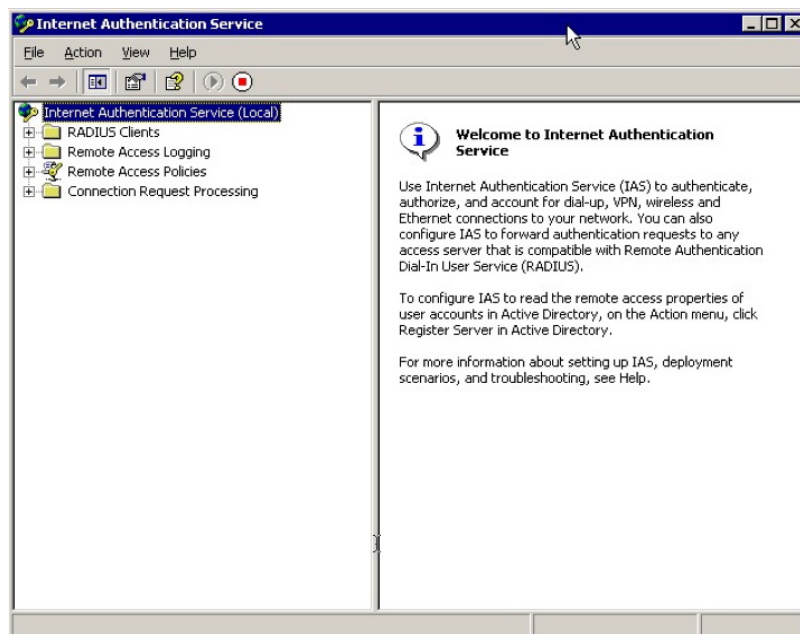


And wait for the installation to be finished. The IAS administrative console can be found under the **Administrative Tools**:

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8



After clicking on “Internet Authentication Service“ in the start menu the IAS console will start:

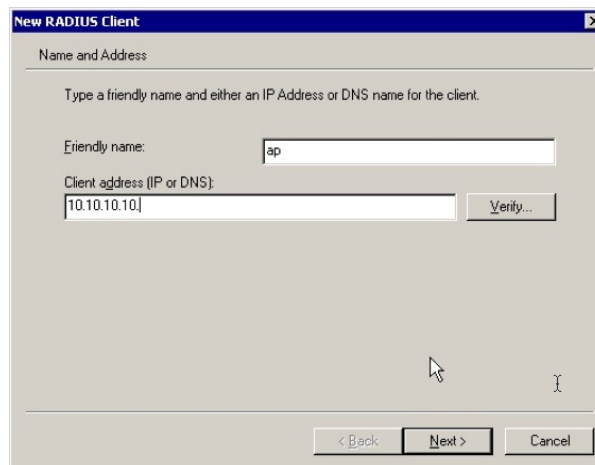


Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

## A.5.2 Configuring IAS to act as a university RADIUS server in the eduroam hierarchy

### A.5.2.1 Configuring IAS for access points and upstream proxies

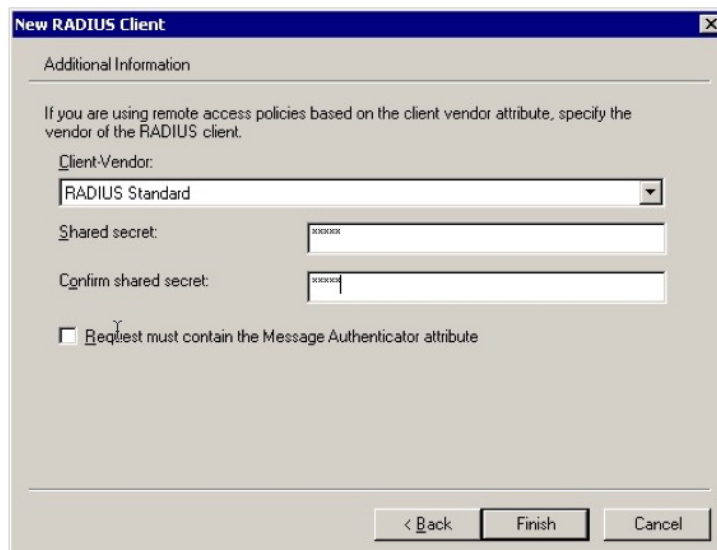
For each access point and upstream proxy (i.e. national eduroam RADIUS server) the parameters of the RADIUS Clients must be configured. When you add a new access point a wizard will start asking for the name and the IP address of the RADIUS client (i.e. Access Point, switch, or upstream RADIUS proxy):



The screenshot shows the 'New RADIUS Client' dialog box with the 'Name and Address' tab selected. The dialog contains the following fields and controls:

- Name and Address** (tab title)
- Instruction: "Type a friendly name and either an IP Address or DNS name for the client."
- Friendly name:** Text input field containing "ap".
- Client address (IP or DNS):** Text input field containing "10.10.10.10".
- Verify...** button next to the client address field.
- < Back**, **Next >**, and **Cancel** buttons at the bottom.

Then you have to specify the shared secret between the RADIUS client and your RADIUS server (IAS):



The screenshot shows the 'New RADIUS Client' dialog box with the 'Additional Information' tab selected. The dialog contains the following fields and controls:

- Additional Information** (tab title)
- Instruction: "If you are using remote access policies based on the client vendor attribute, specify the vendor of the RADIUS client."
- Client-Vendor:** Dropdown menu showing "RADIUS Standard".
- Shared secret:** Password input field with masked characters.
- Confirm shared secret:** Password input field with masked characters.
- Request must contain the Message Authenticator attribute**
- < Back**, **Finish**, and **Cancel** buttons at the bottom.

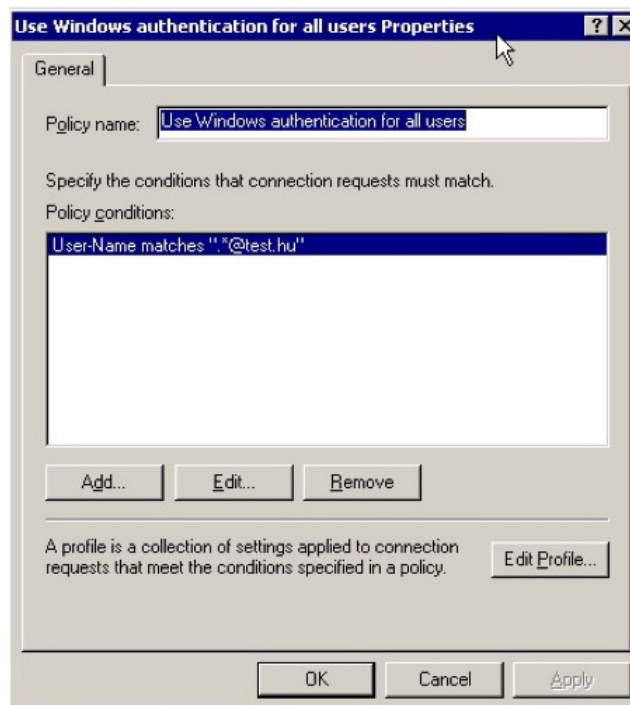
You can select various vendors of RADIUS clients, but in most of the cases you should use **RADIUS Standard**.

### A.5.2.2 Configuring Connection Request Processing Policy

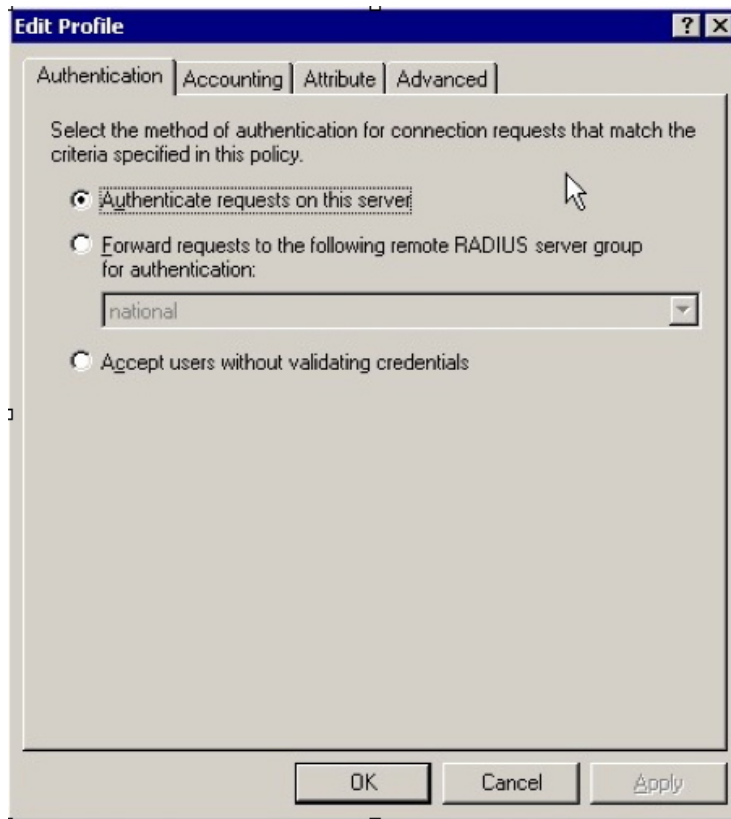
The realm processing should be configured to match eduroam hierarchy. First you have to configure a policy to catch local realms, then you have to configure a policy that forwards rest of the requests to your upstream proxy server.

### A.5.2.3 Configuring policy for local realm

You should configure a Connection Request Processing Policy, that captures all the User-Names that are used for access to local realms using the policy condition ".\*@yourrealm.tld".

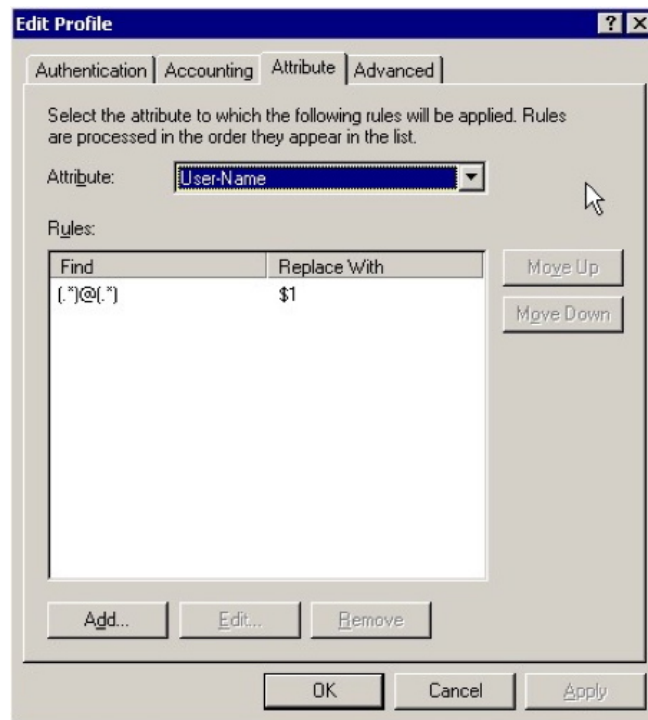


In this case the profile will be more complicated. The authentication should happen on the local server:



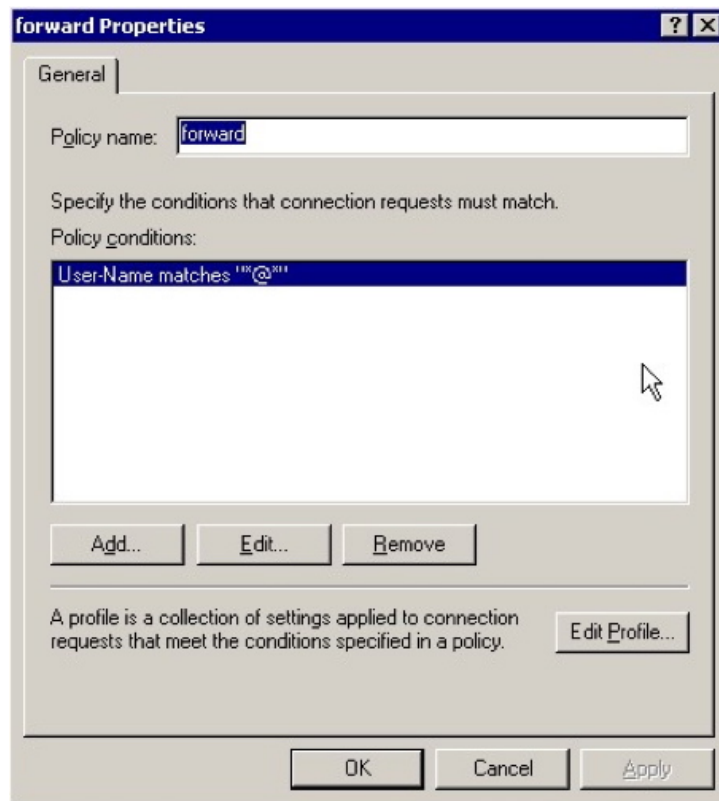
But the RADIUS attributes must be processed. In the case of a matching realm name the realm name must be stripped off:

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

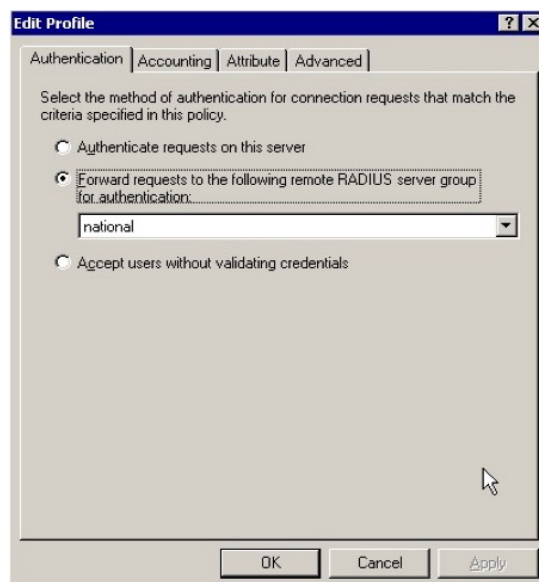


#### A.5.2.4 Configuring policy for upstream RADIUS proxy server

You should configure a Connection Request Processing Policy, that captures all the User-Names that are potentially used for roaming with the policy condition “.\*@.\*”.



Then you should edit the profile to be forwarded to the national proxy server:



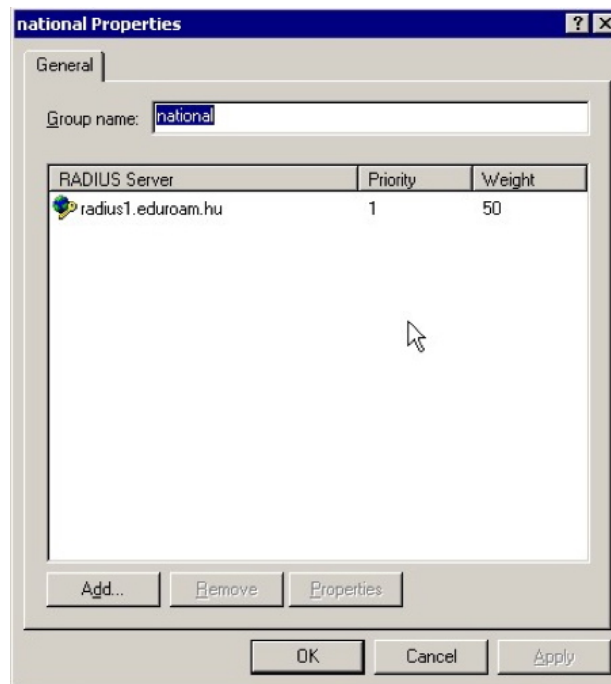
Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8



You should configure ~~first~~ the remote RADIUS server group first in order to be able to select from the list.

### A.5.3 Configuring remote RADIUS servers

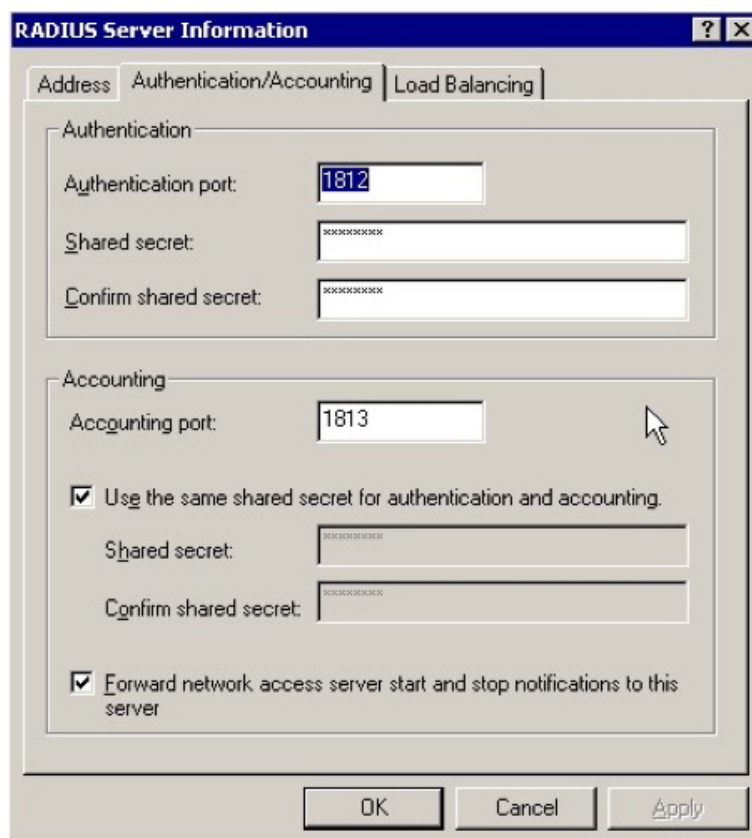
The national RADIUS proxy server must be added to the remote RADIUS server:



The remote RADIUS server address must be specified:



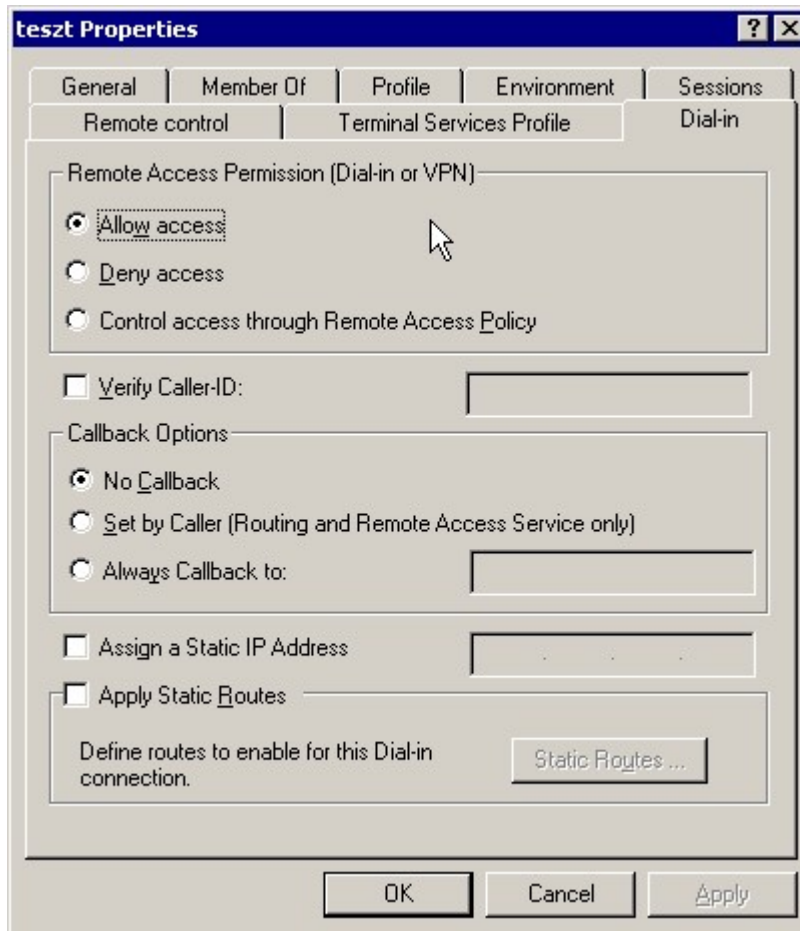
You have to enter the RADIUS server authentication port (usually 1812) and the shared secret of the remote RADIUS proxy server as well as the remote RADIUS server accounting port. You can specify different shared secrets for accounting if you wish:



#### **A.5.4 Configuring Domain Users to be able to use eduroam with their credentials to Windows Domain**

By default the users configured in the Windows Domain are not able to use their Windows Domain username and password to authenticate against IAS. This should be enabled in the Domain to allow access to Remote Access Permission. This can be done via the User Management interface or the Domain Manager interface with the following policy:

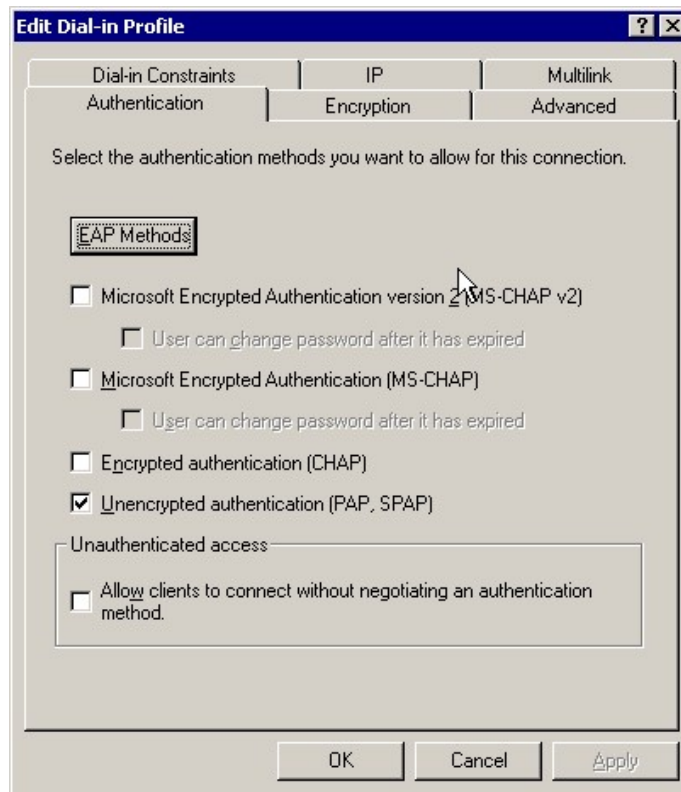
Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8



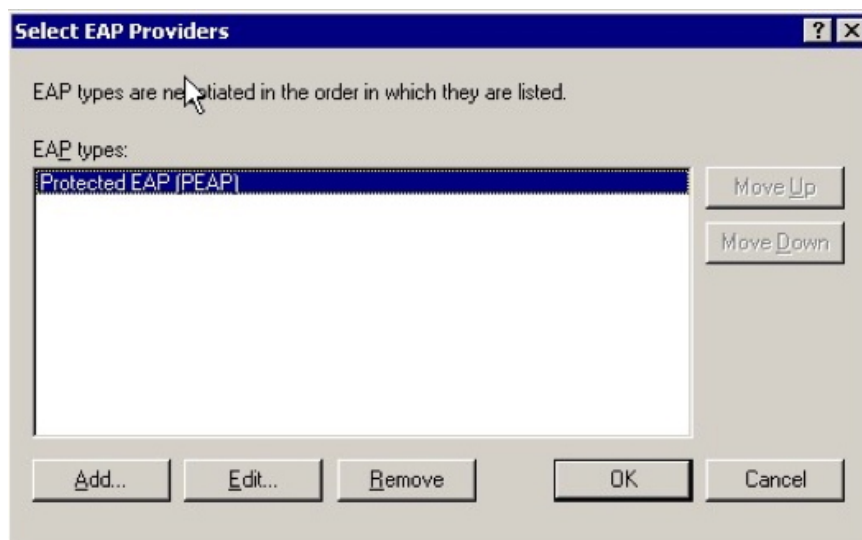
### A.5.5 Configuration of Authentication methods

The authentication methods should be configured in the **Remote Access Policies** under the **Profile** settings. The absolute minimum that needs to be enabled is PEAP under the EAP methods, but it is useful to have PAP as well, for debugging purpose – at least for certain accounts (e.g. for test accounts):

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

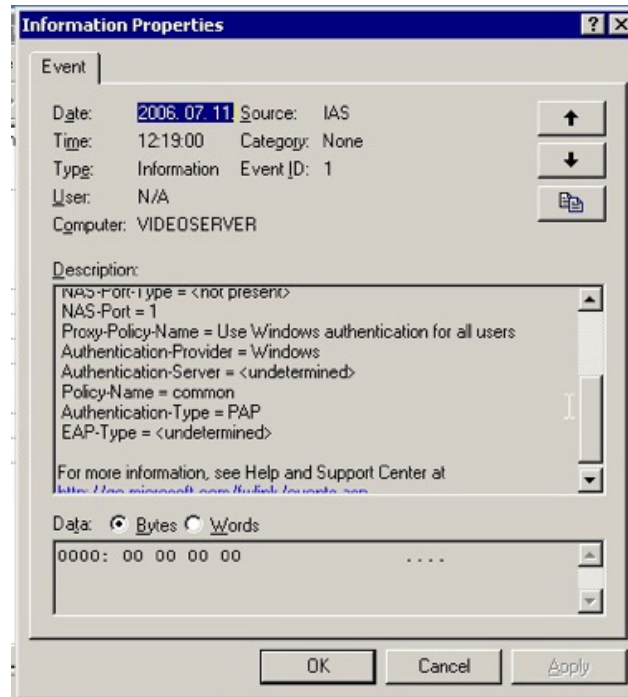


PEAP is the easiest way to deploy eduroam authentication method under Windows. Deploying EAP-TLS can be labour-intensive:



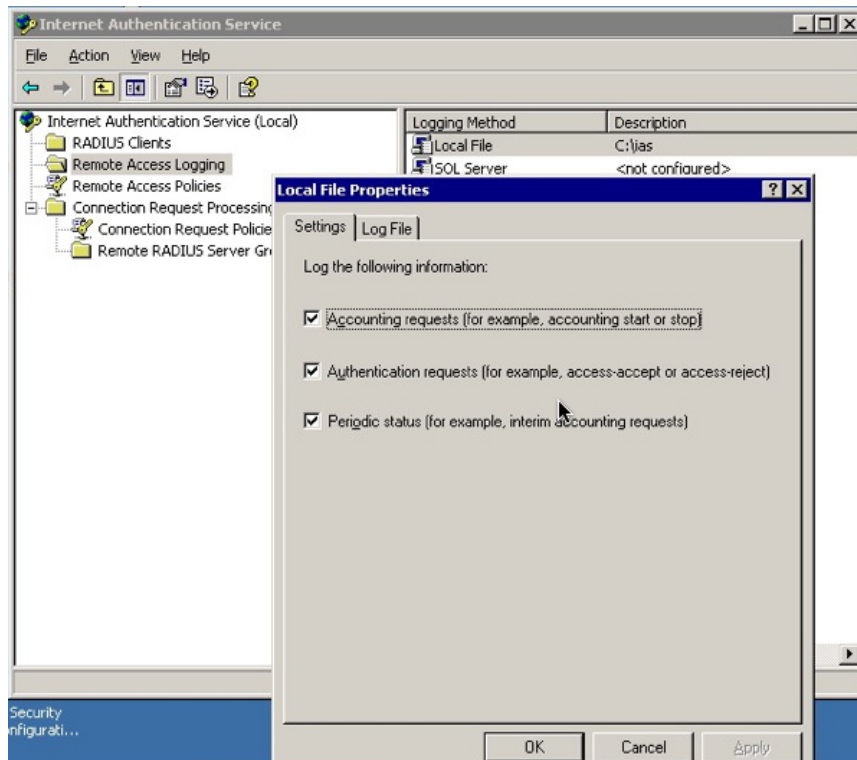
## A.5.6 Troubleshooting

The most useful information can be extracted from the Eventviewer:



But you can obtain these information as well from the log files:

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8



## A.5.7 References

IAS Resources: <http://technet2.microsoft.com/WindowsServer/en/Library/f6985d5d-d4c5-49e2-bbc7-385e105bfe281033.mspx?mfr=true>

Internet Authentication Service <http://technet2.microsoft.com/WindowsServer/en/Library/d98eb914-258c-4f0b-ad04-dc4db9e4ee631033.mspx?mfr=true>

IAS Pattern matching syntax: <http://technet2.microsoft.com/WindowsServer/en/Library/6e5ce48d-e662-435c-a74e-0dce305914ce1033.mspx?mfr=true>

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

## Appendix B Access Points

Complementing the main parts brief and explanatory configuration instructions for the Cisco Aironet 1200 series, this part of the appendix provides a complete configuration example that can be copy pasted for that Access Point. Later on the configuration for a Lancom L-54 as an alternative hardware solution is presented. This two Access Points are again by no means the only ones that can be used to implement eduroam. Alternative hardware solutions might work just as well, but as these two are in use by JRA5 participants, correct functionality has been proven.

### B.1 Cisco Aironet 1200 Series example setup

The configuration described in this section refers to the reference setup for all the VLAN's used.

#### List of configuration commands

**NOTE:** The following commands can be copy pasted into Access Point configuration mode (telnet or console access). The **bold** and *italic bold* text values need to be changed to match the implemented setup. The RADIUS Secret **MUST** match the secret configured on the RADIUS Software used and the Access Point's IP address (defined on the BVI interface) **MUST** be configured as an allowed Client in the used RADIUS Software. Probably some stuff from the default AP configuration like RADIUS groups, default SSID (tsunami), etc has to be deleted. To erase/delete configuration parameters just precede each command with the keyword ,no', e.g. with the command ,no ssid tsunami', the ssid tsunami will cease to exist.

```

version 12.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname ap1200
!
logging buffered warnings
logging monitor warnings
enable secret 0 <your super passwd for accessing configuration level>

```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8



```

!
ip subnet-zero
no ip domain lookup
!
!
aaa new-model
!
!
aaa group server radius radsrv
  server <your RADIUS Server IP address> auth-port 1812 acct-port 1813
!
aaa authentication login default local
aaa authentication login eap_methods group radsrv
aaa authentication login mac_methods local
aaa authorization exec default local
aaa authorization network default group radsrv
aaa accounting send stop-record authentication failure
aaa accounting session-duration ntp-adjusted
aaa accounting update newinfo periodic 15
aaa accounting exec default start-stop group radsrv
aaa accounting network default start-stop group radsrv
aaa accounting network acct_methods start-stop group radsrv
aaa accounting connection default start-stop group radsrv
aaa accounting system default start-stop group radsrv
aaa accounting resource default start-stop group radsrv
aaa nas port extended
aaa session-id unique
no dot11 igmp snooping-helper
!
dot11 ssid eduroam
  vlan 909
  authentication open eap eap_methods
  authentication network-eap eap_methods
  authentication key-management wpa optional
  accounting acct_methods
  guest-mode
!
dot11 ssid guest
  vlan 903
  authentication open
  accounting acct_methods
!
dot11 holdoff-time 60
dot11 location isocc PT cc 351 ac 21
dot11 ids eap attempts 32 period 8
dot11 network-map
dot11 arp-cache
!
!
username <your login> password 0 <your login password>
!
bridge irb
!
!
interface Dot11Radio0

```

```
no ip address
no ip route-cache
!
encryption vlan 906 mode ciphers aes-ccm tkip wep128
!
encryption vlan 909 mode ciphers aes-ccm tkip wep128
!
broadcast-key vlan 906 change 600 membership-termination capability-change
!
broadcast-key vlan 909 change 600 membership-termination capability-change
!
!
ssid eduroam
!
ssid guest
!
countermeasure tkip hold-time 0
speed ofdm separate
speed basic-5.5 6.0 9.0 basic-11.0 12.0 18.0 24.0 36.0 48.0 54.0
power local cck 50
power local ofdm 30
no power client local
power client 50
fragment-threshold 512
station-role root fallback shutdown
rts threshold 2312
beacon period 500
beacon dtim-period 1
no dot11 extension aironet
world-mode dot11d country PT indoor
no cdp enable
dot1x reauth-period 300
bridge-group 1
bridge-group 1 subscriber-loop-control
bridge-group 1 block-unknown-source
no bridge-group 1 source-learning
no bridge-group 1 unicast-flooding
!
interface Dot11Radio0.903
encapsulation dot1Q 903
no ip route-cache
no cdp enable
bridge-group 3
bridge-group 3 subscriber-loop-control
bridge-group 3 block-unknown-source
no bridge-group 3 source-learning
no bridge-group 3 unicast-flooding
bridge-group 3 spanning-disabled
!
interface Dot11Radio0.906
description - eduroam VLAN for local users
encapsulation dot1Q 906
no ip route-cache
no cdp enable
bridge-group 6
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```

bridge-group 6 subscriber-loop-control
bridge-group 6 block-unknown-source
no bridge-group 6 source-learning
no bridge-group 6 unicast-flooding
bridge-group 6 spanning-disabled
!
interface Dot11Radio0.909
description - eduroam VLAN for roamers (foreign users)
encapsulation dot1Q 909
no ip route-cache
no cdp enable
bridge-group 9
bridge-group 9 subscriber-loop-control
bridge-group 9 block-unknown-source
no bridge-group 9 source-learning
no bridge-group 9 unicast-flooding
bridge-group 9 spanning-disabled
!
interface FastEthernet0
no ip address
no ip route-cache
load-interval 30
duplex auto
speed auto
no cdp enable
!
interface FastEthernet0.902
description - Administrative VLAN
encapsulation dot1Q 902 native
no ip route-cache
no cdp enable
bridge-group 1
bridge-group 1 port-protected
no bridge-group 1 source-learning
bridge-group 1 spanning-disabled
!
interface FastEthernet0.903
description - guest VLAN - no internet connectivity on this network
encapsulation dot1Q 903
no ip route-cache
no cdp enable
bridge-group 3
no bridge-group 3 source-learning
bridge-group 3 spanning-disabled
!
interface FastEthernet0.906
description - eduroam VLAN for local students
encapsulation dot1Q 906
no ip route-cache
no cdp enable
bridge-group 6
no bridge-group 6 source-learning
bridge-group 6 spanning-disabled
!
interface FastEthernet0.909

```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
description - eduroam VLAN for roamers (foreign students)
encapsulation dot1Q 909
no ip route-cache
no cdp enable
bridge-group 9
no bridge-group 9 source-learning
bridge-group 9 spanning-disabled
!
interface BV11
description
ip address <your AP's IP address> <your AP's network mask>
no ip route-cache
!
no ip http server
no ip http secure-server
ip http help-path http://www.cisco.com/warp/public/779/smbiz/prodconfig/help/eag
ip radius source-interface BV11
!
no cdp run
radius-server attribute 8 include-in-access-req
radius-server attribute 32 include-in-access-req format %h
radius-server dead-criteria time 60 tries 10
radius-server host <your RADIUS server IP address> auth-port 1812 acct-port 1813
radius-server retransmit 2
radius-server deadtime 15
radius-server key 0 <your radius secret>
radius-server vsa send accounting
radius-server vsa send authentication
!
control-plane
!
bridge 1 route ip
!
!
!
line con 0
line vty 5 15
!
end
```

**IMPORTANT:** It is mandatory to save the configuration to the Access Point flash memory by executing the “write” command in the normal command prompt, or the AP will boot with the previous saved configuration (of the previous boot)

**Disclaimer:** The presented Access Point configuration lacks security measures to control management to the Access Point’s such as access-lists. Nevertheless the users associated to the access point (authenticated in SSID eduroam or just in the SSID guest) can’t access the managing interface (BVI) unless there is some VLAN routing elsewhere on the hotspot.

This configuration should be considered only as a starting point to make the access points eduroam compatible.

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

## B.2 LANCOM L-54 Series Access Points

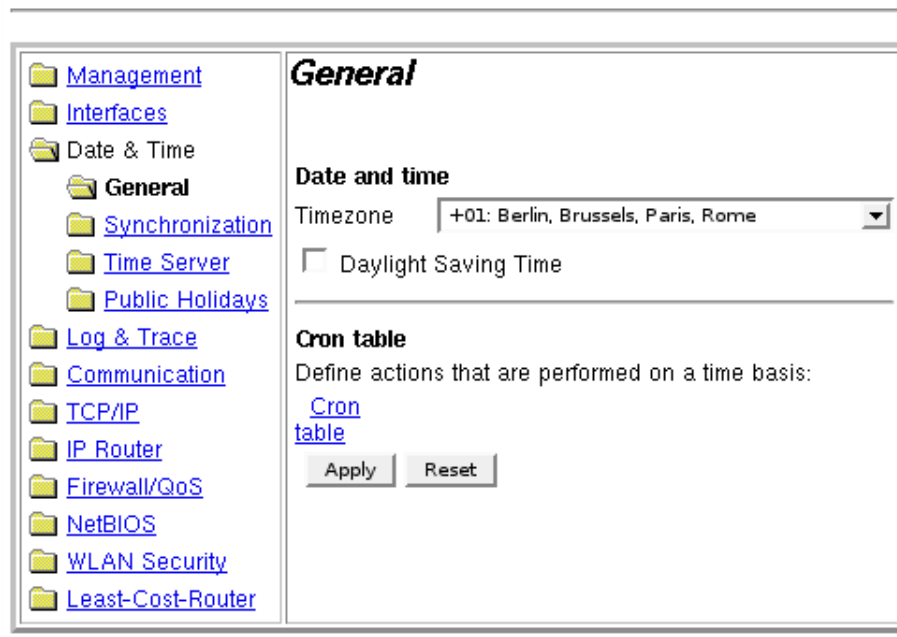
This series of Access Points offers a wide range of features for a mid-range price. One of the outstanding features in its price class is the ability to use ARP sniffing to determine a client's IP address even if it changes during a user session. Activating this feature fulfils the requirement for MAC to IP correlation from the confederation policy and obsoletes logging of DHCP leases.

The following steps are needed to set up eduroam on a Lancom L-54 access point. It describes the setup via the web interface and is current as of LCOS Version 6.10:

### B.2.1 NTP setup (confederation requirement: reliable timing source)

First select your Timezone under “Configuration - Date & Time – General“:

*(LANCOM L-54g Wireless 6.06.0012 / 27.03.2006)*



🕒 [11/10/2006 12:09](#)

👉 [Previous Page](#)    ↻ [Entry Page](#)    🏠 [LANCOM Systems Homepage](#)

Next choose “Synchronization” and check the radiobutton “Synchronize...”, then click on the link “Time Server” (NOT the menu “Time Server” on the left-hand side; this is only relevant if you want the AP to be the time server for its clients), click on add and enter your server details:

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

( LANCOM L-54g Wireless 6.06.0012 / 27.03.2006)

🕒 11/10/2006 11:11

🔍 [Previous Page](#) ↻ [Entry Page](#) 🏠 [LANCOM Systems Homepage](#)

## B.2.2 Logging

Select Configuration – Log &Trace – Syslog and check the box “Send information”, then click on “Syslog clients” and “add”, now add at least localhost: IP 127.0.0.1, activate all sources:

( LANCOM L-54g Wireless 6.06.0012 / 27.03.2006)

IP address	System Login	System time	Console login	Connections	Accounting	Administration	Router	Alert	Error	Warning	Information	Debug
<input checked="" type="checkbox"/> <input type="checkbox"/> 127.0.0.1	On	On	On	On	On	On	Off	On	On	On	On	On
<input checked="" type="checkbox"/> <input type="checkbox"/> 158.64	On	On	Off	On	Off	On	Off	On	On	On	On	On

🕒 11/10/2006 11:12

🔍 [Previous Page](#) ↻ [Entry Page](#) 🏠 [LANCOM Systems Homepage](#)

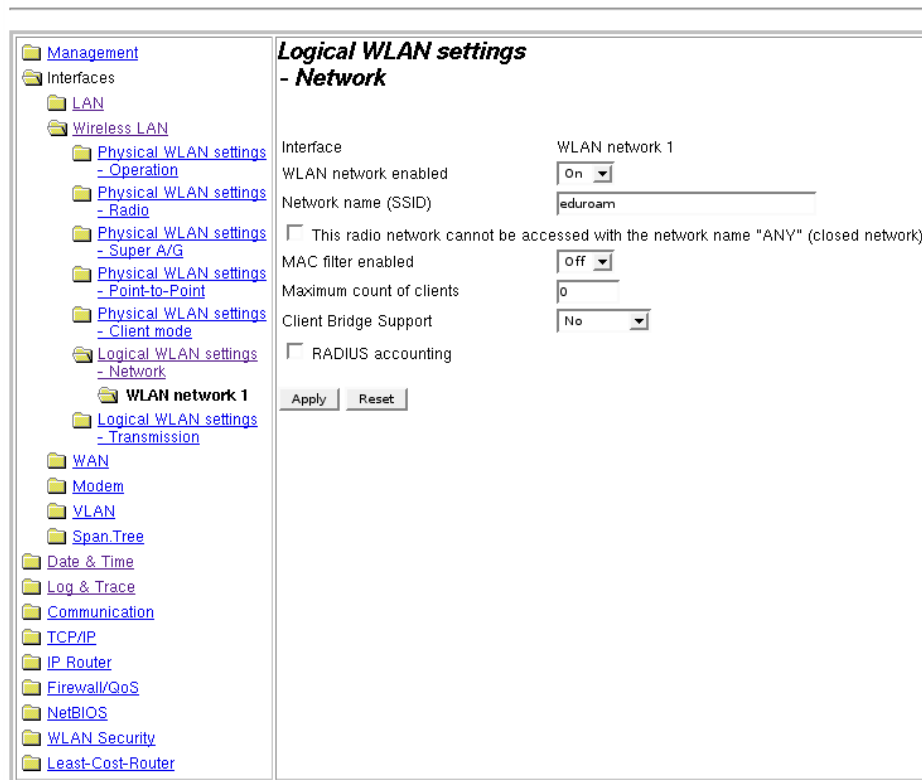
The logs that are collected with the localhost setting will show up under Expert Configuration – Status – TCP/IP - Syslog

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

### B.2.3 Configuring the SSID

Select Configuration – Wireless LAN – Logical WLAN setting – Network and click on one of the available slots, then set “WLAN network enabled” to “On” set the Network name (ssid) to eduroam and uncheck the box labelled "This radio network cannot be accessed with the network name "ANY" (closed network)", then set MAC filter enabled to Off, Maximum count of clients to 0 and Client Bridge support to No. Optionally you can check the box “RADIUS accounting”:

( LANCOM L-54g Wireless 6.06.0012 / 27.03.2006)



11/10/2006 11:14

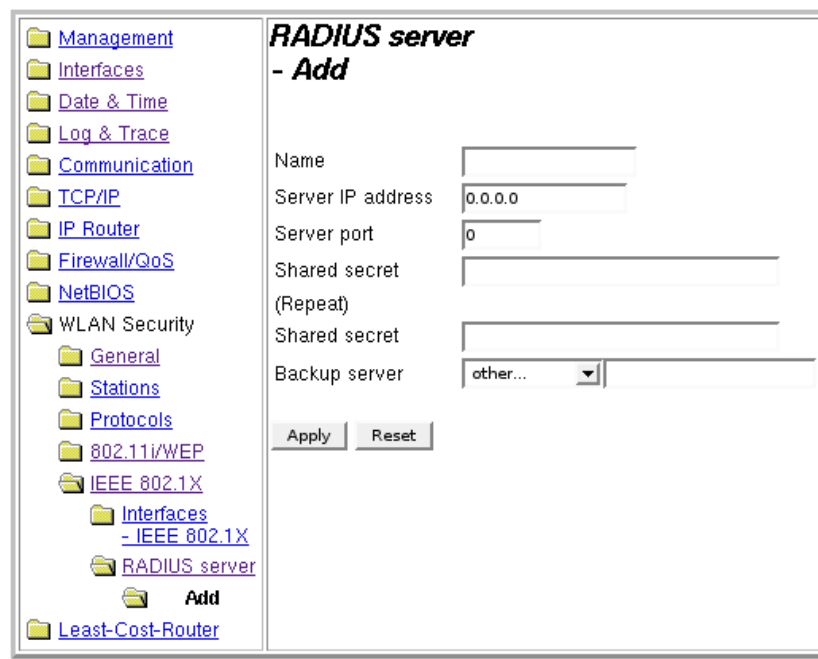
[Previous Page](#) [Entry Page](#) [LANCOM Systems Homepage](#)

### B.2.4 WPA Enterprise security

First configure the RADIUS server to use. Select Configuration – WLAN security – IEEE 802.1X – RADIUS server click on add and enter your server details:

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

(LANCOM L-54g Wireless 6.06.0012 / 27.03.2006)



The screenshot shows a web-based configuration interface for a LANCOM device. On the left is a navigation tree with folders for Management, Interfaces, Date & Time, Log & Trace, Communication, TCP/IP, IP Router, Firewall/QoS, NetBIOS, WLAN Security (containing General, Stations, Protocols, 802.11i/WEP, IEEE 802.1X, and RADIUS server), and Least-Cost-Router. The 'RADIUS server' folder is expanded, showing an 'Add' button. The main area is titled 'RADIUS server - Add' and contains the following fields: Name (empty), Server IP address (0.0.0.0), Server port (0), Shared secret (empty), (Repeat) (empty), Shared secret (empty), and Backup server (other... dropdown). At the bottom are 'Apply' and 'Reset' buttons.

🕒 11/10/2006 11:15

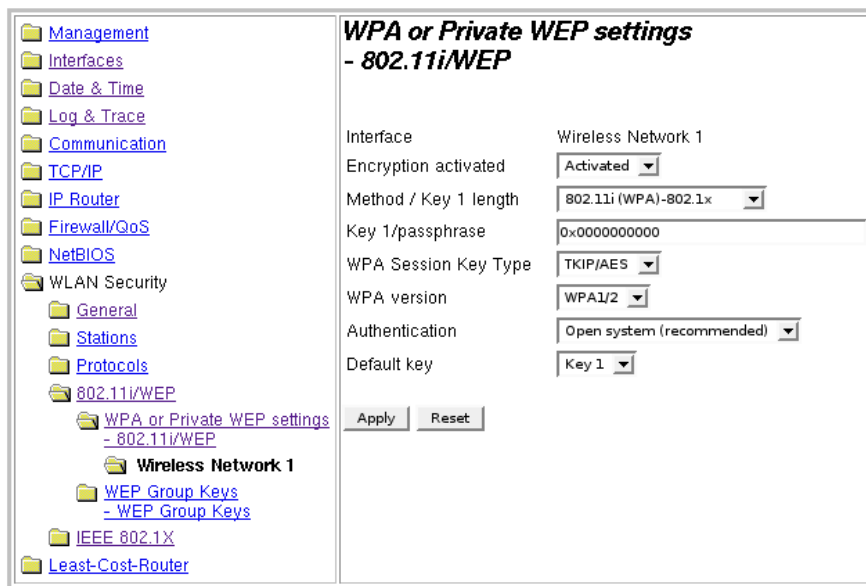
🔍 [Previous Page](#) ↻ [Entry Page](#) 🏠 [LANCOM Systems Homepage](#)

Next apply the RADIUS server and encryption scheme to the SSID eduoam. Select Configuration – WLAN security – 802.11i/WEP, click on WPA or Private WEP setting – 80211.i/WEP and click on the slot in which you previously configured the SSID eduoam. Then configure the following settings: “Encryption Activated” to “Activated”, “Method/Key 1 Length” to “802.11i(WPA)-802.1x”, “WPA Session Key Type” to “TKIP/AES” and “WPA Version” to “WPA1/2”. The other settings are irrelevant with WPA-Enterprise:

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8



(LANCOM L-54g Wireless 6.06.0012 / 27.03.2006)



The screenshot displays the LANCOM configuration interface for WPA or Private WEP settings. On the left is a navigation tree with folders for Management, Interfaces, Date & Time, Log & Trace, Communication, TCP/IP, IP Router, Firewall/QoS, NetBIOS, WLAN Security, and Least-Cost-Router. Under WLAN Security, there are sub-folders for General, Stations, Protocols, 802.11i/WEP, WEP Group Keys, and IEEE 802.1X. The 'WPA or Private WEP settings - 802.11i/WEP' folder is selected. The main area shows settings for 'Wireless Network 1':

Interface	Wireless Network 1
Encryption activated	Activated
Method / Key 1 length	802.11i (WPA)-802.1x
Key 1/passphrase	0x0000000000
WPA Session Key Type	TKIP/AES
WPA version	WPA1/2
Authentication	Open system (recommended)
Default key	Key 1

Buttons for 'Apply' and 'Reset' are located at the bottom of the settings area.

🕒 11/10/2006 11:16

👉 [Previous Page](#)   🔄 [Entry Page](#)   🏠 [LANCOM Systems Homepage](#)

## B.2.5 RADIUS accounting server (optional)

If RADIUS accounting for the eduroam SSID above is enabled, you have to configure a RADIUS server to send the accounting messages to. Go to “Expert Configuration” and select Setup – WLAN – RADIUS-Accounting and fill in the server details:

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

(LANCOM L-54g Wireless 6.06.0012 / 27.03.2006)








---

[Expert Configuration](#)

 [Setup](#)

 [WLAN](#)

## RADIUS-Accounting

 <a href="#">Server-Address</a>	158.64. 
 <a href="#">Accnt-Port</a>	1813
 <a href="#">Secret</a>	*
 <a href="#">Backup-Server-IP-Address</a>	0.0.0.0
 <a href="#">Backup-Accnt-Port</a>	1813
 <a href="#">Backup-Secret</a>	
 <a href="#">Client-Brg.-Handling</a>	All-Traffic
 <a href="#">Interim-Update-Period</a>	300
 <a href="#">Excluded-VLAN</a>	0

---

 [11/10/2006 11:17](#)

 [Previous Page](#)

 [Entry Page](#)

 [LANCOM Systems Homepage](#)

## Appendix C **Supplicants**

The main part of this document presented the usage of SecureW2 as a preconfigured supplicant for MS Windows, there are of course respective supplicants for all common Operating Systems, some of them are described in the following sections. These supplicants are either Open Source or integrated into the Operating System, so there are no licence fees to consider. The main difference between them is the degree to which they can be preconfigured. Whereas the first section of this appendix shows how to use a non-preconfigured SecureW2, followed by a brief explanation how to setup the Mac OS X integrated supplicant, the last part demonstrates how to pre-configure the wpa\_supplicant which can basically be used on either Unix or Windows environments because of its open source nature.

### C.1 **SecureW2**

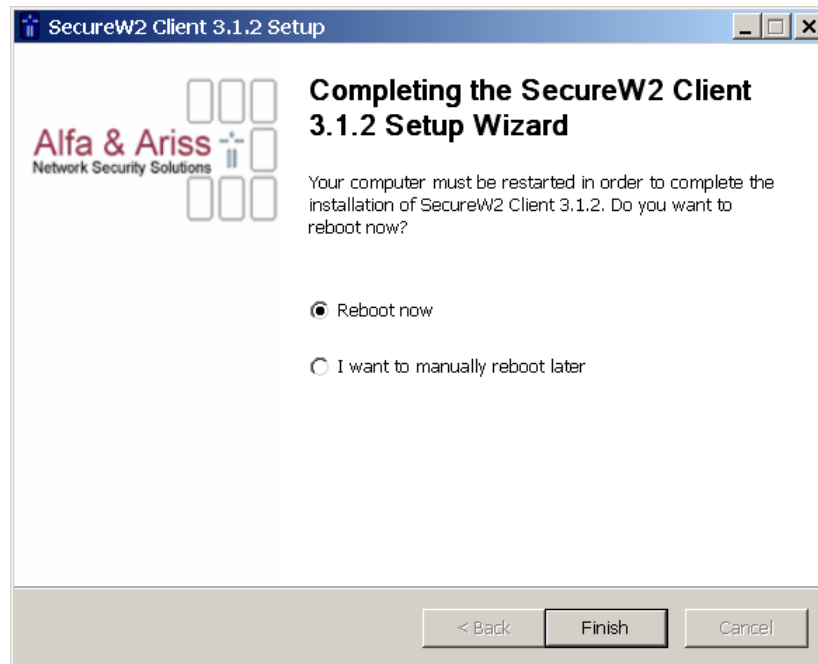
In addition to the SecureW2 part in the reference section, it is also possible to use a non-preconfigured SecureW2 supplicant to connect to eduroam. For testing purposes or for very small institutions where it does not seem worth the effort to prepare a preconfigured SecureW2 the following section illustrates the steps necessary.

Start the SecureW2 installer which can be downloaded from:

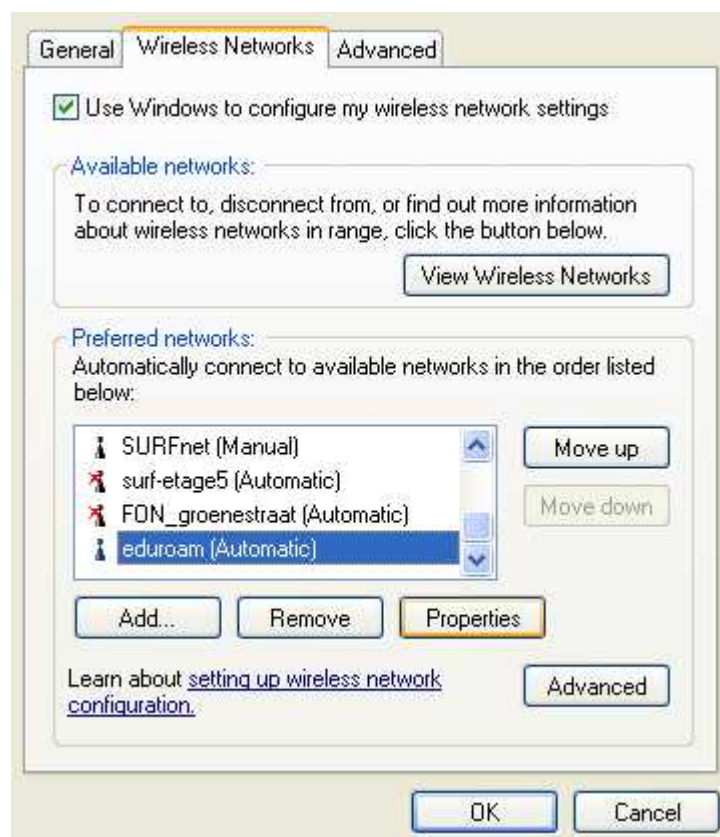
<http://securew2.alfa-ariss.com/uk/download/index.htm>

After starting the installer click on „next“, accept the EULA and click on „install“ in the options panel. After successful installation you should see the following screenshot asking you to reboot:

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8



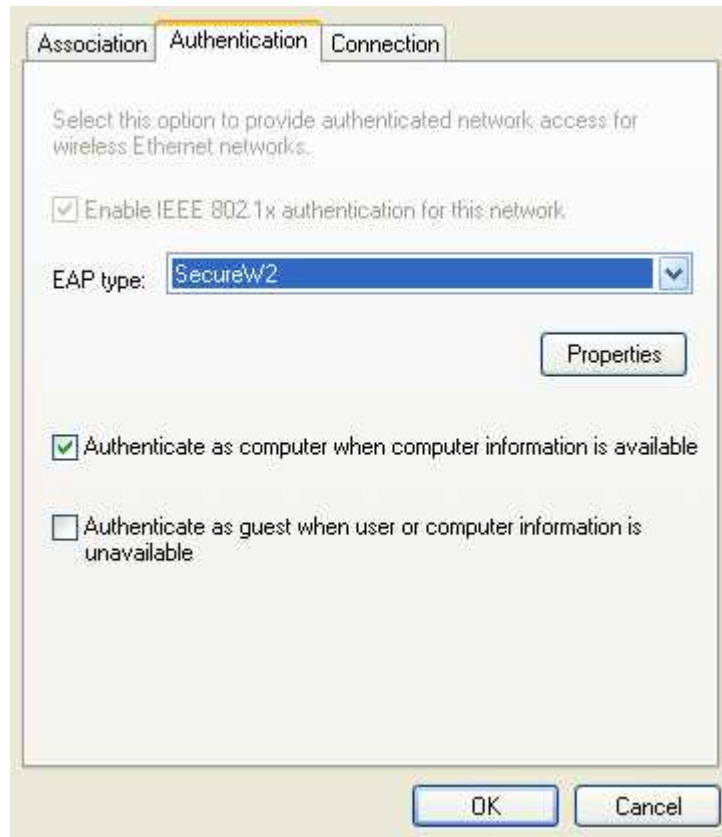
Now the WLAN connection needs to be configured, go to the network properties and right-click on „wireless networks“, where you choose properties as well. Now select „Wireless networks“ and click on add:



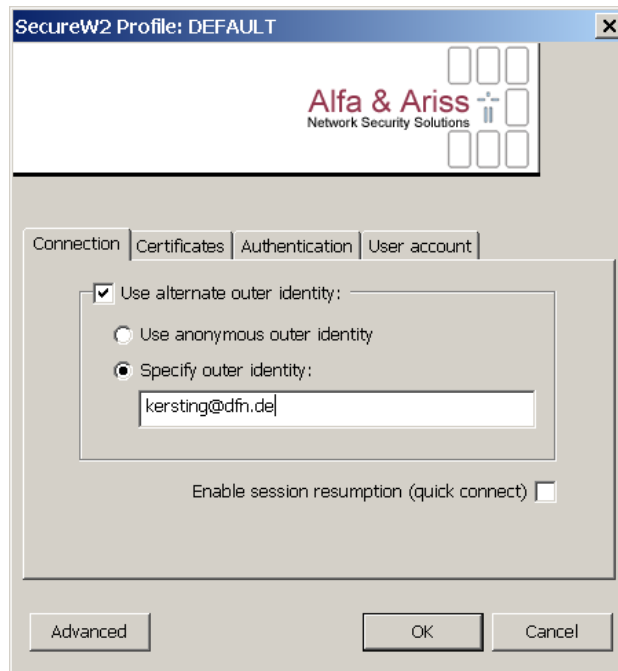
Now enter the SSID „eduroam“ and choose WPA/TKIP as encryption. Not being able to select these might be due to two reasons: first the firmware of your WLAN adapter might need to be updated, second you need to install the WPA patch for XP SP2. (Knowledge Base Article 893357).



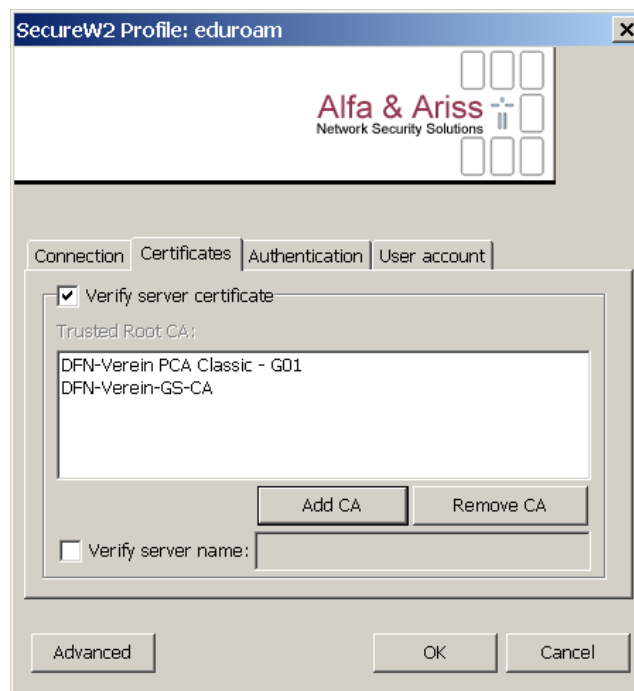
Now click on the authentication tab and choose SecureW2 as EAP-Type. Then click on properties:



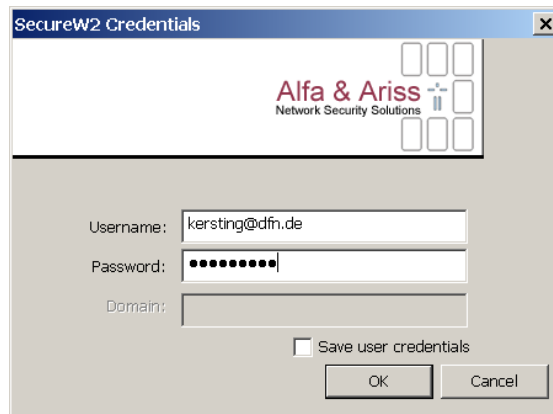
Click on „new“, select a meaningful profile name, preferably „eduroam“ and set your outer identity in the form yourname@yourrealm. It is possible to use anonymous as well but the usage of outer=inner identity is recommended.



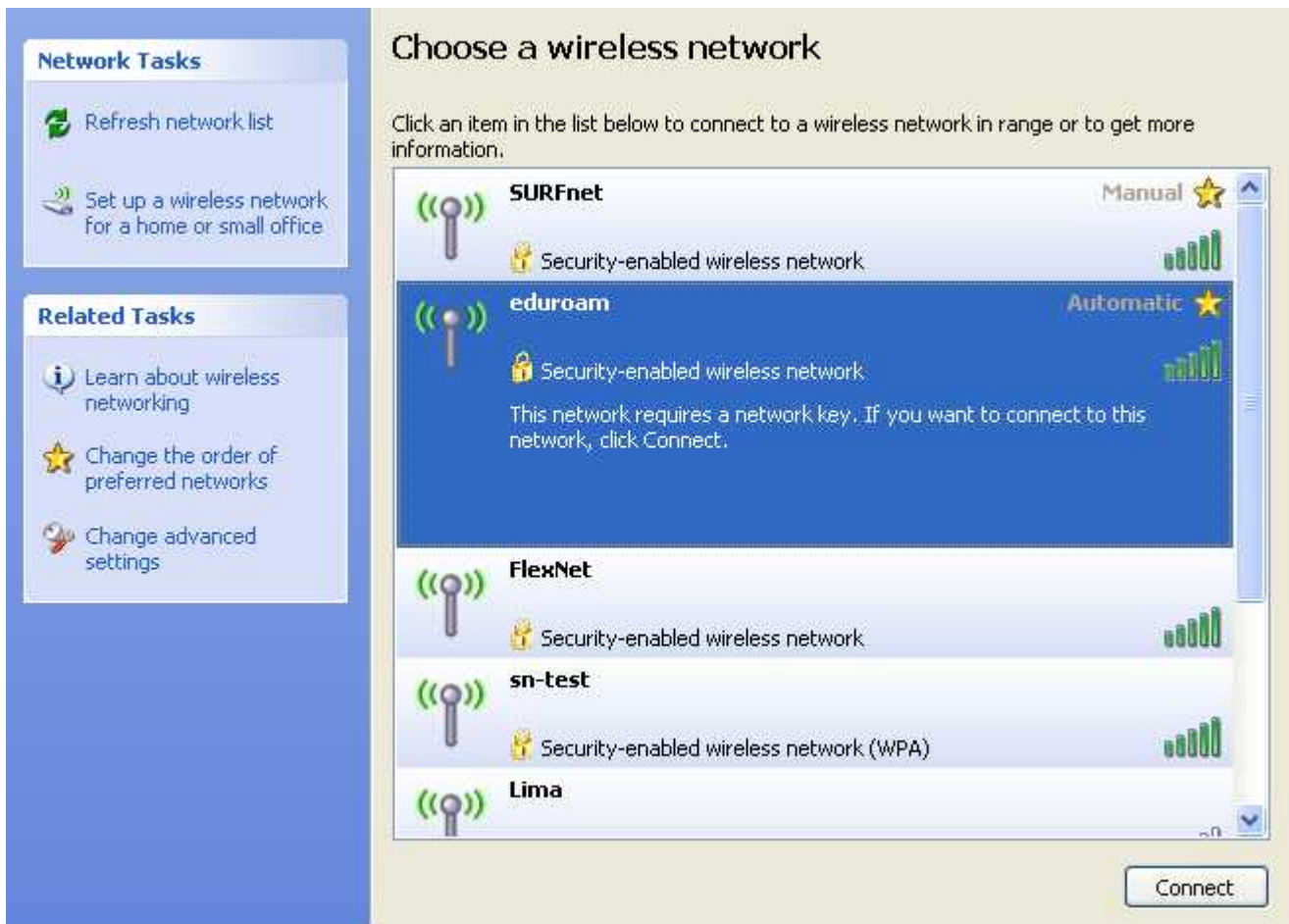
Check the „Verify server certificate“ checkbox and add the certificates from your ca. Now check the „Verify server name“ and add the DNS name of your RADIUS server.



Now click on the „Authentication“ tab, verify that „Authentication Method“ is set to pap and click on the „User account“ tab. Now enter your username in the format „yourname@yourrealm“, your password and you are almost done.



After clicking on „ok“ to close all open windows you are ready to connect. Click on the WLAN network symbol in the task bar to see all wireless networks in your area, select eduroam and click on connect.

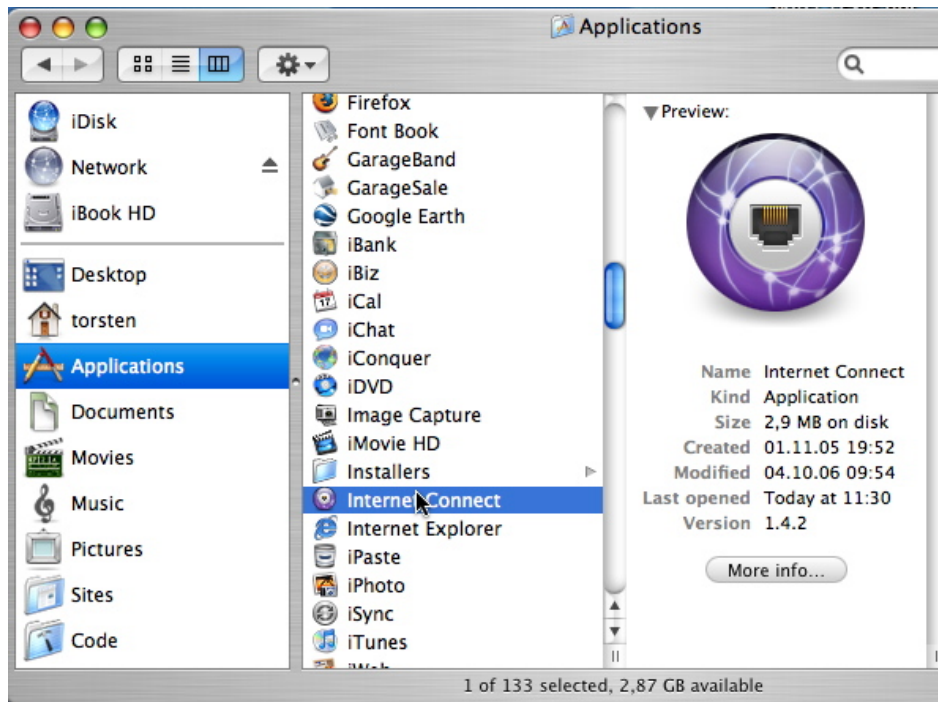


Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

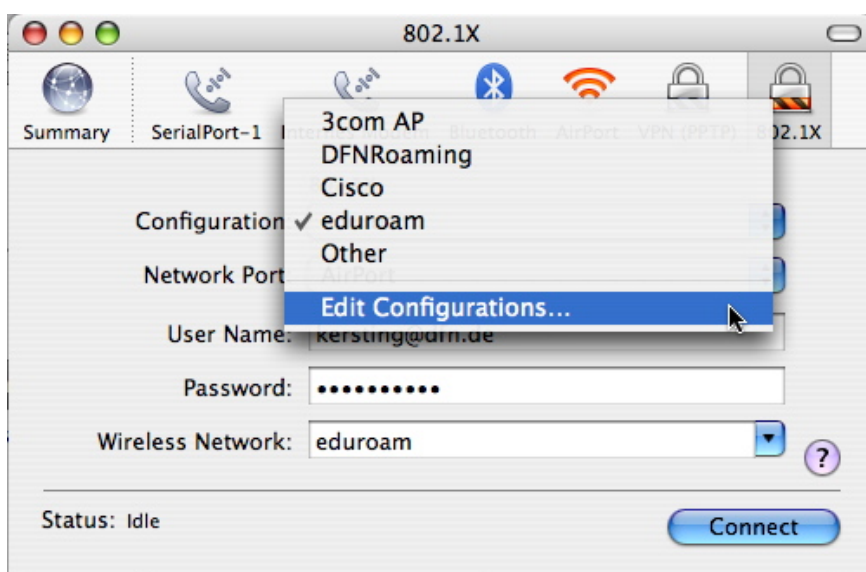


## C.2 MacOS

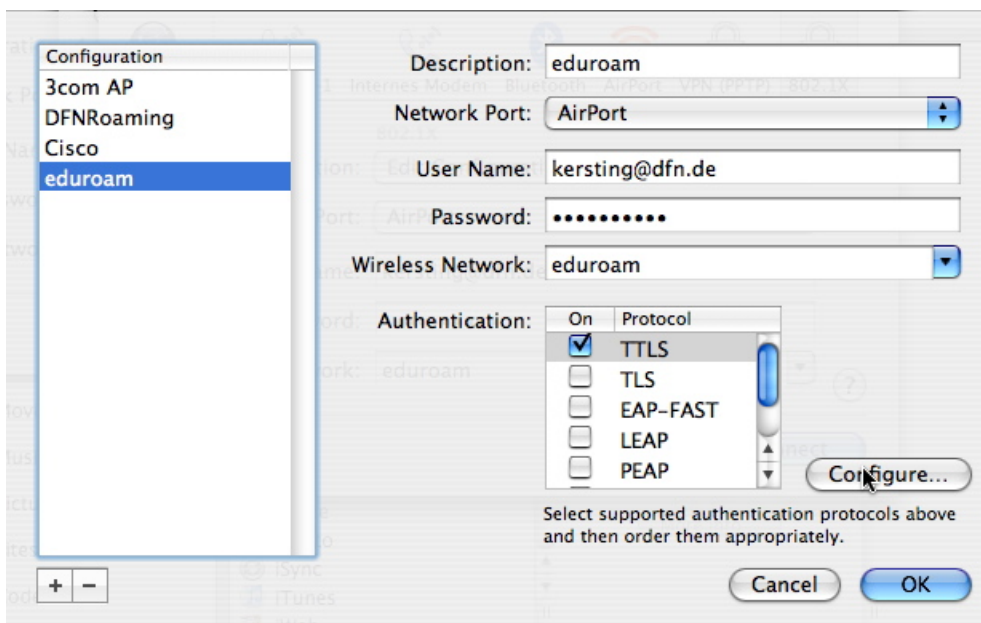
Configuring the MacOS X Supplicant is pretty much straight forward, everything works right out of the box. To begin the configuration start the program "Internet Connect" which is part of the MacOS X default installation:



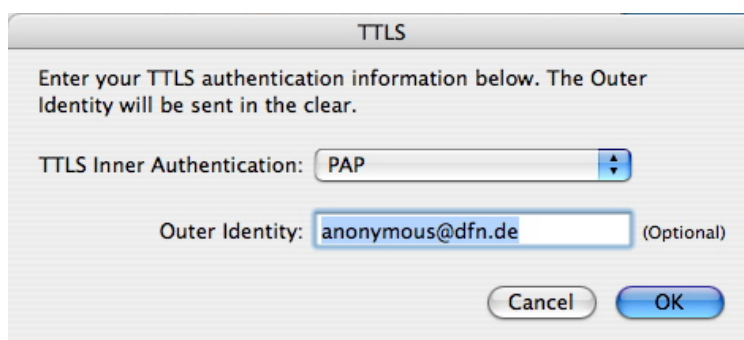
Select „802.1x“ on the top bar and then choose „Edit Configurations“ from the configuration pull down menu:



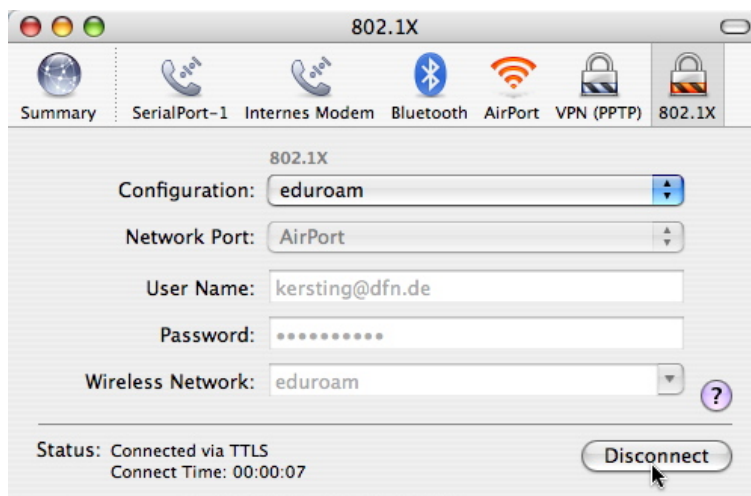
Now click on the „+“ mark at the bottom of the configurations list to add the configuration for eduroam. Fill in a meaningful description, preferably „eduroam“, and enter your username in the format user@realm and your password (the outer identity). Select eduroam as the wireless network (it is possible to just type the SSID in case there is no eduroam network available for selection in the pull down menu at the time of the configuration). Check the box selecting TTLS as the authentication protocol, and click on configure:



Leave „Inner authentication“ set to „PAP“ and enter „yourname@yourrealm“ as Outer Identity, as shown in the screenshot, it is possible to use anonymous as the outer identity as well:



After clicking „OK“ the client immediately tries to connect and given the circumstance that an eduroam network is available you are online:



### C.3 WPA\_Supplicant

Wpa\_supplicant ([http://hostap.epitest.fi/wpa\\_supplicant/](http://hostap.epitest.fi/wpa_supplicant/)) is an open source 802.1x supplicant for Linux, BSD and MS Windows. This section describes its use on the Linux platform. Wpa\_supplicant is available for most modern Linux distributions and seems to be the focal point of 802.1X development on the \*nix platform.

It is out of scope of this cookbook to describe how wpa\_supplicant can be compiled from source or what options need to be enabled in the Linux kernel to make eduroam authentication work. Modern Linux distributions with standard kernel, wireless tools and wpa\_supplicant should work “out of the box”.

With below technical information it is possible to implement eduroam support so that it will be seamlessly integrated with the OS, this is however very distribution specific and therefore out-of-scope for this document.

This chapter shows the basic elements of the configuration and provides simple means of configuring eduroam in a universal way. It is assumed that the user has a working wireless card (this can be verified by using the `iwconfig` command).

wpa\_supplicant is responsible for the (layer 2) authentication of the user, and must be followed by some means of setting up the (layer 3) IP connection by using a DHCP-client. Wpa\_supplicant typically runs in the background controlling the connection, taking care of re-authentications, roaming between access points, etc. It is started with the command:

```
wpa_supplicant -i interface -c configuration_file -D driver -B
```

where **interface** is the system name for the wireless interface (like eth1, ath0, wlan0, etc.), **configuration\_file** is the location of the file, that will be described later on, **driver** is one of: wext, ipw, madwifi and ndiswrapper (described below) and **-B** option means ‘run in the background’.

Project:	GN2
Deliverable Number:	DJ5.1.5.1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

The driver setting depends on the particular card used.

The wext driver currently supports most existing cards (Atheros chipset based cards being an exception, madwifi should be used there). Hence, the wext setting should be tested first.

The configuration file depends on the EAP type of choice. Example configurations are provided for EAP-TTLS, EAP-PEAP and EAP-TLS. Each of the examples contains two, nearly identical, network blocks, the only difference is that one is for WPA and the second for dynamic WEP.

In principle, one block with 'key\_mgmt=WPA-EAP IEEE8021X' should be sufficient, but it has been found out that under certain conditions this may fail, whereas two separate blocks seem to work correctly.

The ca\_cert points to the certificate file of the CA which has provided the certificate for the RADIUS server. This file should contain certificates for the whole certification chain, up to the root. All certificates and keys should be in PEM format.

```
# EAP-TTLS configuration
ctrl_interface=/var/run/wpa_supplicant

network={
    ssid="eduroam"
    key_mgmt=WPA-EAP
    ca_cert="/etc/eduroam/ca.cer"
    identity="user@your.domain"
    eap=TTLS
    password="xxxx"
    phase2="auth=PAP"
}
network={
    ssid="eduroam"
    key_mgmt=IEEE8021X
    ca_cert="/etc/eduroam/ca.cer"
    identity="user@your.domain"
    eap=TTLS
    password="xxxx"
    phase2="auth=PAP"
}
```

Note: this example will set the outer identity to be the same as the real, inner identity of the user. It is possible to set the outer identity to a different name (for instance to an opaque id), but for simplicity this is not shown here.

```
# EAP-PEAP configuration
ctrl_interface=/var/run/wpa_supplicant

network={
    ssid="eduroam"
    key_mgmt=WPA-EAP
    ca_cert="/etc/eduroam/ca.cer"
    identity="user@your.domain"
    eap=PEAP
    password="test"
    phase2="auth=MSCHAPV2"
}
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
network={
    ssid="eduroam"
    key_mgmt=IEEE8021X
    ca_cert="/etc/eduroam/ca.cer"
    identity="user@your.domain"
    eap=PEAP
    password="test"
    phase2="auth=MSCHAPV2"
}

# EAP-TLS configuration
ctrl_interface=/var/run/wpa_supplicant

network={
    ssid="eduroam"
    key_mgmt=WPA-EAP
    ca_cert="/etc/eduroam/ca.cer"
    identity="user@your.domain"
    eap=TLS
    client_cert="/etc/eduroam/user.crt"
    private_key="/etc/eduroam/user.key"
    private_key_passwd="xxxx"
}
network={
    ssid="eduroam"
    key_mgmt=IEEE8021X
    ca_cert="/etc/eduroam/ca.cer"
    identity=" user@your.domain "
    eap=TLS
    client_cert="/etc/eduroam/user.crt"
    private_key="/etc/eduroam/user.key"
    private_key_passwd="xxxx"
}
```

Note: Most wpa\_supplicant compilations will accept user key/certificate in one PFX (p12) file. If that is used, this file should be pointed to by private\_key and client\_cert should be commented out.

The script provided below starts and stops the eduroam connection. It needs to be configured by assigning correct values to the variables in the configuration section. The script kills possible wpa\_supplicant processes, and DHCP clients for the particular interface. Then it starts wpa\_supplicant and monitors its state with wpa\_cli. If no authentication takes place during the REAUTH\_TIMEOUT period, wpa\_supplicant is restarted. After authentication, the DHCP client is started.

```
#!/bin/sh
#
WPA_SUPPLICANT="/sbin/wpa_supplicant"
WPA_CLI="/sbin/wpa_cli"
DRIVER="wext"
WPA_CONF="/etc/eduroam/wpa_supplicant.conf"
DHCPD="/sbin/dhclient"
INTERFACE="eth1"
REAUTH_TIMEOUT=40
# end of configuration section
dhclient=`basename $DHCPD`
case "$1" in
```

Project:	GN2
Deliverable Number:	DJ5.1.5,1
Date of Issue:	02/02/07
EC Contract No.:	511082
Document Code:	GN2-06-258v8

```
start)
echo "starting network on ${INTERFACE}"
pkill wpa_supplicant
kill `ps -ef | awk "/$dhclient/ && /eth1/ && ! /awk/ {print $2}"` 1>/dev/null 2>&1
${WPA_SUPPLICANT} -B -D ${DRIVER} -c ${WPA_CONF} -i ${INTERFACE} -P \
    var/run/wpa_supplicant .pid 1>/dev/null 2>&1
if [ "$WPA_CLI" ] ; then
    i=1
    echo "waiting for connection"
    while ! $WPA_CLI status | grep -q AUTHENTICATED ; do
        sleep 1
        i=`expr $i + 1`
        if [ $i -gt $REAUTH_TIMEOUT ] ; then
            echo "restarting wpa_supplicant"
            echo "waiting for connection"
            pkill wpa_supplicant
            sleep 1
            ${WPA_SUPPLICANT} -B -D ${DRIVER} -c ${WPA_CONF} -i ${INTERFACE} -P \
                /var/run/wpa_supplicant.pid 1>/dev/null 2>&1
            i=1
            sleep 1
        fi
    done
    echo "connected to eduroam"
else
    sleep 10
fi
echo "setting IP"
${DHCPD} ${INTERFACE}
;;
stop)
echo "stopping network on ${INTERFACE}"
pkill wpa_supplicant
kill `ps -ef | awk "/$dhclient/ && /eth1/ && ! /awk/ {print $2}"` 1>/dev/null 2>&1
;;
*)
echo "Usage $0 {start|stop}"
exit 1
;;
esac
```

This script has to be run with administrator's rights, which is not very convenient. With a little extra effort one can create wrappers, which can then be connected to panel buttons, and so the network can be started and stopped by clicking with the mouse and providing the administrator's password. Since, for a casual user, finding the right values of all variables, creating the correct configuration file, creating correct wrapper scripts may be far too complicated, an alternative way has been proposed and implemented in a form of a simple package, which can be downloaded from:

[http://eduroam.pl/Files/prepare\\_eduroam\\_config.tgz](http://eduroam.pl/Files/prepare_eduroam_config.tgz).

This utility allows campus administrators to create a configuration script that can be distributed to the users. The script contains all necessary certificates, scans the system for the needed tools and creates configuration files, certificate files, sets up the main starting script and wrappers. A full description is beyond the scope of this document, but can be found in the documentation of the package.