Cookbook eduroam-br

v.2.0

Edelberto Silva
Otávio Roma

Abril de 2011

Sumário

Introdução

O eduroam é um projeto já consolidado na comunidade acadêmica europeia. Ele tem como objetivo, permitir que estudantes, pesquisadores e a equipe de instituições participantes obtenham conectividade à Internet, através de conexão sem fio, dentro de seus campi e quando visitam as instituições parceiras.

No Brasil ele surge através de uma iniciativa da Rede Nacional de Pesquisa (RNP), visando implantar uma infraestrutura de autenticação e acesso, semelhante à europeia, entre as universidades brasileiras. Neste primeiro momento, foram escolhidas algumas instituições para integrar a fase piloto do projeto. São elas: Universidade Federal Fluminense (UFF), Universidade Federal do Rio de Janeiro (UFRJ) e Universidade Federal do Mato Grosso do Sul (UFMS).

Para garantir conectividade, o eduroam se vale de uma arquitetura semelhante à da Figura . Nela, percebemos uma estrutura hierárquica de servidores de autenticação RADIUS, onde cada servidor possui sua base de dados LDAP associada. Todo usuário está cadastrado na base LDAP da instituição a que pertence.

Quando um suplicante tenta se conectar a um dos pontos de acesso (*Access Point - AP*) da estrutura, este AP irá consultar seu servidor de autenticação que, por sua vez, irá verificar se o suplicante consta em sua base de dados LDAP. Em caso positivo, o servidor autoriza o acesso à rede, já em caso negativo, serão consultados os servidores de hierarquia superior até que seja localizado o servidor RADIUS responsável pelo domínio daquele suplicante.

Se a entrada correspondente ao suplicante não for encontrada em nenhuma base de dados do sistema, ele terá o acesso à rede negado pelo servidor de autenticação inicial.

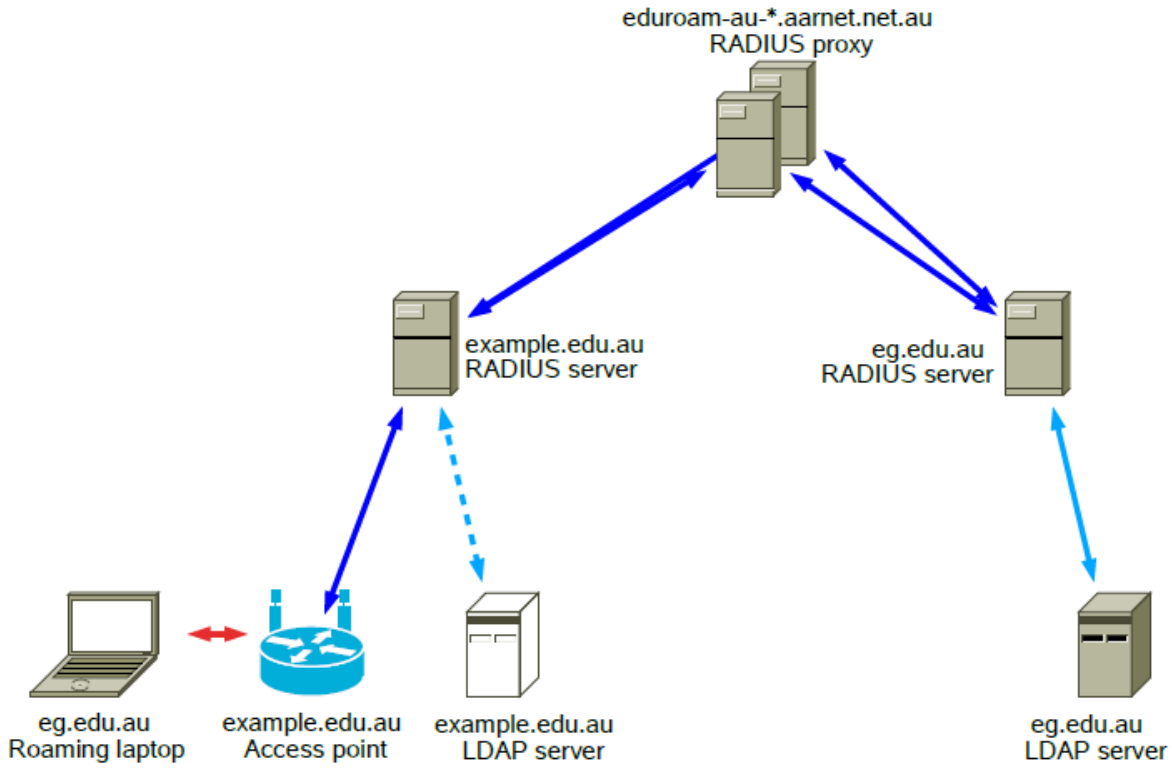
Este documento visa complementar a documentação já existente, acrescentando ao projeto as experiências vivenciadas pelos pesquisadores do eduroam-br. No primeiro tópico é apresentada a configuração de uma base de dados LDAP, desde sua criação até uma pequena descrição das ferramentas necessárias para manipulá-la.

O segundo tópico trata da migração de uma base NIS para uma base LDAP. Neste documento, optamos por utilizar os scripts da ferramenta *Migration Tools*.

O terceiro tópico se refere à instalação do servidor RADIUS, desde o *download* dos fontes no sítio do desenvolvedor até sua descompactação e compilação.

O quarto tópico trata da configuração do RADIUS. Neste cenário, não há ainda a capacidade de *roaming* entre as instituições, pois os servidores de federação não estão presentes.

No quinto tópico tratamos da configuração dos pontos de acesso para uma arquitetura que contemple os servidores de autenticação.



Figura

Configurando o LDAP

A configuração foi baseada no tutorial disponível em [4].

Primeiro instalamos o LDAP e uma série de ferramentas úteis à sua configuração com o seguinte comando:

```
# sudo apt-get install slapd ldap-utils
```

Para popular a base, é necessário carregar arquivos adicionais de *schema.* Para isso é preciso digitar os seguintes comandos:

```
# sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
# sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
#      sudo      ldapadd      -Y      EXTERNAL      -H      ldapi:///      -f
/etc/ldap/schema/inetorgperson.ldif
```

Agora, criamos um arquivo de *backend.* Segue o arquivo criado:

```
# Load dynamic backend modules
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: back_hdb

# Database settings
dn: olcDatabase=hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcSuffix: dc=uff,dc=br
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=admin,dc=uff,dc=br
olcRootPW: teste123
olcDbConfig: set_cachesize 0 2097152 0
olcDbConfig: set_lk_max_objects 1500
olcDbConfig: set_lk_max_locks 1500
olcDbConfig: set_lk_max_lockers 1500
olcDbIndex: objectClass eq
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcAccess: to attrs=userPassword by dn="cn=admin,dc=uff,dc=br" write by
anonymous auth by self write by * none
olcAccess: to attrs=shadowLastChange by self write by * read
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=admin,dc=uff,dc=br" write by * read
```

Nomeamos nosso arquivo como *backend.uff.br.ldif*. Feito isso, adicionamos o mesmo à base de dados com o seguinte comando:

```
# sudo ldapadd -Y EXTERNAL -H ldapi:/// -f backend.example.com.ldif
```

A partir deste momento a base de dados de *frontend* está pronta para ser populada. Criamos um arquivo de *frontend* com o seguinte conteúdo:

```
# Create top-level object in domain
```

```
dn: dc=uff,dc=br
objectClass: top
objectClass: dcObject
objectclass: organization
o: UFF Universidade Federal Fluminense
dc: uff
description: LDAP UFF

# Admin user.
dn: cn=admin,dc=uff,dc=br
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword: XXXXXX

dn: ou=People,dc=uff,dc=br
objectClass: organizationalUnit
ou: People

dn: ou=Group,dc=uff,dc=br
objectClass: organizationalUnit
ou: Group

dn: uid=root,ou=People,dc=uff,dc=br
uid: root
cn: root
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: XXXXXX
shadowLastChange: 14866
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 0
gidNumber: 0
homeDirectory: /root
gecos: root

dn: uid=bin,ou=People,dc=uff,dc=br
uid: bin
cn: bin
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: XXXXXXX
shadowLastChange: 14054
shadowMax: 99999
shadowWarning: 7
loginShell: /sbin/nologin
uidNumber: 1
gidNumber: 1
homeDirectory: /bin
```

```
gecos: bin

dn: aemon,ou=People,dc=uff,dc=br
uid: daemon
cn: daemon
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: XXXXXXX
shadowLastChange: 14054
shadowMax: 99999
shadowWarning: 7
loginShell: /sbin/nologin
uidNumber: 2
gidNumber: 2
homeDirectory: /sbin
gecos: daemon

dn: uid=adm,ou=People,dc=uff,dc=br
uid: adm
cn: adm
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: XXXXXXX
shadowLastChange: 14054
shadowMax: 99999
shadowWarning: 7
loginShell: /sbin/nologin
uidNumber: 3
gidNumber: 4
homeDirectory: /var/adm
gecos: adm
                                        .
                                        .
                                        .
```

Também não há um lugar obrigatório onde ele deve estar. O nosso arquivo de *frontend* foi criado a partir da conversão da base NIS em uma base de dados LDAP. O procedimento usado para migração NIS LDAP será comentado mais adiante.

Nomeamos nosso arquivo como *frontend.uff.br.ldif*. Basta agora adicionarmos estas entradas presentes no arquivo à base de dados LDAP:

```
# sudo ldapadd -x -D cn=admin,dc=example,dc=com -W -f frontend.uff.br.ldif
```

Este comando adiciona entradas à base desde que as mesmas não contenham erros. Caso uma entrada contenha erro, ele interrompe o processo e retorna o erro encontrado, não adicionando as entradas restantes. Como não encontramos suporte, por parte do LDAP, ao uso de acentos, nossa base de dados convertida do NIS pode nos trazer problemas em virtude de eventuais entradas que possuam tais caracteres. Podemos ter dificuldades em adicionar uma

base dados muito extensa por conta da quantidade destes caracteres uma vez que a cada acento ou cedilha encontrado o processo de inserção é interrompido.

Para solucionar este problema, devemos retirar os acentos presentes na base de dados e utilizar o seguinte comando:

```
# sudo ldapmodify -c -a -x -D cn=admin,dc=uff,dc=br -W -f frontend.uff.br.ldif
```

Este comando irá adicionar à base as entradas restantes.

Outra questão é a inserção dos *passwords* no arquivo de *frontend.* Podemos inseri-lo em texto claro no atributo *userPassword* de cada entrada. Podemos inserir também, em vez do texto claro, um *hash* da senha. Para extrairmos esse *hash* devemos utilizar o comando *slappasswd:*

```
1. # sudo slappasswd
2. New password:
3. Re-enter new password:
4. {SSHA}AmM9c3YpiKNR3y97QbXnQxHXkiQQ3npq
```

A linha 4 é o *hash* da senha que foi dada como entrada para o comando, que neste exemplo foi a senha “teste”.

Uma vez que populamos a base com êxito, podemos verificar se o conteúdo foi corretamente adicionado realizando uma busca com a ferramenta *ldapsearch.* Como exemplo:

```
# sudo ldapsearch -xLLL -b "dc=uff,dc=br" uid=fsilva sn givenName cn
```

Para nossa base, temos como saída:

```
dn: uid=fsilva,ou=People,dc=uff,dc=br
cn: Felipe Jose da Silva
```

Migração da base de usuários NIS para LDAP

Foram utilizados *scripts* de auxílio à migração da base de usuários utilizada pelo *NIS* (*Network Information Service*), em nosso caso o *passwd* e também o *shadow* (que armazena as senhas criptografadas) para o formato padrão de importação do LDAP (*Lightweight Directory Access Protocol*). Os *scripts* estão disponíveis em [1].

Uma característica do *script* de migração é o suporte a senhas criptografadas e armazenadas no *shadow,* porém é tomado como localização padrão do arquivo *shadow* o caminho */etc/shadow*. Em nosso caso, como migramos arquivos de uma máquina de terceiros

e os arquivos, portanto, não se encontravam em suas localizações padrão (ou seja, abaixo no */etc*) foi realizada a alteração do script *'migrate_passwd.pl*' na linha 188

```
#open(SHADOW, "/etc/shadow") || return;
open(SHADOW, "./shadow") || return;
```

Sendo assim o *script* irá solicitar a entrada dos parâmetros de localização do *passwd* e sua saída (*passwd.ldif*) e utilizará a localização do arquivo *shadow* como a pasta corrente.

Foram migrados, portanto, o *passwd* e o *group* para seus formatos equivalentes em LDAP, como segue:

```
# ./migrate_passwd.pl passwd passwd.ldif
# ./migrate_group.pl group group.ldif
```

Observação: Há dois comandos interessantes no Linux quando se trabalha com o *passwd* e *shadow.* São eles: *pwconv* e *pwunconv,* onde o primeiro é responsável por gerar o arquivo *shadow* a partir do *passwd e* o segundo realizar o caminho inverso, ou seja, a partir da existência do *passwd* e *shadow*, gerar o arquivo *passwd* único novamente.

Instalando o RADIUS

A instalação do servidor é bem simples e demanda poucos comandos.

Primeiro, teremos que baixar os fontes da última versão estável do FreeRadius. Atualmente esta versão é a 2.1.10.

```
#       wget       ftp://ftp.freeradius.org/pub/freeradius/freeradius-server-
2.1.10.tar.gz
# tar -xvzf freeradius-server-2.1.10.tar.gz
# cd freeradius-server-2.1.10
# ./configure
# make
# make install
```

Configurando o RADIUS sem Federação

A configuração do servidor RADIUS independe da construção de uma base de dados LDAP. Ele pode funcionar autenticando usuários e validando acesso de diversas maneiras. Podemos, por exemplo, criar uma base de dados no próprio servidor RADIUS, adicionando clientes no arquivo */usr/local/etc/raddb/users.*

Nossa abordagem será dividir nossa tarefa em pequenas metas, visando tornar mais simples o trabalho de solucionar eventuais falhas de configuração.

Será configurado um servidor RADIUS capaz de autenticar usuários a partir do */usr/local/etc/raddb/users*, sem qualquer tipo de consulta à base LDAP.

Como primeiro passo, editaremos o arquivo /usr/local/etc/raddb/clients.conf para que possamos realizar um teste de autenticação em *localhost* e em texto plano. O referido arquivo deverá conter as seguintes linhas:

```
client 127.0.0.1 {
      secret            = teste123
      require_message_authenticator = no
      shortname         = localhost
      nastype     = other
}
```

No arquivo /usr/local/etc/raddb/users, deveremos acrescentar as linhas:

```
clientex    Cleartext-Password := "password"
            Reply-Message = "Hello, %{User-Name}"
```

Onde *clientex* corresponde ao *login* do cliente e "*password*" é a senha do referido cliente.

Feito isso podemos iniciar o servidor RADIUS em modo *debug* com o seguinte comando:

```
      # radiusd -X
```

Agora, é só testar a parte de autenticação do usuário através do comando:

```
      # radtest -t pap clientex password localhost:1812 0 teste123
```

Onde *clientex* e *password* são os campos editados no arquivo *users* e *teste123* refere-se ao campo editado no arquivo *clients.conf*.

A resposta para o cliente de que a conexão foi realizada com sucesso é algo como a destacada a seguir:

```
Sending Access-Request of id 169 to 127.0.0.1 port 1812
    User-Name = "clientex"
    User-Password = "password"
    NAS-IP-Address = 127.0.0.1
    NAS-Port = 0
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=169,
length=39
    Reply-Message = "Hello, clientex"
```

A configuração do ponto de acesso sem fio para encaminhar os pedidos de autenticação ao servidor RADIUS será comentada na próxima seção. Devemos ainda incluir a liberação do ponto de acesso (em um arquivo de configuração no servidor) para envio de requisições ao servidor RADIUS. Este passo é similar ao realizado para a liberação do *localhost.* A seguir podemos visualizar as linhas que deverão ser incluídas no arquivo */usr/local/etc/raddb/clients.conf*, devendo conter, portanto, seu IP, a senha compartilhada e configurada no ponto de acesso, assim como o nome do ponto de acesso.

```
client 192.168.0.3 {
      secret            = teste123
      shortname         = eduroam-uff-1
      nastype           = other
}
```

Uma vez que configuramos o servidor RADIUS para autenticação de clientes em uma base local com senha em texto plano, seguiremos para uma segunda etapa. Aqui o servidor RADIUS, que agora sabemos estar funcionando corretamente, será configurado para, em vez de consultar sua própria base dados, consultar a base LDAP. Esta configuração é interessante sob muitos pontos de vistas, o principal deles é que desta maneira poderemos garantir maior segurança à nossa infraestrutura de autenticação, hospedando nossa base LDAP em um servidor diferente daquele que executa o RADIUS. Esta segmentação distribui as tarefas e evita que tenhamos um único ponto de falha.

Para a realização dos testes de autenticação utilizando a base local não houve nenhuma modificação na configuração original dos arquivos do RADIUS no que diz respeito aos métodos de consulta, sendo eles: *EAP-PEAP* e *MSCHAPv2*. Por meio dessa configuração foi possível, sem nenhuma modificação, autenticar os usuários por texto plano.

A fim de acrescentar maior segurança ao nosso ambiente utilizaremos *EAP-TTLS* e *PAP* para acesso a senhas armazenadas em *hash MD5* na base LDAP (base essa migrada anteriormente do servidor *NIS* como apresentado no início deste documento). Será necessário, portanto, habilitar o suporte ao LDAP e ao método de autenticação agora utilizado. Essas etapas serão descritas a seguir.

Primeiramente deveremos configurar o arquivo */usr/local/etc/raddb/eap.conf* contendo as seguintes informações

```
        eap {
                default_eap_type = peap
                timer_expire     = 60
                ignore_unknown_eap_types = no
                cisco_accounting_username_bug = no
                        max_sessions = 4096

                        md5 {
                        }
```

```
                leap {
                }

                gtc {
                }

        tls {
                certdir = ${confdir}/certs
                cadir = ${confdir}/certs
                private_key_password = SENHA_COMPARTILHADA
                private_key_file = ${certdir}/server.key
                certificate_file = ${certdir}/server.pem
                CA_file = ${cadir}/ca.pem
                dh_file = ${certdir}/dh
                random_file = /dev/urandom
                fragment_size = 2048
                include_length = yes
                check_crl = no
                cipher_list = "DEFAULT"
        }

        ttls {
                default_eap_type = mschapv2
                copy_request_to_tunnel = yes
                use_tunneled_reply = yes
                virtual_server = "eduroam"
        }

        peap {
                default_eap_type = mschapv2
                copy_request_to_tunnel = yes
                use_tunneled_reply = yes
                virtual_server = "eduroam"
        }

        mschapv2 {
        }

}
```

Neste arquivo é configurado o tipo padrão de autenticação (*EAP*) a ser utilizado em nosso ambiente, o *TTLS*. Assim como, os suportes a *TLS* e também ao *PEAP*. Para a configuração do *TLS* são expostos os caminhos dos certificados necessários à autenticação. Tanto para a utilização de *TTLS* ou *PEAP* é referenciado o arquivo */usr/local/etc/raddb/sites-enable/eduroam* como *virtual_server* e é nele que deveremos configurar os métodos de autenticação suportados. A seguir o arquivo */usr/local/etc/raddb/sites-enable/eduroam* é descrito por completo.

```
server eduroam {
      authorize {
            suffix
            preprocess
            auth_log
            ldap
            chap
            mschap
            pap
            eap {
                  ok = return
              }

      }
      authenticate {
            Auth-Type LDAP{
                  ldap
            }
            Auth-Type PAP{
                  pap
            }
            Auth-Type MS-CHAP{
                  mschap
            }
            Auth-Type EAP {
                  eap
            }

            Auth-Type CHAP {
                  chap
              }


            eap
      }
      preacct {
            preprocess
      }

      accounting {
            detail
            radutmp
            unix
            attr_filter.accounting_response
      }

      session {
            radutmp
```

```
	}

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

	pre-proxy {
			attr_filter.pre-proxy
			pre_proxy_log
	}

	post-proxy {
		eap
			post_proxy_log
			attr_filter.post-proxy
		Post-Proxy-Type Fail {
			detail
		}
	}
}
```

No arquivo */usr/local/etc/raddb/sites-enable/eduroam* são descritos os módulos autorizados a serem utilizados assim como o suporte à autenticação.

Como complemento, é relevante citar que as informações não constantes no arquivo mencionado serão pesquisadas no arquivo */usr/local/etc/raddb/sites-enable/default*, que como o próprio nome sugere, carrega as configurações padrão do aplicativo. O arquivo *default* encontra-se descrito a seguir.

```
authorize {
	preprocess
	auth_log
	chap
	mschap
	digest
	suffix
	eap {
		ok = return
	}
	unix
	files
	ldap
	daily
	expiration
```

```
      logintime
      pap
}

authenticate {
      Auth-Type PAP {
            pap
      }

      Auth-Type CHAP {
            chap
      }

      Auth-Type MS-CHAP {
            mschap
      }

      digest
      unix

      Auth-Type LDAP {
            ldap
      }

      eap
}


preacct {
      preprocess
      acct_unique
      suffix
      files
}

accounting {
      detail
      daily
      unix
      radutmp
      exec
      attr_filter.accounting_response
}

session {
      radutmp

}
```

```
post-auth {
      exec

      Post-Auth-Type REJECT {
            attr_filter.access_reject
      }
}

pre-proxy {
}

post-proxy {
      eap
      Post-Proxy-Type Fail {
            detail
      }
}
```

A indicação da base LDAP assim como a localização dos usuários dentro dessa base são configuradas no módulo do LDAP localizado no arquivo */usr/local/etc/raddb/modules/ldap*

```
     ldap {
      server = "localhost"
      basedn = "dc=uff,dc=br"
      password_attribute = "SEU_PASSWORD_DO_LDAP"
      filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
      start_tls = no
      dictionary_mapping = ${raddbdir}/ldap.attrmap
      ldap_connections_number = 5
      timeout = 4
      timelimit = 3
      net_timeout = 1
      edir_account_policy_check = yes
    }
```

Configurando o roteador sem fio Linksys WRT54G

Os roteadores sem fio *Linksys* utilizados estavam com versões OpenWRT instaladas. Foi decidida a recuperação da versão original do *firmware*. E para tanto foi seguido o passo-a-passo do site do próprio OpenWRT [2].

Realize o *download* em [3] (A versão do *firmware* utilizada por nós foi a v4 - verifique abaixo do aparelho qual a sua versão). Se presume que esteja *logado* no terminal do aparelho via ssh.

```
# cd /tmp
# wget http://www.example.org/original.bin
```

É necessário transformá-lo de .bin para o formato .trx. Para isso, utilize o comando:

```
# dd bs=32 skip=1 if=original.bin of=original.trx
```

Após isto, é necessário somente o comando a seguir:

```
# mtd -e linux -r write original.trx linux
```

Onde “-e” equivale a reescrever a imagem do roteador e “-r” a ordem de reiniciar o roteador após a escrita do *firmware* original ser concluída.

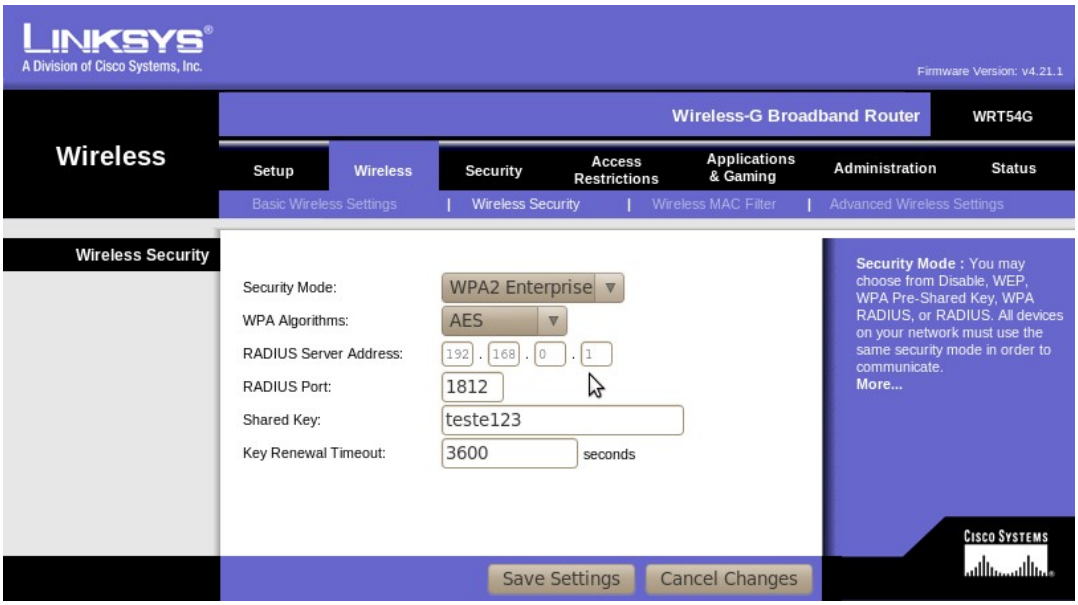
Aguarde e acesse o endereço 192.168.1.1 pelo seu navegador.

Dica: utilize DHCP (*Dynamic Host Configuration Protocol*) em sua máquina cliente que está ligada ao roteador.

Para que o sistema funcione da maneira almejada, o ponto de acesso sem fio também deve ser configurado de forma que encaminhe as requisições para o servidor de autenticação. A opção que permite tal configuração está em *Wireless* --> *Wireless Security*, como na Figura .

Figura

Em *Security Mode*, selecionaremos *WPA2 Enterprise*, isto indica que utilizaremos uma infraestrutura de acesso baseada em *WPA2* no ambiente sem fio, mas em vez de utilizarmos uma única senha para todos os clientes, iremos nos valer de um servidor de autenticação RADIUS. Isto nos provê uma série de recursos adicionais, como a possibilidade de definirmos uma senha diferente para cada cliente. Uma vez selecionada esta opção, outras aparecerão, como na Figura .



Figura

Em *WPA Algorithms*, selecionaremos *AES*, que nada mais é do que um algoritmo de criptografia simétrica que será utilizado para proteger as informações trocadas entre o ponto de acesso e o servidor. No *RADIUS Server Address* colocaremos o endereço IP do servidor RADIUS, que é para onde o ponto de acesso encaminhará as requisições feitas pelos clientes. Em *RADIUS Port,* colocaremos a porta pela qual o servidor RADIUS estará ouvindo as requisições. Em *Shared Key*, colocaremos a chave simétrica que o ponto de acesso irá utilizar para se comunicar com o servidor RADIUS. O campo *Key Renewall Timeout* não precisa ser alterado.

Com isso, encerra-se a etapa de configurações propostas neste capítulo. Através delas, é possível realizar autenticação e autorização, por intermédio do servidor RADIUS e com consulta a uma base de dados LDAP.

Agora, cada usuário pode acessar a infraestrutura de rede sem fio com o *login* e senha cadastrados na base LDAP. Como primeiro método de autenticação será utilizado o *EAP-TTLS* e, como segundo método, será utilizado o *PAP*.

Referências

[1]. Migration Tools. [Online]
http://www.padl.com/OSS/MigrationTools.html.

[2]. OpenWRT. *Deinstalling.* [Online]
http://wiki.openwrt.org/oldwiki/OpenWrtDocs/Deinstalling.

[3]. Suport. *Linksys.* [Online] http://homesupport.cisco.com/en-us/wireless/lbc/WRT54G?referrer=www.linksysbycisco.com.

[4]. Ubuntu Documentation. *OpenLDAP Server.* [Online]
https://help.ubuntu.com/10.04/serverguide/C/openldap-server.html.