Tutorial

John Mitchell, Glen Johnson *
Dave Worth, Philippe Hanset**
Jeff Hagley***

*University of Alaska
**University of Tennessee
***Internet2
# Table of contents

<table>
<thead>
<tr>
<th>Pages</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-12</td>
<td>Introduction to eduroam</td>
</tr>
<tr>
<td>13-23</td>
<td>Structure of an eduroam Top Level RADIUS</td>
</tr>
<tr>
<td>24-44</td>
<td>Campus RADIUS configs: FreeRADIUS</td>
</tr>
<tr>
<td>45-73</td>
<td>Campus RADIUS configs: RADIATOR</td>
</tr>
<tr>
<td>74-100</td>
<td>EAP methods</td>
</tr>
<tr>
<td>101-107</td>
<td>Auth. backend and RADIUS: MIT Kerberos</td>
</tr>
<tr>
<td>108-113</td>
<td>Auth. backend and RADIUS: SAMBA, openLDAP</td>
</tr>
<tr>
<td>114-115</td>
<td>Network Design Considerations: UA</td>
</tr>
<tr>
<td>116-120</td>
<td>Network Design Considerations: Internet2</td>
</tr>
<tr>
<td>121-125</td>
<td>Network Design Considerations: UTK</td>
</tr>
<tr>
<td>126-134</td>
<td>Example of Wireless Configs: Cisco LWAP</td>
</tr>
<tr>
<td>135-141</td>
<td>Example of Wireless Configs: LWAP remote office</td>
</tr>
<tr>
<td>142-151</td>
<td>Example of Wireless Configs: Aruba Networks</td>
</tr>
<tr>
<td>152-180</td>
<td>Supplicant Provisioning: scripting tools, Xpressconnect</td>
</tr>
<tr>
<td>181-184</td>
<td>Security, Policies, Support</td>
</tr>
<tr>
<td>185-198</td>
<td>RadSec</td>
</tr>
<tr>
<td>199-205</td>
<td>Lessons Learned</td>
</tr>
</tbody>
</table>
Introduction to eduroam
What is eduroam?

From www.eduroam.org:
eduroam (education roaming) is the secure, world-wide roaming access service developed for the international research and education community. eduroam allows students, researchers and staff from participating institutions to obtain Internet connectivity across campus and when visiting other participating institutions by simply opening their laptop.
eduroam relies on a worldwide federation of RADIUS servers to facilitate network access for roaming academic affiliates using IEEE 802.1x as the vehicle.
eduroam is Standard based (802.1x, RADIUS, EAP, WPA2)
Why eduroam?

When roaming:

- Data over cellular is limited by cost, Wi-Fi is generally not
- Web portals are time consuming for users
- Authentication is useful to the user and the service provider
- Encryption over the air is not a bad idea
- Compatibility across the world
Building block of eduroam: 802.1x on campus
802.1x on campus: more details

Suppliant  
802.11 Association  
WPA/WPA2  
802.1x  
EAPOL-Start  
EAP-Request Identity  
EAP-Response Identity  
(netid@utk.edu)  
SSL/TLS Tunnel  
EAP-Request Credentials  
(TTLS/PEAP)  
EAP-Response Credentials  
SSL/TLS Tunnel  
EAP-Success/Failure  
802.11 Keying  
WPA/WPA2  

Authenticator  
RADIUS Access-Request  
RADIUS Access-Challenge  
RADIUS Access-Request  
RADIUS Access-Accept/Reject  

Authentication Server  

802.1x beyond campus: eduroam
eduroam features

- With eduroam, an institution can verify the legitimacy of a visitor without direct access to the user's credentials (SSL/TLS tunnel goes between user's device and user's home RADIUS server)

- With eduroam, a user can verify the legitimacy of an infrastructure (When verifying the SSL/TLS certificate during the EAP exchange, the user also makes sure that the infrastructure is a valid member of the federation)

- With eduroam, the wireless traffic is encrypted between the user's device and the institution's infrastructure
eduroam pros and cons

Pros

- Instant Access
- Authentication (device/user/network)
- Encryption over the air
- Automatic provisioning
- Free data when traveling abroad
- Layer2: IPv4 and IPv6 compatible (no worries about non v6 portals)

Cons

- 802.1x can be cumbersome to setup
- No way to communicate information to visitors
- RADIUS uses UDP: can be problematic across many links (Frame Fragmentation, MTU, 2048 bit certs)
eduroam quick facts:

- 2000 institutions worldwide

- 34 members and 1 SP only in the US

- Anyone can be an SP (Service Provider, e.g. Wireless Provider)

- Not everyone can be an IdP (Identity Provider, in the case of eduroam: RADIUS + Authentication backend like LDAP or AD)

- EU data: International traffic = 1/10 National = 1/10 Regional

- No bad story so far (2004>2011)

- EAP method is between user and home RADIUS, Wireless encryption is specific to campus visited (WPA2 recommended)
eduroam developments

-RADSEC (TLS and TCP based)
Can solve Frame Fragmentations issues with RADIUS

-F-Ticks (Monitoring/Troubleshooting tool across federations)

-CUI attribute (Chargeable User Identity). Helps blacklisting and tracking users without interfering with the whole realm

-Project Moonshot (not eduroam, but using the RADIUS infrastructure)
Structure of an eduroam
Top-Level RADIUS Server
It's a Router!

All a Top-Level RADIUS Server (TLRS) does is route RADIUS requests. No authentication! We only pass authentication requests to where it needs to land.
TLRS Rules of the Road

- Granular separation of configuration is the key to flexibility
  - Intra-Federation and Inter-Federation configs are stored separately
  - Logging and monitoring are separate configs to allow changes to these isolated elements
TLRS Rules of the Road (con't)

- Generality where it makes sense (default routing) and specificity elsewhere.
  - Inter-Federation TLRS configs are broken into four chunks to allow for exceptions and careful handling of forwarding
  - APAN has 8 TLDs, The US has 3 (.edu, .us, .gov), Europe has many so they are "default"
    - default isn't necessarily blindly forwarded - blackhole as appropriate (.com, .mil, etc.)
  - Allow for exceptions in routing
    - We blackhole .net, except two .nets in Europe
    - We handle all .edus, except those we don't
Separate configs for each school.
  - Again, flexibility.
  - Every school is built from a standard config template but configured in obvious ways: Shared Secret, DNS names / IPs of RADIUS servers, realm(s)
  - Lots of potential for variation per institution
    - More clients than auth gateways (common)
    - per-client or auth gateway secrets (less common)
    - asymmetric secrets (client vs. auth gateway)
  - In the future, potentially less obvious variation per-institution: dynamic filtering based on identity, specific filtering for incident management, etc.
TLRS Rules of the Road (con't)

- Additional handlers to catch exceptional conditions and handle them gracefully
  - log and drop, rather than forwarding to default for certain cases:
    - institutions forwardings names without realms (and alert the institution if possible)
    - blackholed TLDs (.org, .com, .net, etc..)
    - any other exceptional condition (often defined as they are discovered)
  - We do periodic log analysis to find configuration bugs in our own and institutional configs and work to fix them iteratively.
TLRS RADIUS Config overview

radius.cfg:
1. Generic Configuration information - port bindings, log levels, etc.
2. Logging and Monitoring setup
3. TLRS Clients and Authentication Methods (sets up RADIUS peers) for later reference
4. eduroam-US peer configurations (one per school)
5. International Exceptions (JA.net, NORDUNet, etc.)
6. All unfederated realms for which we are authoritative are logged and dropped, then all realms that nobody handles (.com, etc) are logged and dropped as well.
7. TLRS Forwarding / Routing - APAN (explicit realm list: .au, .nz, .tw, .jp, etc...) and EU (default route)
TLRS_RADIUS Config overview (con't)

TLRS_Clients.cfg:
- APAN and EU clients are configured with secrets here

TLRS_Authentication.cfg:
- This mirrors the clients configuration but has authentication information instead. These can be referenced later wherever needed (exceptions and standard routing)
EduroamUS_Institutions/\langle\text{institution}\rangle\.cfg (filled in for any given institution):
- Configures client and forwarding
- Loop prevention
- Empty realms are logged and dropped. Later the institution can be contacted re: bad forwarding to the TLRS
TLRS RADIUS Config overview (con't)

authoritative_fallthrough.cfg:
- catch .edu, .us, .gov realms that are not federated but users have attempted auth with them. Drop and log attempts

Blackhole.cfg:
- catch .com, .net, etc. Drop and log attempts
TLRS RADIUS Config overview (con't)

TLRS_Routing.cfg

- Forwards APAN realms to APAN
  - APAN TLDs: .au, .cn, .hk, .jp, .mo, .nz, .pg, .ph, .tw
- Forward all other realms to EU
FreeRADIUS

- Open Source RADIUS server
- Documentation on how to deploy FreeRADIUS in a generic sense and for eduroam, although poor compared to commercial alternatives
- Supports the major EAP types
- Can talk to most backend authentication
Why we chose FreeRADIUS

- Ability to use MIT Kerberos for backend authentication
  - The plan is to move to EAP-TLS and use certs, but EAP-TTLS was easy to get us started
- Open Source
- Has pre-built binaries for Redhat
  - This allows for easier server management
- Has IPv6 support
- Easy to bring up other servers
  - Just install and copy the configs over from a working server, make a few tweaks and you are good to go
Some Caveats

● Listen * would only listen on one interface
  ○ You need to define each address the server should listen on

● It is listed as freeradius2 when you are looking for binaries on Redhat

● Troubleshooting
  ○ Restart to view debug
  ○ Failure reason not clear in accounting records
    ■ Access-Reject... Ok, why?

● EAP-TLS
  ○ Re-reading CRL requires restart
  ○ OSCP in trunk, but not yet released
FreeRADIUS EAP-TTLS config

- There are 5 main files you need to configure to have basic 802.1x functionality with eduroam
  - radiusd.conf, eap.conf, clients.conf, proxy.conf, and users
  - You will probably also have to edit some other files depending on the authentication backend you choose
radiusd.conf

listen {
    ipaddr = 10.10.10.10
    port = 0
    type = auth
}

listen {
    ipv6addr = 2001:468:1::100
    port = 0
    type = auth
}

log {
    destination = files
    file = ${logdir}/radius.log
    syslog_facility = daemon
    stripped_names = no
    auth = yes
    auth_badpass = no
    auth_goodpass = no
}

eap {  
  default_eap_type = ttls  
  timer_expire = 60  
  ignore_unknown_eap_types = no  
  cisco_accounting_username_bug = no  
  max_sessions = 2048  
}  

tls {  
  certdir = \$(confdir)/certs  
  cadir = \$(confdir)/certs  
  private_key_file = \$(certdir)/serverkey.key  
  certificate_file = \$(certdir)/servercert.cert  
  dh_file = \$(certdir)/dh  
  random_file = \$(certdir)/random  
  cipher_list = "DEFAULT"  
  make_cert_command = \$(certdir)/bootstrap  
  cache {  
    enable = no  
    max_entries = 255  
  }  
}  

ttls {  
  default_eap_type = md5  
  copy_request_to_tunnel = yes  
  use_tunneled_reply = yes  
  virtual_server = "inner-tunnel"  
}  
}
clients.conf

client localhost {
  ipaddr = 127.0.0.1
  secret = secretkey
  nastype = other
}

client eduroam1.ns.utk.edu {
  secret = differentsecretkey
  nastype = other
}

client 192.168.1.0/24 {
  secret = secretkeyforsomenetwork
  nastype = other
}

proxy.conf

```conf
proxy server {
    default_fallback = no
}
home_server localhost {
    type = auth
    ipaddr = 127.0.0.1
    secret = secretkey
}
realm NULL {
}
realm LOCAL {
}
realm INTERNET2.EDU {
    type = radius
    authhost = LOCAL
    accthost = LOCAL
}
realm DEFAULT {
    type = radius
    authhost = eduroam1.ns.utk.edu
    accthost = eduroam1.ns.utk.edu
    secret = secretkey
    nostrip
}
```
users

Here is a sample for local auth:

User1 Auth-type = Local, User-Password = "password1"
User2 Auth-type = Local, User-Password = "password2"
Here is a sample for using MIT Kerberos:

DEFAULT Auth-Type = Kerberos
FreeRADIUS EAP-TLS Config
UAK/EAP-TLS > eap.conf

Notes

1. check_cert_cn
2. CA_file
3. Cert size / bits
4. certificate_file

```bash
eap {
    default_eap_type = tls
    timer_expire = 60
    ignore_unknown_eap_types = no
    cisco_accounting_username_bug = no
    max_sessions = 2048

tls {
    certdir = ${confdir}/certs
    private_key_file = ${certdir}/server.key
    certificate_file = ${certdir}/server.crt
    CA_file = ${certdir}/ca.pem
    dh_file = ${certdir}/dh
    random_file = /dev/urandom
    cipher_list = "DEFAULT"
    check_cert_cn = %{User-Name}

    make_cert_command = "${certdir}/bootstrap"
}
}
```
1. Simplify incoming requests with huntgroup "tag"
/etc/raddb/users

Username in Deny group?
● YES...
  ○ Bandwidth limit via QoS
  ○ Vlan w/ acl policies
● NO...
  ○ *Handle elsewhere*

MAC Filtering
● Allow by default
  ○ ------->
● Deny specifically
  ○ sites-available/default

DEFAULT  Huntgroup-Name == eduroam,
  Service-Type == Login-User,
Sql-Group == "Deny-Full-Access",
  Auth-Type := PAP
Airespace-QOS-Level = Bronze,
  Airespace-Interface-Name = "vlan 154",
  Reply-Message := "Guest authorized"

DEFAULT  Huntgroup-Name == eduroam,
  Service-Type == Call-Check,
NAS-Port-Type == Wireless-802.11,
  Auth-Type := ACCEPT
Within the Authorize { }, call modules

**preprocess**
- tag requests with Huntgroup-Name

**suffix**
- Stow `domain.tld` (in Realm) from `un@domain.tld`

**sql**
- Execute `authorize_check_query`, if results, execute `authorize_reply_query`

**files**
- process directives in `users file`

```python
authorize {
    # Discover Huntgroup-Name
    preprocess

    # Discover realm
    if ( User-Name =~ /@/ ) {
        suffix
        ntdomain
    }

    # Mac Filter
    if ( Service-Type == Call-Check
        && NAS-Port-Type == Wireless-802.11 ) {
        sql
        # Handle Whitelist / Blacklist
        sql

        # Otherwise, accept all (via
        # users file)
        files
    }

    ... continued ...
}
Within the Authorize { }, call modules, (continued)

```plaintext
authorize {
...
#
# If EAP-Message, assume this is EDUROAM
# and bypass everything else in the
# authorize{} section (since eduroam is
# handled via EAP in the authentication
# section).
#
if ( EAP-Message != "" ) {
  eap {
    ok = return
  }
}
...
}
```

eap
  * see eap.conf file
Within Authenticate { }, call modules

eap
  • see eap.conf file

authenticate {
  Auth-Type EAP {
    eap
  }
}

post-proxy {
  eap
}

/etc/raddb/sites-available/default
Testing your RADIUS server

- FreeRADIUS includes a great tool called eapol_test
- They also have a version that is a Nagios plugin
  - We use the Nagios plugin to test authentication to our local servers at set intervals
  - We also use it to test our peering with eduroamus by testing credentials from another domain
Log Reporting

- We use a perl script to parse the logs and email out a report
- The report gives the number of good and bad auths per user and per domain
- The metrics this provides has been invaluable in proving eduroam is a good thing to executives
Download and Help

● Download at www.freeradius.org
● Great documentation there
● More specific eduroam configuration
  ○ www.eduroamus.org
  ○ www.eduroam.org
Campus RADIUS Configs

Radiator
Configuration Overview

Two different cases are handled:
1. Connections from a campus "secure" WPA2-enterprise wireless network with SSID "campus-SSID"
2. eduroam also broadcast on campus.

If eduroam is the only WPA2 network then only the eduroam handler is required and discrimination between the cases may be ignored.
Notes on Configuring Radiator

Most Important:

- RTFineM - The Radiator documentation is very complete, updated with each revision, and lives on my desktop for daily reference
Notes on Configuring Radiator (con't)

- Radiator **Handlers** match in order from top to bottom of configuration files and stop searching once all parameters match. Thus more specific handlers **must** come before less specific handlers. Otherwise your handler just won’t be called and you’ll be confused. Debugging this can be painful.
  - Example: Tunneled Handlers (for PEAP or TTLS) must be defined before un-tunneled because a tunneled request is more specific than an un-tunneled request and will match all of the un-tunneled requirements short-circuiting the search!
As is well documented in the Radiator docs, but is very easy to forget, comments start with a # as the first character on the line! “Trailing” comments are taken as part of the parameter very often. This can be a pain to track down too!

Build a custom authentication logger to log both successful and failed authentications. Very often it is handy to watch until a user authentication is successful (particularly when they are struggling while away from home). This is also handy for accounting (without using RADIUS accounting).

The difference between log levels 3 and 4 is huge. Stay in 3 unless you are debugging!
Configuration sketch

Client: Campus Wireless => Client-Identifier: fromCampus

NAS-Identifier = "campus-SSID"
and Realm =~ /(?:^(?:[a-z0-9_]+\.)+institution\.edu$|\^$)/
  (regex matches institution.edu and all sub-realms and an empty realm)
=> Outer-Handler => TTLS/PEAP Inner Handler => Authentication

NAS-Identifier = "eduroam"
and Realm =~ /(?:[a-z0-9_]+\.)+institution\.edu$/
  (regex matches institution.edu and all sub-realms)
=> Outer-Handler => TTLS/PEAP Inner Handler => Authentication

and Realm =~ /\^$/i
  => Drop  (eduroamers must have a realm)
=> Forward to eduroam-US TLRS  (they are eduroamers with a routed realm)
Configuration sketch (con't)

Client: eduroam-US TLRS => Identifier: fromEduroam

Realm =~ /^(?:[a-z0-9]+\.[a-z0-9]+)*institution\.edu$/i
  => Outer-Handler => TTLS/PEAP Inner Handler => Authentication

Realm =~ /\$/ or (Realm !~ /institution\.edu$/i
  and TunnelledByPEAP/TTLS)
  => Drop
Note: we are using the `NAS-Identifier` string to differentiate the SSID to which the local user associated (this is configured per VirtualAP in Aruba.) YMMV based on your wireless architecture.

Moreover, to use the `NAS-Identifier` in both inner and outer handlers but it is only present in the outer-handler. We require a custom hook (written in perl) to pass the `NAS-Identifier` attribute to the tunneled handler with a "PreHandlerHook"
Setup logging

Log authentication attempts for both successes and failures. This goes a long way in user-support.

<AuthLog FILE>
Identifier authlogger
  Give the logger a name to refer to later

Filename %L/%Y/auth_log.%y%m%d
  Setup the log path

LogSuccess 1
LogFailure 1
  Log both success and failures
</AuthLog>
Campus Authentication Blocks

Configure each AuthBy Block necessary for your campus separately (outside of any handlers) and refer to them by an identifier. One can play tricks with multiple types in groups to achieve complicated ends.

The underlying authentication method must be configured and working before Radiator can use it.
Campus Authentication Blocks (con't)

For example, to use AuthBy NTLM first ntlm_auth must be working on the system.

**ntlm_auth requires that the machine has joined an ActiveDirectory (kinit <username>; klist; net ads join -U <username>; net info -U <username>) and winbindd is working (check via wbinfo -p and wbinfo -t)**

ActiveDirectory or Samba+OpenLDAP authentication:

```
<AuthBy NTLM>
  Identifier NTLMAuthentication
  Give this a good unique name by which we refer to it later

  EAPType MSCHAP-V2
  Domain Institution
  UsernameMatchesWithoutRealm

  Most directories do not use the realm, so strip it before authenticating
```

</AuthBy>
Campus Authentication Blocks

Depending on the authentication methods used on your campus, Radiator provides support for numerous Directory services and authentication back-ends:

```xml
<AuthBy LDAP2>
  Identifier LDAP2Authenticator
  ...
</AuthBy>

<AuthBy KRB5>
  Identifier KerberosAuthenticator
  ...
</AuthBy>
```
Define the NASs and/or other RADIUS servers able to connect to this instance if they know the shared secret:

```
# Campus Wireless Network(s)
<Client 10.0.0.0/16>
   IdenticalClients 192.168.23.0/24
   Secret <snip>
   Identifier fromCampus
</Client>

# a remote site that is considered part of campus
<Client remote.institution.edu>
   Secret <snip>
   Identifier fromCampus
</Client>
```
eduroam-US Client

Make sure that your travelers at other eduroam enabled institutions can connect back by allowing connections from the eduroam Top-Level RADIUS server (TLRS)

# eduroam-US top-level server
<Client eduroam1.ns.utk.edu>
  Secret <snip>
  Identifier fromEduroam
    Name this identifier for later reference

    StripFromRequest NAS-Identifier, ...
    Strip internal-use RADIUS attributes to prevent injections
</Client>
Example Outer-Handler for eduroam

```xml
<Handler Client-Identifier=fromCampus, \ 
NAS-Identifier="eduroam", \ 
Realm=/^[a-z0-9]+\.[a-z0-9]+\.[a-z0-9]+\.*institution\.(edu$)/i>
/AuthBy FILE>
   Filename %D/dot1x_anon
   EAPType PEAP, TTLS, TLS, ...
   EAPAnonymous %{User-Name}

PreHandlerHook file:'/etc/radiator/nasid_hook.pl'

   EAPTLS_CertificateType PEM
   EAPTLS_CAFFile <path to CA certificate>
   EAPTLS_CertificateFile <path to infrastructure cert>
   EAPTLS_PrivateKeyFile <path to private key>
   EAPTLS_PEAPVersion 0
   EAPTLS_NoAckRequired
   AutoMPPKEkys
</AuthBy>
</Handler>
```
Example Outer-Handler - details

<Handler Client-Identifier=fromCampus, \nUser is on campus...
Example Outer-Handler - details

<Handler Client-Identifier=fromCampus, \  
User is on campus...

NAS-Identifier="eduroam", \  
  but using the eduroam SSID
Example Outer-Handler - details

<Handler Client-Identifier=fromCampus, \
User is on campus...

NAS-Identifier="eduroam",  \
but using the eduroam SSID

Realm=/^(?:[a-z0-9]+\.)*institution\.edu$/i> \
and has provided a valid realm
<AuthBy FILE>
  Filename %D/dot1x_anon
There is a file on the filesystem against which we are authenticating for now

  EAPType PEAP, TTLS, TLS, ...
Defines which EAP types you are using on campus. Order the types by how common they are on your campus.

*Example:* If a TTLS user joins and PEAP is the default then the RADIUS server tries PEAP first and the client replies with an EAP-NAK and starts the new method... this adds latency
These directives setup Radiator's encrypted tunnel between client and
the RADIUS server. These are the second most important definitions
and easiest to make mistakes in during setup. Verify that the CA
Certificate, the Server Certificate and the Server's private key are
readable by the \texttt{EUID/EGID} of the radiusd process.

\texttt{EAPTLS\_CertificateType=PEM}
\texttt{EAPTLS\_CAFile \textless path to CA certificate\textgreater}
\texttt{EAPTLS\_CertificateFile \textless path to infrastructure cert\textgreater}

This certificate is the one presented to the user in some supplicants. It
should either be automagically installed on their devices in the correct
key-store (trust level) \textbf{or} be issued/signed by a well-known CA that is
already in their OS key-store (Comodo/Verisign/Thawte/etc.)!

\texttt{EAPTLS\_PrivateKeyFile \textless path to private key\textgreater}
EAPTLS_NoAckRequired
Workaround for broken TTLS clients (like some versions of Mac OSX according to the old documentation and source)

AutoMPPEKeys
The *most* important directive in the config. This is the 802.1x magic-sauce which returns the necessary public/private keying material to the NAS and client to perform per-user, per-session encryption over WPA2 networks. Without this even successful authentications will not result in a successful association to the infrastructure as the keying material will be absent.
Example Inner-Handler - details (con't)

<Handler Client-Identifier=fromCampus, \n   User is on Campus

Recall: This handler must occur before the outer-handler in the configuration or it will never be reached!
Example Inner-Handler - details (con't)

<Handler Client-Identifier=fromCampus, \nUser is on Campus
NAS-Identifier="eduroam", \nUser is using the eduroam SSID
Example Inner-Handler - details (con't)

<Handler Client-Identifier=fromCampus, \       
  User is on Campus
NAS-Identifier="eduroam", \                     
  User is using the eduroam SSID
TunnelledByPEAP=1, \                              
  ... and the user is now in a tunneled mode (PEAP but could be TTLS)
Example Inner-Handler - details (con't)

<Handler Client-Identifier=fromCampus,
   User is on Campus
NAS-Identifier="eduroam",
   User is using the eduroam SSID
TunneledByPEAP=1,
   ... and the user is now in a tunneled mode (PEAP but could be TTLS)
Realm=/^(?:[a-z0-9]+\.)+institution\.edu$/i >
   just like the Outer-Handler example the user's realm is valid
Example Inner-Handler - details (con't)

Now we can actually authenticate if all of the check-items are valid:

AuthBy NTLMAuthentication
  Or a different AuthBy identifier as appropriate for your institution

AuthLog authlogger
  This handler logs both successful and failed authentications

AcctLogFileName %L/%Y/acct_log.%y%m%d
  Setup your RADIUS accounting per institution policy
</Handler>
PreAuthHandler - some magic sauce

nashid_hook.pl - magic sauce for pushing RADIUS attributes into tunneled methods:

```perl
sub {
  my $p = @{$_[0]}; # Request Object reference
  my $outer = $p->{outerRequest};
  my $nasid = $outer->get_attr('NAS-Identifier');
  $p->add_attr('NAS-Identifier', $nasid);
  return;
}
```

Of course NAS-Identifier could be any other RADIUS attribute that must be present in the inner-tunnel for filtering.
Default handler to eduroam-US

<Handler>
  <AuthBy RADIUS>
    Secret <snip>
    Host eduroam1.ns.utk.edu
    AuthPort 1812
    AcctPort 1813

    Radiator defaults to 1645/1646 for Auth/Acct but eduroam uses "modern" ports of 1812/1813 by default. The eduroam-US TLRS also binds 1645/1646 just to eliminate configuration mistakes but we prefer to use 1812/1813 for consistency.
  </AuthBy>

  AuthLog authlogger
</Handler>
Other Handlers

Other, very similar handlers to those above, must be created for requests from eduroam, to handle TTLS if necessary, and from campus but on the campus secure wireless SSID (if it exists). Simple permutations of the above should suffice.

If eduroam is the only WPA2-enterprise network on campus the NAS-Identifier is not required as a filter, so the PreAuthHandler may be omitted. The only important check is that every request on the eduroam SSID, whether local or to be forwarded, contains a realm. If it does not, drop and log it, and make sure the help-desk has access to that information.
EAP methods

Extensible Authentication Protocol
Extensible Authentication Protocol

Wikipedia says...

- LEAP
- EAP-TLS
- EAP-MD5
- EAP-PSK
- EAP-TTLS
- EAP-IKEv2
- EAP-FAST
- EAP-SIM
- EAP-AKA
- EAP-AKA'
- EAP-GTC
- EAP-EKE
- PEAP

- Authentication framework
- Message format (not wire protocol)
- Used to validate identity
- WPA/WPA2-Enterprise certification mandates support for...
  - EAP-TLS
  - EAP-TTLS/MSCHAPv2
  - PEAPv0/EAP-MSCHAPv2
  - PEAPv1/EAP-GTC
  - PEAP-TLS
  - EAP-AKA (May 2009)
  - EAP-SIM (May 2009)
Typically, this subset...

- **EAP-TLS**
  - Transport Layer Security
  - Foundation of EAP-TTLS and PEAP

- **EAP-TTLS**
  - Tunneled-TLS
  - Secures clear-text passwords

- **PEAP**
  - Protected-EAP
  - Secures clear-text passwords

- Others...
EAP-TLS (Transport Layer Security)

● Benefits
  ○ Trusted security model
  ○ Validates end user device and infrastructure
  ○ Phishing is harder

● Drawbacks
  ○ Requires Public Key Infrastructure
EAP-TTLS (Tunneled TLS)

- Benefits
  - Flexible
  - Validates infrastructure

- Drawbacks
  - No (native) windows support! (need SecureW2/etc)
  - Potential for MITM (need user education!)
  - No password expired/locked/etc notification
  - Identity exposure
PEAP (Protected EAP)

● Benefits
  ○ Works on most platforms
  ○ Validates infrastructure (e.g. https for radius)

● Drawbacks
  ○ Potential for MITM (need user education!)
  ○ No password expired/locked/etc notification
  ○ Identity exposure
  ○ Silently fails without correct CA in Trusted Root Store
LEAP (Lightweight EAP)

- **Benefits**
  - ?

- **Drawbacks**
  - Relies on MSCHAPv1 which is easily compromised and has been broken since 2004
EAP-FAST (Flexible Authen. via Secure Tunneling)

● Benefits
  ○ ?

● Drawbacks
  ○ Relies on MSCHAPv1 which is easily compromised and has been broken since 2004
In development: EAP-FASTv2

● Benefits
  ○ EAP Channel Binding
    ■ http://tools.ietf.org/html/draft-ohba-eap-channel-binding-02
  ○ Additional password verification methods
  ○ TLS enhancements

● Drawbacks
  ○ Still in IETF Draft, too early to judge
    ■ http://www.ietf.org/id/draft-zhou-emu-eap-fastv2-00.txt
In development: TEAM
(Tunneled Extensible Authentication Method)

● Benefits
  ○ Notification methods!
    ■ Bad password
  ○ Fragmentation Reassembly
    ■ RadSec addresses this too

● Drawbacks
  ○ Still in IETF Draft, too early to judge
    ■ http://tools.ietf.org/id/draft-zorn-emu-team-01.txt
University of Alaska case

- **Planning**
  - Existing RADIUS
  - FreeRADIUS
  - S/MIME!?  
  - TTLS

- **Build out**
  - TTLS
  - Joined eduroam
  - PEAP added for Windows

- **Pilot/beta**
  - Not broadcast
  - Several client-side issues
University of Alaska case: pre-prod

- Back to the drawing board
  - Tokens
  - TLS

- Rush to implement
  - Mistakes
Univ of Alaska case: production

- KISS
  - Least complicated, customer-facing
  - Certs easier behind the scenes
  - Trade-off

- Security, user-/client-side
  - Harder to Phish
  - Compromised un/pw irrelevant
Making the Certs Available

certificate authority

- **University of Alaska eduroam CA Root**
  - Must be installed in the operating system, under *Trusted Certificate Authorities*.
  - eduroam installer does this automatically.

identity certificates

- gfjohnson@alaska.edu -- download in *PEM* or *PKCS12* format

  **NOTE** Certificate password is *gfjohnson*.

eduroam installers

- *UAK_eduroam_Win_0.0.0.9.exe*
Getting to the Certs

[Image of a login page for University of Alaska]

Username or ID: gfjohnson
Password: ********
Login

Password Reset

Support: helpdesk@alaska.edu
Installing the Identity Cert - 1
Installing the Identity Cert - 2

Certificate Import Wizard

File to Import
Specify the file you want to import.

File name:

Browse...

Note: More than one certificate can be stored in a single file in the following formats:

- Personal Information Exchange- PKCS #12 (.PFX, .P12)
- Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)
- Microsoft Serialized Certificate Store (.SST)
Installing the Identity Cert - 3

Certificate Import Wizard

Password
To maintain security, the private key was protected with a password.

Type the password for the private key.

Password:
**********

- Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.

- Mark this key as exportable. This will allow you to back up or transport your keys at a later time.
Installing the Identity Cert - 4

Certificate Import Wizard

Certificate Store
Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for

- Automatically select the certificate store based on the type of certificate
- Place all certificates in the following store

Certificate store:  

Browse...
Installing the Identity Cert - 5
Deploying on WinXP/Vista/7

● Certificate Authority

● Installer Components
  ○ Null-Soft Installation System
    ■ WinXP SP3 through Win7

● Some assembly required...
Customer Experience

University of Alaska; iPhone
Step 1: Open Safari

Open Safari from your device's home screen.

Step 2: Open profile page

Navigate to https://nah.alaska.edu/iphone/

Step 3: Login

Username or ID:

Password:

Login

Password Reset

Support: helpdesk@alaska.edu

You need to login before you can download the profile.

1) Enter your UA Username.

2) Enter your password - the same one you use for Google Apps @ UA.

3) Tap the Login button.

EAP-TLS (Univ Alaska, iPhone)
Step 4: Select the Eduroam Profile

Tap on the "eduroam profile" from the list of profiles.

Step 5: Install the Eduroam Profile

Click the "Install" button.

Step 6: Install Root Certificate

When prompted, click the "Install Now" button to install the "University of Alaska eduroam CA Root" certificate.

Step 7: Enter Your Passcode

If you have a passcode set on your device, you must enter your passcode to authorize the installation of this certificate. This screen will look different if you use a password instead of passcode to lock your device.

EAP-TLS (Univ Alaska, iPhone)
**Step 8: Enter your UA Password**

Enter the password for the certificate "the password is _____________."

Enter your **UA Username** -- the same one you use for Google Apps @ UA -- without @alaska.edu. Then tap **Next**.

**Step 9: Eduroam Profile Installed**

The eduroam profile has been successfully installed. Tap **Done**.

**Step 10: Connect to Eduroam**

Return to the Home screen by pressing the Home button on your device.

Open the **Settings** application, then **Wifi**.

Tap on the eduroam network option to connect to it.
Deploying eduroam; iPhone

- Certificate Authority

- Enterprise Deployment - iOS
  - Very nice!

- How to DIY
  1. Generate a working .mobileconfig
  2. Take apart enough to understand (it's Plist formatted)
  3. Build script, to generate custom .mobileconfig
Provisioning iPhone/iPad - Gotchas

● Don't mess with PayloadVersion

● Careful with GUIDs

● Embedded Identity Certificates
  ○ PKCS12 format, wrapped with Base64
  ○ Password required

● "Verified" status: sign with web-server cert
  ○ webserver private key required

    cat iphone.mobileconfig | /usr/bin/openssl smime -sign -signer server.crt -inkey server.key -certfile intermediate-chain.crt -outform der -nodetach >iphone
Connecting authentication backends and RADIUS

MIT Kerberos
Using MIT Kerberos for Authentication

- Easy to do in FreeRADIUS, although it wasn’t well documented when I did it
- Allows users to use EAP-TTLS with their kerberos username and password
- Great to use if you already have MIT Kerberos deployed extensively, otherwise you should look somewhere else
Getting Started

● Compile FreeRADIUS with MIT Kerberos support, or install the appropriate binary (RedHat uses freeradius2-krb5)
● Generate a keytab for your RADIUS servers
● Setup your RADIUS server to use EAP-TTLS with PAP as the inner authentication
  ○ Details on how to do this in FreeRADIUS on the next slide
● Configure your supplicant appropriately
FreeRADIUS Config

- Edit krb5 module under the $raddb/modules directory
  
  krb5 {
    keytab = $raddb/keytab
    service_principal = host/hostname.domain.edu
  }

FreeRADIUS Config

- Add DEFAULT Auth-Type = Kerberos to your users file
- Modify the *inner-tunnel* and *default* files under `/etc/raddb/sites-available`. In both files under the authenticate section and right below the PAP configuration line add the following
  - Auth-Type Kerberos {
    krb5
  }
Supplicant Support

- EAP-TTLS with PAP inner-auth
- Operating systems with built in support
  - Mac OSX
    - Need to configure it to use pap as the inner auth, not the default EAP-TTLS config
  - Linux
  - IOS
    - Need to use a profile to configure it
  - Android
- Windows needs a third party client
  - I’ve used SecureW2 as a supplicant
Password Changes

- If the user changes passwords, it doesn't fail gracefully.
- The user will need to touch each device to update the password stored there.
Connecting authentication backends and RADIUS

Samba and OpenLDAP for MSCHAPv2 without ActiveDirectory
Samba + OpenLDAP

(P)EAP(-TTLS) + MSCHAPv2
without Active Directory
The goal:

The native Windows supplicant only supports PEAP (TLS and MSCHAPv2) by default. MSCHAPv2 authentication either requires an ActiveDirectory and a RADIUS server that can speak AD natively or has the “glue” to talk to AD (e.g. ntlm_auth on *NIX) or something that emulates and ActiveDirectory, for example, Samba with OpenLDAP!
Implementation Overview

1. Setup `slapd` (OpenLDAP) for basic LDAP functionality (Ubuntu: [https://help.ubuntu.com/11.04/serverguide/C/openldap-server.html](https://help.ubuntu.com/11.04/serverguide/C/openldap-server.html))
   ○ Note: This document outlines not only basic setup but replication for High-Availability, Encryption, etc. YMMV.
   ○ If you have an existing LDAP backend you will need to modify your existing Schema to support Samba’s schema.

1. Fix your `cn=admin,cn=config` DN password via [http://ubuntuforums.org/showthread.php?t=1515119](http://ubuntuforums.org/showthread.php?t=1515119) (but replace MD5 with SHA) using the same password you configured earlier. Now you can use the DN appropriately.

Adding MSCHAPv2 support to an existing LDAP installation (outline):

- Configure Samba as an Domain Controller
- Add Samba schema to LDAP
- Install and configure `smbldap_tools` package to update Samba passwords
- Update a test user password and test that change via `ntlm_auth`
- Require all users to “reset” their passwords (perhaps to the same password) to update the existing hashed value (MD5, SHA1, etc) and an NTLM hash (sambaNTPassword). This is true unless passwords are stored in plaintext in LDAP then you can script the migration (and maybe even migrate to hashed passwords in one pass)
Connecting authentication backends and RADIUS

Caveats:

- Users pick inner authentication methods (PAP, MSCHAPv2, etc...). Cannot be disabled on RADIUS

- If PAP is used RADIUS admin of home institution can see the password!
eduroam Network Design Considerations

University of Alaska
Network Design: University of Alaska

- Private IPv4
  - Address conservation
  - Handle transient utilization spikes

- Public IPv6 (planned)

- Common access method (hopefully)
  - Regional and Community campuses

- Security
  - Local network limitations
    - May arbitrarily be applied by administrators
    - GuestACL: Only allow http/s, imap/s, pop3/s, vpn, icmp, & dns
    - bps restrictions
eduroam Network Design Considerations

Internet2
Current eduroam Deployment

- We use eduroam as the main SSID for all of our staff to connect to
- We have one wireless controller and use a GRE tunnel to connect our 2 offices
- We have a RADIUS server in each office, as well as a Kerberos server
- We are starting to have one-off APs broadcasting the eduroam SSID at teleworker locations
- We use a web portal for guests that do not have eduroam
Future eduroam Deployment

- Add a 2nd controller at the remote office
- Point all of our access points at both
  - This will provide good redundancy
    - It will mean that if the main office goes down then the web portal will still work at the remote office
    - It will also mean that all of the great features of having a controller will work even in a partial outage
Conference eduroam setup

- We provide an open SSID with a web registration page for everyone without eduroam
- Historically we have used Fat APs, but have since moved to a controller based architecture
- We have the controller onsite with us, but we are planning on keeping the controller back at our office and using GRE tunnels in the future
- We use offsite RADIUS servers, but make sure they are geographically diverse
Future eduroam Conference Setup

- In the future we will leverage our existing controller infrastructure to manage the conference APs.
- This will give us the same redundancy as our offices, and make deployment simpler.
- Build a GRE tunnel from conference to each office and tunnel controller traffic over that to APs.
eduroam Network Design Considerations
University of Tennessee
RADIUS

- Two RADIUS servers serving all 802.1x needs for the University, peering with top level eduroam-US
- Separate REALM based on SSID
  - ut-wpa2 comes as one handler
  - eduroam comes as another handler and enforces the form user@utk.edu to prepare users for traveling
eduroam network design: today

- Public IP addresses different from ut-wpa2 (University's 802.1x based network) for firewall purposes

- eduroam subnets are currently routed to our core and pass IPS on the way out AND on the way in. no IPS for internal traffic (unlike our web based visitor network)

- 1022 available IP addresses (eduroam is used by UTK users as well)

- no web page required to join (defeats eduroam instant access)

- HelpDesk supports all calls (even from non utk-eduroam)
eduroam network design: tomorrow

- migrate ut-wpa2 and eduroam in one SSID: eduroam

- differentiate traffic by REALM: @utk.edu receives more privileges than all others. Using role based networking we can either assign different IP addresses OR filter traffic based on REALM and/or attributes.

- migrating SSID and doing REALM (with roles) based networking simplifies IP distribution
eduroam network design: tomorrow (cont.)

- Two SSID for the entire WLAN: ut-open and eduroam
  - ut-open: Web based SSID with 4 options on page:
    - Visitor without credentials
    - Visitor with temporary credentials
    - NetReg based network for non 802.1x devices (DHCP enforce option turned on)
    - 802.1x configurator
  - eduroam: for all encrypted Wireless LAN purpose

- The web based SSID also gives the option to communicate with the public. 802.1x is pretty terse in that respect!

- and a third SSID for ticket scanners: wpa2-psk
Example Wireless Configs

Cisco LWAP
Login!
### RADIUS Authentication Servers

- **Call Station ID Type**
  - IP Address

- **Use AES Key Wrap**
  - Designed for FIPS customers and requires a key wrap compliant RADIUS server

- **MAC Delimiter**
  - Hyphen

<table>
<thead>
<tr>
<th>Network User</th>
<th>Management</th>
<th>Server Index</th>
<th>Server Address</th>
<th>Port</th>
<th>IPSec</th>
<th>Admin Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enabled</td>
<td></td>
</tr>
</tbody>
</table>

**Cisco Security > AAA > RADIUS > Auth**
Add new radius server...

RADIUS Authentication Servers > New

Server Index (Priority) 2
Server IP Address
Shared Secret Format ASCII
Shared Secret
Confirm Shared Secret
Key Wrap
Port Number 1812
Server Status Enabled
Support for RFC 3576 Enabled
Server Timeout 2 seconds
Network User Enable
Management Enable
IPSec

(Designed for FIPS customers and requires a key wrap compliant RADIUS server)
### WLANs > Create New... > General

WLANs > Edit 'eduroam'

<table>
<thead>
<tr>
<th>General</th>
<th>Security</th>
<th>QoS</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Name</td>
<td>eduroam</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>WLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSID</td>
<td>eduroam</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Enabled</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Security Policies

- MAC Filtering [WPA + WPA2][Auth(802.1X)]
  
  (Modifications done under security tab will appear after applying the changes.)

#### Additional Configuration

- Radio Policy: **All**
- Interface/Interface Group(G): management
- Multicast Vlan Feature: Unchecked
- Broadcast SSID: Enabled
WLANs > Create new... > Security > Layer 2

WLANs > Edit 'eduroam'

Layer 2 Security 6: WPA+WPA2
- MAC Filtering

WPA+WPA2 Parameters:
- WPA Policy: checked
- WPA Encryption: AES, not TKIP
- WPA2 Policy: checked
- WPA2 Encryption: AES, not TKIP
- Auth Key Mgmt: 802.1X

MITM packet injection vulnerability since 2008
WLANs > Create new... > Security > Layer 3
WLANs > Create new... > Security > Advanced

Select AAA servers below to override use of default servers on this WLAN

**Radius Servers**

<table>
<thead>
<tr>
<th>Authentication Servers</th>
<th>Accounting Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

Server 1: 199.299.299.299.

Server 2: 299.1.99.999.

Server 3: 199.299.299.299.

**Local EAP Authentication**

Local EAP Authentication: Enabled
WLANs > Create new... > Advanced

- Allow AAA Override: Enabled
- Coverage Hole Detection: Enabled
- Enable Session Timeout: Unchecked
- Aironet IE: Enabled
- Diagnostic Channel: Unchecked
- IPv6 Enable: Unchecked
- Override Interface ACL: None
- P2P Blocking Action: Disabled
- Client Exclusion: Enabled, Timeout Value (secs): 60

Off Channel Scanning Defer
- Scan Defer Priority: 0 1 2 3 4 5 6 7
- Scan Defer Time (msecs): 100

H-REAP
- H-REAP Local Switching: Enabled
- Learn Client IP Address: Enabled

DHCP
- DHCP Server: Override
- DHCP Addr. Assignment: Required

Management Frame Protection (MFP)
- MFP Client Protection: Optional

DTIM Period (in beacon intervals)
- 802.11a/n (1 - 255): 1
- 802.11b/g/n (1 - 255): 1

NAC
- State: Enabled

Load Balancing and Band Select
- Client Load Balancing: Unchecked
- Client Band Select: Unchecked

Voice
- Media Session Snooping: Enabled
- Re-anchor Roamed Voice Clients: Enabled
Example Wireless Configs

Cisco LWAP for Remote Offices
eduroam on Cisco Controller when using HREAP and remote offices
Design Considerations

- Do you want all of your wireless traffic funneling back through central controllers
  - We decided that we didn't want that
  - However we did want the features of a controller based wireless system
    - channel management, ease of deploying a new access point, ease of configuration

- Is your wireless all within one LAN? Does it span a WAN?
  - Ours spanned a WAN between our 2 locations

- If it spans a WAN how will remote access points talk back to the controller?
  - We built a GRE tunnel between the 2 locations and tunnel some RFC-1918 addresses over it
What mode to run the APs in

- We chose to use Cisco's HREAP mode for our wireless gear.
- It allowed us to have all the features we wanted with a controller-based system, and let the APs switch traffic locally.
  - We did not want our remote office traffic coming to the central controller before going out.
- We currently only have 1 controller, so it meant that we could lose the controller and wireless would still function.
What extra config needs to happen?

- APs need to know about VLANs and what SSID belongs on which one
  - You need to manually map the VLANs to the SSIDs and configure the backup RADIUS servers
  - You can't just plug in an AP, have the controller discover it, and it magically work
Advantages

● Rely on current switching infrastructure to distribute the wireless traffic load
● Your wireless controller does not need large amounts of connectivity
  ○ It is only handling administrative traffic, the real wireless traffic never touches it
● Great for people who have multiple locations over a WAN and don't want multiple controllers at each location
Drawbacks

● This works great in a relatively small deployment
  ○ 50 or less APs
● More work in larger deployments, but may be worth it depending on how your network is designed
Example Wireless Conﬁgs

Aruba Networks
Aruba Wireless Configuration for eduroam - Overview

Configuring Aruba Wireless controllers for eduroam is no different than any other 802.1x wireless network on your campus.

1. Create RADIUS Server(s)
   ○ Configuration > Authentication > Servers > RADIUS Server > Add
1. Create RADIUS Server Group
   ○ Configuration > Authentication > Servers > Server Group > Add
2. Create 802.1x Group Auth. Profile
   ○ Configuration > Authentication > L2 Auth. > 802.1x Auth. Profile > Add
3. Create User Roles
   ○ Configuration > Access Control > User Roles > Add
4. Create AAA Profile
   ○ Configuration > Authentication > AAA Profiles > Add
5. Create SSID Profile
   ○ Configuration > All Profiles > Wireless LAN > SSID Profile > Add
6. Create Virtual AP
   ○ Configuration > All Profiles > Wireless LAN > Virtual AP Profile > Add
   ○ Select SSID and AAA Profiles created above
Create RADIUS Server(s)

<table>
<thead>
<tr>
<th>Host</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAS ID</td>
<td>eduroam</td>
</tr>
<tr>
<td>Auth Port</td>
<td>1812</td>
</tr>
<tr>
<td>Retransmits</td>
<td>3</td>
</tr>
<tr>
<td>NAS IP</td>
<td>[IP-Address]</td>
</tr>
<tr>
<td>Use MD5</td>
<td>[Tick]</td>
</tr>
<tr>
<td>Mode</td>
<td>[Tick]</td>
</tr>
</tbody>
</table>

Configuration > Authentication > Servers > RADIUS Server > Add
Create RADIUS Server Group

Configuration > Authentication > Servers > Server Group > Add

server configs include NAS-Identifier of "eduroam"
Create 802.1x Group Auth. Profile

<table>
<thead>
<tr>
<th>Basic</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max authentication failures</td>
<td>0</td>
</tr>
<tr>
<td>Enforce Machine Authentication</td>
<td>□</td>
</tr>
<tr>
<td>Machine Authentication: Default Machine Role</td>
<td>guest</td>
</tr>
<tr>
<td>Machine Authentication: Default User Role</td>
<td>guest</td>
</tr>
<tr>
<td>Reauthentication</td>
<td>□</td>
</tr>
<tr>
<td>Termination</td>
<td>□</td>
</tr>
<tr>
<td>Termination EAP-Type</td>
<td>□ eap-tls □ eap-peap</td>
</tr>
<tr>
<td>Termination Inner EAP-Type</td>
<td>□ eap-mschapv2 □ eap-gtc</td>
</tr>
</tbody>
</table>
Create User Roles

<table>
<thead>
<tr>
<th>Name</th>
<th>Firewall Policies</th>
<th>Bandwidth Contract</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>default-via-role</td>
<td>allowall</td>
<td>Up: Not Enforced Down: Not Enforced</td>
<td>Show Reference Edit Delete</td>
</tr>
<tr>
<td>eduroam-logon</td>
<td>blockall</td>
<td>Up: Not Enforced Down: Not Enforced</td>
<td>Show Reference Edit Delete</td>
</tr>
</tbody>
</table>

Configuration > Access Control > User Roles > Add
Create AAA Profile

<table>
<thead>
<tr>
<th>AAA Profile &gt;</th>
<th>eduroam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial role</td>
<td>eduroam-logon</td>
</tr>
<tr>
<td>MAC Authentication Default Role</td>
<td>guest</td>
</tr>
<tr>
<td>802.1X Authentication Default Role</td>
<td>eduroam-authenticated</td>
</tr>
<tr>
<td>User derivation rules</td>
<td>--NONE--</td>
</tr>
<tr>
<td>Wired to Wireless Roaming</td>
<td></td>
</tr>
<tr>
<td>MAC Authentication Profile</td>
<td>default</td>
</tr>
<tr>
<td>MAC Authentication Server Group</td>
<td>default</td>
</tr>
<tr>
<td>802.1X Authentication Profile</td>
<td>eduroam</td>
</tr>
<tr>
<td>802.1X Authentication Server Group</td>
<td>eduroam</td>
</tr>
<tr>
<td>RADIUS Accounting Server Group</td>
<td>eduroam</td>
</tr>
<tr>
<td>XML API server</td>
<td></td>
</tr>
<tr>
<td>RFC 3576 server</td>
<td></td>
</tr>
</tbody>
</table>

Configuration > Authentication > AAA Profiles > Add
Create SSID Profile

<table>
<thead>
<tr>
<th>Network</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Name (SSID)</td>
<td><strong>eduroam</strong></td>
</tr>
<tr>
<td><strong>802.11 Security</strong></td>
<td></td>
</tr>
<tr>
<td>Network Authentication</td>
<td>None, 802.1x/WEP, WPA, WPA-PSK, WPA2</td>
</tr>
<tr>
<td>Encryption</td>
<td>AES</td>
</tr>
</tbody>
</table>

**Keys**

- AES

---

Configuration > All Profiles > Wireless LAN > SSID Profile > Add
SSID Profile (cont')

Advanced Settings (identical to Basic)
Create Virtual AP

Configuration > All Profiles > Wireless LAN > Virtual AP Profile > Add
Supplicant Provisioning

Scripting Tools and XpressConnect
Aruba Networks 1x Configurator

- Located on support site under tools
- Windows XP and Vista only
- DOS shell script
- Designed to help customers (not for general consumption)
SU1X

http://www.ja.net/services/authentication-and-authorisation/janet-roaming/su1x.html

SU1X 802.1X Configuration Deployment Tool

Configuring Windows supplicant software is not technically a difficult task, even with the additional complication of including details about an institutional RADIUS server certificate or certificate distribution. However users are generally students and staff who don’t have much knowledge about or interest in wireless networks or login mechanisms. For such users, configuring devices properly for use on 802.1X networks can be difficult. Due to the nature of the many different configuration options, step by step instruction guides, even with screen-shots, can be quite daunting for the average user who does not wish to know about wireless ciphers; username including realm; domain blank; roaming identity; authentication type: EAP TTLS/PAP, EAP TTLS/MSCHAP, PEAP/MSCHAPv2; RADIUS server certificate validation; RADIUS server name.

A major step has now been taken towards solving at least this latter problem of wide scale deployment of 802.1X configuration on Windows devices. JANET(UK) is pleased to have supported the development of the open source SU1X 802.1X Configuration Deployment Tool developed by Gareth Ayres at Swansea University in association with Loughborough University.

The SU1X Tool is now for available for general use by network managers and can be freely downloaded, complete with comprehensive documentation.

Download SU1X now
SU1X

- Windows XP, Vista, 7
Xpressconnect

- Commercial: From CloudPath Networks
- Unique Niche (no competition so far)
- Increases 802.1x adoption by leaps and bonds
- Amazing versatilty
  - Auto-detection of OS
  - Support of "all" OS (Windows, Mac, iOS, Android, Unbuntu)
  - BlackBerry in the works
- Mini NAC built-in (can turn OS' features ON)
Xpressconnect-script-setup-init
Step 1 of 12: Network Name

Please provide a name and a description for this new network. The network name is typically something like "Sample Corp Wireless Network" or "Sample University Student & Faculty Network".

If only one network exists, this information will not be displayed to the user.

When multiple networks exist, this information assists the user in selecting the appropriate network. To see how it is displayed to the user, click here. In the 'Description' field, it is helpful to mention who is expected to use the network (employees vs guests). Also, it is helpful to mention the requirements for using the network, such as the need to authenticate, accept terms & conditions, etc.

Provide a name for this new network:

- Network Name: $eduram$
- Description: The encrypted Wireless Network for UTK. Will also allow you to roam freely to other locations that have "eduram" (check website for locations)
Step 2 of 12: Connection Method

A single network may support wired and/or wireless network connections. If the network is configured to support both, XpressConnect will configure the machine based on the active network connection.

Select the connection method(s) this network supports:

- [ ] Wired Connections
  - If selected, XpressConnect will configure wired connections.
- [x] Wireless Connections
  - If selected, XpressConnect will configure wireless connections.
  - The following fields are necessary for wireless connections:
    - SSID:
    - Authentication:
      - WPA2
    - Encryption:
      - AES
    - Protocol Preference:
      - Any (Default)
        - N Only
        - A Only
        - A or G Only
        - G Only
        - B Only
        - A or B Only
        - A or G or B Only
        - A or B or G Only
        - A or B or G Only
        - N Only
        - A or B or G or N Only
    - Is the SSID broadcast?
      - [ ] Any
        - [ ] Always
        - [ ] Never

Step 3 of 12: Conflicting SSIDs

The following settings control the manner in which XpressConnect resolves conflicts with other SSIDs in the environment. XpressConnect will always ensure that the configured SSID is at the top of the priority list. However, operating systems will occasionally make a sub-optimal decision to roam away from the secure SSID to open SSIDs in the area.

This setting is used to prevent the machine from making a sub-optimal decision to roam to other SSIDs in the area. We recommend specifying the list of open SSIDs within your environment, such as the help SSID and guest SSID if appropriate. The 'Conflicting SSIDs' field may be a single SSID (mySSID1), a semicolon separated list of SSIDs (mySSID1;mySSID2;mySSID3), or a wildcard (*). A wildcard causes the currently associated SSID to be set to 'connect manually' or deleted.

Specify how conflicting SSIDs should be handled:

Behavior: Set these SSIDs to 'Connect Manually'.

Conflicting SSIDs: 

[Options for specifying SSIDs]
Select the type of authentication used on this network:

- **This network uses 802.1X.**
  
  If selected, XpressConnect will configure the user for access based on 802.1X. The selection of an EAP type is necessary for 802.1X:
  
  - Supplicant Preference: **Native**
  - EAP Type: [ ] EAP-FAST
  - [ ] PEAP/MSCHAPv2
  - [ ] EAP/TLS
  - [ ] TLS

- **This network uses a web-based login.**
  
  If selected, XpressConnect will automatically open the browser to the login page once they are migrated to the new network.
  
  Login URL: [http://](http://)

- **This network does not use authentication.**
Step 5 of 12: Operating Systems

A network may support one or more operating systems. After the wizard is completed, a profile will be created for each operating system selected below. If an operating system is excluded, support for it may be added later.

Select the operating systems that will be supported:

- Windows XP
- Windows Vista & 7
- Mac Tiger
- Mac Leopard, Snow Leopard & iPhone
- Ubuntu
- Android
Step 6 of 12: Authentication Behavior

XpressConnect contains the ability to prompt for credentials prior to performing the migration. The credentials are simply held in memory before being provided to the supplicant. Allowing XpressConnect to prompt for credentials before migration has the following advantages:

- The prompting is not time-sensitive, so the user can type as slowly as they wish.
- Messaging can be customized to assist the user with their credentials.
- If the user needs to create an account or reset their password, they can be provided a link to the appropriate website since the user is still attached to the open SSID or guest VLAN.
- A regular expression can be used to help validate the username.

What behavior do you want for each operating system?

**Windows XP**
- Prompt for credentials before migration.
  - The user will be prompted to supply their credentials before they are migrated to the secure network. This allows the user to bypass the bubble.
- Use only the standard Windows behavior.
  - The user is prompted for their credentials using the standard Windows bubble and 802.1X login form.

**Windows Vista & 7**
- Prompt for credentials before migration.
  - The user will be prompted to supply their credentials before they are migrated to the secure network. This allows the user to bypass the bubble.
- Use only the standard Windows behavior.
  - The user is prompted for their credentials using the standard Windows bubble and 802.1X login form.

**Mac Tiger**
- Use the standard Mac behavior.
  - The user is prompted for their credentials using the standard Mac Internet Connect form.

**Mac Leopard, Snow Leopard & iPhone**
- Prompt for credentials before migration.
  - The user will be prompted to supply their credentials before they are migrated to the secure network. The credentials are pre-cached for the supplicant in the keychain. This helps work-around some issues with the supplicant wherein it occasionally fails to update the keychain with the new password.
- Use only the standard Mac behavior.
  - The user is prompted for their credentials using the standard Mac popup form.

**Ubuntu**
- Prompt for credentials before migration.
  - The user will be prompted to supply their credentials before they are migrated to the secure network.

**Android**
- Prompt for credentials before migration.
  - The user will be prompted to supply their credentials before they are migrated to the secure network.
Step 7 of 12: Authentication Behavior

Most environments refer to their username format by a name. Sometimes, this is simply the user's email address. Other times, it has a special name like EID. For example, when a user asks how to authenticate, you would tell them that their user name is their __________.

What is a descriptive name of the user's username?
NaID@utk.edu

What is a descriptive name of the user's password?
Password for your NaID

Do you use the domain field?
No, we leave the domain field blank.
The image to the right is the screen XpressConnect will display when it prompts the user for their credentials. The settings below fine-tune the user experience to match your environment. If your configuration utilizes Windows login credentials, this screen will not display to Windows users.

**User Name Suffix**

If the user is prompted for credentials before migrating, XpressConnect may ensure that the user name ends with a specified suffix. If this field is set to "@test", the value entered by the user will be appended with '@test'. You may also include a semi-colon-separated list of suffixes that are allowed. (ie @test; @foo) If the user name doesn't have one of these suffixes included, it will be replaced with the first suffix in the list.

**Login Directions**

NetID@utk.edu and Password for your NetID are required to access this network.

**Windows Credentials Submission**

- **User Name Description**: NetID@utk.edu
- **Default User Name**: [Your NetID@utk.edu]
- **User Name RegEx**: 
- **User Name Replacement RegEx**: Public@Xpressconnect.edu
- **User Name Suffix**: @students@faculty
- **User Name Case**: Leave user name unchanged.
- **Password Description**: Password for your NetID
- **Help Link Caption**: Forgot your password?
- **Help Link URL**: https://its.utk.edu/passwords/ or call the HelpDesk at 865-974-9980
Within 802.1X, server certificate validation is an important security feature. When enabled, the client will only authenticate to a server that provides a certificate signed by the selected trusted certificate authority. If server certificate validation is disabled, the client will authenticate to any server. Enabling server certificate validation is a security best practice.

Server certificate validation may be changed later using the "Define Networks" tab. If you need to upload multiple CA certificates, upload one here, and upload the additional certificates on the "Define Networks" tab. If you are uncertain about the certificate configuration, contact support@cloudpath.net and we can assist you.

If you choose not to enforce server certificate validation, we recommend adding the server CA certificate to the Mac Leopard & Snow Leopard profile so that XpressConnect can mark the certificate trusted and avoid the user prompt to accept the certificate. This may be done on the Mac Leopard & Snow Leopard tab by clicking Add Additional Application Settings.

Select the appropriate setting for server certificate validation:

- Disable server certificate validation.
  - If selected, the user will not validate the certificate provided by the authentication server.

- Enable server certificate validation.
  - If selected, the user will validate the certificate provided by the authentication server. If you have a custom Root CA certificate, you may upload it, and it will be installed for the user. If you use a public Root CA certificate, select it from the list of CAs.

  - My own CA certificate file [Choose File] no file selected
  - Select from the list of common & previously uploaded certificate authorities.

In addition to verifying the CA, some operating systems can verify the name of the RADIUS server, which is stored in the server certificate. When using a public CA, verifying the server name is a good practice to ensure that computers only authenticate against your RADIUS server, since anyone may purchase a certificate from the public CA.

If multiple RADIUS servers exist, you have two options. You may specify a name using a wildcard ("*.sample.edu") or you may specify a semicolon-separated list ("radius1.sample.edu;radius2.sample.edu"). If using a wildcard, specify the name using an asterisk ("*.sample.edu") and XpressConnect will translate if another format is needed.

- Restrict server name to

When server certificate validation is enabled, it is important that the computer’s system clock be reasonably accurate. If the system clock is off considerably, it may incorrectly reject the server certificate. This setting will verify that the user’s system clock is within 7 days of being correct. If not, the user will be asked to set it appropriately.

- Check system clock accuracy.
Step 9 of 12: Server Certificate Validation

Within 802.1X, server certificate validation is an important security feature. When enabled, the client will only authenticate to a server that provides a certificate signed by the selected trusted certificate authority. If server certificate validation is disabled, the client will authenticate to any server. Enabling server certificate validation is a security best practice.

Server certificate validation may be changed later using the "Define Networks" tab. If you need to upload multiple CA certificates, upload one here, and upload the additional certificates on the "Define Networks" tab. If you are uncertain about the certificate configuration, contact support@cloudpath.net and we can assist you.

If you choose not to enforce server certificate validation, we recommend adding the server CA certificate to the Mac Leopard & Snow Leopard profile so that XpressConnect can mark the certificate trusted and avoid the user prompt to accept the certificate. This may be done on the Mac Leopard & Snow Leopard tab by clicking Add Additional Application Settings.

Select the appropriate setting for server certificate validation:

1. **Disable server certificate validation.**
   - If selected, the user will not validate the certificate provided by the authentication server.

2. **Enable server certificate validation.**
   - If selected, the user will validate the certificate provided by the authentication server. If you have a custom Root CA certificate, you may upload it, and it will be installed for the user. If you use a public Root CA certificate, select it from the list of CAs.

   - **My own CA certificate file**
     - Select from the list of common & previously uploaded certificate authorities.

- **AAA Certificate Services**
- **AAA Certificate Services**
- **AddTrust External CA Root**
- **AddTrust External CA Root**
- **Baltimore CyberTrust Root**
- **Class 1 Primary CA**
- **Class 1 Public Primary Certification Authority**
- **Class 1 Public Primary Certification Authority**
- **Class 2 Primary CA**
- **Class 2 Public Primary Certification Authority**
- **Class 2 Public Primary Certification Authority**
- **Class 3 Primary CA**
- **Class 3 Public Primary Certification Authority**
- **Class 3 Public Primary Certification Authority**
- **COMODO Certification Authority**
Step 10 of 12: Additional Options

Based on your answers so far, profiles will be created to handle the configurations you have specified. The options below are settings which are commonly used. You may add, delete, or modify these and other settings at any time on the "Define Networks" tab. If you would like to include any of these settings at this time, select them below.

Do you want to add any of the following settings to the new profiles?

**Windows XP**
- Enable Windows Auto Updates if not enabled.
- Enable Windows Firewall if a firewall is not running.
- Install Impulse SafeConnect NAC Agent 13
- Enable Microsoft Network Access Protection (NAP) (XP SP3 Only)
- Install WPA2 Hotfix If Necessary
- Enable 'Automatically Detect' in IE LAN Settings

**Windows Vista & 7**
- Enable Windows Auto Updates if not enabled.
- Enable Windows Firewall if a firewall is not running.
- Install Impulse SafeConnect NAC Agent 13
- Enable Microsoft Network Access Protection (NAP)
- Enable 802.1X Single Sign-on
- Disable Wireless Hosted Network (Win7 Only)
- Enable 'Automatically Detect' in IE LAN Settings

**Mac Tiger**
- Install Impulse SafeConnect NAC Agent 13

**Mac Leopard, Snow Leopard & iPhone**
- Install Impulse SafeConnect NAC Agent 13
- Enable Leopard Firewall if not running.

**Ubuntu**
- No additional options available.

**Android**
- No additional options available.
Step 11 of 12: Completion Behavior

After the user is connected to the secure network, XpressConnect will verify that the machine receives an IP address. The following settings control the behavior of XpressConnect after the IP address is received.

**How should the wizard behave after completing the migration?**

- Open Browser to URL
  
  If populated, this URL will be opened in the user's default browser as the final page of the wizard is displayed. This is useful for opening a web-login page, a self-registration page, a portal page, or a network policy page.

  **Open Browser to URL:**
  
  [Input Field]

- When Completed:
  
  After migrating to the secure network, XpressConnect may close automatically or display a 'Connected' page to the user. This setting controls whether or not the 'Connected' page displays and if so, whether or not the user is presented with the option to revert.

  Typically, the Connected page is displayed with the revert options.

---

![Connected page with revert option]

*Successfully connected.*

You are now connected to the network.
Your IP address is 192.168.4.160.
Click here to view the modifications made to your computer.

If you would like to revert the changes later, click here to create a shortcut on the desktop.
XpressConnect will now generate the network based on the information provided. Once generated, you may fine-tune the settings within the network.

Profiles will be created based on the information below:

- **Operating Systems:** Windows XP, Windows Vista & 7, Mac Tiger, Mac Leopard, Snow Leopard & iPhone, Ubuntu, Android
- **Connection Methods:** Only wireless. Wireless uses ‘WPA2’ using AES and SSID ‘edu roam’.
- **Authentication Methods:** 802.1X using PEAP/MSCHAPv2
- **Server Cert Validation:** Enabled with 2 predefined CA(s). Also, server name must be ‘radius1.ns.utk.edu;radius2.ns.utk.edu’.
- **Additional Options:** On Windows XP, the following will be enforced: WPA2 Hotfix.

If this information is correct, click ‘Done’.
The "eduroam" network is displayed below. This network should mimic a network within your environment, such as your student or guest network. If multiple networks exist, the end-user will be asked to select the desired network based on the name and description. Once an end-user selects this network, their computer will be configured based on the information contained here.
Cloudpath Administrative Console | University of Tennessee Knoxville

One or more settings are configured for installation-based repair. After downloading a deployment package, please ensure that the following one or two files are placed in the appropriate directory: WindowsXP-KB917021-09-x86-ENU-Language-English.exe

**Ad hoc Deployment**

1. Select your deployment method: **Web Server with HTML**

   This package is designed to be deployed on a web server which supports the scripting type selected. Once downloaded, simply untar the file into a directory on your web server.

   - For Windows, use a utility like WinZip.
   - For Linux, use the command `tar -xvf XpressConnect.tar.gz`.

   For assistance with installing, configuring or troubleshooting a web server, consult the [web server guide](#).

   To verify that the web server is properly configured, you may run the server verification at: http://[Server URL]/info/test.html

2. Where will the package be deployed? **http://wireless.utk.edu**

   [Download button]
Xpressconnect-the user side

Live Demo

comment:
The demo succeeded during Joint-Techs Summer 2011 in Fairbanks and took less than a minute to connect a laptop to eduroam.

For the purpose of archiving, we are including screenshots of the demo in the next few slides.
Welcome To The
University of Tennessee Knoxville Network

This wizard will configure your computer for secure network access.

To ensure your security, the University of Tennessee Knoxville network utilizes a secure authentication mechanism known as 802.1X. This security mechanism protects your user name and password. In a wireless environment, it also protects your data with network encryption. To utilize this secure network, your network connection requires specific settings. This wizard will ensure that your machine is properly configured.

XpressConnect will attempt to load automatically using a Java Applet or ActiveX. If you prefer, you may load XpressConnect manually.

To get started, simply accept the End-User License Agreement and click 'Start >'.

I accept the terms of the End User License Agreement
Start >
Welcome To The University of Tennessee Knoxville Network

Select the location for this network:

Your computer currently has multiple network locations defined. The current location is 'Automatic'. We may add the new network to this location or you may select a different location below. If uncertain, select 'Automatic'.

Location: Automatic

Refresh

< Back Continue >
Welcome To The University of Tennessee Knoxville Network

Enter your credentials for this network:

Your NetID@utk.edu and Your NetID password are required to access this network.

NetID@utk.edu: bogusguy@utk.edu
Your NetID pass...: 

Forgot your password? Click the link below or call the HelpDesk at 865-974-9900

< Back  Continue >
Welcome to the University of Tennessee Network.

Administrator Privileges Required:

To allow XpressConnect to configure your computer, please provide a user name and password for an administrative account on your system.

Unlicensed Evaluation Version
University of Tennessee Knoxville

Enter Your **Computer Admin Account & Password**, Not Your Network Password

Name: bogusguy
Password: ************
Welcome To The University of Tennessee Knoxville Network

Connecting to the Secure Network

Please wait while we associate to the wireless network 'eduroam'...
Welcome To The University of Tennessee Knoxville Network

Validating Connectivity

Waiting for a network address, which will allow the computer to participate on the network. This may take a couple moments.
Welcome to the University of Tennessee Knoxville Network

Successfully connected.

You are now connected to the network.
Your IP address is 172.20.227.163.
No modifications were made to your machine.
Security, Policies, Support
How to handle "abuse"?

- Direct handling: block the MAC address
- Worst case scenario: block the REALM (but this will block all users from that institution)
- Granular blocking: CUI (coming to you soon)
  - Can block a user alone instead of an entire REALM
  - Persistent identifier while visiting one institution
- Delayed handling: contact the federation, but it will require legal documents
- International Policies require to hold logs for at least 6 months
Policies

• each RO (Regional Operator) follows the compliance statement of the GeGC (Global eduroam Governance Committee).

• What is a user?
  ○ current staff/faculty/students
  ○ Alumni?
  ○ Retirees?

• Logs retention policy (6 months)
How to handle eduroam users' support?

● The eduroam rule: "user must first contact home institution"
  ○ the reality is: doesn't work well across time zones
  ○ So far every international debugging has taken longer than the user's visit
● If your HelpDesk can afford it, handle all eduroamers locally
● F-Ticks to the rescue of debugging international connectivity issues
RadSec

Next Generation RADIUS
The two RadSecs

Two things can be discussed under the name RadSec:

1: Classical RADIUS over TCP (as opposed to UDP)
   ● Solves MTU problems with EAP-TLS
   ● Solves unreliability of UDP
   ● Referred to as RADIUS+TCP(+TLS/SSL) for the duration
The two RadSecs (con't)

2: "Peer-to-Peer" RadSec (via NAI-based Dynamic Discovery)
- The responsible RADIUS server for a given realm is discovered via DNS (referred to as P2P-RadSec)
- Does everything the previous version does plus dynamic federation.
- Probably will be used by TLRSSs for routing with disparate TLDs that aren’t physically located in the nation that is authoritative for the TLD (e.g. .edu .net):
  - berkley.edu - Spain
  - monash.edu - Australia
  - ja.net - U.K.
  - nordu.net - Norway
  - etc..
RadSec DNS Resolution - overview

1. If RadSec server has an explicit route use it for Step 3
2. If no route is defined then
   1. Lookup NAPTR record for realm
   2. Lookup appropriate SRV records for realm
   3. Lookup appropriate AAAA or A records for realm
3. Connect and forward request

Given a realm, restena.lu, let's resolve its RadSec address via DNS (edited for brevity):

```
> dig restena.lu NAPTR
;; QUESTION SECTION:
;restena.lu. IN NAPTR

;; ANSWER SECTION:
restena.lu. 3112 IN NAPTR 150 10 "s" "x-eduroam:radius.tls" "" _radbackup._tcp.eduroam.lu.
restena.lu. 3112 IN NAPTR 100 10 "s" "x-eduroam:radius.tls" "" _radsec._tcp.eduroam.lu.
```

The "s" flag denotes a service; the **service tag** indicates this is for eduroam; and the **DNS name** specifies which server to resolve next. Note the _tcp for the next lookup.
RadSec DNS Resolution details (con't)

Since we found NAPTR records with the "s" flag previously, let's resolve the SRV record for the DNS names in the NAPTR record:

```
> dig _radsec._tcp.eduroam.lu SRV 
;; QUESTION SECTION:
;_radsec._tcp.eduroam.lu. IN SRV

;; ANSWER SECTION:
_radsec._tcp.eduroam.lu. 43200 IN SRV 0 0 2083
tld1.eduroam.lu.
```

The _tcp port and RadSec server to connect to are specified in the SRV record above.
Suppose there are no NAPTR records with the correct service tag, then prefix the realm with "_radiustls._tcp." and resolve the SRV record for the new realm as if the original NAPTR resolution had worked and provided the new realm.

If this fails then the institution is not RadSec NAI ready.
RadSec DNS Resolution details (con't)

Repeat these resolution steps for all "s" flags in the NAPTR record.

Order the various resulting as the were presented in the NAPTR record, preferring IPv6 over IPv4 addresses as possible.

Connect to the resolved servers, and once a connection is established, forward RADIUS requests over TCP.
RADIUS Servers supporting RadSec

Radiator:
- RadSec (RADIUS+TLS) available in <v4.7 via RADSECCClient and RADSECServer directives
- P2P RadSec supported in v4.8+ via DNSROAM and Resolver directives thanks to patches by Stefan Winter
RADIUS Servers supporting RadSec (con't)

radsecproxy:

radsecproxy is meant to serve as glue for non-RadSec compliant RADIUS infrastructure. RADIUS+TCP and P2P-RadSec are supported.
RADIUS Servers supporting RadSec (con't)

FreeRADIUS:

No support in the 2.x codebase but support is claimed in future releases.

All others (including FreeRADIUS): use radsecproxy as the "glue" to do RadSec without changing/breaking current architecture. That's what it's there for.
Configuring RadSec+TCP in Radiator

Overview: Replace `<Client>` blocks with `<ServerRADSEC>` and `<AuthBy RADIUS>` with `<AuthBy RADSEC>`.

To use TLS make sure `UseTLS` is specified otherwise you've configured RADIUS+TCP(-TLS)

RADIUS Secrets are still required since the protocol inside the TLS Tunnel is still classical RADIUS, and the secret is used in the same ways (to encrypt passwords and generate message authenticators).
ServerRADSEC example

<ServerRADSEC>
  Secret <Secret>

  UseTLS
  TLS_CertificateType PEM
  TLS_CAFile <CA Certificate(s) file>
  TLS_CertificateFile <Server Certificate path>
  TLS_PrivateKeyFile <Server Private Key path>
  TLS_RequireClientCert

  Identifier <Some Identifier>
</ServerRADSEC>
AuthBy RADSEC example

<AuthBy RADSEC>
  UseTLS
  TLS_CertificateType PEM
  TLS_CAFilename <CA Certificate(s) file>
  TLS_CertificateFile <Client Certificate path>
  TLS_PrivateKeyFile <Client Private Key path>
  TLS_ExpectedPeerName <CN to accept>
  ConnectOnDemand

<Host radsec-TLRS>
  UseTLS
  Secret <Secret>
</Host>
AuthMPPEKeys
</AuthBy>
Lessons Learned
Jeff Hagley's Lessons Learned

- Make sure to test your RADIUS infrastructure thoroughly and often
  - We make sure that not only are the RADIUS servers up, but the backend authentication is also working
  - Also make sure your peering with the eduroamUS servers is fully functional all the time
Jeff Hagley's Lessons Learned

- We found it best to have our users live on the eduroam SSID all the time
  - For us it makes sure that they are properly configured
  - Can have some security implications, but you can have RADIUS put certain users in certain VLANs to help separate guest and staff traffic
Jeff Hagley's Lessons Learned

- The first 2 times we deployed eduroam at a conference we used fat APs
  - This made it incredibly difficult to fix bugs

- An open SSID is easy to manage, and very little can go wrong. Once you start using 802.1x there are lots of things that might not work as expected when you plug the AP in, and changing a bunch of APs by hand is a pain
  - Authentication by the user was slow

- Every time a user roamed to a new AP they had to re-authenticate with their home RADIUS server
  - Use a controller based system at conferences
Dave Worth's Lessons and Best Practices

Lessons learned:
● In case of failing routing check names or IPs, ports, and most importantly, secrets.
● Know your supplicants in the case of debugging. Use eapol_test from the CLI for easy testing.
● Additional granularity in configs (for security) may required bigger hoops to jump through (hooks!)

Best Practices:
● Generate minimal working configurations first (applies to FreeRADIUS more than Radiator), then iterate until both features and flexibility meet your technical and/or business needs.
Glen Johnson's Lessons Learned

- Save time/frustration, consider EAP-TLS first
- Longer pilot/beta phase, when switching to TLS
  - Three weeks = too short
- TTLS
  - Lack of pw expired/locked/etc notification
    - Annoying
    - Repeated customer account lock-outs
      - Even more annoying; more so for them
      - Blackeye for you...
- Separate radius servers and SSIDs: test / prep / prod
  - At minimum: test / prod
Speaker and Contributor Contact Info

Philippe Hanset - University of Tennessee (eduroam-US)
  phanset@utk.edu

David Worth - University of Tennessee (eduroam-US)
  dworth@utk.edu

Glen Johnson - University of Alaska
  gfjohnson@alaska.edu

John Mitchell - University of Alaska
  jpmitchell@alaska.edu

Jeff Hagley - Internet2
  hagleyj@internet2.edu