

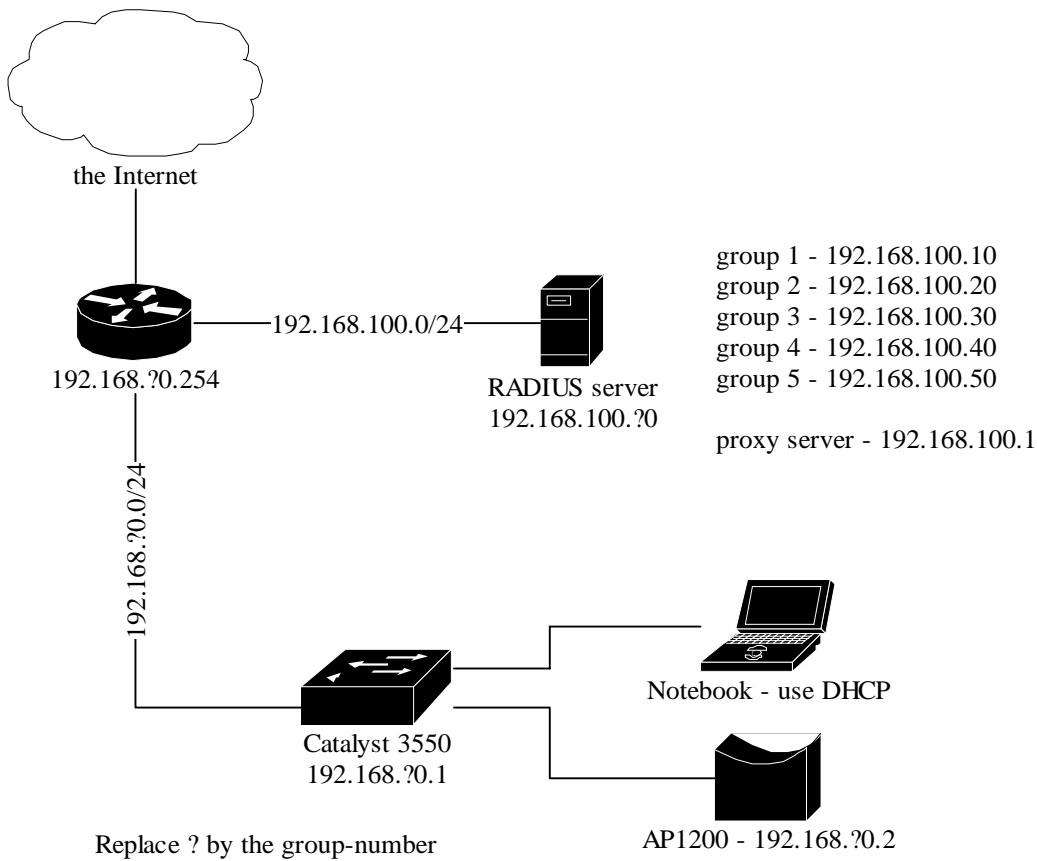
# 802.1x workshop, hands-on

## Introduction

The network for the hands-on session consists of identical set-ups for every group, each consisting of a Cisco Catalyst 3550 switch, a Cisco Aironet AP-1200 Access Point and a laptop with Windows XP. These group set-ups are coupled with a central switch. Every group has its own VLAN (with private IP-ranges). The central switch is connected to the Internet through NAT.

For the RADIUS configuration a FreeBSD server has been setup with so-called jails that allow each group to have its own 'virtual' Radiator RADIUS-server. On this server an initial configuration has been set up already.

For each group, the configuration looks like:



Have a look at Appendix A for the complete setup.

## Exercises

### Exercise 1: Get connected

In order to gain access to the Internet the configuration of the switch needs to be changed. First connect the notebook to the switch with the provided serial console cable. Create the VLAN for your group, and provide it with the correct IP-identity and gateway information. (This example is for group 1. If there is an enable password configured, try 'Cisco')

```
switch>enable
switch#configure terminal
switch(config)#vtp mode transparent
switch(config)#vlan 10
switch(config-vlan)#interface vlan 10
switch(config-if)#ip address 192.168.10.1 255.255.255.0
switch(config-if)#ip route 0.0.0.0 0.0.0.0 192.168.10.254
```

Configure all switch-ports for this new VLAN, except for the uplink to the central switch (in this case fastEthernet 0/48)

```
switch(config)#interface range fastethernet0/1 - 47
switch(config-if)#switchport mode access
switch(config-if)#switchport access vlan 10
switch(config-if)#spanning-tree portfast
```

The uplink can be defined with

```
switch(config)#interface fastethernet0/48
switch(config-if)#switchport trunk encapsulation dot1q
switch(config-if)#switchport trunk native vlan 10
switch(config-if)#switchport mode trunk
```

Confirm that you get an IP-address using DHCP on the notebook, both wired and wireless. (There is already a DHCP server configured for your network.)

The next step is to provide the Access-Point (AP) with an IP-address. Connect the serial console cable to the Access-Point and enter configuration mode again:

```
ap#configure terminal
ap(config)#interface bvi 1
ap(config-vlan)#ip address 192.168.10.2 255.255.255.0
```

Ensure that everything is correct with ping:

```
ap(config-vlan)# ^Z
ap#ping 192.168.100.1
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

Now that we have configured both devices with a static IP-address, we can move on with the configuration of RADIUS.

## Exercise 2: Setting up RADIUS

In exercise 2 the RADIUS server will be configured. Connect with putty (SSH-client, present at the desktop of the laptop) to the RADIUS-server of your group (this is 192.168.100.?, where ? is your group number) with username 'radius' and password 'radius'. You can get root-access with 'su' and the password 'radius'.

To set up RADIUS a (small) PKI is required. The required server certificates can be found in the home directory of the user 'radius'. Note that the server's private key and the public key are in the same file (server.pem or group?.pem).

Copy these files along with the dictionary file to the /etc/radiator directory. If it doesn't exist, create it first.

The Radiator RADIUS-server needs a configuration file /etc/radiator/radius.cfg. Then create this configuration file with your favourite editor, f.i.

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

### General variables

Radiator processes the configuration file linear. It is best to first set some general variables. Some practical ones can be found below:

```
#Foreground
LogStdout
```

"Foreground" prevents Radiator from running in the background after startup, "LogStdout" makes the logging appear on the screen. This combination is very handy for debugging purposes.

If everything works as desired these commands can be escaped by putting a '#' before the command. A '#' marks the start of a comment, meaning that the command will not be processed by Radiator.

```
Trace 4
```

Trace indicates the logging level. The higher the number, the more details in the logfiles (and the faster the server runs out of disk space!). For production systems this level needs in general to be low, for testing purposes 4 is a good choice. To allow the creation of the logfile the directory /var/log/radius needs to be created.

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

These 2 variables configure the location of the logfiles and the directory where a number of additional files like the RADIUS dictionary<sup>1</sup> can be found. These values can be used later in the configuration as %L and %D.

```
AuthPort 1812
AcctPort 1813
```

For RADIUS two pair of ports are being used, 1645 for authentication with 1646 for accounting and 1812 and 1813. Nowadays 1812 and 1813 are the preferred ports, Radiator however still uses 1645/1646 by default, above variables override this.

## Clients

A client in the `radius.cfg` is for instance a Network-Access-Server (NAS) like a switch or an access point or another RADIUS-server (in the role of client) that are allowed to send requests to the server. An example of such a client clause is:

```
<Client 192.168.10.2>
    Secret 6.6obaFkm&RNs666
    Identifier ACCESSPOINT1
</Client>
```

The *secret* is a shared secret that client and server use to encrypt and decrypt all messages between the two. It is also possible to configure a default client clause for use when there is no other client clause that matches. This is done by using the `<Client DEFAULT>` clause. Use of a DEFAULT client is not recommended as this means that every client uses the same secret. A Client-clause is always terminated with `</Client>`.

## Realms

The processing of authentication of accounting request is done by linear processing of the present `<Realm>`- or `<Handler>`-clauses. Handler-clauses are more potent than Realm clauses in terms of filtering things besides realms and are therefore the preferred method. A *realm* is the part behind a username to indicate the origin of a user, usually username and realm are separated with a "@". With `Klaas.Wierenga@SURFnet.nl` for instance `Klaas.Wierenga` is the username and `SURFnet.nl` the realm.

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

A typical example for catching all users with realm "group1" can be found below (command in bold):

---

<sup>1</sup> The dictionary is a file with RADIUS attributes with the corresponding values as described in the RADIUS RFC's. This dictionary can be found in the Radiator-3.9 directory and needs to be copied to `/etc/radiator`

```

<Handler Realm=group1>
# With RewriteUsername (may be present more than once) the realm can be
# stripped from the username allowing for user databases without the
# realm. If you need PEAP, do not strip the realm, and keep it in the
# user database!
# RewriteUsername s/^(^[^@]+).*/$1/
# The AuthBy-clause indicates the backend the Handler uses to verify
# the user credentials , for instance LDAP or SQL, here we use
# <AuthBy FILE> for authenticating against a plain-text file.
<AuthBy FILE>
# Filename indicates the user database, for both tunneled
# auth and regular PAP. The variabele %D comes in handy here.
Filename %D/users
# The EAPType indicates the EAP-types that can be used.
EAPType TTLS, PEAP, MSCHAP-V2
# The EAPTLS CAFile indicates the CA-certificate
# that signed the server certificate
# Use rootca.pem, and put it for instance in
# /etc/radiator, as below.
EAPTLS_CAFile %D/root.pem
# The server-certificate itself
EAPTLS_CertificateFile %D/server.pem
# The type of the certificate is PEM
EAPTLS_CertificateType PEM
# The file in which the private key of the server certificate can be
# found. This can be the same PEM-file as with the certificate
# itself in it.
EAPTLS_PrivateKeyFile %D/server.pem
# The password protecting the private-key-file
EAPTLS_PrivateKeyPassword serverkey
# The TLS FragmentSize is the size of the EAP-Messages that are
# being sent to the client: these messages should in any case fit in
# the RADIUS request, i.e. 4096 bytes.
EAPTLS_MaxFragmentSize 1024
# The AutoMPPE keys-optie ensures that in an Access-Accept an
# MS-MPPE-Send-Key and an MS-MPPE-Recv-Key are being sent.
# With these keys it is for instance in Windows XP possible to
# enable "automatic WEP-key"
AutoMPPEKeys
# When using PEAP the username for anonymous inner authentication
# must be defined with EAPAnonymous; make sure the proper realm is
# defined here.
EAPAnonymous anonymous@group1
# The AuthBy- and the Handler-clauses need to be terminated
</AuthBy>
</Handler>

```

With this Radiator configuration RADIUS requests can be processed, credentials are checked against the file /etc/radiator/users. The most straightforward syntax of this file is:

```
username@group1 Password=password
```

or, if you strip the realm:

```
username Password=password
```

Now radiator can be started with 'radiusd'. It will become a daemon, unless configured with "Foreground".

After adding users, you have to restart radiator (unless otherwise configured), with e.g.

```
kill -HUP `cat /var/log/radius/radiusd.pid`
```

If this configuration works ok, try splitting the inner- from the outer authentication with multiple handlers. (Filter this with TunneledByTTLS=1 and TunneledByPEAP=1.)

### Exercise 3: Configure AP and switch to use RADIUS

Now that the RADIUS server is configured, we can enable both the switch and the AP to use it.

On both the switch and the access-point the following commands needs to be entered in global configuration mode:

```
Switch-and-ap# configure terminal
Switch-and-ap(config)# aaa new-model
Switch-and-ap(config)# radius-server host 192.168.100.?0 auth-port 1812 key <secret>
```

On the Access-Point we also need to define authentication-groups, for both eap-authentication, and RADIUS-accounting. This can be done with:

```
Ap(config)# aaa group server radius rad_eap
Ap(config-sg-radius)# server 192.168.100.?0 auth-port 1812 acct-port 1813
Ap(config-sg-radius)# aaa group server radius rad_acct
Ap(config-sg-radius)# server 192.168.100.?0 auth-port 1812 acct-port 1813
Ap(config-sg-radius)# aaa authentication login eap_methods group rad_eap
Ap(config-sg-radius)# aaa accounting network acct_methods start-stop group rad_acct
```

### Exercise 5: Enable 802.1X on AP and switch

On the switch the following commands need to be executed in global and in interface configuration mode:

```
Switch# configure terminal
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# dot1x system-auth-control
Switch(config)# interface fastethernet0/12
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
```

The configuration of the AP is a bit more complicated, since we also need to configure some basic wireless settings.

First, create a new SSID on the base station in the interface settings of Dot11Radio 0, and enable eap authentication and encryption.

```
ap(config)# interface dot11Radio 0
ap(config-if)# encryption mode ciphers wep40
ap(config-if)# ssid group_1
ap(config-if-ssid)# authentication open eap eap_methods
ap(config-if-ssid)# accounting acct_methods
ap(config-if-ssid)# guest-mode
```

It's also wise to define a channel, otherwise the AP will search for an available channel, which can be changed in time. Choose f.i. the channel-name of your group.

```
ap(config-if-ssid)# channel 1
```

Also delete the default SSID (tsunami):

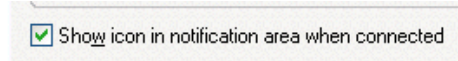
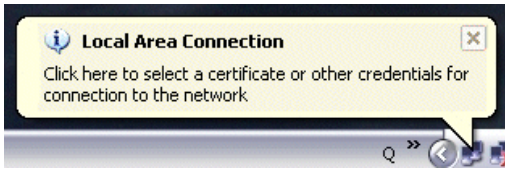
```
ap(config-if-ssid)# no ssid tsunami
```

If there are many groups, it's wise to lower the transmit-power of the AP:

```
ap(config-if)# power local 5
```

## Exercise 4: Configure the laptop for 802.1X

When using a wired connection under windows we must resolve an issue where the popup for credentials does not appear.

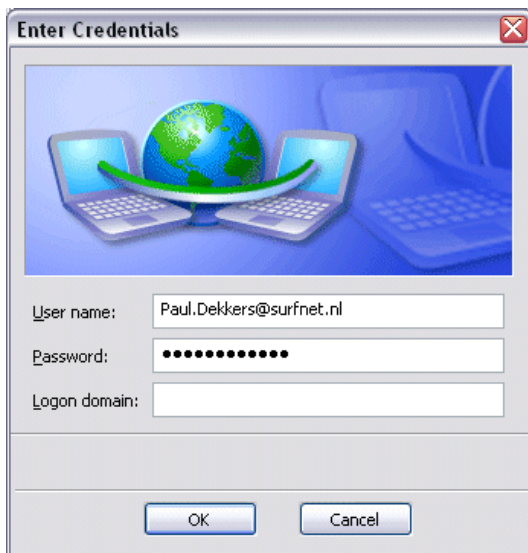


In order to see this popup, the icon for the wired connection must be visible in the system-tray. This can be configured using the properties of the LAN connection.

In this exercise we will configure PEAP, and the laptop will be made suitable for use with TTLS as well. Because XP doesn't have a builtin TTLS client one needs to be installed. For this purpose the SecureW2 installer can be downloaded from [www.alfa-ariss.com](http://www.alfa-ariss.com) or found at the laptop in the "shared documents" directory.

TTLS will be configured for use with the wireless interface to start with. This can be done by editing the properties of the wireless adapter (to be found in control panel/network connections) for the SSID that has been configured at the AP. Select in the configuration the authentication tab and select "SecureW2". Fill in the User-setup properties of SecureW2 the user credentials in as configured in Radiator.

Finally, do the same for the wired connection and try to get connected to the switchport that has dot1x enabled. If you succeed, do the same with Wireless LAN.



After succeeding with SecureW2, try to configure PEAP instead. In the properties screen of PEAP the inner authentication (MSCHAP-v2) can be configured.

Disable the use of the Windows credentials in the MS-CHAPv2 properties. This enables you to enter a username and password like in the image left.

With PEAP you can only do this once, if the authentication succeeds.

Have a look at the logfile of Radiator to see if everything works as expected.

If you need to remove the stored credentials entered here, you can do this by removing entries from your registry using regedit, out of:

```
HKEY_CURRENT_USER\Software\Microsoft\EAPOL\UserEapInfo
```

If authentication fails, try disabling the validation of server certificates. With PEAP, the certificates cannot be added as easily as with SecureW2. Also, check the client system's time and date.

## Exercise 6: Enable trunking on AP and switch

An extra VLAN (VLAN-60) will be created and added on the AP and the switch.

For the switch this means putting an extra port in trunk mode, the port that connects the AP. (We already configured the port that connects to the central switch as a trunk). For this the following commands are needed:

```
Switch# configure terminal
Switch(config)# interface FastEthernet 0/1
    (the port to which the Access-Point is connected)
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport trunk native vlan 10
Switch(config-if)# switchport mode trunk
Switch(config-if)# end
```

The native VLAN is the VLAN that will be sent over the wire without 802.1q tags. This VLAN is necessary for the maintenance interface of the Access-Point.

You also need to create an extra VLAN on your switch in order to work with it:

```
Switch(config)# vlan 60
Switch(config-vlan)# exit
```

On the access-point we have to create sub-interfaces for every VLAN. This is done in pairs for the both wired and wireless interface. Both get the same bridge-group assigned.

The first VLAN we create must be the native VLAN, on which we also have the management-interface of the Access-Point. In this example we use VLAN 10 as the Native VLAN.

```
Ap(config)# interface dot11Radio 0.10
Ap(config-if)# encapsulation dot1Q 10 native
Ap(config-if)# bridge-group 1
Ap(config)# interface fastEthernet 0.10
Ap(config-if)# encapsulation dot1Q 10 native
Ap(config-if)# bridge-group 1
```

And now the new VLAN (note the bridge-group is only 1 for the native interface):

```
Ap(config)# interface dot11Radio 0.60
Ap(config-if)# encapsulation dot1Q 60
Ap(config-if)# bridge-group 60
Ap(config)# interface fastEthernet 0.60
Ap(config-if)# encapsulation dot1Q 60
Ap(config-if)# bridge-group 60
```

For every VLAN we need to define how we deal with encryption:

```
Ap(config)# interface dot11Radio 0
Ap(config-if)# encryption vlan 10 mode ciphers wep40
Ap(config-if)# encryption vlan 60 mode ciphers wep40
Ap(config-if)# broadcast-key vlan 10 change 1800
Ap(config-if)# broadcast-key vlan 60 change 1800
```

## Exercise 7: Configure VLAN assignment

To assign a VLAN by the RADIUS server Radiator needs to be configured to send extra attributes in the "Access-Accept".

This can be configured per user (and gotten from the backend), but it is also possible to assign this for a complete realm, thus enabling traffic separation between groups of users (employees, students, guests etc.).

The extra needed attributes are:

```
Tunnel-Type=1:VLAN
Tunnel-Medium-Type=1:Ether_802
Tunnel-Private-Group-ID=1:<VLAN NUMBER>
```

In the users file a user can be assigned these attributes:

```
username      Password=password
              Tunnel-Type=1:VLAN
              Tunnel-Medium-Type=1:Ether_802
              Tunnel-Private-Group-ID=1:10
```

VLAN assignment can also be done in radius.cfg itself, by adding to the AuthBy-clause the following:

```
AddToReply Tunnel-Type=1:VLAN,Tunnel-Medium-Type=1:Ether_802,Tunnel-Private-Group-ID=1:10
```

To prevent for instance the use of parameters assigned by other servers all attributes can be stripped BEFORE the AddToReply clauses:

```
StripFromReply Tunnel-Type,Tunnel-Medium-Type,Tunnel-Private-Group-ID
```

Radiator needs to be restarted after changing the userfile or radius.cfg. This can be done by issuing the command:

```
kill -HUP `cat /var/log/radius/radiusd.pid`
```

## Exercise 8: Configure RADIUS proxying

To allow for guest usage RADIUS-requests for an unknown realm need to be proxied to a central RADIUS-server. This can be done by adding an <AuthBy RADIUS>-clause.

```
<Handler Realm=/.*/>
  <AuthBy RADIUS>
    Host 192.168.100.1
    Secret 802.1x-secret
    AuthPort 1812
    AcctPort 1813
  </AuthBy>
</Handler>
```

It is a good idea to add here an "AddToReply" and "StripFromReply", like with the own realm. This enables the possibility to assign guests their 'own' VLAN.

You can try this proxy-ing with one of the other groups and vice versa. The other group should define a client for your RADIUS-server.

## Exercise 9: Create a non-1X guest VLAN

Add a new VLAN with VLAN-ID 70. Create a new SSID at the AP, and select default-VLAN 70 for this SSID.

Don't enable EAP-authentication for this SSID and verify that the laptop can associate with this SSID.

By creating an SSID without 802.1X and assigning a fixed VLAN to this it is possible to give (limited) access for guests without authentication or to create a VLAN where instructions and software for 802.1X can be downloaded to circumvent the bootstrap problem of needing network access to get the software for network access.

On the switch you can provide users with a guest-VLAN. This is a fallback VLAN that will be connected if all attempts for authentication on the port fail.

This can be arranged with something like:

```
switch(config)# interface fastEthernet 0/15  
switch(config-if)# dot1x guest-vlan 70
```

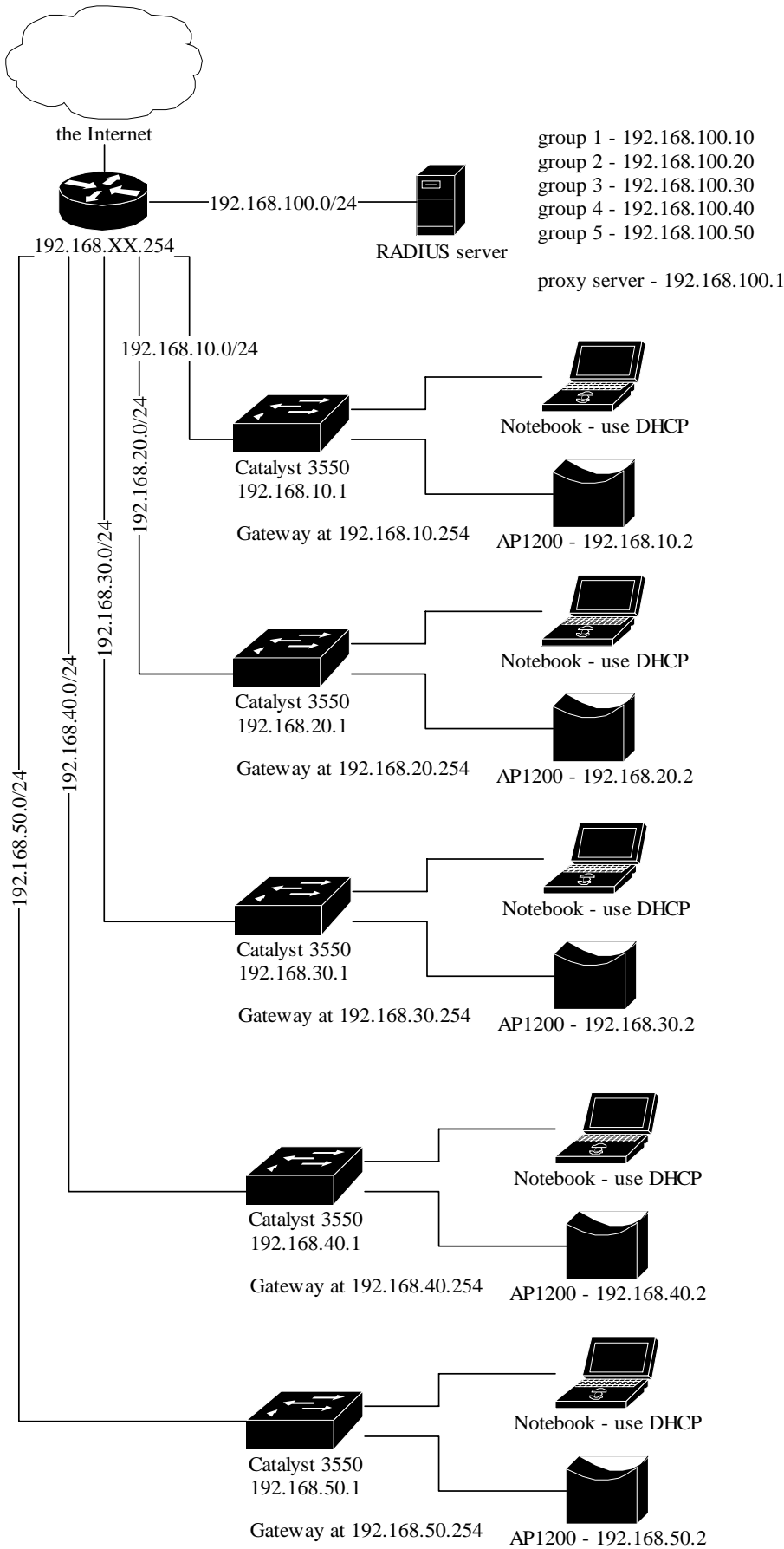
Revision 6, april 2004

This document was written by Klaas Wierenga and Paul Dekkers (SURFnet)

Thanks to Cisco Systems for providing equipment for the exercises!

# Appendix A

## Overview of the handson-network



## Appendix B

In this appendix you will find the configuration examples for Radiator, the Switch and the Access-Point. The examples are more complex than the ones used in the exercises.

### **Radiator**

Trace 4

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

AuthPort 1812
AcctPort 1813

<Client 192.168.10.2>
  Secret very_secret!
  Identifier AP-and-Switch
  IdenticalClients 192.168.10.1
</Client>

<Client 192.168.100.1>
  Secret super_secret!
  Identifier Proxy-Identifier
</Client>

<Handler TunnelledByPEAP=1, Realm=group1>
  <AuthBy FILE>
    Filename %D/peap-users
    EAPType MSCHAP-V2
  </AuthBy>
</Handler>

<Handler TunnelledByTTLS=1, Realm=group1>
  RewriteUsername s/^(^[^@]+).*/$1/
  <AuthBy FILE>
    Filename %D/ttts-users
  </AuthBy>
</Handler>

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

# Proxy only requests that do not come from the proxy-server, to prevent loops.
<Handler Client-Identifier=/^(?!Proxy-Identifier$)/>
  <AuthBy RADIUS>
    Host 192.168.100.1
    Secret super_secret!
    AuthPort 1812
    AcctPort 1813
    StripFromReply Tunnel-Type, Tunnel-Medium-Type, Tunnel-Private-Group-ID
    AddToReply Tunnel-Type=1:VLAN, Tunnel-Medium-Type=1:Ether_802,
      Tunnel-Private-Group-ID=1:70
  </AuthBy>
</Handler>
```

## Switch

Only relevant parts of the IOS-configuration are included.

```
!  
aaa new-model  
aaa authentication dot1x default group radius  
!  
vtp mode transparent  
!  
vlan 10,20,30,60,70  
!  
interface FastEthernet0/1  
description --- NORMAL SWITCHPORT IN VLAN 10 ---  
switchport access vlan 10  
switchport mode access  
no ip address  
spanning-tree portfast  
!  
interface FastEthernet0/12  
description --- SWITCHPORT WITH 802.1X: VLAN 10 WITH AUTH, VLAN 70 WITHOUT AUTH ---  
switchport access vlan 10  
switchport mode access  
no ip address  
dot1x port-control auto  
guest-vlan 70  
spanning-tree portfast  
!  
interface FastEthernet0/48  
description --- SWITCHPORT THAT IS THE TRUNK PORT WITH ALL VLANS TO CENTRAL SWITCH ---  
switchport trunk encapsulation dot1q  
switchport trunk native vlan 10  
switchport mode trunk  
no ip address  
!  
interface Vlan10  
ip address 192.168.10.1 255.255.255.0  
!  
ip routing  
ip route 0.0.0.0 0.0.0.0 192.168.10.254  
!  
radius-server host 192.168.100.10 auth-port 1812 acct-port 1813 key very_secret!  
!
```

## Access-Point

Only relevant parts of the IOS configuration are included.  
This example enables the assignment of both VLAN 10 and VLAN 70 by RADIUS.  
Default is VLAN 10, in case no attributes are provided by RADIUS.

```
!  
aaa new-model  
!  
aaa group server radius rad_eap  
  server 192.168.100.10 auth-port 1812 acct-port 1813  
!  
aaa group server radius rad_acct  
  server 192.168.100.10 auth-port 1812 acct-port 1813  
!  
aaa authentication login eap_methods group rad_eap  
!  
interface Dot11Radio 0.10  
!  
  encapsulation dot1Q 10 native  
  bridge-group 1  
!  
interface fastEthernet 0.10  
!  
  encapsulation dot1Q 10 native  
  bridge-group 1  
!  
interface Dot11Radio 0.70  
!  
  encapsulation dot1Q 70  
  bridge-group 70  
!  
interface fastEthernet 0.70  
!  
  encapsulation dot1Q 70  
  bridge-group 70  
!  
interface Dot11Radio0  
!  
  encryption vlan 10 mode ciphers wep40  
  encryption vlan 70 mode ciphers wep40  
  broadcast-key vlan 10 change 1800  
  broadcast-key vlan 70 change 1800  
!  
  dot1x reauth-period 1800  
!  
  ssid group_1  
    authentication open eap eap_methods  
    accounting acct_methods  
    guest-mode  
    vlan 10  
!  
  ! ssid wide_open  
  !   vlan 70  
!  
radius-server host 192.168.100.10 auth-port 1812 acct-port 1813 key very_secret!
```