Zentyal

for Network Administrators

VERSION 2.2

zentyal

Linux Small Business Server

Preparation for the certification exam

Zentyal Certified Associate (ZeCA)

STUDENT GUIDE

http://www.zentyal.com/training/

zentyal TRAINING

VERSION 2.2

Linux Small Business Server

Credits

Produced by:

eBox Technologies S.L.
CEEI Aragon, Nave 19
C/ Maria de Luna 11
50018 - Zaragoza (Spain)
www.zentyal.com

Copyright notice

Index

5. **ACTION.** Finally verify from *Dashboard* that the address appearing in the *DHCP leases widget* is displayed[17].

2.3.3 Proposed exercises

☐ EXERCISE A

Configure DHCP service to always assign the same IP address to a host. Check that the host is working properly.

☐ EXERCISE B

Change the maximum lease time. Change the parameters of **dhclient** in your configuration file to send a request with a lease greater than allowed. What happens? Check the lease contents in */var/lib/dhcp3/dhclient.leases*.

☐ EXERCISE C

Integrate DHCP and DNS modules to serve names of DHCP clients. Provide examples using static and dynamic domains by using static ranges and assignments.

2.4 Certification Authority (CA)

2.4.1 Public Key Infrastructure (PKI)

Encryption technologies ensure authenticity, privacy and integrity in data transmission. However, the main problem in all the encryption schemes is how to distribute the keys to their users without interception by third parties. The solution consists of the use of a *Public Key Infrastructure*[18] (PKI). This technology allows the use of the key in insecure medium while avoiding forgery, interception or modification of keys by anyone who snoops the communication.

PKI means each participant generates two keys: a *public key* and a *private key*. The public key can be distributed publicly and the private one must remain secret. Any participant who wants to encrypt a message can do it with his own private key and the recipient's public key. Therefore, the message can only be decrypted with the recipient's private key. Moreover, as the message has been encrypted with the sender's private key, it is possible to guarantee its integrity by checking the sender's public key.

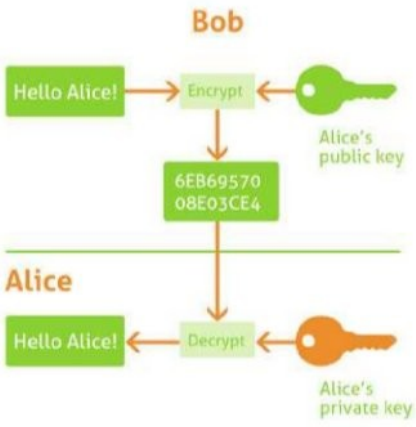

Image 2.30. Public key encryption.

17 Bear in mind that the static allocations do not appear in the DHCP widget.
18 *http://en.wikipedia.org/wiki/Public_key_infrastructure.*

Image 2.31. Public key signature.

However, this solution creates a new problem: if anyone could present a public key, how can you guarantee that a participant is really who he claims to be and is not impersonating another identity? To solve this problem, **certificates** were created[19].

A certificate is a file that contains a public key, signed for someone that is trusted. This trusted participant is used to verify identities and is called **Certification Authority** (CA)[20].

Image 2.32. Diagram to issue a certificate.

Zentyal uses **OpenSSL**[21] for the management of the Certification Authority and the life cycle of the issued certificates.

2.4.2 Importing certificates in clients

To validate any certificate issued by a *Certification Authority* operated by Zentyal, you should import the certificate into the system.

19 *http://en.wikipedia.org/wiki/Public_key_certificate*
20 *http://en.wikipedia.org/wiki/Certification_authority*
21 *www.openssl.org/.*

In Windows XP you should click on *Start* → *Configuration* → *Control Panel*, and once in this window select *Network and Internet connections*.

Image 2.33. Control panel.

Next, select *Internet options*.

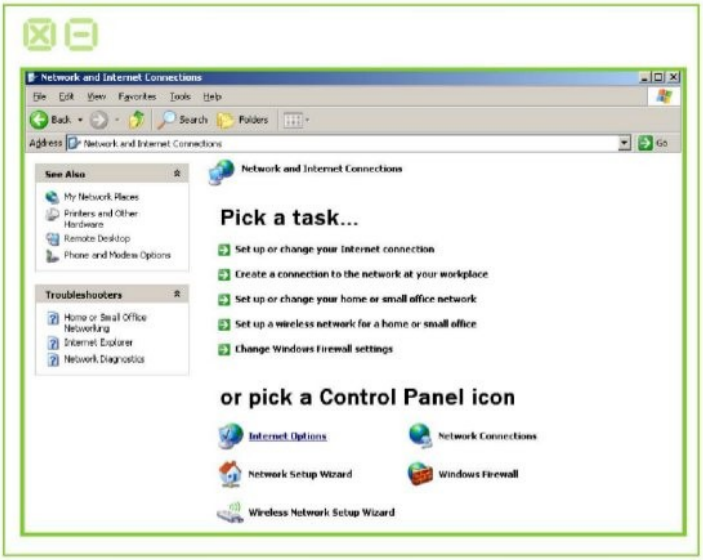

Image 2.34. Network and Internet connections.

You will access a new *Internet properties* window where you should select the *Content:* tab and go to *Certificates...*

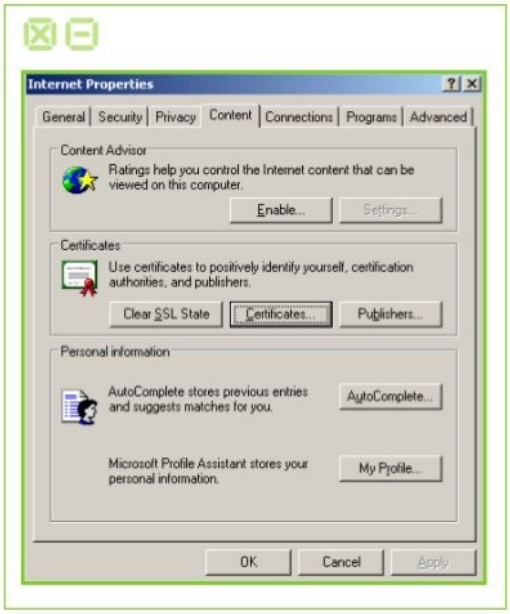

Image 2.35. Internet properties.

In the *Certificates* window you can see different tabs classifying the different types of stored certificates. To import your certificate, just click on *Import...*

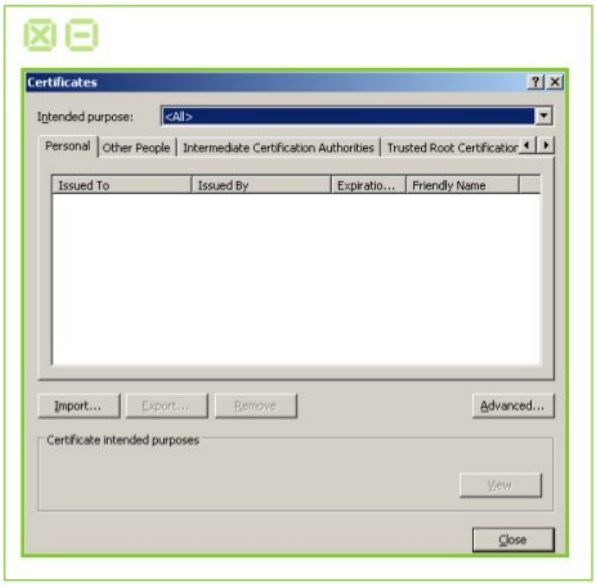

Image 2.36. Certificates.

A wizard will start to import new certificates. Just continue with the *Next* step.

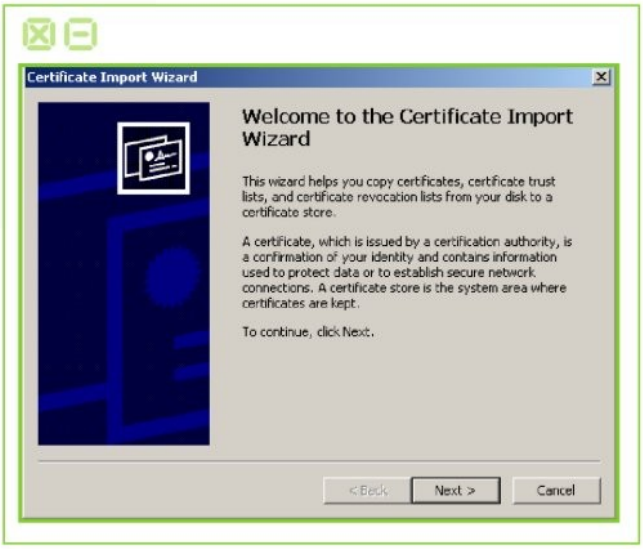

Image 2.37. Certificate import wizard.

In *File name* select the path to the file with your certificate by using *Examine...*. Once you have selected the certificate to import just continue to the next step.

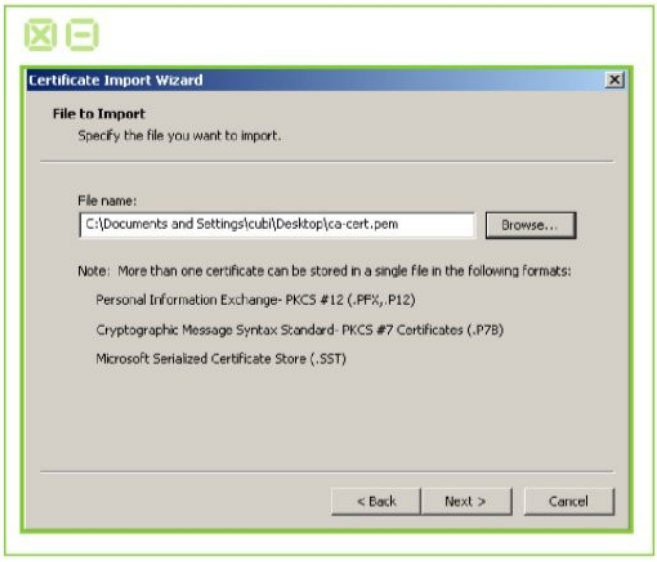

Image 2.38. Certificate import wizard.

On the *Certificate store* window select the *Automatically select the certificate store based on the type of certificate* option and continue again to the next step.

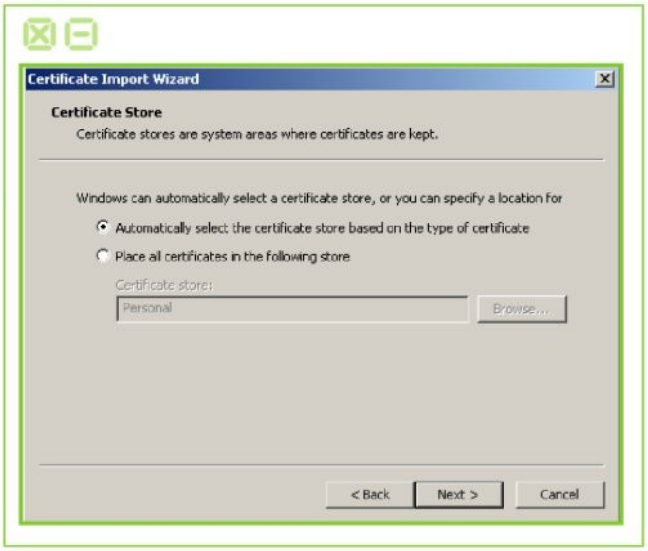

Image 2.39. Certificate import wizard.

You will see a summary of the actions to be executed and you just have to click on *Finish*.

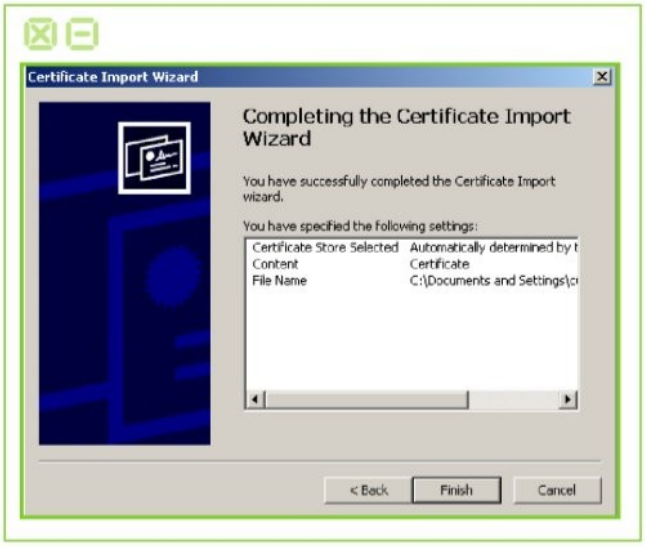

Image 2.40. Certificate import wizard.

If the import was succesful, you will see a confirmation message.

Image 2.41. Certificate import wizard.

Verify that the certificate from your CA appears in the list of certificates.

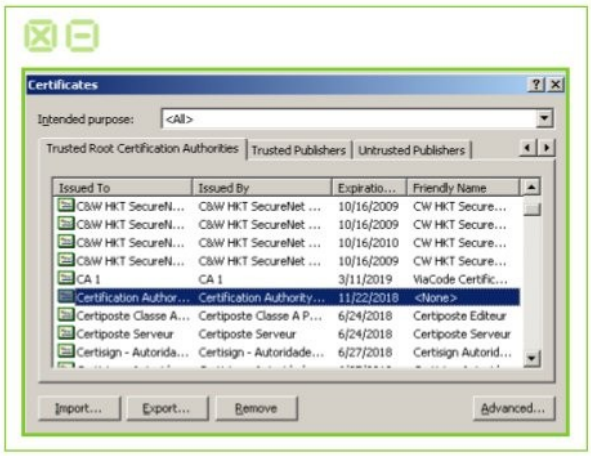

Image 2.42. Certificates.

You can also add a certificate from the CA to a web browser such as Mozilla Firefox. Let's see how to do this on Ubuntu.

The first step is to launch the browser and go to *Edit → Preferences*. In this window select the *Advanced* tab, then the *Encryption* tab and click on *View certificates*.

Image 2.43. Advanced preferences.

Much like in the previous case, a list of different types of certificates classified in different tabs will be displayed. Select the *Authorities* tab.

Image 2.44. Certificates from Authorities.

Proceed to *Import* the CA certificate by selecting the file where it is stored. The following window will be displayed to select in which situations you want to trust this new **Certification Authority**.

Image 2.45. Import new certificate.

By clicking *View* you will see the details of the certificate.

Image 2.46. Details of the certificate.

Once you have verified that the certificate is correct and you have selected the situations in which you are going to trust this CA, you just have to click on *Accept* and verify that the certificate appears on the list.

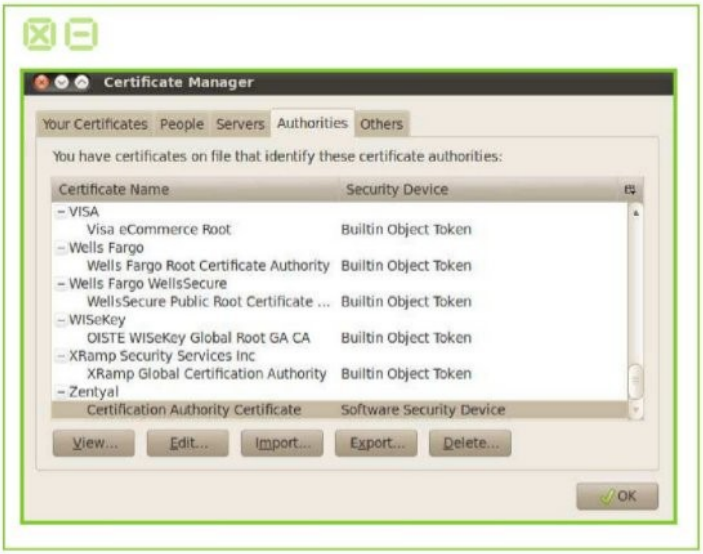

Image 2.47. Certificates.

2.4.3 Certification Authority configuration with Zentyal

In Zentyal, the *Certification Authority* module is self-managed, which means that it does not need to be enabled in *Module status*. However, you have to initialize the CA to make the functionality of the module available.

Go to *Certification Authority* → *General* and you will find the form to create the CA. You are required to fill in the *Organization Name* and *Days to expire* fields. Optionally, it is possible

to specify the *Country code* (a two-letter acronym following the ISO-3166-1 standard[22]), *City* and *State*.

Image 2.48. Create the CA certificate.

When setting the expiration date you have to take into account that at the moment of expiration all certificates issued by this CA will be revoked, stopping all services depending on those certificates.

Once the CA has been initialised, you will be able to issue certificates. The required data are the *Common Name* of the certificate and the *Days to expire*. This last field is limited by the fact that no certificate can be valid for a longer time than the CA. In case you are using the certificate for a service such as a web server or mail server, the *Common Name* of the certificate should match the domain name of that server. For example, if you are using the domain name *zentyal.home.lan* to access the web administrative interface in Zentyal, you will need a certificate with the same *Common Name*. In case you are setting a user certificate, the *Common Name* will usually be the user's email address.

Optionally, you could set *Subject Alternative Names*[23] for the certificate. These are useful when setting common names to a certificate: a domain name or an IP address for a HTTP virtual host or an email address when signing email messages.
Once the certificate is issued, it will appear in the list of certificates and it will be available for the administrator and for the rest of modules. Through the certificate list you can perform several actions on the certificates:

- Download the public key, private key and the certificate.
- Renew the certificate.
- Revoke the certificate.
- Reissue a previously revoked or expired certificate.

22 *http://en.wikipedia.org/wiki/ISO_3166-1*
23 For more information about subject alternative names, visit *http://www.openssl.org/docs/apps/x509v3_config.html#Subject_Alternative_Name*

Image 2.49. Certificate list page.

The package with the keys contains also a PKCS12 file with the private key and the certificate and it can be installed directly into other programs such as web browsers, mail clients, etc.

If you renew a certificate, the current certificate will be revoked and a new one with the new expiration date will be issued. And if you renew the CA, all certificates will be renewed with the new CA trying to keep the old expiration date. If this is not possible because it is after the date of expiry of the CA, then the date of expiration is set as the one of the CA.

Image 2.50. Renew a certificate.

If you revoke a certificate you will not be able to use it anymore as this action is permanent and it cannot be undone. Optionally, you can select the reason of the certificate revocation:

- **unspecified:** reason non specified,
- **keyCompromise:** the private key has been compromised,
- **CACompromise:** the private key for the certification authority has been compromised,
- **affilliationChanged:** the issued certificate has changed its affiliation to another certification authority from other organization,
- **superseded:** the certificate has been renewed and it is now replaced by a new one,
- **cessationOfOperation:** the certification authority has ceased its operations,
- **certificateHold:** certified suspended,
- **removeFromCRL:** currently unimplemented, it provides delta CRLs support, that is, lists of certificates whose revoked status has changed.

Image 2.51. Revoke a certificate.

When a certificate expires all the modules are notified. The expiration date of each certificate is automatically checked once a day and every time you access the certificate list page.

Services Certificates

On *Certification Authority* → *Services Certificates* you can find the list of Zentyal modules using certificates for their operation. Each module generates its own self-signed certificates, but you can replace them with others issued by your CA.

You can generate a certificate for each service by defining its *Common Name*. If a previous certificate with the name does not exist, the CA will create it automatically.

Image 2.52. Services Certificates.

Once enabled, you need to restart the service to force the module to use the new certificate. This also applies if you renew a certificate for a module.

TIP. If you want to use certificates signed by a commercial Certification Authority, you have to follow the usual procedure to generate a private key and then a CSR (Certificate Signing Request) so they will forward the signed certificate that you will use in the service. Let's see an example on how to do this for your mail server:

1.- Generate the private key

```
openssl genrsa -out dominio.tld.key 2048
```

2.- Using the private key, generate the CSR:

```
openssl req -new -key dominio.tld.key -out dominio.tld.csr
```

3.- Send the CSR file to the Certification Authority, which will return a certificate that you will save in a file named domain.tld.crt.

4.- Check where to store the certificate in the configuration files of the desired service; in the case of Postfix, this directory is */etc/postfix/sasl/postfix.pem*, so proceed to overwrite that file setting exclusive read permissions for the root user:

```
cat dominio.tld.crt dominio.tld.key > /etc/postfix/sasl/postfix.pem
chmod 400 /etc/postfix/sasl/postfix.pem
```

5.- You can proceed in the same way for Zarafa:

```
cat dominio.tld.crt dominio.tld.key > /etc/zarafa/ssl/ssl.pem
chmod 400 /etc/zarafa/ssl/ssl.pem
```

6.- It is a good practice to save a backup of the private key and the certificate, and also double-check that all files containing the private key can only be read by the root user and/or the username used by the service.

2.4.4 Practical examples

PRACTICAL EXAMPLE A

The company ContaFoo SL is implementing security protocols in internal communications to comply with current legislation. The various intranets will work under HTTPS and email will use SSL/TLS, but for that they need to import the certificate of the **Certification Authority** operated in their Zentyal server. Create the CA and then import its certificate in the Windows XP clients.

1. **ACTION.** Go to *Certification Authority* → *General*. In the form called *Create Certification Authority Certificate*, fill in the fields *Organization Name* and *Days to expire* with reasonable values. Press *Create* to generate the Certification Authority.

   EFFECT. The key pair of the Certification Authority is generated and its certificate will be issued. Your new CA will be displayed in the certificate list. The form for creating the CA will be replaced by another one intended to issue normal certificates.

2. **ACTION.** From the certificate list, you will download the one of the CA, a file with a name such as *CA-key-and-cert.tar.gz* containing the public *key ca-public-key.pem* and the certificate *ca-cert.pem*. Following the previously described procedure, import the certificate file *ca-cert.pem* on the Windows XP clients.

   EFFECT. The new certificates will appear in the certificate list, and any certificate issued by this CA will be accepted by the Windows XP clients.