Seguridad en Redes
Practica 2.3

David Antuña Rodríguez
Javier Carrión García

1 OpenSSL

1.1 Creación de una CA

Si intentamos utilizar otros directorios dará error a no ser que se modifique el fichero de configuración, concretamente los datos de la sección [CA_default].
La clave privada se ha almacenado en *demoCA/private/cakey.pem* y el certificado en *demoCA/cacert.pem*. Como hemos empleado la opción -new la clave generada es de tipo RSA y su tamaño viene especificado en el fichero de configuracion, valor de default_bits en la sección [req], por defecto son 2048 bits.
Comandos
mkdir demoCA
mkdir demoCA/newcerts
mkdir demoCA/private
touch demoCA/index.txt
echo 01 > demoCA/serial
echo 01 > demoCA/crlnumber
openssl req -x509 -new -days 3650 -keyout demoCA/private/cakey.pem -out demoCA/cacert.pem

1.2 Creación de solicitudes de firma de certificado

El algoritmo por defecto se encuentra en la variable default_md del fichero de configuración, en nuestro caso hemos empleado la opción -new que crea una clave rsa nueva y la ha almacenado en userkey1.pem y userkey2.pem, también podriamos haber utilizado una que tuvieramos previamente con la opción -key.
Comandos
openssl req -new -keyout userkey1.pem -out usercsr1.pem CN: usuario1
openssl req -in usercsr1.pem -noout -text > usercsr1.txt
openssl req -verify -in usercsr1.pem
openssl req -new -keyout userkey2.pem -out usercsr2.pem CN: usuario2
openssl req -in usercsr2.pem -noout -text > usercsr2.txt
openssl req -verify -in usercsr2.pem

1.3 Creación y verificación de certificados

Comandos
openssl ca -in usercsr1.pem -out usercert1.pem
openssl verify -CAfile demoCA/cacert.pem usercert1.pem
openssl ca -in usercsr2.pem -out usercert2.pem
openssl verify -CAfile demoCA/cacert.pem usercert2.pem

1.4 Consulta y manipulación de certificados

Comandos

openssl x509 -in usercert1.pem -noout -text > usercert1.txt
openssl x509 -in usercert1.pem -noout -pubkey > usercert1pubkey.txt
openssl x509 -in usercert2.pem -noout -text > usercert2.txt
openssl x509 -in usercert2.pem -noout -pubkey > usercert2pubkey.txt openssl x509 -in demoCA/cacert.pem -noout -text > cacert.txt
openssl x509 -in demoCA/cacert.pem -noout -pubkey > cacertpubkey.txt
PEM
openssl x509 -in usercert1.pem -out usercert1.der -outform DER
openssl x509 -in usercert1.der -inform DER -out usercert1.pem
openssl pkcs12 -export -in usercert1.pem -inkey userkey1.pem -out usercert1.p12

1.5 Revocación de certificados

Comandos
openssl ca -revoke usercert2.pem
openssl ca -gencrl -out crl.pem
openssl crl -in crl.pem -noout -text > crl.txt
openssl crl -CAfile demoCA /cacert.pem -in crl.pem
openssl verify -crl_check -CAfile demoCA/cacert.pem -CRLfile crl.pem usercert2.pem

2 GnuPG

2.1 Firma de claves (Web of trust)

C será valida porque aunque desconocemos la validez de C si conocemos la de B, y nos fiamos de ella, y la clave C ha sido firmada por B.
Comandos
gpg2 −−gen-key nombre: claveA, psw: seguridad
gpg2 −−import pubkeysSER.gpg
gpg2 −−sign-key −−local-user idA idB
gpg2 −−sign-key −−local-user idB idC
gpg2 −−edit-key idB trust quit Nivel 4
gpg2 −−edit-key idC trust quit Nivel 1
gpg2 −−check-trustdb > check_trustdb.txt
gpg2 −−list-options show-uid-validity –list-keys > validity.txt

En el segundo caso C no será válida porque B es dudosa y para validar una clave con firmas dudosas son necesarias al menos 3, también debe cumplir que el camino que lleva hasta nuestra firma es menor de 5, como hemos firmado B con la clave A esto se cumple pero estas condiciones no son excluyentes.
Comandos
gpg2 −−edit-key idB trust quit Nivel 3
gpg2 −−check-trustdb > check_trustdb2.txt
gpg2 −−list-options show-uid-validity –list-keys > validity2.txt

Ahora C vuelve a ser válida porque como hemos explicado antes ha sido firmada por 3 claves dudosas (B, D y E) y ademas el camino a nuestra clave es menor de 5.

Comandos

gpg2 −−sign-key −−local-user idA idD
gpg2 −−sign-key −−local-user idA idE
gpg2 −−edit-key idD trust quit Nivel 3
gpg2 −−edit-key idE trust quit Nivel 3
gpg2 −−sign-key −−local-user idD idC
gpg2 −−sign-key −−local-user idE idC
gpg2 −−check-trustdb > check_trustdb3.txt
gpg2 −−list-options show-uid-validity –list-keys > validity3.txt