

Instalación e configuración do DNle en Debian

Introdución

Queremos empregar o noso DNle baixo Debian para poder autenticarnos pola rede ou firmar documentos.

Situación de partida:

- Debian testing jessie con kernel 3.2.0-4-amd64
- Lector de tarxetas C3PO LTC31v2



Comprobamos o funcionamento do lector

Enchufamos o lector de tarxetas

Examinando se detecta o lector de tarxetas

```
root@Port-Profes4:/home/usuario# lsusb -t
/: Bus 08.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 07.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 06.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
   |__ Port 2: Dev 2, If 0, Class=Chip/SmartCard, Driver=, 12M
```

Amósamos información detallada sobre o dispositivo e comprobamos que non ten un driver asociado.

```
root@Profes1:/home/usuario# lsusb -s 6:2 -v
Bus 006 Device 002: ID 0783:0006 C3PO LTC31v2
Device Descriptor:
  .....
  idVendor           0x0783 C3PO
  idProduct          0x0006 LTC31v2
  bcdDevice          0.38
  iManufacturer      1 C3PO
  iProduct            2 USB SMART CARD READER
```

Instalación das ferramentas para traballar co lector de tarxetas

Necesitamos ter instaladas as seguintes ferramentas:

- **pcscd**: (*Personal Computer/Smart Card*) É un **Middleware**, é dicir, proporciona unha API que permite a outras ferramentas interactuar cos lectores de tarxetas intelixentes.
- **pcsc-tools**: Ferramentas para interactuar cos lectores de tarxetas intelixentes
- **pinentry-gtk2**: Para que nos pida o pin do DNIE en modo gráfico.

```
apt-get install pcscd pcsc-tools pinentry-gtk2
```

Unha vez instalado pcscd o sistema xa recoñece o lector de tarxetas e lle asigna un módulo para xestionalo.

```
root@Port-Profes4:/home/usuario# lsusb -t
/: Bus 08.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 07.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 06.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
    |__ Port 2: Dev 2, If 0, Class=Chip/SmartCard, Driver=usbfs, 12M
```

Comprobamos o que funciona correctamente

Empregamos unha das ferramentas proporcionadas por pcsc-tools para comunicarnos co lector de tarxetas intelixentes e obter información da tarxeta intelixente inserida.

```
root@Port-Profes4:/home/usuario/DNIE# pcsc_scan
PC/SC device scanner
V 1.4.21 (c) 2001-2011, Ludovic Rousseau <ludovic.rousseau@free.fr>
Compiled with PC/SC lite version: 1.8.7
Using reader plug'n play mechanism
Scanning present readers...
0: C3PO LTC31 v2 (00416036) 00 00
Fri May 31 14:57:44 2013
Reader 0: C3PO LTC31 v2 (00416036) 00 00
    Card state: Card inserted,
    ATR: 3B 7F 38 00 00 00 6A 44 4E 49 65 10 02 4C 34 01 13 03 90 00
.....
.....
Possibly identified card (using /usr/share/pcsc/smartcard_list.txt):
3B 7F 38 00 00 00 6A 44 4E 49 65 10 02 4C 34 01 13 03 90 00
3B 7F 38 00 00 00 6A 44 4E 49 65 [1,2]0 02 4C 34 01 13 03 90 00
    DNI electrónico (Spanish electronic ID card)
    http://www.dnielectronico.es
```

Podemos probar cunha tarxeta de crédito, pero seguramente non a recoñecerá, xa que non estará na lista de tarxetas coñecidas.

Instalación do “driver” específico para o DNle

O lector de smartcard funciona correctamente, permite ler a información da tarxeta. Pero para que os programas entendan a información almacenada no DNle é preciso instalar **OpenSC-OpenDNle**.

É dicir, OpenSC con soporte para el DNI electrónico Español. Este é un driver de código aberto para traballar cos lectores de DNle. É a fusión de dous proxectos:

- **OpenSC**: é o standard de manexo de smartcards ou tarxetas criptográficas pkcs#15.
- **OpenDNle**: Incorpora o necesario para comunicarse co DNle Español.

O paquete proporcionado na [páxina oficial](#) está discontinuado así que empregaremos este. A última versión desenrolada foi para **Debian Squeeze**.

Obteremos a versión actualizada do driver dende <https://github.com/clopez/OpenSC-OpenDNle-Debian>

Git é un dos [sistemas de control de versións](#) máis populares entre os desenroladores. Parte culpa da súa popularidade tena [GitHub](#), un excelente servizo de aloxamento de repositorios de software con este sistema

Como o código do noso driver está aloxado en GitHub podemos descargar o código e compilalo para a nosa versión de Debian para crearnos o .deb. Para elo seguimos as instrucións da páxina

- **Instalamos dependencias**

```
apt-get install build-essential devscripts dh-autoreconf git-buildpackage
apt-get build-dep opensc
```

Este último comando instala as dependencias necesarias para compilar o paquete opensc, para elo temos que ter engadida no sources.lst unha liña do tipo deb-src. Por exemplo

```
deb-src http://ftp.es.debian.org/debian/ jessie main contrib non-free
```

- **Clonar repositorio con gbp:**

É dicir descargamos o código fonte do repositorio github para compilalo.

Como usuario normal

```
mkdir DNIE
cd DNIE
gbp-clone https://github.com/clopez/OpenSC-OpenDNle-Debian
Esto crea unha carpeta chamada OpenSC-OpenDNle-Debian co código fonte
cd OpenSC-OpenDNle-Debian
```

- **Compilamos o paquete**

```
git-buildpackage
```

Se se produce algún erro e non crea o .deb podemos executalo como

```
git-buildpackage -git-ignore-new
```

O proceso finaliza coa seguinte mensaxe de erro. Pero é normal

Now signing changes and any dsc files...

```
signfile opensc_0.12.2-OpenDNIE.dsc Carlos Alberto Lopez Perez  
<clopez@igalia.com>
```

```
gpg: directory `/home/usuario/.gnupg' created
```

```
gpg: creouse un novo ficheiro de configuración
```

```
`/home/usuario/.gnupg/gpg.conf'
```

```
gpg: AVISO: as opcións de `/home/usuario/.gnupg/gpg.conf' aínda non están  
activas nesta execución
```

```
gpg: chaveiro `/home/usuario/.gnupg/secring.gpg' creado
```

```
gpg: chaveiro `/home/usuario/.gnupg/pubring.gpg' creado
```

```
gpg: skipped "Carlos Alberto Lopez Perez <clopez@igalia.com>": a chave  
secreta non está dispoñible
```

```
gpg: /tmp/debsign.ZUVJifRs/opensc_0.12.2-OpenDNIE.dsc: clearsign failed: a  
chave secreta non está dispoñible
```

```
debsign: gpg error occurred! Aborting....
```

```
debuild: fatal error at line 1278:
```

```
running debsign failed
```

```
gbp:error: Couldn't run 'debuild -i -I': debuild -i -I returned 29
```

É normal que falle, está tratando de firmar o paquete empregando gpg, pero é imposible que o firmemos nos, xa que non somos o autor do mesmo e non temos a súa clave privada.

Se todo vai ben obteremos no directorio DNIE o paquete .deb para instalar

- **Instalamos o paquete**

Como root

```
dpkg -i opensc_0.12.2-OpenDNIE_amd64.deb
```

.

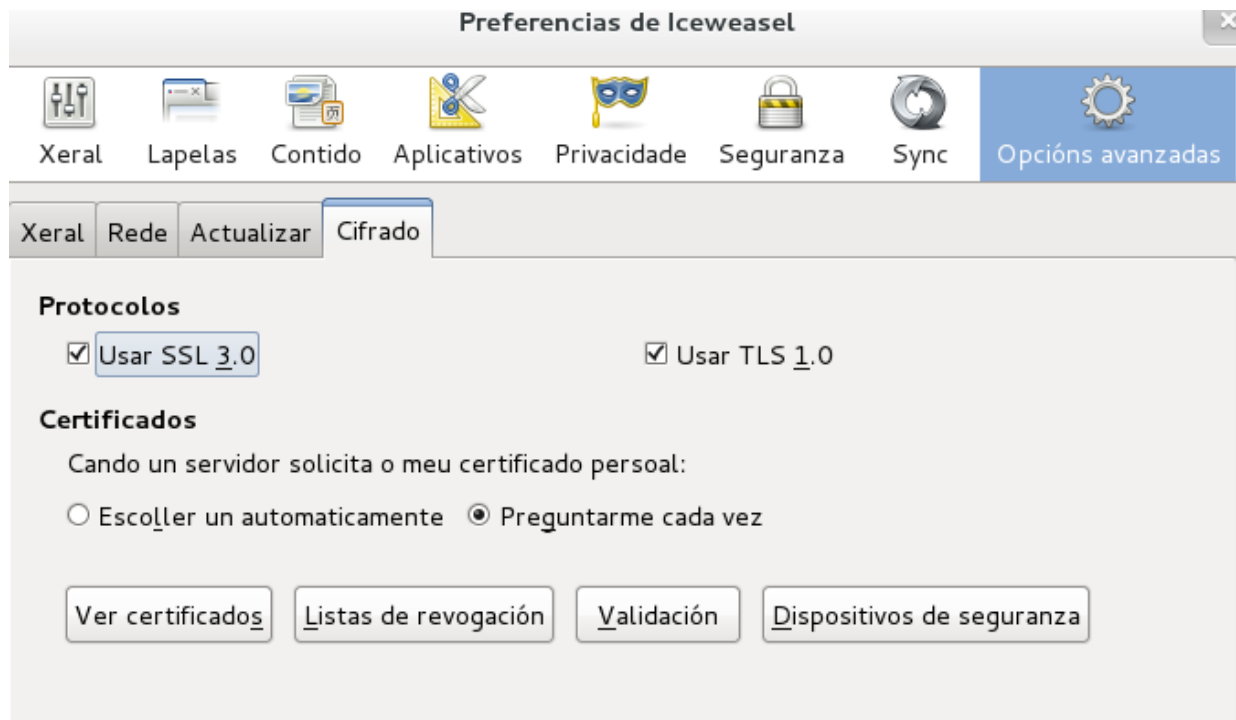
Configurar o navegador para autenticarnos co DNIE

Imos a configurar o iceweasel. Temos que facer dúas cousas

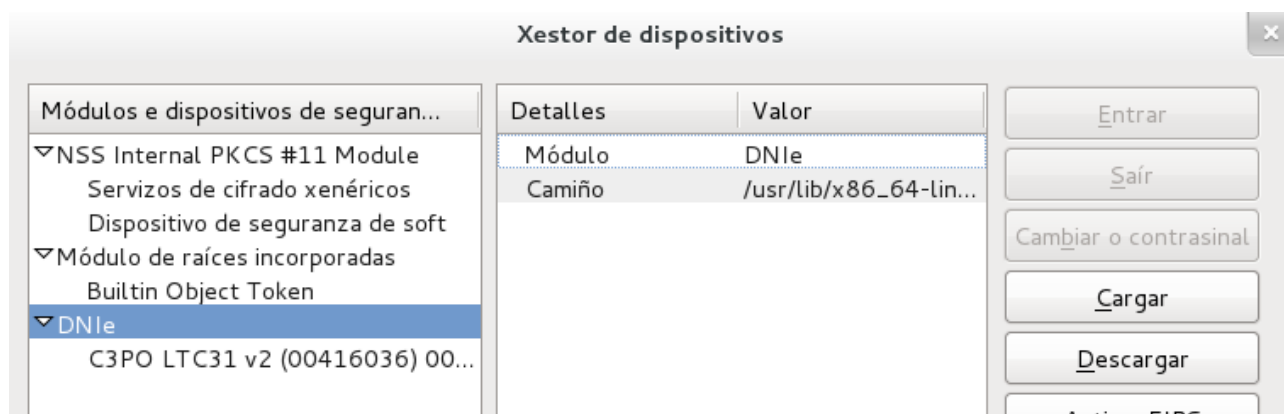
- Importar certificado da FNMT
 - É necesario importar no navegador o certificado raíz da DGP
- Imos a [área de descargas](#) de certificados da páxina do DNIE.
- Descargamos [Certificado pkcs1-sha256WithRSAEncryption](#)
- Descomprimos o arquivo e importamos o certificado no navegador



- Engadir dispositivo de seguridade



Engadimos `/usr/lib/x86_64-linux-gnu/opensc-pkcs11.so`



Comprobando o seu funcionamento

Imos a unha páxina onde podamos empregar o DNLe, por exemplo a www.dgt.es para comprobar o noso saldo de puntos. Escollemos “Saldo de puntos con certificado”



Pode que todo pareza ir ben pero non abre a ventá para introducir o pin. Para solucionalo en

`/etc/opensc/opensc.conf`. Temos que descomentar `enable_pinpad = false;`

```
reader_driver pcsc {
    .....
    #
    # Enable pinpad if detected (PC/SC v2.0.2 Part 10)
    # Default: true
    enable_pinpad = false;
    #
}
```

Agora o acceder á páxina xa debería solicitar o PIN