Yubikeys

TFA au bout des doigts

Journées FedeRez 2011

Thomas DUBOUCHER
thomas@duboucher.eu

19 Mars 2011

Présentation

Nous sommes là pour :

- Découvrir les Yubikeys.

Présentation

Nous sommes là pour :

- Découvrir les Yubikeys.

Nous allons voir :

- Comment configurer rapidement une gestion locale ;
- Des démonstrations.

Présentation

Nous sommes là pour :

- Découvrir les Yubikeys.

Nous allons voir :

- Comment configurer rapidement une gestion locale ;
- Des démonstrations.

Nous ne sommes pas là pour :

- Un cours sur l'authentification.

Un peu de théorie quand même

Authentification à l'aide d'un OTP

Authentification à l'aide d'un mot de passe statique

Identification des Yubikeys

Identification vs. Authentification

Identification

« *J'affirme que je suis* 𝔄 »

- identité publique (nom d'utilisateur, . . .).

Authentification

« *Je prouve que je suis* 𝔄 »

- ce que je sais (mot de passe, question, . . .) ;
- ce que je possède (carte à puce, dongle, . . .) ;
- ce que je suis (empreintes digitales, . . .).

OTP & TFA

One-Time Password

Mot de passe unique valide pour une seule authentification :

- basé par exemple sur une horloge ou un compteur ;
- résistant aux attaques par rejeu ;
- difficile à manipuler sans équipement pour les gérer.

Two-Factor Authentication

Utilisation de deux éléments pour prouver son identité :

- pensez à la carte bleue française.

La Yubikey c'est quoi dans tout ça ?

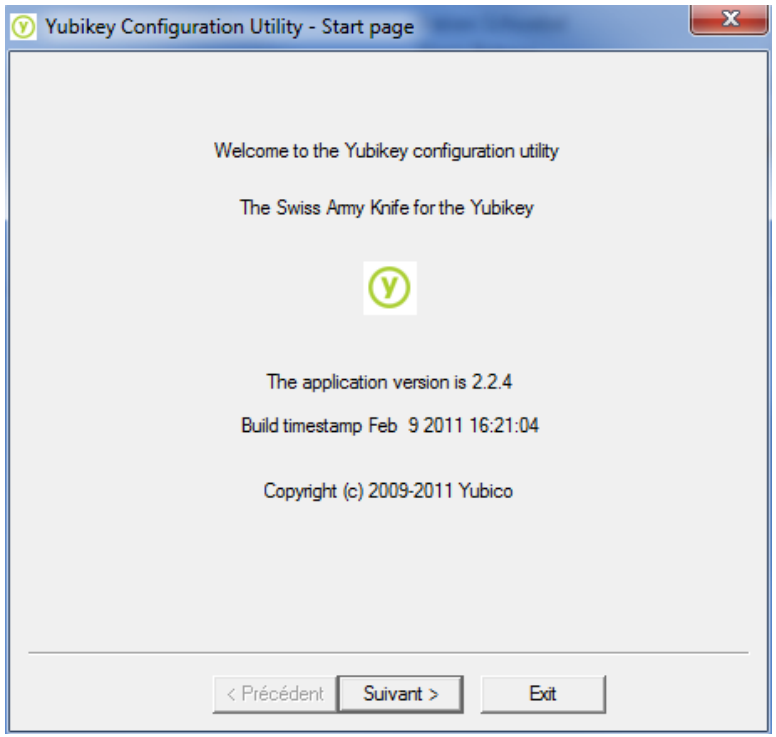
Il s'agit d'un dongle USB assurant l'authentification

- OTP ;
- OATH-HOTP (RFC 4226) ;
- mot de passe statique ;
- *challenge-response* ;
- RFID (MIFARE) sur certains modèles.

Avec plusieurs avantages :

- pas de pilote ou d'application cliente (USB HID) ;
- protocole ouvert ;
- support *open-source*.

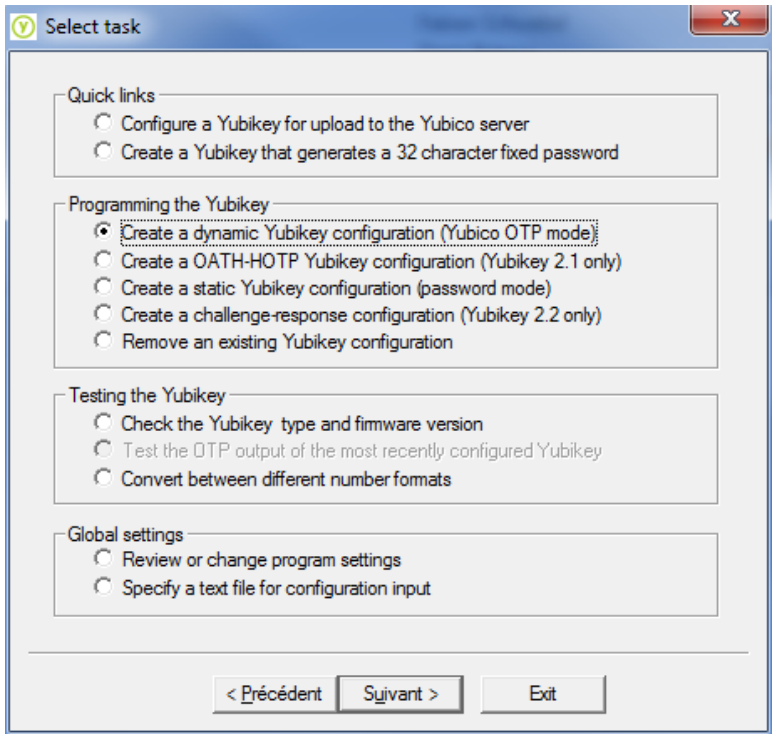
Configuration d'une Yubikey



On utilise le couteau suisse

- Ici, l'outil (très pratique) sous Windows ;
- Un autre outil en ligne de commande est disponible pour Linux.

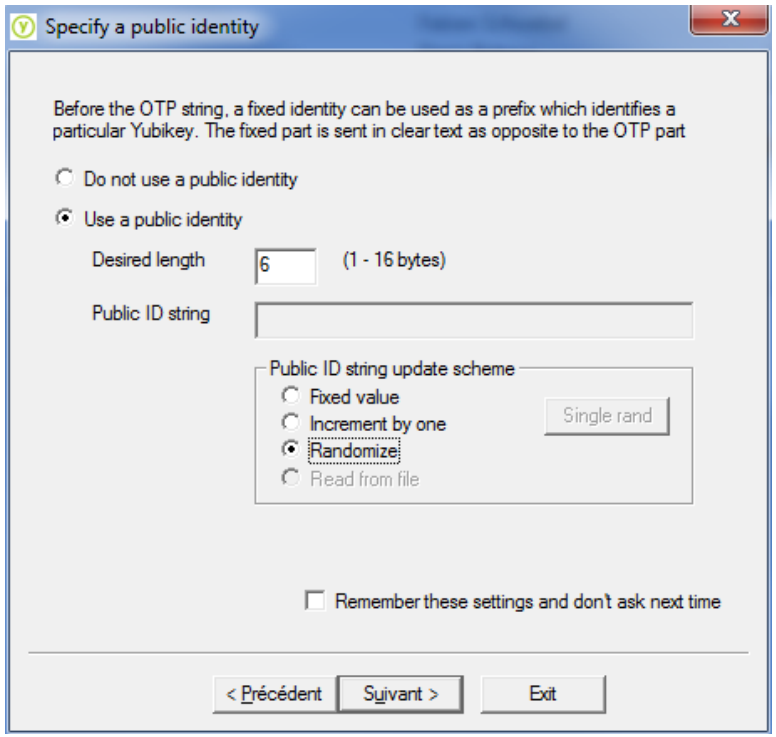
Configuration d'une Yubikey



Select Task

- *Configure a Yubikey for upload to the Yubico server*;
- ou *Create a dynamic Yubikey configuration (Yubico OTP mode).*

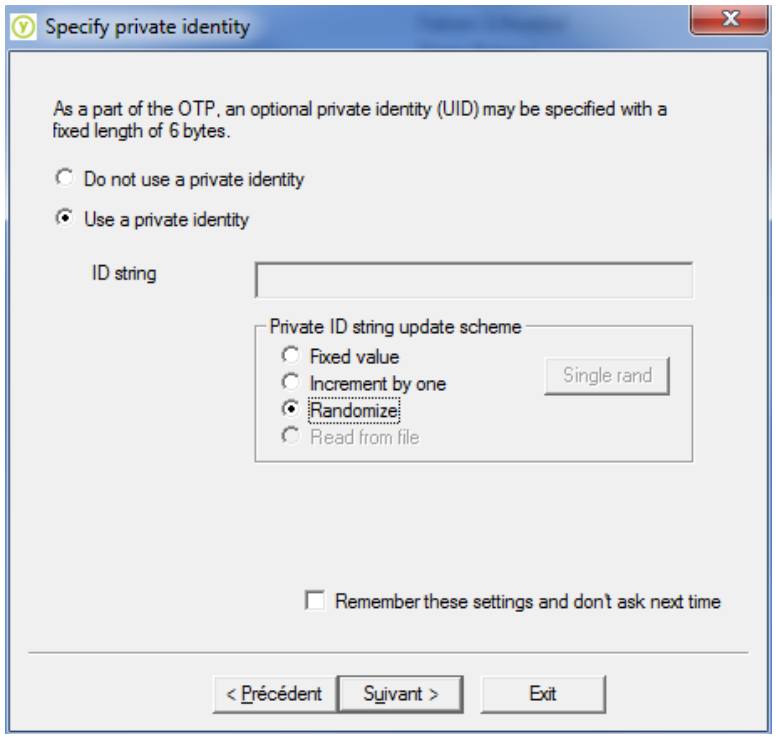
Configuration d'une Yubikey



Specify a public identity

- Est envoyée au début d'un OTP ;
- Permet d'identifier une Yubikey.

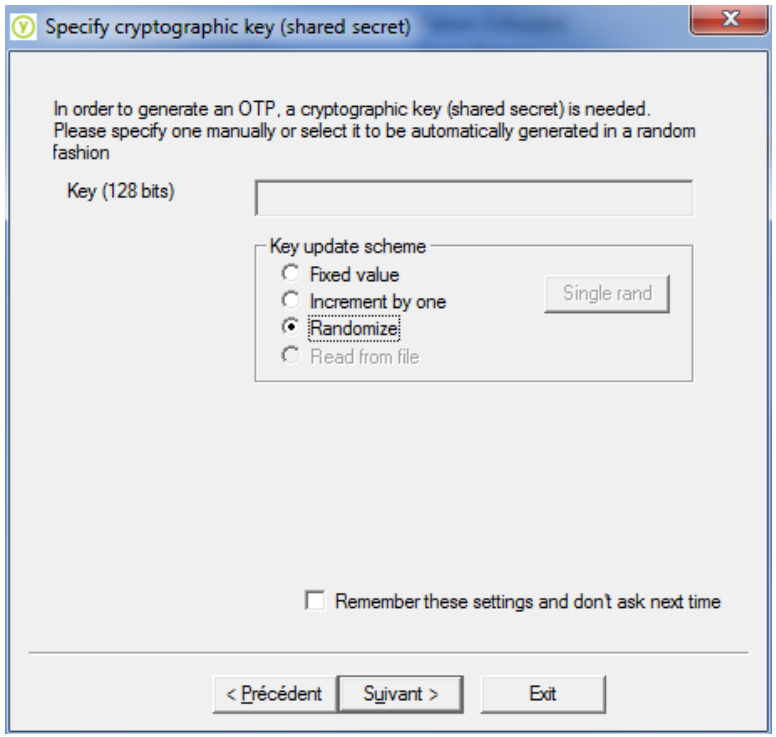
Configuration d'une Yubikey



Specify a private identity

- Secret unique utilisé pour dériver les OTP.

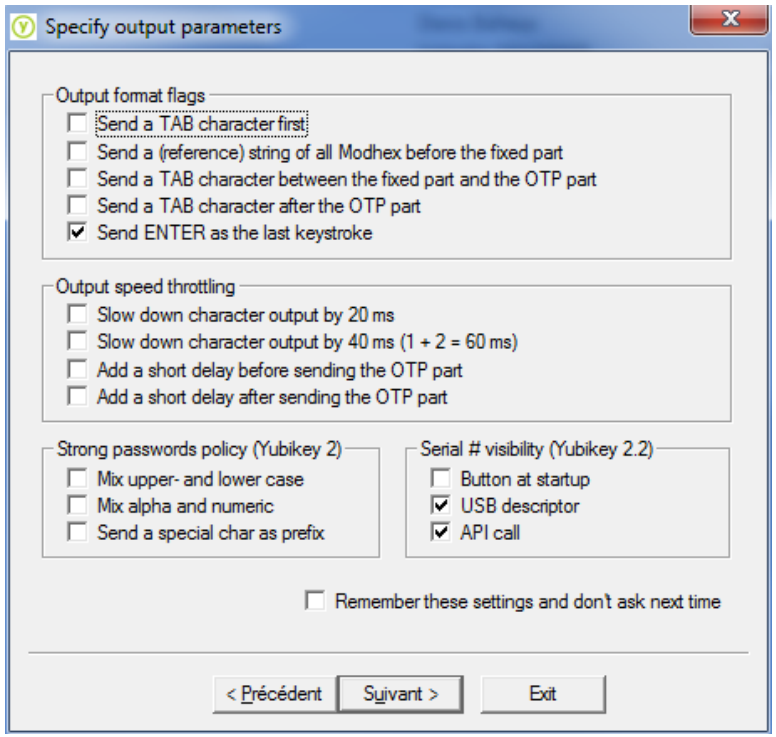
Configuration d'une Yubikey



Specify cryptographic key (shared secret)

- Clef symétrique partagée avec le serveur (AES).

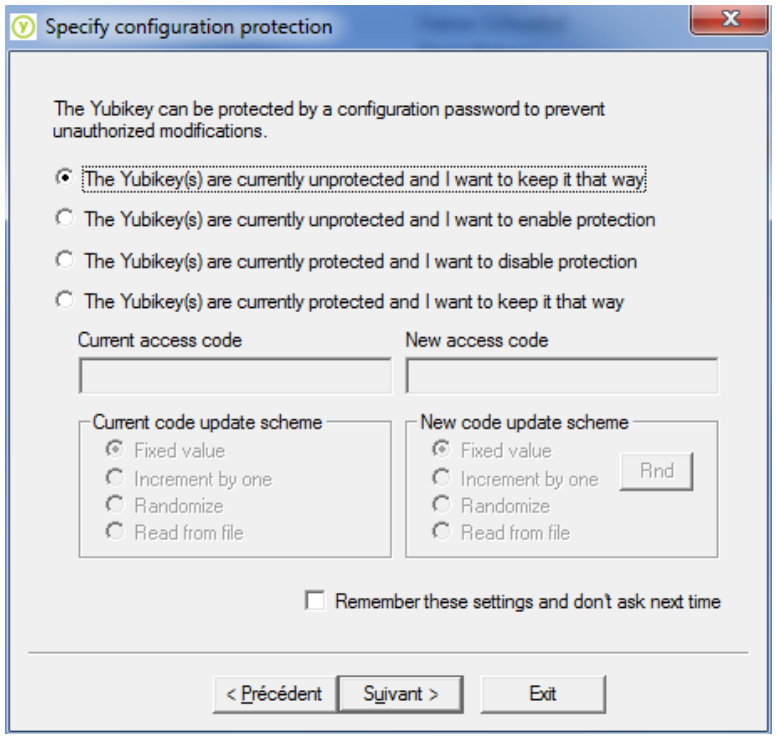
Configuration d'une Yubikey



Specify output parameter

- Insertion de `Tab` ou `Enter` ;
- Gestion de la vitesse de sortie pour les terminaux lents ;
- Caractères utilisés ;
- Visibilité du *serial*.

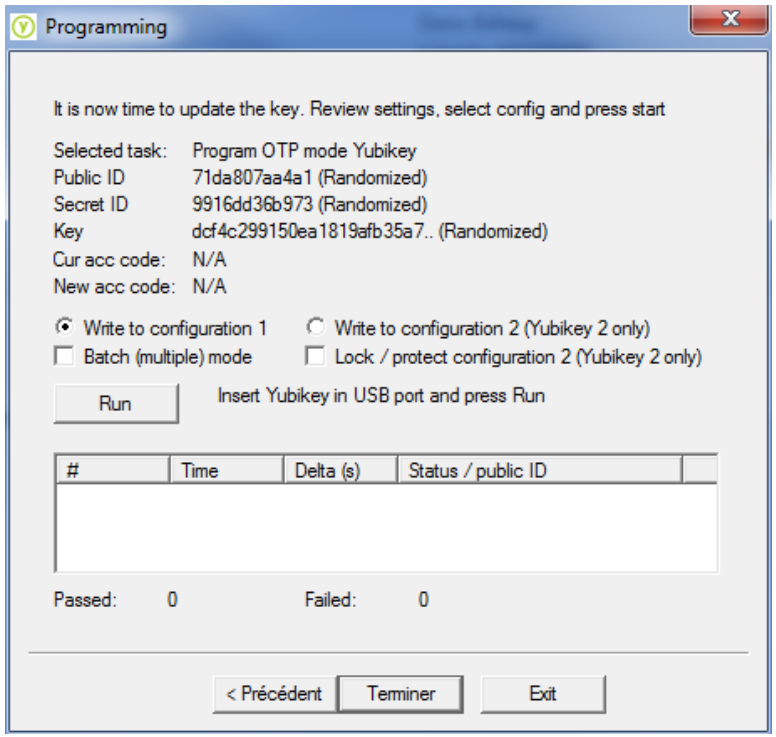
Configuration d'une Yubikey



Specify configuration protection

- Permet de protéger en écriture par une clef la configuration de la Yubikey ;

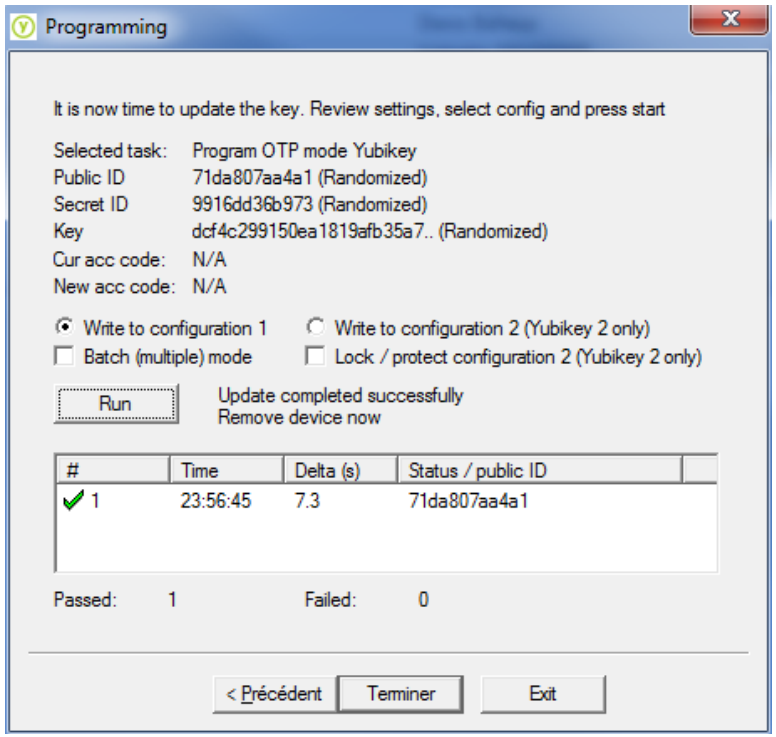
Configuration d'une Yubikey



Programming

- *Batch mode* pour flasher plusieurs clefs à la suite ;
- Choix de la configuration à écraser.

Configuration d'une Yubikey



Programming

- *Batch mode* pour flasher plusieurs clefs à la suite ;
- Choix de la configuration à écraser.

- En voila une !

Construire `libykclient`

Disponible sur `http://code.google.com/p/yubico-c-client/`

- Librairie de validation des OTP ;
- Packages Debian/Fedora disponibles ;
- Penser aux dépendances : `libcurl4-openssl-dev`, ...

Compilation

```
user@localhost:~/libyubikey-1.7$ autoreconf --install
user@localhost:~/libyubikey-1.7$ ./configure
user@localhost:~/libyubikey-1.7$ sudo make check install
```

Construire pam-yubico

Disponible sur `http://code.google.com/p/yubico-pam/`

- Module PAM pour Yubikey ;
- PPA disponible sur `ppa:fredrikt/yubico-pam` ;
- Penser aux dépendances : `libpam-dev`, ...

Compilation

```
user@localhost:~/pam_yubico-2.4$ autoreconf --install
user@localhost:~/pam_yubico-2.4$ ./configure
user@localhost:~/pam_yubico-2.4$ sudo make check install
user@localhost:~/pam_yubico-2.4$ ln -s /usr/local/lib/security/pam_yubico.so \
  /lib/security/
```

Configuration basique de PAM

- la vérification peut se faire à distance en HTTPS avec un LDAP ;

Configuration basique de PAM

- la vérification se fera ici en local en HTTP ;

Configuration basique de PAM

- la vérification se fera ici en local en HTTP ;
- `sufficient` pour utiliser la Yubikey seule ;

/usr/share/pam-config/yubico-ofa

```
Name: Yubikey authentication (one-factor)
Default: yes
Priority: 512
Auth-Type: Primary
Auth-Initial:
        sufficient      pam_yubico.so authfile=/etc/yubikey_mapping \
    url=http://localhost:8700/wsapi/2.0/verify?foo=%d&otp=%s
```

Configuration basique de PAM

- la vérification se fera ici en local en HTTP ;
- `requisite` pour ajouter la Yubikey à la configuration existante ;

/usr/share/pam-config/yubico-tfa

```
Name: Yubikey authentication (two-factor)
Default: yes
Priority: 384
Auth-Type: Primary
Auth-Initial:
        requisite       pam_yubico.so authfile=/etc/yubikey_mapping \
    url=http://localhost:8700/wsapi/2.0/verify?foo=%d&otp=%s
```

Configuration basique de PAM

- la vérification se fera ici en local en HTTP ;
- `requisite` pour ajouter la Yubikey à la configuration existante ;
- l'utilitaire `pam-auth-update` permet une gestion globale de la configuration.

Mise à jour de la configuration

```
1 user@localhost:~$ sudo pam-auth-update
```

Configuration basique de PAM

- la vérification se fera ici en local en HTTP ;
- `requisite` pour ajouter la Yubikey à la configuration existante ;
- l'utilitaire `pam-auth-update` permet une gestion globale de la configuration.
- enfin on associe les Yubikeys aux utilisateurs Unix.

`/etc/yubikey_mapping`

```
1 serianox:jucfhhclerie
```

Configuration de yubiserve

Disponible sur http://code.google.com/p/yubico-yubiserve/

- Serveur léger écrit en Python ;
- Pratique pour une gestion locale ;
- Quelques bugs ...

Installation et lancement

```
user@localhost:~$ sudo aptitude install python python-crypto python-openssl python-sqlite
user@localhost:~$ openssl req -new -x509 -keyout yubiserve.pem -out yubiserve.pem \
  -days 365 -nodes
user@localhost:~$ ./yubiserve.py &
```

Configuration de yubiserve

Disponible sur `http://code.google.com/p/yubico-yubiserve/`

- Serveur léger écrit en Python ;
- Pratique pour une gestion locale ;
- Quelques bugs ...

yubikey.csv

```
1 71da807aa4a1;9916dd36b973;dcf4c299150ea1819afb35a78ac1223c00000000
```

Configuration de yubiserve

Disponible sur http://code.google.com/p/yubico-yubiserve/

- Serveur léger écrit en Python ;
- Pratique pour une gestion locale ;
- Quelques bugs ...

Ajout d'une clef

```
user@localhost:~$ ./dbconf.py -ya serianox-white jucfhhclerie 9916dd36b973 \
  dcf4c299150ea1819afb35a78ac1223c
user@localhost:~$ ./dbconf.py -yl
 1 key  into database:
[Nickname]              >> [PublicID]           >> [Active]
 serianox-white         >>  jucfhhclerie        >>  yes
```

Démo

- Test du serveur en HTTP ;

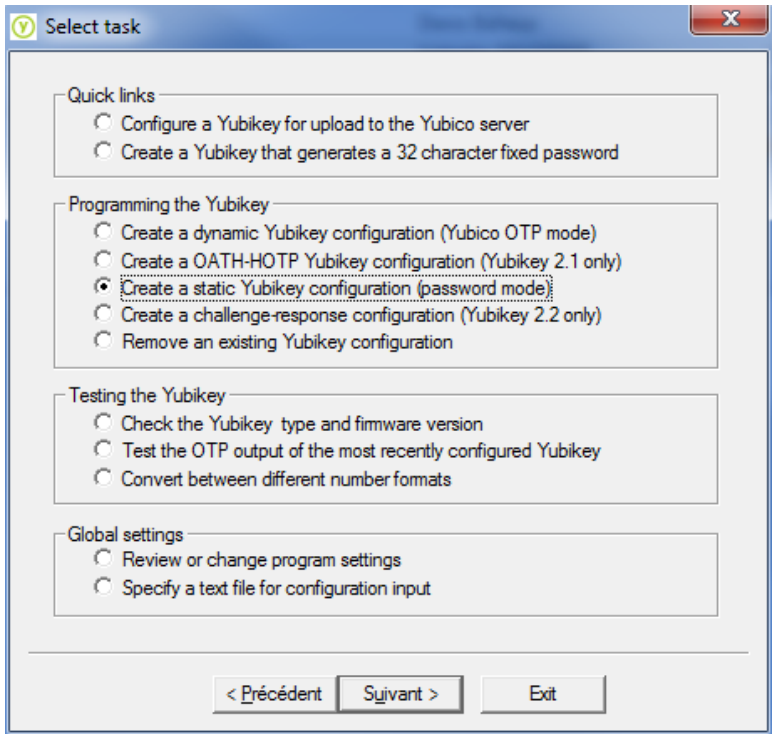
Démo

- Test du serveur en HTTP ;
- Exemple avec sudo ;

Démo

- Test du serveur en HTTP ;
- Exemple avec sudo ;
- Exemple avec FreeRadius.

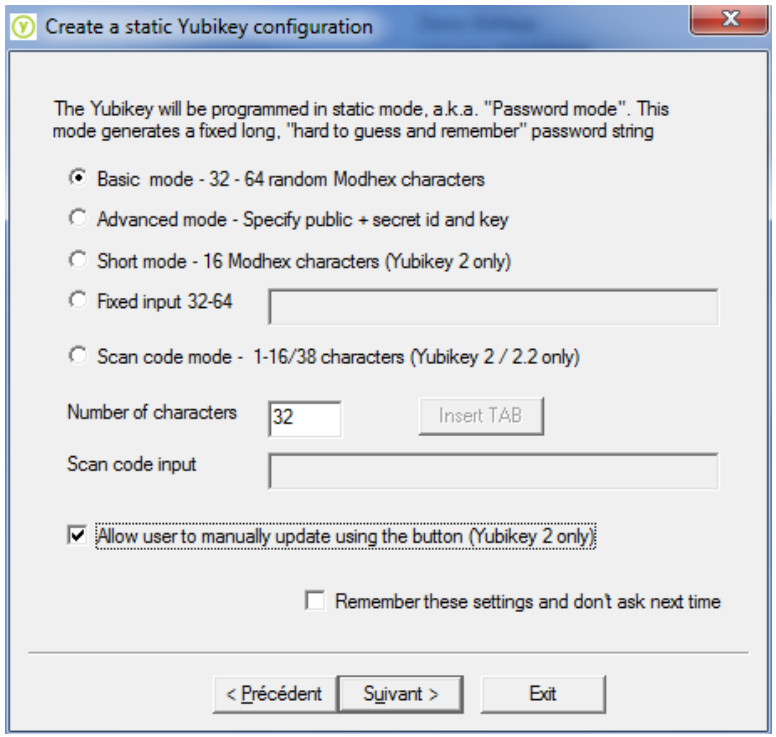
Configuration en mode statique



Select task

- *Create a Yubikey that generate a 32 character fixed password*;
- *Create a static Yubikey configuration (password mode).*

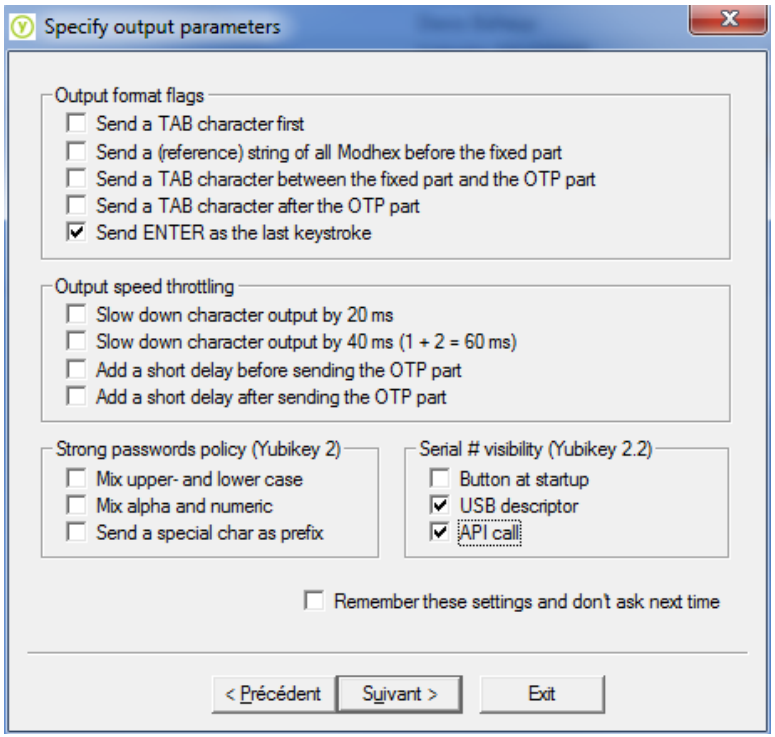
Configuration en mode statique



Create a static Yubikey configuration

- Choix de la longueur ;
- Possibilité de spécifier directement des *scan codes* ;
- Possibilité de laisser l'utilisateur mettre à jour lui-même le mot de passe.

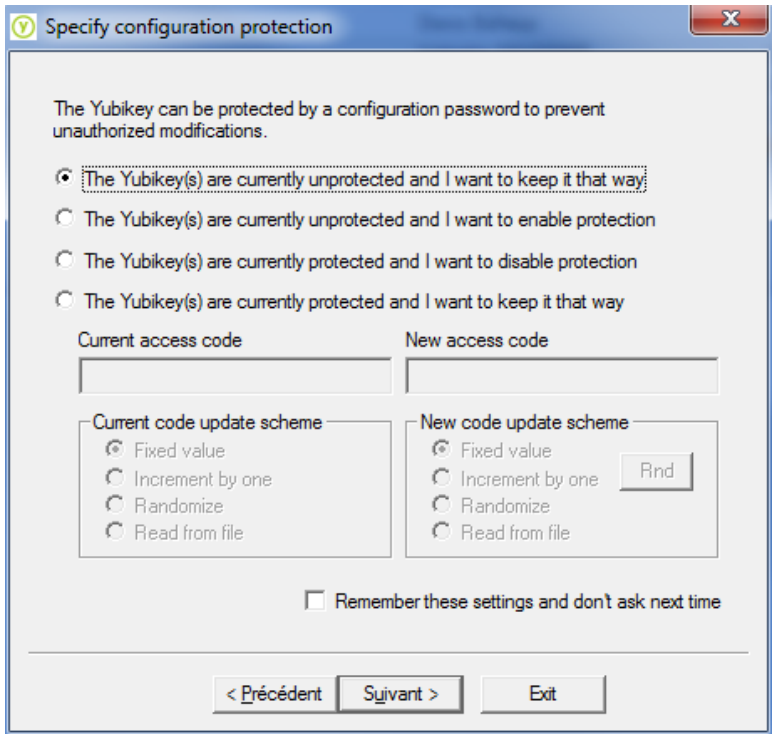
Configuration en mode statique



Specify output parameters

- *idem*

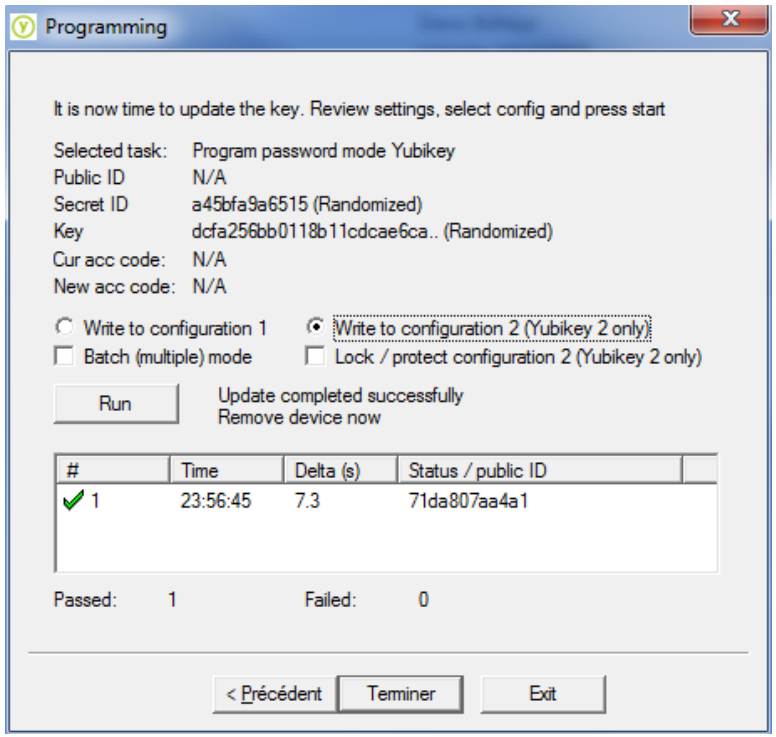
Configuration en mode statique



Specify configuration protection

- *idem*

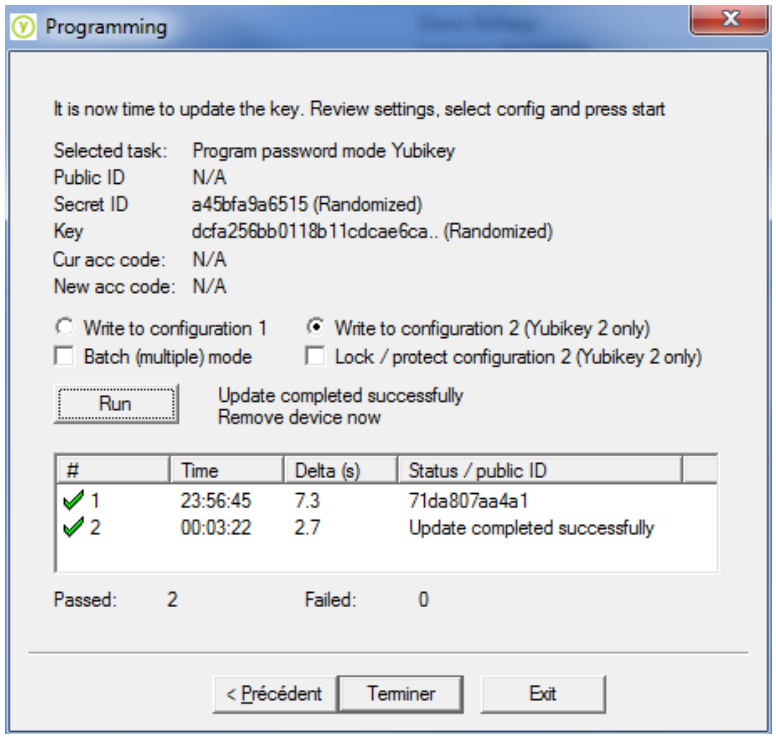
Configuration en mode statique



Programming

- On utilise ici la seconde configuration.

Configuration en mode statique



Programming

- On utilise ici la seconde configuration.

- En voila une autre !

Démo

- Utilisation des deux configurations ;

Démo

- Utilisation des deux configurations ;
- Changement du mot de passe statique.

Utilisation de udev

Console

```
user@localhost:~$ sudo lsusb -vvvv |grep Yubikey -A 20
Bus 002 Device 006: ID 1050:0010 Yubico.com Yubikey
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.00
  bDeviceClass            0 (Defined at Interface level)
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0         8
  idVendor           0x1050 Yubico.com
  idProduct          0x0010 Yubikey
  bcdDevice            2.23
  iManufacturer           1 Yubico
  iProduct                2 Yubico Yubikey II
  iSerial                 3 0000510594
  bNumConfigurations      1
  Configuration Descriptor:
    bLength                 9
...
```

Utilisation de udev

- Le *serial* est visible parmis les descripteurs ;

Utilisation de udev

Console

```
user@localhost:~$ udevadm info -a -p $(udevadm info -q path -n /dev/input/by-id/usb-
    ↪Yubico_Yubico_Yubikey_II_0000510594-event-kbd)
  looking at device  /devices/pci0000:00/0000:00:1d.0/usb2/2-2/2-2:1.0/input/input13/
      ↪event7 :
    KERNEL=="event7"
    SUBSYSTEM=="input"
    DRIVER==""

  looking at parent device  /devices/pci0000:00/0000:00:1d.0/usb2/2-2/2-2:1.0/input/
      ↪input13 :
    KERNELS=="input13"
    SUBSYSTEMS=="input"
    DRIVERS==""
    ATTRS{name}=="Yubico Yubico Yubikey II"
    ATTRS{phys}=="usb-0000:00:1d.0-2/input0"
    ATTRS{uniq}=="0000510594"
    ATTRS{modalias}=="input:b0003v1050p0010e0111-e0,1,4,11,14,k77,7D,7E,7F,ram4,l0
        ↪,1,2,3,4,sfw"

...
```

Utilisation de udev

Console

```
user@localhost:~$ udevadm test --action=add $(udevadm info -q path -n /dev/input/by-id/
    ↪usb-Yubico_Yubico_Yubikey_II_0000510594-event-kbd)
...
udevadm_test: ID_INPUT=1
udevadm_test: ID_INPUT_KEY=1
udevadm_test: ID_INPUT_KEYBOARD=1
udevadm_test: ID_VENDOR=Yubico
udevadm_test: ID_VENDOR_ENC=Yubico
udevadm_test: ID_VENDOR_ID=1050
udevadm_test: ID_MODEL=Yubico_Yubikey_II
udevadm_test: ID_MODEL_ENC=Yubico\x20Yubikey\x20II
udevadm_test: ID_MODEL_ID=0010
udevadm_test: ID_REVISION=0223
udevadm_test: ID_SERIAL=Yubico_Yubico_Yubikey_II_0000510594
udevadm_test: ID_SERIAL_SHORT=0000510594
udevadm_test: ID_TYPE=hid
udevadm_test: ID_BUS=usb
udevadm_test: ID_USB_INTERFACES=:030101:
udevadm_test: ID_USB_INTERFACE_NUM=00
...
```

Utilisation de udev

- Le *serial* est visible parmis les descripteurs ;
- Celui est reconnu par udev ;

Utilisation de udev avec gnome-screensaver

- Le *serial* est visible parmis les descripteurs ;
- Celui est reconnu par udev ;
- On peut s'en servir pour activer l'écran de veille en cas de retrait de la Yubikey ;

/etc/udev/rules.d/85-yubikey.rules

```
1 ACTION=="add", ENV{ID_VENDOR}=="Yubico", \
    RUN+="/usr/local/bin/gnome-screensaver-unlock %E{ID_SERIAL_SHORT}"
  ACTION=="remove", ENV{ID_VENDOR}=="Yubico", \
    RUN+="/usr/local/bin/gnome-screensaver-lock %E{ID_SERIAL_SHORT}"
```

Utilisation de udev avec gnome-screensaver

- Le *serial* est visible parmis les descripteurs ;
- Celui est reconnu par udev ;
- On peut s'en servir pour activer l'écran de veille en cas de retrait de la Yubikey ;
- Pour cela, on associe les clefs aux comptes Unix et on fait un peu de bash.

/etc/yubikey_ownership

```
1 serianox:0000510594:0000505941
```

Utilisation de udev avec gnome-screensaver

/usr/local/bin/gnome-screensaver-lock

```
#!/bin/sh

YUBIKEYS_OWNERSHIP_FILE="/etc/yubikey_ownership"

if [ -n "$1" ]; then
    for proc in $(pgrep -f gnome-screensaver); do
        user=$(ps -p ${proc} u | sed -e  1d  | awk  {print $1} )
        for key in $(grep -m 1 ${user} ${YUBIKEYS_OWNERSHIP_FILE} | sed -e  s
            ↪/[^:]*:\([0-9]\{10\}\(:[0-9]\{10\}\)*\).*/\1/  -e  s/:/ /g ); do
            if [ $1 = $key ]; then
                export $(grep -z DBUS_SESSION_BUS_ADDRESS /proc/${proc}/environ)
                if [ $(basename $0) = "gnome-screensaver-lock" ]; then
                    su ${user} -c "qdbus org.gnome.ScreenSaver / SetActive true"
                elif [ $(basename $0) = "gnome-screensaver-unlock" ]; then
                    su $user -c "gnome-screensaver-command --poke"
                fi;
            fi;
        done;
    done;
fi;
```

Démo

- Surprise ! :)

Questions

?