Universial 2nd Factor (U2F)

Universal 2nd Factor (U2F) is an open standard that strengthens and simplifies two-factor authentication (2FA) using specialized USB or NFC devices based on similar security technology found in smart cards.

For support of U2F in major web browsers and system authentication install the following packages:

```
pacman -S libfido2 pam-u2f
```

Generate U2F key for PAM

> **NOTE:** Generate keys as a regular user!

To start using a U2F key for system-level authentication, keys need to be created first.

The default directory these keys will usually be looked for is at `~/.config/Yubico` (since the `pam-u2f` package is developed by Yubico for use with their Yubikeys). Create the directory under your home directory:

```
mkdir ~/.config/Yubico
```

The `pam-u2f` package comes with a utility to create keys from the USB device. Create new keys with `pamu2fcfg`:

> **WARNING:** This takes your machine's current host name and assumes it is not re-assigned on network changes! Changing your machine's host name might render the key unable to authenticate you until your machine returns to the original host name.

> **NOTE:** You might be silently prompted to interact with a physical button on your key.

```
pamu2fcfg -o pam://$HOST -i pam://$HOST > ~/.config/Yubico/u2f_keys
```

System-wide U2F prompts

To use your physical security key system-wide and not just for specific use-cases, add the following line **before** the first `auth` line in `/etc/pam.d/system-auth`:

> **NOTE:** Be sure to replace `hostname` with the actual host name of your machine!

```
auth        sufficient    pam_u2f.so cue origin=pam://hostname appid=pam://hostname
```

This will prompt you to touch your physical security key during every attempt at authenticating with your user, whether it's in conjunction with graphical system administrator prompts, `sudo` prompts, display manager login prompts, TTY logins, etc.

If the security key is not connected, the system will fall back to regular password prompts.

Passwordless `sudo`

> **WARNING:** Changes to PAM configuration files apply immediately! Before making any changes to your configuration, start a separate shell with root permissions (e.g. `sudo -s`). This way you can revert any changes if something goes wrong.

Open `/etc/pam.d/sudo` and add the following as the first line:

```
auth        sufficient    pam_u2f.so cue origin=pam://hostname appid=pam://hostname
```

Be sure to replace the `hostname` with the actual host name of your machine.

To test, open a new terminal and type `sudo ls`. Your key's LED should flash and after clicking it the command is executed. The option `cue` causes an instruction to appear on what to do, e.g. `Please touch the device`.

Display manager login

GDM

You can use a U2F key as an actual 2nd factor to your password or as a passwordless login method.

2nd factor to password

Open `/etc/pam.d/gdm-password` and add the following line **after** the existing `auth` lines:

> **NOTE:** Be sure to replace `hostname` with the actual host name of your machine!

```
auth        required    pam_u2f.so nouserok cue origin=pam://hostname appid=pam://hostname
```

This will require you to have your U2F physical key inserted to authenticate and log you in with your local user account.

> **WARNING:** If you lose your key you will also lose your ability to authenticate and log in to your user account. You could theoretically use `sufficient` instead of `required` but this would render the security benefits of this endeavour pointless, as the password would still be enough to gain access to your account.

Please note the use of the `nouserok` option which allows the rule to fail if the user did not configure a key or the key is not connected. The `cue` option will display a prompt to let you know the physical key is waiting for you to touch it.

Passwordless login

Open `/etc/pam.d/gdm-password` and add the following line **before** the existing `auth` lines:

> **NOTE:** Be sure to replace `hostname` with the actual host name of your machine!

```
auth        sufficient  pam_u2f.so cue origin=pam://hostname appid=pam://hostname
```

This will prompt you to touch your physical key in order to log into your local user account. If the key is not present you will be asked for your normal password instead.

> **ATTENTION:** Logging into your local user account like this will prevent auto-unlocking your `Login` keychain in GNOME's password manager and you will be prompted to provide it after logging in. There is currently no elegant workaround to this other than removing any password on the default keychain to stop the prompts, but this will leave sensitive information inside it unprotected.

Unlock LUKS container during boot

A FIDO2 key can also be used to unlock your LUKS encrypted drives. To register the key, you will need to use the `systemd-cryptenroll` utility and have a systemd-based initrd.

Run the following command to list your detected keys:

```
systemd-cryptenroll --fido2-device=list
```

Then you can register the key in a LUKS slot, specifying the path to the FIDO2 device, or using the `auto` value if there is only one device:

> **ATTENTION:** Make sure to pass the device node of your actual LUKS container!

```
systemd-cryptenroll --fido2-device=auto /dev/nvme0n1p2
```

To make systemd use the FIDO2 key for unlocking during boot, add the following option to your `rd.luks.options` list of options:

```
rd.luks.options=fido2-device=auto
```

When booting your system, watch for the indicator on your FIDO2 hardware key prompting you to touch it.

Revision #5
Created 2 March 2022 20:55:22 by Sebin
Updated 29 July 2023 22:48:01 by Sebin