What Use Is a Confined User Shell?

Alan Griffiths
What use is a confined user shell?

Alan Griffiths
alan.griffiths@canonical.com
What use is a confined user shell?

Confinement
A confined process has limited access to the system

User shell
A shell interacts with the computer on a user’s behalf: A way to control other programs
Traditionally

- The programs you run can access everything you can
- Installation mechanisms use root access

Confinement

Restricts access to your computer to only those things a program needs to work
Confinement

[DEMO] confined command-line
Confinement
Code running on a computer can be divided into “kernel” and “userspace”

Snap confinement

Snap confinement is Canonical’s chosen approach to confining programs for Ubuntu. Snaps use AppArmor “under the hood”.

The rest of this talk covers snaps and AppArmor because I work with this.

Other packaging and confinement technologies I’m aware of:

- Flatpak uses SELinux
- “clicks” which also use AppArmor

While details differ the principles...
**Confinement**
Code running on a computer can be divided into “kernel” and “userspace”

**Userspace**
The **userspace** is everything that runs within a normal program.

- **Kernel**
  - Hardware: Memory, Filesystems, Outputs, Inputs,...

- **System call interface**

- **App**

- **Userspace**
Confinement
Code running on a computer can be divided into “kernel” and “userspace”

Debian, Ubuntu etc
AppArmor is common in Debian derived distros

Red Hat, Android, etc
SE Linux is common in Red Hat based distros and in Android

Confinement checks system calls
Confinement

Code running on a computer can be divided into “kernel” and “userspace”
**AppArmor**

AppArmor configuration is based on text files. These contain rules for matching resources on the system and specify the access that is permitted.

For example the line:

```
owner /run/user/[0-9]*/wayland-[0-9]* rw,
```

allows read and write access to any files matching the pattern that have the same owner (i.e. user) as the app’s process.
$ wc -l /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.*
  1353 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.daemon
  1373 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.egmde-confined-desktop
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.configure
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.connect-plug-login-session-control
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.connect-plug-wayland
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.disconnect-plug-login-session-control
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.disconnect-plug-wayland
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.install
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.post-refresh
  11889 total
### Confinement

Snap interfaces

Snap connections egmde-confined-desktop

<table>
<thead>
<tr>
<th>Interface</th>
<th>Plug</th>
<th>Slot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>alsa</td>
<td>egmde-confined-desktop:alsa</td>
<td>:alsa</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>audio-playback</td>
<td>egmde-confined-desktop:audio-playback</td>
<td>:audio-playback</td>
</tr>
<tr>
<td>avahi-observe</td>
<td>egmde-confined-desktop:avahi-observe</td>
<td>:avahi-observe</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>content[gtk-3-themes]</td>
<td>egmde-confined-desktop:gtk-3-themes</td>
<td>gtk-common-themes:gtk-3-themes</td>
</tr>
<tr>
<td>content[sound-themes]</td>
<td>egmde-confined-desktop:sound-themes</td>
<td>gtk-common-themes:sound-themes</td>
</tr>
<tr>
<td>locale-control</td>
<td>egmde-confined-desktop:locale-control</td>
<td>:locale-control</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>login-session-control</td>
<td>egmde-confined-desktop:login-session-control</td>
<td>:login-session-control</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mount-observe</td>
<td>egmde-confined-desktop:mount-observe</td>
<td>:mount-observe</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>network-bind</td>
<td>egmde-confined-desktop:network-bind</td>
<td>:network-bind</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>network-observe</td>
<td>egmde-confined-desktop:network-observe</td>
<td>:network-observe</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>opengl</td>
<td>egmde-confined-desktop:opengl</td>
<td>:opengl</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pulseaudio</td>
<td>egmde-confined-desktop:pulseaudio</td>
<td>:pulseaudio</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>removable-media</td>
<td>egmde-confined-desktop:removable-media</td>
<td>:removable-media</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shutdown</td>
<td>egmde-confined-desktop:shutdown</td>
<td>:shutdown</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>system-observe</td>
<td>egmde-confined-desktop:system-observe</td>
<td>:system-observe</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wayland</td>
<td></td>
<td>egmde-confined-desktop:wayland</td>
</tr>
<tr>
<td>manual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x11</td>
<td>egmde-confined-desktop:x11</td>
<td>:x11</td>
</tr>
</tbody>
</table>

Snaps make use of lists of AppArmor rules called “interfaces” each of which covers identifiable capabilities. These can be enabled (or disabled) by the end user.
What use is a confined user shell?

**Confinement**
A confined process has limited access to the system

**User shell**
A shell interacts with the computer on a user’s behalf: A way to control other programs
Mir-kiosk is a simple embedded shell based on Mir

[DEMO] confined graphical shell
Mir-kiosk is a simple embedded shell based on Mir.

The fast, open and secure display server for any device

Whether you want an information kiosk, digital signage display, in-car entertainment stack, or home automation interface, Mir on Ubuntu is your fastest path to deployment.

Get the Mir datasheet
**Graphical shell**

**Confinement**
Shell and App are confined separately.

**Kernel**
- Hardware: Memory, Filesystems, Outputs, Inputs,...
Shell and apps are different

A graphical shell needs...
- User input & output
- Graphics

A web-kiosk needs...
- Network
- Graphics
Shell and apps are different

```bash
$ snap connections mir-kiosk

<table>
<thead>
<tr>
<th>Interface</th>
<th>Plug</th>
<th>Slot</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>opengl</td>
<td>mir-kiosk:opengl</td>
<td>:opengl</td>
<td>-</td>
</tr>
<tr>
<td>wayland</td>
<td>wpe-webkit-mir-kiosk:wayland</td>
<td>mir-kiosk:wayland</td>
<td>manual</td>
</tr>
<tr>
<td>x11</td>
<td>mir-kiosk:x11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Shell and apps are different

```
$ snap connections wpe-webkit-mir-kiosk

<table>
<thead>
<tr>
<th>Interface</th>
<th>Plug</th>
<th>Slot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avahi-observe</td>
<td>wpe-webkit-mir-kiosk:avahi-observe</td>
<td>-</td>
</tr>
<tr>
<td>hostname-control</td>
<td>wpe-webkit-mir-kiosk:hostname-control</td>
<td>-</td>
</tr>
<tr>
<td>network</td>
<td>wpe-webkit-mir-kiosk:network</td>
<td>:network</td>
</tr>
<tr>
<td>network-manager</td>
<td>wpe-webkit-mir-kiosk:network-manager</td>
<td>-</td>
</tr>
<tr>
<td>opengl</td>
<td>wpe-webkit-mir-kiosk:opengl</td>
<td>:opengl</td>
</tr>
<tr>
<td>process-control</td>
<td>wpe-webkit-mir-kiosk:process-control</td>
<td>-</td>
</tr>
<tr>
<td>Upower-observe</td>
<td>wpe-webkit-mir-kiosk:upower-observe</td>
<td>-</td>
</tr>
<tr>
<td>wayland</td>
<td>wpe-webkit-mir-kiosk:wayland</td>
<td>manual</td>
</tr>
</tbody>
</table>
```
Ubuntu Core

A whole OS of snaps

Confinement can be applied to more than apps and shells.

A whole operating system can be built with this technology
A whole OS of snaps
Forme Life is a company based in California developing Studio, the full-length mirror that transforms into personal training.

Mir-kiosk was used on this innovative mirror to provide the foundation for the graphical implementation.
mirr.OS one is the further development of the individual smart home concept. The completely revised system now adapts even better to your needs. mirr.OS one comes with its own web app and uses the new security advantages of Ubuntu Core. On the new board with a grid you can arrange your widgets how you want and as often as you want.

glancr.de
Embedded in IoT devices

This is a picture of a test gateway running a Raspberry CM3 module, connected to a Siemens S7-300 PLC through a MOXA E1212 remote-io

http://tiny.cc/85ebtz
Different shells

**Kiosk mode**

We’ve see few examples of a minimal shell...

- A single fullscreen app
- Launched automatically

**Desktop environment**

- Multiple windowed applications
- Launched by the user
Egmde shell

Egmde is an “example desktop environment” used for testing and demonstration.
This snap confines egmde, and a variety of applications, to illustrate the possibilities and limitations.
This snap confines egmde, and a variety of applications, to illustrate the possibilities and limitations.
Having to package and confine all the applications and shell in a single snap is a limitation, and I’ll talk about that shortly.
[DEMO] confined “desktop” shell
We can run a terminal emulator included in the snap.

```
alan@Octopull-X1C3:~$ whoami
bash: /usr/bin/whoami: Permission denied
alan@Octopull-X1C3:~$ 
```
We can run a browser included in the snap.

The confinement restrictions apply to the browser, so even if compromised by a website it cannot access the host environment.
Embedded in IoT devices

Running on Ubuntu Core on a RPi3b
A login option that restricts access to specific applications.
As a window within a traditional “desktop”

The confinement restrictions apply within the “Mir-on-X” desktop. It cannot access the host system.
The “egmde-confined-desktop” snap is a proof-of-concept, not a finished product

- A confined desktop environment
  - On Ubuntu Core
  - On “classic” Linux (where snapd is supported)
  - A variety of applications included

Including a bespoke set of applications in a new snap is the simplest way to customize this
Mircade is an example snap based on a modified egmde and some games from the Ubuntu archive. This shell launches a single fullscreen app.
Mircade is an example snap based on a modified egmde and some games from the Ubuntu archive.

This shell launches a single fullscreen app.
What use is a confined user shell?

Having to package and confine all the applications in a shell snap is a limitation...

...and I’ll talk about that now!
Shell and apps are different

A shell: needs access to...
- User input and output
- Launching apps

A desktop environment...
- Helpers for keyring & policy kit
- Screensaver, screen lock, suspend, logout, shutdown

An app needs access to...
- $HOME directory & your files
- Network
- Removable media
- Other devices & filesystems
Shell and apps are different

**A shell: needs access to...**
- User input and output
- Launching apps

**A desktop environment...**
- Helpers for keyring & policy kit
- Screensaver, screen lock, suspend, logout, shutdown

**An app needs access to...**
- $HOME directory & your files
- Network
- Removable media
- Other devices & filesystems
Shell and apps are different

A shell: needs access to...

➢ User input and output ✓
➢ Launching apps (currently only within the same snap)

A desktop environment...

➢ Helpers for keyring & policy kit ❌
➢ Screensaver, screen lock, suspend, logout, shutdown ❌
There are several issues to be addressed in order to launch apps...

➢ Identifying the available apps
   ○ There is a standard
     ■ The “Desktop Entry Specification”
       https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html
     ■ But how does it apply in a confined environment?
➢ Confined snaps cannot directly invoke other snaps
Launching apps

Confined snaps cannot directly invoke other snaps

➢ But there is a “userd” process that can...
  ○ ...so we can send it a message
  ○ But, how does userd “police” the requests?
  ○ And, userd only runs on “Classic” systems

➢ There’s “Prior art” in Ubuntu Touch
  ○ Clicks have lomiri-app-launch (formerly ubuntu-app-launch)
What use is a confined user shell?

Mir-kiosk
- “Kiosk” shell
  - On Ubuntu Core
  - Apps must “launch themselves”
    - E.g. wpe-webkit-mir-kiosk

Egmde-confined-desktop
- A confined desktop environment
  - A variety of applications included
  - On classic Linux
  - On Ubuntu Core

Mircade
- Bespoke shell with some games
  - On classic Linux
  - On Ubuntu Core

Future directions
We’re working on ways to enable other snaps to be launched from within a confined snap.

Other desktop environments could be confined with some effort.
Before the “hands on” Questions?
Making a confined user shell

To “play along” you need a computer with:

1. Linux
2. Snaps working: https://snapcraft.io/docs/installing-snapd
3. Git installed
4. A working internet connection
Making a confined shell

Installing the build tools

➢ Snapcraft
➢ Multipass

$ snap install --classic snapcraft
$ snap install --classic multipass
Making a confined shell

Cloning the confined desktop example

➢ Get egmde confined desktop
➢ Switch to the project directory

$ git clone https://github.com/MirServer/egmde-confined-desktop.git
$ cd egmde-confined-desktop
Making a confined shell

```
$ ls -hl
total 12K
drwxr-xr-x 4 alan alan 4.0K Mar 9 16:27 glue
-rw-r--r-- 1 alan alan  259 Mar 9 16:27 README.md
drwxr-xr-x 4 alan alan 4.0K Mar 9 16:27 snap
$ ls -hl snap
total 16K
drwxr-xr-x 2 alan alan 4.0K Mar 9 16:27 hooks
drwxr-xr-x 2 alan alan 4.0K Mar 9 16:27 plugins
-rw-r--r-- 1 alan alan 6.2K Mar 9 16:27 snapcraft.yaml
```
Making a confined shell

$ snapcraft
Launching a VM.
...
Snapped egmde-confined-desktop_139-mir2.3.2-snap80_amd64.snap

$ snap install --dangerous *.snap
egmde-confined-desktop 139-mir2.3.2-snap80 installed

$ /snap/egmde-confined-desktop/current/bin/setup.sh
...

$ egmde-confined-desktop
A look at the snapcraft.yaml
Making a confined shell

**egmde-confined-desktop**

The “egmde-confined-desktop” snap is a proof-of-concept, not a finished product

- A confined desktop environment
  - On Ubuntu Core
  - On “classic” Linux (where snapd is supported)
  - A variety of applications included

Including a bespoke set of applications in a new snap is the simplest way to customize this

**Future directions**

Other snapped applications can be run on the “egmde-confined-desktop” desktop, but need to be launched from outside the snap. We’re investigating ways to enable other snaps to be launched from within a confined snap

Other desktop environments could be confined with some effort. There’s work being done with GNOME
Thank you. Questions?