ATOMIC OBJECT

March 10, 2015 by: Justin Kulesza 9 Comments

Protecting the Root Filesystem on Ubuntu with Overlayroot

In certain situations, it is desirable to have a read-only root filesystem. This prevents any changes from occurring on the root filesystem that may alter system behavior, and it allows a simple reboot to restore a system to its pristine state. Examples of such applications include kiosks and embedded devices. Using overlayroot on Ubuntu makes creating a read-only root filesystem quick and easy.

Motivation

Let us imagine that we have an embedded Linux device that needs to run a particular program. The program does not need to store any data (except perhaps some logs), and we want to protect the system against any changes. If the program or any system processes make filesystem modifications, we want to toss them out and return to the original state. We want a robust system that will run our program, but be restored to it's 'factory' state with a power-cycle.

Background

Overlayroot is a package for Ubuntu which utilizes OverlayFS, a union filesystem implementation. OverlayFS presents a unified view of two different filesystems; the presented filesystem is the result of overlaying one filesystem over another.

In OverlayFS, there is an 'upper' filesystem and a 'lower' filesystem. If a particular object exists in both the *upper* filesystem and the *lower* filesystem, the object from the *upper* filesystem is presented, and the object from the *lower* filesystem is hidden. If the object is a directory, the contents of the directory on the *upper* and *lower* filesystems are merged and presented.

With overlayroot, the *lower* filesystem is a read-only mount of the root filesystem, and the *upper* filesystem is a read-write mount of another block device. That block device can be `tmpfs`, a standard block device, or an encrypted block device (à la `dmcrypt`).

All changes made on top of the root filesystem are stored elsewhere and do not affect the root filesystem. Depending on the block device chosen, those changes can persist across reboots. While the root filesystem will not be affected, the view presented by OverlayFS will include any changes stored in the *upper* filesystem.

Situation

For our embedded Linux device, we are not interested in preserving any changes to the root filesystem. Any data that we want to preserve will be stored on a separate filesystem dedicated to data storage.

Here is our filesystem structure:

- `/dev/sda1` on `/`
- `/dev/sda2` on `/data`

We want to mount `/` using overlayroot so that we have a read-write filesystem available but not persist any changes to `/`, and we want to mount `/data` normally so that our program can write out and persist log files.

Solution

Overlayroot has been available since Ubuntu 12.10, and has been back-ported to 12.04 LTS. It is quick to install:

```
apt-get install overlayroot
```

The configuration file is stored at `/etc/overlayroot.conf`, and contains a wealth of in-line documention.

The only item to change is the `overlayroot` variable. By default, it is blank:

```
overlayroot=""
```

Start Simple

Since we want to use `tmpfs` mode for overlayroot, we set the variable accordingly.

```
overlayroot="tmpfs"
```

The other modes (specifying a device, or `crypt`) are documented in `overlayroot.conf`, with some nice diagrams on the post introducing overlayroot.

Add Swap

By default, overlayroot disables swap. This can be re-enabled by providing another option to the configuration:

```
overlayroot="tmpfs:swap=1"
```

(The default value for `swap` is `0`).

Don't Recurse

By default, overlayroot will mount all filesystems under `/` in the specified mode. This can be prevented by adding another option to the configuration:

```
overlayroot="tmpfs:swap=1,recurse=0"
```

Note that `swap=1` and `recurse=0` are separated by a comma, not a colon.

(The default value for `recurse` is `1`).

This will prevent overlayroot from overlaying a read-write `tmpfs` on top of a read-only `/data` filesystem. All of `/` will be protected from any modifications, except `/data`, which is what we want.

The Result

After making changes to `overlayroot.conf` and rebooting, the root filesystem should now be mounted as a read-write `tmpfs` overlaying the read-only `/` filesystem.

Running `mount`, we should see something like:

```
overlayroot on / type overlayfs (rw,lowerdir=/media/root-ro/,upperdir=/media/root-rw)
proc on /proc type proc (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,noexec,nosuid,nodev)
none on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/fuse/connections type fusectl (rw)
devtmpfs on /dev type devtmpfs (rw,mode=0755)
none on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
none on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
/dev/sda1 on /media/root-ro type ext4 (ro)
tmpfs-root on /media/root-rw type tmpfs (rw,relatime)
/dev/sda2 on /data type ext4 (rw)
```

Note how the root filesystem is of type `overlayfs` with specified upper and lower filesystems that correspond to the read-write `tmpfs` and read-only mounts listed.

Disabling Overlayroot

Once overlayroot is enabled, it is no longer possible to make changes to the root filesystem. This introduces problems if we legitimately need to make changes (such as for updates, tweaks, etc.). Fortunately, we can override any overlayroot configuration by passing `overlayroot=disabled` to the kernel at boot. This can be done as a one-off operation as needed, or can be set up as an alternate boot configuration in a boot loader such as GRUB.

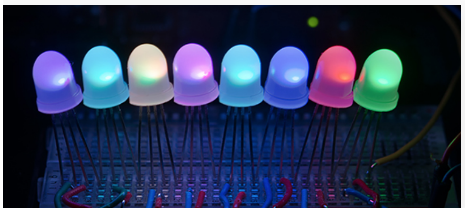
For example, with GRUB2, something like the following could be used:

```
menuentry 'Ubuntu, with Linux 3.5.0-54-generic (Writable)' --class ubuntu --class gnu-linux --class gnu --cl
        recordfail
        gfxmode $linux_gfx_mode
        insmod gzio
        insmod part_msdos
        insmod ext2
        set root='(hd0,msdos1)'
        search --no-floppy --fs-uuid --set=root 28adfe9d-c122-479a-ab81-de57d16516dc
        linux   /vmlinuz-3.5.0-54-generic root=/dev/mapper/faramir-root ro overlayroot=disabled
        initrd  /initrd.img-3.5.0-54-generic
}
```

Conclusion

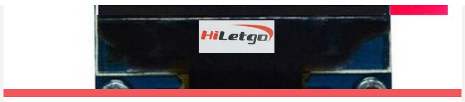
Overlayroot makes the process of mounting the root filesystem as read-only on Ubuntu very easy. Prior to the availability of overlayroot packages, custom scripts were added to the `initramfs` configuration. These scripts were often fragile, and not always compatible across versions of Ubuntu. With the availability of OverlayFS and overlayroot, creating robust kiosks and embedded Linux devices with rollback capabilities is now extremely easy.

Related Posts



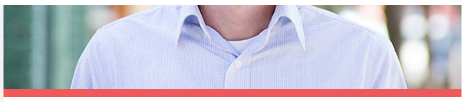
A Library for Driving NeoPixels with the ESP32 Micro Controller

by Jordan Schaenzle



Adding an OLED to Your Particle Device

by Jordan Schaenzle



A Simple Message Queue for C

by Jordan Schaenzle