DRM Security

David Herrmann
<dh.herrmann@gmail.com>

DRM

rendering *mode-setting*

open(“/dev/dri/card0”, ...)

(Dumb) Buffer Allocation

ioctl(fd, DRM_IOCTL_MODE_CREATE_DUMB, &creq)

mreq.handle = creq.handle;

ioctl(fd, DRM_IOCTL_MODE_MAP_DUMB, &mreq)

mmap(fd, size, …, MAP_SHARED, …, mreq.offset)

mmap-offsets are globally accessible

```
for (i = 0; i < 0xffffffff; ++i) {
    void *p = mmap(fd, …, i);
    if (p != MAP_FAILED)
        break;
}
```

```
for (i = 0; i < 0xffffffff; ++i) {
    void *p = mmap(fd, …, i);
    if (p != MAP_FAILED)
        break;
}
```

FIXED

ioctl(fd, DRM_IOCTL_MODE_CREATE_DUMB, &creq)

mreq.handle = creq.handle;

ioctl(fd, DRM_IOCTL_MODE_MAP_DUMB, &mreq)

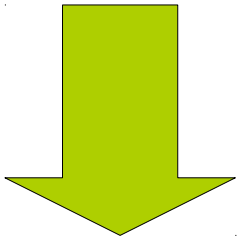
mmap(fd, size, …, MAP_SHARED, …, mreq.offset)

Buffer Passing

lreq.handle = mreq.handle;

ioctl(fd, **DRM_IOCTL_GEM_FLINK**, &lreq);

send(somewhere, lreq.name, sizeof(lreq.name));



recv(somewhere, &oreq.name, sizeof(oreq.name));

ioctl(fd, **DRM_IOCTL_GEM_OPEN**, &oreq);

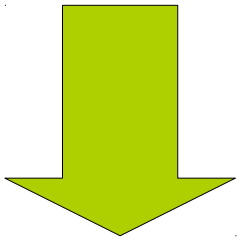
```
for (i = 1; i < 0xffffffff; ++i) {
    req.name = i;
    r = ioctl(fd, DRM_IOCTL_GEM_OPEN, &req);
    if (!r)
        break;
}
```

```
for (i = 1; i < 0xffffffff; ++i) {
    req.name = i;
    r = ioctl(fd, DRM_IOCTL_GEM_OPEN, &req);
    if (!r)
        break;
}
```

DEPRECATED

dma-buf

```
req.handle = mreq.handle;
ioctl(fd, DRM_IOCTL_PRIME_HANDLE_TO_FD, &req);
send_unix_fd(somewhere, req.fd);
```



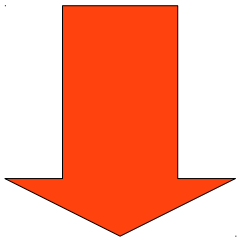
```
recv_unix_fd(somewhere, &req.fd);
ioctl(fd, DRM_IOCTL_PRIME_FD_TO_HANDLE, &req);
```

DRM Authentication

open(“/dev/dri/card0”, …);

ioctl(fd, **DRM_IOCTL_GET_MAGIC**, &magic);

send(somewhere, &magic, sizeof(magic));



recv(somewhere, &magic, sizeof(magic));

ioctl(fd, **DRM_IOCTL_AUTH_MAGIC**, &magic);

ioctl(fd, **DRM_IOCTL_GET_MAGIC**, &magic);

send(somewhere, &magic, sizeof(magic));

recv(somewhere, &magic, sizeof(magic));

ioctl(fd, **DRM_IOCTL_AUTH_MAGIC**, &magic);

Access-Management is done via file-system modes!

There is a reason FD-passing
is call **SCM_RIGHTS**

open(“/dev/dri/renderD128”, …);

Render Nodes

- No GEM_FLINK
- No DRM AUTH/MAGIC
- No Mode-setting
- No global resources
- No legacy DRM API
- No DRM-Master

OpenGL with Render-Nodes

fd = **open**(“/dev/dri/renderD128”, …);

dev = **gbm_create_device**(fd);

disp = **eglGetDisplay**(dev);

eglInitialize(disp, major, minor);

eglBindAPI(EGL_OPENGL_API);

eglChooseConfig(disp, …, &conf);

ctx = **eglCreateContext**(disp, conf, 0, &attrs);

eglMakeCurrent(disp, 0, 0, ctx);

```
surf = gbm_surface_create(dev, … attrs ...);
wnd = eglCreateWindowSurface(disp, conf, surf, …);
eglMakeCurrent(disp, wnd, wnd, ctx);
```

DRM Master

- acquire DRM-Master:
  - **open()**

    or
  - **drmSetMaster()**

- drop DRM-Master
  - **close()**

    or
  - **drmDropMaster()**

- acquire DRM-Master:
  - **open()**

    or
  - **drmSetMaster()**

- drop DRM-Master
  - **close()**

    or
  - **drmDropMaster()**