

## [patch 4/6] Add MMC password protection (lock/unlock) support V4

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-01/msg11533.html>

---

- *From:* Carlos Aguiar <carlos.aguiar@xxxxxxxxxxxx>
  - *Date:* Tue, 31 Jan 2006 16:16:40 -0400
- 

Implement MMC password force erase, remove password, change password, unlock card and assign password operations. It uses the sysfs mechanism to send commands to the MMC subsystem.

Signed-off-by: Anderson Briglia <anderson.briglia@xxxxxxxxxxxx>  
Signed-off-by: Anderson Lizardo <anderson.lizardo@xxxxxxxxxxxx>  
Signed-off-by: Carlos Eduardo Aguiar <carlos.aguiar@xxxxxxxxxxxx>

Index: linux-omap-2.6.git/drivers/mmc/mmc\_sysfs.c

```
=====
--- linux-omap-2.6.git.orig/drivers/mmc/mmc_sysfs.c 2006-01-31 15:22:25.000000000 -0400
+++ linux-omap-2.6.git/drivers/mmc/mmc_sysfs.c 2006-01-31 15:22:29.000000000 -0400
@@ -16,6 +16,7 @@
#include <linux/device.h>
#include <linux/idr.h>
#include <linux/key.h>
+#include <linux/err.h>

#include <linux/mmc/card.h>
#include <linux/mmc/host.h>
@@ -23,6 +24,12 @@

#include "mmc.h"

+#ifdef CONFIG_MMC_DEBUG
+#define DBG(x...) printk(KERN_DEBUG x)
+#else
+#define DBG(x...) do { } while (0)
+#endif
+
+
#define dev_to_mmc_card(d) container_of(d, struct mmc_card, dev)
#define to_mmc_driver(d) container_of(d, struct mmc_driver, drv)
#define cls_dev_to_mmc_host(d) container_of(d, struct mmc_host, class_dev)
@@ -64,6 +71,101 @@ static struct device_attribute mmc_dev_a

static struct device_attribute mmc_dev_attr_scr = MMC_ATTR_RO(scr);

+#ifdef CONFIG_MMC_PASSWORDS
```

[patch 4/6] Add MMC password protection (lock/unlock) support V4

```
+
+static ssize_t
+mmc_lockable_show(struct device *dev, struct device_attribute *att, char *buf)
+{
+ struct mmc_card *card = dev_to_mmc_card(dev);
+
+ if (!mmc_card_lockable(card))
+ return sprintf(buf, "unsupported\n");
+ else
+ return sprintf(buf, "%slocked\n", mmc_card_locked(card) ?
+ "" : "un");
+}
+
+/*
+ * implement MMC password functions: force erase, remove password, change
+ * password, unlock card and assign password.
+ */
+static ssize_t
+mmc_lockable_store(struct device *dev, struct device_attribute *att,
+ const char *data, size_t len)
+{
+ struct mmc_card *card = dev_to_mmc_card(dev);
+
+ if (!mmc_card_lockable(card))
+ return -EINVAL;
+
+ if (mmc_card_locked(card) && !strncmp(data, "erase", 5)) {
+ /* forced erase only works while card is locked */
+ mmc_lock_unlock(card, NULL, MMC_LOCK_MODE_ERASE);
+ return len;
+ } else if (!mmc_card_locked(card) && !strncmp(data, "remove", 6)) {
+ /* remove password only works while card is unlocked */
+ struct key *mmc_key = request_key(&mmc_key_type, "mmc:key",
+ "remove");
+
+ if (!IS_ERR(mmc_key)) {
+ int err = mmc_lock_unlock(card, mmc_key,
+ MMC_LOCK_MODE_CLR_PWD);
+ if (!err)
+ return len;
+ } else
+ DBG("request_key returned error %ld\n",
+ PTR_ERR(mmc_key));
+ } else if (!mmc_card_locked(card) && !strncmp(data, "change", 6)) {
+ /* change */
+ struct key *mmc_key = request_key(&mmc_key_type,
+ "mmc:key", "change");
+ if (!IS_ERR(mmc_key)) {
+ int err = mmc_lock_unlock(card, mmc_key,
+ MMC_LOCK_MODE_SET_PWD);
+ if (!err)
```

[patch 4/6] Add MMC password protection (lock/unlock) support V4

```
+ return len;
+ } else
+ DBG("request_key returned error %ld\n",
+ PTR_ERR(mmc_key));
+ } else if (mmc_card_locked(card) && !strcmp(data, "unlock", 6)) {
+ /* unlock */
+ struct key *mmc_key = request_key(&mmc_key_type,
+ "mmc:key", "unlock");
+ if (!IS_ERR(mmc_key)) {
+ int err = mmc_lock_unlock(card, mmc_key,
+ MMC_LOCK_MODE_UNLOCK);
+ if (err)
+ DBG("Wrong password\n");
+ device_release_driver(dev);
+ device_attach(dev);
+ if (!err)
+ return len;
+ } else
+ DBG("request_key returned error %ld\n",
+ PTR_ERR(mmc_key));
+ } else if (!mmc_card_locked(card) && !strcmp(data, "assign", 6)) {
+ /* assign */
+ struct key *mmc_key = request_key(&mmc_key_type,
+ "mmc:key", "assign");
+ if (!IS_ERR(mmc_key)) {
+ int err = mmc_lock_unlock(card, mmc_key,
+ MMC_LOCK_MODE_SET_PWD);
+ if (!err)
+ return len;
+ } else
+ DBG("request_key returned error %ld\n",
+ PTR_ERR(mmc_key));
+ }
+
+ return -EINVAL;
+}
+
+static struct device_attribute mmc_dev_attr_lockable =
+ __ATTR(lockable, S_IWUSR | S_IRUGO,
+ mmc_lockable_show, mmc_lockable_store);
+
+ #endif
+
static void mmc_release_card(struct device *dev)
{
@@ -233,6 +335,11 @@ int mmc_register_card(struct mmc_card *c
if (ret)
device_del(&card->dev);
}
+ #ifdef CONFIG_MMC_PASSWORDS
```

