Replicant - Issue #2043

Investigate if it's possible to detect and recover EFS backups corrupted by 'adb shell "cat /dev/block/platform/*/by-name/EFS" > /EFS.img'

06/08/2020 06:57 PM - Denis 'GNUtoo' Carikli

Status:	New	**Start date:**	06/08/2020
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	Replicant 6.0	**Spent time:**	0.00 hour
Resolution:		**Grant:**	
Device:	Galaxy Nexus (GT-I9250), Galaxy Nexus (I9250), Galaxy Note (GT-N7000), Galaxy Note (N7000), Galaxy Note 2 (GT-N7100), Galaxy Note 2 (N7100), Galaxy Note 8.0 (N51xx), Galaxy S (I9000), Galaxy S 2 (I9100), Galaxy S 3 (I9300), Galaxy S 3 4G (I9305), Galaxy Tab 2 10.1 (P51xx), Galaxy Tab 2 7.0 (P31xx), Nexus One, Nexus S (I902x)	**Type of work:**	C programming, Unknown

Description

History

#1 - 06/08/2020 07:08 PM - Denis 'GNUtoo' Carikli

- Subject changed from Test the recovery of EFS images backed up incorrectly to Investigate if it's possible to detect and recover EFS backups corrupted by 'adb shell "cat /dev/block/platform//by-name/EFS > /EFS.img"'*

Before, EFS backups used the following as part of the procedure:

```
adb shell "cat /dev/block/platform/*/by-name/EFS > /EFS.img"
```

At some point, this started to create corrupted images.

There might be a way to fix the images.

In BackupResearch revision 15, doak complex tested the following as part of research to use adb pull in one command:

```
Be aware of the @head --bytes=-2@ which is needed to get rid of the nasty @0x0d 0x0a@ line ending returned by @adb@.
(Tested on Recovery of Replicant 6.0 0003.)
```

So the images that were backuped with the bad command have might just have the two extra bytes mentioned above.

#2 - 06/08/2020 07:16 PM - Denis 'GNUtoo' Carikli

- Subject changed from Investigate if it's possible to detect and recover EFS backups corrupted by 'adb shell "cat /dev/block/platform//by-name/EFS > /EFS.img"' to Investigate if it's possible to detect and recover EFS backups corrupted by 'adb shell "cat /dev/block/platform/*/by-name/EFS" > /EFS.img'*

The two bytes seem trivial to remove, however detecting if that has already been done in a way that is 100% reliable is harder:

- The two bytes are appended, so we could still try to parse the filesystem but that would be error prone.
- The last 2 bytes of a non corrupted backup might be the same than the ones we know are appended. Removing them would corrupt good backups.
- We don't know if the EFS partition size differ across various versions (16G, 32G) of a same device variant (GT-I9300). However we could try to find that with adb. If the backup is bigger than the partition it's supposed to be restored to, and that the two bytes are 0x0d and 0x0a, we could create a <file>.fixed for instance.

#3 - 06/08/2020 07:27 PM - Denis 'GNUtoo' Carikli

I've tried With a GT-I9100

We can reproduce it easily:

```
$ adb shell "cat /dev/block/platform/*/by-name/EFS" > corrupted_EFS.img
$ adb shell "cat /dev/block/platform/*/by-name/EFS > /EFS.img"
$ adb pull /EFS.img ./
/EFS.img: 1 file pulled. 2.7 MB/s (20971520 bytes in 7.505s)
```

Unfortunately we have more than 2 bytes of difference:

```
$ cmp EFS.img corrupted_EFS.img
EFS.img corrupted_EFS.img differ: byte 1293, line 1
$ du corrupted_EFS.img EFS.img
20488    corrupted_EFS.img
20480    EFS.img
```

Here's the first diff we have with vbindiff.

From the good backup:

```
0000 0500: 00 00 00 00 00 00 00 00  00 00 00 00 0A F3 01 00  ........ ........
```

And the bad backup:

```
0000 0500: 00 00 00 00 00 00 00 00  00 00 00 00 0D 0A F3 01  ........ ........
```

So from time to time some 0D are inserted, and the position (0x500) is very far from the end of the file (0x5000) .

#4 - 02/27/2021 08:40 AM - Denis 'GNUtoo' Carikli

- Device Galaxy Nexus (GT-I9250) added

#5 - 02/27/2021 08:50 AM - Denis 'GNUtoo' Carikli

- Device Galaxy Note (GT-N7000) added

#6 - 02/27/2021 10:06 AM - Denis 'GNUtoo' Carikli

- Device Galaxy Note 2 (GT-N7100) added

#7 - 03/26/2021 05:10 PM - _I3^ RELATIVISM

- Type of work C programming, Unknown added