



Backup You Can TrustSM

Proving The BRU Advantage

In 1985, Fred Fish was using tar to protect his business data and discovered he was unable to successfully recover it. Burned, he wrote an elegant piece of code for himself that implemented mechanisms to absolutely assure the accurate recovery of data. The BRU format, and subsequent BRU product, was born.

```
[root@pc-00067 /root]# tar -cvf test.tar /etc/hosts*
tar: Removing leading `/' from member names
etc/hosts
etc/hosts.allow
etc/hosts.deny
[root@pc-00067 /root]# bru -cvf test.bru /etc/hosts*
/
/etc
/etc/hosts
/etc/hosts.allow
/etc/hosts.deny
```

From its inception to this day, the development of BRU has been a classic example of "form follows function." Design focus has always centered on the accurate recovery of data, and the backup function is the "form" that supports the recovery "function." This design philosophy is what differentiates BRU from virtually all other backup approaches. Because of Fred's vision and resolve, untold users around the world can rest easy knowing their data is safely protected by BRU. The excellence of his code, continually updated and enhanced over the last 17 years, has stood the test of time. After all, it's all about the restore, not the backup.

BRU implements multiple techniques during the backup and restore operations to assure that the data returned is accurate. *The above test, using a tar format example, is but one example of the robustness of the BRU technology. Prove it yourself!*

Here is the /etc/hosts file before we tar'd or BRU'd it.

```
--snip--
127.0.0.1          localhost.localdomain  localhost
192.168.1.75      accounting
192.168.1.1       gw
209.149.147.185   serve1
66.1.56.100       bleaf
66.1.47.197       fwc
--snip--
```

```
[root@pc-00067 /root]#
```

```
[root@pc-00067 /root]# vi test.tar
```

Skip past the tar header and modify one byte or character ie. change one of the IP's in the hosts file, we changed 209.149.147.185 to 209.349.147.185 in the 'serve1' entry.

```
[root@pc-00067 /root]# vi test.bru
```

Skip past the BRU header and modify one byte or character ie. change one of the IP's in the hosts file, we changed 209.149.147.185 to 209.349.147.185 in the 'serve1' entry. **PLEASE NOTE:** we made the same change to both the test.tar and the test.bru, changing the 149 to a 349 in the '209.149.147.185 serve1' entry.

Now let's try to extract the files from the tar archive first.

```
[root@pc-00067 /root]# tar -xvf test.tar
etc/hosts
etc/hosts.allow
etc/hosts.deny
```

It extracted normally, now let's look at the file after it's been modified.

```
--snip--
127.0.0.1                localhost.localdomain localhost
192.168.1.75            accounting
192.168.1.1             gw
209.349.147.185         serve1
66.1.56.100             bleaf
66.1.47.197            fwc
--snip--
```

The 'tar' utility returned inaccurate data without notification. Now let's try to extract it with BRU and see what the result is.

```
[root@pc-00067 /root]# bru -xvrvvf test.bru
archive ID = 3c9a03a856e2
buffer size = 20k bytes
media size =
bru: [I245] "/" : skipped file, directory exists
bru: [I245] "/etc": skipped file, directory exists
bru: [W011] warning - file synchronization error; attempting recovery ...
x 8K [1] -rw-r--r-- 1 root          root          161 Jan 12 2000 292040 /etc/hosts.allow
x 10K [1] -rw-r--r-- 1 root          root          347 Jan 12 2000 292041 /etc/hosts.deny
bru: [I181] read 10 blocks (20 KBytes) on volume [1], 0:00:00, 9999 Kb/sec

**** bru: execution summary ****

Started:                Thu Mar 21 09:11:23 2002
Completed:              Thu Mar 21 09:11:23 2002
Archive id:             3c9a03a856e2
Messages:               1 warnings, 0 errors
Archive I/O:            0 blocks (0Kb) written
Archive I/O:            10 blocks (20Kb) read
Files written:          0 files (0 regular, 0 other)
Files read:             2 files (2 regular, 0 other)
Files skipped:          2 files
Write errors:           0 soft, 0 hard
Read errors:            0 soft, 0 hard
Checksum errors:        1

[root@pc-00067 /root]#
```

BRU recognizes the data has been corrupted.

Conclusion: The structure of traditional tar, and cpio archive formats as well, do not allow for the assurance of data integrity. Today's data is much too critical to leave to chance. The BRU format accounts for every bit being backed-up and restored. In the example above, BRU reports the data is corrupted. It would then advance the tape until good data was read, and continue the recovery versus aborting the entire restore