

## Impact Analysis: Bug - Re-adding Torrent Corruption

---

**Bug:** If the user removes a torrent and then adds the same torrent back, the torrent resumes as if the data was in the state before its removal. This is an issue when the user completes a torrent, removes it, then adds the same torrent with incomplete data (causing an empty file to be considered complete). This also causes an issue with substituting a incomplete torrent with completed data, causing the file to be overridden with the same data. It is inconclusive whether a false positive torrent could corrupt others issues by seeding bad data, but seeding does take place with falsely incomplete data.

**Possible Cause:** Validate the data file for its progress and resume from there instead of where last left off. Possibly caused by properties files called s\_ files. One s\_ file exists for each torrent added.

### Tests:

1. New torrent : **Download torrent and complete**
2. Complete torrent with s\_ file : **shown as complete**
3. Complete torrent without s\_ file : **shown as complete**
4. Incomplete torrent without s\_ file : **shown as incomplete**
5. Incomplete torrent with completed s\_ file : **shown as complete**
6. Complete torrent with incompleted s\_ file : **shown as incomplete and downloads**

### Estimated impact:

SnarkManager.java  
Snark.java

### Estimated impact: 2 Files

**Actual Impact:** We learned that the cause of the bug is from residual 'sP' files. These files store the state of the torrent and are not removed when the torrent is removed. By swapping in and out these files (some that say the torrent is complete, others saying it is not), we can trick I2PSnark into believing the torrent is in a false state. For example, using an empty data file with a s\_ file that says it's 128mb complete will trick I2PSnark to resume the torrent on the remaining data, ignoring the 128mb that was listed as downloaded. The solution likely includes removing the s\_ files when a torrent is removed, or always making a new s\_ file overwriting any incorrect s\_.

**Solution:** We found that the remove commands function properly in Snark.java and SnarkManager. This lead us to I2PSnarkServlet.java which calls functions based on the user's gui. When the gui action is "remove" or "delete" for a torrent, the code checks if the meta does not exist and removes the magnet-torrent and continues. This is an issue when the torrent is not a magnet file, which does get removed. We discovered this bug is present in two locations where a magnet is removed without considering the torrent if the metadata exists. We fixed this bug by adding a condition checking if there is metadata associated with the torrent and calls the removeTorrentStatus method on the SnarkManager, removing the

metadata. This solution redirects the tests provided earlier by eliminating conditions that would lead to tests 2, 5 and 6.