CS111 Qinyi Yan (UID:704406413)
Lab4 Report Xiaoxuan Wang (UID:804406399)

Part I Vulnerabilities and Fixes

1. Buffer Overrun Bug:
   The filename length can exceed the maximum allowed filename length (FILENAMESIZE-1) bytes.
   To fix this bug, simply use strcpy instead of strncpy to restrcit the number of bytes copied to the filename buffer. Also, add a '\0' at the end of the string. In addition, throw out an error message if the filename length is too long to notify the users.
2. Unlimited Large File:
   An unlimited large file may use up all our disk space.
   To fix this bug, putting a upper limit of file (for an example, 10Mbyte) to limit the file size. If the file size exceeds the limit, throw out an error message.
3. Download from other Directories
   Peers may request to download files from directory other than the current directory, which is the upload directory.
   To fix this bug, compare requested directory and the current directory, if the two differ, terminate the upload.
4. Task Buffer too Small:
   Given that there are so many peers in popular tracker, the task buffer size might be too small.
   To fix this, increase TASKBUFSIZ from 4096 to a larger number.
5. Too much Concurrent Peers:
   If too many downloads/uploads are processing in parallel, the CPU is very likely to be used up.
   To fix this, set a upper limit of number of processes, if the number of processes exceeds the limit, stop forking new child process and wait for others to finish.
6. Slow Transimssion:
   Slow peers might slow cut down the overall throughput and make other peers wait in a queue.
   To fix this, set a minimum transfer rate and compare peer's tranfer rate with it. If the transfer rate is lower than the minimum value, schedure another peer.
7. File corruption:
   The file we downloaded might be different from the one when evil peer register with the tracker.
   To fix this, compare MD5 checksum the peer tell the tracker with the checksum of the file we downloaded.

Part II Attack Approaches

1. Using filename bufferflow to overrun the buffer, when the connection is open, build a file which the size is twice as the FILENAMESIZ, in which case, without proper protection, it will overwrite other variables and may cause segmentation fault. Even worse, if we design our

file name well, we may be able to control what the peer upload to us by changing fieids in the task struct.

2. Making the file open a different file rather than intended for upload, hence, the peer who is requesting the file will get the wrong file which could also be a virus.
3. Downloader attack: Ask for a file infinitely many times. This way it is overloaded with requests and does not actually ever send a file.
4. When a peer request a file from us, send it an endless or useless file. For example, we send the peer a null device (/dev/null) as long as we are in evil mode. If the peer does properly check what we sent, the null device could be very dangerous to their system.
5. Upload attack: Uploading junk data forever to a file which will cause disk overrun.