# Homework 4

**Due:** November 18, 2016 **Points:** 100

## Questions

Remember to justify your answers where appropriate.

1. (*28 points*) Consider the first example in Section 26.4.1.

   (a) Why does the router not save time by opening a connection to the destination host before the pending connection completes its three-way handshake?

   (b) The router is protecting a target from being flooded. Is the router itself vulnerable to a flooding attack? If not, why not, and why won't the same property make the target immune? If so, does the attack on the router differ from the attack on the target?

2. (*20 points*) Does using passwords with salts make attacking a specific account more difficult than using passwords without salts? Explain why or why not.

3. (*28 points*) Most operating systems define two types of names. A *direct alias* (name or link) identifies the specific entry in a file allocation table (such as an inode), and an *indirect alias* is itself a file containing the path name of a second file. When one opens an indirect alias for certain actions (such as reading or writing), the operating system instead opens the file named in the indirect alias. Specific commands operate on the indirect alias itself (as opposed to the file it names).

   (a) Can indirect aliases ever loop; that is, can there exist a chain of indirect aliases $i_1, \ldots, i_n$ such that $i_1 = i_n$? If so, how would the system detect such loops? What should it do when one is discovered?

   (b) Can a loop with direct aliases occur?

   (c) The text points out the difference between a file name and a file descriptor. How does the introduction of indirect aliases complicate the resolution of an alias to a device number and inode?

   (d) On some systems, a direct alias cannot refer to an inode on a different device. Suppose the system were altered to allow a device number to be included in the alias, so a direct alias could refer to a file on another device. What complications might arise? Do indirect aliases, which can reference files on other devices, have the same complications?

4. (*24 points*) Consider a Multics procedure $p$ and a data segment $d$. Procedure $p$ is executing and needs to access segment $d$. Segment $d$'s access bracket is (5, 6). Assume that $d$'s access control list gives $p$ full (read, write, append, and execute) rights to $d$. In which ring(s) must $p$ execute for each of the following to happen?

   (a) $p$ can read, write to, and append to $d$.

   (b) $p$ can read $d$ but not write to or append to $d$.

   (c) $p$ cannot access $d$.

## Extra Credit

5. (*10 points*) The second example in section 16.2 asserts that UNIX file descriptors are in fact capabilities. Please explain in detail why this is true. (*Hint:* How are file descriptors used? What is in a file descriptor?)