Homework 4: Denial of Service

This homework is due **Friday, November 17** at **6 p.m.** and counts for 6% of your course grade (4% if you are a graduate student taking CMSC 33250). You will have a budget of four extensions (24-hour periods) over the course of the quarter that you can use to turn assignments in late without penalty and with no questions asked. You cannot consume partial days. Once your extensions are used up, further extensions will only be granted in extraordinary circumstances.

We encourage you to discuss the problems in general terms with other students in the class. However, the answers you turn in must be your own original work, and you are bound by the University's policy on Academic Honesty and Plagiarism. Also, please document any material discussions you had with others about this assignment (e.g., "Note: I discussed this exercise with Jane Smith").

Solutions should be submitted electronically via chisubmit in plain text format using the template found at `hw4/hw4.txt` in the upstream repository.

Concisely answer the following questions. (Limit yourself to at most 80 words per subquestion.)

1. **Client puzzles.** Denial-of-service (DoS) attacks attempt to overwhelm a server with a huge volume of requests. Researchers have proposed a defense against DoS attacks called *client puzzles*: For each request, the server sends the client a freshly generated random challenge r and a difficulty parameter n, and the client has to produce a solution s such that $\text{HMAC}_r(s)$ ends in n zero bits. Clients must present a valid solution to receive service.

   (a) What is the expected number of HMAC computations for the client to compute the solution? How many HMAC computations does it take for the server to check the solution?

   (b) Suppose a "unit of work" is equivalent to the difficulty of computing one HMAC. If an attacker enjoys an *amplification factor* of 64 (i.e., the attacker can cause the server to do 64 units of work by expending one unit of work), what should n be to negate this advantage using client puzzles?

   (c) Some denial-of-service attacks employ a large number of malicious clients to overwhelm the server. Briefly, how can the system adjust the puzzles to ensure that legitimate clients receive service during such attacks without requiring them to do excessive work solving puzzles when the system is not under attack? Hint: think about the scenario in terms of supply and demand.

2. **Distributed denial-of-service.** A popular attack tool among novice hackers recently has been the Low Orbit Ion Cannon (LOIC), which features a user-friendly GUI as well as an option to voluntarily add yourself to a botnet controlled via an IRC channel. *We do not recommend installing or using LOIC!*

   (a) LOIC is a fairly simple program. The source file at https://github.com/NewEraCracker/LOIC/blob/master/src/HTTPFlooder.cs contains the primary attack mechanism. Briefly, how does this mechanism work?

   (b) The LOIC command and control system ("Hive Mind mode") is also fairly simple. It is described in the README file at https://github.com/NewEraCracker/LOIC/blob/master/README.md. Briefly, how does this mechanism work?

   (c) Other than client puzzles, what are some things a website could do to defend itself against a LOIC Hive Mind attack? If the attack involves thousands of bots, how can the server distinguish them from legitimate clients?

   (d) Briefly, what was Operation Payback? (To answer this question, you can obviously use external sources. Please cite the sources you use.)

   (e) Who is Christopher Wayne Cooper? What was he charged with when he was indicted? (Once again, you can use and cite external sources.)

   (f) Briefly, compare and contrast LOIC Hive Mind mode to a typical botnet.

□