Homework 4: HTTPS, Denial of Service

This homework is due Tuesday, March 24 at 6 p.m. You have a budget of five late days (24-hour periods) over the course of the semester that you can use to turn assignments in late without penalty and without needing to ask for an extension. Once your late days are used up, extensions will only be granted in extraordinary circumstances.

We encourage you to discuss the problems and your general approach with other students in the class. However, the answers you turn in must be your own original work, and you must adhere to the Code of Academic Integrity. Solutions should be submitted electronically via Canvas in plain text format by completing the template at the end of this document.

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

1. HTTPS. A self-signed certificate makes the claim that a public key belongs to a particular server, without any trusted certificate authority (CA) to verify it. Browsers display a warning message when a site presents such a certificate, but users often override these warnings. Some websites use self-signed certs to avoid the trouble of obtaining a cert from a trusted CA.

   (a) Briefly explain how using HTTPS with a self-signed certificate provides protection against a passive eavesdropper.

   (b) How might a man-in-the-middle attacker compromise a site that uses a self-signed certificate, assuming that the client ignores browser certificate warnings?

   (c) Some sites use HTTPS with a certificate signed by a trusted CA for their login pages, then set a session cookie and use HTTP for the other pages on the site. Briefly compare the security of this design to the use, for all pages on the site, of (i) a self-signed certificate and (ii) a certificate signed by a trusted CA.

2. Authentication protocols. A large Ivy league university has implemented a central sign-on facility where users authenticate themselves to an official site then receive a token that confirms their identity to all other campus sites.

   (a) Assuming the protocol is competently implemented and deployed, how might deploying this service improve security on campus?

   (b) Under the same assumptions, how might it hurt security?
Suppose the sign-on protocol proceeds as follows: When the user visits site $A$, which requires authentication, site $A$ redirects the user to the central sign-on site. Following authentication, the central sign-on site redirects the user’s browser back to a standardized HTTPS URL at site $A$ with the following parameters: $u$, the user’s username, and $\text{Sign}(u)$, a digital signature produced with the sign-on site’s private key. (Assume that the corresponding public key is widely known.) The site checks that the signature is valid for $u$, and considers the user authorized if so.

(c) If site $A$ is controlled by an attacker, how can it trivially impersonate the user to other sites that trust the sign-on protocol?

(d) Propose a simple change to the protocol that would fix the problem identified in (c).

3. **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 hash 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.

4. **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/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 section at https://github.com/NewEraCracker/LOIC. 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?

(e) Who are the Paypal 14? What were they charged with when they were indicted?

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

(g) Briefly, compare and contrast LOIC Hive Mind mode to a political protest.

Submission Template

Use the template below to organize your submission. Make sure each answer is formatted as a single line, and that you submit through the Canvas text entry form. You may use LaTeX-style math syntax or Canvas’s equation editor if you wish.

# Problem 1

1a. [Answer ...]

1b. [Answer ...]

1c. [Answer ...]

# Problem 2

2a. [Answer ...]

2b. [Answer ...]

2c. [Answer ...]

2d. [Answer ...]

# Problem 3

3a. [Answer ...]

3b. [Answer ...]

3c. [Answer ...]

# Problem 4
4a. [Answer ...]
4b. [Answer ...]
4c. [Answer ...]
4d. [Answer ...]
4e. [Answer ...]
4f. [Answer ...]
4g. [Answer ...]