CS426: Computer Security **Fall 2010**

Homework #2

Due date & time: 1:30pm on September 24, 2010. Hand in at the beginning of class (preferred), or email to the TA (twykoff@purdue.edu) by the due time.

Late Policy: You have three extra days in total for all your homeworks and projects. Any portion of a day used counts as one day; that is, you have to use integer number of late days each time. If you emailed your homework to the TA by 1:30pm the day after it is due, then you have used one extra day. If you exhaust your three late days, any late homework won't be graded.

Additional Instructions: (1) The submitted homework must be typed. Using Latex is recommended, but not required.

Problem 1 (4 pts) Recall the distinction we made among users, principals, and subjects in class, and recall that a user refers to a human user of the computer system. What corresponds to a principal in a UNIX environment? What corresponds to a subject in a UNIX environment? Why we want ensure that for each principal, there is only one user corresponding to it?

Problem 2 (6 pts) Why it is necessary for some files to be setuid root (i.e., owned by root and has setuid bit set)? What security problems a setuid root program may cause? What can a developer do to alleviate the problem?

Problem 3 (10 pts) What permissions are necessary and sufficient to perform the following operations under UNIX operating systems? For example, to read the file /d1/d2/f3, one needs x on /, x on /d1, x on /d1/d2, and r on /d1/d2/f3.

1. delete the directory /d1/d2, where /d1/d2 contains one file /d1/d2/f3
2. rename a file from /d1/d2/f3 to /d1/d2/f4
3. create a hard link /d1/d2/f3, which points to /d4/f5
4. read the file /d1/d2/f3/f5, where /d1/d2/f3 is a symbolic link pointing to the directory /d4, and /d4 contains a file /d4/f5
5. delete the file /d1/d2/f3/f5, in the same setting as above

Problem 4 (10 pts) The UNIX `crypt` function is a hash function that only looks at the first eight bytes of the input message. For example, `crypt(helloworld)` returns the same value as `crypt(hellowor)`.

Some web sites use the following authentication method to authenticate users: (1) the user types in a user-id and a password P into his web browser, (2) the site, upon verification of the password P, computes $T =$ `crypt(`user-id$||K$`)`, where $||$ denotes string concatenation, and K is a ℓ-byte site secret key $\ell \leq 8$, (3) the site sends a cookie back to the user containing T, (4) the user can use T to authenticate himself to the site in future connections.

Show that by choosing clever user-id's (of varying length) an attacker can expose the site's secret key K in time approximately 128ℓ. More concretely, the user creates an account, logs in and receives the corresponding T; he then creates another account (with a different user-id, logs in and receives another T. By repeating this sufficient times, the user recovers K completely. We are assuming there are 128 possible values for each character in a string.

Hint: Try to recover one character of K for each account created. The attack is described in the paper "Dos and Don'ts of Client Authentication on the Web" in USENIX Security Symposium, 2001. Reading the paper is allowed.

Problem 5 (20 pts) Read the following paper and write a summary of what you have learned from it.

- Ross Anderson: Why Cryptosystems Fail. (Available from the handout page.)

Problem 6 (20 pts) Read "Authentication in an Internet Banking Environment" by Federal Financial Institutions Examination Council (available from the handout page).

1. Write a set of guidelines on how users use passwords for an online bank. The guidelines should cover how the passwords are chosen, used, and updated.
2. Analyze the user authentication mechanism in one online bank that you use. Try to see whether it is vulnerable to various attacks, including phishing attacks.

Problem 7 (20 pts) Read the article "Password protection for modern operating systems" at *http://www.usenix.org/publications/login/2004-06/pdfs/alexander.pdf*, and summarize the key weaknesses of password schemes in modern operating systems.

Problem 8 (10 pts) You are asked to implement a web server that requires each user to log in. You are asked to come up with two designs to store users' passwords securely. You can use a cryptographic hash function h, a symmetric cipher E, and a message authentication code function C.

- **(a)** One design is to store the passwords on the website. However, as the file that stores these passwords may be leaked, we want to make dictionary attacks on the password file very difficult. What will you store in the file? And how would you authenticate a user? (Give precise description using mathematical formula.)
- **(b)** The other design is not to store the password on the server. When a user creates an account, the account number is stored on the server and the user's password is stored in a cookie on the user's machine.

  Identify the possible attacks in this scenario. To prevent various attacks, what should the cookie contain? And how would you authenticate a user? (Give precise description using mathematical formula.)