3 problems. 40 points. 25 minutes Closed book. Closed notes. No electronic device. Write your name above.

1. [10 points]

A website maintains an SQL table YY with a row for every user and columns NAME, PWD, AGE and others.

- To change its password, a user sends a GET request with path /chpw.php?a1=<name>&a2=<opwd>&a3=<npwd>.
  The server looks for an entry in YY with NAME =<name> and PWD =<opwd>; if found, it sets PWD field to <npwd>.
- To get a user's age, a user sends a GET request with path
  /getage.php?a1=<name>.
  The server looks for an entry in YY with NAME =<name>; if found, it returns the entry's AGE field.

The server does no additional checks on these operations.

Among the users are Ted and Bob (these are their NAME entries). Ted does not know Bob's PWD value.

Give the path of a GET request that Ted can issue in order to change Bob's password to fqr123.

Solution

Assume

```
chpw.php(name,opwd,npwd):
  UPDATE YY
  SET PWD="npwd" WHERE NAME="name" AND PWD="opwd"

getage.php(name):
  SELECT AGE FROM YY WHERE NAME="name"
```

Possible attack paths

- /chpw.php?a1=bob"/*&opwd=*/-&npwd=fqr123
  // SQL: UPDATE YY SET PWD="fqr123" WHERE NAME="bob"- ...
- getage.php?a1=bob;UPDATE+YY+SET+PWD="fqr123"+WHERE+NAME="Bob"
  // SQL: UPDATE YY SET PWD="fqr123" WHERE NAME="bob"

End of solution

2. [10 points]

This problem concerns a browser c1, website s1, and attacker website s2.

- c1 clicks http://s1/p1.html. In the response, s1 sets a cookie for domain s1.
- Then c1 clicks http://s2/p2.html.
- Then p1.html regularly issues POST requests to s1. Each POST request contains the cookie value in its data. The server treats a request as valid iff the cookie value (in the request header) matches the value in the data.

For each of the following cases, answer whether p2.html can send a POST request to s1 that the latter treats as valid. Write "YES" if it can, and "NO" if it cannot. (Below, "unguessable" is equivalent to "randomly generated".)

s1-cookie name	**s1-cookie value**	**your answer**	
guessable	guessable	YES	**[2.5 pts]**
guessable	unguessable	NO	**[2.5 pts]**
unguessable	guessable	YES	**[2.5 pts]**
unguessable	unguessable	NO	**[2.5 pts]**

If s2 knows the cookie value, p2.html can have a form element that posts to s1 with the cookie value in its data. Client c1 includes the cookie in the header.

3. [20 points]

This problem concerns a browser c1, a website s1, and an attacker website s2 that can *also* eavesdrop, intercept and send messages on the network link between c1 and s1. (Note: both https and http are used.)

- c1 clicks https://s1/p1.html. In the response, s1 sets a *secure* cookie for domain s1.
- Then c1 clicks http://s2/p2.html.
- Then p1.html regularly issues POST requests to https://s1 and GET requests to http://s1.
  Each POST request contains the cookie value in its data.
  s1 accepts a POST request iff the value of the cookie (in the request header) matches the value in the data.

Part a Suppose s1-cookie has a guessable name and an unguessable value. Can p2.html send a POST request to s1 that the latter treats as valid. If yes, give the steps of the attack. If no, explain *briefly*.

Solution

Yes, an attack is doable.

Let scname denote the cookie's name; s2 knows this because it's guessable.

- s2 eavesdrops and waits for c1 to send an http://s1 request (which is in an unencrypted TCP message).
- s2 intercepts this http request and sends (in a TCP message) an http response to c1 that sets cookie scname to a value, say scval (using response header Set-Cookie: scname=scval).
- c1 receives the http response and sets scname's value to scval (even though it received scval the Set-cookie header over http).
- p2.html learns of scval; eg, it periodically queries s2.
- p2.html can now use a form to construct a POST request to https://s1 with scval in its data. c1 will attach the cookie in the header.

Grading

5 pts max: if you say s2 can obtain cookie name/value by eavesdropping.

7 pts: if you say s2 cannot obtain cookie name/value by eavesdropping because it is only sent via https.

Lose points for incorrect/irrelevant claims.

End of solution

Part b Same as part a but now suppose s1-cookie has an unguessable name and a guessable value.

Solution

Yes, an attack is doable.

Let scval denote the cookie's value; s2 knows this because it is guessable.

p2.html can use a form to construct a POST request to https://s1 with scval in its data. c1 will attach the cookie in the header. (Note that scname is not needed in the data.)

Grading

5 pts max: if you say s2 can obtain cookie name/value by eavesdropping.

Lose points for incorrect/irrelevant claims.

End of solution