Q 1. Assume that you logged in to your account at bbmrocks.com web site in your browser, which uses a cookie to manage sessions. At the same time, you got an e-mail which says that you should connect to a link to update some information in your account of bbmrocks.com site. When you click to this link, what kind of attack(s) can be performed on your account? How does the attack(s) work? What can the attacker do with this attack?

Q 2. In a UNIX system, a server program “bomb” is running with setuid bit set by "root". This program maintains a configuration file for each user under the home directory of the user, which is named as “bomb.config”. When the user runs the program, this configuration file is read and some operations are done by the program. If there is an error in the file, a message showing the error lines of the file is displayed on the screen. Also, a temporary file “bomb.tmp” is created under the user’s directory. if the temporary file already exists it’s truncated. A summary of the operations are given in the below box. Which attack(s) can be carried out on this program? What can be obtained by attack? How can we prevent it?

```
read “$USER_HOME/bomb.config”

create/truncate “$USER_HOME/bomb.tmp”

/* do some other initialization tasks */

open the main window to the user
```

Q 3. A server software is accepting HTTP URLs as inputs and do some search operations on the page specified with the user input. However, this software only accepts URLs from “edu.tr” domain. If the user inputs the below URL, should the server accept or deny it? Why? Explain your answer to get full credit.

http://www.metu.edu.tr@0xC1.140.0330.0x2B

Q 4. The function on the below box concatenates two strings (buf1, buf2) on "mybuf" string and do some operations with this string. What kind of attacks can be performed on this code? How can you prevent this attack? Explain attack details. (Note: `memcpy(mybuf,buf1,len1)` function copies `len1` number of characters from `buf1` to `mybuf`).

```
int catstr(char *buf1, char *buf2, char len1, char len2){
   char mybuf[256];

   if((len1 + len2) > 256)
     return -1;

   memcpy(mybuf, buf1, len1);
   memcpy(mybuf + len1, buf2, len2);

   do_some_stuff(mybuf);

   return 0;
}
```

Q 5. Assume that there is a web page like below, which contains some PHP code. On the URL string of this page, `filename` parameter is provided. The file whose name is specified with `filename` parameter is backed up on the UNIX machine with the "backup" command. Which attack(s) can be performed on this page? Explain the attack(s) with example. Also explain that how we can prevent the attack(s).

```
<HTML>
<head><title>Hack World</title></head>
<body>
.....
<?php
   $file=$_GET["filename"];
   print "The file to be backed up: $file";
   print "<br>";
   system("backup $file");
   print "The file is backed up!";
?>
.....
</body>
</HTML>
```

Q 6.

```
$query = "UPDATE customers SET city='" +
escape($_GET["city"]) + "' WHERE customerID='" +
$customerID + "' AND purchaseID='" + $purchaseID + "'"";
```

```
          TABLE customers (
            customerID VARCHAR(20),
            purchaseID VARCHAR(20),
            date varchar(20),
            amount varchar(20),
            city varchar(20)
          );
```

A PHP page's URL is requested like this "`update.php?city=Ankara`". The "`city`" parameter sent to this PHP page is used to construct the SQL query string shown in the top box and then this query is executed as a PHP code. The "`city`" parameter is escaped for special characters by using the escape function. "`customerID`" and "`purchaseID`" parameters come from the web session. Definition of CUSTOMERS table is given in the second box.

However, the escape function for the "`city`" parameter is forgotten in the below query. In this query, "`customerID`" parameter comes from the web session. "`city`" parameter is selected by the user from a dropdown box on the web page, where the values are loaded from the database. How does an attacker perform **second-order SQL injection** on this page to change the amount of a customer's all purchases to 1 TL ? Explain attack details.

```
$query = "SELECT purchaseID, date, amount FROM customers
WHERE city='" + $city + "' AND customerID='" + $customerID
+ "'";
```

Q 7. What changes have been made in IIS 5.0 to prevent directory traversal attacks? What should be considered when implementing these changes? Please explain.

Q 8. Assume that you are viewing a blog web site in your browser, which might contain malicious scripts. At the same time, you logged in to your account at `hubbm.com` web site in another tab of your browser. `hubbm.com` web site does not use cookies and embed session IDs as a parameter to every generated HTML page. Is it possible that the blog page can launch a CSRF attack on `hubbm.com` web page? Explain your reasoning in your answer (How does the attack work or not?).