UNIVERSITY OF BIRMINGHAM

School of Computer Science

Security and Networks

Second Class Test May 2021

Security and Networks

Question 1

You review a PHP application that interacts with an MySQL database server.

(a) You come across the following piece of code, where `pw` contains the password as entered by the user in a web form and the variable `score` contains an integer, and is set based on calculations done in the PHP script.

```
 1  $scoreDir= "/var/www/scores";
 2  $user = $_GET["user"];
 3  $pw = $_GET["pw"];
 4  $result = mysqli_query("SELECT * FROM users WHERE user = '$user'
 5                          AND password = '$pw'");
 6  if ($mysqli_num_rows($result) > 0) {
 7    // Logic successful
 8    // calculate score and store result in $score
 9    system ("echo $score > $scoreDir/$user.txt");
10    echo "New score $score added";
11  }
```

Which vulnerabilities are present in this code? Explain how these vulnerabilities can be exploited.

[7 marks]

(b) Explain why the way the passwords are stored in the database is very bad practice, and describe a way to securely store the passwords in the database. **[6 marks]**

(c) You inspect the server and find that some security-critical settings have been disabled through the admin web interface. There is a well-known URL that could lead to this if invoked by a user with admin rights. The administrator recalls that they have visited some potentially malicious websites, but are pretty sure that no actual malware has been installed. Is there still a way that these modifications could have been made by the malicious website? Explain your answer. **[7 marks]**

Question 2

(a) Modern operating systems have mechanisms to limit the effect of buffer overflows in binaries running on the system, without requiring recompilation or other changes. Explain the operation of two such mechanisms. As well as describing each independently, you should explain how your two mechanisms work together.

Note: a "stack canary" requires recompilation.

[8 marks]

(b) Consider this piece of code:

```
1        void admin_command (char *password) {
2            char command[50];
3            int privileged = 0;
4
5            /* check the password is correct */
6            privileged = check_password (password);
7            /* load the command */
8            gets (command);
9
10           if (privileged) {
11              if (strncmp (command, "reboot", 6) == 0) {
12                 reboot ();
13              }
14              /* other privileged commands here */
15           }
16       }
```

Assuming that the machine has the protections you described in the previous part, explain how this piece of code might permit a user who does not know the password to reboot the machine. Explain how the attack would be carried out. Indicate two places where you would repair the code to remove the vulnerability and guard against its later recurrence. **[6 marks]**

(c) Referring again to your answer in part (a), do the mechanisms you have described guard against this attack? Explain your answers. **[6 marks]**