Jaypee Institute of Information Technology
Test-2 Examination 2020-21

Course Name: Web Technology & Cyber Security **Course Code: 21B12CS315**
Max. Time: 1 Hr **Max. Marks: 20**

Q1 to Q10 (1 * 10 = 10 Marks)

[C331-4.2] Q1. Illustrate the difference between setTimeout and setInterval.

Ans 1:
setTimeout – Used to schedule code execution after a designated amount of milliseconds
setInterval – Used to execute a block of code multiple times, starting after the given interval of time, then repeating continuously at that interval.

```
setTimeout(() => alert('Hello'), 3000);    // Will alert 'Hello', once after 3 second.

setInterval(() => alert('Hello'), 3000);    // Will alert 'Hello' repeatedly after every 3 seconds.
```

[C331-4.2] Q2. Interpret and present the output of the following code snippet:

```
console.log("first");
setImmediate( () => { console.log("second");});
console.log("third");
```

Ans 2:

first
third
second

[C331-4.3] Q3. After creating an app in a Django project, how do we get Django to notice it?

Ans 3: Add the name of the created app to the INSTALLED_APPS variable in setting.py file.

[C331-4.5] Q4. Explain how to achieve confidentiality with block cipher.

Ans 4: Confidentiality is achieved with encryption.

The encryption algorithm: Perform the XOR operation on the plaintext and a block of bits that is the same size.

The decryption algorithm: Perform the XOR operation on the cipher text and the same block of bits used in the encryption process.

For both encryption and decryption, symmetric key will be used.

[C331-4.4] Q5. Mr. Ramu is a network security administrator, who has identified that the company server is running slow all of a sudden. He also observed that the requests

made by the network computers are unable to serve. Identify what causes this problem and list other targets that may get affected due to this problem?

Ans 5: The problem has appeared due to Denial of Service Attack.

DoS attacks result in a multitude of consequences. It may affect

a. Web servers – A successful DoS attack and subsequent compromise of a web server constitutes the widest public exposure against specific target.
b. Back end resources – DoS attack take down the back end resources such as customer database or server farm essentially render all front-end resources unavailable.
c. Network or computer specific resources – DoS attacks are also launched from within a local area network with intent to compromise the network itself or to compromise a specific node such as a server or client system..

[C331-4.4] Q6. Discuss ping of death with related commands.

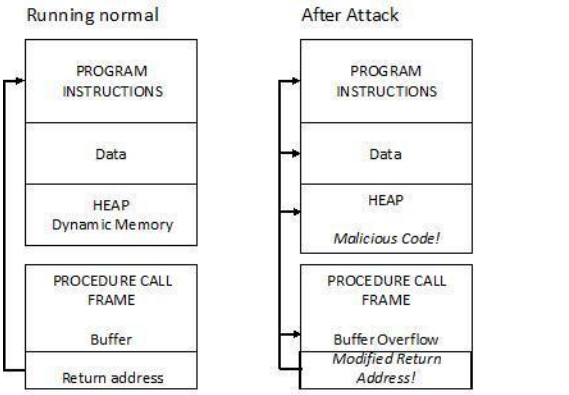
Ans 6: Ping of death is a ping packet that was larger than the allowable 64K. Although not much of a significant threat today due to ping blocking, OS patching, and general awareness, this method is extremely easy to use DoS exploit. To perform ping of death the following command is used

ping –l 65540 <hostname or IP address>

[C331-4.4] Q7. Discuss the reason for buffer overflow DoS attack.

Ans 7: Absence of Bound checking is the root cause of buffer overflow attack.

Ex: functions such as gets(), scanf(), strcpy(), strcat() in C programming.



[C331-4.3] Q8. Build a jQuery code to set a 2px red border of all table elements, and 2px green border of all <h2> headings, on a button click event.

Ans 8:

```
$("#que").click(function() {
        $("table").css("border", "2px solid red" )
        $( "h2" ).css( "border", "2px solid green" );

});
```

[C331-4.3] Q9. Consider the CSS given below. Using jQuery, add the class "*p_font_color*" to the first paragraph element.

```
p { margin: 8px; font-size: 16px; }
.p_font_color{color: red; }
.w3r_background { background: yellow; }
```

Ans 9: $("p").first().addClass("p_font_color");

[C331-4.3] Q10. Using Bootstrap, develop logic to create navigation bar with two menu items "Academics" and "Department". The menu item "Department" will be a dropdown with menu items "Department 1" and "Department 2".

Ans 10:

```
<nav class="navbar navbar-expand-sm bg-light">
  <div class="container-fluid">
    <ul class="nav navbar-nav">
      <li class="nav-item"><a  href="#">Academics</a></li>
      <li class="dropdown">
        <a class="dropdown-toggle"  data-toggle="dropdown" href="#">Department
        <span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="#">Department 1</a></li>
          <li><a href="#">Department 2</a></li>
        </ul>
      </li>
    </ul>
  </div>
</nav>
```

[C331-4.2] Q11. Assume, there is a form with two fields, *username* and *password*, and an *update* button. Whenever a user enters data into the provided fields and clicks the button, a request is sent to the server, and a file corresponding to each username either gets created or updated (if it already exists). The naming convention for the file is *username.txt*. Username and password values are passed in the url as query parameters.

Write the code for creating a web server in node.js that handles/ performs the following tasks:

a) **If the file already exists then the provided password should be appended to the existing file, and the file should be renamed as *username_updated.txt*.**

b) **If the file does not exist then a new file should get created, and the password should be written in that file.**

Marks: 3 (1.5 + 1.5)

Ans 11:

```
var http = require('http');
var url = require('url');
```

```
var fs = require('fs');

http.createServer(function (req, res) {
   var q = new url(req.url);
   const uname = q.searchParams.get('uname')
   const pwd = q.searchParams.get('pwd')
   var filename = uname + '.txt'

   fs.access(path, fs.F_OK, (err) => {
      if (err) {
         // file does not exist
         fs.writeFile(filename, pwd, function (err) {
            if (err) {
               console.log(err);
               return res.end();
            }
         });
      }

      // file exists
      fs.appendFile(filename, pwd, function (err) {
         if (err) {
            console.log(err);
            return res.end();
         }

         var newfilename = uname + '_updated.txt'
         fs.rename(filename, newfilename, function (err) {
            if (err) console.log(err);
         });
      });
   });
}).listen(8080);
```

[C331-4.2] Q12. Assume you are a startup which offers car rental service at a cheap cost across five cities. You have created a web application to handle all the bookings. In the first month of business only 20 customers avail your services, and your booking rate is around 5 bookings per hour. So, initially when you start, you save all the data corresponding to the users, cars, locations, booking history, in a single database and on a single machine. But thanks to the ads and promotions, soon a lot of people start signing up for your service and the booking rate increases to 50 bookings per hour.

Now with the increased load, your web app's performance has started to degrade, like response time has increased, availability has gone down, some transactions deadlock or fail, all leading to customer dissatisfaction. To this end, as a designer outline a mechanism to improve the performance of the app.

Marks: 2

Ans 12: Both horizontal and vertical scaling should be used for improving the app's performance.

Horizontal Scaling – Increasing the number of machine instances

Using the technique of sharding, the single database (db) server can be split into multiple localized db servers. Each localized db servers will have the same schema but will hold datafor one location/ city. Further, each localized db server can have their own replicas (cloning), which can be of used for increasing the app's availability. Scaling the app horizontally, will also assist in balancing the load on each server. Load balancers can be engaged to distribute the load evenly across the servers, so that no one server is heavily loaded and all servers can efficiently service the incoming requests without going into a deadlock or wait situation.

Vertical Scaling –Enhancing a single machines performance

In addition to localizing the db server, vertical scaling techniques can also be used to further enhance the app's performance. Upgrading the system hardware, like upgrading the RAM, or upgrading the disk space, or increasing the processing power can further decrease the app's response time.

[C331-4.5] Q13. Mr. Anil used public key cryptography with key pairs (Pk, Sk) to send a file securely to Mr. Bhuvi. This file consisting of a message of 100 characters (mixture of numbers and characters). Once the file has been received, Mr. Bhuvi wants to check its integrity. Mr. Anil used a hash function H(x) that works as follows: first it converts the characters to ASCII values, and the sum of these values are modulo with 153. In this connection, explain the following:

a) **Satisfaction of hashing properties by the hash function H(x)**
b) **Making Mr. Anil to not deny the sending of message in future**

Marks: 2.5 (1.5 + 1)

Ans 13:

a) The algorithm meets the first property (**pre-image resistenace/ one wayness**). It is not possible to find the original numbers if the digest is given, For example, if we know the digest is 76, we cannot find the original ten numbers, and they can be any set of 10 numbers. The algorithm does not meet the second criteria (**second pre-image resistance/ weak collision**). If the digest is given, we can create 10 numbers that has to the same digest. For example, without knowing the original set of number, if Mr. Eve can intercept the digest of 151, create the set, and send it with the digest 151 to Mr. Bhuvi. Mr. Bhuvi is fooled and believes that the set is authentic. The algorithm does not meet the third criteria (**collision resistance/ strong collision**). If the digest is given, we can create at least two sets of 10 numbers that hash to the same digest.

b) To make Mr. Anil to be accountable, it is recommended to use signature using Public Keys provided. Let (Pk, Sk) are the key pairs, the signature will be created as

E(H(file), Sk) such that anyone knowing Pk of Mr. Anil can verify the signature. This is also called as non-repudiation property of information security.

[C331-4.2] Q14. Consider you have an array of social media user's objects which contains unique id, name, list of friends, date of birth and phone number. Demonstrate with suitable code how a REST API can be used to

a) **Update the date of birth and phone number of a user**
b) **Add a new user**
c) **Remove a user**

Marks: 3 (1 + 1+ 1)

Ans 14:

```
const express=require('express');
const app=express();

app.use(express.json()); \
const users=[
   {id:1, name:'Krishna', friend:['f1','f2'],dob:'01-01-1990',phone:9789},
   {id:2, name:'Shyam',friend:['f1','f2'],dob:'05-03-1992',phone:9899},
   {id:3, name:'Vasu',friend:['f1','f2'],dob:'27-06-1990',phone:9076}
]

/// add new user
app.post('/api/users, (req,res)=>{
        const usr={id: users.length+1,
                name: req.body.name,
                friend:req.body.friend,
                dob:req.body.dob,
                phone:req.body.phone
        };
        users.push(usr);
        res.send(usr);
});

// update date of birth and phone number
app.put('/api/users/:id', (req, res)=>{
        const usr=users.find(c => c.id === parseInt(req.params.id));
         if(!usr) return res.status(404).send(User not found');
        usr.dob=req.body.dob;
        usr.phone=req.body.phone;
        res.send(usr);
});

/// Handling delete request
app.delete('/api/users/:id', (req, res)=>{
```

```
    const usr=users.find(c => c.id === parseInt(req.params.id));
    if(!usr) return res.status(404).send('User not found');
    const index=users.indexOf(usr);
    users.splice(index, 1);
    res.send(usr);
});

app.listen(3000, ()=> console.log('listening....'));
```