Examen IIC2513 2021-1

SECCIÓN: Fundamentos

Regla para TODAS las preguntas:
Cada pregunta correcta equivale a un punto.

P1.- Cuando vemos en un módulo de React lo siguiente:

```
function App() {
  return (
    <div>
       <h1>Este es un título/h1>
      </div> )
};
```

Podemos afirmar que en el caso de React, permite trabajar con Javascript y HTML incrustado. **Se pide que comente la aseveración anterior**

La afirmación es imprecisa, en realidad lo que ocurre es que React, en este caso, está usando JSX el cual tiene una sintaxis parecida a la de HTML pero sólo para mejorar la legibilidad. Eso que sale en el código NO es HTML.

P2.- Comente cuál es la salida de este código (en el browser) y **justifique su respuesta**.

```
import React, { useState, useEffect } from "react";

var timeNow = new Date().toLocaleTimeString();

const [theTime, setTime] = useState(timeNow);

function App() {
  setInterval(setUpTime, 1000);

  function setUpTime() {
    const newTime = new Date().toLocaleTimeString();
    setTime(newTime);
  }

  return (
    <div>
      <h2>Time: {theTime} </h2>
    </div>
  );
```

```
}

export default App;
```

La salida será un error. ***[0,2 puntos por indicar la salida]***
El error se produce al tener const [theTime, setTime] = useState(timeNow); fuera de la función que usará el hook useState. El contexto y alcance (scope) de useState está dado por la función que lo contiene. ***[0,8 puntos por justificación correcta]***

P3. Comente cuál es la salida de este código (en el browser) y **justifique su respuesta**. (asuma que en el HTML que usa React hay un elemento de id “root”)

```
import React from "react";
import ReactDOM from "react-dom";

ReactDOM.render(
     <h1>El título para la página</h1>
     <h2>Un header tipo 2..</h2>,
     document.getElementById("root")
);
```

La salida será un error. ***[0,2 puntos por indicar la salida]***
*React no permite renderizar dos elementos individuales de HTML, si se quiere tener más de un elemento HTML, estos se deben encerrar (o envolver) en algún contenedor como <form> o * ***[0,8 puntos por justificación correcta]***

P4. Comente cuál es la estructura HTML que se crea y renderiza en este trozo de código particular. **Justifique su respuesta**. (asuma que en el HTML que usa React hay un elemento de id “root”)

```
import React from "react";
import ReactDOM from "react-dom";

const rootElement = document.getElementById("root");

ReactDOM.render(
React.createElement('div',null,React.createElement('h1',nu
ll,'Hola Mundo')),
  rootElement
);
```

la salida renderizada en HTML será
* *

<h1>hola Mundo</h1>
* * ***[0,5 puntos por HTML bien formado]***

Esto se debe a que se han creado dos elementos HTML (usando la API de react) anidados. ***[0,5 puntos por justificación correcta]***

P5.- En una página web cualquiera, existe la siguiente url:

http://www.cualquierPagina.com?id=317

Un hacker hace lo siguiente:

http://www.cualquierPagina.com?id=317 AND 1=1

La página responde adecuadamente con los recursos de acuerdo al ID entregado.

a.- ¿Qué tipo de ataque se acaba de realizar? ***[0,3 puntos]***
SQL Injection
b.- ¿Qué puede inferir el atacante al recibir una respuesta OK al agregar AND 1 = 1 ? ***[0,7 puntos]***
Al recibir un OK, el atacante puede inferir que el código no está aplicando medidas preventivas ante un ataque de SQL injection y puede proceder a realizar ataques más sofisticados de SQL injection para hacerse de información desde la Base de datos.

P6.- Tenemos una página web que actúa de la siguiente forma:

http://www.miPagina.com/recurso?q=Fotos

La cual retorna lo siguiente:
Me preguntaste por ‘Fotos’

Es decir, se incluye en la respuesta, el parámetro de búsqueda (en este caso, “Fotos”)

Nuestro hacker favorito envía un correo masivo que incluye el siguiente link:

http://www.miPagina.com/recurso?q=Fotos+%3Cscript%3algo_malvado()%3C/script%3E

a.- Si una persona cae en esta página y busca “Fotos”, ¿qué estará ejecutando? ***[0,3 puntos]***
En este caso estará ejecutando un código de la forma:
http://www.miPagina.com/recurso?q=Fotos<script>algo_malvado()</script>
Lo que equivale a ejecutar algo_malvado() como script “silencioso”

b.- ¿Qué tipo de ataque acaba de ocurrir? ***[0,2 puntos]***
XSS

c.- ¿Cómo se previene este tipo de ataque? (mencione al menos una estrategia válida) ***[0,5 puntos]***

Hay varias estrategias que se pueden aplicar para evitar este tipo de ataques, se mencionan tres, pueden haber otras:

- *Filtrar las entradas en el momento en que se recibe la entrada del usuario, en general los filtros se hacen en función de lo que se espera que sea una entrada válida.*
- *Utilizar cabeceras de response HTTP adecuadas. Se puede utilizar las cabeceras Content-Type y X-Content-Type-Options para asegurarse de que los navegadores interpretan las respuestas de la forma que se espera.*
- *Generar listado de caracteres prohibidos (una lista negra) que filtre, evite o rechace ciertos caracteres y también mezclar lo anterior con un listado de caracteres permitidos*