Profesor

Miguel Salazar

Alumno

Fernando Martinez Bautista

Matricula

1702824

Riesgos/aspectos de seguridad con html y/o javascript

Existen muchos riesgos al utilizar html o javascript debido a que los sitios como de comercio electrónico, servicios, bancos e incluso redes sociales contienen información sensible la cual fácilmente puede ser obtenida si algún programador distraído no hace uso de las buenas prácticas para cuidar esta información.

Ataques de url de tipo semántico

En el uso de formularios se debe de contemplar el uso de validaciones para evitar ya que el mal uso de métodos como GET hacen que una contraseña se adhieran directamente a la url por lo que seria fácilmente detectado por el atacante.

Cross site scripting

Es típicamente encontrada en aplicaciones web que permiten la inyección de código por usuarios maliciosos en paginas web. Los atacantes se valen del código html y de scripts ejecutados en el cliente

Peticiones http falsificadas

Para estos se emplean herramientas de línea de comando o plugins agregados a los navegadores, con estos se pone a la escucha de los servicios web que típicamente se conectan a través del puerto 80.

Una fortaleza del sistema implementado será su capacidad para detectar que peticiones deberá recibir y ser escuchadas de acuerdo a los parámetros y valores que se vayan a ejecutar

Exposición de credenciales de acceso

La intercepción no autorizada de esta información podría comprometer los servidores de bases de datos o gestores de contenidos en donde se aloja nuestra información o la del cliente

Esto se evitaría si configuramos el servidor web para rechazar las peticiones de recursos que no deben ser accesibles.

SQL Injection

Para que existan este tipo de vulnerabilidades debe de haber dos factores;

1. Fallas en el filtrado de datos
2. Fallas en el escapado de datos al enviarlos a la base de datos

```
Sentencias del tipo:

select * from usuarios where id=$id;

Donde no se filtre correctamente la variable $id la entrada
podría ser algo parecido a $id = '10 or 1=1;--'


Dando como resultado:

select * from usuarios where id=10 or 1=1;--;

Podemos observar que el código inyectado nos devolvería toda
la información al ser verdadera la condición inyectada.
```

Sniffing

Cuando un atacante tiene los medio para analizar el trafico entre los usuarios y el servidor de la aplicación debemos preocuparnos por la exposición que pueden tener los datos en el trayecto, sobre todo si es tu usuario y contraseña.

Mitigar este tipo de problemas es sencillo gracias al uso https para poder cifrar el canl de comunicación por el que enviaremos nuestra información

Referencias

- https://www.owasp.org/index.php/Top_10_2013-A3-Cross-Site_Scripting_(XSS)
- https://www.seguridad.unam.mx/content/aspectos-b%C3%A1sicos-de-la-seguridad-en-aplicaciones-web