Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Lic. En Seguridad de Tecnologías de la Información

Riesgos y aspectos de seguridad en HTML y JavaScript

Alumno: Roberto Iván Hernández Chavarría

Matricula: 1452359

Materia: Diseño Orientado a Objetos

Profesor: Lic. Miguel Ángel Salazar Santillán

Riesgos de HTML

HTML tiene diversos riesgos que se han ido corrigiendo a los largo de actualizaciones en sus versiones, pero algunos siguen hasta nuestros días, volviéndose cada vez más audaces e ingeniosos en la forma de atacar, a continuación se mostraron algunos ejemplos de ataques.

- SQL injection

Los ataques SQL Injection explotan una vulnerabilidad en la validación de las entradas a la base de datos de una aplicación. Una inyección SQL consiste en inyectar un código SQL invasor dentro de otro código SQL para alterar su funcionamiento normal haciendo que se ejecute el código invasor en la base de datos.

La mayoría de aplicaciones webs consisten en un conjunto de operaciones con la base de datos, por ejemplo, en un foro para crear nuevos comentarios se insertan registros en la base de datos y para listar las entradas se seleccionan registros en la base de datos. Estas operaciones reciben el nombre de CRUD (Create, Retrieve, Update y Delete). Las aplicaciones son un conjunto de módulos separados que interaccionan entre sí a través de los enlaces y los formularios. Por lo tanto es normal que el desarrollador enlace las distintas páginas utilizando variables que después utilizará para hacer las consultas pertinentes a la base de datos.

- Ataques XSS

Los ataques XSS, Cross-Site Scripting están basados en la explotación de una vulnerabilidad del sistema de validación HTML. Consisten en inyectar código JavaScript en el interior de un documento HTML, por esto a veces se le conoce con el nombre de HTML Injection.

Esto es posible debido a que no se filtran correctamente los datos de entrada que se pasan a la aplicación. Como dije en el anterior artículo dedicado a los ataques RFI, la principal medida de seguridad a tener en cuenta cuando desarrollamos una aplicación web es nunca confiar en los datos que puedan modificar los usuarios, aquellos que se pasan por la URL, e incluso los que se pasan mediante formularios.

Existen diversos tipos de XSS

-XSS indirecto (se le conoce como reflejado): Este tipo de XSS es más frecuente encontrarlo en las aplicaciones web. El código JavaScript que se incrusta en el documento HTML no es persistente en el tiempo, es decir, solo afecta al usuario que utiliza el navegador.

–XSS directo (se le conoce como persistente): Este tipo de XSS es persistente a la aplicación ya que el código a ejecutar se queda en el repositorio de datos, ya sea base de datos, o ficheros. En este caso no hace falta enviar a la víctima un enlace manipulado ya que todos los usuarios de la aplicación se convierten en víctima afectadas.

Suele encontrarse en aplicaciones donde se permitan subir datos, por ejemplo, en un libro de visitas, en forma de comentarios.

Ya sea en uno de los dos tipos diferentes de ataques XSS se pueden ejecutar scripts entre las etiquetas <script> </script>, aunque también puede ejecutarse código JavaScript dentro de otras etiquetas HTML.

- Ataques RFI

Los ataques RFI, Remote File Inclusion, permite incluir archivos remotos desde otro servidor debido a la utilización errónea de las funciones para incluir archivos en PHP, como require(), include(), include_once(), require_once(). Una de las reglas básicas para mejorar la seguridad en las aplicaciones web es nunca confiar en los datos que pueda introducir el usuario. Así que es recomendable filtrar todas las variables de los formularios, tanto las que recibimos por el método de envío GET, como las recibidas por el método de envío POST.

Riesgos de usar JavaScript

Uno de los principales riesgos de usar JavaScript, es el plagio de código, ya que como se ejecuta desde una página web, es posible ver el código web, por lo que cualquier persona con conocimientos puede copiar el código, modificarlo un poco, o nada, y asegurar que él es el desarrollador, por lo que se sugiere, aunque no sea una buena práctica de la programación, realizar el código de una forma poco entendible, ya que esto dificultará a los plagiarios realizar este acto, pero si mantener algún archivo que nos indique de lo que estemos hablando al programar, esto para que nuestros colaboradores puedan entender nuestro código con facilidad.

Otra de las vulnerabilidades, son los ataques cross-site, ya que por medio de este tipo de ataques, se puede ejecutar código malicioso en una aplicación web, y afectar el rendimiento y seguridad de la página web, así como de las bases de datos a las cuales se pudiera tener acceso.

Aunado a lo mencionado anteriormente, se ha demostrado en base a diversos artículos de investigación, que hasta un tercio de todas las páginas web, utilizan librerías obsoletas, lo que aumenta el riesgo de ser atacadas usando una vulnerabilidad de estas bibliotecas.

Uno como desarrollador, debe de considerar este tipo de vulnerabilidades, y si somos administradores de páginas web, tratar de siempre realizar actualizaciones de las bases de datos y librerías que usemos, así como siempre tener un respaldo general de nuestro código.

Referencias:

https://desarrolloweb.com/articulos/principales-vulnerabilidades-web.html

https://books.google.com.mx/books?id=GSuZBgAAQBAJ&pg=PA44&lpg=PA44&dq=vulnerabilidades+html&source=bl&ots=10-yUFRahW&sig=1YbQkRxOB_upLWClvorHNuZaO3k&hl=es-419&sa=X&ved=0ahUKEwjasciv1oXWAhWD64MKHf7YCl8Q6AEITjAG#v=onepage&q=vulnerabilidades%20html&f=false

https://www.redeszone.net/2017/03/10/librerias-javascript-vulnerables/

http://www.catrian.com/problemas-mas-graves-que-posee-javascript/