Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Diseño Orientado a Objetos

Nombre: Jesús Manuel Muñoz Escobedo

Matrícula: 1723308

Aula: 413

Riesgos de JavaScript

JavaScript es una de las tecnologías que más se habla en la actualidad y se está utilizando en todas partes desde el navegador web de escritorio para teléfonos móviles. El lenguaje de programación se ha contenido dinámico a un nuevo nivel. Sin embargo, existen algunos riesgos asociados con el uso de JavaScript para cualquier trabajo de desarrollo graves.

Velocidad

Y las cuestiones relacionadas con la velocidad han estado plagando JavaScript desde sus primeros días. La situación ha mejorado significativamente, pero la velocidad sigue siendo un grave problema para ciertos dominios y plataformas. Esto es particularmente cierto para los juegos. Su nuevo y estremecedor juego puede funcionar de maravilla en su pc de escritorio de doble núcleo, pero trate de cargarlo en su iPhone o dispositivo Android. Es probable que las animaciones que has trabajado tan duro están muy por debajo de los 30 fotogramas por segundo que necesita para tener una buena experiencia para los usuarios.

Diferencias motor

No hay un motor de JavaScript. Google, Apple y otras organizaciones tienen sus motores preferidos. Son similares, pero no idénticas, y no puede haber diferencias de rendimiento. Esto se nota especialmente en los dispositivos móviles que Apple y google están encerrados en una lucha para producir el motor más rápido y menos intensivo de la batería.

Plagio

Usuarios puede acceder a su código fuente de la mayoría de navegadores web comunes, simplemente haciendo clic en el botón " ver código fuente". Los visitantes del sitio pueden, sin su conocimiento, copie su código y hacerlo pasar como propio. Es poco lo que se puede hacer para combatir esto con excepción de ofuscar el código, o intencionalmente escribir el código de una manera que es difícil de leer y entender. Por supuesto, eso no impide que cualquier persona de mayor robo de su código, pero puede disuadir a alguien que quiera modificar su código. Debe tenerse en cuenta que este problema no existe cuando se trabaja con JavaScript embebido en dispositivos móviles.

Seguridad

Seguridad sigue siendo un problema con JavaScript, aunque la situación ha mejorado mucho desde los primeros días del idioma. Algunos de los problemas de seguridad más comunes relacionados con la caída lenguaje bajo la amplia categoría de "vulnerabilidades de cross-site ". Esto es cuando un atacante es capaz de conseguir una página web de confianza, como un sitio de banca en línea, para incluir un script malicioso con sus propios guiones benignos, el script malicioso normalmente registrará su log-in credencial y enviarla al atacante utilizarse en un momento posterior.

Vulnerabilidades de JavaScript

Los hackers utilizan herramientas de explotación de JavaScript para atacar sitios web, organizaciones e individuos. Las vulnerabilidades de JavaScript pueden ser tanto problemas del cliente como pesadillas empresariales, ya que los hackers pueden robar datos del lado del servidor e infectar a los usuarios con malware.

Ataques Cross-Site Scripting (XSS)

El uso más común de la vulnerabilidad de las aplicaciones en las aplicaciones web es el cross-site scripting (XSS). A través de la manipulación de scripts JavaScript y HTML, los hackers ejecutan scripts maliciosos (también conocidos como "maliciosos payloads") utilizando un navegador web de usuario desprevenido que puede resultar en el script que se incrusta en la página web que están visitando. Cada vez que el usuario visita la página web o se realiza una acción predefinida, se desencadena y ejecuta la secuencia de comandos malintencionada.

Los ataques XSS tienen el potencial de causar serias amenazas a empresas y cuentas empresariales que pueden resultar en robo de identidad y robo de datos. Al ejecutar ataques XSS, los hackers pueden inyectarse y propagar virus y gusanos en toda la red de la compañía, acceder a los datos del portapapeles e historias de navegación e incluso obtener el control remoto del navegador, lo que les ayuda a buscar e identificar otras posibles vulnerabilidades que pueden utilizarse para más información. XSS ataques. Debido a la presencia de JavaScript en casi todos los elementos de la experiencia de navegación web, las aplicaciones escritas con JavaScript son las víctimas más comunes de los ataques XSS.

Cross-Site Solicitud de Falsificación

Cross-Site Request Forgery (CSRF) es una forma de exploit que se produce cuando los comandos no autorizados, que normalmente se rechazan, lo que resulta en el sitio web que se cree que el usuario malicioso es un usuario autorizado a través de una autorización falsificada. Tras una explotación exitosa de esta vulnerabilidad, el hacker puede acceder a las funciones de la aplicación web que normalmente se negaría. Los riesgos asociados a los ataques CSRF incluyen la suplantación de identidad y la identidad, la modificación de los datos de la aplicación utilizando las credenciales y permisos de la víctima, el lanzamiento de ataques organizados contra todos los usuarios de la aplicación, la explotación de routers DSL vulnerables y más. CSRF es a menudo pronunciado "mar-surf" y es alternativamente abreviado como XSRF.

Vulnerabilidades de HTML

La inyección de HTML es un ataque similar al de Cross-site Scripting (XSS). Mientras que en la vulnerabilidad XSS el atacante puede inyectar y ejecutar código Javascript, el ataque de inyección HTML sólo permite la inyección de ciertas etiquetas HTML. Cuando una aplicación no maneja correctamente los datos suministrados por el usuario, un atacante puede proporcionar código HTML válido, normalmente a través de un valor de parámetro, e inyectar su propio contenido en la página. Este ataque suele utilizarse en combinación con alguna forma de ingeniería social, ya que el ataque está explotando una vulnerabilidad basada en código y la confianza de un usuario.

Riesgos en HTML

Gran parte de los problemas de seguridad en las aplicaciones web son causados por la falta de seguimiento en dos rubros muy importantes de los que depende cualquier aplicación, las entradas y salidas del sistema. es importante considerar la exposición accidental de datos que pueden ser empleados en un posible ataque sobre el sistema. Los mensajes de error enviados por el servidor, que suelen ser de gran utilidad durante el proceso de desarrollo de la aplicación, pueden ser empleados maliciosamente cuando siguen apareciendo en un entorno de producción, por lo que es necesario deshabilitar todos estos mensajes y editar algunos otros (como los que se envían cuando el servidor no encuentra algún archivo en particular) los cuales también pueden ser utilizados por los atacantes para obtener información sobre nuestro sistema.

Las Desventajas

La seguridad sigue siendo el talón de Aquiles de JavaScript. Los fragmentos de código de JavaScript una vez añadidos a las páginas web en los servidores, estos son descargados y ejecutados en el navegador del cliente permitiendo así que cierto código malicioso pueda ser ejecutado en la máquina del cliente con el objetivo de explotar alguna vulnerabilidad de seguridad conocida en una de las aplicaciones, navegadores o el mismo sistema operativo. Es verdad que hoy día existen estándares de seguridad que restringen la ejecución de código por parte de los navegadores, pero, aun así, se puede ejecutar código que dañe, robe o destruya información del lado del cliente.

Otra desventaja de JavaScript es que este tiende a introducir una cantidad enorme de fragmentos de código en nuestros sitios web. Por suerte, el problema de grandes fragmentos de código JavaScript se resuelve fácilmente mediante el almacenamiento del código JavaScript dentro de archivos separados del código HTML con la extensión. *. Js, dejando una página web mucho más limpia y legible de cara al desarrollador.

Bibliografía

http://www.ordenador.online/Programacion/JavaScript-Programaci%C3%B3n/Los-riesgos-con-JavaScript-.html

https://www.checkmarx.com/sast-supported-languages/javascript-overview-and-vulnerabilities/

https://www.acunetix.com/vulnerabilities/web/html-injection

https://www.seguridad.unam.mx/historico/documento/index.html-id=17

http://blog.capacityacademy.com/2012/10/19/que-es-javascript-ventajas-y-desventajas/