En un esfuerzo por mitigar una gran clase de posibles problemas de secuencias de comandos en sitios cruzados en Firefox, Mozilla bloqueó la ejecución de todas las secuencias de comandos inline y funciones de evaluación potencialmente peligrosas para «about:pages» incorporadas que son la puerta de entrada a preferencias sensibles, configuración y estática del navegador.

El navegador web Firefox cuenta con 45 páginas internas de este tipo alojadas localmente, algunas de las cuales se enumeran a continuación que podría haber notado o utilizado en algún momento:

- about:config – Panel para modificar las preferencias y configuraciones críticas de Firefox.
- about:downloads – Las descargas recientes con Firefox.
- about:memory – Muestra la memoria en uso por Firefox.
- about:newtab – Nueva pestaña por defecto.
- about:plugins – Muestra todos los plugins e información relacionada.
- about:privatebrowsing – Abre una nueva ventana privada.
- about:networking – Muestra información de red.

Cabe mencionar que estos cambios no afectan el funcionamiento de los sitios web de Internet en el navegador Firefox, pero en el futuro, Mozilla promete *«auditar y evaluar de cerca»* el uso de funciones dañinas en extensiones de terceros y otros mecanismos incorporados.

Firefox deshabilita Inline JavaScript por seguridad

Debido a que todas las páginas están escritas en HTML/JavaScript y se muestran en el contexto de seguridad del navegador, también son propensas a ataques de inyección de código que, en caso de vulnerabilidad, podría permitir a los atacantes remotos inyectar y ejecutar código arbitrario en nombre del usuario, es decir, ataques de secuencias de comandos entre sitios (XSS).

Para agregar una primera línea de defensa sólida contra los ataques de inyección de código, incluso cuando hay una vulnerabilidad, Mozilla bloqueó la ejecución de todos los scripts en línea, por lo tanto, también los scripts inyectados, mediante la implementación de una estricta Política de Seguridad de Contenido (CSP), para poder garantizar que el código JavaScript solo se ejecuta cuando se carga desde un recurso empaquetado mediante el protocolo interno.

Para lograr esto, Mozilla tuvo que reescribir los controladores de eventos inline y mover todo el código JavaScript outline en archivos separados para las 45 páginas.

> *«No permitir ningún script en línea en ninguna de las páginas about:pages limita la superficie de ataque de la ejecución de código arbitrario y, por lo tanto, proporciona una primera línea de defensa sólida contra los ataques de inyección de código»,* dijo Mozilla en su blog.

Cuando los atacantes no pueden inyectar script de forma directa, utilizan la función eval() de JavaScript y métodos similares para engañar a las aplicaciones de destino para que conviertan el texto en un JavaScript ejecutable para lograr la inyección de código.

Por lo tanto, además de las secuencias de comandos en línea, Mozilla también eliminó y bloqueó las funciones de evaluación, que el fabricante del navegador cree que es otra *«herramienta peligrosa»*, ya que analiza y ejecuta una cadena arbitraria en el mismo contexto de seguridad.

> *«Si ejecuta eval() con una cadena que podría verse afectada por una parte maliciosa, puede terminar ejecutando código malicioso en la máquina del usuario con los permisos de su página web/extensión»,* explicó Mozilla en sus documentos web de MDN.

Google también comparte el mismo pensamiento, *«evaluar es peligroso dentro de una extensión porque el código que ejecuta tiene acceso a todo en el entorno de alto permiso de la extensión».*

Para esto, Mozilla reescribió todo el uso de funciones de evaluación de contextos privilegiados del sistema y el proceso padre en la base de código de su navegador web Firefox.

Además, la compañía también agregó aserciones eval() que no permitirán el uso de la función eval() y sus familiares en contextos de script con privilegios del sistema e informarán al Equipo de Seguridad de Mozilla sobre instancias que aún se desconocen de eval().