Micro Focus

Contenido de seguridad del software Fortify

Actualización 3 de 2020

viernes, 25 de septiembre de 2020

Acerca del Equipo de investigación de seguridad para software Micro Focus Fortify

El Equipo de investigación de seguridad para software Fortify transforma la investigación más avanzada en inteligencia de seguridad que impulsa la cartera de productos de Fortify, incluidos Fortify Static Code Analyzer (SCA), Fortify WebInspect y Fortify Application Defender. Actualmente, el contenido de seguridad del software Micro Focus Fortify admite 1.032 categorías de vulnerabilidad en 27 lenguajes de programación y abarca más de un millón de API distintas.

Puede obtener más información en: https://software.microfocus.com/en-us/software/security-research

El Equipo de investigación de seguridad para software (SSR) se complace en anunciar la disponibilidad inmediata de actualizaciones para Fortify Secure Coding Rulepacks (en inglés, versión 2020.2.0), Fortify WebInspect SecureBase (disponible mediante SmartUpdate) y Fortify Premium Content.

Micro Focus Fortify Secure Coding Rulepacks [SCA]

Con esta versión, Fortify Secure Coding Rulepacks detecta 815 categorías únicas de vulnerabilidades en 27 lenguajes de programación y abarca más de un millón de API distintas. En resumen, esta versión incluye lo siguiente:

Compatibilidad con .NET System.Text.Json (versión 3.1)[1]

System.Text.Json es un espacio de nombres integrado de alta velocidad conforme a los estándares que funciona como alternativa al paquete Json.NET de Newtonsoft. También proporciona tipos de lectura y escritura de texto JSON codificado como UTF-8. La compatibilidad incluye flujo de datos mediante los diversos métodos de serialización y deserialización, así como mayor cobertura para las siguientes categorías:

- Cross-Site Scripting: reflejados
- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Privacy Violation
- System Information Leak: External
- System Information Leak: Internal

Compatibilidad con corrutinas y bibliotecas estándar de Kotlin[2]

Cobertura extendida de las bibliotecas estándar de Kotlin, incluida compatibilidad con corrutinas. Las corrutinas proporcionan una API rica y flexible para el desarrollo de código simultáneo y asincrónico. Esta actualización incorpora un análisis de flujo de datos mejorado en las aplicaciones que utilizan corrutinas.

Interoperabilidad de Java y Kotlin, y mejoras para Android[3]

Compatibilidad de lenguajes mejorada en las categorías de debilidades existentes para proyectos que contienen código fuente Java y Kotlin, así como para aplicaciones Android desarrolladas mediante Kotlin.

Mejoras de Java 14[4]

Mejoras en las categorías de debilidades existentes para admitir cambios de versiones hasta Java 14.

Actualizaciones de versiones de ECMAScript[5]

Se actualizó la cobertura de reglas para versiones de ECMAScript a fin de admitir todas las versiones hasta ECMAScript 2020.

---

[1] Debido a su diseño, el espacio de nombres System.Text.Json no es vulnerable a los mismos problemas de inyección de JSON informados en el paquete Json.NET de Newtonsoft.

[2] Las API de corrutinas experimentales, como aquellas que utilizan determinadas expresiones, no se admiten en esta actualización.

[3] Se requiere SCA 20.2 o una versión posterior.

[4] Se requiere SCA 20.2 o una versión posterior.

[5] La compatibilidad completa requiere SCA 20.2 o una versión posterior.

Logrus para Go (versión 1.6.0)

Logrus es un registrador estructurado para Go, totalmente compatible por API con el registrador de bibliotecas estándar. Logrus es un paquete de registro común que puede introducir riesgos relacionados con el registro en las aplicaciones empresariales. Entre las categorías compatibles se incluyen:

- Poor Logging Practice: Use of a System Output Stream
- Privacy Violation
- System Information Leak

Compatibilidad con Dockerfile[6]

Docker permite que el código de empaquetado y sus dependencias en un entorno de espacio aislado (también denominados contenedores) se ejecuten en cualquier entorno informático. Dockerfile es la configuración de imágenes de contenedor. Esta actualización aborda las vulnerabilidades más comunes en Dockerfile.
La compatibilidad inicial cubre la categoría actual "Password Management: Password in Configuration File", así como las siguientes nuevas categorías:

- Dockerfile Misconfiguration: Default User Privilege
- Dockerfile Misconfiguration: Privileged Container
- Dockerfile Misconfiguration: Privileged Port
- Dockerfile Misconfiguration: Sensitive Host Discovery
- Dockerfile Misconfiguration: SSH Service

DISA STIG 4.11

Para ofrecer soporte a nuestros clientes federales en lo que respecta al cumplimiento, se agregó una correlación de la taxonomía de Micro Focus Fortify con la versión 4.11 de la STIG de seguridad y desarrollo de aplicaciones de la Agencia de sistemas de información de defensa (DISA) estadounidense.

Otras erratas

En esta versión, seguimos invirtiendo recursos para garantizar la reducción del número de falsos positivos y para mejorar la capacidad de auditoría de los problemas por parte de los clientes. Los clientes también verán cambios en los problemas comunicados en relación con lo siguiente:

Aviso de desuso para versiones de SCA anteriores a 17.x:
Informamos a nuestros usuarios que esta será la última versión de Rulepacks compatible con versiones de SCA anteriores a 17.x. En la próxima versión, las versiones de SCA anteriores a 17.x no cargarán la instancia de Rulepacks. Se deberá cambiar a una versión inferior de Rulepacks o actualizar la versión de SCA.
En las versiones futuras, continuaremos admitiendo las últimas cuatro versiones principales de SCA.

[6] Se requiere SCA 20.2 o una versión posterior.

Asignaciones obsoletas - Guía de implementación técnica de seguridad versiones 3.x (STIG 3.x):
A lo largo de los años, el Equipo de investigación de seguridad para software continuó creando asignaciones a partir de categorías de Fortify para prácticas recomendadas y estándares externos. Por lo tanto, hoy en día la cantidad de versiones de diferentes estándares que son compatibles ha crecido notablemente, aunque muchas versiones anteriores no se usarán.
En este sentido, marcamos las versiones de STIG previas a la versión 4.0 como "obsoletas" mediante un nuevo atributo XML en el archivo externalmetadata.xml. Este cambio significa que las aplicaciones de Fortify Software Security Center y de SCA a partir de la versión 20.2 no mostrarán estas versiones anteriores de forma predeterminada. Si debe comprobar estas versiones antiguas, una solución alternativa es cambiar las asignaciones de "obsoletas" a "falsas", o bien quitar el atributo directamente.

Nomenclatura de lista negra y lista blanca:
Siguiendo las tendencias del sector, el Equipo de investigación de seguridad para software, como parte de Micro Focus Fortify, decidió dejar de usar los términos "lista negra" (blacklist) y "lista blanca" (whitelist) en favor de conceptos más neutros y semánticamente más específicos. Decidimos usar términos más descriptivos, como "lista de denegados" (deny list) y "lista de permitidos" (allow list), respectivamente.
Este cambio también implica la modificación de los nombres de las siguientes categorías, lo cual podría traducirse en problemas nuevos y otros eliminados:

- “Deserialization Bad Practice: Blacklist” ahora se denomina “Deserialization Bad Practice: Deny List”
- "Django Bad Practices: Blacklisted Attributes" ahora se denomina "Django Bad Practices: Attributes in Deny List"
- "Setting Manipulation: User-Controlled Whitelist" ahora se denomina "Setting Manipulation: User-Controlled Allow List"

Actualización de Weak Encryption: Insecure Mode of Operation
Consideramos que ya no es seguro usar el modo de cifrado simétrico con encadenamiento de bloques de cifrado (CBC). A partir de esta versión, el uso de CBC se notificará como Weak Encryption: Insecure Mode of Operation para las bibliotecas admitidas actualmente. Luego de considerar los ataques BEAST, POODLE, Zombie POODLE, GOLDENDOODLE, LUCKY13 y de oráculo de relleno, determinamos que CBC ya no es seguro en función de la investigación en criptografía conocida actualmente.

Mejoras en falsos positivos:
Seguimos escuchando a nuestros clientes y nos esforzamos por mejorar las tasas de falsos positivos. En esta versión, trabajamos en los siguientes aspectos para reducir el número de falsos positivos:

- Cross-Site Request Forgery
  Reducción para dar cuenta de otros tipos de tokens CSRF y campos ocultos internos de ASP.NET
- Server-Side Request Forgery
  Eliminaciones en algunas aplicaciones de Spring
- Poor Logging Practice: Use of a System Output Stream
  Eliminación cuando el contexto se basa en utilidades o aplicaciones de línea de comandos

- Dead Code: Unused Field
  Expresiones lambda internas
- Privacy Violation
  Eliminación de problemas duplicados con contraseñas dentro de expresiones lambda de Java, Scala y Kotlin
- Las propiedades del sistema Java ahora se abordan por separado en las propiedades integradas, lo cual reduce los falsos positivos en muchos casos y permite encontrar posibles nuevas filtraciones de información del sistema.

Vista previa:[7]
Se reajustaron las reglas para los problemas Null Dereference y Redundant Null Check a fin de permitir la reducción de tasas de falsos positivos. En particular, se agregó la capacidad de escribir reglas personalizadas para controlar funciones internas de comprobación de valores NULL.
Sin embargo, no queda claro si los beneficios serán universales. Por lo tanto, estas mejoras disponibles en SCA 20.2 se muestran desactivadas de forma predeterminada. Si desea probar estas mejoras, póngase en contacto con la atención al cliente.

Micro Focus Fortify SecureBase [Fortify WebInspect]

Fortify SecureBase combina las comprobaciones para miles de vulnerabilidades con las directivas que guían a los usuarios en las siguientes actualizaciones disponibles inmediatamente con SmartUpdate:

Compatibilidad de vulnerabilidades

Weak Cryptographic Signature: User-Controlled Key Source

Los ataques de confusión de claves ponen en riesgo la integridad de las instancias de JSON Web Token (JWT). Las implementaciones de JWT que no seleccionan cuidadosamente una fuente de claves para verificar un token recibido del cliente y permiten que un usuario malintencionado falsifique la carga útil de tokens son vulnerables a estos ataques. Esta versión incluye una comprobación para evaluar el uso de JWT frente a los ataques de confusión de claves.

JSON Web Token: Missing Protection Claims

Las instancias de JSON Web Token que no implementan medidas de protección como "jti", "exp" y "aud" pueden ser susceptibles a ataques de suplantación y reproducción. Esta versión incluye varias comprobaciones que detectan si los tokens encontrados durante un análisis carecen de estas medidas.

[7] Se requiere SCA 20.2 o una versión posterior.

Access Control: Authorization Bypass

JSON Web Token (JWT) ofrece protección contra la manipulación de datos, ya que la información que contiene incluye una firma digital hecha mediante el algoritmo HMAC o RSA. Las aplicaciones que no logran verificar la firma pueden ser susceptibles al acceso no autorizado y a los ataques de suplantación. Esta versión incluye varias comprobaciones para evaluar si la implementación de JWT es vulnerable a ataques de firmas faltantes o arbitrarias.

OAuth2: Insufficient state Parameter Entropy

Las implementaciones del protocolo OAuth2 que no usan un valor de parámetro "state" (estado) con suficiente entropía son susceptibles a ataques de suplantación. Esta versión incluye una comprobación para evaluar si el parámetro de estado no logra cumplir con los requisitos de entropía necesarios.

OAuth2: Insufficient Refresh Token Revocation

Los servidores de autorización que no verifican el identificador del cliente, el secreto del cliente y la caducidad del token de actualización antes de volver a emitir un nuevo token de acceso pueden ser vulnerables a ataques de suplantación. Esta versión incluye una comprobación para detectar si se puede usar un token de actualización con el fin de generar varios tokens de acceso sin el identificador ni el secreto del cliente.

OAuth2: Insufficient Authorization Code Expiration

Los servidores de autorización de OAuth2 que no hacen caducar oportunamente los códigos de autorización pueden ser susceptibles a ataques de suplantación, ya que pueden seguir otorgando un nuevo token de acceso válido para códigos de autorización robados. Esta versión incluye una comprobación para detectar el uso indebido de códigos de autorización.

Cross-Site Request Forgery

Las aplicaciones que no logran usar un parámetro "state" (estado) en el flujo de autorización de OAuth2 son susceptibles a las acciones no autorizadas que se hacen pasar por clientes. Asimismo, las aplicaciones que usan almacenamiento de cookies para instancias de JSON Web Token (JWT) corren el riesgo de sufrir ataques de CSRF similares. Esta versión incluye varias comprobaciones para detectar instancias de JWT y OAuth2 vulnerables a CSRF.

Weak Cryptographic Signature: Tamaño de clave insuficiente

Las instancias de JSON Web Token contienen información firmada digitalmente para demostrar la integridad de la carga útil. Ciertos algoritmos simétricos como HS256, que tienen una clave secreta débil, podrían ser atacados por fuerza bruta, lo cual le permitiría al atacante generar una carga útil arbitraria y firmarla con una clave válida. Esta versión incluye una comprobación para detectar si se usa un algoritmo débil como HS256 para firmar los tokens.

SQL Injection[8]

SQLite es una biblioteca de C que ofrece una base de datos ligera basada en disco, la cual no requiere un proceso de servidor por separado y permite acceder a la base de datos mediante una variante no estándar del lenguaje de consultas SQL. Esta versión incluye una mejora en la comprobación de SQL Injection de WebInspect para detectar vulnerabilidades de SQL Injection en las aplicaciones web mediante la biblioteca SQLite versión 3.16.0 o superior.

Informe de cumplimiento

DISA STIG 4.11

Para proporcionar soporte a nuestros clientes federales en lo que respecta al cumplimiento, esta versión contiene una correlación de las comprobaciones de WebInspect con la versión más reciente (4.11) de la STIG para la seguridad y el desarrollo de aplicaciones de la Agencia de sistemas de información de defensa estadounidense.

Actualizaciones de directivas

DISA STIG 4.11

Se incorporó una directiva personalizada a la lista existente de directivas admitidas en WebInspect SecureBase para incluir las comprobaciones pertinentes a DISA STIG 4.11.

Web API

Se incorporó una directiva personalizada a la lista existente de directivas admitidas en WebInspect SecureBase para incluir las comprobaciones pertinentes a la evaluación de seguridad de API.

[8] La inclusión de información de una base de datos para SQLite como parte del contenido del reporte requiere WebInspect 20.2 o una versión posterior.

Otras erratas

En esta versión, seguimos invirtiendo recursos para garantizar la reducción del número de falsos positivos y para mejorar la capacidad de auditoría de los problemas por parte de los clientes. Los clientes también verán cambios en los problemas comunicados en relación con lo siguiente:

El contenido del informe Insecure Transport: Weak SSL Cipher:
Se modificó la comprobación identificada con el número 11285 para notificar que todas las instancias de cifrados de modo CBC son débiles en comparación con la descripción del contenido del informe en Weak SSL Protocol al detectar SSLv3 o TLS1.0 junto con los cifrados de CBC. Luego de considerar debilidades como POODLE, GoldenPOODLE y ZombieDOODLE, ya no recomendamos los cifrados de modo CBC para la seguridad de la capa de transporte.

Nomenclatura de lista negra y lista blanca:
Siguiendo las tendencias del sector, el Equipo de investigación de seguridad para software, como parte de Micro Focus Fortify, decidió dejar de usar los términos "lista negra" (blacklist) y "lista blanca" (whitelist) en favor de conceptos más neutros y semánticamente más específicos. En cambio, decidimos usar términos más descriptivos, como "lista de denegados" (deny list) y "lista de permitidos" (allow list), respectivamente, en el contenido de las comprobaciones de seguridad.

Micro Focus Fortify Premium Content

El equipo de investigación crea, amplía y mantiene diversos recursos independientes de nuestros principales productos de inteligencia de seguridad.

DISA STIG 4.11

Para acompañar las nuevas correlaciones, esta versión también contiene un nuevo paquete de informes para Fortify Software Security Center compatible con DISA STIG 4.11 que se puede descargar de Fortify Customer Support Portal, en la sección Premium Content.

Taxonomía de Micro Focus Fortify: errores en la seguridad del software

El sitio Taxonomía de Fortify, que contiene descripciones de la compatibilidad con las nuevas categorías añadidas, está disponible en https://vulncat.fortify.com/es. Los clientes que busquen el sitio antiguo con la última actualización compatible pueden encontrarlo en Micro Focus Fortify Support Portal.

Comuníquese con el soporte técnico de Fortify
Micro Focus Fortify
https://softwaresupport.softwaregrp.com/
+1 (844) 260-7219

Comuníquese con SSR
Alexander M. Hoole
Director del Equipo de investigación de seguridad para software
Micro Focus Fortify
hoole@microfocus.com
+1 (650) 258-5916

© Copyright 2020 Micro Focus o una de sus empresas afiliadas. La información que contiene este documento está sujeta a cambios sin previo aviso. Las únicas garantías para productos y servicios de Micro Focus son las que se especifican en las declaraciones expresas de garantía que acompañan a dichos productos y servicios. El contenido de este documento no debe interpretarse como garantía adicional. Micro Focus no se hace responsable de los errores u omisiones técnicos o editoriales aquí contenidos.

25 de septiembre de 2020