Micro Focus

Contenido de seguridad del software Fortify

Actualización 2 de 2020

viernes, 26 de junio 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 1.022 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 810 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 la biblioteca principal de Kotlin (versión 1.3)[1]

La compatibilidad inicial con las bibliotecas estándar de Kotlin abarca los siguientes paquetes dirigidos a JVM:

- kotlin
- kotlin.collections
- kotlin.comparisons
- kotlin.io
- kotlin.properties
- kotlin.random
- kotlin.ranges
- kotlin.sequences
- kotlin.streams
- kotlin.text

La interoperabilidad limitada con Java implica que todas las categorías y las bibliotecas admitidas por Java son todas parcialmente compatibles con Kotlin. Las siguientes son compatibles específicamente con las bibliotecas estándar de Kotlin:

- Cross-Site Scripting: Inter-Component Communication
- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: reflejados
- Denial of Service: expresión regular
- Denial of Service: StringBuilder
- Formula Injection
- Insecure SSL: Certificado de confianza excesivamente permisivo
- Log Forging
- Path Manipulation
- Poor Logging Practice: Use of a System Output Stream
- Privacy Violation
- Server-Side Request Forgery
- System Information Leak
- System Information Leak: External
- System Information Leak: Internal
- Violación de límite de confianza
- Unchecked Return Value

[1] Se requiere la versión de revisión SCA 20.1.2. El uso de SCA 20.1 dará como resultado múltiples errores en torno a reglas y funciones lambda específicas, que no se producen con la versión de revisión.

Mejoras de C# 8 y .NET Core

Se actualizó la compatibilidad con las bibliotecas C# 8 y .NET Core, incluidas las versiones 2.1, 2.2, 3.0 y 3.1. Con esto se incluye la compatibilidad total con las reglas para .NET Standard 2.1, así como un flujo de datos mejorado cuando se utilizan las operaciones asincrónicas de C# 8 *foreach*, *range* y *using*.

Mejoras en la biblioteca principal de Python[2]

Se actualizó la compatibilidad con las bibliotecas principales de Python, incluidas las versiones 3.7 y 3.8. Además de las categorías previamente compatibles, ahora se admite la siguiente categoría nueva:

- Python Bad Practices: Leftover Debug Code

Mejoras en la compatibilidad con las bibliotecas principales de Go (versión 1.13)

Se amplió la compatibilidad con las bibliotecas principales de Go para cubrir crypto/tls. Go es un lenguaje de código abierto con un sistema de tipos estático diseñado por Google™ que facilita la compilación de software para que resulte sencillo, confiable y eficiente. En términos de sintaxis, Go es similar a C, pero presenta mecanismos de seguridad de memoria, recopilación de elementos no utilizados y tipado estructural. Entre las categorías compatibles se incluyen:

- Insecure SSL: verificación de la identidad del servidor deshabilitada
- Path Manipulation
- Privacy Violation
- Server-Side Request Forgery
- Setting Manipulation
- System Information Leak: External

GORM (versión 1.9.12)

Se admite la biblioteca de GORM, que es un proyecto de mapeo objeto-relacional (ORM) de código abierto para Golang escrito por Jinzhu. GORM es una biblioteca de ORM común que puede introducir riesgos relacionados con la base de datos en las aplicaciones empresariales. Entre las categorías compatibles se incluyen:

- Access Control: base de datos
- Connection String Parameter Pollution
- Dynamic Code Evaluation: inyección de código
- Insecure Transport
- Log Forging
- Password Management: Empty Password
- Password Management: contraseña codificada de forma rígida
- Password Management: criptografía débil
- Privacy Violation
- Setting Manipulation
- SQL Injection
- System Information Leak: Internal
- Weak Cryptographic Hash

[2] Se requiere SCA 20.2 para obtener mejores resultados cuando los proyectos incluyen una sintaxis recientemente introducida, como las expresiones de asignación.

OWASP ASVS 4.0

Para ayudar a los clientes en busca de marcos que permitan ejecutar su programa appsec, esta versión admite la correlación entre nuestras categorías de taxonomía de Micro Focus Fortify y los requisitos de Application Security Verification Standard (ASVS) de OWASP, versión 4.0. ASVS proporciona una lista de requisitos o pruebas que los clientes pueden utilizar para definir, compilar, probar y verificar aplicaciones seguras.

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:

Cambios en el reino:
En algunos lugares determinados, se cambió el reino de categorías específicas por motivos de coherencia. En versiones anteriores de SCA (versiones inferiores a 6.30, es decir, todas las anteriores al 17-04-2015), esto puede hacer que algunos problemas se consideren nuevos/eliminados durante la fusión, en lugar de permanecer como "actualizados". Este problema no se producirá en ninguna versión de SCA 6.30 o versiones posteriores.
Esto también puede afectar de forma general el filtrado, si los usuarios filtran por reino. Entre las categorías afectadas se incluyen:

- Access Control: Manipulación de ACL. Se cambió de "Representación y validación de entrada" a "Funciones de seguridad"
- Access Control: Omisión de autorización Se cambió de "Representación y validación de entrada" a "Funciones de seguridad"
- System Information Leak: Struts 2. Se cambió de "Representación y validación de entrada" a "Encapsulación".

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:

- Se eliminó una regla de fuente de JavaScript específica que generaba falsos positivos en varias categorías.
- Se mejoraron las reglas de "Unreleased Resource: Database" de Java para tener en cuenta las conexiones que cierran los recursos subyacentes según la especificación JDBC 4.3.
- Se eliminaron los problemas de "Path Manipulation" con falsos positivos en Java relacionados con File.createTempFile().
- Se deshabilitó la notificación de problemas de archivos de definición de TypeScript.
- Se eliminaron algunos falsos positivos de "Server-Side Request Forgery" de .NET en SendAsync.
- Se redujeron los falsos positivos de "Spring Security Misconfiguration: Incorrect Request Matcher Type" de Java.
- Se eliminaron los problemas duplicados de "Weak Encryption" de .NET.
- Se eliminaron los falsos positivos de "ASP.NET Bad Practices: Non-Serializable Object Stored in Session" de .NET en los tipos de cadenas.
- Se eliminaron los falsos positivos de Golang que se producían por la corrupción no intencional de los objetos net.http.Response.

- Se identificaron y se eliminaron los inusuales falsos positivos de "Code Correctness: Class Does Not Implement equals" dentro de Java.
- En JavaScript, se eliminaron muchos problemas de falsos positivos de "Key Management". Se continuará avanzando en esto en las versiones posteriores.

Errores y advertencias:
Con SCA 19.2, para todos los lenguajes de programación compatibles, la carga de reglas durante un análisis generará la siguiente advertencia en los registros de compilación y análisis cuando esto se combina con el conjunto completo de paquetes de reglas 2020.2.0:

```
[WARN  20599]
The rule language kotlin has no corresponding mapping in SCA
```

Este es un problema conocido en SCA 19.2 que se puede solucionar si se elimina el siguiente archivo de paquete de reglas de Kotlin:

<*FORTIFY_DIR*>\Fortify_SCA_and_Apps_19.2.0\Core\config\rules\core_kotlin.bin

El análisis de Kotlin requiere SCA 20.1.2.

Mejoras de VulnCat
Mejoramos la clasificación de las categorías y las subcategorías de filtro para que las entradas se muestren en un orden lógico.

Compatibilidad con la validación de ESAPI
Escuchamos a nuestros clientes, y muchos muestran preocupación por las reglas de SCA que permiten a las API de ESAPI eliminar problemas como si se hubieran solucionado, debido a preocupaciones en torno a la seguridad en la biblioteca.
En esta versión, eliminamos la validación de las bibliotecas ESAPI, y ahora el flujo de datos continúa a través de las funciones de codificación sin evitar que se produzcan problemas de flujo de datos.
Como sabemos que algunos clientes desean seguir usando la biblioteca ESAPI, específicamente en los productos antiguos, estamos poniendo a disposición un paquete de reglas separado para incluir nuevamente esta validación.

Si desea continuar usando ESAPI, puede utilizar nuestro paquete de reglas de extensión de ESAPI que se encuentra en Support Portal, en la sección Premium Content, Fortify Exchange.

Tenga en cuenta que esta actualización también elimina todos los problemas marcados como "Obsolete: Deprecated by ESAPI", los cuales, a su vez, se pueden encontrar al usar el paquete de reglas separado mencionado anteriormente.

Actualización de Weak Encryption
3DES, también conocido oficialmente como el algoritmo de cifrado de datos triple (TDEA), ahora se informará en Weak Encryption dadas las recomendaciones del Instituto Nacional de Estándares y Tecnología (NIST).

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

Web Server Misconfiguration: Information Disclosure

Spring Boot incluye una serie de características adicionales también conocidas como accionadores, gracias a las cuales los administradores pueden supervisar y administrar sus aplicaciones web. Esta versión incluye una verificación para detectar información confidencial de Spring Boot Actuator disponible para los usuarios sin privilegios. Mediante el uso de extremos de accionador, un atacante puede recopilar información de auditoría, estado y métricas, o incluso realizar un ataque de ejecución de código remoto en algunas configuraciones.

Cross-Site Scripting: reflejados

Se utiliza la API ResolveUrl de ASP.NET para resolver la ruta relativa a la raíz de la aplicación en el tiempo de ejecución. La API no valida la ruta relativa en la que se incluyen valores formateados de ID de estado sin cookies y permite que un atacante realice un ataque Cross-Site Scripting en la aplicación. Esta versión incluye una comprobación que permite detectar esta vulnerabilidad en aplicaciones .NET.

Cookie Security: Missing SameSite Attribute

El atributo SameSite de las cookies proporciona un mecanismo simple para proteger las aplicaciones contra los ataques de falsificación Cross-Site Request Forgery. Las versiones recientes de exploradores pueden rechazar las cookies que no establecen el atributo SameSite. Esta versión incluye una comprobación para detectar las instancias de cookies configuradas que no logran establecer el atributo SameSite.

Cookie Security: Misconfigured Prefix

Las cookies con un nombre que contiene el prefijo _Host- o _Secure- aplican restricciones de nivel de host y el requisito de que la cookie solo debe enviarse mediante una conexión HTTPS segura para protegerla contra escuchas y suplantaciones accidentales. El explorador puede rechazar las cookies que no tienen los atributos Path, Domain y Secure configurados correctamente. En esta versión, se incluye una comprobación para detectar las cookies con prefijos mal configurados.

Cookie Security: Overly Permissive SameSite Attribute

El valor Strict para el atributo SameSite en las cookies obliga a los exploradores a que solo agreguen esa cookie a las solicitudes producidas por la navegación de nivel superior hacia el dominio o cuando el mismo host realiza la solicitud desde varias etiquetas HTML con vínculos de iframe, enlace y formulario, etc. Es necesario configurar los identificadores de sesión con un valor Strict en el atributo SameSite para brindar la máxima protección a la aplicación contra los ataques Cross-Site Request Forgery. Esta versión incluye una comprobación para detectar las instancias en las que el atributo SameSite no se configuró en Strict para las cookies de sesión.

Informe de cumplimiento

OWASP Application Security Verification Standard (ASVS)

OWASP ASVS ofrece una metodología para probar las aplicaciones web con fines de control de seguridad y también proporciona pautas de desarrollo seguro. Esta versión contiene una correlación de las comprobaciones de WebInspect con la última versión de OWASP ASVS 4.0.

Actualizaciones de directivas

Directiva de OWASP Application Security Verification Standard (ASVS)

Además de la correlación de las comprobaciones de WebInspect con la última versión de OWASP ASVS, esta versión incluye una directiva para identificar las vulnerabilidades por las que se produce una correlación con OWASP ASVS.

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:

- Mejoras en el contenido de seguridad de Often Misused: Weak SSL Certificate. Este ahora refleja con mayor precisión la información sobre las razones por las que un certificado se considera débil. Se agregó una nueva comprobación con el identificador 11635.

Micro Focus Fortify Premium Content

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

OWASP Application Security Verification Standard (ASVS)

Application Security Verification Standard (ASVS) es un listado de requisitos de seguridad y pruebas de aplicaciones que deben realizarse durante la configuración y el ciclo de vida de desarrollo de software (SDLC) para desarrollar software seguro. Esperamos que esta asignación siga evolucionando a medida que colaboramos con socios de la industria para mejorar la forma en que se diseñan las asignaciones. Para acompañar las nuevas correlaciones, esta versión también contiene un nuevo paquete de informes para Fortify SSC compatible con OWASP ASVS 4.0, que se puede descargar de Fortify Customer 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. 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

26 de junio de 2020