Anuncio de publicación de Software Security Research

Contenido de seguridad del software Fortify

Actualización 1 de 2022

viernes, 25 de marzo de 2022

Acerca de CyberRes Fortify Software Security Research

Fortify Software Security Research 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 CyberRes Fortify Software Security admite 1.166 categorías de vulnerabilidad en 29 lenguajes de programación y abarca más de un millón de API distintas.

Fortify Software Security Research (SSR) se complace en anunciar la disponibilidad inmediata de actualizaciones para Fortify Secure Coding Rulepacks (en inglés, versión 2022.1.0), Fortify WebInspect SecureBase (disponible mediante SmartUpdate) y Fortify Premium Content.

CyberRes Fortify Secure Coding Rulepacks [SCA]

Con esta versión, Fortify Secure Coding Rulepacks detecta 946 categorías únicas de vulnerabilidades en 29 lenguajes de programación y abarca más de un millón de API distintas. En resumen, esta versión incluye lo siguiente:

Actualizaciones de Log4j (versión compatible: 2.17)

Log4j es un marco de registro popular para Java que ha estado en el punto de mira en los últimos meses debido a las vulnerabilidades de gran importancia descubiertas dentro del marco. Esta versión incluye un soporte mejorado para identificar exactamente qué partes de su código fuente son susceptibles a la vulnerabilidad de Log4Shell, y las incluye dentro de la categoría *Dynamic Code Evaluation: JNDI Reference Injection.*
Además, el soporte actualizado de Log4j cubre las últimas versiones de Log4j para el siguiente espacio de nombres:

- org.apache.logging.log4j

El soporte también mejora la cobertura en las siguientes categorías de debilidades:

- Code Correctness: Stack Exhaustion
- Dynamic Code Evaluation: JNDI Reference Injection
- Log Forging
- Log Forging (debug)
- Privacy Violation
- System Information Leak

Azure Functions (Python, versión compatible: 3.10.x)

Azure Functions es una solución de computación en la nube sin servidor que puede ejecutar código en respuesta a eventos predefinidos, como llamadas API o transacciones de bases de datos, o administrar colas de mensajes en otros servicios de Azure. En esta versión, ampliamos la compatibilidad con Azure Functions para cubrir las funciones de desencadenador HTTP en Python. El desencadenador HTTP ayuda a invocar una función con una solicitud HTTP y se puede usar para crear API sin servidor y responder a webhooks.
El soporte incluye la cobertura de las siguientes categorías:

- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: Reflected
- Header Manipulation
- Header Manipulation: Cookies
- Privacy Violation
- System Information Leak: External

Compatibilidad con GraphQL: Python Graphene (versión compatible: 3.0.0)

Esta versión incluye compatibilidad inicial con el servidor GraphQL para Python Graphene. GraphQL es un proyecto de código abierto desarrollado por Facebook que presenta un lenguaje de consulta fuertemente tipado y un motor de tiempo de ejecución del servidor para las API. GraphQL ha sido un estándar abierto desde 2015 y actualmente es compatible con más de una veintena de lenguajes de programación. Graphene es una arquitectura de servidor GraphQL popular para aplicaciones de Python. Esta versión añade las siguientes dos categorías para detectar debilidades en las API de GraphQL desarrolladas con Graphene:

- GraphQL Bad Practices: Introspection Enabled
- GraphQL Bad Practices: GraphiQL Enabled

Actualizaciones de Kotlin (versión compatible: 1.5)

Kotlin es un lenguaje de propósito general con un sistema de tipos estático que ofrece interoperabilidad con Java. Esta versión incluye una mejor compatibilidad con las API de la biblioteca estándar introducidas en Kotlin 1.5 dirigidas a la máquina virtual Java (JVM).

Sequelize (versión compatible: 6.17)

Sequelize es una herramienta de mapeo objeto-relacional (ORM, por sus siglas en inglés) basada en promesas diseñada para simplificar el trabajo con muchos dialectos SQL populares dentro de las aplicaciones Node.js. El soporte incluye la cobertura de las siguientes categorías:

- Access Control: base de datos
- Password Management: Empty Password
- Password Management: contraseña codificada de forma rígida
- Password Management: Null Password
- SQL Injection

Archivos referenciados no seguros en HTML

Todas las referencias a sitios de terceros dentro de las páginas web se deben realizar a través de una conexión segura. Por ello, esta versión incluye soporte para las siguientes categorías nuevas dentro de los archivos HTML:

- Dynamic Code Evaluation: Insecure Transport
- Insecure Transport: External Link

Detección de base de datos de contraseñas compartida

Una base de datos de contraseñas es un archivo o conjunto de archivos creados para almacenar contraseñas de forma segura. Las bases de datos de contraseñas normalmente se cifran mediante una contraseña maestra o una clave maestra. Sin embargo, no deben usarse para mantener el uso de contraseñas dentro de una aplicación a lo largo del ciclo de vida de desarrollo. En esta versión, notificamos la existencia de bases de datos como: *Password Management: Shared Password Database*. Las bases de datos de contraseñas compatibles incluyen lo siguiente:

- KeePass
- 1Password
- Password Safe
- MacOS Keychain
- Gnome Keyring
- KDE KWallet

Infraestructura como código en la nube

Esta versión incluye compatibilidad ampliada con infraestructura como código (IaC, por sus siglas en inglés) en la nube. La infraestructura como código es el proceso de administrar y aprovisionar recursos informáticos a través de código, en lugar de procesos manuales. Las tecnologías compatibles incluyen AWS, AWS CloudFormation, Azure ARM, Kubernetes K8S y Azure Kubernetes Service. Los problemas más comunes relacionados con la configuración de los servicios mencionados ahora se comunican al desarrollador.

Entre las categorías adicionales compatibles se incluyen las siguientes:

- Malas prácticas de Ansible: CloudWatch Log Group Retention Unspecified
- Malas prácticas de Ansible: Unrestricted AWS Lambda Principal
- Malas prácticas de Ansible: User-Bound AWS IAM Policy
- Error de configuración de Ansible: Azure Monitor Missing Administrative Events
- Insecure Storage: Missing EC2 AMI Encryption
- Insecure Storage: falta cifrado de EFS
- Insecure Storage: Missing Kinesis Stream Encryption
- Insecure Transport: Azure App Service
- Insecure Transport: Azure Storage
- Malas prácticas de Kubernetes: Automated iptables Management Disabled
- Malas prácticas de Kubernetes: Kernel Defaults Overridden
- Malas prácticas de Kubernetes: Kubelet Streaming Connection Timeout Disabled
- Malas prácticas de Kubernetes: Missing NodeRestriction Admission Controller
- Malas prácticas de Kubernetes: Missing PodSecurityPolicy Admission Controller
- Malas prácticas de Kubernetes: Missing Security Context
- Malas prácticas de Kubernetes: Missing SecurityContextDeny Admission Controller
- Malas prácticas de Kubernetes: Missing ServiceAccount Admission Controller
- Malas prácticas de Kubernetes: Service Account Token Automounted
- Malas prácticas de Kubernetes: Shared Service Account Credentials
- Error de configuración de Kubernetes: Insecure etcd Client Transport
- Error de configuración de Kubernetes: Insecure etcd Peer Transport
- Error de configuración de Kubernetes: Missing Kubelet Certificate Authentication
- Error de configuración de Kubernetes: Missing Service Account Token Authentication
- Error de configuración de Kubernetes: Weak SSL Certificate for Kubelet

Claves y paquetes criptográficos externos

Las claves criptográficas se pueden almacenar en archivos separados del código fuente, pero persisten en un sistema de control de versiones. Además, las claves criptográficas se pueden almacenar en un paquete criptográfico, un archivo que almacena objetos criptográficos, como certificados y claves de cifrado. En esta versión, notificamos sobre la existencia de archivos como: *Key Management: Hardcoded Encryption Key*. Los paquetes criptográficos y los archivos clave compatibles incluyen:

- Estándares de criptografía de clave pública #12 KeyStore
- Java KeyStore, formato KeyStore de Oracle
- Claves maestras de Ruby On Rails
- Clave privada de PuTTY
- Clave de descifrado de Microsoft BitLocker

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:

Insecure Transport: Weak SSL Protocol

Secure Sockets Layer (SSL) y Transport Layer Security (TLS) proporcionan mecanismos para proteger los datos a través de las redes. Nesta versão, atualizamos o suporte para *Insecure Transport: Weak SSL Protocol*. Además de marcar el uso de cualquier versión de SSL, a partir de esta versión también marcamos el uso de las versiones 1.0 o 1.1 de TLS.

Insecure Transport: Weak SSL Cipher

Los conjuntos de cifrado especifican los algoritmos criptográficos utilizados con Secure Sockets Layer (SSL) o Transport Layer Security (TLS). Anteriormente notificados por Fortify WebInspect, Fortify Static Code Analyzer (SCA) ahora también notifica los resultados de *Insecure Transport: Weak SSL Cipher.*

Weak Cryptographic Signature

Una firma digital es una técnica utilizada para determinar la autenticidad e integridad de mensajes digitales. El algoritmo de firma digital (DSA, por sus siglas en inglés) ahora está obsoleto y ya no debe usarse. Esta versión incluye soporte para notificar una *Weak Cryptographic Signature* cuando se usa DSA en Java, Ruby y PHP.

Ligeras mejoras en los nodos

Mejoramos la compatibilidad con los paquetes de Node.js, incluyendo "net", "http", "https" y "os". Los clients pueden esperar resultados más precisos en las categorías *Cross-Site Scripting, Server-Side Request Forgery* y *System Information Leak* .

Mejoras en falsos positivos:

Se ha seguido trabajando con gran esfuerzo para eliminar los falsos positivos en esta versión. Además de las otras mejoras, los clientes disfrutarán de una eliminación adicional de falsos positivos en las áreas siguientes:

- Credential Management: Hardcoded API Credentials, al identificar tokens de acceso de GitHub.
- Cross-Site Scripting: Content Sniffing en aplicaciones Java
- Falsos positivos intermitentes para "Portability Flaw: Locale Dependent Comparison"
- Falsos positivos intermitentes para "OGNL Expression Injection: Double Evaluation"
- Password Management: Contraseña codificada cuando se establece dentro de un dominio de ejemplo, como ejemplo.com
- SQL Injection: iBatis Data Map

CyberRes 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:

Soporte de vulnerabilidad Inclusión de archivos peligrosos: Local

Grafana es una plataforma de código abierto para la monitorización y la observación. Algunas versiones de Grafana son vulnerables al salto de directorios según lo identificado por CVE-2021-43798. Esta vulnerabilidad permite el acceso a archivos locales. Los atacantes pueden obtener el contenido de los archivos en el servidor, lo que puede dar lugar a la divulgación de datos confidenciales y la posible recuperación de la lógica empresarial propia. Esta versión incluye una comprobación que permite detectar esta vulnerabilidad en Grafana.

Actualizaciones de la política Log4Shell Aggressive[1]

Se ha agregado una nueva política Log4Shell Aggressive a la lista de políticas admitidas de SecureBase. En comparación con las políticas existentes, puede realizar análisis más precisos, agresivos y decisivos para una evaluación de seguridad integral de las aplicaciones web que utilizan Log4j. Esto incluye *JNDI Reference Injections* en versiones vulnerables de las bibliotecas Apache Log4j.

Otras erratas

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

Log4Shell[1]

Esta versión incluye mejoras en la verificación de Log4Shell para añadir soporte para la nueva política Log4Shell Aggressive, que proporciona un escaneo más preciso para las *JNDI Reference Injections* en versiones vulnerables de las bibliotecas de Apache Log4j.

Actualización de CSRF

Esta versión mejora la comprobación de CSRF para reducir los falsos negativos y mejorar la precisión de los resultados.

---

[1] La verificación *Log4Shell* y la política *Log4Shell Aggressive* requieren el parche WebInspect 21.2.0.117 o una versión posterior.

CyberRes Fortify Premium Content

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

Taxonomía de CyberRes 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 CyberRes Fortify Support Portal.

Póngase en contacto con el soporte técnico de Fortify

CyberRes Fortify
http://softwaresupport.softwaregrp.com/
+1 (844) 260-7219

Póngase en contacto con SSR

Alexander M. Hoole
Director sénior de Software Security Research de CyberRes Fortify
hoole@microfocus.com
+1 (650) 258-5916

Peter Blay
Director de Software Security Research
CyberRes Fortify
peter.blay@microfocus.com