Anuncio de publicación de Software Security Research

Contenido de seguridad del software Fortify

Actualización 4 de 2022

viernes, 16 de diciembre de 2022

Acerca de CyberRes Fortify Software Security Research

El equipo de Fortify Software Security Research transforma la investigación más avanzada en inteligencia de seguridad que impulsa la cartera de productos de Fortify, que incluye Fortify Static Code Analyzer (SCA) y Fortify WebInspect. En la actualidad, el contenido de seguridad del software Fortify admite 1.286 categorías de vulnerabilidad en 30 lenguajes y abarca más de un millón de API distintas.

CyberRes
A Micro Focus Line of Business

Fortify Software Security Research (SSR) se complace en anunciar la disponibilidad inmediata de actualizaciones para Fortify Secure Coding Rulepacks (en inglés, versión 2022.4.0), Fortify WebInspect SecureBase (disponible mediante SmartUpdate) y Fortify Premium Content.

Fortify Secure Coding Rulepacks [Fortify Static Code Analyzer]

Con esta versión, Fortify Secure Coding Rulepacks detecta 1.066 categorías únicas de vulnerabilidades en 30 lenguajes de programación y abarca más de un millón de API distintas. En resumen, esta versión incluye lo siguiente:

Actualización de Flask (versión compatible: v2.2.x)

Flask es un marco micro web escrito en Python que no requiere un conjunto particular de herramientas o bibliotecas listas para usar. Es un marco ligero y bien establecido, generalmente más adecuado para proyectos pequeños y medianos, pero también capaz de manejar proyectos relativamente complejos, como API pequeñas y microservicios. Entre las categorías compatibles se incluyen:

- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: Reflected
- JSON Injection
- Often Misused: File Upload
- Open Redirect
- Path Manipulation
- Privacy Violation
- Server-Side Template Injection
- System Information Leak: External
- System Information Leak: Internal

Actualizaciones del SDK de iOS para Swift (versión compatible: 16)[1]

El SDK de iOS de Apple proporciona una recopilación de marcos que permiten a los desarrolladores crear aplicaciones móviles para dispositivos iPhone y iPad de Apple. Esta versión contiene actualizaciones adicionales que complementan nuestra compatibilidad con SDK de iOS para Swift. Las reglas nuevas y actualizadas amplían nuestra cobertura de API del marco Foundation en SDK 15 y 16 de iOS y para aplicaciones Swift iOS y iPadOS. Estas actualizaciones mejoran la detección de problemas para muchas categorías de debilidad existentes, entre otras:

- Insecure SSL: Overly Broad Certificate Trust
- Insecure Transport: Weak SSL Protocol
- Privacy Violation
- Resource Injection
- System Information Leak

[1] Las nuevas reglas para iOS SDK 16 requieren Fortify Static Code Analyzer 22.2 o una versión posterior.

Actualizaciones de Salesforce Apex y Visualforce (versión compatible: v55)[2]

Salesforce Apex es el lenguaje de programación utilizado para crear aplicaciones de Salesforce, como, por ejemplo, transacciones comerciales, administración de bases de datos, servicios web y páginas de Visualforce. Esta actualización mejora nuestro soporte en operaciones de Database, servicios web SOAP, servicios web REST, API del sistema principal de Apex, API criptográficas y componentes de página de Visualforce. Las categorías admitidas recientemente para Apex incluyen:

- Cookie Security: Cookie not Sent Over SSL
- Cookie Security: Missing SameSite Attribute
- Cookie Security: Overly Broad Path
- Cookie Security: Overly Permissive SameSite Attribute
- Cookie Security: Persistent Cookie
- Header Manipulation
- Header Manipulation: Cookies
- Insecure Transport
- Key Management: Hardcoded Encryption Key
- Log Forging (debug)
- Open Redirect
- Password Management: Empty Password
- Password Management: Hardcoded Password
- Path Manipulation
- Privacy Violation
- Server-Site Request Forgery
- System Information Leak: External
- System Information Leak: Internal
- Weak Cryptographic Hash
- Weak Encryption: Insecure Initialization Vector

Además, se han realizado mejoras adicionales en las siguientes categorías admitidas:

- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: Reflected

Mejoras en el escaneo de secretos

El escaneo de secretos es el concepto de encontrar secretos en varios códigos fuente y archivos de configuración. SSR ya es compatible con muchos tipos de secretos y SCA aplica las reglas de escaneo de secretos a todos los tipos de archivos, lo que permite encontrar secretos específicos independientemente del lenguaje de código. El soporte se expande para cubrir los siguientes secretos:

- Credential Management: Hardcoded API Credential, para tokens de portador codificados
- Password Management: Hardcoded Password, para contraseñas codificadas en cadenas de conexión en el servidor SQL
- Password Management: Password in Comment, para contraseñas en comentarios XML[3]

[2] Se requiere Fortify Static Code Analyzer 22.2 o una versión posterior.
[3] Se requiere Fortify Static Code Analyzer 22.2 o una versión posterior.

Cobertura inicial para Google Guava (versión compatible: v31.1)

Guava es un conjunto de bibliotecas Java de Google que incluye nuevos tipos de colección (como multimap y multiset), colecciones inmutables, una biblioteca de gráficos y utilidades para concurrencia, E/S, almacenamiento en caché hash, primitivos, cadenas y más. Es ampliamente utilizado en proyectos Java dentro de Google y otras empresas. Entre las categorías compatibles se incluyen:

- Null Dereference
- Path Manipulation
- Privacy Violation
- Setting Manipulation
- System Information Leak
- Unreleased Resource
- Unsafe Reflection
- Weak Cryptographic Hash
- Weak Cryptographic Hash: User-Controlled Minimum Bits
- Weak Cryptographic Hash: User-Controlled Seed
- Weak Encryption

Cobertura inicial para Hot Chocolate (versión admitida: 12.15.2)

Hot Chocolate es un servidor GraphQL de código abierto construido sobre la plataforma Microsoft .NET. Hot Chocolate permite a los desarrolladores crear e implementar rápidamente API basadas en GraphQL para sus aplicaciones. Esta versión agrega compatibilidad inicial con Hot Chocolate, incluida la detección de las siguientes categorías de debilidad en las API de GraphQL desarrolladas con Hot Chocolate:

- Cross-Site Scripting: Inter-Component Communication (Cloud)
- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: Reflected
- GraphQL Bad Practices: Introspection Enabled
- Privacy Violation
- System Information Leak: External
- Trust Boundary Violation

Expansión de gRPC para Java y cobertura inicial para Python (versión admitida: 1.49.1)

Google Remote Procedure Call (gRPC) es un marco RPC de alto rendimiento de código abierto moderno, multientorno y multilenguaje. gRPC conecta servicios con soporte para equilibrio de carga, seguimiento y autenticación. A diferencia de JSON tradicional sobre HTTP, gRPC se basa en HTTP2 y típicamente usa el formato de búfer de protocolo binario (protobuf) para los mensajes.

Se expandió el soporte para Java gRPC para cubrir las siguientes categorías adicionales:

- Access Control: gRPC Authentication Bypass
- Insecure SSL: Overly Broad Certificate Trust
- Log Forging
- Setting Manipulation
- Unreleased Resource: Streams

Se estableció el soporte para Python gRPC para cubrir las siguientes categorías:

- Insecure Transport
- Insecure Transport: gRPC Channel Credentials
- Insecure Transport: gRPC Server Credentials
- Privacy Violation
- System Information Leak: External

Infraestructura de la nube como código (IaC)

IaC es el proceso de administrar y aprovisionar recursos informáticos a través de código en lugar de varios procesos manuales. El soporte mejorado incluye configuraciones de Terraform para la implementación en AWS, Azure y GCP. Los problemas comunes relacionados con la configuración de estos servicios ahora se notifican al desarrollador.

Configuraciones de Terraform para Amazon AWS

Terraform es una herramienta de código abierto de infraestructura como código para construir, cambiar y versionar la infraestructura de la nube. Utiliza su propio lenguaje declarativo conocido como HashiCorp Configuration Language (HCL). La infraestructura de la nube está codificada en archivos de configuración para describir el estado deseado. Los proveedores de Terraform son compatibles con la configuración y administración de la infraestructura de Amazon Web Services (AWS). En esta versión, identificamos las siguientes categorías para las configuraciones de Terraform:

- AWS Terraform Misconfiguration: Amazon API Gateway Publicly Accessible
- AWS Terraform Misconfiguration: Amazon EBS Insecure Storage
- AWS Terraform Misconfiguration: Amazon ElastiCache Insecure Transport
- AWS Terraform Misconfiguration: Amazon MQ Publicly Accessible
- AWS Terraform Misconfiguration: Amazon Neptune Publicly Accessible
- AWS Terraform Misconfiguration: Amazon RDS Insecure Storage
- AWS Terraform Misconfiguration: Amazon RDS Proxy Insecure Transport
- AWS Terraform Misconfiguration: Amazon RDS Publicly Accessible
- AWS Terraform Misconfiguration: Amazon Redshift Publicly Accessible
- AWS Terraform Misconfiguration: Amazon SNS Insecure Storage

Configuraciones de Terraform para Microsoft Azure

Terraform es una herramienta de código abierto de infraestructura como código para construir, cambiar y versionar la infraestructura de la nube. Utiliza su propio lenguaje declarativo conocido como HashiCorp Configuration Language (HCL). La infraestructura de la nube está codificada en archivos de configuración para describir el estado deseado. Los proveedores de Terraform admiten la configuración y administración de la infraestructura de Microsoft Azure. En esta versión, identificamos las siguientes categorías para las configuraciones de Terraform:

- Azure Terraform Bad Practices: AKS Cluster Missing Host-Based Encryption
- Azure Terraform Bad Practices: Azure Disk Snapshot Missing Customer-Managed Key
- Azure Terraform Bad Practices: Azure MySQL Server Missing Infrastructure Encryption
- Azure Terraform Bad Practices: Azure PostgreSQL Server Missing Infrastructure Encryption
- Azure Terraform Bad Practices: Container Registry Missing Customer-Managed Key
- Azure Terraform Bad Practices: Cosmos DB Missing Customer-Managed Key
- Azure Terraform Bad Practices: Missing Azure Storage Infrastructure Encryption

- Azure Terraform Bad Practices: Missing SQL Database Backup Encryption
- Azure Terraform Bad Practices: Scale Set Missing Host-Based Encryption
- Azure Terraform Bad Practices: Shared Image Missing Customer-Managed Key
- Azure Terraform Bad Practices: SQL Database Missing Customer-Managed Key
- Azure Terraform Bad Practices: Storage Account Missing Customer-Managed Key
- Azure Terraform Bad Practices: Storage Encryption Scope Missing Customer-Managed Key
- Azure Terraform Bad Practices: VM Missing Host-Based Encryption
- Azure Terraform Misconfiguration: AKS Cluster Missing Customer-Managed Key
- Azure Terraform Misconfiguration: Managed Disk Missing Customer-Managed Key
- Azure Terraform Misconfiguration: Missing SQL Database Encryption
- Azure Terraform Misconfiguration: VM Storage Missing Customer-Managed Key

Configuraciones de Terraform para Google Cloud

Terraform es una herramienta de código abierto de infraestructura como código para construir, cambiar y versionar la infraestructura de la nube. Utiliza su propio lenguaje declarativo conocido como HashiCorp Configuration Language (HCL). La infraestructura de la nube está codificada en archivos de configuración para describir el estado deseado. Los proveedores de Terraform son compatibles con la configuración y administración de la infraestructura de Google Cloud Platform. En esta versión, identificamos las siguientes categorías de debilidad para las configuraciones de Google Cloud Platform Terraform:

- GCP Terraform Bad Practices: Apigee Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: BigQuery Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Cloud Bigtable Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Cloud Functions Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Cloud Spanner Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Filestore Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Pub/Sub Missing Customer-Managed Encryption Key
- GCP Terraform Bad Practices: Secret Manager Missing Customer-Managed Encryption Key
- GCP Terraform Misconfiguration: Compute Engine Missing Confidential Computing Features
- GCP Terraform Misconfiguration: Edge Cache Service Missing HTTP-to-HTTPS Redirect
- GCP Terraform Misconfiguration: Insecure App Engine Domain Transport
- GCP Terraform Misconfiguration: Insecure App Engine Transport
- GCP Terraform Misconfiguration: Insecure Cloud Function HTTP Trigger Transport
- GCP Terraform Misconfiguration: Insecure Edge Cache Service Transport
- GCP Terraform Misconfiguration: Insecure Supply Chain
- GCP Terraform Misconfiguration: URL Map Missing HTTP-to-HTTPS Redirect

Otras erratas

En esta versión, seguimos invirtiendo recursos para garantizar la reducción del número de falsos positivos, lograr una mejor consistencia 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:

Desuso para versiones de Fortify Static Code Analyzer anteriores a 19.x

Como se anunció en la presentación del lanzamiento de 2022.3, [enlace a la presentación del lanzamiento R3] esa fue la última versión de Rulepacks compatible con versiones de Fortify Static Code Analyzer anteriores a 19.x. En esta versión, las versiones de Fortify Static Code Analyzer anteriores a 19.x no cargarán las instancias 2022.4 de Rulepacks. Se deberá cambiar a una versión inferior de Rulepacks o actualizar la versión de Static Code Analyzer a 19.x o posterior. En las versiones futuras, continuaremos admitiendo las últimas cuatro versiones principales de Fortify Static Code Analyzer.

Refactorización de metadatos de orden de prioridad de Fortify para categorías de debilidad

Dado que es importante que nuestros usuarios puedan remediar los problemas de manera efectiva, hemos estado investigando mecanismos para determinar la categorización de los problemas de manera más objetiva dentro del modelo Fortify Priority Order. Como parte de este esfuerzo mencionado en el anuncio de lanzamiento de R3 2022 [enlace al anuncio de lanzamiento de R3], comenzamos a revisar las categorías que cubren todas nuestras reglas e identificamos algunas áreas donde se necesitaban actualizaciones. Las siguientes 96 categorías han cambiado sus metadatos de Fortify Priority Order asociados y, como tal, es posible que vea problemas que aparecen en el mismo grupo de gravedad o menor (p. ej., crítico, alto, medio, bajo). Los filtros existentes pueden hacer que se oculten problemas adicionales debido a los cambios en el valor de Fortify Priority Order y sus componentes individuales.

- Android Bad Practices: Encryption Secret Held in Static Field
- Android Bad Practices: Use of Released Camera Resource
- Android Bad Practices: Use of Released Media Resource
- Android Bad Practices: Use of Released SQLite Resource
- ASP.NET Bad Practices: Non-Serializable Object Stored in Session
- Buffer Overflow
- Buffer Overflow: Signed Comparison
- Code Correctness: Arithmetic Operation on Boolean
- Code Correctness: Function Not Invoked
- Code Correctness: Incorrect Serializable Method Signature
- Code Correctness: Memory Free on Stack Variable
- Code Correctness: Negative Content-Length
- Code Correctness: Premature Thread Termination
- Code Correctness: readObject() Invokes Overridable Function
- Code Correctness: Readonly Collection Reference
- ColdFusion Bad Practices: Unauthorized Include
- Cookie Security: Cookie not Sent Over SSL
- Cross-Site Scripting: Handlebars Helper
- Cross-Site Scripting: Inter-Component Communication
- Cross-Site Scripting: Inter-Component Communication (Cloud)
- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Reflected
- Cross-Site Scripting: Untrusted HTML Downloads
- Dangerous Method
- Denial of Service
- Denial of Service: Parse Double
- Denial of Service: Regular Expression

- Denial of Service: Stack Exhaustion
- Denial of Service: StringBuilder
- Double Free
- Dynamic Code Evaluation: Code Injection
- Dynamic Code Evaluation: Script Injection
- File Disclosure: Django
- File Disclosure: J2EE
- File Disclosure: Spring
- File Disclosure: Spring Webflow
- File Disclosure: Struts
- Format String: Argument Number Mismatch
- Header Manipulation: Cookies
- Header Manipulation: SMTP
- J2EE Bad Practices: Non-Serializable Object Stored in Session
- Log Forging
- Null Dereference[4]
- Often Misused: Authentication
- Often Misused: Boolean.getBoolean()
- Path Manipulation: Base Path Overwriting
- Path Manipulation: Zip Entry Overwrite
- Portability Flaw: File Separator
- Portability Flaw: Locale Dependent Comparison
- Privacy Violation
- Privacy Violation: Android Internal Storage
- Privacy Violation: BREACH
- Privacy Violation: Heap Inspection
- Privacy Violation: HTTP GET
- Privacy Violation: Image
- Privacy Violation: Keyboard Caching
- Privacy Violation: Screen Caching
- Privacy Violation: Sensitive Data Accessible From iTunes
- Privacy Violation: Shoulder Surfing
- Privacy Violation: Unobfuscated Logging
- Privilege Management: Android Disable
- Privilege Management: Missing API Permission
- Privilege Management: Missing Content Provider Permission
- Privilege Management: Missing Intent Permission
- Process Control
- Query String Injection: Android Provider
- Race Condition: PHP Design Flaw

[4] Los cambios en Null Dereference solo afectan a Java y .NET.

- Race Condition: Singleton Member Field
- Race Condition: Static Database Connection
- SOQL Injection
- SOSL Injection
- System Information Leak: Overly Broad SQL Logging
- System Information Leak: PHP Errors
- System Information Leak: PHP Version
- Unreleased Resource: Cursor Snarfing
- Unsafe JNI
- Unsafe JSNI
- Unsafe Mobile Code: Access Violation
- Unsafe Mobile Code: Database Access
- Unsafe Native Invoke
- Use After Free
- Weak Encryption: Insecure Initialization Vector
- Weak Encryption: Insecure Mode of Operation
- Weak Encryption: Insufficient Key Size
- Weak Encryption: Missing Required Step
- Weak Encryption: User-Controlled Key Size
- Weak XML Schema: Lax Processing
- Weak XML Schema: Type Any
- Weak XML Schema: Unbounded Occurrences
- Weak XML Schema: Undefined Namespace

React Bad Practices: Dangerously Set InnerHTML

El uso de "dangerouslySetInnerHTML" dentro de las aplicaciones React ahora se marca como una mala práctica.

Mejoras en falsos positivos

Se ha seguido trabajando con el fin de eliminar los falsos positivos en esta versión. Además de otras mejoras, los clientes pueden esperar una mayor eliminación de falsos positivos en las siguientes áreas:

- *Access Control: Database*: problemas eliminados en Apex cuando la entrada proviene de otra llamada de base de datos
- *ASP.NET MVC Bad Practices: Model With Required Non-Nullable Property* : se reducen los falsos positivos al usar atributos que aplicarían ciertas características de los datos
- *Dockerfile Misconfiguration: Dependency Confusion*: los falsos positivos se reducen cuando la imagen se crea para que sea mínima mediante la ampliación desde cero
- *GraphQL Bad Practices: Introspection Enabled* : se reducen los falsos positivos en las aplicaciones de Flask al registrar vistas de Flask basadas en clases
- *Dynamic Code Evaluation: JNDI Reference Injection*: los falsos positivos se reducen cuando los proyectos Spring Boot establecen la propiedad Maven 'log4j2.version' en una versión que no se ve afectada por Log4Shell
- *Memory Leak*: se reducen los falsos positivos cuando se usa std::unique_ptr
- *Mass Assignment: Insecure Binder Configuration*: los falsos positivos se reducen al usar la anotación JSONConverter con DataContract, DataMember o IgnoreDataMember

- *Privacy Violation*: se reducen los falsos positivos en los valores de enumeración en .NET
- *SQL Injection* : se reducen los falsos positivos cuando se utilizan declaraciones preparadas que contienen '$.' en las anotaciones de consulta de MyBatis
- *Unreleased Resource*: se reducen los falsos positivos en los análisis C/C++ cuando se agrupan recursos en una colección

Configuración incorrecta de PHP: categorías magic_quotes eliminadas

Las siguientes tres categorías de debilidad se han eliminado porque ya no son relevantes en las versiones compatibles de PHP:

- PHP Misconfiguration: magic_quotes_gpc Enabled
- PHP Misconfiguration: magic_quotes_runtime Enabled
- PHP Misconfiguration: magic_quotes_sybase Enabled

Como resultado, todos los problemas de las categorías anteriores se eliminarán de los resultados del escaneo.

Cambios de categoría

Junto con las eliminaciones de falsos positivos, identificamos algunos lugares en los que las categorías deberían haberse unificado o estaban mal etiquetadas. Cuando se producen cambios en el nombre de la categoría de debilidad, los resultados del escaneo al fusionar escaneos anteriores con nuevos escaneos darán como resultado categorías añadidas o eliminadas.

- *Code Correctness: Class Does Not Implement equals* ahora se notifica como *Code Correctness: Class Does Not Implement Equivalence Method*
- *Code Correctness: Class Does Not Implement Equals* ahora se notifica como *Code Correctness: Class Does Not Implement Equivalence Method*
- *Code Correctness: toString on Array* ahora se notifica como *Code Correctness: ToString on Array*
- *Code Correctness: null Argument to equals()* ahora se notifica como *Code Correctness: null Argument To Equivalence Method*
- *Code Correctness: null Argument to Equals()* ahora se notifica como *Code Correctness: null Argument To Equivalence Method*

Además, las siguientes 24 categorías de nuestro reciente soporte IaC se han refactorizado para mejorar la consistencia.

- *Access Control: Azure Container Registry* ahora se notifica como *Azure ARM Misconfiguration: Improper Container Registry Network Access Control*
- *Access Control: Azure SQL Database* ahora se notifica como *Azure ARM Misconfiguration: Improper SQL Server Network Access Control*
- *Access Control: Cosmos DB* ahora se notifica como *Azure ARM Misconfiguration: Improper DocumentDB Network Access Control*
- *Access Control: Kubernetes Admission Controller* ahora se notifica como *Kubernetes Bad Practices: Improper Admission Controller Access Control*
- *Access Control: Kubernetes Image Authorization Bypass* ahora se notifica como *Kubernetes Misconfiguration: Image Authorization Bypass*
- *Ansible Bad Practices: CloudWatch Log Group Retention Unspecified* ahora se notifica como *AWS Ansible Misconfiguration: Insufficient CloudWatch Logging*
- *Ansible Bad Practices: Redshift Publicly Accessible* ahora se notifica como *AWS Ansible Misconfiguration: Improper Redshift Network Access Control*

- *Ansible Bad Practices: Unrestricted AWS Lambda Principal* ahora se notifica como *AWS Ansible Misconfiguration: Improper Lambda Access Control Policy*
- *Ansible Bad Practices: User-Bound AWS IAM Policy* ahora se notifica como *AWS Ansible Bad Practices: Improper IAM Access Control Policy*
- *Ansible Misconfiguration: Azure Monitor Missing Administrative Events* ahora se notifica como *Azure Ansible Misconfiguration: Insufficient Azure Monitor Logging*
- Azure Resource Manager Bad Practices: *Cross-Tenant Replication* ahora se notifica como *Azure ARM Misconfiguration:* Improper Storage Account Network Access Control
- *Azure Resource Manager Bad Practices: Remote Debugging Enabled* ahora se notifica como *Azure ARM Misconfiguration: Improper App Service Access Control*
- *Azure Resource Manager Bad Practices: SSH Password Authentication* ahora se notifica como *Azure ARM Misconfiguration: Improper Compute VM Access Control*
- *Azure Resource Manager Misconfiguration: Insecure Transport* ahora se notifica como *Azure ARM Misconfiguration: Insecure App Service Transport*
- *Azure Resource Manager Misconfiguration: Overly Permissive CORS Policy* ahora se notifica como *Azure ARM Misconfiguration: Improper CORS Policy*
- *Azure Resource Manager Misconfiguration: Security Alert Disabled* ahora se notifica como *Azure ARM Misconfiguration: Insufficient Microsoft Defender Monitoring*
- *Azure SQL Database Misconfiguration: Insufficient Logging* ahora se notifica como *Azure ARM Misconfiguration: Insufficient SQL Server Monitoring*
- *Insecure Storage: Missing EC2 AMI Encryption* ahora se notifica como *AWS CloudFormation Misconfiguration: Insecure EC2 AMI Storage*
- *Insecure Storage: Missing EFS Encryption* ahora se notifica como *AWS CloudFormation Misconfiguration: Insecure EFS Storage*
- *Insecure Storage: Missing Kinesis Stream Encryption* ahora se notifica como *AWS CloudFormation Misconfiguration: Insecure Kinesis Data Stream Storage*
- *Insecure Transport: Azure App Service* ahora se notifica como *Azure Ansible Misconfiguration: Insecure App Service Transport*
- *Kubernetes Bad Practices: API Server Publicly Accessible* ahora se notifica como *Azure ARM Misconfiguration: Improper AKS Network Access Control*
- *Privacy Violation: Exposed Default Value* ahora se notifica como *Azure ARM Misconfiguration: Hardcoded Secret*
- *Privilege Management: Overly Permissive Role* ahora se notifica como *Azure ARM Misconfiguration: Improper Custom Role Access Control Policy*

Fortify SecureBase [Fortify WebInspect]

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

Compatibilidad de vulnerabilidades

Server-Side Request Forgery[5]

Server-Side Request Forgery (SSRF) ocurre cuando un atacante puede influir en una conexión de red realizada por el servidor de aplicaciones. La conexión de red se originará desde la IP interna del servidor de aplicaciones y un atacante puede usar esta conexión para eludir los controles de red y escanear o atacar recursos internos que de otro modo no estarían expuestos. Esta versión incluye una comprobación que permite detectar las vulnerabilidades de SSRF en aplicaciones web que aceptan entradas de usuario.

Expression Language Injection[6]

CVE-2022-42889 identificó una vulnerabilidad crítica de ejecución remota de código en la popular biblioteca Apache Commons Text, versiones 1.5 a 1.9. La configuración predeterminada podría permitir la evaluación de secuencias de comandos no seguras y la ejecución de código arbitrario. Esta versión incluye una comprobación que permite detectar la vulnerabilidad CVE-2022-42889 en los servidores web de destino. Debido a que esta verificación envía una cantidad significativa de solicitudes, se excluye de la política estándar. Utilice la política Todas las comprobaciones, personalice una política existente para incluir la comprobación o cree una política personalizada para ejecutar esta comprobación.

Insecure Transport: Weak SSL Cipher

Los protocolos Transport Layer Security (TLS) y Secure Sockets Layer (SSL) proporcionan un mecanismo para ayudar a proteger la autenticidad, confidencialidad e integridad de los datos transmitidos entre un cliente y un servidor web. El uso de un cifrado débil o una clave de cifrado de longitud insuficiente, por ejemplo, podría permitir que un atacante derrote el mecanismo de protección y robe o modifique información confidencial. Esta versión incluye una nueva verificación identificada por ID 11285, para marcar la vulnerabilidad Protección insuficiente de la capa de transporte: Cifrado inseguro con gravedad crítica. Los conjuntos de cifrado inseguros tienen múltiples vulnerabilidades conocidas y tienen ataques que son triviales.

Otras erratas

En esta versión hemos invertido recursos para reducir aún más el número de falsos positivos y para mejorar la capacidad de auditar problemas por parte de los clientes. Los clientes también verán cambios en los resultados comunicados en relación con lo siguiente:

Insecure Transport: Weak SSL Cipher

---

[5] Requiere funciones OAST que están disponibles en el parche WebInspect 21.2.0.117 o una versión posterior.
[6] Requiere funciones OAST que están disponibles en el parche WebInspect 21.2.0.117 o una versión posterior.

Esta versión incluye mejoras en Insufficient Transport Layer Protection – Weak Cipher check (11716). Los clientes deberían ver que la severidad de esta verificación se reduce de Crítico a Alto porque los ataques contra estos cifrados débiles son sofisticados y requieren recursos considerables. La gravedad de esta verificación se reduce y se introdujo una nueva verificación para identificar "Insufficient Transport Layer Protection – Insecure Cipher" que marcará los problemas con gravedad "Crítica". En el futuro, los cifrados recomendados no incluirán conjuntos de cifrado que no tengan Perfect Forward Secrecy (PFS).

XML External Entity Injection[7]

La comprobación identificada por ID 11337 se modificó para usar cargas útiles que admitan la función de pruebas de seguridad de aplicaciones fuera de banda (OAST). Las mejoras de esta comprobación reducen los falsos positivos y aumentan la eficiencia y la precisión de sus resultados.

Fortify Premium Content

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

Fortify Taxonomy: errores en la seguridad del software

El Fortify Taxonomy sitio 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 el Fortify Support Portal.

[7] Requiere funciones OAST que están disponibles en el parche WebInspect 21.2.0.117 o una versión posterior.

Comuníquese con el soporte técnico de Fortify

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

Comuníquese con SSR

Alexander M. Hoole
Director sénior del Equipo de Investigación de seguridad para software
CyberRes Fortify
hoole@microfocus.com
+1 (650) 258-5916

Peter Blay
Director del Equipo de Investigación de seguridad para software
CyberRes Fortify
peter.blay@microfocus.com

Copyright 2023 Open Text. The only warranties for products and services of Open Text and its affiliates and licensors (“Open Text”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.