Software Security Research Release Announcement

Fortify Software Security Content

2022 Update 1

March 25, 2022

About CyberRes Fortify Software Security Research

The Fortify Software Security Research team translates cutting-edge research into security intelligence that powers the Fortify product portfolio – including Fortify Static Code Analyzer (SCA), Fortify WebInspect, and Fortify Application Defender. Today, CyberRes Fortify Software Security Content supports 1,166 vulnerability categories across 29 languages and spans more than one million individual APIs.

Fortify Software Security Research (SSR) is pleased to announce the immediate availability of updates to Fortify Secure Coding Rulepacks (English language, version 2022.1.0), Fortify WebInspect SecureBase (available via SmartUpdate), and Fortify Premium Content.

CyberRes Fortify Secure Coding Rulepacks [SCA]

With this release, the Fortify Secure Coding Rulepacks detect 946 unique categories of vulnerabilities across 29 programming languages and span over one million individual APIs. In summary, this release includes the following:

Log4j updates (Version supported: 2.17)

Log4j is a popular logging framework for Java that has come under scrutiny in recent months due to high profile vulnerabilities discovered within the framework. This release includes improved support to identify exactly which parts of your source code are susceptible to the Log4Shell vulnerability, flagging them under the category *Dynamic Code Evaluation: JNDI Reference Injection.*
Additionally, the upgraded Log4j support covers the latest versions of Log4j for the following namespace:

- org.apache.logging.log4j

Support also improves coverage in the following weakness categories:

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

Azure Functions (Python, Version supported: 3.10.x)

Azure Functions is a serverless cloud computing solution that can execute code in response to pre-defined events, such as API calls, database transactions, or manage message queues in other Azure services. In this release, we expanded the support for Azure Functions to cover HTTP Trigger functions in Python. The HTTP trigger helps to invoke a function with an HTTP request and can be used to build serverless APIs and respond to webhooks.
Support includes coverage of the following categories:

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

GraphQL support: Python Graphene (Version supported: 3.0.0)

This release includes initial GraphQL server support for Python Graphene. GraphQL is an open-source project developed by Facebook that features a strongly-typed query language and a server-side runtime engine for APIs. GraphQL has been an open standard since 2015 and is currently supported by more than two dozen programming languages. Graphene is a popular GraphQL server framework for Python applications. This release adds the following two categories to detect weaknesses in GraphQL APIs developed with Graphene:

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

Kotlin updates (Version supported: 1.5)

Kotlin is a general-purpose, statically-typed language featuring Java interoperability. This release includes updated support for standard library APIs introduced in Kotlin 1.5 targeting the Java Virtual Machine (JVM).

Sequelize (Version supported: 6.17)

Sequelize is a promise-based Object-Relational Mapping (ORM) tool designed to simplify working with many popular SQL dialects within Node.js applications. Support includes coverage of the following categories:

- Access Control: Database
- Password Management: Empty Password
- Password Management: Hardcoded Password
- Password Management: Null Password
- SQL Injection

Insecure referenced files in HTML

All references to third-party sites within web pages should be over a secure connection, and therefore this release includes support for the following new categories within HTML files:

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

Shared password database detection

A password database is a file, or set of files that are made to securely store passwords. Password databases are typically encrypted using a master password or master key. However, they should not be used to persist password usage within an application through the development lifecycle. In this release, we report the existence of such databases as: *Password Management: Shared Password Database*. Supported password databases include:

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

Cloud Infrastructure as Code

This release includes expanded support for cloud Infrastructure as Code (IaC). Infrastructure as code is the process of managing and provisioning computer resources through code, rather than manual processes. Technologies supported include AWS, AWS CloudFormation, Azure ARM, Kubernetes K8S, and Azure Kubernetes Service. Common issues related to the configuration of the services mentioned are now reported to the developer.

Additional categories supported include:

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

External cryptographic keys and bundles

Cryptographic keys can be stored in files separate from the source code, but persisted in a version control system. In addition, cryptographic keys can be stored in a Cryptographic bundle, a file that stores cryptographic objects, such as certificates and encryption keys. In this release, we report the existence of such files as: *Key Management: Hardcoded Encryption Key*. Supported Cryptographic bundles and key files include:

- Public-Key Cryptography Standards #12 KeyStore
- Java KeyStore, Oracle's KeyStore format
- Ruby On Rails master keys
- PuTTY Private Key
- Microsoft BitLocker decryption key

Miscellaneous Errata

In this release, we have continued to invest resources to ensure we can reduce the number of false positive issues and improve the ability for customers to audit issues. Customers can also expect to see changes in reported issues related to the following:

Insecure Transport: Weak SSL Protocol

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) provide mechanisms to protect data over networks. In this release, we updated support for *Insecure Transport: Weak SSL Protocol*. In addition to flagging the use of any version of SSL, starting with this release, we also flag the use of TLS versions 1.0 or 1.1.

Insecure Transport: Weak SSL Cipher

Cipher suites specify the cryptographic algorithms used with Secure Sockets Layer (SSL) or Transport Layer Security (TLS). Previously reported by Fortify WebInspect, *Insecure Transport: Weak SSL Cipher* results are now reported by Fortify Static Code Analyzer (SCA) as well.

Weak Cryptographic Signature

A digital signature is a technique used to determine the authenticity and integrity of digital messages. The Digital Signature Algorithm (DSA) is now obsolete and should no longer be used. This release includes support to flag *Weak Cryptographic Signature* when DSA is used in Java, Ruby, and PHP.

Minor Node improvements

We improved support for Node.js packages including 'net', 'http', 'https', and 'os'. Customers can expect more accurate findings in *Cross-Site Scripting, Server-Side Request Forgery,* and *System Information Leak* categories.

False Positive improvements:

Work has continued with the effort to remove false positives in this release. In addition to other improvements, customers can expect further removal of false positives in the following areas:

- Credential Management: Hardcoded API Credentials, when identifying GitHub access tokens
- Cross-Site Scripting: Content Sniffing in Java applications
- Intermittent false positives for "Portability Flaw: Locale Dependent Comparison"
- Intermittent false positives for "OGNL Expression Injection: Double Evaluation"
- Password Management: Hardcoded Password, when set within an example domain, such as example.com
- SQL Injection: iBatis Data Map

CyberRes Fortify SecureBase [Fortify WebInspect]

Fortify SecureBase combines checks for thousands of vulnerabilities with policies that guide users in the following updates available immediately via SmartUpdate:

Vulnerability Support

Dangerous File Inclusion: Local

Grafana is an open-source platform for monitoring and observability. Some versions of Grafana are vulnerable to directory traversal as identified by CVE-2021-43798. This vulnerability allows access to local files. Attackers might obtain the contents of files on the server, which can lead to sensitive data disclosure and potential recovery of proprietary business logic. This release contains a check to detect this vulnerability in Grafana.

Policy Updates

Aggressive Log4Shell[1]

A new Aggressive Log4Shell policy has been added to the SecureBase list of supported policies. Compared to existing policies, it can perform more accurate, aggressive, and decisive scans for a comprehensive security assessment of web applications that use Log4j. This includes *JNDI Reference Injections* in vulnerable versions of Apache Log4j libraries.

Miscellaneous Errata

In this release, we have continued to invest resources to reduce the number of false positive issues and improve the ability for customers to audit issues. Customers can also expect to see changes in reported issues related to the following:

Log4Shell[1]

This release includes improvements in the Log4Shell check to add support for the new Aggressive Log4Shell policy, which provides more accurate scanning for *JNDI Reference Injections* in vulnerable versions of Apache Log4j libraries.

CSRF Update

This release includes improvements for the CSRF check to reduce false negatives and improve the accuracy of results.

---

[1] The *Log4Shell* check and the *Aggressive Log4Shell* policy require the WebInspect 21.2.0.117 patch or later.

CyberRes Fortify Premium Content

The research team builds, extends, and maintains a variety of resources outside our core security intelligence products.

CyberRes Fortify Taxonomy: Software Security Errors

The Fortify Taxonomy site, which contains descriptions for newly added category support, is available at https://vulncat.fortify.com. Customers looking for the legacy site, with the last supported update, can obtain it from the CyberRes Fortify Support Portal.

Contact Fortify Technical Support

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

Contact SSR

Alexander M. Hoole
Senior Manager, Software Security Research
CyberRes Fortify
hoole@microfocus.com
+1 (650) 258-5916

Peter Blay
Manager, Software Security Research
CyberRes Fortify
peter.blay@microfocus.com