Software Security Research Release Announcement

# Fortify Software Security Content

## 2022 Update 3

## September 30, 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) and Fortify WebInspect. Today, Fortify Software Security Content supports 1,244 vulnerability categories across 30 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 2022.3.0), Fortify WebInspect SecureBase (available via SmartUpdate), and Fortify Premium Content.

## Fortify Secure Coding Rulepacks [Fortify Static Code Analyzer]

With this release, the Fortify Secure Coding Rulepacks detect 1,024 unique categories of vulnerabilities across 30 programming languages and span over one million individual APIs. In summary, this release includes the following:

### ASP.NET Core Updates (version supported: 6.0)[1]

In the Model-View-Controller (MVC) pattern, views are *.cshtml* files that use the C# programming language embedded in Razor markup. Razor markup is code that interacts with HTML markup to produce a webpage sent to the client. Views handle the application's data presentation and user interaction. Using Fortify Static Code Analyzer version 22.2.0 and later, rules now support finding issues within views.

Support includes coverage of the following weakness categories:

- ASP.NET MVC Bad Practices: Form Without AntiForgery Token
- Cross-Site Scripting: Inter-Component Communication (Cloud)
- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: Reflected
- Open Redirect
- Privacy Violation
- System Information Leak

### Entity Framework Core (version supported: 6.0)

Entity Framework (EF) Core is an open-source data access technology for .NET applications. EF Core allows developers to map .NET objects to database schemas and invoke database operations through standard APIs and LINQ queries. Support includes coverage of the following weakness categories:

- Access Control: Database
- ASP.NET Bad Practices: Leftover Debug Code
- Connection String Parameter Pollution
- Insecure Transport: Database
- Password Management: Hardcoded Password
- Setting Manipulation
- SQL injection
- System Information Leak: Overly Broad SQL Logging

[1] Requires Fortify Static Code Analyzer version 22.2.0 or later.

### GitHub Actions

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows for automation of build, test, and deployment pipelines. Recent weaknesses have come to light that result in command injection attack vectors across a variety of systems. This release includes coverage to detect common instances of this command injection weakness under the following category:

- Command Injection: GitHub Actions

### React (version supported: 18.2)[2]

React, or ReactJS, is an open-source JavaScript library for building component-based user interfaces. While no new weakness categories are supported in this release, coverage has been refactored for React to be more accurate and reduce false positives.

### React Native (version supported: 0.70) [2]

React Native is an open-source UI framework for developing multiplatform user interfaces in JavaScript and JSX. React Native enables developers to write mobile applications that are rendered by the target platforms native rendering APIs to produce a polished and consistent user experience. In addition to the weakness categories supported for React, the following weakness categories are added for React Native:

- Open Redirect
- Privacy Violation
- System Information Leak: Internal

### React Native Async Storage (version supported: 1.17) [2]

Async Storage is an unencrypted, asynchronous, key-value storage library for React Native based upon the community *react-native-async-storage project*. Async Storage provides an abstraction on top of native iOS and Android platform specific storage mechanisms. Support enables dataflow through Async Storage and reporting of existing JavaScript and platform/library specific weakness categories.

### Secret Scanning Improvements

Secret scanning is the concept of finding secrets in various source code and configuration files. Fortify Static Code Analyzer applies the secret scanning coverage to all file types, which allows for finding specific secrets regardless of code language. Support for the following secrets has been added and are reported as *Password Management: Hardcoded Password* or *Credential Management: Hardcoded API Credentials*:

- HTTP Basic authentication tokens
- JWT (JSON Web Tokens)
- NPM (Node Package Manager) access tokens
- Postman API keys
- PyPI API token

---

[2] Requires Fortify Static Code Analyzer version 22.2.0 or later.

### Initial gRPC Support for Java and Go (version supported: 1.49.0)

Google Remote Procedure Call (gRPC) is a modern multi-environment and multi-language open-source high performance RPC framework. gRPC connects services with support for load balancing, tracing, and authentication. Unlike traditional JSON-over-HTTP, gRPC is based on HTTP2 and normally uses the binary Protocol Buffers (protobuf) format for messages. For gRPC projects, users should include the code generated from the .proto file definitions during the translation phase of Fortify Static Code Analyzer.

Support has been added for Go gRPC v1.49.0 to cover the following weakness categories:

- Header Manipulation
- Privacy Violation
- System Information Leak: External

Support has been added for Java gRPC v1.49.0 to cover the following weakness categories:

- Denial of Service
- gRPC Metadata Manipulation
- Insecure Transport
- Insecure Transport: gRPC Server Credentials
- Insecure Transport: gRPC Channel Credentials
- Privacy Violation
- Resource Injection
- System Information Leak: External

### Initial Flask Support (version supported: 2.2.x)

Flask is a web framework written in Python. Initially a wrapper for *Werkzeug* and *Jinja* libraries, Flask has become one of the most popular Python web application frameworks. To complement our Google Cloud Functions support for Python, this release contains support for the Flask Response objects only.
Support includes coverage of the following weakness categories:

- Cookie Security: Cookie not Sent Over SSL
- Cookie Security: HTTPOnly not Set
- Cookie Security: Missing SameSite Attribute
- Cookie Security: Overly Broad Domain
- Cookie Security: Overly Broad Path
- Cookie Security: Overly Permissive SameSite Attribute
- Cookie Security: Persistent Cookie
- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: Reflected
- Header Manipulation
- Header Manipulation: Cookies
- HTML5: Misconfigured Content Security Policy
- HTML5: Overly Permissive Content Security Policy
- HTML5: Overly Permissive CORS Policy
- HTML5: Unenforced Content Security Policy
- Open Redirect

- Privacy Violation
- System Information Leak: External

### Google Cloud Functions (version supported: 403.0.0)

Google Cloud Functions is a serverless execution environment for building and connecting cloud services. It can execute code in response to pre-defined events, such as API calls, database transactions, file upload to Cloud Storage, or an incoming message on a Pub/Sub topic.
Cloud Functions offers two product versions: Cloud Functions (1st gen), the original version, and Cloud Functions (2nd gen), a new version built on *Cloud Run* and *Eventarc* to provide an enhanced feature set. This release includes support for Google Cloud Functions in Python and updated support for Google Cloud Functions in Java.

Weakness categories supported for Python include those supported by Flask APIs, along with the following:

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

For Python Google Cloud Functions, users should either include the JSON or YAML cloud build file. Alternatively, users can set the following properties at scan-time:

- *com.fortify.sca.rules.GCPFunctionName* should be set to the function name.
- *com.fortify.sca.rules.GCPHttpTrigger* should be set to `true` if the trigger type is HTTP, `false` for other trigger types.

Updated rules support for 2nd gen Java Google Cloud Functions identifies sources of dangerous input originating from CloudEvents requests.

### Initial Apollo Server Support (version supported: 3.6.8)

Apollo Server is an open-source GraphQL server used in JavaScript applications to build GraphQL APIs. This release adds initial GraphQL server support for Apollo Server, including detection of the following weakness categories in GraphQL APIs developed with Apollo Server:

- Cross-Site Scripting: Persistent
- Cross-Site Scripting: Poor Validation
- Cross-Site Scripting: Reflected
- GraphQL Bad Practices: Introspection Enabled
- Privacy Violation
- System Information Leak: External

### Infrastructure as Code (IaC)

IaC is the process of managing and provisioning computer resources through code rather than various manual processes. Supported technologies include Terraform configurations for deployment to GCP, OpenAPI Specification, and MuleSoft. Common issues related to the configuration of these services are now reported to the developer.

### Google Cloud Platform (GCP) Terraform Configurations

Terraform is an open-source IaC tool for building, changing, and versioning cloud infrastructure. It uses its own declarative language known as HashiCorp Configuration Language (HCL). Cloud infrastructure is codified in configuration files to describe the desired state. Terraform providers support the configuration and management of GCP infrastructure. This release includes coverage of the following weakness categories for GCP Terraform configurations:

- GCP Terraform Misconfiguration: Cloud SQL Database Publicly Accessible
- GCP Terraform Misconfiguration: Cloud Storage Bucket Uniform Access Disabled
- GCP Terraform Misconfiguration: Compute Engine IP Forwarding Enabled
- GCP Terraform Misconfiguration: Compute Engine Serial Console Enabled
- GCP Terraform Misconfiguration: Compute Engine Shielded VM Option Disabled
- GCP Terraform Misconfiguration: GKE Cluster Node Auto-Repair Disabled
- GCP Terraform Misconfiguration: GKE Cluster Publicly Accessible
- GCP Terraform Misconfiguration: Overly Permissive Role
- GCP Terraform Misconfiguration: Permissive Firewall

### OpenAPI Specification

The OpenAPI specification defines a standard, programming language-agnostic description for HTTP APIs. OpenAPI documents that conform to the OpenAPI specification can be represented either in a JSON or YAML format. This standard defines the capabilities of a service without access to the implementation, documentation, or through network inspection. This release includes coverage of the following weakness categories for OpenAPI configurations:

- OpenAPI Misconfiguration: Credential Leakage
- OpenAPI Misconfiguration: Empty Global Security Requirement
- OpenAPI Misconfiguration: Empty Operation Security Requirement
- OpenAPI Misconfiguration: Insecure Transport
- OpenAPI Misconfiguration: Missing Error Handling
- OpenAPI Misconfiguration: Missing Global Security Requirement
- OpenAPI Misconfiguration: Missing Operation Security Requirement
- OpenAPI Misconfiguration: Missing Security Schemes
- OpenAPI Misconfiguration: Optional Global Security Requirement
- OpenAPI Misconfiguration: Optional Operation Security Requirement
- OpenAPI Misconfiguration: Weak Authentication

### Mule

Mule Runtime, often referred to as simply Mule, is an enterprise service bus and integration framework provided by MuleSoft. Mule enables integrations of existing systems such as Web Services, HTTP, Java Database Connectivity (JDBC), and more. Mule allows different applications to communicate with each other by acting as a transit system between applications within an enterprise network or across the internet. This release includes coverage of the following weakness categories for Mule configurations:

- Mule Misconfiguration: Hardcoded Password
- Mule Misconfiguration: Insecure Database Transport
- Mule Misconfiguration: Insecure Transport
- Mule Misconfiguration: Server Identity Verification Disabled

## 2022 CWE Top 25

The Common Weakness Enumeration (CWE™) Top 25 Most Dangerous Software Weaknesses (CWE Top 25) was introduced in 2019 and replaces SANS Top 25. Released in June, the 2022 CWE Top 25 was determined using a heuristic formula that normalizes the frequency and severity of vulnerabilities reported to the National Vulnerability Database (NVD) over the past two years. To support our customers who want to prioritize their auditing around the most commonly reported critical vulnerabilities in the NVD, a correlation of the CyberRes Fortify Taxonomy to the 2022 CWE Top 25 has been added.

## Miscellaneous Errata

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

### *Deprecation of Fortify Static Code Analyzer Versions Prior to 19.x*

As observed with the 2021.4 release, we are continuing to support the last four major releases of Fortify Static Code Analyzer. Therefore, this will be the last release of the Rulepacks that support Fortify Static Code Analyzer versions prior to 19.x. For the next release, Fortify Static Code Analyzer versions prior to 19.x will not load the most recent Rulepacks. This will require either downgrading the Rulepacks or upgrading the version of Fortify Static Code Analyzer. For future releases, we will continue to support the last four major releases of Fortify Static Code Analyzer.

### *Renaming of Infrastructure as Code (IaC) Weakness Categories*

As support for detecting misconfigurations and bad practices related to IaC continues to mature, our next release of security content will include category name changes to a subset of the weakness categories (2022 Update 4). When weakness category name changes occur, scan results when merging prior scans with new scans will result in added/removed categories.

### *Refactoring of Fortify Priority Order Metadata for Weakness Categories*

As the application security domain continues to mature, our collective knowledge and understanding of the impact of weakness categories to confidentiality, integrity, and availability evolves. Our next release of security content will include changes to weakness metadata fields “accuracy” and “impact” for a subset of weakness categories (2022 Update 4). When weakness metadata field changes occur, future scan results may have issues appearing in different filter set folders (e.g., critical, high, medium, low). The initial updates will cause some issues to move from higher Fortify Priority Order (FPO) folders to lower FPO folders. Customers should be prepared for how this change can impact existing filter sets and templates.

### *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:

- *Cross-Site Request Forgery* – false positives removed in .NET applications using versions of .NET Framework later than 4.5.2
- *JavaScript Hijacking* – issues (see section below)

- *Key Management* – false positives reduced across JavaScript scans
- *Key Management* – false positives reduced that primarily affect SAPUI5 projects
- *Key Management* – issues based on comparisons produced a lot of false positives and have been removed
- *Password Management: Hardcoded/Empty/Null Password* – false positives prevented for C# conditional statements
- *Password Management* – false positives reduced from NPM, Yarn, and Bower files
- *Privacy Violation: Autocomplete* – false positives reduced when setting new passwords
- *Setting Manipulation* – false positives reduced when clearing environment variables
- *Weak Cryptographic Signature* – false positives prevented in java.security package
- *XML Entity Expansion Injection* – false positives reduced in Java programs using JAXP transformers

### *JavaScript Hijacking Removal*

The following categories are no longer relevant in modern ECMAScript and were removed:

- JavaScript Hijacking
- JavaScript Hijacking: Constructor Poisoning
- JavaScript Hijacking: Vulnerable Framework

As a result, all issues from the above categories will be removed from scan results.

### *Category Changes*

Along with false positive removals, we identified some places that categories should have been unified or were mislabeled. When weakness category name changes occur, scan results when merging prior scans with new scans will result in added/removed categories.

- *Insecure SSL: Android Hostname Verification Disabled* now reports as *Insecure SSL: Server Identity Verification Disabled*
- In Dockerfiles, *Password Management: Hardcoded Password* issues are now reported as *Password Management: Password in Configuration Files*
- In .NET, some instances of *Setting Manipulation* when setting a database connection string are now reported as *Connection String Parameter Pollution*

## 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

### Insecure Deployment: Unpatched Application

dotCMS is a Content Management System that provides the ability to create and reuse content, images, and assets in one centralized location. The ContentResource API is susceptible to a remote code execution (RCE) vulnerability identified by CVE-2022-26352. The file name used to store the content is constructed from user input provided in the multipart request and is not sanitized by dotCMS. It enables

an attacker to upload arbitrary files on the system, resulting in RCE. This release includes a check to detect this vulnerability on a target server that runs affected dotCMS versions.

### Insecure Deployment: Unpatched Application

Apache APISIX is an open-source API gateway that provides traffic management features such as load balancing, dynamic upstream, and more. This API gateway is susceptible to an RCE vulnerability identified by CVE-2022-24112. An attacker can bypass IP restrictions on Apache APISIX through the batch-request plugin. If APISIX uses a default Admin key, with Admin API enabled and no custom admin port assigned, an attacker can invoke the Admin API via the batch-requests plugin, resulting in RCE. This release includes a check to detect this vulnerability on target server that runs affected Apache APISIX versions.

### Dynamic Code Evaluation: JNDI Reference Injection[3]

Java Naming and Directory Interface (JNDI) is a Java API that enables clients to discover and look up data and objects by name. These objects can be stored and retrieved through different naming or directory services such as Remote Method Invocation (RMI), Common Object Request Broker Architecture (CORBA), Lightweight Directory Access Protocol (LDAP), or Domain Name Service (DNS). If attackers obtain control of the argument to a JNDI lookup operation, they could point the lookup to a Naming or Directory service under their control and return a JNDI reference that uses a remote factory for object instantiation. This attack can enable execution of arbitrary remote code on the target server that performs the lookup operation. This release includes a check to detect this vulnerability on target web servers.

### Dynamic Code Evaluation: Unsafe Deserialization[3]

A pre-authorization insecure Java deserialization vulnerability in ADF Faces components of Oracle Fusion Middleware versions 12.2.1.3.0 and 12.2.1.4.0 has been identified by CVE-2022-21445. It impacts all applications that rely on ADF Faces components, including Business Intelligence, Enterprise Manager, Identity Management, SOA Suite, WebCenter Portal, Application Testing Suite, and Transportation Management. This issue enables attackers to execute arbitrary code on the server, abuse application logic, or mount Denial of Service (DoS) attacks. This release includes a check to detect this vulnerability on target web servers.

## Compliance Reports

### *2022 CWE Top 25*

The Common Weakness Enumeration (CWE™) Top 25 Most Dangerous Software Weaknesses (CWE Top 25) was introduced in 2019 and replaces SANS Top 25. Released in June, the 2022 CWE Top 25 is determined using a heuristic formula that normalizes the frequency and severity of vulnerabilities reported to the National

[3] Requires OAST features that are available in the WebInspect 21.2.0.117 patch or later.

Vulnerability Database (NVD) over the past two years. This SecureBase update includes checks that map either directly to the category identified by the CWE Top 25, or a CWE-ID related to a CWE-ID in the Top 25 via "ChildOf" relationship.

### Policy Updates

#### 2022 CWE Top 25

A policy customized to include checks relevant to 2022 CWE Top 25 has been added to the WebInspect SecureBase list of supported policies.

#### Miscellaneous errata

In this release, resources have been invested to further reduce the number of false positives and improve the ability for customers to audit issues. Customers can also expect to see changes in reported findings related to the following:

##### Dynamic Code Evaluation: Unsafe Deserialization[4]

The check identified by ID 11504 has been modified to use payloads that support the OAST feature. Improvement of this check reduces false positives and increases the efficiency and accuracy of its results.

## Fortify Premium Content

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

#### 2022 CWE Top 25

To accompany the new correlations, this release also contains a new report bundle for Fortify Software Security Center with support for the 2022 CWE Top 25, which is available for download from the Fortify Customer Support Portal under Premium Content.

#### 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 Fortify Support Portal.

[4] Requires OAST features that are available in the WebInspect 21.2.0.117 patch or later.

## 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