Fortify Software Security Content

2023 Update 1
March 31, 2023

About OpenText 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 1,399 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 2023.1.0), 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 1,177 unique categories of vulnerabilities across 30 programming languages and span over one million individual APIs. In summary, this release includes the following:

GoLang Updates (version supported: 1.17)

Updated support for the Go standard library to support up to version 1.17. Go is a statically typed open-source language designed by Google™ that aims to make it easy to build simple, reliable, and efficient software. Go is syntactically similar to C, but with memory safety mechanisms, garbage collection, and structural typing. This update improves the coverage of the standard library namespaces to include the following additional categories:

- Header Manipulation: SMTP
- Mail Command Injection: SMTP

Support improves detection of weaknesses under existing namespace coverage and expands to include the following new namespaces:

- io/fs
- math/big
- math/random
- net/smtp
- net/textproto
- text/template

Python Updates (version supported: 3.10)

Python is a general-purpose, powerful programming language, with dynamic typing and efficient high-level data structures. It supports multiple programming paradigms, including structured, object-oriented, and functional programming. This release increases our coverage to the Python 3.10 by expanding our support for changes in Python's standard library API. Updated categories include:

- Path Manipulation
- Privacy Violation
- System Information Leak

ECMAScript Updates (version supported: 2022)[1]

ECMAScript 2022, also known as ES2022 or ES12, is the latest version of the ECMAScript standard for the JavaScript language. Key features of ES2022 are private methods and accessors, extended numeric literals, logical assignment operators, and improved error handling. Support for ES2022 extends coverage of all relevant JavaScript vulnerability categories to the latest version of the ECMAScript standard.

Vue 2 (version supported: 2.7)

Initial support for Vue 2. Vue is an open-source reactive framework for building user interfaces and single-page-applications for all ECMAScript 5 compatible browsers. Vue focuses on the view layer of a web application and was created as a minimalist alternative to common frameworks such as Angular and React.

iOS SDK Updates (version supported: 16)[2]

Apple's iOS SDK provides a collection of frameworks that enable developers to build mobile applications for Apple iPhone and iPad devices. This release contains incremental updates to our iOS SDK support for Swift and Objective-C. New and updated rules extend our API coverage of the DataDetection, Foundation, Security, SwiftUI, and UIKit frameworks in iOS SDK 15 and 16 for Swift iOS and iPadOS applications. These updates improve issue detection for many existing weakness categories, including:

- Biometric Authentication: Insufficient Touch ID Protection
- Format String
- Insecure Transport: Weak SSL Protocol
- Log Forging
- Privacy Violation
- System Information Leak: External
- System Information Leak: Internal
- Weak Encryption: Inadequate RSA Padding

Additionally, two new weakness categories are introduced in this release for iOS and iPadOS applications:

- Insecure Storage: Persistent Named Pasteboard
- Insecure Storage: Universal Clipboard

Salesforce Apex and Visualforce Updates (version supported: v57)[3]

Salesforce Apex is the programming language used for creating Salesforce applications such as business transactions, database management, web services, and Visualforce pages. This update improves our

[1] Requires Fortify Static Code Analyzer 23.1.
[2] New rules for iOS SDK 16 require Fortify Static Code Analyzer 22.2 or later.
[3] Requires Fortify Static Code Analyzer 23.1 or later for some new rules that target the Apex v57 APIs.

support for Apex v57 APIs and Visualforce APIs. In addition to improving support of existing weakness categories, the following have been added to our Apex support:

- Cross-Site Request Forgery
- Poor Error Handling: Empty Catch Block
- Unsafe Reflection

Additionally, the following new weakness category is introduced for Apex applications:

- Access Control: Unenforced Sharing Rules

Initial Support for Google Dataflow with Java Apache Beam (version supported: 2.46.0)

Apache Beam is an open-source, unified programming model for building data processing pipelines capable of being run on a variety of data processing backends. Initial support for Apache Beam enables data processing pipelines, such as Google Dataflow, and is limited to the Java programming language by identifying sources of data within Apache Beam pipelines. Support enables reporting of relevant Java vulnerability categories such as Command Injection, Privacy Violation, and Log Forging within Apache Beam transforms.

.NET 7 (version supported: 7.0)

.NET is a general programming platform that enables programmers to write code in languages such as C# and VB.NET with a standardized set of APIs. This release increases our coverage to the latest version of .NET, improving dataflow, as well as expanding our API coverage for the following categories:

- Denial of Service: Regular Expression
- Path Manipulation
- Path Manipulation: Zip Entry Overwrite
- Permission Manipulation
- Privacy Violation
- Setting Manipulation
- System Information Leak

ASP.NET Core 7 (version supported: 7.0)

ASP.NET Core is the flagship web framework for use with .NET. The framework includes functionality to create many types of applications including MVC web applications and Web APIs. This release increases our coverage to the latest version of ASP.NET Core, expanding our categories supported to include:

- Denial of Service
- Privacy Violation
- Setting Manipulation
- System Information Leak

Additionally, the following new weakness category is introduced for ASP.NET applications:

- ASP.NET Misconfiguration: Logging Sensitive Information

Cloud Infrastructure as Code (IaC)

IaC is the process of managing and provisioning computer resources through code rather than various manual processes. Improved support includes Terraform configurations for deployment to AWS and Azure as well as improved coverage for Azure Resource Manager (ARM). Common issues related to the configuration of these services are now reported to the developer.

Amazon AWS and Microsoft Azure Terraform Configurations

Terraform is an open-source infrastructure as code 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 Microsoft Azure infrastructure and Amazon Web Services (AWS). In this release, we report the following categories for Terraform configurations:

- AWS Terraform Misconfiguration: AMI Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: Aurora Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: Aurora Publicly Accessible
- AWS Terraform Misconfiguration: CloudTrail Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: Database Migration Service Publicly Accessible
- AWS Terraform Misconfiguration: DocumentDB Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: DocumentDB Publicly Accessible
- AWS Terraform Misconfiguration: EBS Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: EC2 Image Builder Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: EFS Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: Elasticache Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: File Cache Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: FSx Lustre Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: FSx Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: FSx ONTAP Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: FSx OpenZFS Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: FSx Windows Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: Insecure AMI Storage
- AWS Terraform Misconfiguration: Insecure Aurora Storage
- AWS Terraform Misconfiguration: Insecure DocumentDB Storage
- AWS Terraform Misconfiguration: Insecure EC2 Image Builder Storage
- AWS Terraform Misconfiguration: Insecure EFS Storage
- AWS Terraform Misconfiguration: Insecure Neptune Storage
- AWS Terraform Misconfiguration: Insecure Redshift Storage
- AWS Terraform Misconfiguration: Insufficient Aurora Monitoring
- AWS Terraform Misconfiguration: Insufficient DocumentDB Monitoring
- AWS Terraform Misconfiguration: Insufficient RDS Monitoring

- AWS Terraform Misconfiguration: Kinesis Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: Lightsail Publicly Accessible
- AWS Terraform Misconfiguration: Location Service Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: Neptune Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: RDS Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: RDS Publicly Accessible
- AWS Terraform Misconfiguration: Redshift Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: SageMaker Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: Secrets Manager Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: S3 Missing Customer-Managed Encryption Key
- AWS Terraform Misconfiguration: Timestream Missing Customer-Managed Encryption Key
- Azure Terraform Misconfiguration: Improper App Service CORS Policy
- Azure Terraform Misconfiguration: Improper Cognitive Services Network Access Control
- Azure Terraform Misconfiguration: Improper CosmosDB CORS Policy
- Azure Terraform Misconfiguration: Improper Functions CORS Policy
- Azure Terraform Misconfiguration: Improper Healthcare CORS Policy
- Azure Terraform Misconfiguration: Improper IoT Central Network Access Control
- Azure Terraform Misconfiguration: Improper IoT Hub Network Access Control
- Azure Terraform Misconfiguration: Improper Key Vault Network Access Control
- Azure Terraform Misconfiguration: Improper Logic App CORS Policy
- Azure Terraform Misconfiguration: Improper Media Services Network Access Control
- Azure Terraform Misconfiguration: Improper Service Bus Network Access Control
- Azure Terraform Misconfiguration: Improper SignalR CORS Policy
- Azure Terraform Misconfiguration: Improper SignalR Network Access Control
- Azure Terraform Misconfiguration: Improper Spring Apps CORS Policy
- Azure Terraform Misconfiguration: Improper Storage CORS Policy
- Azure Terraform Misconfiguration: Improper Storage Network Access Control
- Azure Terraform Misconfiguration: Improper Web PubSub Network Access Control
- Azure Terraform Misconfiguration: Insecure Event Hubs Transport
- Azure Terraform Misconfiguration: Insecure Front Door Transport
- Azure Terraform Misconfiguration: Insecure Functions Transport
- Azure Terraform Misconfiguration: Insecure Redis Transport
- Azure Terraform Misconfiguration: Insecure Service Bus Transport
- Azure Terraform Misconfiguration: Insecure SQL Database Transport
- Azure Terraform Misconfiguration: Insecure SQL Managed Instance Transport

Microsoft Azure Resource Manager (ARM) Configurations
ARM is the deployment and management service for Azure. ARM provides a management layer that enables you to create, update, and delete resources in your Azure account. In this release, we report the following weakness categories for ARM configurations:

- Azure ARM Misconfiguration: Automation Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: Batch Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: Cognitive Services Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: Databricks Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: Event Hubs Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: Insecure CDN Transport

- Azure ARM Misconfiguration: Insecure Database for MySQL Storage
- Azure ARM Misconfiguration: Insecure Database for PostgreSQL Storage
- Azure ARM Misconfiguration: Insecure DataBricks Storage
- Azure ARM Misconfiguration: Insecure EventHub Storage
- Azure ARM Misconfiguration: Insecure EventHub Transport
- Azure ARM Misconfiguration: Insecure IoT Hub Transport
- Azure ARM Misconfiguration: Insecure Recovery Services Backup Storage
- Azure ARM Misconfiguration: Insecure Recovery Services Vaults Storage
- Azure ARM Misconfiguration: Insecure Redis Enterprise Transport
- Azure ARM Misconfiguration: Insecure Redis Transport
- Azure ARM Misconfiguration: Insecure Service Bus Storage
- Azure ARM Misconfiguration: Insecure Service Bus Transport
- Azure ARM Misconfiguration: Insecure Storage Account Storage
- Azure ARM Misconfiguration: IoT Hub Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: NetApp Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: Service Bus Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: Storage Account Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: Weak App Service Authentication
- Azure ARM Misconfiguration: Weak SignalR Authentication

Customizable Password Management and Key Management Regular Expressions[4]
Sometimes the only way to match passwords and encryption keys within source code is by making an educated guess using regular expressions. These are now customizable via properties, have been made more consistent across languages, and the default regular expressions have been restricted to minimize false positives.

The global regular expressions can be configured with one of the following properties:

```
com.fortify.sca.rules.key_regex.global
```
or
```
com.fortify.sca.rules.password_regex.global
```

You can set more specific variants per-language with the following properties:

```
com.fortify.sca.rules.key_regex.abap
com.fortify.sca.rules.key_regex.actionscript
com.fortify.sca.rules.key_regex.cfml
com.fortify.sca.rules.key_regex.cpp
com.fortify.sca.rules.key_regex.golang
com.fortify.sca.rules.key_regex.java
com.fortify.sca.rules.key_regex.javascript
com.fortify.sca.rules.key_regex.jsp
com.fortify.sca.rules.key_regex.objc
com.fortify.sca.rules.key_regex.php
```

[4] Requires Fortify Static Code Analyzer 23.1.

```
com.fortify.sca.rules.key_regex.python
com.fortify.sca.rules.key_regex.ruby
com.fortify.sca.rules.key_regex.sql
com.fortify.sca.rules.key_regex.swift
com.fortify.sca.rules.key_regex.vb

com.fortify.sca.rules.password_regex.abap
com.fortify.sca.rules.password_regex.actionscript
com.fortify.sca.rules.password_regex.cfml
com.fortify.sca.rules.password_regex.cobol
com.fortify.sca.rules.password_regex.config
com.fortify.sca.rules.password_regex.cpp
com.fortify.sca.rules.password_regex.docker
com.fortify.sca.rules.password_regex.dotnet
com.fortify.sca.rules.password_regex.golang
com.fortify.sca.rules.password_regex.java
com.fortify.sca.rules.password_regex.javascript
com.fortify.sca.rules.password_regex.json
com.fortify.sca.rules.password_regex.jsp
com.fortify.sca.rules.password_regex.objc
com.fortify.sca.rules.password_regex.php
com.fortify.sca.rules.password_regex.properties
com.fortify.sca.rules.password_regex.python
com.fortify.sca.rules.password_regex.ruby
com.fortify.sca.rules.password_regex.sql
com.fortify.sca.rules.password_regex.swift
com.fortify.sca.rules.password_regex.vb
com.fortify.sca.rules.password_regex.yaml
```

For more specific details about the default regular expressions, see the Fortify Static Code Analyzer User Guide.

DISA STIG 5.2

To support our federal customers in the area of compliance, correlation of the Fortify Taxonomy to the Defense Information Systems Agency (DISA) Application Security and Development STIG version 5.2 has been added.

PCI DSS 4.0

To support our e-commerce and financial services customers in the area of compliance, this release supports correlation between our Fortify Taxonomy categories and the requirements specified in the latest version of the Payment Card Industry Data Security Standard, version 4.0.

PCI SSF 1.2

To support our e-commerce and financial services customers in the area of compliance, this release supports correlation between our Fortify Taxonomy categories and the control objectives specified in the new "Secure Software Requirements and Assessment Procedures", defined in the Payment Card Industry (PCI) Secure Software Standard (SSS) as part of the new Software Security Framework (SSF), version 1.2.

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:

Removal of "Denial of Service: Parse Double"

The *Denial of Service: Parse Double* category has been removed because the vulnerability only exists in Java version 6 update 23 and earlier versions. Customers who are using those vulnerable Java versions can still download the removed rules in a separate Rulepack from the Fortify Customer Support Portal under Premium Content.

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:

- *Access Control: Database* – false positives reduced when data originates from a database
- *Android Bad Practices: Unnecessary Component Exposure* – false positives reduced when an Android receiver is marked with `android:exported="false"`
- *ASP.NET MVC Bad Practices: Controller Action Not Restricted to POST* – false positives reduced when controller actions pass their input directly to a view without state change
- *Credential Management: Hardcoded API Credentials* – is no longer found in `google-services.json` when recommended
- *Credential Management: Hardcoded API Credentials* – false positives reduced on Facebook revision keys
- *Cross-Site Scripting* – false positives removed that were triggered in VB6 Windows Forms applications
- *Dead Code: Unused Field* – false positives reduced in Java lambdas
- *Dockerfile Misconfiguration: Dependency Confusion* – false positives reduced when using a local library definition
- False positives removed across multiple categories in all supported languages when a dataflow issue is reported on a `Boolean` variable

- False positives removed across multiple categories in C/C++ applications when retrieving file information via WinAPI functions
- *HTTP Parameter Pollution* – false positives reduced on URL encoded values
- *Insecure Randomnes: Hardcoded Seed* and *Insecure Randomness: User-Controlled Seed* – false positives reduced when using `Random` and `SplittableRandom` classes in Java applications
- *Insecure Storage: Unspecified Keychain Access Policy, Insecure Storage: Externally Available Keychain,* and *Insecure Storage: Passcode Policy Unenforced* – false positives reduced in Swift iOS applications when recommended remediations are applied
- *Memory Leak* – false positives reduced when adding a pointer to a boost program options description
- *Memory Leak* – false positives reduced when using `std::unique_ptr`
- *Null Dereference* – false positives removed when casting `0` to a `byte` in .NET applications
- *Password Management: Hardcoded Password* – false positives reduced on passwords in comments
- *Privacy Violation: Android Internal Storage* – false positives reduced when using `EncryptedSharedPreferences` objects in Android applications
- *SOQL Injection* and *Access Control: Database* – false positives reduced when using `Database.getQueryLocator()` in Salesforce Apex applications

Category Changes

When weakness category name changes occur, analysis results when merging prior scans with new scans will result in added/removed categories.

To improve consistency, the following category was renamed:

- *ASP.NET Bad Practices: Leftover Debug Code* now reports as *.NET Bad Practices: Leftover Debug Code* when triggering in general .NET code.

Furthermore, to improve consistency in the reporting of IaC flaws across categories, an additional 121 categories have been renamed in this release (see Appendix A).

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[5]

Cacti is a framework that provides logging and graphing capabilities for users to monitor devices on the network. The *remote_agent.php* file is susceptible to a remote code execution (RCE) vulnerability identified by CVE-2022-46169 in Cacti versions earlier than 1.2.23. The `poller_id` parameter is passed when polling data calls proc_open method using user input. As this value is not sanitized, it enables an attacker to execute commands on the target machine. Combining this command injection issue with an authentication bypass using an `X-Forwarded-For` header results in an unauthenticated attacker compromising the entire application. This release includes a check to detect this vulnerability on a target server that runs affected Cacti versions.

SAML Bad Practices: Insecure Transform

SAML messages are cryptographically signed to guarantee validity and integrity of the assertion. One of the steps the service provider must perform for signature verification is the transformation of the data that is pointed to by the `Reference` element. Usually, the `Transform` operation aims at selecting just a subset of the referenced data. However, an attacker could use some types of transforms to cause a denial of service and in some environments even arbitrary code execution. This release includes a check that triggers if the service provider allows insecure types of transforms in XML references.

Compliance Reports

DISA STIG 5.2

To support our federal customers compliance needs, this release contains a correlation of the WebInspect checks to the latest version of the Defense Information Systems Agency Application Security and Development STIG, version 5.2.

PCI DSS 4.0

To support our e-commerce and financial services customers compliance needs, this release contains a correlation of the WebInspect checks to the requirements specified in the latest version of the Payment Card Industry Data Security Standard, version 4.0.

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

PCI SSF 1.2

To support our e-commerce and financial services customers compliance needs, this release contains a correlation of the WebInspect checks to the control objectives specified in the new "Secure Software Requirements and Assessment Procedures", defined in the Payment Card Industry (PCI) Secure Software Standard (SSS) as part of the new Software Security Framework (SSF), version 1.2.

Policy Updates

DISA STIG 5.2

A policy customized to include checks relevant to DISA STIG 5.2 has been added to the WebInspect SecureBase list of supported policies.

PCI DSS 4.0

A policy customized to include checks relevant to PCI DSS 4.0 has been added to the WebInspect SecureBase list of supported policies.

PCI SSF 1.2

A policy customized to include checks relevant to PCI SSF 1.2 has been added to the WebInspect SecureBase list of supported policies.

Miscellaneous Errata

In this release, we invested resources 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:

Password Management: Weak Password Policy[6]

This release includes minor improvements for the password entropy check where password/username fields have improved detection of custom username and password fields. This fix helps reduce false positives in findings related to check IDs 11496, 11498 and 11661.

Fortify Premium Content

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

[6] Requires WebInspect 23.1 or later.

DISA STIG 5.2, PCI SSF 1.2, and PCI DSS 4.0

To accompany the new correlations, this release also contains a new report bundle for Fortify Software Security Center with support for DISA STIG 5.2, PCI DSS 4.0, and PCI SSF 1.2, 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.

Appendix A: IaC Weakness Categories Renamed

Removed Category	Added Category
Access Control: Azure Blob Storage	Azure Ansible Misconfiguration: Improper Blob Storage Access Control
Access Control: Azure Blob Storage	Azure ARM Misconfiguration: Improper Blob Storage Access Control
Access Control: Azure Network Group	Azure Ansible Misconfiguration: Improper Security Group Network Access Control
Access Control: Azure Network Group	Azure ARM Misconfiguration: Improper Security Group Network Access Control
Access Control: Azure Storage	Azure Ansible Misconfiguration: Improper Storage Account Network Access Control
Access Control: Azure Storage	Azure ARM Misconfiguration: Improper Storage Network Access Control
Access Control: EC2	AWS Ansible Misconfiguration: Improper EC2 Network Access Control
Access Control: EC2	AWS CloudFormation Misconfiguration: Improper EC2 Network Access Control
Access Control: Overly Broad IAM Principal	AWS CloudFormation Misconfiguration: Improper S3 Access Control Policy
Access Control: Overly Permissive S3 Policy	AWS Ansible Misconfiguration: Improper S3 Bucket Network Access Control
Access Control: Overly Permissive S3 Policy	AWS CloudFormation Misconfiguration: Improper S3 Bucket Network Access Control
AKS Bad Practices: Missing Azure Monitor Integration	Azure Ansible Misconfiguration: Insufficient AKS Monitoring
AKS Bad Practices: Missing Azure Monitor Integration	Azure ARM Misconfiguration: Insufficient AKS Monitoring
Ansible Bad Practices: Missing CloudWatch Integration	AWS Ansible Misconfiguration: Insufficient CloudTrail Logging
Ansible Misconfiguration: Log Validation Disabled	AWS Ansible Misconfiguration: Missing CloudTrail Log Validation

AWS Ansible Bad Practices: Improper IAM Access Control Policy	AWS Ansible Misconfiguration: Improper IAM Access Control Policy
AWS Ansible Misconfiguration: Amazon RDS Publicly Accessible	AWS Ansible Misconfiguration: RDS Publicly Accessible
AWS CloudFormation Bad Practices: Missing CloudWatch Integration	AWS CloudFormation Misconfiguration: Insufficient CloudTrail Logging
AWS CloudFormation Bad Practices: Missing CloudWatch Integration	AWS CloudFormation Misconfiguration: Insufficient DocumentDB Logging
AWS CloudFormation Bad Practices: Missing CloudWatch Integration	AWS CloudFormation Misconfiguration: Insufficient Neptune Logging
AWS CloudFormation Bad Practices: Redshift Publicly Accessible	AWS CloudFormation Misconfiguration: Improper Redshift Network Access Control
AWS CloudFormation Bad Practices: User-Bound IAM Policy	AWS CloudFormation Misconfiguration: Improper IAM Access Control Policy
AWS CloudFormation Misconfiguration: API Gateway Unauthenticated Access	AWS CloudFormation Misconfiguration: Improper API Gateway Access Control
AWS Cloudformation Misconfiguration: Insecure EC2 AMI Storage	AWS Ansible Misconfiguration: Insecure EC2 AMI Storage
AWS Cloudformation Misconfiguration: Insecure EFS Storage	AWS Ansible Misconfiguration: Insecure EFS Storage
AWS Cloudformation Misconfiguration: Insecure Kinesis Data Stream Storage	AWS Ansible Misconfiguration: Insecure Kinesis Data Stream Storage
AWS CloudFormation Misconfiguration: Insecure Transport	AWS CloudFormation Misconfiguration: Insecure Redshift Transport
AWS CloudFormation Misconfiguration: Insufficient RedShift Logging	AWS CloudFormation Misconfiguration: Insufficient Redshift Logging
AWS CloudFormation Misconfiguration: Insufficient S3 Logging	AWS CloudFormation Misconfiguration: Insufficient S3 Bucket Logging
AWS CloudFormation Misconfiguration: Log Validation Disabled	AWS CloudFormation Misconfiguration: Missing CloudTrail Log Validation
AWS CloudFormation Misconfiguration: root User Access Key	AWS CloudFormation Misconfiguration: Improper IAM Access Control
AWS CloudFormation Misconfiguration: Unrestricted Lambda Principal	AWS CloudFormation Misconfiguration: Improper Lambda Access Control Policy
AWS Terraform Misconfiguration: Amazon API Gateway Publicly Accessible	AWS Terraform Misconfiguration: API Gateway Publicly Accessible
AWS Terraform Misconfiguration: Amazon EBS Insecure Storage	AWS Terraform Misconfiguration: Insecure EBS Storage
AWS Terraform Misconfiguration: Amazon ElastiCache Insecure Transport	AWS Terraform Misconfiguration: Insecure ElastiCache Transport
AWS Terraform Misconfiguration: Amazon MQ Publicly Accessible	AWS Terraform Misconfiguration: MQ Publicly Accessible
AWS Terraform Misconfiguration: Amazon Neptune Publicly Accessible	AWS Terraform Misconfiguration: Neptune Publicly Accessible
AWS Terraform Misconfiguration: Amazon RDS Insecure Storage	AWS Terraform Misconfiguration: Insecure RDS Storage

AWS Terraform Misconfiguration: Amazon RDS Proxy Insecure Transport	AWS Terraform Misconfiguration: Insecure RDS Proxy Transport
AWS Terraform Misconfiguration: Amazon Redshift Publicly Accessible	AWS Terraform Misconfiguration: Redshift Publicly Accessible
AWS Terraform Misconfiguration: Amazon SNS Insecure Storage	AWS Terraform Misconfiguration: Insecure SNS Storage
Azure ARM Misconfiguration: Improper Storage Account Network Access Control	Azure ARM Misconfiguration: Improper Storage Network Access Control
Azure Monitor Misconfiguration: Insufficient Logging	Azure ARM Misconfiguration: Insufficient Application Insights Monitoring
Azure Resource Manager Misconfiguration: Public Access Allowed	Azure ARM Misconfiguration: Improper Storage Network Access Control
Azure Resource Manager Misconfiguration: Public Access Allowed	Azure ARM Misconfiguration: Public Access Allowed
Azure Terraform Bad Practices: Azure Disk Snapshot Missing Customer-Managed Key	Azure Terraform Misconfiguration: Azure Disk Snapshot Missing Customer-Managed Key
Azure Terraform Bad Practices: Container Registry Missing Customer-Managed Key	Azure Terraform Misconfiguration: Container Registry Missing Customer-Managed Key
Azure Terraform Bad Practices: Cosmos DB Missing Customer-Managed Key	Azure Terraform Misconfiguration: Cosmos DB Missing Customer-Managed Key
Azure Terraform Bad Practices: Shared Image Missing Customer-Managed Key	Azure Terraform Misconfiguration: Shared Image Missing Customer-Managed Key
Azure Terraform Bad Practices: SQL Database Missing Customer-Managed Key	Azure Terraform Misconfiguration: SQL Database Missing Customer-Managed Key
Azure Terraform Bad Practices: Storage Account Missing Customer-Managed Key	Azure Terraform Misconfiguration: Storage Account Missing Customer-Managed Key
Azure Terraform Bad Practices: Storage Encryption Scope Missing Customer-Managed Key	Azure Terraform Misconfiguration: Storage Encryption Scope Missing Customer-Managed Key
Azure Terraform Misconfiguration: Insecure PostgresSQL Transport	Azure Terraform Misconfiguration: Insecure PostgreSQL Transport
GCP Terraform Bad Practice: Overly Permissive Service Account	GCP Terraform Bad Practices: Overly Permissive Service Account
GCP Terraform Bad Practices: Apigee Missing Customer-Managed Encryption Key	GCP Terraform Misconfiguration: Apigee Missing Customer-Managed Encryption Key
GCP Terraform Bad Practices: BigQuery Missing Customer-Managed Encryption Key	GCP Terraform Misconfiguration: BigQuery Missing Customer-Managed Encryption Key
GCP Terraform Bad Practices: Cloud Bigtable Missing Customer-Managed Encryption Key	GCP Terraform Misconfiguration: Cloud Bigtable Missing Customer-Managed Encryption Key
GCP Terraform Bad Practices: Cloud Functions Missing Customer-Managed Encryption Key	GCP Terraform Misconfiguration: Cloud Functions Missing Customer-Managed Encryption Key
GCP Terraform Bad Practices: Cloud Spanner Missing Customer-Managed Encryption Key	GCP Terraform Misconfiguration: Cloud Spanner Missing Customer-Managed Encryption Key
GCP Terraform Bad Practices: Filestore Missing Customer-Managed Encryption Key	GCP Terraform Misconfiguration: Filestore Missing Customer-Managed Encryption Key
GCP Terraform Bad Practices: Pub/Sub Missing Customer-Managed Encryption Key	GCP Terraform Misconfiguration: Pub/Sub Missing Customer-Managed Encryption Key

GCP Terraform Bad Practices: Secret Manager Missing Customer-managed Encryption Key	GCP Terraform Misconfiguration: Secret Manager Missing Customer-Managed Encryption Key
Insecure SSL: Inadequate Certificate Verification	Kubernetes Misconfiguration: Inadequate Certificate Verification
Insecure SSL: Overly Broad Certificate Trust	Kubernetes Misconfiguration: Overly Broad Certificate Trust
Insecure SSL: Server Identity Verification Disabled	Kubernetes Misconfiguration: Missing API Server Identity Verification
Insecure Storage: Missing DocumentDB Encryption	AWS CloudFormation Misconfiguration: Insecure DocumentDB Storage
Insecure Storage: Missing EBS Encryption	AWS Ansible Misconfiguration: Insecure EBS Storage
Insecure Storage: Missing EBS Encryption	AWS CloudFormation Misconfiguration: Insecure EBS Storage
Insecure Storage: Missing Elasticache Encryption	AWS CloudFormation Misconfiguration: Insecure Elasticache Storage
Insecure Storage: Missing Neptune Encryption	AWS CloudFormation Misconfiguration: Insecure Neptune DB Storage
Insecure Storage: Missing RDS Encryption	AWS Ansible Misconfiguration: Insecure RDS Storage
Insecure Storage: Missing RDS Encryption	AWS CloudFormation Misconfiguration: Insecure RDS Storage
Insecure Storage: Missing Redshift Encryption	AWS Ansible Misconfiguration: Insecure Redshift Storage
Insecure Storage: Missing Redshift Encryption	AWS CloudFormation Misconfiguration: Insecure Redshift Storage
Insecure Storage: Missing S3 Encryption	AWS Ansible Misconfiguration: Insecure S3 Bucket Storage
Insecure Storage: Missing S3 Encryption	AWS CloudFormation Misconfiguration: Insecure S3 Bucket Storage
Insecure Storage: Missing SNS Topic Encryption	AWS CloudFormation Misconfiguration: Insecure SNS Topic Storage
Insecure Transport: Azure Storage	Azure Ansible Misconfiguration: Insecure Storage Account Transport
Insecure Transport: Azure Storage	Azure ARM Misconfiguration: Insecure Storage Account Transport
Insecure Transport: Database	AWS CloudFormation Misconfiguration: Insecure DocumentDB Transport
Insecure Transport: Database	Azure Ansible Misconfiguration: Insecure MySQL Server Transport
Insecure Transport: Database	Azure Ansible Misconfiguration: Insecure PostgreSQL Server Transport
Insecure Transport: Database	Azure ARM Misconfiguration: Insecure MySQL Server Transport
Insecure Transport: Database	Azure ARM Misconfiguration: Insecure PostgreSQL Server Transport

Insecure Transport: Missing Elasticache Encryption	AWS CloudFormation Misconfiguration: Insecure Elasticache Transport
Insecure Transport: Weak SSL Protocol	Azure ARM Misconfiguration: Insecure Active Directory Domain Service Transport
Key Management: Excessive Expiration	AWS CloudFormation Misconfiguration: Improper IAM Access Control Policy
Key Management: Excessive Expiration	Azure ARM Misconfiguration: Improper KeyVault Access Control Policy
Kubernetes Bad Practices: Automated iptables Management Disabled	Kubernetes Misconfiguration: Automated iptables Management Disabled
Kubernetes Bad Practices: Default Namespace	Kubernetes Misconfiguration: Default Namespace
Kubernetes Bad Practices: Host Write Access	Kubernetes Misconfiguration: Host Write Access
Kubernetes Bad Practices: Kernel Defaults Overridden	Kubernetes Misconfiguration: Kernel Defaults Overridden
Kubernetes Bad Practices: Kubelet Streaming Connection Timeout Disabled	Kubernetes Misconfiguration: Kubelet Streaming Connection Timeout Disabled
Kubernetes Bad Practices: Missing API Server Authorization	Kubernetes Misconfiguration: Missing API Server Authorization
Kubernetes Bad Practices: Missing Kubelet Authorization	Kubernetes Misconfiguration: Missing Kubelet Authorization
Kubernetes Bad Practices: Missing Node Authorization	Kubernetes Misconfiguration: Missing Node Authorization
Kubernetes Bad Practices: Missing NodeRestriction Admission Controller	Kubernetes Misconfiguration: Missing NodeRestriction Admission Controller
Kubernetes Bad Practices: Missing PodSecurityPolicy Admission Controller	Kubernetes Misconfiguration: Missing PodSecurityPolicy Admission Controller
Kubernetes Bad Practices: Missing RBAC Authorization	Kubernetes Misconfiguration: Missing RBAC Authorization
Kubernetes Bad Practices: Missing Security Context	Kubernetes Misconfiguration: Missing Security Context
Kubernetes Bad Practices: Missing SecurityContextDeny Admission Controller	Kubernetes Misconfiguration: Missing SecurityContextDeny Admission Controller
Kubernetes Bad Practices: Missing ServiceAccount Admission Controller	Kubernetes Misconfiguration: Missing ServiceAccount Admission Controller
Kubernetes Bad Practices: NamespaceLifecycle Enforcement Disabled	Kubernetes Misconfiguration: NamespaceLifecycle Enforcement Disabled
Kubernetes Bad Practices: readOnlyPort Enabled	Kubernetes Misconfiguration: readOnlyPort Enabled
Kubernetes Bad Practices: Service Account Token Automounted	Kubernetes Misconfiguration: Service Account Token Automounted
Kubernetes Bad Practices: Shared Service Account Credentials	Kubernetes Misconfiguration: Shared Service Account Credentials
Kubernetes Bad Practices: Static Authentication Token	Kubernetes Misconfiguration: Static Authentication Token

Kubernetes Bad Practices: Unconfigured API Server Logging	Kubernetes Misconfiguration: Unconfigured API Server Logging
Kubernetes Misconfiguration: Insecure Transport	Kubernetes Misconfiguration: Insecure Kubelet Transport
Kubernetes Misconfiguration: Server Identity Verification Disabled	Kubernetes Misconfiguration: Missing Kubelet Identity Verification
Often Misused: Weak SSL Certificate	Kubernetes Misconfiguration: Weak etcd SSL Certificate
Poor Logging Practice: Excessive Cloud Log Retention	AWS CloudFormation Misconfiguration: Insufficient Log Group Logging
Poor Logging Practice: Insufficient Cloud Log Retention	Azure ARM Misconfiguration: Insufficient Application Insights Logging
Poor Logging Practice: Insufficient Cloud Log Retention	Azure ARM Misconfiguration: Insufficient SQL Server Logging
Poor Logging Practice: Insufficient Cloud Log Retention	Kubernetes Misconfiguration: Insufficient API server Log Retention
Poor Logging Practice: Insufficient Cloud Log Rotation	Kubernetes Misconfiguration: Insufficient Cloud Log Rotation
Poor Logging Practice: Insufficient Cloud Log Size	Kubernetes Misconfiguration: Insufficient Cloud Log Size
Privilege Management: Overly Broad Access Policy	AWS Ansible Misconfiguration: Improper IAM Access Control Policy
Privilege Management: Overly Broad Access Policy	AWS CloudFormation Misconfiguration: Improper IAM Access Control Policy
System Information Leak: Kubernetes Profiler	Kubernetes Misconfiguration: API Server Profiling
System Information Leak: Kubernetes Profiler	Kubernetes Misconfiguration: Controller Manager Profiling

Contact Fortify Technical Support

OpenText Fortify
http://softwaresupport.softwaregrp.com/
+1 (800) 509-1800

Contact SSR

Alexander M. Hoole
Senior Manager, Software Security Research
OpenText Fortify
hoole@opentext.com
+1 (650) 427-9973

Peter Blay
Manager, Software Security Research
OpenText Fortify
pblay@opentext.com
+1 (669) 309-1634