Fortify 软件安全内容

2023 更新 3
2023 年 9 月 29 日

关于 OpenText Fortify Software Security Research

Fortify Software Security Research 团队将尖端研究成果转换为助力 Fortify 产品组合（包括 Fortify Static Code Analyzer (SCA) 和 Fortify WebInspect）的安全情报。现在，Fortify 软件安全内容支持超过 33 种语言的 1,627 个漏洞类别，且涵盖的单独 API 超过一百万个。

Fortify Software Security Research (SSR) 团队荣幸地宣布，我们即将推出 Fortify Secure Coding Rulepacks（2023.3.0 英文版）、Fortify WebInspect SecureBase（可通过 SmartUpdate 获取）和 Fortify Premium Content 的更新。

Fortify Secure Coding Rulepacks [Fortify Static Code Analyzer]

这次发行的 Fortify Secure Coding Rulepacks 可以检测超过 33 种语言的 1,403 个不同的漏洞类别，且涵盖的单独 API 超过一百万个。概括来说，此版本包括以下内容：

改进了对 Android 13 的支持（支持的版本：33）

Android 平台是专为移动设备设计的开源软件堆栈。Android 的主要组件是 Java API 框架，它向应用程序开发人员开放 Android 功能。此版本扩展了使用 Java 或 Kotlin 编写并利用 Android Java API 框架的原生 Android 应用程序中的漏洞检测。此版本中为 Android 应用程序引入了三个新的缺陷类别：

- Intent Manipulation: Implicit Internal Intent
- Intent Manipulation: Implicit Pending Intent
- Intent Manipulation: Mutable Pending Intent

对 Android Jetpack (AndroidX) 的初始支持

Android Jetpack 是一组库、工具和指南，可帮助开发人员更轻松地创建 Android 应用程序。Jetpack 涵盖 androidx.* 软件包，并且与平台 API 分离，这有助于促进向后兼容性并允许更频繁的更新。在此版本中，我们提供了对此软件套件的初始覆盖。

对 Android Jetpack 的初始覆盖范围支持检测以下库中的缺陷：

- androidx.appsearch (version supported: 1.1.0-alpha03)
- androidx.compose.foundation (version supported: 1.5.1)
- androidx.compose.material (version supported: 1.5.1)
- androidx.compose.material3 (version supported: 1.1.2)
- androidx.compose.ui (version supported: 1.5.1)
- androidx.core (version supported: 1.12.0)
- androidx.credentials (version supported: 1.2.0-beta04)
- androidx.datastore (version supported: 1.0.0)
- androidx.security.crypto (version supported: 1.0.0)
- androidx.sqlite (version supported: 2.3.1)

类别覆盖范围改进示例包括：

- Access Control: Database
- Command Injection
- Denial of Service
- Denial of Service: Regular Expression
- Header Manipulation

- Insecure Transport
- Open Redirect
- Password Management: Empty Password
- Password Management: Hardcoded Password
- Path Manipulation
- Privacy Violation
- Resource Injection
- Server-Side Request Forgery
- SQL Injection
- System Information Leak
- System Information Leak: Internal

MySQL Connector/Python 支持（支持的版本：8.1.0）

MySQL Connector/Python 是一个软件库，可促进 Python 应用程序与 MySQL 数据库之间的交互。它充当 Python 编程语言和 MySQL 数据库管理系统之间的桥梁或连接器，使开发人员能够使用 Python 代码轻松连接、查询和操作 MySQL 数据库中的数据。

改进的类别覆盖范围包括：

- Access Control: Database
- Denial of Service
- Insecure Transport: Client Identity Verification Disabled
- Insecure Transport: Database
- Insecure Transport: Weak SSL Protocol
- Password Management
- Password Management: Empty Password
- Password Management: Hardcoded Password
- Password Management: Weak Cryptography
- Path Manipulation
- Server-Side Request Forgery
- SQL Injection

改进了对 Django 的支持（支持的版本：3.2）

Django 是一个使用 Python 编写的 Web 框架，旨在促进安全、快速的 Web 开发。开发的速度和安全性是通过框架中的高级抽象来实现的，其中使用代码构造和生成来大幅减少样板代码。在此版本中，我们更新了现有的 Django 覆盖范围以支持最高版本 3.2。

改进的覆盖范围包括以下命名空间：*Django.contrib.auth.models*、*Django.db.models* 和 *Django.http.response*。此外，改进的缺陷类别覆盖范围包括：

- Cookie Security: Overly Permissive SameSite Attribute
- Header Manipulation
- Password Management
- Password Management: Empty Password

- Password Management: Hardcoded Password
- Password Management: Null Password
- Password Management: Weak Cryptography
- Privacy Violation
- System Information Leak
- System Information Leak: External

对 Bicep 的初始支持（支持的版本：0.21.1）[1]

Microsoft Bicep 是一种用于基础架构即代码 (IaC) 解决方案的开源域特定语言 (DSL)，由 Microsoft 开发，旨在简化和促进 Azure 资源的部署。它充当 Azure 资源管理器 (ARM) 模板之上的抽象层，提供更直观、更易读的方式来定义和管理 Azure 基础架构。凭借 Bicep，用户可以编写简洁易读的代码，用于描述 Azure 资源、配置和依赖项。

缺陷类别的初始覆盖范围包括：

- 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: Hardcoded Secret
- Azure ARM Misconfiguration: HTTPS Not Required
- Azure ARM Misconfiguration: Improper AKS Network Access Control
- Azure ARM Misconfiguration: Improper App Service Access Control
- Azure ARM Misconfiguration: Improper Blob Storage Access Control
- Azure ARM Misconfiguration: Improper Compute VM Access Control
- Azure ARM Misconfiguration: Improper Container Registry Network Access Control
- Azure ARM Misconfiguration: Improper CORS Policy
- Azure ARM Misconfiguration: Improper Custom Role Access Control Policy
- Azure ARM Misconfiguration: Improper DocumentDB Network Access Control
- Azure ARM Misconfiguration: Improper KeyVault Access Control Policy
- Azure ARM Misconfiguration: Improper Security Group Network Access Control
- Azure ARM Misconfiguration: Improper SQL Server Network Access Control
- Azure ARM Misconfiguration: Improper Storage Network Access Control
- Azure ARM Misconfiguration: Insecure Active Directory Domain Service Transport
- Azure ARM Misconfiguration: Insecure App Service Transport
- 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 MySQL Server Transport
- Azure ARM Misconfiguration: Insecure PostgreSQL Server Transport

[1] 需要 Fortify Static Code Analyzer 23.2.0 及更高版本。Bicep 的初始安全内容随 Fortify Static Code Analyzer 23.2.x 一起分发。

- 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: Insecure Storage Account Transport
- Azure ARM Misconfiguration: Insufficient AKS Monitoring
- Azure ARM Misconfiguration: Insufficient Application Insights Logging
- Azure ARM Misconfiguration: Insufficient Application Insights Monitoring
- Azure ARM Misconfiguration: Insufficient Microsoft Defender Monitoring
- Azure ARM Misconfiguration: Insufficient SQL Server Logging
- Azure ARM Misconfiguration: Insufficient SQL Server Monitoring
- Azure ARM Misconfiguration: IoT Hub Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: NetApp Missing Customer-Managed Encryption Key
- Azure ARM Misconfiguration: Public Access Allowed
- 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
- Password Management: Empty Password
- Password Management: Hardcoded Password
- Privacy Violation
- Privacy Violation: Missing Secure Decorator

对 Solidity 的初始支持（支持的版本：0.8.x）[2]

Solidity 是一种面向对象的编程语言，用于在各种去中心化区块链环境中开发智能合约，尤其是在以太坊区块链中。使用 Solidity 编写的智能合约主要运行在以太坊虚拟机 (EVM) 上，但也可以在其他兼容的虚拟机上运行。

缺陷类别的初始覆盖范围包括：

- Authorization Bypass: tx.origin
- Code Correctness: Failing Assertion
- Code Correctness: Reentrancy
- Code Correctness: Typographical Error
- Dead Code
- Denial of Service: External Call
- Dynamic Code Evaluation: Delegatecall
- Integer Overflow
- Obsolete
- Often Misused: Block Values
- Poor Style: Confusing Naming

[2] 需要 Fortify Static Code Analyzer 23.2.0 及更高版本。 Solidty 的初始安全内容随 Fortify Static Code Analyzer 23.2.x 一起分发。

- Poor Style: Variable Never Used
- Solidity Bad Practices: Default Function Visibility
- Solidity Bad Practices: Ether Balance Check
- Solidity Bad Practices: Hardcoded Gas Amount
- Solidity Bad Practices: Lack of Explicit Variable Visibility
- Solidity Bad Practices: Missing Constructor
- Solidity Misconfiguration: Compiler With Known Vulnerabilities
- Solidity Misconfiguration: Floating Pragma
- Unchecked Return Value
- Uninitialized Variable

云基础设施即代码 (IaC)

基础设施即代码是通过代码而非各种手动过程来管理和配置计算机资源的过程。受支持技术的扩展覆盖范围包括用于部署到 Microsoft Azure 的 Terraform 配置以及 AWS Ansible 的配置。与上述服务配置相关的常见问题现已报告给开发人员。

Microsoft Azure Terraform 配置

Terraform 是一种开源 IaC 工具，用于构建、更改云基础设施以及对其进行版本控制。它使用自己的声明性语言，称为 HashiCorp Configuration Language (HCL)。云基础设施被编码入配置文件以描述所需状态。Terraform 提供程序支持 Microsoft Azure 基础设施的配置和管理。改进了对 Terraform 配置的缺陷类别的覆盖范围，包括以下内容：

- Azure Terraform Misconfiguration: App Service Auto Upgrade Disabled
- Azure Terraform Misconfiguration: Improper AKS Access Control
- Azure Terraform Misconfiguration: Improper AKS Network Access Control
- Azure Terraform Misconfiguration: Improper App Service Access Control
- Azure Terraform Misconfiguration: Improper Cognitive Search Network Access Control
- Azure Terraform Misconfiguration: Improper Container Registry Access Control
- Azure Terraform Misconfiguration: Improper Functions Access Control
- Azure Terraform Misconfiguration: Improper MariaDB Network Access Control
- Azure Terraform Misconfiguration: Improper MySQL Network Access Control
- Azure Terraform Misconfiguration: Improper SQL Database Network Access Control
- Azure Terraform Misconfiguration: Improper Storage Account Access Control
- Azure Terraform Misconfiguration: Improper Virtual Network Access Control
- Azure Terraform Misconfiguration: Insecure Disk Storage
- Azure Terraform Misconfiguration: Insecure PostgreSQL Storage
- Azure Terraform Misconfiguration: Insufficient AKS Monitoring
- Azure Terraform Misconfiguration: Insufficient Application Gateway Monitoring
- Azure Terraform Misconfiguration: Insufficient Defender for Cloud Monitoring
- Azure Terraform Misconfiguration: Insufficient Front Door Monitoring
- Azure Terraform Misconfiguration: Insufficient MariaDB Backup
- Azure Terraform Misconfiguration: Insufficient Monitor Logging
- Azure Terraform Misconfiguration: Insufficient Network Watcher Logging
- Azure Terraform Misconfiguration: Insufficient PostgreSQL Monitoring
- Azure Terraform Misconfiguration: Insufficient SQL Database Monitoring
- Azure Terraform Misconfiguration: Redis Cache Auto Upgrade Disabled
- Azure Terraform Misconfiguration: Reduced Virtual Network Availability

- Azure Terraform Misconfiguration: Weak App Service Authentication
- Azure Terraform Misconfiguration: Weak Functions Authentication
- Azure Terraform Misconfiguration: Weak Linux Virtual Machines Authentication
- Azure Terraform Misconfiguration: Weak Service Fabric Authentication

Amazon Web Services (AWS) Ansible 配置

Ansible 是一种开源自动化工具，可提供对各种环境的配置管理、应用程序部署、云配置和节点编排。Ansible 包含支持配置和管理 Amazon Web Services (AWS) 的模块。改进了对 AWS Ansible 配置的缺陷类别的覆盖范围，包括以下内容：

- AWS Ansible Misconfiguration: EFS Missing Customer-Managed Encryption Key
- AWS Ansible Misconfiguration: Improper API Gateway Network Access Control
- AWS Ansible Misconfiguration: Improper ECR Access Control
- AWS Ansible Misconfiguration: Improper ECS Network Access Control
- AWS Ansible Misconfiguration: Improper S3 Access Control
- AWS Ansible Misconfiguration: Improper Stack Access Control
- AWS Ansible Misconfiguration: Insecure API Gateway Transport
- AWS Ansible Misconfiguration: Insecure CloudFront Transport
- AWS Ansible Misconfiguration: Insecure CloudTrail Storage
- AWS Ansible Misconfiguration: Insecure CodeBuild Storage
- AWS Ansible Misconfiguration: Insecure RDS Transport
- AWS Ansible Misconfiguration: Insufficient API Gateway Logging
- AWS Ansible Misconfiguration: Insufficient CloudFront Logging
- AWS Ansible Misconfiguration: Insufficient Lambda Logging
- AWS Ansible Misconfiguration: Insufficient RDS Backup
- AWS Ansible Misconfiguration: Insufficient S3 Backup
- AWS Ansible Misconfiguration: Insufficient S3 Logging
- AWS Ansible Misconfiguration: Insufficient S3 Monitoring
- AWS Ansible Misconfiguration: Insufficient Stack Monitoring
- AWS Ansible Misconfiguration: Privileged Batch Container
- AWS Ansible Misconfiguration: RDS Auto-Upgrade Disabled
- AWS Ansible Misconfiguration: RDS Missing Customer-Managed Encryption Key
- AWS Ansible Misconfiguration: Reduced CloudFront Availability
- AWS Ansible Misconfiguration: Reduced EC2 Availability
- AWS Ansible Misconfiguration: Reduced ELB Availability
- AWS Ansible Misconfiguration: Weak IAM Password Policy

2023 Common Weakness Enumeration (CWE™) Top 25

Common Weakness Enumeration (CWE™) Top 25 Most Dangerous Software Weaknesses (CWE Top 25) 于 2019 年引入，取代了 SANS Top 25。2023 CWE Top 25 于 6 月发布，是使用启发式公式确定的，该公式可规范化过去两年中向国家漏洞数据库 (NVD) 报告的漏洞的频率和严重程度。对于要针对 NVD 中最常报告的关键漏洞确定审核优先级的客户，为了向其提供支持，我们增加了 Fortify Taxonomy 与 2023 CWE Top 25 之间的关联。

OWASP API Security Top 10 2023

Open Worldwide Application Security Project (OWASP) API Security Top 10 2023 提供了 2023 年影响 API 的首要安全风险列表。该列表旨在提高人们对 API 安全缺陷的认识，并帮助那些参与 API 开发和维护的人员，例如所有需要保护 Web API 的开发人员、设计人员、架构师、经理和/或组织。

OWASP API Security Top 10 重点关注影响 Web API 的缺陷，它并非旨在仅仅单独使用，而是旨在与其他标准和最佳实践结合使用，以全面记录所有相关风险。例如：它应该与 OWASP Top 10 结合使用，以识别与输入验证（例如注入）相关的问题。为了向希望降低 Web 应用程序风险的客户提供支持，我们增加了 Fortify Taxonomy 与新发布的 OWASP API Security Top 10 2023 之间的关联。

Center for Internet Security (CIS) 基准

Center for Internet Security (CIS) 基准是社区开发的安全配置建议的集合，这些建议映射到 CIS Critical Security Controls。这些建议旨在确保云基础设施的安全并证明其符合行业标准。CIS 基准持续更新，以适应所涵盖的超过 25 个供应商产品系列的不断变化的网络安全状态。支持的产品系列包括：

- Amazon Elastic Kubernetes Service (EKS) Benchmark v1.3.0
- Amazon Web Services Foundations Benchmark v2.0.0
- Azure Kubernetes Service (AKS) Benchmark v1.3.0
- Google Cloud Computing Platform Benchmark v2.0.0
- Google Kubernetes Engine (GKE) Benchmark v1.4.0
- Kubernetes Benchmark v1.7.1
- Microsoft Azure Foundations Benchmark v2.0.0

Smart Contract Weakness Classification (SWC)[3]

Smart Contract Weakness Classification (SWC) 是一个对智能合约中的漏洞进行分类和解释的系统框架。它提供了一种标准化的方法来理解和解决这些在以太坊等区块链上运行的自动执行代码片段的缺陷。值得注意的是，SWC 注册表的内容自 2020 年以来尚未全面更新，因而存在已知的不完整性、错误和重要遗漏。为了支持想要降低智能合约风险的客户，我们添加了 Fortify Taxonomy 与当前版本 SWC 的关联。

[3] 需要从 Fortify Static Code Analyzer 23.2.0 及更高版本进行扫描。

杂项勘误表

在此版本中，我们已投入大量资源来确保能够减少误报问题的数量、重构以实现一致性并提高客户审核问题的能力。此外，客户可能还会发现与以下各项相关的已报告问题发生了变化：

弃用低于 Fortify Static Code Analyzer 20.x 的版本

正如您在 2022.4 版本中所观察到的，我们将继续支持 Fortify Static Code Analyzer 的最后四个主要版本。因此，这将是支持低于 Fortify Static Code Analyzer 20.x 版本的最后一个规则包版本。下次发布新版本时，低于 Fortify Static Code Analyzer 20.x 的版本将不加载最新的规则包。在此情况下，将需要对规则包进行降级或升级 Fortify Static Code Analyzer 版本。以后发布新版本时，我们将继续支持 Fortify Static Code Analyzer 的最后四个主要版本。

误报改进

此版本仍在继续努力改进，消除误报。除了其他改进之外，客户可能还会发现以下方面的误报得到了进一步消除：

- *ASP.NET MVC Bad Practices: Optional Submodel With Required Property* – 消除了与 ASP.NET 应用程序中的虚拟字段相关的该误报
- *Code Correctness: Double-Checked Locking* – 消除了 Java 应用程序中的该误报
- *Cross-Site Request Forgery* – 消除了在 .NET 应用程序中使用“AntiForgery.GetHtml()”或“Html.AntiForgeryToken()”的 HTML 表单的该误报
- *Cross-Site Scripting: Persistent* – 消除了 Django 应用程序中与“cycle”标签相关的该误报
- *Double Free* – 消除了从 Boost 库使用“throw_error()”的 C/C++ 应用程序中的该误报
- *HTML5: Missing Content Security Policy* – 消除了 Java 应用程序中的该误报
- *JSON Injection* – 消除了 PHP 应用程序中的该误报
- *Mass Assignment: Insecure Binder Configuration* – 消除了 .NET 应用程序中与 Enum 类型相关的该误报
- *Often Misused: File System* – 消除了与 C++ 应用程序中的“GetFullPathNameW()”和类似函数调用相关的该误报
- *Path Manipulation* – 消除了使用 Amazon AWS SDK 的 Java 应用程序中的该误报
- *Type Mismatch: Signed to Unsigned* – 消除了与 C/C++ 应用程序中布尔值相关的该误报
- *Unreleased Resource* – 消除了在 C++ 应用程序中使用“CreateFileW()”时出现的该误报

类别更改

当缺陷类别名称发生更改时，将先前的扫描与新的扫描合并后的分析结果将导致增加/移除某些类别。

为了提高一致性，我们已对以下 14 个类别进行了重命名：

删除的类别	添加的类别
AWS CloudFormation Misconfiguration: Insecure Elasticache Storage	AWS CloudFormation Misconfiguration: Insecure ElastiCache Storage
AWS CloudFormation Misconfiguration: Insecure Elasticache Transport	AWS CloudFormation Misconfiguration: Insecure ElastiCache Transport
AWS Terraform Misconfiguration: Elasticache Missing Customer-Managed Encryption Key	AWS Terraform Misconfiguration: ElastiCache Missing Customer-Managed Encryption Key
Azure Terraform Bad Practices: AKS Cluster Missing Host-Based Encryption	Azure Terraform Misconfiguration: AKS Cluster Missing Host-Based Encryption
Azure Terraform Bad Practices: Azure MySQL Server Missing Infrastructure Encryption	Azure Terraform Misconfiguration: MySQL Missing Infrastructure Encryption
Azure Terraform Bad Practices: Azure PostgreSQL Server Missing Infrastructure Encryption	Azure Terraform Misconfiguration: PostgreSQL Missing Infrastructure Encryption
Azure Terraform Bad Practices: Missing Azure Storage Infrastructure Encryption	Azure Terraform Misconfiguration: Storage Account Missing Infrastructure Encryption
Azure Terraform Bad Practices: Missing SQL Database Backup Encryption	Azure Terraform Misconfiguration: SQL Server Backup Missing Encryption
Azure Terraform Bad Practices: Scale Set Missing Host-Based Encryption	Azure Terraform Misconfiguration: Scale Set Missing Host-Based Encryption
Azure Terraform Bad Practices: VM Missing Host-Based Encryption	Azure Terraform Misconfiguration: VM Missing Host-Based Encryption
GCP Terraform Bad Practices: Overly Permissive Service Account	GCP Terraform Misconfiguration: Improper Compute Engine Access Control
GCP Terraform Misconfiguration: Weak Key Management	GCP Terraform Misconfiguration: Compute Engine Missing Customer-Managed Encryption Key
Kubernetes Bad Practices: Improper Admission Controller Access Control	Kubernetes Misconfiguration: Improper Admission Controller Access Control
Kubernetes Misconfiguration: Missing Service Account Admission Controller	Kubernetes Misconfiguration: Missing ServiceAccount Admission Controller

Fortify Priority Order 变更

为了提高与缺失的客户管理的加密密钥相关的漏洞类别之间的一致性，以下 20 个类别的 Fortify Priority Order 已更改为“low”：

- *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: 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 Terraform Misconfiguration: AKS Cluster Missing Customer-Managed Key*
- *Azure Terraform Misconfiguration: Azure Disk Snapshot Missing Customer-Managed Key*
- *Azure Terraform Misconfiguration: Container Registry Missing Customer-Managed Key*
- *Azure Terraform Misconfiguration: Cosmos DB Missing Customer-Managed Key*
- *Azure Terraform Misconfiguration: Managed Disk Missing Customer-Managed Key*
- *Azure Terraform Misconfiguration: Shared Image Missing Customer-Managed Key*
- *Azure Terraform Misconfiguration: SQL Database Missing Customer-Managed Key*
- *Azure Terraform Misconfiguration: Storage Account Missing Customer-Managed Key*
- *Azure Terraform Misconfiguration: Storage Encryption Scope Missing Customer-Managed Key*
- *Azure Terraform Misconfiguration: VM Storage Missing Customer-Managed Key*
- *GCP Terraform Misconfiguration: Compute Engine Missing Customer-Managed Encryption Key*

Fortify SecureBase [Fortify WebInspect]

Fortify SecureBase 不仅能对成千上万的漏洞进行检查，还可通过策略引导用户通过 SmartUpdate 立即获得以下更新。

漏洞支持

Insecure Deployment: Unpatched Application

vBulletin 版本 5.6.0 至 5.6.8 中的预授权 Remote Code Execution (RCE) 漏洞已标识为 CVE-2023-25135。vBulletin 是一种用于构建动态在线社区和论坛的流行软件，它会错误地清理用户提供的输入以进行未经身份验证的反序列化。此问题使攻击者能够在服务器上执行任意代码、滥用应用程序逻辑或发起拒绝服务 (DoS) 攻击。此版本包括用于检测目标服务器上是否存在此漏洞的检查功能。

原型污染：服务器端

当攻击者可以操纵对象的原型时，就会发生服务器端原型污染。这在基于原型的语言（例如 JavaScript）中是可能的，该语言允许在运行时更改属性和方法。该漏洞的严重性取决于应用程序中使用污染对象的位置。攻击包括 Denial of Service、更改应用程序配置以及某些情况下的 Remote Code Execution。此版本包括用于检测 Web 应用程序中是否存在原型污染的检查功能。

合规性报告

2023 Common Weakness Enumeration (CWE™) Top 25

Common Weakness Enumeration (CWE™) Top 25 Most Dangerous Software Weaknesses (CWE Top 25) 于 2019 年引入，取代了 SANS Top 25。2023 CWE Top 25 于 6 月发布，是使用启发式公式确定的，该公式可规范化过去两年中向国家漏洞数据库 (NVD) 报告的漏洞的频率和严重程度。此次 SecureBase 更新所包括的检查，要么直接映射到标识为 CWE Top 25 的类别，要么映射到通过 "ChildOf" 关系与 Top 25 中的 CWE-ID 关联的 CWE-ID。

OWASP API Security Top 10 2023

Open Worldwide Application Security Project (OWASP) API Security Top 10 2023 提供了 2023 年影响 API 的首要安全风险列表。该列表旨在提高人们对 API 安全缺陷的认识，并帮助参与 API 开发和维护的人员，例如所有需要保护 Web API 的开发人员、设计人员、架构师、经理和组织。
OWASP API Security Top 10 重点关注影响 Web API 的缺陷，它并非旨在单独使用，而是要与其他标准和最佳实践结合使用，以全面记录所有相关风险。例如：将 OWASP API Security Top 10 2023 与 OWASP Top 10 结合使用以识别与输入验证（例如注入）相关的问题。此 SecureBase 更新包括一个新的合规性报告模板，可提供 OWASP API Security Top 10 2023 类别与 WebInspect 检查之间的关联。

策略更新

2023 CWE Top 25

在受 WebInspect SecureBase 支持的策略列表中，添加了一项自定义策略以纳入与 2023 CWE Top 25 相关的检查。

OWASP API Security Top 10 2023

在 WebInspect SecureBase 的支持策略列表中，我们添加了一项自定义策略以纳入与 OWASP API Security Top 10 2023 相关的检查。该策略包含部分可用的 WebInspect 检查，支持客户运行特定于合规性的 WebInspect 扫描。

杂项勘误表

在此版本中，我们已投入大量资源来进一步减少误报数量，并提高客户审核问题的能力。此外，客户可能还会发现与以下各项相关的报告结果发生了变化。

LDAP Injection

此版本包括对 LDAP Injection 检查的改进，以减少误报并提高结果的准确性。

SSL 证书主机名差异

SSL 证书主机名差异检查报告内容现在包含更详细的信息，可帮助客户针对此安全问题应用正确的解决方案。

通过检查输入进行积极覆盖

对于某些 WebInspect 检查，可以启用积极覆盖，引导 WebInspect 发送针对更广泛端点的更长攻击列表。此版本包括对这些检查的改进，使客户能够通过更改检查输入来配置积极覆盖，而不是向扫描策略添加单独的检查。具有积极覆盖能力的检查包括以下内容：*Log4Shell*、*JNDI Reference Injection*、*Server-Side Request Forgery*、*OS Command Injection* 和 *Server-Side Prototype Pollution*。启用积极覆盖的检查可提供更准确的扫描，但是，重要的是要考虑请求数量和扫描时间可能会大幅增加。因此，Fortify 强烈建议您在单独的策略中启用积极覆盖的情况下运行检查，而不进行其他检查。

Web Server Misconfiguration: 不受保护的文件

此版本包含一个微小错误修复，以改进对 Java 相关配置文件的检测。

Fortify Premium Content

此研究团队负责构建、扩展并维护我们的核心安全情报产品之外的各种资源。

2023 CWE Top 25

除新关联之外，此版本还包含附带 2023 CWE Top 25 支持的 Fortify Software Security Center 新报告包，您可从 Premium Content 下的 Fortify 客户支持门户下载该报告包。

OWASP API Security Top 10 2023

除新关联之外，此版本还包含附带 OWASP API Security Top 10 支持的 Fortify Software Security Center 新报告包，您可从 Premium Content 下的 Fortify 客户支持门户下载该报告包。

Fortify Taxonomy：软件安全错误

要访问 Fortify Taxonomy 站点以查看对新增类别支持的说明，请访问：https://vulncat.fortify.com。

联系 Fortify 技术支持

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

联系 SSR

Alexander M. Hoole
Software Security Research 高级经理
OpenText Fortify
hoole@opentext.com
+1 (650) 427-9973

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