Micro Focus

Fortify 软件安全内容

2020 更新 3

2020 年 9 月 25 日

关于 Micro Focus Fortify 软件安全研究

Fortify 软件安全研究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)、Fortify WebInspect 和 Fortify Application Defender）的安全情报。现在，Micro Focus Fortify 软件安全内容支持 27 种编程语言的 1,032 个漏洞类别，且涵盖的单独 API 超过一百万个。

如需了解详细信息，请访问：https://software.microfocus.com/en-us/software/security-research

Fortify 软件安全研究 (SSR) 团队荣幸地宣布，我们即将推出 Fortify Secure Coding Rulepack（2020.2.0 英文版）、Fortify WebInspect SecureBase（可通过 SmartUpdate 获取）和 Fortify Premium Content 的更新。

Micro Focus Fortify Secure Coding Rulepack [SCA]

这次发行的 Fortify Secure Coding Rulepack 可以检测 27 种编程语言的 815 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：

.NET System.Text.Json 支持（版本 3.1）[1]

System.Text.Json 命名空间提供了高速、内置且符合标准的 Newtonsoft Json.NET 软件包替代方案。System.Text.Json 命名空间还提供类型以读取和写入编码为 UTF-8 的 JSON 文本。支持包括通过各种序列化/反序列化方法得到的数据流，以及扩大对以下类别的覆盖范围：

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

Kotlin 协程和标准库支持 [2]

扩展 Kotlin 标准库的覆盖范围，包括对协程的支持。协程提供丰富且灵活的 API 用于开发并发和异步代码。此更新在使用协程的应用程序中增加了改进的数据流分析。

Kotlin Java 互操作性和 Android 改进 [3]

针对包含 Java 和 Kotlin 源代码的项目以及使用 Kotlin 开发的 Android 应用程序的现有漏洞类别，改进了语言支持。

Java 14 改进 [4]

针对现有漏洞类别进行了改进，以支持 Java 14 之前的所有版本更改。

ECMAScript 版本更新 [5]

已升级 ECMAScript 版本的规则覆盖范围，支持 ECMAScript 2020 之前的所有版本。

---

[1] 由于 System.Text.Json 命名空间的设计，其不易出现 Newtonsoft Json.NET 软件包中报告的 JSON 注入问题。
[2] 此更新不支持使用 Select 表达式等的实验性协程 API。
[3] 需要 SCA 20.2 或更高版本。
[4] 需要 SCA 20.2 或更高版本。
[5] 完整支持需要 SCA 20.2 或更高版本。

Go Logrus（版本 1.6.0）

Logrus 是 Go 的结构化记录器，与标准库记录器完全 API 兼容。Logrus 是一个常见的日志记录包，可能会给企业应用程序带来日志记录相关的风险。支持的类别包括：

- Poor Logging Practice: Use of a System Output Stream
- Privacy Violation
- System Information Leak

Dockerfile 支持 [6]

Docker 允许在沙盒环境（也称为容器）中打包代码及其依赖项，以便在任何计算环境中执行。Dockerfile 是容器映像的配置。此次更新解决了 Dockerfile 中最常见的漏洞。
初始支持包括现有的“Password Management: Password in Configuration File”类别，以及以下新类别：

- Dockerfile Misconfiguration: Default User Privilege
- Dockerfile Misconfiguration: Privileged Container
- Dockerfile Misconfiguration: Privileged Port
- Dockerfile Misconfiguration: Sensitive Host Discovery
- Dockerfile Misconfiguration: SSH Service

DISA STIG 4.11

为了向联邦客户提供合规性方面的支持，我们增加了 Micro Focus Fortify Taxonomy 与美国国防信息系统局 (DISA) 应用安全与开发 STIG 版本 4.11 之间的关联。

杂项勘误表

在此版本中，我们继续投入资源，以确保能够减少误报问题的数量，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已报告问题发生了变化：

SCA 17.X 之前版本的弃用通知：
请注意，这将是支持 SCA 17.x 之前版本的最后一个规则包版本。对于下一个版本，SCA 17.x 之前的版本将不加载规则包。这将需要对规则包进行降级或升级 SCA 版本。
对于以后的版本，我们将继续支持 SCA 的最后四个主要版本。

弃用的映射 - 安全技术实施指南 3.x 版 (STIG 3.x)：
软件安全研究团队多年来基于 Fortify 类别以及外部标准和最佳实践不断创建各种映射。因此，我们现在已拥有大量支持不同标准的版本，尽管许多旧版本将不再使用。
因此，我们现在已经使用 externalmetadata.xml 文件中的新 XML 属性将 STIG 4.0 之前的版本标记为“obsolete”。此更改意味着自 20.2 版本起的 Fortify Software Security Center 和 SCA 应用程序默认将不会显示这些旧版本。如果需要检查这些旧版本，解决方法是更改映射，将 obsolete 设置为“false”，或者完全删除该属性。

[6] 需要 SCA 20.2 或更高版本。

黑名单和白名单命名：
为了遵循该领域的趋势，作为 Micro Focus Fortify 整体的一部分，软件安全研究团队决定使用更中性且语义更具体的术语来取代术语“blacklist”和“whitelist”。我们决定分别改用更具描述性的术语“deny list”和“allow list”。
此更改还意味着以下类别名称已更改，可能会导致已删除和新的问题：

- “Deserialization Bad Practice: Blacklist”现在为“Deserialization Bad Practice: Deny List”
- “Django Bad Practices: Blacklisted Attributes”现在为“Django Bad Practices: Attributes in Deny List”
- “Setting Manipulation: User-Controlled Whitelist”现在为“Setting Manipulation: User-Controlled Allow List”

弱加密：不安全运行模式更新
我们认为使用对称加密的密码块链 (CBC) 模式已不再安全。从此版本起，对于当前支持的库，任何使用 CBC 的行为都将被报告为“弱加密：不安全运行模式”。考虑到 BEAST、POODLE、Zombie POODLE、Goldendoodle、LUCKY13 以及 Padding Oracle 攻击，我们根据目前已知的加密研究确定 CBC 不再安全。

误报改善：
我们继续听取客户的意见，并努力改善误报率。在此版本中，我们致力于下列工作以减少误报次数：

- Cross-Site Request Forgery
  减少对其他类型的 CSRF 令牌，以及 ASP.NET 隐藏字段内的令牌的说明
- Server-Side Request Forgery
  删除部分 Spring 应用程序
- Poor Logging Practice: Use of a System Output Stream
  当上下文基于命令行应用程序或实用程序时删除。
- 死代码：未使用字段
  Lambda 内部
- Privacy Violation
  消除了与 Java、Scala 和 Kotlin Lambda 内部的密码有关的重复问题
- Java 系统属性现在被单独处理为内置属性，减少了许多情况下的误报，并有可能发现新的系统信息泄漏。

预览：[7]
对空指针引用和冗余空值检查的规则进行了修改以降低误报率。特别是增加了编写自定义规则以处理内部空值检查函数的功能。
然而，目前尚不清楚这些优势是否具有普遍性。因此，SCA 20.2 中的这些改进在默认情况下是关闭的。如果您想测试这些改进，请联系客户支持。

[7] 需要 SCA 20.2 或更高版本。

Micro Focus 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

Weak Cryptographic Signature: User-Controlled Key Source

密钥混淆攻击对 JSON Web 令牌 (JWT) 的完整性构成威胁。JWT 实现没有仔细选择密钥源来验证从客户端接收的令牌，从而使得攻击者可以伪造令牌有效负载，因此容易受到这些攻击。此版本包含一项检查，以评估针对密钥混淆攻击的 JWT 使用情况。

JSON Web Token: Missing Protection Claims

不具备保护声明的 JSON Web 令牌（如“jti”、“exp”和“aud”）可能容易受到模拟和重播攻击。此版本包含多项检查，用于检测扫描期间遇到的令牌是否缺少这些声明。

Access Control: Authorization Bypass

JSON Web 令牌 (JWT) 提供保护以防止数据篡改，因为它包含的信息是用 HMAC 或 RSA 算法进行数字签名的。未验证签名的应用程序可能容易受到未经授权的访问和模拟攻击。此版本包括多项检查，以评估 JWT 实现是否缺少签名以及是否会受到任意签名攻击。

OAuth2: Insufficient state Parameter Entropy

如果 OAuth2 协议实现不使用具有足够熵的“state”参数值，则容易受到模拟攻击。此版本包含一项检查，以评估状态参数是否未能满足足够的熵要求。

OAuth2: Insufficient Refresh Token Revocation

如果授权服务器在重新颁发新的访问令牌之前未验证客户端 ID、客户端机密和刷新令牌有效期，则可能容易受到模拟攻击。此版本包含一项检查，以检测刷新令牌是否可用于生成不含客户端 ID 和客户端机密的多个访问令牌。

OAuth2: Insufficient Authorization Code Expiration

未能及时终止授权代码的 OAuth2 授权服务器可能容易受到模拟攻击，因为它们可能会继续为被盗的授权代码授予新的有效访问令牌。此版本包含一项检查，以检测授权代码滥用情况。

Cross-Site Request Forgery

未在 OAuth2 授权流中使用“state”参数的应用程序容易受到模拟客户端的未授权操作的影响。此外，将 cookie 存储用于 JSON Web Token (JWT) 的应用程序也面临与 CSRF 攻击类似的风险。此版本包括多项检查，以检测易受 CSRF 攻击的 JWT 和 OAuth2 实例。

Weak Cryptographic Signature: Insufficient Key Size

JSON Web Token 包含数字签名的信息，以证明有效负载的完整性。 对称算法（如使用弱机密的 HS256）可能会遭到暴力攻击，使攻击者能够生成任意有效负载并使用有效密钥对其进行签名。此版本包含一项检查，用于检测是否使用了 HS256 之类的弱算法来对令牌进行签名。

SQL Injection[8]

SQLite 是一个 C 库，它提供了一个轻量级的基于磁盘的数据库，不需要单独的服务器进程，并且允许使用 SQL 查询语言的非标准变体访问数据库。该版本增强了对 WebInspect SQL Injection 的检查，以使用 SQLite 库 3.16.0 或更高版本在 Web 应用程序中检测 SQL injection 漏洞。

合规性报告

DISA STIG 4.11

为了向我们的联邦客户提供合规性支持，此版本包含 WebInspect 检查与最新版本的美国国防信息系统局应用安全与开发 STIG 版本 4.11 之间的关联。

策略更新

DISA STIG 4.11

在 WebInspect SecureBase 内现有的支持策略列表中，已添加为纳入与 DISA STIG 4.11 相关的检查而自定义的策略。

Web API

在 WebInspect SecureBase 内现有的支持策略列表中，已添加为涵盖与 API 安全评估相关的检查而自定义的策略。

[8] 将 SQLite 的数据库类型信息作为报告内容的一部分需要 WebInspect 20.2 或更高版本。

杂项勘误表：

在此版本中，我们继续投入资源，以确保能够减少误报问题的数量，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已报告问题发生了变化：

Insecure Transport: 弱 SSL 密码：
当 SSLv3 或 TLS1.0 以及 CBC 密码同时被检测到时，已将 ID 11285 标识的检查修改为将 CBC 模式密码的所有实例报告为弱，而在弱 SSL 协议中报告内容简介。考虑到 POODLE、GoldenPOODLE 和 ZombieDOODLE 等漏洞，不再建议将 CBC 模式密码用于传输层安全。

黑名单和白名单命名：
为了遵循该领域的趋势，作为 Micro Focus Fortify 整体的一部分，软件安全研究团队决定使用更中性且语义更具体的术语来取代术语“blacklist”和“whitelist”。我们决定在“检查安全内容”中分别改用更具描述性的术语“deny list”和“allow list”。

Micro Focus 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。

DISA STIG 4.11

除新关联之外，此版本还包含附带 DISA STIG 4.11 支持的 Fortify 软件安全中心新报告包，您可从 Premium Content 下的 Fortify 客户支持门户下载该报告包。

Micro Focus 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/zh-cn。如果客户要从旧站点上查找最新支持的更新，可从 Micro Focus Fortify 支持门户获取此更新内容。

联系 Fortify 技术支持
Micro Focus Fortify
https://softwaresupport.softwaregrp.com/
+1 (844) 260-7219

联系 SSR
Alexander M. Hoole
软件安全研究团队经理
Micro Focus Fortify
hoole@microfocus.com
+1 (650) 258-5916

2020 年 9 月 25 日