Red Hat build of Apicurio Registry 2.5

Apicurio Registry 2.5 发行注记

红帽构建的 Apicurio Registry 中的新功能

Last Updated: 2024-05-30

Red Hat build of Apicurio Registry 2.5 Apicurio Registry 2.5 发行注记

红帽构建的 Apicurio Registry 中的新功能

法律通告

摘要

描述红帽构建的 Apicurio Registry 产品，并提供了有关本发行版本中新内容的最新详情。

目录

前言

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 CTO Chris Wright 的信息。

对红帽文档提供反馈

我们感谢您对我们文档的反馈。

要改进，创建一个 JIRA 问题并描述您推荐的更改。提供尽可能多的详细信息，以便我们快速解决您的请求。

前提条件

- 您有一个红帽客户门户网站帐户。此帐户可让您登录到 Red Hat Jira Software 实例。
  如果您没有帐户，系统会提示您创建一个帐户。

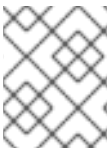
流程

1. 单击以下链接： 创建问题。
2. 在 **Summary** 文本框中输入问题的简短描述。
3. 在 **Description** 文本框中提供以下信息：
   - 找到此问题的页面的 URL。
   - 有关此问题的详细描述。
     您可以将信息保留在任何其他字段中的默认值。
4. 点 **Create** 将 JIRA 问题提交到文档团队。

感谢您花时间来提供反馈。

第 1 章 APICURIO REGISTRY 2.5 发行注记

红帽构建的 Apicurio Registry 是标准事件模式和 API 设计的数据存储，它基于 Apicurio Registry 开源社区项目。



注意

红帽构建的 Apicurio Registry 是 Red Hat Integration Service Registry 的新产品名称。红帽构建的 Apicurio Registry 2.x 和 Red Hat Integration Service Registry 2.x 的功能相同。

您可以使用 Apicurio Registry 通过 Web 控制台、REST API、Maven 插件或 Java 客户端管理和共享数据的结构。例如，客户端应用程序可以对或从 Apicurio Registry 动态推送或拉取最新的 schema 更新，而无需重新部署。您还可以创建可选规则来管理 Apicurio Registry 内容随时间变化的方式。这些规则包括验证内容、工件引用的完整性，以及向后兼容或转发 schema 或 API 版本的兼容性。

1.1. APICURIO REGISTRY 安装选项

您可以使用以下数据存储选项之一在 OpenShift 上安装 Apicurio Registry：

- PostgreSQL 数据库
- Red Hat AMQ Streams

如需了解更多详细信息，请参阅在 OpenShift 上安装和部署红帽构建的 Apicurio Registry。

1.2. APICURIO REGISTRY 支持的平台

Apicurio Registry 2.5 支持以下核心平台：

- Red Hat OpenShift Container Platform: 4.15, 4.14, 4.13, 4.12
- Red Hat OpenShift Service on AWS: 4.13
- Microsoft Azure Red Hat OpenShift: 4.13
- PostgreSQL: 15, 14, 13, 12
- Red Hat AMQ Streams: 2.6, 2.5, 2.2
- OpenJDK: 17, 11

如需了解更多详细信息，请参阅以下文章：

- 红帽构建的 Apicurio Registry 支持的配置。

1.2.1. 支持的与其他产品集成

Apicurio Registry 2.5 还支持与以下产品集成：

- Red Hat Single Sign-On (RH-SSO) 7.6
- 红帽构建的 Debezium 2.3

1.2.2. Operator 元数据版本

有关用于安装和部署 Apicurio Registry 的对应 Service Registry Operator 元数据版本的详情，请查看以下文章：

- Red Hat Integration - Service Registry Operator 元数据版本 。

1.3. APICURIO REGISTRY 新功能

Apicurio Registry 2.5 包括以下新功能：

Apicurio Registry 核心新功能

升级到 Quarkus 3.x

- Apicurio Registry 服务器运行时已从 Quarkus 2.x 升级到 Quarkus 3.x。此升级提供了更高的安全性、性能和维护。如需了解更多详细信息，请参阅 https://quarkus.io/quarkus3/。Apicurio Registry 2.5 基于 Quarkus 3.2 构建。

avro SerDes 改进

- 在使用 Apache Avro serializers/deserializers 时，支持生成带有 null 字段的模式。如需了解更多详细信息，请参阅 Registry-3862。

模式缓存容错

- 添加了选项以使用现有模式缓存条目，而不是在 schema 缓存加载失败时抛出错误。如需了解更多详细信息，请参阅 Registry-3807。

解引用工件内容

- 在某些情况下，返回带有引用的内容的工件内容可能会很有用。在这些情况下，Core Registry API v2 添加了对某些操作中 **dereference** 查询参数的支持。如需了解更多详细信息，请参阅 Apicurio Registry v2 core REST API 文档 。
- 在 2.5.11 之前，只有在 API 操作中指定 **dereference** 参数时，才会对 Avro 和 Protobuf 工件实现这个支持。这个参数不支持任何其他工件类型。如需了解更多详细信息，请参阅 Registry-2865。

注意

对于 Protobuf 工件，只有在所有模式都属于同一软件包时才支持取消引用内容。

从 2.5.11 开始，支持已扩展到 JSON 架构、Async API 和 OpenAPI。

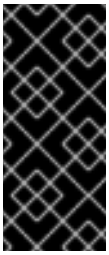
注意

对于 JSON Schema 工件，仅针对引用独立工件完整内容的工件支持解引用内容。**不支持在** 工件引用第二个工件的一部分时取消引用

Apicurio Registry Maven 插件改进

- 添加 选项，**以跳过** Maven 插件中的注册目标。如需了解更多详细信息，请参阅 Registry-3817。

- 使用 **pom.xml** 文件中的 **autoRef** 选项，自动检测 Maven 插件中的引用。如需了解更多详细信息，请参阅 Registry-3439。这是一个技术预览功能。



重要

技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

Apicurio Registry Operator 新功能

改进了对 SQL 数据源配置的支持

- Apicurio Registry Operator 支持使用环境变量配置 SQL 数据源，作为 **spec.configuration.sql.dataSource** 字段的替代选择。现在，您可以在 **ApicurioRegistry** 自定义资源中使用 Kubernetes secret 而不是明文提供 SQL 凭证。如需了解更多详细信息，请参阅 https://access.redhat.com/solutions/7059053。

- 这个版本改进了 Apicurio Registry Operator，以更好地支持这个用例。现在，您可以使用 **spec.configuration.sql.dataSource** 和 **spec.configuration.env** 字段来定义配置的部分。例如，以下配置现在有效：

```
apiVersion: registry.apicur.io/v1
kind: ApicurioRegistry
metadata:
  name: myregistry
spec:
  configuration:
    persistence: sql
  sql:
    dataSource:
      url: "jdbc:postgresql://..."
      userName: "postgres-user"
    env:
      - name: REGISTRY_DATASOURCE_PASSWORD
        valueFrom:
          secretKeyRef:
            name: postgres-secret
            key: password
```

Operator 还检测到这种类型的配置，并在无需其他用户干预的情况下立即应用。

Apicurio Registry 用户文档和示例

文档库已使用版本 2.5 中的新功能更新：

- 在 OpenShift 上安装并部署红帽构建的 Apicurio Registry
- 迁移红帽构建的 Apicurio Registry 部署
- 红帽构建的 Apicurio Registry 用户指南
- Apicurio Registry v2 core REST API 文档

开源演示应用程序也已更新：

- https://github.com/Apicurio/apicurio-registry-examples

1.4. APICURIO REGISTRY 已弃用的功能

Apicurio Registry 核心已弃用的功能

- *Confluent Schema Registry API 版本 6 (compatibility API)* : Apicurio Registry 目前支持独立端点上的 Confluent Schema Registry API 的两个版本：版本 6 和版本 7。v6 API 端点已弃用，并将在以后的发行版本中删除。确保将对 v6 API 端点的所有引用替换为对 v7 API 端点的引用。

- *Apicurio Registry Core API 版本 1* : Apicurio Registry Core API 的原始版本 1 的支持现已弃用。此 v1 旧 API 将在下一个主发行版本中删除。

- *动态日志级别配置* ： v2 Apicurio Registry Core API 中弃用了 **/admin/loggers** 和 **/admin/loggers/{logger}** API 端点。这些端点将在以后的发行版本中被删除。

- *Registry V1 export 工具*: Apicurio Registry 对命令行导出工具的支持现已弃用。导出工具用于将 Apicurio Registry 1.x 中的数据导出为可导入到 2.x 的格式，将不再发布或维护。所有客户都应该已从 1.x 升级到 2.x。

Apicurio Registry Operator 已弃用的功能

- *JAVA_OPTIONS 环境变量* ： **JAVA_OPTIONS** 环境变量不再是配置 Apicurio Registry 的 Java 选项的首选方法。您可以使用 **JAVA_OPTS_APPEND** 环境变量。**JAVA_OPTS** 环境变量也可用，它取代了 Java 选项的默认内容。但是，最好避免使用 **JAVA_OPTS**，因为它可能会影响一些 Apicurio Registry Operator 功能。

- *通过编辑 Deployment 资源来设置环境变量* ：在以前的版本中，您可以通过直接编辑其 **Deployment** 资源(Apicurio Registry Operator 支持)来为 Apicurio Registry 设置环境变量。现在，您可以使用 **ApicurioRegistry** CRD 文件中的 **spec.configuration.env** 字段来管理环境变量，前面的流程已弃用，并将删除对它的 Operator 支持。确保使用 **spec.configuration.env** 字段来设置 Operator 未设置的所有环境变量。

- *为未启用的功能保留环境变量* ： Apicurio Registry Operator 设置环境变量以启用和配置各种功能，如使用 Kafka 存储时 Salted Challenge Response Authentication Mechanism (SCRAM)安全性。当禁用此类功能时，Operator 当前会保留关联的环境变量，这可能会导致问题。此类环境变量的保留已弃用，Operator 对它的支持将被删除。确保您的部署不依赖于这些环境变量的保留。

- *环境变量优先级* ： Apicurio Registry Operator 可能会尝试设置已在 **spec.configuration.env** 字段中明确指定的环境变量。如果环境变量具有冲突的值，则 Apicurio Registry Operator 设置的值会默认具有优先权。此行为将在以后改变，以便用户覆盖 Operator 设置的大部分环境变量。确保您的部署不依赖于原始优先级行为。

1.5. 升级和迁移 APICURIO REGISTRY 部署

您可以在 OpenShift 上自动将 Apicurio Registry 服务器从 Apicurio Registry 2.x 升级到 Apicurio Registry 2.5。没有从 Apicurio Registry 1.x 到 Apicurio Registry 2.x 的自动升级，需要迁移过程。

1.5.1. 更新 2.x 客户端依赖项

这不是更新本发行版本的客户端依赖项所必需的。现有的 Apicurio Registry 2.x 客户端应用程序继续使用 Apicurio Registry 2.5。

但是，在 Apicurio Registry 的下一个发行版本前，您必须更新所有客户端依赖项，以使用最新版本的 Apicurio Registry。客户端依赖项包括 Apicurio Registry Kafka serializers/deserializers (SerDes)、Maven 插件和 Java 客户端应用程序的依赖项。

例如，要更新 Java 客户端应用程序的 Maven 依赖项，请在 **pom.xml** 文件中指定版本，如下所示：

```
<dependency>
    <groupId>io.apicurio</groupId>
    <artifactId>apicurio-registry-client</artifactId>
    <version>2.5.11.Final-redhat-00001</version>
</dependency>
```

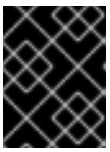
如需了解更多详细信息，请参阅 默认启用旧 REST API 日期格式。

1.5.2. 在 OpenShift 上从 Apicurio Registry 2.x 升级

您可以在 OpenShift 4.11 上从 Apicurio Registry 2.x 升级到 OpenShift 4.12 或更高版本上的 Apicurio Registry 2.5。您必须升级 Apicurio Registry 和 OpenShift 版本，并一次升级 OpenShift 的一个次版本。

先决条件

- 您已在 OpenShift 4.11 或更高版本上安装了 Apicurio Registry 2.x。
- 您已在 Kafka 主题或 PostgreSQL 数据库中备份了现有的 Apicurio Registry 存储数据。如需了解更多详细信息，请参阅在 OpenShift 上安装和部署红帽构建的 Apicurio Registry。



重要

在 OpenShift 上的生产环境中，为了帮助确保在升级前备份存储，最好将 Apicurio Registry 的 Operator 更新批准策略设置为 manual 而不是 automatic。

流程

1. 在 OpenShift Container Platform Web 控制台中，点 **Administration**，然后点 **Cluster Settings**。
2. 点 **Channel** 字段旁边的铅笔图标，然后选择下一个次 **candidate** 版本（例如，从 **stable-4.11** 改为 **candidate-4.12**）。
3. 点 **Save** 然后点 **Update**，等待升级完成。
4. 如果 OpenShift 版本小于 4.13，请重复步骤 2 和 3，然后选择 **candidate-4.13** 或更高版本。
5. 点 **Operators** > **Installed Operators** > **Red Hat Integration - Service Registry**。
6. 确保 **更新频道** 已设置为 **2.x**。
7. 如果 **Update approval** 设为 **Automatic**，则升级应在设置 **2.x** 频道后立即批准并安装。
8. 如果 **Update approval** 设置为 **Manual**，点 **Install**。
9. 等待 Operator 部署好并部署了 Apicurio Registry pod。
10. 验证您的 Apicurio Registry 系统是否已启动并在运行。

其他资源

- 有关如何在 OpenShift Container Platform Web 控制台中设置 Operator 更新频道的更多详细信息，请参阅 更改 Operator 的更新频道。

1.5.3. 从 OpenShift 上的 Apicurio Registry 1.1 迁移

有关从 Apicurio Registry 1.1 迁移到 Apicurio Registry 2.x 的详情，请参阅 迁移 Apicurio Registry 部署的红帽构建。

1.6. APICURIO REGISTRY 解决的问题

表 1.1. 解决了 Apicurio Registry 2.5.11 中的问题

问题	描述
IPT-1059	Service Registry 的身份验证错误被错误地记录在 DEBUG 级别下。
IPT-882	在 API 中实施 dereference 参数支持。

表 1.2. 解决了 Apicurio Registry 2.5.10 中的问题

问题	描述
IPT-1091	Apicurio Registry Operator 应该支持 JAVA_OPTS_APPEND 和 JAVA_OPTIONS （已弃用）环境变量。
IPT-1092	Apicurio Registry 升级会破坏 Confluent v6 兼容性 API。

表 1.3. 解决了 Apicurio Registry 2.5.9 中的问题

问题	描述
IPT-1071	当使用 KafkaSQL 存储和带有 Protobuf 工件升级 Apicurio Registry 时，可能会出现数据丢失。
IPT-1035	Operator 升级到 2.2.3 后 CrashLoop 中的 Apicurio Registry Pod。
registry-4417	无法从 Apicurio Registry 中正确删除孤立的内容。
registry-4283	当为单个工件创建具有相同名称的两个引用时，Apicurio Registry 服务器应该会失败。
registry-4226	删除所有规则 REST API 操作不会删除 **INTEGRITY** 规则（仅限KafkaSQL 存储）。
registry-4215	带有不同字段顺序的 avro 以规范形式被视为相等。
Registry-4107	以只读模式在 Apicurio Registry web 控制台中不会显示有效性、兼容性和完整性规则值。

表 1.4. 解决了 Apicurio Registry 2.5.5 中的问题

问题	描述
Registry-4104	当配置了 AMQ Streams 存储和 OAuth 时，Apicurio Registry 可能会因为缺少的 kafka-oauth-client 类而无法启动。

表 1.5. 解决了 Apicurio Registry 2.5.4 中的问题

问题	描述
registry-4019	即使计数器到达限制，一些健康检查也始终为 UP。
Registry-3956	即使在本地缓存中已存在 schema (SerDes)，也会调用 schema registry。
Registry-3725	资源所有者密码 grant - 基本身份验证 - java.lang.IllegalStateException: Client is closed.
Registry-3647	protobuf 内容规范过时的值被检测到。

1.7. APICURIO REGISTRY 解决了 CVE

在 Apicurio Registry 2.5 中解决了以下常见漏洞和风险(CVE)：

表 1.6. Apicurio Registry 2.5.11 中解决的 CVE

CVE	描述
CVE-2024-1023	Eclipse Vert.x 工具包中的漏洞会导致内存泄漏，因为使用 Netty FastThreadLocal 数据结构。
CVE-2024-1300	Eclipse Vert.x 工具包中的漏洞会导致使用 TLS 和 SNI 支持配置的 TCP 服务器中的内存泄漏。
CVE-2024-26308	在 Apache Commons Compress 中发现没有限制或节流的漏洞的资源分配。
CVE-2024-25710	Apache Common Compress 中发现了一个带有无法访问的退出状态(Infinite Loop)漏洞的循环。
CVE-2024-29025	io.netty:netty-codec-http 软件包中发现了一个安全漏洞。由于 HttpPostRequestDecoder 中数据积累，所以这个软件包的分配容易受到资源分配（不限制）或 Throttling 的影响。

表 1.7. Apicurio Registry 2.5.9 中解决的 CVE

CVE	描述
CVE-2024-20952 CVE-2024-20921 CVE-2024-20919 CVE-2024-20918	很难利用漏洞，攻击者可以通过多个协议访问网络，从而破坏了用于 JDK、Oracle GraalVM Enterprise Edition 的 Oracle Java SE、Oracle GraalVM Enterprise Edition 的 Oracle Java SE、Oracle GraalVM Enterprise Edition。
CVE-2024-20945	很难利用漏洞，在日志中对 Oracle Java SE、Oracle GraalVM for JDK、Oracle GraalVM Enterprise Edition 执行的低特权攻击者、Oracle GraalVM Enterprise Edition 对 JDK、Oracle GraalVM Enterprise Edition 的 Oracle Java SE、Oracle GraalVM Enterprise Edition 造成影响。
CVE-2024-20932	易利用的漏洞可让通过多个协议进行网络访问的未经身份验证的攻击者破坏了用于 JDK、Oracle GraalVM Enterprise Edition 的 Oracle Java SE、Oracle GraalVM Enterprise Edition。
CVE-2023-39615	在 Libxml2 中发现了一个安全漏洞，它在 /libxml2/SAXX2.c 的 xmlSAX2StartElement （） 函数中包含全局缓冲区溢出。
CVE-2023-38473	Avahi 中发现了一个漏洞。avahi_alternative_host_name （）函数中存在可访问断言。
CVE-2023-38472	Avahi 中发现了一个漏洞。avahi_rdata_parse （）函数中存在可访问断言。
CVE-2023-38471	Avahi 中发现了一个漏洞。dbus_set_host_name 函数中存在可访问断言。
CVE-2023-38470	Avahi 中发现了一个漏洞。avahi_escape_label （）函数中存在可访问断言。
CVE-2023-38469	Avahi 中发现了一个漏洞，它在 avahi_dns_packet_append_record 中存在一个可访问的断言。
CVE-2023-27043	通过 3.11.3 的 Python 的电子邮件模块错误地解析包含特殊字符的电子邮件地址。
CVE-2023-7104	SQLite3 中发现了一个漏洞。此问题会影响 make alltest Handler 组件中的 ext/session/sqlite3session.c 函数的 sessionReadRecord 功能。操作可能会导致基于堆的缓冲区溢出。
CVE-2023-5981	发现了一个漏洞，在 RSA-PSK ClientKeyExchange 中格式密码文本的响应时间与 ciphertexts 的响应时间不同，并带有正确的 PKCS#1 v1.5 padding。
CVE-2023-5678	OpenSSL 中发现了一个安全漏洞，这会导致生成或检查较长的 X9.42 DH 密钥或参数要比预期要慢得多。此问题可能会导致拒绝服务。
CVE-2023-5388	发现，在 NSS 中使用用于 RSA 加密的数字库会泄漏信息，无论 RSA 解密结果的高顺序是零。
CVE-2023-3817 CVE-2023-3446	OpenSSL 中发现了一个漏洞。发生此安全问题的原因是，使用 DH_check （）、DH_check_ex （）或 EVP_PKEY_param_check （）函数的应用程序来检查 DH 密钥或 DH 参数可能会长时间延迟。

CVE	描述
CVE-2022-48564	在 plistlib.py 文件中的 read_ints （）函数中的 Python core plistlib 库中发现了一个漏洞。
CVE-2022-48560	通过 heapq 模块中的 heappushpop 函数在 Python 中发现一个 use-after-free 漏洞。
CVE-2021-3468	在 avahi 中发现了一个安全漏洞。在 avahi Unix 套接字上指示客户端连接终止的事件没有在 client_work 功能中正确处理，允许本地攻击者触发无限循环。

表 1.8. 解决了 Apicurio Registry 2.5.4 中的 CVE 问题

问题	描述
IPT-1034	CVE-2023-5072 JSON-java: parser混淆会导致 OOM 错误。
IPT-1030	CVE-2023-31582 jose4j: Insecure iteration count 设置。
IPT-1021	CVE-2023-44487 undertow: HTTP/2: 启用多个 HTTP/2 的 Web 服务器会受到 DDoS 攻击(Rapid Reset Attack)的影响。
IPT-1013	CVE-2023-39410 avro: apache-avro: Apache Avro Java SDK: Memory when deserialing not data in Avro Java SDK.
IPT-995	CVE-2023-4853 quarkus-vertx-http: quarkus: HTTP 安全策略绕过。
IPT-993	CVE-2023-39321 CVE-2023-39322 integration-service-registry-operator-container: 各种漏洞。
IPT-953	CVE-2023-29409 integration-service-registry-operator-container: golang: crypto/tls: slow 验证包含大型 RSA 密钥的证书链。
IPT-948	CVE-2023-29406 integration-service-registry-operator-container: golang: net/http: insufficient sanitization of Host 标头。
IPT-940	CVE-2023-34462 netty: SniHandler 16MB 分配会导致 OutOfMemoryError。
IPT-936	CVE-2023-34455 snappy-java: 未检查的块长度会导致 DoS。
IPT-935	CVE-2023-35116 jackson-databind: 通过 cyclic 依赖项拒绝服务。
IPT-874	CVE-2023-1584 quarkus-oidc: ID，通过授权代码流访问令牌泄漏。

表 1.9. Apicurio Registry 2.5.4 中解决的额外 CVE

CVE	描述
CVE-2023-44483	所有版本的 Apache Santuario - 2.2.6、2.3.4 和 3.0.3 之前的 Java 的 XML 安全性都会受到使用 JSR 105 API 的问题的影响，当生成带有 debug 级别的 XML 签名和日志记录时，可以在日志文件中披露私钥。
CVE-2023-43642	在 snappy-java 中的 SnappyInputStream 中发现了一个安全漏洞，它是 Java 中的数据压缩库。当因为块长度缺少上限检查而解压缩带有太大的块数据时，会发生此问题。
CVE-2023-42503	Apache Commons Compress: Denial of service via CPU consumption for malformed TAR 文件。
CVE-2023-40217	Python 3 ssl.SSLSocket 容易受到对 HTTPS 服务器在特定实例中绕过 TLS 握手的攻击，以及使用 TLS 客户端身份验证的其他服务器端协议，如 mTLS。
CVE-2021-39194	在解析带有 kaml 中标记的 polymorphism 风格的 polymorphic 输入时拒绝服务
CVE-2023-34454 CVE-2023-34453	在 Snappy-java 的 shuffle 函数中发现了一个安全漏洞，它在开始操作前不会检查输入大小。
CVE-2023-29491	在 ncurses 中发现了一个漏洞，由 setuid 应用程序使用时进行。
CVE-2023-28118	在使用定位符和别名解析输入时，kaml 的潜在拒绝服务。
CVE-2022-24823	当在 netty 中使用多部分解码器时，如果启用了在磁盘上存储上传，则本地信息披露可能会通过本地系统临时目录发生。
CVE-2023-4911	在处理 GLIBC_TUNABLES 环境变量时，GNU C 库的动态加载程序 ld.so 中发现了一个缓冲区溢出。
CVE-2023-4813	glibc 中发现了一个安全漏洞。在不常用的情形中，gaih_inet 函数可以使用已释放的内存，从而导致应用程序崩溃。
CVE-2023-4806	glibc 中发现了一个安全漏洞。在非常罕见的情况下，getaddrinfo 函数可以访问已释放的内存，从而导致应用程序崩溃。
CVE-2023-4527	glibc 中发现了一个安全漏洞。当使用 AF_UNSPEC 地址系列调用 getaddrinfo 函数，且系统通过 /etc/resolv.conf 配置为 no-aaaa 模式时，通过 TCP 大于 2048 字节的 DNS 响应可以通过函数返回的地址数据披露堆栈内容，并可能导致崩溃。

1.8. APICURIO REGISTRY 已知问题

Apicurio Registry 2.5 中应用以下已知问题：

Apicurio Registry 核心已知问题

registry-3413 - 默认启用旧 REST API 日期格式

为获得最大兼容性，并从旧版本的 Apicurio Registry 更轻松地升级，Apicurio Registry REST API 中使用的日期格式与 OpenAPI 标准不兼容。这是因为旧版本中的一个错误。

在 Apicurio Registry 的下一个发行版本前，您必须升级所有客户端应用程序以使用最新的 Apicurio Registry 客户端版本。下一个版本将修复日期格式 bug，这会导致旧的客户端不再与 REST API 兼容。

要将 REST API 更新至兼容 OpenAPI，您可以修复此 Apicurio Registry 中的日期格式错误，如下所示：

1. *将所有客户端应用程序更新至 **2.5.11.Final-redhat-00001**，如 更新 2.x 客户端依赖项 中所述。*
2. *将以下环境变量设置为显示的值：*

```
REGISTRY_APIS_V2_DATE_FORMAT=yyyy-MM-dd'T'HH:mm:ss'Z'
```

IPT-814 - Apicurio Registry logout 功能与 RH-SSO 7.6 不兼容

*在 RH-SSO 7.6 中，与 logout 端点一起使用的 **redirect_uri** 参数已弃用。如需了解更多详细信息，请参阅 RH-SSO 7.6 升级指南。因此，当 Apicurio Registry 使用 RH-SSO Operator 保护时，点 **Logout** 按钮会显示 **Invalid parameter: redirect_uri** 错误。*

有关临时解决方案，请参阅 https://access.redhat.com/solutions/6980926。

IPT-701 - CVE-2022-23221 H2 允许通过 JNDI 从远程服务器加载自定义类

当 Apicurio Registry 数据存储在 AMQ Streams 中时，H2 数据库控制台允许远程攻击者使用 JDBC URL 执行任意代码。在默认情况下，Apicurio Registry 不会受到攻击，需要进行恶意配置更改。

Apicurio Registry Operator 已知问题

operator-42 - 自动生成 OpenShift 路由可能会使用错误的基本主机值

*如果指定了多个 **routerCanonicalHostname** 值，则 Apicurio Registry OpenShift 路由可能会使用错误的基础主机值。*

附录 A. 使用您的订阅

Apicurio Registry 通过软件订阅提供。要管理您的订阅，请访问红帽客户门户中的帐户。

访问您的帐户

1. 转至 access.redhat.com。
2. 如果您还没有帐户，请创建一个帐户。
3. 登录到您的帐户。

激活订阅

1. 转至 access.redhat.com。
2. 导航到 **My Subscriptions**。
3. 导航到 **激活订阅** 并输入您的 16 位激活号。

下载 ZIP 和 TAR 文件

要访问 ZIP 或 TAR 文件，请使用客户门户网站查找下载的相关文件。如果您使用 RPM 软件包，则不需要这一步。

1. 打开浏览器并登录红帽客户门户网站 **产品下载页面**，网址为 access.redhat.com/downloads。
2. 在 Integration **和 Automation 类别中找到Red Hat Integration**条目。
3. 选择所需的 Apicurio Registry 产品。此时会打开 **Software Downloads** 页面。
4. 单击组件的 **Download** 链接。

更新于 2024-05-30