



# API 攻击剖析

了解 BOLA 和清单管理漏洞



## 简介

---

现在,大多数安全团队已经明白,一款有效的企业级安全程序必须具有主动威胁搜寻能力,尤其是对于应用程序编程接口 (API) 安全而言更是如此。API 通常支持对数据、功能和工作流的直接访问。尽管企业为保护应用程序而广泛使用了基础边界安全措施,但 API 滥用和其他类型的攻击却愈演愈烈。事实上,近年来登上新闻头条的一些热门安全事件都与 API 相关。为了更好地了解这些攻击模式,例如失效的对象级授权 (BOLA) 和清单管理不当的漏洞,本白皮书将:

- 回顾 API 基础知识
- 探讨为什么 API 安全越来越重要
- 借用一些高关注度 API 安全事件来重点强调几个关键的 API 安全领域
- 阐述有效执行 API 威胁搜寻所需具备的能力

## API 和端点基础知识

---

首先,我们来回顾一些基本术语。API 用途广泛,包括用于实现企业对消费者 (B2C) 功能、企业对企业 (B2B) 协作和集成,以及内部开发和集成功能等。Web API 是最常见的实施模型,通过 Web 浏览器使用的相同 HTTP 协议进行通信。这些 API 所提供的特定功能有时也称为服务或 API 产品。

要想确保 API 安全,了解端点的概念也至关重要。虽然端点有时用于指代最终用户计算设备,但 API 端点却有着其他含义。您可以将 API 端点视为组成 API 的单个可访问资源以及对资源执行的操作。

举个简单的例子。一个返回特定公司订单信息的 API 端点可以表示为: `GET /orders/{orderId}`。其中, `GET` 是具体的 HTTP 方法,而 `orders` 和 `orderId` 代表通过 API 请求的具体资源。

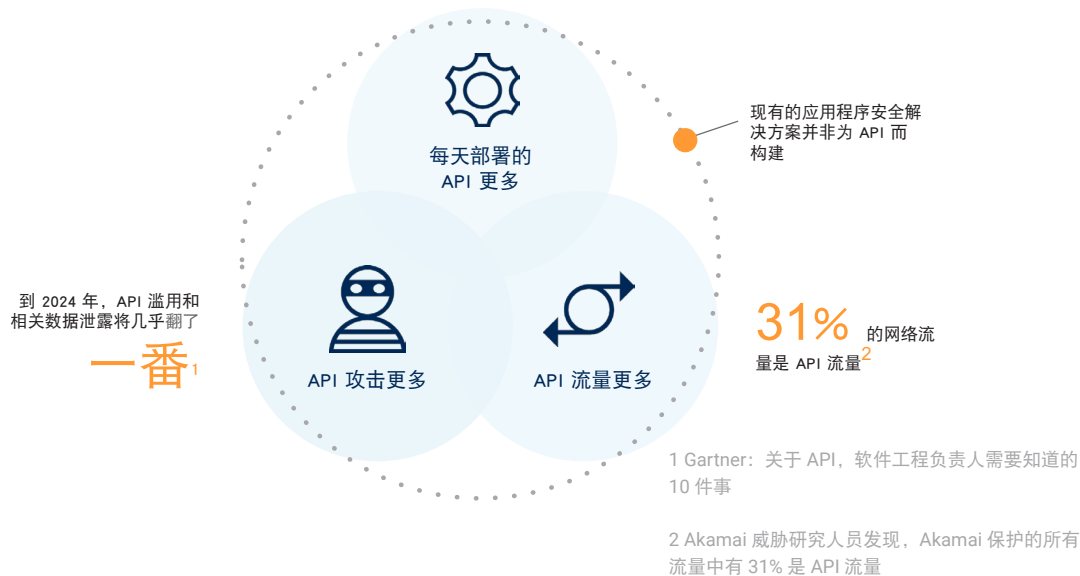
## 为什么说 API 是下一个重大安全挑战？

过去，攻击者可能专注于入侵企业数据中心，进而从特定的服务器访问和窃取企业数据。他们也可能尝试检查企业网络流量，伺机捕捉敏感数据。在这些场景中，主动威胁搜寻能力可能会重点搜寻渗透测试等活动，以切断攻击者可能的侵入点。

在当今 API 广泛应用的世界里，情况已经发生变化。许多 API 本身可供外部的任何人访问，有时只有凭据和密钥这一道防线。而攻击者越来越擅长窃取凭据和密钥。此外，获得 API 访问权限的用户如果以未经批准的方式使用其权限，也会造成 API 滥用，有些最具破坏性的 API 滥用类型便是由此产生。

## 现实世界中的 API 攻击

在 Akamai 保护的所有流量中，有 31% 是 API 流量。API 流量的增加会导致下游效应，例如攻击和滥用的增加。Gartner 预计 2024 年 API 滥用和数据泄露事件将增加一倍。与此同时，许多安全团队仍停留在追赶模式。API 数量成倍增加，而现有的应用程序安全工具提供的 API 保护非常有限。



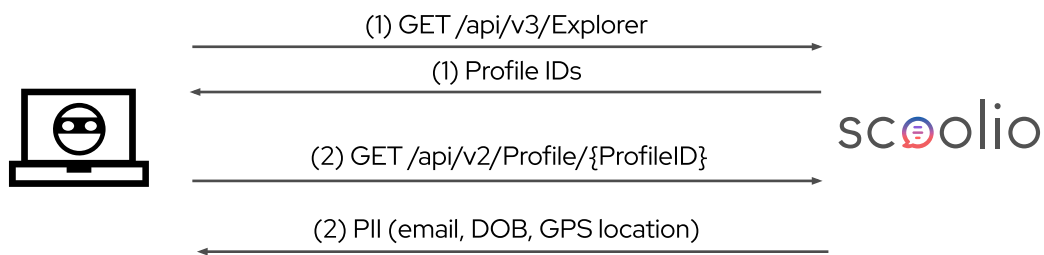
为了在真实场景中解释这个问题，我们通过一个案例研究来说明 API 攻击可能对企业及其客户产生的现实影响。

## 案例研究

### 帐户接管 | Scoolio

一个备受瞩目的例子是 2021 年影响德国教育应用程序 Scoolio 的事件。该应用程序会从学生用户收集大量的信息。例如进行性格测试, 提供社交网络和聊天功能, 以及管理学习计划和辅导等活动。这些功能收集了大量的个人身份信息 (PII)。安全研究人员 Lilith Wittmann 在该应用程序的 API 中发现一个 BOLA 漏洞, 攻击者可以利用两个 API 调用来访问任何使用该应用程序的其他用户的 PII 和其他数据。

操作方式如下:



#### 第 1 步

发送 GET /api/v3/Explorer API 调用。

该调用返回 UUID, 在此实例中称为 ProfileID。

#### 第 2 步

发送 GET /api/v2/Profile/{ProfileID} API 调用。

该请求返回相关用户的大量 PII, 包括电子邮件、出生日期、GPS 位置等。

#### 使用 UUID 的价值

这两种场景都重点使用了 UUID, 事实上, 使用 UUID 确实是一种很好的做法。使用随机生成的数字而非可预测的用户标识符序列会让攻击者更难获取全体用户的信息。问题在于 UUID 信息被随意暴露并且存在 BOLA 漏洞。

## 清单管理不当

该 API 漏洞攻击的另一个方面是利用清单管理不当, 后者在 OWASP 十大 API 安全风险清单中排第 9 位。仔细观察其攻击序列, 您就会注意到第一步适用于程序的 API 版本 3, 第二步则是针对版本 2。版本 3 中进行了改进, 提供了更严格的 PII 访问控制措施。然而, 更容易受到攻击的版本 2 仍然可供所有人访问, 因而削弱了这些

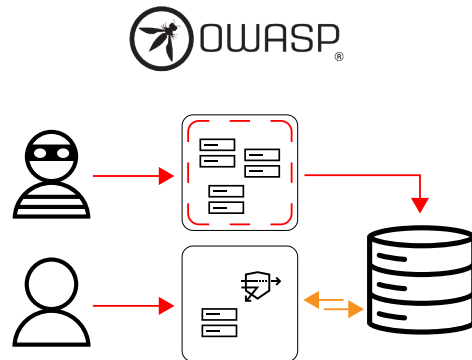
改进的效果。最终, 版本 2 和版本 3 都受到了 BOLA 漏洞的影响。但没有必要继续存在的版本 2 使得该漏洞的影响更加严重。

## 现在企业采取哪些措施来保护 API?

许多企业侧重于通过以下三个关键做法来确保 API 安全:

1. 集中授权——首先, 为所有 API 访问端口实施集中授权引擎可防止开发过程中出现错误而造成授权机制有缺陷, 从而可以降低 API 漏洞风险。
2. API 测试——第二个重要的做法是 API 测试。使用静态代码分析和动态测试方法来测试所有漏洞 (特别是失效的授权) 可以在开发过程中提早发现问题。
3. 运行时保护——第三个重要的关键做法是对生产环境应用一系列运行时保护措施。即使最主动的团队也无法在部署之前发现每个漏洞。因此, 检查用户对生产数据的访问权限并且尽可能防止对已知类别漏洞的利用至关重要。

这三个做法为 API 安全策略奠定了良好的基础。但还有一点很重要, 那就是要记住这些做法并非毫无破绽, 也并非面面俱到。例如, 即使企业采用了集中授权模式, 也不能保证开发人员始终遵循最佳做法。最后, 现有应用程序保护工具在检测已知攻击模式时通常表现良好, 但对于检测 BOLA 等更隐蔽的威胁则不尽如人意。



## 如何在这三个做法的基础上采用更先进的 BOLA 检测技术?

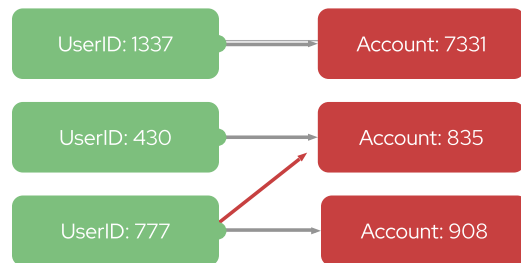
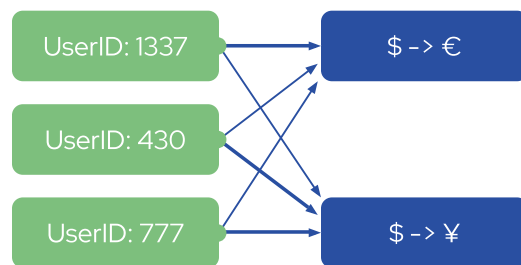
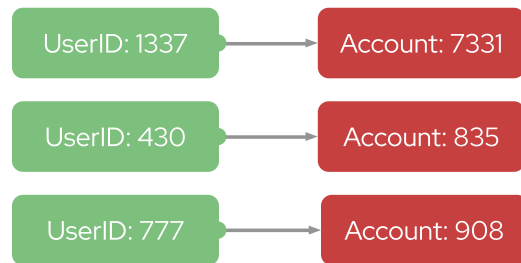
要想检测和应对 BOLA 及其他隐蔽的 API 漏洞,一个关键做法是对 API 活动涉及的实体之间的关系进行建模。这些实体除了资源本身之外,还包括尝试访问资源的参与者(例如用户)。如果可以对参与者实体和与 API 交互的业务流程实体之间的这些连接进行映射,就可以在分析原本看似相同的 API 事件时区分合法活动与非法活动。

### 关系映射图解说明

为了更好地理解关系映射,可以参考这里的简单示例。某银行应用程序支持两种操作。第一种操作是读取帐户数据,包括帐户余额、近期交易等信息。第二种操作是查看货币汇率。这两种操作中用户与资源之间的关系大不相同。对帐户信息的访问应仅限于单个用户。与之不同,汇率功能通常应该对所有用户开放。

尽管这是一个非常简单的示例,但为实体之间的关系映射构建更复杂的模型可以让预防或检测 BOLA 更加切实可行。

在此例中,我们看到一个用户试图访问并非其自己的帐户。具体的 API 调用可能是相同的,但实体映射所提供的附加背景信息却清楚表明不应该允许访问。



## 实践中的高级 BOLA 攻击检测

接下来，我们将此概念运用到更复杂的示例，例如案例研究中的漏洞。以下是该案例场景涉及的实体片段：

scoolio

GET/api/v3/Profile/{ProfileID}

标头：

- Authorization: <MyAccessToken>

参与者实体以绿色突出显示，请求的资源（个人资料 ID）以红色突出显示。理解了这些关系，就可以采取措施来执行通用逻辑，例如在适当的情况下限制参与者对单个资源的访问。这绝不是轻而易举的事，因为实际的关系可能比这更复杂并且包含一对多的层面。但机器学习和行为分析等技术使其成为可能。例如，假设我们为某客户成功检测到 BOLA 漏洞，结果可能如下所示：

The screenshot displays a security dashboard with the following components:




- Alert Summary:** "Suspicious Data Access" (Open), 21 SEP 2022 | 18:24:50.00 | Data Leak, OWASP A5, OWASP API1, OWASP API5. Status: Open. Category: Account Takeover. Severity: Medium.
- Description:**
  - Endpoint "PUT/users/v1/username/password" in service "users"
  - A User should not access more than one username
  - The User "MyDemoUser" accessed more than one username: "MyDemoUser", "admin"
- Timeline:** Shows a sequence of events on 21 September:
  - 18:24:17: PUT vampi-nginx-neosec-dev-internal.com/users/v1/MyDemoUser/password
  - 18:24:24: PUT vampi-nginx-neosec-dev-internal.com/users/v1/admin/password
  - 18:24:50: Suspicious Data Access (Alert)
- Log Table:** Shows 2 rows of log entries:

TL	ENTITY TYPE	ENTITY ID	ENDPOINT	S.	S.	LABELS	CONTENT
21 Sep 2022 18:24:24	User	MyDemoUser	PUT vampi...	204	10.3...		→application/json(27) ←application/json(0)
21 Sep 2022 18:24:17	User	MyDemoUser	PUT vampi...	204	10.3...		→application/json(27) ←application/json(0)

对于本示例，我们在实验室环境中模拟了 BOLA 漏洞。通过实体映射和行为分析，我们的平台检测到 BOLA 并生成了信息丰富的告警。查看告警的安全分析师或威胁搜寻人员将看到 MyDemoUser 访问了自己的用户个人资料并更改了密码，这是经过批准的操作。然而，随着时间轴的推移，我们可以看到该用户不久就执行了另一个 API 调用来更改管理员密码。根据参与者和资源之间的关系，这显然是未经批准的操作，因此平台生成了告警。

## 从哪里开始实施 API 安全计划

对于大多数企业来说，API 安全是一项持续进行中的工作。所以可能很难知道应该从哪里开始入手。上述三个基础的关键做法很适合作为起点，而如果在实施安全计划时遵循以下三点建议，将会显著提高工作成效：

-  1. 确保 API 清单始终处于最新状态
-  2. 监控非生产性和生产性 API 环境
-  3. 配置实体之间的关系

我们无法保护未知的 API。因此，有效的 API 保护始于随时更新的 API 清单和安全态势评估。同样，在开发 API 安全监控功能时，将功能扩展到生产性和非生产性 API 的实施也很重要。最重要的是，API 监控和执行不能仅限于单纯的操作，还要考虑 API 活动中涉及的实体之间的关系。这有助于发现漏洞和防护缺口，并确保符合预期的 API 使用模型。了解 API 中的行为将有助于发现任何滥用迹象。

想要了解关于 API 攻击的更多信息以及如何防范这些攻击吗？敬请查看我们的 **OWASP 十大 API 风险明细表**。



无论您在何处构建内容，以及将它们分发到何处，Akamai 都能在您创建的一切内容和体验中融入安全屏障，从而保护您的客户体验、员工、系统和数据。我们的平台能够监测全球威胁，这使得我们可以灵活调整和增强您的安全格局，让您可以实现 Zero Trust、阻止勒索软件、保护应用程序和 API 或抵御 DDoS 攻击，进而信心十足地持续创新、发展和转型。如需详细了解 Akamai 的云计算、安全和内容交付解决方案，请访问 [akamai.com](https://akamai.com) 和 [akamai.com/blog](https://akamai.com/blog)，或者扫描下方二维码，关注我们的微信公众号。



扫码关注 · 获取最新 CDN 前沿资讯