白皮书

如何防范 API 漏洞

探索 5 类 API 漏洞及其防范方式

报告内容

前言

API 可以帮助企业与合作伙伴、供应商及客户交换数据，从而连通企业的各个环节。但在大多数企业中，API 安全防护仍然不够全面。事实上，对于众多公司来说，近年来易受攻击的 API 已然成为众矢之的，使得攻击者不断滥用它们来获取敏感数据，然后将这些数据出售给其他攻击者或者公之于众。2024 年，消费者电信、企业计算和虚拟协作领域的全球众多品牌都遭遇了 API 攻击事件，大量客户数据及其他敏感数据遭到泄露，造成巨大经济和声誉损失。

什么是 API 漏洞?

简而言之，只要出现任何故意不当使用或滥用 API 的行为就表明存在 API 漏洞，而此类行为的目的通常是获取敏感数据。API 漏洞的类型可以根据不同的标准进行细分。为了识别风险并避免生产运营中出现漏洞，可以考虑以下将风险分为五种类别的方案：

1. **已知漏洞**
   - 攻击者会利用尚未修补的已知漏洞。
2. **影子 API、恶意 API、僵尸 API 和已弃用的 API**
   - 不受管理以及被遗忘的 API 会导致运营容易受到攻击。
3. **外部暴露**
   - 凭据、密钥和其他暴露可能超出了您的控制范围。
4. **错误配置和操作员错误**
   - 基础架构和服务中的安全系统配置不当可能会为攻击者创造利用漏洞的入口点。
5. **未发现的漏洞和错误**
   - 即便您已经竭尽全力来建立安全屏障，但攻击者仍会想方设法从生产环境中找出任何可能的错误和漏洞。

本电子书将说明这五类 API 漏洞中发生安全故障的位置以及如何进行防范。另外还将帮助您重点关注 API 安全计划中的特定薄弱环节，以最大限度地提高 API 安全性并降低风险。

漏洞类型：已知漏洞

利用尚未修补的已知漏洞发起的 API 攻击屡见不鲜。如果犯罪分子想要获取您的数据，那么对他们来说，第一步往往是检查您的企业是否留有任何后门。

2024 年 1 月，某个攻击者通过利用缺少身份验证控制措施的 API 端点，成功入侵一款使用广泛的项目管理工具。在入侵此 API 后，该攻击者未经授权便访问了数百万用户的信息，并且几个月后在互联网上泄露了超过 21 GB 的数据，包括电子邮件地址和董事会成员信息。

身份验证和授权问题是最常见的 API 问题之一。OWASP 十大 API 安全风险提供了有关各企业必须防范的 10 个最严重 API 漏洞（包括失效身份验证）的信息。

除了保护 API 免受 OWASP 十大风险中各类风险的侵扰之外，各企业还应该保护 API 代码，使其免受常见漏洞和风险 (CVE) 完整列表中相关漏洞的影响，该列表由 MITRE 运营的美国国家网络安全联邦资助研究与发展中心 (FFRDC) 编制。您可能还记得广为人知的 Apache Log4j 2 漏洞 (CVE-2021-44228)，它也称为“Log4Shell”。Log4j 库是一个用于 Java 编程语言的热门开源日志记录库，由于该库中存在的一个缺陷，攻击者能够远程执行任意代码来获取系统访问权限。他们会定期探查企业系统中是否存在与之类似的已知漏洞。

在美国，网络安全和基础架构安全局 (CISA) 负责维护已知 CVE 的目录。其他国家或地区也可能维护着类似的目录。

OWASP 十大 API 安全风险列表创建于 2019 年，并于 2023 年进行了更新。尽管它很有用，却跟不上攻击面的变化速度。仅在 2024 年，CISA 的目录中就新增了 24,000 多个新的 CVE，其中超过 500 个 CVE 与 API 相关（截至 2024 年 8 月中旬）。

若想全面保护贵企业免受已知漏洞的影响，需要双管齐下：

1. 确保您的开发和测试过程足够可靠，能够避免将已知漏洞带入生产环境中。

2. 在发现新的漏洞后，尽快修补这些漏洞。

很多企业都难以完成这两个步骤。除此之外，他们还使用来自第三方来源的 API 和代码，这可能会带来一些单独的漏洞。2022 年，一个研究团队发现了一些严重 API 缺陷，这些缺陷影响了整个汽车行业中多家制造商。这些缺陷会暴露敏感的客户数据甚至车辆的位置，使攻击者能够通过被入侵的远程管理系统来解锁、启动汽车或使其无法运转。

如何防范

贵企业要抵御已知漏洞可能造成的 API 攻击，一个众所周知的方法就是在安全补丁发布后迅速更新软件和系统。另外，还必须确保您的开发和测试过程综合全面，并严格遵循 API 安全防护最佳实践。其中包括：

- **保护您的软件供应链：**确保您使用的所有库、开源软件 (OSS) 和其他第三方代码都是安全的。
- **实施左移安全测试：**将与 API 安全和软件测试相关的任务移至开发过程早期阶段。这可以帮助您发现一些漏洞，例如开发人员团队在需要快速发布软件或更新的压力下出现的代码编写错误和配置错误。
- **利用 API 安全态势管理：**此功能将 API 发现与敏感数据识别及漏洞检测功能相结合，可确保修复工作首先集中在最关键的 API 上。

Akamai API Security 如何为您提供帮助

借助 Akamai API Security，您的团队可以减少每个新的构建版本中的已知漏洞，同时无需牺牲速度。API Security 是一个专门构建的 API 安全测试解决方案，可全面覆盖 API 相关的漏洞。主动测试可帮助将 API 安全测试融入开发的每个阶段。

- 根据对应用程序业务逻辑的了解，**寻找并测试每个 API**。
- 通过集成到整个软件开发生命周期中来实现**左移**。在整个 CI/CD 流程中，团队可以跨多个状态和环境中获得动态的 API 监测能力。
- **为开发人员提供出色的可用性**，包括简单的设置和自动化、联合测试结果以及用于修复已识别问题的情境指导。

此外，API Security 的态势管理让您可以全面了解流量、代码和配置，以评估您的 API 安全态势。API Security 会尽可能查看更广泛的来源以检测漏洞，包括日志文件、历史流量回放、配置文件等。它还会检测是否存在 OWASP 十大 API 安全风险中的漏洞（要详细了解态势管理，请参阅“错误配置和操作员错误”一节）。

漏洞类型：影子 API、恶意 API、僵尸 API 和已弃用的 API

您无法保护看不见的资产，而在很多公司内，很大一部分 API 都不受管理，这使得影子 API、恶意 API、僵尸 API 和已弃用的 API（请参阅下一页的侧栏）成为您的 API 资产中未被察觉或未加考虑的目标。此外，攻击者往往会查看某个企业已暴露的 API，然后进行模糊测试或修改值以查找旧版本，从而搜寻可利用的 API 变体。

这正是澳大利亚一家大型电信公司的遭遇，他们意外暴露了超过 1120 万条客户记录，包括姓名、地址、出生日期和政府签发的一些身份证件号码，因此遭受了攻击。此次攻击利用了一个用于测试的 API，而该 API 出于某种未知原因可通过开放的互联网进行访问。由于此异常 API 缺少身份验证检查，攻击者乘机请求并接收数百万条记录。

大多数企业在运营中会用到各种遗留 API 和新 API。不幸的是，很多人都会发现自己身边存在着各种恶意 API、僵尸 API 和影子 API，而这些 API 导致企业面临一系列网络安全风险和运营难题。

这些被忽视的 API 有各种来源：

- **商业 API：**一些商业软件包会包含用于连接其他应用程序及外部数据源的 API。这些 API 可能会被激活却无人觉察（这是可以通过全面 API 发现解决的问题）。
- **旧 API 版本：**在很多情况下，某个 API 的旧版本可能安全防护能力较差或者存在已知漏洞，但从未被删除。在软件更新期间，旧版本可能需要与新版本共存一段时间，但当进程故障导致无法关闭旧 API 时，该 API 便会变为僵尸 API。
- **走捷径和进程故障：**那些虽被创建但未告知 IT 或安全团队的 API 就会形成影子 API。例如，某个业务线团队可能会创建用于满足特定需求的 API 而未告知 IT 或安全团队，或者某个开发人员可能并未遵循相关过程。
- **继承的 API：**因合并或收购而继承的 API 也经常会被忽略而成为影子 API。
- **重新激活的代码：**在某些情况下，API 的旧版本可能会意外被重新激活。

如何防范

若想通过手动 API 审核来记录必须准确清点的所有输入，可能需要花费几个小时的时间，考虑到还要评估和处理发现的每个 API，所需时间会更多。这对本已负担过重的安全团队来说无疑是不现实的。为了保护您的企业免受恶意 API、僵尸 API 和影子 API 被利用的影响，您需要能够识别正在使用的各类 API 的自动化 API 发现功能。然后通过它来找到并清点您的整个运营过程中的每个 API，并且发现不受 API 网关管理的 API 和 API 域，这对您至关重要。

Akamai API Security 如何为您提供帮助

API Security 会利用广泛的集成来源来提取 API 数据，例如原始流量、日志记录等。利用从这些来源中得到的数据，API Security 可以识别 API 及其错误配置和漏洞，以及 API 滥用。我们的发现工具可以检测是否存在 OWASP 十大 API 安全风险中的漏洞。

借助附加的发现功能，您可以：

- 找到并清点所有 API，无论配置或类型如何（包括 RESTful、GraphQL、SOAP、XML-RPC、JSON-RPC 和 gRPC）
- 发现休眠、遗留和僵尸 API
- 识别被遗忘、被忽视或未知的影子域名
- 维护 API 清单并确保 API 文档记录准确无误

攻击者寻找的不受管理的高风险 API

影子 API（也称为“未明确记录的 API”）存在并运行于企业官方监控的渠道之外。它们可能是善意的开发人员为了加快工作速度而创建的，也可能是先前的软件版本的残余部分。

恶意 API 是未经授权或恶意的 API，会对系统或网络构成安全风险。

僵尸 API 包含被新版本或其他 API 完全取代后仍处于运行状态的任何 API。

已弃用的 API 是由于 API 发生了变化而不再推荐使用的 API。虽然已弃用的类、方法和字段仍处于实现状态，但它们可能会在未来的实现中被删除，因此您不应该在新代码中使用它们。

漏洞类型：外部暴露

外部的 API 漏洞通常由不良实践或程序错误所导致，例如 API 密钥和凭据泄露、API 代码和架构暴露、文档管理松散和代码库漏洞。因此，使用合适的功能来发现运营边界外的潜在攻击媒介已成为当务之急。去年，多起备受瞩目的数据泄露事件就是外部来源的 API 密钥或其他凭据遭到意外暴露所致。例如，黑客使用网络钓鱼活动对 Dropbox 的 130 个源代码库进行了未经授权的访问。这让他们可以获取 GitHub 上未被妥善保存的 API 密钥。此类暴露已变得非常普遍，以至于 GitHub 不得不采取措施来阻止发生 API 密钥和其他密文泄露，但其他公共代码库可能仍然容易受到攻击。

在另一个广为人知的外部暴露示例中，研究人员发现超过 3,000 款移动应用程序将 Twitter API 密钥公之于众。出乎意料的是，此类错误很常见，因为在开发过程中，开发人员往往会为了方便而在应用程序代码中嵌入 API 密钥。如果他们未能在公开发布前移除这些嵌入的密钥，这便会成为密钥暴露的潜在来源。

如何防范

减少或消除这些类型的外部暴露需要采用双管齐下的方法：

- 加强对相关过程的管理，以识别和消除暴露来源，例如泄露的密钥和凭据、代码库的不当使用等。
- 定期扫描外部攻击面以检测和修复漏洞。

要保护您自己免受广泛 API 威胁的侵扰，您同时需要由内而外的发现（如“恶意 API 造成的漏洞”一节所述）以及由外而内的发现，这可以识别暴露问题并减小您的外部攻击面。

Akamai API Security 如何为您提供帮助

API Security 可模拟黑客使用的侦察技术并让您能够快速找到并修复问题，从而帮助您抢先一步防范攻击者的攻击。借助由外而内的发现，API Security 会定期自动扫描您的外部攻击面，以便在攻击者之前找到漏洞，从而让您可以：

- **找到公开的漏洞：**快速找到并修复关键问题，如 API 密钥和凭据泄露、代码暴露、错误配置、代码库漏洞等。
- **发现与贵公司相关的域名和子域名：**利用从各种来源（包括互联网注册商、证书注册商以及开放来源）收集的数据。
- **融入真实的攻击方法：**模仿攻击者进行外部侦察，通过执行对公司域名或子域名的有限查询来收集相关信息。

漏洞类型：错误配置和操作员错误

很多网络攻击者会利用服务器、网络、API 网关以及防火墙（用于代理和保护 API 流量）的错误配置来实施入侵。IBM Security X-Force 进行的一项研究发现，三分之二的云漏洞都与错误配置的 API 相关。导致安全系统错误配置的原因可能包括不安全的默认配置、没有访问控制的云存储（出人意料地常见）以及不完整或临时的配置。随着您的数字足迹不断扩大，您的运营可能会扩展到更多位置，包括多个公有云可用区域或 AWS、Microsoft Azure 和 Google Cloud 等公有云。这些环境往往在不同的安全控制措施下运行，这使得确保在每个地方都进行正确的安全配置变得复杂且困难。

如何防范

要想从基础架构层面有效防范安全错误配置，您应该尽可能地避免对服务器、网络设备、网关和防火墙进行手动配置。如果贵公司的管理员团队定期手动配置基础架构和应用程序安全控制措施，也就是定期“调整”它们，则引入配置漏洞的几率会增加。

在安全方面，自动化是您最好的朋友。为避免人为错误，部分公司开始接受不可变基础架构的理念。

即使您已尽己所能，采取了一切措施来确保基础架构、服务和 API 万无一失，您仍然需要 API 态势管理。态势管理为您提供了用于在 API 整个生命周期中管理、监控和保持 API 安全性的工具。

API Security 如何为您提供帮助

API Security 的态势管理模块可以分析 API 调用和基础架构，以识别是否存在错误配置。这些错误配置通常是 Amazon S3 存储桶问题、与未经身份验证的 API 相关的敏感数据以及不同的基于 Kubernetes 访问权限的错误配置。

态势管理模块可让您全面了解流量、代码和配置，并让您可以了解跨 API 和 Web 应用程序的整个攻击面，包括通过 API 移动的所有形式的敏感数据，例如个人身份信息。它还可以帮助您确认 API 管理工具是否正在使用安全系数高的协议和密码，从而避免使用可能会暴露这些敏感数据的弱加密。此外，API 不得接受过期的 JSON Web 令牌，因为这样做会允许未经授权的访问并增加安全风险。此模块还可以帮助避免出现错误配置，例如应用程序负载均衡器在没有重定向的情况下侦听不安全的端口。所有这些措施可以共同增强 API 的安全态势，从而确保对潜在威胁的防御措施具备更强的恢复能力。

漏洞类型：未发现的漏洞

和大多数漏洞类型一样，网络犯罪分子会定期扫描您的基础架构以寻找 CVE、OWASP 十大 API 安全风险和其他常见错误配置，以及恶意 API、僵尸 API 和影子 API。他们还会探查您的已暴露的 API，查找可在库、开源代码和其他类型的公共代码中利用的新漏洞，以及您的 API 资产中是否存在可以利用的代码编写错误、缺陷及错误配置。这些漏洞让网络犯罪分子能够操纵 API 调用并将模糊测试字符串插入请求中。因此，网络犯罪分子使用的技术也在不断地演进变化。

如何防范

要防范此类漏洞，务必尽可能地确保您的代码没有错误和漏洞（请参阅“已知漏洞”一节）。但是，您仍然应该假设攻击者会发现缺陷或者获得让他们能够利用 API 的密钥或凭据。

API 运行时保护旨在识别利用任何已知或未知漏洞的黑客。只有这样，才能保护您的 API 资产，使其免受先前未识别而进入生产环境中的缺陷和错误配置的影响，并且这也是防范已遭到泄露的凭据和密钥的最佳保护措施。

运行时保护可以识别出 API 使用和数据访问中的异常模式和异常，从而可以在数千或数百万条数据记录被提取之前，发现并修复可能未被察觉且正在进行的攻击。

API 运行时保护可帮助您识别并拦截恶意 API 请求，包括：

- 从 API 中提取大量敏感数据的攻击
- 失效的对象级授权 (BOLA) 攻击

API 运行时保护解决方案可以检测：

- 数据泄漏
- 数据政策违规
- API 安全攻击
- 数据篡改
- 可疑行为

此外，运行时保护还会记录 API 流量、监控敏感数据访问、检测威胁以及屏蔽或修复攻击媒介。

API Security 如何为您提供帮助

您可以将运行时保护作为最后一道防线，以便弥补其他防范措施的缺失。运行时保护的主要作用是实时检测并阻止 API 攻击。基于自主机器学习 (ML) 的监控功能用于执行实时流量分析，并提供对数据泄露、数据篡改、数据政策违规、可疑行为和 API 安全攻击的情境洞察。API Security 可检测出您的 API 流量中存在的异常和潜在威胁，并根据预先选择的事件响应策略来帮助进行修复。

通过使用 ML，API Security 可以为每个 API 构建一个行为模型。然后，可以将此正常行为基准用于检测 API 业务逻辑攻击。运行时保护生成的每个问题都包含严重性、状态、与 OWASP 十大 API 安全风险的对应关系以及攻击者详细信息（如果适用）。这些问题还包含攻击者的会话详细信息以及 API 请求和响应的副本等证据，以帮助对问题进行分类和修复。

除了很多用于简化操作和修复的常用工作流集成之外，API Security 运行时保护还可以实时检测和防范 API 攻击并且持续检测是否存在 API 错误配置。

对您的团队来说，可能最好的消息是 API Security 可以与 WAF、API 网关、ITSM、SIEM 及其他工作流工具相集成，以提供全面的攻击防御措施。您可选择以全自动的方式完成威胁修复，也可以要求进行不同程度的人工干预来实现更强的监测能力和控制。

5 种漏洞类型，5 项防范原则

您已经详细了解了网络犯罪分子如何使用 API，接下来可以集中精力进行防范。在遇到相关漏洞时，您可以将以下五种防范工具及策略观点结合起来运用：

1. 将 API 安全防护左移

- 将 API 安全防护左移意味着需要在开发中进行广泛测试，这样就不会在生产环境中暴露漏洞，让进行探查的网络犯罪分子失去可乘之机

2. 由内而外的发现

- 识别整个运营环境中的所有 API

3. 由外而内的发现

- 识别并消除暴露来源，例如泄露的密钥和凭据以及代码库的不当使用，并且定期扫描外部攻击面以检测和修复漏洞

4. 全面的态势管理

- 通过避免出现错误配置和漏洞，始终在 API 安全方面保持最佳状态

5. 运行时保护

- 检测异常的 API 活动并防范所有可能的威胁，包括先前未识别的漏洞和缺陷

申请演示

观看 Akamai API Security 的实际应用，体验如何轻松识别并修复 API 中的错误配置以及如何保护自己免受恶意 API 攻击。亲自了解为何出色的企业选择我们的 API 安全解决方案。

申请演示

Akamai

扫码关注 - 获取最新云计算、云安全与 CDN 前沿资讯