威胁猎人 THREAT HUNTER | CAICT 中国信通院

API安全发展白皮书

（2023）

深圳永安在线科技有限公司

中国信息通信研究院云计算与大数据研究所

版 权 声 明

参 编 人 员

毕裕、黄巍、吴越林、李忆晨、卫斌、郭雪、孔松

目 录

图 目 录

表 目 录

一、 API 安全概述

（一）API 已成为数字化转型的重要抓手，安全监管日趋严格

API 指应用程序接口（Application Programming Interface）。中华人民共和国国家标准《信息安全技术术语》（GB/T25069-2022）将其定义为“一组预先定义好的功能，开发者可通过该功能（或功能的组合）便捷地访问相关服务，而无需关注服务的设计与实现”。应用程序接口作为促进数据流通的重要技术，不仅可以帮助企业建立与客户沟通的渠道，还承担着不同复杂系统环境、组织机构之间的数据交互、传输的任务。API 安全指对应用程序接口的完整性进行有效的防护。API 安全通常包括接口的信息安全、数据安全以及应用安全。

API 成为企业数字化转型的重要抓手。自新冠疫情以来，各企业及机构纷纷加快数字化转型的步伐。国家互联网信息办公室发布的《数字中国发展报告（2022 年）》显示，2022 年我国数字经济规模达 50.2 万亿元，数字经济成为稳增长促转型的重要引擎。随着企业将越来越多的数据和应用迁移至云端，API 也正成为企业成功数字化转型的战略重点之一。IDC 报告显示，到 2021 年，超过 50%的全球 2000 强企业，将用 API 开放生态系统完成其平均 1/3 的数字化服务交互。开发、管理和保护 API 安全，将成为数字化时代下企业及机构需要重点考量的关键因素。

云计算引爆 API 安全危机。随着云计算、大数据、人工智能的蓬

勃发展，越来越多的应用开发深度依赖于 API 之间的相互调用。与此同时，随着全球云上业务快速发展，API 作为系统间的通信桥梁，也正成为攻击者重点光顾的目标，其安全漏洞随着调用量增多而暴露无遗。Gartner 在《如何建立有效的 API 安全策略》报告中预测，API 滥用将成为导致企业 Web 应用程序数据泄露的最常见攻击媒介，到 2024 年，API 滥用和相关数据泄露将几乎翻倍。通过攻击 API 来破坏信息系统和窃取数据，已成为数字时代黑产活动最集中的方向之一。

我国 API 相关监管逐渐清晰化，但针对不同领域的 API 安全攻防体系难以进行规模化落地。2021 年 9 月 1 日，《中华人民共和国数据安全法》正式实施，为开展数据安全监管和保护工作提供了法律依据，对数据的有效监管实现了有法可依，并完善了网络空间安全治理的法律体系。API 作为数据传输间的桥梁，应遵守相关法律进行安全监管。2020 年 2 月 13 日，中国人民银行发布《商业银行应用程序接口安全管理规范》（JR/T 0185—2020）金融行业标准，规定了商业银行应用程序接口的类型与安全级别、安全设计、安全部署、安全集成、安全运维、服务终止与系统下线、安全管理等安全技术与安全保障要求。我国传统产业规模庞大，数字化进程呈现行业细分的特点，不同垂直领域的安全需求差异较大。数据开放共享过程中涉及个人隐私、商业机密等数据的安全保护制度不够完善、防护技术亟待提升。现有 API 相关标准和 API 安全攻防体系难以实现规模化复制，导致数据质量规范标准、数据价值量化标准匮乏，因此发展难度也更高。

（二）API 屡受攻击，安全治理存在众多难点

国际国内 API 遭受攻击的事件屡见不鲜，已引发广泛关注。受经济利益驱动的攻击者对各个行业的暴露 API 进行广泛攻击，非法窃取隐私数据，造成严重社会影响。国际方面，2023 年 1 月 19 日，美国某运营商暴露的 API 遭到攻击，导致超三千万名用户的姓名、账单地址、电子邮件、电话号码、出生日期被泄露。2023 年 1 月，数十家全球汽车制造商生产的车辆和车联网服务被披露存在众多 API 安全缺陷，攻击者可利用 API 漏洞非法窃取车辆行驶路线信息。2022 年，美国某平台网站上超五百万账户的电话号码和电子邮件地址遭泄露，而泄露数据是通过漏洞赏金计划中披露的 API 漏洞收集的。国内方面，2020 年，某社交平台 API 遭到爬虫攻击，导致超 5 亿用户信息在暗网出售。2020 年 11 月，某企业被曝出有内部人员有偿租借员工账号，导致数十万条公民个人信息被泄露事件。被泄露的信息中包括地址、姓名、电话等信息。2021 年，某企业的 API 遭到爬虫攻击，涉及用户账号、昵称、手机号、商品等超亿条信息。

新型 API 攻击手段频现，攻击手段呈多样化、隐蔽化发展。随着 API 访问环境的愈发开放、API 数量的极速攀升，攻击者正开发出更多更先进的攻击工具和方法对暴露在外的 API 加以利用，早期防护技术已经无法满足现有安全问题的防护需求。新型 API 攻击主要呈现两个趋势。**第一是攻击手段多样化。**面对受害者的防御策略，攻击者会同时采用多种攻击方法，形成 “地毯式轰炸攻击”模式。针对 API 的

新型安全威胁包括注入攻击、内容篡改、参数篡改等。在新型攻击手段下，传统 API 安全网关提供的身份认证、权限管控、速率限制、请求内容校验等安全机制逐渐失灵。**第二是攻击途径隐蔽化。**合法应用程序所使用的加密通道、端口和协议也为攻击者提供了掩盖其足迹的方法。对于安全专业人员而言，从经由 API 接口传入和传出的众多 HTTPS 请求中拦截、筛选和分析可疑流量则变得更加麻烦。

API 安全治理存在多个难点，传统 API 体系问题日趋明显。为了适应数字化进程的快速发展，政府、企业都会开放大量的 API，由此造成的数据安全风险敞口，传统的 API 安全体系并不能完全应对解决。传统 API 安全体系主要存在四大问题。**第一是 API 资产难以进行全面梳理。**API 资产盘点仍然停留在主机、服务、端口维度，无法清晰盘点所有 API 接口，以及可以输出敏感数据的 API 接口。**第二是难以针对 API 接口进行全面的脆弱性评估。**API 资产数量庞大且其业务相关性强、迭代速度快，安全运营人员难以对 API 遗留的安全漏洞进行安全性评估。**第三是针对 API 接口的威胁检测机制不完善。**当前普遍缺乏对 API 的细粒度检测，只能基于已知特征的 API、勒索软件、木马、病毒进行威胁监测，无法实时发现数据泄露以及针对 API 接口特征的攻击。**第四是发生 API 攻击事件后的事后追溯难。**企业针对 API 的日志管理普遍存在分散存储、日志记录不全、格式不规范等问题，一旦发生网络故障或安全事件时，无法在短时间内定位责任人和泄露路径。

二、 API 爆发式增长催生新的安全挑战

（一） API 攻击趋势：新型攻击手段频现且难以防范

数字化时代与线上化趋势下，API 作为连接不同应用和系统的桥梁，正迅速成为现代企业的核心组件。API 的爆发式增长在为企业提供了更强大、灵活的数据交换和功能扩展能力、创造更多商业机会的同时，也带来了新的安全挑战。API 接口数量的暴增与其安全发展的不匹配，使其成为恶意攻击的首选目标，API 安全问题受到广泛关注。

1. 攻击手段多样化

API 是企业数字化转型的关键组成部分，攻击者可以利用各种不同的手段攻击 API 并窃取敏感数据或者干扰企业的业务流程。常见的 API 攻击手段包括但不限于:

1.1 **拒绝服务攻击**（Denial of Service，DoS）。攻击者通过发送大量占用计算资源的请求，例如大文件下载、高负荷的搜索请求等，使得 API 服务器无法分辨并响应合法请求，从而导致 API 服务器资源耗尽，服务不可用。

1.2 **注入漏洞利用攻击。**攻击者通过在 API 请求中注入恶意代码，以获取未授权的数据或执行未授权的操作。例如，攻击者通过在 API 请求中注入 SQL 注入代码的方式，获取数据库中的敏感信息。

1.3 **未授权漏洞利用攻击。**系统对用户的访问控制无限制，攻击者可直接获取未授权的数据或执行未授权的操作。例如，攻击者可在 API 请求中使用未经过授权的密钥进行 API 调用，从而获取或修改数

据，导致信息泄露，甚至破坏系统功能。

1.4 **安全配置漏洞利用攻击。**攻击者通过发现 API 服务器未正确配置安全措施，利用错误配置发起漏洞攻击。例如，攻击者可以通过访问 API 服务器的管理接口修改 API 的安全配置，以绕过安全措施获取敏感数据。

1.5 **越权访问漏洞利用攻击。**系统对用户的访问控制限制不正确，攻击者可直接获取超出授权的数据或执行超出授权的操作。越权访问包括水平越权攻击和垂直越权攻击两种形式。水平越权攻击是指攻击者通过利用 API 的漏洞，获取其他用户或角色拥有的权限范围。垂直越权攻击是指攻击者通过利用 API 的漏洞，获取比其本身权限更高的权限。例如，攻击者可能会通过利用该 API 漏洞，通过普通用户的凭证来执行管理员级别的操作。

1.6 **参数遍历攻击。**攻击者通过修改 API 请求参数来遍历 API 中的文件或目录，获取未授权的数据或执行未授权的操作。例如，攻击者可通过修改 API 请求参数，对 API 的敏感数据或服务器上的文件进行非法访问。

1.7 **明文传输利用攻击。**攻击者通过截获 API 请求，以获取未加密的敏感信息。

1.8 **代码设计漏洞攻击。**攻击者利用目标 API 中的代码设计漏洞进行攻击。这些漏洞通常是由于编码或设计错误而导致的，可能涉及不安全的数据处理、缓冲区溢出、格式字符串漏洞、逻辑错误、错误

处理等。例如，某些 API 可能会设计特定数据类型的处理方式，这可能引发攻击者利用该设计漏洞构造特定类型的恶意数据，从而导致 API 崩溃或执行恶意代码。

1.9 **第三方组件漏洞利用攻击。**许多应用程序和系统都使用第三方组件，例如开源库、框架和插件，以便快速构建和部署应用程序。此类组件通常由其他开发者编写和维护，在开发过程中可能存在开放性的安全漏洞。攻击者可以利用安全漏洞进行 API 攻击。例如，攻击者可以利用已知的第三方组件漏洞，绕过 API 的安全措施并获取系统权限。这种攻击可能会导致数据泄露、拒绝服务等安全问题。

2. 攻击途径隐蔽化

为了规避现有安全防护设备的检测，提高攻击成功率，越来越多的攻击者会采用如下方式来隐蔽其攻击途径:

2.1 **使用代理服务器。**攻击者使用代理服务器隐藏真实 IP 地址和其他攻击特征（如攻击次数、攻击频率）。这可以使攻击者更难被追踪和识别，从而提高攻击的成功率。

2.2 **滥用逻辑漏洞。**攻击者滥用 API 逻辑漏洞发起隐蔽攻击，例如未授权访问、越权访问、过度的错误提示等。此类行为可能被认为是正常的系统操作，从而不会被管理员或安全人员察觉，更不会引起警报。

2.3 **模拟真实用户请求。**攻击者可以使用一些工具来模拟各种类型的真实用户请求，包括 GET、POST、PUT、DELETE 等请求类型，

也可以模拟多个用户的流量，并使用分布式攻击工具避免被识别。

2.4 **破解 API 密钥。**攻击者可以尝试破解 API 密钥或伪造 API 密钥，以此来绕过 API 的安全措施，如攻击者可以使用字典库来暴力破解 API 密钥。

2.5 **重定向攻击。**攻击者可以利用 API 中的重定向功能来跳过安全检测、混淆或隐藏攻击路径。

2.6 **滥用 API 版本。**API 旧版本可能存在漏洞或者安全问题，攻击者可以利用此类问题进行攻击。攻击者可尝试使用较旧的 API 版本以便绕过最新版本的安全措施，或者利用最新版本中的漏洞来进行攻击。

3. 攻击场景自动化

攻击者对 API 进行自动化攻击的主要目的是提高攻击效率和成功率，同时降低攻击成本和风险。通过自动化攻击工具和脚本，攻击者可以快速发现 API 漏洞，大量尝试不同的攻击方式和参数组合，并快速利用漏洞进行攻击，从而避免人工操作的不稳定性和易出错性。

攻击者针对 API 发起攻击场景自动化的趋势主要有以下方向:

3.1 **增加攻击深度和复杂度。**攻击者倾向于组合多种不同的攻击方法，实现更深入和复杂的攻击。例如，在针对 API 的自动化攻击中，攻击者可能会结合使用扫描、爬虫、渗透测试等多种工具，以实现更广泛和深度的攻击。

3.2 **模块化攻击方式。**攻击者将攻击过程分解成多个模块，每个

模块都可以独立完成特定的攻击任务。这些模块可以根据攻击者需要进行灵活组装，以快速完成攻击工具的制作，使攻击更高效。因此，攻击者越来越倾向于使用模块化的攻击工具来发起针对 API 的攻击行为。

3.3 **与 SaaS 攻击平台结合。**攻击者日益倾向于将自动化攻击工具与 SaaS 攻击平台结合使用。例如攻击者会使用 SaaS 化攻击平台进行自动化攻击工具开发。SaaS 化攻击平台包括代理 IP 平台、接码平台等，能够为攻击者提供更强大的攻击工具和更高效的攻击方法。

3.4 **攻击工具市场化。**攻击者将自动化攻击工具的使用权出售给其他攻击者，以获取更高的收益。这些攻击工具可以被打包成易于使用的软件，称为"攻击套件"，并在地下黑市上出售。攻击者甚至会为攻击工具提供售后和更新服务，以保持工具的有效性。

3.5 **利用 AI 和机器学习技术进行攻击。**随着 AI 和机器学习技术的迅速发展，攻击者开始利用新技术实现自动化攻击。例如，攻击者可以利用 AI 技术自动生成攻击代码或规避安全防御措施。

（二） API 安全挑战：API 安全已成为网络安全的最大威胁之一

1. API 资产缺乏可见性，或将带来无法量化的风险

随着 API 的广泛应用和数量的快速增长，企业可能面临大量的 API 资产，包括公开的、私有的和第三方的 API。这些 API 资产可能存在于企业的各种应用程序、云服务和微服务中，形成了庞杂的 API

暴露面。企业对于 API 暴露面的管理和保护是确保 API 安全的重要组成部分。然而，API 资产的发现却是一个具有挑战性的任务，企业进行 API 资产发现时可能遇到如下难点:

1.1 API **资产的隐秘性**。API 资产不会自主对外宣告，这使得它们很难被发现。内部系统的 API 资产、已静默的 API 资产、东西向调用的 API 资产、第三方提供的 API 资产等由于其隐蔽性将成为 API 资产发现过程中的盲区。

1.2 API **资产的动态性**。API 资产可能随着业务需求而不断变化，这将导致 API 资产变更愈加频繁和快速，需要企业建立持续动态的 API 发现流程。

1.3 API **协议的多样性**。现代 API 协议的多样性使得 API 资产的发现更加具有挑战性。例如，GraphQL 和 RPC 是两种不同的 API 协议，它们具有不同的语法和语义，需要采用不同的方法进行资产发现。

1.4 **部署方式的复杂性**。API 资产可能采用不同的部署方式，包括网关发布和私有化部署。网关发布通常是一种中心化的方式，它允许在单个入口点上集中所有 API。然而，私有化部署则会使 API 资产更难以发现，因为它们可能被分散在多个环境中，包括本地网络和云环境。此外，API 资产的创建者、使用者和管理者也可能不同，这也增加了 API 资产发现的难度。

1.5 **开发管理困难**。企业的 API 资产累积以及开发管理规范化是逐步建设的过程。然而，在这个过程中，历史建设或第三方创建的 API

资产可能无法被纳入开发管理的范围。另外开发管理规范也无法完全保证 API 开发过程的执行质量。这使得 API 资产的开发管理变得更加困难，也使得企业无法只从开发管理规范来实现 API 资产的可见性。

1.6 **组件引入的不确定性。**在开发过程中，开发人员可能会引入各种组件来加速开发进度、提高开发效率。这些组件包括但不限于第三方库、框架、工具等。然而，这些组件包含许多 API 接口，这些 API 接口可能被用于开发新的功能，或是作为开发现有功能的基础，极有可能导致 API 资产失去管理，带来一系列潜在风险，尤其是当这些组件由第三方提供的时候。

1.7 **程序框架的多样性。**API 资产往往使用不同的程序语言和框架，相应地就需要考虑不同的技术来进行 API 资产的发现。

2. API 攻击面不断变化，企业难以管理和把控

企业在完成 API 资产梳理以后，需要对 API 资产所形成的攻击面进行安全管理。而 API 攻击面的动态变化属性导致了企业需要实时且可持续的安全管理手段。引发 API 攻击面动态变化的原因有如下几点：

2.1 **业务和技术变化。**企业的业务需求和技术环境都在不断变化。API 资产可能会随着业务场景的变化而上线或下线，或者在技术架构的升级过程中发生变化。这些变化都会导致 API 的攻击面发生动态变化。

2.2 **持续演化的安全威胁。**安全威胁是一个不断演化的过程，攻击者会不断发现新的漏洞和攻击技术。即使在完成 API 资产梳理时没有发现漏洞，但在之后的时间里，新的漏洞也可能会出现。攻击技术的演化导致了 API 攻击面的安全威胁的不确定性。

2.3 **第三方组件及服务。**企业在 API 开发部署过程中常使用第三方组件、容器以及第三方服务 API。这些组件、容器和服务 API 可能存在潜在的漏洞和安全问题，攻击者可以利用它们来攻击企业的 API 系统。组件、容器及第三方服务 API 的安全问题不可控也导致了 API 攻击面的不可控。

基于上述业务、技术及安全威胁的不断发展及演进，API 攻击面的动态变化是不可避免的。可实时持续发现 API 资产的状态变化，感知 API 资产的未知漏洞和已知漏洞的修复状态是企业实现 API 攻击面管理的决定性技术要素。API 的攻击面管理需要使用实时监测和自动化分析工具，帮助企业及时发现 API 资产的状态变化，对 API 资产进行实时的安全评估，从而针对问题资产快速做出反应并采取适当的管理和保护措施。

3. 现有防护体系难以对 API 风险进行有效识别

攻击者针对于 API 资产的攻击手段存在多样化、隐蔽化、自动化的趋势，可以轻松突破企业对于 API 的限频、限量及认证鉴权等基础的防护手段。除此之外，大部分攻击者利用 2023 OWASP API Security Top 10（对象级别授权失效、身份认证失效、对象属性级别授权失效、

资源消耗无限制、功能级别授权失效等)所列举的逻辑漏洞进行攻击，企业很难从流量中提取区别于正常用户的特征信息，从而对攻击进行阻断。

针对 API 的逻辑攻击，传统的安全防护体系常常存在如下难点:

3.1 **攻击流量无特征。**攻击者利用 API 逻辑漏洞时，通常会模拟正常的 API 请求，使用合法的 API 参数和协议。这使得企业现有安全设备（API 网关、WAF 等）难以区分攻击流量和正常流量。因为攻击流量与正常流量在协议和参数上并无明显区别，传统的基于规则的安全设备难以准确检测和识别这种攻击。

3.2 **攻击行为跨越多个请求。**API 逻辑漏洞通常涉及多个 API 请求上下文的交互，攻击者通过多次请求的组合来实现攻击目的。然而，现有安全设备难以理解攻击的完整上下文，无法对多个 API 请求的关系和交互进行深入分析。这限制了它们在识别和检测 API 逻辑攻击的准确性和有效性。

3.3 **缺失敏感数据的理解。**API 逻辑攻击通常涉及敏感数据的泄露、篡改或者未经授权的访问。然而，现有安全设备主要关注网络协议和数据格式，难以理解流量中传递的敏感数据的类型及内容，这导致它们无法对涉及敏感数据的恶意攻击进行准确的检测和识别。

虽然 API 网关设备和 WAF 设备都开始引入人工智能学习算法来不断分析攻击者的攻击方法和行为模式，并根据这些信息提炼出攻击者的特征向量，自主地调整自身的规则和策略，但是在应对攻击风险

的过程中，仍存在三大问题难以解决。**第一是导致误报或漏报。**AI 学习需要大量的数据集进行训练，如果数据集中存在偏差或噪声，可能会导致误报或漏报攻击。同时，攻击者也可以采用无特征的攻击流量使 AI 学习无法归纳出特征向量或者归纳出错误的特征向量来规避检测，导致设备的失效。**第二是难以应对新型攻击。**尽管 AI 学习能够识别历史攻击模式并对其进行预测和防范，但是对于新型攻击模式，设备可能无法及时识别和防范，需要进行额外的人工干预和调整。**第三是需要大量的计算资源。**AI 学习需要大量的计算资源来训练模型和执行实时检测，这需要企业投入更多的成本来购买和维护高性能的设备和系统。

除上述因素以外，企业内部系统以及东西向的流量也很难接入到 WAF 设备及 API 网关设备，无法对上述系统或者流量进行 API 风险的识别。因此，企业现有的防护体系难以有效应对 API 攻击所带来的新的挑战。

4. API 安全治理成企业数据安全合规的必要举措

对企业而言，围绕数据的全生命周期，从数据采集、存储、访问、使用、销毁构建完善的数据安全体系成为安全建设工作的重点。

目前，大多数企业在数据采集、存储、数据库访问方面做了比较全面的安全建设，在数据的流动访问方面的安全建设意识也正逐步萌芽和发展；而 API 作为连接数据与应用的主要通道，成为了数据传输中最薄弱的环节之一。传统的安全防护手段主要以边界安全为主，在

安全能力无法覆盖到 API 敏感数据的保护，从而导致 API 数据泄露和违规访问的风险依然无法规避。

由于 API 防护的缺失，企业对外暴露的 API、开放的 API、API 通信中的敏感数据等问题均未得到应有的重视。攻击者可以利用 API 的认证授权、数据过度暴露、数据可遍历、配置或设计错误等漏洞发起攻击进行数据窃取。企业有效的落实 API 的安全治理，已经成为其满足监管合规要求的必要举措。

5. 企业跨部门协同存难题，API 安全全生命周期治理难实现

API 安全的治理需要从 API 生命周期的角度考虑，即从 API 的规划、开发、测试、部署、运行和下线等各个阶段全面考虑安全性，避免安全漏洞被潜在的攻击者利用。而实现全生命周期的安全治理需要跨多个部门，包括开发、测试、安全、运维等部门，达成共识并建立协作机制，共同保障 API 的安全。当前企业在 API 安全治理的跨部门合作中可能存在五方面问题。**第一是缺乏有效的沟通渠道和协作机制。**不同部门之间缺乏有效的沟通和协作机制，导致信息共享不畅，合作难度大。**第二是职责划分不清。**不同部门对 API 安全治理的职责划分不清，导致工作重叠或空缺，影响协作效率。**第三是缺乏协同工具和流程支持。**缺乏协同工具和流程支持，导致协作效率低下，难以有效跨部门合作。**第四是优先级和利益冲突。**不同部门的工作重心和利益有所不同，导致协作中可能存在优先级和利益的冲突。**第五是技**

术难度大。API 安全涉及到技术和业务的复杂性，不同部门可能缺乏相应的技术和经验，导致难以合作和共同解决问题。

三、 API 安全防护体系建设思路

（一） 总述

建立 API 安全防护体系是确保企业信息安全的必要举措，也是企业可持续发展的重要保障，该体系的构建过程是长期且持续的，且涉及多部门参与。安全部门作为 API 安全的第一责任人，可以优先基于 API 生命周期构建安全防护模型，再结合自身业务情况，进一步基于 IPDRR 模型实现 API 安全管理的闭环。

注: IPDRR 模型是一种安全管理框架，其核心思想是通过识别、防护、检测、响应和修复五个步骤来全面评估和管理组织的信息系统、网络和应用程序的安全。通过实施 IPDRR 模型，组织可以制定综合的安全策略，包括技术、运营、人员、法律和合规等方面，以保护组织的信息和资产免受安全威胁的侵害。

（二） 基于 API 生命周期构建安全防护模型

对企业而言，想要做好安全管理，全生命周期的 API 管理很有必要。首先，API 是企业的私有化资产，从设计和开发就是在企业内部完成，API 问题的产生通常也是由企业自身产生。所以，企业在管理角度上，有必要从开发和设计环节就开始考虑整个 API 的管理。

其次，API 在整个业务生命周期当中变化非常快，API 实现逻辑一旦发生变化，很容易引入新的风险。因此，从 API 全生命周期来考

虑 API 安全，围绕规划、开发、测试、部署、运行到下线的每一个环节加强安全建设显得尤为重要。

图 1 API 生命周期安全管理模型

1. 规划阶段：引入威胁建模理论

在 API 全生命周期中，规划阶段至关重要，它决定了整个 API 开发过程的方向和目标。在规划阶段，企业应根据自身业务需求对风险进行评估，并制定 API 安全规划和策略，以确保 API 的安全性和可扩展性。威胁建模作为一种系统方法，是安全规划中的重要部分，它帮助企业评估 API 存在的威胁和漏洞，并采取相应措施保护 API 的安全。

在规划阶段，威胁建模可帮助企业评估 API 的安全风险，为后续的保护措施提供指导。

企业可采用 ASTRIDE 和攻击树等模型进行威胁建模，分析攻击者的目标、手段和途径，并根据分析结果制定相应的威胁建模。同时，结合自身业务特点和历史安全事件总结，制定安全策略和措施，为后续的开发、测试、部署和运行提供指导，确保 API 在全生命周期中的安全性。

ASTRIDE 安全分析模型（Asset, Threat, Vulnerability, Risk, Impact, Detection, and Response）是一种综合了攻击树和组合方法的安全风险评估模型，它通过对系统进行建模和分析，可以识别出系统中存在的安全漏洞和潜在攻击路径，并且可以对这些漏洞和攻击路径进行量化的评估和排序。

当企业在规划阶段引入威胁建模理论时，安全工程师应参与其中以确保 API 的安全防护能力：安全工程师可以在设计阶段对 API 进行威胁建模，识别 API 的安全威胁和漏洞，并提出改进方案；安全工程师还可以提供安全规范和最佳实践，以确保 API 的安全性和可用性，资源允许的情况下在设计阶段对每个 API 进行威胁建模，线上的 API 风险将会大大减少；资源有限的情况下，安全工程师可建设自动化威胁建模的能力，采用“重点业务人工评审”和“非重点业务自动化威胁建模”相结合的方式，对核心高风险 API（如账号登录、文件上传、营销活动等关联 API）进行覆盖。

此外，安全工程师还可以在规划阶段为 API 安全设计提供指导和建议，包括让开发人员遵循安全编码标准和最佳实践，加强代码审计和安全测试，以及设计强有力的认证和授权机制等，同时协助企业评估 API 的安全性，提出适当的安全措施和流程，最大程度降低风险，加强企业数据资产保护。

下表中的安全检查表和最佳实践可以作为威胁建模阶段的参考：

表 1 API 安全检查表及最佳实践

序号	名称	来源
1	OWASP REST 安全检查表	https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html
2	API 安全工具和资源（APIsec）	https://github.com/arainho/awesome-api-security
3	API 安全检查表	https://github.com/shieldfy/API-Security-Checklist
4	JWT 安全检查表	https://pragmaticwebsecurity.com/files/cheatsheets/jwt.pdf

2. 开发阶段：加强安全开发建设，引入安全工具

在开发阶段，企业需要引入一系列安全措施来加强 API 的安全性。首先，可以制定安全开发规范，明确 API 开发的安全要求和规范，确保开发人员在设计和编写 API 时充分考虑安全因素，从而减少潜在漏洞的存在。

其次，企业需要开展安全开发培训，将安全意识渗透到开发人员

的日常工作中。培训应该包括最新的安全技术和最佳实践，同时也应该提供实践指导，帮助开发人员了解如何构建安全的 API，如何使用安全开发工具和技术来发现和修复漏洞。

为了加强安全开发意识，安全团队和开发团队可以共同建立安全社区，定期举办安全相关的活动，分享最新的安全知识和案例，让团队成员共同学习、成长和分享。

此外，企业可以使用安全开发工具来提高 API 的安全性。这些工具可以帮助开发人员发现并修复 API 中的安全漏洞，例如静态代码分析工具、API 安全验证工具等。这些工具还可以自动检测代码中的漏洞和潜在安全风险，并及时提醒开发人员进行修复。

通过上述安全措施的引入和实施，可以在 API 开发阶段有效提升企业的 API 安全防护能力，减少潜在安全风险和漏洞。如下列表中的安全编码和开发规范，可供借鉴及参考：

表 2 API 安全编码和开发规范

序号	名称	来源
1	OWASP 安全编码实践	https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/
2	腾讯代码安全指南	https://github.com/Tencent/secguide
3	威胁猎人 API 安全开发规范	https://www.threathunter.cn/reportDetail/3c4pMlAzZMlEuHstjOSPMCg12EbXgt

在开发阶段可以考虑使用如下工具加强安全建设:

2.1 **引入静态代码分析工具**。嵌入到 CI/CD 或者研发流程中，对代码进行静态扫描，识别潜在的代码缺陷和安全漏洞。

2.2 **引入 API 管理工具**。对 API 的文档进行集中统一管理，监测 API 的变更迭代的过程数据，帮助团队实现 API 的快速迭代和管理。

2.3 **引入 API 安全验证工具**。验证 API 安全实现的正确性，以保障验证工作尽可能做到全面，相关工具如 FuzzDB、个人数据隐私监测类工具。

3. 测试阶段：使用 AST 类工具进行自动化漏洞测试，提高覆盖率

在测试阶段，企业需要在实现 API 业务逻辑、API 稳定性、API 性能测试的基础上增加 API 安全的自动化测试。进行 API 安全自动化测试的目的是为了自动化扫描每一个 API，从“请求输入”与“应答响应”两部分去分析 API 是否存在漏洞。API 安全测试的常规内容主要包含 API 身份验证、授权、输入验证、异常处理、数据保护、安全传输以及 HTTP Header 安全性等。

API 数量多且在不断地迭代更新，因此需要借助自动化工具来辅助进行安全测试，以便在测试阶段提前发现 API 在研发过程中存在的安全漏洞。常用的工具有以下三类：SAST、DAST、IAST 工具，可以在测试阶段提前发现存在的 API 安全漏洞。

静态安全检测（Static Application Security Testing，SAST），其特

点是分析应用程序的源代码或二进制文件，通过语法、结构、过程、接口等来发现应用程序的代码是否存在漏洞。

动态安全检测（Dynamic Application Security Testing，DAST），其特点是在应用程序的动态运行状态下，模拟黑客攻击行为，分析应用程序的响应，而确定应用程序是否存在漏洞。

交互式安全检测（Interactive Application Security Testing，IAST），相当于是 DAST 和 SAST 结合的一种安全检测技术，通常会在应用程序中添加探针或 Agent 代理，收集应用程序、Web 容器、JVM 中的执行日志和函数调用信息，结合请求输入与响应消息，分析应用程序中是否存在漏洞。

4. 部署阶段：确保 API 的部署和配置的安全性

在 API 部署阶段，企业需要采取一系列措施来保证 API 的安全性。

首先，在 API 部署前，企业需要对服务器进行安全扫描及加固，包括对服务器系统、应用、中间件等进行安全配置，开启防火墙等安全措施，减少攻击面，降低服务器受攻击的风险。

其次，企业需要确保数据在传输过程中的安全性，可以通过使用 SSL/TLS 等加密协议来加密数据传输，防止数据在传输过程中被窃取或篡改。

此外，企业需要根据业务需求限制对 API 的访问，确保只有授权的用户或系统能够访问 API。可以通过 IP 白名单、Token 鉴权等方式

进行访问控制，避免未授权的用户或系统访问 API。

最后，企业可以使用安全配置检测工具来检测 API 部署和配置的安全性，包括服务器的操作系统、中间件、数据库等的配置，防止因配置错误或漏洞而导致的安全风险。

在部署阶段可以考虑使用如下工具加强安全建设：

4.1 **漏洞扫描工具。**通过模拟攻击，帮助企业自动发现 API 部署时存在的常见漏洞和安全配置问题，并提供修复建议。

4.2 **安全配置管理工具。**可以帮助企业管理 API 的安全配置，监控配置的变更，及时发现不安全的配置并采取措施进行修复。

5. 运维阶段：利用工具及时感知 API 攻击威胁

在运维阶段，企业可以利用各种安全工具来及时感知 API 的攻击威胁，实现 API 的分层防护，确保 API 的安全性。这些工具包括 DDoS 防火墙、Web 应用程序防火墙（WAF）、API 网关、API 安全审计工具等。

5.1 DDoS **防火墙。**针对分布式拒绝服务攻击（DDoS）的安全工具，DDoS 防火墙可以通过各种技术手段，包括 IP 黑名单、限制流量等，识别和过滤 DDoS 流量，保护 API 免受 DDoS 攻击。

5.2 Web **应用程序防火墙**（WAF）。通过分析来自客户端的 HTTP 请求，并根据其规则库对其进行检查，以检测和阻止恶意攻击。在 API 防护过程中，WAF 通常用于阻止 SQL 注入、跨站脚本攻击（XSS）等攻击请求，以及常规的 BOT 攻击等。WAF 可以提供实时监测和分

析 API 的访问日志，并能够迅速发现异常行为。WAF 可以记录所有请求数据，包括来源 IP、用户代理、参数等，并能够根据事先设置的安全策略进行分析和识别。当异常行为被发现时，WAF 可以立即采取行动，根据预设的安全策略阻止恶意请求，从而快速响应和快速阻止攻击。此外，WAF 也提供了一些高级功能，如基于 AI 的安全分析、虚拟补丁等，可以根据攻击行为自动调整安全策略。在运维人员发现新的攻击行为后，可以在 WAF 上进行配置和调整，以提高 API 的安全性。

WAF 在实际使用中也存在着一些片面性。例如，WAF 不能解决所有的安全问题，大部分利用 API 逻辑漏洞进行攻击的行为可以绕过 WAF 的防御。另外，如果 WAF 的规则库更新不及时或者设置不当，就会导致误报或漏报，从而影响 API 的正常使用。因此，企业在使用 WAF 时需要进行定期维护和更新，以保证其有效性和准确性。

5.3 API 网关。能够实现认证、授权、访问控制和数据脱敏，从而提高 API 的安全性。API 网关具有身份认证、访问控制、数据校验、限流熔断等功能，可以帮助安全团队管理 API。但是，对于内部开发人员来说，使用 API 网关需要打通持续集成（CI/CD）与 API 网关的接口，准备好 API 导入数据供 CI/CD 调用，这会增加开发人员的工作量，并影响发布进度。此外，当所有后端服务的流量都必须通过 API 网关进行通信时，会对原有通信性能和 API 的稳定性产生影响，这是推进 API 网关工作的负责人员需要面对的最大挑战。因此，是否需要

使用 API 网关需要根据业务的实际情况进行评估。

5.4 API **安全审计工具**。能够通过流量或者 WEB 日志的持续分析，动态识别企业的 API 资产、API 资产关联的敏感数据、账号信息，具备对于各类资产进行自动分级分类的能力，支持通过 API 资产的流量差异比对发现僵尸 API、影子 API，老版本、功能重复的 API 资产。该工具通过对 API 资产的流量上下文及敏感数据的持续分析，可以实时发现 API 资产的授权类、认证类、数据暴露类、配置及设计不合理等各类逻辑漏洞，通过外部情报和机器学习模型来感知针对 API 的低频慢速的攻击风险，使用账号、IP、访问时间、访问 API、访问敏感数据等多重维度建设的 UEBA 模型来感知账号共用、借用、盗用等行为导致数据泄露的攻击风险。

6. 下线阶段：及时下线僵尸影子 API

在 API 下线阶段，企业需要进行特定的安全下线处理措施来保障系统安全，数据清除和安全审计是两个关键的方面。数据清除是指将与 API 相关的数据和配置清除干净，以防止敏感信息被未经授权的人员获取。安全审计则是指对 API 下线过程中对 API 资产进行审计和维护，以及查看是否存在安全风险，以便在今后的安全事件处理中作为重要依据。API 资产审计工具支持企业动态 API 资产的维护，识别哪些 API 是应该下线的，如僵尸 API，老版本 API、功能重复的 API，通过审计工具的安全事件对已经完成下线操作的 API 资产进行二次确认，对于使用频率较低的 API（如营销活动 API、招聘 API 等）也

可以进行有效的资产分类及业务监控。

（三） API 安全闭环管理要求和建议

在日益严峻的网络安全环境下，API 安全建设绝非易事。API 全生命周期管理涉及企业各部门角色的参与，投入工作量极大，企业应该根据实际情况来调整投入产出比。而从高效防御的角度出发，企业可以从 API 的上线运行阶段入手，基于 IPDRR 安全模型实现 API 安全闭环管理，结合自身的业务情况有序开展 API 安全实践。

基于 IPDRR 模型，企业可通过持续识别 API 资产潜在威胁和漏洞、提供安全保护措施、实施实时监测和事件检测、迅速响应安全事件、以及实施恢复策略和业务持续性计划，全面保护 API 的安全性和可靠性，最大化降低甚至避免 API 风险。

图 2 API 安全闭环管理

1. Identify：构建可持续的识别能力

1.1 API 资产动态识别

API 资产动态识别能力是 API 提供者和攻击者之间的竞赛，要在攻击者之前发现 API，提前感知和了解到系统可能的攻击面。因为 API 是不断在研发迭代的，所以持续动态梳理 API 的能力显得至关重要。企业可以从 API 安全网关、负载平衡或交换机镜像的网络流量中对 API 资产进行动态识别。

API 资产的动态识别需要遵循全面性原则。首先，API 资产的动态识别需要覆盖企业的全部应用系统类型，包含终端用户系统、内部应用系统、合作方系统、第三方组件管理系统等范围；其次，API 资产的动态识别需要全面支持各类应用协议，如 REST、RPC、GraphQL 等。

API 资产的动态识别需要遵循准确性原则。API 资产的动态识别过程中首先需要具备有效去除扫描流量等非正常请求的干扰的能力，避免非正常流量导致无效 API 资产数量无限扩大，此外需要具备针对存在参数变量类似、路径高度重合的 API 资产合理规约的能力，避免未合理规约导致同属性 API 资产数量无限扩大，最后需要具备针对接口类型参数化的 API 资产合理拆分的能力，避免未合理拆分导致 API 资产安全特征模糊。API 资产的数量无限扩大或者未合理拆分，都会导致 API 安全运营的无效工作增大，最终使得 API 安全运营工作难以开展。

1.2 敏感数据动态检测

API 资产作为企业流动数据的载体，企业应对流动数据进行数据分级分类治理，对 API 承载的敏感数据进行持续监测评估，动态识别 API 接口中的敏感数据类型并自动生成 API 接口与敏感数据类型的映射关系。

敏感数据动态识别能力应支持自定义的敏感数据类型识别规则，并支持自定义敏感数据类型的分级分类。

1.3 账号资产自动提取

账号资产自动发现能力是企业数据流动风险发现的基础。API 资产的交互过程中承载了账号及其鉴权信息，企业可以通过从 API 交互过程中自动提取出账号以及账号的访问日志信息，从而可以清晰掌握“谁通过什么 API 访问了什么敏感数据”。企业基于上述访问日志信息可以对账号异常行为进行有效的治理及溯源。

1.4 API 资产自动化管理

API 资产自动化管理能力是 API 资产安全运营的基础。企业可从 API 资产的网络属性、应用属性、关联敏感数据类型、业务场景属性以及使用组件属性等维度对资产进行自动化分级分类。企业可以根据 API 资产的分级分类结果确定安全运营工作的优先级，有效提升 API 安全运营工作的效率。

1.5 漏洞自动发现

企业需要建设具备持续监测 API 存在的授权、认证、数据过度暴

露、配置、逻辑设计错误等漏洞的自动化识别能力。

漏洞的自动化识别能力需要遵循全面性原则，不仅需要对自身业务存在的漏洞进行覆盖，也需要覆盖对第三方开源组件及中间件进行覆盖。

漏洞的自动化识别能力需要遵循准确性原则，在漏洞识别的过程中需要去除扫描流量的干扰，可以有效降低扫描流量导致的漏洞误判，从而极大降低安全人员对漏洞事件审计工作的工作量。

2. Protect: 构建实时的防护能力

2.1 认证授权

建设有效的认证授权是 API 安全的关键，是保护 API 系统安全的第一道防线。企业应建设统一的认证授权体系，确保只有通过认证的用户或者应用程序才能访问被授权的 API 接口。企业可以基于 OAuth2.0 或 Open ID Connect 等协议完成认证授权体系的建设。

2.2 访问控制

企业应采取访问控制措施，限制 API 接口的访问权限，以确保 API 系统仅被授权用户或者应用程序所访问，并提供完整的访问控制日志用于审计。访问控制的具体实施措施可以包括基于角色的访问控制、基于属性的访问控制、多因素身份认证等方式。企业可以根据实际需求选择合适的访问控制策略。

2.3 限流限速

企业应针对 API 接口的访问采取限制措施，限制 API 接口的访

问频率、访问次数等，以避免 API 接口被恶意攻击者利用导致资源无效消耗。API 接口的限制措施的具体实施可以包括 IP 地址限制、请求次数限制、黑白名单限制等。

2.4 信息传输保护

企业应保障 API 在信息传输过程中的机密性和完整性。信息传输保护可以通过采用 TLS/SSL 协议、HTTPS 等方式来实现。TLS/SSL 协议是一种安全传输协议，可以保证通信过程中的数据传输安全。

2.5 数据加密

企业应对敏感信息的交互进行加密和脱敏处理，特殊的敏感信息需要被加密或做脱敏处理，例如身份证号、银行卡号、手机号等，以减少敏感信息的泄漏风险。

3. Detect：构建全面的检测能力

3.1 API 逻辑攻击检测

当攻击者利用 API 的逻辑漏洞发起数据爬取、恶意注册、扫号、爆破、短信轰炸、营销欺诈的恶意攻击时，为了突破现有安全设备（如 WAF、API 网关）的检测策略，一般选择接入代理 IP SaaS 服务商来实现低频访问，接入虚拟手机卡 SaaS 服务商来突破账号身份限制。当攻击者针对 API 发起低频无特征攻击时，原有安全设备的风险检测模型就会失效，因此，企业在基于流量特征及行为特征构建针对 API 逻辑漏洞的攻击检测模型时，应当引入攻击资源情报作为检测模型的判定依据，从而准确识别出针对 API 的逻辑攻击风险，有效降低攻击

检测的误判率，提升 API 安全运营的工作效率。

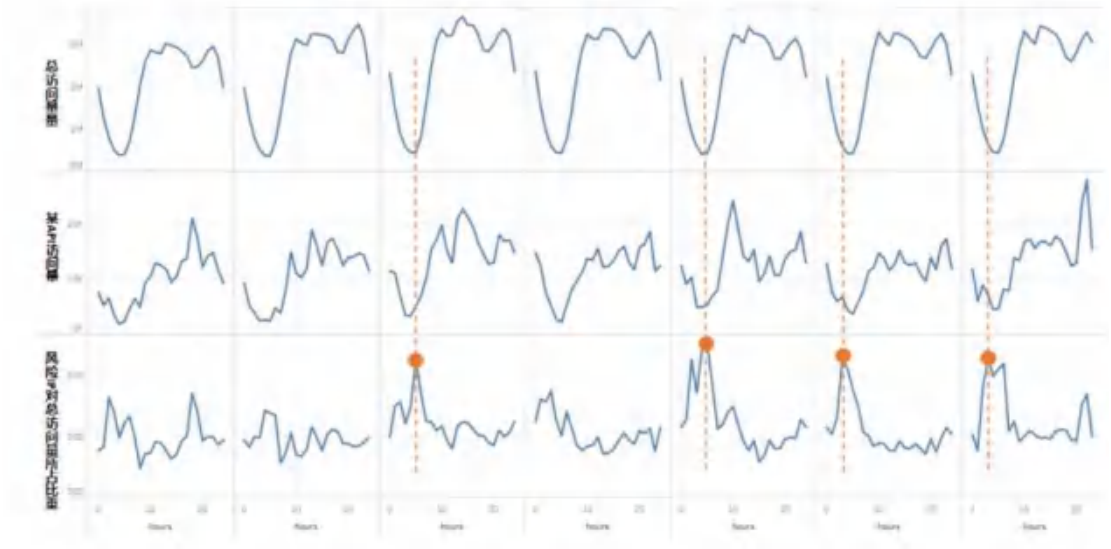

图 3 某互联网企业业务请求量

图 3 中第一条曲线是总业务流量趋势，非常规律；第二条曲线是某个关键 API 的访问量，波动正常，如果基于规则引擎识别，容易产生误判；第三条曲线是该 API 下风险 IP 请求量占比，在 API 访问量低的情况下，仍然能够精准发现攻击风险。

3.2 账号违规操作检测

攻击者可以利用内鬼来盗取敏感数据，也可以借助撞库、账号暴力破解、社工钓鱼等方式获得敏感的账号，进而利用授权账号进行违规的操作，如违规获取大量敏感数据。账号违规操作行为的检测，需围绕账号的历史行为以及账号和同组织的人在登录环境、登录 IP、获取数据的时间、访问站点和数据的情况等行为方面构建基线，基于 UEBA（User and Entity Behavior Analytics 用户实体行为分析）的模型来监测账号违规操作的行为。

3.3 IP 异常行为检测

攻击者的攻击都需要通过 IP 资源来完成，针对 IP 构建历史行为画像如数据访问、地理位置、API 访问序列、风险情报画像等，基于机器学习的方法检测 IP 的异常行为，如路径扫描、漏洞扫描、单一 API 请求过多、境外 IP 获取敏感数据等异常行为。基于 IP 风险情报来构建的检测模型，准确率高，可解释性强。

4. Respond：构建高效的响应能力

4.1 攻击威胁分析与响应

企业应建设安全大数据分析平台、安全信息与事件管理系统（SIEM）等，制定相应的应急预案和流程，帮助企业对所有 API 安全事件进行统一监测，并在发生威胁事件时快速感知和响应。通过实时监测和数据分析，企业可以更快地发现潜在安全问题，及时进行预警和处置。同时，企业还可以在日常运营中积累足够的数据和经验，进一步提高 API 的安全性和可靠性。

4.2 攻击威胁预警和溯源

企业需要借助风险情报和大数据分析技术，对 API 的访问日志进行分析，挖掘攻击流量，并对攻击流量进行解释和溯源，做到及时预防和响应。可从业务流量中细化攻击行为，结合攻击者历史情报信息对攻击者的资源、手法、意图、 团伙情况、潜在目标进行关联跟踪分析，进而构建攻击者情报库，对攻击者进行持续跟踪。

4.3 数据泄漏预警和溯源

企业应对存在数据信息贩卖的情况进行有效监测。基于暗网等数据源的监测数据，进行脏数据过滤、二手信息过滤，结合数据售卖者的置信度对交易数据准确性和及时性进行分析，提炼出准确的数据泄漏预警信息，及时预警数据泄漏事件。同时，利用大数据日志平台，对泄漏的数据进行追踪溯源，找到泄漏源头，针对性进行修复，避免大规模的数据泄漏事件发生。

5. Recover：构建闭环式的修复能力

5.1 漏洞修复管理

企业应构建漏洞修复管理机制，统一收集 SRC、渗透测试、基于旁路流量的 API 安全工具持续挖掘安全事件中暴露的漏洞，做好漏洞的修复过程管理，以避免已知漏洞被攻击者利用。企业可以通过修复漏洞的方式提高 API 接口的安全性和可靠性。

5.2 安全政策和规范更新

针对安全事件中暴露出来的安全问题，及时更新安全政策和规范，以确保员工遵守最新的安全要求。同时，企业还可以针对安全事件的经验和教训，进行相应的安全策略和规范的调整和改进。

5.3 加强员工安全教育

通过培训和教育，提高员工的安全意识和技能，让员工能够更好地理解安全政策和规范，并在日常工作中遵守相应的安全要求。这样可以有效减少因员工失误而导致的安全事件。

5.4 安全事件总结和分析

对检测到的安全事件进行总结和分析，评估安全事件对企业的影响和损失，及时调整和改进安全策略和措施，以避免类似事件的再次发生。通过总结和分析，企业可以不断完善 API 接口的安全性和可靠性，提高企业的整体安全水平。

四、 API 安全未来发展趋势展望

在数字化浪潮下，企业加速转型步伐，API 漏洞所造成的安全挑战不断扩大，攻击手段频现且难以防范，API 攻击事件以及防护也受到了业内的广泛关注。新形势下，对于 API 安全攻防的发展趋势及未来展望如下。

建立基于不同行业需求的 API 安全体系标准。在引导和鼓励企业、安全厂商深化数字化业务的同时，还应鼓励安全厂商加快针对 API 安全新技术新方法的研究与实践，推动 API 安全的体系化发展。针对不同细分行业，积极开展符合行业发展规律的 API 安全防护标准制定，推动引导 API 防护技术发展。2022 年，中国信息通信研究院联合多家企业撰写了《应用程序接口全生命周期安全管理成熟度要求》，从 API 开发、测试、发布、运维、迭代、下线等各个阶段规范 API 安全能力，对助力企业不断优化升级安全技术，落地 API 全生命周期安全防护体系具有重要意义。企业可参照该标准对自身 API 安全防护体系进行设计、管理和评价。

提高人员安全意识。企业安全管理及运维人员应充分认识到 API 安全的重要性以及泄露的危害性。增强人员安全意识可以大幅降低安

全风险。企业可通过定期培训等方式对从业人员进行安全意识、安全素养的训练。

强化发展 API 融合防护技术和解决方案。强化发展融合动态防护、机器学习、情报分析、行为分析、特征识别等技术的新型 API 安全防护解决方案，以解决 API 面临的新型威胁，引导和鼓励企业、安全厂商在深化数字化业务的同时，加快针对 API 安全新技术新方法的研究与实践，从而推动 API 安全的体系化发展。

附录 1 2022 年全球 API 攻击重要事件

（一） 某大型社交平台 API 安全漏洞暴露超 500 万用户数据

2022 年，某大型社交平台报告了从 2021 年底到 2022 年发生的 API 泄露事件，暴露了超 500 万用户账户的个人信息数据（实际账户数量可能要高得多）。该 API 漏洞允许任何一方在没有任何认证的情况下，通过提交电话号码或电子邮件，获得任何用户的 ID，使得攻击者可以利用该漏洞进行大规模“扫号”攻击。

虽然该平台修复了这个漏洞，但仍然暴露了数百万用户的私人电话号码和电子邮件，部分数据在暗网上出售，甚至是被免费发布的。网络犯罪分子可能会利用从这次泄露中收集的信息，针对用户进行电子邮件网络钓鱼、语音网络钓鱼和诈骗，试图欺骗用户交出个人信息和登录凭据。

（二） 某大型电信公司 API 漏洞导致巨大经济损失

2022 年，某大型电信公司报告称有 1000 万个账户被攻破，攻击者随后勒索了 100 万美元，并声称拥有超过 1100 万条用户记录。这名黑客威胁说，如果该公司在一周内不付款，他就将这些数据打包出售。

未验证 API 是此次事件的罪魁祸首，用于测试的 API 被暴露在互联网上，并且缺少身份验证检查，也就是说互联网上的任何人都可以向此 API 发出数据请求，而无需提供任何凭据或令牌来证明其身

份。还有另一个问题是攻击者通过简单地更改 ID 号便能轻松请求数百万条记录。根据 2023 OWASP API 安全十大漏洞，用户身份验证漏洞是第二大 API 漏洞。

（三） 某跨国运营商的 API 数据泄露导致超 3000 万客户账户受损

某跨国运营商 2022 年 12 月透露发现了一起数据泄露事件，并且影响了超 3000 万个付费和预付费账户，预计将因此损失大量资金。攻击者通过暴露的未经授权的 API 窃取了超 3000 万个付费和预付费客户帐户的个人信息。受影响的 API 泄漏了客户帐户数据，包括客户姓名、账单地址、电子邮箱、电话号码、出生日期、该平台账户号和账户订阅条目数与套餐功能等信息

该跨国运营商已通知美国相应的联邦当局，并正在与他们合作调查此次违规事件。在此次黑客攻击事件中被泄露敏感信息的客户现在已收到运营商的通知。

（四） API 安全漏洞影响多家汽车制造商

2022 年安全研究人员发现，黑客可以利用新的 API 安全漏洞远程控制、跟踪和转移车辆，此外他们还可能危及数百万汽车制造商和经销商的帐户，获得对内部系统的管理访问权限，并访问客户和员工信息，泄露来自十多家知名汽车制造商的个人信息。

其中某国际知名豪车品牌在 CMS 中存在 SSO(单点登录)漏洞，暴露了后端 API，攻击者将有机会从 Java script 中提取凭据，这意味

着允许攻击者创建、修改、删除任意的该汽车品牌客户账户，甚至可以将自己设定为车主从而接管任何该车企个人账户。

附录 2　API 安全最佳实践

（一）金融行业

1. 背景与挑战

某证券公司在数字化转型期间，内部建设的数字化应用系统的安全设计不够细致，容易遭受撞库、扫号等攻击，导致账户信息泄露。另外，攻击者利用未授权，敏感数据未脱敏等缺陷，爬取大量的用户资料，交易信息等数据，造成严重的数据泄露。企业需要全面梳理和掌控 API 资产，了解敏感数据的流动情况，及时发现和修复缺陷、漏洞，防止系统被攻击，同时确保符合数据安全合规要求。

2. 实践

通过借助 API 安全管控平台的能力，该证券公司构建了 API 安全审计体系，一方面通过接入的流量识别线上 API 存在的缺陷并及时联动业务进行闭环修复，尤其是外部开发团队的系统，未授权类缺陷过多，这类缺陷很容易被攻击者利用，造成严重的数据泄露。期间共检测出 20 类缺陷，累计协助修复 300 余个 API 缺陷。其中涉及有接口存在未授权访问，关键涉敏数据未脱敏漏洞，该接口可以不经授权，直接通过用户 ID 获取到用户的保单信息，其中包括了姓名、生日、手机号、账号、银行卡号、家庭地址等大量未脱敏的敏感信息。

另一方面也结合了平台内置的威胁情报特征库与异常行为模型，发现外部的恶意攻击并及时部署 API 风控策略、处置恶意请求。共监测到风险事件 23 起，涉及大量敏感信息爬取、撞库攻击等风险，存

在严重的数据泄漏风险。比如攻击者利用多个恶意 IP 代理平台爬取了大量客户的账号信息，包括姓名、手机号、风险投资等级等。同时发现攻击者利用某数据中心 IP 发起了超 8 万次撞库攻击，尝试获取账户登录信息。同时平台也将该部分攻击情报同步到 WAF 设备进行阻断，避免了数据泄露事态进一步扩大导致的用户权益受损与监管责罚风险。

（二） 互联网行业

1. 背景与挑战

某互联网公司是内容社区领域中的 top 企业，在激烈的竞争环境中通过敏捷开发保持较高的产品更新频率以获得更多的市场，但也导致存在大量老旧业务 API 未做下线处理，长期处在安全人员的的视野范围之外，安全策略和监控措施跟不上。此外，黑产攻击时会利用大量的、低成本的动态代理 IP 伪装成正常的请求流量，绕过传统 WAF 和 API 网关的策略对敏感数据进行低频、慢速地爬取，企业往往很难发觉。

2. 实践

通过部署 API 安全管控平台，该互联公司建设了 API 资产管理与风险发现于一体的治理流程。基于平台 API 自动化规约的图模型技术，梳理 APP、小程序、网页中面向用户、内部员工、合作伙伴及第三方供应商等开放场景中的 API，建立完整可视化的 API 资产清单，共梳理 API 资产一万五千余个，其中涉敏 API 超四千个，包含了大

量多版本 API 与僵尸 API。

同时基于平台内置的威胁情报特征库与异常行为模型，发现针对企业业务的攻击风险，共识别 API 风险 4 类 81 个事件，包含利用秒拨代理 IP 进行数据爬取、撞库攻击和营销欺诈等。其中涉及攻击者利用 4.9 万个代理秒拨对其中 2 个站点，超 10 个服务接口发起 90 万次爬虫攻击，主要集中在 PC 端站点，爬取用户、内容、关注好友等页面的信息，从而针对风险及时采取限制措施，避免进一步数据泄露。另一起事件是平台发现移动端营销活动的签到接口遭遇批量攻击，攻击者使用虚假手机号与代理 IP 对签到接口发起超 7 万次请求，通过签到手段获取商家的活动积分以兑换现金或商品福利。API 安全管控平台及时对攻击事件进行告警，进一步关联出恶意用户，并及时联系业务运营方取消了未被兑换的营销奖励，挽回了 80%的损失。

（三） 传统数字化

某连锁酒店集团公司，近些年为支撑迁移至线上的业务快速发展，企业采购或自研大量内部运营系统，这部分系统往往缺乏较严格的访问权限管控与日志留存，甚至不同的内部系统有不同的账户体系，分散式的账户权限管理需要花费较大的成本且容易出现遗漏。因此很难有人能够明确企业内有多少内部系统多少账号，以及账号通过 API 访问了什么数据。同时产业化的黑产团伙上游在匿名平台大肆高价收购数据，驱使内部员工违规窃取数据的事件频发。

通过部署 API 安全管控平台，该连锁酒店集团公司基于 API 流

量分析还原了“员工”和“数据”之间的双向可视化追溯通道。不仅识别出内部系统资产、账号资产，并且对账号的安全状态与行为进行安全审计。过程中共识别超过 200 个内部系统，共梳理访问账号资产 1600 余个，可访问涉敏的账号约 1000 个，超 1500 个内部员工账号、100 个三方调用账号。

通过平台发现在内部的运营系统存在未授权、越权等缺陷，内部人员盗取敏感数据的门槛极低，业务方做了及时修复。并且发现存在弱密码帐号 200 余个，存在着被恶意爆破、盗号的风险，也及时让相关业务通报纠正，并且宣贯密码安全意识。同时发现多名内部员工账号在非工作时间获取涉敏数据过多，比如有 1 个采购部门的账号 12:00 访问客户信息，获取涉敏数据类型包括账号、姓名、手机号、酒店信息，安全人员及时对该账号进行冻结以及联系业务做进一步处理。

联系我们

若您对本自皮书有任何建议，请联系：

marketing@threathunter.cn （深圳永安在线科技有限公司）
welcome@caict.ac.cn（中国信息通信研究院）