金融行业

9 月报
2021 年

安全月报

安全观点 | 行业研究 | 漏洞聚焦 | 安全态势

绿盟科技金融事业部出品

安全观点

数据安全法实施之企业应对指南

行业研究

【云原生应用安全】云原生应用安全风险思考

基金行业信息安全白皮书解读

银行业第三方软件开发工具包(SDK)
安全接入指南规范解读

加密货币交易所 Bilaxy 遭到
黑客攻击，攻击者控制了

联合国遭网络入侵，大量内部
数据或泄露

NSFOCUS

让安全更有效
绿盟科技安全服务
专业|灵活|高效

咨询服务
安全规划
合规咨询
信息安全管理体系咨询
应急体系建设
……

可管理安全服务
远程安全运维
全评估/测试服务
安全基线服务
应急响应
……

安全评价
外部检查辅导
安全指标体系度量
……

安全研究
渗透测试
源代码审计
业务安全测试
漏洞挖掘
……

教育培训
安全技能培训
安全意识教育
……

THE EXPERT
BEHIND GIANTS
巨人背后的专家

多年以来，绿盟科技致力于安全攻防的研究，为运营商、政府、金融、能源、互联网以及教育、医疗等行业用户，提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。在这些巨人的背后，他们是备受信赖的专家。

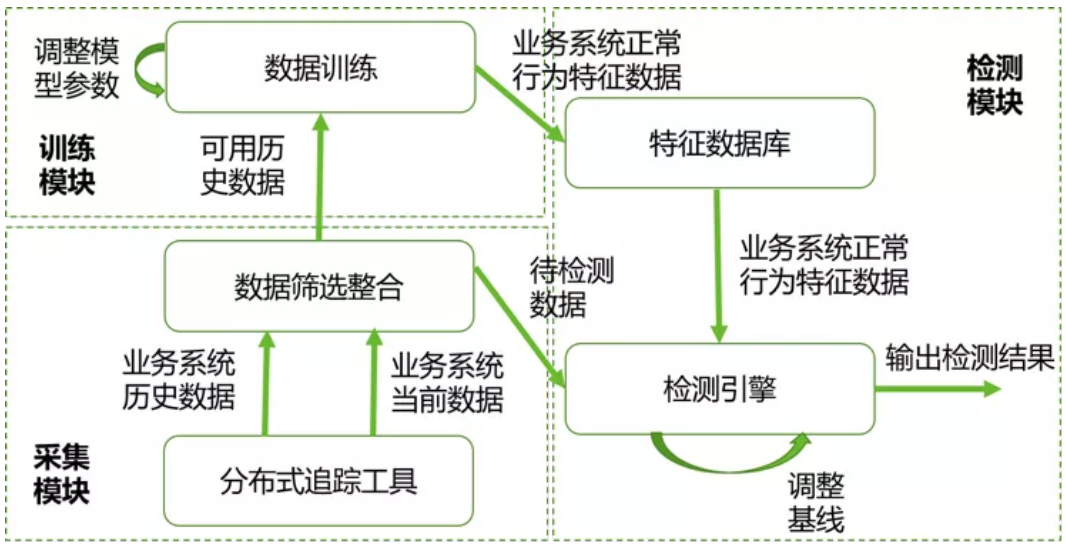
客户支持热线：400-818-6868

NSFOCUS绿盟科技

本|期|看|点

P04 数据安全法实施之企业应对指南

P19 【云原生应用安全】云原生应用安全防护思考（一）



安全月报

2021年第9期

绿盟科技金融事业部

目录 CONTENTS

安全月报在线阅读

绿盟科技官方微信

NSFOCUS

安全观点

数据安全法实施之企业应对指南

绿盟科技 安全服务部

2021年9月1日《中华人民共和国数据安全法》（以下简称“《数据安全法》”）正式施行，标志着我国数据安全建设逐步法制化、规范化、体系化。《数据安全法》中第二十七条、第二十九条、第三十条等明确规定了企业在开展数据处理活动的过程中应满足的数据安全保护义务，本文将以企业视角来讨论如何应对。

一、协同治理，优化企业组织架构

法律依据

第二十七条（第二款） 重要数据的处理者应当明确数据安全负责人和管理机构，落实数据安全保护责任。

企业应对

重要数据的处理者（企业）应当在现有组织架构的基础上增加数据安全、个人信息保护、关键信息基础设施以及重要数据等方面的管理职责，而对于中小企业同样需要明确相应的管理职责，因为“第六章 法律责任”明确提出了对直接负责的主管人员和其他直接责任人员的处罚机制。而企业在实际执行方面，需强化协同治理新理念，在企业现有的网络安全领导小组中进行补充和完善，包括明确直接负责的主管人员和各环节的直接责任人员，同时签署《数据安全保密协议》、《安全责任承诺书》等，以规范企业数据安全组织建设，为数据安全建设落地实施提供坚实的后盾。

二、树立旗帜，完善数据安全制度

法律依据

第二十七条 开展数据处理活动应当依照法律、法规的规定，建立健全全流程数据安全管理制度，组织开展数据安全教育培训，采取相应的技术措施和其他必要措施，保障数据安全。利用互联网等信息网络开展数据处理活动，应当在网络安全等级保护制度的基础上，履行上述数据安全保护义务。

企业应对

从《数据安全法》“第三章 数据安全制度”来看，企业主要涉及数据分类分级、数据安全风险评估与监测、数据安全应急处置、数据安全审查、重要数据管理等方面的管理制度，同时借鉴行业实践，建议企业也需考虑企业数据访问权限管理、数据全生命周期管理、数据合作方管理、数据安全投诉举报、数据安全教育培训等管理制度。建立制度相关要求可参照《信息安全技术 数据安全能力成熟度模型》(GB/T 37988-2019)中“制度流程”能力维度的执行要求，需要从企业整体进行全面考虑和分层设计，并强调层与层之间、同一层不同模块之间的关联逻辑，在内容上不能重复或矛盾，进而形成数据安全制度体系框架。通过制度体系分层思路，企业在管理制度建设过程中需结合实际情况，一般可按照四个层级的制度文件来建设，从而保障制度有效落地执行。

三、梳理现状，开展企业合规分析

法律依据

第二十九条 开展数据处理活动应当加强风险监测，发现数据安全缺陷、漏洞等风险时，应当立即采取补救措施；发生数据安全事件时，应当立即采取处置措施，按照规定及时告知用户并向有关主管部门报告。

第三十条 重要数据的处理者应当按照规定对其数据处理活动定期开展风险评估，并向有关主管部门报送风险评估报告。风险评估报告应当包括处理的重要数据的种类、数量，开展数据处理活动的情况，面临的数据安全风险及其应对措施等。

企业应对

为满足以上数据安全风险监测、评估等保护义务，企业应当开展数据安全现状分析、数据安全评估、数据安全风险监测等方面的工作。

（1）数据安全现状分析包括对数据资产的梳理、数据的分类分级（如涉及重要数据的，则需重点关注）、数据应用场景梳理、数据流转梳理、数据接口梳理（如涉及共享接口的，则需重点关注）、敏感操作场景梳理等多个方面。

（2）而数据安全评估则包括了数据安全合规性评估、数据安全风险评估、个人信息安全影响评估、APP隐私保护评估等，这几类评估中，企业需要理清评估目标、评估方式、评估对象、关注重点等方面的区别，在实际的评估工作中可结合不同类型的评估方式开展。如：合规性评估关注对管理动作的有无以及合理性，需对企业整体合规情况进行评估分析；风险评估关注数据活动的开展环境，需对数据资产和数据应用场景所处环境的风险情况进行综合分析；而影响评估和APP隐私保护评估则是关注数据活动对用户权益的侵害，需着重围绕个人信息主体权益（如是否违反了“告知—同意”的个人信息处理规则）进行综合分析。

（3）数据安全风险监测，则在网络安全的基础上，强调对数据层面的风险监测，如涉及数据重大操作的（如数据批量复制、传输、处理、开放共享和销毁等）、未经客户授权查询的、异常时间/IP登录的、异常帐号增加和权限变更的、异常增删改查的。建议企业制定数据安全风险监测实施细则，完善风险监测流程和处置措施等。

另外，关键信息基础设施的运营者或企业涉及（重要）数据出境的，则需按照有关管理规定进行报备审批和安全评估等。从事数据交易中介服务的机构则需要履行审查义务并做好记录存档；而在数据交易过程中还需界定好数据权属问题，即“数据到底属于谁”，建议在数据交易之前做好安全评估工作。

四、查缺补漏，夯实技术支撑能力

法律依据

第二十七条 开展数据处理活动应当依照法律、法规的规定，建立健全全流程数据安全管理制度，组织开展数据安全教育培训，采取相应的技术措施和其他必要措施，保障数据安全。利用互联网等信息网络开展数据处理活动，应当在网络安全等级保护制度的基础上，履行上述数据安全保护义务。

第十六条 国家支持数据开发利用和数据安全技术研究，鼓励数据开发利用和数据安全等领域的技术推广和商业创新，培育、发展数据开发利用和数据安全产品、产业体系。

企业应对

在企业应用实践中，目前大多以传统的数据安全处理技术为主，包括数据加密（采用国密或者安全的国际算法）、数据脱敏、数据防泄漏、数据水印、UEBA等，这些措施在企业一部分的主要场景中一般是可以应对合规要求的。而在一些内部环境（比如大部分内部用户可以访问和下载）或外部共享环境中，通过传统的数据安全技术处理后的数据仍然面临多种多样的隐私攻击，包括背景知识攻击、差分攻击和重标识攻击等，在这些攻击下敏感信息仍然可能会被泄露。因此，在国家支持数据开发利用和数据安全技术研究的背景下，企业需加大对数据安全新型技术的研发投入，包括差分隐私、知识图谱、流程自动化等，才能更好保护企业数据资产安全和用户合法权益。详细的可参阅绿盟科技在2020年12月发布的《拥抱合规、超越合规：数据安全前沿技术研究报告》。

五、持续改进，强化安全运营能力

法律依据

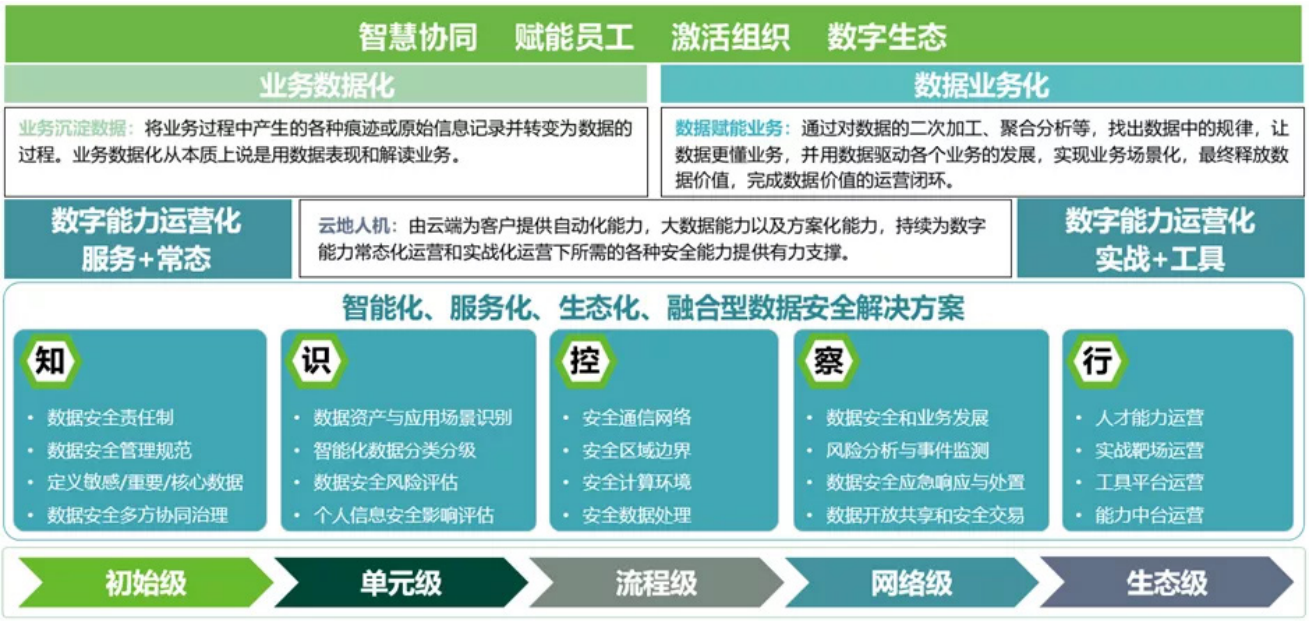
第十三条 国家统筹发展和安全，坚持以数据开发利用和产业发展促进数据安全，以数据安全保障数据开发利用和产业发展。

第十四条 国家实施大数据战略，推进数据基础设施建设，鼓励和支持数据在各行业、各领域的创新应用。

企业应对

单从“第十三条、第十四条”字面上去理解，和“安全运营”没有多大关联，但细细一想，企业保障数据安全与发展之间的平衡，缺少不了安全运营能力。所以，从这个角度去理解，常态化的安全运营能力是多么的重要。那么，企业如何才能在现有能力的基础上再度凸显安全运营的能力呢？2021年4月，绿盟科技正式发布智慧安全3.0理念体系，该理念提出以体系化建设为指引，构建“全场景、可信任、实战化”的安全运营能力，达到“全面防护，智能分析，自动响应”的防护效果。安全运营归根结底还是要靠人，本文认为，企业应首先重视人

才培养，建设数据安全的人员能力2.0（主要包括数据安全管理能力、数据安全运营能力、数据安全技术能力和数据安全合规能力四个维度）的梯队，并同时结合智能化的技术工具，在全场景、实战化中得以运用，才能真正让安全运营的效果发挥最大。



本文通过以上五个步骤讨论了数据安全实施之企业应该如何应对。借此，在数据化转型的大浪潮中，本文提出数据安全落地实践新框架，以“智慧协同、赋能员工、激活组织、数字生态”为远景，为企业在数据化转型过程中如何做好数据安全能力建设提供一个新的角度。

NSFOCUS

行业研究

【云原生应用安全】云原生应用安全风险思考

绿盟科技 星云实验室

一、概述

随着云计算技术的不断发展，当前绝大多数企业正在数字化转型的道路上砥砺前行，其中企业上云是必经之路，在相应实践过程中，传统应用存在升级缓慢、架构臃肿、无法弹性扩展及快速迭代等问题，于是近年来云原生 的概念应运而生，凭借着云原生弹性、敏捷、资源池和服务化等特性，解决了业务在开发、集成、分发和运行等整个生命周期中遇到的问题。

云原生环境中，应用由传统的单体架构转向微服务架构，云计算模式也相应的从基础设施即服务（Infrastructure as a Service，IaaS）转向为容器即服务（Container as a Service，CaaS）和函数即服务（Function as a Service，FaaS）。应用架构和云计算模式的变革是否会导致进一步的风险，这些风险较之传统应用风险又有哪些区别。在讲述云原生应用具体风险前，笔者首先提出以下三个观点，这些观点有助于各位读者较好的理解本文所讲述的内容。

观点一 云原生应用继承了传统应用的风险和API的风险云原生应用源于传统应用，因而云原生应用风险也就继承了传统应用的风险。此外，由于云原生应用架构的变化进而导致应用API交互的增多，可以说云原生应用中大部分交互模式已从Web请求/响应转向各类API请求/响应 ，例如RESTful/HTTP、gRPC等，因而API风险也进一步提升。

观点二 应用架构变革将会带来新的风险由于应用架构变革，云原生应用遵循面向微服务化的设计方式，从而导致功能组件化、服务数量激增、配置复杂等问题，进而为云原生应用和业务带来了新的风险。

观点三 计算模式变革将会带来新的风险随着云计算的不断发展，企业在应用的微服务化后，会进一步聚焦于业务自身，并将功能函数化，因而出现了无服务器计算（Serverless Computing）这类新的云计算模式，进而引入了Serverless应用和Serverless平台的新风险。综上，我们可以看出云原生应用带来的风险是不容小觑的，本文笔者将从传统应用风险、应用架构变革带来的新风险、云计算模式变革带来的新风险三个维度为各位读者分别进行介绍，希望可以引发大家更多的思考。

二、传统应用面临的风险

由于云原生应用也是应用，因而云原生应用风险可以参考传统应用风险，传统应用风险则以Web应用风险为主，主要包含注入、敏感数据泄露、跨站脚本、使用含有已知漏洞的组件、不足的日志记录和监控等风险。

此外，云原生环境中，应用的API交互模式逐渐由“人机交互”转变为

“机机交互”，虽然API大量出现是云原生环境的一大特点，但本质上来说，API风险并无新的变化，因而其风险可以参考现有的API风险，主要包含安全性错误配置、注入、资产管理不当、资源缺失和速率限制等风险。

有关传统应用风险和API风险的更多细节可以分别参考OWASP组织在2017和2019年发布的应用十大风险报告[1]和API十大风险报告[2]。

三、应用架构变革带来的新风险

3.1 云原生应用带来的新风险

云原生应用面临的新风险主要“新”在哪里，笔者看来“新”主要体现在新应用架构的出现，我们知道，新应用架构遵循微服务化的设计模式，通过应用的微服务化，我们能够构建容错性好、易于管理的松耦合系统，与此同时，新应用架构的出现也会引入新的风险，为了较为完整地对风险进行分析，本文我们将以信息系统安全等级三要素，即机密性（Confidentiality）、完整性（Integrity）、可用性（Availability）作为导向为各位读者介绍应用架构变化带来的新风险。

机密性受损的风险

典型的如信息泄露风险，攻击者可通过利用资产脆弱性和嗅探、暴力破解等攻击方式窃取用户隐私数据，从而造成信息泄露风险。

完整性受损的风险

典型的如未授权访问风险，攻击者可通过利用资产脆弱性和中间人攻击等行为绕过系统的认证授权机制，执行越权操作，从而造成未授权访问的风险。

可用性受损的风险

典型的如系统被拒绝服务的风险，一方面，攻击者可通过畸形报文、SYN泛洪等攻击方式为目标系统提供非正常服务，另一方面，系统供不应求的场景也会导致系统遭受拒绝服务风险。

本小节接下来的内容，将以“信息泄露”、“未授权访问”、“拒绝服务”为例，分别介绍上述三类风险。

3.1.1 数据泄露的风险

云原生环境中，虽然造成应用数据泄露风险的原因有很多，但都离不开以下几个因素：

应用漏洞：通过资产漏洞对应用数据进行窃取。

密钥不规范管理：通过不规范的密钥管理对应用数据进行窃取。

应用间通信未经加密：通过应用间通信未经加密的缺陷对传输中数据进行窃取，进而升级到对应用数据的窃取。

3.1.1.1 应用漏洞带来的风险

我们知道，应用中存储的数据多是基于API进行访问，若应用中某API含有未授权访问漏洞，例如Redis未授权访问漏洞，攻击者便可利用此漏洞绕过Redis认证机制，访问到内部数据，进而导致了敏感信息泄露的风险。

传统单体应用架构下，由于API访问范围为用户到应用，攻击者只能看到外部进入至应用的流量，无法看到应用内部的流量，所以针对恶意使用API漏洞进行数据窃取造成的损失范围通常是有限的。

反观微服务化应用架构，当单体应用被拆分为若干个服务后，这些服务会根据业务情况进行相互访问，API访问范围变为服务到服务（Service toService），若某服务因API漏洞导致攻击者有利可图，那么攻击者将会看到应用内部的流量，这无疑为攻击者提供了更多的攻击渠道，因而针对数据泄露的风险程度而言，微服务架构相比传统单体应用架构带来的风险更大。此外，随着服务数量达到一定规模，API数量将不断递增，进而扩大了攻击面，增大了数据泄露的风险。

3.1.1.2 密钥不规范管理带来的风险

在应用的开发过程中，开发者常疏于对密钥的管理从而导致数据泄露的风险，例如开发者将密钥信息、数据库连接密码等敏感信息硬编码在应用程序中，从而增大了诸如应用程序日志泄露、应用程序访问访问密钥泄露的风险。

传统单体应用架构中，开发者常将配置连同应用一起打包，当需要修改配置时，只需登录至服务端进行相应修改，再对应用进行重启便可实现，这种单个集中式配置文件的存储方式从密钥管理风险的角度上讲是相对可控的。

微服务应用架构中，应用的配置数量与服务数量的逐渐增多是成正比的，例如微服务应用中会存在各种服务、各种数据库访问、各种环境变量的配置，且各配置支持动态调整。同时，微服务应用架构对服务的配置管理也提出了更高的要求，例如代码与配置可分离、配置支持分布式、配置更新的实时性、配置可统一进行治理等，因而微服务下的配置管理更加复杂，对运维人员的要求更高，密钥管理的难度也在不断提升，最终会造成更大的数据泄露风险。

3.1.1.3 应用通信未加密带来的风险

如我们所知，如果应用采用HTTP协议进行数据传输，那么HTTP页面的所有

信息将都以纯文本形式进行传输，默认是不提供任何加密措施的，因而在数据传输过程中易被攻击者监听、截获和篡改，典型的攻击流程为攻击者通过Fiddler、Wireshark等抓包工具进行流量监听，之后截获传输的敏感信息，例如数据库密码，登录密码等，最后攻击者根据自身意图对敏感数据进行篡改并发送至服务端，进而导致数据泄露的风险。

传统单体应用架构中，由于网络拓扑相对简单，且应用通信多基于HTTP/HTTPS，因而造成的数据泄露风险多是因为采用了HTTP协议。微服务应用架构中，网络拓扑相对复杂，因遵循分布式的特点，应用间的通信不仅采用HTTP/HTTPS协议，还采用gRPC等协议，由于gRPC协议默认不加密，因而将会导致攻击面的增多，为数据泄露带来了更多的风险。

3.1.2 未授权访问的风险

云原生环境中，应用未授权访问的风险多是由于应用自身漏洞或访问权限错误的配置导致。

3.1.2.1 应用漏洞带来的风险

应用漏洞是造成未授权访问的一大因素，如我们所知，未授权访问漏洞非常之多，较为常用的如Redis、MongoDB、Jenkins、Docker、Zookeeper、Hadoop等应用都曾曝光过相关漏洞，例如Docker曝出的Docker RemoteAPI (Docker Remote API是一种RESTful API，它替代了Docker的远程命令行（rcli），可远程对Docker容器进行操作。) 未授权访问漏洞，攻击者可通过Docker Client或HTTP请求直接访问Docker Remote API，进而对容器进行新建、删除、暂停等危险操作，甚至是获取宿主机shell权限。再如MongoDB未授权访问漏洞，该漏洞造成的根本原因在于MongoDB在启动时将认证信息默认设置为空口令，从而导致登录用户可通过默认端口无须密码对数据库进行任意操作并且可以远程访问数据库。

从漏洞成因的出发点来看，认证及授权机制的薄弱是其主要原因，在单体应用架构下，应用作为一个整体对用户进行认证授权，且应用的访问来源相对单一，基本为浏览器，因而风险是相对可控的，微服务应用架构下，其包含的所有服务均需对各自的访问进行授权，从而明确当前用户的访问控制权限，此外，服务的访问来源除了用户外还包含内部的其他服务，因而在微服务架构下，应用的认证授权机制更为复杂，为云原生应用带来了更多的攻击面。

3.1.2.2 访问权限错误配置带来的风险

由于运维人员对用户的访问权限进行了错误配置，进而会增大被攻击者利用的风险。例如，运维人员对Web应用访问权限进行相应配置，针对普通用户，运维人员应只赋予其只读操作，若运维人员进行了错误的配置，例如为普通用户配置了写操作，那么攻击者便会利用此缺陷绕过认证访问机制对应用发起未授权访问攻击。传统应用架构中，应用由于设计相对单一的特点，其访问权限也相对单一，几乎只涉及用户对应用的访问权限这一层面，因此对应的访问权限配置也相对简单，诚然，也因访问权限配置简单的特点，用户身份凭据等所有敏感信息常存储在应用的服务端，一旦攻击者利用配置的缺陷对应用发起未授权访问入侵，就有可能拿到所有保存在后端的数据，从而造成巨大风险。

微服务应用架构下，由于访问权限还需涉及服务对服务这一层面，因此将会导致权限映射关系变得更加复杂，相应的权限配置难度也在同步增加，例如一个复杂应用被拆分为100个服务，运维人员需要精密地对每个服务赋予其应有的权限，如果因疏忽导致为某个服务配置了错误的权限，攻击者就有可能利用此缺陷对服务展开攻击，若该服务中包含漏洞，进而可能会导致单一漏洞扩展至整个应用的风险。所以如何对云原生应用的访问权限进行高效率管理成为了一个较难的问题，这也是导致其风险的关键因素。

3.1.3 被拒绝服务的风险

被拒绝服务是应用程序的面临的常见风险，笔者看来，造成拒绝服务的主要原因包含两方面，一方面是由于应用自身漏洞所致，例如ReDoS漏洞、Nginx拒绝服务漏洞等。另一方面是由于访问需求与资源能力不匹配所致，例如某电商平台的购买API由于处理请求能力有限，因而无法面对突如其来的大量购买请求，导致了平台资源（CPU、内存、网络）的耗尽甚至崩溃，这种场景往往不带有恶意企图，而带有恶意企图的则主要以ACK、SYNC泛洪攻击及CC（Challenge Collapsar）等攻击为主，其最终目的也是应用资源的耗尽。

3.1.3.1 应用漏洞带来的风险

应用漏洞可以导致应用被拒绝服务，那么具体是如何导致的呢？以ReDoS（Regular expression Denial ofService）漏洞为例，ReDoS为正则表达式拒绝服务，攻击者对该漏洞的利用通常是这样的一个场景，应用程序为用户提供了正则表达式的输入类型又没有对具体的输入进行有效验证，那么攻击者便可通过构造解析效率极低的正则表达式作为输入进而在短时间内引发100%的CPU占用率，最终导致资源耗尽，甚至应用程序崩溃的风险。

3.1.3.2 访问需求与资源能力不匹配带来的风险

此处笔者以CC攻击举例，其攻击原理通常是攻击者通过控制僵尸网络、肉鸡或代理服务器不断地向目标主机发送大量合法请求，从而使正常用户的请求处理变得异常缓慢。

传统Web场景中，攻击者利用代理服务器向受害者发起大量HTTPGET请求，该请求主要通过动态页面向数据库发送访问操作，通过大量的连接，数据库负载极高，超过其正常处理能力，从而无法响应正常请求，并最终导致服务器宕机。

在微服务应用架构下，由于API数量会随着服务数量的递增而递增，因而可能将会导致单一请求生成数以万计的复杂中间层和后端服务调用，进而更容易引起被拒绝服务的风险，例如若微服务应用的API设计未考虑太多因单个API调用引起的耗时问题，那么当外部访问量突增时，将会导致访问需求与资源能力不匹配的问题，使服务端无法对请求作出及时的响应，造成页面卡死的现象，进而会引起系统崩溃的风险。

3.2 云原生业务带来的新风险

在之前的概述小节中，笔者提到应用架构的变革也会为云原生应用业务带来新的风险，说到此处，读者们可能会产生疑问，云原生应用业务风险和上一小节提到的云原生应用风险有何区别，笔者看来，云原生应用风险主要是Web应用风险，即网络层面的风险，而云原生应用业务风险无明显的网络攻击特征，多是利用业务系统的漏洞或规则对业务系统进行攻击来牟利，从而造成一定的损失。

此外，与传统应用架构中的业务风险不同，微服务应用架构中，若服务间的安全措施不完善，例如用户授权不恰当、请求来源校验不严格等，将会导致针对微服务业务层面的攻击变得更加容易，例如针对一个电商应用，攻击者可以对特定的服务进行攻击，例如通过API传入非法数据，或者直接修改服务的数据库系统等。攻击者可以绕过验证码服务，直接调用订单管理服务来进行薅羊毛等恶意操作。攻击者甚至可以通过直接修改订单管理和支付所对应的服务系统，绕过支付的步骤，直接成功购买商品等。

综上，笔者认为，应用微服务化的设计模式带来的业务风险可包含两方面，一方面是未授权访问风险，典型场景为攻击者通过权限绕过对业务系统的关键参数进行修改从而造成业务损失，另一方面则是API滥用的风险，典型的是对业务系统的薅羊毛操作。

3.2.1 未授权访问的风险

云原生业务环境中，笔者针对造成未授权访问风险的原因进行了分析，可以大致分为业务参数异常和业务逻辑异常两方面，为了更为清晰的说明上述异常如何导致未授权访问的风险，笔者以一个微服务架构的电商系统举例说明。如图1所示：

图1 某电商系统流程图

3.2.1.1 业务参数异常带来的风险

API调用过程中往往会传递相关的参数。参数的取值根据业务场景的不同会有不同的取值范围。例如商品数量必须为非负整数，价格必须大于0等。若API对相应参数的监测机制不完善，那么攻击者便可通过输入异常参数导致业务系统受到损失。例如在图1所示的电商系统中，若商品价格只在商品介绍服务中进行校验，而未在订单管理和支付服务中进行校验，那么攻击者则可以通过直接调用订单管理和支付服务的API将订单价格修改为0元或者负值，从而给业务系统造成损失。

3.2.1.2 业务逻辑异常带来的风险

相比于前一类异常，此类异常一般较为隐蔽。攻击者采用某些方法使API调用的逻辑顺序出现异常，包括关键调用步骤缺失、颠倒等。例如在图1所示的电商系统中，攻击者可以利用漏洞绕过支付的步骤直接提交订单。这样就会出现业务逻辑关键步骤缺失的情况，进而会为业务系统带来损失，例如验证码绕过异常就属于业务逻辑异常的一种。

3.2.2 API 滥用的风险

针对此类风险，通常指的是攻击者对业务系统的薅羊毛操作，风险成因则是由于业务频率异常所致，这里笔者依然以电商系统举例说明。

业务频率异常主要指针对一个或一组API的频繁调用，如我们所知，业务系统往往通过图形验证码的方式来避免机器人刷单的操作。例如在图1所示的电商

系统中，攻击者可以绕过验证码所对应的服务，直接对订单进行操作，进而实现机器刷单，对电商进行薅羊毛。

四、云计算模式变革带来的新风险

作为一种新的云计算模式，Serverless具备许多特性，典型的主要有输入源的不确定性、服务器托管云服务商、供应商锁定等，这些特性可能会给Serverless带来新的风险。

此外，由于Serverless最终呈现的还是多个函数组成的应用，且被Serverless提供的服务端运行，因此Serverless风险还应包括Serverless应用的风险及Serverless平台的风险。

最后Serverless因购买、部署成本低、函数访问域名相对可信等将会使Serverless面临被滥用的风险。

本节笔者将针对以上提到的风险进行一一分析。

4.1 Serverless特征带来的风险

4.1.1 输入源不稳定带来的风险

在讲述输入源的不确定性可以带来什么风险前，可能会有读者想了解为什么输入源的不确定性会带来风险，我们知道，Serverless函数是由一系列事件触发的，如云存储事件（S3,Blobs和其他云存储）、流数据处理（如：AWS Kinesis）、通知（如：SMS、电子邮件、IoT）等，鉴于此特性，我们不应该把来自API调用的输入作为唯一攻击面。此外，我们不再控制源到资源间的这条线，如果函数被邮件或数据库触发，将无处可设置防火墙或任何其他控制措施来验证事件源[4]。可见输入源的不确定性将可能导致一定的风险。

在传统应用程序开发中，开发者根据自身实践经验，在数量有限的可能性中可判定出恶意输入来源，但Serverless模式下函数调用是由事件源触发，输入来源的不确定性限制了开发者的判定。例如当函数订阅一个事件源后，该函数在该类型的事件发生时被触发，这些事件可能来源于FaaS平台，也可能来源于未知的事件源，对于来源未知的事件源可以被标注为不受信任。在实际应用场景中，如果开发者没有良好的习惯对事件源进行分类，则会经常导致将不受信任的事件错认为是FaaS平台事件，进而将其视为受信任的输入来处理，最终带来了风险。

具体地，输入来源的不确定性会为Serverless应用带来注入的风险，与传统应用相同的是，注入攻击过程与并无太大区别，不同的是攻击向量得变化，传统应用中用于注入攻击的向量通常指攻击者可以控制或操纵应用输入的任何位置，但Serverless应用由于输入得不确定性因而带来了更大的攻击面。

4.1.2 服务托管云服务厂商带来的风险

传统应用中，例如Web应用常部署在本地/远程服务器上，关于服务端的操作系统漏洞修补、网络拓扑的安全、应用在服务端的访问日志及监控等均需要特定的运维人员去处理，而Serverless的服务器托管云服务商的特点将导致开发者无法感知到服务器的存在，实际上开发者也无须对服务器进行操作，只需关注应用本身的安全即可，服务器的安全则交由云厂商管理，所以在我们也可以认为Serverless的这一特征实际上降低了安全风险。

4.1.3 供应商锁定带来的风险

“供应商锁定”是指用户依赖特定供应商提供的产品及服务，并且在不产生实质性转换成本或运营影响的情况下无法使用其他供应商的云服务，在Serverless中，“供应商锁定”是目前存在的一大问

题，例如用户选择AWS作为应用的运行环境，由于一些原因，该应用需迁移至MicrosoftAzure平台，但“供应商锁定”的问题导致无法轻易得将之前运行的应用及使用的相应资源如S3存储桶等平滑迁移至Microsoft Azure平台中，进而导致企业面临应用转换成本的风险。

4.2 Serverless应用风险

Serverless应用属于云原生应用，其应用本身与传统应用基本是相同的，唯一区别是应用代码编写需要参照云厂商提供的特有代码模版，而传统应用通常没有这个限制。

Serverless应用属于云原生应用，云原生应用又源于传统应用，因而传统应用面临的风险几乎可以全面覆盖Serverless应用风险，关于风险分析部分我们可以参考之前传统应用风险的内容，更详细的内容可以参考OWASP组织在2017年发布的Serverless应用十大风险报告[4]。

4.3 Serverless平台风险

Serverless平台主要指FaaS平台，目前主流的FaaS平台分为两种类型，一种是面向公有云提供商的FaaS平台，常见的有AWSLambda、Microsoft Azure Functions、GoogleCloud Functions等，另一方面则是面向私有云的FaaS平台，此类以开源项目居多，且均支持在Kubernetes上进行部署，常见的有Apache OpenWhisk[7]、Kubeless[8]、OpenFaaS[9]、Fission[10]等。类似在IaaS平台上运行虚机、PaaS平台上运行操作系统和应用，FaaS平台较之上述平台的主要区别为其运行的是一个个Serverless函数。FaaS平台自身负责云环境地安全管理，主要包括数据、存储、网络、计算、操作系统等。如IaaS平台，PaaS平台一样，FaaS平台也面临未授权访问和数据泄露的风险。例如AWS Lambda平台由于自身函数运行时的脆弱性将会导致攻击者轻易拿到运行时shell，结合脆弱的访问权限错误配置可以最终达到攻击目的，这一部分的详细内容可以参考笔者之前在绿盟研究通讯公众号发表的《【云原生攻防研究 】针对AWS Lambda的运行时攻击》文章。

此外，与其他云计算模式不同的是，Serverless为FaaS平台引入了新的攻击源，例如针对FaaS平台账户的拒绝钱包服务攻击，因而Serverless将面临FaaS平台账户的风险，针对此特定类型的攻击解析可参考笔者之前在绿盟研究通讯公众号发表的《Serverless安全研究 — Serverless安全风险》文章。

4.4 Serverless被滥用的风险

Serverless被滥用指具体是指攻击者通过恶意构建Serverless函数并利用其充当整个攻击中的一环，这种方式可在一定程度上规避安全设备的检测，导致Serverless被滥用的原因个人认为主要包括以下几点：

1. 云厂商提供 Serverless 函数的免费试用

近些年，各大云厂商为了用户体验，均对用户提供免费的Serverless套餐，包括每月免费的函数调用额度，这种方式虽然吸引了更多的用户去使用Serverless函数, 但也使得攻击者的攻击成本大幅降低。

2. 用户部署 Serverless 函数的成本低

由于Serverless服务端托管云厂商的机制，故用户只需实现函数的核心逻辑，而无须关心函数是如何被部署及执行的，利用这些特点，攻击者可以编写对其有利的Serverless函数并能省去部署的成本。

3. Serverless 函数访问域名可信

当用户部署完Serverless函数后，需要通过触发器去触发函数的执行，通常用户使用云厂商提供的API网关作为触发器，创建API网关触发器之后，云厂商会为用户提供一个公网的域名，用于访问用户编写的Serverless函数。需要注意的是，该公网域名通常是云厂商域名相关的子域名，因而是相对可信的，鉴于此，攻击者可以利用函数访问域名的可信去隐藏其攻击资产，躲避安全设备的检测。

五、总结

本文较为详细的为各位读者分析了云原生应用面临的风险，可以看出，云原生应用相比传统应用面临的风险主要为应用架构变革及新的云计算模式带来的风险，而针对应用本身的风险并无较大变化，因而对云原生应用架构和无服务器计算模式的深度理解将会有助于理解整个云原生应用安全。

参考文献

[1] https://owasp.org/www-project-top-ten/

[2] https://owasp.org/www-project-api-security/

[3] https://netflixtechblog.com/starting-the-avalanche-640e69b14a06

[4] https://www.owasp.org/index.php/OWASP_Serverless_Top_10_Project

[5]【云原生攻防研究 】针对AWS Lambda的运行时攻击 https://mp.weixin.qq.com/s/duF1Z0EDC3n_G378Aq_XYA

[6]《Serverless安全研究 — Serverless安全风险》https://mp.weixin.qq.com/s/rbS0_42RBiFu8UFFQW4kew

[7] https://github.com/apache/openwhisk

[8] https://github.com/kubeless/kubeless

[9] https://github.com/openfaas/faas

[10] https://github.com/fission/fission

【云原生应用安全】云原生应用安全防护思考（一）

绿盟科技 星云实验室

一、概述

应用是云原生体系中最贴近用户和业务价值的部分，笔者在之前《云原生应用安全风险思考》一文中分析了云原生应用面临的风险，相信各位读者已经有所了解，本文为云原生应用安全防护系列的第一篇，主要针对传统应用安全、API安全、云原生应用业务安全这三方面风险提出笔者的一些防护见解及思考。另外，文章篇幅较长，且内容上与前述风险篇相互对应，若结合在一起阅读，思路会更清晰些，希望本文可为各位读者带来更多思考。

二、传统应用安全防护

从《云原生应用安全风险思考》一文中对传统应用风险的介绍，我们得知传统应用为云原生应用奠定了基石，因而笔者认为云原生应用安全防护也可参照传统应用安全防护，接下来笔者将为各位读者介绍传统应用的安全防护方法，笔者认为其主要包含以下四方面。

应用程序代码漏洞缓解

如《云原生应用安全风险思考》一文中对传统应用安全的分析，应用程序的已知漏洞几乎是造成所有风险的主要原因，因而针对应用程序的漏洞缓解措施是必要的。

应用程序依赖库漏洞防护

应用程序的漏洞缓解措施只能一定程度上规避开发者不规范编码造成的风险，而应用程序本身除了开发者编写的代码，还可能需要引入第三方依赖库，那么依赖库是否含有已知漏洞将会直接决定该应用程序是否相对安全，因而针对应用程序依赖库漏洞的防护也是必要的。

应用程序访问控制

笔者在之前的风险篇中多次提到“访问权限的错误配置”，“脆弱的函数运行时”等会导致应用存在未授权访问风险，做好应用程序的访问控制非常重要。

应用程序数据安全防护

我们知道，应用程序最终为业务服务，而数据为业务产生了价值，从《云原生应用安全风险思考》一文的分析中我们得知数据泄露风险是目前应用程序面临的巨大风险之一，如何防止数据泄露是我们需要关心的一大问题。

2.1 应用程序代码漏洞缓解

应用程序代码漏洞缓解应当从两方面考虑，一方面是安全编码，另一方面是使用代码审计工具。

2.1.1 安全编码

针对安全编码，开发者需要具备安全编码的能力。例如面对SQL注

入漏洞，开发者需要将数据和命令语句及查询语句分离，那么最佳的选择便是使用相对安全的API，而避免使用解释器，提供参数化界面的接口及迁移至ORM或实体框架。此外，对参数输入的有效过滤，例如白名单机制，也有助于防御注入攻击。再如针对XSS类型的漏洞，主要的防护原则为将不可信的输入源与动态的浏览器内容分离，具体实现的手段也非常多，例如使用从设计上就会将危险输入进行编码或转义以防止XSS攻击的Web框架，例如Ruby on Rails或ReactJS等。由于漏洞类型较多，本文由于篇幅限制，不再赘述，更多的针对代码漏洞的防护方法可以参考OWASP组织在2017年发布的应用十大风险报告[1]。

2.1.2 使用代码审计工具

应用程序代码在未部署至服务器前是静态的，我们可以通过手动编写规则脚本去进行漏洞筛查，但往往效率较低，可行的方法是使用自动化代码审计工具，业界比较主流的有AppScan、Fortify、Burp等。需要注意的是这些工具也不是万能的，可能会产生误报或漏报的现象。

2.2 应用程序依赖库漏洞防护

针对应用程序依赖库漏洞造成的风险，我们可以使用受信任的源或软件组成分析技术进行防护。

2.2.1 使用受信任的源

使用受信任的源是最直接的方法，应用开发者可以仅从官方渠道获取第三方组件，同时也可以关注已含有CVE、NVD漏洞的第三方组件，避免试错过程，这些含有漏洞可在官方网站上进行查询，例如Node.js库CVE漏洞列表[2]、Java库CVE漏洞列表[3]、Python库CVE漏洞列表[4]。

2.2.2 使用软件组成分析工具

如果应用程序较为复杂涉及的组件较多，仅通过手动移除含有漏洞的第三方组件往往效率较低，且容易造成漏洞遗漏，鉴于此，业界通常采取软件组成分析(Software Component Analysis SCA)技术，其原理是通过对现有应用程序中使用的开源依赖项进行统计，并同时分析依赖项间的关系最后得出依赖项的开源许可证及其详细信息，详细信息具体包括依赖项是否存在安全漏洞、漏洞数量、漏洞严重程度等。最终SCA工具会根据这些前提条件判定应用程序是否可以继续运行。目前主流的SCA产品有OWASPDependency Check[5]、SonaType[6]、Snyk[7]、Bundler Audit[8]，其中SonaType、Snyk、Bunder Audit均为开源项目。

2.3 应用程序访问控制

实现访问控制，但随着业务量逐渐复杂，用户数量不断增多，准确识别每个用户需要哪些权限、不需要哪些权限是一件具有挑战性的工作，且为每个用户赋予单一权限的方法易造成权限泛滥的风险，因而我们应遵循最小特权原则，即给予每个用户必不可少的特权，从而可以保证所有的用户都能在所赋予的特权之下完成应有的操作，同时也可以限制每个用户所能进行的操作。

使用基于角色的访问控制是实现最小特权原则的经典解决方案，基于角色的访问控制就是将主体（用户、应用）划分为不同的角色，然后为每个角色赋予权限，例如上述提到的业务量大，用户数多的应用程序中，使用基于角色的访问控制就显得很有效，因为我们可以定义每类角色所具备的访问权限，这样即便有成千上万个用户，我们只需按照用户的类型去划分角色，从而可能只需要有限个数的用户角色即可完成访问控制。

2.4 应用程序数据安全防护

笔者认为应用程序的数据安全防护应当覆盖安全编码、密钥管理、安全协议三方面。安全编码涉及敏感信息编码，密钥管理涉及密钥的存储与更换，安全协议涉及函数间数据的安全传输。

2.4.1 安全编码

在应用的开发过程中，开发者常常为方便调试将一些敏感信息写在日志中，随着业务需求地不断增多，开发者容易忘记将调式信息进行删除，从而引发了敏感信息泄露的风险。更为严重的是这种现象在生产环境中也频频出现，例如python的oauthlib依赖库曾被通用缺陷列表（Common Weakness Enumeration CWE）指出含有脆弱性风险[9]，原因是其日志文件中写入了敏感信息，以下为此依赖库对应含有风险的代码：

```
if not request.grant_type== 'password':
raise errors.UnsupportedGrantTypeError(request=request)
log.debug('Validating username %s and password %s.', request.username,request.

if not self.request_validator.validate_user(request.username,request.password,
raise errors.InvalidGrantError('Invalid credentials given.',request=request)
```

以上可以看出开发者将用户名密码记录在了Debug日志中，这是非常危险的写法，因为攻击者可能会利用此缺陷获取用户的登录方式，并进行未授权访问，甚至窃取用户隐私数据，因而针对应用程序的数据安全，安全编码十分重要。

安全编码具体需要怎么做是读者们关心的问题，笔者认为，最重要的是禁止将敏感信息（如：用户名密码、数据库连接方式）存储至源码、日志及易被攻击者发现的地方，同时我们应对存储的所有敏感数据进行加密。

此外，一些开源项目可以帮助开发者避免敏感信息被硬编码至源码中，例如AWS的开源项目git-secrets[10]和Yelp的开源项目detect-secrets[11]，各位读者可以参考。

2.4.2 使用密钥管理系统

为了应用程序环境的安全，我们应当使用密钥管理机制，该机制主要用于对密钥进行创建、分配、更换、删除等操作，目前许多企业采用密钥管理系统（Key Management System）的方式，例如国外主要以AWSKMS[12]、Azure Key Vault[13]、Google CKM（Cloud Key Managemet）[14]等为主，国内则以阿里云密钥管理服务[15]、腾讯云密钥管理服务[16]等为主。

2.4.3 使用安全协议

为避免敏感数据在传输过程中泄露，应确保传输中的数据是加密的，例如Web应用中，我们可以通过使用HTTPS协议替代HTTP协议，确保用户传输的数据不被窃取和篡改，从而在一定程度上避免被中间人攻击的风险。

三、API安全

云原生应用面临的新风险主要“新”在哪里，笔者看来“新”主要体现在新应用架构的出现，我们知道，新应用架构遵循微服务化的设计模式，通过应用的微服务化，我们能够构建容错性好、易于管理的松耦合系统，与此同时，新应用架构的出现也会引入新的风险，为了较为完整地对风险进行分析，本文我们将以信息系统安全等级三要素，即机密性（Confidentiality）、完整性（Integrity）、可用性（Availability）作为导向为各位读者介绍应用架构变化带来的新风险。

通过《云原生应用安全风险思考》一文中API风险分析，我们知道，虽然云原生应用架构的变化导致了API数量的不断增多，但在造成的API风险上并无太大区别，因而在相应的API防护上我们可以参考传统的API防护方法。此外，我们还可采用API脆弱性检测的方式防止更多由于不安全的配置或API漏洞造成的种种风险。最后，在云原生应用架构下，我们可使用云原生API网关，其与传统的API网关有何不同，能为云原生应用风险带来哪些新的防护是我们关心的问题。因此，本小节笔者将API安全分为传统API防护、API脆弱性检测、云原生API网关三个部分进行介绍。

3.1 传统API防护

针对传统的API风险，我们可以使用传统的API防护方式，例如针对失效的认证，我们可以采取多因素认证[17]的方式或采用账号锁定、验证码机制来防止攻击者对特定用户的暴力破解，再如针对失效的功能授权，我们应当默认拒绝所有访问，并显式授予特定角色访问某一功能，更多典型的API防护方式各位读者可以参考OWASP组织在2019年发布的API十大风险报告[18]，该报告针对每种典型风险均提出了较为详细的防护方法，本文限于篇幅，不再赘述。

3.2 API脆弱性检测

API脆弱性主要针对的是服务端可能含有的代码漏洞、错误配置、供应链漏洞等，目前较为可行的方式是使用扫描器对服务端进行周期性的漏洞扫描，国内各大安全厂商均提供扫描器产品，例如绿盟科技的远程安全评估系统（RSAS）[19]和Web应用漏洞扫描系统（WVSS）[20]，其中RSAS已支持针对容器镜像的扫描。同时，我们也可以使用其它商业版扫描器，例如AWVS（Acunetix Web Vulnerability Scanner）、AppScan、Burp Suite、Nessus等。

3.3 云原生API网关

云原生API网关，顾名思义指云原生应用环境下的API网关，笔者认为，云原生API网关与传统API网关的区别主要有两方面，一方面是应用架构带来的区别，另一方面是部署模式的区别。

针对应用架构带来的区别，传统API网关更关注于管理API带来的挑战，而云原生API网关由于应用微服务化后，每个服务都可能会由一个小团队独立开发运维，以快速向客户交付相应的功能，因而为了让每个团队能够独立工作，服务应当具备及时发布、更新以及可观测性的特点，鉴于此，云原生API网关更关注于业务层面，例如可通过为终端用户提供静态地址，并动态地将请求路由至相应的服务地址实现服务发布，又如可在终端用户访问服务过程中通过收集关键可观测性指标实现对服务的监控，再如可支持动态地将终端用户的请求路由至服务的不同版本以便进行金丝雀测试。

针对部署模式的区别，传统的API网关通常在虚拟机或Docker容器中进行部署，而云原生API网关则主要在微服务编排平台部署，典型的为Kubernetes。

微服务应用环境中，云原生API网关充当着非常重要的一环，它不仅要负责外部所有的流量接入，同时还要在网关入口处根据不同类型请求提供流量控制、日志收集、性能分析、速率限制、熔断、重试等细粒度控制行为。云原生API网关为云原生应用环境的防护带来了一定优势，首先，由于云原生API网关接管南北向流量，因而将外部访问与微服务进行了一定隔离，从而保障了后台微服务的安全。其次，在早期的微服务治理框架中，例如SpringCloud，由于其将服务治理逻辑嵌入了具体服务代码中，因而导致了应用的复杂性变高，而云原生网关具备一定的服务治理能力，从而可节省后端服务的开发成本，进而有益于应用层面的扩展。最后，云原生API网关也具备解决外界访问带来的一些安全问题，例如TLS加密、数据丢失防护、防止跨域访问、认证授权、访问控制等特性。

云原生API网关以开源项目居多，近些年来，随着技术的不断发展，Kubernetes显然已成为容器编排平台的业界标准，因而云原生API网关也都相应支持在Kubernetes上进行部署，目前主流的云原生API网关有Ambassador、Zuul、Gloo、Kong等。为了让各位读者一览以上提到的云原生API网关在安全功能上的支持，笔者进行了相应统计，以供各位读者参考，具体下表所示：

表1 主流开源云原生API网关安全功能支持

	Ambassador	Zuul	Gloo	Kong
Web 应用防火墙	支持	支持	支持	支持
访问控制	支持	支持	支持	支持
基本认证授权	支持	支持	支持	支持
SSL 证书管理	支持	支持	不支持	支持
数据丢失防护	不支持	支持	支持	不支持
跨域（CORS）	支持	支持	支持	支持
JWT	支持	支持	支持	支持
限速服务	支持	支持	支持	支持

通过上述表格可以看出Zuul全项支持，但因Zuul与Spring Cloud的深度集成，故只能针对使用Java环境的微服务进行防护。其余云原生API网关均有一项不支持，主要为Ambassador针对数据丢失防护不支持，Gloo针对SSL证书管理不支持，Kong也是对数据丢失防护不支持，需要注意的是，这三个API网关相比于Zuul有较为明显的区别，Ambassador与Gloo均为Kubernetes原生网关，且从官

方网站上[21][22]都能看到他们兼容微服务治理框架Istio的方案，因而如果各位读者使用Istio治理微服务，可以选择Ambassador和Gloo。Kong属于这四者中开源社区最为活跃及成熟的，从官方的解决方案中[23]可以看到，其以支持Kubernetes的部署方案，凭借Kong在API安全上的积累，相信很快可以在云原生API网关上占据一席之地，成为大多数人的选择。

四、云原生应用业务安全

云原生应用面临的新风险主要“新”在哪里，笔者看来“新”主要体现在新应用架构的出现，我们知道，新应用架构遵循微服务化的设计模式，通过应用的微服务化，我们能够构建容错性好、易于管理的松耦合系统，与此同时，新应用架构的出现也会引入新的风险，为了较为完整地对风险进行分析，本文我们将以信息系统安全等级三要素，即机密性（Confidentiality）、完整性（Integrity）、可用性（Availability）作为导向为各位读者介绍应用架构变化带来的新风险。

针对《云原生应用安全风险思考》一文中提到的云原生应用业务层面安全问题，基于基线的异常检测是一类比较有效的方法：首先建立正常业务行为与参数的基线，进而找出偏移基线的异常业务操作，其中，基线的建立需要结合业务系统的特性和专家知识共同来完成。

在电商系统中，业务参数基线主要基于专家知识来建立。例如商品价格不仅与商品本身相关，也与时间和各类优惠活动等相关。这类基线需要运维人员持续的维护。对于业务逻辑基线的建立，由于业务系统在正式上线运行以后，其操作逻辑一般不会有较大的变化，同时异常操作所占的比例较少。因此可以采集业务系统历史的操作数据，结合统计分析与机器学习的方法建立业务逻辑的基线。相比于人工方法，这种方法可以提高基线建立的效率，有效减轻运维人员的工作量。

为此，可利用分布式追踪工具对云原生应用中产生的数据进行采集，笔者对当前主流的分布式追踪工具Zipkin，Jaeger，Skywalking，Pinpoint进行了调研，这些分布式追踪工具大体可分为三类，基于SDK的，基于探针的，基于Sidecar的。

基于SDK的分布式追踪工具。以Jaeger为例，Jaeger提供了大量可供追踪使用的API，通过侵入微服务业务的软件系统，在系统源代码中添加追踪模块实现分布式追踪。此类工具可以最大限度地抓取业务系统中的有效数据，提供了足够的可参考指标；但其通用性较差，需要针对每个服务进行重新实现，部署成本较

高，工作量较大。

基于探针的分布式追踪工具。以SkyWalking Java探针为例，在使用SkyWalkingJava探针时，需将探针文件打包到容器镜像中，并在镜像启动程序中添加-javaagent agent.jar命令实现探针的启动，以完成SkyWalking在微服务业务上的部署。SkyWalking的Java探针实现原理为字节码注入，将需要注入的类文件转换成byte数组，通过设置好的拦截器注入到正在运行的程序中。这种探针通过控制JVM中类加载器的行为，侵入运行时环境实现分布式追踪。此类工具无需修改业务系统的源代码，相对SDK有更好的通用性，但其可获取的有效数据相对SDK类工具较少。

基于Sidecar实现。Sidecar作为服务代理，为其所管理的容器实现服务发现，流量管理，负载均衡和路由等功能。在流量管理过程中，Sidecar可以抓取进出容器的网络请求与响应数据，这些数据可以记录该服务所完成的一次单个操作，可与追踪中的跨度信息对应，因此可将sidecar视为一种基于数据收集的分布式追踪工具。Sidecar无需修改业务系统代码，也不会引入额外的系统的开销。但由于sidecar所抓取的跨度不包含追踪链路上下文，要将sidecar所抓取的跨度数据串联成追踪链路是很困难的。

通过使用以上分布式追踪工具进行数据采集后，针对《云原生应用安全风险思考》一文中提出的三种业务异常场景（业务参数异常、业务逻辑异常、业务频率异常），笔者设计并实现了业务异常检测引擎，如下图所示。其中，采集模块主要用于采集业务系统的运行数据，训练模块主要针对业务系统历史数据进行训练以提取行为特征数据，检测模块主要对正在运行的业务系统进行异常检测。

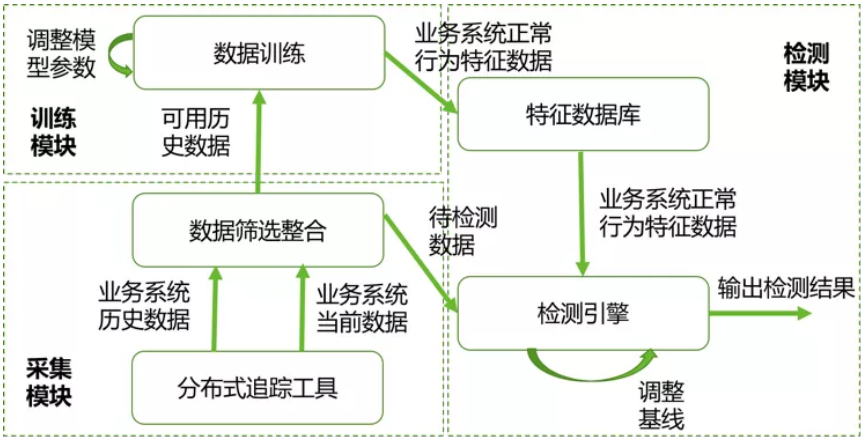


图1 业务异常检测引擎设计图

检测引擎中每部分的具体功能为：

分布式追踪工具。相比Skywalking、Sidecar，Jaeger可获取的数据字段最多，能够检测的异常场景最丰富，然而，Jaeger需要在业务系统的源代码中进行插桩，对开发团队而言有较强的侵入性。相反，Sidecar模式没有代码和镜像的侵入性，但通过反向代理截取流量的模式也决定了它不能获得丰富的上下文，如云原生应用的API调用关系树（TraceID）是无法获得的。如何利用侵入性更低的采集工具收集到的数据来实现覆盖更多场景的异常检测仍需要很多后续工作。

数据筛选与整合模块。此模块主要功能为过滤掉数据集中的脏数据，以及提取出可以表示业务系统行为的数据。在云原生应用中，可以表示业务系统行为的数据为API调用关系树、服务名、操作名、HTTPPOST参数等。

数据训练模块。将预处理后的历史数据利用机器学习或统计学的方法，训练出业务系统中的正常行为，并生成与业务系统正常行为匹配的特征数据。这里进行训练的先验知识为，我们认为业务系统中大量存在的行为是正常行为，而数量很少的行为是异常行为。在训练过程中，需要根据专家知识对训练结果的检验不断调整训练模型的参数。

检测引擎。将业务系统当前数据与特征数据库中数据进行检索匹配，并利用序列相似性计算等方法找出特征数据库中与当前行为最为匹配的特征数据。检测引擎需要将特征数据与当前数据的相似性与基线进行比较，若比较结果显示当前行为与正常行为的差异在基线限制范围内，则为正常行为，若超出基线限制范围，则判定为异常行为。对于基线，首先需要根据专家知识设置合理的初始基线，并根据不同场景，或利用无监督模型自行调整基线，或由运维人员手动维护基线。

五、总结

云原生应用面临的新风险主要“新”在哪里，笔者看来“新”主要体现在新应用架构的出现，我们知道，新应用架构遵循微服务化的设计模式，通过应用的微服务化，我们能够构建容错性好、易于管理的松耦合系统，与此同时，新应用架构的出现也会引入新的风险，为了较为完整地对风险进行分析，本文我们将以信息系统安全等级三要素，即机密性（Confidentiality）、完整性（Integrity）、可用性（Availability）作为导向为各位读者介绍应用架构变化带来的新风险。

本文为各位读者介绍了云原生应用在传统应用安全、API安全、云原生应用业务安全三个维度的相应防护方法，结合之前风险篇的相应介绍，首先我们可以看出传统应用防护技术适用于云原生应用，因而深刻理解传统应用防护内容非常重要。其次，云原生应用架构的变化为API带来了更多特点，也带来了新的防护方法，如云原生API网关的合理使用可以有效改善用户环境下的API安全状况。最后云原生应用业务方面的异常会给相应的业务系统带来巨大的损失。而由于API业务安全与业务场景的强耦合性，需要在系统设计之初就考虑各种业务场景下的API安全问题。一方面加强API的认证授权机制；另一方面要加入必要的数据采集功能，为后续业务异常场景的分析提供支撑。

云原生应用安全防护系列的第二篇，笔者会为各位读者介绍微服务架构下的应用安全及Serverless安全的防护手段，欢迎各位读者持续关注。

参考文献

[1] https://owasp.org/www-project-top-ten/
[2] https://www.npmjs.com/advisories
[3] https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=java
[4] https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=python
[5] https://owasp.org/www-project-dependency-check/
[6] https://www.sonatype.com/
[7] https://snyk.io/
[8] https://github.com/rubysec/bundler-audit
[9] https://cwe.mitre.org/data/definitions/534.html
[10] https://github.com/awslabs/git-secrets
[11] https://github.com/Yelp/detect-secrets
[12] https://aws.amazon.com/cn/kms/
[13] https://azure.microsoft.com/en-us/services/key-vault/
[14] https://cloud.google.com/security-key-management
[15] https://www.aliyun.com/product/kms
[16] https://cloud.tencent.com/product/kms
[17] https://owasp.org/www-project-api-security/
[18] https://zh.wikipedia.org/wiki/%E5%A4%9A%E9%87%8D%E8%A6%81%E7%B4%A0%E9%A9%97%E8%AD%89
[19] https://www.nsfocus.com.cn/html/2019/209_1009/66.html
[20] https://www.nsfocus.com.cn/html/2019/206_0911/8.html
[21] https://www.getambassador.io/user-guide/with-istio/
[22] https://www.solo.io/blog/istio-1-5-api-gateway-with-gloo/
[23] https://konghq.com/solutions/kubernetes-ingress/

【云原生应用安全】云原生应用安全防护思考（二）

绿盟科技 星云实验室

一、概述

本文为云原生应用安全防护系列的第二篇，也是最终篇，本文笔者主要针对微服务架构下的应用安全、Serverless安全提出一些防护见解及思考。文章篇幅较长，内容上与之前笔者发表的若干文章有相互交叉对应的部分，希望能为各位读者带来帮助。

二、微服务架构下的应用安全

针对《云原生应用安全风险思考》一文中对云原生应用的新风险分析，我们可以看出应用的微服务化带来的新风险主要包含数据泄露、未授权访问、被拒绝服务攻击，那么如何进行相应的防护也应从以上三方面去考虑，笔者通过调研和一些实践发现使用传统的防护方法是可行的，但当服务随业务的增多而逐渐增多时，传统的防护方法由于需要开发人员进行大量配置而变得非常复杂，例如用户的应用部署在Kubernetes上，该应用包含上百个服务，当我们做访问控制时可以依托Kubernetes的RBAC机制对目的服务进行授权，进而我们就需要依赖Kubernetes的API以完成配置，每次配置是会耗费一定时间的，因此需要大量服务授权时，开发者往往感到力不从心，为解决诸如以上服务治理带来的难题，我们可以使用微服务治理框架进行相应防护，笔者在2018年发表的《Service Mesh实践之Istio初体验》文章中对什么是Service Mesh及Istio的基本概念进行了相应介绍，从中可以看出Istio目前已成为第二代微服务治理框架的代表，因而笔者认为Istio的安全防护能力也是基本能够覆盖微服务应用安全范畴的。事实上，笔者在2019年发表的《Istio系列一：Istio的认证授权机制分析》一文中也对Istio的安全

机制进行了介绍，不过当时Istio处于较早期版本，虽然与目前最新版本的Istio在功能实现稍有不同，但底层原理几乎未变，可供各位读者参考。

综上，笔者认为面向微服务架构下的应用安全，可以采用传统的防护方式或微服务治理框架进行防护，具体的防护方法可以包含以下几方面。

认证服务

由于攻击者在进行未授权访问前首先需要通过系统的认证，因而确保认证服务的有效性非常重要，尤其在微服务应用架构下，服务的不断增多将会导致其认证过程变得更为复杂。

授权服务

授权服务是针对未授权访问风险最直接的防护手段，微服务应用架构下，由于服务的权限映射相对复杂，因而会导致授权服务变得更难。

数据安全防护

与《云原生应用安全风险思考》一文中分析数据安全防护的必要性一样，但微服务应用架构下，服务间通信不仅使用HTTP协议，还会使用gRPC协议等，这是我们需要注意的地方。

其它防护

除了上述防护方法之外，微服务治理框架与API网关/WAF可以结合以进行深度防护，例如可以在一定程度上缓解微服务环境中被拒绝服务攻击的风险。

2.1 认证服务

微服务架构下，服务可以采用JWT或基于Istio的认证方式，下面笔者将分别进行说明。

2.1.1 基于 JWT(JSON Web Token) 的认证

微服务架构下，每个服务是无状态的，传统的session认证方式由于服务端需要存储客户端的登录状态因此在微服务中不再适用。理想的实现方式应为无状态登录，流程通常如下所示：

1. 客户端请求某服务，服务端对用户进行登录认证；

2. 认证通过，服务端将用户登录信息进行加密并形成令牌，最后再返回至客户端作为登录凭证；

3. 在2步骤之后，客户端每次请求都需携带认证的令牌;

4. 服务端对令牌进行解密，判断是否有效，若有效则认证通过，否则返回失败信息；

为了满足无状态登录，我们可通过JWT实现，JWT是JSON风格轻量级认证和授权规范，也就是上述流程中提到的令牌，主要用于分布式场景，其使用流程如图1所示：

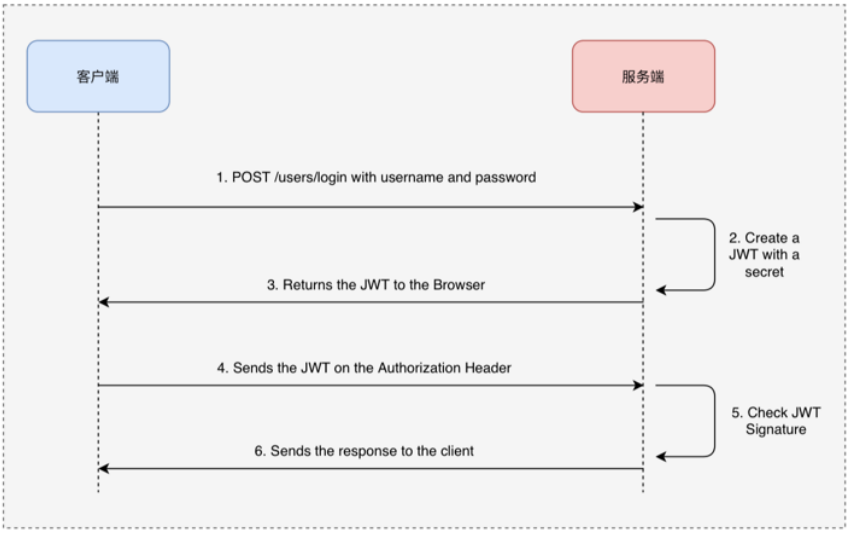


图1. JWT交互流程图

从图1我们可以看出，JWT交互流程与上述提到的理想流程基本上是相似的，需要注意的是，JWT令牌中会包含用户敏感信息，为防止被绕过的可能，JWT令牌采用了签名机制。此外，传输时需要使用加密协议。

2.1.2 基于 Istio 的认证

本节主要为各位读者介绍基于Istio的认证，在具体介绍前，我们首先为各位读者介绍Istio的安全架构，如下图所示：

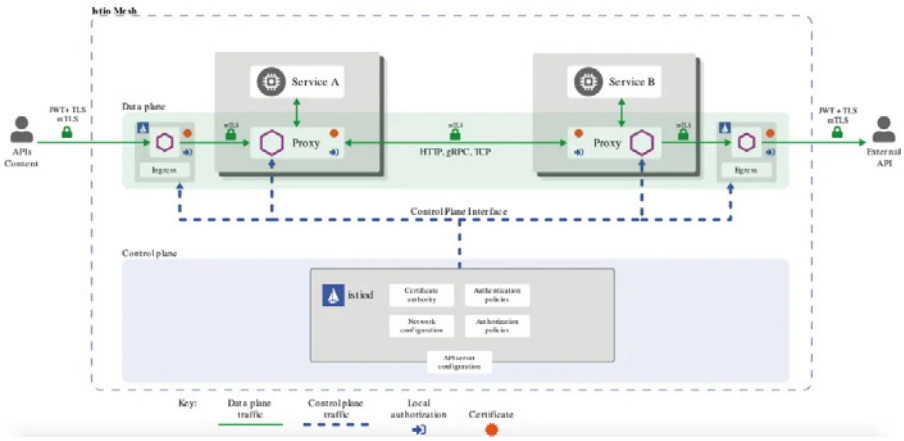


图2. Istio安全架构

图2展示了Istio的认证和授权两部分，Istio的安全机制涉及诸多组件，控制平面由核心组件Istiod提供，其中包含密钥及证书颁发机构（CA）、认证授权策略、网络配置等；数据平面则由Envoy代理、边缘代理（Ingress和Egress）组件构成。

借助控制平面Istiod内置的CA模块，Istio可实现为服务网格中的服务提供认证机制，该认证机制工作流程包含提供服务签名证书，并将证书分发至数据平面各个服务的Envoy代理中，当数据平面服务间建立通信时，服务旁的Envoy代理会拦截请求并采用签名证书和另一端服务的Envoy代理进行双向TLS认证从而建立安全传输通道，保障了数据安全。

下面笔者将为各位读者介绍Istio的两种主要认证类型。

2.1.2.1 传输认证(Peer authentication)

传输认证是Istio的一种认证类型，其主要用于微服务应用架构中服务到服务的认证，从而可验证所连接的客户端。针对此类型的认证，Istio提供了双向TLS的解决方案，该解决方案提供以下功能[1]：

1. 确保服务到服务间的通信安全；

2. 提供密钥管理系统，从而自动进行密钥及证书的生成、分发和轮换；

3. 为每个服务提供一个代表其角色的身份，从而实现跨集群的互操作性。

具体的我们可以通过使用传输认证策略为Istio中的服务指定认证要求，例如命名空间级别TLS认证策略可以指定某命名空间下所有的Pod间的访问均使用TLS加密，Pod级别TLS认证策略可以指定某具体Pod被访问时需要进行TLS加密等，更多关于Istio的双向TLS解决方案内容可以参考官方文档[2]。

2.1.2.2 请求级认证(Request authentication)

请求级认证是Istio的一种认证类型，主要用于对终端用户的认证，与传输认证的主要区别为，请求级认证主要用于验证用户请求服务时携带的凭据，而非服务到服务的认证。

请求级认证主要通过JSON Web Token（JWT）机制实现，实现原理与前面“基于JWT的认证”小节中提到的内容类似，区别为Istio在其基础上进行了一层封装，使用户可以以yaml的方式进行策略配置，用户体验更为友好。

Istio的JWT认证主要依赖于JWKS（JSON Web Key Set），JWKS是一组密钥集合，其中包含用于验证JWT的公钥，在实际应用场景中，运维人员通过为服务部署JWT认证策略实现请求级认证，为方便理解，下面展示了JWT认证策略的核心部分配置：

```
issuer: https://example.com
jwksUri: https://example.com/.well-known/jwks.json
triggerRules:
- excludedPaths:
  - exact: /status/version
  includedPaths:
  - prefix: /status/
```

其中：

issuer：代表发布JWT的发行者；

jwksUri：JWKS获取的地址，用于验证JWT的签名，jwksUri可以为远程服务器地址也可以为本地地址，其通常以域名或URL形式展现；

triggerRules（重要）：triggerRules为使用JWT验证请求的规则触发列表，如果满足匹配规则就进行JWT验证，此参数使得服务间的认证变得弹性化，用户可以按需配置下发规则。上述策略中triggerRules的含义为对于任何带有“/status/”前缀的请求路径，除了/status/version以外，都需要JWT认证。

当JWT认证策略部署完成后，外部对某服务有新的请求时，请求级认证会根据策略内容验证请求携带的令牌（Token），若与策略内容匹配则返回认证失败，反之认证成功。

2.2 授权服务

微服务架构下，授权服务可以通过基于角色的授权以及基于Istio的授权实现，以下笔者将分别进行说明。

2.2.1 基于角色的授权服务

基于角色的授权服务为RBAC（RoleBased Access Control），通过角色关联用户，角色关联权限的方式间接赋予用户权限。在微服务环境中作为访问控制被广泛使用，RBAC可以增加微服务的扩展性，例如微服务场景中，每个服务作为一个实体，若要分配服务相同的权限，使用RBAC时只需设定一种角色，并赋予相应权限，再将此角色与指定的服务实体进行绑定即可。若要

分配服务不同的权限，只需为不同的服务实体分配不同的角色，而无需对服务具体的权限进行修改，通过这种方式不仅可以大幅提升权限调整的效率，还降低了漏调权限的概率。

如果用户选择在Kubernetes中部署微服务应用，则可以直接使用Kubernetes原生的RBAC策略，具体使用方式可参考官方文档[3]。

2.2.2 基于 Istio 的授权服务

有了前述提到的Istio认证机制作为基础，Istio还提供授权机制，其主要用于对服务进行授权。在Istio 1.4版本之前，其授权机制依赖于Kubernetes的RBAC策略，相比Kubernetes的原生RBAC策略，Istio对其进行了进一步的封装，可让用户直接通过Istio的声明式API对具体的服务进行授权，不过Istio为了更好地用户体验，在其1.6版本中引入了AuthorizationPolicyCRD[16]（Custom Resource Definition），相比1.4版本，AuthorizationPolicy CRD带来了更多的优势，一方面该CRD将RBAC的配置变得更为简化为从而大幅提升了用户体验，另一方面该CRD支持了更多的用例，例如对Ingress/Egress的支持，且同时不会增加复杂性。

此外，Istio的授权模式也是基于其提供的授权策略去实现。

图3展示了Istio的授权架构：

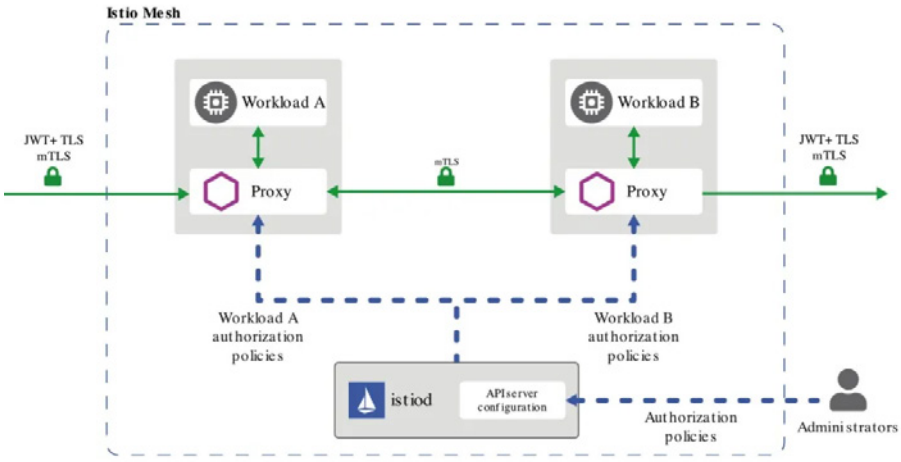


图3 Istio授权架构图

如图3所示，Istio授权流程可以归纳总结为以下内容：

Administrator使用yaml文件指定Istio授权策略并将其部署至Istiod核心组件中，Istiod通过API Server组件监测授权策略变更，若有更改，则获取新的策略，

Istiod将授权策略下发至服务的Sidecar代理，每个Sidecar代理均包含一个授权引擎，在引擎运行时对请求进行授权。

以下是一个简单的Istio授权策略：

```
apiVersion:security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
 name: httpbin
 namespace: foo
spec:
 selector:
  matchLabels:
   app: httpbin
   version: v1
 rules:
 - from:
  - source:
     principals:["cluster.local/ns/default/sa/sleep"]
  to:
  - operation:
     methods: ["GET"]
  when:
  - key: request.headers[version]
   values: ["v1", "v2"]
```

可以看出，以上策略适用于foo命名空间下，且满足标签为app:httpbin和version: v1的目标Pod, 并设置授权规则为当访问源为“cluster.local/ns/default/sa/sleep”服务，且请求头中包含 v1 或 v2 的 version 字段时，才允许访问。默认情况下，任何与策略不匹配的请求都将被拒绝。

2.3 数据安全

如《【云原生应用安全】云原生应用安全防护思考（一）》一文中提到的，传统应用架构中，我们可以通过安全编码、使用密钥管理系统和使用安全协议的方式防止数据泄露，在微服务应用架构中，我们可以考虑使用Kubernetes原生的安全机制或微服务治理框架的安全机制去进行防护。

针对Kubernetes原生的安全机制，例如Secret机制，我们可以使用其进行密钥存储，从而规避了敏感信息硬编码带来的数据泄露风险，更详细的内容可以参考官方文档[4]。

针对微服务治理框架的安全机制，如Istio支持服务间的TLS双向加密、密钥管理及服务间的授权，因而可以有效规避由中间人攻击或未授权访问攻击带来的数据泄露风险。

2.4 其他防护机制

通过以上三小节的介绍，我们能从中看出采用微服务治理框架的防护方式可在一定程度上有效规避云原生应用的新风险，但其防护点主要针对的是微服务架构下应用的东西向流量，针对南北向的流量防护稍显脆弱，由于微服务架构下的应用防护应当是全流量防护，因而针对南北向所存在的问题，我们可以考虑将微服务治理框架与API网关和WAF相结合，从而提升南北向的防护能力。

本节笔者将以微服务治理框架Istio为例，为大家介绍Istio和API网关协同的全面防护以及Istio与WAF结合的深度防护。

2.4.1 Istio 和 API 网关协同的全面防护

针对应用的南北流量而言，Istio采取的解决方案为使用边缘代理Ingress与Egress分别接管用户或外

界服务到服务网格内部的入/出站流量，Ingress与Egress实则为Istio部署的两个Pod，Pod内部为一个Envoy代理，借助Envoy代理的安全Filter机制，在一定程度上可对恶意Web攻击进行相应防护，但现有的Envoy安全Filter种类相对较少，面对复杂变化场景下的Web攻击仍然无法应对，可行的解决方案为在服务网格之外部署一层云原生API网关，具体如图4所示：

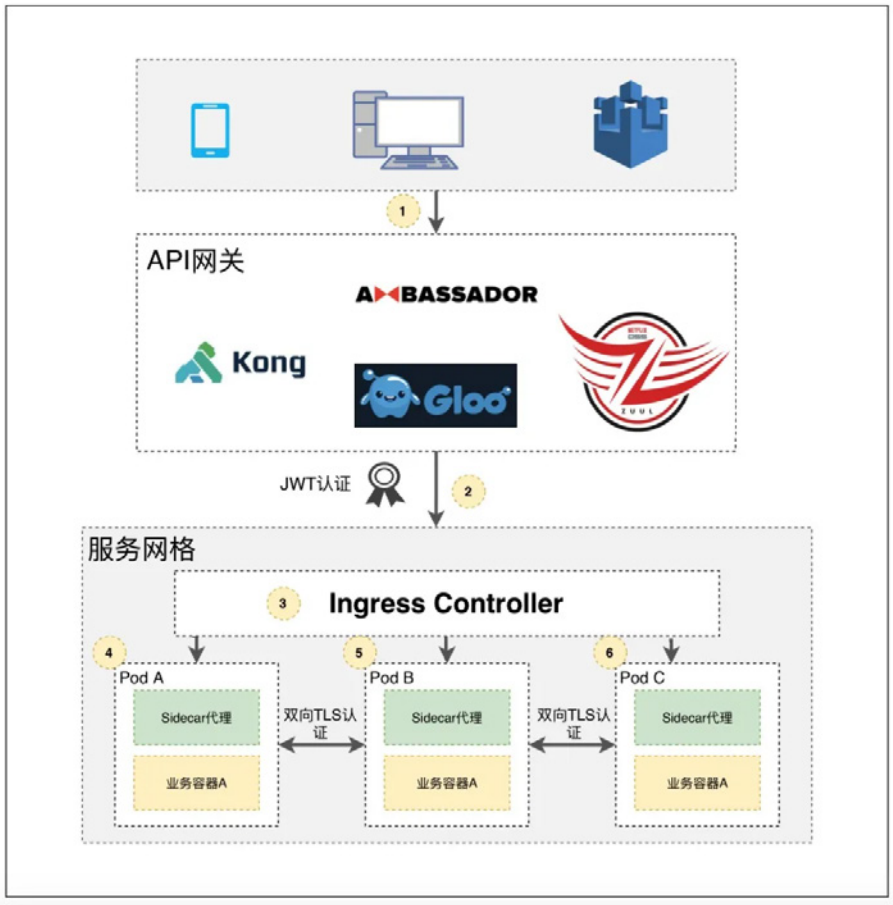


图4 Istio与API网关结合防护图

安全功能上，云原生API网关可提供全方位的安全防护，例如访问控制、认证授权、证书管理、Bot流量检测、数据丢失防护、黑白名单限制等，在这些有效防护基础之上，应用的南北向得到了控制。

此外，该解决方案的好处还在于应用内部的东西流量无需通过外部网关层，这样可以从边缘到端点进行一站式防护。

2.4.1 Istio 和 WAF 结合的深度防护

WAF作为一款抵御常见Web攻击的主流安全产品，可以有效对Web流量进行深度防护，并且随着云原生化概念的普及，国内外安全厂商的容器化WAF产品也在迅速落地，未来容器化WAF与Istio的结合将会在很大程度上提升微服务安全。

根据近期市场调研，Signal Sciences、Fortiweb、Wallarm、Radware这几家公司已有了各自的容器化WAF解决方案。值得注意的是Signal Sciences 公司的解决方案支持WAF服务与Envoy或Istio结合，其设计如图5所示，该方案主要运用了Envoy的Filter机制，通过External Authorization HTTP Filter可以将流经业务容器的东西/南北向流量引流至WAF容器，从而可阻断恶意请求，保护了微服务的安全。

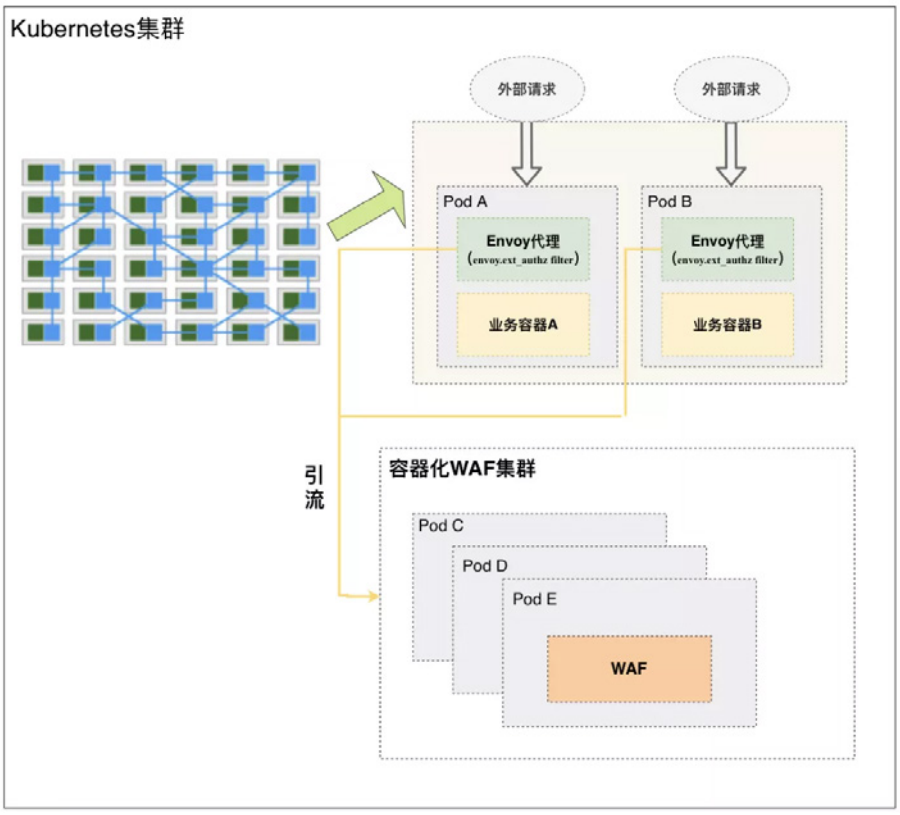


图5 Istio与云原生WAF结合防护图

此方案带来的好处是对业务入侵较小，实现较为容易、且容器化WAF规模不会随用户业务更改而更改。但同时也有一些弊端，比如需要单独部署容器化WAF、Envoy引流模块的性能问题、引流方式对WAF处理的延迟等。

另一种解决方案是Radware提出的Kubernetes WAF方案，该方案基于Istio实现，其中WAF被拆分为Agent程序和后端服务两部分，Agent程序作为Sidecar容器置于Pod的Envoy容器和业务容器间，该Sidecar的主要作用为启动一个反向代理，以便将外部请求流量代理至Pod外部的WAF后端服务中，如图6所示。该套方案带来的好处是无需关心外部请求如何路由至Pod、与Istio结合的理念更接近云原生化、实现了以单个服务为粒度的防护。但同时也存在着一些不足，例如流量到达业务容器前经历了两跳，这在大规模并发场景下可能会影响效率。

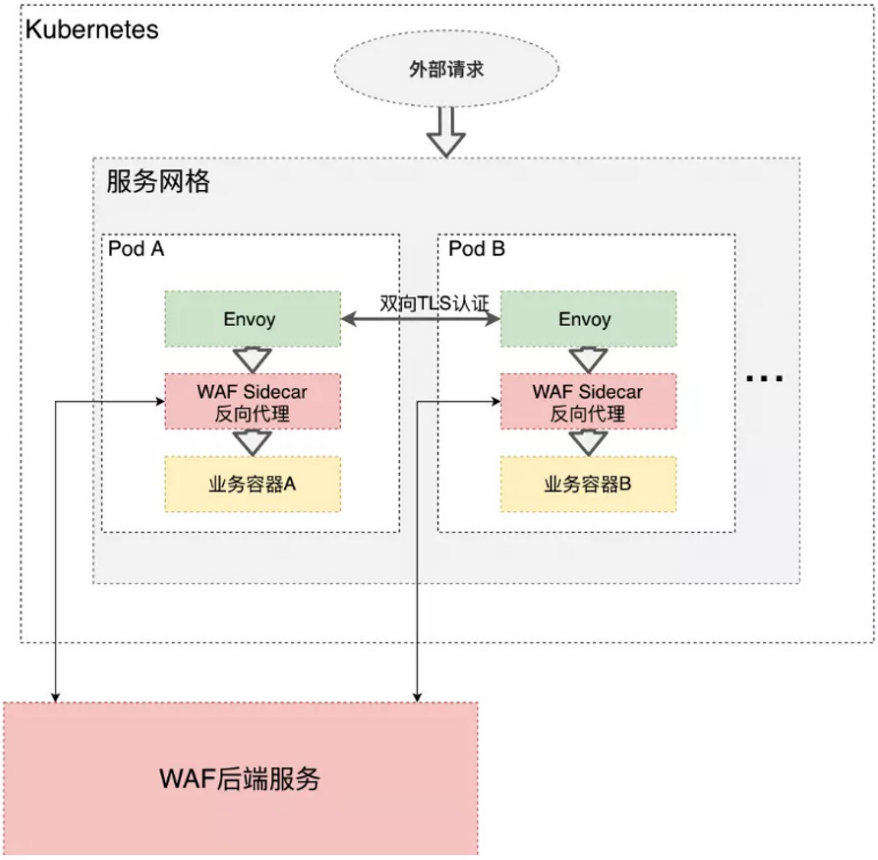


图6 WAF Sidecar化示意图

此外，由于Istio的数据平面为微服务应用安全防护提供了引擎，而数据平面默认采取Envoy作为Sidecar代理，因此Envoy自身的扩展性成为了安全厂商较为关心的问题，近些年Envoy也在不断提升着其适配性，例如Envoy提供Lua过滤器[5]和Wasm过滤器[6]，以便安全厂商将安全能力，例如WAF的能力融入Envoy，从而对微服务应用进行防护。

三、Serverless安全防护

3.1 Serverless应用安全防护

通过《云原生应用安全风险思考》一文中的Serverless风险分析，我们了解到传统应用的风险几乎可以覆盖Serverless应用的风险，因而针对Serverless应用的安全防护各位读者可以大体参考《【云原生应用安全】云原生应用安全防护思考（一）》一文中传统应用安全的防护方式，尤其是应用程序的代码漏洞缓解、依赖库漏洞防护、数据安全防护。

针对应用程序访问控制，除了《【云原生应用安全】云原生应用安全防护思考（一）》中提到的使用基于角色的访问控制之外，由于Serverless云计算模式带来的变化，还需要进行更深层次的防护，笔者认为函数隔离及底层资源隔离是较为合适的防护方法。

函数隔离

函数间进行隔离可有效降低安全风险。一个FaaS应用通常由许多函数以既定的序列和逻辑组成，每个函数可以独立进行扩展、部署等，但也同时可能被攻破，如果安全团队没有对函数进行有效隔离，那么攻击者也可同时访问应用中的其它函数。再如随着应用设计不断变化，这些函数更改了执行序列，从而使攻击者有机可乘并发起业务逻辑攻击，这些是FaaS产生的碎片化问题。正确的做法应当是将每个函数作为边界，使得安全控制粒度细化至函数级别，这对于创建能够长期保持安全的FaaS应用是非常必要的。

为了更好地将函数进行隔离，笔者认为应当从以下几方面进行考虑：

1. 不要过度依赖函数的调用序列，因为随着时间推移调用序列可能会改变；如果序列发生了变化，要进行相应的安全审查；

2. 每个函数都应当将任何事件输入视为不受信任的源，并同时对输入进行安全校验；

3. 开发标准化的通用安全库，并强制每个函数使用；

4. 使用FaaS平台提供的函数隔离机制，例如AWSLambda采用Amazon弹性计算云（ElasticCompute Cloud EC2）模型[7]和安全容器Firecracker模型[8]机制进行隔离。

底层资源隔离

仅仅对函数层面进行访问控制是不够的，例如攻击者仍可以利用函数运行时环境的脆弱性以获取服务端的shell权限从而进行滥用，笔者在之前发表的《【云原生攻防研究】针对AWS Lambda的运行时攻击》有详细的利用过程，可供各位读者参考。

为了预防上述场景的发生，我们应当从底层进行资源隔离，例如可通过KataContainer[9]从上至下进行防护，再如可通过Kubernetes的网络策略(NetworkPolicy)[10]实现由左至右的网络层面隔离。

3.2 Serverless平台安全防护

针对《云原生应用安全风险思考》一文中提出的Serverless平台风险，我们可以考虑通过以下几种防护方式进行相应缓解。

3.2.1 使用云厂商提供的存储最佳实践

为了尽量避免用户在使用云厂商提供的Serverless平台时因不安全的错误配置造成数据泄露的风险，主流云厂商均提供了相应的存储最佳实践供各位开发者参考，例如Howto secure AWS S3 Resources[11]、Azure Storage SecurityGuide[12]、Best Practices for Google Cloud Storage[13]等。

3.2.2 使用云厂商的监控资源

现今各大云厂商均为Serverless配备了相应的监控资源，例如AzureMonitor、AWS CloudWatch、AWSCloudTrail等，使用云这些监控

资源可以识别和报告异常行为，例如未授权访问、过度执行的函数、过长的执行时间等。

3.2.3 使用云厂商的账单告警机制

针对拒绝钱包服务（DoW）攻击，公有云厂商提供了账单告警机制进行缓解[14]，如AWS开发者可通过在Lambda控制台为函数调用频度和单次调用费用设定阈值进行告警；或提供资源限额的配置，主流的云厂商已提供了以下资源选项供开发者配置：

1. 函数执行内存分配；
2. 函数执行所需临时的磁盘容量；
3. 函数执行的进程数和线程数；
4. 函数执行时常；
5. 函数接收载荷大小；
6. 函数并发执行数 ‘

通过上述选项的合理配置可以在一定程度上缓解DoW攻击。

3.3 Serverless被滥用的防护措施

针对《云原生应用安全风险思考》一文提出的Serverless被滥用的风险，我们可以采取以下方式进行防护[15]：

1. 通过IDS等安全设备监测木马在本机的出口流量，诸如“/pixel”、“/utm.gif”、“ga.js”等URL的流量应进行重点监测；

2. 确认自己的资产中是否有云厂商提供的Serverless函数业务，如果没有可以通过浏览器禁用相关云厂商的子域名；

3. 采取断网措施，从根源上直接禁止所有网络访问。

3.4 其他防护措施

3.4.1 Serverless 资产业务梳理

由于云厂商通常缺乏一套自动化机制对现有Serverless应用中包含的函数，数据及可用API进行分类、追踪，评估等操作，因此开发者在不断完善应用的同时，可能疏于对应用数据及API的管理，从而导致攻击者利用敏感数据、不安全的API发起攻击。为了避免这种情况，开发者需要在应用的设计阶段对资产业务进行详细梳理。其中包括但不限于以下几个部分：

1. 确认应用中函数间的逻辑关系；

2. 确认应用的数据类型及数据的敏感性；

3. 评估Serverless数据的价值；

4. 评估可访问数据API的安全；

有了一个较为全面的应用全景图，便可在一定程度上降低应用被攻击的风险。

3.4.2 定期清理非必要的Serverless 实例

由于Serverless应用通常遵循微服务的设计模式，因此一套完整的工作流应由许多函数组成，而开发者可能部署了非常多的Serverless应用，在这些应用中，必定存在一些长时间不被调用的实例，为了避免被攻击者利用，应当定期对Serverless应用进行检测，清理非必要的实例，从而降低安全隐患。

3.4.3 限制函数策略

开发者首先应当限制函数策略，给予其适当的访问权限，删除过于宽松的权限，这样即便攻击者拿到了访问凭证也无法对所有资源进行访问。

四、总结

本文较为系统地对微服务架构下的应用安全及Serverless安全提供了相应的防护思路，其中：

针对《云原生应用安全风险思考》一文提出的云原生应用的新风险，我们可以看出应用架构的变化是带来新风险的主要原因，鉴于此，本文笔者针对具体的风险提出了防护方法，其中，使用微服务治理框架Istio可以在一定程度上缓解应用架构带来的风险，此外，也介绍了Istio与API网关和WAF结合的业界方案，从而可实现微服务应用的全流量防护。

针对《云原生应用安全风险思考》一文提出的Serverless风险，笔者较为系统地从Serverless应用及平台两方面对前述提到的Serverless风险进行了相应防护介绍。可以看出，与传统安全防护不同的是Serverless模式带来的新型云原生下的应用安全场景，因而，我们需要适应云计算模式的不断变化，并不断总结新场景下的防护方法，才能最终将安全落实到底。

参考文献

[1]https://istio.io/latest/docs/concepts/security/#authentication

[2]https://istio.io/latest/docs/concepts/security/#mutual-tls-authentication

[3]https://kubernetes.io/docs/reference/access-authn-authz/rbac/

[4]https://kubernetes.io/docs/concepts/configuration/secret/

[5]https://www.envoyproxy.io/docs/envoy/latest/configuration/http/http_filters/lua_filter

[6]https://www.envoyproxy.io/docs/envoy/latest/configuration/http/http_filters/wasm_filter#config-http-filters-wasm

[7]https://docs.aws.amazon.com/lambda/latest/dg/services-ec2.html

[8]https://firecracker-microvm.github.io/

[9]https://github.com/kata-containers

[10]https://kubernetes.io/docs/concepts/services-networking/network-policies/

[11]https://aws.amazon.com/cn/premiumsupport/knowledge-center/secure-s3-resources/

[12]https://docs.microsoft.com/en-us/azure/storage/blobs/security-recommendations

[13] https://cloud.google.com/storage/docs/best-practices#security

[14]https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/monitor_estimated_charges_with_cloudwatch.html

[15]https://mp.weixin.qq.com/s?__biz=MzA5MDc1NDc1MQ==&mid=2247487065&idx=1&sn=5503dd715b3f2131aacba91d0075be9e&chksm=90079589a7701c9f6b808de703de88dd517f31e054a90af9cd61bd8b9ea4281d27d43b163666&mpshare=1&scene=1&srcid=0422DzTAylEepRGhnBx93iDS&sharer_sharetime=1619074342871&sharer_shareid=0f3da91f6fd8e207294f6347709786ec&version=3.0.40.6184&platform=mac#rd

[16]https://istio.io/latest/docs/reference/config/security/authorization-policy/

基金行业信息安全白皮书解读

绿盟科技 张凯

一、当前基金行业面临的问题

1.1 组织架构尚不完善

仍有部分公司存在对信息安全重视程度不够、 组织体系不完善、负责信息安全的信息技术人员数量不足、较多信息安全人员身兼多职、岗位职责不能相互制衡等问题，甚至有部分基金公司尚未设立专职的信息安全管理人员

1.2 信息安全投入不足

总体来看，目前国内基金公司的信息安全投入和银行证券等金融机构以及国外同行相比尚存在较大差距。从行业内部来看，不同基金公司的信息安全投入水平差异较大。其根本原因在于大部分公司对于整体统一的信息安全规划建设还有欠考虑，导致公司对基础技术架构和管理支持系统的投入资源有限。

1.3 应用安全保障体系不健全

目前，基金公司普遍建立了系统上线和变更评审，对于测试和生产环境进行区分和隔离，并对访问、操作等活动进行记录和审计。尽管如此，基金公司仍需进一步优化应用安全保障体系，如完善对代码检测及代码托管的管理，加强代码安全审计，确保测试环境下关键数据的脱敏， 对互联网和移动应用进行加固保护等。

1.4 数据安全保障体系缺失

基金公司的业务系统每天产生大量的应用数据，其中高价值和敏感数据占据极大比例。而目前基金公司对数据安全管理规范、数据安全操作指南和流程、数据安全管理人员能力等方面仍然存在部分缺失，需进一步完善数据安全保障体系。

1.5 内部流程管控不完善

目前，基金行业也已经在密码强度、特权账号管理、权限分离、防范社工攻击、培养公司全员信息安全防范意识等方面投入大量资源以此来提升行业内信息安全应对综合能力水平。

1.6 信息安全防御体系需进一步提升

有部分基金公司还未能形成完备的信息安全防御体系，如：基础设施建设不健全，已有的信息安全防护体系未形成纵深防御能力，较难应对与时俱进的网络攻击方式和手段；公司运维体制不完善，大部分运维保障仍停留在人工操作阶段，系统自动化运维的安全和风险管理能力不足，导致信息安全事件频发。信息安全防御体系的不足导致了行业内部分基金公

司在信息安全战略规划、安全指标监测、安全事件处置、安全防控策略调优等方面技能有所欠缺，无法应对日益严峻的信息安全形势

1.7 业务连续性建设需持续完善

有部分基金公司的基础设施较为集中化，仅建有同城数据中心。一旦发生区域性风险或灾难，这类基金公司将很难在规定时间内完成核心业务系统的恢复和重建工作。

二、基金行业主要法律法规标准

我国的基金监管机构主要为中国证券监督管理委员会、中国人民银行、证券交易所、证券业协会。基金公司网络安全方面建设需要参考以上机构颁发的各项标准、政策、指引等。

标准编号	实施时间	发布机构	标准名称
【证监会令 [第 82 号]】	2012 年 11 月 1 日	证监会	证券期货业信息安全保障管理办法
【证监会公告 [2011]10 号】（JRT0059-2010）	2011 年 4 月 14 日	证监会	证券期货经营机构信息系统备份能力标准
【证监会公告 [2013]7 号】（JRT0099-2012）	2013 年 1 月 31 日	证监会	证券期货业信息系统运维管理规范
【证监会公告 [2016]2 号】			
(JR/T0133—2015)	2016 年 1 月 13 日	证监会	证券期货业信息系统托管基本要求
中证协发 [2012]228 号	2012 年 12 月 3 日	中国证券业协会	证券公司证券营业部信息技术指引
【证监会公告 [2013]30 号】	2013 年 7 月 18 日	证监会	《关于加强证券期货经营机构客户交易终端信息等客户信息管理的规定》
【证监会公告 [2019]27 号】	2018 年 9 月 14 日	证监会	《关于进一步加强期货经营机构客户交易终端信息采集有关事项的公告》
【证监会公告 [2014]58 号】（JRT0112-2014）	2014 年 12 月 26 日	证监会	证券期货业信息系统审计规范 -

标准编号	实施时间	发布机构	标准名称
【证监会公告 [2014]56 号】（JRT0110-2014）	2014 日 12 月 26 日	证监会	证券公司客户资料管理规范
【证监会公告 [2012]47 号】（JRT0084-2012）	2012 年 12 月 26 日	证监会	证券期货业网络时钟授时规范
【证监会公告 [2012]46 号】	2012 年 12 月 24 日	证监会	证券期货业信息安全事件报告与调查处理办法
【证监会公告 [2018]28 号】JR/T 0158—2018	2018 年 9 月 27 日	证监会	证券期货业数据分类分级指引
证监会令【第 152 号】	2019 年 6 月 1 日	证监会	证券基金经营机构信息技术管理办法
银 发 [2019]237 号（JRT 0092-2019）	2019 年 9 月 27 日	人民银行	《移动金融客户端应用软件安全管理规范》
银发 [2018]343 号	2019 年 1 月 1 日	人民银行、银保监会、证监会	会关于金融行业贯彻推进互联网协议第六版（IPv6）规模部署行动计划的实施意见
JR/T 0171—2020	2020 年 2 月 13 日	人民银行	个人金融信息保护技术规范
JR/T 0197-2020	2020 年 9 月 23 日	人民银行	金融数据安全数据安全分级指南
JR/T 0184—2020	2020 年 2 月 5 日	人民银行	金融分布式账本技术安全规范
JR/T 0223-2021	2021 年 4 月 8 日	人民银行	金融数据安全 数据生命周期安全规范
JR/T 0213-2021	2021 年 2 月 10 日	人民银行	金融网络安全 --Web 应用服务安全测试通用规范
GB/T 22239-2019	2019 年 12 月 1 日	公安部	信息安全技术 网络安全等级保护基本要求
	2017 年 6 月 1 日	全国人民代表大会常务委员会	网络安全法
	2021 年 9 月 1 日	全国人民代表大会常务委员会	数据安全法
	2020 年 1 月 1 日	全国人民代表大会常务委员会	密码法

标准编号	实施时间	发布机构	标准名称
	2021 年 11 月 1 日	全国人民代表大会常务委员会	个人信息保护法
	2021 年 9 月 1 日	国务院	关键信息基础设施安全保护条例

注：加粗表示重点关注的重要标准。

三、基金行业信息安全建设解决方案

根据基金行业当前面临的风险、当前以及未来的信息安全建设方向与我司对应的标准解决方案。

信息安全建设方向	建设内容	解决方案
信息安全治理架构	将信息技术风险纳入公司全面风险管理；建立信息安全三道防线体系、加强三道防线间循环联动机制	信息安全规划咨询服务、信息安全管理体系咨询服务、信息安全等级保护咨询服务
人力与资金保障	与发展战略及规划的匹配性；资源配备的针对性、	信息安全规划咨询服务、信息安全管理体系咨询服务、信息安全等级保护咨询服务
信息安全治理目标	治理及管理目标设计；治理系统组件、	信息安全规划咨询服务、信息安全管理体系咨询服务、信息安全等级保护咨询服务
信息技术服务机构安全管理	信息技术控制范围；信息技术服务机构管理流程	信息安全规划咨询服务、信息安全管理体系咨询服务、信息安全等级保护咨询服务
风险识别与业务连续性管理	加强识别、评估和监测，及时发现信息与网络安全事件；加强业务连续性管理，增强小概率事件的风险防范能力	信息安全规划咨询服务、信息安全管理体系咨询服务、信息安全等级保护咨询服务、业务连续性咨询服务、渗透测试服务、源代码审计服务
安全审计	丰富第三道防线审计方式（加强信息安全常规审计。加强新系统、新业务、新制度的前置化敏捷审计。加强信息安全敏感岗位审计。运用科技手段，开展连续审计。）；	信息安全管理体系咨询服务、信息安全合规咨询服务、渗透测试服务、源代码审计服务、安全漏洞扫描服务、安全配置核查服务、日志分析服务、系统上线前安全评估服务

信息安全建设方向	建设内容	解决方案
企业信息安全风险意识	结合基金公司自身发展情况，适当借助外部资源；将信息科技风险意识纳入公司内部控制文化；	网络安全意识培训服务、钓鱼邮件测试服务
重大活动网络安全保障	国家攻防演习、建党周年庆、建国周年庆、G20 会议、两会。	重要时期安全保障服务、应急响应服务、安全漏洞扫描服务、安全配置核查服务、安全加固服务、信息安全技术策略制定服务、渗透测试服务、网络架构分析服务网站安全监测服务、现场值守服务
远程办公	终端安全管理；确保 WiFi 安全；防止非法登录；警惕钓鱼攻击；数据安全传输；做好数据备份；做好审计留痕	绿盟终端检测与响应系统（EDR）、安全无线防御系统（SWD）、企业版杀毒软件 KSV9、堡垒机 OSMS、SDP 零信任、邮件安全网关 SEG、威胁分析系统（TAC）、数据防泄漏（终端 DLP、网络 DLP）数据灾备一体机、备份软件、日志审计 LAS、数据库审计 DAS
数据安全	数据外部非授权访问及修改；数据内部非授权访问及修改；数据加密；数据非故意丢失；数据交换安全	数据库审计 DAS、SDP 零信任、堡垒机 OSMS、渗透测试服务、数据防泄漏（终端 DLP、网络 DLP）、数据分类分级 IDR、数据库加密系统、安全隔离与信息交换系统 SIES、数据库脱敏 DMS、web 应用防火墙 WAF、入侵防御设备 IPS、综合威胁探针 UTS、数据安全态势感知平台 ISOP-DS、数据安全咨询服务、数据分类分级服务
无线安全	企业内部 WiFi 上网安全；访客无线网络安全；办公网与外部无线网络隔离	安全无线防御系统（SWD）、防火墙 NF
开发安全	漏洞管理流程；网站漏洞；网页防篡改；SQL 注入预防；SQL 注入预防；SDLC（安全开发管控流程）	应用开发管控 SDL 服务、源代码审计服务、渗透测试服务、web 应用防火墙 WAF、网页防篡改 HDS
自动化运维	制度和资源；标准和工具	ISOP 智能安全运营平台（SOAR 模块）、安全管理平台（ESP-H HD5000 型号）、信息安全规划咨询服务

银行业第三方软件开发工具包（SDK）安全接入指南规范解读

绿盟科技 刘红涛 郭逸

一、SDK概述

SDK 就是 Software Development Kit 的缩写，翻译过来——软件开发工具包。一般都是一些软件工程师为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件时的开发工具的集合。开发者使用非自研协助开发的SDK称为第三方SDK。

第三方SDK工具包按照集成、工作方式，可分为以下类型：

a) 无交互类 SDK：SDK 完全嵌入到宿主应用中，SDK 服务的对象为银行应用的用户，SDK 不主动与第三方服务端交互，当 SDK 需要获取信息时，由宿主应用经由银行服务端向第三方服务端发起信息获取请求，第三方服务端反馈无差别的数据，框架见图 1。

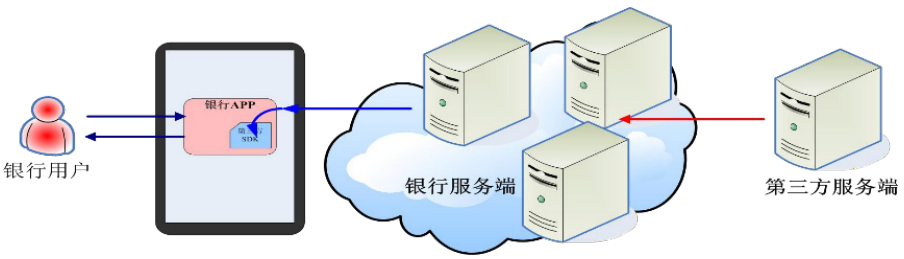


图 1 无交互类 SDK 框架示意图

b) 推送类 SDK：SDK 由嵌入到宿主应用部分和第三方服务端部分组成，SDK 服务的对象为宿主应用（银行应用）的用户，SDK 需要主动与第三方服务端进行通信，SDK 仅向第三方服务端发送简单参数信息（不包括能够定位到个体的个人金融信息或支付敏感信息），第三方服务端根据参数向 SDK 推送数据，框架见图 2。

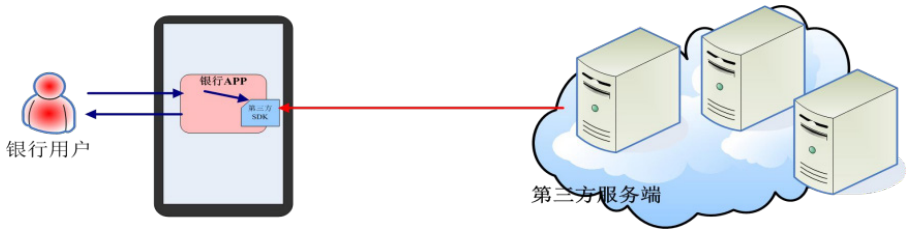


图 2 推送类 SDK 框架示意

c) 交互服务类 SDK：SDK 由嵌入到宿主应用部分和后台服务器部分组成，SDK 服务对象为宿主应用（银行应用）的用户和第三方应用的用户，工具包需要主动与后台服务器进行通信，工具包 将用户输入信息反馈至后台服务器，通过交互实现功能，框架见图 3。

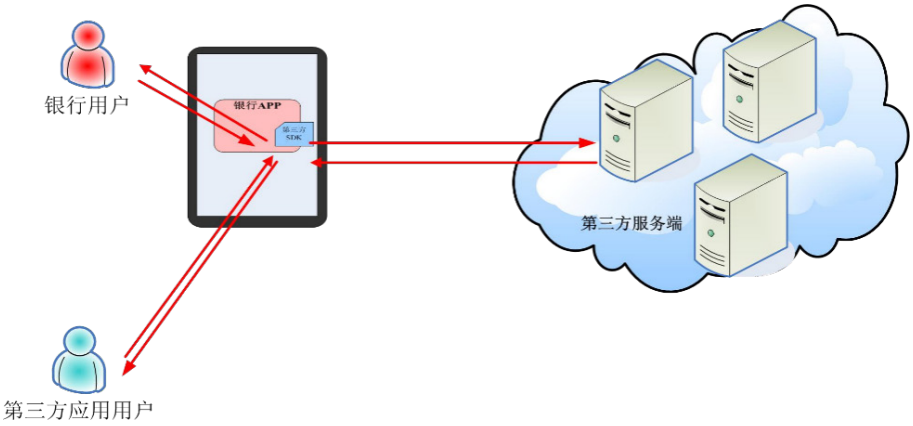


图 3 交互服务类 SDK 框架示意

二、第三方SDK风险

集成在App里的第三方工具包，它们可以帮助App低成本地实现各种功能，如地图，支付，统计，社交，广告等。开发者引入第三方SDK，可降低自己的开发难度。但SDK自身也具备获取设备信息和用户个人信息的能力。由第三方SDK引入也可能存在安全问题。有些非法SDK能潜藏在手机App中，窃取用户隐私数据回传至自己后台，甚至包括验证码等关键信息。

在互联网金融发展的浪潮下，国内商业银行都加快了创新步伐，在现有应

用中加入合作方 SDK 的开发模式已经成为国内商业银行应用创新的重要手段。以银行业为例，近年来开始尝试在手机银行等移动应用中引入互联网公司开发的 SDK，如在手机银行App 中引入指纹识别 SDK，实现指纹登录、指纹支付等功能。合作方技术能力的引入推动银行移动应用架构向多源化、灵活化发展，给银行应用创新注入了活力。

但近年来不断发生的安全事件也为商业银行与合作方的合作敲响警钟。

2018 年 4 月，国内某广告联盟的广告组件被植入木马，导致使用该广告组件应用的用户信息被泄露，支付宝、京东等多个平台超过 20 万客户信息被泄露；2018 年 7 月，微信支付 SDK 被报存在 XXE 漏洞，黑客可利用该漏洞攻击嵌入微信支付 SDK 的商户服务器，获取其安全密钥等关键数据，进而实现 0 元支付购买商品。2020年7月16日，央视315晚会今晚举行，曝光了一些手机应用中存在第三方SDK插件，窃取用户信息的情况。

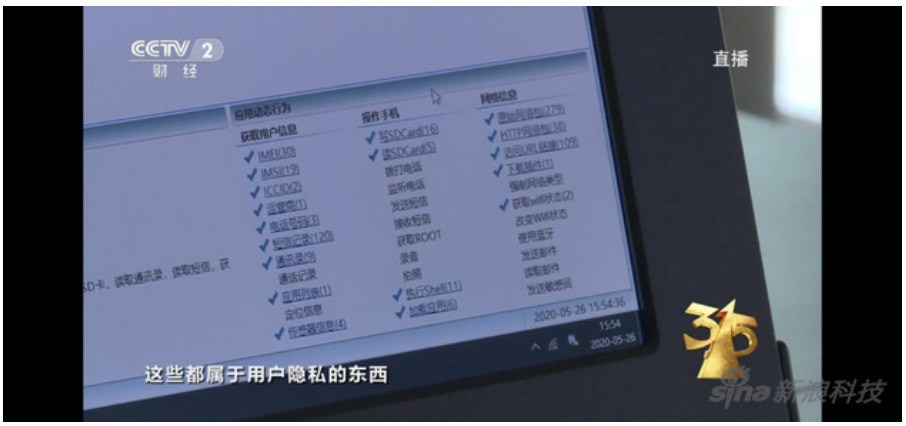


图4 央视315晚会报导相关应用窃取用户信息

典型案例：2015年8月起，欧某等20余人开始研发“广告SDK”，装有“广告SDK”的手机在用户首次开机联网时，即通过互联网与后台服务器连接，在用户不知情的情况下上传用户信息、自动更新“广告SDK”版本等，并根据与手机商达成的运营方案向用户推送商业性电子信息，从而产生广告费收入。为了实现公众号粉丝量快速增长，2017年2月起，欧某等人开始研发“一键达apk”，利用“广告SDK”的静默安装功能自动下载并安装“一键达apk”，在用户点击推送的文章或新闻后自动下载二维码图片，利用手机辅助功能模拟用户操作，使用户微信自动识别下载的二维码图片，关注该团伙运营的公众号，并定期自动清理二维码图片。经查，欧某等人利用上述方式非法控制移动终端1.3亿余部，利用

“广告SDK”“一键达apk”关注公众号的移动终端800余万部，共计非法获利3000万元以上。

图5 精准广告推送

本案系源头上侵害用户权益的计算机信息系统领域犯罪，在预售手机内植入“广告SDK”，其危害性、破坏性、影响力比一般的侵入式控制后果更为严重。对于以预装方式植入“广告SDK”的行为，其本质上仍是未经授权的非法行为。对于利用“广告SDK”获取的系统权限，在未经用户允许的情况下，获取用户信息、根据配置情况决定弹送广告方式、频率、静默下载其他软件等的行为，均属于非法控制行为。2019年1月25日，浙江省平湖市法院作出判决，认定被告人欧某等28人的行为构成非法控制计算机信息系统罪，判处欧某有期徒刑四年零六个月，并处罚金30万元；其余27名被告人，分别判处有期徒刑十个月至二年零八个月不等，并处罚金。一审宣判后，被告人欧某等13人提出上诉。嘉兴市中级法院二审裁定驳回上诉，维持原判。

通过对近年来因引入第三方应用组件导致的信息安全事件进行归类和分析，我们认为银行移动应用中引入合作方 SDK 面临的安全风险主要包括以下方面。

1. 对银行系统造成的风险

对银行系统造成的风险指由于合作方 SDK 存在缺陷、漏洞、后门等原因，给银行应用系统或基础设施带来信息安全风险，表现形式有：

（1）预置恶意或隐藏功能。

（2）防护不足。

（3）利用银行渠道发起攻击

（4）合作方抵赖。

2. 对银行客户造成的风险

对银行客户造成的风险指合作方对银行客户数据保护不足，或通过 SDK 对银行客户进行欺诈或非法获取银行客户数据的风险，表现形式有：

（1）客户数据泄露。

（2）违背客户知情权收集用户信息。

（3）客户信息滥用。

（4）夹带违法、违规内容。

三、监管要求

为了约束不安全的第三方软件开发工具包引入银行应用系统可能带来用户信息泄露、资产窃取等风险，规范第三方软件开发工具包引进的安全性，促进第三方软件开发工具包在银行各类应用中集成的安全、健康发展。中国人民银行于2021年7月22日发布了《JR/T 0231-2021 银行业第三方软件开发工具包（SDK）安全接入指南》。

总体原则要求第三方开发工具包做到信息保护、信息透明、无主观恶意、全生命周期管理。

其中对第三方工具包使用支付

敏感信息和个人金融信息时提出了具体的控制要求。主要包括以下九个方面：资源控制、身份认证、访问控制、数据安全、软件容错、攻击防护、安全审计、个人信息收集、第三方工具包交付。

附录A中对第三方工具包恶意行为做出了详细描述；附录B中对银行集成第三方软件开发工具包做出了详细指导。

四、规范解读

在国家总体安全观的引领下，我国不断出台多项法律规范与条例，银行业也在逐步完善一些监管体系的内容。中国人民银行发布的《JR/T 0231-2021 银行业第三方软件开发工具包（SDK）安全接入指南》于 2021 年 7 月 22 日正式施行，对第三方SDK的风险控制也提出了监管要求。银行业可参考该接入指南对已引入的第三方SDK进行评估，并为后续第三方SDK的建设需求提供监管依据。

绿盟科技主要在原文条款的基础上，一一做了细化的解读。在第三方工具包设计安全过程中从九个方面提出了控制要求。

（1）资源控制

1. 组件仅允许在自身需求权限和宿主应用授权范围内进行操作，不得提权；

2. 以宿主应用分配资源和调度为主，宿主应用为服务提供者设定服务优先级策略和调度资源，组件作为服务消费者，依照宿主应用的优先级和资源分配策略进行执行；服务优先级调度的策略有：基于线程调度器/优先级队列/加权配置/服务迁入迁出的优先级调度策略；

3. 第三方工具包如需要与外部建立会话，需要限定最大会话连接数和单位时间会话建立数，防止影响宿主应用的会话连接，宿主应用的连接数也需要依赖中间件设定的最大连接数；

4. 宿主应用需要控制第三方工具包的权限，防范权限滥用影响宿主应用，如宿主应用或者第三方工具包需要获取个人信息，需要进行声明，向用户明示且获得用户点击同意后才可搜集和上送个人信息。

（2）身份认证

第三方工具包应通过其嵌入到宿主应用的部分与后台服务器进行通信，并且在通信时要进行数字签名技术进行身份认证，不得私自与未知（或不受宿主应用

控制）的后台服务器通信，防范窃取用户或宿主应用敏感数据

所采取的身份认证方式及数字签名技术要符合《网上银行系统信息安全通用规范》中的6.2.2专用安全机制的管理规定。

（3）访问控制

1、对于第三方工具包申请的各项权限应授予最小权限，不得滥用权限。

2、第三方工具包不可与任意指定的服务器进行数据交互，仅能接受合法限定的后台服务器推送的数据，建议通过宿主应用的后台与推送数据的服务器进行数据交互。

3、宿主应用的后台通过以白名单的方式固定互联网协议地址及域名的方式限定第三方工具包的通信范围，防范DNS欺骗和流量劫持攻击影响用户；

4、后台服务器不得与工具包通信主动唤醒宿主应用，如需要唤醒操作，需要经过用户授权许可；主动唤醒的应用在后台运行，将会产生耗电及消耗流量，影响用户体验。

5、并且对传输会话设定超时机制，超过指定时间无回应，则断开连接。

6、第三方工具包明确提供自己所用到权限以及所使用的敏感信息情况给宿主应用，宿主应用在安装至用户终端后，在启动时向用户申请权限并明确敏感信息使用范围。

（4）数据安全

1和2、第三方工具包不能主动收集、存储转发宿主应用信息及用户个人信息，提供UI组件且独立提供服务的场景除外，如需存储个人信息的场景，应通过宿主应用申请存储空间的读写权限，并将个人信息进行加密存储；

3和4、独立提供UI组件的工具包，应当保证个人敏感信息展示时经过部分遮盖处理，如银行卡号、身份证号和手机号等，需要对中间部分字符经过*遮盖后才可向用户展示。同时，宿主应用需要实现防截屏和录屏，防范展示的敏感信息被截屏或录屏对外泄露。

5、当敏感信息存储于共享系统资源区域时，应通过加密的保护措施，以防范被其他软件或者进程获取

6、当一个第三方工具包被多个宿主应用嵌入时，应当以宿主应用为主；当多个这类的应用都安装在终端设备上时，后台服务器推送消息不得影响非本宿主应用程序。

7、工具包与后台服务器通信的报文需要经过加密处理，并且对传输的报文进行签名防止篡改、伪造，并加入时间戳防范重放攻击。工具包和后台服务端应通过https通信，且对数字证书进行双向校验（客户端校验服务器端证书的有效性，并且服务器端校验客户端证书的有效性），防范中间人攻击。

8、独立提供UI组件的工具包，应当保证个人敏感信息展示时经过部分遮盖处理，如银行卡号、身份证号和手机号等，需要对中间部分字符经过*遮盖后才可向用户展示。同时，宿主应用需要实现防截屏和录屏，防范展示的敏感信息被截屏或录屏对外泄露。

9、涉及个人金融信息和支付敏感信息的操作，工具包不得收集和处理，建议由银行的客户端进行输入和处理；

10、通常情况下不允许在SDK中硬编码存储敏感信息，防范被攻击者反编译获得敏感信息。

（5）软件容错

1和4、工具包嵌入宿主应用部分或者后端服务器发生故障、崩溃时，应不会影响宿主应用的使用。

2、当工具包或者后台服务端程序发生异常时，应当对用户屏蔽产生的错误信息，工具包应向宿主应用提供可枚举的异常代码进行备案，以便根据异

常代码向用户展示提示信息；

3、对于同台设备的不同宿主应用可以嵌入相同或者同类型的工具包。

4、工具包嵌入宿主应用部分或者后端服务器发生故障、崩溃时，应不会影响宿主应用的使用。

（6）攻击防护

1、工具包中使用了第三方组件或中间件时，应当确认当前组件版本不存在漏洞，当使用的组件或中间件是商业版本时，应注意获得使用或者销售许可授权；

2、3、4、5、6、8、9：应用发布时，进行加壳，保证第三方组件及宿主应用的安全性。

7、后台服务器应经过漏洞扫描、配置检查和渗透测试，防范常见漏洞的存在；

10、对于安全键盘类的工具包要对敏感信息及时加密，防止在内存中被转存，导致信息泄露。

（7）安全审计

1、2、3、4：工具包要具有独立于宿主应用的安全审计日志记录功能，审计日志要包含工具包的操作类型、操作时间、操作结果等内容，但不能包含敏感信息。要按照宿主应用的需求上报安全审计日志，确保日志的有效性；

5、6、后端服务器对嵌入宿主应用的工具包的数据通信的行为进行监测，并记录日志，记录内容应包括通信认证信息、通信起止时间、通信流量、以及是否传输敏感信息等。若工具包的审计日志涉及了银行交易日志，需要按照国家会计准则的规定，系统日志保存期限不少于1年；

（8）个人信息审计

工具包个人信息收集的行为要对用户及银行保持透明，不能欺诈、诱骗、强迫用户提供个人信息，不能收集法律法规命令禁止收集的个人信息。对于收集的个人信息类型要与实现产品或服务的业务功能有直接关联。

工具包提供方要将工具包采集的个人信息、用途及采集频率告知银行；对于客户，工具包的信息获取行为要取得客户的明示同意，要在声明中明确说明数据采集和使用的主体，声明内容要包括工具包提供方收集、使用个人信息的规则。

（9）第三方工具包交付

1、第三方工具包交付用户使用时，应提供安全测试报告，测试项需要包含规范中的所有要求项，并满足本文要求的SDK安全接入要求；

2、除安全测试报告外，需要提供工具包集成手册。包含工具包的设计、功能、调用方法等内容。

3、5、工具包的提供方如果发现工具包、宿主应用或者工具包集成的组件存在安全缺陷，需要首先通知银行单位，未经银行许可，不得私自将缺陷奚杰透漏给第三方；工具包提供方也应配合银行方进行问题复盘和事件分析；

4、当工具包发生功能或者API接口变更时，要配合银行评估分析对宿主应用的影响，并配合银行制定变更方案和应急预案，防止影响银行业务。

本次解读专家结合了一些其他的标准规范和一些实际的业务特点，在相关金融客户依据该指南进行第三方SDK评估时，能给出可落地的实际操作建议。

五、总结

基于近些年来发生的安全事件，第三方SDK的使用风险已经到了需要监管把控的边缘了。第三方 SDK 引入银行业信息系统，不仅给系统风险防控框架带来了新的需求，还给行业风险防控的重心和思路带来了变化。以往银行信息系统开发团队相对固定，开发过程相对规范，因此风险防控的重心自然被放在了应用的研发阶段，通过不断优化开发流程、加强对应用的评估验证来提升应用的安全性，在应用运行阶段，重点对外部网络层面安全风险进行监测和防控。以广泛引入合作方为特征的移动应用模式下，难以按照银行应用开发的模式和标准对合作方进行约束，势必要求风险防控的重心向事中的检测防护和事后的处置倾斜，同时也由于合作方的引入，银行应用的风险防控需要在考虑外部风险的同时，增加对应用内部违规操作、越权使用等行为的监测和防控。

风险防控思路由确保系统万无一失，向细分各功能安全边界，划清各方责任转变。以往银行业系统完全由自身营运，理所应当承担系统所有安全责任，随着合作方的引入，不再享有对系统的绝对控制权，更是难以彻底消除合作方 SDK 的安全风险，因此势必需要将应用中的各类风险进行精细化的区分，划清各方的责任边界。

《JR/T 0231-2021 银行业第三方软件开发工具包（SDK）安全接入指南》就在这个关键时刻为金融行业的第三方SDK引入建设指引了方向。

加密货币交易所 Bilaxy 遭到黑客攻击，攻击者控制了

ERC20 钱 包

摘要：位于塞舌尔的加密货币交易所 Bilaxy 周日宣布，其 ERC20 热钱包遭到黑客攻击，造成重大损失。

关 键 词：标 签 （加 密 货 币 、Bilaxy、黑客攻击、ERC20），技术问题（安全事件）。

内容：

位于塞舌尔的加密货币交易所 Bilaxy 周日宣布，其 ERC20 热钱包遭到黑客攻击，造成重大损失。

尽管该交易所鲜为人知，但这次黑客攻击可能是业内最大的黑客攻击之一。 该交易所没有正式透露黑客攻击的规模，但据许多估计，黑客窃取了约 4.5 亿美元的数字资产。

该交易所首先向其用户通报了其 Telegram 频道上的黑客行为，并详细说明它在8 月 28 日世界标准时间下午 6 点至晚上 7 点之间遭受了“严重黑客攻击”。它还透露，黑客从钱包中转移了 295 个不同的 ERC20 代币。 被泄露的代币包括 Tether (USDT)、USD Coin (USDC)、Uniswap (UNI) 等。

作为一项紧急措施，加密货币交易所暂停了其平台上的活动，并将“数百个代币” 从其热钱包转移到冷钱包中。 它进一步敦促用户停止在交易所存款。

⚠⚠顶级紧急-Bilaxy 黑客入侵通知

请注意 Bilaxy Hot 钱包被黑了， 请不要再向你的 bilaxy 帐户发送任何资金。 我们正在抓紧时间检查和修复。 请等待进一步通知。@ICODrops @ICO_Analytics @CoinMarketCap @比拉西团队 — Bilaxy (@Bilaxy_exchange) 2021 年 8 月 29 日

重大违规

该交易所现在正在调查这次攻击，并分享了资金转移到的黑客的地址。根据区块链数据，截至发稿时，该地址仅持有 100 多个以太币，价值约 322,000 美元。

基于以太坊的去中心化金融协议 Hoge Finance 表示，近 10 亿个 HOGE 代币在攻击者的 Bilaxy 黑客攻击中遭到破坏。 唯一被抽走的 HOGE 代币的法定价值约为2200 万美元。

加密货币交易所和钱包平台一直是网络犯罪分子的热门目标。 从比特币的早期开始，该行业就一直面临着攻击。

最近，攻击者破坏了日本加密货币交易所 Liquid 的安全性。 在另一起事件中， 跨链协议 Poly Network 遭受了 6.11 亿美元的黑客攻击，但黑客后来归还了大部分受损资金。

信息来源：

https://0xzx.com/20210830173716984 36.html

遭遇大规模 DDoS 攻击，俄罗斯银行业集体曝出访问故障

摘要：近日，一次针对俄罗斯银行业的大规模 DDoS 攻击，致使当地电信运营商Orange Business Services 陷入“瘫痪”状态，由此引发了连锁反应，多家头部银行的在线业务访问出现波动甚至故障。

关键词：标签（金融保险、DDoS 攻击、银行、俄罗斯），技术问题（安全事件）。

内容：

9 月 2 日深夜，一次大规模 DDoS 攻击导致多家俄罗斯银行系统宕机，部分服务无法正常使用。在此期间，各银行用户纷纷遭遇支付与卡片服务问题。

俄罗斯最大银行 Sberbank、第二大银行 VTB 以及最大私营商业银行 Alfa-Bank 虽然经受住了考验，但他们的互联网服务商 Orange Business Services 却遇上了不小的麻烦。

一位银行代表指出，“通过互联网服务商执行的所有操作，包括通过固网线路、自动取款机、POS 终端所接入的登陆点，都出现了一段时间的服务瘫痪。”

VTB 则报告称，“我们的合作伙伴及其通信服务商的 IT 服务遭遇 DDoS 攻击，导致我们远程服务渠道中的客户支付业务受到影响。”

Sberbank 则报告称，9 月 2 日有外部服务商出现故障状况，可能导致个别服务的操作发生短暂延迟。

Afla-Bank 也报告称，“Downdetector 资源记录中的部分报告，可能与当地一家互联网服务商遇到的问题有关。”

Orange Business Services 俄罗斯及独联体地区运营总监 Olga Baranova 表示， 自 8 月 9 日以来，公司的网络威胁监控中心持续记录到使用放大式攻击等手段对金融客户发动攻击，并利用加密协议（HTTPS）实施其他攻击。

她补充道，“这些攻击甚至目前仍在持续，峰值状态下流量可达 100 Gbps 左右。另外，从检测到的攻击数量而言，今年 8 月单月的攻击总量就与去年全年持平。”

正如 Qrator Labs 创始人兼 CEO Alexander Lyamin 所言，放大攻击针对的是通信渠道，而 HTTPS 或应用层攻击则将矛头直接指向应用程序本体。他总结道，“这类DDoS 攻击危险度最高，因为它们能够模拟合法流量，因此难以得到检测与消除。”

信息来源：

https://www.secrss.com/articles/34261

联合国遭网络入侵，大量内部数据或泄露

摘要：联合国披露，内部网络在今年 4 月遭到入侵；据报告该事件的安全公司Resecurity 称，攻击者可能使用了暗网泄露的联合国员工账户，并窃取了大量内部数据（联合国未予置评）；攻击者的身份和动机均未知，泄露账号在暗网仅售 1000 美元，未开启二次验证。

关键词：标签（网络攻击、数据泄露、联合国），技术问题（安全事件）。

内容：今年早些时候，黑客成功入侵联合国的计算机网络，并窃取到大量可用于攻击联合国组织的机构内部数据。黑客获取联合国网络访问权的方法似乎非常简单： 他们很可能使用了从暗网上购买到的联合国员工的失窃用户名与密码。

联合国秘书长发言人斯特凡·杜加里克昨天（9 月 9 日）在一份声明中表示：“我们可以确认，不明身份的攻击者能够在 2021 年 4 月入侵联合国的部分基础设施。” “联合国经常成为网络攻击的目标，包括持续的入侵行动。我们还可以确认，已经发现并正在响应与之前的违规行为有关的进一步攻击。”

联合国和它的关联机构之前也曾沦为黑客攻击的目标。2018 年，荷兰与英国执法部门联手挫败了俄罗斯对禁止化学武器组织实施的网络攻击，当时该组织正在调查俄罗斯在英国本土使用致使神经毒剂。据福布斯报道，2019 年 8 月，在一次针对微软 SharePoint 平台已知漏洞的网络攻击中，联合国“核心基础设施”遭到破坏，在外部机构曝光前这次事件一直没有公开披露。

泄露账号未开启二次验证，联合国内部数据或泄露

泄露凭证属于联合国专有项目管理软件 Umoja 上的某个账户。据发现此次事件的网络安全公司 Resecurity 表示，黑客能够借此深入访问联合国网络。目前了解到， 黑客掌握联合国系统访问权的最早日期为 4 月 5 日，截至 8 月 7 日他们在联合国网络上仍然保持活跃。

Resecurity 公司在今年早些时候向联合国通报了此次最新违规事件，并与联合国内部安全团队合作确定了攻击的范围。联合国的杜加里克表示，内部已经发现了这次袭击。

Resecurity 公司称，联合国官员告知 Resecurity 公司，这次黑客攻击属于侦察行为，黑客只是在内部网络上截取了屏幕截图。而当 Resecurity 公司提交了失窃数据证据之后，联合国停止了与该公司联系。

黑客使用的 Umoja 账户并未启用双因素身份验证，这是一项基础安全功

能。根据今年 7 月 Umoja 网站上的公告，该系统已经迁移至微软 Azure，这套云平台直接提供多因素身份验证机制。Umoja 的迁移公告称，此举“降低了网络安全风险”。 Resecurity 公司表示，在最近这次入侵中，黑客试图找出关于联合国计算机网络

架构设计的更多信息，并攻破了 53 个联合国账户。目前尚无法确定黑客究竟是谁， 他们又为什么要入侵联合国网络。

攻击者的身份和动机均未知，泄露账户在暗网仅售 1000 美元

黑客组织的侦察活动可能是为了筹备后续攻击，也可以是打算把信息出售给试图入侵联合国的其他团伙。

Resecurity 公司的首席执行官 Gene Yoo 称，“像联合国这样的组织是网络间谍活动中的高价值目标。攻击者入侵的目的是窃取联合国网络中的大量用户数据，以进一步进行长期的情报收集。”

Recorded Future 公司高级威胁分析师 Allan Liska 表示，“从传统上讲，像联合国这样的组织一直是民族国家攻击者的主要目标。但随着网络犯罪分子逐渐找到更有效的被盗数据货币化方法，也随着初始访问者频繁出售自己掌握的入侵资源，如今的攻击活动正表现出愈发明确的针对性与渗透趋势。”Liska 还提到，他自己曾亲眼在暗网上见到过在售的联合国雇员用户名和密码。

威胁情报公司 Intel 471 首席执行官 Mark Arena 表示，这些凭证来自多名讲俄语的网络犯罪分子，售价则仅为区区 1000 美元。

Arena 总结道，“自 2021 年初以来，我们已经发现多名出于经济动机的网络犯罪分子在出售联合国 Umoja 系统的访问权限。这些参与者会同时出售来自多个犯罪团伙的泄露凭证。结合之前的经验，这些被盗的凭证会被出售给其他网络犯罪分子， 再由他们用于实施后续入侵活动。”

彭博社查看了相应的暗网广告，发现其中至少有三个市场，最晚到 7 月 5 日还仍在出售这些联合国账户凭证。

信息来源：

https://mp.weixin.qq.com/s/3ZcKE25EaLJ238Erl11-Yw

BladeHawk 组织利用 Facebook 钓鱼攻击库尔德组织

摘要：新发现的 Android 888 RAT 已被 Kasablanka 组织和 BladeHawk 使用。

关键词：标签（钓鱼攻击、BladeHawk 组织、库尔德组织），技术问题（安全事件）。

内容：ESET 研究人员最近调查了针对库尔德族群的有针对性的移动间谍活动，该活动至少从 2020 年 3 月开始就很活跃。通过 Facebook 专门的个人资料传播两个名为 888 RAT 和 SpyNote 的安卓后门，伪装成合法的应用程序。这些个人资料似乎以库尔德语提供 Android 新闻，以及为库尔德人的支持者提供新闻。一些个人资料故意将其他间谍应用程序传播到带有亲库尔德内容的 Facebook 公共群。数据显示， 仅在 Facebook 的几篇帖子中，就有至少 1481 次来自 URL 的下载。

新发现的 Android 888 RAT 已被 Kasablanka 组织和 BladeHawk 使用。他们都使用了不同的名称来指代相同的 Android RAT——分别是 LodaRAT 和 Gaza 007。

BladeHawk Android 间谍活动

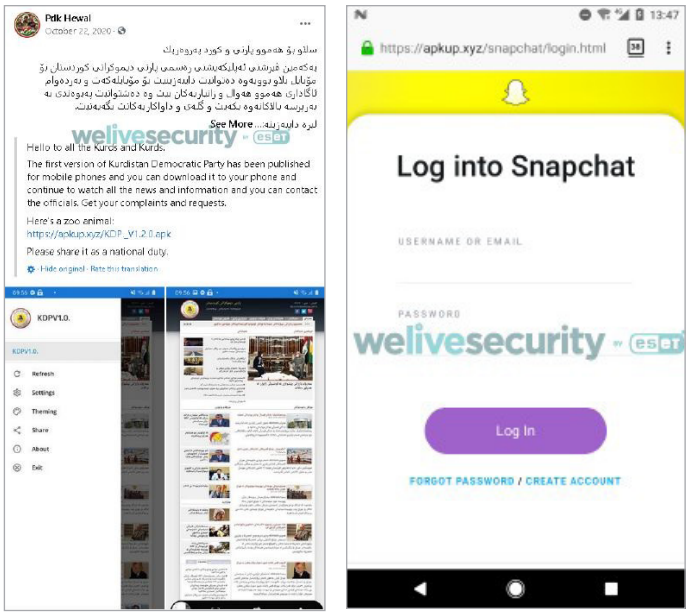
本文所指的间谍活动与 2020 年公开披露的两起案件直接相关。QiAnXin 威胁情报中心将这些攻击背后的组织命名为 BladeHawk，本文延续了这个名称。这两个活动都是通过 Facebook 传播的，使用的是由商业自动化工具（888 RAT 和 SpyNot e）构建的恶意软件，恶意软件的所有样本都使用相同的 C&C 服务器。

传播

研究人员锁定了六个 Facebook 个人资料网页，都属于 BladeHawk 活动的一部分，这些网页都分享了这些 Android 间谍应用程序。他们向 Facebook 报告了这些个人资料，目前这些页面都已被删除。其中两个页面所分享的配置文件针对的是技术用户，而另外四个配置文件的网页则伪装成库尔德支持者。所有这些资料都是在 2

020 年创建的，创建后不久，他们就开始发布这些虚假应用程序。除了一个伪装成合法应用程序的 Android RAT 帐户外，这些帐户没有发布任何其他内容。

这些个人页面还负责向 Facebook 的各类社群分享间谍应用程序，其中大部分是库尔德斯坦地区前总统马苏德·巴尔扎尼的支持者，这些目标群体总共有超过 1.1 万名粉丝。



在如下的示例中，研究人员发现了通过网络钓鱼网站以及捕获 Snapchat 凭据的尝试过程。

捕获 Snapchat 凭据的尝试

研究人员跟踪分析了 28 个有代表性的帖子作为 BladeHawk 活动的一部分。这些帖子都包含虚假的应用程序描述和下载应用程序的链接，研究人员能够从这些链接下载 17 个独特的 APK。一些 APK 网络链接直接指向恶意应用程序，而另一些则指向第三方上传服务 top4top.io，它跟踪文件下载的数量。这样，研究人员就可以从 top4top.io 获得了这八个应用程序的总下载量。从 2020 年 7 月 20 日到 2021 年 6 月 28 日，这八个应用程序总共被下载了 1481 次。

关于托管在第三方服务上的一个 RAT 样本的信息

样本分析

据我们所知，该攻击活动仅针对 Android 用户，攻击者集中使用了两个商业 A ndroid RAT 工具——888 RAT 和 SpyNote。在研究人员的分析中，他们只发现了SpyNote 的一个样本。由于它是使用旧的、已经分析过的 SpyNote 构建器构建的， 因此本文只对 888 RAT 样本进行分析。

Android 888 RAT 攻击样本分析

这个商业、多平台 RAT 最初仅以 80 美元的价格面向 Windows 生态系统发布。 2018 年 6 月，它在 Pro 版本中进行了扩展，增加了构建 Android RAT 的功能（150 美元）。后来，Extreme 版本也可以创建 Linux 有效载荷（200 美元）。

它是通过开发人员的网站 888-tools[.]com 出售的：

888 RAT 的价格

2019 年，Pro 版本（Windows 和 Android）被发现破解，并在一些网站上免费提供。

888 RAT 破解版

888 RAT 以前没有直接参与任何有组织的攻击活动，这是该 RAT 首次被指定为网络间谍组织。

在这个发现之后，研究人员能够将 Android 888 RAT 与另外两个有组织的活动联系起来，详情请点此 Spy TikTok Pro 和 Kasablanka Group 的活动描述。

该间谍活动自 2020 年 3 月以来就一直非常活跃，其只仅针对 Android 设备。它通过至少 28 个恶意 Facebook 帖子来进行钓鱼攻击，这些帖子会导致潜在受害者下载 Android 888 RAT 或 SpyNote。大多数恶意 Facebook 帖子都使用的是 888 RAT 。

信息来源：

https://www.4hou.com/posts/K9XG

NSFOCUS

漏洞聚焦

Apache Shiro 身份验证绕过漏洞（CVE-2021-41303）通告

发布时间：2021-09-17

一、漏洞概述

近日，绿盟科技 CERT 监测到 Apache Shiro 官方发布安全通告，修复了一个新的权限绕过漏洞（CVE-2021-41303）。当在 Spring Boot 中使用 Apache Shiro 时，攻击者可以构造特定的 H TTP 请求绕过身份验证访问后台功能；请相关用户采取措施进行防护。

Apache Shiro 是一个功能强大且易于使用的 Java 安全框架，功能包括身份验证、授权、加密和会话管理。使用 Shiro 的 API，可以轻松地、快速地保护任何应用程序，范围从小型的移动应用程序到大型的 Web 和企业应用程序。

参考链接：

https://www.mail-archive.com/announce@apache.org/msg06740.html

二、影响范围

受影响版本

☐ Apache Shiro < 1.8.0

不受影响版本

☐ Apache Shiro = 1.8.0

三、漏洞检测

人工监测

相关用户可通过版本检测的方式判断当前应用是否存在风险。在 config\pom.xml 的 version 标签中查看当前使用的 shiro 版本号：

```
<!-- shiro start -->
<dependency>
    <groupId>org.apache.shiro</groupId>
    <artifactId>shiro-core</artifactId>
    <version>1.2.5</version>
</dependency>
<dependency>
    <groupId>org.apache.shiro</groupId>
    <artifactId>shiro-ehcache</artifactId>
    <version>1.2.5</version>
</dependency>
<dependency>
    <groupId>net.sf.ehcache</groupId>
    <artifactId>ehcache-core</artifactId>
    <version>2.4.8</version>
</dependency>
<dependency>
    <groupId>org.apache.shiro</groupId>
    <artifactId>shiro-spring</artifactId>
    <version>1.2.5</version>
</dependency>
<dependency>
    <groupId>org.apache.shiro</groupId>
    <artifactId>shiro-web</artifactId>
    <version>1.2.5</version>
</dependency>
<!-- end shiro -->
```

若版本在受影响范围内则可能存在安全风险。

四、漏洞防护

官方升级

目前官方已在最新版本中修复了该漏洞，请受影响的用户尽快升级版本进行防护，官方 下载链接：https://shiro.apache.org/download.html

Microsoft MSHTML 远程代码执行漏洞（CVE-2021-40444）通告

发布时间：2021-09-08

一、漏洞概述

北京时间 9 月 8 日，绿盟科技 CERT 监测到微软发布安全通告披露了 Microsoft MSHTM L 远程代码执行漏洞，攻击者可通过制作恶意的 ActiveX 控件供托管浏览器呈现引擎的 Micr osoft Office 文档使用，成功诱导用户打开恶意文档后，可在目标系统上以该用户权限执行任意代码。微软在通告中指出已检测到该漏洞被在野利用，请相关用户采取措施进行防护。

MSHTML（又称为 Trident）是微软旗下的 Internet Explorer 浏览器引擎，也用于Office 应用程序，以在Word、Excel 或PowerPoint 文档中呈现Web 托管的内容。AcitveX 控件是微软 C OM 架构下的产物，在 Windows 的 Office 套件、IE 浏览器中有广泛的应用，利用 ActiveX 控件即可与 MSHTML 组件进行交互。

参考链接：

https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-40444

二、影响范围

受影响版本

- ☐ Windows Server, version 20H2 (Server Core Installation)
- ☐ Windows Server, version 2004 (Server Core installation)
- ☐ Windows Server 2022 (Server Core installation)
- ☐ Windows Server 2022
- ☐ Windows Server 2019 (Server Core installation)
- ☐ Windows Server 2019
- ☐ Windows Server 2016 (Server Core installation)
- ☐ Windows Server 2016
- ☐ Windows Server 2012 R2 (Server Core installation)
- ☐ Windows Server 2012 R2
- ☐ Windows Server 2012 (Server Core installation)
- ☐ Windows Server 2012
- ☐ Windows Server 2008 for x64-based Systems Service Pack 2 (Server Core installation)
- ☐ Windows Server 2008 for x64-based Systems Service Pack 2
- ☐ Windows Server 2008 for 32-bit Systems Service Pack 2 (Server Core installation)

- ☐ Windows Server 2008 for 32-bit Systems Service Pack 2
- ☐ Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core installation)
- ☐ Windows Server 2008 R2 for x64-based Systems Service Pack 1
- ☐ Windows RT 8.1
- ☐ Windows 8.1 for x64-based systems
- ☐ Windows 8.1 for 32-bit systems
- ☐ Windows 7 for x64-based Systems Service Pack 1
- ☐ Windows 7 for 32-bit Systems Service Pack 1
- ☐ Windows 10 for x64-based Systems
- ☐ Windows 10 for 32-bit Systems
- ☐ Windows 10 Version 21H1 for x64-based Systems
- ☐ Windows 10 Version 21H1 for ARM64-based Systems
- ☐ Windows 10 Version 21H1 for 32-bit Systems
- ☐ Windows 10 Version 20H2 for x64-based Systems
- ☐ Windows 10 Version 20H2 for ARM64-based Systems
- ☐ Windows 10 Version 20H2 for 32-bit Systems
- ☐ Windows 10 Version 2004 for x64-based Systems
- ☐ Windows 10 Version 2004 for ARM64-based Systems
- ☐ Windows 10 Version 2004 for 32-bit Systems
- ☐ Windows 10 Version 1909 for x64-based Systems
- ☐ Windows 10 Version 1909 for ARM64-based Systems
- ☐ Windows 10 Version 1909 for 32-bit Systems
- ☐ Windows 10 Version 1809 for x64-based Systems
- ☐ Windows 10 Version 1809 for ARM64-based Systems
- ☐ Windows 10 Version 1809 for 32-bit Systems
- ☐ Windows 10 Version 1607 for x64-based Systems
- ☐ Windows 10 Version 1607 for 32-bit Systems

三、漏洞检测

目前微软官方暂未针对此漏洞发布修复补丁，请相关用户采取下列措施进行防护：

3.1 版本检测

在 Internet Explorer 中禁用所有区域的 ActiveX 控件安装可缓解此漏洞攻击，可通过创建注册表文件禁用 ActiveX 控件（建议备份后再进行操作）：

```
Windows Registry Editor Version 5.00 [HKEY_LOCAL_MACHINE\SOFTWARE\
Policies\Microsoft\Windows\CurrentVersion\Intern  et Settings\Zones\0]
"1001"=dword:00000003 "1004"=dword:00000003
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\
CurrentVersion\Intern  et Settings\Zones\1]
"1001"=dword:00000003 "1004"=dword:00000003
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\
CurrentVersion\Intern  et Settings\Zones\2]
"1001"=dword:00000003 "1004"=dword:00000003
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\
CurrentVersion\Intern  et Settings\Zones\3]
"1001"=dword:00000003
"1004"=dword:00000003
```

1. 将以下方框中的内容粘贴到文本文件中并使用 .reg 文件扩展名保存：
2. 双击 .reg 文件，将其应用到策略配置单元。
3. 重新启动系统以确保应用新配置。

注：以上操作会将 64 位和 32 位进程的所有 Internet 区域的URLACTION_DOWNLOAD_SIGNED_ACTIVEX (0x1001) 和URLACTION_DOWNLOAD_UNSIGNED_ACTIVEX (0x1004) 设置为 DISABLED (3)。并不会安装新的 ActiveX 控件，之前安装的 ActiveX 控件将继续运行。撤消此缓解措施：

删除在实施此操作时添加的注册表项。

3.2 产品更新防护

目前 Microsoft Defender Antivirus 和 Microsoft Defender for Endpoint 都支持为已知漏洞提供检测和保护，请相关用户及时更新反恶意软件产品，使用自动更新的用户无需采取额外措施。管理更新的企业客户应选择检测版本 1.349.22.0 或更高版本，并在环境中进行部署。Microsoft Defender for Endpoint 警报将显示为：“可疑的 Cpl 文件执行”。

开放管理基础设施（OMI）多个高危漏洞通告

发布时间：2021-09-15

一、漏洞概述

9 月 15 日，绿盟科技 CERT 监测到微软发布 9 月安全更新补丁，修复了 86 个安全问题， 其中包括 Open Management Infrastructure 中的几个高危漏洞，某些Azure 产品（例如Configur ation Management），当开放了侦听 OMI 的HTTP/S 端口（默认为 5986）时，受下列漏洞影响。

OMI 远程代码执行漏洞（CVE-2021-38647）：未经身份认证的攻击者可通过 HTTPS 协议发送特制的数据包到目标系统的 OMI 端口，可实现远程代码执行。

OMI 权限提升漏洞（CVE-2021-38648/CVE-2021-38645/CVE-2021-38649）：经过身份验证

的普通用户可利用此类漏洞提升至系统 root 权限。

开放管理基础设施(OMI) 是一个开源项目，旨在进一步开发 DMTF CIM/WBEM 标准的生产质量实施。支持大多数的 UNIX 和 Linux 系统发行版，适用于嵌入式系统和其他基础设施组件。

参考链接：

https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-38647
https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-38648
https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-38645
https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-38649

二、影响范围

受影响版本

☐ Azure Open Management Infrastructure < omi-1.6.8-1

不受影响版本

☐ Azure Open Management Infrastructure = omi-1.6.8-1

三、漏洞检测

3.1 人工检测

用户可通过以下命令查看 Azure Linux 节点监听 OMI 端口的情况，检测系统是否受以上漏洞影响：

```
netstat -an | grep <port-number>
```

注：对于不同的服务，端口号可能不同。

四、漏洞防护

4.1 官方升级

目前官方已于 8 月 12 日在最新版本中修复了以上漏洞，请受影响的用户尽快升级版本进行防护，官方下载链接： https://github.com/microsoft/omi-kits/tree/master/release

一、根据您使用的 Linux 操作系统，将 MSRepo 安装到系统中，参考链接： https://docs.microsoft.com/en-us/windows-server/administration/Linux-Package-Repository- for-Microsoft-Software

二、使用您平台的打包工具来升级 OMI，如 sudo apt-get install omi 或 sudo yum install omi

VMware vCenter Server 多个高危漏洞通告

发布时间：2021-09-22

一、漏洞概述

9 月 22 日，绿盟科技 CERT 监测到 VMware 官方发布安全通告披露了 VMware vCenter S erver 中的多个漏洞，攻击者可利用这些漏洞造成信息泄露、权限提升、远程代码执行等。目前官方已更新版本修复，请相关用户采取措施进行防护。

vCenter Server 是 VMware 公司的一种服务器管理解决方案，可帮助 IT 管理员通过单个控制台管理企业环境中的虚拟机和虚拟化主机。

参考链接：

https://www.vmware.com/security/advisories/VMSA-2021-0020.html

二、重点漏洞简述

vCenter Server 任意文件上传漏洞 (CVE-2021-22005)：

未经身份验证的攻击者可以通过 Analytics 服务上传特制的文件到 vCenter Server 的 443端口，从而在目标系统上以该用户权限执行任意代码，CVSS 评分：9.8。

vCenter Server 权限提升漏洞 (CVE-2021-21991)：

由于 vCenter Server 处理会话令牌的方式不正确，非管理用户访问权限的攻击者利用该漏洞可将权限提升到 vSphere Client (HTML5) 或 vCenter Server vSphere Web Client (FLEX/Fla sh)的管理员权限，CVSS 评分：8.8。

vCenter Server 反向代理绕过漏洞 (CVE-2021-22006)：

由于端点处理 URL 的方式异常，未经身份验证的攻击者可利用该漏洞通过vCenter Server的 443 端口访问受限制的端点，CVSS 评分：8.3。

vCenter Server 未经身份验证的 API 端点漏洞 (CVE-2021-22011)：

由于 vCenter Server 内容库中包含一个未经身份验证的 API 端点漏洞，未经身份验证的攻击者可利用该漏洞执行 VM 网络设置操作，CVSS 评分：8.1。

vCenter Server 本地提权漏洞 (CVE-2021-22015)：

由于对文件和目录权限控制不当，导致 vCenter Server 包含多个本地提权漏洞，具有非管理用户访问权限的攻击者可利用此类漏洞在vCenter Server Appliance 上将系统权限提升为roo t，CVSS 评分：7.8。

三、影响范围

受影响版本

- ☐ VMware vCenter Server 7.0 系 列 < 7.0 U2c
- ☐ VMware vCenter Server 6.7 系 列 < 6.7 U3o
- ☐ VMware vCenter Server 6.5 系 列 < 6.5 U3q
- ☐ Cloud Foundation (vCenter Server) 4.x 系 列 < 4.3
- ☐ Cloud Foundation (vCenter Server) 3.x 系 列 < 3.10.2.2

不受影响版本

- ☐ VMware vCenter Server = 7.0 U2c
- ☐ VMware vCenter Server = 6.7 U3o
- ☐ VMware vCenter Server = 6.5 U3q
- ☐ Cloud Foundation (vCenter Server) = 4.3
- ☐ Cloud Foundation (vCenter Server) = 3.10.2.2

四、漏洞防护

4.1 官方升级

目前官方已在最新版本中修复了该漏洞，请受影响的用户尽快升级版本进行防护，对应产品版本的下载链接及文档如下：

产品版本	下载链接	操作文档
vCenter Server 7. 0 U2d	https://customerconnect.vmw are.com/downloads/details? downloadGroup=VC70U2D&productId=974&rPId=74352	https://docs.vmware.com/en/VMware-vSphere/7.0/rn/vsphere-vcenter-ser ver-70u2d-release-notes.html
vCenter Server 6. 7 U3o	https://customerconnect.vmw are.com/downloads/details?downloadGroup=VC67U3O&productId=742&rPId=73667	https://docs.vmware.com/en/VMware-vSphere/6.7/rn/vsphere-vcenter-ser ver-67u3o-release-notes.html
vCenter Server 6.5 U3q	https://customerconnect.vmw are.com/downloads/details?downloadGroup=VC65U3Q&productId=614&rPId=74057	https://docs.vmware.com/en/VMware-vSphere/6.5/rn/vsphere-vcenter-ser ver-65u3q-release-notes.html
VMware vCloud Foundation 4.3.1	https://docs.vmware.com/en/VMware-Cloud-Foundation/4.3.1/rn/VMware-Cloud-Foundation-431-Release-Notes.html	
VMware vCloud Foundation 3.10.2.2	https://docs.vmware.com/en/VMware-Cloud-Foundation/3.10.2/rn/VMware-Cloud-Foundation-3102-Release-Notes.html	

4.2 临时防护措施：

针对 VMware vCenter Server 任意文件上传漏洞（CVE-2021-22005），可参考官方给出的措施进行临时缓解：https://kb.vmware.com/s/article/85717

海康威视产品命令注入漏洞 (CVE-2021-36260)

发布时间：2021-09-22

一、漏洞概述

近日，绿盟科技 CERT 监测到海康威视发布安全通告，修复了海康威视部分产品中的 we b 模块存在的一个命令注入漏洞，由于对输入参数校验不充分，未经身份验证的攻击者通过构造带有恶意命令的报文发送到受影响设备，可实现远程命令执行。

海康威视是以视频为核心的智能物联网解决方案和大数据服务提供商，业务聚焦于智能　物联网、大数据服务和智慧业务，构建开放合作生态，为公共服务领域用户、企事业用户和　中小企业用户提供服务，致力于构筑云边融合、物信融合、数智融合的智慧城市和数字化企　业。

参考链接：

https://www.hikvision.com/cn/support/CybersecurityCenter/SecurityNotices/20210919

二、影响范围

1. 易受攻击的网络摄像机固件

产品类型	影响版本
IPC_E0	IPC_E0_CN_STD_5.4.6_180112
IPC_E1	未知
IPC_E2	IPC_E2_EN_STD_5.5.52_180620

产品类型	影响版本
IPC_E4	未知
IPC_E6	IPCK_E6_EN_STD_5.5.100_200226
IPC_E7	IPCK_E7_EN_STD_5.5.120_200604
IPC_G3	IPC_G3_EN_STD_5.5.160_210416
IPC_G5	IPC_G5_EN_STD_5.5.113_210317
IPC_H1	IPC_H1_EN_STD_5.4.61_181204
IPC_H5	IPCP_H5_EN_STD_5.5.85_201120
IPC_H8	Factory installed firmware mid 2021
IPC_R2	IPC_R2_EN_STD_V5.4.81_180203

2. 易受攻击的 PTZ 摄像机固件

产品类型	影响版本
IPD_E7	IPDEX_E7_EN_STD_5.6.30_210526
IPD_G3	IPDES_G3_EN_STD_5.5.42_210106
IPD_H5	IPD_H5_EN_STD_5.5.41_200911
IPD_H7	IPD_H7_EN_STD_5.5.40_200721
IPD_H8	IPD_H8_EN_STD_5.7.1_210619

3. 易受攻击的旧固件

产品类型	影响版本
IPC_R7	5.4.x
IPD_R7	
IPC_G0	
IPC_H3	
IPD_H3	

4. OEM 固件

三、漏洞防护

目前海康威视官方已发布新版本修复该漏洞，请受影响用户尽快更新进行防护，下载链接：https://www.hikvision.com/cn/support/CybersecurityCenter/SecurityNotices/20210919/

各受影响的产品版本与修复程序下载链接如下：

序号	产品名称	受影响版本号	修复程序下载
1	DS-2CVxxxx	版本 build 日期在 210625 之前	https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/01%EF%BC%9ADS-2CVxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
2	DS-2CD1xxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/02%EF%BC%9ADS0-2CD1xxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
3	IPCxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/03%EF%BC%9AIPCxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
4	DS-IPC-Bxx DS-IPC-Txx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/05%EF%BC%9ADS-IPC-Exx%E3%80%81Sxx%E3%80%81Axx%E3%80%81DS-2XDxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
5	DS-IPC-Exx DS-IPC-Sxx DS-IPC-Axx DS-2XDxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/05%EF%BC%9ADS-IPC-Exx%E3%80%81Sxx%E3%80%81Axx%E3%80%81DS-2XDxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
6	DS-2CD2xxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/06%EF%BC%9ADS-2CD2xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
7	DS-2CD3xxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/07%EF%BC%9ADS-2CD3xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
8	(i)DS-2DCxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/08%EF%BC%9A(i)DS-2DCxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip

序号	产品名称	受影响版本号	修复程序下载
9	(i)DS-2DExxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/09%EF%BC%9A(i)DS-2DExxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
10	(i)DS-2PTxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/10%EF%BC%9A(i)DS-2PTxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
11	(i)DS-2SE7xxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/11%EF%BC%9A(i)DS-2SE7xxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
12	DS-2DBxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/12%EF%BC%9ADS-2DBxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
13	DS-2DYHxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/13%EF%BC%9ADS-2DYHxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
14	DS-DY9xxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/14%EF%BC%9ADS-2DY9xxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
15	iDS-2DY5Cxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/15%EF%BC%9AiDS-2DY5Cxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
16	iDS-2DP9Cxxx-T4		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/16%EF%BC%9AiDS-2DP9Cxxx-T4%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
17	DS-2DY7xxx-CX(S5) DS-2DF6xxx-CX(S6) DS-2DF6Cxxx-CX(T2)		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/17%EF%BC%9ADS-2DY7xxx-CX%EF%BC%88S5%EF%BC%89%E3% 80%812DF6xxx-CX%EF%BC%88S6%EF%BC%89%E3%80%812DF6Cxxx-CX%EF%BC%88T2%EF%BC%89%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
18	iDS-2VY4xxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/18%EF%BC%9AiDS-2VY4xxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip

序号	产品名称	受影响版本号	修复程序下载
19	iDS-EGDxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/19%EF%BC%9AiDS-EGDxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
20	DS-2CD4xxx DS-2CD5xxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/20%EF%BC%9ADS-2CD4xxx%E3%80%815xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
21	DS-2CD6xxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/21%EF%BC%9ADS-2CD6xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
22	DS-2CD7xxx DS-GPZxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/22%EF%BC%9ADS-2CD7xxx%E3%80%81DS-GPZxxx%E7%B3%BB% E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
23	DS-2CD8xxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/23%EF%BC%9ADS-2CD8xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
24	DS-2XA8xxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/24%EF%BC%9ADS-2XA8xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
25	DS-FCNxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/25%EF%BC%9ADS-FCNxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
26	iDS-2XM/CD6xxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/26%EF%BC%9AiDS-2XM%E3%80%81CD6xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
27	DS-2DF5xxxx DS-2DF6xxxx DS-2DF6xxxx-Cx DS-2DF7xxxx DS-2DF8xxxx DS-2DF9xxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/27%EF%BC%9ADS-2DF5%E 3%80%816%E3%80%817%E3%80%818%E3%80%819xxx%E7%B3%BB%E5%88%97%E3%80%81DS-2DF6xxx-CX%E7%B 3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
28	iDS-2VPDxxxx iDS-2DPxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/28%EF%BC%9AiDS-2VPDxxxx%E3%80%812DPxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip

序号	产品名称	受影响版本号	修复程序下载
29	iDS-2PT9xxxx		https://hiknow-cn-s3.s3.cn-north-1.amazon aws.com.cn/29%EF%BC%9AiDS-2PT9xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip https://hiknow-cn-s3.s3.cn-north-1.amazon
30	iDS-2SK7xxxx iDS-2SK8xxxx		https://hiknow-cn-s3.s3.cn-north-1.amazon aws.com.cn/30%EF%BC%9AiDS-2SK7%E 3%80%818xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
31	iDS-2SR8xxxx		https://hiknow-cn-s3.s3.cn-north-1.amazon aws.com.cn/31%EF%BC%9AiDS-SR8xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
32	iDS-2VSxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazon aws.com.cn/32%EF%BC%9AiDS-2VSxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
33	iDS-2VTxxxx		https://hiknow-cn-s3.s3.cn-north-1.amazon aws.com.cn/33%EF%BC%9AiDS-2VTxxxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
34	iDS-GPZ2xxxx		https://hiknow-cn-s3.s3.cn-north-1.amazon aws.com.cn/34%EF%BC%9AiDS-GPZ2xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
35	DS-2XE62x7FWD(D) DS-2XE30x6FWD(B) DS-2XE60x6FWD(B) DS-2XE62x2F(D) DS-2XC66x5G0 DS-2XE64x2F(B)		https://hiknow-cn-s3.s3.cn-north-1.amazon aws.com.cn/35%EF%BC%9ADS-2XE%E3%80%81XC6xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
36	KBA18(C)-83x6FWD		https://hiknow-cn-s3.s3.cn-north-1.amazon aws.com.cn/36%EF%BC%9AKBA18%EF%BC%88C%EF%BC%89-8xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA% A7%E5%8C%85.zip
37	DS-2TBxxx DS-Bxxxx DS-2TDxxxx B TBC-12xxx TBC-26xxx	版本 build 日期在 210702 之前	https://hiknow-cn-s3.s3.cn-north-1.amazon aws.com.cn/37%EF%BC%9ADS-TBxxx% E3%80%81DS-Bxxxx%E3%80%81DS-2T DxxxxB%E3%80%81TBC-12xxx%E3%80%81TBC-26xxx%E7%B3%BB%E5%88%9 7%E5%8D%87%E7%BA%A7%E5%8C%85.zip

序号	产品名称	受影响版本号	修复程序下载
38	DS-2TD1xxx-xx DS-2TD2xxx-xx		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/38%EF%BC%9ADS-TD1xxx%E3%80%81DS-2TD2xxx%E7%B3%BB% E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
39	DS-2TD51xx-xx/W/GL T DS-2TD55xx-xx/W DS-2TD65xx-xx/W		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/39%EF%BC%9ADS-2TD51xx-xx%E3%80%81DS-2TD55xx-xx%E3%80%81DS-2TD65XX-XX%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
40	DS-2TD41xx-xx/Wxx DS-2TD62xx-xx/Wxx DS-2TD81xx-xx/Wxx DS-2TD91xx-xx/W DS-2TD4xxx-xx/V2 DS-2TD55xx-xx/V2 DS-2TD6xxx-xx/V2 DS-2TD81xx-xx/V2 DS-2TD91xx-xx/V2		https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/40%EF%BC%9ADS-2TD41xx-xx%E3%80%81DS-TD62xx-xx%E3%80% 81DS-2TD81xx-xx%E3%80%81DS-2TD91 xx%E3%80%81DS-2TD4xxx%E3%80%81 DS-2TD55xx%E3%80%81DS-TD6xxx%E7%B3%BB%E5%88%97%E5%8D%87%E7%BA%A7%E5%8C%85.zip
41	DS-76xxN-Exx DS-78xxN-Kxx DS-NVR-K1xx DS-NVR-K2xx	V4.30.210 Build201224-V4.31.000 Build210511	https://hiknow-cn-s3.s3.cn-north-1.amazonaws.com.cn/41%EF%BC%9ADS-76xxN-E xxxxx%E3%80%81DS-78xxN-Kxxxxx%E3%80%81DS-NVR-K1xx%E3%80%81DS-NVR-K2xx.zip

NSFOCUS

安全态势

互联网安全威胁态势

行业动态回顾

1. Marketo组织窃取了PUMA公司1GB的数据

【概述】

8月29日，研究人员发现，Marketo组织窃取了PUMA公司1GB的数据，该组织作为一个“被盗数据市场”的运营商，不同于典型的勒索软件集团，他们是通过阻止受害者的网络，加密各种数据存储上的可用文件来分发恶意代码，破坏IT运营服务。窃取PUMA公司的敏感数据在暗网平台上进行公开拍卖。其中包含链接到公司产品管理门户内部管理应用程序的源代码，攻击者可以使用这些数据来策划对公司更复杂的攻击。

【参考链接】

https://ti.nsfocus.com/security-news/llMOp

2. 攻击者向用户分发虚假的电子邮件进行网络钓鱼活动

【概述】

近日，研究人员检测到多起以新冠疫苗COVID-19为主题的网络钓鱼活动。攻击者利用虚假的网络钓鱼电子邮件为诱饵向用户发送恶意构造的样本链接欺骗用户点击，此次诱饵文件名字为“Certification-Vaccination-Status-Form.pdf”，受害者通过点击诱饵文件启动PowerShell程序并执行恶意脚本后，程序会从指定的网络地址请求并获取后续的PowerShell恶意脚本，会将目标员工带到一个冒充MicrosoftOutlookWeb应用程序登录页面的恶意域，最终他们将被带到一个假冒受信任品牌的被劫持网页。

【参考链接】

https://ti.nsfocus.com/security-news/llMOj

3. 波士顿公共图书馆遭到黑客攻击

【概述】

研究人员表明，波士顿公共图书馆计算机网络遭到黑客严重攻击，导致该图书馆整个系统中断，受影响的系统已下线，暂停了公共计算机和公共打印服务，以及一些在线资源。目前，该图书馆仍然开放，但大部分电子功能处于离线状态，现在所有的工作站点都在手动处理交易。

【参考链接】

https://ti.nsfocus.com/security-news/llMOi

4. 攻击者利用KonniRAT恶意软件攻击俄罗斯

【概述】

MalwarebytesLabs的研究人员发现了一个正在进行的恶意软件活动，此次攻击活动的目标是俄罗斯。攻击者利用两份俄语编写的武器化文件作为诱饵，其中一份利用俄罗斯与朝鲜半岛之间的贸易和经济问题，第二份文件以俄罗斯-蒙古政府间委员会的会议为诱饵。当攻击者启用宏后，它执行的感染链将开始部署一个经过严重混淆的新KonniRAT变体，攻击者试图在文档内容的末尾隐藏其主要活动开始的恶意JS，并没有将其直接放入宏中，以避免被AV产品检测到并隐藏其他们的主要意图。

【参考链接】

https://ti.nsfocus.com/security-news/llMO1

5. 闪电贷黑客窃取了金融平台CREAM Finance 2900万美元的加密货币财产

【概述】

近日，研究人员发现，闪电贷黑客从金融平台CREAMFinanceDefi窃取了超过2900万美元的加密货币资产。CREAMFinanceDefi是一种去中心化借贷协议，供个人、机构和协议访问金融服务。它向被动持有ETH或wBTC的用户承诺收益。在攻击活动中，攻击者在其“闪电贷”功能中进行了“重入攻击”，窃取了418,311,571个AMP代币和1,308.09个ETH代币。黑客在转移资产过程中通过重新借用Amp代币进行了500ETH的闪贷，然后在17笔单独的交易中更新第一个借入资产。提供了一个示例事务，黑客使500ETH的flashloan和存入的资金作为抵押。CREAMFinance宣布，已经暂停Amp代币的供应和借用合约来阻止黑客利用。

【参考链接】

https://ti.nsfocus.com/security-news/llMOx

6. LockBit勒索软件组织攻击曼谷航空公司

【概述】

8月30日，LockBit勒索软件组织成功攻击了曼谷航空公司的内部系统，窃取了该公司103GB与其客户有关的个人数据，被盗数据包括姓名、国籍、性别、电

话号码、电子邮件、地址、联系信息、护照信息、历史旅行信息、部分信用卡信息和航空公司乘客的特殊膳食信息。攻击者在其泄密网站上发布了一条消息，如果曼谷航空公司不支付赎金，就会泄露被盗数据，消息还显示他们有更多的数据要泄露。

【参考链接】

https://ti.nsfocus.com/security-news/llMOC

7. 诈骗者通过冒充OpenSea公司员工窃取数字资产

【概述】

OpenSea是一个基于区块链的数字资产市场，诈骗者冒充OpenSea公司员工以窃取数字资产，OpenSea用户和艺术家JeffNicholas成为该骗局的受害者，攻击者从Ledger钱包中窃取了他持有的数字资产以及价值约14,600美元的4.5以太币。攻击的方式是，诈骗者使用Discord聊天平台提供客户支持，人们被告知去OpenSeaDiscord并发布他们的支持票，攻击者正在监视这些渠道，然后联系冒充OpenSea支持的人，并提供有关他们支持索赔的信息。

【参考链接】

https://ti.nsfocus.com/security-news/llMOA

8. 攻击者使用开放重定向链接引诱用户访问恶意网站获取Office365凭据

【概述】

根据最近发布的一份研究报告称，存在“广泛”的网络钓鱼活动，欺诈者使用开放重定向链接引诱用户访问恶意网站，以获取Office365凭据。除了使用社会工程技术模仿生产力工具和服务来引诱用户点击之外，欺诈者还会部署一个恶意的CAPTCHA验证页面，帮助引诱用户点击恶意链接并帮助欺诈者避开某些安全工具，用户会单击重定向链接，这会打开一个他们必须填写的虚假CAPTCHA站点，一旦受害者完成伪造的CAPTCHA页面，用户就会被发送到一个恶意域，该域旨在看起来像一个合法的Office365或其他登录网站。会要求用户输入两次凭据，以确保欺诈者收集正确的用户名和密码组合。一旦用户第二次输入密码，该页面就会定向到一个合法的Sophos网站，该网站声称该电子邮件已被释放，从而为攻击增加了另一层合法性。

【参考链接】

https://ti.nsfocus.com/security-news/llMOF

9. 英国电信公司遭受DDOS攻击

【概述】

英国两家基于互联网的电信公司VoipUnlimited和Voipfone在其网站上报告说，它们已经遭到了分布式拒绝服务DDOS持续的攻击，导致公司系统核心网络已经中断了服务，以及中断了呼叫、注册和客户门户访问等服务，造成的服务瘫痪每小时损失低于1000英镑，攻击媒介“不断变化”，其网络团队正在根据需要应用缓解措施。

【参考链接】

https://ti.nsfocus.com/security-news/llMPu

10. FIN7团伙利用Word文档来投放恶意负载攻击美国销售点服务提供商

【概述】

Anomali Threat Research 专家监测了最近由FIN7团伙进行的鱼叉式网络钓鱼攻击活动，此次攻击的目标是美国销售点 (PoS) 服务提供商， 该团伙通过采用Word文档来投放恶意负载攻击链侵入PoS服务商网络，攻击链始于一个 Microsoft Word 文档

(.doc)，其中包含一个声称使用 Windows 11 Alpha 制作的诱饵图像。该图像要求收件人启用编辑和启用内容以访问其内容，启用宏后，将执行高度混淆的 VBA 宏以检索 JavaScript 负载。恶意脚本还会检查虚拟机以防止在虚拟化环境中进行分析，为了避免发现，该组织还在 VBA 宏中插入垃圾数据。

【参考链接】

https://ti.nsfocus.com/security-news/llMPj

11. AVOSLockerRansomware运营商攻击太平洋城市银行

【概述】

PacificCityBank是一家美国太平洋城市银行，专注于位于加利福尼亚州的韩裔美国人社区，并提供商业银行服务，该银行遭到AVOSLockerRansomware运营商的攻击，并且从金融机构窃取了敏感文件。2021年9月4日，勒索软件团伙将该银行信息添加到其泄密站点，并发布了一些屏幕截图作为攻击的证据。

【参考链接】

https://ti.nsfocus.com/security-news/llMPI

12. RagnarLocker勒索软件团伙窃取台湾威刚公司1.5TB的数据

【概述】

RagnarLocker勒索软件团伙成功窃取台湾威刚公司1.5TB的数据，被盗数据包含敏感信息，如保密协议、财务文件、合同和其他文件。该公司拒绝支付黑客要求的赎金。如果受害者试图联系执法机构，RagnarLocker勒索软件团伙威胁要泄露被盗数据。该组织在其暗网泄漏站点上发布了一条消息，宣布了其新战略，如果受害者试图联系执法机构，RagnarLocker勒索软件运营商威胁要泄露被盗数据。

【参考链接】

https://ti.nsfocus.com/security-news/llMPS

13. BladeHawk组织针对库尔德族群Android用户有针对性发起攻击

【概述】

研究人员发现，BladeHawk组织针对库尔德族群Android用户有针对性发起

攻击。该组织利用两个商业AndroidRAT工具，分别是888RAT和SpyNote。利用Android888RAT能够执行从其C&C服务器收到的42个命令，从设备中窃取和删除文件、截取屏幕截图、获取设备位置、钓鱼Facebook凭据、获取已安装的应用程序列表、窃取用户照片、拍照、记录周围的音频和电话、拨打电话、窃取短信信息、窃取设备的联系人列表、发送短信等。

【参考链接】

https://ti.nsfocus.com/security-news/llMPW

14. 攻击者利用Conti勒索软件攻击HSE爱尔兰国家卫生服务提供商

【概述】

研究人员发现，针对爱尔兰国家卫生服务提供商HealthServiceExecutive的勒索软件攻击活动，攻击中使用了Conti勒索软件，此次攻击对HSE的系统造成了广泛破坏，攻击者声称从HSE窃取了700GB患者的个人数据，包括个人文件、电话号码、联系人、工资单和银行对帐单，然后要求支付2000万美元，但爱尔兰总理迈克尔·马丁拒绝支付任何赎金，并告诉全国媒体，政府没有与袭击者沟通。然而，一周后，被指控的攻击者向HSE提供了一个解密密钥，条件是它支付1900万美元的赎金或公开其患者数据。

【参考链接】

https://ti.nsfocus.com/security-news/llMPN

15. REvil勒索软件运营商攻击Kaseya基于云的MSP平台

【概述】

REvil勒索软件团伙袭击了Kaseya基于云的MSP平台，影响了MSP及其客户。该团伙破坏了KaseyaVSA的基础设施，然后推出了VSA内部部署服务器的恶意更新，以在企业网络上部署勒索软件。该组织要求提供价值7000万美元的比特币来解密所有受Kaseya供应链勒索软件攻击影响的系统。这次袭击引起了媒体和警察当局的注意，增加了对该组织的压力。

【参考链接】

https://ti.nsfocus.com/security-news/llMQf

16. 攻击者利用Trojan.Win32.BreakWi恶意软件攻击伊朗网络安全公司

【概述】

近日，研究人员发现，攻击者利用Trojan.Win32.BreakWi恶意软件攻击伊朗网络安全公司，伊朗铁路道路与城市发展系统部成为网络攻击的目标。黑客在全国各地车站的信息板上显示火车延误或取消的信息，并敦促乘客拨打电话以获取更多信息，之后，伊朗道路和城市化部的网站出现“网络中断”后停止服务，攻击者在网络安全公司网络中开发并部署了至少3种不同版本的工具（Meteor、Stardust、Comet）。攻击主要有效载荷是msapp.exe，其目的是锁定受害者机器并擦除其内容使其停止服务。执行时恶意软件会隐藏此可执行文件的控制台窗口。

【参考链接】

https://ti.nsfocus.com/security-news/llMQj

17. APT组织使用GoldenSAML攻击获取对ActiveDirectory的访问权限

【概述】

APT组织使用GoldenSAML攻击来绕过身份验证控制并访问Office365环境。大多数受害者采用混合身份验证模型，破坏ADFS服务器令牌签名

证书会导致访问Azure/Office365环境。默认情况下，证书的有效期为一年，这允许APT组织以AD中的任何用户身份保持持久性并重新进入Azure/Office365环境，而不管任何密码重置或多重身份验证。

【参考链接】

https://ti.nsfocus.com/security-news/llMQ8

18. 俄罗斯互联网服务提供商Yandex受到大规模拒绝服务（DDOS）攻击

【概述】

美国公司Cloudflare证实了大规模DDoS攻击的事件，此次攻击的目标是俄罗斯互联网服务提供商Yandex，Yandex公司服务器遭遇俄罗斯网史上最强的一次DDoS攻击，该公司表示，此次攻击持续时间长达6小时，导致互联网已陷入瘫痪。影响了大量的社交媒体用户。

【参考链接】

https://ti.nsfocus.com/security-news/llMQz

19. 新西兰多个银行和邮局遭到DDOS攻击

【概述】

研究人员表明，新西兰多个银行和邮局遭到持续的分布式拒绝服务DDOS攻击，此次攻击导致该国的银行和邮局的网站已经关闭，部分业务已经中断，包括应用程序、网上银行、电话银行和网站已经中断。其他受害者包括澳新银行新西兰有限公司，周三，ANZ Bank New Zealand Ltd.在 Facebook 上发帖称，它经历了一次中断，影响了对其部分在线服务的访问。官员们表示他们正在与网络攻击作斗争。一些受影响的组织能够使他们的服务重新上线，但他们仍然遇到间歇性中断。

【参考链接】

https://ti.nsfocus.com/security-news/llMQB

20. SANGKANCIL黑客窃取了CITY4U网站700万以色列人的个人信息

【概述】

9月7日，以色列地方当局的CITY4U网站被一名名为SANGKANCIL的黑客入

侵，该黑客声称已经窃取了该网站700万以色列人的详细信息。在他的Telegram帐户中，他分享了一些随机照片，其中包含几位以色列公民的个人详细信息，例如身份证及其照片、地址、全名、电话号码、财产税支付情况等。

【参考链接】

https://ti.nsfocus.com/security-news/llMQp

21. APT组织利用SideWalk恶意软件针对美国计算机零售公司发起攻击

【概述】

研究人员发现了APT组织利用SideWalk恶意软件针对美国一家计算机零售公司发起攻击。攻击者通过入侵公司的Microsoft Exchange 或 MySQL 服务器，一旦在Microsoft Exchange 或 MySQL Web 服务器上部署 Web shell，攻击者就会在目标网络中横向传播。

【参考链接】

https://ti.nsfocus.com/security-news/llMQF

22. 黑客窃取Fortinet虚拟专用网络近50万客户的账户和密码

【概述】

9月8日，一名网络攻击者窃取了近50万Fortinet VPN登录名和密码，因为虚拟专用网络凭据可能允许威胁行为者访问网络执行数据外泄、安装恶意软件和执行勒索软件攻击。

【参考链接】

https://ti.nsfocus.com/security-news/llMPS

23. 黑客窃取了联合国用户的数据

【概述】

研究人员发现，黑客攻击联合国的基础设施，并且窃取联合国系统网络中的大量用户数据，并且将窃取的大量信息出售给试图入侵联合国的其他团伙，以进一步收集长期情报。之后，黑客试图获取更多有关联合国计算机网络架构的信息，并入侵了53个联合国账户，将窃取的内网账号和密码在暗网出售。

【参考链接】

https://ti.nsfocus.com/security-news/llMQV

24. 攻击者利用BlackMatter 勒索软件攻击科技巨头奥林巴斯

【概述】

研究人员发现，针对科技巨头奥林巴斯的勒索软件攻击活动，攻击中使用BlackMatter勒索软件，此次攻击对奥林巴斯的系统造成了广泛破坏，导致奥林巴斯停止了客户的所有文件传输。

【参考链接】

https://ti.nsfocus.com/security-news/llMRi

25. Sextortion勒索软件诈骗者利用虚假的电子邮件获取受害者设备的访问权限

【概述】

Sextortion 是一个通过电子邮件或任何其他媒介勒索受害者的骗局团伙，并威胁要公开照片、网页浏览历史记录、聊天记录等私人数据。该勒索软件诈骗者通过发送电子邮件获得对受害者设备的访问权限，当受害者点击勒索电子邮件时，通过登录电子邮件在设备上安装了木马病毒。为了使邮件看起来更真实，攻击者通常会发送主题为“从您的帐户付款”的电子邮件。然后声称您的活动正在通过您的设备（如相机、麦克风等）的控制器被记录。受害者有 48 小时的时间将 1550 美元转移到骗子的比特币钱包，标识如果没有向比特币地址付款， 攻击者会威胁受害者公开私人数据。

【参考链接】

https://ti.nsfocus.com/security-news/llMQ1

26. 泰国肾脏医院四万名患者数据被盗

【概述】

近日，研究人员发现，泰国一所肾脏专科医院系统遭网络攻击者入侵，导致该医院患者数据库无法访问，随后，技术人员在对系统进行了检查之后，发现有四万多名患者的个人信息和病例信息被网络攻击者盗取。此次数据泄露事件破坏了医院的数据系统，导致医生无法正常访问患者的X光档案信息。之后，攻击者试图通过电话谈判来勒索医院，以向其支付数据赎金并拿回被盗的患者数据。

【参考链接】

https://ti.nsfocus.com/security-news/llMRu

27. 攻击者利用 maxtrilha 银行木马攻击欧洲和南美银行的客户

【概述】

研究人员表明，发现了一种名为maxtrilha的新银行木马，该银行木马正在传播，攻击目标是欧洲和南美银行的客户，攻击者在执行期间，通过短URL打开目标合法页面，在目标机器上创建持久性，它使用 TinyURL 在线服务，该服务在恶意软件执行期间由受害者计算机上安装并可用的默认 Web 浏览器打开。短 URL 指向与网络钓鱼模板相关的特定页面以引诱受害者。

【参考链接】

https://ti.nsfocus.com/security-news/llMR2

28. 南非司法部和宪法发展部遭到勒索软件攻击

【概述】

研究人员表明，南非司法部和宪法发展部系统遭到勒索软件攻击，攻击者利用勒索软件包含恶意附件的网络钓鱼电子邮件或通过偷渡式下载进行传播，系统被攻击后，导致该部

门信息技术系统以及电子邮件和保释多项服务瘫痪，以及所有的信息系统加密。

【参考链接】

https://ti.nsfocus.com/security-news/llMR4

29. Anonymous黑客团体攻击网络托管服务提供商Epik

【概述】

研究人员称，Anonymous黑客团体入侵了网络托管服务提供商Epik，并窃取了该公司180GB用户的数据、注册、转发等信息，并在 DDoSecrets 非盈利举报网站上泄露。据黑客称，被盗数据包括：域名购买，域名转移，WHOIS历史，DNS更改，电子邮件转发，支付历史，账户凭证，超过500000个私钥等信息。

【参考链接】

https://ti.nsfocus.com/security-news/llMRD

30. 攻击者利用Operation Layover恶意软件攻击航空航天和旅游行业

【概述】

近日，研究人员发现，一项有针对性的网络钓鱼活动由在尼日利亚开展业务的攻击者带头发起，目标是攻击航空航天和旅游行业，此次攻击活动利用Operation Layover恶意软件，通过鱼叉式网络钓鱼电子邮件分发积极开发的加载程序，然后提供 RevengeRAT 或 AsyncRAT，攻击者将多个 RAT 编织到他们的活动中，将基础设施用作 Cybergate RAT、AsyncRAT 和批处理文件的命令和控制 (C2) 服务器作为恶意软件链的一部分来下载和执行其他恶意软件。

【参考链接】

https://ti.nsfocus.com/security-news/llMS9

31. 攻击者利用Numando新型银行木马针对墨西哥和西班牙用户发起攻击

【概述】

研究人员发现了一种新的 LATAM 银行木马，被命名为 Numando，它滥用 YouTube、Pastebin 和其他公共平台作为 C2 基础设施并进行传播。专家发现了攻击者利用Numando恶意软件针对墨西哥和西班牙用户的发起攻击。与其他拉

丁美洲银行木马一样，它是用 Delphi 编写的，并利用虚假的覆盖窗口来诱骗受害者提供敏感信息。一旦 Numando 安装在目标机器上，每当受害者访问金融组织的网站并捕获他们提供的凭据时，它都会创建虚假的覆盖窗口。

【参考链接】

https://ti.nsfocus.com/security-news/IlMSj

32. APT-C-36间谍组织通过伪装成哥伦比亚政府机构向用户传播虚假的电子邮件

【概述】

研究人员发现了一项针对南美组织的鱼叉式网络钓鱼邮件的垃圾邮件活动，这些攻击归因于一种被称为APT-C-36的高级持续威胁组织 (APT)，该组织主要通过伪装成哥伦比亚政府机构，向客户分发欺诈性的电子邮件，当邮件收件人打开诱饵 PDF 或 Word 文档时，感染链就开始，该邮件声称是与其相关的扣押令，银行帐户，然后单击从 URL 缩短器服务生成的链接。APT-C-36 会根据位置和电子邮件收件人的财务状况来选择他们的目标。

【参考链接】

https://ti.nsfocus.com/security-news/IlMSn

33. TeamT.NT僵尸网络团伙攻击全球各地的组织机构

【概述】

研究人员发现了一项名为Chimaera的新活动，这项活动由TeamT.NT僵尸网络团伙进行，目标是攻击全球各地的组织机构。为了窃取全球各地的组织机构登录凭证，TeamT.NT在他们的攻击设备库中添加了许多工具，包括shell脚本、隐蔽挖矿、IRC和开源软件等。他们一直在攻击Windows系统和各种Linux操作系统，以及AWS、Docker和Kubernetes，人们注意到其过去也攻击过MacOS系统。全球超过5000多例恶意软件感染事件归因于TeamTNT。

【参考链接】

https://ti.nsfocus.com/security-news/IlMSo

34. 明尼苏达州农业供应合作社水晶谷遭遇勒索软件攻击

【概述】

明尼苏达州农业供应合作社水晶谷遭遇勒索软件攻击，该合作社为 2,500 名农民和牲畜生产者提供服务，并拥有 260 名员工。这次攻击已经感染了水晶谷的计算机系统，严重中断了公司的日常运营，该公司已经关闭了 IT 系统，暂停了所有使用 Visa.Mastercard 和 Discover 信用卡的付款。

【参考链接】

https://ti.nsfocus.com/security-news/IlMSV

35. Turla组织攻击阿富汗的政府组织和机构

【概述】

一个与俄罗斯有关联的Turla 组织一直以政府组织的系统为目标，通过在该机构上部署恶意软件，以保持在受感染设备中的持久性，研究人员将恶意软件称为TinyTurla，在过去两年中已针对美国和德国系统部署。最近，Turla 在阿富汗于 8 月被塔利班占领之前就使用了该恶意软件来攻击阿富汗的政府组织和机构。 Turla 将恶意软件伪装成一个名为“Windows 时间服务”的合法 Microsoft 文件，该文件允许恶意代码在后台运行并与受感染设备上的合法应用程序混合。

【参考链接】

https://ti.nsfocus.com/security-news/IlMSS

36. 攻击者利用商业RAT诱饵攻击针对印度政府人员发起攻击

【概述】

Cisco Talos 最近发现了一个针对印度次大陆政府雇员和军事人员的恶意活动，其中有两个 RAT 诱饵系列，称为NetwireRAT和WarzoneRAT。攻击者向其印度政府人员发送了各种诱饵，主要伪装成与印度政府基础设施和操作相关的指南，当受害者使用恶意文档下载和检测加载程序，加载器负责下载或解密最终的 RAT 负载并将其部署在受感染的端点上。

【参考链接】

https://ti.nsfocus.com/security-news/llMT1

37. 攻击者窃取了1.06亿泰国游客

【概述】

研究人员发现，攻击者入侵了泰国的Elasticsearch数据库，并且窃取了1.06亿泰国游客，窃取的个人信息包括旅行者的全名、护照号码、居留身份、抵达泰国的日期、移民入境卡号码和签证类型。

【参考链接】

https://ti.nsfocus.com/security-news/llMSR

38. 俄罗斯的Yandex遭受史上最大的DDOS攻击

【概述】

Yandex是俄罗斯的第一大搜索引擎，很多人会将其称为俄罗斯的“百度”，以及是俄罗斯最大的互联网服务提供商，涵盖了搜索引擎、电子商务、电子邮件等互联网业务，称之为俄罗斯的“BAT”。 据外媒报道，近日Yandex遭受了俄罗斯互联网历史上最大的DDoS攻击，攻击峰值达到了每秒2180万次请求。Yandex内部人士称本次DDoS攻击难以遏制，截止本周仍在继续遭受攻击。本次DDoS攻击是由一个新的僵尸网络发起的，该僵尸网络被标记为Méris，由大约20多万台设备组成。

【参考链接】

https://ti.nsfocus.com/security-news/llMSA

39. Marketron遭到 Black Matter 团伙发起的勒索软件攻击

【概述】

Marketron是一家为媒体行业提供企业收入和管理解决方案的供应商，Marketron Broadcast Solutions遭到 Black Matter 团伙发起的勒索软件攻击，该攻击已下架了该营销公司的许多产品。此次攻击直接影响了该公司的 6,000 家媒体行业客户，大部分服务仍处于离线状态。这个数字肯定会在下游效应中呈指数级增长。

【参考链接】

https://ti.nsfocus.com/security-news/llMSD

40. 航运巨头CMA CGM遭遇攻击者袭击

【概述】

法国航运公司CMA CGM称，在遭受勒索软件攻击导致其系统离线数天后，该公司遭遇数据泄露近一年。在对集团 API 的监控操作期间，检测到有关有限客户信息（姓名、雇主、职位、电子邮件地址和电话号码）的数据泄露，此次攻击导致其 IT 系统瘫痪。

【参考链接】

https://ti.nsfocus.com/security-news/llMSq

贴身服务 加油干

绿盟科技城商行信息安全解决方案

无缝衔接 | 密切配合

THE EXPERT
BEHIND GIANTS
巨人背后的专家

多年以来，绿盟科技致力于安全攻防的研究，为金融、政府、运营商、能源、互联网以及教育、医疗等行业用户，提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。在这些巨人的背后，他们是备受信赖的专家。

NSFOCUS绿盟科技

安全月报

绿盟科技金融事业部出品

主办 / 绿盟科技金融事业部
地址 / 北京市海淀区北洼路4号益泰大厦3层
邮编 / 100089
电话 / 010-59610688-1159
传真 / 010-59610689
网站 / www.nsfocus.com
客户支持热线 / 400-818-6868
股票代码 / 300369
月报电子版下载 / https://www.nsfocus.com.cn/html/7/20/34/