零信任网络安全最佳实践

Zero Trust Network Security Best Practices

软件定义边界（SDP）

Software Defined Perimeter

安全架构技术指南

（白皮书合订本）

CSA 大中华区 SDP工作组 编著

中国云安全与新兴技术安全创新联盟

零信任网络安全最佳实践

Zero Trust Network Security Best Practices

软件定义边界（SDP）

安全架构技术指南

Technical Guide for

Software Defined Perimeter (SDP)

Security Architecture

（白皮书合订本）

CSA 大中华区 SDP 工作组 编著

中国云安全与新兴技术安全创新联盟

2019 年 8 月

关于云安全联盟 CSA

CSA（Cloud Security Alliance）云安全联盟是中立的非盈利性国际行业组织，致力于国际云计算安全的全面发展。云安全联盟的使命是“倡导使用最佳实践为云计算提供安全保障，并为云计算的正确使用提供教育以帮助确保所有其计算平台的安全”。

云安全联盟发起于 2008 年 12 月，2009 年在美国正式注册，并在当年的 RSA 大会上宣布成立。2011 年美国白宫在 CSA 峰会上宣布了美国联邦政府云计算战略，目前云安全联盟已协助美国、欧盟、日本、澳大利亚、新加坡等多国政府开展国家网络安全战略、国家身份战略、国家云计算战略、国家云安全标准、政府云安全框架、安全技术研究等工作。云安全联盟在全球拥有 4 个职业化大区实体（包括美洲区，欧洲区，亚太区，大中华区），近百个业余性地方分会，8 万位个人会员，4 百多家公司/机构会员，为业界客户们提供安全标准认证和教育培训。大中华区包括台湾，香港，澳门，北京，上海，华南，杭州，深圳分会，中国最早的分会自 2010 年成立，云安全联盟中国办事处于 2014 年 5 月在中国落地，2015 年与协调司合并，在北京，深圳，东莞等地设有办公室或工作组。

中国云安全与新兴技术安全创新联盟

中国云安全与新兴技术安全创新联盟(简称“中国云安全联盟”或“C-CSA”）挂靠在中国产学研合作促进会下，得到国务院和各部委认可，是中国第一个在安全行业全面对接国际产业和标准组织的非盈利性组织。 C-CSA 现有上百家机构会员，5 千多位个人会员，同时管理十多个地方分会。

联盟作为国际产业组织化的运营单位，与国际云安全联盟（CSA），隐私专家国际协会（IAPP）、INFOforum 信息安全论坛等国际权威安全机构合作，代表其在华运营，包括引入标准，技术，课程等先进国际安全与隐私的优秀理念，并且协助网信办等中国政府机构把国内安全政策和最佳实践介绍到国外，这使得国际安全业务在中国自主可控。C-CSA 致力将联盟发展为在国际有影响力的中国联盟，为中国在国际平台上发声。

序言：

软件定义边界 Software Defined Perimeter (SDP) 是一种具有创新性的网络安全解决方案，这种解决方案又称零信任网络 Zero Trust Network (ZTN)。SDP 或 ZTN 是基于云安全联盟 CSA 提出的理念，用安全隐身衣取代安全防弹衣保护目标，使攻击者在网络空间中看不到攻击目标而无法攻击，从而使企业或服务商的资源受到保护。

SDP 的灵感来源于中央情报局情报社区和美国国防部高度安全网络设计，因此 CSA 聘请了 CIA 原 CTO 为联盟 SDP 研究工作组组长。ZTN 灵感的最早发明者与实践者是美国微软公司，2007 年由比尔盖茨在 RSA 大会发布的微软 Anywhere Access 安全战略就是 ZTN 的实现，微软通过这项技术使公司员工甚至 Windows 使用者可以在互联网直接访问公司内网，摈弃了传统的网络边界、VPN、Firewall。

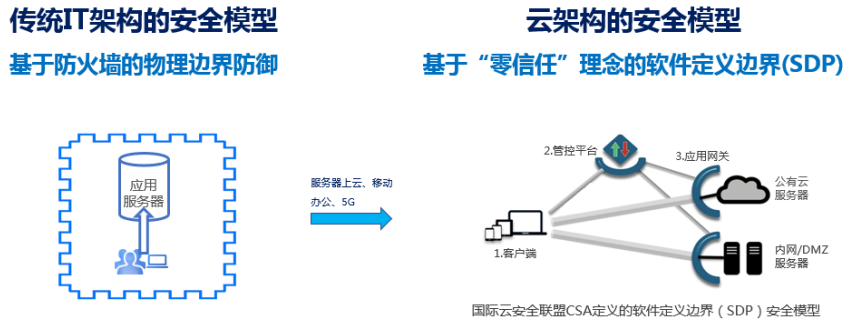
本书是 CSA 贡献给业界的一本重磅白皮书合集，它涵盖了 SDP 标准规范、设计指南与参考架构、迁移上云指南，以及业零信任网络安全先行者 Google 的 BeyondCorp 研究项目的论文合集。SDP 适用于企业网络环境、IaaS 云环境、IoT 物联网环境、BYOD 移动互联网环境等，本书不仅对 SDP 的优势与价值做了阐述，还给出了具体技术设计指导。

感谢及 CSA 大中华区 SDP 工作组专家们无私文献，特别是工作组组长陈本峰投入的大量精力，及中国云安全联盟秘书处工作人员辛勤组织和志愿者的支持。

李雨航 Yale Li
云安全联盟大中华区主席
中国云安全与新兴技术安全创新联盟执行理事长
原微软全球首席安全架构师、华为首席网络安全专家兼国际 CSO

前言：

随着云计算、移动互联网、IoT、5G 等新技术的崛起，传统的网络安全架构已经不适应时代的发展需求，一场网络安全架构的技术革命正在发生并已被越来越多的企业 CIO 们认可。过去，企业的服务器和办公设备主要运行在内网环境中，所有的企业安全都是围绕着内网的“墙”来建设的，这就是大家所熟知的基于防火墙的物理边界防御的安全模型。然而，物理边界有天然的局限性，今天的企业不可能把 IT 局限在自己的办公大楼里面：企业要拥抱云计算，但不可能把阿里云、腾讯云都装到自己的防火墙里面；企业要拥抱移动办公、IoT，也不可能把防火墙修到外边的酒店机场。因此，基于防火墙的物理边界防御模型在万物互联的新时代正在变得过时，成为企业拥抱新兴技术的障碍，企业亟需新的网络安全模型。



国际云安全联盟CSA定义的软件定义边界（SDP）安全模型

零信任网络 Zero Trust Network（ZTN）的理念于 2010 年由 Forrester 的分析师 John Kindervag 提出，给安全架构开辟了新的思路：不再默认信任物理边界内的安全性，而是始终校验用户的身份和设备的合理性/匹配性/合规性，也就是“Never Trust，Always Verify”。这种思路从某种程度上已经打破了物理边界的局限性，让用户和服务器可以分布在任何位置。然而，零信任网络只是一个理念，企业需要具体落地的解决方案。Google 内部孵化的 BeyondCorp 项目就是对零信任实践解决方案的探索。BeyondCorp 项目从 2011 年开始实施，到 2017 年对外宣布成功完成，并且已经广泛应用于大部分 Google 员工的日常办公。BeyondCorp 为业界提供了很好的零信任网络理念的架构参考，Google 也在《:login:》杂志上发表了 6 篇关于 BeyondCorp 项目实践的

论文，本书也包含了这 6 篇论文的中文翻译版。BeyondCorp 项目历经 6 年时间完成，相对而言，落地实施是非常复杂的，Google 也只是把它作为一个内部平台在使用，并没有对外商用的产品发布。目前市面上能看到的有 Google Cloud 云平台上的一个身份感知代理(Identity-Aware Proxy，即 IAP)，但它只是 BeyondCorp 其中的一个模块。

国际云安全联盟（Cloud Security Alliance，CSA）于 2014 年发布了《软件定义边界（Software Defined Perimeter，即 SDP）标准规范 1.0》。SDP 的安全理念和零信任网络 ZTN 的理念是完全一致的：

1）无论用户和服务器资源在什么位置，确保所有的资源访问都是安全的

2）记录和检查所有的流量

3）对于所有授权执行最小权限原则（Need-to-Know)

除此之外，SDP 还提出了一个创新的安全理念——网络隐身。传统的安全理念更多关注的是矛与盾之间的攻防关系，然而攻和防并不是对等的：对于攻而言，100 公里防线只要有 1 公里攻破就成功了；而防守方的角度来说，100 公里防线必须要 100%全部防住。当企业业务系统上云之后，资源暴露在公网上，7x24 小时接受来自全球各地黑客攻击，而且黑客攻击技术日新月异，软硬件安全漏洞层出不穷，一定是防不胜防的。因此，云时代的安全应该转变思路，从攻防到隐身，从穿“安全防弹衣”到穿“安全隐身衣”。敌人再高级的武器也无法攻击看不见的目标。网络隐身的安全理念更符合云时代的场景。

CSA 的《SDP 标准规范 1.0》白皮书发布之后，在业界引起了广泛的反响，美国以及以色列涌现出了一批 SDP 或 ZTN 的创业公司，行业发展如火如荼。其中比较有代表性的创业公司如 Zscaler 和 Okta，都已经在纳斯达克上市，并且市值都是在短时间内从 20 亿美金飞速发展到超过 100 亿美金，还有一些创业公司被传统的安全巨头公司以数亿美金收购，例如赛门铁克收购 Luminate，思科收购 Duo Security 等等。SDP/ZTN 行业的高速发展充分说明了其技术的先进性以及市场的光明前景。基于 SDP 标准规范的成功，CSA 又在 2017 年发布了《SDP 帮助企业安全迁移上云》白皮书以及 2019 年发布了《SDP 架构指南》白皮书，进一步对 SDP 的使用场景以及实践应用进行描述。本书包含了上述的 SDP 相关白皮书的中文版翻译。

由于 SDP/ZTN 的市场认知度日益扩大，越来越多的企业 CIO 在积极寻找 SDP/ZTN 商业化产品和落地解决方案。因此，全球知名 IT 咨询机构 Gartner 于 2019 年 4 月发布了 SDP 的市场指南《Market Guide for Zero Trust Network Access》（零信任网络访问 ZTNA 市场指南）（ID：G00386774）。指南中 Gartner 对于 SDP 市场做了如下定义：

“零信任网络访问（ZTNA），也称为软件定义边界（SDP），是围绕某个应用或某一组应用创建的基于身份和上下文的逻辑访问边界。应用是隐藏的，无法被发现，并且通过信任代理限制一组指定实体访问。在允许访问之前，代理会验证指定访问者的身份，上下文和策略合规性。这个机制将把应用资源从公共视野中消除，从而显著减少可攻击面。”

Gartner 报告还对这个市场做了如下的预测：

- 到 2020 年，80％向生态系统合作伙伴开放的数字业务应用程序将通过零信任网络访问（ZTNA）访问。
- 到 2023 年，60％的企业将淘汰大部分 VPN，转而使用 ZTNA。
- 到 2023 年，40％的企业将采用 ZTNA 用于报告中描述的其他使用场景。

可见，SDP 安全架构在国际上的普及速度是飞快的，其优势已经得到了企业 CIO 们的广泛认可，其安全性和易用性也得到了无数企业的实践验证。今天的中国市场上，云计算、移动、IoT 等新技术也正在被千千万万的企业应用，然而安全问题一直是困扰企业 CIO 们的噩梦，成为企业发展进步的阻碍。为了促进 SDP 技术在中国的落地，在中国云安全联盟的支持下 ，CSA 大中华区成立 SDP 工作组。

SDP 工作组的成立受到业界广泛的关注，行业知名的 IaaS 云厂商（如：阿里云、腾讯云等）以及安全厂商（如：深信服、奇安信、绿盟、天融信等）都委派专家参与进来。在诸多专家的积极细心的努力工作下，SDP 工作组的工作进展顺利，连续完成了多篇 SDP 相关文献的翻译编著工作，为业界贡献了宝贵的学习材料，为零信任网络安全理念、软件定义边界（SDP）安全架构在中国的落地奠定了重要的基础。

本书是对 SDP 工作组过去翻译编著的文献做一个合集，并且以合适的顺序来编排，解决 SDP 技术资料比较零散的问题。为 SDP 技术的学习者、SDP 产品开发者、SDP 安全架构实践者提供一个统一的、完整的资料合集。

本书的主要内容：

本书主要由 SDP/ZTN 的相关白皮书和论文的中文翻译稿组成，内容组织如下：

- 第一章：《SDP 标准规范 1.0》，了解什么是 SDP，其架构及协议是什么样的。
- 第二章：《SDP 架构指南》，从实践角度来阐述 SDP 的安全架构的优势、部署方式以及在企业安全体系中与其他的安全产品的关系。
- 第三章：《SDP 帮助企业安全迁移上云》，针对企业上云场景，阐述 SDP 安全架构如何与 IaaS 云环境结合。
- 第四章：《Google BeyondCrop 系列论文合集》，讨论 Google BeyondCrop 项目的实践经验。

致谢：

感谢云安全联盟 CSA 以及中国云安全联盟的大力支持，尤其是李雨航主席对工作组的指导和帮助，以及 CSA 秘书处许木娣、朱晓璐、高健凯对于工作组工作的大力协作和付出。

特别感谢奇安信身份安全实验室在《Google BeyondCrop 系列论文合集》翻译工作中与 SDP 工作组的紧密合作和全力付出。

最后，特别感谢所有 SDP 工作组成员在上述文档翻译、审校工作中的辛勤付出和努力，他们是（排名不分先后）：程长高、方伟 、李钠、刘德林、刘洪森、靳明星、马韶华 、莫展鹏、沈传宝 、孙刚、王贵宗、吴涛、杨洋、姚凯、于新宇 、余强、袁初成、张泽洲。

陈本峰 Benjamin Chen
云安全联盟大中华区 SDP 工作组组长
国家“千人计划”特聘专家
云深互联（北京）科技有限公司 创始人及 CEO

目录

第一章：《SDP 标准规范文档 1.0》

第二章：《SDP 架构指南》

第三章：《SDP 帮助企业安全迁移上云》

第四章：《Google BeyondCrop 系列论文合集》

目录

软件定义边界（SDP）工作组

SDP 标准规范 1.0

英文版 2014 年 4 月 / 中文版 2019 年 5 月

致谢

作者：

Brent Bilger

Alan

Boehme

Bob Flores

Zvi

Guterman

Mark

Hoover

Michaela

Iorga Junaid

Islam Marc

Kolenko

Juanita

Koilpilla

Gabor Lengyel

Gram Ludlow

Ted

Schroeder Jeff

Schweitzer

中文翻译版说明

由中国云安全联盟(C-CSA)秘书处组织 CSA 大中华区 SDP 工作组专家对《SDP 标准规范 1.0》(SDP_Specification_1.0)进行翻译。

参与本文档翻译的专家（排名不分先后）：

组长：陈本峰（云深互联）

组员：靳明星（易安联）、李钠（奇安信）、吴涛（华云数据）、张泽洲（奇安信）、刘洪森、王贵宗、袁初成、姚凯

C-CSA 工作人员：

朱晓璐、高健凯

关于 CSA 大中华区 SDP 工作组：

随着云计算和移动互联网的发展，传统的基于边界防御的企业安全模型已经无法适应需求，取而代之是 Software Defined Perimeter（软件定义边界，即 SDP）安全模型。目前，SDP 已经在国外逐渐被普遍采用，为了推动 SDP 在中国企业的应用，并根据本土市场需求制定出更适应中国市场的 SDP 实践指南，在中国云安全联盟的支持下 ，CSA 大中华区成立 SDP 工作组。工作组于 2019 年 3 月成立，首批参与单位有：阿里云、腾讯云、京东云、奇安信、深信服、绿盟科技、Ucloud、顺丰科技、天融信、云深互联、中宇万通、华云数据、三未信安、上元信安、安全狗、易安联、联软科技、上海云盾、缔盟云等三十多家单位。

关于 SDP 工作组更多的介绍，请点击中国云安全联盟官网 https://www.c-csa.cn/ruanjiandingyibianjieSDP.html 查看，联盟联系邮箱：info@c-csa.cn。

序言

在云环境下，应用系统不断迭代，快速更新，安全边界的防护已不再是一成不变。

在边界内部的移动设备的增长，以及应用程序资源向外部的迁移已经扩展了企业使用的传统安全模型。现有的解决方案涉及将用户回传到数据中心以进行身份验证和数据包检查，无法很好地扩展。因此需要一种新方法，使应用程序所有者能够保护公共云或私有云中的基础架构、数据中心中的服务器、甚至保护应用程序服务器内部。SDP 改变了传统的网络控制模式，原来通过网络 TCP/IP、路由做寻址，现在通过身份寻址。SDP 旨在使应用程序所有者能够在需要时部署边界，以便将服务与不安全的网络隔离开来。可以说，SDP 是在网络边界模糊和消失趋势下给资源节点提供的隐身衣，它使网络黑客看不到目标而无法发动攻击。

在中国云安全联盟支持下，CSA 大中华区完成《SDP 标准规范 1.0》的翻译工作。规范描述了云安全联盟（CSA）提议的软件定义边界（SDP）初始协议规格，旨在让更多终于从业者、供应商、推广者能够更进一步了解 SDP 的架构规范，指导日常生产工作。

在此，感谢 CSA 大中华区研究院与 C-CSA 专家委员会的专家们把白皮书翻译成中文，供大家学习。

李雨航 Yale Li

云安全联盟大中华区主席

中国云安全与新兴技术安全创新联盟执行理事长

文档声明

本文档概述了云安全联盟（CSA）为软件定义边界（Software Defined Perimeter，即 SDP）规范发起的协议，并征求讨论和改进建议。本文档的分发无限制。本文档是基于 RFC 4301 IP 安全架构构建的。

摘要

本文档描述了“软件定义边界（SDP）协议”，旨在提供按需、动态配置的安全隔离网络。安全隔离网络是与所有不安全网络隔离的可信网络，避免受到网络攻击。SDP 协议基于的工作流程是由美国国防部（DoD）发明并被一些联邦机构（Federal Agencies）使用。基于这些工作流程的网络拥有更高级别的安全性，但与传统企业网络相比，它们被认为非常难以使用。

软件定义边界（SDP）虽然基于广义的 DoD 工作流程，但已为商业用途而将其修改，使其能与现有的企业安全控制兼容。在适用的情况下，SDP 遵循 NIST 关于加密协议的指南。 SDP 可用于政府应用，例如安全访问 FedRAMP 认证的云网络以及企业应用程序，或实现对公共云的安全移动访问。

术语

Software Defined Perimeter 软件定义边界
Air-gapped networks 安全隔离网络
Initiating Hosts (IH) SDP 连接发起主机
Accepting Hosts (AH) SDP 连接接受主机
Controller SDP 控制器
Department of Defense 美国国防部
Intelligence Communities 美国情报体系
Need-to-know model 需知模型

Virtual Desktop Infrastructure 虚拟桌面基础架构

Single Packet Authorization 单包授权

Dynamical Tunnel Mode 动态隧道模式

1 介绍

本文档定义了软件定义边界（SDP）兼容系统的基础架构。协议分为两部分：一部分描述控制平面，另一部分描述数据平面。控制平面描述了 SDP 连接发起主机（IH）和 SDP 连接接受主机（AH）如何与 SDP 控制器通信。数据平面描述了 SDP 连接发起方如何与 SDP 连接接受方通信。

1.1 目标读者

本文档的目标读者是 SDP 协议的实践者。

2 设计目标

SDP 协议的设计目标是为 IPv4 和 IPv6 提供可交互操作的安全控制，包括控制器和受 SDP 连接接受方保护服务的隐藏和访问控制，以及从 SDP 连接发起方到控制器再到 SDP 连接接受方的通信机密性和完整性。

该规范提供了控制平面的协议和数据平面的一个选项。预计将为数据平面提供额外选项。

3 系统概述

本节的目标是提供协议的整体概述并定义协议中涉及的组件。详细实现请见后文。

3.1 变化的边界

纵观历史，企业通过在其数据中心部署边界安全，来抵御企业应用的外部威胁。然而，传统的边界模型正在迅速变得过时，原因有两个：

1.黑客可以轻松劫持边界内的设备（例如通过网络钓鱼攻击）并从内部攻击企业应用。此外，由于自带设备（BYOD）、外包工作人员和合作伙伴的存在，边界内部设备增多，导致漏洞不断增加。

2.除了传统数据中心，企业正在不断采用外部云计算资源，如 PaaS，IaaS 和 SaaS。因此，边界安全网络设备在拓扑上并不能很好地保护企业应用基础设施。

在边界内部的设备数量的不断增长，以及企业应用程序不断向外部的迁移，已经破坏了企业使用的传统安全模型。 现有的解决方案涉及将用户回传到数据中心以进行身份验证和数据包检查，无法很好地规模化。因此需要一种新方法，使应用程序所有者能够保护公共云或私有云中的基础架构、数据中心中的服务器、甚至保护应用程序服务器内部。

3.2 SDP 概念

SDP 旨在使应用程序所有者能够在需要时部署安全边界，以便将服务与不安全的网络隔离开来。SDP 将物理设备替换为在应用程序所有者控制下运行的逻辑组件。SDP 仅在设备验证和身份验证后才允许访问企业应用基础架构。

SDP 背后的原理并不是全新的。美国国防部（DoD）和美国情报体系（IC）内的多个组织已经实施了在网络访问之前进行身份验证和授权的类似网络架构。通常在分类或高端网络中使用（由美国国防部定义），每个服务器都隐藏在远程访问网关设备后面，在授权服务可见且允许访问之前

用户必须对其进行身份验证。 SDP 采用分类网络中使用的逻辑模型，并将该模型整合到标准工作流程中（第 2.4 节）。

SDP 保持了上述“需知模型”的优点，并去除了需要远程访问网关设备的缺点。在获得对受保护服务器的网络访问之前，SDP 要求发起方进行身份验证并首先获得授权，然后在请求系统和应用程序基础架构之间实时创建加密连接。

3.3 SDP 架构

SDP 的体系结构由两部分组成：SDP 主机和 SDP 控制器。SDP 主机可以发起连接或接受连接。这些操作通过安全控制通道与 SDP 控制器交互来管理（请参见下页的图 1）。因此，在 SDP 中，控制平面与数据平面分离以实现完全可扩展的系统。 此外，为便于扩展与保证正常使用，所有组件都可以是多个实例的。

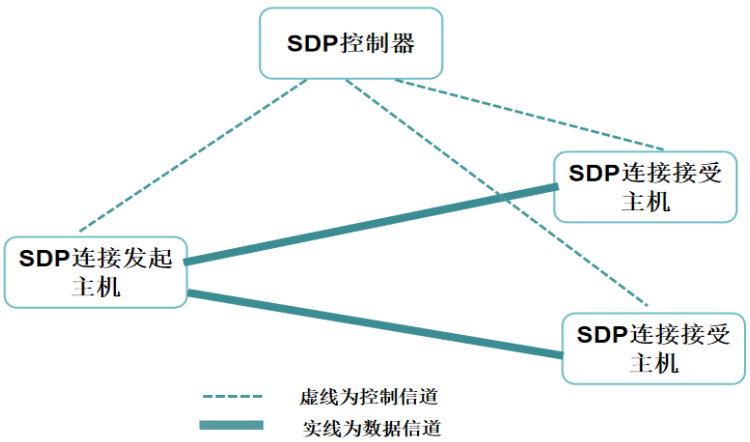


图 1：SDP 架构由两大组件组成：SDP 主机以及 SDP 控制器

3.3.1 SDP 控制器

SDP 控制器确定哪些 SDP 主机可以相互通信。SDP 控制器可以将信息中继到外部认证服务，例如认证地理位置和/或身份服务器。

3.3.2 SDP 连接发起主机（Initiating Host,即 IH）

SDP 连接发起主机（IH）与 SDP 控制器通信以请求它们可以连接的 SDP 连接接受方（AH）列表。 在提供任何信息之前控制器可以向 SDP 连接发起主机请求诸如硬件或软件清单之类的信息。

3.3.3 SDP 连接接受主机（Accpeting Host，即 AH）

默认情况下 SDP 连接接受主机（AH）拒绝来自 SDP 控制器以外的所有主机的所有通信。 只有在控制器指示后，SDP 连接接受主机才接受来自 SDP 连接发起主机的连接。

3.4 SDP 工作流

SDP 工作流如下：

1.一个或多个 SDP 控制器上线并连接至适当的可选认证和授权服务（例如，PKI 颁发证书认证服务、设备验证、地理定位、SAML、OpenID、Oauth、LDAP、Kerberos、多因子身份验证等服务）。

2.一个或多个 SDP 连接接受主机（AH）上线。这些主机连接到控制器并由其进行身份验证。但是，他们不会应答来自任何其他主机的通信，也不会响应非预分配的请求。

3. 每个上线的 SDP 连接发起主机（IH）都与 SDP 控制器连接并进行身份验证。

4.在验证 SDP 连接发起主机（IH）之后，SDP 控制器确定可授权给 SDP 连接发起主机（IH）与之通信的 SDP 连接接受主机（AH）列表。

5.SDP 控制器通知 SDP 连接接受主机（AH）接受来自 SDP 连接发起主机（IH）的通信以及加密通信所需的所有可选安全策略。

6.SDP 控制器向 SDP 连接发起主机（IH）发送可接受连接的 SDP 连接接主机（AH）列表以及可选安全策略。

7.SDP 连接发起主机(IH)向每个可接受连接的 SDP 连接接受主机(AH)发起单包授权，并创建与这些 SDP 连接接受主机（AH）的双向 TLS 连接。

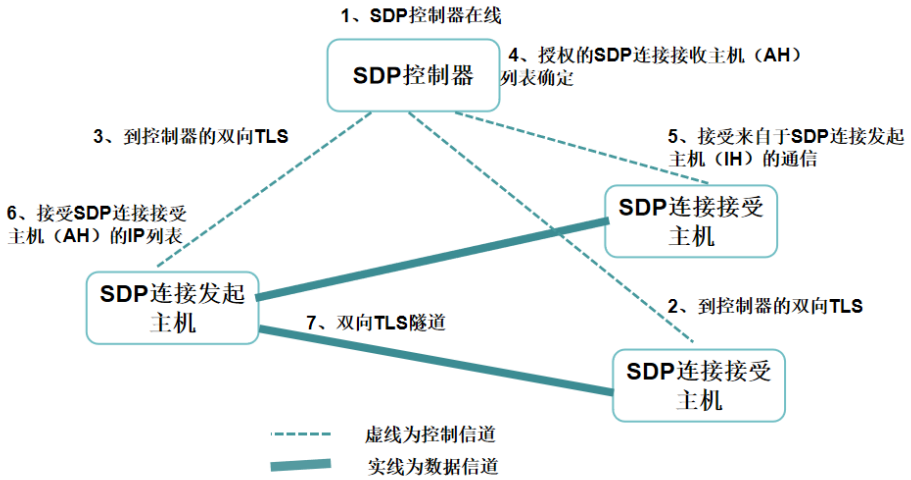


图 2: SDP 架构的工作流展示了控制平面和数据平面的隔离

3.5 SDP 协议实现

虽然所有 SDP 实施方案都保持相同的工作流，但是针对不同的应用场景会有不同的实现方式。

3.5.1 客户端—网关模型

在客户端—网关的实施模型中，一个或多个服务器在 SDP 连接接主机（AH）后面受到保护，这样，SDP 连接接受主机（AH）就充当客户端和受保护服务器之间的网关。此实施模型可以在企业网络内执行，以减轻常见的横向移动攻击，如服务器扫描、操作系统和应用程序漏洞攻击、中间人攻击、传递散列和许多其他攻击。或者，它可以在 Internet 上实施，将受保护的服务器与未经授权的用户隔离开来，并减轻诸如拒绝服务（DoS）、SQL 注入、操作系统和应用程序漏洞攻击、中间人攻击、跨站点脚本（XSS）、跨站点请求伪造（CSRF）等攻击。

3.5.2 客户端—服务器模型

客户机到服务器的实施在功能和优势上与上面讨论的客户机到网关的实施相似。然而，在这种情况下，受保护的服务器将运行可接受连接主机（AH）的软件，而不是位于运行该软件的服务器前面的网关。客户机到网关实施和客户机到服务器实施之间的选择通常基于受保护的服务器数量、负载平衡方法、服务器的弹性以及其他类似的拓扑因素。

3.5.3 服务器—服务器模型

在服务器到服务器的实施模型中，可以保护提供代表性状态传输（REST）服务、简单对象访问协议（SOAP）服务、远程过程调用（RPC）或 Internet 上任何类型的应用程序编程接口（API）的服务器，使其免受网络上所有未经授权的主机的攻击。例如，对于 REST 服务，启动 REST 调用的服务器将是 SDP 连接发起主机（IH），提供 REST 服务的服务器将是可以接受连接的主机（AH）。为这个用例实施一个软件定义边界可以显著地减少这些服务的负载，并减轻许多类似于上面提到的攻击。这个概念可以用于任何服务器到服务器的通信。

3.5.4 客户端—服务器—客户端模型

客户端到服务器到客户端的实施在两个客户端之间产生对等关系，可以用于 IP 电话、聊天和视频会议等应用程序。在这些情况下，软件定义边界会混淆连接客户端的 IP 地址。作为一个微小的变化，如果用户也希望隐藏应用服务器，那么用户可以有一个客户端到客户端的配置。

3.6 SDP 应用

软件定义边界可以保护所有类型的服务器免受基于网络的攻击。下面介绍了一些更有趣的软件定义边界应用场景。

3.6.1 企业应用隔离

对于涉及知识产权，财务信息，人力资源数据以及仅在企业网络内可用的其他数据集的数据泄露，攻击者可能通过入侵网络中的一台计算机进入内部网络，然后横向移动获得高价值信息资产的访问权限。 在这种情况下，企业可以在其数据中心内部署 SDP，以便将高价值应用程序与数据中心中的其他应用程序隔离开来，并将它们与整个网络中的未授权用户隔离开来。 未经授权的用户将无法检测到受保护的应用程序，这将减轻这些攻击所依赖的横向移动。

3.6.2 私有云和混合云

除了有助于保护物理机器，SDP 的软件覆盖特性使其可以轻松集成到私有云中，以利用此类环境的灵活性和弹性。 此外，企业可以使用 SDP 隔离隐藏和保护其公共云实例，或者作为包含私有云和公共云实例和/或跨云集群的统一系统。

3.6.3 软件即服务（SaaS）

软件即服务(SaaS)供应商可以使用 SDP 架构来保护他们提供的服务。在这种应用场景下，SaaS 服务是一个 SDP 连接接受主机（AH），而所有连接服务的终端用户就是 SDP 连接发起主机（IH）。 这样使得 SaaS 产商可以通过互联网将其服务提供给全球用户的同时不再为安全问题担忧。

3.6.4 基础设施即服务（IaaS）

基础设施即服务（IaaS）供应商可以为其客户提供 SDP 即服务作为受保护的入口。 这使他们的客户可以充分利用 IaaS 的灵活性和性价比，同时减少各种潜在的攻击。

3.6.5 平台即服务（PaaS）

平台即服务（PaaS）供应商可以通过将 SDP 架构作为其服务的一部分来实现差异化。 这为最终用户提供了一种嵌入式安全服务，可以缓解基于网络的攻击。

3.6.6 基于云的虚拟桌面基础架构（VDI）

虚拟桌面基础架构（VDI）可以部署在弹性云中，这样 VDI 的使用按小时支付。 然而，如果 VDI 用户需要访问公司网络内的服务器，VDI 可能难以使用，并且可能会产生安全漏洞。 但是，VDI 与 SDP 相结合，可通过更简单的用户交互和细粒度访问解决了这两个问题。

3.6.7 物联网（IoT）

大量的新设备正在连接到互联网上。管理这些设备或从这些设备中提取信息抑或两者兼有的后端应用程序的任务很关键，因为要充当私有或敏感数据的保管人。软件定义边界可用于隐藏这些服务器及其在 Internet 上的交互，以最大限度地提高安全性和正常运行时间。

3.6.8 SDP 与 IKE/IPsec 和 TLS 的关系

如前面部分所述，SDP 可以使用 IKE / IPsec 和 TLS 等协议在 SDP 连接发起主机（IH）和 SDP 连接接受主机（AH）之间创建 VPN。但是，SDP 与 VPN 不同。它们之间的差异概述如下：

- 与受 VPN 网关保护的服务器相比，创建受 SDP 保护的服务器需要不同的工作量。在 SDP 情况下，一旦 SDP 控制器上线，用户可以通过软件设置，根据需要创建尽可能多的受保护服务器，并且可以通过 LDAP 关联区分授权用户和未授权用户。
- 与 SDP 相比，设置 VPN 网关以保护单个服务器的资本和运营成本更高。 SDP 是可以部署在云环境中的软件架构。
- SDP 可以同时用于安全和远程访问，而 VPN 网关则不能。如果要尝试在企业内使用 VPN 客户端和 VPN 网关来保护某个服务器，则用户无法使用远程访问 VPN 来访问服务器（因为 VPN 客户端已连接到远程访问 VPN 网关）然而 SDP 通信则可以在远程访问 VPN 之上进行。
- SDP 可防止 DDoS 攻击，而 VPN 网关则不会。 SDP 连接接受方可以部署在与其保护的应用服务器不同的拓扑不同的位置，甚至从而对授权用户隐藏真实位置。

4 术语汇编

本文档使用以下术语:

- SDP 连接接受主机（AH）

  在控制器验证并授权连接后，SDP 连接接受方接受来自 SDP 连接发起方的通信。

- 代理 ID（Agent ID）

  代理 ID（AID）是一个 32 位唯一无符号值，用于标识指定的 SDP 连接发起主机（IH）和/或 SDP 连接接受主机（AH）。 它主要用于单数据包授权。

- SDP 连接接受主机-控制器路径

  SDP 连接接受主机—控制器路径是指每个 SDP 连接接受主机（AH）和控制器之间通信的信道。

- SDP 连接接受主机（AH）会话

  SDP 连接接受方会话是 SDP 连接接受主机（AH）连接到控制器的特定时间段。

- SDP 连接接受主机（AH）会话 ID

  由控制器管理的 256 位随机化 NONCE，用于指代特定的 SDP 连接接受方（AH）会话。

- 动态隧道模式（Dynamical Tunnel Mode，即 DTM）

  动态隧道模式（DTM）是 SDP 连接发起主机（IH）与一个或多个 SDP 连接接受主机（AH）通信的建议协议和封装。 预计替代协议将被提出。

- SDP 连接发起主机（IH）

  SDP 连接发起方（IH）是发起与控制器和 SDP 连接接受方的通信的主机。

- SDP 连接发起主机（IH）会话

  SDP 连接发起主机会话是 SDP 连接发起主机（IH）连接到控制器的特定时间段。

- SDP 连接发起主机（IH）会话 ID

  由控制器管理的 256 位随机化 NONCE，用于指代特定的 SDP 连接发起主机（IH）会话。

- Mux ID

  64 位 Mux ID（MID）用于在动态隧道模式下在单个 SDP 连接发起主机（IH）-SDP 连接接受主机（AH）隧道上复用连接。 其中最重要的 32 位组成了控制器为每个远程服务分配的唯一值， 它被称为 MID 的服务 ID。剩余 32 位形成由 SDP 连接发起主机（IH）和 SDP 连接接受主机（AH）维护的值，以区分特定远程服务的不同 TCP 连接。 这被称为 MID 的会话 ID。

- 服务

  服务是指受 SDP 连接接受主机（AH）保护的应用程序及其关联数据。

- 服务 ID

  服务 ID 是 MUX ID 的最重要的 32 位。

- 会话 ID

  会话 ID 是 MUX ID 的次重要的 32 位。

- 单包授权一次性密码(SPA OTP)

基于 RFC4226 的单包授权（Single Packet Authorization，即 SPA），但修改后包含计数器值（见下文）。它作为唯一标识用于在向控制器和可接受连接主机（AH）发起通信时辨认 SDP 连接发起主机（IH）。

5 SDP 协议

下面将解释 SDP 协议。如下面图 3 描述的软件定义边界体系结构：

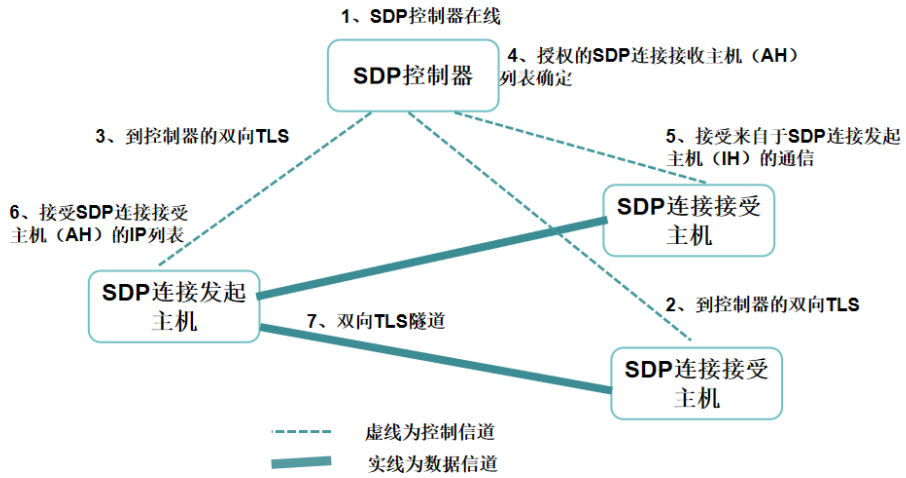


图 3： SDP 架构

5.1 服务启动

一个或多个 SDP 控制器、SDP 连接发起主机 IHs、SDP 连接接受主机 AHs 的服务启动方法不在本文档的讨论范围之内。典型的方法是通过 CHEF 或 PUPPET 或其它主机托管服务（例如，RightScale、AWS CloudFormation 等。）

5.2 单包授权（SPA）

单包授权（SPA）用于在以下所有情况下启动通信：IH 控制器、AH 控制器和 IH-AH。SPA 为受 SPA 保护的服务器提供以下安全作用。

- **保护服务器：**在提供真正的 SPA 之前，服务器不会响应来自任何客户端的任何连接。
- **缓解对 TLS 的拒绝服务攻击：**面向 Internet 的运行 HTTPS 协议的服务器极易受到拒绝服务（DoS）攻击。SPA 可以缓解

这些攻击，因为它允许服务器在进入 TLS 握手之前放弃 TLS DoS 尝试。

- **攻击检测：**从任何其他主机发送到 AH 的第一个数据包必须是 SPA。如果 AH 接收到任何其他数据包，则应将其视为攻击。因此，SPA 使得 SDP 可以根据一个恶意数据包来检测到攻击。

SPA 基于 RFC 4226（HOTP）标准，参数如下：

- **客户端：**RFC4226 使用术语“客户端”指 SPA 包的生成器。在 SDP 的架构中，客户机是 IH 或 AH。
- **服务器：**RFC4226 使用术语“服务器”来指 SDP 架构中 SPA 包的验证者，在这里服务器指是控制器或 AH。
- **种子：**种子是指每个通信双方（即 IH-Controller、AH-Controller 和 IH-AH）之间共享的 32 位无符号整数。种子必须保密。
- **计数器：**计数器是一个 64 位无符号整数，必须在通信双方之间同步。在 RFC4226 中，这是通过“前瞻窗口”完成的（因为 RFC4226 的典型用例是硬件 OTP 令牌）。但是，对于 SDP 协议来说，计数器可以在 SDP 包中发送。从而避免了对前瞻窗口的需求以及通信双方不同步的可能性。注意计数器不需要被保密。
- **密码：**由 RFC4226 加密算法生成的 HOTP 值。
- **密码长度：**密码长度固定为 8 位。

对于 SPA 协议来说，单个 SPA 包从客户端发送到服务器，服务器不需要回复。数据包的格式为：

IP	TCP	AID (32-bit)	Password (32-bit)	Counter (64-bit)

在接收到数据包后，服务器必须允许客户端通过端口 443 上的双向 TLS 进行连接。

5.3 双向 TLS 或者 IKE

在进一步的设备验证和/或用户身份验证之前，需要先保证所有主机之间的连接必须使用带有相互身份验证的 TLS 或互联网密钥交换（IKE），以将该设备验证为 SDP 的授权设备。所有弱密码套件和不支持相互身份验证的套件都必须被禁止。

TLS（IPsec）客户端和服务器的根证书必须绑定到已知的合法根证书，并且不应该由大多数用户浏览器信任的数百个根证书组成，这可以避免伪装者攻击（即攻击者可以通过被攻陷的证书颁发机构 CA 伪造证书）。根证书安装到到 IH、AH 和控制器的方法不在本文档讨论范围之内。典型的方法是通过 Chef 或 Puppet 或其托管服务等类似方法（例如，RightScale、AWS CloudFormation 等）。

TLS（IPSec）服务器应使用 IETF 工作草案《X.509v3 扩展：OCSP 连接所需的 draft-hallambaker-muststaple-00》中定义的 OCSP 响应连接（OCSP response stapling），该草案引用了 RFC 4366《传输层安全性（TLS）扩展》中的连接实现。OCSP 响应连接可以减少对 OCSP 响应的 DoS 攻击，还可以有效防止服务器证书吊销前因过时 OCSP 响应产生的中间人攻击。

5.4 设备验证

双向 TLS（IKE）证明了请求访问 SDP 的设备具有一个未过期且未被吊销的私钥，但它不证明该密钥未被窃取。设备验证的目的是证明适当的设备拥有私钥，并且设备上运行的软件是可信的。在 SDP 中控制器默认是受信任的设备（因为它存在于最受控制的环境中），而 IHs 和 AHs 必须经过其验证。设备验证减轻了用户账密被盗和由此产生的伪装者攻击。设备验证超出了此版本 SDP 文档的范围，但将在未来版本中阐述。

5.5 AH-控制器协议

以下小节定义了在 AH 和控制器之间传递的各种消息及其格式。基本协议的形式如下：

命令（8 字节）	命令特定数据（命令特定长度）

5.5.1 登录信息请求

AH 向控制器发送登录请求消息，以指示它是可用的，并且能够接受来自控制器的其他消息：

0x00	无命令特定数据

5.5.2 登录响应信息

控制器发送登录响应消息，验证登录请求是否成功，如果成功，则提供 AH 会话 ID。

0x01	状态码（16 位）	AH 会话 ID（256 位）

5.5.3 登出信息请求

登出请求消息由 AH 发送至控制器，用于表示 AH 不再提供服务，不再接收来自控制器的其他消息了。本消息无需响应。

0x02	无命令特定数据

5.5.4 保活信息

Keep-Alive 消息由 AH 或控制器发出，表示其仍处于激活状态。

0x03	无命令特定数据

5.5.5 AH 服务信息

服务消息由控制器发送至 AH，用于通告 AH 所保护的服务列表。

0x04	JSON 格式定义的服务的数组

JSON 规范如下：

格式	实例
{“services”: [“port”: <Server port>, “id”: <32-bit Service ID>, “address”: <Server IP>, “name”: <service name>] }	{“services”: [“port”: “443”, “id”: “123445678”, “address”: “100.100.100.100”, “name”: “SharePoint”] }

5.5.6 认证完成信息

IH 认证完成消息由控制器发送至 AH，通知 AH 一个新的 IH 已经验证通过，AH 应当允许此 IH 访问指定的服务。

0x05	JSON 格式定义的 IH 信息的数组

JSON 规范如下：

{“sid

: <256-bit IH Session ID>, “seed”: <32-bit SPA seed>, “counter”: <32-bit SPA counter> [“id”:<32-bit Service ID>

]

}

5.5.7 保留的自定义信息

命令（0xff）保留用于 AH 和控制器之间的任意非标准消息。

0xff	用户自定义

5.6 AH 到控制器的序列图

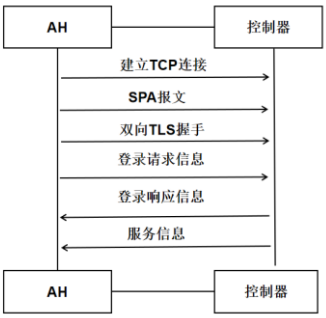


图 4. SDP 连接接受主机（AH）连接至控制器

AH 连接至控制器的协议序列图如下所示。本序列图只描述了初始登录阶段的消息交互。IH 连接至控制器的消息交互会在 IH-控制器序列图中描述。

5.7 IH-控制器协议

本章节定义 IH 和控制器之间传输的各种消息及其格式，基本协议格式如下：

命令（8 位）	特定长度的命令特定数据

5.7.1 登录请求信息

登录请求消息由 IH 发送至控制器，用于表示 IH 服务已经就绪并希望加入 SDP。

0x00	无命令特定数据

5.7.2 登录响应信息

登录响应消息由控制器发至 IH，用于表示登录请求成功与否，若成功，同时提供 IH 会话 ID。

0x01	状态码（16 位）	IH 会话 ID（256 位）

5.7.3 登出请求消息

登出请求消息由 IH 发送至控制器，用于表示 IH 将退出 SDP。本消息无需响应。

0x02	无命令特定数据

5.7.4 Keep-Alive 保活信息

Keep-Alive 消息由 IH 或控制器发出，表示其仍处于激活状态

0x03	无命令特定数据

5.7.5 IH 服务信息

服务消息由控制器发送至 IH，用于通告 IH 可用的服务列表以及保护服务的 AH 的 IP 地址列表。

0x06	JSON 格式定义的服务的数组

JSON 规范如下：

格式	示例
{"services": [{"address" : <AH IP>, "id": <32-bit Service ID>, "name": <service name>, “type” : <service type>[1] }] }	{"services": [{"address" : “200.200.200.200”, "id": “12345678”, "name": “SharePoint”, “type” : “https” }] }

5.7.6 保留的自定义消息

该命令（0xff）保留用于 IH 和控制器之间的任意非标准消息。

0xff	用户自定义

5.8 IH 到控制器序列图

IH 连接至控制器的协议序列图如下所示

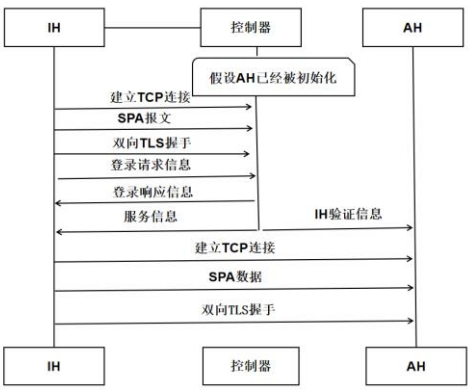


图 5. SDP 连接发起主机（IH）连接至控制器及 SDP 连接接受主机（AH）

[1] 类型（Type）是用来区分不同的服务的连接方式。例如：HTTP 是一种服务类型，而 HTTPS 也是一种服务类型。

5.9 动态隧道模式（DTM）下的 IH-AH 协议

本节定义了 DTM 模式下 IH 与 AH 的传输消息及格式

命令（8 位）	命令特定长度的命令特定数据

5.9.1 保活消息

Keep-Alive 消息由 IH 或 AH 发出，表示其仍处于激活状态。

0x03	无命令特定数据

5.9.2 建立连接请求消息

该消息由 IH 向 AH 发出，表示将建立特定服务的连接

0x07	Mux ID 64 位

5.9.3 建立连接响应消息

该消息由 AH 向 IH 发出，表示建立连接请求是否成功

0x08	状态码 16 位	Mux ID 64 位

5.9.4 数据消息

该消息由 IH 或 AH 发出，用来在打开的连接上推送数据，该消息没有响应。

0x09	数据长度 16 位	Mux ID 64 位

5.9.5 连接关闭信息

该消息由 AH 发出表示 AH 已经关闭连接，由 IH 发出表示请求关闭连接，该消息没有响应。

0x0A	Mux ID (64 位)

5.9.6 自定义信息

该消息表示 IH 和控制器之间的任意非标准消息。

0xff	自定义

5.10 IH 到时序图（示例）

IH 和 AH 之间的示例协时序图如下图所示，这个序列图只描述与初始登录相关联的消息序列。IH 连接到控制器时发送的消息显示在 AH 到控制器的时序图中。

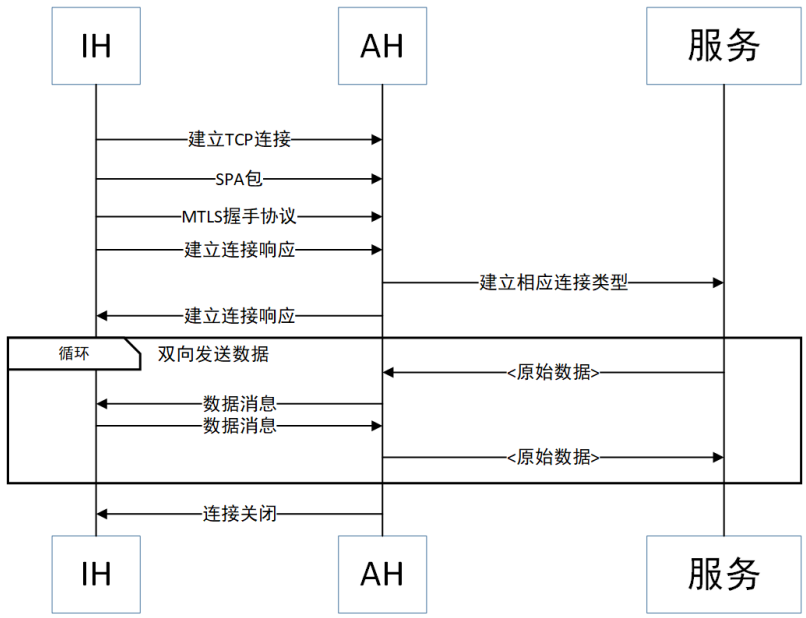


图 5: SDP 连接发起主机（IH）与 SDP 连接接受主机（AH）建立连接并且向服务发送数据

5.11 控制器确定 IH 可连接 AH 列表

控制器确定 IH 可连接的 AH 列表的方法不在本地协议规范文档中描述。如果这是一个物联网应用，列表可能是静态的或是基于连接到软件定义的边界的软件类型确定的。如果这是一个服务器到服务器的应用程序，列表可能来自一个受保护的数据库服务器。如果这是一个客户端/服务器应用程序，列表可能来自 LDAP 服务器。其他的应用程序可能以其他方式确定该列表。

6 日志

所有系统都要求通过创建日志确定服务可用性和性能以及服务器的安全性。

6.1 日志信息字段

所有日志应该包括以下字段：

字段名称	含义
时间	日志记录发生的时间
名称	事件的可读名称。注意：不包括任何可变的数据段，例如用户名、ip 地址、主机名等。日志记录的额外字段已经包含这些信息，我们不想重复。
严重程度	该事件从 debug 到 critical 的严重程度（见下文）
设备地址	创建日志记录的机器的 IP 地址

6.2 操作

下面是一个需要记录日志的操作用例或活动的清单。

签名符（signature_id） 是一个标识符，能够确定事件的类型。第三列包含需要记录特定日志消息的额外字段。

活动（activity）	签名符（signature_id）	**需要记录的数据/信息**
组件启动、关闭、重启（例如控制器启动，主机重启）	*ops:startup* *ops:shutdown* *ops:restart*	***原因：***说明为什么会发生重启或关 ***组件：***说明哪个组件受影响
组件之间的连接(控制器、IH、AH、第三方组件、DB)上线、下线、重新连接	*ops:conn:up* *ops:conn:dow* *ops:conn:reco* *ne* *ct*	***src：***连接源地址，报告主体可见的地址 ***dst：***连接目的地址，报告主体可见的 ip 地址 ***reconnect_count：***记录有多少次接尝试 ***原因：***说明为什么沟通中断

举个简单例子来描述了一个完整的故障场景发生时什么日志条目记录在什么地方。在这个场景中，我们假设一个控制器关机:

1. 控制器下线 [没有日志，失效组件不能记录日志]
2. IH 多次试图连接控制器
   记录 **ops:conn:reconnect** 日志信息
3. 多次尝试后，客户端声称到控制器的连接中断，并寻找新的控制器
   记录 **ops:conn:down** 日志信息，严重程度是 error
4. IH 连接到新发现的控制器
   记录 **ops:conn:up** 日志信息
5. 如果没有其他控制器
   记录 **ops:conn:down** 日志信息，严重程度是 **critical**

另一个类似的情况是一个客户端掉线并且没有发出警告（如笔记本电脑关机）。在这种情况下，控制器和 AH 都检测到失败的连接。每个设备都将记录 **ops:conn:down** 日志信息，严重程度是 **error**。

6.3 安全性

安全日志是 SDP 的核心，同时对于检测更广泛的大规模基础设施攻击方面也至关重要。因此，当这些日志被发送到 SIEM 系统时，它们的价值就变得极高。

签名符（signature_id）作为一个标识符，用于标识不同的事件类型。第三列中的包含了一些特定日志信息需要记录的额外域值。

动作（activity）	签名符（signature_id）	需要记录的数据/信息
AH 登录成功	sec:login	***src***：AH 的控制器可见的 IP 地址 ***AH Session ID*** ： AH 的会话 ID
AH 登录失败		
IH 登录成功	sec:login	***src***：IH 的控制器可见的 IP 地址 ***IH Session ID*** ： IH 的会话 ID
IH 登录失败		
组件认证(例如：IH->控制器)		
拒绝接入请求	*sec:fw:denied*	src： 尝试连接的源地址 dst： 尝试连接的目的地址

下面是一个完整的用户登录过程的日志例子（IH 向 AH 发起连接）：

1. IH 向控制器请求连接
记录 ops:conn:up 日志信息

2. IH 和控制器相互验证
记录 sec:auth 日志信息

3. IH 向 AH 请求连接

记录 ops:conn:up 日志信息

4. IH 和控制器相互验证
记录 sec:auth 日志信息

6.4 性能

性能信息的差异通常不适合采用传统的日志方式来记录。大量的衡量指标可能使得日志系统崩溃死机，而且分析系统的设计初衷也不是用于处理类似信息的。因此，我们建议提供一个独立的关于性能日志处理系统。

6.5 合规性

如果日志规范遵守得当，所有的合规性要求诸如 PCI（Payment Card Industry，支付卡行业数据安全标准）、SOX（萨班斯法案），就变得简单了。比如说，SOX 中要求记录所有针对财务系统的特权访问，甚至记录任何可能对于财务系统状态或结果造成影响的行为。当我们覆盖了关于“安全”部分的所有的用例时候，我们就已经覆盖登录用例的合规性。

6.6 安全信息和事件管理集成性（SIEM）

我们建议把所有的安全事件推送到一个特定的 SIEM 系统之上。这样就可以帮忙 SIEM 系统生成网络安全态势的整体画像。因为 SDP 安全日志作为画像组成部分，使得对于环境的可视化和可感知性得到了提升。

操作日志记录可以被用于管理产品可用性和性能。这个信息对于离开 SDP 的边界的环境是作用不大，但是我们建议用户可以指定把相关的日志转发到中央控制台（比如 SIEM 系统）。如果有用户从这些信息中获取到一定有价值的内容，他们自然就会来按照这种方式操作。

7 SDP 标准规范

本文档描述了云安全联盟（CSA）提议的软件定义边界（SDP）协议的初始规范。本文档的分发不受限制。

软件定义边界

架构指南

致谢

云安全联盟（CSA）对所有为制定本指南做出贡献和提供支持的人员表示感谢。

主要作者

Jason Garbis

Juanita Koilpillai

贡献者

Junaid Islam

Nya Murray

Aaron Palermo

Preeta Raman

Michael Roza

云安全联盟员工

Shamun Mahmud

中文翻译版说明

由中国云安全联盟(C-CSA)秘书处组织 CSA 大中华区 SDP 工作组专家对《SDP 架构指南》(SDP_Architecture_Guide)进行翻译。

参与本文档翻译的专家（排名不分先后）：

组长：陈本峰（云深互联）
组员：程长高（安全狗）、靳明星（易安联）、李钠（奇安信）、吴涛（华云数据）、袁初成（缔安科技）、余强（中宇万通）、刘德林、刘洪森、孙刚、王贵宗、杨洋、姚凯

关于 CSA 大中华区 SDP 工作组：

随着云计算和移动互联网的发展，传统的基于边界防御的企业安全模型已经无法适应需求， 取而代之是 Software Defined Perimeter（软件定义边界，即 SDP）安全模型。目前，SDP 已经在国外逐渐被普遍采用。为了推动 SDP 在中国企业的应用，并根据本土市场需求制定出更适应中国市场的 SDP 实践指南，在中国云安全联盟的支持下 ，CSA 大中华区成立 SDP 工作组。工作组于 2019 年 3 月成立，首批参与单位有：阿里云、腾讯云、京东云、奇安信、深信服、绿盟科技、UCloud、顺丰科 技、天融信、云深互联、中宇万通、华云数据、三未信安、上元信安、安全狗、易安联、联软科技、上海云盾、缔盟云、缔安科技等三十多家单位。

关于 SDP 工作组更多的介绍，请点击中国云安全联盟官网 https://www.c-csa.cn/ruanjiandingyibianjieSDP.html 查看，联盟联系邮箱：info@c-csa.cn

序言

软件定义边界 Software Defined Perimeter (SDP) 是一种具有创新性的网络安全解决方案，这种解决方案又称零信任网络 Zero Trust Network (ZTN)。

SDP 或 ZTN 是基于云安全联盟 CSA 提出的理念，用安全隐身衣取代安全防弹衣保护目标，使攻击者在网络空间中看不到攻击目标而无法攻击，从而使企业或服务商的资源受到保护。

SDP 的灵感来源于中央情报局情报社区和美国国防部高度安全网络设计，因此 CSA 聘请了 CIA 原 CTO 为联盟 SDP 研究工作组组长。ZTN 灵感的最早发明者与实践者是美国微软公司，2007 年由比尔盖茨在 RSA 大会发布的微软 Anywhere Access 安全战略就是 ZTN 的实现，微软通过这项技术使公司员工甚至 Windows 使用者可以在互联网直接访问公司内网，摒弃了传统的网络边界 VPN、Firewall。

本白皮书是 CSA 贡献给业界的又一篇重磅白皮书，它是 SDP 规范之后的设计指南与参考架构，适用于企业网络环境、IaaS 云环境、IoT 车联网环境、BYOD 移动互联网环境等，不仅对 SDP 的优势与价值做了阐述，还给出了具体技术设计指导。

我代表 CSA 对大中华区参与此项翻译工作的专家们表示由衷的感谢，特别是工作组组长陈本峰投入的大量精力，及 CSA 志愿工作者们的支持。

李雨航主席 Yale LI

CSA 云安全联盟大中华区

中国云安全与新兴技术安全创新联盟执行理事长

介绍

SDP 方案结合了技术和架构组件，可以比传统的安全工具更有效、更高效地保护网络应用程序和基础架构。

当今的网络安全体系结构、工具和平台无法应对当前安全威胁带来的挑战。无论您是在阅读主流媒体的头条新闻，还是作为网络防御者进行日常工作，或者您是安全供应商，这些潜在安全威胁都可能会影响到您。各种来源的持续攻击会影响商业企业、政府组织、关键基础设施等。

现在是时候让我们信息安全行业拥抱创新的网络安全工具，即软件定义边界（SDP）技术，将其应用于所有的网络层。SDP 方案结合了技术和架构组件，已经证明可以比传统的安全工具更好地保护网络应用程序和基础架构。由云安全联盟 CSA 于 2014 年 4 月发布的“SDP 规范 1.0”概述了 SDP 技术的基础知识：

“SDP 背后的原理并非全新。美国国防部（DoD）和美国情报体系（IC）内的多个组织在网络访问之前已经实施了基于认证和授权的类似网络架构。通常用于机密或高端网络（由国防部定义），每台服务器隐藏在远程访问网关设备后面，用户必须先通过该设备身份验证，才能查看授权服务并进行访问。SDP 利用分类网络中使用的逻辑模型，并将该模型纳入标准工作流程中。在获得对受保护服务器的网络访问之前，SDP 要求端点进行身份验证并首先获得授权。然后在请求系统和应用程序基础架构之间实时创建加密连接。[14]”

14 https://downloads.cloudsecurityalliance.org/initiatives/sdp/SDP_Specification_1.0.pdf

目的

作为一个由安全从业者和解决方案提供商组成的组织，我们对信息安全和网络安全充满热情。我们相信 SDP 是一个重要的创新解决方案，可以应对我们所有人面临的安全威胁。

自“SDP 规范 1.0”发布以来，我们作为一个由软件供应商、系统、安全架构师和企业组成的工作组，已经构建并部署了许多符合这些准则的系统。同时，我们了解了很多关于 SDP 实现的知识-特别是在缺乏原始规范的领域。

通过本指南，我们将帮助企业和从业人员获取有关 SDP 的信息；展示其可提供的经济和技术效益；并帮助用户在其组织中成功实施 SDP。如果实现以下目标，我们将认为此文档是成功的:

- 提高 SDP 的市场认知度、可信度和企业采用率
- 提高人们对 SDP 在不同环境中的应用的理解
- 提升企业使用 SDP 解决问题的动机
- 使用本文档向内部业务相关者介绍 SDP
- 企业根据本白皮书中的体系结构建议成功部署 SDP 解决方案。

受众目标

文中的信息将使考虑或正在组织机构中实施 SDP 项目的团队在安全性、体系结构和技术网络中受益。

主要受众包括从事信息安全、企业架构和安全合规角色的专业人员。这些人员主要负责 SDP 解决方案的评估、设计、部署和运营。

此外，作为解决方案提供商、服务提供商和技术供应商的人员也将从本文提供的信息中获益。

概述

软件定义边界（SDP）简介

SDP 旨在利用基于标准且已验证的组件，如数据加密、远程认证（主机对远程访问进行身份验证）、传输层安全（TLS，一种加密验证客户端信息的方法）、安全断言标记语言（SAML，它依赖于加密和数字签名来保护特定的访问及通过 X.509 证书公钥验证访问），将这些技术和其它基于标准的技术结合起来，确保 SDP 与企业现有安全系统可以集成。

自云安全联盟（CSA）首次发布软件定义边界（SDP）规范以来，CSA 已经看到了 SDP 无论在知名度还是在企业的 SDP 创新应用方面都取得了巨大的增长。虽然传统的网络安全方法在所有行业中似乎都让 IT 和安全专业人员感到身心疲惫，但 SDP 技术使用和兴趣却在不断增加，例如：

- 五个 SDP 工作组在其重点领域取得了重大进展，包括用于 IaaS 的 SDP、防 DDoS 攻击和汽车安全通信 [14]。
- 已经有多个供应商提供多种商业 SDP 产品，并已在多个企业中被使用。
- 针对 SDP 的防 DDoS 用例实施了开源（参考 [15]）。
- 已举办四个针对 SDP 的黑客松，并且攻破成功率保持为零。
- 行业分析师报告已开始将 SDP 纳入研究和演示。

14 SDP-for-IaaS: https://cloudsecurityalliance.org/download/sdp-for-iaas/ Anti-DDoS: http://www.waverleylabs.com/open-source-sdp/Software-Defined Perimeter Working Group Initiatives: https://cloudsecurityalliance.org/ group/software-defined-perimeter/# initiatives

15 http://www.waverleylabs.com/open-source-sdp/demo/

SDP 安全优势

- SDP 通过最小化攻击面来降低安全风险。
- SDP 通过分离访问控制和数据信道来保护关键资产和基础架构，使其中的每一个都看起来是“黑”（不可见）的，从而阻止潜在的基于网络的攻击。
- SDP 提供了一个集成的安全体系结构，这个体系结构是现有安全产品（如 NAC 或反恶意软件）难以实现的。SDP 集成了以下独立的架构元素:
    - » 用户感知的应用程序
    - » 客户端感知的设备
    - » 网络感知的防火墙/网关
- SDP 提供了基于连接的安全架构而不是基于 IP 的替代方案，因为当今 IP 环境的爆炸式增长和云环境中的边界缺失使得基于 IP 的安全性变得脆弱。
- SDP 允许根据预先审查谁可以连接（从哪些设备、哪些服务、基础设施和其他参数）来控制所有连接。

SDP 商业优势

SDP 提供了许多业务优势，我们在这里概述这些优势以供您快速参考。我们期待与 SDP 社区合作，在未来的出版物中对这些益处进行深入的定性和定量检验。

业务领域	实施 SDP 的优势
节省成本及人力	使用 SDP 替换传统网络安全组件可降低采购和支持成本。 使用 SDP 部署并实施安全策略可降低操作复杂性，并减少对传统安全工具的依赖。 SDP 还可以通过减少或替换 MPLS 和租用线路利用率来降低成本，因为组织机构可以减少或消除对专用主干网的使用。 SDP 可以为组织机构带来效率和简便性，最终有助于减少人力需求。
提高 IT 运维的灵活性	IT 流程可能会拖累业务流程。相比之下，SDP 的实现可以由 IT 或 IAM 事件自动驱动。这些优势加快了 IT 的速度，使其更快地响应业务和安全需求。
GRC 好处	与传统方法相比，SDP 降低了风险。 SDP 可以抑制威胁并减少攻击面，防止基于网络或者应用程序漏洞被利用的攻击。 SDP 可以提供并响应 GRC 系统（例如与 SIEM 集成），以简化系统和应用程序的合规性活动。
合规范围增加及成本降低	通过集中控制从注册设备上的用户到特定应用程序/服务的连接，SDP 可以改进合规性数据收集、报告和审计过程。 SDP 可为在线业务提供额外的连接跟踪。 SDP 提供的网络微隔离经常用于减少合规范围，这可能会对合规报告工作产生重大影响。

安全迁移上云	通过降低所需安全架构的成本和复杂性，支持公有云、私有云、数据中心和混合环境中的应用程序，SDP 可以帮助企业快速、可控和安全地采用云架构。 与其他选项相比，新应用程序可以更快地部署，且有更好的安全性。
业务的敏捷性和创新	SDP 使企业能够快速、安全地实施其优先任务。例如： • SDP 支持将呼叫中心从企业内部机构转换为在家办公的工作人员 • SDP 支持将非核心业务功能外包给专业的第三方 • SDP 支持远程第三方网络和位置上用户自助服务的设备 • SDP 支持将公司资产部署到客户站点，与客户建立更强的集成并创造新的收入

SDP 主要功能

SDP 的设计至少包括五层安全性：（1）对设备进行身份认证和验证；（2）对用户进行身份验证和授权；（3）确保双向加密通信；（4）动态提供连接；（5）控制用户与服务之间的连接并且同时将这些连接隐藏。这些和其他组件通常都包含在 SDP 实现中。

信息/基础设施隐藏

SDP 架构组件	减轻或减少安全威胁	额外效益
服务器“变黑”	所有外部网络攻击和跨域攻击	SDP 组件（控制器、网关）在尝试访问的客户主机通过安全协议（如单包授权（SPA））进行身份验证授权之前，不会响应任何连接请求。

减少拒绝服务（DoS）攻击	带宽和服务器 DoS 攻击（但请注意，SDP 应该通过 ISP 提供的上游反 DoS 服务来增强。）	面向 Internet 的服务通常位于“拒绝所有”SDP 网关（充当网络防火墙）后面，因此能够抵御 DoS 攻击。 SPA 可以保护 SDP 网关免受 DoS 攻击。
检测错误包	快速检测所有外部网络和跨域攻击。	从任何其他主机到接受主机（AH）的第一个数据包是 SPA 数据包（或类似的安全构造）。如果 AH 收到任何其他数据包，则将其视为攻击。

双向加密的连接		
SDP 架构组件	**减轻或减少安全威胁**	**额外效益**
验证用户和设备身份	来自未授权用户和设备的连接	所有主机之间的连接必须使用相互身份验证来验证设备和用户是否是 SDP 的授权成员。
不允许伪造证书	针对身份被盗的攻击	相互身份验证方案将证书固定到由 SDP 管理的已知且受信任的有效根目录。
不允许中间人攻击	中间人攻击	相互握手技术可以防止在撤销服务器证书之利用在线证书状态协议（OCSP）响应的中间人攻击。

“需知（NEED TO KNOW）”访问模型		
SDP 架构组件	**缓解或降低的安全威胁**	**额外效益**
取证简化	恶意数据包和恶意连接	对所有恶意数据包进行分析和跟踪，以便进行取证行动。
细粒度访问控制	来自未知设备的外部用户的数据窃取	只允许授权用户和设备与服务器建立连接。
设备认证	来自未授权设备的威胁；证书窃取	密匙被证实由请求连接的适当合法设备持有。
保护系统免受已被入侵设备的攻击	来自被入侵设备的“内网漫游”的威胁	用户只能访问授权的应用程序（而非整个网络）。

动态访问控制		
SDP 架构组件	**缓解或降低的安全威胁**	**额外效益**
动态的、基于会员认证体系的安全隔离区	基于网络的攻击	通过动态创建和删除访问规则（出站和入站）来启用对受保护资源的访问。

应用层访问		
SDP 架构组件	**缓解或降低的安全威胁**	**额外效益**
取消广域网接入	攻击面最小化; 消除了恶意软件和恶意用户的端口和漏洞扫描	设备只能访问策略允许的特定主机和服务，不能越权访问网段和子网。

应用程序和服务访问控制	攻击面最小化; 恶意软件和恶意用户无法连接到资源	SDP 控制允许哪些设备和应用程序可访问特定服务，例如应用程序和系统服务。

SDP 潜在应用领域

因为 SDP 是一种安全架构，所以它能够很好提供多种不同场景的安全，无法简单把它归类到现有的安全常见类别。下表列出了部分可由 SDP 实施保护的几种场景。

网络场景	现有技术的局限性	SDP 优势
基于身份的网络访问控制	传统的网络解决方案仅提供粗粒度的网络隔离，并且以 IP 地址为导向。即使 SDN 这样的新平台，企业仍然难以及时实现以身份为中心且精确的用户访问控制。	SDP 允许创建与组织相关的以身份为中心的访问控制，且访问控制是在网络层实施。例如，SDP 支持仅允许财务用户在公司允许的受控设备上通过 Web 访问财务管理系统。SDP 还允许只有 IT 用户才能安全地访问 IT 系统（SSH）。
网络微隔离	通过传统的网络安全工具，使用微隔离服务来提高网络安全性，是一种劳动密集型工作。	SDP 能够实现基于用户自定义控制的网络微隔离。通过 SDP 可以自动控制对特定服务的网络访问，从而消除了手动配置。
安全的远程访问（VPN 替代）	VPN 为用户提供安全的远程访问，但范围和功能有限。这种方式不保护本地用户，并且通常仅提供粗粒度访问控制（访问整个网段或子网）。这种安全和遵从风险通常违反最小权限原则。	SDP 可以保护远程用户和本地用户。公司组织可以使用 SDP 作为整体解决方案，摒弃 VPN 解决方案。而且，SDP 解决方案还专为细粒度访问控制而设计。用户无法访问所有未经授权的资源，这符合最小权限原则。

网络场景	现有技术的局限性	SDP 优势
第三方用户访问	安全团队通常尝试通过 VPN，NAC 和 VLAN 的组合来控制第三方访问。这些解决方案通常是孤岛式的，无法在复杂环境中提供细粒度或全面的访问控制。	控制第三方访问权限使企业能够进行创新和适应。例如，用户可以从公司办公过渡到家庭办公以降低成本或者有时可以远程工作，而且某些功能可以安全地外包给第三方专家。SDP 可以轻松控制和保护第三方用户的本地访问。
特权用户访问安全	特权用户（通常是管理员）访问通常需要更高的安全性监控和合规性监督。一般特权访问管理（PAM）解决方案通过凭证加密存储来管理访问，但是该凭证加密存储不提供网络安全性、远程访问或敏感内容访问。	对特权服务的访问可以设制为授权用户，并在网络层受到保护，并且可以向未经授权的用户隐藏特权服务，从而限制攻击范围。SDP 确保只有在满足特定条件时（例如，在定义的维护窗口期或仅从特定设备）才允许访问，然后可以记录访问日志以进行合规性报告。
高价值应用的安全访问	目前，对具有敏感数据的高价值应用程序提供细粒度授权可能需要对多个功能层进行复杂且耗时的更改。（例如：应用程序、数据外部访问。）	可以通过集成用户/身份感知，网络感知和设备感知在不暴露完整的网络的情况下限制对应用程序的访问;并依靠应用程序或应用程序网关进行访问控制。SDP 还可以促进应用程序升级，测试和部署，并为 DevOps CI / CD 提供所需的安全框架。
托管服务器的访问安全	在托管安全服务提供商（MSSP）和大型 IT 环境中，管理员可能需要定期对在重叠 IP 地址范围的网络上对托管服务器进行网络访问。这一点通过传统的网络和安全工具很难实现，并且要求繁琐的合规性报告。	可以通过业务流程来控制对托管服务器的访问。SDP 可以覆盖复杂的网络拓扑、简化访问，同时记录用户活动以满足合规性要求

网络场景	现有技术的局限性	SDP 优势
简化网络集成	要求组织定期快速集成之前不同的网络，例如，在并购或灾难恢复方案中	借助 SDP，网络可以快速无中断地互连，而无需进行大规模更改
安全迁移到 IaaS 云环境	采用基础架构即服务（IaaS）的组织急剧增加，但许多安全性问题仍待解决。例如，IaaS 访问控制可能与企业原有的访问控制无法衔接，范围仅限于云提供商环境内部。	SDP 方案改进了 IaaS 安全性。不仅将应用程序隐藏在默认防火墙之外，还会对流量进行加密，并且可以跨异构企业定义用户访问策略。请参考《SDP 在 IaaS 中的应用》白皮书。
强化身份认证方案	对已有的应用程序在安全性和合规性上可能需要额外的 2FA。但这在非网络应用和不易更改的程序上是很难实现的。	SDP 需要在对特定应用程序授予访问权限之前添加 2FA。并通过部署多因素身份验证（MFA）系统来改善用户体验，并可以添加 MFA 以增强遗留应用程序的安全性。
简化企业合规性控制和报告	合规性报告需要 IT 团队付出极其耗时且成本高昂的工作。	SDP 降低了合规范围（通过微隔离），并自动执行合规性报告任务（通过以身份为中心的日志记录和访问报告）.
防御 DDoS 攻击	传统的远程访问解决方案将主机和端口暴露在 Internet，并受到 DDoS 攻击。所有的完整的数据包都被丢弃，而且低带宽 DDoS 攻击绕过了传统的 DDoS 安全控制。	SDP 可以（让服务器）对未经授权的用户不可见，并通过使用 default-drop 防火墙，只允许合规的数据包通过。

4 具体来说，我们正在讨论用于远程企业用户访问的 VPN，而不是站点到站点 VPN 或消费者 VPN 方案。

5 在 Gartner 于 2016 年 9 月 30 日发表的一篇论文中，作者写道：“到 2021 年，60% 的企业将逐步淘汰数字商业通信的网络 VPN，转而采用软件定义边界，而 2016 年不到 1 % ”。“现在是时候将你的服务从互联网沼泽中隔离出来了 ”https://www.gartner.com/doc/3463617/time-isolate-services-internet-cesspool。

SDP架构

SDP 架构的主要组件包括客户端/【发起主机（IH）】，服务端/【接受主机（AH）】和【SDP 控制器】，AH 和 IH 都会连接到这些控制器。【SDP 主机】可以启动连接（发起主机或 IH），也可以接受连接（接受主机或 AH）。 IH 和 AH 之间的连接是通过【SDP 控制器】与安全控制信道的交互来管理的。该结构使得控制层能够与数据层保持分离，以便实现完全可扩展的安全系统。此外，所有组件都可以是冗余的，用于扩容或提高稳定运行时间。通过遵循此处概述的工作流程，可以使用图 1 中概述的技术来保护这三个组件之间的连接。

工作原理

IH 上的【**SDP 客户端软件**】启动与 SDP 的连接。

包括笔记本电脑、平板电脑和智能手机在内的IH设备面向用户，也就是说 SDP 软件在设备自身上运行。网络可以是在部署 SDP 的企业的控制之外。

AH 设备接受来自 IH 的连接，并提供由 SDP 安全保护的一组服务。AH 通常驻留在企业控制下的网络（和/或直接的代表）。

SDP 网关 为授权用户和设备提供对受保护程序和服务的访问。网关还可以对这些连接进行监视、记录和报告。

IH 和 AH 设备连接到【**SDP 控制器**】，【**SDP 控制器**】可以是一种设备或程序，它确保用户是经过身份验证和授权、设备经过验证、通信是安全建立的、网络中的用户流量和管理流量是独立的，来确保对隔离服务的安全访问。

AH 和控制器使用单包授权（SPA）进行保护，这样让未授权的用户和设备无法感知或访问。第 21 页描述了单包授权（SPA）的参考实现。

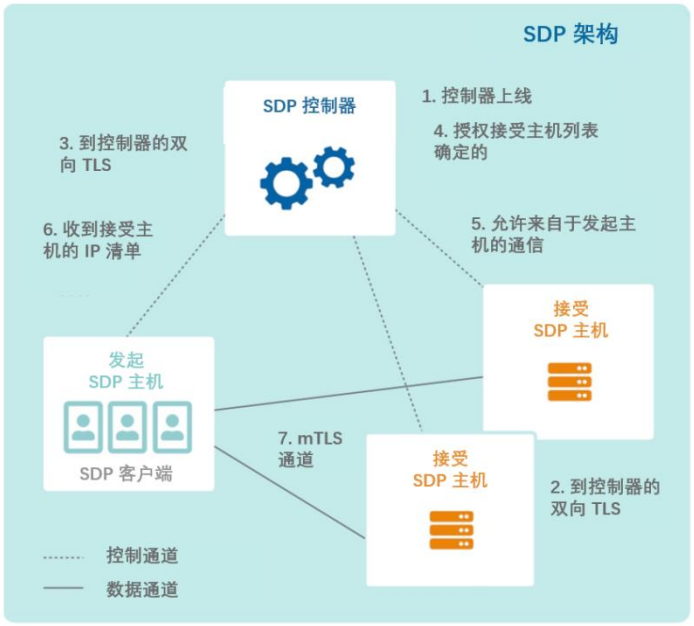


图 1：SDP 架构（已经在 《SDP 标准规范 1.0》 中发布）

SDP 的安全性遵循以下特定各步骤的工作流程：

1. 在 SDP 中添加并激活一个或多个【SDP 控制器】并连接到身份验证和授权服务，例如 AM、PKI 服务、设备验证、地理位置、SAML、OpenID、OAuth、LDAP、Kerberos、多因子身份验证、身份联盟和其他类似的服务。
2. 在 SDP 中添加并激活一个或多个 AH。它们以安全的方式连接控制器并进行验证。 AH 不响应来自任何其他主机的通信，也不会响应任何未许可的请求。
3. 每个 IH 会在 SDP 中添加和激活，并与【SDP 控制器】连接并进行身份验证。
4. IH 被验证之后，【SDP 控制器】确定 IH 被授权可以连接的 AH 列表。
5. 【SDP 控制器】指示 AH 接受来自 IH 的通信，并启动加密通信所需的任何可选策略。
6. 【SDP 控制器】为 IH 提供 AH 列表，以及加密通信所需的任何

可选策略。

7. IH 向每个授权的 AH 发起 SPA。然后 IH 和这些 AH 创建双向加密连接（例如，双向验证 TLS 或 mTLS）。
8. IH 通过 AH 并使用双向加密的数据信道与目标系统通信。（注意：上一页的图 1 中未描述步骤 8）。

SDP 部署模型

CSA 的 **SDP 标准规范 1.0** 中定义了以下几种在组织机构中部署 SDP 的可能架构：

- 客户端-网关
- 服务器-服务器
- 客户端-网关-客户端
- 客户端-服务器
- 客户端-服务器-客户端
- 网关-网关

【客户端-网关】

当一个或多个服务器必须在网关后面受到保护时，无论底层网络拓扑如何，客户端/IH 和网关之间的连接都是安全的。网关既可以位于同一位置，也可以跨域的分布。

在这个部署模式下，客户端/IH 通过 mTLS 隧道直接连接到网关，并在网关终结 mTLS 隧道。如果要确保与服务器的连接是安全的，必须采取其他预防措施。【SDP 控制器】可以位于云中或受保护服务器附近，因此控制器和服务器使用相同的 SDP 网关。

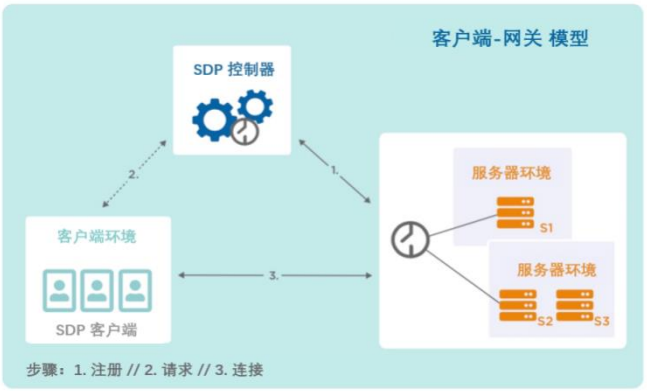


图 2：【客户端-网关】模型：一个或者多个服务器被网关保护

在图 2 中，（在一个或多个环境中的）服务器作为 AH 在 SDP 网关后受到保护。要确保穿过网关的服务器的连接安全性，服务器所处的环境应由运行 SDP 的组织控制。

网关和控制器被 SPA 和采用“默认丢弃”（default-drop）策略防火墙所保护，除非通信来自于正常的客户端/IH，服务器是不可访问的。因此，服务器对于非授权用户和潜在的攻击者而言，这些服务器是不可见且不可访问的。

受保护的服务器是无法访问的，除了来自正常的客户端/ IH，并且网关和控制器使用带有默认防火墙的SPA进行保护，因此它们是“黑暗的”并且对于未经授权的用户和潜在的攻击者是不可访问的。

受保护的服务器可以包含在 SDP 中，而无需对服务器进行任何更改。但是，它们所在的网络需要配置为仅允许从网关到受保护服务器的入站连接，这将防止未经授权的客户端绕过网关。

这种部署模式下，因为可以在 SDP 网关和受保护服务器之间部署安全组件，从而保留了组织机构使用其现有网络安全组件（如 IDS/IPS）的能力。从连接客户端到网关的流量从 mTLS 隧道中流出之后，可以进流量监

控。

客户端/IH 既可以是终端设备，也可以是服务器。（参考第 72 页的【服务器-服务器】模型）

【客户端-网关】模型适用于将其应用程序迁移到云的组织。无论服务器环境位于何处（云、本地或附近），组织机构都希望必须确保网关和应用程序之间的数据安全。

此模型还适用于保护本地遗留应用程序，因为 IH 不需要进行任何更改。

【客户端-服务器】

当组织机构将应用程序移动到 IaaS 环境并提供程序端到端地保护连接时，此模型将服务器和网关组合在一个主机中。客户端/IH 可以位于与服务器相同的位置，也可以是分布式的。在任何一种情况下，客户端/IH 和服务器之间的连接都是端到端的。

该模型为组织提供了极大的灵活性，因为“服务器-网关”组合可以根据需要在多个IaaS 提供商之间移动。此模型也适用于保护无法升级的本地遗留应用程序。

在此模型中，客户端/IH 通过 mTLS 隧道直接连接到安全服务器，并终结于安全服务器。 SDP 控制器可以位于服务器上（因此控制器和服务器使用相同的网关）或者位于云中。

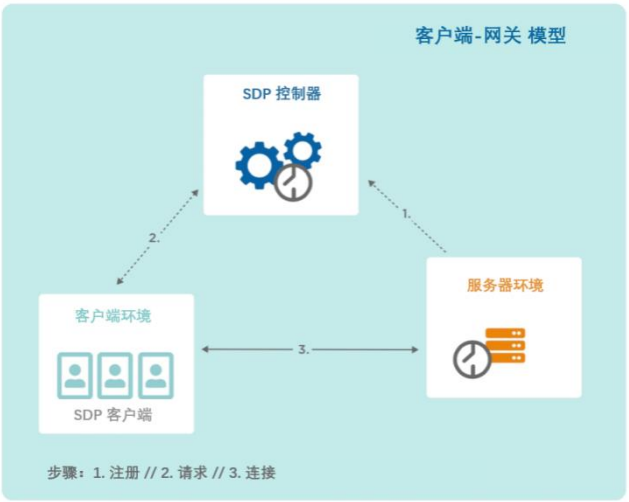


图 3：【客户端-服务器】模型：服务器上直接运行网关软件

服务器受 SDP 网关保护（作为 AH ）。通过网关连接到服务器（在服务器环境中）的安全连接可以在服务器上的应用程序/服务的所有者控制下，使所有者完全控制这些连接。

因为网关和控制器通过使用“默认丢弃”（default-drop）策略的防火墙以及 SPA 进行保护，因此除了来自被允许的客户端/IH 的请求之外，受保护的服务器是不可访问的。这意味着服务器对于内部、外部攻击者以及未经授权的用户是无法访问的，这可以抵御对内部的安全威胁。

使用此模型，受保护的服务器将需要配备网关。服务器所在的网络不需要配置为限制到受保护服务器的入站连接。这些服务器上的网关（执行点）使用 SPA 来防止未经授权的连接。

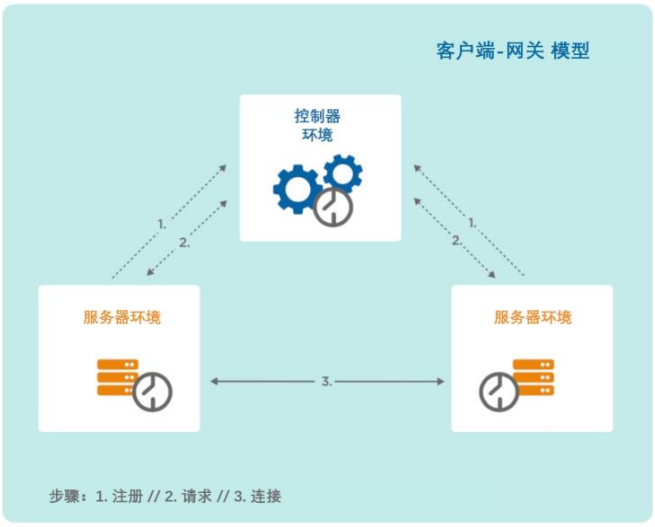
此模型可以更轻松地使用现有的网络安全组件，例如 IDS/IPS 或 SIEM。可以通过分析来自SDP 网关/受保护服务器的丢弃数据包来监控流量，从而保留客户端/IH 与服务器之间的 mTLS连接。（另请注意，客户端/IH 虽然描述为用户设备，但它本身可能是服务器。在这种情况下，请参阅下面的服务器到服务器模型。）

【客户端-服务器】模型非常适合将应用程序迁移到云的组织。无论服务器环境位于何处（云或本地），组织都可以完全控制与云中应用程序的连接。

【服务器-服务器】

此模型最适合物联网（IoT）和虚拟机（VM）环境，并确保服务器之间的所有连接都加密，无论底层网络或 IP 基础结构如何。服务器到服务器模型还确保组织的 SDP 白名单策略明确允许通信。跨不受信任的网络的服务器之间的通信是安全的，并且服务器使用轻量级 SPA 协议保持对所有未授权连接保持隐藏。

此模型类似于上一页中的客户端到服务器模型，除了 IH 本身是服务器，并且还可以充当 SDP AH。与【客户端-服务器】模型一样，【服务器-服务器】模型要求在每个服务器上安装 SDP 网关或类似的轻量级技术，并使得所有【服务器-服务器】的流量相对整个环境中其他元素而言不可见。基于网络的 IDS/IPS 需要配置从 SDP 网关而不是从外部获取数据包。此外，组织可能依赖基于主机的 IDS/IPS 。



图示 4 ：【服务器-服务器】模型：任何通信包括了 API 调用和系统服务

SDP 控制器可以位于服务器上，以便控制器和服务器使用相同的 SDP 网关。【SDP 控制器】也可以保留在云端。

服务器在作为 AH 的SDP 网关后面而受到保护。通过网关的服务器（在服务器环境中）的安全连接默认由服务器上的应用程序/服务的所有者控制，这使得所有者可以完全控制这些连接。

受保护的服务器除了来自其他白名单服务器外是不可访问的，网关和控制器由SPA通过防火墙“默认丢弃”（default-drop）模式进行保护，因此服务器是不可见的（Dark），攻击者和未经授权的用户(内部和外部)无法访问这些服务器，从而提供了额外的保护免受内部威胁。

使用此模式，受保护的服务器将需要配备网关或轻量级SPA协议。受保护的服务器所在的网络不需要配置为限制inbound(流量)连接。这些服务器上的网关(执行点)利用SPA协议防止内部和外部未经授权的连接。

该模式使应用IDS/IPS和SIEMs等网络安全组件变得更加容易。可以通过分析来自SDP网关/受保护服务器的所有丢弃包来监控流量，从而保持受保护服务器之间的mTLS连接。

该模式非常适合所有组织将物联网和VM环境迁移到云上的环境。无论服务器环境位于何处(云环境还是本地环境)，企业组织都可以完全控制到云环境的连接。

【客户端-服务器-客户端】

在某些情况下，点对点通信通过中介服务器，例如IP电话、聊天和视频会议服务。在这些情况下，SDP连接客户端的IP地址，组件连接通过加密网络，并通过SPA保护服务器/AH免受未经授权的网络连接。

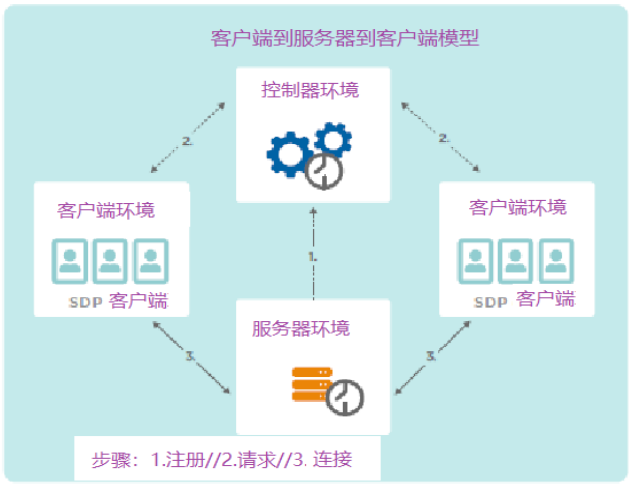


图 5:客户端到服务器到客户端模式:用于对等连接的模式，如 IP 电话或聊天。

SDP控制器可能位于服务器上(因此控制器和服务器使用相同的SDP网关)或云中。如上所述，服务器在充当AH的SDP网关后面受到保护。默认情况下，通过网关到服务器的安全连接由服务器上的应用程序/服务的所有者控制。

受保护的服务器是不可访问的，除非来自其他被允许的客户端，而且网关和控制器由SPA通过防火墙“默认丢弃”（Default-drop）进行保护，因此服务器是不可见的，攻击者和未经授权的用户(内部和外部)无法访问服务器，以提供额外的保护免受内部威胁。

使用此模式，受保护的服务器将需要配备网关或轻量级SPA协议。受保护服务器所在的网络不需要限制入向（inbound）连接。服务器上的网关(执行点)使用SPA来防止内部和外部未经授权的连接。

该模式使应用IDS/IPS和SIEMs等网络安全组件变得更加容易。可以通过分析来自SDP网关/受保护服务器的所有丢包来监控流量，从而保持客户端和受保护服务器之间的mTLS连接。

该模式非常适合于组织机构将其对等应用程序迁移到云中。无论服务

器环境位于何处(云环境还是本地环境)，组织都可以完全控制到客户端的连接。

【客户端-网关-客户端】

此模式是上面客户机到服务器到客户机的变形。该模式支持对等网络协议，要求客户端在执行SDP访问策略时直接相互连接。

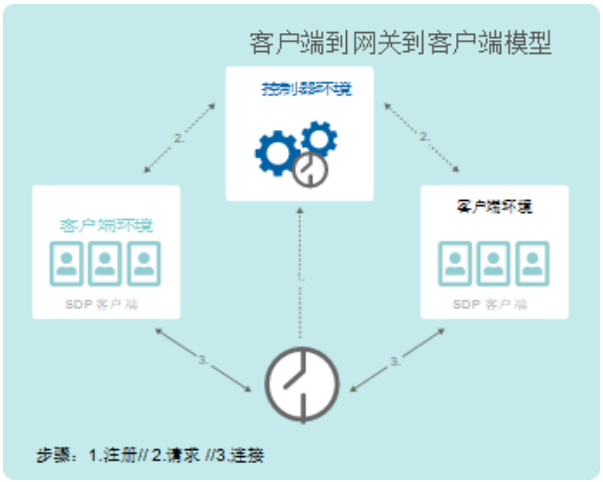


图 6:客户端到网关到客户端模型:用于保护客户端到客户端的通信。

这将导致客户机之间的逻辑连接(每个客户机都充当IH、AH或两者的角色，具体取决于应用程序协议)。注意，应用程序协议将决定客户端如何进行彼此连接，SDP网关充当它们之间的防火墙。

未来将发布更多的关于这个模式的信息。

【网关到网关】

网关到网关模式没有包含在SDP规范1.0的初始发布中。该模式非常适合于某些物联网环境。在此场景中，一个或多个服务器位于AH后面，因此AH充当客户端和服务器之间的网关。与此同时，一个或多个客户端位于IH后面，

因此IH也充当网关。

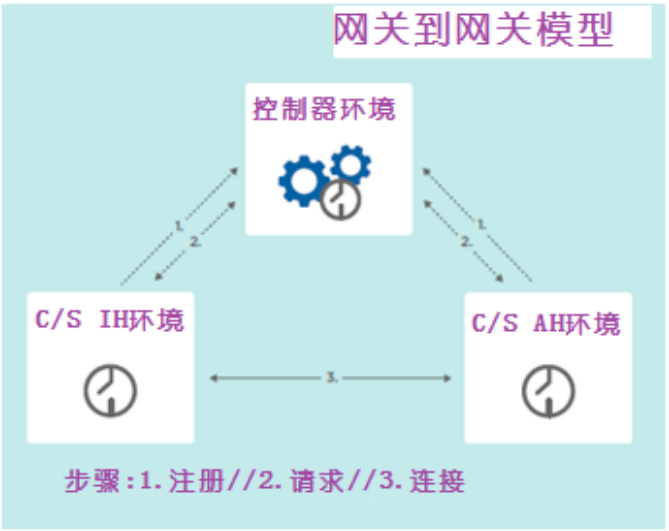


图 7:网关到网关模式:一个或多个服务器或客户端在网关后面受到保护

在这个模型中，客户端设备不运行SDP软件。这些设备可能包括那些不需要或不可能安装SDP客户机的设备，例如打印机、扫描仪、传感器和物联网设备。在这个模型中，网关作为防火墙，也可能作为路由器或代理，具体取决于实现部署方式。

SDP 部署模式和相应的场景

下表显示了哪些部署模式可以对应到哪些SDP场景。每种类型的部署都需要保护不同的连接。

网络场景	客户端到网关	客户端到服务器	服务器到服务器	客户端到服务器到客户端	客户端到网关到客户端	网关到网关
基于身份的网络访问控制	Y	Y*	Y		Y Y	Y**
所有的 SDP 模式都支持身份驱动的网络访问控制。 *此模式提供到网络和服务的安全连接。 ** 此模式为，SDP 识别设备的程度取决于特定的 SDP 实现执行设备识别和验证的方式。例如，MAC 地址提供的身份验证比 802.1x 更弱。						
网络微隔离	Y*	Y**	Y***		Y Y	Y
所有的 SDP 模式都通过保护单个连接来提供网络微隔离。 *此模式通过保护客户端和网关之间的连接来提供微隔离，但不提供到网关后面服务器的微分隔连接。 **该模式通过保护到服务器的所有连接来提供网络微隔离。此外，承载网关的服务器是隐藏的。 ***该模式通过保护到指定服务器的所有连接来提供网络微隔离。此外，承载网关的服务器是隐藏的。						
安全远程访问（VPN 替代）	Y	Y	Y		Y Y	Y
SDP 是传统 VPN 的替代品。在所有情况下，控制器和网关/AH 必须能够被远程设备访问 他们可以使用 SPA 启动连接。						
第三方用户访问	Y	Y	Y*		Y Y	Y

根据需要保护的连接，SDP 支持对所有场景的第三方访问。第三方可能是远程或现场，也可以有一个独立的身份提供程序对其进行身份验证。 * SDP 模式为，提供保护连接从第三方应用对内部应用程序的访问，第三方应用程序作为客户端。						
特权用户访问安全	Y	Y	N	Y	Y	N
SDP 保护来自客户端特权用户的访问连接。通常，特权用户访问指的是访问服务器的客户机（身份或权限），但可以应用于所有模式，具体取决于所涉及的应用程序。						
高价值应用的安全访问	Y	Y	Y	Y	Y	N
除了网关到网关模式以外，所有保护模式都提供特定的方式来进行高价值应用程序的访问保护。						
托管服务器的访问安全	Y	Y*	Y	Y	N	Y
此场景用于服务提供者访问托管服务器。服务器可以完全由网关隐藏，或者在托管服务环境中，只有管理界面由网关隐藏。 *在该模型中，SDP 网关软件部署在服务器上。服务器被隐藏，MSSP/托管服务被检测和控制服务进行连接。						
简化网络集成	Y	Y*	Y*	Y	Y	Y
所有 SDP 部署模式都支持此场景，不同模式有不同的安全连接。 *对于这些模式，另一个优点是服务器上的服务可以通过网关隐藏。						
安全迁移到 IaaS 云环境	Y	Y	Y	Y	Y	Y
这个场景涉及到将服务从本地迁移到云。						
强化身份验证方案	Y	Y	Y*	Y	Y	Y
所有 SDP 模式都提供增强身份验证的能力，通常通过多因素/逐步验证。 *此模式下没有用户，无法提示输入一次性密码。但是，它可以支持多因素身份验证，比如使用 PKI 或基于服务器的 HSM。身份管理系统可以(也应该)用于系统或设备，而不仅仅是用户。						
简化企业合规性控制和报告	Y	Y	Y	Y	Y	Y

所有 SDP 模式都通过集成控制方式帮助企业简化合规性。								
防御 DDoS 攻击	Y	Y	Y			Y	Y	Y
因为所有的 SDP 模型都在网关中使用 SPA，它们提高了组织对 DDoS 攻击的弹性。在这种情况下,我们不使用拒绝网关服务，与内部托管服务相比，面向互联网的服务更频繁的受到 DDoS 攻击。								

SDP 连接安全

SDP架构提供的协议在网络所有层都对连接提供保护。图8描述了被各种SDP部署模式保护的连接。通过在关键位置部署网关和控制器，实施人员能够专注于保护对组织最关键的连接，并保护这些连接免受网络攻击和跨域攻击。

单包授权

SDP技术最关键的组成部分之一是要求并强制实施“先认证后连接”模型，该模型弥补了TCP/IP开放且不安全性质的不足。SDP通过单包授权（SPA）实现这一点。SPA是一种轻量级安全协议，在允许访问控制器或网关等相关系统组件所在的网络之前先检查设备或用户身份。

包括请求方的IP地址等在内的连接请求的信息，在单一的网络消息中被加密和认证。SPA的目的是允许服务被防火墙隐藏起来并被默认丢弃。该防火墙系统应该丢弃所有TCP和UDP数据包，不回复那些连接尝试，从而不为潜在的攻击者提供任何关于该端口是否正被监听的信息。在认证和授权后，用户被允许访问该服务。SPA对于SDP不可或缺，用于在客户端和控制器、网关和控制器、客户端和网关等之间的连接中通信。

尽管各种SPA的实现可能有轻微差别，这些实现都应该能满足以下原则：

1. 数据包必须被加密和认证
2. 数据包必须自行包含所有必要的信息；单独的数据包头不被信任
3. 生成和发送数据包必须不依赖于管理员或底层访问权限；不允许篡改原始数据包
4. 服务器必须尽可能无声地接收和处理数据包；不发送回应或确认

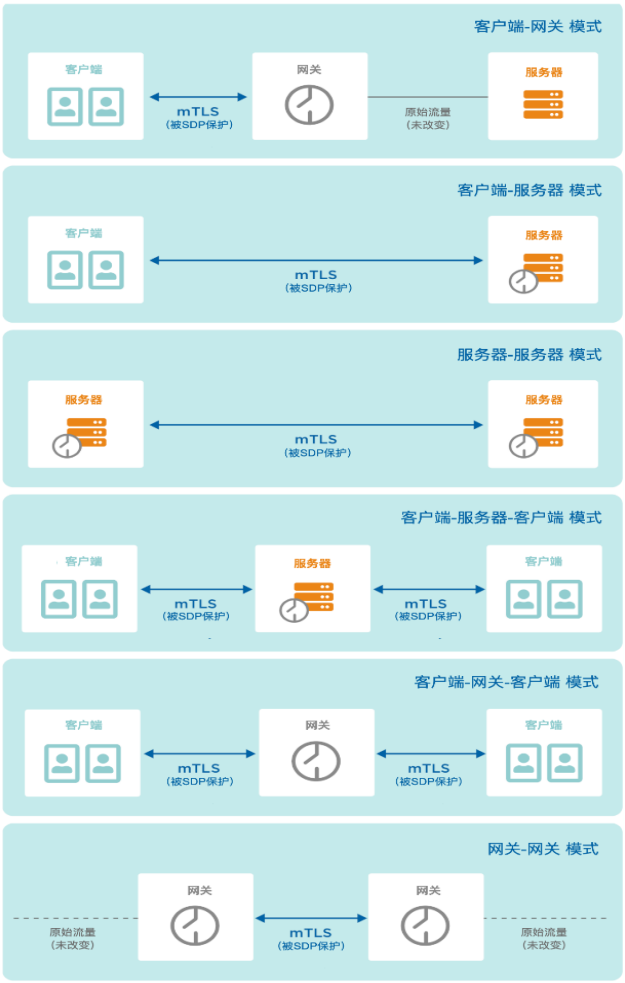


图 8：被各种 SDP 部署模式保护的连接

SPA 的好处

SPA在SDP中起很大作用。SDP的目标之一是克服TCP/IP开放和不安全的基本特性。TCP/IP的这个特性允许“先连接后认证”。鉴于今天的网络安全威胁形势，允许恶意行为人员扫描并连接到我们的企业系统是不可被接受的。与SDP组合的SPA通过两种方式应对这个弱点。使用SDP架构的应用被隐藏在SDP网关/AH后面，从而只有被授权的用户才能访问。另外，SDP组件自身，如控制器和网关也被SPA保护。这允许它们被安全地面向互联网部署，确保合法用户可以高效可靠地访问，而未授权用户则看不到这

些服务。**SPA提供的关键好处是服务隐藏。**防火墙的Default-drop（默认丢弃）规则缓解了端口扫描和相关侦查技术带来的威胁。这种防火墙使得SPA组件对未授权用户不可见，显著减小了整个SDP的攻击面。相比与VPN的开放端口以及在很多实现中都存在的已知弱点，SPA更安全。

SPA相对于其他类似技术的另一个优势是**零日(Zero-day)保护**。当一个漏洞被发现时，只有被认证的用户才能够访问受影响的服务，使该漏洞的破坏性显著减小。

SPA也可以抵御分布式拒绝服务（DDoS）攻击。如果一个HTTPS服务暴露在公共互联网而能被攻击，很少的流量就可能使其死机。SPA使服务只对认证的用户可见，因而所有DDoS攻击都默认由防火墙丢弃而不是由被保护的服务自己处理。

SPA 的局限

SPA只是SDP多层次安全的一部分，其自身并不完整。虽然SPA实现应该设计成能够抵御重放攻击，但是SPA仍然可能遭受中间人（MITM）攻击。具体而言，如果一个MITM敌方能够捕获并修改SPA数据包，虽然该敌方不能建立到被授权客户端的连接，但是可能有能力建立到控制器/AH的连接。但该敌方将不能在没有客户端证书的情况下完成mTLS连接。因此控制器/AH应该拒绝这个连接尝试并关闭TCP连接。即使是在MITM场景下，SPA也远比标准TCP安全。

不同供应商的SPA实现可能有轻微差异。Fwknop（FireWall KNock Operator）项目[14]提供了一个开源的SPA参考实现，请参考第38页附录2。另一个很好的参考是Evan Gilman和Doug Barth《零信任网络》(O'Reilly Media, Inc., 2017)一书的《信任其流量》一章。

[1]https://www.cipherdyne.org/fwknop/

SDP 和访问控制

SDP作为一个新兴架构的价值在于加强了访问控制管理，并为实施用户访问管理、网络访问管理和系统认证控制等设定了标准。SDP有能力通过阻止来自于未授权用户和/或使用未批准设备的网络层访问的方式实施访问控制。因为SDP部署了“全部拒绝”（Deny-all）的防火墙，可以阻止、允许或防止网络数据包在IH和AH间流动。至少，SDP使组织机构能够定义和控制自己的访问策略，决定哪些个体能够从哪些被批准的设备访问哪些网络服务。

SDP并不尝试去替代已有的身份和访问管理方案，但对用户认证的访问控制进行了加强。SDP 通过将用户认证和授权与其它安全组件集成（见第60页的“SDP的主要功能”）显著减小了潜在攻击面。例如，用户Jane可能没有登录公司生产财务管理服务器的密码，但该服务器即使只是简单地在网络上对Jane的设备可见，就仍然存在风险。如果Jane的公司部署了SDP架构，财务管理服务器就对Jane的设备隐藏了。所以，即使攻击者已经在Jane的设备上立足，SDP将阻止从该设备连接到财务管理服务器。即使Jane确实有允许访问财务管理服务器的密码，在她的设备上安装SDP客户端也提供了额外的保护。攻击者仍然将被多因子身份认证加上强力的设备验证拒之门外。

补充架构

零信任和 BeyondCorp

在当今的安全蓝图中，除了软件定义边界之外还有另外两个新思路：由行业分析公司Forrester最早推动的“零信任”概念和Google内部的BeyondCorp举措。

Forrester 的零信任模型

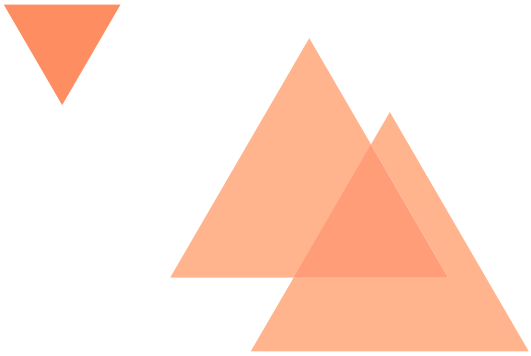
Forrester的零信任模型[2]在过去的几年中扩大了其范围，零信任模型建立在三个原则之上：

- 不管用户和资源所处位置，确保所有资源访问都是安全的
- 记录和检查所有流量
- 执行最小权限原则

这些原则与SDP所提供的一致。SDP架构可能是实施这些零信任原则的最佳方法。SDP对用户和设备进行强认证，并对网络连接进行加密，从而保证不管用户所在位置，所有资源都被安全地访问。SDP通常作为覆盖层部署在现有网络上，因此也确保无论资源是部署在内部、云上或其他位置，都可以被安全地访问。在SDP实现中，网络连接也受控制，提供了

[15]https://www.forrester.com/report/The+Forrester+Wave+Zero+Trust+eXtended+ZTX+Eco-system+Providers+Q4+2018/-/E-RES141666 and https://go.forrester.com/blogs/next-genera- tion-access-and-zero-trust/

一个中心位置记录哪些个体（人或机器）正在访问哪些资源。如果需要对数据包进行深度检测，SDP能够很容易地与网络流量探测系统集成。最后，也许是最重要的，SDP天然地且强力地执行最小特权原则。



SDP从一个默认的、拒绝所有的网关开始，并严格按照白名单访问模型访问，个体只有在明确被SDP控制器允许时才能访问SDP自身和联网的应用。这是最小特权原则的本质。

Google 的 BeyondCorp 模型

BeyondCorp是Google内部网络的安全访问平台，用于帮助其员工访问内部资源。BeyondCorp强调通过设备许可管理企业提供的Chromebook 。这个系统已经被充分研究和记录[3]，并且已经成为Google在过去五年中的一个成功的内部项目。

与SDP模型有所区别的是，BeyondCorp是一个基于Web代理的方案，

[16]https://cloud.google.com/beyondcorp/#researchPapers

[17]https://cloud.google.com/istio/

支持HTTP、HTTPS和SSH协议。SDP实现通常支持更多IP协议，在某些实现中甚至支持所有IP协议。SDP也比BeyondCorp支持更细颗粒度的访问控制。在Google的系统中，应用都被分配成若干“可信级别”。通过用户和设备上下文信息，SDP支持更细颗粒度和独立的访问控制。

尽管Google的BeyondCorp不能在市场中买到，但Google已经在其用于保护微服务的开源平台Istio[4]中包括了BeyondCorp平台的一些组件。Google也发布了一个称为身份感知代理(Identity- Aware Proxy，IAP)的免费组件[5]，控制对Google云平台（Google Cloud Platform，GCP）中资源的访问。IAP不是SDP，也不具有BeyondCorp的全部能力。据Google称，“云IAP是BeyondCorp的一个构成模块”。

如果你的企业在考虑构建建一个零信任安全环境，或者你的团队喜欢BeyondCorp方法，你可能也不妨评估一下SDP，因为它提供了类似的好处且有多个可在市场中购买到的产品。

总之，SDP架构能够保证零信任原则的成功实现。BeyondCorp实现为读者提供了将SDP架构结合到BYOD战略中的成功参考。

[18]https://cloud.google.com/iap/

软件定义边界SDP与您的企业

因为组织机构中的许多利益相关者都存在安全风险和顾虑，企业信息安全架构很复杂。无论底层IP基础设施如何，软件定义边界SDP都能确保安全连接。软件定义边界SDP因为包含以下关键概念，所以可以作为企业安全架构的基础：

1.在允许连接之前授权用户并验证设备

2.双向加密通信

3.拒绝一切（Deny-all）的防火墙动态规则和服务器隐身功能

4.集成应用上下文和细颗粒度的访问控制

在本节中，我们提出了架构师们在其企业中规划部署软件定义边界SDP时应考虑的一些问题。这些问题将帮助架构师们考虑安全性的各个方面，这些方面包括用户群、网络、服务器环境以及安全性和合规性要求。

SDP的部署如何适应现有的网络技术？

架构师们必须决定使用哪个软件定义边界SDP部署模型，同时必须理解某些模型中网关可能代表一个额外的在线网络组件。这可能会影响到他们组织的网络，例如需要对防火墙或路由进行一些更改，确保受保护的服务器是不可见的，并且只能通过SDP网关访问。

SDP如何影响监控和日志系统？

由于软件定义边界在SDP连接发起方IH和SDP连接接受方AH之间使用mTLS协议，因此网络流量对于可能用于安全、性能或可靠性监控目的的中介服务不透明。架构师们必须了解哪些系统正在运行，以及软件定义边界对网络流量的相关更改如何影响这些系统。由于软件定义边界通常为用户访问提供更丰富的、以身份为中心的日志记录，因此它们还可以

用于补充和增强现有的监控系统。此外，所有软件定义边界网关和控制器丢弃的数据包都可被记录到安全信息和事件平台中进行进一步分析。每个连接的“谁、何时、何地”信息变得更容易收集。

软件定义边界如何影响应用程序发布/DevOps流程和工具集，以及API集成？

许多组织机构都采用了DevOps或CI/CD(持续集成/持续交付)[19]等快速应用程序发布流程。这些流程及其支持的自动化框架与安全系统的集成需要经过深思熟虑，SDP也不例外。SDP可以有效地保护授权用户在DevOps时可与开发环境连接。SDP还可以在操作期间保护连接，即使是合法用户到受特殊保护的服务器和应用程序间的连接也得以保护。

安全架构师们必须理解他们的SDP部署模型，以及他们组织的DevOps机制将如何与之集成。因为API集成通常是DevOps工具集集成的需求，安全团队应该查看他们的SDP实现所支持的一组API。

SDP如何影响用户，特别是业务用户？

安全团队经常努力使其解决方案对用户尽可能透明，SDP支持这种方法。如果实现了最小权限原则，用户将可以完全访问他们需要的一切，而且不会收到不必要的访问被拒绝。根据SDP部署模型，用户将在其设备上运行SDP客户端软件。安全架构师应该与IT部门协作，对用户体验、客户端软件分发和设备安装过程进行规划。

19 https://en.wikipedia.org/wiki/DevOps and https://en.wikipedia.org/wiki/CI/CD

企业信息安全的元素

图9表明了企业安全架构的主要元素。该图是混合企业的简化视图，描述了安全基础设施原型的主要元素以及这些元素之间的关系。

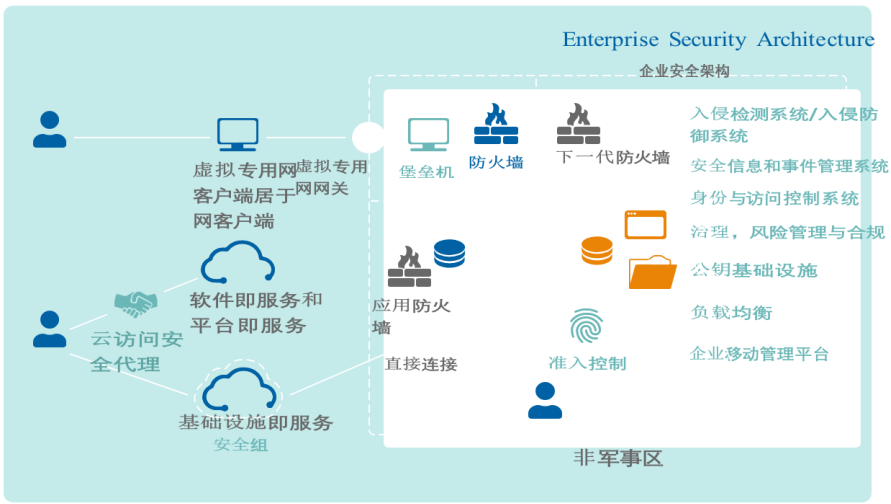
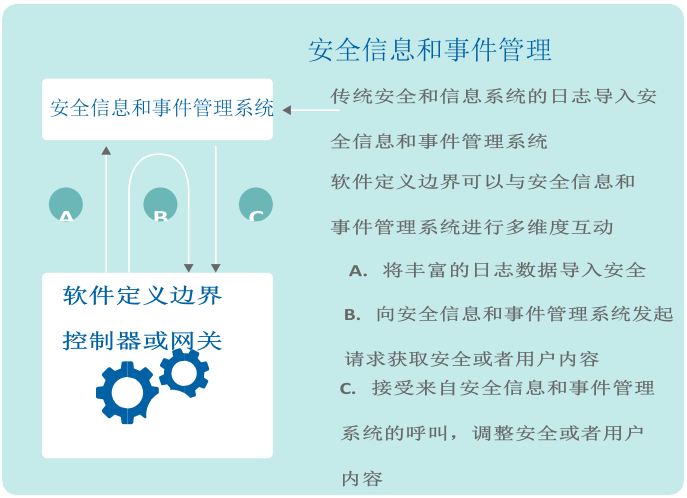


图 9：安全基础设施原型的主要元素

此企业安全体系结构示例由内部部署和基于云的资源（IaaS和SaaS / PaaS）组成，其中包含一组标准的安全、IT和合规性组件。以下几页将详细地探讨这些标准组件如何与SDP集成。

安全信息和事件管理（SIEM）

SIEM系统[14]提供对应用程序和网络组件生成的日志信息和安全警报进行分析的功能。SIEM系统集中存储并解析日志，支持近乎实时地分析，这使得安全人员能够快速采取防御措施。SIEM系统还提供了基于法律合规通常所需的自动化集中报告。



图表10：安全信息和事件管理SIEM和软件定义边界SDP

安全信息和事件管理系统无论是部署在内网还是在托管在云中，都是IT和安全管理系统中一个成熟的主流部分。虽然商业化的SDP解决方案通常提供内部日志记录功能，但当SDP日志被定向到从多个来源聚合信息的SIEM系统时，它们的价值会被放大。企业系统可能直接从分布式SDP组件接收反馈，也可能以分层方式部署多个收集代理。SIEM系统通过将预定义和定制的事件转发到集中管理控制台或通过以电子邮件向指定的个人发送警报的形式执行检查并标记异常

因为SDP以审查身份和设备的方式控制访问，所以为SIEM系统提供比典型的网络和应用程序监视工具更为丰富的信息。SDP实时提供有关每个连接的“谁、什么、在哪里”信息，从而增加了SIEM系统的价值。就这一点与SIEM系统当前用于日志记录的方式进行比较：安全分析人员必须将多个日志中的信息拼凑在一起识别未经授权的用户（“谁”），在识别从“什么”到“哪里”的未授权连接时非常具挑战性。但是，如果SDP客户端安装在用户的设备上，就可以从设备收集特定信息。所有从SDP网关丢弃的数据包都可以存储起来，以便进一步分析潜在的黑客企图或评估消耗。

这种级别的记录优于传统防火墙生成的IP地址和端口列表。SDP还增

强了SIEM系统关联跨多个设备发生的用户活动的能力。如果没有SDP，以这种方式关联用户活动通常很难实现，特别是随着自带设备办公和移动设备（BYOD）的出现时更为困难。

将SIEM系统与SDP部署集成有助于实现将安全操作从被动操作转移到主动操作的目标。为了控制风险，现有的SIEM除了作为SDP日志信息的接收器之外，还应被视为重要的信息源。SIEM系统可以通过断开用户连接、禁止来自未验证设备或某些主机的连接以及删除可疑连接帮助控制风险。例如，如果SIEM系统指示高于正常风险级别，指示未经授权的用户活动，则SDP将断开用户的所有连接，直到可以执行进一步的分析。SDP通过在几秒钟内寻址和控制连接补充了SIEM系统的功能。

与所有生成日志信息的系统一样，SDP日志产生了企业潜在的数据隐私问题。由于网络连接（及其元数据）可能与日志中的特定用户关联，因此组织需要在部署SDP期间采取预防措施解决此问题。

SDP增强并提高了SIEM系统预防、检测和响应不同类型攻击的能力。下一页显示了可以减轻攻击类型的一些示例。（通过将SDP与SIEM集成可以预防的攻击的更详细列表将在未来的CSA出版物中给出。）

安全攻击类型	缓解措施	如何将 SDP 和 SIEM 集成
端口扫描/ 网络侦察	封锁并通知	SDP 阻止所有未经授权的网络活动，并可以记录所有连接请求以供 SIEM 系统使用。
拒绝服务 DDoS 攻击	封锁并通知	由于 SDP 受单包授权（SPA）保护，拒绝服务 DDoS 攻击在很大程度上无效。单包授权会丢弃坏数据包，这些数据包可以被记录到 SIEM 系统
恶意使用授权资源	检测和定位	SDP 允许授权用户访问授权资源，但 SIEM 系统可以分析用户活动是否存在异常行为，然

		后 SDP 可以禁止授权用户访问，直到可以执行进一步的分析。
使用被盗凭证	封锁并通知	SDP 在连接之前需要进行多因素验证，使得被盗密码不足以让攻击者获得访问权限。

传统防火墙

传统防火墙基于七层开放系统互连（OSI）模型，按照一组规则监控网络流量，其中OSI的第2、3、4层分别为：数据链路层（2）、网络层（3）、传输层（4）。它们遵循5-元组[14]方法，该方法基于源和目标IP和端口过滤网络包数据，并定义流经连接的网络协议。防火墙还可以支持其他功能，例如网络地址转换（NAT）和端口地址转换（PAT）。

几十年来，防火墙一直是企业网络安全的支柱。但是，因为它们只是安全基础设施的一部分，并且只在5元组的有限世界中运行，确实存在诸多限制。通常，传统防火墙只能表示静态规则集，不能基于身份信息来表示或执行规则。

软件定义边界SDP使用防火墙或实现类似的网络流量强制功能，显著改善企业当前使用防火墙的方式。SDP可以实现许多企业组织试图通过防火墙控制的网络访问控制。企业可以通过SDP大大减少防火墙规则集。SDP可以抛开在5元组的约束，对以身份为中心的访问控制进行建模，允许对访问控制进行更准确的表示和执行。除了减少在复杂环境中编写、测试、调试和部署防火墙规则所需的工作量，SDP还支持更丰富和更精确的访问控制机制。

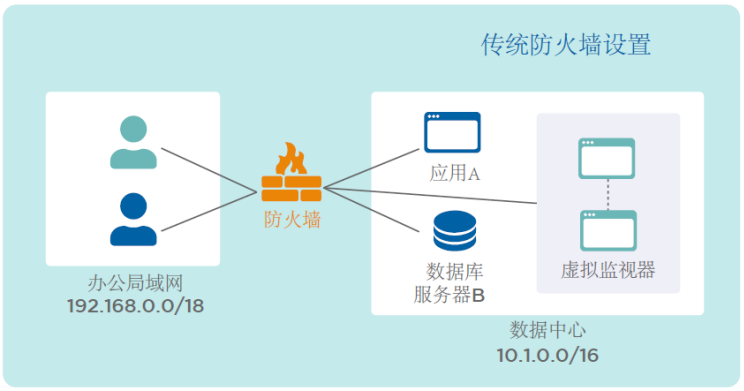


图 11: 传统防火墙设置

图11描述了在传统办公局域网环境下，通过单独的防火墙控制从用户子网（192.168.100.0/18）到当地的数据中心子网的连接，试图安全访问的困难性。

因为办公网上的各个用户仅仅通过IP地址标识，防火墙并不能区别他们。此外，因为很多用户定期地连接或断开其笔记本电脑，所有用户的IP地址会频繁变化。

一个典型的数据中心承载着包括测试和生产系统在内的大量负载。

虽然一些应用是长期存活的并使用静态IP地址，但是另外的应用则部署在虚拟机之上，这些应用经常会被创建和销毁，因此IP地址不可预测。虽然没有一个用户需要访问数据中心中所有的服务器，包括这些服务器之上的所有端口，但实际上在这个环境中，防火墙有一个规则即会强制放行在办公局域网内的所有IP地址都可以访问在数据中心网络中的所有IP地址。相比于传统防火墙设置，图12中描述了一个简化的客户端到网关的SDP模型。为了清晰起见，忽略了控制器。另外需要说明，其他SDP模型部署也是类似的。

在这个例子中，网络防火墙已经被扮演相似功能的SDP网络代替[15]。因为SDP基于明确的用户身份和他们使用的设备信息，所以SDP网关可以实现对数据中心访问的细颗粒度控制。这个开放的、扁平化的网络表示一个巨大的攻击面已经变得最小化了。注意，通过在数据中心服务器附近或者在其上增加更多的SDP网关可以实现对特定服务的更加细颗粒度的访问。（请查看第70页客户端到服务器的介绍）

在实际的部署中，防火墙仍在相应的位置之上，但是只设定最小权限规则集，例如：只能允许办公局域网的流量到SDP网关之上，然后SDP网关强制用户使用特定的设备连接特定的服务。

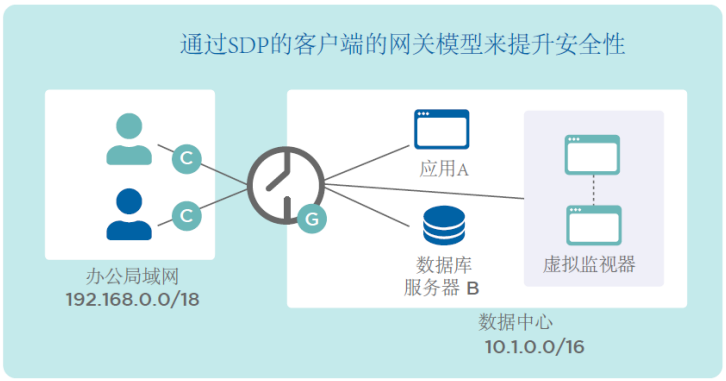


图12: 通过SDP的客户端的网关模型来提升安全性

入侵检测和入侵防御系统（IDS/IPS）

入侵检测和入侵防御系统（IDS/IPS）在这里被看成是同义词，是用作检测网络或系统恶意行为及策略违规的安全组件。它们是基于网络的（检查流量）或者基于主机的（检查活动和潜在的网络流量）。尽管需要更改基于IDS的网络，SDP可以支持IDS/IPS系统的部署。在单网络远程办公室等小型运营环境中，部署SDP可以不需要部署IDS/IPS，从而降低成本。

另外，因为SDP采用mTLS技术加密客户端和网关间的通信，所以对于IDS系统而言，网络流量变得不透明了。IDS可以采用引入证书的方式代理TLS数据流，但这会带来增加攻击面的副作用[17]。通常，因为SDP是基于mTLS（通常基于临时证书）通信的并且它可以反弹IDS扮演的中间人攻击（MITM），所以一般不会增加攻击面。

因为这些逻辑连接通过SDP证书进行加密处理，这些连接的mTLS 分段（图中蓝色标识）对于任意的外部系统是不透明的。同时，在设计上，尝试做流量分析的系统也不能访问这些连接。这个变化对于中间安全和网络监控系统有一定影响，特定使用场景不再适用，这个情况和从TLS1.2升级到1.3类似。[18]（对于每种SDP部署模型的图形化展示请参考第80页“SDP连接安全”一节）

此外，SDP支持把未加密的网络数据流（例如：被丢弃的数据包）推送到远端IDS设备。另外，基于本地部署的IDS要比基于网络部署的IDS/IPS更能增强安全操作。当然，SDP并非是影响基于网络的IDS的唯一。应用向云端迁移也提升了基于云部署的IDS的有效性和使用。

虽然部署SDP系统可能会需要对IDS系统进行一定的变更，但通过阻止未认证的网络流量的方式有助于降低系统噪声。这种改变使得IDS及其操作团队更关注已授权应用的网络流量，同时把资源有效倾斜到内部威胁检测方面。

SDP同样也可以简化和增强“蜜罐”系统的创建和有效性。因为所有的被保护系统针对攻击者而言都是不可见的，而SDP就增加了恶意攻击者发现和攻击蜜罐的可能性。一个基于SDP的“蜜罐”系统可以更快定位网络上的恶意软件行为。

虚拟专用网（VPN）

VPN用于跨越非可信的公用网络构建一个安全的访问连接。VPN通常被用作远程访问（例如：外出的员工访问公司站点），安全的内部通讯，甚至是在不同公司之间通信（点到点的外部网）。VPN通常使用TLS或者IPSec方式。[19]

虽然可以使用VPN封装和加密网络流量，但使用VPN会遇到一些限制，而SDP可以更好的解决这些问题。虽然VPN的授权成本可能很低，但是其运维需要投入大量的人力。VPN通常提供广泛的、过于宽松的网络访问能力。VPN的典型使用方式是只提供基于子网范围等方式的基本访问控制能力。在很多组织机构这些限制带来安全和合规性方面的风险。在分布式的网络环境中，VPN可能会将用户的大量不必要的流量都导到企业的数据中心，加重企业的带宽成本以及网络延迟。VPN服务器作为一个服务是暴露在公共互联网上，其可见性将导致容易被攻击者攻入。

此外，VPN给用户带来了相当大的负担和较差的用户体验。用户被要求记忆哪些应用需要使用VPN访问，哪些不需要，同时，他们也被要求手动连接或者断开VPN。对于那些有多个远程地点需要登录的用户来说，VPN无法支持同时连接，而是要求在不同环境之间进行切换。只要涉及云业务迁移，VPN的管理就爆炸式地变得复杂，使得IT管理员需要在不同的物理节点之间配置和同步VPN和防火墙访问策略。这种操作的复杂度使得消除过期的访问权限更为困难。

替代VPN是SDP最基本的目标。和VPN类似，SDP同样要在客户端设备上部署一个客户端。通过使用SDP代替VPN，组织机构可以对远程用户、

内部用户、移动设备用户等提供同一套访问控制平台。也正是因为SDP，尤其是那些部署在互联网上的SDP设备，通过SPA（单包认证）技术和动态防火墙技术，可以比传统的VPN服务器抵御更多的攻击。

下一代防火墙（NGFW）

一般而言，NGFW[20]具备传统防火墙的能力，同时添加了额外的属性使得他们成为了“下一代”。NGFW基于预定义的规则策略监视访问并检测网络数据包，并且用OSI模型2到4层的数据信息过滤数据包。NGFW同样也使用5到7层（会话层、表示层、应用层）增加额外的功能。

NGFW提供如下的能力，不同的供应商会有所差异：

- **应用识别：**根据应用决定进行何种攻击扫描
- **入侵检测（IDS）：**监视网络的安全状态
- **入侵防护（IPS）：**为了阻止安全漏洞而拒绝通信
- **身份识别（用户和组控制）：**管理用户可以访问的资源
- **虚拟专用网（VPN）：**NGFW可以提供在不信任网络上的远程用户的访问能力

虽然NGFW相比传统防火墙有很大提升，但与SDP相比仍然存在一些限制：

- 时延：和任何的IDS/IPS一样，会对网络流量造成额外时延，在执行文件审查时尤其如此。
- 可扩展性：需要很多硬件资源进行弹性扩展

- 规则复杂度：一些NGFW厂家提供了用户和分组属性等相关的身份识别能力，但是这些能力的配置很复杂。

SDP是已经部署的NGFW的天然补充。企业可以使用SDP确保用户访问策略，同时使用NGFW进行核心防火墙保护，使用IDS/IPS进行流量监测。

SDP和NGFW进行集成后带来的好处包括：强制实现不可见，并使得NGFW更加动态（后续章节会有详细描述）。虽然将NGFW和IAM或AD集成同样可以强化用户访问策略，但是使用SDP可以提供可控的、真正安全的连接。

在某些情况下，NGFW的架构和SDP存在竞争和重叠。在过去一段时间里，NGFW厂商已经成功地、创新式的解决了SDP范围内的一些问题。通过组合使用NGFW和VPN并配以用户和应用识别，企业可以在一定程度上实现SDP的许多目标。但是，在架构设计实现方面，这种方案和SDP的实现不同。NGFW是基于IP地址的，而SDP是基于连接的。NGFW可以提供有限的身份认证和以应用为中心的功能。NGFW的访问模型是典型的粗颗粒度方式，提供给用户比他们严格需要更广泛的访问能力。相比SDP，NGFW提供了较少的针对外部系统的访问决策动态管理能力。比如说：SDP系统可以只允许开发人员在经过批准的变更管理窗口期访问开发服务器。SDP有能力强化逐步认证，但通常NGFW不支持这一点。

NGFW仍然还是防火墙，所以还是工作在传统的以边界为中心的体系架构下站点到站点连接的场景中。SDP部署通常支持更加分散和灵活的网络，从而具备灵活地网络分段能力。SDP是基于Need-to-know“需知”（白名单）的安全方式设计的，这样就可以屏蔽未授权的用户和未授权的设备的未授权访问服务。SDP使用SPA和动态防火墙技术保护和隐藏认证的连接。NGFW则在一个高度暴露的环境中进行相关操作。

身份及访问管理 (IAM)

IAM系统为用户和设备提供了验证其身份(通过身份验证)的机制，并存储关于这些身份的管理属性和组成员关系。SDP体系结构旨在与现有的企业IAM提供者集成，例如LDAP、活动目录（Active Directory）和安全断言标记语言（SAML）等。

SDP的控制访问通常基于IAM属性和组成员关系以及用于连接的设备

的属性等因素。用户和设备授权的组合有助于建立更细颗粒度的访问规则，进行授权或予以限制，确保只有授权设备上的授权用户才能对授权应用程序进行访问。

SDP与IAM的集成不仅用于初始用户身份验证，还用于加强身份验证，例如提示使用动态口令（OTP）访问敏感系统，或者在某些情况下(例如远程访问与本地访问)加强验证。IAM系统还可以通过应用程序编程接口（API）调用SDP进行通信，响应身份的生命周期行为，例如禁用帐户、更改组成员、删除用户连接或更改用户角色。

在SDP中使用IAM对用户进行身份验证，为SDP提供用于做出授权决策的信息，并对用户从注册设备发出的所有授权访问提供丰富的审计日志。将应用程序访问(不是网络访问)与用户(而非IP地址)绑定在一起，可以为日志记录提供有用的连接信息，并在出于安全或合规性原因需要审计历史访问记录时显著降低IT开销。

IAM工具通常关注维护身份生命周期的业务流程，并对如何使用身份信息控制对资源的访问进行标准化。例如，授予用户访问的机制通常是手动和自动流程的组合。因此这些流程依赖于由IAM工具管理的身份属性和组成员关系，所以SDP支持这些流程。当用户属性或组成员关系发生更改时，SDP会自动检测这些更改，并在不更改IAM流程的情况下更改用户访问权限。

SDP与SAML可以集成[21]。在SDP的部署中，IAM提供者可以充当用户属性的身份提供者和/或身份验证提供者(例如多因子认证)。除了**SAML 之外**，还有许多开放身份验证协议，如OAuth[22]、OpenID Connect[23]、W3C Web身份验证 (WebAuthn)[24]、和FIDO Alliance Clientto-Authenticator协议(CTAP)。[25](这些协议将在未来与SDP相关的研究中进行探索。)

21 https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language

22 https://en.wikipedia.org/wiki/OAuth and https://oauth.net/

23 https://en.wikipedia.org/wiki/OpenID_Connect

24 https://www.w3.org/TR/webauthn/

25 https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-client-to-authenticator-proto- col-v2.0-id-20180227.html and https://fidoalliance.org/fido2/

26 https://en.wikipedia.org/wiki/Network_Access_Control

网络准入控制(NAC)解决方案

NAC解决方案通常控制哪些设备可以连接到网络，以及哪些网络主体可被访问。这些解决方案通常使用基于标准的硬件(802.1X)和软件来验证设备，然后授予设备访问网络的权限，这些操作运行在OSI模型的第2层。

当设备首次出现在网络上时，NAC执行设备验证，然后将设备分配给网络段(VLAN)。在实际中，NAC将设备粗略地分配给少量网络。大多数组织只有几个网络，比如“访客”、“员工”和“生产”。由于NAC运行在网络的第2层，它们通常需要特定的网络设备，不能运行在云环境中，也不能远程使用。

SDP是一个集成了用户和设备访问的NAC的现代化解决方案。然而，在某些环境中使用NAC是有意义的——例如，像打印机、复印机、固定电话和安全摄像头这样的硬件设备。这些设备通常内置802.1X支持，不支持安装SDP客户端。通过SDP网关到网关模型来保护这些设备并控制用户对它们访问是一个更好的选择，也是未来SDP研究的一个主题。

终端管理(EMM/MDM/UEM)

许多企业使用终端管理系统，通常分为企业移动化管理(EMM)、移动设备管理(MDM)或统一端点管理(UEM)。这些是企业IT和安全的重要元素，它们的价值和重要性通过SDP部署得到增强。

终端管理系统可用于跨用户设备自动化分发和安装SDP客户端。由

于这些系统通常使用与SDP相同的身份及访问管理系统，因此可以紧密协调部署以简化用户体验。这些系统通常还提供了功能丰富的设备自检和配置评估功能。SDP可以对设备管理平台进行API调用，获取特定设备的信息，然后根据这些信息做出动态访问决策。

或者，没有部署终端管理系统的企业组织可以直接利用软件定义边界SDP来管理和控制设备。

Web 应用防火墙(WAF)

Web应用程序防火墙(WAFs)用于过滤、监视和阻止Web应用程序进出的HTTP(S) Web流量。Web应用程序防火墙检查应用程序协议的流量，阻止源于应用程序安全漏洞的攻击，如SQL注入、跨站点脚本(XSS)和文件包含[27]。Web应用程序防火墙尽管通常在用户和应用程序之间以类似于入侵检测（IDS）/入侵防御（IPS）的方式联机运行，但却不是网络访问控制或网络安全解决方案。Web应用程序防火墙主要检查HTTP(S)协议流量，检测并阻止恶意内容。

Web应用程序防火墙是SDP的补充。例如，在客户端到网关模型中，Web应用程序防火墙部署在SDP网关之后，在从SDP的mTLS隧道中提取本地Web应用程序流量之后，对流量进行操作。在客户机到服务器和服务器到服务器模型中，Web应用程序防火墙与服务器上的SDP网关集成，以便对检查的HTTP流量进行进一步分布式控制。

负载均衡

负载均衡是许多网络和应用程序架构的一部分。负载均衡包括基于DNS和基于网络的解决方案，架构师需要在规划SDP部署时了解企业组织如何使用它们。

例如，基于网络的负载均衡通常联机部署在网络上，类似于上面讨

论的WAF一样位于客户机和服务器之间，可能无法检查SDP组件之间的mTLS连接。SDP部署和负载均衡方法的细节需要仔细分析，确保可以最大限度地部署SDP。

云访问安全代理 (CASB)

CASB位于云服务用户和云应用程序之间，监视与执行安全策略的相关的所有活动。它们提供各种各样的服务，包括监视用户活动、警告管理员潜在的危险行为、强化安全策略合规性和自动防止恶意软件。CASB既能位于用户和云服务之间，也能使用SaaS API的方式部署于SaaS系统内部，这取决于供应商和SaaS平台对API的支持水平。

CASB功能通常不与SDP 功能重叠，因为CASB通常在第7层(应用层)操作，检查应用程序流量。CASB通常不提供网络安全或访问控制。但是，还是可以通过SDP进行数据保护和用户行为分析，从而简化其运维。

基础设施即服务 (IaaS)

IaaS平台的安全性是围绕行业标准的“共享责任”模型构建的[29]，其中云提供商承担一定的责任(云自身的安全性)，而客户负责保护其应用程序(云上的安全性)。IaaS中客户使用云网络安全组[30]控制对其云资源的访问。这些网络安全组作为简单的防火墙配置和使用。这些安全措施可以与软件定义边界SDP集成，创建一个更加健壮的安全环境。

软件即服务 (SaaS)

Salesforce.com和Office 365等SaaS应用程序是多租户的，并可以在公共互联网上公开访问。目前，防止未经授权的用户进行网络级访问并不是这些系统的目标。组织机构可能希望在采用SaaS 应用程序时加强安全性，原因如下:

- 确保只有授权设备上的授权用户才能访问该特定组织租用的SaaS
- 确保SaaS应用程序使用管理的企业IAM身份凭证进行身份验证
- 确保用户访问SaaS应用程序时强制进行多因子身份验证
- 确保对SaaS应用程序访问的所有行为都被识别并记录

越来越多的SaaS供应商认识到他们的企业客户想要“限制源IP地址和设备”功能。这些特性在软件定义边界SDP和传统VPN上同样有效，并使SaaS客户能够限制用户通过特定的IP地址访问(登录和使用)他们的域(租用的服务)。对于软件定义边界而言，源IP是系统(网关)的一个元素，用户流量通过它进行路由、被授权或被拒绝。

平台即服务（PaaS)

与标准硬件或基于IaaS的系统相比，PaaS产品允许企业以更小成本构建和部署定制应用程序。与IaaS和SaaS不同，对PaaS系统的网络访问控制(以及SDP的相关程度)取决于PaaS提供者提供的功能以及启用外部访问控制的方式。

然而，主要的PaaS提供者的PaaS和IaaS平台支持相同的网络安全模型。例如，微软 Azure PaaS安全模型通过Azure网络安全组支持源IP地址限制。谷歌云平台App Engine和亚马逊 Elastic Beanstalk也可以部署不同的SDP模型，这取决于PaaS应用程序是什么以及需要保护哪些连接。

治理、风险管理及合规(GRC)

治理、风险管理和合规（GRC）[36]通常是企业整体安全框架的一部分，帮助确保组织实现安全目标并行事正直。GRC系统通常通过购买的治理、风

险管理及合规软件[37]实现，通过标准和指南(如SOX、PCI等)定义并强化对包括IT在内的许多组织系统的控制。

SDP可以通过强制执行和记录GRC系统所需的访问控制的方式与GRC系统交互并支持GRC系统。例如，GRC系统可能要求生产系统与非生产系统隔离，并记录所有用户对生产系统的访问。软件定义边界可以执行这种网络分割，并且可以为GRC系统提供审核日志进行验证。

公钥基础设施(PKI)

公钥基础设施(PKI)是“创建、管理、分发、使用、存储和吊销数字证书，以及管理用于加密、解密、散列和签名的私有和公共密钥所需的一组角色、策略和过程”。 SDP可以使用PKI生成TLS证书和安全连接。即使不存在公钥基础设施，SDP也可以提供TLS证书保护连接[39]。现有的PKI是SDP的一个自然集成点，因为它们可被SDP用于生成证书以及验证用户身份。

软件定义网络(SDN)

软件定义网络是通过API驱动IT网络基础设施，用于协调IT网络内的网络导流。SDN支持高效的网络配置，提高性能和监测能力。软件定义网络的重点是流量效率，而不是安全性和授权。运行良好的SDN系统为企业提供可靠、高效和自适应的网络带宽。

无论底层网络基础设施如何，SDP协调网络上对象之间的连接。SDP可以与SDN集成在一起从而获得部署SDN带来的益处，但这种集成不是必须的。例如可以把SDP和SDN的控制器集成在一起。SDN还可以为加密的、非透明的mTLS连接提供QoS。

无服务器计算模型

随着计算模型的发展，安全工具和体系结构也必须随之发展。一个例子是“无服务器”计算模型[41]的增长，云提供商提供了在“函数即服务”模型中运行自定义代码或在“无服务器数据库”中预构建代码集的能力。

“函数即服务”模型可以向整个互联网公开通用公共节点，并使用API密钥控制身份验证和授权。在这种情况下，因为这些接口被设计为公共的，SDP模型将不适用。然而，其他服务(或其它云提供商)可能选择遵循不同的安全模型，其中每个客户都有自己的专用接入点实现“作为服务”功能。在这样的模型中，可以使用SDP网关保护私有接入点的安全。

架构关注点

虽然SDP所涉很广，可以涵盖大量的网络访问场景，但它并不能解决所有的安全问题。这些领域不在关注的范围之内：

- 保护或控制对公共网络服务(例如不需要身份验证的网站)的访问——SDP更适合会员制的服务（针对特定人群）
- 终端防护
- 某些计算模型，如无服务器计算
- 特别的网络连接拓扑，如点对点，取决于SDP部署模型(参见第19页的“SDP部署模型和相应场景”)

29 https://aws.amazon.com/compliance/shared-responsibility-model/ and https://blogs.
msdn.microsoft.com/azuresecurity/2016/04/18/what-does-shared-responsibility-in-the- cloud-mean/

30 https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html and
https://docs.microsoft.com/en-us/azure/virtual-network/security-overview

31 https://downloads.cloudsecurityalliance.org/assets/research/sdp/sdp_for_iaas.pdf

32 https://en.wikipedia.org/wiki/Platform_as_a_service

33 https://docs.microsoft.com/en-us/azure/security/security-paas-deployments

34 https://cloud.google.com/vpc/docs/firewalls

35 https://aws.amazon.com/premiumsupport/knowledge-center/security-group-elas-
tic-beanstalk/ and https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.
managing.ec2.html

36 https://en.wikipedia.org/wiki/Governance,_risk_management,_and_compliance

37 https://searchcio.techtarget.com/definition/GRC-governance-risk-management-and-com- pliance-software

38https://en.wikipedia.org/wiki/Public_key_infrastructure

39https://cloudsecurityalliance.org/download/software-defined-perimeter-glossary/

40https://en.wikipedia.org/wiki/Software-defined_networking

41https://en.wikipedia.org/wiki/Serverless_computing

结论

当今，企业和政府机构都面临着信息安全的严峻挑战，必须采取更有效的方法保护数据资产。SDP是组织机构的安全专业人员寻求的工具，为稳健的企业开发、操作、安全提供健壮的、可适应的、可管理的基础架构。我们希望本文能帮助安全人员更好地理解SDP体系结构是如何工作的，以及如何将其部署到他们独特的环境中。

当然，我们的工作并不止于此。未来我们将涉及更多的主题研究，包括上文提到的离散SDP部署模型以及集成的文章，以及SDP对各种业务的收益的详细解读，基于SDP部署的合规控制映射工具，以及更多的出版物。

最重要的是我们认识到我们没有所有的答案，我们诚挚邀请您加入我们的SDP工作组参与讨论，并做出贡献。作为支持安全、开放和可用的互联网的安全人员，为道德所激励的人，我们努力工作确保一个更美好的未来。我们希望你能加入我们的旅程。更多参与信息请关注。

国际云安全联盟SDP工作组：https://cloudsecurityalliance.org/working-groups/softwaredefined-perimeter/.

中国云安全联盟SDP工作组：https://www.c-csa.cn/ruanjiandingyibianjieSDP.html

附录1：

参考文献

Software-Defined Perimeter Working Group: SDP Specification 1.0.Brent Bilger, Alan Boehme, Bob Flores, Zvi Guterman, Mark Hoover, Michaela Iorga, Junaid Islam, Marc Kolenko, Juanita Koilpilla, Gabor Lengyel, Gram Ludlow, Ted Schroeder, and Jeff Schweitzer (CSA, 2014年4月)

《SDP标准规范1.0》
Software-Defined Perimeter for Infrastructure as a Serviceby Jason Garbis and Puneet Thapliyal (CSA, 2016)

《SDP在IaaS中的应用》
Software-Defined Perimeter Working Group Glossary," Cloud Security Alliance (CSA, 2018)

《SDP工作组术语》
Zero Trust Networks: Building Secure Systems in Untrusted Networks," Evan Gilman and Doug Barth (2017年6月)

《零信任网络：在非受信网络中构建安全体系》
http://shop.oreilly.com/product/0636920052265.do

"fwknop: Single Packet Authorization > Port Knocking," Michael Rash
fwknop:单包授权>端口碰撞

http://www.cipherdyne.org/fwknop/
Open Source Software-Defined Perimeter," Waverley Labs

开源SDP
http://www.waverleylabs.com/open-source-sdp/

附录2：

SDP详解

下面是SDP 1.0标准规范中定义的SPA包格式。

密文	Nonce临时随机数	防止接受过期的SPA包
	Timestamp时间戳	最常见的:服务访问请求
	Message Type消息类型	可能弃用：访问请求，NAT访问请求，网关命令消息
	Message String消息串	被允许的源IP地址，打开的目标服务ID(s)
	Optional Fields可选字段	注意：网关知道打开哪个端口，是否和向哪里转发连接
	Digest摘要	注意：可能用于请求服务流量隧道
明文	HMAC	在加密之前，这个SHA256哈希是在消息的密文部分上计算的，然后由服务器在成功解密消息后用于验证消息完整性。

对于该规范的一个改进意见是增加一个明文客户端ID，以便更高效处理进入的数据包。目前二进制SPA格式正在被设计中，而且描述该格式的RFC文档也会被创建。

SPA作为单个UDP数据包发送是最有效的。但在某些场景中是不可行的，因为（某些）网络环境可能会阻止一些或所有传出的UDP数据包。在

这种情况下，可以通过TCP连接发送SPA包。这在技术上违反了SPA的“单包”特性，但有时从实际出发考虑是必要的。

SPA包也可以通过连接机器或其他设备发送。这种做法的一个例子就是移动设备被用于代表台式计算机发送SPA包。在某些场景中，尤其在阻止UDP包的网络环境中这也是一个合理的变通方案。

软件定义边界 SDP 帮助企业安全迁移上云 (IaaS)

软件定义边界(SDP)工作组

cloud security alliance®

致谢

在此，我们由衷地感谢所有为《软件定义边界SDP帮助企业安全迁移上云(IaaS)》提供意见和反馈的个人，您的贡献对这份报告带来了更多的价值和意义。

谢谢您们！

主要作者:

Jason Garbis Puneet Thapliyal

SDP 联合主席:

Bob Flores Junaid Islam

编辑:

John Yeoh

共同审稿人：

Brent Bilger
Vince Campitelli
Matthew Carter
Aradhna Chetal
Gerald Greer
Kevin Fletcher
Jeff Huegel
Scott Kennedy
Juanita Koilpillai
Dan Logan
Nya Murray
Elamurian R
Vijay Rangayyan
John Reel
Reza Reza
Colin Robbins
Puneet Thapliyal
Yoshio Turner

Flavio Villanustre
Manish Yadav
Erkki Yli-Juuti
Xing Zhang

设计者:

Stephen Lumpe (封面设计者)
Kendall Scoboria

中文翻译版说明

由中国云安全联盟(C-CSA)秘书处组织翻译《软件定义边界 SDP 帮助企业安全迁移上云(IaaS)》（Software Defined Perimeter for Infrastructure as a Service），中国云安全联盟 SDP 工作组的专家翻译并审校。

参与本文档翻译的专家（排名不分先后）：

组长：陈本峰（云深互联）

组员：于新宇 、马韶华 、姚凯 、沈传宝 、方伟 、莫展鹏

C-CSA 工作人员：

朱晓璐

关于 CSA 大中华区 SDP 工作组：

随着云计算和移动互联网的发展，传统的基于边界防御的企业安全模型已经无法适应需求，取而代之是 Software Defined Perimeter（软件定义边界，即 SDP）安全模型。目前，SDP 已经在国外逐渐被普遍采用，为了推动 SDP 在中国企业的应用，并根据本土市场需求制定出更适应中国市场的 SDP 实践指南，在中国云安全联盟的支持下 ，CSA 大中华区成立 SDP 工作组。工作组于 2019 年 3 月成立，首批参与单位有：阿里云、腾讯云、京东云、奇安信、深信服、绿盟科技、Ucloud、顺丰科技、天融信、云深互联、中宇万通、华云数据、三未信安、上元信安、安全狗、易安联、联软科技、上海云盾、缔盟云等三十多家单位。

关于 SDP 工作组更多的介绍，请点击中国云安全联盟官网 https://www.c-csa.cn/ruanjiandingyibianjieSDP.html 查看，联盟联系邮箱：info@c-csa.cn。

序言

随着数字化及云服务的普及，企业对网络安全产品及解决方案的需求与日俱增。软件定义边界（Software Defined Perimeter，SDP）作为新一代网络安全解决理念，最早由云安全联盟（CSA）于 2013 年提出，其整个中心思想是通过软件的方式，在移动+云时代，构建起一个虚拟的企业边界，利用基于身份的访问控制，来应对边界模糊化带来的控制粒度粗、有效性差问题，以此达到保护企业数据安全的目的。

2017 年 2 月，CSA 正式发布《Software Defined Perimeter for Infrastructure as a Service》白皮书。该白皮书全面介绍了当前 IaaS 面临的安全挑战，为什么 SDP 可以改变 IaaS 安全现状，以及 SDP 在 IaaS 中的应用场景，从而为企业了解云安全问题及如何解决问题提供了有价值的参考。中国云安全与新兴技术安全创新联盟（简称：中国云安全联盟）特组织业界专家将该白皮书翻译为中文版本，相信一定会有助于更多的中国企业和 IaaS 公司从中获益。

中国云安全联盟和云安全联盟大中华区非常感谢翻译和支持工作者们和中国云安全联盟专家委员会专家们的无私贡献。

李雨航 Yale Li
CSA 云安全联盟大中华区主席
中国云安全与新兴技术安全创新联盟常务副理事长

目标

软件定义边界（SDP）的应用正在迅速普及[1]，其有效性在许多企业和案例中得到了广泛的验证。如今，随着越来越多的企业战略性地拥抱云计算 IaaS 平台，并且迫切需要云上资源的安全访问。我们相信，致力于保护云上资源的安全架构——SDP 的时机已经到来。

本报告旨在探索和解释软件定义边界（SDP）部署于IaaS时，对提高安全性、合规性和运维效率的相关优势。通过本报告，读者能够清楚认识到企业IaaS所面临的安全挑战（基于共享责任模型），原有的IaaS访问控制与传统网络安全工具结合产生的安全问题，以及软件定义边界在各种场景中的解决之道。

1 Gartner 预测，到 2017 年底，使用 SDP 来保护网络服务的企业将从 1%增加到 10%，而 2021，60%的企业将用 SDP 解决方案取代 VPN。（Gartner，《迎接新时代：隔离互联网污染环境与你的网络服务》，发布于 2016 年 9 月 30 日。）市场和市场预测未来 5 年的年复合增长率为 34%（http://www.marketsandmarkets.com/PressReleases/software-defined-perimeter.asp）。其他分析师如 ESG 也预测增长：http://www. networkworld.com/article/3141930/security/goodbye-nac-hello-software-defined-perimeter-sdp.html

方法和范围

- 本报告内容主要是基于公有云的IaaS产品，例如Amazon Web Services、Microsoft Azure、Google Compute Engine和Rackspace Public Cloud。其相关用例和方法同样适用于私有化部署的IaaS，如基于VMware或OpenStack的私有云。
- 不管是按照SDP规范实现商业化的厂商，还是没有严格按照V1标准进行产品开发的厂商，在构建产品的过程中，都有各自不同的架构、方法和能力。在本报告中，我们对厂商保持中立，并且避免头轮产商相关的能力。如果有因为厂商能力产生的差异化案例，我们会使用“也许、典型的、通常”等词汇来解释这些差异，以不牺牲报告的可读性。
- 由于大多数公有云IaaS提供商目前只支持IPv4，因此我们所讨论的内容在一定程度上有所束缚。不过，随着IPv6的应用在未来普及，在下一版本的报告中，我们将进一步完善相关内容。
- 与核心SDP规范相一致，我们专注用户到服务（user-to-service）的访问控制（南北方向）。服务器到服务器Server-to-Server（也称为东西方向）通信不在本报告的范围之内（为响应市场发展趋势，在核心SDP规范和本报告的未来修订中，我们将会解决这一问题[2]）。服务器到服务器是核心规范V1中所提到的一个支持模型，但目前，该模型还未像用户到服务模型那样被高度采用。
- 高可用性和负载均衡不在本报告讨论范围之内。
- SDP策略模型不在本报告讨论范围之内。报告中讨论的SDP用例和方法也可以适用于平台即服务的系统PaaS，这取决于它们如何支持和管理网络访问控制[3]。

在撰写这份文件时，我们努力做到内容聚焦。我们考虑了很多值得探讨的话题，但这些话题要么更适合包含在整个V2规范中，要么我们认为与本报告无关。请参阅“增强SDP规范的建议”部分，这些建议提及到V2规范的相关重点，比IaaS有更广泛的适用性，其非常重要。

虽然我们避开了这些话题，但该报告的内容仍超过了目标页数，不过我们相信，在平衡内容长度和范围方面我们做出了正确的选择。该报告也将为我们下一次内容的修订提供了良好的基础。

2 注意，在 IaaS 环境中，实际上，在某些情况下，与内网环境相比，IAAS 网络安全组更容易控制东西方向流量，因为 IAAS 网络安全组默认地拒绝跨服务器流量，这必须明确启用。

3 例如，如果 PAAS 系统支持源 IP 地址限制，则它可以被配置为只接受来自 SDP 网关的访问，这样可以让 SDP 策略来控制用户访问。

执行摘要

如今，IT 和安全管理者已深刻认识到，企业和云提供商有共同的责任共同面对 IaaS 安全挑战。IaaS 与传统的内网相比，有着不同的（并且在某些方面更具挑战）用户访问需求和安全需求，然而，这些需求并不能完全由传统安全工具或者 IaaS 供应商提供的安全架构来满足。

例如，企业往往需要对用户访问网络资源进行一定程度的限制，但传统的网络访问控制（NAC）和虚拟局域网（VLAN）解决方案在IaaS环境中并不适用，因为它是多租户、虚拟化的网络基础设施。另一个例子：在IaaS环境中，所有用户都需要对云资源进行“远程访问”，最成熟的手段无它，只有VPN。但是，随着当今移动办公、跨公司协作或动态云环境等场景广泛存在企业当中，VPN通过管理IP地址和端口的访问控制并不适用。企业越来越需要以用户为中心建立安全和访问模型。

使用软件定义边界（SDP）架构，企业用户可以安全地访问他们的IaaS资源，且不妨碍业务用户或IT生产力。事实上，当正确部署时，SDP可以成为整个企业中改变网络安全实践的催化剂——无论是在内网还是公有云的环境。有了SDP，企业可以有一个集中管控并且策略驱动的网络安全平台，覆盖他们的整个基础设施（无论是在内网还是公有云环境）和他们的整个用户群体，这是一个引人注目的愿景。事实上，SDP也正在实现这一愿景。目前，世界各地的许多企业组织都在使用SDP来增强他们的网络安全，减少网络攻击面，增加业务和IT人员的生产力，并减少他们的合规负担——同时节省资金。

- 本研究的重点是如何将SDP部署于（IaaS）基础设施的环境中，重点为以下用例：

- 开发人员安全访问IaaS环境
- 业务用户安全访问内部公司应用服务
- 管理员安全访问公共对外服务
- 在创建新服务器实例时更新用户的访问权限
- 服务提供商的硬件管理后台访问
- 多企业帐户访问控制

此外，本研究报告还解释了为什么传统的网络安全方法不适用于IaaS环境，以及SDP部署在混合环境中的价值。

软件定义边界和云安全联盟提出的十二大安全威胁

云安全联盟公布了一个值得关注的网络安全威胁的报告，以此帮助企业对云计算的采用做出明智的风险管理决策。该报告反映了安全专家在CSA社区中就最重要的云上的安全问题所达成的一致意见：SDP可有效减少受攻击面，缓解或者彻底消除安全报告中提到的威胁、风险和漏洞，从而帮助企业能够集中资源于其他领域。

下表列出了十二大威胁（《十二大网络安全威胁》），并分析SDP对于解决这些威胁的作用：

	安全威胁	SDP作用
1	数据泄露	SDP通过添加预验证和预授权层来减少公开暴露的主机的攻击面，实现服务器和网络的安全性的“最小访问权限”模型，从而有助于减少数据泄露的许多攻击方式。 剩余风险：数据泄露的几个其他攻击方式不适用于SDP，包括钓鱼、错误配置和终端保护。授权用户对

		授权资源的恶意访问将不会被SDP直接阻止。
2	弱身份、密码与访问管理	过去，企业VPN访问密码被盗往往导致企业数据丢失。这是因为VPN通常允许用户对整个网络进行广泛的访问，从而成为弱身份、密码与访问管理中的薄弱环节。 相比之下，SDP不允许广泛的网络访问，并限制对这些主机的访问权限。这使得安全体系结构对弱身份、证书和访问管理有更大的弹性。SDP还可以在用户访问资源之前执行强认证。 剩余风险：企业必须有一个积极的参与者来调整IAM流程，并确保访问策略被正确定义。过于宽泛的准入政策会给企业带来风险。
3	不安全的界面和API	保护用户界面不被未授权用户访问是SDP的核心能力。使用SDP，未经授权的用户（即攻击者）无法访问UI，因此无法利用任何漏洞。 SDP还可以通过在用户设备上运行的进程来保护API。目前SDP部署的主要焦点一直是保护用户对服务器的访问。 服务器到服务器的访问至今还不是SDP的一个重点，但是我们希望这将在不久的将来被包含在SDP范围内。 剩余风险：服务器到服务器API调用在这个时候不是SDP的常见用例，因此这种API服务可能不会受到SDP系统的保护。
4	系统和应用程序漏洞	SDP显著减少攻击面，通过将系统和应用程序的漏洞隐藏起来，对于未授权用户不可见。 剩余风险：授权用户可以访问授权的资源，存在潜在的攻击可能性。其它安全系统如SIEM或IDS必须用来监控访问和网络活动（见下文的内部恶意人员威胁）。

5	账号劫持	基于会话cookie的帐户劫持被SDP完全消除。如果没有预先认证和预先授权，并且携带适当的SPA数据包，应用服务器会默认拒绝来自恶意终端的网络连接请求。因此，即使网络请求中携带被劫持的会话cookie，也不会被SDP网关准入。 剩余风险：钓鱼或密码窃取仍然是一个风险，但SDP可以通过执行强身 份验证来降低这种风险，并有基于诸如地理定位等属性来控制访问的策略。
6	内部恶意人员威胁	SDP将限制内部人员造成安全威胁的能力。适当配置的SDP系统将具有限制用户仅能访问执行业务功能所需的资源，而所有其他资源都将被隐藏。 剩余风险：SDP不阻止授权用户对授权资源的恶意访问。
7	高级持续威胁攻击（APT）	APTS本质上是复杂的、多方面的，不会被任何单一的安全防御所阻止。 SDP通过限制受感染终端寻找网络目标的能力，并且在整个企业中实施多因子认证，有效减少攻击面，从而降低APT的存在可能性和传播。 剩余风险：预防和检测APTS需要多个安全系统和过程结合起来进行深入的防御。
8	数据丢失	SDP通过执行最小权限原则，并将网络资源对未授权用户隐藏起来，来减少数据丢失的可能性。SDP还可以通过适当的DLP解决方案来增强。 剩余风险：SDP不阻止授权用户对授权资源的恶意访问。
9	尽职调查不足	SDP不适用这种情况
10	滥用和非法使用云服务	SDP并不直接适用，但SDP供应商的产品可能有能力检测和了解云服务使用状况。
11	DDoS攻击	SDP架构中的单包授权（SPA）技术使得SDP控制器和网关对阻止DDoS攻击更有弹性。SPA与典型的TCP握手连接相比可花费更少的资源，使服务器能够大规

		模处理、丢弃恶意的网络请求数据包。与TCP相比，基于UDP的SPA进一步提高了服务器的可用性[4]。 剩余风险：虽然SPA显著降低了由无效SPA包所施加的计算负担，但它仍然是非零的，因此面向公众的SDP系统仍然可能受到大规模DDoS攻击的影响。
12	共享技术问题	SDP可以由云服务提供商使用，以确保管理员对硬件和虚拟化基础设施的访问管理。有关服务提供商的硬件管理控制面板访问，请参阅下面的讨论用例。 剩余风险：云服务提供商除了SDP之外，还必须使用各种安全系统和流程。

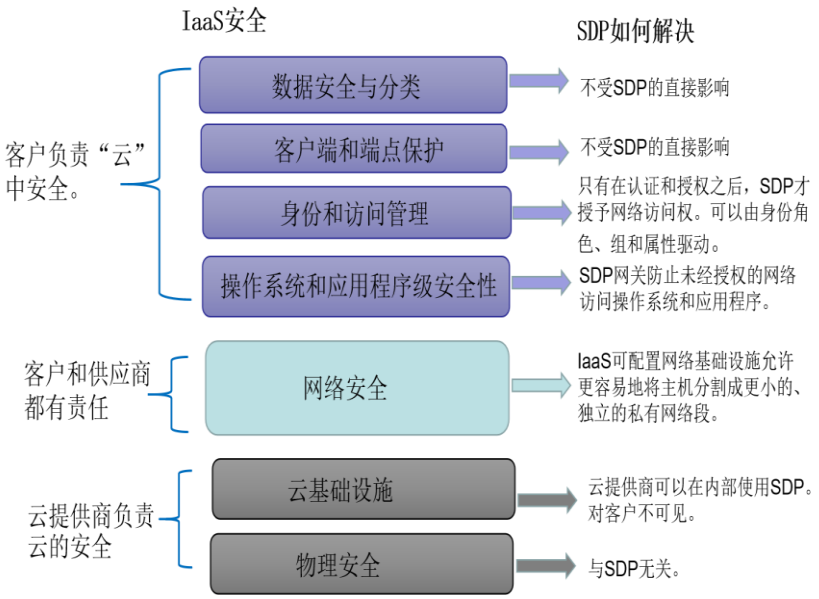
4 抗 DDoS SDP 工作组对这个问题提供了一些有趣的研究，一些正在进展中的的性能指标来对比传统 TCP 连接和 SPA 对服务器的负载的影响。值得注意的是，基于 UDP 的 SPA 甚至比基于 TCP 的 SPA 更有弹性，因为它消耗更少的服务器资源，并能更好地抵御无效的数据包流量攻击。

IaaS安全概述

业界对云上运行的应用程序的安全性往往存在诸多误解。众所周知，如果部署恰当，基于云的应用程序比起内部部署更安全。但是，云环境遵

循的是与传统内部部署不一样的安全模型，而这些不同可能无意间导致安全降低[5]。因此，向云端迁移工作业务系统不会自动让工作更安全，无论厂商还是企业都需要谨慎考量并采取行动。

IaaS 供应商通常会创建和推动“责任共享模型”，这个模型定义了 IaaS 供应商负责云的安全，而客户（企业）负责自己在云中的安全。下图是融合了几个领先 IaaS 供应商[6]的理念而创建的责任共享模型。



许多企业正在尝试拥抱云安全责任共享概念，尤其是IaaS提供商的工具集由自己创建时更是如此。这些工具倾向基于静态 IP 地址而不是基于用户（或身份），客户不能通过这种方法行之有效地管理基于用户的云资源的访问。因此，客户公司依赖应用级的身份验证来保护对这些资源的访问，致使内网里任何人都可对整个云网络进行访问。

从安全角度来看这自然存在相应风险——基于网络级的资源访问有太多可以被未经身份验证的攻击者利用的弱点。同时还有一个合规问题——企业经常要在敏感和受控环境中报告“谁访问了什么”。

如上图所示，SDP 架构在与 IaaS 供应商的责任共享模型中有重要作

用。通过 SDP，云客户可以在他们自身的安全共享控制部分采用更有效的方式

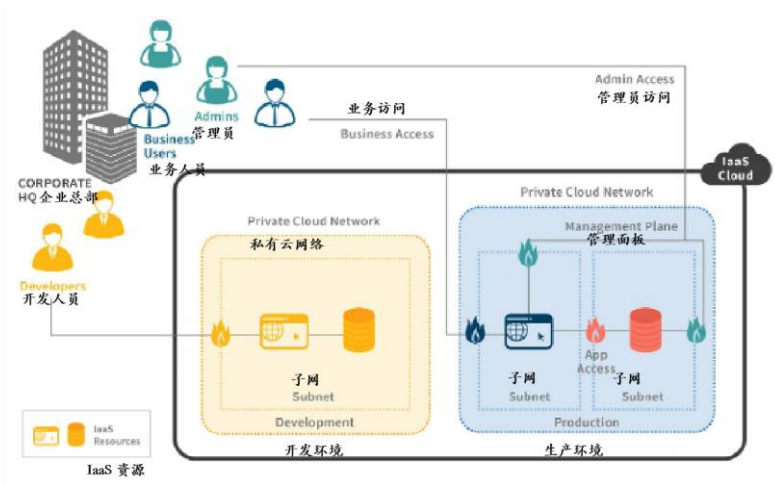
5 例如， 2017 年 1 月在不安全的、公开的 NoSQL 数据库上进行的 ransom 攻击是一个很好的例子，这些数据库大多运行在 IaaS 环境中

6 特别是这些来自于 AWS 模型 https://aws.amazon.com/compliance/shared-responsibility-model/和 Mircosoft Azure 模型 https://blogs.msdn.microsoft.com/azuresecurity/2016/04/18/what-does-shared-responsibility-in-the-cloud-mean/

技术原理

IaaS参考架构

本文读者对 IaaS 的组件和架构很熟悉， 不需在此介绍。下图是一个对公有云和私有云部署都适用的 IaaS 环境的简化架构。



该图显示了一个包含两组 IaaS 资源 (虚拟机)，分成两个私有云网络的 IaaS 云环境。这些私有云网络可以对应不同的帐户， 或云环境中不同的私有区域 (如 AWS 虚拟私有云)。从网络访问的角度来看,这些私有云网络受到云防火墙的保护， 防火墙在逻辑上控制这些网络的访问进出。对进入这些私有云网络(间)的访问控制很快就会变得复杂,不同的云提供者有不同的工具集。本文中， 我们有意省略路由表、网关或 NACLs 等构件的复杂性， 以便我们能够集中精力于管理用户访问 IaaS 资源所面临的挑战。

这个简单的模型不管供应商是谁，我们都可以持续地谈论云安全性和网络。具体地说， 我们将在本研究中使用以下术语:

术语	描述	示例
云防火墙	控制云环境的网络流量进出的安全构件。通过将服务器实例分配给云防火墙组来进行管理。	AWS: Security Group Azure: Network Security Group
私有云网络	云环境中由单个帐户控制的独立网络区域。可能包括多个子网，并且可以由一个企业中的许多人访问。	AWS: Virtual Private Cloud Azure: Virtual Network
标签	IaaS 系统支持为服务器实例指派name-value键值对。这些标签在 IaaS 系统中没有语义含义，但可以作为一个 SDP 系统进行访问策略决策的基础，非常有用。	AWS Tags Azure Tags
直接连接	IaaS供应商与电信运营商合作，提供从企业内部网络到 IaaS 环境的专用网络连接(通常使用 MPLS)。具有可靠和专用带宽的优点，通常可以将其细分为多个虚拟网络。	AWS Direct Connect Azure Express Route

为什么 IaaS的安全性更复杂？

IaaS的网络访问存在一个重大的安全挑战。作为云安全责任共享模型的一部分，网络安全直接依赖于企业。将私有云资源公开到公共互联网通常不是一个可接受的选项——仅依赖于身份验证来保护，显然不符合安全和合规要求。因此，企业需要在网络层弥补这一差距。

由于如下几个原因，这是一个典型的复杂的安全挑战。

位置只是一个普通的属性而已

不同的开发人员 （即使是座位相邻的开发者） 也可能需要不同类型的网络来访问不同的资源。例如， Sally 是数据库管理员，需要访问运行数据库的所有服务器的 3306 端口。Joe 坐在 Sally 旁边， 管理 Purple 项目的应用程序代码， 并需要使用 SSH 连接到那些运行 Purple 项目的应用服务器。Chris 和小组其他人员不一样，他是远程工作的。他是 Purple 项目的应用程序开发员，尽管相隔千里也要求与 Joe 有相同的访问。

位置可能仅仅是访问策略需要考虑的属性之一，而非传统网络环境中网络访问层的主要驱动因素。

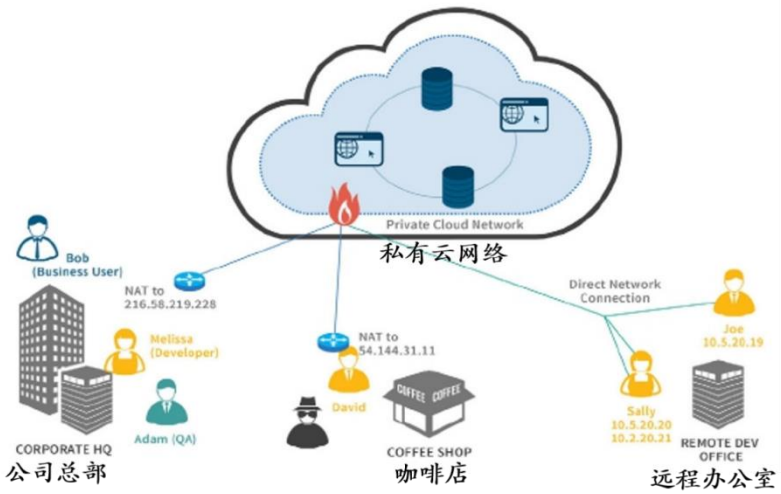
唯一不变的是变化

这是一个真理，在云环境中尤其如此。首先，IaaS 环境中的计算资源是高度动态的，服务器实例不断地被创建和销毁[7]。手动管理和跟踪这些访问几乎不可能。其次，开发者也是动态的（尽管从个人的角度来看不一定如此)——至少他们可能同时在不同项目中担任不同的角色。

这个问题在 DevOps 环境中被放大。开发、QA、发布和运维角色混合在一个团队中，对“生产环境”资源的访问可以迅速改变。

IP 地址难题

也许我们不需要传说中的网络管理员福尔摩斯 (CCIE #1)，但我们的 IPv4 世界确实面临着严峻的挑战：不仅用户的 IP 地址定期更改，用户和 IP 地址之间也没有一对一的对应关系。下图说明了当访问规则完全由 IP 地址驱动时，即使是简单的环境也会很复杂：



位置	网络设置	安全隐患
公司总部	所有用户都映射到单个 IP 地址。在此位置有许多用户需要广泛的网络访问能力	网络安全组无法区分用户，并且必须授予每个人所有资源的完全访问权限。这意味着恶意用户、攻击者或恶意软件可以从本地到云网络不受阻碍地穿越。
远程开发办公室	直接网络连接会保留每个用户的 IP 地址	IP 地址是动态分配的，并每天更改。用户还可以从多个设备访问云。 IT 运营团队不断更新安全组的规则（增加业务延迟）或网络完全对云开放（降低安全性）。

咖啡店	一个 （或很少） 用户需要从不同的位置远程访问， 可能是 NAT的方式	来自这些位置的网络访问将会同步开放给同一网络上的任何恶意用户。IT管理员很难根据用户的位置和访问需求的变化来手动调整网络访问策略。

安全要求和传统安全工具

从根本上说， 有两个问题需要解决：

- 安全地远程访问
- 用户访问的可见性和可控性

安全专业人员普遍认为向公网开放敏感服务是一个坏主意， 并希望使用其中的一种或两种方法来

保护敏感服务。

安全地远程访问

首先， 让我们考虑安全的远程访问问题。直到今天， 我们还没有发明一种将开发人员上传到云中的方法， 所以，所有的云用户都是远程的，这意味着无论网络连接是公共互联网还是专用的直接连接，与云的通信都是在网络连接上发生。

企业通常通过使用 VPN 解决这一问题， 通过建立站点到站点的VPN(如上面的图中的公司总部位置蓝色线所示)， 或者从用户的设备通过VPN 集中器直接连到云。或者， 结合上述两个方案，将用户从其设备通过VPN 连接到企业网络， 再从那里通过站点到站点的 VPN 进入云。

使用 VPN 在技术上解决了上面的问题的第一部分（安全地远程访问），它为从用户设备到云网络的网络通信提供了安全、加密的隧道。这有一些缺点，特别是如果所有的用户流量都需要先到公司网络，然后再去访问云，这额外的延迟， 造成单点故障，并可能会增加带宽成本和 VPN 授权的购买

成本。通过 VPN 直接从每个用户的设备连接到云有助于解决其中的一些问题，但可能会与用 VPN 同时进入企业网络的需求（例如访问内部开发资源）发生冲突。

普遍来说，如果 VPN 上应用程序通讯协议已经是加密的，例如 HTTPS 和 SSH，并不会增强安全的保密性和完整性，

VPN 可以提供价值的一个方面是安全的可用性，因为被 VPN 保护的资源可以确保不会公开可见，从而防止 DDoS 一类的攻击。

这是我们下一节的一个很好的话题，在这里我们谈及查看和控制用户访问的需求，而这点 VPN 无法帮助到企业用户。

用户访问的可见性和控制

不管用户如何进入 IaaS 环境的（无论是否通过 VPN），安全团队仍然需要控制（并监视和报告）在 IaaS 环境中哪些用户可以访问哪些资源。

IaaS 平台提供了内置的工具来管理这一点，例如 AWS 中的安全组和 Azure 中的网络安全组（在本文中我们称为云防火墙），基于 IP 地址控制对服务器的访问。

这是**安全访问面临的最基础的挑战**——企业需要解决用户访问问题，但只被赋予了基于 IP 地址的访问控制工具。

让我们来看一个关于云防火墙的例子：

类型	协议	端口范围	源地址
HTTP	TCP	80	173.76.247.254/32
HTTP	TCP	80	50.255.155.113/32
HTTP	TCP	80	73.68.25.221/32

HTTP	TCP	80	98.217.113.192/32
HTTP	TCP	80	209.64.11.88/32
HTTP	TCP	80	172.85.50.162/32
HTTP	TCP	80	68.190.210.117/32
RDP	TCP	3389	173.76.247.254/32
RDP	TCP	3389	110.142.238.207/32
RDP	TCP	3389	50.255.155.113/32
RDP	TCP	3389	73.68.25.221/32
RDP	TCP	3389	98.217.113.192/32
RDP	TCP	3389	209.64.11.88/32

上述的防火墙配置片段展示了IaaS平台提供的简单IP地址规则方法。所有被分配到此防火墙组的虚拟机实例都将继承这个规则集，允许网络访问特定的端口。任何IaaS的用户都可以证明这种方法存在以下几个问题：

- 它提供对此云防火墙中所有服务器的粗粒度访问
- IP地址不能与用户对应
- 没有任何策略的概念，也没有解释为什么指定的源IP地址会在这个列表中。因此，依照用户的访问控制策略去实现任何一种复杂的访问控制都是相当困难和耗费时间的。
- 上述列表是静态的，不能依据用户位置和权限的变化而做出相应的变化。
- 上述方法没有考虑任何信任的概念（比如身份验证强度，设备配置文件或客户端行为），并相应调整访问权限。
- 任何更改都需要对管理对IaaS账户进行管理访问，将导致以下两种之一发生：
    - 需要进行集中化处理，因而导致性能下降
    - 需要对更多用户设置管理员访问权限，

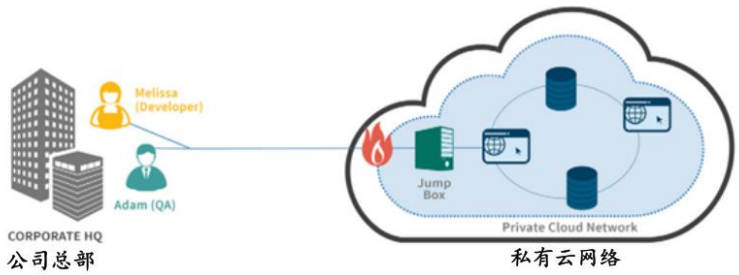
从而产生安全性、合规性和操作性问题

在IaaS环境下，安全远程访问控制已经不再是一个特殊场景。所有用户都是“远程的”，因此，网络安全团队需要关心所有用户是如何访问资源的，而不仅仅是用户的一个子集。也就是说，安全的远程访问控制必须成为一个核心关注点，并且是采用IaaS的任何企业的整体策略的一部分。

注意：除了上述方法（将多个源 IP 地址添加到单个云防火墙）以外，在另外一些云平台上，你可以使用略微有所差异的方法——创建多个云防火墙（例如，针对每个用户的 IP 创建一个防火墙），或者每个云服务器实例关联多个云防火墙。这些都与上述方法具有相同的逻辑效果。虽然可以给防火墙分配有意义的名字（例如“Sally home and work”）让它能行之有效，但是这会带来额外的开销，并且这仍然是个静态的解决方案。

跳板机：三思而后行

跳板机，也被称作跳转服务器或跳转主机，使不安全区域的用户访



问在更安全区域中运行的服务器或服务。对于本文档，使用跳板机的场景是使用跳板机来代理访问云环境中的服务器。

如上图所示，跳板机的网络访问可以是公开的，通过直接连接或者VPN来访问。访问跳板机桌面本身需要用户认证（多因子）。跳板机通过用对受管理的服务的强制单点访问，来控制云资源的访问。然而，跳板机中的诸多限制使得它不适合用于海量的云资源访问控制：

- 它不是典型的多用户系统，是用于单用户访问受保护的服务器
- 它是为特殊场合的访问控制设计的，比如系统管理员访问，而不是为持续的访问控制设计的
- 它只能对跳板机网络中的所有服务器一刀切地提供“要么全有，要么全无”的网络访问控制
- 它是一个非常有价值的攻击目标。一旦攻破一个跳板机，或者一台可以访问跳板机的用户设备，就对攻击者开放了整个网络
- 难以跟踪用户访问以实现合规性检查

很显然，跳板机不是云系统用户访问控制的合适解决方案。

为什么是SDP而不是VPN

VPN是一种广泛用于安全远程用户访问控制的普遍技术。但是为什么企业不能继续使用这种被验证过的技术呢？

VPN很好地为远程用户提供对虚拟局域网或网段的安全访问，就好像他/她们实际物理地存在于企业网络一样。这种技术，在与多因子身份认证结合时，对于具有传统边界的企业以及静态用户和服务器资源来说效果很好。但是正如Gartner的调研报告所说，“DMZ和传统VPN是为上世纪90年代的网络设计的，由于缺乏保护数字业务所需的敏捷性，它们已经过时[8]。”

VPN有两个缺点，使它们不适合当今的需求。首先，它们对所分配的网络提供非常粗粒度的访问控制。它们的目标是让远程用户的行为就像在本地网络上一样，这意味着所有用户都可以对整个虚拟局域网VLAN进行完全的网络访问。尝试配置VPN以为不同用户提供不同级别的访问是不现实的。它们也不能很容易地适应网络或服务器集群的变化。VPN根本无法跟上当今的动态发展的企业的需要。

其次，即使公司对VPN所提供的控制级别感到满意，但VPN只是一种控

制远程用户的竖井式解决方案——它们不会帮助保护本地内网中的用户，这意味着组织需要一组完全不同的技术和策略来控制本地用户的访问。这将使协调和匹配这两个解决方案所需的工作量成倍增加。而且，随着企业采用混合和基于云计算的计算模型，VPN就更难被有效地使用。

Gartner 指出：“到 2021 年，60%的企业将逐步淘汰 VPN，换而使用软件定义边界（SDP），（尽管 2016 年 SDP 的使用量不到 1%）。”[9]

虚拟桌面基础设施（VDI）

虚拟桌面基础设施（VDI）是一组技术，可以让企业在企业数据中心的集中式服务器机群中托管大量的桌面操作系统实例。这些实例可以是桌面操作系统的虚拟化实例，也可以是许多用户并发登录到的桌面操作系统的多用户版本。与VPN一样，VDI一直是企业用来远程访问其网络和应用程序的一种重要机制。

总的来说，在今天的云计算和移动世界中，VDI有一些缺点。首先，远程桌面的用户体验往往在小型的移动设备上表现不佳。它不会以一种响应的方式呈现，而且非常难以使用，因此会阻碍生产力。

其次，很多基于桌面的客户端/服务器（C/S）应用程序已经被重新编写为Web应用程序，从而减少了VDI的价值。第三，VDI集群的采购成本很高，尤其是如果它们是基于硬件的。最重要的是，随着越来越多的工作业务系统转移到云上，企业已经意识到VDI并不能解决远程应用程序的用户访问的问题。

事实上，VDI确实解决了部分远程访问问题——通常通过对从客户端设备到VDI服务器的流量进行加密，但它不能帮助解决核心的用户访问问题——控制一个特定用户可以访问的网络资源。在某些情况下，VDI会使多个用户出现在一个多用户操作系统中，从而使这个问题变得更加困难。

在这种情况下，通过传统的网络安全解决方案进行网络访问控制实际上是不可能的。

VDI 无疑是有一定的好处，但是它并不是为了控制用户对云网络和服务器资源的访问而设计的，因此在某些方面甚至会使这个问题变得更加困难。

SDP怎么解决这个问题？

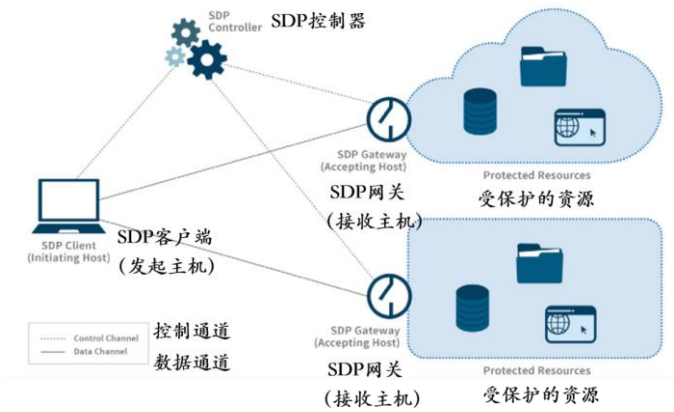
SDP可以解决上面讨论的所有安全问题，为企业提供对IaaS环境的安全远程访问，并对用户访问做到细粒度的可见性和控制。

通过使用SDP，企业的云资源对于未经授权的用户完全不可见。这样完全消除了许多攻击方式，包括暴力攻击；网络流量攻击，以及TLS漏洞攻击，如著名的“心脏出血Heartbleed漏洞” 和 “贵宾犬Poodle漏洞”。SDP通过在企业的服务器周围建立一张“暗网”，帮助它们成功地为云计算安全负责。

SDP 以预认证和预授权作为它的两个基本支柱。通过在单数据包到达目标服务器之前对用户和设备进行身份验证和授权，SDP 可以在网络层上执行最小权限原则，可以显著地缩小攻击面。

什么是软件定义边界（SDP）？

软件定义边界[10]（SDP）架构由三个主要组件组成，如图所示：



CLIENT 客户端（发起主机）	每个用户的设备上运行的客户端
CONTROLLER 控制器	用户身份认证的组件（可选择性地与用户身份管理系统集成），并在授予每个用户个性化的网络权限之前对其进行验证
GATEWAY 网关(接受主机)	网关代理访问受保护的资源。客户端的流量通过加密的通道发送到每个网关，在那里它被解密并发送到适当的应用程序（受保护的资源）。如上图所示，SDP 体系结构支持多个分布式网关，每个网关保护一组应用程序或系统资源。

用SDP术语来说，客户端（用户设备）指的是发起主机，网关指的是接收主机。在通过控制器进行身份验证后，客户端为每个网关建立加密的隧道(上面的图表显示了两个分布式网关，每个网关保护一组不同的资源，由单个控制器管理)。

SDP规范中的一个重要元素是单包授权（SPA)。使用这种技术，客户端基于一个共享密钥（seed）创建一个基于HMAC的一次性口令，并将其提交给SDP控制器和网关，作为连接建立过程发送的第一个网络数据包。(它

也用于网关与控制器的连接建立）

因为 SDP 控制器和网关拒绝无效的数据包（可能来自未经授权的用户），所以它们可以防止和未经授权的用户或设备建立 TCP 连接。由于非法的客户端可以通过分析单个数据包来区分，所以 SDP 控制器和网关所产生的计算负载是最小的。这极大降低了 DDoS 攻击的有效性，并使得 SDP 服务可以在面向公众的网络中可以更安全、更可靠地部署。

基于用户而不仅仅是IP地址策略

由于 SDP 系统是以用户为中心的（也就是说，在允许任何访问之前，它们先验证用户和设备），因此它们允许企业基于用户属性创建访问策略。通过使用诸如目录组成员、IAM-分配的属性和角色等机制，公司可以以一种对业务、安全和合规性团队有意义的方式定义和控制对云资源的访问。相比之下，传统的网络安全系统完全基于 IP 地址，根本不考虑用户。

SDP的优势

部署 SDP 的企业将在改善安全性的几个维度受益，我们希望在本文档的其余部分中清楚地表达出来。SDP 的其他好处包括运维效率、简化的合规性工作和降低成本等。下面我们将进行一一探讨。

运维效率

与达到特定级别的安全性所需的手工工作相比，SDP的自动化策略执行在运维上体现了显著的好处。从另一个角度来看，一个企业可以通过SDP轻松获得的安全性级别实际上是不可能通过传统的安全工具实现的。

简化的合规性工作

SDP的实施产品通常提供每个用户访问权限和活动的详细记录，这是由于网关对所有客户端网络流量进行日志记录和控制。因此，SDP可以根据这些细节提供自动化的合规性报告。

而且，由于 SDP 支持对用户访问的细粒度控制，企业通过将其网络分割成更小的、隔离的部分来获得降低合规性需要的范围。

降低成本

SDP可以帮助企业以多种方式降低成本。首先，对访问策略的自动执行减少了为响应用户或服务器更改而手动更新和测试防火墙规则的需求。在较大的企业中，这通常是其IT人员日常工作的一部分，因此这提供了一个减少工作量和人工成本的机会（特别是在外包模型中）。它还将提高业务和技术人员的生产效率，同时也可以有效降低硬成本（特别是对于小时工或外包的工人）。

其次，简化的合规性工作将减少准备和执行审计所需的时间和精力。这两项活动都需要第三方咨询师，节省的每一小时都是直接的成本节约。

最后，SDP 还可以给企业带来一种替代其他技术的方案，从而降低成本。例如，我们已经看到一些企业在考虑升级传统 NAC 的网络交换机时选择了 SDP，这为他们节省了数十万美元的资本支出。

SDP作为变革的催化剂

我们特别欣慰的是，SDP可以成为变革的催化剂。我们相信，SDP代表了安全架构的突破，并将很快成为广泛被采用的保护网络服务的方法。

我们越来越多地看到企业公开支持 SDP 作为它们实施安全的新方式，并将其作为一个机会来取代传统的安全技术，如 VPN、NAC、或 DMZ，因为它是一种更有效、更动态、更安全的替代品。

SDP、身份及访问管理

SDP、身份及访问管理（IAM）在很多方面都是互补的。首先，SDP能实现对已经部署的IAM系统进行身份验证，这可以加速SDP的上线。这种身份验证可能通过连接到本地LDAP或AD服务器，或者使用SAML之类的标准来实现。

其次，SDP实施产品通常使用IAM用户属性（如目录组成员、目录属性或角色）作为SDP策略的元素。例如，一个SDP策略可能会定义为“目录组中的所有销售用户都可以在443端口上访问销售门户的服务器。”这是一个很好的例子，说明SDP系统如何为现有的IAM系统增值（并扩展能力）。[12]

最后，SDP系统也可以包含在由IAM系统管理的身份生命周期中。通常被称为“加入，移动，离开”流程，IAM系统管理着与用户帐户和访问权限相关的业务和技术流程。部署SDP的企业应该将SDP管理的网络权限包含到它们的IAM供应系统中。例如，当IAM系统在应用程序X中为Sally Smith创建一个新帐户时，SDP系统应该同时创建相应的网络权限。

综上所述，这种集成很好地支持了第三方用户访问SDP系统。SDP控制器信任第三方IAM系统提供的身份验证和用户身份生命周期的所有权管理。因此，当第三方用户在它们的IAM系统中被禁用时（这对企业的用户禁用流程非常关键），用户将自动无法访问SDP保护的资源，因为他们不能通过关联进行身份验证。这个关联很好地解决了第三方访问的一个常见问题。

关于 SDP 和 IAM 如何一起工作还有很多内容要写，但是这样的分析在这个文档中是不可能的（尽管我们很喜欢这两种技术）。我们正在考虑将其纳入 SDP v2 规范中。

8 Gartner:G00315586，《迎接新时代：隔离互联网污染环境与你的网络服务》，2016 年 9 月 30 日

9 如上

10 SDP 版本 1.0 规范在这里提供：https://cloudsecurityalliance.org/download/sdp-specification-v1-0/

11 防 DDoS SDP 工作小组对这个主题有一些有趣的研究，还有一些正在研究的性能指标，对比了使用传统 TCP 连接和 SPA 在服务器负载上的区别。请注意，基于 UDP 的 SPA 比基于 TCP 的 SPA 更有弹性，因为它消耗的服务器资源更少，并且能够更好地抵御无效的数据包流量攻击。

12 这个例子是一个真实的策略，但是它在一些更大的环境中可能会面临挑战。供应商应该考虑支持参数化的策略，例如，一个使用身份和系统属性的策略有效地声明“只有部门中的用户可以访问他们在端口 443 上的部门门户”。

IaaS使用场景

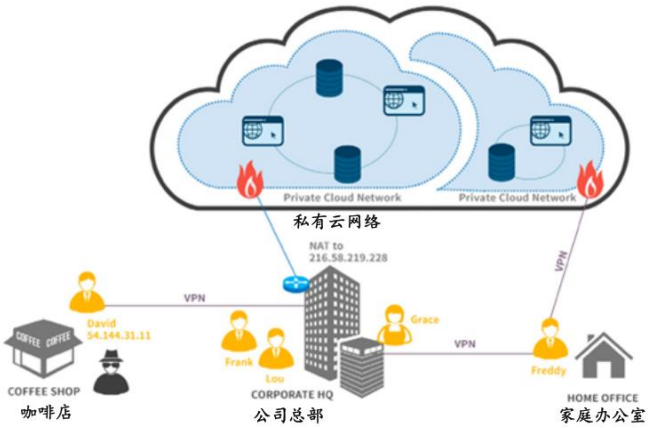
本节包含六个用例，代表了 SDP 可以为 IaaS 访问的核心需求提供帮助。

用例场景：开发人员安全访问IaaS环境

开发人员需要访问IaaS资源，用于开发，测试和部署工作。这些用户需要访问各种协议和端口，以及访问不断变化的IaaS资源。

开发人员可能会处理敏感数据，在 DevOps 环境的生产系统中工作。因此，在安全性和合规性的需求下，组织对系统访问行为必须是可视而且可控的。

不使用SDP的访问

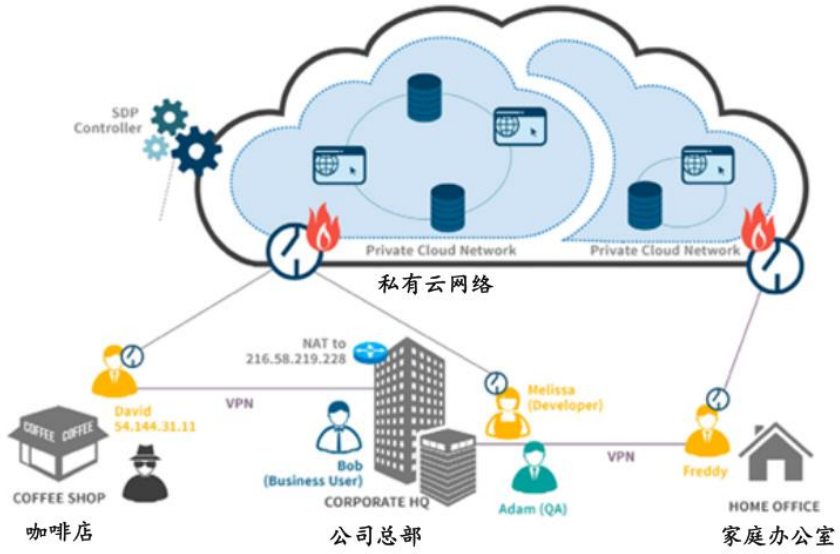


如上图所示，各种开发人员需要访问两个私有云网络环境。这些开发人员具有不同的访问要求，并且处于不同的位置。云防火墙是网络流量的唯一控制点，实质上是一个包含所有授权连接的简单表格。表格包含源 IP 地址到目标服务器和端口的映射。

使用SDP的访问

SDP 部署如下。控制器（如下页图所示）在所有用户均可访问的位置运行（为了清楚起见，连接未在图中显示）。它可能在云端的公共可访问位置运行，或者可能在总部的 DMZ 区运行。访问控制器受单包授权（SPA, Single-Packet Authorization）保护，因此将其暴露在互联网并不会显著增加风

险。



用户通过 SDP 控制器正确验证身份后，他们通过 SDP 网关访问私有云网络上的资源。网关也受 SPA 保护，所有用户流量都通过互联网上的加密隧道传输。网关对每个用户执行访问控制策略，以实现最小权限原则。网关位于每个私有云网络的入口点，并控制所有入站流量。

对照

需求：Grace，Lou和Frank在总部工作，需要通过应用程序进行协作，在多个服务器实例上访问端口22（SSH），443（HTTPS），3306（MySQL）和3389（RDP）。

挑战： 总部的所有系统都 NAT 到单个 IP 地址 216.58.219.228

不使用 SDP	使用 SDP

方法：必须将防火墙配置为允许从216.58.219.228到私有云网络中的所有服务器上的所有服务器上的所有端口的流量通过。这些服务器必须分配可公开访问的IP地址。	方法：每个用户建立一个从其设备（发起主机）到SDP网关互相认证的隧道，然后再通过网关连接到云中的目标资源。 云防火墙配置变得更简单： ● SDP网关对整个互联网的所有流量开放。因为它只允许通过SPA连接，所以它可以在一定程度上减轻DDoS和其他攻击。[13] ● 受保护的资源位于SDP网关后面的私用IP地址上，无法从互联网访问。云防火墙配置为只接受来自网关IP地址的访问连接。
效果：公司网络上的所有用户和系统都可以完全访问私有云网络，违反了最小权限原则，增加了攻击面。云网络可以被扫描并且可以被攻击者利用漏洞进行攻击。服务器访问仅通过身份验证保护，而不受网络级别的保护。密钥管理可能成为开发人员的负担。合规性检查更加困难，因为所有用户都可以访问所有系统。	**效果：**因为每个用户到 SDP 网关的连接都是单独建立并经过高强度认证的，与源 IP 地址是否经过 NAT 不再相干。SDP 网关可以以细粒度的方式对每个用户实施对云资源的访问控制。组织可以定义与用户、设备和角色相关的策略。

需求：David是一名远程开发人员，他必须定期从不安全的网络（如咖啡店）访问云端系统中的多个服务器。他还需要访问总部网络上的开发资源。这些服务使用多个协议和端口（22,443,3389）。

挑战：咖啡店网络 NAT 到单个 IP 地址 54.144.131.11。

不使用 SDP	使用 SDP
方法：开放云防火墙使其面向整个互联网访问是不能接受的，甚至允许来自54.144.131.11的所	**方法：**David 的设备向 SDP 控制器进行身份验证，然后授予访问受 SDP

有流量都具有太大的安全风险，因此David首先将VPN连接到办公网络，然后像访问云网络那样访问企业局域网	网关保护的资源的权限。David不再需要VPN进入办公网络，从而提高网络性能并减少网络带宽使用成本。
效果：David需要通过VPN连接到总部网络（他需要访问本地资源），所有流量都必须回到公司网络，从而增加延迟和带宽成本。该解决方案变成了上面表格中的要求，即允许企业网络上的所有用户和设备都可以完全访问云网络。	**效果：**由于流量是从David的设备加密到网关，因此他即使使用公共无线网络或公共互联网也没有太大风险。云防火墙的配置不必改变 - 网关对互联网开放（但受SPA保护） - 所以David无论身在何处都可以高效工作，并且安全基础架构无论位于何处都能始终如一地工作。

13 有关更多信息，请参阅防DDoS工作组的SDP研究以及CSA赞助的年度SDP Hackathons。

需求：Freddy是一位在其家里工作的开发人员，需要访问与其团队其他成员分开的私有云网络。这个环境包含敏感的，受管制的信息，所以他建立了一个VPN来进行访问。他还需要访问总部网络上的开发资源。

挑战：Freddy的位置不变，但他需要持续访问云和总部资源。出于安全目的，需要安全的网络连接。但他不能在同一台机器上同时运行两个VPN。

不使用SDP	使用SDP
方法：Freddy通过他的开发机器上的不同环境访问这些资源 。他通过虚拟机进入云端网络，并	**方法：**Freddy建立到SDP网关的安全连接，以访问受保护的云资源。

通过运行在他的自己主机操作系统上的 VPN 访问总部网络	
效果：这种方法会影响到Freddy的工作效率，因为他的一些工具和开发任务需要从同一个系统访问这两个环境。 由于 Freddy 是目前唯一访问此环境的人员，因此合规性和审计报告不成问题。但他知道，几个星期后，随着其他团队成员加入该项目，他将会面临跟踪和报告所有人的访问行为以及管理这些访问权限的问题。他还不知道他将如何启用这种访问。他应该向办公室的每个人开放云防火墙吗？ 那么远程开发人员呢？他需要管理每个人的 VPN 访问吗？	**效果：**他可以同时使用自己的VPN连接到办公室网络，与访问云资源没有冲突或问题，因为SDP连接看起来像是常规网络连接，而不是VPN。所以Freddy的工作将更高效。 Freddy 可以通过他设计的一系列策略轻松控制和报告对这些资源的访问行为。提供对新用户的访问只需编辑他的策略或编辑用户属性即可，使他能够以细粒度的方式控制访问。

总结

对于这个用例，SDP为企业提供了强大的优势：

- 无论位置如何，都可确保开发者的访问需求
- 通过服务和端口精确控制每个开发人员可以访问的服务
- 更简单的合规报告
- 更简单的安全策略配置
- 提高开发人员的生产力

用例场景：保障业务人员访问内部企业应用系统的安全

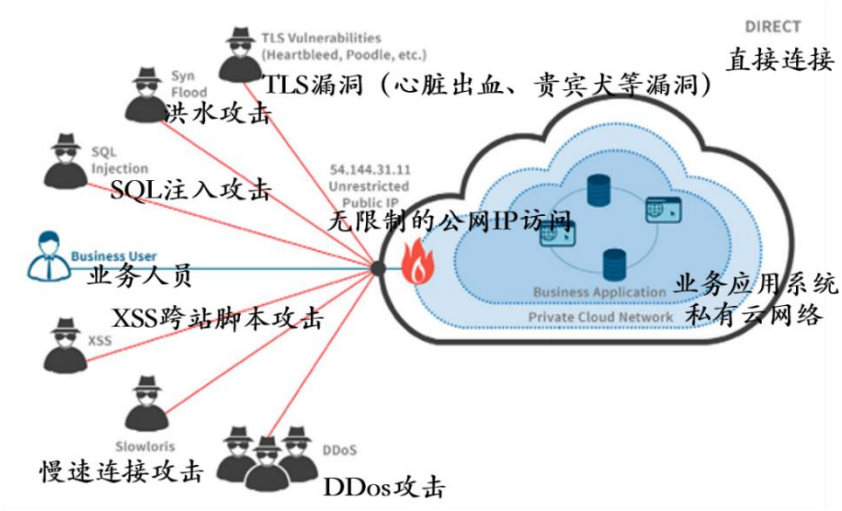
业务人员需要安全访问在IaaS环境中运行的企业应用系统，如人力资源管理系统、财务、采购、费用、供应链等。这些应用可能是供应商提供的系统，可能是内部IT开发人员开发的应用系统，也可能处于生产环境或者测试/QA环境。

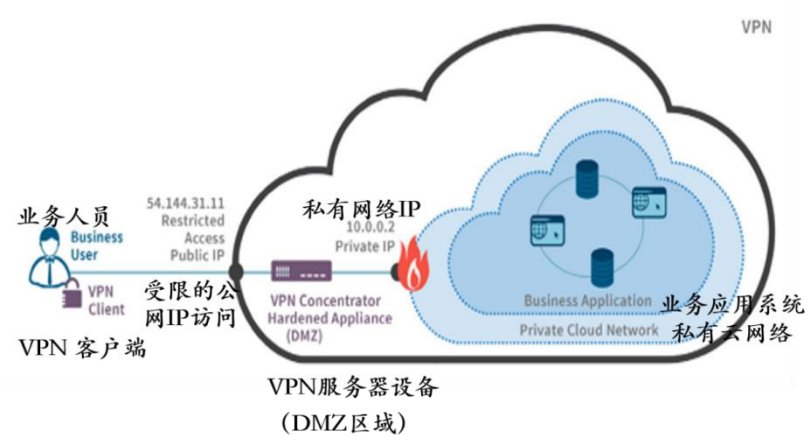
在这些情况下，业务人员通常不需要考虑网络或计算机的底层访问协议（例如SSH或RDP）。

不使用SDP的访问

向业务人员提供应用系统的安全远程访问有三种常见方式 ： 1）直接连接、2）VPN 和3）VDI。

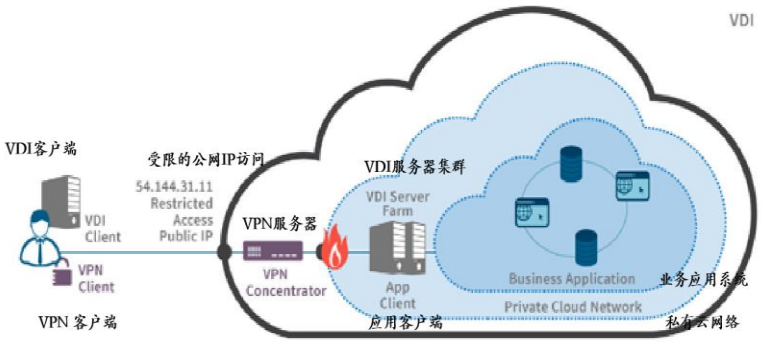
直接连接：在直接访问的情况下，应用系统通常是一个Web应用程序，配置到公共互联网环境下，不考虑访问限制。在这些情况下，应用系统会暴露于各种因素（安全威胁）下，容易受到各种形式的攻击，包括暴力破解，DDoS，XSS和任何TLS漏洞（如心脏出血漏洞Heartbleed或贵宾犬漏洞Poodle）。



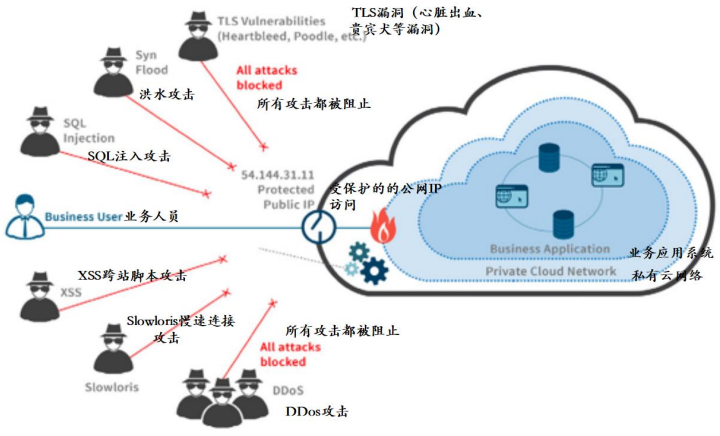


VPN： 通过VPN，内网及其所有的资源都将对该业务人员的设备开放。

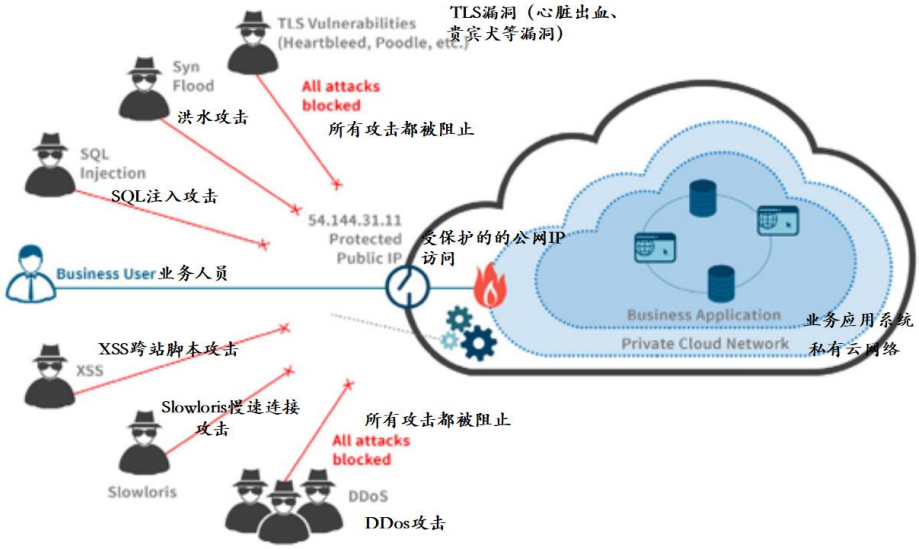
VDI：通过VDI，业务人员可以操作虚拟计算机（通常是Windows操作系统），这个虚拟机可以用作企业应用系统的启动平台。业务应用系统通常是一个需要“厚Windows客户端”（较多在客户端及服务器端的运算，较少的通信链接）的客户端/服务器应用程序。



使用SDP的访问



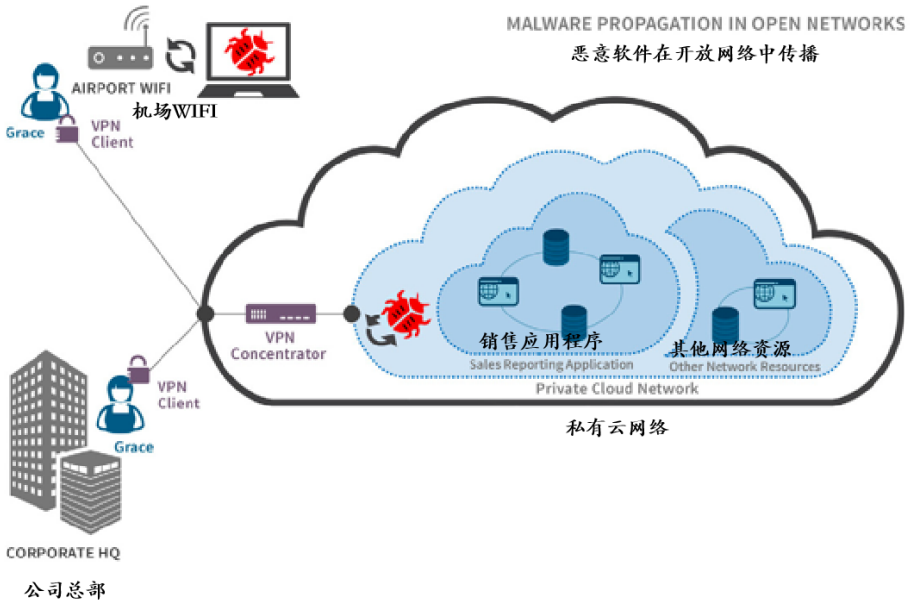
通过SDP解决方案，只有授权用户才能访问业务应用系统。实际上，未经授权的用户甚至无法访问SDP网关 – 因为它受到单包授权SPA的保护，所以对攻击者来说实际上是完全不可见。

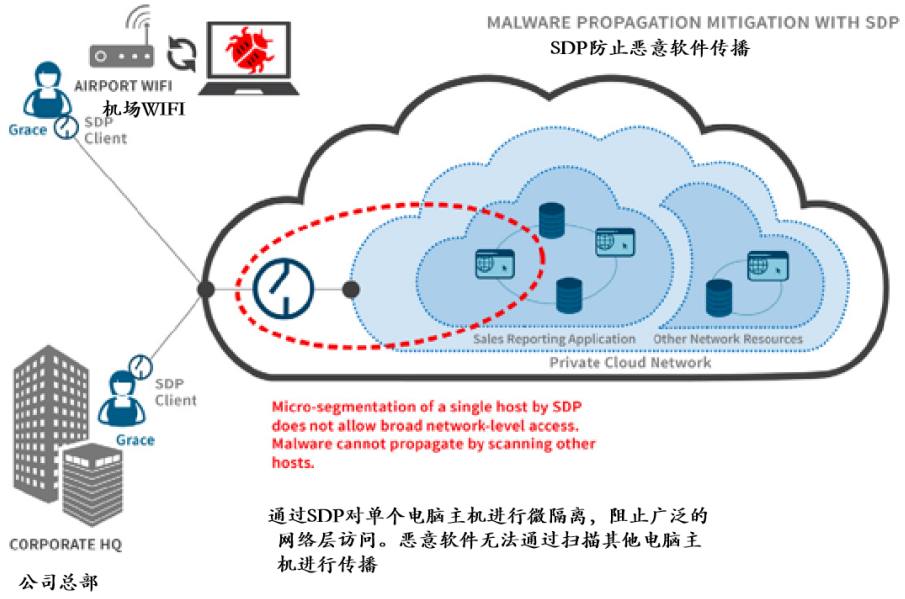


我们来看看不同的用户的访问需求，以及如何管理这种访问。

需求：Grace在公司总部的销售部门工作。她需要通过由IT团队开发的新销售报告系统访问重点客户销售报告。她经常拜访不同地方的客户，需要远程运行报告，且该应用系统托管在Amazon AWS上。

挑战：Grace 在出差期间在机场和咖啡店访问多个免费网络。过去，IT 安全部门发现了她的笔记本电脑上的恶意软件，他们担心当 Grace 通过 VPN 访问新的重点客户销售报告或返回公司总部时，恶意软件可能会传播到 AWS 基础架构中（“星期一恶意软件”）。





不使用 SDP	使用 SDP
方法：恶意软件不应该能够在网络内传播。因此，IT 安全人员创建了一个网络分段以隔离 AWS 上的应用服务器。	**方法：**SDP 提供的好处能够将网络隔离到单个服务器（而不是整个网络）。SDP 为 Grace 提供安全的远程访问，同时防止

	恶意软件通过公网访问带入[14]（例如 VPN）。
效果：使用传统网络工具创建微分段很复杂。随着应用程序数量的增长，ACL 记录的数量往往呈指数增长[15]。很快，网络管理员由于需要支持数千个 ACL 而负担过重。每个新建的 ACL 请求都需要数天才能分析和处理。生产力降低，IT 失去敏捷性，新应用的部署会变慢。	**效果：**业务人员现在可以完全访问托管在公有云网络上的企业应用系统。SDP在默认情况下提供有限制的网络访问，因此不违背最小权限访问的原则。 网络管理员能够阻止通过公网访问带入的恶意软件，不会因公司日益复杂的网络问题而负担过重。

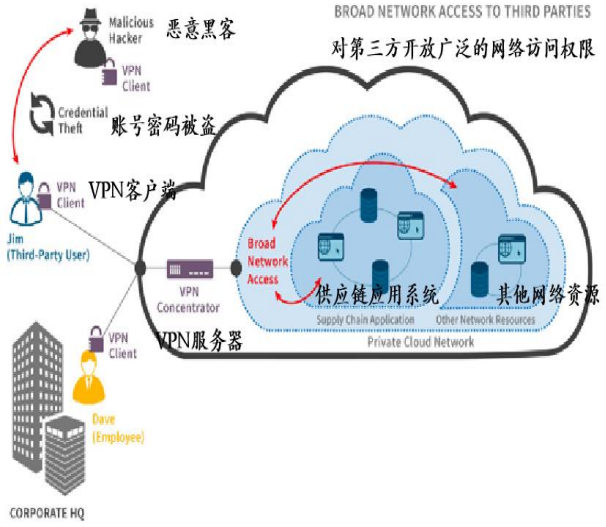
需求：Dave负责IT部门的供应链应用系统。新的业务流程要求其供应商员工Jim在货物发运后立即在其供应链应用系统中输入货件的详细信息。

挑战：这需要授予第三方供应商的一部分人员对应用系统的访问权限，这些业务人员（例如Jim）不是公司的员工，而Dave对其安全策略及安全培训等方面的控制是有限的。

Dave 不希望授予他们进入公司内网的广泛网络访问权限（VPN），他担心如果供应商端的账号被盗，他的整个公司网络将会受到伤害。他该如何限制“爆炸半径”？

14 使用 SDP 控制东西向流量将在 v2 SDP 规范中得到实质性解决。如本文档其他地方所述，这是 IaaS 环境与内部部署相比较容易解决的问题。

15 从技术上讲，O（n ^ 2）是为了建立应用程序到应用程序的连接模型。



不使用 SDP	使用 SDP
方法：Dave 为这一个应用服务器创建一个虚拟内网 VLAN，这样就能将其与其它网络隔离。但是，供应链应用系统非常复杂，并已集成到网络中的其他几个系统中。这里存在跨 VPC 的连接和防火墙规则，以及网络层访问控制表。因此，维护复杂的网络配置非常麻烦，因为任何系统中的 IP 地址更改都可能会导致整个供应链应用程序无法运行。	**方法：**使用 SDP 的情况下，他们可以从针对单个服务器进行网络隔离中获益（而不是隔离整个网络）。远程连接不会向供应商暴露任何其他网络资源。恶意的攻击者无法进行嗅探或探索网络中的其他漏洞或者易攻陷的资源。Dave 选择使用 SDP 来为供应商人员提供安全和保密的远程接入。

效果：Dave不再向第三方供应商提供应用系统访问权限，因为从网络安全角度来看其过于复杂且存在风险。供应链用户希望他们能更好的规划来自供应商的供应。这样做的影响是，上下游不再能够看到即将到来的发货数据，业务面临错误订单和延迟发货的风险。	**效果：**尽管 Dave 无法控制第三方供应商业务人员执行安全最佳实践和培训，但通过 SDP 他可以在发生账号失窃时限制影响范围。授予第三方访问权限变得更安全，不仅提高了供应链的效率而且同时保证业务增长。

需求：Jim每周都会准备一份业务分析报告，生成报告的是一个只能在Windows系统上运行的客户端/服务器(C/S)应用程序。所以IT部门为Jim和他的团队部署了一个VDI解决方案。Jim每次需要通过VDI登录远程桌面，然后启动报告程序。

挑战：这个（C/S）报告程序是从一个大型供应商处购买的打包的应用程序。建议的部署模式仅是“私有部署”，即供应商建议不要通过公共互联网访问应用程序的服务端，因为其并不稳定/安全。也就是说，服务器必须与客户端在同一网络内。

因此，对于远程用户，IT安全团队决定使用VDI，其中客户端和服务器始终保持在同一网络中。但是，构建和维护VDI服务器的成本非常高，并且会随着远程/出差雇员数量的增加而增加。

组织面临的挑战是如何安全地开放（C/S）应用程序的服务器部分，以便业务用户可以直接在他们自己的 Windows 笔记本电脑上运行客户端。

不使用 SDP	使用 SDP

方法： Jim 持续增加更多基础设施来支持 VDI 集群	**方法：** 使用SDP，Jim可以安全的向经过认证/授权的选定业务用户开放服务器/端口。对于其他人而言，服务器保持“不可见” Jim选择使用SDP来为有需求的业务分析用户提供 C/S 应用的安全远程接入。
效果： 建立一个大型的VDI集群既增加了用于硬件的资本支出，也增加了用于维护和保养VDI服务器的运营支出。 因此，IT 部门必须削减其他重要项目的预算，而VDI集群正在耗尽所有的资金。	**效果：** 使用SDP方法减少了维护VDI基础架构的成本。业务人员变得更加高效，因为他们运行应用程序之前省去了日常登录远程桌面的步骤。 不久以后，VDI 集群就可以淘汰，从而为其他设施腾出 IT 预算。

总结

在这个使用场景中，SDP为为企业提供了强大的优势：

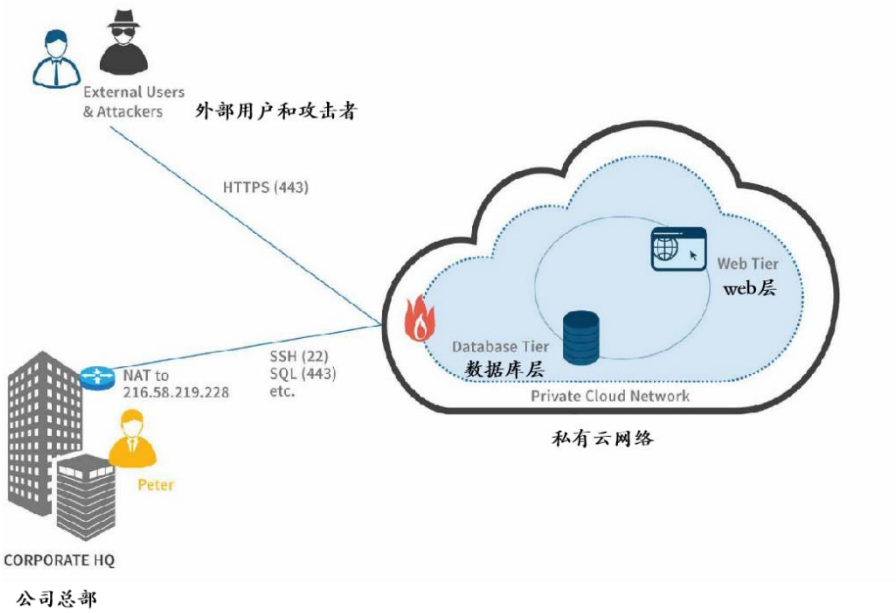
- 为远程业务用户提供安全访问企业应用的途径
- 精确控制用户可以访问的应用程序。
- 增加第三方业务集成。
- 更简单的合规报告
- 降低VDI相关基础设施成本
- 更简单的安全策略配置
- 提高业务流程的生产效率

使用场景：安全的管理面向公众的服务

当一个应用程序在云端提供服务时，系统管理员、开发人员以及其他高级用户都需要远程访问它的很多后端服务。这些服务可能包括：

- 通过SQL接口来访问数据库层（例如SQL Navigator for Oracle, PgAdmin for Postgres等）
- 通过SSH访问服务器
- 应用程序的管理员界面（例如WordPress博客的管理界面）
- 通过HTTPS访问数据库工具（例如PhpMyadmin for MySQL）
- 在这些案例中，在公共互联网上允许这些服务不受限制的访问是不明智的。暴露这些服务会导致暴力破解、错误配置、0day利用等攻击的几率上升。

不使用SDP时，这些后端服务需要被暴露到公共互联网，或者在云防火墙上手动维护限制某些源地址访问，但即使如此也有可能将服务暴露给许多用户。



当使用 SDP 方案时，只有那些需要公共访问的服务（如 HTTPS）才会暴露于互联网。所有其他服务都被 SDP 网关所隐藏，而且访问受到接入策略的控制，不需要接入额外的 VPN。

需求： Peter 是一个财务应用系统的数据库管理员。他的公司已经将他们的基础设施转移到一个行业领先的 IaaS 上。Peter 负责调整数据库 SQL 查询以改善应用性能。所以，他通过防火墙开放了数据库端口并且使用 SQL 浏览工具连接数据库来完成优化计划。

挑战： 现在数据库端口对外开放，恶意的自动机器人会迅速发现开放端口，并且尝试暴力破解管理员密码。他们有可能在几天内通过字典破解口令并获得公司的重要财务数据，也可能发现默认密码，或者利用已知的数据库平台漏洞（未被修复）进行攻击。

不使用 SDP	使用 SDP
方法：Peter 为云端网络设立 VPN	**方法：**使用 SDP，Peter 可以让端口对其他地方保持关闭。恶意黑客不会意识到数据库服务正在运行。数据库端口只能通过授权和认证后从 Peter 的设备访问。
效果：Peter 不得不在另一个不是他们原有数据中心私有网络上设立 VPN。他不得不在他的设备上配置两个不同的VPN并且每次访问网络资源时都需要选择连接到哪个 VPN。Peter 是一位 SQL 开发人员，并不是 IT 管理员，他不完全了解 VPN 配置。他并不喜欢会拖慢他的 VPN。	**效果：**Peter 不需要切换不同配置，也不需要配置网络安全策略就可以安全地访问数据库。

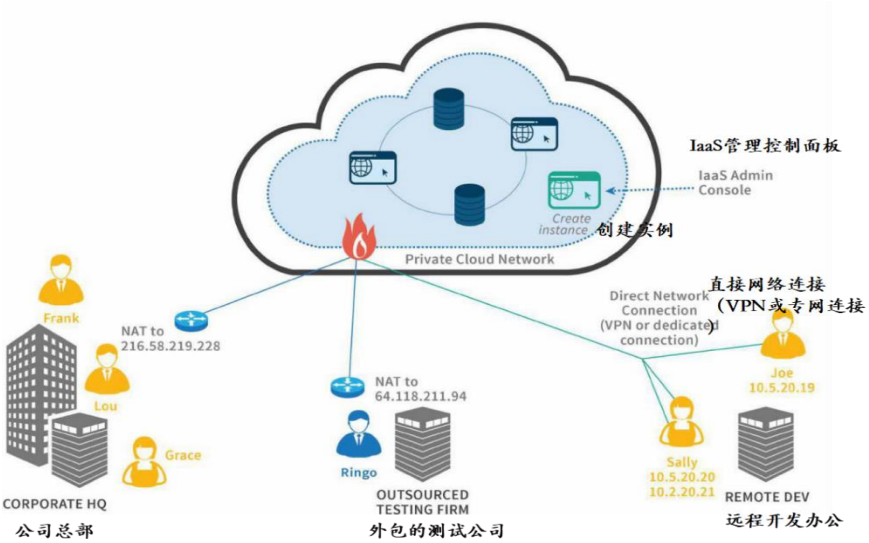
使用场景：当新服务实例创建时更新用户访问权限

云环境本质上是动态的，实际上，大多数企业利用IaaS的这一特点来提高其开发速度和敏捷性。特别是在IaaS环境中，创建和销毁服务器实例非常简单，所以企业可以频繁地(如果不是连续的)创建和销毁服务器实例[16]。

如下图所示，一个人使用IaaS控制台（或者调用API接口的系统）创建一个新的服务器实例。所需的网络更改取决于其位置，云连接类型和需求，并在以下页面的表中讨论。

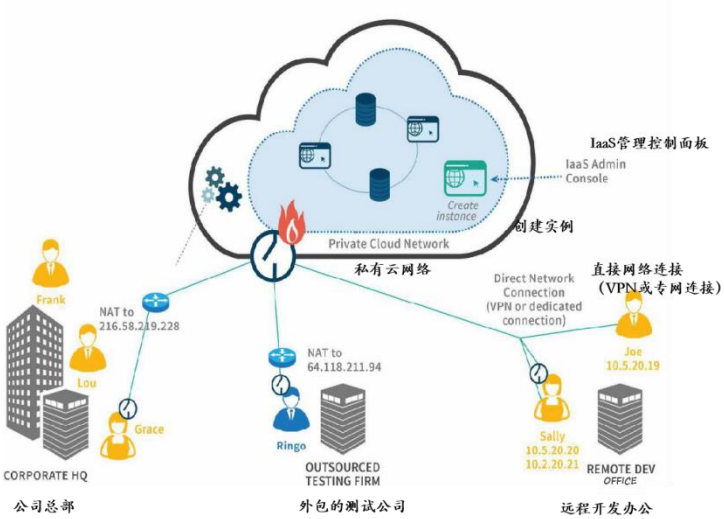
16 备注 1：对于这个用例，我们所描述的是由 IaaS 管理控制台手动启动的服务器实例，或者从开发或部署脚本通过 IaaS API 启动的服务器实例。此处不讨论 IaaS 基础架构自动扩展的服务器实例（例如，基于 CPU 负载阈值）。访问这些示例已在已有用例的覆盖范围内，因为它们实际上是纯粹用于负载平衡和扩展目的的已运行资源的克隆。对它们的访问将由与控制池中服务器核心集相同的访问策略来管理，本文档中的其他使用场景将对此进行介绍。

备注 2：由于通过将实例分配给云防火墙组来授予网络访问权限，因此在实例终止时无需执行任何操作即可移除访问权限。云防火墙并不授予特定服务器地址访问权限，因此也不存在由于 IP 地址重用导致错误继承访问权限。



用SDP接入

云服务器全部受 SDP 系统保护，网关充当私有云网络的唯一网络入口点。新的服务器实例具有元数据（标签）。



需求：Grace启动一个实例并且只需要SSH访问（端口22）

挑战：Grace位于一个通过NAT地址访问云的网络

不使用 SDP	使用 SDP
方法：这个实例必须分配一个公网 IP 地址，并且云防火墙必须授予 NAT 地址 216.58.219.228 或者整个互联网(0.0.0.0)对实例的完全访问。	**方法：**访问新的服务器实例必须通过SDP网关，SDP网关可公开访问并且受SPA的保护。 SDP 系统检测到这个新的服务器实例，并根据其元数据（标签）自动授予 Grace 对端口 22 的访问权限。

效果：虽然云防火墙不需要立即进行更改，但对任何用户或在公司网络中的设备来说，访问此服务器都是不受限的。这代表着重大的安全风险。 由于 IP 地址是 NAT 的，因此不可能将网络访问权限限制为单个用户或单个端口。因此，安全团队要求通过单独的密钥来控制对这些实例的 SSH 访问。管理和跟踪这些密钥文件是一件令人头疼的事情，并且对开发者来说也存在安全风险。	**效果：**Grace 会自动获得生产所需的最低访问权限，而无需任何手动重新配置或 IT 部门参与操作。

需求：Sally 启动一个实例，并需要从她的所有设备上访问 HTTPS，RDP 和 MySQL 端口。

挑战：Sally 通过她直连到云的办公室网络来访问云端——这些资源就好像在本地网络一样。

不使用 SDP	使用 SDP
方法：如果目的是只允许 Sally 访问，云防火墙则必须更新以允许 Sally 当前 IP 地址访问这个特定的服务器实例。如果目标是不产生任何延迟的情况下授予 Sally 访问权限，则必须将实例配置为允许所有本地网络上的设备访问新服务器实例的所有端口。	**方法：**SDP 系统检测到这个新的服务器实例，并根据其元数据（标签）自动授予 Sally 适当的端口访问权限。

效果：要求对云防火墙进行持续更改是大多数企业不愿意承担的运营负担，因为这会增加时间和成本。大多数企业授予开放式网络访问权限，而仅依靠认证进行控制。	**效果：**Sally 会自动获得必要的最低访问权限，无需任何手动重新配置或 IT 部门参与操作。

需求：Ringo 作为一个外包的测试人员，需要 Web 访问（443 端口）才能测试这个新实例。

挑战：Ringo 的办公室网络被 NAT 转换为一个不会改变的单一公共 IP 地址。由于 Ringo 在另一个时区，他有时必须在家工作实时与团队协作。

不使用 SDP	使用 SDP
方法：云防火墙必须允许 Ringo 从特定的公共 IP 地址访问。通过特定服务器分配给此云防火墙组，可以将其限制为特定的服务器实例。防火墙还必须允许 Ringo 从家用 IP 地址（定期更改）访问。	**方法：**SDP 系统检测到这个新的服务器实例，并根据其元数据（标签）自动授予 Ringo 对端口 443 的访问权限。
效果：用户启动的这个新服务器实例必须将其分配给允许从 Ringo 的 NAT 地址访问的安全组。所有使用 Ringo 的家用网络的用户也都可以通过端口 443 访问此实例。 每次 Ringo 的家用 IP 地址变化时，他必须向 IT 部门申请更新云防火墙更新。这可能需要长达 24 小时，并影响整个团队的生产效率。	**效果：**Ringo 自动获得必要的最低访问权限，无需任何手动重新配置或 IT 部门介入操作。 因为访问权限是授予作为用户的 Ringo，所以不会绑定到他的 IP 地址。这意味着 Ringo 能立即工作，并且无论他在哪里工作，都具有相同的安全访问权限。

总结

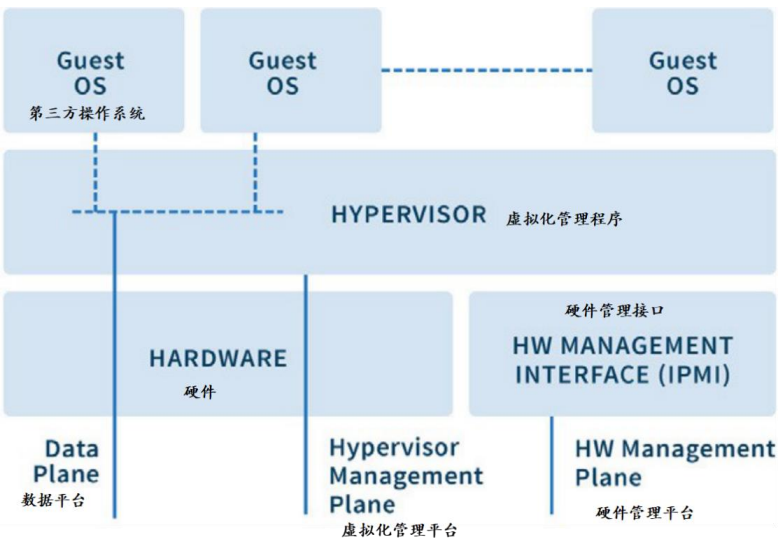
在这个使用场景中，SDP 为企业提供了强大的优势：

- 自动检测新服务器实例，并根据实例元数据自动分配用户访问权限
- 无论位置如何，都可确保开发者安全访问
- 完全的高效生产-不用等待网络访问控制更改的延迟
- 由策略驱动的访问控制，而不是基于云防火墙配置
- 减少 IT 运营工作量和成本

使用场景：对于服务提供商的硬件管理平台访问

尽管我们提出了一个无限可扩展且完全虚拟化的平台概念，但它实际上是运行在某个层面上的网络和计算硬件上。当然，这些硬件必须由服务提供商管理，无论他们是提供私有云平台的内部IT部门、服务器托管商或共址提供商、还是商业IaaS供应商。

下面的逻辑图说明了要管理的网络访问路径：



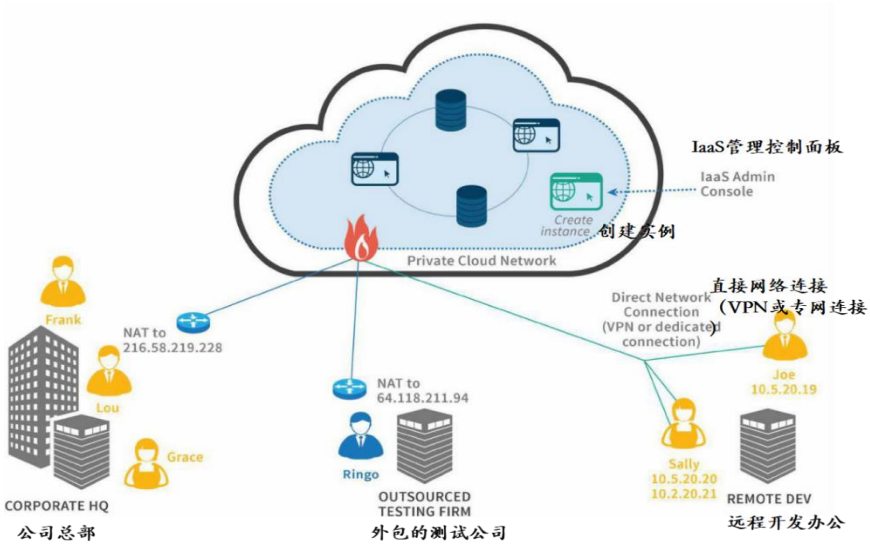
Data Plane数据平台是用于访问第三方操作系统实例的标准网络。我们在整个文档中的讨论都集中在这个网络上。

Hardware Management Plane 硬件管理平台是内置于许多硬件平台的网络，通常基于称为IPMI（智能平台管理接口）的英特尔规范构建。这可以通常称为底板管理控制器，或者非正式地称为“熄灯网络”。大多数服务器制造商通过具有其自身物理网络连接的嵌入式硬件卡，比较著名的如Dell服务器的DRAC）和HP的ILO for ILO都支持此功能。[17]

但是，这些IPMI服务因许多漏洞而众所周知，包括不修改的默认账号密码以及无法抵御简单攻击。

Hypervisor Management Plane 虚拟化管理平台是管理员通过GUI控制台或API访问读虚拟化进行管理。虽然虚拟机管理程序通常具有比IPMI系统更好的访问控制，但它们仍应配置为仅通过单独的网卡，在物理上独立的网络或虚拟局域网VLAN上进行访问，并受SDP保护。下面的讨论虽然侧重于IPMI，但也适用于虚拟化管理平台。

安全管理员访问各种端口上的IPMI网络接口。此访问必须具有强身



份验证，并且出于安全性和合规性报告目的而被记录。理想情况下，管理员需要全天候访问到这个网络。但是，这种按需访问通常具有时间敏感性，因为IT可能会响应服务器中断。同时应该有业务流程 – 例如请求和批准 – 来控制访问，并且使用闭环机制确保一旦不再需要访问就被删除。

IPMI 有许多已知的弱点，从可利用的漏洞到有限的管理功能。IPMI 需要单独的网络，无论是物理上分离还是通过 VLAN。

不使用 SDP	使用 SDP
方法：强烈建议不要使用默认方法，即依赖IPMI系统中的默认账号密码，并将访问IPMI网络的权限仅限于授权用户。 较好的方法虽然会产生很大的开销，但却是单独管理每个服务器的访问账号。 更好的方法是将 IPMI 身份验证与企业的 LDAP / RADIUS 系统联通。	**方法：**使用SDP，IPMI服务器可以简单地放置在受SDP网关保护的网段上。也就是说，除非SDP策略允许，否则任何网络流量都无法到达任何IPMI接口。 SDP 系统可以利用各种用户和系统属性 – 例如组成员资格、设备配置文件、位置或时间。
效果：这些解决方案都不够完善 – 保留IPMI系统的默认账号密码是在自讨苦吃 – 恶意攻击者很容易获得对网络的访问权限，例如通过错误的配置。 在每个服务器的基础上配置用户访问账号可以提供更好的安全性，但在任何规模的环境中都是行不通的 ，因为需要较多的手动工作和账号跟踪就来实现这一点。 利用组织的 LDAP / RADIUS 系统进行身份验证要好得多，但仍然要求任	**效果：**用户对IPMI接口的访问可以由策略驱动，并且可以根据即时的“按需授权”策略轻松动态调整。例如，SDP系统可以在允许用户访问之前，验证工单系统中是否存在特定用户和特定服务器的需求。这很容易支持对敏感授权的请求、批准等流程。 并且，SDP系统可以基于位置实施访问规则，例如仅允许从本地公司网络访问，以及阻止来自远程位置的任何访问。

何可能在某些时候需要 IPMI 访问权限的用户始终可以完全访问 IPMI 网络。通过防火墙规则控制网络访问在技术上是可行的,但会引入过多的进程开销,并会延迟对服务器的管理员访问。	SDP 还可以与组织的 IAM 系统集成，以实施强身份验证。

总结：

对于此场景，SDP 提供以下好处：

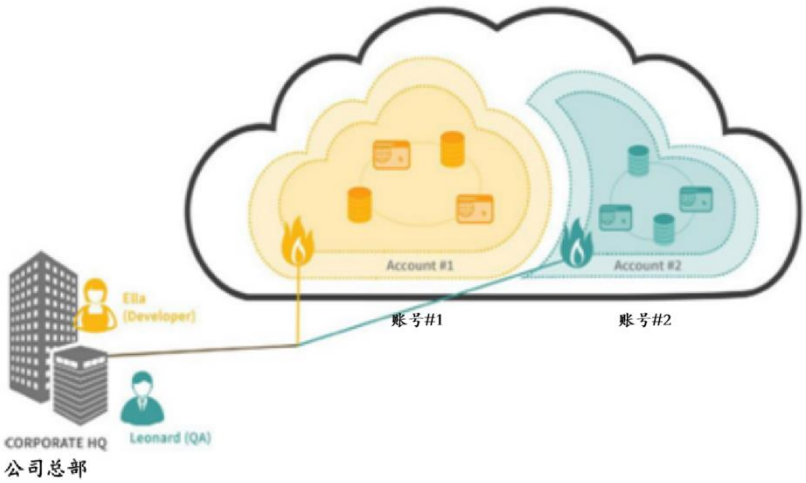
- 安全可控地访问高风险和易受攻击的 IPMI 网络
- 细粒度控制哪些用户可以访问这些系统，以及何时访问
- 通过简单的、以用户为中心的策略进行控制
- 通过与 IAM 集成实施强身份验证
- 在保障安全性或合规性的情况下，实现紧急服务器中断情况的快速访问
- 全面详细的日志，了解谁可以访问哪些系统（以及何时）以实现合规性
- 随着数据中心的增长和动态而扩展的解决方案

17 参考 https://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface 来了解更多信息。

18 参考 https://community.rapid7.com/community/metasploit/blog/2013/07/02/a-penetration-testers-guide-to-ipmi

使用场景：通过多企业账号控制访问

在此场景中，一个企业在IaaS上具有多个不同的帐户。出于安全性或合规性原因，这可能是故意的，或者由于不同的组独立设置帐户而偶然发生的的。下图显示了两个帐户的情况，但企业通常还有更多帐户，例如，如果他们采用了“一系统一个帐户”策略，则会有更多帐户。



从企业到云的连接（如上图中的墨绿色部分所示）可以通过共享云直连（实际上是专用站点到云的 VPN）或通过 Internet 进行连接。在任何一种情况下，都有相同的挑战，如上图所示。

要求：针对不同用户设置不同的访问权限，并且在该用户的所有帐号上保持一致。

挑战：虽然有多个单独的帐户可能在计费或访问云控制台，但它们不共享云防火墙。

不使用 SDP	使用 SDP
方法：如果企业试图严格控制用户对资源的访问，上述用例中提到的问题仍然存在，并且实际上随着帐户数量越来越严重。 他们可能会尝试通过使用云 API 来更新安全组策略以自动执行一些操作，但最终仍然面临着以 IP 地址为中心的云防火墙模型被应用到以用户为中心的安全控制的问题。	**方法：**通过在每个云防火墙前部署 SDP 网关，企业可以立即简化并提高其安全性。每个帐户的云防火墙可以简化为一个简单的规则 - 仅允许从 SDP 网关到受保护服务的连接。
效果：结果是企业实际上向网络上的所有用户开放所有云资源，只是依赖身份验证来保护它们。正如我们上面所述，这不是一个强有效的安全或合规的方案。	**效果：**无论使用的云帐户的数量或类型如何，都可以始终如一地应用 SDP 策略。这些策略提供完整的合规性跟踪，并始终与企业的 IAM 系统集成。

总结：

对于此例，SDP体现出明显的优势：

- 在不会牺牲安全性前提下，允许使用多个云提供商的账号
- 简化云防火墙规则相关的手动管理工作
- 始终对所有用户群实施所有帐户的访问策略

增强SDP规范的建议

在进行这项研究的整个过程中，我们有意地从一个从业者的角度来看SDP，查看实际可行的用例，这些案例通常是由可行的SDP实现和体系结构所支持的。因为这些很有必要，所以这些用例和需求超出了当前（V1）SDP

规范。

在我们的辩论、对话和写作中，我们会记录那些我们认识到的一个完整的SDP实现所需的领域（如SDP供应商产品已经解决的问题所证明的）。

这些话题中的许多超出了IaaS，它们是整个SDP规范的必要一部分，然而我们决定它们不在本文档的范围内，而是放到当前正在进行的V2规范中。我们期待着接下来的创作，就以下主题进行讨论：

- 网络权限的策略模型
- IAM 集成
- 目录属性和 SAML 验证作为策略模型的一部分
- 逐步认证触发器
- 高可用性和负载均衡方法
- 除了 HTTP 外的网络协议支持（如 SSH、UDP）
- 深度包检测和会话代理
- 增量部署到现在的企业环境
- 实例元数据和云环境自动探测
- SDP 成本节约/ROI 模型
- 探讨服务器与服务器之间的流量微隔离（东西向流量）
- 遗留的风险和潜在的 SDP 系统威胁（例如，中间人攻击、控制器或网关攻击）

混合云以及多云的环境

在可预见的未来中，多数企业都拥有一个复杂的IT环境。安全团队与其将此看成是一个需要消除的问题，不如去拥抱这种丰富的场景，因为这是商业与生俱来的复杂性。不同行业的商业有不同需求，我们可以非常负责任地预测，世上没有“放之四海而皆准”的IT架构可以适用所有的企业。

这表明安全团队需要寻找正确类型的工具和技术来为不同的环境提供持续的安全保障。虽然总是有不少平台强关联的工具，例如系统管理、自动化、或者是终端管理，但我们相信从安全的角度来说，企业在不同平台上建立统一的以用户为中心的策略和流程是非常关键的。

例如，企业肯定非常希望有一个统一的平台，他们可以定义并且执行“谁可以访问什么系统”的策略和流程。这个平台必须是统一管理内网部署的、多地部署的、物理的、虚拟的、私有云的、公有云的资源。如果不是这样，企业将面临增加的复杂性、风险、以及运维成本。

我们相信SDP，因为它以用户为中心、平台无关、网络层访问控制强制执行的能力，是今天企业解决复杂环境下安全问题的正确选择。

替代计算模型和SDP

我们看到“无服务器”计算模型的可用性和采用率正在稳步提高，云提供商在其功能线中添加了新的‘PaaS’。这些是对传统（如关系数据库或消息队列）的“as a service”转变到更新颖的“function as a service”（例如AWS Lambda、Azure Functions、以及 Google Cloud Functions），以及其他许多新的以物联网为中心的业务。

所有这些共同点是它们不向客户公开传统操作系统，这意味着要解决的网络访问控制问题可能与IaaS平台相关的问题不同。

在某些情况下，这些服务完全符合我们在此讨论的IaaS场景。例如，作为服务的关系数据库恰好是受SDP保护的服务类型。实际上，许多IaaS提供商使用相同的网络访问模型来控制对其IaaS实例的关系实例的访问，因此我们在此描述的SDP方法是完全相关的。

在其他情况下，其中一些服务使用其他安全模型。例如，“function as a service ”通常可以通过公开暴露的URL或通过某种API网关访问。由于客户端到网关到服务器的模型没有任何意义，今天可能与SDP方法不兼容。我们相信这些模型将会像SDP一样发展，并且这将成为未来一个有趣的领域。

无论如何，如果您的企业正在使用（或考虑）其中一些替代计算模型，请确保您和您的安全团队与开发人员进行互动，以了解该工具的安全模型，以及如何与您的安全架构向适应。

容器和SDP

容器是另外一个正在快速发展的趋势，许多企业采用它们作为基础技术，实现高速的DevOps方案/周期。容器带给他们一些有趣的新的安全和访问挑战。对于不同的容器和集群技术，当然有不同的网络访问模型，但为了简化起见，它们映射到以下方面：

每个pod群集（单个OS进程中的一组容器）获取一个由其容器共享的公共IP（Kubernetes模型）

每个容器都有一个私有IP，它被NAT连接到pod群集的公共IP（Docker模型）。

在这两种情况下，SDP都可以有效地应用。pod群集和它们的容器可以放在SDP网关后面，SDP制定策略来控制用户对服务的访问。受保护的服务对应于容器内动态解析的IP地址或元数据，就像IaaS环境一样。从端口到容器的任何特定于pod群集的映射都可以在SPD网关后面工作，添加SDP没有任何影响。

当然，还有其他方法可以在容器内联网，因此请仔细查看您的团队使

用的工具。但总的来说，上面列出的主流方法与SDP兼容，实际上可以很好地与SDP配合使用。这是另一个未来研究和验证的领域。

结论与下一步计划

无论你是一个企业、一个服务提供者、还是一个独立的实践者，我们都希望这项研究能够给您带来帮助。该文档将提高您对与IaaS环境相关的特定网络访问所面临的挑战，并通过软件定义边界SDP来帮助您解决这些问题。

我们希望您能够不仅仅将IaaS资源视为内部网络的扩展。拥抱云有很多好处，但往往需要很多改变才能充分利用。我们希望这篇研究能帮助您对云有不同的看法，并且改变用户访问这些资源的方式，使其更安全、更灵活、更高效。

我们相信SDP在安全方面是一个重要的进步——这是第一次使动态的、以身份为中心的安全性被应用在网络层上，并且我们热衷于看到它更广泛地被企业所接受，以满足当今的安全和业务需求。正如Gartner所说的那样：

“连接复杂性使得旧的安全体系不可持续，这需要一种新的方法来满足数字业务对复杂性、大流量和灵活性的需求，同时避免从旧模型中继承漏洞。”[19]

当然，SDP并不能解决所有的安全问题——有很多信息安全问题并不在SDP的范围内，也有可能会从特定产品中产生残余风险，或者由企业实施的细节产生。

但是总的来说，软件定义边界作为一种新的方法，不仅适用于当前的IaaS环境，而且正在重塑下一代网络安全解决方案。

19 Gartner:《迎接新时代：隔离互联网污染环境与你的网络服务》，2016年9月30日。

谷歌 BeyondCorp

系列论文合集

CSA 大中华区 SDP 工作组

奇安信身份安全实验室　　译

二零一九年五月

前言：

随着企业大规模的采用移动互联网和云计算技术，传统的采用防火墙建立的“城堡”安全模式，变得越来越不安全。2014 年 12 月起，Google 先后发表 6 篇 BeyondCorp 相关论文，论文提供了一种新的安全模式，设备和用户只能获得经过验证的资源，构建软件定义安全的雏形。另外，论文也介绍了 BeyondCorp 的架构和实施情况，为传统网络架构迁移至 BeyondCorp 架构提供依据参考。

为推动国内安全技术和理论与国际同步，在国内传播国际优秀实践，中国云安全联盟秘书处组织专家翻译 BeyondCorp 相关论文，供大家学习参考。特别感谢 CSA 大中华区 SDP 工作组与奇安信身份安全实验室对本次翻译工作的贡献及支持！

本文档一共由 BeyondCorp 的六篇论文组合而成：

[1] BeyondCorp：一种新的企业安全方案

[2] 谷歌 BeyondCorp：从设计到部署

[3] BeyondCorp：访问代理

[4] 迁移到 BeyondCorp：提高安全性的同时保持生产力

[5] BeyondCorp ：用户体验

[6] BeyondCorp：构建健康机群

关于 CSA 大中华区 SDP 工作组：

CSA （Cloud Security Alliance）2008 年 12 月在美国发起，以云计算安全为开端，2011 年白宫在 CSA 峰会上宣布了美国联邦政府云计算战略，之后 CSA 演化成为独立、中立、非盈利的世界性产业组织，致力于全球下一代 IT 与新兴技术安全的全面发展。

为提高 Software Defined Perimeter（软件定义边界，即 SDP）在中国企业的应用，在中国云安全联盟的支持下 ，CSA 大中华区成立 SDP 工作组。工作组于 2019 年 3 月成立，首批参与单位有：阿里云、腾讯云、京东云、奇安信、深信服、绿盟科技、Ucloud、顺丰科技、天融信、云深互联、中宇万通、华云数据、三未信安、上元信安、安全狗、易安联、联软科技、上海云盾、缔盟云、缔安科技等三十多家单位。

关于奇安信身份安全实验室：

奇安信身份安全实验室，是奇安信集团下属专注“零信任身份安全架构”研究的专业实验室。该团队以“零信任安全，新身份边界”为技术思想，探索“企业物理边界正在瓦解、传统边界防护措施正在失效”这一时代背景下的新型安全体系架构，推出“以身份为中心、业务安全访问、持续信任评估、动态访问控制”为核心的奇安信天鉴零信任身份安全解决方案。该团队结合行业现状，大力投入对零信任安全架构的研究和产品标准化，积极推动“零信任身份安全架构”在业界的落地实践，其方案已经在部委、金融、央企等进行广泛落地实施，得到市场、业界的高度认可。

参与本文档翻译的专家（按照姓名拼音排序）：

组长：陈本峰（云深互联）

组员：高健凯、刘德林、张泽洲（奇安信）

【第一篇】BeyondCorp：一种新的企业安全方案

如今，几乎所有企业都会采用防火墙来建立安全边界，然而，这种安全模型存在问题：一旦边界被突破，攻击者可以畅通无阻地访问企业的特权内部网络。另一方面，随着企业大规模地采用移动互联网和云计算技术，边界防护变得越来越难。谷歌采用了不同的网络安全方法，逐步摆脱对特权内网的依赖，越来越多地将企业应用程序从内网迁移至公网。

从 IT 基础设施诞生以来，企业一向使用边界防御措施来保护对内部资源的访问。边界安全模型通常被比作中世纪的城堡：有着厚厚的城墙，被护城河环绕，仅有一个守卫森严的入口和出口，任何墙外的东西都被认为是危险的，任何墙内的东西都认为是安全可信的，这也就意味着任何能通过吊桥的人都能获得城堡内的资源。

当所有员工都只在企业办公大楼中工作时，边界安全模型确实很有效；然而，随着移动办公的出现、办公使用的设备种类激增、云计算服务的使用越来越广泛、新的攻击向量也随之增加，如上因素逐渐导致传统安全手段形同虚设。边界安全模型所依赖的关键假设不再成立：边界不再由企业的物理位置决定，边界之内也不再是个人设备和企业应用运行的安全地带。

大部分企业假设内部网络是安全的环境并且企业应用可以放心暴露在内网，但谷歌的经验证明了这种观念是错误的。应该假设企业内网与公网一样充满危险，并基于这种假设构建企业应用。

谷歌 BeyondCorp 的目标是摒弃对企业特权网络（内网）的依赖并开创一种全新安全访问模式，在这种全新的无特权内网访问模式下，访问只依赖于设备和用户身份凭证，而与用户所处的网络位置无关。无论用户是

在公司“内网”、家庭网络、酒店还是咖啡店的公共网络，所有对企业资源的访问都要基于设备状态和用户身份凭证进行认证、授权和加密。这种新模式可以针对不同的企业资源进行细粒度的访问控制，所有谷歌员工不再需要通过传统的 VPN 连接进入内网，而是允许从任何网络成功发起访问，除了可能存在的网络延迟差异外，对企业资源的本地和远程访问在用户体验上基本一致。

BeyondCorp的关键组件

BeyondCorp 由许多相互协作的组件组成，以确保只有通过严格认证的设备和用户才能被授权访问所需的企业应用，各组件描述如下（见图 1）。

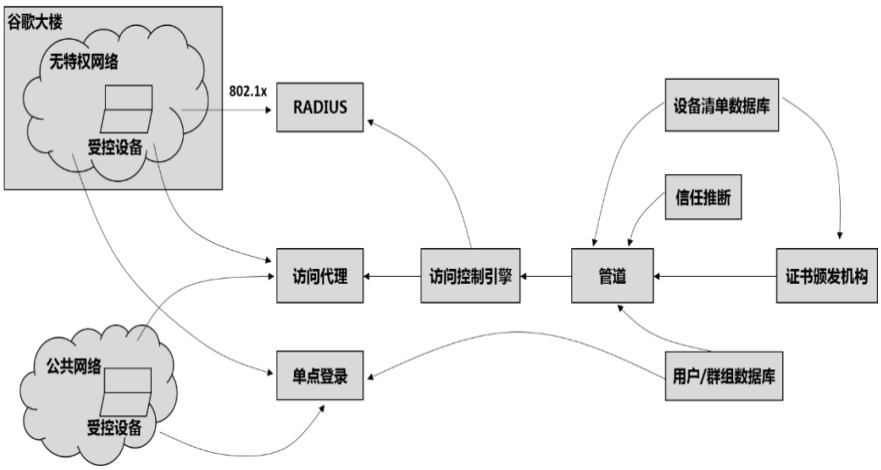


图 1 BeyondCorp 的组件和访问流

安全识别设备

设备清单数据库

BeyondCorp 使用了“受控设备”的概念——由企业采购并管理可控的

设备。只有受控设备才能访问企业应用。围绕着设备清单数据库的设备跟踪和采购流程管理是这个模型的基础之一。在设备的全生命周期中，谷歌会追踪设备发生的变化，这些信息会被监控、分析，并提供给 BeyondCorp 的其他组件进行分析和使用。因为谷歌有多个清单数据库，所以需要使用一个元清单数据库对来自多个数据源的设备信息合并和规一化，并将信息提供给 BeyondCorp 的下游组件。通过元清单数据库，我们就掌握了所有需要访问企业应用的设备信息。

设备标识

所有受控设备都需要一个唯一标识，此标识同时可作为设备清单数据库中对应记录的索引值。实现方法之一是为每台设备签发特定的设备证书。只有在设备清单数据库中存在且信息正确的设备才能获得证书。证书存储在硬件或软件形态的可信平台模块（Trusted Platform Module, TPM）或可靠的系统证书库之中。设备认证过程需要验证证书存储区的有效性，只有被认为足够安全的设备才可以被归类为受控设备。当进行证书定期轮换时，这些安全检查也会被执行。一旦安装完毕，证书将用于企业服务的所有通信。虽然证书能够唯一地标识设备，但仅凭证书不能获取访问权限，证书只是用来获取设备的相关信息。

安全识别用户

用户和群组数据库

BeyondCorp 还跟踪和管理用户数据库和用户群组数据库中的所有用

户。用户/群组数据库系统与谷歌的 HR 流程紧密集成，管理着所有用户的岗位分类、用户名和群组成员关系，当员工入职、转岗、或离职时，数据库就会相应更新。HR 系统将需要访问企业的用户的所有相关信息都提供给 BeyondCorp。

单点登录系统

外化的单点登录（SSO）系统是一个集中的用户身份认证门户，它对请求访问企业资源的用户进行双因子认证。使用用户数据库和群组数据库对用户进行合法性验证后，SSO 系统会生成短时令牌（short-lived tokens），用来作为对特定资源授权流程的一部分。

消除基于网络的信任

部署无特权网络

为了不再区分内部和远程网络访问，BeyondCorp 定义并部署了一个与外网非常相似的无特权网络，虽然其仍然处于一个内网的地址空间。无特权网络只能连接互联网、有限的基础设施服务（如，DNS、DHCP 和 NTP）、以及诸如 Puppet 之类的配置管理系统。谷歌办公大楼内部的所有客户端设备默认都分配到这个网络中，这个无特权网络和谷歌网络的其他部分之间由严格管理的 ACL（访问控制列表）进行控制。

有线和无线网络接入的 802.1x 认证

对于有线和无线接入，谷歌使用基于 802.1x 认证的 RADIUS 服务器将设备分配到一个适当的网络，实现动态的、而不是静态的 VLAN 分配。这种方法意味着不再依赖交换机/端口的静态配置，而是使用 RADIUS 服务器来通知交换机,将认证后的设备分配到对应的的 VLAN。受控设备使用设备证书完成 802.1x 握手，并分配到无特权网络，无法识别的设备和非受

控设备将被分配到补救网络或访客网络中。

将应用和工作流外化

面向公共互联网的访问代理

谷歌的所有企业应用都通过一个面向公共互联网的访问代理开放给外部和内部客户。通过访问代理，客户端和应用之间的流量被强制加密。一经配置，访问代理对所有应用都进行保护，并提供大量通用特性，如全局可达性、负载平衡、访问控制检查、应用健康检查和拒绝服务防护。在访问控制检查（详述见后文）完成之后，访问代理会将请求转发给后端应用。

公共的 DNS 记录

谷歌的所有企业应用均对外提供服务，并且在公共 DNS 中注册，使用 CNAME 将企业应用指向面向公共互联网的访问代理。

实现基于设备清单的访问控制

对设备和用户的信任推断

每个用户和/或设备的访问级别可能随时改变。通过查询多个数据源，能够动态推断出分配给设备或用户的信任等级，这一信任等级是后续访问控制引擎（详述见后文）进行授权判定的关键参考信息。

例如，一个未安装操作系统最新补丁的设备，其信任等级可能会被降低；某一类特定设备，比如特定型号的手机或者平板电脑，可能会被分配特定的信任等级；一个从新位置访问应用的用户可能会被分配与以往不同的信任等级。信任等级可以通过静态规则和启发式方法来综合确定。

访问控制引擎

访问代理中的访问控制引擎，基于每个访问请求，为企业应用提供服务级的细粒度授权。

授权判定基于用户、用户所属的群组、设备证书以及设备清单数据库中的设备属性进行综合计算。如果有必要，访问控制引擎也可以执行基于位置的访问控制。另外，授权判定也往往参考用户和设备的信任等级，例如，可以限制只有全职工程师、且使用工程设备才可以登录谷歌的缺陷跟踪系统；限制只有财务部门的全职和兼职员工使用受控的非工程设备才可以访问财务系统。访问控制引擎还可以为应用的不同功能指定不同的访问权限和策略，例如，在缺陷跟踪系统中，与更新和搜索功能相比，查看某一条记录可能不需要那么严格的访问控制策略。

访问控制引擎的消息管道

通过消息管道向访问控制引擎源源不断地推送信息，这个管道动态地提取对访问控制决策有用的信息，包括证书白名单、设备和用户的信任等级，以及设备和用户清单库的详细信息。

一个端到端示例

应用

本例中，我们假设一个应用将被“BeyondCorp 化”，这个应用用于工程师审核、注释、更新源代码，并且经审核者批准后可以提交代码。进一步假设权限设定为：允许全职和兼职工程师从任何受控设备上对这一应用(codereview.corp.google.com)进行访问。

配置面向互联网的访问代理

codereview.corp.google.com 的所有者为这一服务配置访问代理。配置

指定了应用后端的网络位置和每个后端可承受的最大流量。codereview.corp.google.com 的域名在公共 DNS 中注册，其 CNAME 指向访问代理。例如：

```
$ dig @8.8.8.8 codereview.corp.google.com

; <<>> DiG 9.8.1-P1 <<>> @8.8.8.8 codereview.corp.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12976
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0,
ADDITIONAL: 0

;; QUESTION SECTION:
;codereview.corp.google.com. IN  A

;; ANSWER SECTION:
codereview.corp.google.com. 21599 IN  CNAME
accessproxy.l.google.com.
accessproxy.l.google.com.    299   IN  A     74.125.136.129

        ;; Query time: 10 msec
        ;; SERVER: 8.8.8.8#53(8.8.8.8)
        ;; WHEN: Wed Aug 20 19:30:06 2014
        ;; MSG SIZE rcvd: 86
```

配置访问控制引擎

访问控制引擎提供了一个默认规则，限制只有全职员工使用受控设备才能进行访问。codereview.corp.google.com 的所有者可以提供一个更具化的规则进一步限制针对此应用的访问，包括：限制只有最高信任等级的受控设备可以访问、限制只有最高信任等级的全职和兼职工程师可以访问。

当一位工程师访问网络

如果网络位于企业办公大楼外：工程师使用谷歌配发的笔记本电脑，接入任何 Wi-Fi 网络，这个网络可能是一个有登录验证门户的机场 Wi-Fi，也可能是咖啡馆的公共 Wi-Fi。不再需要配置和通过 VPN 来连接到企业网络。

如果网络位于企业办公大楼内：工程师使用谷歌配发的笔记本电脑访

问企业网络，这台电脑在与 RADIUS 服务器进行 802.1x 握手过程中提供设备证书。当证书有效时，为这台电脑在无特权网络上分配一个地址；如果电脑不是由公司配发的或者其设备证书过期了，就为这台电脑分配一个补救网络上的地址，而且这个地址的访问权限非常有限。

访问应用，无需区分网络

工程师从公司配发的笔记本电脑上访问 codereview.corp.google.com，读者可以参考图 1 的访问流程。

1、请求指向访问代理，笔记本电脑提供设备证书。

2、访问代理无法识别用户，重定向到单点登录系统。

3、工程师提供双因子认证凭据，由单点登录系统进行身份认证，颁发令牌，并重定向回访问代理。

4、访问代理现在持有标识设备的设备证书，标识用户的单点登录令牌。

5、访问控制引擎为 codereview.corp.google.com 执行对应的授权检查。授权检查基于每个请求进行：

a) 确认用户是工程组成员。

b) 确认用户拥有足够的信任等级。

c) 确认设备是一个良好的受控设备。

d) 确认设备拥有足够的信任等级。

e) 如果所有这些检查通过，请求被转发到应用后端获取服务。

f) 如果上述任何检查失败，请求被拒绝。

基于上述方法和流程实现了丰富的、服务级的认证及针对每个请求的授权检查。

迁移到BeyondCorp

与世界上几乎所有企业一样，多年来谷歌一直为其用户和应用维护一个特权网络（内网），这种模式使得基础设施对公司的日常运作至关重要。尽管公司的所有组件都应该迁移到 BeyondCorp，但一下子将每个网络用户和每个应用都迁移到 BeyondCorp 环境，对业务连续性来说非常危险。因此，谷歌投入大量资源进行分阶段迁移，在不影响公司生产力的情况下，成功地将大批网络用户逐步迁移到 BeyondCorp，如图 2 所示。下面将详细介绍我们所做的一些工作。

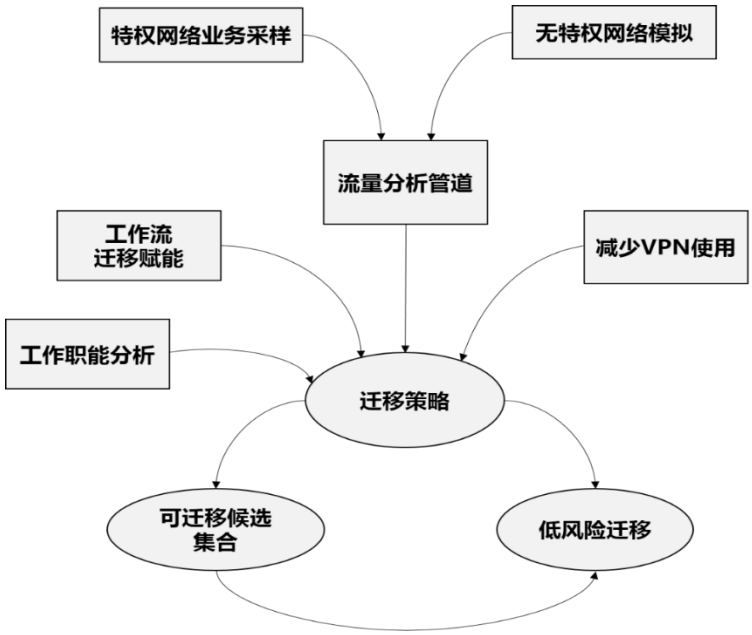


图 2 迁移到 BeyondCorp

工作流迁移评估

需要确保谷歌的所有应用都可以迁移以便最终通过访问代理进行访问。BeyondCorp 会发起对所有应用的检查和迁移评估，确保其平滑迁移。每个应用要实现迁移的难易不一，有的只需要简单的支持 HTTPS 流量，有的比较复杂，如，需要实现单点登录集成。需要对每个应用配置访问代

理，在大多数情况下还需要在访问控制引擎中进行特定的策略配置。每个应用都会经历以下阶段实现迁移：

1、可以通过特权网络直接访问；以及在外网通过 VPN 访问。

2、可以通过特权网络直接访问；以及在外网及无特权网络中通过访问代理访问。此阶段需要使用 DNS 分离解析，内部域名服务器直接指向应用，外部域名服务器指向访问代理。

3、在外部、特权和无特权网络中通过访问代理均可访问。

工作职能分析

通过检查整个公司的工作职能，并和工作流迁移评估进行交叉对比，我们能够确定用户群组迁移的优先级。基于当时对用户工作流和 BeyondCorp 组件功能的全面理解，我们从财务、销售、法务或工程师团队中选择网络用户进行迁移。

减少 VPN 的使用

随着越来越多的应用通过访问代理访问，我们开始阻止用户使用 VPN，策略如下：

1、只有经证实确有需要的用户才能使用 VPN 访问。

2、监控 VPN 的使用，删除在一段时期内未使用 VPN 的用户的访问权限。

3、监控 VPN 活跃用户的 VPN 使用情况，如果他们所有的工作流都可以通过访问代理实现，将强烈建议用户放弃使用 VPN。

流量分析管道

只有当我们确信（或者非常接近确信）所有用户的工作流都可以从无特权网络中访问时，才能将用户转移到无特权网络，这一点对迁移的平滑

性非常重要。为了建立一个相对的确定性，我们建立了一个流量分析管道。从公司的每个交换机中采样网络流量数据并输入管道，将这些数据与无特权网络和公司其余网络之间的预设访问控制列表对比分析，通过分析，能够识别命中访问控制列表的流量，以及没有命中访问控制列表的流量，并分别形成列表。对于没有命中访问控制列表的“逃逸”流量被关联到特定的工作流和/或特定的用户和/或特定的设备上，并进一步解决这些“逃逸”流量的问题，使其在 BeyondCorp 环境中能够工作。

无特权网络模拟

作为补充，除了通过交换机采样流量并进行流量分析外，我们还通过安装在访问谷歌网络所有用户设备上的流量监视器，对整个公司的无特权网络行为进行模拟。流量监视器检查了每个设备的所有流入和流出的流量，与无特权网络和公司网络其余部分之间的预设访问控制列表对比验证，并记录没有通过验证的非法流量。流量监视器有两个模式：

- 记录模式：捕获非法流量，但仍然允许上述流量流出设备。
- 强制模式：捕获并丢弃非法流量。

迁移策略

通过流量分析管道和无特权网络模拟，可以定义并实施分阶段的迁移策略，包括以下内容：

1、通过工作职能和/或工作流和/或位置来确认潜在的可迁移候选集。

2、模拟器开启记录模式，确认在连续 30 天内合格流量比例大于 99.9%的用户和设备。

3、如果在该时期内，用户和设备的合格流量比例大于 99.99%，

则为用户和设备启动强制模式。当然，若有必要，用户可以将模拟器恢复到记录模式。

4、在强制模式下成功运行 30 天之后，将此状态记录在设备清单中。

5、包含在候选集中，且在模拟器执行模式下成功工作 30 天是一个非常强烈的合格信号，下一次 Radius 服务器提供 802.1x 身份验证服务时，设备将被分配到无特权网络。

豁免处理

除了尽可能自动化地将用户和设备从特权网络转移到新的无特权网络外，我们还采用了一个简单的办法允许用户请求临时免除这种迁移。我们维护了一个未获得 BeyondCorp 能力评估、尚未达到迁移标准的工作流列表。用户可以搜索这些工作流，在经过适当的审批后，将自己和设备标记为特定工作流的活跃用户。当工作流完成 BeyondCorp 赋能，达到迁移标准后，相关用户会收到通知，再次进入迁移候选名单并进行迁移。

完成 BeyondCorp

谷歌的 BeyondCorp 迁移正在进行中，所需的大部分工作流已经评估确认完毕。我们的迁移工具和策略允许主动将用户、设备和工作流迁移到 BeyondCorp，而不会影响日常工作生产力。

我们预计 BeyondCorp 迁移的收尾工作还很多，需要花费一段时间。例如，使用专有协议与服务器交互的客户端应用将是一个挑战。我们正在研究将此类应用迁移到 BeyondCorp 的方法，也许会为它们配套使用一种特殊的认证服务。

随着我们向 BeyondCorp 迁移工作的推进，我们打算后续发表一系列

文章解释谷歌为何以及如何向 BeyondCorp 迁移，同时也希望可以鼓励其他企业实施类似的实践。

【第二篇】谷歌BeyondCorp：从设计到部署

谷歌 BeyondCorp 项目的目标是为了提高员工和设备访问企业内部应用的安全性。与传统的边界安全模型不同，BeyondCorp 不基于物理位置或发起请求的网络位置来授予用户访问服务和应用的权限；相反，访问策略的制定完全基于设备的信息、状态和当前设备的使用者信息等等。BeyondCorp 模型默认内部网络和外部网络都是不可信的，需要动态评估当前访问请求的安全等级，并确保此等级满足应用的最低安全要求。

本文将介绍谷歌如何从传统的安全基础设施迁移到 BeyondCorp 模式，以及在迁移过程中所面对的挑战和获得的经验教训。关于 BeyondCorp 的架构讨论，请参见第一篇“BeyondCorp：一种新的企业安全方案”[1]。

概述

BeyondCorp 系统的关键组件包括信任引擎（Trust Inferer）、设备清单服务(Device Inventory Service)、访问控制引擎(Access Control Engine)、访问策略(Access Policy)、网关（Gateways）和资源(Resources)，如图 1 所示，BeyondCorp 所使用的各术语定义如下：

- 访问需求被划分为不同的**信任等级（Trust Tiers）**，不同的等级代表着不同的敏感度，等级越高，敏感度越高。
- **资源(Resources)**代表所有访问控制机制将覆盖的应用、服务和基础设施。包括在线知识库、财务数据库、链路层访问、实验室网络等等，需要为每个资源都分配一个访问所需的最小信任等级。
- **信任引擎（Trust Inferer）**是一个持续分析和标注设备状态的系统。该系统可设置设备可访问资源的最大信任等级，并为设备分配对应的

VLAN，这些数据都会记录在设备清单服务中。任何设备状态的更新，或者信任引擎无法接收到设备的状态更新消息，都会触发对其信任等级的重新评估。

- **访问策略(Access Policy)**是描述授权判定必须满足的一系列规则，包含对资源、信任等级和其他影响授权判定的因子的程序式表示。
- **访问控制引擎(Access Control Engine)**是一种集中式策略判定点，它为每个访问网关提供授权决策服务。授权过程一般基于访问策略、信任引擎的输出结果、请求的目标资源和实时的身份凭证信息，并返回成功/失败的二元判定结果。
- **设备清单服务(Device Inventory Service)**是 BeyondCorp 系统的中心，它不断收集、处理和发布所有在列设备状态的变更。
- **网关（Gateways）**是访问资源的唯一通道，如 SSH 服务器、Web 代理或支持 802.1x 认证的网络等。网关负责对授权决策进行强制执行，例如强制最低信任等级或分配 VLAN。

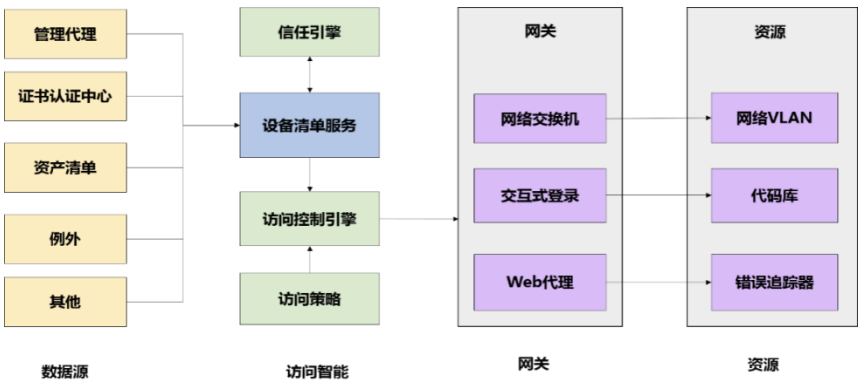


图 1: BeyondCorp 系统的关键组件

BeyondCorp的组件

通过使用下列组件，BeyondCorp 将各类已有系统组件、新系统组件进行集成，以便实现灵活而细粒度的信任判定。

设备（Device）和主机(Host)

要实现基于清单的访问控制，基本前提就是要建立清单库。基于环境和安全策略，团队需要先针对设备和主机的定义达成一致。**设备（device）**是物理或虚拟“计算机”，而**主机（host）**是指某特定时间点上设备状态的快照。例如，**设备**可能是一台笔记本电脑或一部手机，而**主机**则是运行在该设备上的操作系统和软件的详细信息。设备清单服务包含设备信息、运行于设备上的主机信息、以及对二者的信任评估。在下面的章节中，根据不同的访问策略配置，“设备”既可能指代物理设备也可能指代主机。建立基础清单库后，其余组件就可以按需部署，以提供安全性更高、覆盖率更广、颗粒度更细、延迟性更低、灵活性更佳的基于清单的访问控制服务。

基于信任等级的访问

信任度可以划分为若干信任等级，并由信任引擎为每个设备分配信任等级，另外，需要为每个资源都事先分配一个访问所需的最低信任等级，简称访问信任等级。设备被分配的信任等级必须大于等于资源的访问信任等级才可访问该资源。简单举一个婚庆餐饮公司的例子：送货员只需要查询婚礼的地址，这种访问请求所需的信任等级较低，事实上，他们也并不需要访问更敏感的服务，比如账单系统，这类系统一般会分配一个较高的访问信任等级。

分配访问信任等级有几个优点：降低了高安全要求的设备相关的运维

成本（主要是与技术支持和生产力相关的成本），同时也提高了设备的可用性。如果允许设备访问更多高敏感数据，则需要更频繁地检测以确保设备使用者确实“在场”，因此越是信任某个设备，其身份凭证有效期应该越短。因此，按照潜在访问需求所需的最低信任等级来要求/限定设备所需的信任等级，就意味着设备使用者在访问过程中受到干扰的程度会降到最小。比如，为了维持较高的信任等级，就需要设备在几个工作日内必须完成操作系统最新升级包的安装；而对于信任等级需求较低的设备，安装升级包的时间窗口就可以稍微宽松些。

再举一个例子，一台由公司集中管理的笔记本电脑，由于在一段时间内没有连接到网络，因此没有更新到最新状态。如果操作系统缺少几个*非关键补丁*，其信任等级可能被降为中等，仅允许访问部分业务应用，而被拒绝访问需要更高信任等级的业务应用。但如果它缺少*关键补丁*，或者防病毒软件报告该设备已感染病毒，那就只允许它连接补救服务。在最极端的情况下，一台明确的遗失或被盗设备会被拒绝访问所有企业资源。

除了分配信任等级，信任引擎还通过标注设备可访问的 VLAN 来进行网络分段。网络分段允许我们基于设备状态来限制对诸如实验室和测试环境的特定网络的访问权限。当一个设备变得不可信时，可以将它分配到隔离网络，在设备恢复信任之前仅提供有限的资源访问权限。

设备清单服务

设备清单服务（如图 2 所示）是一个不断更新的数据管道，能够从广泛的数据来源中导入数据。系统管理数据源可能包括活动目录（Active Directory）、Puppet 和 Simian，其他设备代理、配置管理系统和企业资产管理系统也会向该管道导入数据。外部（Out-of-band）数据源包括漏洞扫描

系统、证书颁发机构和诸如 ARP 映射表等网络基础设施单元。每个数据源都可以发送设备相关的完整数据或增量数据。

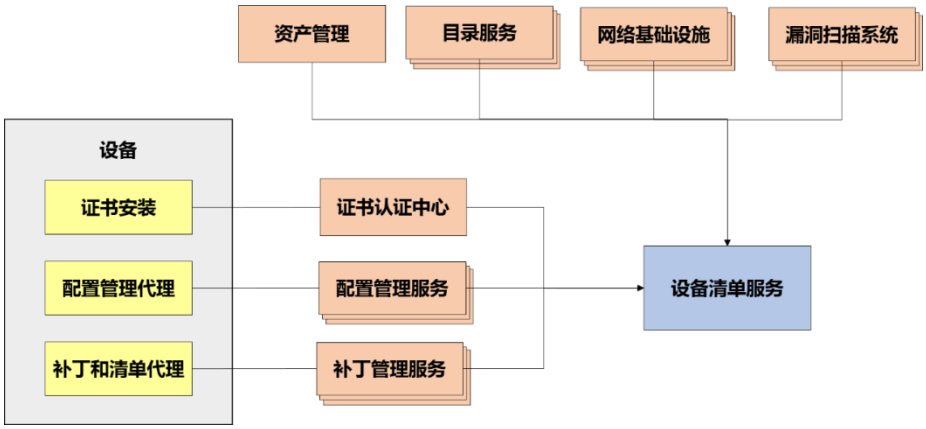


图 2　设备清单服务

BeyondCorp 设备清单服务自实施初期，已经从超过 15 个数据源中吸收了数十亿的增量数据，速度约 300 万条/天，总量超过 80TB。保留历史数据非常重要，因为这样才能更好地了解特定设备的端到端生命周期、跟踪和分析总体趋势、执行安全审计和调查取证。

数据类型

数据有两种主要的类型：观察数据和预设数据。

观察数据由程序产生，包括以下项目：

- 最近一次在设备上执行安全扫描的时间,及扫描结果
- 活动目录的最后同步策略和时间戳
- 操作系统版本和补丁等级
- 已安装的软件

预设数据通过 IT 运维手动维护，包括以下内容：

- 为设备分配的所有者

- 允许访问该设备的用户和组
- 分配的 DNS 和 DHCP
- 对特定 VLAN 的显式访问权限

在数据不足或客户端平台不可定制的情况下，就需要明确分配（比如打印机就属于这种情况）。与观察数据所表现的易变性相比，预设数据通常是静态的。通常需要分析许多来自不同来源的数据，用以识别潜在的数据冲突，而不要盲目地相信单个或少量系统的数据真实性。

数据处理

数据格式的转换与统一

为了使设备清单服务保持最新状态，涉及几个处理阶段。首先，所有数据必须转换成一种通用数据格式。一些数据源，比如内部自研系统或开源解决方案，可以通过系统改造，在提交数据时主动发布给清单服务。而其他来源，特别是那些第三方数据源，可能无法扩展或改造，难以支持主动的变更发布，这种情况需要通过定期轮询来获得更新。

数据关联

当输入数据格式统一后，就进入数据关联阶段。在这个阶段，所有来自不同数据源的数据都被聚合、关联到某一设备，当确定两条记录描述的是同一设备时，就将它们合并为单条记录。数据关联过程看似简单，但在工程实践中却相当复杂，因为许多数据源之间并不具备数据关联所必须的重叠的“标识符”。

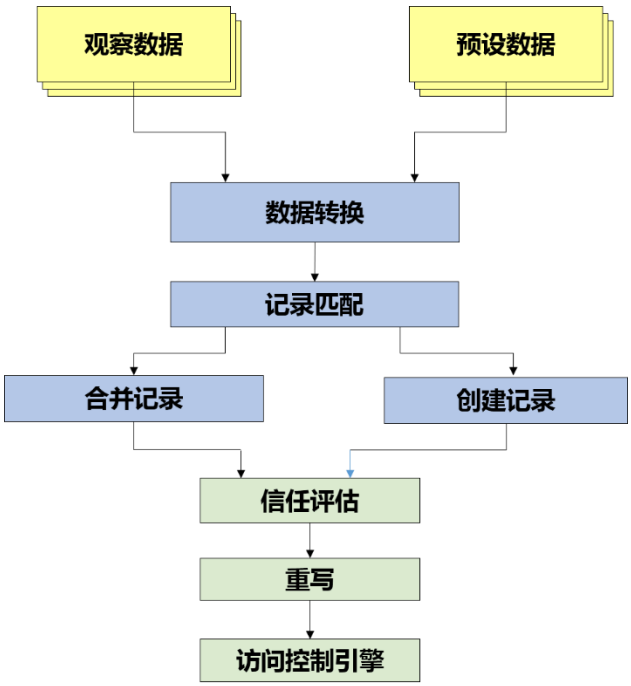


图 3 数据处理管道

例如，资产管理系统可能存储资产 ID 和设备序列号，而磁盘加密托管系统存储硬盘序列号，证书颁发机构存储证书指纹，ARP 数据库存储 MAC 地址。这些数据不具备一个重叠的“标识符”，难以确定来自这些独立系统中的增量是否描述的是同一个设备，只有在清单报告代理同时报告几个或全部这些标识信息之后，这些多源的、没有交集的记录才可能合并为单条记录。

如果再考虑到设备的全生命周期，相关的信息及其关联过程将更加一团糟，因为硬盘、网卡、机箱和主板都有可能被替换，甚至会在设备之间交换。另外，如果还考虑人为的数据录入错误，情况会更加复杂。

信任评估

一组输入记录一旦完成关联合并，就会触发引擎进行重新评估。为了

分配信任等级，评估分析过程引用多种字段并聚合产生最终结果。信任引擎目前从不同的数据源引用了数十个字段，包括针对特定平台的和平台无关的；随着系统的不断演化，还有数百万个额外字段可供分析。例如，为了获得较高信任等级，可能需要一个设备满足以下所有（或更多）需求：

- 加密
- 成功执行所有的管理和配置客户端程序（agents）
- 安装最新的操作系统安全补丁
- 从所有输入源中获得的数据状态一致

这种信任等级预计算减少了必须被推送到网关的数据量，以及在为访问请求做授权判定时所需的计算量。这一步也确保所有的执行网关都使用了一致的数据集合。在这个阶段，我们甚至可以对非活跃设备修改信任等级。比如在以前，我们会预先拒绝所有可能受到 Stagefright[2]漏洞影响的设备的访问权限，即便它们还没发起实质性的访问请求。预计算同样为我们提供了一个实验框架，在此框架中可以对变更进行预验证，以及在不影响整个公司的情况下，对策略或信任引擎进行小幅度的局部调整和验证。

当然，预计算也有它的局限性，还不能完全依赖它。比如，访问策略可能要求进行实时的双因子认证，或者限制来自某已知恶意网段的访问请求。策略或设备状态变更与网关真正执行这个变更之间的延迟并不是什么大问题，因为更新延迟通常在 1 秒以内。事实上更本质的问题是，并不是所有的信息在预计算阶段都能够获取。

特殊处理

信任引擎对于设备信任等级的分配有最终决定权。信任评估还需要考

虑设备清单服务中已存在的特殊处理。通过特殊处理，允许对通用访问策略进行覆盖和重写。特殊处理主要为了降低策略变更或新策略的生效延迟。比如，由于安全扫描设备可能尚未升级，检测不出某种零日攻击，但可以通过特殊处理立即阻止某台可能遭受零日攻击的设备；同样，可以采用特殊处理，允许将某台不可信设备连接到实验室网络。物联网设备安装和维护设备证书可能并不可行，同样可以通过特殊处理，直接为其分配适合的信任等级以确保正常访问。

部署

首次上线

BeyondCorp 上线的第一阶段包含一部分网关和初步的元清单服务，这些服务仅由少数几个数据源构成，主要是一些预设数据。最初实现的访问策略模拟了谷歌已有的基于 IP 的边界安全模型，并将这个策略集应用到不可信设备上，为来自特权网络的设备保留不变的访问权限。这种策略能够确保在系统完善之前，能安全地部署系统的一些组件，而不会影响用户的平滑使用。

与此同时，BeyondCorp 团队也在设计、开发并持续迭代一个规模更大、延迟更低的元清单解决方案。这个设备清单服务从超过 15 个数据源收集数据，根据正在主动生成数据的设备数量，每秒钟可能有 30 至 100 个不等的数据变更。设备清单服务主要提供的是企业设备的信任资格标注和强制授权。随着元清单解决方案的成熟，可以获得更多的设备信息，能够逐步地依靠信任等级分配，逐步替代基于 IP 的策略。在验证了低信任等级

设备的工作流后，对更高信任等级的访问进行细粒度限制，并逐步迈向最终目标：随着时间的推移，有序扩大设备和企业资源的信任等级分配范围，并基于信任等级进行访问控制。

考虑到前文提到的从不同来源关联数据的复杂性，BeyondCorp 采用 x.509 证书作为固定的设备标识符。x.509 证书提供了两个核心功能：

- 如果证书发生变化，即使所有其他标识符都保持相同，设备也被标记为不同设备。
- 如果证书安装在不同的设备上，关联逻辑会发现证书冲突以及与辅助标识不匹配，随即做出反馈，降低设备信任等级。

证书并未降低数据关联的必要性，其本身也不足以获得访问权限。但它确实能提供一个基于密码学的 GUID，访问网关还可将其用于流量加密，并持续、唯一地标识设备。

移动设备

谷歌一直力图使移动设备成为主流平台，移动设备必须能够完成与其他平台相同的任务，因此也需要相同的访问等级。与其他平台相比，在移动平台上部署信任等级访问模型更容易。移动设备的特点是没有太多传统遗留通信协议和访问方法，因为几乎所有通信都是基于 HTTP 的。安卓设备使用加密的安全通信，允许在设备清单中识别设备。值得一提的是，由于 API 也位于与访问控制引擎集成的访问代理之后，因此原生应用程序与通过 Web 浏览器访问的资源都能通过相同的授权机制进行保护。

遗留（Legacy）平台和第三方平台

为了支持遗留平台和第三方平台，我们需要采用比移动设备更广泛的访问方法。为此任意 TCP 和 UDP 流量，我们通过 SSH 隧道和客户端

SSL/TLS 代理技术提供的隧道通信。 而网关只允许符合访问控制引擎中策略的隧道业务通过。RADIUS[3]是一个特例：它与设备清单服务集成，但它从信任引擎接收的是 VLAN 的分配结果，而不是信任等级的分配。在网络连接时，RADIUS 使用 802.1x 认证的证书来作为设备标识符，通过信任引擎分配的结果，动态设置 VLAN。

避免干扰用户

在部署 BeyondCorp 的过程中，面临的最大挑战之一是如何在不干扰用户的情况下完成如此大规模的任务。为了制定策略，需要先确认现有的工作流。从现有的工作流中，可以确定：

- 哪些工作流，可以与无特权网络兼容
- 哪些工作流允许进行超出预设的访问或哪些工作流允许用户绕过已经存在的限制

为了确认工作流，我们采用双管齐下的模式。一方面，开发了一个模拟管道，它可以检查 IP 级元数据，将流量划分到服务，并在模拟环境中应用了我们预期的网络安全策略；另一方面，将安全策略转换为每个平台本地防火墙配置语言。在企业网络上，这种手段可以很好的记录流量元数据，这些流量是访问谷歌企业服务所必须的，稍有差池，迁移到无特权网络后，这些服务很可能无法访问。在此过程中，我们还有一些令人惊讶的意外发现，比如那些早就应该下线的服务，却不明就里的仍在运行。

收集了这些数据之后，通过与服务所有者合作，将他们的服务迁移到支持 BeyondCorp 的网关。有些服务很容易迁移，但还有些服务则比较困难，需要一些特殊处理机制。不过，这种情况都明确指定了责任人，确保服务所有者能在限定期限内消除例外。随着越来越多的服务进行了更新和

改造，越来越多的用户在不执行任何例外处理的情况下也可以正常工作很长一段时间，此时，就可以将用户的设备分配到一个无特权的 VLAN。通过这种方法进行过渡，用户使用不兼容 BeyondCorp 的应用不会感到不太方便；迁移压力基本都在服务提供者和应用程序开发人员身上，这可以促使他们正确地配置相关服务。

特殊处理增加了 BeyondCorp 生态系统的复杂性，随着时间的推移，“为什么我的访问被拒绝了？”这个问题的答案已经不那么明了。基于清单数据和实时请求数据，需要非常明确地判断特定请求在特定时间点失败或成功的原因。回答上述问题的第一步是与终端用户建立沟通（警告其潜在的问题，以及如何进行自我修复或联系支持），并培训 IT 运维人员。此外，还开发了一种服务，它可以分析信任引擎的决策树和影响设备信任等级分配的事件的时间顺序，从而提出补救措施。有些问题用户可以自己解决，不需要权限更高的支持人员。拥有额外访问路径的用户通常能够自我修复，例如，如果用户认为他的笔记本电脑信任评估不当,但手里还有一只信任等级足够的手机，我们可以将诊断请求转发给这个手机进行评估。

挑战和经验教训

数据质量及相关性

资产管理的数据质量问题可能导致设备无意中失去对企业资源的访问权限。拼写错误、标识错误和信息丢失都是常见问题。此类问题可能由于采购团队收到资产并将其添加至系统时的人为失误，也可能是由于制造商工作流程的失误导致。数据质量问题也经常发生在设备维修过程中，主要原因在于替换设备的零部件或在设备之间交换某个部件。这些问题可能会破坏设备记录，除非人工检查这些设备，否则很难修复这些记录上的差

错。例如，单条设备记录可能实际上包括两个不同设备的数据，要自动修复和分离数据甚至需要调整设备硬件的资产标签甚至主板序列号。

这时最有效的解决方案是通过本地工作流程改进并增加自动输入验证，以便在输入时发现并减少人为错误。复式记账法有一定帮助但是并不能发现所有错误。做出准确的信任评估需要设备清单库提供高精度的数据，所以这又迫使人们不得不重新关注设备清单库中的数据质量。这种数据的精确性要求是前所未见的，也带来了前所未有的价值。比如，我们能精确地知道终端信息，安装最新补丁的情况，进而提高整个系统安装最新补丁的百分比。

稀疏数据集

如前所述，上游数据源未必有重叠的设备标识符。以下列举一些潜在的场景：新设备可能有资产标签，但没有主机名；在设备生命周期的不同阶段，硬盘序列号可能与不同的主板序列号相关联，又或者 MAC 地址可能会发生冲突。一组简单的启发式算法可以将大部分增量与数据源某个子集相关联，但为了将精度提高到接近 100%，需要一组非常复杂的启发式算法来处理看似无穷无尽的边缘情况。一小部分数据不匹配的设备，可能会使数百甚至数千名员工无法使用他们工作中的必需应用。为了减少这种情况的发生，监控并验证各种综合数据可能的情况，精细设计和验证信任评估路径，最终确保符合预期的信任等级评估结果。

管道延迟

由于设备清单服务从几个不同的数据源中获取数据，所以每个源可能都需要一个特定的实施方案。自研系统或基于开源系统的数据源很容易扩展，以便异步地向我们现有管道发布增量。对于其他来数据源必须定期轮

询，这需要在轮询频率和由此产生的服务器负载之间取得平衡。尽管将变更信息传递到网关通常不到一秒，但是对于轮询的场景，一些变更可能需要几分钟才能获悉。此外，串行处理本身也会增加时延。因此，需要采用流式处理。

沟通

对安全基础设施的根本性改变可能会对整个公司的生产力产生负面影响。与用户沟通改变的潜在影响、会出现的问题和可能的补救措施十分重要，但是很难找到过度沟通和沟通不足之间的平衡点。沟通不足会让用户感到惊讶和困惑，造成补救措施效果差，IT 支持人员的工作也会超负荷。过度沟通也有问题：不愿改变的用户会倾向于高估变化带来的影响并企图寻求不必要的豁免。过于频繁的沟通也会让用户对潜在的影响判断出现偏差，由于谷歌的企业基础设施在许多互不关联的方面同时开展工作，用户很容易将访问相关的问题与其他正在进行的项目问题混淆，这也会降低补救措施的效率，增加支持人员的操作负荷。

灾难恢复

正因 BeyondCorp 基础设施的组成是非常复杂的，而灾难性的失败甚至会导致支持人员无法访问恢复所需的工具和系统，因此 BeyondCorp 系统中构建了各种故障保护系统。除了监测信任等级分配的潜在或明显的变化，我们已经利用了现有的一些灾难恢复实践，以确保在发生灾难性紧急情况时，BeyondCorp 仍能发挥作用。BeyondCorp 的灾难恢复协议基于最小依赖关系，并允许极少的一部分特权维护人员重放清单变更的日志记录，以便恢复到设备清单和信任评估工作以前的良好状态。我们也有能力在紧急情况下细粒度地变更访问策略以确保维护人员启动恢复流程。

下一步

与任何大项目一样，我们在部署 BeyondCorp 时面临的挑战，有些是预期内的，而有些在意料之外。在谷歌，越来越多的团队正在寻找新的、有趣的方式来整合系统，并提供更详细、更有层次的防护以对抗恶意攻击者。在没有牺牲易用性的前提下，BeyondCorp 已经大幅改善了谷歌的安全态势，同时还提供了一个灵活的基础设施，能够基于策略进行授权决策而不受具体技术限制。BeyondCorp 在谷歌自身的系统和规模内已取得了相当大的成功，也欢迎其他组织基于这些原则和流程进行部署和完善。

参考文献：

[1] Architectural discussion of BeyondCorp: http://research .google.com /pubs/pub43231.html.
[2] Stagefright: https://en.wikipedia.org/wiki/Stagefright _(bug).
[3] RADIUS: https://en.wikipedia.org/wiki/RADIUS.

【第三篇】BeyondCorp：访问代理

本文将详细介绍 BeyondCorp 前端基础设施——访问代理（Access Proxy, AP）的实现，关注其实施过程中遇到的挑战,以及设计和上线中学到的经验教训。此外，对于我们正在开展的，旨在提高员工访问内部应用时使用体验的项目，本文也有所涉及。

向 BeyondCorp 模型迁移过程中（之前在“BeyondCorp：一种新的企业安全方案”[1]和“谷歌 BeyondCorp：从设计到部署”[2]中有讨论），有许多难题需要解决，比如，如何将公司策略应用到所有内部服务就是一个重大挑战。传统方法通常要把每个业务后端与设备信任引擎集成，以便进行应用级策略的评估，然而，这种方法会明显降低产品发布和迭代的速度。

为了解决这个问题，谷歌通过前端访问代理（AP）作为中心化的策略强制执行点，实现粗粒度的公司安全策略。访问代理的设计具有足够的通用性，基于同一套代码我们实现了不同逻辑的网关。目前， 访问代理已支持 Web 代理和 SSH 网关组件[2]。由于 AP 是员工访问内部 HTTP 服务的唯一机制，所有内部服务都需要迁移到 AP 的后面。

事实证明，我们一开始只打算支持 HTTP 协议是完全不够的，随着项目的推进，不得不为更多的协议（其中多数都需要端到端加密,如 SSH）提供解决方案。支持这些协议通常需要对客户端进行改造，以确保 AP 准确识别设备。

结合访问代理（AP）和集中的访问控制引擎（Access Control Engine, ACE）（共享的 ACL 评估系统）主要有两个好处：一是所有请求都途经同

一个日志记录点，便于更有效地进行流量分析；二是能够更迅速、更统一地改变执行策略。

BeyondCorp的前端基础设施

任何大规模部署的现代 Web 应用程序都会采用前端基础设施——通常是负载均衡和/或 HTTP 反向代理的组合，企业 Web 应用也不例外，前端基础设施为策略执行点的部署提供了理想位置。因此，谷歌的前端基础设施对于 BeyondCorp 访问策略的强制执行至关重要。

谷歌前端基础设施的主要组件是 HTTP/HTTPS 反向代理集群，即谷歌前端服务（Google Front Ends, GFEs）[3]。GFE 有很多优点，例如负载均衡和 TLS 卸载服务。这样 Web 应用的后端可以专注于服务请求的具体内容，而几乎不必考虑请求的路由细节。BeyondCorp 将 GFE 作为访问策略强制执行的逻辑中心。通过逻辑上的集中，带来请求的汇集，在此基础上就可以自然而然地扩展 GFE 的功能，比如自助服务开通、认证、授权和集中式日志记录。扩展后的 GFE 即访问代理（AP）。下文将详细阐述访问代理提供的具体服务。

扩展后的GFE特性：产品需求

GFE 有一些内置功能，并不是专门为 BeyondCorp 设计的但可以为 BeyondCorp 所用：如，为后端提供负载均衡服务、通过 GFE 实现 TLS 卸载。AP 通过引入**认证**和**授权**策略扩展了 GFE。

认证

为了正确处理一个授权请求，AP 需要识别发出请求的用户和设备。

在多平台环境中，设备认证面临许多挑战，将在后文的“多平台身份认证的挑战”中进行详细讨论，本节重点介绍用户认证。

AP 通过集成谷歌的身份提供服务（Identity Provider, IdP）完成用户身份认证。如果要求后端服务必须修改它们自身的身份认证机制才能迁移到 AP 不具备伸缩性，所以 AP 需要支持一系列的认证机制,包括：OpenID Connect、OAuth 和一些定制化协议。

AP 还需要处理不能提供用户凭证的请求场景，例如，一个软件管理系统试图下载最新的安全补丁，这种情况下，AP 可以禁用用户认证。

当 AP 认证用户后，将用户凭证相关信息从请求中去除后再转发至后端服务，这样做至关重要，有两点原因：

- 确保后端不能通过访问代理重放请求（或凭证），进行重放攻击。
- 代理对后端服务透明。这样做的好处在于后端业务可以独立于访问代理的数据流叠加自身的认证逻辑，并且也避免了将 cookie 和用户凭证不必要的暴露给后端业务。

授权

以下两个设计推动 BeyondCorp 中授权机制的实施：

- 一个可通过远程过程调用（Remote Procedure Calls, RPC）查询的集中访问控制列表（Access Control List, ACL）
- 采用领域特定语言（domain-specific language, DSL）表达访问控制列表（ACL），使其同时兼顾可读性和可扩展性

以服务形式提供 ACL 评估能够保证多种前端网关的一致性（如 RADIUS 网络访问控制基础设施、AP 和 SSH 代理）。

集中式授权有好有坏。好处是，通过集中策略执行点，由前端访问代理负责授权可以将后端开发者从处理授权的细枝末节中解放出来，并且一致性更强。坏处是，代理可能无法执行细粒度策略，细粒度授权还是要交由后端处理更好（例如，“用户 A 被授权去修改资源 B”）。

从过去的实践经验来说，将 AP 提供的粗粒度、集中式授权与后端实现的细粒度授权结合对于前后端来说都是最佳选择。这种方法不会导致重复工作，因为针对特定应用的细粒度策略通常与前端基础设施所执行的企业级策略相互独立。

代理和后端之间的双向身份认证

因为后端业务将访问控制逻辑完全交由前端的 AP 进行，迫切需要适当的机制确保后端业务能信任 AP 转发的业务流量已经通过了认证和授权。这种机制尤其重要，因为 TLS 握手和传输在前端代理就终结了，前端代理是通过另外的加密通道传输 HTTP 请求给后端业务。

为满足上述要求，需要一个能够建立加密通道的双向认证机制----举个例子：一种可能的实现是基于 TLS 双向证书认证和企业公钥基础设施。BeyondCorp 采用了内部开发的认证和加密框架 LOAS(Low Overhead Authentication System, 低开销认证系统)，它可以对代理和后端之间的所有通信进行双向认证和加密。

前端和后端之间进行双向认证和加密的一个好处是，后端可以信任 AP 插入的任何元数据（通常以 HTTP 消息头的形式）。在反向代理和后端之间额外插入元数据、使用自定义协议（比如， Apache JServe 协议）并不是什么新方法,但 AP 的双向认证机制，确保了元数据的完整性。

采用此方法的另一个好处是，当 AP 逐渐部署了更多新功能时，后端可以通过简单地解析相应的消息头，获取 AP 插入的新功能数据，并选择所需信息。使用这个功能可以将设备的安全等级传递到后端，后端可据此调整服务内容。

ACL 语言

将领域特定语言（domain-specific language, DSL）用于表述 ACL 是解决集中式授权挑战的关键。这种语言支持静态编制 ACL（有助于提高性能和可测试性），同时减少了策略表述和具体实现之间的逻辑鸿沟。这一策略提高了了以下各方的职责分离：

- **安全策略团队：**负责对访问策略进行抽象和静态编制
- **清单管道团队：**根据发起访问请求的用户和特定设备，负责提供对资源的访问决策的具体实例化（请参阅“谷歌 BeyondCorp：从设计到部署”[2]了解关于清单管道的更多细节）
- **访问控制引擎团队：**负责评价和执行安全策略
- ACL 语言语义上采用首次匹配（first-match）模型，和传统防火墙规则比较类似。虽然这种模型存在一些极端情况（例如，规则之间会相互覆盖），但好在这些情况已经众所周知，安全团队理解起来还是相对容易。当前采用的 ACL 结构包括两大部分：

- **全局规则：**通常是粗粒度的，影响所有服务和资源。例如，“安全等级低的设备不允许提交源代码”。
- **针对特定服务的规则：**专属于某个服务或主机，通常包括和用户有关的断言。例如，“群组 G 中的所有厂商允许访问 Web 应用 A”。

以上结构基于一个假设，即服务所有者可以识别应用策略的 URL 地址范围，除非请求对象不在 URL 中指定而在报文主体中指定（可以通过修改 AP 来处理这种情况）。不可避免地，针对特定服务的规则规模会越来越大，因为访问代理会对越来越多的服务负责，而这些服务都需要特定的 ACL 规则。

全局规则在处理一些特殊的安全状况（例如，员工离职）和应急响应（例如，浏览器漏洞利用或设备被盗）时具有极大的便利性。比如，这种机制曾帮助我们成功处置 Chrome 浏览器某个第三方插件的 0Day 漏洞风险,通过创建一条全新的高优先级规则，使用老版本的 Chrome 浏览器时将会被重定向到一个带有更新指南的页面，该规则 30 分钟内就在整个公司完成部署和强制执行，最终，存在漏洞的浏览器的数量急剧减少。

集中式日志记录

为了进行必要的事件响应和取证分析，所有请求日志必须进行持久化存储。AP 提供了一个理想的日志记录点。日志记录主要包括部分请求头、HTTP 响应码、调试或重构访问决策和 ACL 评估过程所需的元数据，一般包括访问请求的设备标识和用户标识。

访问代理的特性：运维弹性

自助服务开通

一旦访问代理准备就绪，企业应用的开发人员和所有者就可以着手配置通过代理的服务访问模式。

当谷歌逐渐从网络层开始限制用户对公司资源的访问，访问代理就成为了能在迁移过程中保持服务正常运行的最快方案。显然，单个团队无法支撑对 AP 配置的全部更改，因此将 AP 配置过程结构化，使用户可以更为便利地使用自助服务。用户保留了他们自己的配置片段的所有权，而 AP 团队保留构建配置系统的所有权，可以校对、测试、灰度发布（金丝雀发布）和更新配置。

这种设置有几个主要好处：

- 解放了 AP 团队，让他们不再需要根据每个用户请求持续修改配置
- 鼓励服务所有者拥有他们的配置片段（并为其编写测试）
- 确保开发速度和系统稳定性之间的平衡

仅仅只需几分钟就可以在 AP 后设置一个服务，用户也可以在不请求 AP 团队支持的情况下迭代自己的配置片段。

多平台身份认证的挑战

目前，已经理清了 BeyondCorp 前端在服务侧的情况，包括了其实现

及由此带来的困难和挑战。现在将采用类似方法来梳理 BeyondCorp 中客户端方面的情况。

准确的设备识别至少需要以下两个组件：

- 某种形式的设备标识
- 能追踪任何指定设备最新状态的清单数据库

BeyondCorp 的目标之一是以适当的设备信任替代基于网络的信任。每个设备都必须有一个一致的、不可克隆的标识，设备的软件、用户和位置的相关信息必须集成到清单数据库中。正如在前两篇 BeyondCorp 论文中所说，构建和维护设备清单库可能面临着诸多挑战。下面几个小节将更详细地描述与设备认证相关的挑战和解决方案。

台式机和笔记本电脑

台式机和笔记本电脑使用 x.509 证书，以及系统证书库中对应的私钥。密钥存储是现代操作系统的标准功能，它确保了通过 AP 与服务器通信的命令行工具（和守护进程）可以与正确的设备标识匹配。由于 TLS 要求客户端提供拥有私钥的加密证明，而且设备标识存储在类似可信平台模块（Trusted Platform Module, TPM）的安全硬件中，这能确保标识的不可欺骗性且不可克隆性。

但这种实现方式有一个主要缺点：证书验证提示通常会影响用户体验。幸好大多数浏览器都支持通过策略配置或插件扩展自动提交证书。但如果客户端提供了无效证书，服务器拒绝 TLS 握手,此时也会对用户有所影响。TLS 握手失败，浏览器会显示特定的错误消息，且大多不可定制。为了提

升用户体验，AP 可以接受没有有效客户端证书的 TLS 会话，但必要时会按需弹出一个 HTML 拒绝页面。

移动设备

上述解决证书提示问题的策略，在几个主流移动平台中都无需考虑。移动设备的认证可以不依赖证书，因为移动操作系统本身就可以提供安全性高的设备标识。比如，ios 设备可以使用苹果的 Vendor 标识符（Identifier For Vendor, IDFV），安卓设备使用企业移动管理（EMM）应用提供的设备 ID。

一些特殊情况和例外

虽然在过去的几年中已经将绝大多数 Web 应用程序迁移到访问代理，但是仍然有些特殊的用例，要么自身无法与访问代理模式兼容，要么需要经过特殊处理才能兼容。

非HTTP协议

有些谷歌企业应用程序使用了非 HTTP 协议，这些协议需要端到端加密。为了通过 AP 为这些协议提供服务，需要将它们封装在 HTTP 请求中。

幸好有现成的 ProxyCommand 工具，因此在 TLS 上将 SSH 业务封装成 HTTP 流量并不难。我们开发了一个类似 Corkscrew 的本地代理，不同之处在于我们使用了 WebSockets 进行封装。虽然 WebSockets 和 HTTP CONNECT 请求都能兼容 AP 的 ACL 评估，但 WebSockets 本身能从浏览器继承用户和设备的身份凭据，这一点比 CONNECT 机制更占优势。

对于 gRPC 和 TLS 流量，最终选择使用 HTTP CONNECT 请求进行封装。封装有个很明显的缺点，它会给传输带来性能损失（虽然可以忽略不计）。但封装有一个重要优势，它能够将设备标识和用户标识分离，在协议栈的不同层来实现。这种方案缘于基于清单库的访问控制是一个相对新的概念，虽然通常现有协议支持用户认证（例如，LOAS 和 SSH 都支持），但要扩展到支持设备认证并不容易。

在封装 CONNECT 请求的 TLS 层执行设备认证，就不需要重写应用来识别设备证书。以 SSH 为例：客户端和服务器之间能够使用 SSH 证书来进行用户认证，但是 SSH 原本并不支持设备认证。此外，不能通过修改 SSH 证书来传递设备身份，因为 SSH 客户端证书默认是可移植的：一个 SSH 证书可以用在多个设备上。类似于 HTTP 的处理方式，CONNECT 封装确保了用户认证和设备认证的良好分离。使用 TLS 客户端证书来认证设备的时候，也可以使用用户名和密码的方式来认证用户。

远程桌面

在 Chrome 代码库中公开可用的 Chrome 远程桌面[5]，是谷歌 BeyondCorp 主要使用的远程桌面解决方案。虽然 HTTP 的封装协议可以满足很多使用场景，但还有些专门用于远程桌面的协议，它们对通过 AP 后可能产生的额外延迟格外敏感，需要单独考虑。

为了确保请求得以授权，Chrome 远程桌面在连接建立的交互流程中引入了基于 HTTP 的授权服务器。这个服务器位于 Chromoting 客户端和 Chromoting 主机之间充当第三方授权服务器，同时也帮助两个实体共享密钥，与 Kerberos 协议工作方式类似。

我们将授权服务器作为 AP 的一个简化的后端服务来实现，并为其配置特殊的 ACL。这种实现效果还不错：通过 AP 带来的额外延迟仅在每个远程桌面会话发起时发生一次，并且也确保了访问代理能对每个会话创建请求都实施 ACL。

第三方软件

第三方软件通常比较麻烦，因为它可能无法提供 TLS 证书，也可能其实现逻辑假设网络总是直连的。为了适配这些软件，我们设计了一种可以自动建立点到点加密隧道（使用 TUN 设备）的方案。软件对隧道无感知，就像是直连到服务器一样。理论上来看，隧道建立机制与远程桌面方案类似：

- 客户端运行辅助程序来建立隧道
- 服务端同样运行辅助程序作为 AP 的后端
- AP 执行访问控制策略并且协助会话信息和加密密钥在客户端和服务端的辅助程序之间交换

经验教训

ACL很复杂

推荐下面的最佳实践来减少 ACL 相关的困难：

- **确保语言的通用性。**AP 的 ACL 改变了无数次，而且还持续不断

地增加新信息（如用户和组）。因此需要定期更新可用功能，并且确保语言自身不会妨碍这些更新。

- **尽早启动 ACL。**原因有两个方面：
  - 确保用户尽快了解 ACL 以及访问被拒绝的可能原因。
  - 确保开发者尽快开始调整代码来满足 AP 的要求。例如，为了处理用户和设备认证，我们甚至重新开发了软件来替换 cURL。
- **完善自助服务。**正如前面提到的，单个服务配置团队无法支撑多个团队。
- **建立能将数据从 AP 传递给后端的机制。**正如前面提到的，AP 能够安全地将额外数据传递给后端，允许其能够进行细粒度的访问控制。尽早规划所需要的功能。

紧急情况

事先充分测试，充分准备，以应对意外紧急情况。尤其注意以下两类紧急事件：

- **产品类紧急事件：**由于服务访问的逻辑链路上关键部件的中断或失灵造成的紧急事件。
- **安全类紧急事件：**由于迫切需要授权/撤回特定用户和/或资源的访问造成的紧急事件。

产品类紧急事件

为了确保 AP 在大多数宕机期间还能存活,请根据 SRE 最佳实践进行

设计和运维[3]。为了避免可能出现的数据源中断，需要定期对所有数据进行快照以便能本地访问。此外，还需要设计不依赖于 AP 本身的 AP 修复路径。

安全类紧急事件

安全紧急事件比产品紧急事件更为敏感，因为在设计时往往容易被忽略。在用户撤销/设备撤销/会话撤销时均需考虑到 ACL 推送频率和 TLS 问题。

用户撤销相对简单：作为撤销过程的一部分，已撤销的用户将自动添加到特殊组，通过一条靠前的 ACL 全局规则（请参阅上面的“ACL 语言”）确保这些用户访问任何资源的权限都被禁止。会话令牌（例如，OAuth 和 OpenID Connect 令牌）和证书有时候会泄露或丢失，同理也需要撤销。

正如第一篇 BeyondCorp 论文中所说[1]，除非收到设备清单管道的状态上报，否则设备标识不可信。这意味着即使丢失 CA 密钥（意味着不能撤销证书）也不会失控，因为直到被列入清单管道的目录中，新的证书才可信。

由于上述特性，我们决定彻底忽略证书撤销过程: 如果怀疑证书相应的私钥丢失或者泄露，不再发布证书撤销列表（certificate revocation list, CRL），而是降低证书的清单信任等级。清单本质上就是可信设备标识的白名单，并且不依赖于 CRL。这种方法的主要缺点是它可能会带来额外延迟。不过通过在清单和访问代理服务器之间设计快速传播通道，可以相对容易地解决这种延迟。

为了保证执行策略的及时可达，需要一个 ACL 的标准快速推送机制。ACL 超出一定规模后,必须要将部分 ACL 定义过程委托给服务所有者,这就会导致一些不可避免的错误。虽然单元测试和冒烟测试通常可以发现明显错误，但逻辑错误会通过安全措施渗透，并进入生产阶段。工程师必须具备快速回滚 ACL 变更的能力，才能恢复丢失的访问权限、锁定意外的访问权限。引用之前 Chrome 插件的 0Day 漏洞为例，快速推送 ACL 是应急响应团队的关键能力，通过快速推送自定义 ACL 可以强制用户进行更新。

工程师需要支持

迁移到 BeyondCorp 不可能一蹴而就，需要多个团队之间的协调和沟通。在大型企业中，将整个迁移任务委托给单个团队是不可能的。迁移很可能涉及一些不能向后兼容的变更，这需要得到管理层的强大支持。

迁移的成功很大程度上取决于团队在访问代理背后配置服务的难易程度。以减轻开发人员的开发负担为目标，要把异常情况的出现维持在最低限度。提供合理的默认设置，为常见用例撰写指南和文档。使用沙箱应对更高级和更复杂的变化，比如可以创建一个访问代理的单独实例，负载均衡器会忽略这个实例，但开发人员还可以访问（如临时覆盖其 DNS 配置）。沙箱在大部分情况都非常有用，比如在对 x.509 证书或底层 TLS 库进行重大变更之后，需要确保客户端 TLS 连接能成功进行。

展望未来

虽然 BeyondCorp 的前端实现在很大程度上是相当成功的，但仍然有一些问题尚未解决。首当其冲的，就是台式机和笔记本使用证书进行身份

认证，而移动设备则使用设备标识。证书的轮换仍然很痛苦，因为出示一个新的证书需要重启浏览器才能确保现有的套接字已经关闭。

为了解决上述问题，计划将台式机和笔记本电脑同样采用移动设备的方式，以消除对证书的需求。构建一个桌面设备管理器来处理这种迁移，该桌面管理器看起来与移动设备管理器非常相似。它将提供一个通用的标识，以设备-用户-会话-ID（DUSI）的形式出现，DUSI 会在所有浏览器和工具间共享，也许会使用一个通用 OAuth 令牌授予守护进程来实现。一旦迁移完成，不再需要通过证书验证台式机和笔记本电脑，并且在各类操作系统中的所有的控制都可以持续使用 DUSI。

结论

作为 BeyondCorp 的核心组件，访问代理的部署实施考虑了谷歌特有基础设施架构和用例。访问代理的最终设计实践与常见**网站可靠性工程（Site Reliability Engineering, SRE）**最佳实践一致，并且已证明其具有较高的稳定性及伸缩性——在部署过程中，访问代理已经完成了几个量级的增长。

任何想要实现类似 BeyondCorp 安全模型的组织都可以采用类似访问代理设计和部署的解决方案。希望本文分享的谷歌如何解决多平台认证、特殊案例和例外等挑战的解决方案，以及在这个项目中所学到的经验，可以帮助其他组织能以最小的代价实现类似项目。

参考文献：

[1] R. Ward and B. Beyer, “BeyondCorp: A New Approach to Enterprise Security,” login:, vol. 39, no. 6 (December 2014): https://www.usenix.org/system/files/login/articles/login_dec14_02_ward.pdf.

[2] B. Osborn, J. McWilliams, B. Beyer, and M. Saltonstall, “Beyond Corp: Design to Deployment at Google,” login:, vol. 41,no. 1 (Spring 2016): https://www.usenix.org/system/files/login/articles/login_spring16_06_osborn.pdf.

[3] B. Beyer, C. Jones, J. Petoff, and N. Murphy, eds., Site Reliability Engineering (O’Reilly Media, 2016).

[4] Apache JServer Protocol: https://tomcat.apache.org/connectors-doc/ajp/ajpv13ext.html.

[5] https://src.chromium.org/viewvc/chrome/trunk/src/remoting/.

【第四篇】迁移到BeyondCorp： 提高安全性的同时保持生产力

如果你熟悉过去两年发表在《login》[1-3]上的谷歌 BeyondCorp 网络安全模型文章，可能会想：“这一切听起来不错，但我的组织如何从现有模式转变到 BeyondCorp 类似模式？需要做些什么？这种转变对公司和员工会有什么影响？”本文讨论了从现有网络迁移到 BeyondCorp 模型所采用的方法，探讨如何实现在根本上改变网络访问模式的同时，却不影响公司生产力。

向 BeyondCorp 及其类似模型迁移面临诸多挑战，其中有几个尤其值得注意:

- 这个过程会影响整个公司。要让每个人都参与进来并保持一致和确保知情，就需要得到各级管理层的承诺和支持，这就需要与各方广泛沟通，从拥有个性化服务的团队，到管理层，再到支持团队，最后到用户。

- 迁移不可能一蹴而就。这个过程是多层次和渐进式的，包含信息收集、实验部署、流程和技术修正等各阶段，以及必要地点与时间的例外和补救。

- 这个过程涉及到在业务/技术栈中的多层、甚至所有层上进行更改：网络、安全网关、客户端平台和后端服务。为了保证不同层面工作进展相互独立，需要各层分治，确保多线程并行且易于管理和实现。

下面将讨论我们如何将 BeyondCorp 迁移工作进行分解，介绍各个层面平滑、一致的迁移所需的技术和工具，当然，必须确保整个过程对用户导致的负面影响最小化。

先决条件：认同和沟通

着手迁移到一个类似 BeyondCorp 的模型之前，需要来自公司高层及其他干系人的支持。第一步是理解和沟通迁移的动机：减少成功网络攻击所造成的威胁，同时保持生产力。需要将迁移背后的基本原理、威胁模型以及维持“业务照常运行”所需的成本形成文档。然后，准备好向每一个业务部门解释迁移过程的价值和必要性。与所有的安全项目一样，部署新模型需要付出代价：新工具、额外流程和使用习惯的改变。高层管理者需要积极支持这种改变，并将这种改变的动机和认同理念在所有干系人中推广。

有了管理者的准许和认同，接下来确定并争取到关键领域负责人的支持：安全、身份、网络、访问控制、客户端和服务器平台软件、关键业务应用程序服务，以及任何第三方合作伙伴或 IT 外包。负责人应该梳理和确定各领域专家，获得其承诺，并确保他们投入时间和精力。谷歌 BeyondCorp 团队是一个分布到全球各地的虚拟团队，有负责决策的总监，有项目技术经理负责协调落地执行。随着时间的推移，团队参与成员虽然会有所变化，但是高层领导、团队负责人和其他参与者会通过在线文档、邮件组和定期会议（面对面的和远程的）联系，始终保持对当前进展和项目状态的了解。

随着迁移工作的推进，通用的变更管理规则同样适用，因为每个工作组都有自己的关注点和优先级。要倾听反馈，调整兼顾每个参与者或受影

响群体的特殊情况和要求。及时公开计划和资讯很有必要，但仅仅这样还不够，还需要互动沟通（最好是当面沟通，至少也要通过视频或音频会议进行）才能加深团队间的协同、更易获取帮助和得到认可。

分步推进

BeyondCorp 的总体目标是，从允许客户端直接访问服务器的网络，过渡到无特权网络：取消客户端直接访问后端服务器的特权。详情请参见 BeyondCorp 系列文章的第一篇《BeyondCorp，一种新的企业安全方案》[1]。为此，谷歌曾考虑依次阻断每个应用或服务器，以便逐步移除遗留 VLAN 的访问特权。但这一策略并不理想，有两个原因：一是在网络层部署和协调很困难；二是在应用层增加了影响生产力的风险。因此，决定在最终的 BeyondCorp 配置中部署一个新的 VLAN。这个 VLAN 只允许通过访问控制网关访问服务器网络，确保所有流量都经过身份认证、授权和加密。这一策略不是逐步限制遗留 VLAN 的特权，而是逐步将设备最终都转移到这个新的 VLAN 上。

VLAN 迁移项目实现了一个复杂但至关重要的目标，迁移遗留“特权”网络的用户设备，并将它们分配给新的受控无特权客户端（Managed Non-Privileged Client, MNP）VLAN。这次迁移有一关键约束：对于运行在新 VLAN 工作站上的任何遗留应用，无论是预期还是必需，直接访问服务器网络都将失败。因此，近期目标是在不破坏业务关键操作的情况下实现迁移工作。为此，采用了三管齐下的策略来实现这一目标：

1、广泛分析网络流量日志

2、 识别和修复不符合迁移要求的应用程序

3、 在确定设备可以在新网络上成功运行之后，迁移设备

这种策略允许网络层面相对稳定地应用新的配置，且能够独立于 BeyondCorp 的其他部分进行。BeyondCorp 的设计包括基于 802.1x 认证进行网络准入以及 VLAN 分配，这种方式能够将网络层与迁移策略的细节隔离开。更高层的软件和数据分析决定了设备的 VLAN 分配，并由 RADIUS 服务器将其返回给网络层。

实现这一系列目标任务艰巨，需要对技术/业务栈的每一层进行修改。但迁移团队并没有试图在一次过渡中修改所有层（毫无疑问这会引起灾难性的崩溃），而是分步实现：

- 解耦网络层：新的 VLAN、802.1x、RADIUS 策略服务器
- 解耦客户端平台升级：证书生成和安装，用户认证工具
- 完成不符合迁移要求的服务和工作流的修复，逐步地迁移设备
- 持续修正流程和程序

第一步：802.1x网络

在 BeyondCorp 的第一阶段，为每个用户设备安装证书并基于 802.1x 认证实现所有的网络访问准入。这个看似简单的步骤暗含了几个新的开发项目：证书颁发机构(CA)，为公司受控设备（针对所有操作系统）安装证书的工具，在网络交换机上启用 802.1x，集成一个策略驱动的 RADIUS 服

务。以上开发项目并行开展。

安全团队设计了一个新的证书颁发机构，通过提供 API 接口的方式，使每个操作系统平台管理团队能够在对应的平台上获取并安装证书。每个平台团队独立部署软件、工具和监测系统，执行和监测每个设备的证书安装。在与接入交换机集成的同时，我们还创建了批量分发和维护证书的流程。

同步开展的还有对接入交换机的重新配置工作，为接入交换机配置新的 VLAN 定义，开启 802.1x 认证，支持基于 RADIUS 的 VLAN 分配。自动脚本通过审计交换机的升级，来识别尚未配置新 VLAN 的交换机。这样，RADIUS 服务器就不会为这些交换机分配其尚未开通的 VLAN。

采用 802.1x 认证，就可以将 VLAN 分配的控制权从网络层转移到 VLAN 策略服务器。为了减少新 RADIUS 服务器可能引发的故障，初始策略仅匹配现有 VLAN 分配（包括复杂的黑名单和白名单）。一开始，配置策略服务器在审计模式工作，比对新的 VLAN 分配与既有的 VLAN 分配。当两者差异足够小，就启用新策略。此后，就可以使用软件和数据驱动的策略，接近实时地管理设备的 VLAN 分配。这个简单初始策略的使用，使得最终状态（和过渡）策略仍在开发中时，在网络层面率先启用动态 VLAN 分配。

以成功为导向的迁移

全面部署 802.1x 认证花费了数年时间，随后又花了更长的时间来实现基于清单、按信任等级动态分配 VLAN，并将其作为 RADIUS 策略服务

器的输入[2]。在这些开发工作进行时，需要识别出两类主要用户群和应用服务：那些准备好采用 BeyondCorp 的，和那些需要升级网络和安全能力才能兼容 BeyondCorp 的。首先，捕获和分析网络路由器的流量。通过日记记录和分析经过公司路由器的全部流量的部分采样，发现使用模式不兼容的情况。此外，这种分析还可以协助发现网络上的异常、意外和未经授权的流量。识别出这些不兼容 BeyondCorp 的应用，就可以尽早对这些应用进行兼容性改造，并避免对这些应用的使用者造成干扰。

有些网络用例，比如使用 NFS/CIFS 文件服务器的工作站，显然是不兼容 BeyondCorp 的。虽然 NFS / CIFS 文件服务器是实现文档共享和协同的最简单方法，但其底层协议不支持我们所需的安全属性（强加密和身份认证）。为了消除对 NFS / CIFS 的依赖，我们很早就启动了一个项目，来实现两个目标：一是将 NFS 主目录移动到本地磁盘，并通过自动备份同步至安全的云存储；二是使用 Google Drive 或其他安全的文件共享技术取代其它 NFS 的使用。即便如此，还是有些应用程序非常依赖 NFS，如 CAD（计算机辅助设计）编辑器，对于这种情况，我们在将其用户和工作站移动到受限的 MNP VLAN 之前，就需要定制解决方案。在下文的“修复困难用例”一节中将讨论如何处理这些特殊需求的框架细节。

还有些不兼容的工作流不那么容易判断出来，但一旦受到 MNP 网络的 ACL 限制时，这些业务就会运行失败。让其失败是必要的，因为我们无法假设 NFS、RDP、SQL 等具有足够的身份认证、授权和加密能力。当不得不在网络层面进行修复时，检测出这些工作流后通过改变设备的网络分配来恢复其生产力，费时费力。为了避免这种情况对生产力产生巨大影响（更不用提影响用户的情绪），需要一个分析驱动的策略，在将用户分配到

MNP VLAN 之前，预先检测并修正可能失败的工作流。

为了方便在无特权网络上进行简单的分析和用户工作流测试，我们创建了一个基于 C/S 架构的网络 ACL 仿真器，仿真器能识别被 MNP ACL 阻塞的网络数据包。底层技术采用 Capirca（参见源代码[4]），并依据真实的 MNP ACL，创建本地 iptable 规则或其他的包过滤规则。在分析和迁移阶段，用户设备继续在特权网络上运行，而 MNP 仿真器监视网络流量，并将所有非 MNP 兼容的流量的源和目的地址记录到中心数据库。IP 源地址标识潜在故障用户，IP 目的地址标识潜在故障服务。通过分析日志（必须考虑适当的隐私限制），可以识别出已经兼容 MNP 的设备，从而将它们分配到 MNP VLAN。同样，可以识别出暂不兼容流量的设备、用户和服务，并启动项目将这些服务转移到为其需求其他解决方案。随着时间的推移，更多的设备变为兼容设备并被自动分配到 MNP VLAN。

在第二种模式下，MNP 仿真器实际上也可以阻止/丢弃非 MNP 流量，从而在不依赖 MNP VLAN 和 802.1x 管道网络层部署的情况下强制执行 MNP ACL。尽管 ACL 的最终执行是在网络设备中完成，设备中将 ACL 与用户（或黑客）的滥用隔离，但在试用和过渡阶段，在客户端工作站上启用和禁用这种“强制”模式要更容易、更迅速。客户端强制执行模式既是迁移过程中的重要步骤，也是用于测试验证的自助服务工具。如果当初没有这种工具，BeyondCorp 迁移团队恐怕难以实现最终快速、成功的设备迁移工作。

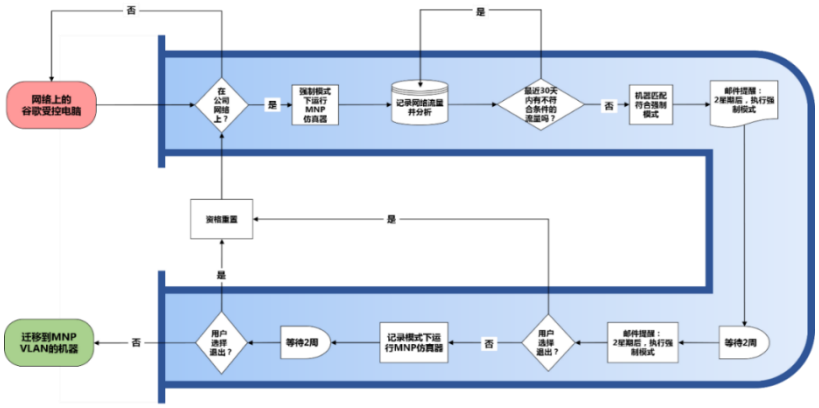


图 1 将谷歌电脑迁移到受控的无特权（MNP）网络的管道

使用访问代理处理简单用例

谷歌的基本安全策略要求所有从工作站流向服务器的业务流量都需要：

- 已认证（识别发出请求的设备和用户）
- 已授权（验证用户和设备是否被允许访问后端资源）
- 已加密（防止窃听）
- 单独记录日志（为了协助取证分析）

对于 HTTP/S 流量和 HTTP 封装的 SSH 流量，访问代理[3]可以满足以上所有要求。

幸运的是，在谷歌内大多数高频使用的应用程序都是基于 B/S 架构的 Web 应用。这种“幸运”并非巧合，因为谷歌有一独特的核心理念在业界“闻名”：尽可能的使用基于 B/S 架构的应用。谷歌为每个 Web 应用提供者准备了工具和文档，让每个应用提供者都可以配置自己的应用运行在访问

代理之后。

当一个应用运行在访问代理后端时，企业和公共 DNS 包含一个可以解析到访问代理的 CNAME，这样此类应用的 URL 在企业和公共网络中都具有同样的易用性和安全性。能从公共网络访问企业应用就意味着，经过身份认证的远程用户可以直接访问企业 Web 应用，而不需要再拨通 VPN 进行访问。因此，使用和支持 VPN 连接用于远程办公所需的费用会立即大幅减少。据粗略估算，由此产生的生产力提升轻松超过了 BeyondCorp 的实施成本。

一旦基于 B/S 架构的应用在访问代理后受到安全保护，我们就可以大刀阔斧地推进了。通过启动一个自动化流程，分析、验证并将设备迁移到无特权网络;在不到一年的时间里，超过 50%的设备迁移到无特权网络访问模式。

修复疑难用例

虽然可以通过访问代理来处理大多数的应用，但还有些应用难以通过此方法处理。整个迁移的时间安排还必须考虑到非 Web 案例的长尾问题，因为这些问题用例需要更多的时间和资源。为保证这些用例能够兼容 BeyondCorp，需要新工具、新技术和工作流改造。

特别是，一些工作小组使用基于非 HTTP 协议的第三方桌面或“胖客户端”应用程序，这就涉及到一系列特殊的问题。例如：

- 有些工具原本就是依赖网络文件共享。
- Java 应用程序可能使用远程方法调用（Remote Method Invocation，

RMI）或其他直接套接字连接。

- 许多工具可能需要使用非 HTTP 套接字和协议连接许可服务器。

即使是基于 HTTP 的应用程序，也可能遇到一些莫名奇妙的、出乎意料的问题。例如，有些应用无法支持客户端证书或适当的用户凭证，而有些应用则内置了一些负载均衡逻辑，导致不易和访问代理整合。对于其中一些案例，通过调整访问代理，允许来自 MNP VLAN 的流量在没有证书的情况下通过。这种临时策略效果还不错，因为设备必须出示证书才能访问 MNP。每个有问题的案例都需要一个诊断和补救项目。

为了解决这类疑难杂症，开发了一个解决方案，使用多端口加密通道来传输客户端和服务器之间的流量：

- 当客户端向服务器发起连接时，访问代理使用常规的用户和设备身份认证及授权。
- 客户端上的路由表将数据包发送到 TUN 设备，该设备可以捕获和加密到特定后端服务器的流量。
- 加密后的数据包采用基于 UDP 的封装协议直接在客户端和加密服务器之间传输。
- 加密服务器只允许应用程序必须的服务和端口流量通过。

用例	**解决方案**
B/S 架构的 HTTP/S 连接	访问代理
HTTP 命令行的原生应用： 提供了一个客户端代理服务器程序，该	本地认证代理程序

服务器提供平台证书，以建立与访问代理的认证与加密的连接。然后，将简单应用定向到本地主机代理。	
单个 TCP 连接： 对于需要 TCP 套接字连接到服务器的应用，一般通过与后端堡垒机建立 SSH 连接来解决，并为简单 TCP 应用端口建立隧道	SSH 隧道和端口转发
多端口或无法预测的端口号	加密服务隧道
对延迟敏感、实时，UDP 流	加密服务隧道

表 1：解决问题工作流的方法

这种方法可以让第三方传统应用更安全地从任何网络连接到它们的服务器，同时也满足了 BeyondCorp 要求的身份认证、授权和加密。

表 1 描述了解决问题工作流的常规方法。详细论述请查阅《BeyondCorp，访问代理》[3]。在有些场景下，表 1 中的解决方案还要求用户通过运行脚本或在访问后端资源之前提供必要的身份认证来修正工作流。

有些基本框架服务也不具备兼容性。当然，这些关键服务的兼容问题并未阻止迁移的整体推进，而是通过开通从 MNP 到特定端口或服务器的临时访问权限进行解决。为了防止这些临时例外变成常态甚至颠覆 BeyondCorp 的基本目标，只有服务所有者给出实现和部署兼容解决方案的明确计划时，我们才允许进行临时的例外放行。

随着一个一个的应用或用例完成整改或调整，借助自动化的分析、验证和迁移工具，越来越多的用户和设备转移到无特权 VLAN 上。随着工作推进，网络日志记录和分析可以用于度量已成功迁移到 MNP 的用户和设

备数量。

逐步上线并不断完善迁移方法

MNP 仿真器，分析管道，以及将设备自动分配到 MNP VLAN，组成了一个重要的软件开发和流程再造项目。所以整个项目的开发和部署也是逐步完成的：首先在针对各个阶段进行小规模测试，持续修复软件，合适的用户调整通告，培训技术支持团队，然后逐步推进到全面部署。

当识别出那些不兼容工作流的用户，仿真和预分析的方法有助于规避对这些用户的负面影响。然而，这种方法将所有新配置的、尚未分析的设备分配给特权网络，并且没有阻止未迁移的用户使用或创建新的不兼容应用，因此它不能作为长期策略。通过纠正大量用例来减少异常案例后，实施方法变为“默认采用 MNP”策略。随着工作逐项推进，全部设备被默认分配到 MNP，同时对那些由于工作职责需要使用未修复应用的用户设备，予以例外处理。这个基于策略的分配完成了从“证明用户会成功，然后迁移设备”到“假设用户会成功，直接迁移设备”的演变。

扩大支持，尽量减少对员工的影响

使用上述工具和流程，能够自动识别、联系和迁移整组用户。但无论是在迁移开始前，还是出现问题时，都需要一些办法来帮助用户，与用户沟通。技术支持的专业培训和增加与用户的沟通和互动，这两点对将工作流迁移到新模型至关重要。

技术支持赋能

在支持团队中培训一批技术人员，将他们培养成为 BeyondCorp 模型的专家和本地的主要接口人。项目上线的初期，这些技术人员帮助受影响用户能够在不影响迁移策略的情况下迅速恢复工作，还能有效地将问题准确地反馈给实施和策略专家。一开始，这些受过专业训练的技术人员比其他部门同事获得了更高的访问修复系统的权限。作为 BeyondCorp 上线的第一批“观察员”，他们可以提前参与思考，接下来的技术支持会需要哪些方式、工具和流程。此外，他们还通过全球科技论坛、讨论列表、午餐时间和办公时间来给其他支持团队做培训。随着信息的不断传播，将系统访问权限赋予全部支持团队人员。

成立本地专家组，使 BeyondCorp 团队能够直接与工作流不兼容的部门进行沟通。在本地专家组中确定一个资深对接人，问题部门就可以与 BeyondCorp 团队项目经理直接沟通，一起找到解决方案。与此同时，允许并鼓励技术人员，让他们在发现问题后立即在内部文档中添加新的临时变通办法或修复手段，以便将解决问题的能力尽可能遍布全网，更有效地实现信息共享并获得规模化支持。

自助服务

为了避免出现海量问询，需要尽量减少员工疑问，并在无需技术人员人工干预的情况下能够对常见问题进行回答。当用户被选中进行迁移时，系统会自动给他们发送一封启动邮件，内含明确时间安排、迁移将如何影响他们的工作、以及项目信息链接、常见问题答疑链接、自助链接和加急服务点。

此外，还提供一个自助服务门户网站，允许受业务关键时间节点约束的用户延迟迁移。为了回答问题，并进一步扩大信息传播范围，创建了一个内部讨论列表，征集员工答案。通过对常见问题的分析，能够快速迭代启动邮件内容和项目文档。

在整个上线过程中，通过专门的 Web 应用，我们还能快速迭代并改进故障处理指南。这个 Web 应用清楚地识别了常见问题（例如，解释为什么用户被拒绝访问某个资源），提供了解决问题的步骤，并链接到知识库文章。用户可以解决诸如组成员关系和证书问题等常见问题，从而减少对技术支持的请求。Web 应用还通过将来自许多不同层面和系统的信息合并成一系列操作，来帮助技术人员解决错误。

内部宣传活动

团队还组织内部宣传活动来提高大家对 BeyondCorp 的认识，比如推出了电脑贴纸、标识和口号，还在办公室张贴随处可见的文章。这些材料都标明了自助服务和办公时间，任何人有任何问题都可以寻求帮助。BeyondCorp 团队坚持宣传、指导、提供帮助，这使其直接与用户建立信任、取得信誉、得到用户的理解支持。在整个过程中，企业内沟通和技术专家参与是至关重要的，尤其是在早期阶段，那时亟需为项目的愿景和潜在影响给定一个清晰的蓝图。

分阶段上线

BeyondCorp 最初是一个小规模试点，试点位置与项目团队很近。随着时间的推移，逐步延伸至具有本地技术专家的试点位置，最终扩展到风险高的工作流和距离项目团队远的地点。直到有了成功经验，用户支持，

以及对策略的信心，我们才开始实现关键业务流的迁移。在此过程中，即便上线规模和受影响工作流在增加，但技术支持负载却在减少。分阶段上线实施是迁移能成功的关键。

最终结果

通过持续分析和改进上面提到的所有方法，BeyondCorp 团队还建立了一个系统，确保 BeyondCorp 能够在全球范围内扩展，而不会对业务、支持或用户体验造成负面影响。并不是简单地通过人海战术，而是通过构建系统和流程来有效地处理问题、进行升级和培训。此外，基于良好的沟通、开放和高度一致的目标，我们确信用户会帮助迁移团队一起实现变革。

随着公司越来越多的人采用 BeyondCorp 模式，我们也仔细追踪了由于 BeyondCorp 上线所产生的支持案例。近几个月来，BeyondCorp 相关问题只占技术支持团队全部处理问题的 0.3%。一开始这个百分比有 0.8%，但随着文档、培训、消息传播和上线方法的不断完善，升级问题已经稳步减少。与谷歌内部其他大规模 IT 变革相比，BeyondCorp 的支持问题少了 30%。

结论

在提高安全的急迫性与改变终端用户的使用习惯之间总是存在矛盾。当基础设施和工作流的改变威胁到生产力的时候，这种矛盾只会升级。在

发展和稳定之间取得平衡，与其说是科学，不如说是艺术。BeyondCorp 能够取得成功、为员工所接受的关键原因是分析、可行的规划和主动沟通。

通过将 BeyondCorp 迁移工作划分为独立任务，可以确保各项任务并行向前推进，确保每个阶段的用户影响最低。尽管部署 BeyondCorp 到各个层级花费了数年时间，但每一个里程碑都实至名归。这个过程逐渐使远程访问变得更容易，更快捷，网络管理变得更简单，安全性得以增强。

开发出能实现 BeyondCorp 安全模型的技术很具挑战性。上线规划和管理用户迁移同样具有挑战性。注意一定要确保每次迁移对用户的影响最小，并且不会中断正在开展的业务。每一次成功迁移都带来了对这个项目价值的全新认识，并为用户和管理人员带来了持续的热情和对项目目标的接纳。通过赋能一个跨职能团队，其中包括每个技术和实施团队的代表、安全策略的责任人还有终端用户支持和通信方面的专家，BeyondCorp 项目最终取得成功。

在谷歌内部，已经能够将在 BeyondCorp 工作中所学到的东西应用到其他项目和服务中。其中最显著的就是最近为谷歌云平台(Google Cloud Platform,即 GCP)增加的新服务（比如基于身份识别的访问代理 IAP）。BeyondCorp 所获得的最大经验教训之一，就是当遇到其它用例时，一定要分步完成项目，并持续完善优化策略。尽管这篇文章关注的是谷歌自己的经验，但它所分享的经验可以在任何组织中采用，无论规模大小，当然，获得相关干系人的坚定支持至关重要。

致谢

感谢以下的人员，有了他们的帮助才完成这篇文章：希瑟•阿德金斯（Heather Adkins）、杰夫•贝尔德（Jeff Baird）、达伦•比比（Darren Bilby）、约翰•布莱迪（John Brady），维克多•埃斯科贝多（Victor Escobedo），辛西娅•霍伊奇（Cynthia Horiguchi），迈克尔•简诺斯科（Michael Janosko），罗伯•皮斯古德（Rob Peasegood），丹•波尔斯比（Dan Polsby），瓦尔•斯蒂里斯（Val Stiris）和罗里•沃德（Rory Ward）。

参考文献：

[1] R. Ward and B. Beyer, “BeyondCorp, A New Approach to Enterp rise Security,” ;login:, vol. 39, no. 6 (December 2014), pp. 6–11: https://www.usenix.org/system/files/login/articles/login_dec14_02_w ard.pdf.

[2] B. Osborn, J. McWilliams, B. Beyer, and M. Saltonstall, “Beyond Corp: Design to Deployment at Google,” ;login:, vol.41, no. 1 (S pring 2016), pp.28–35:https://www.usenix.org/publications/login/spri ng2016/Osborn.

[3] L. Cittadini, B. Spear, B. Beyer, and M. Saltonstall,“ BeyondCorp Part III: The Access Proxy,” ;login:, vol. 41,no. 4 (Winter 201 6), pp.28–33: https://www.usenix.org/publications/login/winter2016/c ittadini.

[4] Capirca is a tool designed to utilize common definitions of netwo rks, services, and high-level policy files to facilitate the develop ment and manipulation of network access control lists: github.co m/google/capirca.

【第五篇】BeyondCorp ：用户体验

BeyondCorp 系列论文的前几篇讨论了在实践过程中我们如何解决各个方面的技术挑战[1-3]。在迁移过程中，除了技术因素，还要考虑人的因素：在整个迁移过程中，必须始终将用户牢记于心，为最终用户尽可能地提供无缝的体验。当出现问题时,我们希望用户知道如何解决,去哪里寻求帮助。本文将讨论谷歌员工在 BeyondCorp 模型中的工作体验,从新员工入职，新设备配置,到遇到问题时如何处理。

创造无缝的新员工体验

对于许多新员工来说,BeyondCorp 模型这个概念是相当陌生的:他们习惯了通过 VPN、公司专属 WiFi、和其他特权环境来访问他们日常工作所需的资源。BeyondCorp 上线之初, 许多新员工继续向我们的 IT 服务台团队（内部称为技术站 Techstop）请求 VPN 访问。用户过去习惯性地认为如果不在办公室的时候需要工作，就要经过复杂的 IT 设置，需要 VPN。BeyondCorp 架构师原本以为用户不在办公室，有远程访问需求时，会尝试直接访问内网资源，并发现可以成功访问。这样看上去非常完美：无需用户申请访问配置，无需技术站的支持负担，简直就是双赢。然而事与愿违，（远程访问需要申请 VPN 权限的）用户习惯根深蒂固。

新员工入职培训

显然，在用户开始谷歌的 IT 之旅时，就应该让其尽早了解这种新的访问模式，因此我们在新员工入职培训时就开始介绍 BeyondCorp。在培训中，我们有意避免去讲解模型的技术细节，而是关注最终的用户体验。我

们强调用户不需要 VPN，可以“自动”获得远程访问权限；用户无需改变他们的工作流就可以在办公室、家里、飞机上，或咖啡馆工作。通过培训，我们向用户展示了 BeyondCorp 的谷歌浏览器（Chrome）扩展程序，作为 BeyondCorp 访问模型中最常见的面向用户的方式（有关扩展的更多细节，请参见下面章节“BeyondCorp 扩展”）；我们还展示了在 BeyondCorp 中代表连接“正确”的图标（参见图 2）。只要有“正确”连接标识，用户就可以通过任何网络连接访问他们需要的绝大多数工具和资源。

新设备安装配置

当用户初次使用公司账号密码登录其公司设备时，其访问设置将被自动配置。为了实现这种无缝的入职体验，清单进程和平台管理工具在后台工作，以配置新的租用设备并进行初始化。如“谷歌 BeyondCorp：从设计到部署”[1]中所述，我们根据大量的数据来判定设备的信任等级，包括观察数据（最近安全扫描时间，补丁级别，安装软件等）和预设数据（分配的所有者，VLAN 等）。为了解决这种判定的复杂性，我们的清单团队遵循自动配置流程，以确保首次登录时正确信任新租用设备。验证必要的用户账密后，我们会自动将自定义 Chrome 扩展程序推送到用户设备。从用户的角度看，只要能够看到扩展中的绿色图标，他们就可以访问企业资源。通过在新员工培训中讲解 BeyondCorp 的 Chrome 扩展，基本消除了新员工困惑，并且可以支持新员工的远程访问请求。

减少VPN使用

尽管新员工在培训中了解了 BeyondCorp，但毕竟他们在入职谷歌的头几天中可是接受了大量的信息冲击，让每个人都能回忆起培训中的每个

细节不太现实。于是我们修改了 VPN 申请流程和工具来强调在培训中讲解的 BeyondCorp 概念。

默认情况新员工没有访问 VPN 网关的权限，他们必须通过在线申请门户来申请 VPN 访问权限。在此门户上，我们明确提醒用户 BeyondCorp 是自动化配置的，他们在请求 VPN 访问之前应尝试直接访问他们需要的资源。

如图 1 中的流程图所示，如果用户跳过这个警告，我们还会对用户通过 VPN 隧道访问的服务进行自动分析。如果用户在过去 45 天内没有访问过任何一个 BeyondCorp 模式不支持的企业服务，我们就会向他们发送电子邮件，邮件中会解释，由于他们访问的所有公司资源都是通过 BeyondCorp 支持的，因此他们的 VPN 访问权限将在 30 天后到期，除非他们访问 BeyondCorp 不支持的服务。我们在 VPN 访问权限失效前 7 天会再发送一个通知，然后在第 7 天结束后取消用户对 VPN 网关的访问许可。这种自动化流程使我们主动剔除对传统访问基础架构的不必要使用，并最终完全拒用 VPN 基础设施。

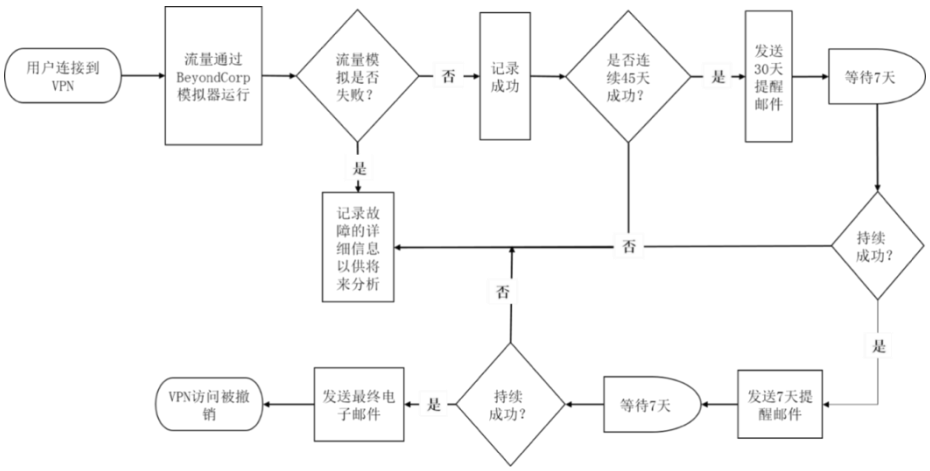


图 1 自动分析和取消员工的 VPN 使用

借用项目

实现 BeyondCorp 自动配置还带来了一个意外好处，为用户改进了其他方面的技术体验。其中一项最明显的改进是我们的借用笔记本电脑项目。像许多现代公司一样，我们员工的工作方式非常灵活，可以在办公桌，会议室，休息室或家中自由工作。移动设备 - 特别是笔记本电脑 - 对其生产力至关重要。为了处理忘带、遗失或被窃的情况，我们提供了一种自助式借用笔记本电脑程序，可以让用户尽快恢复正常工作。

使用遍布全球的自助式谷歌 Chromebook 笔记本电脑借用站，任何用户都可以将借用的笔记本电脑临时注册为自己的工作电脑，最长可达 5 天。从拿到笔记本到开启工作状态可能就几分钟时间，这样简单的流程让用户受益良多。借用设备开通足够简单，所需支持服务也随之减少，技术站的资源就可以释放出来处理其他问题。当用户归还设备或借用时间到期时，系统会自动撤销其证书，并降低其信任等级，为下一个用户重新借用做好准备。

BeyondCorp的Chrome浏览器扩展程序

通过或多或少地消除对 VPN 客户端的需求，我们可以通过 Chrome 扩展程序这个单一入口来封装几乎所有的访问需求——无论是远程访问还是本地访问。Chrome 扩展程序会自动管理用户的代理自动配置（Proxy Auto-Config，PAC）文件，明确将一些特定访问场景路由到访问代理[2]。当用户连接到网络时，该扩展程序会自动下载最新的 PAC 文件并显示“正确”连接的绿色图标。浏览器根据 PAC 文件中的规则自动将企业服务的访

问请求路由到访问代理。这使得内部开发人员可以不用明确配置客户端访问入口参数的情况下部署企业内部 Web 服务：客户端访问入口配置要求开发人员在公网 DNS 中配置 CNAME 指向访问代理，访问代理就会自动处理用户身份认证和授权。

由于 BeyondCorp 扩展程序将所有流量路由到访问代理，用户将无法访问那些访问代理不可达的设备。另外，扩展必须下载正确的 PAC 文件，以便准确路由业务流量。这种设置可能在某些场景下可能会出现问题，比如有强制验证门户的网络连接的场景，或用户需要访问本地网络上与设备而不希望通过访问代理进行路由。我们需要对用户解释这些问题并提供补救方法，最好不要增加技术站的支持负担。Chrome 扩展程序的认证状态图标（如图 2 所示）提示了进一步排除故障的方法信息。

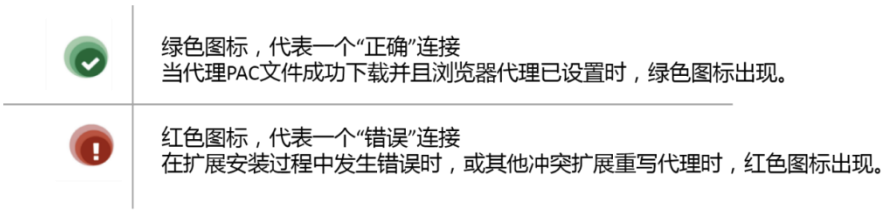


图 2 Chrome 扩展中表示身份认证状态的图标

当出现问题的时候

当出现故障或用户遇到复杂的边界情况时，会发生什么？通过假设用户会遇到的问题，确定常见场景，制定计划尽可能顺利地解决这些问题。让用户能够理解问题，并在可能的情况下自我修复，这是我们一贯的首要目标。

可以自我修复的问题

强制验证门户

因为我们是一家拥有许多差旅员工的全球性公司，当他们在机场、酒店和咖啡馆办公时经常会遇到强制验证门户。这些门户网页通常在私有网络的默认网关上，当用户连接到这样的网络时，BeyondCorp Chrome 扩展程序会尝试下载 PAC 文件，但是强制门户网页会阻拦 PAC 文件的下载。

要解决此问题，当扩展程序检测到网络状态变化时，我们都会确定设备是否位于强制门户之后：通过尝试访问 http://clients3.google.com/generate_204，正常情况下应返回 HTTP 204 的空页面。如果我们收到 HTTP 204 以外的任何内容（最可能出现的是 HTTP 302），就认为该设备需要先通过强制验证门户的认证。这种情况下，Chrome 扩展程序会直接使用内置的预定义 PAC 文件，并警示用户。

当用户碰到强制验证门户的场景，可以点击 Chrome 扩展程序图标，我们会告知他们在连接机场或酒店的网络的时候碰到强制验证门户的这个问题很常见。BeyondCorp 的工作不会受到影响，只需要将 BeyondCorp 的设置更改为“Off:Direct”即可，当用户完成强制门户验证后，浏览器扩展即可成功下载最新 PAC 文件。这个简单的流程允许用户在最短时间内完成自我修复，没有增加技术站的负担。

本地网络设备

用户还经常尝试访问私有网络中的设备，比如许多谷歌员工就会使用公司笔记本电脑来执行配置连接家庭打印机或其他网络设备等任务。但由

于 BeyondCorp 配置通过访问代理来路由所有连接，所以启用 BeyondCorp 扩展后，连接就会失败。与强制验证门户的情况类似，解决方案是将 BeyondCorp 设置更改为 Off：Direct。但不同的是，我们无法轻松检测到此故障状态。因为通常情况下，这种场景下的用户有一个激活的并且功能正常的互联网连接，因此从 BeyondCorp 扩展程序的角度来看，一切正常，用户可以通过 BeyondCorp 访问所有的企业资源，没有理由发出警报。

为了弄清楚在这种情况下如何有效地与用户交互，我们进行了一次典型的用户体验测试：工程师把公司笔记本电脑带回家，想用它来更改家里打印机的设置，两台设备通过 IP 地址连接。用户连接到家庭网络，BeyondCorp 扩展成功连接，下载最新的 PAC 文件，并配置浏览器代理。当用户在新建的浏览器 Tab 标签页中输入打印机的 IP 地址时，对私有网络的访问流量一起重定向到了访问代理。网络请求失败，用户得到错误提示。

我们将解决问题的关键点放到最终的错误页面上，并提出了一个解决方案：通过访问代理展示错误页面。我们创建了一个自定义 HTTP 502 错误提示页面，以便在某些场景下将警示信息插入到错误页面中。具体来说，特别是针对用户试图访问 RFC1918 或 RFC6598 约定的地址时，我们返回的 HTTP 502 错误提示页面可以明确给出提示，用户就会知道如果他们在访问本地网络设备如家用路由器或打印机时（两个最常见情况），需要将 BeyondCorp 扩展修改为“Off:Direct”。通过这种方式，我们能够基于现有的基础设施和流程，让用户自行修复问题。

自定义代理设置

我们的海外员工有时需要配置一些自定义代理来测试广告。如果用户

安装了多个扩展，每个扩展都试图配置代理，那么这些扩展就会相互冲突，这可能会使用户感到困惑，并影响他们访问企业资源。

我们用两种解决方案来处理这种情况。首先，我们将海外的代理配置直接集成到 BeyondCorp 扩展程序中。当用户有业务需求要从特定位置对外访问时，他们可以从支持国家的下拉菜单中选择该位置。这为用户提供了一个单独扩展来满足他们管理最常见的业务代理服务器的需求。

此外，当用户有合理需求运行额外的代理管理扩展时，他们的 BeyondCorp 图标将从绿色变为红色。然后我们给他们一个选项，将状态更改为 Off：System Alternative 并告知他们何时应该使用此设置。同样，这个过程允许用户进行自我修复，提高他们的工作效率并减少对我们支持团队的咨询的工作量。

复杂问题解决：门户

对于上述简单问题，可以通过自定义错误页面或 Chrome 扩展程序让用户可以快速地进行自我修复。然而对于一些看似正常的访问失败场景，用户和支持团队都会迫切需要知道被拒绝的原因。后端基础设施中的 ACL 逻辑复杂、层级多，无论对用户还是支持团队而言，想要理解这特定决策背后的逻辑都有困难。即使是一个经验丰富的 SRE 工程师，也可能需要花费很多时间查询许多内部服务，来找出一个 403 错误页面的原因。考虑到访问代理每天可能产生的 403 错误页面的数量级（仅 HTTP/S 就有约 12M/天），人工参与故障排除是不可规模化的，也是不切实际的。

为了方便诊断和解决更复杂的 BeyondCorp 访问问题，我们设计了一

个门户网站来帮助用户和支持团队。我们不只是用一串通用错误代码来告诉用户他们的访问被拒绝，而是解释他们被拒绝的原因以及如何解决这个问题。门户是独立的，而不是直接集成到访问代理服务器中，因为它使用的是更细粒度的 ACL，取决于最终用户当前信任级别。由于访问代理默认是公开的，所以我们需要限制攻击者从 403 错误页面中获得的信息量。

架构

门户大致分为前端和后端，两者之间采用 API 进行通信。

- 前端是一个交互式 Web 服务。它根据用户的输入内容向后端 API 发出请求。
- 后端可以查询参与访问决策的多个基础设施服务。这个过程会绕开各种缓存层，这样用户就可以接收到最新信息。
- 前端和后端之间的 API 也可以用于其他用途，比如批处理、分析，或者将输出能力嵌入到其他工具中。

解释引擎

除了查询和表示 ACL 外，门户还需要有效地向用户展示这些信息。针对这些被拒请求的响应报文细节，我们构建了一个解释引擎（Explanation Engine）来进行错误诊断。它通过递归遍历负责提供授权决策的子系统来完成操作。

例如，访问代理的 ACL 可能要求设备完全可信才能访问一个特定的

URL。在查询这个 ACL 后，解释引擎会和设备推断管道交互，并获取访问此资源的必要条件，然后将这个信息发送至前端，并翻译成通俗语言，用户就可以通过访问门户来找出他们当前状态存在何种问题以及如何解决这些问题。

为 ACL 定义 ACL

虽然解释引擎可以提供有效信息，但它可能会暴露敏感数据。它会暴露受保护系统存在问题的 ACL，泄露用户账号和设备状态信息，而这些信息都会为潜在攻击者所用。为这些数据定义 ACL 非常棘手，因为我们需要在工具易用性和保护敏感信息之间实现平衡。

根据用户和设备请求故障诊断信息，我们可以使用不太具体的信息替换输出中的敏感信息。在极端的情况下，我们可以将敏感信息替换为联系技术站的提示信息。这样技术站和 SRE 工程师就可以通过验证用户的身份并以用户的名义查看相关信息，在帮助用户的同时不泄露敏感信息。

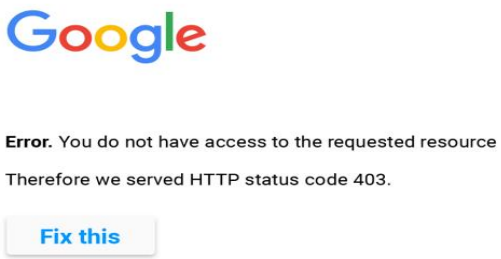


图 3 当 BeyondCorp 阻止请求后展现的错误页面

访问拒绝登录页

一旦门户开发完成，即可将门户集成到访问代理，向用户展示错误消息。当用户遇到 HTTP 403 错误时，他们可以一键返回到门户，查看所有相关错误细节（参见图 3）。然后，门户会向后端重新发送访问请求，并解释导致问题的原因。

例如，如果一个资源要求特定群组成员才可访问，门户会提供群组名和到群组管理系统的超链接，这样用户就可以申请访问权限。门户在后台查询后端的访问控制列表服务来判断该资源的授权要求，与用户当前的归属组信息进行比较，门户前端将比较结果转换为通俗语言（参见图 4）。这一切都发生在几秒钟之内，远远快于用户猜测什么是“组成员问题”或寻求帮助。

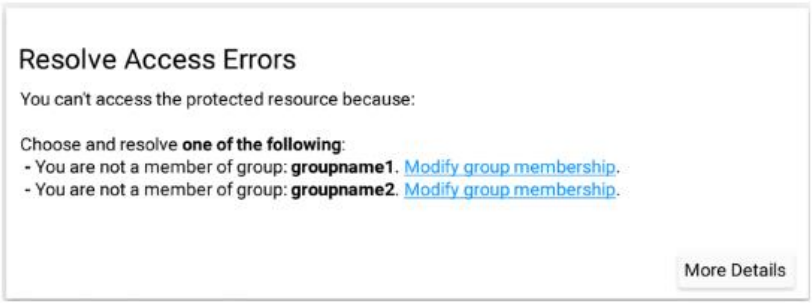


图 4 面向员工的访问拒绝错误故障排除指引

将这个流程直接集成到我们的错误消息中，允许用户可以无缝地完成这个过程——并且，完全通过自助服务。

临时的故障排除

尽管我们期望大多数用户通过错误页面访问到门户，但是我们还提供了一个独立页面来支持更多临时的故障排除。前端门户的登录页面是根据用户身份和访问设备自定义的，它会显示用户及其名下设备的信息，并突出可能导致拒绝访问的问题。我们允许最终用户主动访问这个工具来了解其名下设备的全局视图和潜在访问问题，用户就能一步到位地解决他们任何设备上的问题。去外地出差或者演示之前，使用这个能力进行设备信任度自查非常方便。

支持赋能

门户前端也使技术站能够快速地执行详细的故障诊断，提供立即可执行的方案，大大缩短了解决问题的时间。例如，为了解释一个 403 错误页面，技术人员就可以使用门户登录页面查询特定的用户名或设备标识，锁定到某个特定设备，确认它是否是一个完全可信的公司设备。如果不是，系统会给出设备不可信的具体原因，以及技术人员该如何解决这个问题（参见图 5）。

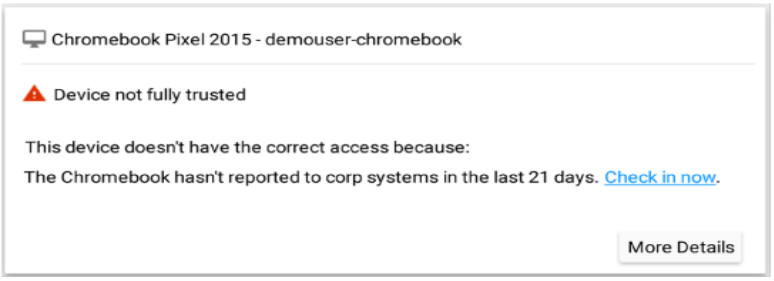


图 5 面向服务台的访问被拒绝错误故障排除指引

未来目标

除了当前的功能外，门户还提供了进一步实现自动化的可能。我们计划在将来持续检查潜在的拒绝访问问题，并会在这些问题真正出现前，告知用户如何自助解决问题的方法。同时，对于不能自我修复的重大问题，会启动自动通知到技术站采取补救措施。我们还希望扩大我们可以自动解决的问题范围，而无需人为干预。

聚焦经验

尽管 BeyondCorp 迁移在多个技术领域困难重重，但它也给予了我们足够的空间可以重新评估用户支持体验。通过关注迁移期间和迁移后的用户体验，我们可以使用户能够轻松地使用复杂的网络模型。专门设计的工具使出现在用户侧的组件变得清晰易用。这些支持界面的意义是为了允许用户尽可能地自我修复，从而节省用户时间，释放出支持团队的资源。当用户需要其他帮助时，我们提供有效工具和信息，使技术站发挥最优价值。

对于绝大多数用户来说，BeyondCorp 是完全透明的。当谷歌员工担心他们自己的业务流程时，BeyondCorp 模型负责处理全部的访问逻辑问题。当用户的确有问题时，我们会快速有效地介入，在合适的时间给他们提供正确的信息协助他们恢复正常。然后我们退回二线，让他们专注于他们最擅长的事情。

参考文献：

[1] B. Osborn, J. McWilliams, B. Beyer, and M. Saltonstall,“BeyondCorp: Design to Deployment at Google,” ;login:, vol.41, no. 1 (Spring 2016), pp. 28–35: https://www.usenix.org/publications/login/spring2016/osborn.

[2] L. Cittadini, B. Spear, B. Beyer, and M. Saltonstall,“BeyondCorp Part III: The Access Proxy,” ;login:, vol. 41,no. 4 (Winter 2016), pp. 28–33: https://www.usenix.org/publications/login/winter2016/cittadini.

[3] J. Peck, B. Beyer, C. Beske, and M. Saltonstall, “Migrating to BeyondCorp: Maintaining Productivity While Improving Security,” ;login:, vol. 42, no. 2 (Summer 2017), pp. 49–55:https://www.usenix.org/publications/login/summer2017/peck.

【第六篇】BeyondCorp：构建健康机群

任何系统的安全等级不可能超过其信任的所有其他系统的安全性。BeyondCorp 项目帮助谷歌基于其信任的计算平台，明确定义并施行访问决策，将安全策略从保护服务转变为保护可信平台。此前发表的 BeyondCorp 系列论文中，详细讨论了谷歌对设备“正本清源”所使用的各类工具，但尚未深入探讨建立设备信任的机制。

我们如此关注计算平台的安全性是有据可依的，业内大量证据已经表明[1]，终端用户已经成为各类攻击的首要目标，并且其攻击手段层出不穷。攻击者通过设计巧妙的社会工程学攻击，将恶意代码安装到设备上，随后即可利用现代操作系统的大量攻击面发起攻击。高级攻击者目的在于利用设备固有的信任、设备的身份凭证、或者已授予用户的信任来进一步利用系统。

为了成功防止这种持续的在可信内容（企业 Web 应用程序、企业凭证）和不可信内容（外部软件库、社交媒体、个人电子邮件等）的混合环境中可能遭受的破坏，平台自身必须具有层次化的、一致的若干控制措施，最终，作为设备机群（fleet）的构成要素，计算平台就成为了新边界。

基于前期工作展开

本文描述的工作内容主要基于白皮书“大规模设备机群管理”[2]和之前发表的 5 篇 BeyondCorp 论文[3]。在这些前期工作的基础上，我们的目标是通过以下方式进一步增强 BeyondCorp 模型：

1. 从通用的控制视角来定义健康设备机群

2. 确保始终如一地全面应用和衡量这些控制措施，并强制执行

3. 基于持续度量推动控制措施的闭环改进

定义待保护环境面临的威胁

与其他安全防护工作类似，我们首先必须定义待保护环境所面临的威胁。在创建威胁列表时，需要考虑一类攻击而不是各种可能的单一攻击向量。攻击者会持续发现新的攻击向量，这意味着很难穷举环境可能面临的所有威胁。但是，如果成功减少了某一类攻击，即可减少对这一类威胁中的多数攻击向量[4]。

从一个较高的层面来看，平台应该考虑的威胁类别包括：

1. 未知设备：由未知的或非受控设备对敏感系统发起的访问；

2. 平台失陷：利用计算平台上操作系统或软件的错误配置；

3. 安全控制旁路：未生效或错误配置的安全策略导致的系统失陷；

4. 非法提权：通过代码的执行导致系统特权被接管，且持续控制；

5. 软件失陷：恶意软件安装和持续存在；

6. 持续攻击：由于缺乏检测而导致攻击者长期攻击；

7. 认证旁路：通过被盗密码或绕过认证机制，造成平台失陷；

8. 数据泄漏：对磁盘、内存或传输中的敏感数据发起未授权访问；

9. 攻击隐蔽：由于缺乏日志和监控手段导致攻击者长期持续存在；

10. 攻击抵赖：攻击者通过掩盖其攻击痕迹而妨碍取证分析；

通过改善设备机群健康来解决环境威胁

定义威胁后，就可以更好地识别出缓解这些威胁所需的控制措施。并且进一步在服务访问时对这些控制措施的状态进行度量（有效性、是否开启等）。表 1 列出了上述各种威胁类别对应的控制措施，对一个理想的可信平台来说，这些措施是必不可少的。

#	威胁	控制
1	未知设备	设备机群清单库和资产管理
2	平台失陷	操作系统&基础软件配置管理
3	安全控制旁路	安全策略管理&强制执行
4	非法提权	防止系统接管
5	软件失陷	软件控制和反恶意软件
6	持续攻击	可远程验证平台状态

7	认证旁路	对平台和用户的可靠认证
8	数据泄漏	数据保护
9	攻击隐蔽	基于日志的检测能力
10	攻击抵赖	平台响应能力/检测&响应

表 1 威胁种类和潜在缓解机制

健康设备的特征

健康的设备机群由健康的设备组成，这些设备的健康由工具、流程和设备健康维护团队提供保障。设备如满足以下条件，即可认为是健康的：

- 可以承受大部分攻击；
- 提供了足够的检测和度量能力，在出现问题时能及时止损；

下面我们将深入研究，为什么这些控制手段至关重要。

设备机群清单库和资产管理

硬件是操作系统和应用程序运行的基础。通过限制硬件配置的差异性，可以更有效地判断出设备机群中设备的能力和不足。设备清单系统通过设备访问开通机制，为可访问敏感系统的设备范围进行了明确界定。

操作系统&软件配置管理

软件管理是保证设备机群健康的关键组件。一个集中式的软件管理基础设施有利于保证平台配置的一致性，以确保可信平台满足以下条件：

- 默认是安全的，并且随着时间的推移仍能保持最小偏离
- 能持续地进行安全的升级

为正在运行的操作系统、敏感的软件栈和安全代理打补丁的能力对于健康的安全态势至关重要，软件配置的管理也同样不可或缺（如软件自动更新策略）。

安全策略强制执行

可信平台应始终如一执行安全策略，并且报告和记录与预期策略之间的偏差。安全策略通常与上面提到的操作系统管理和配置策略交织在一起，但安全策略是独一无二的，是用户无法规避的强制访问控制策略。例如，通过同时登录限制策略可以减少横向移动的威胁；默认禁用 root 权限，有助于缓解恶意进程可能造成的危害。

防止系统接管

通过叠加防护层，确保恶意软件无法破坏系统的安全性。在高级恶意软件屏蔽掉主机日志子系统之前，确保主机可以报告异常行为。

软件完整性及控制

可以限制未授权代码在平台上执行。常用策略包括，仅允许已知的“好”软件运行，明确阻止可疑 “坏”软件运行。我们通常会选择白名单策

略，因为明确定义出工作所需的所有应用程序是可行的，但需要阻止的所有潜在坏人或恶意软件却无穷无尽，无法枚举出来。

可远程验证平台的状态

平台应具备基于密码学的完整性验证机制，在底层平台上提供从固件到运行的操作系统的完整性保证。可行的机制包括：第一命令执行控制[5]、安全启动和远程验证。

平台和用户的可靠认证

在尽可能的情况下，用户账密信息存储应使用系统上的硬件支持或硬件隔离。Windows Defender Credential Guard[6]就是一个很好的例子。

数据保护

我们假设每位用户的系统都有一些敏感数据，因此敏感数据在存储和传输过程中都应该被加密。为了应对设备丢失或被盗的情况，设备应支持远程数据擦除功能，可以销毁一切存储在系统上的数据和长期凭据。

基于日志记录和日志收集进行威胁检测

为了提供纵深防御，平台威胁模型应该假设攻击者能够绕过预设控制措施并且设备存在攻破的可能性。为了缓解此类风险，平台需记录这类事件。日志应该记录发起操作的用户和设备的相关属性，对所有敏感数据的访问或修改，包括对平台安全控制机制、状态和行为的更改都应

该详细记录。这些信息应该输出到一个集中式日志记录工具，理想的日志记录策略必须能够阻止未授权进程对其篡改。

平台响应能力/检测和响应

如果检测到威胁，平台应该提供有效手段，让得到授权的入侵分析师可以进行远程事件响应。诸如 GRR 之类的工具可以提供远程访问能力来执行这类分析[7]。人工检测工作应尽可能保持在最低限度，因为它无法规模化地应对广泛的攻击行为。理想情况下，授权分析师应该能够创建用于调查取证的有效时间线，可以从受影响系统中一次性的拉取数据来进一步调查，通过复现事件，检测和响应团队可以全面了解发生的情况并做出相应的响应。

维护健康的设备机群

具有上述控制特征的一组客户端设备构成了一般意义上健康安全的设备机群。为了达到这个状态，首先要弄清楚如何一步一步地建立平台信任。

建立信任

敏感的服务只能通过可信设备访问，我们为系统信任划分等级，基于特征和行为，设备能够获得不同的信任等级 [8]。

然而，这种方法带来了“鸡生蛋蛋生鸡”的问题：将设备转化为可信状态，需要首先访问客户端软件库，而客户端软件库本身就是一个敏感系统。为了解决这个问题，在设备从不可信到可信的迁移过程中引入了

“已识别”状态。状态为“已识别”的设备是指那些清单库认为信誉良好但由于某种原因还未取得信任的设备。可以允许这些设备访问客户端软件库的某个子集，以便安装补救软件。补救软件使得机器能够报告设备状态、下载和安装所需补丁，并采取所有必要步骤来满足对于可信平台的要求。

随着健康设备机群构建工作的开展，就会更加了解自身环境，就会更有信心地开通访问权限。另一个挑战就是需要确保技术和业务的不断演化过程中，信任能够持续得到保证。下文将讨论随着技术和业务的演进如何保持设备机群良好的健康状态，以及如何在健康状况恶化时迅速纠正。

对抗设备熵

一旦设备发放到用户手中，其安全性会逐步减弱，因为随着时间推移，安全性难以避免的会衰减。在对抗设备熵的过程中，我们发现了一些有用的策略。

第一条策略，也是最强大的策略，即将访问决策与清单系统集成。在获得授权内部资源的访问权限前，所有的设备都应在列且被信任。对机群的每一台设备，在接收和镜像的过程中，确保其信息都添加到公司清单库。对于任何上报为失踪、被盗或丢失的设备,应立即删除其访问权限。为了保证用户能及时报告设备丢失或被盗，要求用户必须在收到新的替换设备之前进行主动报告。

对访问环境的任何设备状态变化采用严密的监测和度量手段非常重要。Facebook 的 OSQuery[9]是一个优秀的开源监测工具，适用于

Linux、OS X 和 Windows 系统：它可以监测设备属性，如设备的操作系统版本、关键软件的补丁级别和加密状态。

另外，补丁和配置管理工具[10]能够改变设备的状态，将不可信的设备转换为可信的设备。BeyondCorp 通过限制用户访问的方式，强制用户进行某些必要操作，例如重新启动或接受更新。

检测不健康的主机

在主机的整个生命周期中，某些操作或不作为都可能会导致设备转变到不健康状态。信任推断引擎[11]通过持续信任评估来检测设备状态变化。当设备不能满足信任标准，则将设备的信任状态降低为“已识别”，通知机器的所有者并为其提供设备的修复指导。

检测与响应团队可以为信任决策提供额外的信息，这个团队有权删除对任何恶意设备的信任。

提供灵活的策略

乍看起来，设备机群健康状况的定义是一项简单的任务。但是，和大多数 IT 环境一样，魔鬼总是隐藏于细节（和例外）之中。在处理大量不同的操作系统和各种用例时，会遇到许多这样的细节问题。

为设备机群上线控制措施时，我们会尝试为策略的合规性设置一些阈值，而不是一上来就提出绝对严格的要求。这种策略允许用户在良好的状态下更灵活地运作，并能避免使用会让用户崩溃的严格规则集（这会导致用户寻求规避手段）。例如，如果用户需要安装非关键补丁，我们会在降低其访问权限前给予一个宽限期。

同时，通过一些防范控制措施为事件监测和响应功能提供信号也非常重要。为此，我们努力将这些控制措施集成到安全信息和事件管理管道中，以便可以报告和记录策略相关数据。捕获访问有关的数据可以帮助后续的调查取证和事件检测， 这些数据可能包括什么时候允许访问，以及根据策略，什么时候对访问进行了拒绝。

试点并推广这些原则

这是一个典型的由安全团队及其合作伙伴主导的项目，从开发到上线，始于设计和原型阶段，小规模测试，从设备机群和用户那儿搜集反馈并逐步完善。我们逐渐达成一项策略，即首先在监控模式下推出控制措施并建立内部试用团队（Dogfood）[12]以方便调试。例如，我们可能会推送一个新的 USB 审计代理到部分硬件工程师组织，因为这部分人员经常与自定义 USB 组件交互。通过这种方式，发现在大样本量的情况下，边缘情况并不会集中涌现。还有种方案，按照地理位置来划分内部测试，当然必须在变更实施前准备好本地支持人员。

在新控制措施上线时，清晰的沟通有助于了解新政策及其存在的原因。将每个控制措施映射到它所解决的威胁可以帮助每个人理解安全团队选择特定操作的原因。高度透明和对标准的清晰解释帮助我们加深了与用户间的理解，建立了与干系人的共识。当他们看到我们并没有任何隐藏的目标或动机时，就会充分参与到这一愿景及目标的规划中。一般来说，负责这种安全驱动的变革团队可以从清晰的全局目标阐述中受益，使得每项行动师出有名，更有利于争取到合作伙伴团队的支持，这种支持会形成一个良性循环，为“如何使设备机群更安全”带来更好的闭环。

平台度量和一致性控制

一旦定义了预期效果的基线，会发现某些控制措施无法普遍应用，因为无论是设备本身还是管理/策略层面，平台的能力都参差不齐（有时甚至能力差别很大）。例如，Chrome 操作系统的 Secure Access 提供了可靠的软件控制，但是 Linux 却没有开箱即用的防恶意软件的能力。为了确保整个设备机群获得一致的安全性，需要对安全评估进行规范和统一。虽然希望在不同平台上 100%实现完全相同的控制效果是不可能的（因为平台能力和威胁模型不同），但也并非绝对，对于所有需要实施的控制，可以将评估标准统一为“对安全风险是否有效”。

为了完成统一评估，分析了所有相关平台目前在满足控制效果方面的现状。然后，评估了现状和理想差距的总体情况。为谷歌管理的每个平台都创建了总体设备机群健康报告——这不是“成绩单”，而是对其能力的分析材料。针对每个平台，都需评估以下方面：

- 平台是否能够支持控制措施？
- 控制措施是否默认开启？
- 控制措施的状态是否可以度量？
- 设备机群是否合规？

要推动各项目标的可度量行和可比较性，可能需要考虑：

- 将这些策略统一到标准的度量单位中：如，自补丁发布以来的时间，地理位置，数量等

- 从相对量的角度来推动度量的标准化：如，和当前版本的偏离量、功能特性的实现比例等

难点在于如何设置这些评估标准。一旦拥有了各平台的可比较的度量标准，探讨设备机群健康状况的能力将大大提高。

当防护控制措施无法生效或者仅部分生效时，可以寻找其他的方式来消除风险，例如，更全面的监控/检测在某些平台下可以作为防护控制措施的补偿。这种评估方式可能看起来有点主观，有点依赖于对平台抵御攻击能力的整体感觉。现代操作系统有非常复杂的攻击面、能力和威胁模型；我们发现聚合所有这些信息的最佳方式，仍然要采用手动比较设备所需特性与实际具备的特性。这种比较允许我们能够围绕项目提出顶层建议，以填补缺失并提高项目的优先级。无论是基于什么数据得出的结论，重点是必须记录下这些结论背后的缘由，或至少记录下产生结论的过程，这能让应急安全工程师之外的人了解设备机群的状态。

与理想情况的偏差

尽管在定义、上线、测量和强制执行控制上，我们都做出了最大努力，但仍不可避免地需要面对这样一个严峻的现实：要部署 100%统一的控制措施过于理想，这种理想也许仅存在于独角兽自由玩耍的上古神秘国度，那里没有恶意软件，没有国家级的网络攻击者。现实是残酷的，我们需要针对偏差制定计划、进行根因分析、妥善处理例外情况。

许多偏差的发生是在所难免的，包括流程中断、管理工具故障、不稳定的版本发布或其他各类原因，都可能造成偏差。例如，为系统打上

最新补丁总是会有延迟的。重要的是要了解什么时候需要对设备机群全范围内的例外情况进行处理，需要防止异常规模的增长，但不要总是通过控制措施来强硬纠偏。如果懂得如何在威胁模型和用户影响之间权衡，就不难做出正确的决策。

例外情况应该可以被度量并且有明确的时间窗口限制。我们建议在整个设备机群范围采用一致的方式对根因进行分类，以便了解当前差距，并明确出哪些控制措施不适用于某些设备机群或某类用户。如果例外在不断更新（又或者永不过期），控制手段就会失效。这时应重新设计控制措施甚至重新审视这项控制措施在设备机群中的角色。

启动

那么，如何将本文讨论的 BeyondCorp 原则应用到你的设备机群呢？一般包含以下 4 个步骤：

1. 定义需要重点考虑的安全控制措施；
2. 找到度量这些控制措施的方法；
3. 判断设备机群的不合规项；
4. 修改工作流，让其符合预定安全策略，或将其定义为例外。

第一步十分关键，目的是为了确定想要实现的目标。我们不应毫无根据地开始创建安全控制，而是应该明确这些控制是针对某项威胁的具体应对机制。明确列举出威胁列表，不仅能够为度量控制的有效性提供

启发，还能为梳理每个特性的优先级提供一个框架。当为这些特性进行定义和排序时，需要咨询合作伙伴的意见（详见下文“经验教训”）。当已经明确威胁及缓解这些威胁的控制措施时，就可以通过测试来评估这些控制措施的有效性，包括单元测试、端到端的红队渗透测试等。基于这些举措，可以进一步明确这些控制是否有助于在实践中达到安全目标。

为了持续确定设备的安全态势，必须能够对其当前状态与理想状态进行监测和度量。如果尚不能实现度量，需要在设备机群安装度量监测软件来收集相关数据。不过，即便获得了原始监测数据也才只完成了一半，我们还需要定义待测量设备的理想状态。由于设备机群所包含的设备多种多样，需要定义多个理想状态，尽可能覆盖所有潜在的用例。

一旦可以度量设备机群的安全态势，就可以开始检查设备实际状态与理想状态的偏差。一些偏差可能不会带来安全风险（因为它们可以通过补偿控制来缓解），但仍有许多偏差将会暴露风险。一开始，我们就需要确保新设备从员工使用它们的第一时间起就符合控制要求。确保所有新设备都能从一个已知的良好状态开始加入机群，这样我们就可以将注意力放在设备机群中的其他设备上来提高设备机群的整体健康状态。

建立一个例外框架，这样在执行一个重要的新控制措施时，就可以先为现有的设备机群创建例外。此时整个设备机群的偏差将保持静止不变，于是就可以保持新设备符合要求的同时逐步修复现有设备。一旦将问题明确隔离到设备机群的例外集合中，就可以将故障原因进行分类，就能够发现整个设备机群或工作流所共有的一些问题。首先解决其中规模最大、风险最大的问题，以最小的代价获得最大的安全性。重复这样的分类及修复过程，直到已经解决了设备机群中的所有问题。如果用户

的工作流与所需安全特性明显不兼容，这种情况可以作为例外特殊处理。

虽然这个系统需要许多不同团队的大量协作和努力工作，但完成这项工作可以提升我们在面对持续攻击时的灵活性。

经验教训

制定一个连贯的计划来衡量和评估信任和设备机群的健康状况并不是一个短期项目。完全达成本文所描述的目标（以及 BeyondCorp 的其他目标）需要许多重要资源。因此，希望谷歌在过去几年中学到的一些经验教训可以为读者们节省一些时间和麻烦。

尽早设置目标里程碑

尽早设定关键目标里程碑。确定重要资产并对它们进行（至少粗略的）排名。这样有助于有效地分配资源，并为实施大型项目提供合理的动机。将设备机群管理系统的数据整合到授权决策过程中，是一个很好的初始里程碑。仅此一项就可以防止未知设备访问关键服务，同时还能生成一份已知良好设备清单。

确定如何处理例外

在项目中尽早定义例外处理框架。每个设备机群中都包含那些无法完全符合理想安全状态的设备。确定例外管理的流程和技术实现是成功部署的关键。定义允许创建例外的各种场景及其理由，记录这些场景，确定允许该例外的最大时间窗口，梳理已有例外的审核过程。

与合作伙伴互动并且尽早影响其他团队

BeyondCorp 的成功实施需要整个 IT 部门的配合。尽早与合作伙伴和可能受影响的团队展开合作，这样会大大提升后续实施上线的顺畅性 。

例如：

- 设备采购和上线团队需要确保在设备加入或退出设备机群时，设备机群管理系统进行更新，保持最新状态。

- 其他安全团队在定义设备的安全属性时可以提供有价值的输入，来自安全团队的各种输入对整个项目的价值重大。

- 传统的 IT 支持团队将负责绝大多数用户升级。 他们必须了解项目的目标，并能够帮助解决用户问题。

同时，我们也需要知道如何与那些会受到影响到的用户进行沟通，确保普通用户能够真正遵循并完成自我修复过程，减少故障排除时间，减少 IT 负担。

结论

保护员工设备安全是保护企业关键信息资产安全的基石。为此，我们彻底评估并定期检查所有企业设备来确保其健康状况，只有已知安全设备可以访问关键的内部系统和信息。

员工及其设备已经引起了坏人的注意，因此我们有义务在保证其安全性的同时维持生产力。为了达成这一目标，需要强烈的设备机群健康意识，明确的策略和衡量标准，以及处理目标偏差的流程。通过一致的控制和强制执行，每个企业都可以提升设备机群的健康和安全性，提高对不断增加的各类攻击和威胁的防御能力。

致谢

BeyondCorp 仍然是谷歌当前一项大型的跨团队项目，该项目有很多贡献者，在此我们想要特别感谢赛勒斯·威苏那（Cyrus Vesuna）在协助完成定义平台通用可信控制方面的工作。

参考文献：

[1] Executive Summary": https://www.verizonenterprise.com/resources/reports/rp_DBIR_2018_Report_execsummary_en_xg.pdf; Mandiant, M-Trends 2018: https://www.fireeye.com/content/dam/collateral/en/mtrends-2018.pdf.

[2] Google, "Fleet Management at Scale," November 2017:https://services.google.com/fh/files/misc/fleet_management_at_scale_white_paper.pdf.

[3] https://cloud.google.com/beyondcorp/#researchPapers.

[4] New variants often stretch the common understanding of classes of attacks, so you can't ignore variants completely. For instance, the industry thought we had a good grasp on microarchitecture security up until 2018—see Jann Horn, Project Zero (Google), "Reading Privileged Memory with a Side-Channel," January 3, 2018: https://googleprojectzero.blogspot.com/2018/01/reading-privileged-memory-with-side.html.

[5] Such as Intel's Boot Guard: https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/4th-gen-core-family-mobile-brief.pdf.

[6] Microsoft's Defender Credential Guard: https://docs.microsoft.com/enus/windows/security/identity-protection/credential-guard/credential-guard.

[7] https://github.com/google/grr.

[8] For a description of trust levels and calculation, see B. Osborn, J. McWilliams, B. Beyer, M. Saltonstall, "BeyondCorp: Design to Deployment at Google": https://ai.google/research/pubs/pub44860.

[9] https://osquery.io/.

[10] For more on the tools we use at Google, see "Fleet Management at Scale: How Google Manages a Quarter Million Computers Securely and

Efficiently”: https://ai.google/research/pubs/pub46587.

[11] For more on the trust inference system and the other moving parts of our BeyondCorp model, see B. Osborn, J. McWilliams, B. Beyer, M. S altonstall, “BeyondCorp: Design to Deployment at Google”: https://ai.go ogle/research/pubs/pub44860.

[12] Dogfood: early release of products to employees to get feedback and ca tch bugs before a wider release.

编者信息：

关于 CSA SDP 工作组：

为提高 Software Defined Perimeter（软件定义边界，即 SDP）在中国企业的应用，在中国云安全联盟的支持下 ，CSA 大中华区成立 SDP 工作组。工作组于 2019 年 3 月成立，首批参与单位有：阿里云、腾讯云、京东云、IBM、奇安信、深信服、绿盟科技、Ucloud、顺丰科技、天融信、云深互联、中宇万通、华云数据、三未信安、上元信安、安全狗、易安联、联软科技、上海云盾、缔盟云、信诺时代、齐治科技、德塔博思等三十多家单位。

关于 SDP 工作组更多的介绍，请查看中国云安全联盟官网 https://www.c-csa.cn/ruanjiandingyibianjieSDP.html ，联盟联系邮箱：info@c-csa.cn。

关于 CSA SDP 工作组组长 — 陈本峰：

云深互联创始人 CEO、国家“千人计划”特聘专家、北京市海外高层次人才（“海聚人才”）、海淀区“海英人才”、中关村“十大海归新星”。曾就职于美国微软总部，专注于互联网底层基础技术研究十五年以上，参与了新一代互联网技术标准 HTML5 的国际标准制定，就任国际互联网标准联盟 W3C 中国区 HTML5 布道官。获得过国内外多项发明专利，以及微软最有价值技术专家(MVP)、微软最佳产品贡献奖等荣誉称号。

云深互联是一家云安全公司，取名自中国唐代古诗“只在此山中，云深不知处”。古诗描绘的意境和 SDP 安全理念不谋而合：通过 SDP（软件定义边界）网络隐身技术，使企业数据“隐身”于互联网之中，只对授权用户可见，让黑客无从发起攻击，从而有效保护企业数据资产。让每一家企业的数据可以安全上云并高效地互联互通。

封面封底设计：郭锦霞、陈旭

校验：高婧、高志明

Technical Guide for

Software Defined Perimeter (SDP)

Security Architecture

联系邮箱：info@c-csa.cn

云安全联盟官网：https://www.c-csa.cn

扫描关注联盟公众号