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

目录

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提供了更高级别的安全性，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，或实现对公有云的安全移动访问。

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 协议的实现者。

2 设计目标

SDP 协议的设计目标是为 IPv4 和 IPv6 提供可互操作的安全控制，包括控制器和受 SDP 连接接受方保护的服务的隐藏和访问控制，以及从 SDP 连接发起方到控制器和 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解决方案，来防止对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 利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

最简单的形式是，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。

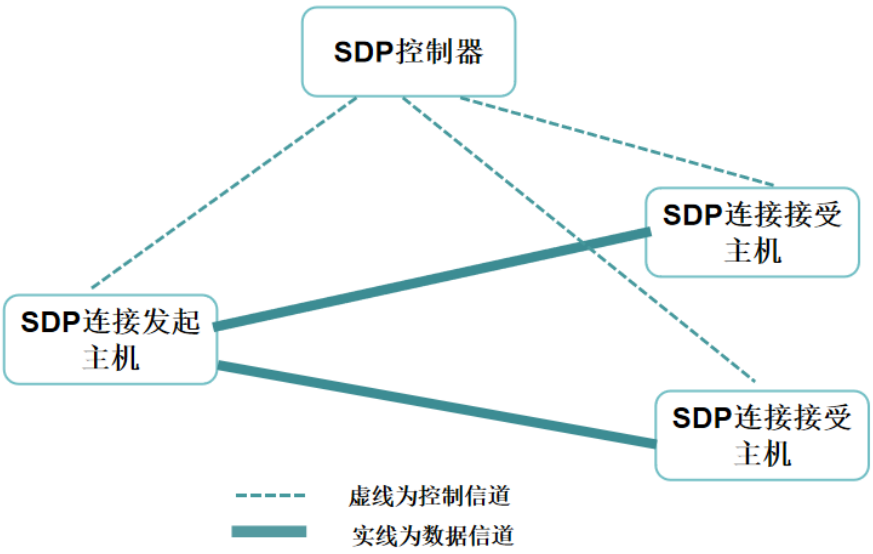

图 1：SDP 架构由两大组件组成：SDP 主机以及 SDP 控制器

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，控制器可以从 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 控制器服务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。

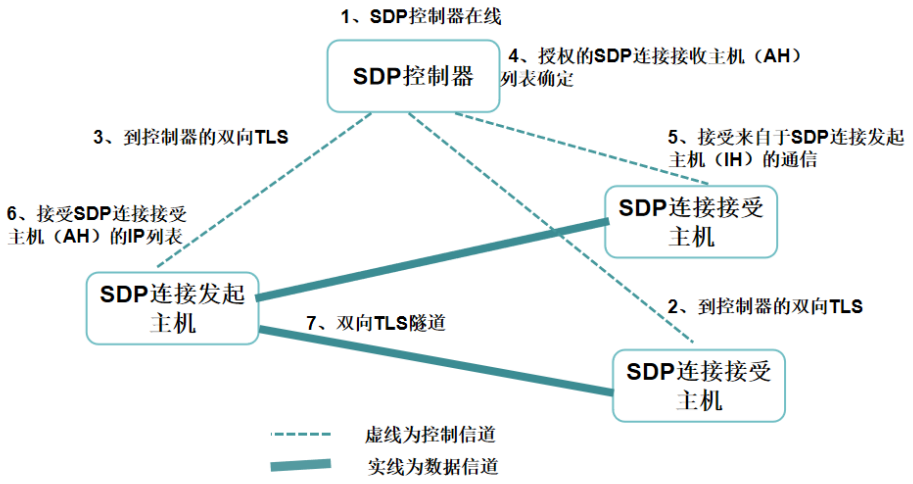

图 2: SDP 架构的工作流展示了控制平面和数据平面的隔离

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，用于标识给定的 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：

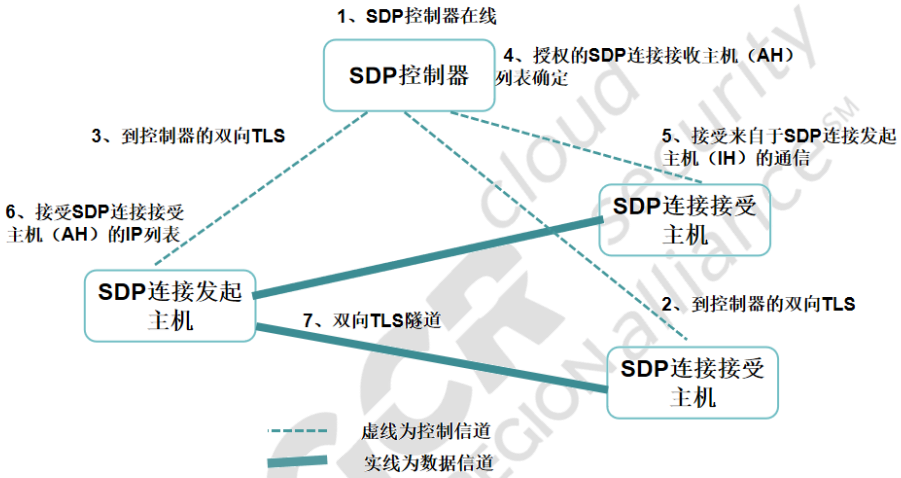

图 3： SDP 架构

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 必须经过其的验证。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，以指示该它是可用的，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

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。

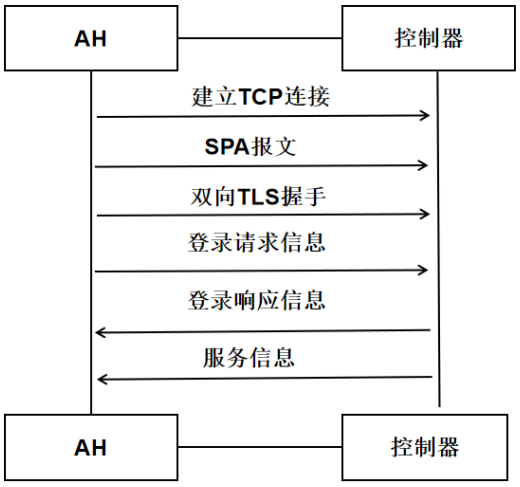

图4. SDP连接接受主机（AH）连接至控制器

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 登陆请求信息

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 登陆响应信息

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” }] }

[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。

5.7.6 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

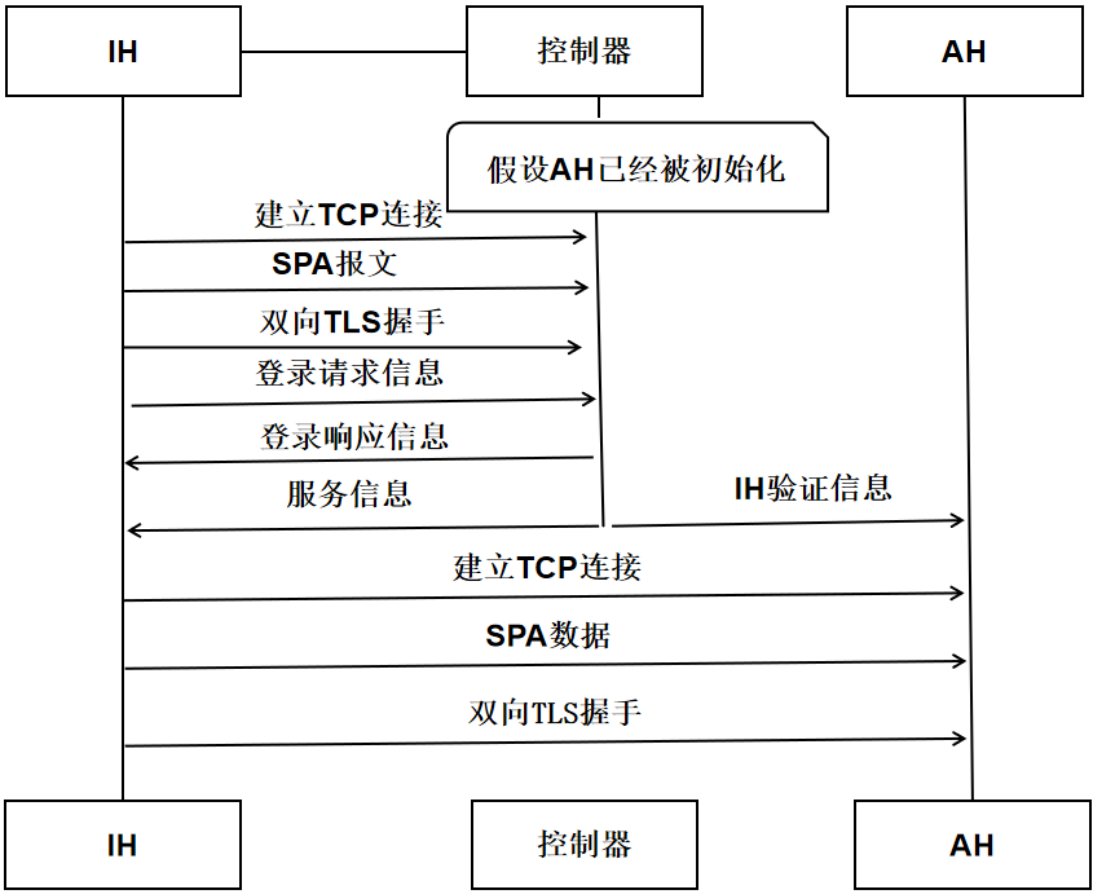

图 5. SDP 连接发起主机（IH）连接至控制器及 SDP 连接接受主机（AH）

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。

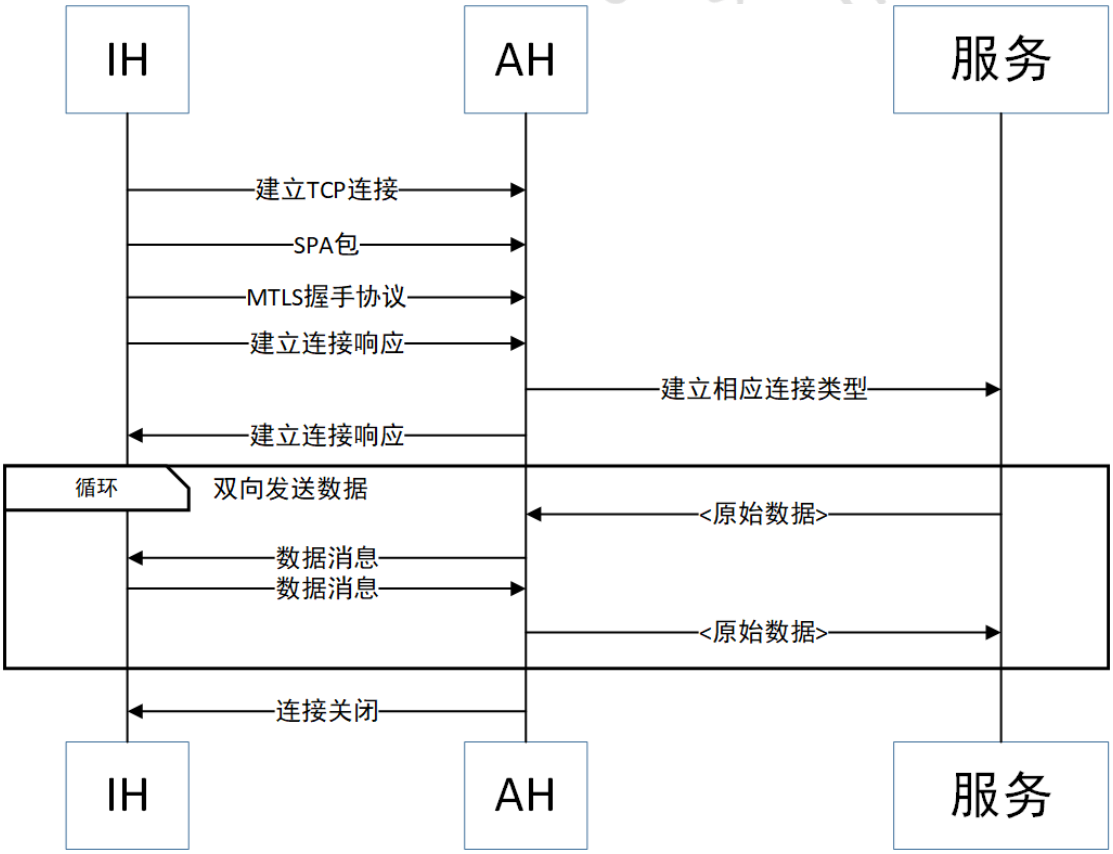

图 5: SDP 连接发起主机（IH）与 SDP 连接接受主机（AH）建立连接并且向服务发送数据

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 列表的方法不在本的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闭 ***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:down* *ops:conn:reconnect*	***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：***记录有多少次连尝试 ***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奔溃宕机，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）。如果有用户从这些的信息中获取一定有价值的内容，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。