E ND B OX：使用客户端可信 执行的可扩展中间盒功能

d一个VID Goltzsch ë*，SIGNE ř ūSC ħ*，曼努埃尔啮ķ ë*，小号'èBASTIEN Vaucher †，尼科W¯eichbrod吨*，Valerio SCHI AV上我†，皮埃尔-路易Aubli Ñ‡，P aolo成本一§，克里斯托夫Fetze ř¶，P ASCAL Felbe ř†，彼得Pietzuc ħ‡和- [R ūdiger Kapitz一个*

*德国不伦瑞克工业大学，goltzsche@ ibr.cs.tu-bs.de ，rrkapitz@ibr.cs.tu-bs.de

†的Un我v緩的Neuc ^ h一个电话，瑞士，pascal.felber@unine.ch

‡英国伦敦帝国理工学院，prp@imperial.ac.uk

§英国Microsoft Research，paolo.costa @ microsoft.com

¶德国德累斯顿工业大学christof.fetzer@tu-dresden.de

***摘要*—许多组织通过将*中间盒*作为其核心网络的一部分集中部署，来增强其托管网络的性能，安全性和功能。尽管这简化了维护，但同时也增加了成本，因为中间盒硬件必须随客户端数量扩展。一个有前途的替代方法是将中间盒功能外包给客户端本身，从而利用其CPU资源。但是，这种方法对关键中间盒功能（如防火墙和入侵检测系统）提出了安全挑战。**

我们描述了E ND B OX，这是一个在网络边缘的客户端计算机上安全地执行中间盒功能的系统。它的设计将虚拟专用网（VPN）与中间盒功能结合在一起，中间盒功能由受信任的执行环境（TEE）进行硬件保护，这是英特尔软件保护扩展（SGX）提供的。通过在SGX区域内维护VPN连接端点，E ND B OX确保中间盒处理所有客户端流量，包括加密的通信。尽管采用了分散模式，但E ND B OX的中间盒功能保持可维护性：它们是集中控制的，可以有效地进行更新。我们证明Ë ND乙OX有两个方案 涉及

(i) 一家大公司；（ii）同时需要保护其网络和连接的客户端的Internet服务提供商。我们 通过将E ND B OX与常见中间盒功能的集中部署（例如负载平衡，入侵检测，防火墙和DDoS防护）进行比较，来评估E ND B OX 。我们展示的是

Ë ND乙OX实现了高达3 。吞吐量提高了8倍， 并与客户端数量呈线性比例增长。

I. 我导论

中间盒是 由 组织（*受管 网络*）管理的大型网络的骨干网的一部分， 并 实现与安全性（例如防火墙和入侵 检测） 以及性能 （例如 缓存 和 负载平衡）相关的各种功能集。同时，他们必须处理不断增长的网络流量[1]和不断增长的基于网络的攻击[2] [3]，同时保持有效的可管理性和成本效益。

尽管基础架构和管理成本很高，但目前的最佳实践是将中间盒作为网络的一部分集中部署[4]。相反，最近的研究建议调查了将中间盒外包到云基础设施的好处[4]，[5]。虽然这减少了维护工作量，反过来，成本，外部部署重要的网络功能，并重定向 敏感 网络 TRAF科幻ç异地 介绍 潜在的安全 风险 ，并 可能 是 非法的。

为了解决这些限制，我们提出了一种新的*分散式* 部署方法，其中将中间盒功能放置在网络边缘的客户端计算机上。因此，中间盒函数可以利用客户端计算机潜在的空闲资源来处理客户端流量。这种方法特别有效，因为客户端流量构成了托管网络中流量的很大一部分[6]，[7]。

对于中间件分散部署模型提出了两个 新 的挑战： （i）它 需要 客户端 向 被 信任的 忠实地执行中间盒的功能，和（ii）网络管理员 必须 保留 控制 过 中间盒 功能 [6]，这 是更 有挑战性 与 分布式 中间件。 尽管这对于服务器等严格管理的机器是可以实现的，但与当今的IT管理实践相反，在IT管理实践中，作为开发 人员 工作的员工保留了 他们计算机上的管理特权。由于缺少补丁，配置不当，用户粗心大意或内部流氓，客户端计算机更容易受到恶意软件的攻击，这些软件试图绕过客户端中间盒 功能。

因此，许多企业会考虑必要的中期dlebox 功能， 例如 作为网络rewalls 或 入侵 检测 太重要的是要委托给客户机不属于系统 的控制权。例如，互联网服务提供商（ISP）通常不愿意在 部署客户 的客户端机器，以防止入侵检测和预防系统（IDPSs） 的恶意软件 从 传播; 公司 将 避免 在员工机器上执行数据泄漏防护（DLP），而是将其安装在集中式网关上。因此到目前为止，研究建议主要考虑了为受信任的服务器计算机部署基于主机的网络 功能[4] – [6]。

我们描述了**E ND B OX**，这 是一个用于在客户端计算机上信任地执行中间盒功能的新系统。设计 的ë ND乙OX是基于上一个虚拟专用网络（VPN），即OpenVPN的[8]，它是用来从一非置信一个访问被管理的网络。我们通过单击软件路由器[9]来支持受信任的中间盒功能的执行，从而增强了VPN客户端的功能。E ND B OX拦截客户端与网络之间的所有流量，以及确保由客户端计算机上执行的中间盒功能对其进行处理。这些功能由现代CPU中可用的受信任硬件功能来保护-E ND B OX使用Intel的Software Guard Extensions（SGX）来在客户端与受管网络进行通信时强制使用它们并保护其 完整性。

2158-3927 / 18 / $ 31.00©2018 IEEE
DOI 10.1109 / DSN.2018.00048

为了 支持广泛流行的加密网络流量 [10]，[11]，E ND B OX 利用其受信任的执行模型进行加密流量分析。与此相反，以人在这方面的中间人（MITM）代理，这可能危及加密会话，ē ND乙OX盾加密密钥在本地上的客户端，从而使解密不削弱整体的安全性。

尽管采用了分散式部署模型，但可以安全，快速，无缝地重新配置E ND B OX执行的中间盒功能。E ND B OX使用用户定义的带内VPN控制 消息， 这些消息在受管网络中的控制服务器和所有E ND B OX客户端之间定期交换。控制 消息 宣布 CON组fi guration 更新 到中间件的功能，强制客户端始终使用 的 最新配置版本。交换控制消息和应用新的中间盒配置的开销很低，因为E ND B OX客户端异步检索新的配置。

本文的其余部分围绕其主要贡献进行了组织：

§II为E ND B OX引入了两种方案，并讨论了在将中间盒功能外包给不受信任的客户时我们要考虑的问题陈述以及威胁模型。

§III描述ë ND乙OX设计，由此说明它是如何固定使用英特尔SGX中间件的功能，保持VPN连接 端点 内 SGX 飞地， 并 安全地处理加密的网络TRAF音响C时不损害端至端 安全;

§IV描述了有关E ND B OX如何与VPN客户端和Click软件路由器集成的实施细节。我们还详细介绍了减少SGX飞地过渡数量，启用用例特定流量保护以及优化E ND B OX 客户端之间的通信的方法。

§V对E ND B OX进行评估，表明它不受许多攻击的影响，例如回滚，重放，拒绝服务（DoS）或密码降级攻击。我们显示E ND B OX与所连接客户端的数量成线性比例，并且达到2 。6 × 至3 。8 × 更高 的吞吐量 相比，以 一个 集中 中间盒 部署。 最后， 我们 展示 的是Ë ND乙OX 有一个 低 性能开销为16％。

II. 牛逼OWARDS SECURE客户端-侧中间件

我们首先介绍受益于安全客户端中间盒（II-A）部署的显式方案。然后，我们描述中间盒如何在当今的托管网络中进行部署，以及为什么最新的解决方案不适合实现上述方案（第II-B节）。最后，我们将英特尔软件防护扩展（SGX）解释为解决方案（ §II-C）的一种启用技术，并讨论了针对不可信客户端（ §II-D）的假定威胁模型。

A. 场景

我们描述了两种由E ND B OX提供的安全客户端中间盒受益的代表性方案。

方案1：企业网络。一家大型公司试图使用中间盒来保护其网络。由于增加 由于 集中式 硬件 中间盒的成本较高， 因此 公司 决定不使用中间盒功能。决定让客户端计算机使用E ND B OX执行中间盒功能。 在与从远程位置工作的雇员线，客户端既可以连接到内部网络或加入的网络 远程地 使用 一个 VPN 客户端。

方案2：ISP网络。拥有数十万客户的Internet服务提供商（ISP）希望通过对网络数据包执行深度数据包检查（DPI）来提供额外的保护。目的是保护客户的客户端计算机以及ISP的网络组件免受恶意软件（如勒索软件）的侵害。然而，它是具有挑战性的提供者，因为（i）它们需要实现这样的系统 访问 加密 TRAF音响Ç有效载荷， 这 是 不可能的 ，而无需创建 安全 漏洞 [12]， 改变 完善的协议[13]，[14]或造成不合理的性能开销[15]；（ii）能够对大量交通进行广泛分析的中间箱购置成本太高[16]。 通过数据计划扩展了ISP的产品组合，该数据计划将E ND B OX 部署到客户的客户端计算机上进行 网络 流量分析。该计划包括折扣，以补偿客户端资源的分配。

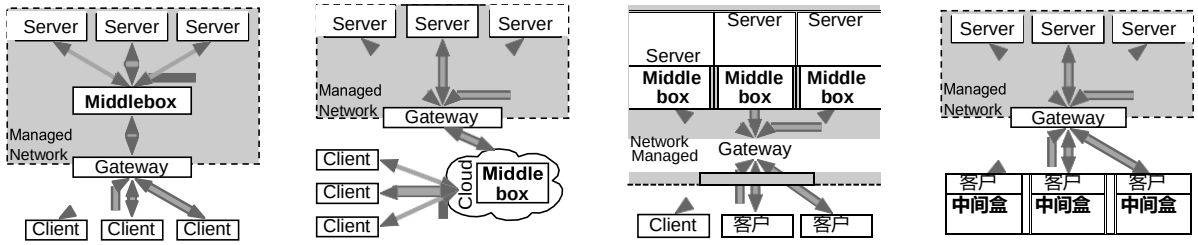
B. 今天的中间盒

中间盒在分析，过滤和操纵网络流量中起着核心作用。典型的示例是防火墙和IDPS，以提高安全性。或缓存和负载平衡器，以获得更好的性能。我们观察到三种基本的中间盒部署方法：（i）作为托管网络一部分的集中式部署；（ii）基于云的部署；（iii）作为终端主机的一部分进行部署。

集中式中间盒部署。这是最COM的部署中管理的网络，其中中间盒被放置到因特网（参见图1a）的服务器和网关之间的周一类型。由于中间件是多种多样的，往往是复杂的，有取代昂贵的专用硬件趋势 电器 通过 基于软件的 解决方案， 运行 在 上面 的商品 硬件 [17]。 随着网络 的不断发展 提供高达100 Gbps容量的流量和企业链接，这需要可扩展的软件解决方案。由于中间盒通常是有状态的（例如，在入侵检测期间），因此执行简单的基于水平数据包的扩展具有挑战性，因为每个网络流都必须分配给单独的中间盒实例 [7]。 集中式 中间盒 部署对于客户端计算机的数量而言 是 不平凡的，而且资源密集，因此成本很高 [4]。

基于云的中间盒部署。与网络功能虚拟化（NFV）的趋势[18]一致，中间盒被外包给第三方[4]运营的公共云 或ISP [19]运营的私有*电信云*（见图1b）。尽管使用公共云从中间盒的管理减轻网络管理员，它带有几个缺点： （i） 在 顺序 来 进行 处理 在 一个 云 基础设施，TRAF音响c必须从而重定向产生额外 等待时间;

(ii) 公共云是外部的，不受信任的基础架构， 即



(a) 集中　(b) 基于云　(c) 服务器端　(d) 客户端

图1：不同的中间盒部署模型：（a）传统的集中式硬件中间盒；（b）将软件中间盒外包到公共或私有云环境中；以及（c，d）最终主机上的软件中间盒

关键功能移到异地；（iii）重定向到云的流量可能会在网络外部被过滤或操纵。中间盒功能在私有电信云中的泛滥可能会导致更少的延迟，并且基础架构可以被认为是更值得信赖的。但是，它仍然需要ISP进行大量投资。总而言之，基于云的中间盒易于管理，但是可能会降低托管网络的可靠性。他们经常因为担心而被丢弃有关安全性，延迟和合法性的信息。

终端主机上的中间盒。最后，中间件功能也可被放置在端主机，无论是 服务器 在 一个 数据 企业环境内中心（参见图1c）[6]或客户端（参见图1d）[20]。这些方法受益于直接在其源或目的地处理网络流量，从而在每个主机处理自己的流量时提高了可伸缩性。然而，完全不可信终端主机还没有考虑，这 是 在 关键的 挑战，介绍了 由 该 场景 中§II-A描述。相比之下，ETTM [20]确实考虑了不可信的终端主机，但是它的方法是有限的：与E ND B相反OX ，ETTM（i）提供较低的安全保证；例如它不能承受物理攻击；（ii）依靠交通被物理交换 机 正 确 转 发，从 而 扩 展 了 整 个 系 统 的 可 信 计 算 基 础（TCB）；（三）建立在一个昂贵的分布式共识算法（参见§VI）。

在本文中，我们的目标是 要 探索 一个 部署 模型，该模型的目标完全不可信的客户端和网络硬件 以便获得以下 好处音响TS： （i） 网络 TRAF音响C可 被 网络过滤的 或 处理在 所述 源 或 目的地;

（ii）处理加密的流量不会产生漏洞，并且很实用；（iii）托管网络中的中央网络设备不必提供中间盒功能；（iv）可以进行扩展部署，因为中间盒功能是由潜在未充分利用的客户端计算机执行的。

C. 英特尔 SGX

近期的Intel CPU以软件保护扩展（SGX）的形式提供了对受信任执行环境（TEE）的支持。SGX使数据和代码的安全，通过所谓的隔间保护 *飞地*。在安全区域内执行的计算与潜在的恶意软件（包括操作系统）隔离开来。

SGX使用特殊的x86指令来创建和管理安全区。飞地 占用隔离的逻辑内存范围 里面 的 地址 空间 的 一个 过程。 SGX 通过校验和和内存加密来保护此范围 的 完整性和机密性。安全区内存存储在称为安全区页面缓存（EPC）的系统保留的内存范围内，该范围已进行透明加密 [21]。

英特尔SGX软件开发工具包（SDK）提供的功能与飞地软件开发，帮助这样的 生命 周期 管理 和 支持 的 功能 调用 跨越的飞地边界。函数调用是穿越从该不可信到可信的环境被称作*eCall的*，而*ocalls*执行相反。

在 除了 对 保护 代码 和 数据， SGX 可以 通过本地或远程认证飞地 *认证*：本地认证提供了用于在同一台机器上的两个飞地来认证方式 的每个 其他 基于 上*测量*，其基本上是飞地的散列。认证依赖于所谓的消息*报道*说可能包含用户自定义网络数据，如用于 将数据绑定到飞地实例。远程认证基于制造过程中融合到CPU中的密钥，并将认证扩展到远程机器[22]。该过程涉及称为*报价的* 数据结构，这些数据结构是由称为报价安全区（QE）的特殊安全区生成的。使用基于Web的英特尔 认证 服务 （IAS）， 报价 可以 是 远程 VERI网络版，以发起 从 一个 真正的 SGX CPU。

SGX的使用涉及一些限制。由于安全区代码必须与不受信任的环境隔离，因此 无法对OS进行系统调用。先前的工作通过将系统支持嵌入到安全区[23]-[25]中，同时增加了TCB的大小来解决了这个问题。当前版本的SGX中的EPC大小限制为每台计算机128 MB。可以通过将EPC页面交换到常规内存来创建更大的安全区，但这会导致性能大幅下降[23]，[26]。而SGX易受到边信道攻击[27] - [29]，研究存在于减轻技术[30]， [31]。

D. 威胁 模型

客户端计算机通常不受信任，因为它们不受 网络所有者的控制。在公司中，并非所有的企业机器都由中央IT部门管理，即，开发人员或管理员通常拥有自己和他人机器的管理权限。在ISP方案的情况下，客户的客户机，应该是 完全 出来 的 控制 中 的 供应商。 另外， 客户端 机器

可能缺少必要的安全补丁或配置不当，因此很容易受到攻击的攻击，这些攻击可能会绕过任何对安全性至关重要的中间盒功能。

我们因此假设该客户机是*不*值得信赖的，并且对手可能在客户端机器的完全控制，包括其运营系统，虚拟机管理程序，和硬件。他们可以把它发送任何TRAF网络c和他们有机会获得入境TRAF网络C，即他们可以丢弃或修改数据包的内容。在此外，他们对操作系统的网络堆栈的完全控制，并可以绕过或修改其任何功能。通过对客户端计算机的物理访问，对手可以读取或写入 到任何内存 地址。

对手还可以对飞地发动DoS攻击，即不启动或进入飞地。但是，我们忽略了对服务器基础架构的分布式拒绝服务（DDoS）攻击：尽管恶意客户端可以串通并将虚假流量发送到服务器，但是可以应用现有的缓解方法[32]。

根据有关托管网络的典型假设，我们认为所有服务器都在中央管理控制之下，因此值得信赖。不允许客户端计算机无限制地访问网络，因为它们可能会受到上述攻击并采取恶意措施。

相反，我们假设用户信任 中间盒功能的提供者（例如公司或ISP）。请注意，此假设对于涉及中间盒的传统方法也是有效的。但是， 通过使用户能够在 运行时 在 SGX 飞地上执行策略，可以削弱或消除这一假设 [33]。

III. d ESIGN

我们描述了E ND B OX，该系统可以安全地在客户端计算机上执行中间盒。根据作为不受信任客户端的一部分的部署方案（请参阅 §II-D），E ND B OX必须满足以下要求：

R1：*灵活性*。E ND B OX应该支持针对广泛使用案例量身定制的中间盒功能的灵活开发。*R2*：*执法*。è ND乙OX应确保客户端和之间的所有网络连接TRAFÇ 所述 管理 网络 是 由中间盒处理 功能。

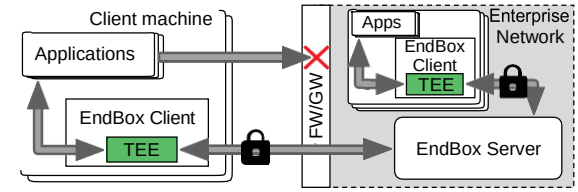
R3：*诚信和隐私*。E ND B OX应保护中间盒功能的完整性和客户业务的私密性。

R4：*可管理性*。尽管被分布式中间件，它应该保持轻松的网络管理员快速，无缝地 管理 中间件 的功能， 例如 作为 更新他们的配置简单。

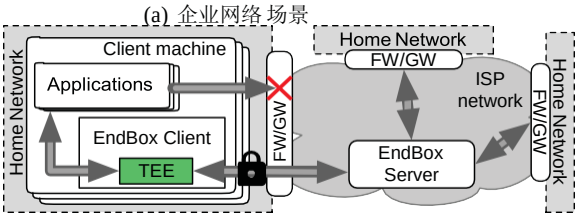
R5：*低开销和良好的可伸缩性*。为了实用，ē ND乙OX应该只介绍一个低性能开销比现有的解决方案和规模线性数量 的 客户， 在 为了 要 支持 FL uctuating 客户 数量和 防止 闲置 中间件 在 了 同一 时间。

A. èND乙OX *在 一言以蔽之*

图2详细说明了在 §II-A中介绍的两种代表性方案中E ND B OX的部署。在这两种情况下，许多E ND B OX *客户端都* 连接到E ND B OX *服务器*。该Ë ND乙OX 客户端允许客户端应用程序 的机器



(a) 企业网络 场景



(b) ISP网络 方案

图2：两种情况（a）和（b）的E ND B OX系统部署，FW / GW为防火墙/网关

访问托管网络。客户端在TEE（我们的原型中为SGX飞地，在本文中以绿色 表示）中执行中间盒功能。TEE保护VPN通信的安全端点并保护 必要的加密密钥。密钥是作为安全引导过程的一部分注入 到 TEE内部的 ，因此 ，计算机的用户和TEE外部的软件都不会被授予对其的访问权限（请参阅 §III-C）。数据包的加密和解密 以及任意处理都在TEE中进行。这样可以实现多种中间盒功能 （*R1*）， 包括 缓存， 恶意软件 检测， 许可控制以及诸如压缩之类的功能，它们都不能对加密的数据包进行操作（请参阅 §III-D）。此外，这使组织能够将用E ND B OX执行的中间盒功能适应 其特定的用 例。

在企业网络场景中（图2a），允许客户端位于网络内部或远程连接（例如，办公室员工）。相反，在ISP网络场景中（图2b），客户端是连接到ISP网络的专用计算机。

在这两种情况下，在访问受管网络时都必须使用E ND B OX，因为E ND B OX 服务器 是唯一的入口点：它仅接受使用正确的E ND B OX 客户端拥有的密钥加密的流量。这确保了所有TRAF网络C被处理Ë ND乙OX 并防止用户绕过 所述 中间盒 的功能性， 因为 漏掉的 TRAF音响c的 任一 堵塞 或 加密， 从而 不 可读 （*R2*）。

此外，êND乙OX 用来保证SGX认证支持 该 （i） 的 飞地 被初始化 与 所述 正确的 代码和数据; 和（ii）的加密和解密的网络数据包 可以 仅 发生 内 的 飞地 （*R3*）。

该Ë ND乙OX 服务器提供了一个 管理 界面 ，使管理员能够部署中间件CON组fi guration变化 （*R4*）， 如 以 发行 更新的 中间件 功能。 更新将分发给所有（已连接和重新连接的）客户端，这些客户端负责获取和应用配置 更改。 后 一个 骗子网络可配置 宽限期 期间， 该

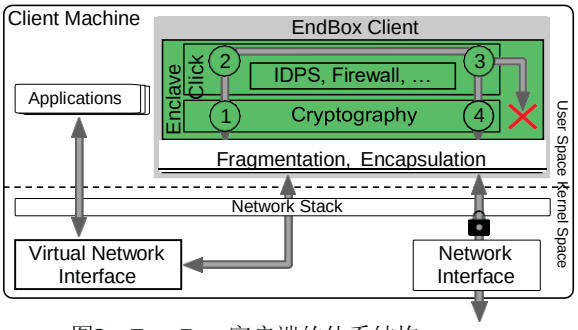


图3：E ND B OX客户端的体系结构

通过仅接受来自E ND B OX的流量来实施更新
具有最新中间盒配置的客户（请参阅§III-E）

E ND B OX旨在降低性能开销，并根据连接的客户端（*R5*）的数量进行扩展。这 是通过（i）减少飞地模式转换的次数来实现的；（ii）通过将中间盒功能移至客户端，从而减轻作为管理 网络一部分的集中式中间盒的负载。

B. ENDBOX *客户端的体系结构*

所述ëND乙OX 客户端 体系结构 示出 在 图3 由 两个部分组成：一个VPN客户端和一组中间设备的功能。该VPN 客户机是 基于 上 的OpenVPN [8] 和被分割; 安全敏感部分（例如加密功能和加密密钥）被移入安全区域，以防止攻击者获得有关机密的知识。对于安全性不重要的其他部分（例如数据包封装和分段）在安全区外部执行。Ë ND乙OX使用Click模块化路由器[9]实现中间盒功能，该路由器可用于实现各种中间盒功能（*R1*）。E ND B OX 通过中间盒功能路由所有流量：在加密出口或解密入口流量之前，OpenVPN将所有数据包交给Click进行 处理。

为了确保所有网络流量都被E ND B OX （*R2*） 拦截，客户端只能通过VPN连接到网络。VPN客户端分四个步骤分别处理每个IP数据包：将数据包复制到安全区域1内后 ， 根据系统配置2由一个或多个中间盒功能对其进行处理。根据特定的功能，可以修改数据包头或有效负载，或者将整个数据包标记为丢弃（例如，由于防火墙或IDPS规则）。在执行了中间盒功能之后，该数据包被接受或拒绝3。最后，对数据包进行签名和加密，然后将其复制到安全区域之外，
它被传递回VPN客户端在运行不可信空间 Ⓧ 的传输OV ER的净W一扫。

从网络到达的每个数据包都以相反的顺序进行处理：首先将其复制到安全区中，在此检查其签名并解密其内容。然后由中间盒函数对其进行处理，接受或丢弃，最后将其复制到安全区域外并传递给应用程序。

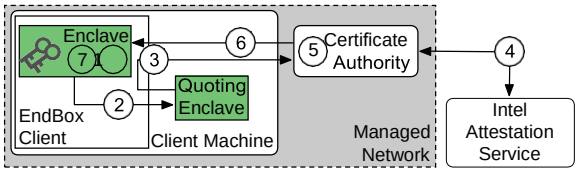


图4：E ND B OX远程认证和密钥管理

C. 证明和密钥 管理

为了实现（安全的我们所希望的水平*R3*），ê ND乙OX 利用描述[22]的英特尔SGX飞地认证设施 在 §II-C 和。 图 4 显示 了为证明飞地的正确性以及保护和签名VPN密钥而执行的 步骤。Ë ND乙OX的密钥管理是基于对一个由网络经营业主CERTI网络凯特颁发机构（CA）。公钥 在系统编译期间，将CA的CA预先部署到安全区二进制文件中，以防止MITM攻击。 在安全区 1中 生成非对称密钥对，私钥永远不会离开安全 区。接下来，VPN客户端创建一个包含上述密钥对的公共密钥的报告，并将其传递 给QE，以获得报价 2 （请参阅§II-C）。这被转发 到CA 3，将其中继 到 的 IAS 和 接收 答复4。如果该答复是肯定的，并且报价中包含已知的度量，则CA签署公钥， 创建 证书5。用安全区的公共密钥加密的证书和对称共享密钥被提供给安全区6 。最后，检查与接收到的CERTI科幻美食后，CA的公共密钥，该飞地持久存储的生成密钥对，以及使用SGX密封部件的CERTI网络美食 7 。客户端现在可以使用该证书连接 到VPN服务器。因此，一个飞地仅需进行一次认证，并且未经认证的客户端由于缺少证书而无法建立连接。对称共享密钥用于 对 解密 CON组fi guration 音响莱 如描述 在 §III-E。

D. 处理加密的网络 流量

尽管当今网络流量的一半已加密[10]，[11]，但许多中间盒功能（例如用于深度数据包检查或缓存）仍需要访问数据包的有效负载，即无法在加密的流量上运行。当Internet工程任务组（IETF）大量讨论是否应降级TLS 1.3中的密钥交换以允许网络监视时，此问题尤其明显。

这里 有 不同的 国家的最先进的 解决方案， 对于 这个 问题：（i）对用户进行MITM攻击的中间盒；（ii）用于一荷兰国际集团MODI音响阳离子到TLS协议以允许中间件以截距TRAF音响Ç [13]，[14]; 和（iii）可搜索的或同态加密方案[15]。这些解决方案解决了问题，但是具有严重的缺点：它们破坏了端到端的安全性，与HTTP公钥固定（HPKP）之类的技术不兼容，不切实际或运行缓慢。因此，E ND B OX 实现了一种解密网络流量的新方法。我们假设客户端应用程序（例如Web浏览器） 已链接到自定义不受信任的TLS库。该库转发所有协商的会话密钥到受信任的点击情况， 运行 里面 的Ë ND乙OX VPN客户端。该键

用于解密特殊Click元素内的数据包。对于我们的原型实现，我们通过添加修改OpenSSL的*单*一调用自定义函数，它向前协商 密钥 通过 的OpenVPN的 管理 界面。 使用这种方法，E ND B OX 可以 对客户端透明地进行流量解密。客户既不需要信任定制CERTI科幻美食权威 也 没有 它 看到 不同 CERTI网络凯茨 比 那些 由访问的服务提供。同样，我们不必更改TLS协议或依靠特殊的加密方案。请注意，键转移到Ë ND乙OX飞地不是为客户端的安全风险：密钥由不可信TLS库中产生 和被 因此 也 存储 在 不可信的 记忆。

我们的 方法 来 分析 加密 TRAF网络Ç 也 适用 与 即将到来的TLS 1.3版本，今天的中间件不能正确处理[35]。此外，我们的解决方案适用于我们的目标方案：在企业网络中，员工在一定程度上信任其雇主，并且应避免使用公司网络中的任何一种来处理私人事务。 在ISP场景中，我们假设客户选择由ISP进行流量分析以提高安全性，即他们知道并 同意。

E. 配置 更新

为了提高可管理性（*R4*），E ND B OX支持在运行时更新Click配置文件。网络管理员可以德音响NE的更新通过指定的重要性*宠限期*的$\tilde{N} \geq 0$秒。在宽限期内，E ND B OX服务器允许新旧配置均处于活动状态。到期后，服务器将阻止来自未应用新配置的客户端的流量。

我们使用来自OpenVPN的带内ping消息将配置更新通知E ND B OX 客户端并执行它们。这些ping消息由VPN客户端和服务器定期发送，以保持连接有效。我们用两个额外的字段扩展了消息格式：最新配置文件的版本号及其宽限期。为了防止恶意客户端发送精心制作的ping消息，将在飞地内部对所有数据包的真实性进行验证。该CA的公共密钥和预共享密钥（请参阅§III-C）用于签名和可选地加密配置文件，例如，以在企业场景中对雇员隐藏IDPS规则。在ISP场景中，未对配置文件进行加密以允许客户检查规则。这些文件存储在托管网络中的受信任服务器上，该服务器可公开访问，以确保客户端在连接之前始终可以获得最新的配置。当网络管理员创建更新的配置文件时，他们进行签名并选择加密，然后将其上传到配置服务器，并指示VPN服务器发送带有新版本号的ping消息。当E ND B OX客户注意到一个 新 的配置 文件 是 可用的， 它 取 了 CON组fi guration网络连接文件，解密它的飞地内，并将其应用于。为了防止客户端重播旧CON组fi guration科幻LES，版本号 的 该 更新 被 纳入 内部 的 更新 本身。

版本号随着每次更新而单调增加。在 整个 更新 过程 被 示出 在 图 5要启动它，

网络管理员上传的CON组fi guration网络文件来 了

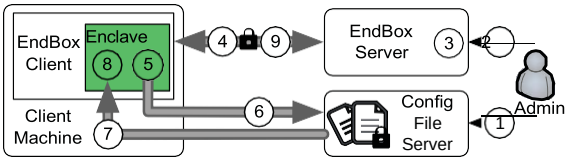


图5：更新E ND B OX中的配置文件

配置服务器1 并在E ND B OX VPN服务器2触发配置更新。VPN服务器启动一个计时器，该计时器到期后将阻止具有旧配置的客户端3 。通过下一次定期ping消息，VPN服务器将新版本号发送给所有客户端 4 。当客户端收到 ping消息时，它会检查是否需要更新 5 。 如果是这样的情况下，获取从CON组fi guration服务器新CON组fi guration 6 - 7 ，对密码进行解密，并取代目前的CON组fi guration 8。最后，客户端发送带有 新 版本 号的ping消息 以 确认 其 成功 更新 9 。

IV. 我MPLEMENTATION

在 实施 的ë ND乙OX 是 基于 上开路VPN V2.4.0 [8]，所述英特尔SGX SDK V1.9 [36]，所述塔洛斯库用于终止SGX飞地[37]内的TLS连接，以及最新版本单击软件路由器[9]。我们使用的OpenVPN作为的基础上进行的é ND乙OX客户端，因为它

（i）是开源的；（ii）具有相对较少的依存关系；（iii）在用户空间中实现；（iv）被广泛使用。这使我们可以将其实施的一部分移植到SGX区域，尤其是考虑到OpenVPN完全在用户空间中执行。TaLoS基于LibreSSL，可作为现有应用程序在SGX飞地中运行的替代产品。

Ë ND乙OX采用 了 英特尔 SGX SDK ，以 德科幻NE *eCall的*和*ocalls*以及处理飞地的生命周期。此外，它使用SDK的信任（但功能限制）C库实现并扩展它与其它功能的使用的OpenVPN和点击。该Ë ND乙OX实施也利用可信时间SDK的支持，以实现TRAF网络Ç整形（见§VB）。此外，SDK提供了一种*仿真模式*，可以执行SGX不支持安全性但运行时 行为类似的不受支持的硬件上的应用程序。

E ND B OX依靠Click来实现中间盒功能。为了配置Click，将所谓的*元素* 相互连接。元素可以从其他元素获取数据包或将数据包转发到其他元素，然后处理数据包。我们选择“点击”是因为

（i）被广泛使用；（ii）具有许多实现各种中间盒功能的现有元素；（iii）提供配置热交换机制；（iv）易于扩展。E ND B OX 使用Click的配置热交换机制来有效地更新中间盒配置。它采用与点击运到中间盒实现功能的元件，并延伸 点击 通过添加 定制 元素 为 一个 IDPS 函数，解密应用程序级TRAF音响c和执行TRAF音响Ç成形 用 一个 可信的 时间 源 提供 由SGX。

单击和OpenVPN的更改。 è ND乙OX需要小的改动 ，以 单击：（i） 所述 ToDevice 元件 被 MODI音响编 到 信号的OpenVPN当 一个 数据包 被 接受 或 拒绝。 有 有

还对Click核心进行了更改：（ii） 由于在飞地内部不支持信号，因此我们禁用了状态清理的信号处理和用于与特定元素通信的控制套接字的功能；（iii）我们使热交换机制适合存储在内存中的配置文件。OpenVPN与TaLoS库链接，这导致所有加密操作都在安全区域内执行。此外，我们将Click编译为一个库，并将其链接到安全区代码以实现快速 交互。

TCB大小。飞地中的代码行总数（LOC）是TCB大小的重要因素。E ND B OX 的受信任部分包括320 kLOC：用于TaLoS的219 kLOC，用于Click的80 kLOC，用于SGX SDK的20 kLOC和用于OpenVPN敏感部分的1 kLOC。TaLoS的代码行数应视为上限：TaLoS提供与LibreSSL相同的API和功能，而E ND B OX仅使用一小部分。

A. 最佳化

E ND B OX进行了一些优化以提高其性能和安全性：（i）减少飞地过渡的次数；（ii）启用用例特定的流量保护；和 (iii) 优化客户与客户之间的沟通。这些optimisa-蒸发散 都 详细 在 了 下面 ，并 评估 在§VG。

飞地 过渡。 该 性能 的 SGX 飞地 是neg- atively通过可信和不可信的代码之间的转换影响。先前的工作[23]，[26]已经表明，飞地过渡的成本比系统调用的成本高。为了降低该成本，êND乙OX OpenVPN的加密逻辑进入飞地的位移部分来减少每个处理的分组飞地转变的数目：èND乙OX仅执行一个*紧急呼叫*每发送或接收的分组。如§VG中所述，此优化极大地提高了整体吞吐量 的 E ND B OX 。

场景特定的流量保护。取决于SCE-其中纳里尼奥ëND乙OX时，较弱的TRAF音响C防护可以 施加。 在 该ISP 的情况下， AES-128-CBC 数据包 加密是可选的，因为信任关系是从企业不同的 使用 情况： 用户 决定 以 让 该ISP 申请ËND乙OX ， 因此事实上，TRAF网络C通过点击路由不必须通过加密来强制执行。但是，事实是通过Click分析*出口*流量需要由ISP通过应用完整性保护来确保。这种优化只针对ISP的情况下，提高了整体吞吐量ËND乙OX，中所述§VG。

客户端到客户端的通信。在客户端到客户端连接的情况下，我们的方法将导致数据包被处理多次，每个客户端一次。对于大多数用例（例如IDPS），这是不合理的。因此，E ND B OX客户端在通过Click处理后会标记传出数据包，从而使其他E ND B OX客户端可以绕过Click。我们实施了FL通过设置质量agging机构 的 服务 质量（QoS） 字节 中 的 IP 报头 ， 以 将 0xEB 。 在 为了 防止 外部 攻击者 从 发送 IP 包 含此字节的èND乙OX 如果服务器中删除所述QoS字节 设置为将0xEB 。最后，所有的数据包都完整性保护的OpenVPN的， FL agged 包 不能被 伪造。 此 优化 如§VG中所述，它可以针对企业场景，但也可以应用于ISP网络，并改善E ND B OX客户端之间的延迟。

B. 安全区域 接口

飞地接口ëND乙OX由90个调用：70个*eCall的*和20个*ocalls*。大多数*ecall*仅在OpenVPN和Click初始化期间被调用。E ND B OX 仅*定义*在正常操作期间执行的 4个*ecall*：（i）数据包加密和解密；（ii）消息认证码（MAC）的生成和验证。而（i） 是 由正常TRAF音响C，（II）触发被用于OpenVPN的完整性保护 控制 信道。 随着 该 异常 的 的éND乙OX - SPECI 科幻çD恩和解密，并点击初始化*eCall的*，所有*eCall的*匹配TALOS / LibreSSL库调用，其执行安全检查。所述*ocalls*执行不同的任务，其中管理不可信存储器和访问（加密）CON组fi guration连接LES。请注意，可以通过使用安全区内 配置 文件和无 出口安全 区 服务来忽略它们 [38]。为了 确保接口的安全性，我们仔细检查了所有*ecall*和*ocall*，并通过对输入（值 返回） 值 的健全性检查来增强它们*eCall的*（相应的*ocalls*），和结合的检查指针或者传递到*eCall的*或返回的从*ocalls*到确保它们指向飞地内存。

V. Ë估价

通过讨论对E ND B OX的不同攻击并执行不同的测量，我们评估E ND B OX的安全性和性能。我们的结果表明：（i）E ND B OX可抵御各种攻击（ §VA）；（ii）其仅影响网络延迟在一个最小的方式（ §VC）；（iii）它诱导的16％可接受的最好的情况下的性能开销（ §VD）；

(iv) 它与客户数量成线性比例；（v）客户可以达到2 。6 × – 3 。吞吐量比传统的集中式中间盒（ §VE）高8倍 ；（vi）我们的运行时重新配置机制比原始Click实现（ §VF）的延迟降低了30％；而最后，（七）我们的优化 中描述 §IV-A实际上提高éND乙OX的对延迟或吞吐量的影响（ §VG）。

A. 安全 评估

在对我们的威胁模型进行了详尽的评估之后，我们讨论了针对E ND B OX的典型攻击，并说明了如何防御这些攻击或为什么它们不适用。

绕过中间盒功能。恶意客户端可能会尝试不使用E ND B OX来访问网络。我们 假设网络受到用于VPN使用的静态防火墙限制流量的保护：如果没有适当配置的E ND B OX客户端建立有效的VPN连接，攻击者就不可能发送会绕过中间盒功能的有效流量。ËND乙OX 。相反，流量将被防火墙丢弃。对于传入的流量，*间接*迫使客户通过E ND路由其流量乙OX客户，如果他们想访问 的 加密的 有效载荷。 èND乙OX确保的真实性 的连接使用远程认证（ §III-C）。

使用旧的或无效的中间盒配置。攻击者可能回滚配置更新，或使用未经授权的配置。一次可调整的宽限期以进行更新

已通过，服务器仅接受使用当前有效配置的E ND B OX客户端，如§III-E中所述。所述ë ND乙OX客户端和服务器周期性地交换的ping含有CON组fi guration信息，以防止客户端的消息，从使用陈旧配置简单。

重播流量。如果恶意客户端重播流量，例如为了在没有真正飞地的情况下建立连接，则由于OpenVPN实施了数据包重播保护，因此E ND B OX服务器会检测到此情况。

拒绝服务攻击。恶意的客户端可以防止恩恒山无法启动或正在输入，作为 飞地 生命 周期是由不可信代码管理。但是，这将导致客户端无法与网络通信。 在 另一方面， 一个 拒绝服务 攻击 上 的Ë ND乙OX服务器将具有作为一个传统的集中式部署中间盒相同的效果，因此，攻击可以通过使用传统技术减轻 [32]。

降级 攻击。攻击者可能试图强迫使用 较弱的TLS版本或密码。但是， OpenVPN实施服务器端检查，以确保使用最低的TLS版本。在客户端，相应的检查在连接建立期间在隔离区内进行，因此无法 规避。

界面攻击。客户端可能会像Iago攻击一样，通过在安全区域接口上操纵参数来尝试闯入安全区域[39]。为了减轻这种攻击，每个*ecall*和*ocall*都增加了对输入参数和返回值的检查（请参阅§IV-B）。此外，ê ND乙OX 公开了一个 有限的 接口 与 一个 限制 攻击 表面。

中间盒的故障。如果中间盒发生故障，则仅会影响运行该中间盒的客户端；其他客户端和受管网络不受影响。这不同于该行为 的 传统的 集中式 中间件 的调校 在其中一个出现故障会影响许多客户甚至整个网络。相反， 管理所有VPN连接的E ND B OX服务器的故障等同于传统的集中式中间盒的故障，从而导致网络中断。

B. 实验设置和用 例

我们评估了E ND B OX在 由两台机器组成的七台机器上的性能。类阿由音响五个机，配有32 GB的存储器SGX能力4核至强V5的CPU，而 类 乙 是 2台 机器具有非SGX 4核Xeon V2 CPU和16 GB的存储器。所有机器都配置了超线程，并且 每台 机器通过两个10 Gbps网络接口连接到10 Gbps交换机。网络链路的 最大 传输 单位 （MTU）配置为9000字节。我们 使用进行吞吐量测量iperf的，而对于延迟测量 我们 依赖 于 ICMP ping命令。 纵观 本 节中，我们评估多种调校，包括这些反复出现的： （一）*香草OpenVPN的*，一个unmodi网络编OpenVPN的V2.4.0;

（ii）*OpenVPN + Click*，与OpenVPN版本相同，但流量由 服务器端 Click 实例处理； （iii）ê ND乙OX*在 模拟 模式* 到显示 的 开销 的 划分 的 VPN客户端 （iv）*硬件模式下的*E ND B OX以显示使用SGX指令的开销。在本节中，我们报告10次连续运行的平均值。如果报告的误差可忽略不计，则结果的方差将被忽略。

在下文中，我们描述了为评估实现的五个中间盒函数。它们基于标准或自定义Click元素。

转发（NOP ）。我们考虑的第一个中间盒功能为我们的测量提供了基准。它转发数据包，而无需 访问 或 修改 任何 标头 或 有效负载。

负载平衡（LB ）。该RoundRobinSwitch 点击元素使我们能够跨越平衡IP数据包或TCP流人几个机器，从而平衡 负载。

IP防火墙（FW ）。的Fi防火墙访问数据包报头和CON组trols TRAF音响C基上的一组规则。我们使用IPFilter Click元素，而没有任何代码修改。对于我们的评价 ，我们 用 一 组 的16条 规则 是 根本 不 匹配的 任何 数据包。

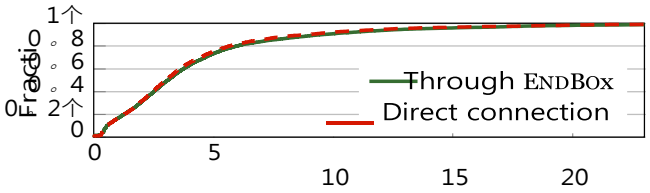
入侵 检测 和 防御 系统 （IDPS ）。一个 IDPS监控网络TRAF网络下未经授权的访问和策略违规。我们支持Snort [40]规则集，并使用[42]中的库执行其字符串匹配算法[41]。IDPS被实现为名为 IDSMatcher的自定义Click元素。为了进行评估，我们使用了Snort社区规则集的377条规则的子集。同样，这些规则与为我们的 评估而生成的数据包不匹配。

DDoS预防（DDoS ）。分布式拒绝服务攻击可以 通常 被 减轻 通过 节流 或 丢弃 数据包 发生反复或者如果检测到源地址spoo音响纳克。我们通过使用自定义的Click元素IDSMatcher 和TrustedSplitter对相同的数据包进行速率限制来实现该中间盒功能。后者允许的TRAF科幻C到成形一个在受信任的方式给 定 带 宽 ： 减 少 昂 贵 的 电 话 获 得 信 任 的 时 候 ，TrustedSplitter在处理了一定数量的数据包之后，通过发出呼叫来对时间戳进行采样。对于我们的测量，此数字设置为500，000。对于*OpenVPN + Click*，我们使用一个类似的Click元素，称为UntrustedSplitter ，它使用系统调用获取时间戳。该用例非常适合ISP场景，因为它使提供商可以直接在客户端检测恶意软件或僵尸网络。

C. 延迟

在下文中，我们评估的延迟影响Ë ND乙OX，因为 这 有 一个显着 的FL uence 的 用户 体验。我们使用转发中间盒功能（ NOP ），并使用A类机器执行本地实验。对于基于云的测量，我们依赖于Amazon Web Services （AWS）弹性计算云（EC2），并在不同区域中使用具有1个虚拟CPU和3.75 GB RAM的m3.medium 实例。

HTTP请求处理。è ND乙OX的对延迟的影响可以在图6中，图表的累积分布函数（CDF），用于HTTP的用Alexa [43]提供1000个流行网站页面加载时间被观察到。结果表明，所需的时间来加载这些网站非常相似，当 使用



页面加载时间[s]

图6：使用和不使用E ND B OX的Alexa前1,000个站点的HTTP页面加载时间的CDF

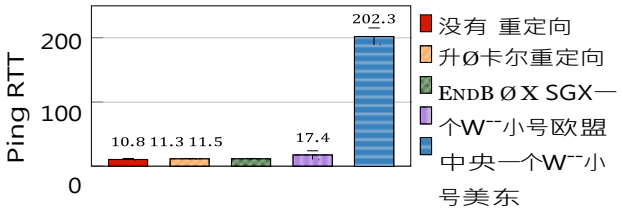


图7：不同重定向方法的平均ping RTT

E ND B OX或直接连接，因此E ND B OX的等待时间开销可以忽略不计。

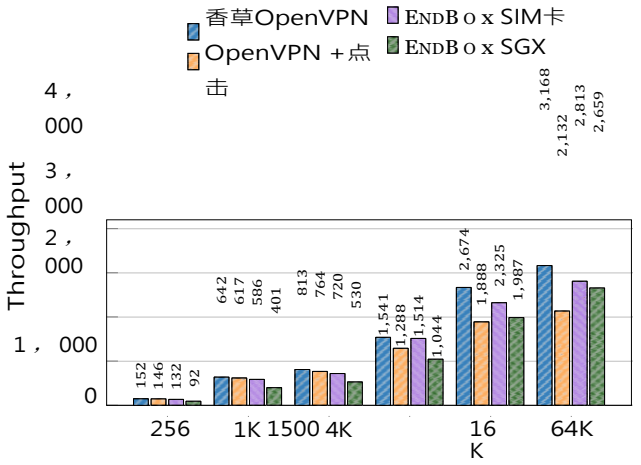
交通重定向。通过进一步探索E ND B OX的影响

关于延迟，我们想证明将中间盒功能引入云环境比仅失去控制和信任有更多的缺点。受[4]的启发，我们创建 了一套在AWS EC2中执行的软件中间盒，并测量了到固定位置的ping往返时间（RTT）。图7示出了平均平。RTT为不同的重定向的方法： （i） 没有 重定向 与 没有 中间盒 或 VPN; （ii）使用*OpenVPN* + *Click*通过VPN和服务器端中间盒进行本地重定向；（iii）重定向槽E ND B OX ；和 （iv）使用*OpenVPN* + *Click*通过部署在不同AWS区域中EC2实例上的中间盒进行 重定向。结果表明，根据云提供商选择的位置，延迟开销在61％到1773％之间，而E ND B OX的延迟开销仅为 6％。

加密流量的处理。如III-D所述，E ND B OX能够透明地解密TLS流量。我们通过让HTTPS客户端从 Web服务器获取大小不同的静态网页来衡量此功能的开销。该客户端正在使用以下一种配置：

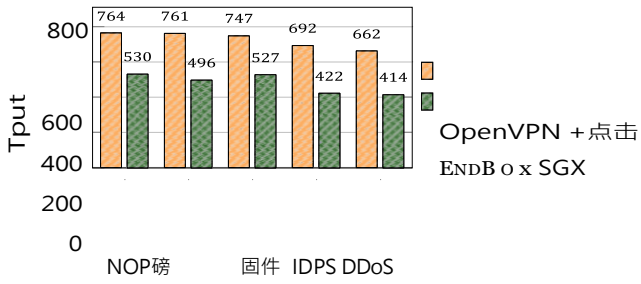
（i）带有自定义OpenSSL的E ND B OX ，并在Click内进行流量解密；（ii）具有自定义OpenSSL的E ND B OX ，但不进行流量解密；或（iii）具有系统OpenSSL且没有流量解密的E ND B OX 。我们测量HTTPS GET 请求的延迟，并在表I中报告结果。它们表明，我们的自定义OpenSSL和流量解密所带来的开销不到8％。间接费用的两个来源是E ND B OX的 将密钥自定义OpenSSL转发到安全区，并进行实际 解密。

E ND B OX OpenSSL香草OpenSSL响应。尺寸w / dec w / o dec w / o dec



封包大小[位元组]

图8：数据包大小为256字节至64 KB时，不同设置的平均最大吞吐量



4 KB	1.08 毫秒	1.04 毫秒	1.00 毫秒
16 KB	1.34 毫秒	1.29 毫秒	1.26 毫秒
32 KB	1.78 毫秒	1.75 毫秒	1.70 毫秒

表I：不同响应大小和配置的HTTPS GET请求延迟

图9：平均最大吞吐量的NOP ，FW ，LB ，IDPS和DDoS的使用案例的OpenVPN +点击和Ë ND乙OX与一个1500点数据包大小 的字节

D. 吞吐量

除了网络延迟外，吞吐量性能也是
影响用户体验的重要参数。所有这些测试 都 进行 上 牛逼w ^ō 类 ⓧ个 机器。

数据包大小。在本实验中，我们测量从256字节到64 kB的各种数据包大小，在不同配置下达到的最大吞吐量。我们比较了四个设置：（i）*香草OpenVPN*，（ii）*OpenVPN + Click*：相同的OpenVPN版本，带有附加的服务器端Click 实例；（iii）*在模拟模式下的*E ND B OX ；（iv）*硬件模式下的*E ND B OX 。结果如图8所示。正如预期的那样，所有配置的吞吐量都随着有效负载大小的增加而增加。此外，我们看到E ND B OX具有可接受的性能开销：在仿真模式下，对于E ND B OX而言，其开销在2％到13％之间变化。使用实际的SGX指令（硬件模式）会增加开销，导致小数据包的最坏情况开销为39％，但是 大型数据包的最佳情况开销仅为16％。这是 由于以下事实：较大的数据包允许 在较少的区域过渡情况下获得更高的吞吐量。我们还看到，服务器端Click实例的平均性能损失为26％。值的范围在5％到29％之间，具体取决于数据包的大小。最后，我们观察到，对于大数据包，服务器端的点击情况下实现了吞吐量比香草近三分之一较低的OpenVPN 由于 到 的 点击 实例的 数据包 抓取。

中间盒功能。图9显示了与E ND B OX相比，传统的具有VPN的中间盒设置所实现的平均最大吞吐量。我们使用一台客户端计算机和1500字节的中等数据包大小来评估 §VB中提供的所有中间盒功能。

以NOP 为基准，我们首先观察到Click配置对OpenVPN + Click的影响很小：在最坏的情况下，对于DDoS预防用例，吞吐量下降了13%，从764 Mbps下降到662 Mbps。其次，对于用例NOP ，LB和FW ， E ND B OX产生约30%的开销。计算密集型用例IDPS和DDoS的开销为39%。请注意，此开销是降低对于较大的数据包，如所示在图8中。

概括。结果表明，对于NOP用例中的大数据包，E ND B OX的吞吐量开销仅为16% 。对于中等大小的数据包，轻型中间盒功能的开销为30%，而特定的重型用例的开销为39%。正如预期的那样，我们观察到E ND B OX的吞吐量随数据包的大小而增加。此外，E ND B OX对HTTP页面加载时间的延迟没有任何用户可感知的影响。其结果是，从一个性能的角度来看，ē ND B OX是现有中间盒部署的可行替代方案。

E. 可扩展性

在 评估了E ND B OX的面向用户的属性（如延迟和吞吐量）之后，我们评估了E ND B OX的可伸缩性，这对于其运营商而言很重要。因此，我们在服务器端测量吞吐量和CPU使用率。吞吐量是在OpenVPN服务器设置的所有虚拟接口上聚合的，每个客户端一个。CPU使用率适用于所有内核，即100%表示所有内核都已被充分利用。可伸缩性测量使用五台A类机器来执行多个E ND B OX 客户端和两台B类机器，每台机器都运行E ND B OX服务器或iperf 服务器。为了进行测量，我们比较了四个设置：（i）不使用中间盒功能作为基准的*香草OpenVPN*；（ii）*硬件模式下的*E ND B OX；（iii）*香草单击*服务器端，不进行加密；（iv）*OpenVPN + Click*：附加到OpenVPN服务器的多个服务器端普通Click实例。在这些实验中，每个客户端产生200 Mbps的工作负载。对于（i），（iii）和（iv），我们每个客户端使用一个OpenVPN服务器实例，作为OpenVPN 它 不 支持 多线程。

首先，我们使用转发器作为中间盒功能（NOP ）评估可伸缩性。图10a中的结果表明，在几乎相同的CPU使用率下，香草OpenVPN和E ND B OX达到了6.5 Gbps的相同吞吐量。这表明 客户端执行中间盒对服务器端的吞吐量或CPU使用率没有影响。对于OpenVPN + Click，瓶颈在于CPU，它比E ND B OX早已得到充分利用，因为点击需要大量的周期。相反，无法处理更多数据包的Click进程将普通Click的吞吐量限制为5.5 Gbps。最后，我们的测量报告甚至 更低 吞吐量 为OpenVPN的+点击 的 2.5 Gbps的， 这与越来越多的客户的不断减小，OpenVPN的+点击 被 限制 通过 该 服务器的 处理器。

用例评估。我们进行同样的测量为我们的网络已经 在展示用例 §VB。在图10B中，我们使用OpenVPN的结果+点击和Ë ND乙OX从先前的测量 为 基准 ，并 展示 如何Ë ND乙OX秤用

	香草相 Click E ND B OX	
取	-	0.86 毫秒
解密	-	0.07 毫秒
热插拔	2.4 毫秒	0.74 毫秒
总计	2.4 毫秒	1.67 毫秒

表II：香草Click和E ND B OX配置更新不同阶段的时间安排

应用不同中间盒配置时的客户端数量。当网络流量加密完全利用VPN服务器（在我们的计算机上有40个客户端）使用时，它将成为E ND B OX的瓶颈：我们观察到所有用例的最大吞吐量为6.5 Gbps。由于 的中间件功能的服务器端执行，OpenVPN的+点击达到此 限制 早些时候 在 30级 的客户 有一个 最大 吞吐量 的
2.5 Gbps FW 和LB 用例。计算密集型IDPS
而DDoS中间盒功能只能达到1.7 Gbps。

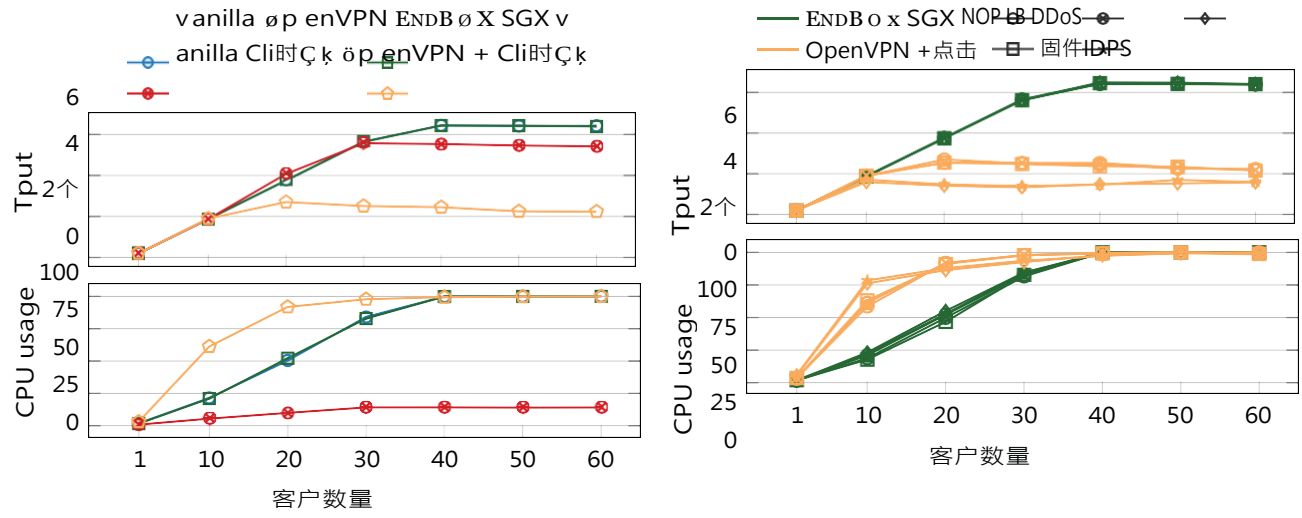
我们的评估表明，E ND B OX 与 客户数量呈线性比例关系。另外，对于60个客户机，êND乙OX达到 一个 2 。6 × 更高的吞吐量 跨越 所有 用 例，并
3 。8 × 用于 IDPS引发的计算密集型工作负载
和 DDoS 。 这 是 不是 一个 普遍 限制 的Ë ND乙OX -它是由于我们的评估设置以及网络数据包上模式匹配的计算 密集型性质，导致中央中间盒更快地过载。因此，我们表明，Ë ND乙OX为CPU密集型的中间设备进行特别好 功能。

概括。结果表明，E ND B OX与客户端数量成线性比例，直到VPN服务器被完全利用为止。他们还表明，通过在客户端执行 中间盒 功能 ，E ND B OX可以实现2 。6 × 至3 。8 ×吞吐量比集中部署中间件更高，取决于上
用例。

F. 配置 费用

集中部署中间盒的一个优势是简单的配置更新机制。对于E ND B OX而言，这更具挑战性，因为中间盒分布在不受信任的客户端计算机上。因此，E ND B OX实现了一种机制，以安全的方式在所有客户端中间盒中应用配置更新，并使管理员能够验证是否应用了正确的配置，如 §III-E中所述。

更新操作的细目分类。表II显示了由香草Click和E ND B OX执行的配置更新的不同阶段。我们使用大小分别为42和59字节的最小配置文件。由于香草点击并不需要获取和解密CON组fi guration网络连接文件，唯一的 操作 是 热交换进行 的CON组fi guration， 这 需要
2.4毫秒 的 平均水平。 在 此相反，ē ND乙OX 花费 的平均提取新配置为0.86毫秒，解密新配置为0.07毫秒。然而，这两个操作不要 在FL uence 的 TRAF网络Ç 网络滤波 的Ë ND乙OX和都在后台执行。最后，热交换配置花费0.74 ms。因此，与香草点击相比，E ND B OX仅需要30%的时间进行实际配置。这是由于到的事实，即香草按需求来设置了网络连接文件



（a）适用于不同中间盒部署的NOP 用例（b）用于OpenVPN + Click和E ND B OX的五个中间盒功能

图10：（a）不同中间盒部署和（b）具体用例的服务器端聚合吞吐量和CPU使用率

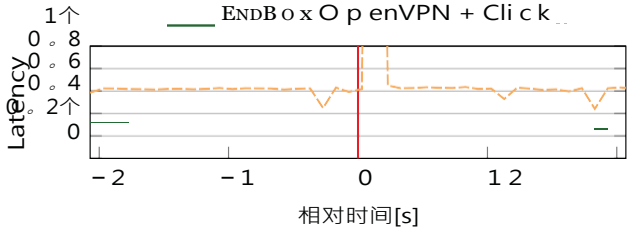


图 11：影响 的 CON组fi guration 更新 上 平 的等待时间显示为 FW 使用 的情况下， 时间 的 侦察组fi guration 在 0 秒

描述 了 该 ToDevice 和 FromDevice 元素， 这是没有必要的é ND乙OX 因为OpenVPN的把这个任务护理 更早。

更新对延迟的影响。另外，我们比较了配置更新对两个设置的 延迟的影响：E ND B OX和OpenVPN + Click，两者均应用了防火墙用例。在我们的实验中，单个客户端以每秒10个请求的速率发送定期ping，然后测量往返时间。如图11所示，我们注意到在重新配置期间，OpenVPN + Click和E ND B OX都丢失了一个ping数据包。这说明开销的*分布式*相比，*本地*侦察组fi guration是可以忽略不计 ，如果实施 正确地。

G. 评价 的优化

最后，我们评估了优化中所述的影响
§IV-A要么上吞吐量或延迟。减少的数目每一个可以通过相当高的数据包的结果飞地跃迁的342%，而避免设置从分组加密在ISP场景导致较高的11%的吞吐量。相比之下，优化客户端到客户端通信对吞吐量没有影响，但降低高达13%的客户之间的延迟对中的IDS使用情况。

VI. [R心花怒放w ^ ORK

我们不是第一个倡导将中间盒移动到最终主机的好处的人，例如[6]，[7]，[20]，[44]。然而， 在 茫茫 大部分 的 这些 解决方案 假设 信任 端 主机 因此，它们不适合本文所针对的客户端部署，因为用户对计算机具有完全的物理访问权限，因此无法被信任。

ETTM [20]是一个值得注意的例外，它依赖于受信任的平台模块（TPM）。这种方法是不灵活的，因为它仅在引导时支持证明，并且在执行过程中缺乏完整性检查。最重要的是，它确实 不能像E ND B OX那样 通过物理访问计算机来防御恶意用户。此外，ETTM是不切实际的，因为它需要对整个管理程序是部分的TCB中; 和物理网络硬件以正确转发流量。虽然对于企业设置来说，可以假定网络硬件是受信任的，但是在ISP方案中这是不可行的。最后，ETTM的设计遵循一种分布式方法，该方法不涉及诸如E ND B OX的 受信任配置服务器。做。因此，ETTM将Paxos [45]应用于共识，但是Paxos不能很好地扩展[46]，会导致高延迟，并且在涉及不稳定 连接的移动节点时不适用，正如我们在企业场景中所讨论的那样。其他提议，例如Eden [6]，则依靠最终主机上的专用硬件来实现中间盒功能。尽管这些解决方案可以实现比E ND B OX更高的性能，但是它们的硬件超过了当今笔记本电脑和普通台式机的规格，因此不能满足我们的 方案要求。

中间盒功能可以完全移动到 所述 云[4]，[5]，[47]。该解决方案避免了用户遭受物理攻击的风险，并可以提供出色的可伸缩性。这些 好处科幻TS，但是，来在该成本的增加费用和更高的延迟，由于到TRAF网络Ç重定向（见 §VC）。此外，外包业务处理会带来安全风险以及隐私和法律问题。

已经提出了在SGX飞地内部执行中间盒功能的方法[48]-[51]。与E ND B OX相反，这些系统并非旨在部署在客户端上。相反，它们在云中执行整个中间盒或特定功能，以 确保网络流量的 完整性 和 机密 性。

如在§III-d详述，E ND乙OX是能够对加密TRAF音响Ç执行中间件功能。以下四个建议也针对此问题。BlindBox [15]提出了一种加密方案，可以对加密的流量执行一组有限的计算，但是其成本要比传统的同态加密低得多。在mcTLS [13]和mbTLS [14]中，对数据包进行加密的方式使得需要访问的中间盒可以对其进行解密。SGX-Box [52]在集中式中间盒上利用SGX在加密的网络流量上启用DPI。类似于E ND B OX，TLS会话密钥与安全区域安全共享。

VII. ç ONCLUSION

在本文中，我们介绍了E ND B OX，这是一个可扩展的系统，能够在不受信任的客户端计算机上安全地部署和执行中间盒功能。对于典型的 中间盒

功能，它与客户数量呈线性比例关系，从而实现2 。6 × 至3 。与传统的托管网络核心部署相比，吞吐量提高了8倍。尽管是分布式的，但对基于E ND B OX的 中间盒服务的配置更改仍是集中 控制 和执行的。

最后，由于加密的应用程序流量位于客户端，因此可以使用E ND B OX对其进行安全有效地解密和过滤。

一个CKNOWLEDGMENTS

作者感谢匿名审稿人的宝贵反馈。根据赠款协议645011（SERECA）和690111（SecureCloud），这项工作已获得欧盟Horizon 2020研究与创新计划的资助。

R参考

[1] 思科视觉网络索引，“ Zettabyte时代-趋势和 分析”*思科白皮书*，2013年。

[2] 卡巴斯基实验室，“全球IT安全风险调查2014 -分布式拒绝 的 服务 （DDoS）攻击 的攻击，” https://goo.gl/dbg3wZ。

[3] 威瑞信博客，“威瑞信Q1 2016 DDos的趋势：攻击活动增加111百分比年过一年，” https://goo.gl/Srm3cW。

[4] J. Sherry， S。Hasan，C。Scott，A。Krishnamurthy等。*ACM SIGCOMM'12中的*“使*中间盒*成为其他人的问题：网络处理作为云服务”。

[5] C.兰，J.雪利酒，RA波帕，S. Ratnasamy和Z.柳，“踏上：安全地 外包 中间盒到 的 云”， 在*USENIX NSDI'16*。

[6] H. Ballani，P. Costa，C. Gkantsidis，M . P. Grosvenor等。“启用终端主机 的网络功能，” 在*ACM SIGCOMM'15*。

[7] W. Zhang，G。Liu，A。Mohammadkhan，J。Hwang等。， “SDNFV在灵活，动态的软件开发管理网络定义的基于应用和流量感知的数据对飞机的控制，” *Middleware'16*。

[8] M. Feilner，*OpenVPN：构建和集成虚拟专用网络*。Packt Publishing Ltd， 2006年。

[9] E. Kohler，R。Morris，B。Chen，J。Jannotti和M. F. Kaashoek，“ Click 模块化路由器”，*ACM Transactions on Computer Systems*，2000年。

[10] EFF，“我们正在加密整个Web的过程已经中途了”， https：//goo.gl/VdUj5b，2017年 。

[11] D.勒，A.费雯莉，I. Leontiadis，Y. Grunenberger 等。“在成本中 的 小号 在HTTPS，” 在*ACM CoNEXT'14*。

[12] “判断伪造SSL LS黄，A.水稻，E. Ellingsen和C.杰克逊， CERTI音响凯茨 在 所述 ，野” 在*IEEE标普2014*。

[13] D. Naylor等。“多情境TLS（mcTLS）：启用安全在-网络功能在TLS，”在*ACM SIGCOMM'15*。

[14] D. Naylor等。，“而且：为更安全的通信，然后有更多 比 2个 政党，” 在*CoNEXT'17*。

[15] J.雪利酒，C.兰，RA波帕，和S. Ratnasamy，：“BlindBox深度包 检查 过 加密TRAF音响C”，在*ACM SIGCOMM'15*。

[16] 1&1 Internet Ltd。，“ IP欺骗：攻击者对数据包 的简单操纵”，https：//goo.gl/Dn1CaV，2017年 。

[17] D. Kreutz等。，“软件定义的网络：全面调查”，*IEEE会议录*， 2015年。

[18] B. Han等。“网络功能虚拟化：挑战和opportu- nities 的 创新，” *IEEE通信杂志*，2015年。

[19] J. Soares等。，“迈向具有服务 功能的telco云环境”，*IEEE通信杂志*，2015年。

[20] C. Dixon，H. Uppal，V. Brajkovic，D. Brandon等。“ETTM：一个可扩展的故障 容错 网络 经理”， 在*USENIX NSDI'11*。

[21] S. Gueron，“适用于通用处理器的内存加密引擎。” *IACR密码学ePrint档案*，2016年。

[22] I. Anati，S。Gueron，S。Johnson和 V. Scarlata，“基于 CPU 的 认证 和 密封的创新技术”， 在*HASP'13中*。

[23] S. Arnautov，B.泽，F.格里，T. Knauth等。“烤饼：安全的Linux 容器 与 英特尔新交所”，在*USENIX OSDI'16*。

[24] C.-C. Tsai，DE Porter和M. Vij，在*USENIX ATC'17中*， “石墨烯-SGX ：在SGX上用于未修改应用程序的实用库OS ”。

[25] S. Shinde，DL Tien，S。Tople和P. Saxena，“ PANOPLY：带有SGX Enclaves的低TBC Linux 应用程序”，在*NDSS'17中*。

[26] S. Brenner，C。Wulf，D。Goltzsche，N。Weichbrodt等。“SecureKeeper：精读网络dential动物园管理员使用英特尔SGX，”在*Middleware'16*。

[27] Y.许， W 崔和M. Peinado，“受控信道攻击：Determin- ISTIC 侧 通道 为 不可信的 工作 系统”， 在*IEEE SP'15*。

[28] N. Weichbrodt，A. Kurmus，P. Pietzuch和R.卡皮查，“AsyncShock：环境与开发同步 错误 在 英特尔 SGX 飞地，” 在*ESORICS'16*。

[29] J.凡Bulck，F. Piessens和R. Strackx，“SGX-步骤：实用攻击框架 用于 精确 飞地 执行 控制，” 2017。

[30] 徐世,、李乙，金圣美。Shih等。“SGX盾：启用地址空间 布局 随机化 的 SGX计划，” 在*NDSS'17*。

[31] M.-W. 施，S.李，T. Kim和M. Peinado，“T-SGX：根除受控信道 攻击 针对 飞地 节目” 在*NDSS'17*。

[32] A. Garg和AN Reddy， “通过QoS规章缓解DoS攻击”，*微处理器和微系统*，2004年。

[33] H. Nguyen和V. Ganapathy，：“EnGarde相互可信任检验 的 SGX 飞地，” 在*IEEE ICDCS'17*。

[34] M.绿，R. Droms，R.豪斯利，P.特纳，S. F输入“在TLS数据中心使用的静态Dif的网络连接的e-Hellman的的1.3节” https://goo.gl/95FaWD。

[35] E. Rescorla，“ TLS 1.3中间盒问题更新”，https：//goo.gl/zCUuRG，2017年 。

[36] 英特尔公司，“用于Linux OS的英特尔软件保护扩展（Intel SGX）SDK”， https：//01.org/intel-software-guard-extensions，2017年。

[37] P.-L. Aublin，F. Kelbert，D.奥基夫，D. Muthukumaran等。，“ TaLoS：SGX Enclaves内部的安全 且 透明的 TLS 终端”，伦敦帝国理工学院，技术。2017年3月，第5/5期。

[38] M. Orenbach，P. Lifshits，M. Minkin和M.西尔伯斯坦，“Eleos：ExitLess OS 服务 为 SGX 飞地，” 在*EuroSys '17*。

[39] S. Checkoway和H.沙哈姆，“伊阿古攻击：为什么系统调用API 是 一个 坏 不可信的 RPC 接口，” 在*ASPLOS'13*。

[40] M. 罗斯奇， “Snort的： 轻量级 入侵 检测 为 。网络” 在*USENIX LISA'99*。

[41] A. V.阿霍和MJ Corasick，“EF音响cient字符串匹配：辅助到书目 搜索，”*通信的的ACM*，1975。

[42] W. Sun和R. Ricci，“快速而灵活：利用GPU 和 Click进行并行数据包处理”， 在*ACM / IEEE ANCS'13中*。

[43] Alexa。，http：//www.alexa.com/，2017年 。

[44] T. Karagiannis 等。，“网络异常处理程序：主机的网络控制 在 企业 网络中，” 在*ACM SIGCOMM'08*。

[45] L. Lamport等人。，“ Paxos变得简单”，*ACM Sigact News*，2001年。

[46] M. V ü ķ OLI ç ，“该任务为可伸缩blockchain ˚F阿布里克：证明-OF-瓦特扫与BFT复制”，在*iNetSec'15*。

[47] 十元，十王，J.林和C.王， “隐私保护的深度包检测 在 外包 中间件，” 在*IEEE INFOCOM'16*。

[48] H.段，十元，和C.王“的灯箱：SGX辅助安全网络 功能 在 接近原生 速度”*的arXiv：1706.06261*，2017年。

[49] M.考芙琳，凯勒E.和E.乌斯特罗， “可信点击：克服安全 问题 的 NFV 中 的云”， 在*ACM SDN-NFV Security'17*。

[50] D. Kuvaiskii，S。Chakrabarti和M. Vij， “ 具有 Intel Software Guard Extension的Snort入侵检测系统”，*arXiv：1802.00508*，2018年。

[51] B. Trach等人。*ACM SOSR'18*， 2018年，“ ShieldBox：使用屏蔽执行保护中间盒” 。

[52] J. Han，*S。Kim，J。Ha*和D. Han，“ SGX-Box：使用安全*中间盒*模块在加密的流量上启用可见性”，在*ACM APNet'17中*。