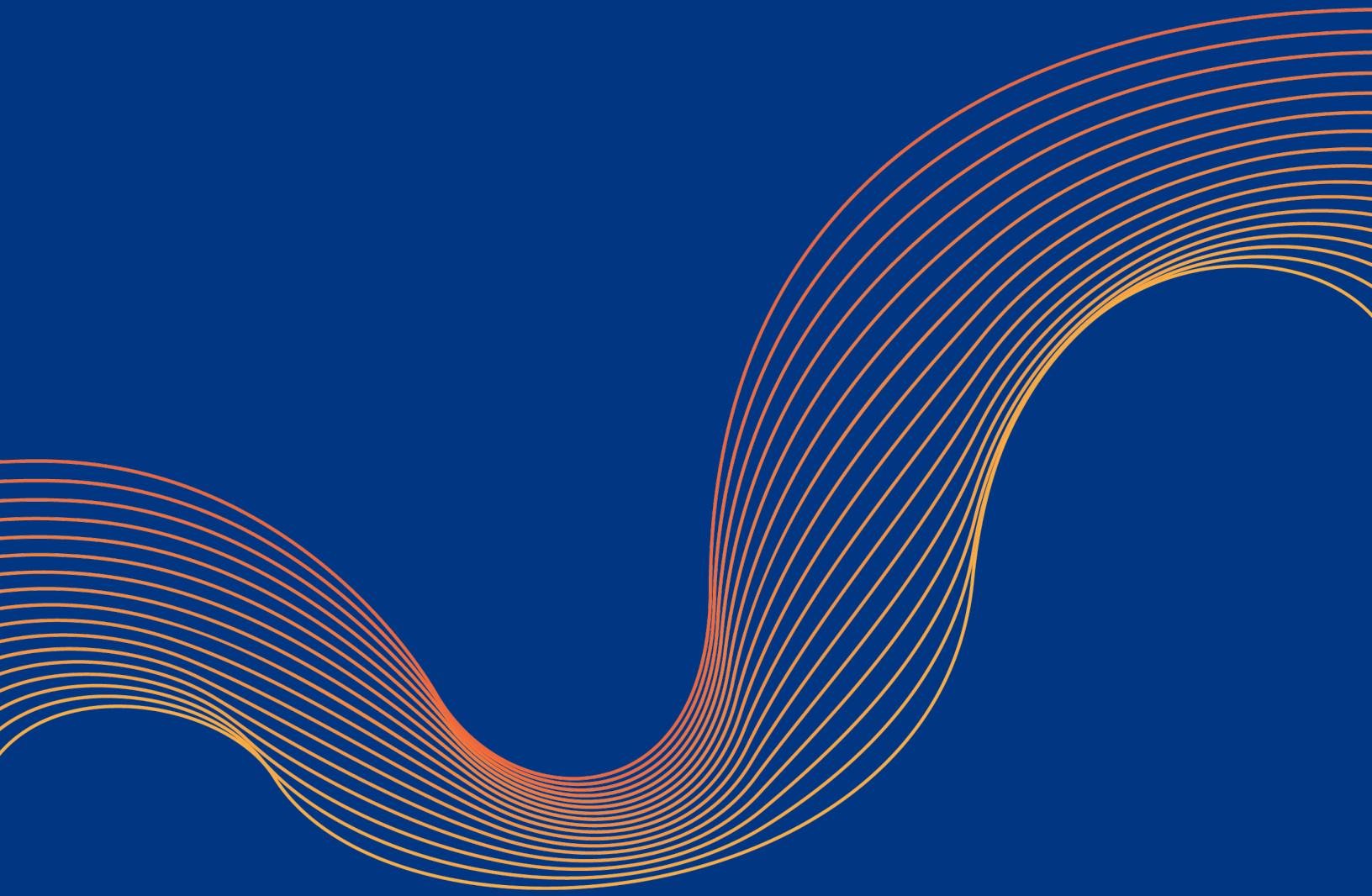
CLOUDFLARE

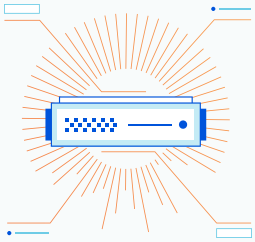
常见的浏览器隔离挑战，及如何克服这些挑战

互联网浏览与零信任安全的融合



010 8524 1783 | enterprise@cloudflare.com | www.cloudflare.com/zh-cn/

索引



简介

IT 和安全团队有充分理由不信任公共互联网。根据 Verizon 的一份报告，网络钓鱼和恶意软件占 2020 年数据泄露的 39%。Cloudflare 委托 Forrester Consulting 进行的一项研究显示，在 2020 年，员工人数超过 1000 人的公司中，有 61% 经历的网络钓鱼攻击超过了前几年。

所有 IT 和安全专业人士都希望避免其组织包含于这些统计数据中。具体而言，他们希望：

- **阻止恶意软件和网络钓鱼**，而两者会不断改变策略来逃避监测。
- **防止一般数据丢失**，不管是通过受感染设备还是用户交互。
- **更好地掌握员工的互联网浏览活动**，以便了解组织的具体威胁状况并对实际发生的泄漏事件做出更快的响应。

这些用例是**零信任安全**的重要元素，互联网资产不应被默认信任——因而必须在用户交互时进行安全处理。

为了实现这些目标，一些组织正在使用浏览器隔离技术，其中员工的互联网浏览与本地网络和基础设施隔离。如能彻底有效地实施，浏览器隔离有望成为缓解互联网攻击的最强大方法。不幸的是，迄今这依然是一种小众技术，原因是存在多种挑战，包括**高成本、浏览体验欠佳、后勤管理障碍**以及**安全漏洞**。

本文详细描述了这些挑战，以帮助安全和 IT 团队更好地了解其互联网浏览安全需求。本文还描述了一种克服这些挑战的方法，最后说明了Cloudflare 如何将这种方法集成到其全球网络中。

常见浏览器隔离策略的挑战

恶意软件、网络钓鱼和数据丢失影响每一个行业和领域的公司级组织。浏览器隔离是最常用于对抗前两种威胁的解决方案，从而增强安全 Web 网关所用的阻止列表、文件匹配和行为方法。

但在实际应用中，浏览器隔离常常无法达到这个目标。

为什么会这样呢？考虑最常见浏览器隔离方法的局限性：

流式传输来自云端的浏览活动截图

这种方法有时被称为“像素推送（pixel-pushing）”，捕获最终用户浏览器中的事件并传送到云托管的远程浏览器中，后者实际执行浏览操作并将远程浏览器窗口的一系列像素图像传回最终用户的浏览器。通过这种方式，任何恶意代码——不管是自动下载还是用户有意操作——都与最终用户的设备保持分离。而潜在的钓鱼网站——例如带有用户名/密码字段的页面——可以只读方式显示或添加警告信息。

这种方法很好地将终端设备与恶意软件和网络钓鱼隔离开来。然而，这种方法有如下问题：

最终用户延迟：当远程浏览器隔离托管于公共云，或一个地理位置受到限制的私有网络时，由于最终用户与浏览器隔离数据中心距离较远，最终用户可能会遇到延迟。如果最终用户的流量通过并非托管于同一数据中心的其他安全工具——例如安全 Web 网关时，或者需要多次“通过”以低效率方式架构的容器时，这个问题变得更加严重。

高成本：从计算角度来看，将远程网页持续编码成视频流并发送到最终用户端点设备的成本极高。这也需要大量带宽，即使经过高度优化。这些成本通常会转嫁给客户。

安全漏洞：由于“像素推送”往往导致糟糕的最终用户体验，很多组织仅要求有权访问特别敏感数据的团队使用——例如财务、人力资源或公司高管。组织也可能仅对一小部分被认为特别危险的网页应用远程浏览。无论哪种情况，组织将继续处于暴露状态——要么通过不受保护的员工，要么通过已经遭到破坏的“可信赖”站点。

高带宽需求：图像流式传输需要大量带宽，这可能会导致网络基础设施不堪重负，并对最终用户体验造成不利影响。此外，像素密度随分辨率呈指数级增长，意味着高 DPI 设备上的远程浏览器会话（尤其是字体）可能会显得模糊不清。

在云端分解网站并清除恶意代码

这种方法通常称为 DOM 操作。在前端开发中，DOM（Document Object Model，文档对象模型）是组成网页结构和内容的对象的数据表示。在 DOM 操作中，云托管远程浏览器检查页面的 HTML、CSS 和其他元素，并试图消除活动代码，例如 JavaScript、已知漏洞和其他潜在恶意内容。然后，远程浏览器将这些代码转发到最终用户的浏览器，由后者用于重新构建一个“干净”版本的的网页。此外，和“像素推送”一样，DOM 操作也可将某些页面标记为存在网络钓鱼风险。

由于 DOM 操作仅传送网站代码，而非浏览体验的完整流，因此需要较少带宽并带来更快的最终用户体验。

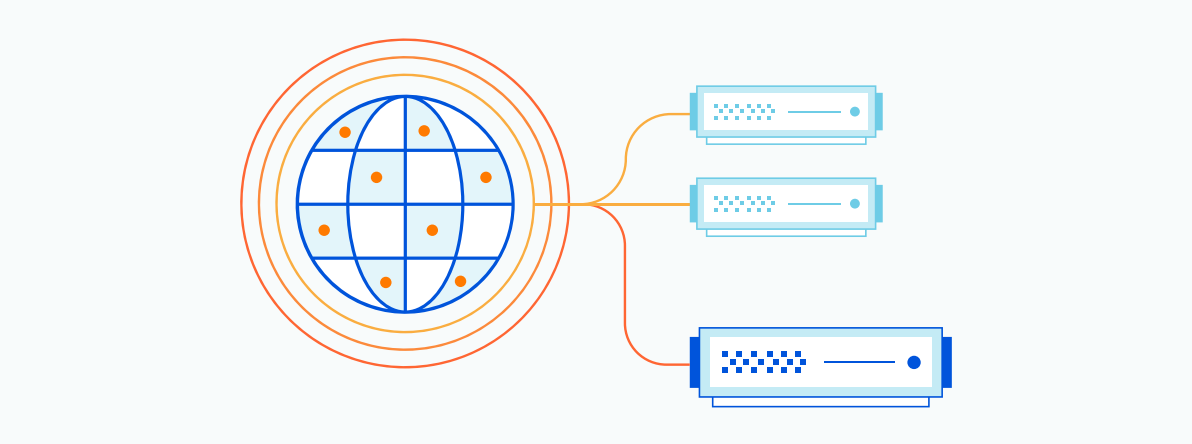
然而，这种方法有如下问题：

最终用户延迟：与“像素推送”一样，如果 DOM 操作浏览器隔离运行于公共云，或地理位置受限的私有网络中，在源服务器距离太远或浏览器隔离和其他安全工具托管于其他数据中心时，最终用户仍会体验到延迟。

破坏网站体验：不可避免的是，试图消除恶意活动代码——以及重构 HTML 和 CSS 和重建不常见的站点架构——会破坏网页，导致渲染不正确或完全不能渲染。此外，今天可行的网站明天未必可行，因为站点发布者也许会每日进行更改，从而破坏 DOM 重建功能。DOM 操作甚至难以支持常见的企业级服务，如 Google G Suite 或 Microsoft Office 365。结果带来无穷无尽的问题，需要大量 IT 资源来解决。

高成本：一些 DOM 操作服务托管于第三方公共云，产生额外成本，这些成本通常会被转嫁到客户身上。无论哪种情况，组织将继续处于暴露状态——要么通过不受保护的员工，要么通过已经被破坏的“可信赖”站点。

安全漏洞：与“像素推送”一样，DOM 操作不良的最终用户体验往往意味着组织仅能偶尔使用这种方法。此外，尽管 DOM 操作是浏览器隔离的一种形式，它仍将不受信任的第三方代码发送到端点设备。如果该服务未能识别恶意代码——鉴于威胁形势不断变化，这种风险始终存在——终端设备仍有可能遭受威胁。



在设备上的虚拟机中隔离浏览活动

在这种方式中，用户终端设备上安装的软件创建一台与设备操作系统其他部分隔离的虚拟机。所有浏览活动都在这台虚拟机上进行，因此任何下载的恶意软件都无法感染设备的其他部分。此外，和其他方式一样，软件可将特定页面标记为具有网络钓鱼风险。

不幸的是，这种方式有如下问题：

高 CPU 和 RAM 要求：运行单独的虚拟机会拖慢大部分个人计算机的速度，导致最终用户的浏览体验变慢。

终端设备管理困难：通过本地软件实现浏览器隔离要求 IT 团队在每一台用户终端设备上安装和更新所述软件——对大型组织而言，这是一个艰巨的任务。如果组织有大量远程员工，或者雇用不使用企业配发设备的第三方合同工时，这种后勤复杂性尤其严重。

移动兼容性问题：基于设备上虚拟机的浏览器隔离对操作系统有特定要求。移动设备往往不受支持。

隔离失败：本地浏览器隔离服务定期遇到允许恶意代码访问主操作系统的漏洞。在这种情况下，IT 团队或终端用户必须人工安装补丁或更新。但如果补丁未正确安装，终端设备可能依然容易受到攻击。

最终用户体验不佳：基于虚拟机的方式往往要求最终用户使用另外的“虚拟化”浏览器、窗口或桌面。这需要培训并增加 IT 支持负担。

这些直接挑战造成额外挑战

直接的后果包括高成本、糟糕的最终用户体验和终端设备管理困难，但这些并非常见浏览器隔离方式的全部挑战。

另一个挑战是设备受损的长期后果。当最终用户通过受破坏的设备登录到敏感应用程序时，恶意软件可在用户不知情的情况下访问该应用程序的数据。

此外，所有这些浏览器隔离方法都难以防止某些类型的数据丢失。内部行为者仍可上传和通过电子邮件发送敏感信息，或将其输入到在线表单中。

远程浏览器隔离的更优方式

尽管存在前述的各种问题，组织仍希望采用远程浏览。

幸运的是，某些技术能助一臂之力：

问题	解决方案
 延迟	**使用大型边缘网络：**将浏览器隔离托管于一个在任何地方均接近最终用户的全球边缘网络，而非有限数量的公共云数据中心。此外，使用与安全 web 网关和其他安全工具位于同一数据中心的浏览器隔离软件。
 高带宽要求	**不使用像素推送：**对于大型企业，从成本和用户体验角度来看，流式传输远程浏览活动的图像都是不切实际的。
 破坏网站体验	**使用原生浏览器技术：**使用已经内置于常见浏览器中的技术来实现远程浏览，在准确重建各种网站时更为可靠。
 高计算量成本	**下一代云计算：**避免托管于公共云的远程浏览器隔离。而是使用高效的无服务器计算技术，通过消除对基础服务器资源的编排和管理来改善虚拟化和容器化，从而更有效地使用这些资源。
 安全漏洞	**使用原生浏览器技术：**无需尝试决定发送或阻止哪些代码，原生浏览技术能完全避免发送代码。取而代之，这种技术能仅发送绘制页面的渲染过程最后一步。
 指定的用户终端困难	**无本地浏览器隔离：**终端设备上的隔离浏览活动速度太慢，难以管理，使这种方式完全过时。

要了解这些技术在实践中如何工作，请考虑 Cloudflare 浏览器隔离实例，后者原生集成于我们的零信任平台。

Cloudflare 如何使远程浏览具备成本效益并减少对最终用户的干扰

总的来说，Cloudflare 避免需指定终端设备困难的方式很简单，就是在一个全球边缘网络而非终端设备上运行。具体而言，Cloudflare 的技术也消除了其他问题：

问题	解决方案	Cloudflare 实施
 延迟	使用大型边缘网络	远程浏览器隔离在 Cloudflare 边缘网络的每一个数据中心上运行，而这个边缘网络遍布 100 个国家的 200 多个城市，与世界上网人口的 95% 连接时间不到 100 毫秒。同一个基础设施提供超低延迟的全球 DNS 和 CDN 服务。 此外，我们的远程浏览与我们的转发代理无缝交互，以应用所有其他过滤器（例如阻止网页的部分内容）和检查，无需在不同的点解决方案之间多次通过和跳跃。
 高带宽要求	原生浏览器技术	Cloudflare 的网络矢量渲染技术（下文将详述）传输绘制指令而非像素图像或“经清洗”的代码。这种方法所需带宽仅为常规浏览或 DOM 操作的很小比例，更不用说“像素推送”了。
 破坏网站体验	原生浏览器技术	Cloudflare 的远程浏览器基于开源的 Chromium 引擎，这是 Google Chrome 和另外 21 个常见浏览器所用的引擎。Chromium 引擎正在进行的重大投入确保最高水平的网站兼容性。 此外，Cloudflare 的网络矢量渲染技术传送绘制指令而非“清洁”代码，从而确保即使最复杂的网页也不会被破坏。
 高计算量成本	下一代云计算	由于 Cloudflare 浏览器隔离在我们自有网络上运行，我们不必将公共云的成本转嫁给客户。而且，我们能非常高效地协调和管理服务器资源，从而避免长达数秒的冷启动，这种冷启动常常会影响到托管于公共云的应用程序。
 安全漏洞	原生浏览器技术	通过传送轻量级的矢量绘制指令——而非任何原始网站代码——Cloudflare 消除了不受信任代码在用户终端设备上运行的风险。未检测到的恶意软件仅会破坏远程浏览器，而不会影响到终端用户。由于 Cloudflare 提供强大的最终用户体验，企业能将远程浏览应用于原本可能不受保护的较低风险用例。
	最终用户行为的精细化控制	一般情况下，数据丢失预防工具通过允许或阻止传输来保护在网络上输送的数据。 Cloudflare 浏览器隔离将赋予管理员对如下方面的精细化控制： • 复制/粘贴/打印权限 • 上传/下载操作 • 一般的键盘活动 • 表单输入权限 • 下载文件存储位置* * 即将推出

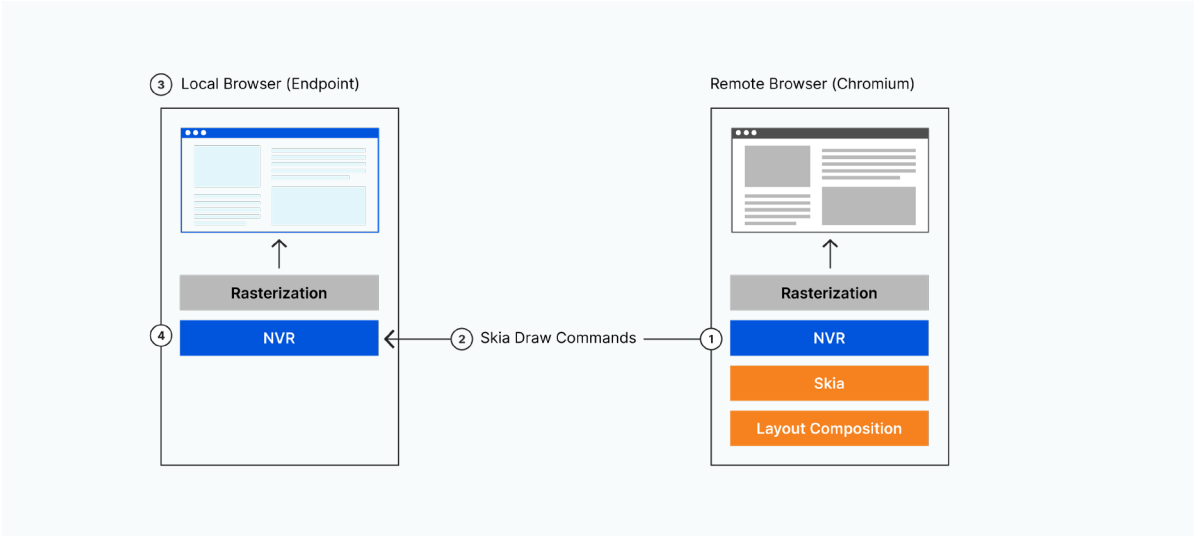
继续阅读，以深入了解上述 NVR 技术：

网络矢量渲染技术提供更佳的最终用户体验并填补安全漏洞

如上所述，Cloudflare 的远程浏览器基于 Chromium。Chromium 浏览器的一个关键架构特征是使用了 Skia——一个广泛用于安卓、Google Chrome、Chrome OS、Mozilla Firefox 和很多其他浏览器的跨平台图形引擎。所有兼容 HTML5 的浏览器都能渲染 Skia。Chromium 浏览器窗口中可见的所有内容都是通过 Skia 渲染层渲染的。其中包括应用程序窗口界面，例如菜单，但更重要的是，网页窗口的全部内容都是通过 Skia 渲染的。Cloudflare 甚至能隔离下载的文件并根据最终用户的需求将文件移动到各种位置。

Cloudflare 的网络矢量渲染（NVR）技术拦截远程 Chromium 浏览器的 Skia 绘制指令，对其进行标记和压缩，加密并传输给端点桌面或移动设备本地运行的任何 HTML5 兼容浏览器。NVR 捕获的 Skia API 命令已预先光栅化，这意味着它们非常紧凑。而且，由于 Skia 应用非常广泛，Cloudflare 的远程浏览适用于任何现代 Web 浏览器。

网络矢量渲染也更加安全。如上所述，由于 Cloudflare 将绘制指令而非实际的网站代码传送给用户终端设备，对应的数据传输不是一种攻击途径。



进一步了解及开始使用

从 Cloudflare 成立之日起，我们的使命就是帮助构建更美好的互联网，并普及以往仅拥有复杂网络、专职 IT 团队和庞大预算的大型企业才能使用的技术。更佳的远程浏览器隔离是以上使命的一个重要部分。

通过使远程浏览器隔离更具成本效益——并提供出色的最终用户体验——我们希望更多组织能体验到这种技术的真正价值。就像不久之前 HTTPS 加密仅用于“敏感的”登录页面和电子商务结帐一样，我们认为，信任任意网站代码将变得和创建零信任浏览新范式一样过时。

要进一步了解 Cloudflare 浏览器隔离及其能如何帮助您实现零信任浏览，请访问 https://www.cloudflare.com/products/zero-trust/browser-isolation/