/dev/SDB：软件定义启动——一种可在任何地方、任何环境进行无盘启动的新型标准

Aditya Mitra Hamza Haroon Amaan Rais Shah Mohammad Elham Rasooli*CyberMACSCyberMACS CyberMACS CyberMACSKadir Has University KadirHasUniversity Kadir Has University Kadir Has University*土耳其 土耳其 土耳其 土耳其 aditya.mitra@stu.khas.edu.tr hamzaharoon@stu.khas.edu.tr amaanshah@stu.khas.edu.tr m.rasooli@stu.khas.edu.tr ORCiD: 0000-0002-9612-0810 ORCiD: 0009-0006-4453-6298 Bogdan Itsam Dorantes Nikolaev Tu˘gc¸e Ballı*CyberMACS* 管理信息系统系 *Kadir Has University Kadir Has University*土耳其 土耳其 bnikolaev@stu.khas.edu.tr tugce.balli@khas.edu.tr ORCiD: 0009-0008-5119-1683

arXiv:2601.20629v1 [cs.CR] 28 Jan 2026

摘要——**计算机不过是一种处理输入指令的设备。然而，随着计算机不断发展，指令或代码也变得日益复杂。随着计算机开始被非技术人员使用，用户无需了解底层代码或硬件就能使用机器，变得至关重要。操作系统由此成为将用户输入转换为硬件实际操作的基础。随着操作系统日益复杂、选择越来越多，人们逐渐认识到，不同群体，尤其是在企业场景中，需要使用不同的操作系统。要在一台机器上为共享计算机安装所有这些系统，变成了一项艰巨的任务，由此催生了基于网络的引导。然而，基于网络的引导仅限于有线连接，使用范围因此局限在非常小的地理区域内。我们提出的系统 /dev/SDB 旨在建立一项标准，让任何用户无须接入企业网络，即可访问获准使用的操作系统。该系统旨在通过 Wi-Fi 和蜂窝网络连接提供同样的服务，确保员工真正能够随时随地工作，同时遵循操作系统相关政策，并且无需冗余硬件。**

索引词—**操作系统、无线启动、无盘、蜂窝启动**

I. 引言

启动通常是打开计算机后加载操作系统的最初步骤之一。操作系统进而负责处理计算机的全部活动。但可以看到，存在用途各异的多种操作系统，并非每个人都需要相同的系统，尤其是在企业场景中。计算机的某些组件（例如存储）往往是冗余的，仅用于存放操作系统。要通过持久化存储介质启动计算机，会使用一种网络启动标准。这通常利用预启动执行环境（PXE）。然而，以往关于基于 PXE 的部署以及企业场景中网络启动的研究，一直局限于有线网络，尤其是数据中心和企业网络。

然而，随着现代社会工作方式的变化，许多组织如今采用了随处办公模式。员工大多使用远程系统，通过互联网从任何地点连接。因此，在现代劳动力中，有线连接已不再像过去那么重要。虽然组织正在部署瘦客户端和虚拟桌面基础架构（VDI），以解决安全问题并让所有操作都通过集中式系统运行，但从安全角度和财务角度来看，这往往还不够。用户通常使用从持久化存储介质启动的通用操作系统，这会带来恶意软件、病毒等风险。此外，对所有员工而言，通用操作系统可能是多余的，因为并非每个人都会在日常工作流程中使用操作系统的所有功能。持久化存储介质通常也是多余的，仅用于存放操作系统，却会扩大攻击面。使用通用操作系统的用户往往会在计算机系统上安装外部应用程序，从而威胁企业资源的机密性。

在现代类似办公室的环境中，即使计算机系统采用有线连接，任何部门的员工通常也可以自由选择任意一台台式计算机，并随时开始工作。这通常不局限于单一地点；员工经常会前往组织的不同地点或办公室

并开始工作。不同员工通常使用不同的操作系统，例如，开发人员可能会使用 Ubuntu 等 Linux 发行版，渗透测试人员可能会使用 Kali、Parrot 或 Black Arch 等发行版，销售人员可能会使用专为会议定制的 Zorin，而会计人员可能会使用专门针对电子表格和其他所需软件定制的操作系统。这就要求从操作系统层面开始，根据每位员工的角色和需求动态配置系统。

所提出的标准旨在构建这样一种系统：计算机系统（包括无线设备和便携式计算机）可根据用户权限和用户需求，启动进入多个操作系统、诊断系统或任何基于便携式操作系统的环境（如 WinPE）之一。该标准解决了所有相关问题，并提供了一种系统，即使在系统启动之前也能对用户进行身份验证。随后，系统会检索分配给用户的操作系统，并使用目标映像启动。该系统采用内存启动，通常会通过互联网从网络文件共享（NFS）加载整个操作系统映像。系统支持通过有线网络、无线网络和蜂窝网络连接到启动服务器，未来最好还能支持卫星网络，从而为用户提供安全的随时随地办公范式，同时确保用户始终获得符合其需求的操作系统。

本文分为以下几个部分：第 1 节介绍引言，第 2 节介绍文献综述，第 3 节介绍本研究的研究方法，第 4 节介绍本研究采用的实验设置，第 5 节则介绍威胁模型。

II. 文献综述

所提出的系统涵盖了与操作系统加载方式相关的各种概念，包括引导加载程序、预启动执行环境（PXE）、iPXE 和 gPXE 等 PXE 开源扩展，以及有关无盘云启动系统的现有研究。

引导加载程序构成了操作系统启动过程的基础：在系统固件运行后初始化硬件并加载内核。Sebastian 和 Sankar [1], 所述的传统多阶段设计依靠主引导记录（MBR）中的小型第一阶段加载程序来查找 GRUB 或 Syslinux 等后续加载程序，后者提供配置菜单和多重引导功能。从 BIOS 转向统一可扩展固件接口（UEFI）后，引入了安全启动功能，要求在内核执行前进行数字签名验证，以防止未经授权的固件 [2]。

现代引导加载程序（如 GRUB 2 和 Syslinux/PXELinux）结合使用链式加载和模块化配置，以支持多种启动来源，包括磁盘和网络 [3], [4]。然而，研究表明，内核前阶段的内存安全缺陷可能会削弱系统信任，这凸显了使用经过验证的安全启动代码的必要性[5]。近期的发展使引导加载程序的应用范围超越了本地环境：Park 等人提出了 FLEX-IoT [6]，该方案采用基于哈希的身份验证和自适应传输调度，以实现物联网设备的安全网络启动；Mitra 等人则推出了 Colaboot [7], ，这是一种无盘云启动模型，利用动态主机配置协议（DHCP）和简单文件传输协议（TFTP）等网络协议将操作系统加载过程虚拟化。

多年来的多项研究支持这样一种观点：网络启动传统上简化了操作系统部署、实现了集中管理，并消除了对本地存储的依赖。研究指出 [8] ，PXE（预启动执行环境）启动系统是一种网络协议，使计算机能够通过网络启动并安装操作系统。它是一种有效的流程，可简化安装过程，降低出错和出现不一致的可能性；尤其是在大型组织中，许多计算机需要采用相同的操作系统和软件配置。Johnston 和 Venaas 的 2006 年 RFC [9]详细说明了 PXE 标准化的作用。该 RFC 规定了 PXE 和 EFI 客户端用于唯一标识正在启动的计算机及其操作系统启动前环境的 DHCP 选项，使 DHCP 和 PXE 服务器能够提供相应的操作系统引导映像。

这些论文始终隐含地依赖有线连接，但这一点在嵌入式系统研究中最为明显。例如，ZUS+设备上的 U-Boot 可通过预启动执行环境（PXE）实现网络启动。标准 ZUS+ 流程中的初始启动步骤包含反映物理基础设施需求的应用特定细节。Fuchs 等人的 Split Boot [10] 研究提出了将所有应用特定设置转移到网络存储设备的方法，该设备随后会在启动过程中被自动访问，尽管有线连接仍然是必需的。

通常用于网络启动的简单文件传输协议（TFTP）存在显著的安全漏洞和性能问题。在网络启动期间，由于大量物联网设备根据各自不同的角色请求系统映像，服务器面临很高的需求，从而导致巨大的网络负载 [6]。当多个启动请求同时发生时，服务器的资源使用量会呈指数级增长。

DHCP 耦合限制在各篇论文所概述的核心架构中显而易见。网络启动系统（NBS）依赖三种主要协议：PXE、DHCP 和 TFTP，体现出三者之间的紧密互联。在此过程中，支持 PXE 的客户端必须通过 PXE 和 DHCP 请求 IP 地址，从而形成对 DHCP 系统的依赖。 [11]

性能研究记录了大规模设备群面临的可扩展性挑战。研究表明，在物联网平台内向物联网设备传输大量系统映像会导致网络流量激增。 [6] 随着传输系统映像的需求不断增长，网络流量消耗还将进一步增加。

iPXE 启动固件被视为传统 PXE 的扩展，提供了超越传统 PXE/TFTP 的功能。Stites 等人指出，iPXE 除了支持 TFTP 外，还支持通过 HTTP 和 iSCSI 启动，并可借助灵活的脚本编写来编排复杂的启动流程 [12]。

链式加载是一项功能：精简的 PXE ROM 将控制权移交给 iPXE，使现有 PXE 基础设施无需更改网卡固件，即可利用 iPXE 的增强功能。与标准 PXE 不同，iPXE 可以在启动脚本中嵌入 CHAP 凭据，使无盘客户端能够安全地从 iSCSI 目标启动，而无需在网卡上存储机密信息。近期研究已将 iPXE 集成到安全且可扩展的架构中；Oliveros 等人为 iPXE 扩展了类似 BitTorrent 的点对点协议，以分发启动映像并消除单服务器瓶颈 [13]。

Moser 展示了一种基于 iPXE 的无状态启动环境，采用 UEFI 安全启动和脚本签名，确保端到端的启动完整性 [14]。然而，iPXE 内置的传输层安全协议支持最高仅到 TLS 1.2，且使用了已弃用的密码套件，因此需要谨慎配置以维护安全性。Moser 还尝试将 iPXE 嵌入网卡固件，以替代厂商 PXE 代码；虽然这实现了无缝的启动流程，但由于固件工具因设备而异，难以在大规模环境中实际应用。在云数据中心，iPXE 作为裸机配置的网络启动层，促使研究人员探讨其中的信任问题。Schear 等人提出了一种云引导架构，利用 TPM/vTPM 安全配置物理机和虚拟机 [15]。例如，M2（可塑金属即服务）使用 iPXE 编排多租户物理服务器的无盘配置：M2 在每次启动时都通过链式加载启动 iPXE，以绕过网卡限制，并通过 HTTP 或 iSCSI 加载操作系统镜像，将裸机节点视作云实例 [16]。

基于云的启动是指从云端启动系统的过程，无需本地存储即可实现可扩展性和安全性 [17]。Colaboot 提出让机器仅从远程下载最精简的引导加载程序，并从云端串流操作系统，在易失性内存中运行操作系统，以确保一致性和韧性 [7]。ClassCloud 等无盘远程启动系统使用 PXE 和 TFTP 等协议从中央服务器启动，可满足大规模企业的需求，但在异构硬件环境中面临挑战 [18]。另一种学术方案提出了一个多步骤引导建立系统，整合云基础设施，以灵活管理工作流 [19]。

专有解决方案包括戴尔的 BIOSConnect：当本地恢复分区不存在时，它可以将操作系统映像下载到 RAM 磁盘，以实现系统恢复 [20]。微软的 Windows 365 Boot 允许用户直接启动进入托管于云端的“云电脑”，适用于共享设备环境 [21]。这种方案主要应用于优化电源使用、可扩展性和管理，但在无盘环境中可能会带来网络延迟方面的挑战。

广泛的文献综述揭示了研究空白：这是首项研究，提出在企业场景中通过软件为每位用户定义所使用的操作系统，同时支持通过任意类型的连接（有线、无线、蜂窝网络等）进行远程启动。该系统旨在将一个硬件模块集成到计算机系统中，以实现这一目标。

III. 研究方法

所提出的标准提供了一种系统，可启动多个系统，不受位置和连接类型的限制，以启动管理员为用户配置的操作系统。

该系统由两个不同部分组成：

- 一个硬件模块，用于在启动前处理网络连接
- 一个云模块，用于处理用户身份验证和操作系统映像。

A. 硬件模块

用于处理网络连接的硬件模块基于一台轻量级单板计算机（SBC），运行 Alpine Linux 等轻量级发行版。该模块具有两个网络接口，其中一个通过 PCIe 或短 UTP 连接器等其他物理介质，在内部连接到主机计算机。第二个接口则连接到上游网络，例如 Wi-Fi 适配器、蜂窝调制解调器或有线连接器。

该模块桥接两个接口。如果在上游网络中发现 DHCP 服务器，它会在桥接接口上启用代理 DHCP 服务器。如果未发现 DHCP 服务器，该模块会在桥接接口上设置静态IP地址并启动 DHCP 服务器。此外，该模块还托管 TFTP 服务器，为目标计算机提供一个自定义的、基于 iPXE 的引导加载程序映像。该映像会尝试从指定域名的 Web 服务器进行链式加载。

根据不同场景启用的代理 DHCP 服务器和标准 DHCP 服务器，都会将基于 iPXE 的引导加载程序指定为引导文件，并将自身的 IP 地址指定为 DHCP 的 next-server 选项。标准 DHCP 服务器会将该模块自身的 IP 地址作为 DNS 服务器提供，而代理 DHCP 服务器不会更改上游 DHCP 服务器提供的 DNS 配置。基于 iPXE 的引导加载程序会从云模块链式加载引导文件，云模块由域名指定。

硬件模块托管一个基于 Flask 的 Web 服务器模块，用于配置 Wi-Fi 和蜂窝网络连接。Web 服务器模块以 iPXE 表单和菜单的形式返回响应，即使目标电脑没有安装操作系统，也可以对其进行渲染。

硬件模块运行着一个 DNS 服务器，该服务器会将所有名称解析为自身地址，而不会向上游网络进行任何递归解析。目标计算机开机后，会进入 PXE环境，并发送 DHCP广播。如果上游网络未连接或没有 DHCP服务器，硬件模块会返回一个 DHCP提供报文，其中包含与其静态 IP 位于同一子网中的 IP 地址。它会查找 next-server字段，找到 TFTP 服务器，并尝试加载基于 iPXE 的引导加载程序。引导加载程序加载完成后，它会尝试从云的域名进行链式加载。

然而，由于提供的 DNS 服务器指向硬件模块本身，因此该域名会解析到硬件模块，并触发基于Flask的应用程序。该应用程序会提供 iPXE表单，要求填写 Wi-Fi SSID、密码或其他蜂窝

网络配置信息。连接到互联网后，同样的流程会再次重复。

现在，当上游网络连接到互联网时，代理 DHCP 服务器允许目标机器使用上游网络本身的 IP 地址。它仍使用自己的 IP 地址作为 TFTP 服务器，并提供相同的引导加载程序以进行链式加载。

DNS 服务器通过域名解析出实际的云服务器后，引导加载程序链便从云端加载，首先显示一个表单，要求用户输入用户名和密码。云模块根据用户名和密码验证用户身份，然后提供分配给该用户的操作系统。图 1 展示了硬件模块的示意图。

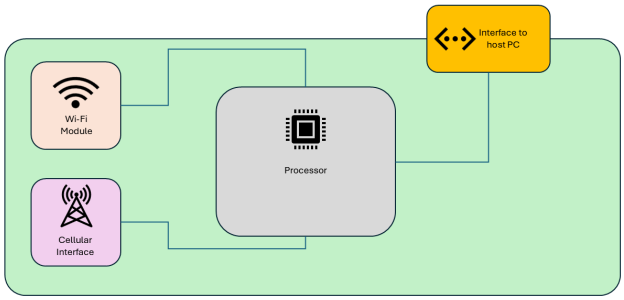


图 1. 硬件模块示意图

B. 云模块

云模块由 Web 服务器、数据库、文件存储以及可选的网络文件共享（NFS）服务器组成。它包含两个子模块：

- 配置模块/管理模块
- 使用模块

配置模块以 Web 应用程序的形式运行，允许管理员定义操作系统并为其分配角色。通过上传所需文件（例如内核和初始 RAM 磁盘）并以脚本格式定义内核参数，即可定义操作系统。配置模块还支持用户管理，包括创建用户、删除用户以及为用户分配操作系统。操作系统元数据和用户详细信息存储在 SQLite 数据库中，而操作系统文件则以文件形式存储。云模块还会针对所用计算机的 MAC 地址，记录所有成功和失败的身份验证尝试，并在 Web 应用程序中显示这些记录。图 2 展示了如何在云模块中定义操作系统。

因此，该系统支持多个用户，并可为每位用户分配操作系统。每位用户只有在完成身份验证后，才能启动进入企业计算机。此外，通过 MAC 地址可以轻松监控哪位员工使用了哪台计算机，从而进一步了解

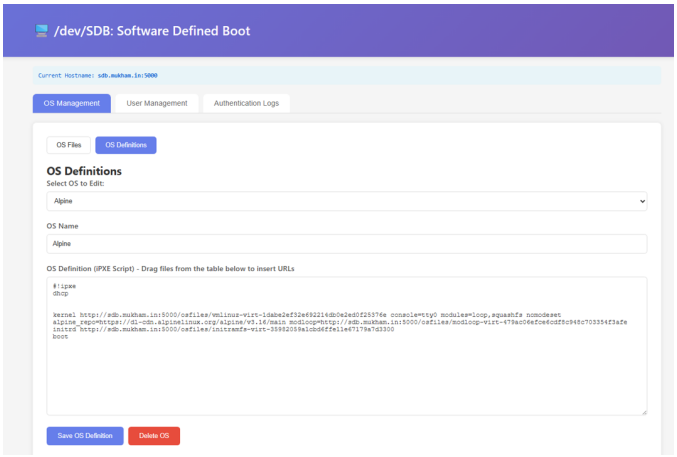


图 2. 操作系统定义

企业的使用模式。该系统为用户提供无缝的入职和离职流程，已离职的用户将无法启动进入企业系统。图 3 展示了该系统的工作流程。

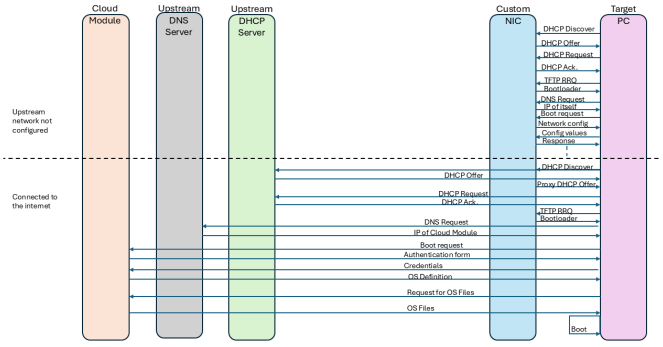


图 3. 系统工作流程

IV. 实验设置

该系统部署在 GNS3 应用程序的模拟环境中。硬件模块网卡基于 Alpine Linux 映像开发为模板，配备 128MB 随机存取存储器，以确保其占用空间尽可能小。它连接到目标电脑，该电脑是一个没有存储介质、配备约 256MB 随机存取存储器的标准 QEMU 设备。虽然实际应用中的目标 PC 将拥有更多内存，但 256MB 对于模拟而言已经足够。我们创建了三个目标 PC 实例，以模拟企业环境中的不同电脑。

云模块运行在主机电脑上，位于自动创建分段网络的 GNS3 模拟之外。这进一步确保了目标电脑与云模块不处于同一广播域中，并且目标电脑位于 GNS3 网络地址转换之后，从而模拟真实的互联网服务提供商级运营商级网络地址转换环境。图4展示了实验设置。

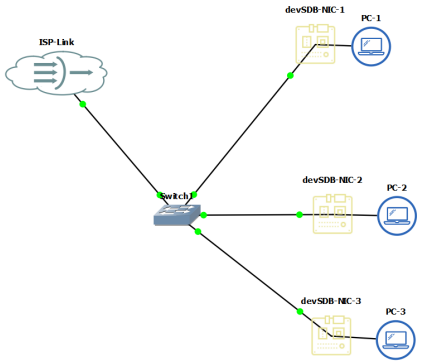


图 4. 实验设置

云模块使用 Python 和 Flask 框架开发，充当 Web 应用程序，并使用 SQLite 存储用户信息和操作系统元数据。Flask 应用程序将上传文件存储在一个目录中。

通过管理模块定义了三种轻量级操作系统：Kolibri OS、Tiny Core Linux 和 Alpine Linux。操作系统文件（包括内核、初始 RAM 磁盘映像和文件系统）已上传至云模块。图 5 展示了云模块的文件管理流程。

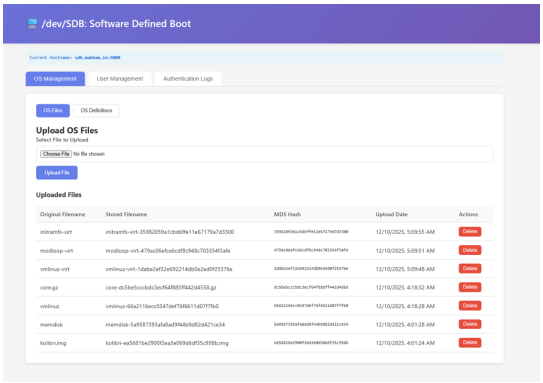


图 5. 操作系统文件

随后创建了三个用户，并为他们分配了要测试的操作系统。图 6 展示了用户管理界面。

用户能够在模拟中启动目标电脑，并看到登录界面。用户输入凭据后，系统会自动启动分配给他们的操作系统。图 7 展示了目标 PC 上的登录提示符。

目标机器从云模块获取操作系统文件并启动，根据操作系统的不同，向用户显示台式计算机桌面或命令行界面。图 8 展示了 Kolibri OS 成功启动后的台式计算机桌面。

所有用户身份验证，无论成功与否，均与目标设备的 MAC 地址一同记录。图 9 展示了身份验证日志。

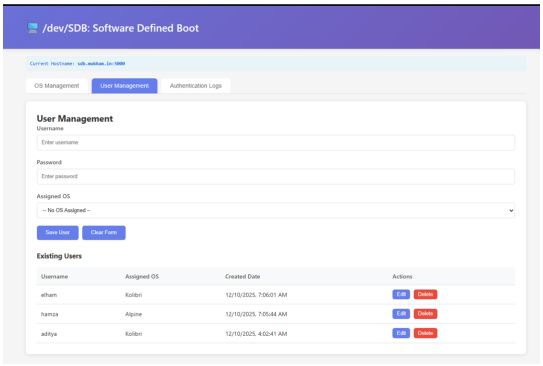


图 6. 用户管理

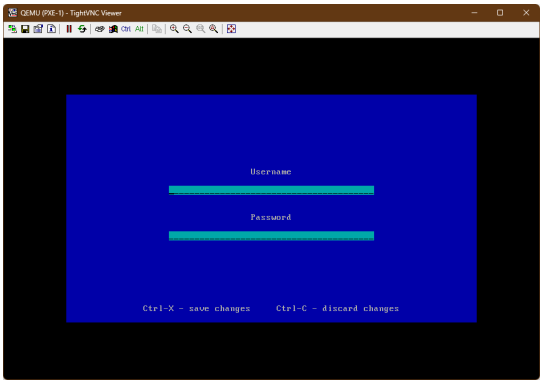


图 7. 登录提示符

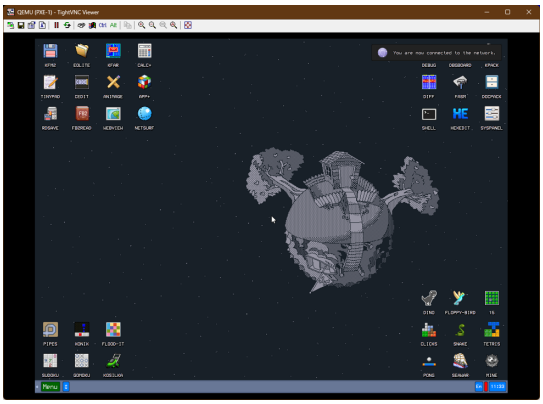


图 8. Kolibri OS 台式计算机

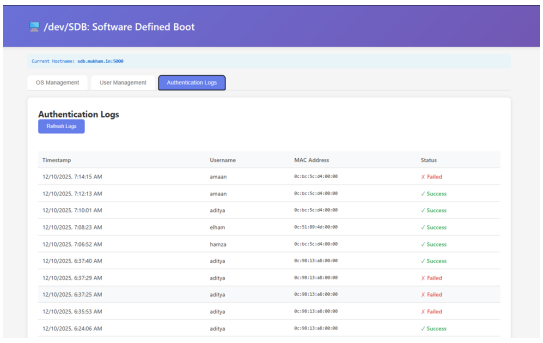


图 9. 身份验证日志

因此，该系统在模拟环境中成功部署并经过测试，结果一致。本研究特意不报告该方案的速度和性能，因为这是在理想条件下进行的模拟，且每次启动都不到 3 秒。然而，在实际环境中，启动过程可能会受到网络、内存及其他条件的影响。

V. 威胁模型

所提出的标准旨在比现有 PXE 和其他网络引导标准更加安全。威胁模型包括：

- 代理 DHCP 投毒：在标准 PXE 系统中，广播域内的恶意设备可能伪装成代理 DHCP 服务器，从而向使用 PXE 的设备提供恶意下一跳服务器（TFTP 服务器）的 IP 地址。在所提出的标准中，代理 DHCP 由连接在目标系统内部的硬件模块处理。因此，这种威胁得以缓解。
- 恶意软件（包括引导扇区恶意软件）：目标 PC 不配备持久化存储介质，因此，任何影响系统的恶意软件都会在计算机关闭后立即从内存中清除。
- 访问未经授权的操作系统：通常，安装了多个操作系统的计算机往往允许用户使用其中任意一个。相比之下，所提出的系统只会在身份验证后提供用户获准使用的操作系统。

VI. 结论与未来研究方向

/dev/SDB 有望成为一种解决方案，支持在企业远程环境中根据角色向不同用户分配软件定义的操作系统。为验证 /dev/SDB 是否切实可行，我们对适用的软件进行了大量模拟，以最低配置模拟目标电脑。所有测试均成功运行。

本研究未来的工作包括实际开发所提出的硬件模块，在模拟环境之外进行测试，并在企业环境中对其进行压力测试。此外，可根据企业的政策，将云模块部署在任何公有云或私有云中。

/dev/SDB 所提出的改进有助于减少计算机中冗余硬件的使用，同时确保每位用户都能使用为其量身定制的操作系统。

七、致谢

本研究是在应用网络安全伊拉斯谟世界联合硕士项目（CyberMACS）的框架内开展的——项目编号 101082683，由欧盟通过 Erasmus+ MUNDUS 计划共同资助。

参考文献

[1] A. Sebastian 和 K. S. Sankar，《操作系统引导加载程序的设计》，《澳大利亚基础与应用科学杂志》，第 9 卷，第 2 期，第 368–374 页，2015年2月。[2] UEFI 论坛，《统一可扩展固件接口（UEFI）规范》，版本 2 10, 2023 。3月。[3] GNU 项目，《GNU GRUB 手册 v2.12》，自由软件基金会，2023年，在线。获取地址：https://www.gnu.org/software/grub/ manual/grub/html node/Overview.html[4] Syslinux 项目，《Syslinux 和 PXELINUX 文档》，2023年，在线。获取地址：https://wiki.syslinux.org/wiki/index.php?title=SYSLINUX[5] N. Sharma 和 P. Kumar，《嵌入式系统中引导加载程序和固件组件的全面内存安全分析》，IEEE 可信与安全计算汇刊，预印本，2024 [6] K.-H. Park、S.-J. Kim、J. Yun、S.-H. Lim 和 K.-W. Park，《FLEX-IoT：面向 Flexible-IoT 平台的安全且节省资源的网络启动系统》，《传感器》，第 21 卷，第 6 期，第 2060 页，2021年3月。[7] A. Mitra、A. Ghosh、S. C. Sethuraman 和 D. Priya，《Colaboot：基于云的无盘 PC 引导机制》，arXiv 预印本 arXiv:2408.17045v1，2024年9月。[8] Jay Kantilal Bhakhar 先生、Prasad Kailas Kalamkar 先生、Om Bapu Hagawane 先生、Pavan Pralhad Aghav 先生和 Sneha B. Sahane 教授，《使用 AIO Boot Creator 和 WinUtil 的 PXE 引导服务器（预启动执行）》，IJARSCT，第 101–109 页，2024年2月，doi：10.48175/ijarsct-15521。[9] M. Johnston，《英特尔预启动执行环境（PXE）的动态主机配置协议（DHCP）选项》，RFC 编辑部，2006年11月。doi：10.17487/rfc4578。[10] M. Fuchs、L. E. Ardila-Perez、T. Mehner 和 O. Sander，《Split Boot——异构 MPSoC 上真正的基于网络的引导》，IEEE 核科学汇刊，第 70 卷，第 6 期，第 1157–1163 页，2023年6月，doi：10.1109/tns.2023.3237968。[11] G. Aryotejo 和 M. Mufadhol，《低成本实验室计算机的网络启动系统》，《物理学杂志：会议系列》，第 1524 卷，第 1 期，第 012101 页，2020年4月，doi：10.1088/1742-6596/1524/1/012101。[12] R. Stites，《Intel E810 iSCSI CHAP 与 NVM》，英特尔公司白皮书，2022年。[13] W. V. A. Oliveros、C. A. M. Festin 和 R. M. Ocampo，《使用类 BitTorrent 协议的 P2P 网络引导方案》，载于《国际 IT 融合与安全会议论文集》，2013年，第 1–4 页。[14] S. Moser，《以 bwLehrpool 和 bwForCluster NEMO 为例保障网络引导 Linux 系统的安全》，硕士论文，弗赖堡大学，2024年。[15] N. A. Schear、P. T. Cable II、T. M. Moyer、B. Richard 和 R. Rudd，《在云中引导建立并维护信任》，载于《年度计算机安全应用会议（ACSAC）论文集》，2016年，第 65–77 页。[16] A. Mohan 等，《M2：可塑金属即服务》，载于《IEEE 国际云工程会议论文集》，2020年，第 61–71 页。[17] Y. Zhang 和 Y. Zhou，《透明计算：冯·诺依曼架构在云服务中的时空扩展》，《清华大学学报（英文版）》，第 18 卷，第 1 期，第 10–21 页，2013年2月。链接：https://ieeexplore.ieee.org/document/6449403 [18] B. S. Sonawane 和 R. R. Deshmukh，《无盘工作站的启动》，《国际科学研究杂志：科学、工程与技术》，第 7 卷，第 2 期，第 290–293 页，2020年，在线，获取地址：https://doi.org/10 32628 2062118 . /CSEIT J. N. Wu、Y. H. Ko、K. M. Huang 和 M. Huang，《云操作系统上的异构无盘远程引导系统》，载于 GPC 2013，第 7861 卷，LNCS，J.J. Park 等编，2013年，第 114–123 页。链接：https://www.researchgate.net/publication/290106881 Heterogeneous Diskless Remote Booting System on Cloud Operating System [19] [20] 戴尔，《使用 BIOSConnect 启动至 SupportAssist 操作系统恢复》，戴尔支持文档，文章编号：000177771，在线，获取地址：https://www.dell.com/support/kbdoc/en-us/000177771/ – – – using-biosconnect-to-recoversupportassist-os-recovery-partition [21] 微软，《什么是 Windows 365 Boot？》Microsoft Learn。 [在线]。获取地址：https://learn.microsoft.com/en-us/windows-365/enterprise/windows-365-boot-overview