FIRM AE：面向物联网固件的大规模仿真以进行动态分析

金敏根
ETRI附属研究所
rla5072@nsr.re.kr

金东宽
KAIST
dkay@kaist.ac.kr

金恩秀
KAIST
hahah@kaist.ac.kr

金素妍
国防部c16192@kaist.ac.kr

张英珍
俄勒冈州立大学yeongjin.
oregonstate.edu

金容大
KAIST
yongdaek@kaist.ac.kr

摘要

评估嵌入式物联网设备安全性的一种方法是对其固件进行动态分析，例如模糊测试。为此，现有方法旨在提供模仿真实硬件/外围设备的行为的仿真环境。尽管如此，在实践中，这样的方法只能仿真固件映像的一小部分。例如，Firmadyne，一个最先进的工具，只能运行183（16.28%）的1，124无线路由器/IP摄像头图像，我们从前八名制造商收集。这种低仿真成功率是由真实和仿真固件执行环境中的差异引起的。

在本研究中，我们分析了大规模数据集中的仿真失败案例，以找出低仿真率的原因我们发现，尽管有不同的根本原因，但通常通过简单的启发式方法避免了广泛的失败案例，从而显著提高了仿真成功率。基于这些发现，我们提出了一个技术，nique，仲裁仿真，我们系统化的仲裁技术，以解决这些故障的几个我们的自动化原型FIRMAE成功运行了1，124个固件映像中的892个（79.36%），包括Web服务器，这比Firmadyne运行的映像多得多（4.8倍）。最后，通过在仿真图像上应用动态测试技术，FIRMAE可以检查320个已知漏洞（比Firmadyne多306⁄4），并且还在23个设备中发现12个新的0天。

CCS概念

• **安全和隐私→嵌入式系统安全;·计算机系统组织→固件**。

关键词

固件，嵌入式设备，仿真，动态分析

ACM参考格式：
Mingeun Kim、Dongkwan Kim、Eunsoo Kim、Suryeon Kim、Yeongjin Jang和Yongdae Kim。2020年。FirmAE：Towards Large-Scale Emulation of IoT Firmware for Dynamic Analysis. 在*年度计算机安全应用会议（ACSAC 2020），2020年12月7日至11日，美国奥斯汀。*ACM，New York，NY，USA，13页。https://doi.org/10.1145/3427228.3427294

ACSAC 2020，2020年12月7-11日

https://doi.org/10.1145/3427228.3427294

1引言

到2025年，活跃的物联网（IoT）设备数量预计将达到342亿[36]。随着大量物联网设备连接到互联网[33]，它们面临着网络威胁。例如，基于Linux的物联网设备（如无线路由器和IP摄像头）经常成为大规模攻击的目标在野外，从这些设备中发现了多个后门[30，38]，并且Mirai和Satori等恶意软件感染了数百万台此类设备[5，31，32，37]。

为了解决如此众多的物联网设备中的安全问题，研究人员一直专注于大规模分析这些设备的固件。具体而言，一系列研究采用了在具有虚拟硬件的仿真环境中运行设备固件的方法，然后将动态分析应用于固件[12，17，23- 25，57，60，64]。通过这种方法，不仅可以在不获得硬件的情况下动态地分析固件，而且还可以利用云基础设施来扩展安全分析。其中，Firmadyne [17]是当前最先进的固件仿真框架，旨在通过提供全系统仿真环境来实现物联网设备的大规模仿真。

问题：现实和虚拟环境的差异。该方法在实践中不是银弹，因为在全系统仿真环境中运行固件经常由于真实和虚拟仿真环境之间的不一致而失败。仿真环境中的任何差异都可能导致固件执行进入意外状态，从而导致仿真和动态安全分析失败。解决这种仿真差异是具有挑战性的，因为这种不一致源于IoT设备硬件和配置的广泛多样性。特别地，每个IoT设备配备有来自过多制造商的特定硬件设备。此外，固件通常依赖于配置向量，诸如NVRAM中的数据，并且仿真环境可能丢失这样的数据，因为数据仅在硬件中可用。这种复杂的情况与Firmadyne的仿真环境不匹配。它的模拟器QEMU [6]只支持很少的通用设备和配置，如果不花大力气模拟每个设备和配置，这个问题永远不会消失。

为了了解此问题在实践中的影响，我们从八大供应商处获得了1，124个无线路由器和IP摄像头固件映像，并使用Firmadyne运行了这些映像。结果令人担忧，因为它只能模拟其中的183个（见表1）。大部分固件映像（83.72%）未进行分析。这样的

低仿真成功率意味着尽管Firmadyne通过提供固件的全系统仿真环境而被设计为通用的，但是这种方法在实践中不起作用，需要许多人工努力来解决仿真环境中的不一致性。

激励的例子。接下来，我们将展示如何手动处理不一致性作为激励示例。首先，我们使用Firmadyne运行D-Link DIR-505 L的固件以测试CVE-2014-3936 [16]。由于该漏洞是在固件上运行的Web服务中的基于堆栈的缓冲区溢出，因此攻击需要通过模拟环境的网络接口发送HTTP请求但是，当我们在Firmadyne上运行固件时，尽管Web服务器运行正常，但我们无法连接到Web服务。从我们的分析中，我们发现固件中的网络配置与模拟环境不匹配，在我们强制配置网络后，我们能够触发漏洞。其次，我们使用Firmadyne运行NETGEAR R6250的固件来测试CVE-2017-5521 在这种情况下，仿真失败，引导过程中出现内核死机。 在我们稍微修改了引导和内核相关的配置以匹配虚拟环境之后，我们能够运行固件并触发漏洞。

观察和目标。从这两个示例中，我们观察到，容易应用的配置或设备设置的轻微改变可以让固件仿真运行而不会遭受难以处理的仿真差异问题。在这方面，我们认为Firmadyne*错过了许多模拟和分析物联网固件映像的机会，不是因为模拟中的基本问题，而是因为设备设置失败，尽管这些可以轻松处理*。为了解决这个问题，我们的目标是通过分析许多仿真失败的情况下，在systematizing这样的启发式，并最终，我们的目标是增加成功的固件仿真比Firmadyne的机会。

我们的方法。我们通过调查许多仿真失败的情况下，作为我们的第一步，实现这一目标。为了进行调查，我们从前八名供应商收集了1124个固件映像[59]：1079个无线路由器和45个IP摄像机。对于仿真，我们特别关注
仿真无线路由器和IP摄像头的Web服务这

是因为Web界面是远程攻击者可以与之交互的部分，并且在这些服务中发现了许多关键漏洞[5，7，12，32，51]。通过使用Firmadyne，我们调查了437个仿真失败案例（在AnalysisSet中的527个固件映像中），发现大多数案例属于以下五类问题：1）与引导相关的问题，例如不正确的引导顺序或缺少文件，2）与网络相关的问题，例如网络接口不匹配或配置不正确
3) 与非易失性RAM（NVRAM）相关的问题，例如缺少库函数或自定义格式，4）与内核相关的问题，例如不支持的硬件或函数，以及5）小问题，例如不支持的命令或定时问题。

我们的调查结果表明，*每个类别中的故障案例都可能通过应用简单的启发式方法来解决这些问题，即使它们源于不同的根本原因*。例如，有227个图像无法设置其网络接口，即使其Web服务器
正确运行。虽然故障的根本原因可能
变化，如可用网络端口数量的差异，[1]http://github.com/pr0v3rbs/FirmAE

网络设备的名称等，强制设置在仿真环境中工作的网络配置的试探法可以解决该问题并实现动态分析。

基于这一观察，我们系统化这些启发式作为一种技术，创造**仲裁仿真**，并开发了几种仲裁技术，绕过失败的情况。代替严格遵循固件的执行行为，仲裁仿真在遵循原始行为或注入适当干预之间进行仲裁，即，有意的操作。因此，它可以稍微改变固件的原始行为然而，我们的目标不是构建与物理设备相同的环境，而是创建有利于动态分析的环境。事实上，我们的方法可以模拟以前的方法无法模拟的许多固件映像，并有效地帮助找到真正的漏洞。

在设计了几个仲裁后，我们自动化和并行化整个固件仿真过程。在测试1，124个固件映像的4小时内，我们的原型F irmAE成功仿真了892个（占总数的79.36%）映像，是Firmadyne的四倍多（表1）。然后，我们在仿真图像上运行了以前已知的漏洞，以验证仲裁仿真是否适用于动态分析。因此，在FirmAE上成功模拟了320个已知漏洞，比Firmadyne多306个成功案例。我们还在F i rmAE上构建了一个简单的模糊器，并在95个最新设备中发现了23个独特的漏洞，并负责任地向供应商报告。

总之，我们的研究贡献如下：

- 我们凭经验调查437固件仿真失败的情况下，系统化的故障处理启发式。
- 我们提出仲裁仿真应用这些启发式仿真环境。我们的原型，FLRMAE，呈现出高得多的仿真成功率（892 *对892*）。*183*）比最先进的框架Firmadyne。
- 我们通过redis确认仲裁模拟是有效的-比Firmadyne多覆盖306个已知漏洞此外，使用简单的模糊器，FIrmAE可以在95个最新设备上找到23个新漏洞，其中12个是
  0-大。 1
- 我们发布源代码以鼓励未来的研究。

2 背景

在本节中，我们将通过引用以前的研究来解释如何分析嵌入式设备，并介绍我们采用的最先进的工具作为我们方法的基础

2.1 嵌入式设备分析过程

为了分析嵌入式设备，可以在有/没有物理设备的情况下获得和分析

固件收集和解包。通常，固件可以从供应商的网站、ftp服务器或第三方存档中获取。这可以手动完成，也可以使用网络爬虫，如Spider [41]。固件也可以直接从设备中的闪存中转储[46]，尽管这需要物理设备。

然后解包固件映像以供以后分析。单个图像可以包括多个内容。例如，基于Linux的固件可以具有引导加载器、内核和文件系统。此映像通常以各种方式压缩，例如LZMA、ZIP或Gzip，以节省存储空间。要解压缩映像，通常使用Binwalk [26]，Firmware-Mod-Kit [27]或FRAK [13]等工具在给定的图像中，这些工具扫描各种文件头的预定义签名。当签名匹配时，他们从图像中提取文件，并继续扫描到最后。也存在加密或定制的图像，无法使用签名匹配;分析它们不在本研究的范围内。

使用物理设备进行分析解包后的固件可以用实际设备进行分析。Zaddach等人[62] Mariu*set al.*[44]使用JTAG接口将进程执行和外围设备访问中继到真实设备和部分仿真的目标代码。类似地，Kammerstette*r et al.*[28，29]开发了一个使用真实设备的代理环境，并将角色设备访问转发给它们。Cui等[14，15]和Kumar*et al.*[33]对连接到公共互联网的嵌入式设备进行了定量研究。

分析w/o器械。另一个研究流集中在分析没有物理设备的固件以扩大分析。研究人员在固件上采用静态方法[11，52];然而，由于缺乏运行时信息，它们经常产生许多误报然而，Costin*et al.*[11]显示了易受攻击的设备的统计数据，这些设备具有容易破解的密码或后门字符串。Shoshitaishvili等 [52]发现使用符号执行的身份验证绕过漏洞。

相比之下，动态分析可以识别漏洞而不会误报，因为它直接运行目标程序然而，执行动态分析不是简单的任务，因为必须仿真设备固件。 最近的研究[12，17，23- 25，57，60，64]集中在固件仿真上，以克服获得真实硬件的困难，我们在以下小节（§2.2）中进一步详细描述了这些研究。

2.2 仿真分析

固件仿真已经引起了人们的注意，因为它不需要真实的设备，并提供了有用的接口进行动态分析。发生仿真的系统被表示为*主机*系统，并且被仿真的系统被称为*客户*系统。通常，有两个仿真级别：用户级和系统级。

用户级仿真。用户级仿真只仿真固件中的目标程序，并充分利用主机系统。一个例子是模拟Web界面。Web界面是嵌入式设备中用于设备管理或维护的代表性服务它提供多种静态内容，如HTML，或由CGI程序生成的动态内容虽然静态内容可以与主机环境一起提供，但是动态内容不可以。 这是因为它们可能与主机系统冲突，或者依赖于主机系统中不存在的自定义库和设备驱动程序。

系统级仿真。系统级仿真完全仿真客户系统，包括内核。 由于它提供了一个独立的执行环境，内核和设备驱动程序中的各种功能也可以被模拟。然而，固件仿真是极其困难的，因为供应商特定的硬件问题或存储器映射的外围设备。如果不处理它们，仿真固件中的程序经常崩溃。

因此，最近的研究一直在努力解决这些问题[12，17，23，25，57]，通过创建尽可能类似于真实设备的仿真环境。流行的仿真器，如QEMU [6]，已经支持更多的硬件类型，包括它们的外围设备。Costin等人 [12]提出了一个可扩展的动态分析框架，以及对各种嵌入式Web接口的几个案例研究。Chen等人 [17]仿真非易失性RAM（NVRAM），它存储仿真固件中程序的各种配置值。Gustafson等人[25]外设通信中的Feng等 [23]试图用机器学习解决同样的问题。最近，Clements等人 [10]建议将硬件与固件解耦。

分析。在仿真之后，可以通过使用先前已知的PoC代码[17]或模糊器[24，60，64]来检查漏洞。TriforceAFL[24]是一个流行的模糊器，目标是QEMU图像，杠杆老化美国模糊lop（AFL）[63]。它也被Hu 等人采用。[60]。在他们的后续研究中，Zheng*et al.*[64]提出了一种用于动态分析的优化仿真方法，该方法在系统级仿真和用户级仿真之间切换上下文。

2.3 固件仿真中的挑战

基于仿真的分析是有利的;然而，在仿真来自不同供应商的固件映像时存在大量挑战，这些挑战源于非标准化的开发过程以及仿真环境与物理环境之间的差异。例如，库、设备驱动程序，甚至设备中的内核在供应商之间都是不同的;除非这些被正确地仿真，否则内部程序无法执行。

访问硬件接口的设备，如LED传感器或摄像头，具有更多的多样性，如以前的研究所指出的[23，25]。主设备与其外围设备之间的通信通常利用具有预定义存储器地址的存储器映射IO（MMIO）操作。然而，这种地址的范围在设备之间显著不同。因此，难以将该方法扩展到各种设备。Chen等人[17]试图在大规模上模拟一种这样的硬件，即NVRAMMuench等人[45]强调了在进行动态分析以识别内存损坏漏洞时特定于设备的挑战

解决这些挑战可能是不可行的，除非在物理设备中完美地实现功能。然而，调查仿真失败案例并解决已识别的问题有助于逐步提高仿真率，并实现动态分析，以提高物联网生态系统的安全性。因此，我们采用了最先进的仿真框架Firmadyne [17]，并研究了故障情况。

2.4 Firmadyne框架

Firmadyne [17]是最先进的固件仿真框架，最初设计用于大规模分析。 许多研究[24，60，64]已将其用于动态分析。 我们还利用Firmadyne进行失效调查。

在解包固件映像后，Firmadyne使用定制的Linux内核和库对其进行仿真，这些内核和库预先构建为

支持各种硬件功能，如NVRAM。对于仿真，Firmadyne会对目标图像进行两次仿真：第一次仿真记录有用信息，而第二次仿真则利用记录的信息。因此，定制的内核包括一个驱动程序，该驱动程序钩住主要的系统调用以记录有用的信息。例如，它们挂钩inet_ioctl（）和inet_bind（）以获取仿真固件中使用的网络接口的名称和IP地址。Firmadyne的自定义库还解决了硬件问题。例如，库libnvram基于硬编码的默认值存储和返回NVRAM值。

虽然Firmadyne很有前途，但其网络可达性和Web服务可用性的仿真率相当低，分别为29.4%和16.3%。为此，我们仔细研究了失败的情况下，并提出了一种技术来解决这些问题。

3 设计

3.1 目标和范围

进球了我们的目标是成功地模拟嵌入式设备的固件映像，特别是运行其Web服务，因为这些设备的Web接口是远程攻击者的关键目标。[5、12、17、32、60、64]。我们不打算解决仿真环境中的所有差异。相反，我们的目标是一个简明的仿真动态测试，我们的仿真目标可以说明以下属性：1）引导没有任何内核恐慌，2）从主机的网络可达性，和3）Web服务的动态分析的可用性。我们的目标是保持这些属性，因为它们是运行Web服务的最低要求，而不会在固件仿真中遇到问题。因此，我们通过检查目标固件的网络可达性和Web服务可用性来检查仿真成功率

范围。在各种嵌入式设备中，我们选择无线路由器和IP摄像头作为我们的分析目标，因为它们存在于我们的日常生活中，并且经常成为攻击目标。事实上，许多僵尸网络[5，32]都以它们为目标，发动大规模的DDoS攻击。请注意，共享类似特性的其他嵌入式设备也可以用我们的方法来解决

3.2 仲裁仿真

为了实现这一目标，我们提出了一种技术，我们称之为*仲裁仿真*。尽管先前的方法[12，17，23，25，57]已经努力确保目标固件像物理设备一样操作，这是一个困难的目标，但仲裁仿真并不完全遵循目标固件的原始执行过程。仲裁仿真背后的关键思想是确保高级行为足以对内部程序执行动态分析，这相对容易做到，而不是找到和修复仿真失败的确切根本原因这里提到的高级行为可以由熟练的分析人员根据他们的目标和仿真目标轻松建模。在本研究中，我们使用§3.1中定义的模型。

仲裁仿真的一个关键特征是它采用了*干预*。干预指示有意添加的动作，其可以不同于物理设备的行为。此操作使得可以绕过未解决的问题，假设它们不会强烈影响仿真固件内的目标程序的行为。仲裁程序

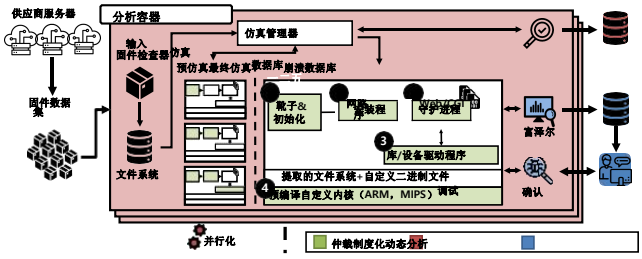

图1：FirmAE架构概述

在按原样遵循固件和应用干预之间的选择被称为*仲裁*。根据需要，干预可以以各种方式实现，并且它可以被注入仿真过程的适当步骤，即*仲裁点*。可以通过分析给定的高级行为模型的违反情况来注意适当的仲裁点。然后，在这些仲裁点中注入干预。由于干预措施集中在高级行为上，因此从一小组固件映像中获得的干预措施可以广泛应用于遭受类似故障情况的其他固件映像，即使它们具有不同的根本原因。

我们的干预措施利用基于Linux的固件的抽象设计。我们对数据集进行了初步研究，发现适当的干预可以帮助模拟器绕过许多未解决的问题。例如，当网络设置过程由于未知外围设备访问或NVRAM支持不足而停止时，强制配置固定网络设置的干预可以解决问题，而不管根本原因如何。虽然仲裁仿真可能会违反全系统仿真的主要概念，我们hypothe- esized的干预措施引入的小差异只有轻微的影响目标程序的行为。事实上，我们通过在1，124个映像中的892个固件映像中成功运行模拟Web服务来支持这一假设，并且我们通过进行动态安全分析发现了12个0天漏洞。

3.3 FirmAE

我们基于Firmadyne [17]实现了我们的仲裁仿真的原型FirmAE。在图1中示出了FmAE的整体架构。FmAE在预构建的定制Linux内核和库上模拟类似于Firmadyne的固件映像，如§2.4中所述。它还模拟目标映像两次，以收集各种系统日志并利用这些信息进行进一步的仿真。我们将前一个仿真步骤称为预仿真，将后一个仿真步骤称为最终仿真。FirmAE中应用的仲裁可以分为五类，这是通过我们在AnalysisSet上的失败案例调查得出的。我们在第4节中描述了每个仲裁的细节，并在第5.1节中将仿真结果与Firmadyne的仿真结果进行了比较。我们构建了用于对FirmAE进行动态分析的附加接口（§5.3），分析结果在§5.4中描述。

自动化。对于大规模的分析，需要完全自动化FirmAE。当然，Firmadyne的许多步骤都是自动化的;但是，它仍然需要一些用户交互。例如，用户必须首先提取具有特定选项的目标固件的文件系统然后，他们评估文件系统是否被成功提取并检索体系结构信息。随后，他们为QEMU制作固件映像并收集信息

在预仿真中。最后，他们运行脚本进行最终仿真并执行动态分析。我们自动化了所有这些交互，并添加了一个网络可达性和Web服务可用性的自动评估过程。为此，我们在FirmAE中构建了一个模块，该模块定期运行ping和curl命令。

并行化。我们还并行化了仿真，以有效地评估大量固件映像，利用Docker的容器化[40]。每个固件映像在每个容器中独立仿真，该容器配备了所有必需的包和依赖项。这使得能够快速且可靠地仿真目标图像。FIRMAE通过运行多个容器实例来并行仿真固件。

通过容器化，我们可以利用抽象主机和客户系统之间的网络连接FMAE用于仿真的QEMU[6]在主机系统中创建附加的网络接口*TAP*。此接口链接到来宾网络接口之一。因此，每个仿真固件应该具有独立的TAP接口，该TAP接口在主机系统中具有唯一的IP地址，否则将发生网络冲突。容器化隔离了每个容器的网络环境因此，即使在并行仿真中，来自主机系统的分组也可以被正确地路由到客户机。我们还将检查器和分析引擎放置在每个容器内。

3.4 实验装置

数据集。我们的数据集包括无线家庭路由器市场的前八名供应商[59]。我们从供应商的网站上收集了1306个固件映像，并通过使用Binwalk [26]解包从收集的映像中提取文件系统，如§2.1所述。然后，我们通过验证每个映像的操作系统是否具有以下三种架构之一来过滤它们：ARM little endian（ARMel），MIPS little endian（MIPSel）和MIPS big endian（MIPSeb）。这些架构占据了我们初始集合的97%以上我们以同样的方式准备了IP摄像机固件

我们的最终数据集包括总共1124个固件图像，其中1079个是无线路由器，45个是IP摄像机。我们将它们分为三个数据集：AnalysisSet，LatestSet和CamSet。它们的简要摘要和仿真结果在表1中给出，其详细版本在附录中的表4中示出。AnalysisSet包含来自3家供应商的526个过时映像，而LatestSet和CamSet仅包含截至2012年12月的最新固件映像2018年。LatestSet包含来自8个供应商的553个最新图像，包括AnalysisSet涵盖的供应商，CamSet包含来自3个供应商的45个最新图像。因此，AnalysisSet可以包括每个设备的多个固件版本，而LatestSet和CamSet每个设备仅具有一个图像数据集之间没有交集，即，它们不共享任何相同的图像。我们使用AnalysisSet来分析仿真失败的案例。通过分析它们，我们发现了几个仲裁点，可以帮助提高仿真率（§4）。我们在FIRMAE中应用了这些仲裁，并使用LatestSet和CamSet对其进行了评估（§5）。

环境。我们所有的实验都是在配备四个Intel Xeon E7- 8867 v4 2.40 GHz CPU、896 GB DDR4 RAM和4 TB SSD的服务器上进行的。我们在服务器上安装了Ubuntu 16.04和Post-greSQL v9.5.14 [42]和Docker v18.09.4 [40]。

4 破产案件的仲裁

仲裁仿真的关键是描述仲裁点，以帮助仿真器绕过故障。因此，我们首先基于高级行为模型分析了AnalysisSet上的失败案例。对于大规模的分析，我们应用了FirmAE的自动化和并行化而没有任何仲裁，使得仿真部分与Firmadyne的仿真部分值得注意的是，只有16.9%的图像的Web服务器被模拟（§5.1）。为了简洁地解释，我们根据仲裁点对失败情况进行了分类：引导（§4.1）、网络（§4.2）、NVRAM（§4.3）、内核（§4.4）和其他（§4.5）。在本节中，我们将详细解释它们。

注意确定仲裁点和设计适当的干预需要实证调查，我们相信我们的研究可以有助于在这一领域的未来研究

4.1 引导仲裁

我们在引导过程的早期遇到了第一个问题，这使得模拟失败并导致内核死机。

引导顺序不正确。引导顺序不正确的主要原因是用于系统初始化的程序没有正确执行。通常，大多数系统在引导过程中需要初始化 在Linux内核中，初始化通常由一个名为init的程序执行，内核试图通过检查预定义的路径（例如/sbin/init、/etc/init和/bin/init）来查找该程序。然而，一些固件映像具有用于初始化程序的定制路径，使得内核无法执行程序并崩溃。

此故障经常发生在NETGEAR固件映像中。在分析它们之后，我们发现它们使用的名称是preinit，这是一个开源嵌入式设备项目Open-Wrt [22]经常使用的名称，我们验证了它们确实是在其上实现的。我们还发现一些TP-Link映像也使用preinit为了解决这个问题，Firmadyne构建了一个脚本，用于搜索和执行一个硬编码的文件列表，这些文件经常被访问以初始化程序。然而，这些候选项不足以说明在野外初始化程序的不同路径。

我们提出了另一种方法，利用从目标固件的内核的信息。具体来说，我们在启动过程开始时创建了一个干预，它可以提取映像内核中的有用信息。具体来说，我们利用内核的命令行字符串，这是请注意，这样的字符串是在开发阶段预定义的，因此它自然嵌入到内核映像中。此信息可能包括初始化程序路径、控制台类型、根目录、根文件系统类型或内存大小。例如，从NET-GEAR固件中的一个内核映像中，我们可以获得console= ttyS 0，115200root=31：08 rootfstype= squashfsinit =/etc/preinit的字符串。我们可以识别/etc/preinit的初始化程序路径、具有115200波特率的ttyS0的控制台类型和squashfs的根文件系统类型。通过配置仿真环境的信息，从原始内核中获得的，客户系统可以正确地初始化没有失败，即使初始化程序有不寻常的路径。如果我们无法提取任何信息，我们会从提取的文件系统中找到preinit或preinitMT等初始化程序。

缺少文件系统结构。由于缺少文件或目录而发生其他失败情况。当内部程序访问这样的路径时，它们会崩溃，并且仿真停止。Firmadyne试图通过在自定义引导脚本的开头创建和安装硬编码路径（如proc、dev、sys或root）来解决此问题一些硬编码的路径当然有效;例如，制作/etc/TZ或/etc/hosts帮助解决了几个这种故障。然而，这种方法不能解释不同的情况。此外，由于它在固件初始化自身之前强制创建文件和目录，因此它与内部程序发生冲突，这些内部程序在相同路径中创建和安装其他文件或目录。

我们通过插入一个干预来对此进行仲裁，这与前面的情况类似，但是从文件系统而不是内核检索信息在模拟给定的映像之前，我们从其文件系统中的可执行二进制文件中提取所有字符串。然后，我们对它们进行过滤，以获得极有可能指示路径的字符串，并根据路径准备文件结构。特别地，我们选择了以通用Unix路径开头的字符串，例如/var或/etc。

4.2 网络仲裁

在引导过程完成之后，网络应当被配置成使得主机系统可以与客户系统通信，并且最终可以执行动态分析。对于网络通信，QEMU要求主机创建一个额外的网络接口*TAP*。此TAP接口连接到客户系统中的网络接口然后，主人和客人通过它进行交流。

但是，正确配置TAP接口并不简单，因为它应该使用与目标网络接口类型对应的特定选项进行设置.此网络接口类型可以是以太网、无线LAN（WLAN）、网桥或虚拟LAN（VLAN）。由于静态区分客户系统中的接口类型并不容易，因此需要对目标映像进行一次仿真。

Firmadyne对给定图像进行两次仿真（第2.4节）。在第一个仿真，即预仿真，Firmadyne收集内核日志通过挂钩的系统调用。由于收集的日志包括在仿真期间访问的网络接口的名称和IP地址，因此它们可以用于最终仿真中的网络配置尽管如此，许多图像仍然失败。

IP别名处理无效将多个IP地址分配给单个网络接口称为IP别名[58]。它在路由器中很普遍，因为它可以通过IP地址单独管理服务。在IP别名中，一个网络接口创建多个自身实例，每个实例都分配有一个唯一的IP地址。例如，IP地址为www.example.com的网桥接口br0192.168.1.1可以具有169.254.39.31.1.1.1分别分配给其实例br0：0和br0：1的IP别名www.example.com和www.example.com。然后，br0链接到以太网接口eth0。在这里，可以使用这些IP地址中的任何一个来访问br0

与此IP混叠相关的故障案例经常出现在D-Link图像中。经过调查，我们发现这些问题是由Firmadyne没有正确处理IP混叠造成的在主机系统中配置Firmadyne网络时出现问题。在预仿真步骤中，内核记录IP别名。然后，Firmadyne解析日志并尝试分配所有记录的IP地址到来宾中的相应接口然后，它为这些IP地址添加静态路由规则，以将它们链接到主机中的TAP接口这里，多个路由规则被添加到单个TAP接口，这使得网络冲突。

在了解IP别名的情况下，FIrmAE通过让主机系统使用其默认路由规则来对此进行仲裁特别地，即使使用IP别名，一旦客户机因此，这些病例不需要干预，证明了在适当情况下进行干预的重要性。

无网络信息。某些固件映像在其内核日志中不包含有关可连接网络接口（例如eth）的任何信息。这些映像仅配置环回接口（lo），而不设置其他网络接口。由于缺乏可连接的网络接口，无法从主机系统访问这些图像。此外，一些图像试图将其Web服务器绑定到不存在的网络接口，并因此崩溃。

在分析案例后，我们发现一些映像使用动态主机配置协议（DHCP）从DHCP服务器检索IP地址，用于其WAN接口。DHCP是在端点设备中设置网络接口的流行协议，因为它不需要任何用户交互。通常，无线路由器本身充当DHCP服务器，将IP地址分配给其客户端所连接的LAN接口然而，它们也可以从外部DHCP服务器检索IP地址，以将它们的WAN接口连接到Internet，除非用户手动配置它。事实上，我们分析的图像试图通过它们的WAN接口和主机系统的TAP接口之间的连接，用DHCP检索IP地址。然而，由于DHCP服务器不存在于仿真环境中，所以仿真固件无法获得IP地址并配置网络接口。此外，由于没有配置网络接口，因此不能布置将多个网络接口分组的桥接接口。因此，绑定到这些网络接口的内部程序无法正常运行。

我们首先尝试使用QEMU的内部DHCP服务器来解决这个问题，这样客户机的网络接口就可以从服务器检索IP地址。但是，即使设置了DHCP服务器，某些映像仍然没有网络接口这可能是由于外围设备支持如果在网络配置期间任何程序访问这样的外围设备，则其崩溃或异常动作，并且最终无法配置网络。

FIR利用强制地用默认设置配置网络的干预来仲裁这些情况。具体来说，我们设置了一个以太网接口eth0，其IP地址为192.168.0.1。设置以太网接口后，它将与内核日志包含桥接接口信息的映像的默认桥接接口br0链接。这种简单的干预可以显著地帮助模拟Web服务（第5.1节）。

ARM中的多个网络接口为了支持多个网络接口，必须选择一个合适的机器，在该机器上将加载目标固件。我们选择了virt，QEMU支持的机器之一，遵循以前研究中采用的方法[17]。这对于几个

固件映像;但是，它无法模拟具有多个网络接口的ARM固件映像。Firmadyne尝试通过准备固定数量（四个）的虚拟接口来解决此多接口问题。它的基本假设是接口的数量应该大于或等于接口名称的后缀，该后缀从内核日志中提取。例如，如果记录了eth1，则很可能也存在eth0然而，几乎所有的ARM镜像仍然没有被仿真。

我们仔细调查了这些病例，但无法确定确切原因。然而，我们可以通过高级干预来解决故障，强制只设置一个以太网接口。更具体地说，我们的干预强制设置了一个以太网接口eth0，并避免设置其他接口。因此，我们设置一个网桥网络接口，并在必要时将其链接到主机。通过这种干预，可以仿真大部分ARM固件映像。

VLAN设置不足 VLAN是路由器的典型功能，因为它提供了一个隔离的网络环境，在逻辑上对子网进行分组。VLAN接口与其他网络接口（如以太网或WLAN）具有不同的特性，因此必须使用其他选项进行设置。 要支持VLAN，TAP接口的类型应设置为VLAN，并为其分配适当的VLAN ID。

另一个故障发生在具有VLAN接口的固件映像在模拟这些映像时，即使以太网接口正确配置了独立的IP地址，来宾网络也无法访问。 Firmadyne尝试通过在设置主机TAP接口时运行命令来解决此问题;但是，其配置不足以处理此问题。特别是，VLAN应设置为将具有相同VLAN ID的主机和访客网络分组。但是，Firmadyne拒绝设置主机网络。通过正确地配置VLAN，MAC对此进行仲裁。

iptables中的过滤规则 许多路由器通过设计设置防火墙来防止未经授权的远程访问。否则，攻击者可以访问管理界面。我们数据集中的一些固件映像也通过使用iptables实现了此策略。因此，客户内核丢弃来自主机的所有数据包。我们在TP-Link中发现了大多数这种情况，即使主机和来宾网络配置正确，来宾也无法访问。

这并不表示模拟失败，因为设置ipta- bles模拟真实设备的原始行为然而，这种过滤阻止了对其潜在漏洞和威胁的分析显然，在分析过程中发现的漏洞可能无法远程利用。然而，许多设备所有者或管理员错误地更改了这些规则，使设备可以公开访问[14，15，51]。

过滤AE通过检查客户系统中的过滤规则并在它们存在的情况下移除它们来对此进行仲裁。这可以简单地通过刷新iptables中的所有策略并设置默认策略以接受所有传入数据包来完成然后，访客网络变得从主机可达，并且可以进行动态分析

4.3 NVRAM仲裁

模拟与真实环境相似的外围设备是固件模拟中最具挑战性的部分之一（§2.3）。NVRAM本质上是闪存，是广泛使用的外围设备之一。在嵌入式设备中用于存储配置数据。嵌入式设备中的内部程序经常在其中存储/从中获取必要的信息。除非NVRAM受支持，否则这些程序通常会崩溃Firmadyne实施了一个自定义NVRAM库，以模拟NVRAM相关功能。通过设置名为LD_PRELOAD的环境变量，可以预先加载此自定义库以包含其他库。 这将拦截与NVRAM相关的函数（如nvram_get（）和nvram_set（）），并在没有物理访问的情况下模拟NVRAM。具体地说，当调用nvram_set（）时，键值对存储在文件中，稍后调用nvram_get（）时将检索键值对。对于这些情况，在调用nvram_set（）之前调用nvram_get（），Firmadyne使用给定固件中的默认文件初始化键值对，这些文件通常用于设备的出厂重置功能Firmadyne有一个默认文件的硬编码路径列表，用于提取键值对。然而，在这方面，我们的数据集中的许多固件映像仍然没有被仿真。

支持自定义NVRAM默认文件。我们发现了许多情况，其中默认文件的路径因设备而异，甚至它们的键值对也有不同的模式。例如，在某些D-Link映像中，默认文件位于/etc/nvram.default或/mnt/nvram_rt.default。此外，一些NET-GEAR映像中的默认文件位于/usr/etc/default。这些文件中的键-值对用不同的分隔符分隔，例如回车或NULL字节。一些默认文件甚至具有供应商特定的格式，例如OBJ或ELM。

为了开发可扩展的方法，FIRM在预仿真期间准备仲裁。具体地，FIRMAE记录在预仿真期间用nvram_get（）和nvram_set（）函数访问的所有键值对。然后，它扫描目标固件的文件系统，并搜索包含记录的键值的多个实例的文件FIR从文件中提取键值对（如果它们存在的话）并且在最终仿真中利用它们。

无NVRAM默认文件。不幸的是，并非所有固件映像都具有默认NVRAM文件。 即使存在默认文件，它也可能不包含请求的键值对。解决此问题的一个简单方法是为未初始化的键返回NULL值，就像Firmadyne所做的那样。然而，我们观察到许多情况下，在nvram_get（）返回NULL之后，由于分段错误而崩溃。通过对崩溃的程序进行逆向工程，我们发现，令人惊讶的是，许多程序都没有验证nvram_get（）的返回值。它们只是将返回值传递给与字符串相关的函数，如strcpy（）或strtok（），并因NULL指针解引用而崩溃。

FIRMAE通过仲裁nvram_get（）函数的行为来处理这一点当访问未初始化的键时，FIrmAE返回指向空字符串的指针，而不是返回NULL值。 这个简单的改变显著地减少了崩溃，特别是在NET-GEAR映像中. 因为我们不能获得真正的键值对与物理设备，这将是最佳的approaches之一，以避免在许多内部程序的错误处理不足造成的崩溃。

4.4 内核仲裁

嵌入式设备中的许多程序通过内核中的设备驱动程序与外围设备协作。通常情况下，他们会交流

外围设备使用ioctl命令。不幸的是，仿真，ING这个过程不是一个简单的任务，因为每个设备驱动程序有不同的特点，这取决于它的开发人员和相应的设备。虽然Firmadyne实现了一些虚拟内核模块，支持/dev/nvram和/dev/acos_nat_cli，但它无法涵盖实际场景中固件映像的各种特征。我们数据集中的许多固件映像也会因此问题而崩溃。

内核模块支持不足 由于Firmadyne使用硬编码的设备名称和ioctl命令实现虚拟模块，因此某些程序在访问具有不同配置的内核模块时会失败。例如，许多NETGEAR映像使用一个名为acos_nat的模块，该模块用于与安装在/dev/acos_nat_cli上的外围设备通信。在这些图像中，Firmadyne模块返回错误的值，并导致httpd的Web服务上的无限循环。此外，我们发现ioctl命令因固件体系结构而异，因此也应该考虑这一点。

FimAE这里的关键直觉是，许多内核模块是通过共享库访问的，这些共享库具有发送相应ioctl命令的函数。因此，与处理NVRAM问题类似地，FIR拦截库函数调用（§4.3）。当程序调用库函数时，FIRMAE返回预定义值。因此，不需要根据设备架构来模拟每个ioctl命令在这个例子中，我们只关注acos_nat，而通过共享库的其他外围设备访问可以以相同的方式处理。

内核版本不正确。我们发现一些固件映像面临的问题与内核版本。Firmadyne在固件仿真中定制了Linux内核v2.6.32然而，最近的嵌入式设备使用了更新版本的内核。升级内核版本似乎是这个问题的一个简单的解决方案。事实上，我们实验性地测试了Linux内核v4.1.17，并成功地模拟了更多的固件映像。但是，一些固件映像，特别是较旧的固件映像，没有在新版本的内核中进行仿真。这些映像失败，libc库崩溃

我们调查了这些情况，并确定Linux内核v4.1.17的地址空间布局随机化与旧版本的libc不兼容。为了解决这个问题，我们在编译新内核时使用了兼容性选项。具体来说，我们设置CONFIG_COMPAT_BRK选项，它排除了堆内存中随机化的brk区域有了这个新内核，FlrmAE能够处理上述情况。可能存在我们的实验中未检测到的其他兼容性问题为了解决这些问题，应该进一步测试具有各种编译选项的多个内核版本，这是我们未来研究的目标之一。

4.5 其他仲裁

一些失败案例通过其他次要干预措施解决

未执行的Web服务器。对于Web服务的动态分析，我们需要同时实现网络可达性和Web服务可用性。在某些映像中，即使网络配置成功，Web服务器也不会运行 我们无法找到这种现象的确切根源。但是，强制执行Web服务器的干预可以解决这个问题。具体来说

它在目标固件的文件系统中搜索诸如httpd、lighttpd、boa或goahead的广泛使用的web服务器以及它们相应的配置文件，并执行它。

超时问题。应强制停止长时间不响应的模拟固件映像。因此，需要设置合适的超时。Firmadyne应用程序使用60秒超时;然而，固件映像（尤其是来自NETGEAR的固件映像）需要很长时间才能完成其引导过程，因此最终会阻止其仿真。我们调查了这种情况，并根据经验发现了240秒的合适超时 虽然这一更改很简单，但成功模拟了60多个固件映像。

缺乏仿真工具嵌入式设备开发人员通常省略不必要的功能以节省存储。因此，固件映像可能没有适当的工具来仿真其自身。由于模拟环境没有任何存储限制，我们可以添加几个必需的工具。 为了成功模拟，应该在文件系统中准备几个Linux命令，如mount或ln。 我们通过将最新版本的busybox添加到目标固件的文件系统中来解决这个问题。 这个简单的添加启用了基本命令，并导致成功的仿真。

5 评价

从对AnalysisSet的调查中，我们发现了几个仲裁点（§4）。 在本节中，我们在我们的数据集上用我们的原型FLRMAE（§3.3）评估每个仲裁。为此，我们在Python和shell脚本中实现了总共3671个LoC 我们还介绍了在动态分析与FIrmAE的漏洞。

5.1 固件仿真结果

我们在每个数据集上比较了FmAE和Firmadyne的仿真率（§3.4）。所有数据集的总仿真时间小于四小时（14289s），因为FLRMAE支持完全自动化和并行化（§3.3）。

总体结果。 由于我们的目标是模拟Web服务以进行动态分析（第3.1节），因此我们验证每个模拟固件的网络可达性和Web服务可用性。此后，我们将Web服务可用性称为仿真率。 最终结果列于表1中。总体而言，仿真率从1628%显著增加到79.36%（增加了487%）。 由于我们的调查是基于AnalysisSet的，它显示了最高的91.83%。与Firmadyne获得的数据相比，LatestSet和CamSet的比率也有很大的提高，我们可以识别其中的漏洞（第5.3节）。在AnalysisSet中，NETGEAR图像的仿真率增加最多，从10.95%增加到93.80%（857%），这是由于解决了ARM网络问题的干预措施，因为大多数NETGEAR图像都是基于ARM的。在LatestSet中，TRENDnet、ASUS、Belkin和Zyxel的仿真率低于60%;这些较低的仿真率归因于这些镜像中大量的内核模块和自定义硬件接口的使用。 我们在§5.2中详细描述了这一点。

CamSet的仿真率表明，解决无线路由器的故障问题也有助于仿真IP摄像机。特别地，没有一个D-Link图像是用Firmadyne模拟的，而FlmAE可以模拟超过65%的图像。尽管如此，FIrmAE无法模拟所有TP-Link图像。 我们调查了这些失败的案例，发现它们不包含Web服务器。

表1：网络和Web服务的仿真率

		Firmadyne FirmAE数据集				
Vendor Images Net Web NetWeb						
分析集	D-Link（93.30%）	179	55	54（30.17%）	177	167例
	TP-Link 73 26 5（6.85%）	7359（80.82%）				
	NETGEAR	274	86	30（10.95%）	259	257（93.80%）
小计		526	167	89（16.92%）	509	483（91.83%）
最新设置	D-Link 58 18 17（29.31%）	54 48（82.76%）			40	
	TP-Link 69 33 10（14.49%）	6954（78.26%）				
	NETGEAR	101 30 7（6.93%）	9279（78.22%）			
	TRENDnet	106 35 23（21.70%）	9163（59.43%）			
	华硕107 27 25（23.36%）	6362（57.94%）				
	贝尔金37 2 2（5.41%）	3022（59.46%）				
	Linksys 55 13 8（14.55%）	4844（80.00%）				
	合勤20 3 0（0.00%）	1810（50.00%）				
小计		553	161	92（16.64%）	465	382例（69.08%）
CamSet	TP-Link 6 0 0（0.00%）	6 0 0（0.00%）				
	TRENDnet	13 2 2（15.38%）	1010（76.92%）			
小计				45 2 2（4.44%）	3527（60.00%）	
总数1124 330 183（16.28%）1009 892（79.36%）						

CamSet的结果表明，许多IP摄像机与无线路由器具有相似的特性，使得无线路由器的仲裁也可以应用于IP摄像机。

每次仲裁的影响 我们还通过从最终版本的FirmAE中省略特定仲裁来调查每个仲裁的有效性，所有仲裁都适用于该最终版本。这是因为多个仲裁点应协作以解决故障，并且扣除特定仲裁直接影响仿真速率。图2显示了这些结果，详细版本见附录中的表5

NVRAM仲裁似乎是最有效的，decias- ing仿真率平均35%，在所有数据集。这与Firmadyne专注于仿真NVRAM的方法删除引导和网络仲裁也显著降低了约30%的仿真率在没有内核仲裁的情况下，所有数据集中只有4.88%的固件映像受到影响其他仲裁影响了22.35%的固件映像。这些结果表明，所提出的仲裁确实是有效的和可扩展的成功的固件仿真。

5.2 仿真后分析

在大规模仿真之后，我们调查了无法通过简单仲裁轻松解决的未处理故障问题，但需要更复杂的虚拟化。

内核模块。 如之前的研究[12，17，23，25，57]中所讨论的，仿真内核模块具有挑战性，因为1）不同的内核版本通常会产生兼容性问题，以及2）一些固件映像可能没有内核，因此无法获得有用的信息。在少数情况下，Web服务器和其他程序访问/proc目录下的内核模块。因为这些文件在仿真环境中不存在，所以这些程序经常崩溃。例如，TP-Link固件映像中的Web服务器访问/proc/simple_config/system_code中的内核模块进行配置，随后会崩溃，因为该模块不存在。

硬件接口。固件的一些内部程序使用它们自己的专用接口进行外围通信，从而强化仿真外围接口。例如，我们挂钩流行的库调用来模拟NVRAM。然而，一些

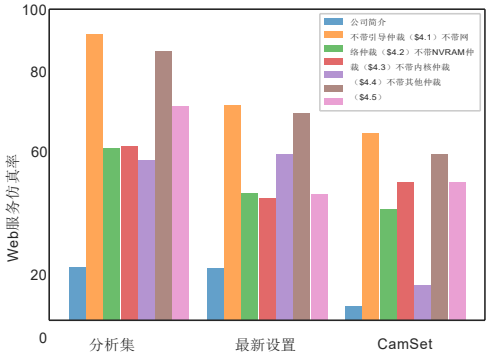

图2：所申请仲裁的有效性

D-Link固件程序调用/bin/flash直接访问/dev/nvram。类似地，一些TP-Link固件映像中的httpd服务器访问闪存/dev/ar7100_flash_chrdev，以检索设备配置信息。同时，Linksys固件中名为webs的Web服务器直接操作/dev/mtd接口。它们甚至验证设备的完整性，并验证给定固件的签名和版本

CGI错误。 即使Web服务器可以访问，其中一些很少响应服务器错误，即*500内部服务器错误*。此错误有几个原因，例如CGI程序中的语法/代码错误，无效的Web界面配置和PHP错误。然而，大多数错误情况是由后端CGI程序崩溃引起的。我们用逆向工程的方法对CGI程序进行了分析，发现它们有着相同的硬件接口问题。因此，它们会尝试访问/proc或/dev下的条目以获取配置值，如果失败则会异常停止。

上述情况呈现了在没有物理设备的情况下仿真外围通信解决这些问题需要一个更复杂的仿真环境，这将在未来的研究中解决。

5.3 应用动态分析

在成功模拟固件映像后，我们将动态分析，模糊，他们的Web服务。通过评估，我们1）验证了仲裁仿真确实适用于应用嵌入式设备的动态安全分析，2）评估了当前的嵌入式设备在野外的安全状况我们的目标LatestSet和CamSet与最新的固件。

动态分析引擎。 对于大规模的分析，我们专注于可扩展性。因此，我们的动态分析工具需要适用于不同的仿真固件映像，很少的用户交互。有了这些标准，我们首先搜索现有的工具[9，19，21，34，36]。

39，43，53，55，56，64]并检查它们是否适用于FirmAE。然而，现有的工具不满足我们的标准，因为它们1）不是公开可用的，2）对于大规模分析不可扩展，以及3）不能发现新的漏洞。例如，Firmadyne [17]利用Metasploit [43]，其检查已知的脆弱性。其他Web扫描器，如Burp Suite [55]，Arachni [34]或Commix [53]只检查预定义的HTTP模式的组合。因此，它们不足以用于实际场景中的各种固件web服务。此外，它们不是为了查找内存损坏漏洞（如缓冲区）而设计的。

溢出或释放后使用。同时，最先进的模糊器Firm-AFL [64]是检测内存损坏漏洞的有前途的工具。然而，它不适用于大规模的分析，因为它需要为目标程序设置单独的环境 由于这些限制，我们构建了自己的分析引擎。开发一个分析引擎本身是一个正交的研究领域，这里我们只提出一个概念设计。我们的概念也可以应用于上述工具。

我们的分析引擎由两部分组成：如果需要，它会自动初始化和登录到网页，并识别包括存储器损坏错误在内的漏洞。为了查找1天漏洞，我们利用了RouterSploit [56]，它具有概念验证功能(PoC)以前已知漏洞的代码我们还添加了

几个定制的PoC代码。为了分析0天漏洞，我们在Python中开发了一个简单的880LoC Web模糊器。

正在初始化Web服务。动态分析的主要步骤是初始化Web服务，除非它们没有收到任何其他请求。我们数据集中的大部分web服务都需要网络和安全配置（例如，admin或AP密码）。然而，该初始化过程在每个固件中也不同。 D-Link、TP-Link、Belkin、Linksys和ZyXEL中大多数固件映像中的Web服务器会自动初始化
在成功仿真后，用户可以自己进行初始化，而华硕和TRENDnet的用户则必须亲自进行初始化幸运的是，它们中的许多都有一个跳过按钮来配置默认选项。一些Web服务没有显式地具有跳过按钮，但是具有行为相同的内部JavaScript函数。同时，有些需要手动管理员密码。

为了实现初始化的自动化，我们分析了Web服务的初始化过程，从中提取了具有代表性的按钮和菜单模式。然后，我们利用这些模式来自动化的过程。在这里，我们利用了Se- lenium [50]，这是一个开源工具，可以提供类似于真实浏览器的界面。

评估漏洞发现性能。成功运行固件映像及其Web服务后，引擎首先使用RouterSploit [56]和我们的定制PoC代码检查1天漏洞。由于RouterSploit由已知漏洞的多个利用组成，因此在此评估中，我们可以1）检查目标设备是否已打补丁，2）找到以前未知但具有相同漏洞的新易受攻击设备为了找到0天漏洞，我们的引擎首先搜索目标固件的文件系统，并通过检查文件的扩展名（如.html、.aspx或.xml。然后，它从候选人中提取可能的参数，并生成检测漏洞的请求例如，对于.htm和.html候选项时，我们的引擎会解析HTML标记（如script、form和input），以提取目标URL、方法和参数信息。当为使用家庭网络管理协议（HNAP）的模糊设备构建请求时，这种方法特别有用; HNAP请求基于XML格式，默认值在.html页面。利用提取的信息，我们可以构造一个有效的请求模板模糊。 因为我们从文件系统中搜索候选者，所以我们还可以检查爬取无法访问的Web服务。

表2：AnalysisSet的1天分析结果

漏洞	Firmadyne FirmAE	类别PoC #图像#（唯一）图像#（唯一）
信息泄露	2 0（0）	17（17）
指令喷射	9	10（6） 152（65）
密码泄露	2	4（3） 146（99）
身份验证绕过	2	0（0） 5（5）
总计	15	14（9） 320（128）

表3：在LatestSet和CamSet上发现的新漏洞

类型	漏洞类别	Vulns数量	器械数量
1天	PHP中的信息泄漏 CGI中的信息泄漏 UPnP SOAP CGI中的命令注入	2	12 HNAP1和3中的命令注入 带后门的命令注入（32764） 2 3 路径遍历 2 9
	小计		
0天	HNAP中的命令注入 CGI中的命令注入	1	3 HNAP% 1% 1中的缓冲区溢出 CGI中的缓冲区溢出
	小计	12	23
	共计	23	95

在各种类型的漏洞，我们专注于命令注入和缓冲区溢出，因为他们经常在嵌入式设备中发现。 为了检测命令注入漏洞，我们的引擎发送有效负载，这些有效负载本质上是候选字符的组合，例如&“”，“”或“”，然后是执行可执行二进制文件的shell命令。我们放置这个二进制文件来记录有用的信息，例如时间和环境变量，从而检查是否触发了漏洞我们还挂接了execve系统调用，以便轻松检测我们的输入是否注入了命令。对于缓冲区溢出检测，当崩溃发生时，FIrmAE提供请注意，由于处理请求所需的时间，我们必须在向目标Web服务发送请求后等待;我们根据经验确定10到15秒就足够了。我们还利用边界值，例如用于模糊输入的大型缓冲区，因为它们更有可能触发漏洞。

我们的分析引擎报告的任何错误都必须经过验证。为此，我们在目标固件的文件系统中添加了strace、gdb和gdbserver等调试程序。注意，我们可以在升级内核版本时使用ptrace系统调用进行调试（第4.4节）。 我们还添加了netcat和telnetd来访问guest shell。使用这些工具，我们手动验证了已识别的错误。

5.4 动力分析结果

为了评估仲裁仿真的有效性，我们对每个仿真固件映像进行了动态分析，其中Web服务已经由我们的引擎初始化特别地，在由FirmAE和Firmadyne中的每一个初始化目标固件映像的web服务之后，我们运行先前已知的PoC漏洞利用。 我们首先使用Router-Sploit [56]在AnalysisSet中的仿真图像上测试已知的漏洞 其中每个都具有FmAE和Firmadyne，并且结果在表2中列出。 不使用任何仲裁（*即*，Firmadyne），我们只能检查14个图像中的漏洞，其中9个是独特的设备。 通过申请所有建议仲裁（*即*，F（rmAE），我们可以检查

320个图像中的漏洞，其中128个是独特的。由于FermAE旨在模拟Web服务（§3.1），所有识别的漏洞都位于Web服务中，例如SOAP CGI、UPnP和HNAP。这一结果表明，F irmAE的成功仿真有助于在动态分析固件图像方面优于Firmadyne。

此外，我们还对LatestSet和CamSet中的最新图像进行了动态分析，包括模糊器。结果，我们在95个独特的设备中发现了总共23个独特的漏洞这些漏洞包括表3中列出的11个1天和12个0天漏洞。对于模糊器，当并行运行50个图像时，每个模糊请求平均花费10 -15s，并且发现每个漏洞所花费的平均时间为70min，最大为150min。模糊吞吐量可以根据系统规格和并行仿真实例的数量而变化。

一个有趣的观点是，一些供应商共享相同的vulner-abilities。例如，D-Link和TRENDnet中的一些设备具有相同的信息泄漏漏洞，以及UPnP和SOAP CGI程序中的命令注入。相反，一些NETGEAR设备与Xiong-mai的共享路径遍历漏洞。另一点是，对目标Web服务的分析可能会揭示与之相关的其他程序的漏洞具体来说，当我们发送一个长有效负载来检测缓冲区溢出时，目标CGI程序将有效负载存储在一个文件中。然后，另一个读取写入文件的程序由于溢出的有效负载而崩溃这种漏洞只能在全系统仿真环境中发现，因为用户模式仿真不考虑文件系统关系。

总之，结果表明，F irmAE是实用的脆弱性分析。我们认为，未发现的漏洞仍然存在，这应该在未来的研究进行调查。

负责任的披露。检测到的0天漏洞分布在四个供应商中。我们在2019年12月之前向供应商报告了所有12个漏洞，最多花了9个月才收到他们的回应。

6 讨论

仲裁仿真中的仿真差异固件AE的目的不是消除真实环境和仿真环境之间的差异，而是运行固件的Web服务器并正确地服务于Web界面。这可能导致与在硬件上运行固件不同的行为。然而，为了应用动态安全分析，我们需要检查的是1）易受攻击的程序是否运行，2）是否接受恶意输入，以及3）是否触发程序中的漏洞。尽管模拟可能不正确，但如果满足以下条件，则可以检查这三项1）我们可以运行固件的Web服务，2）通过网络发送漏洞利用数据包，以及3）验证漏洞利用是否已成功执行。因为我们的仲裁仿真可以支持这些，所以由Fi rmAE发现的漏洞是合法的，并且也在真实设备中工作。

仲裁介入的一般性。虽然我们的启发式仲裁仿真比其他作品表现更好的当前的固件映像，因为我们开发的启发式处理失败的情况下，经验，我们的系统化仲裁仿真可以仅处理观察到的情况，可能不适用于新器械和新的配置。在这方面，我们认为，一个经验寻找这种干预的调查对于处理IoT设备及其配置的复杂性质似乎是必不可少的。为了鼓励未来的研究，我们发布了我们的代码，相信我们的经验发现可以作为参考。

应用其他动态分析技术。在这项研究中，我们开发了一个简单的分析引擎，自动初始化，登录，并分析Web服务的动态分析。然而，每一步都可以通过应用其他有前途的技术来进一步改进。例如，可以通过使用符号执行来分析和绕过登录过程[52]。此外，采用其他模糊策略[8，48]，混合分析方法[54，61]或相似性技术[18，20]，可能会发现更多的漏洞。我们离开这样有前途的改进动态分析引擎作为未来的工作。

应用仿真来构建IoT蜜罐。仲裁模拟还可以用于构建蜜罐，以分析针对IoT设备的大量攻击。事实上，已经有几个利用仿真的蜜罐研究[35，47，49，57]。特别地，Vetterl*et al*. [57]提出了一种名为Honware的基于固件仿真的蜜罐，类似于FmAE的方法。由于蜜罐应该与网络外部的攻击者进行交互，因此作者专注于通过调查仿真失败案例来提高网络可达性。相应地，配置默认网络设置的FIRMAE的网络干预与Honware的方法相当类似。然而，FermAE包括额外的干预来运行web服务以主动分析其中的漏洞，并且这种干预甚至更多地增加了仿真率（表5）。因此，我们相信仲裁模拟也可以用于构建物联网蜜罐[2]

7 结论

嵌入式设备的安全性分析受到了广泛的关注。在这项研究中，我们调查了一个大规模的固件数据集，发现固件仿真可以大大受益于简单的干预。我们提出了仲裁仿真和干预，可以解决高层次的故障问题。通过一个原型，F IRMAE，我们证明了所提出的方法可以提高487%的仿真率的国家的最先进的框架。我们还对仿真固件进行了动态分析，发现了23个独特的漏洞，包括12个0天。

公司简介

我们感谢匿名评论者提供的有用反馈，以及Minkyo Seo开发容器化。本工作得到了韩国政府（MSIT）资助的信息&通信技术规划&评估研究所（Institute of Information Communications Technology Planning Evaluation，IITP）资助（No.2018-0-00831，异构无线网络物理层安全研究，No.2019-0-01343，区域战略产业融合安全核心人才培养业务）

参考文献

[1] 2014. *第23届USENIX安全研讨会论文集（安全）*。加利福尼亚州圣地亚哥

2我们找不到Honware的公共源代码进行评估。

[2] 2016. *2016年度网络和分布式系统安全研讨会（NDSS）*。加利福尼亚州圣地亚哥

[3] 2019. *第28届USENIX安全研讨会论文集（安全）*。加利福尼亚州圣克拉拉

[4] 2020. *第29届USENIX安全研讨会论文集（安全）*。马萨诸塞州波士顿

[5] Manos Antonakakis，Tim April，Michael Bailey，Matt Bernhard，Elie Bursztein，Jaime Cochran，Zakir Durumeric，J. Alex Halderman、Luca Invernizzi、MichalisKallitsis、Deepak Kumar、Chaz Lever、Zane Ma、Joshua Mason、Damian Menscher、Chad Seaman、Nick Sullivan、Kurt Thomas和Yi Zhou。2017年。了解Mirai僵尸网络*第26届USENIX安全研讨会论文集（Security）*。温哥华，BC，加拿大.

[6] 法布里斯·贝拉德2005年QEMU，一个快速和可移植的动态翻译器。在*FREENIX轨道的Proceedings：2005 USENIX年度技术会议，2005年4月10日至15日，美国加利福尼亚州阿纳海姆*。

[7] Roland Bodenheim，Jonathan Butts，Stephen Dunlap，and Barry Mullins.2014年Shodan搜索引擎识别面向互联网的工业控制设备的能力评估*International Journal of Critical Infrastructure Protection* 7，2（2014），114-123.

[8] Sang Kil Cha，Maverick Woo，and David Brumley.2015年。程序自适应变异模糊。*第36届IEEE安全与隐私研讨会*。San Jose，CA，725

[9] 王春雷刘丽刘强。2014年web服务漏洞的自动模糊测试。在*信息和通信技术国际会议（ICT 2014）的会议记录*。IET，中国南京

[10] AbrahamA Clements、Eric Gustafson、Tobias Scharnowski、Paul Grosen、DavidFritz、Christopher Kruegel、Giovanni Vigna、Saurabh Bagchi和Mathias Payer。2020. HALucinator：通过抽象层仿真的固件重新托管，参见[4]。

[11] Andrei Costin，Jonas Zaddach，Aurélien Francillon，and Davide Balzarotti.2014年嵌入式固件安全性的大规模分析，参见[1]。

[12] Andrei Costin，Apostolis Zarras，and Aurélien Francillon. 2016.大规模自动化动态固件分析：嵌入式Web接口的案例研究。*第11届ACM信息、计算机和通信安全研讨会*中国西安

[13] 阿崔。2012.使用FRAK的嵌入式设备固件漏洞搜索。输入*黑帽美国简报（Black Hat USA Briefings）*。内华达州拉斯维加斯

[14] Ang Cui，Michael Costello，and Salvatore J Stolfo.2013年。当固件修改攻击时：嵌入式漏洞利用案例研究 *2013年度网络和分布式系统安全研讨会（NDSS）*。加利福尼亚州圣地亚哥

[15] Ang Cui和Salvatore J Stolfo.2010年。嵌入式网络设备不安全性的定量分析：广域扫描的结果。*年度计算机安全应用会议（ACSAC）*。

[16] CVE 2014。CVE-2014-3936。https://cve.mitre.org/cgi-bin/cvename.cgi?名称=CVE-2014-3936。

[17] 大明湾Chen，Manuel Egele，Maverick Woo，and David Brumley.2016年。To-wards Automated Dynamic Analysis for Linux-based Embedded Firmware，See [2].

[18] Yaniv David，Nimrod Partush和Eran Yahav。2018年。FirmUp：Precise Static Detec-tion of Common Vulnerabilities in Firmware. 程序*设计语言和操作系统架构支持国际会议论文集*。392-404

[19] R·道斯2011年。OWASP WebScarab项目

[20] Steven HH Ding，Benjamin CM Fung，and Philippe Charland.2019年。Asm2Vec：增强二进制克隆搜索对代码混淆和编译器优化的静态表示鲁棒性。在*第40届EEE Symposium on Security and Privacy（Oakland）会议录中*。加利福尼亚州旧金山

[21] 迈克尔·爱丁顿2011年。桃子模糊平台。*第*34集9.1 The Lord of the Lord（2011）

[22] Florian Fainelli 2008. OpenWrt嵌入式开发框架。输入*自由和开源软件开发者欧洲会议论文集*。

[23] Bo Feng，Alejandro Mera，and Long Lu.2020年。P2IM：通过自动外围接口建模的可扩展和硬件独立的固件测试，参见[4]。

[24] NCC Group等.2017年。一个linux系统调用fuzzer使用TriforceAFL.https：//github.com/nccgroup/TriforceAFL的网站。

[25] Eric Gustafson、Marius Muench、Chad Spensky、Nilo Redini、Aravind Machiry、Yanick Fratantonio、Davide Balzarotti、Aurelien Francillon、Yung Ryn Choe、Christophe Kruegel和Giovanni Vigna。2019年。通过自动重托管分析嵌入式固件*第22届攻击、入侵和防御研究国际研讨会（RAID）*论文集。中国北京

[26] 克雷格·海夫纳2010年。固件分析工具。 https://github.com/ReFirmLabs/

[27] Craig Heffner，Jeremy Collake，等. 2011.固件模块套件。https：//github.com/rampageX/firmware-mod-kit。

[28] Markus Kammerstetter，Daniel Burian，and Wolfgang Kastner. 2016. 使用外围设备缓存和运行时程序状态近似进行嵌入式安全测试。在*第十届新兴安全信息国际会议上，系统和技术（SECUWARE）*。

[29] Markus Kammerstetter、Christian Platzer和Wolfgang Kastner。2014.前景：外围设备代理支持嵌入式代码测试。*第九届ACM信息、计算机和通信安全研讨会（ASIACCS）*日本京都

[30] Swati Khandelwal2016年。 在D-Link DWR-932 B LTE路由器中发现多个后门。http://thehackernews.com/2016/09/hacking-d-link-wireless-router的网站。html？m=1。

[31] Swati Khandelwal2017年。 Satori IoT僵尸网络利用Zero-Day僵尸化华为路由器 https://thehackernews.com/2017/12/satori-mirai-iot-botnet.html。

[32] 布莱恩·克雷布斯2016年。 IoT僵尸网络“Mirai”源代码 https：//krebsonsecurity.com/2016/10/source-code-for-iot-botnet-mirai-released 的网站。

[33] DeepakKumar、Kelly Shen、Benton Case、Deepali Garg、Galina Alperovich、DmitryKuznetsov、Rajarshi Gupta和Zakir Durumeric。2019年。 综合考虑：家庭网络上物联网设备的分析，见[3]。

[34] 塔索斯·拉斯科斯2010年。阿拉克尼 http://www.arachni-scanner.com。

[35] Samuel Litchfield，David Formby，Jonathan Rogers，Sakis Meliopoulos，and Ra- heem Beyah.2016年。重新思考网络物理系统的蜜罐 *IEEEInternet Computing* 20，5（2016），9-17.

[36] 克努德·拉塞·卢斯。2018年。2018年物联网状况：物联网设备数量现已达到7B – Market

[37] 大卫·马西亚克。2018.又一个加密采矿僵尸网络？https://www.fortinet的网站。com/blog/threat-research/yet-another-crypto-mining-botnet.html.

[38] 丹尼斯·马克鲁申2018. D-Link后院的后门https://securelist.com/backdoors-in-d-links-backyard/85530.

[39] 哈维·门德斯2014. wfuzz。 https://github.com/xmendez/wfuzz。

[40] 德克·默克尔。2014年Docker：轻量级Linux容器，用于一致的开发和部署。*Linux Journal*2014，239（2014），2.

[41] 瑞恩·米切尔2018年。*使用Python进行网页搜罗：从现代网络收集更多数据*。“奥莱利媒体公司“.

[42] 布鲁斯·莫吉安2001年*PostgreSQL：介绍和概念*。体积一百九十二艾迪生-韦斯利纽约。

[43] HD Moore等.2009年Metasploit项目。 https://www.metasploit.com。

[44] Marius Muench、Aurélien Francillon和Davide Balzarotti。2018年。Avatar2：多目标编排平台。*二进制分析研究研讨会（BAR）*

[45] MariusMuench，Jan Stijohann，Frank Kargl，Aurélien Francillon，and DavideBalzarotti. 2018年。你破坏的不是你崩溃的：模糊嵌入式设备的挑战。*2018年度网络和分布式系统安全研讨会（NDSS）*论文集。加利福尼亚州圣地亚哥

[46] 吴正旭2014年逆向工程闪存的乐趣和利益。输入*黑帽美国简报（Black Hat USA Briefings）*。内华达州拉斯维加斯

[47] YinMinn Pa Pa，Shogo Suzuki，Katsunari Yoshioka，Tsutomu Matsumoto，Takahiro Kasama，and Christian Rossow. 2015年。IoTPOT：分析物联网妥协的兴起USENIX*攻击性技术研讨会（WOOT）*华盛顿特区。

[48] Alexandre Rebert、Sang Kil Cha、Thanassis Avgerinos、Jonathan Foote、David Warren、Gustavo Grieco和David Brumley。2014年 优化用于模糊化的种子选择，参见[1]。

[49] Lukas Rist、Johnny Vestergaard、Daniel Haslinger、Andrea Pasquale和JohnSmith。2013. Conpot ics/scada蜜罐。 http://conpot.org。

[50] Selenium 2004. 硒 https://www.seleniumhq.org。

[51] 肖丹2016年。D-Link互联网报告。https://dlink-report.shodan.io/。

[52] Yan Shoshitaishvili，Ruoyu Wang，Christophe Hauser，Christopher Kruegel，andGiovanni Vigna. 2015年。自动检测二进制固件中的身份验证绕过漏洞 *2015年度网络和分布式系统安全研讨会（NDSS）*。加利福尼亚州圣地亚哥

[53] Anastasios Stasinopoulos、Christoforos Ntantogian和Christos Xenakis。2015. Commix：检测和利用命令注入漏洞。黑*帽美国简报（Black Hat USA Briefings）*内华达州拉斯维加斯

[54] NickStephens、John Grosen、Christopher Salls、Andrew Dutcher、Ruoyu Wang、Jacopo Corbetta、Yan Shoshitaishvili、Christopher Kruegel和Giovanni Vigna。2016. Driller：通过选择性符号执行增强模糊 参见[2]。

[55] 达维德·斯塔德2008. 打嗝套房。 https://portswigger.net/burp。

[56] 威胁9. 2016. RouterSploit。 https://github.com/threat9/routersploit。

[57] Alexander Vetterl和Richard Clayton 2019. Honware：用于捕获CPE和IoT零日的虚拟蜜罐框架。2019年*APWG电子犯罪研究研讨会（eCrime）*。IEEE，1

[58] 维基百科贡献者。2018. IP别名-维基百科，自由的百科全书. https://en.wikipedia.org/w/index.php? title=IP_aliasing oldid=871887325。[在线；2019年8月13日访问]。

[59] 马特·威尔逊2019年。 高级无线路由器市场规模，份额，统计，趋势，类型，应用，分析和预测 |2019-2024年全球行业研究与预测。https://marketersmedia.com/premium-wireless-routers-market-size-share-statistics-trends-types-applications-analysis-and-forecast-global-industry-research-and-forecast-2019-2024/520294.

[60] 胡恒银，郑耀文2018年。物联网设备的可扩展动态分析框架。*黑帽美国简报（Black Hat USA Briefings）*内华达州拉斯维加斯
[61] Insu Yun，Sangho Lee，Meng Xu，Yeongjin Jang，and Taesoo Kim.2018年。QSYM：一个实用的concolic执行引擎为混合fuzzing量身定制 在*第27届USENIX安全研讨会（安全）的会议记录中*。巴尔的摩，MD，745
[62] Jonas Zaddach，Luca Bruno，Aurelien Francillon，and Davide Balzarotti.2014年Avatar：一个支持嵌入式系统固件动态安全分析的框架。在*2014年度网络和分布式会议记录中系统安全研讨会（NDSS）*。加利福尼亚州圣地亚哥
[63] 米哈尔·扎莱夫斯基2017年。美国绒头洛普（AFL）。http://lcamtuf.coredump.cx/afl。（2017年）。
[64] Yaowen Zheng，Ali Davanian，Heng Yin，Chengyu Song，Hongsong Zhu，and Limin Sun.2019年。FIRM-AFL：经由增强的过程仿真的iot固件的高吞吐量灰盒模糊，参见[3]，1099-1114。

表4：固件数据集的完整统计

数据集供应商图像		数量	架构数量				Web服务数量									
			arm32el	mips32el	mips32eb	等等	httpd	uhttpd	简体中文	lighttpd	alphapd	前进	博	jjhttpd	等等	
AnalysisSet	D-Link	179	22人（12.29%）	82例（45.81%）	75（41.90%）（0.00%）	0	102例（56.98%）	0（0.00%）	0（0.00%）	9人（5.03%）	39（21.79%）（5.59%）	10	2人（1.12%）	14人（7.82%）	5人（2.79%）	
	TP-Link	73	10人（13.70%）	15人（20.55%）	48（65.75%）（0.00%）	0	64例（87.67%）	9人（12.33%）	0（0.00%）	0（0.00%）	0（0.00%）（0.00%）	0	0（0.00%）	0（0.00%）	0（0.00%）	
	网络齿轮	274	105（38.32%）	56人（20.44%）	113（41.24%）（0.00%）	0	125（45.62%）	77人（28.10%）	69人（25.18%）	5人（1.82%）	0（0.00%）（0.00%）	0	3人（1.09%）	0（0.00%）	0（0.00%）	
小计		526	137例（26.05%）	153例（29.09%）	236（44.87%）（0.00%）	0	291例（55.32%）	86（16.35%）	69人（13.12%）	14人（2.66%）	39（7.41%）（1.90%）	10	5（0.95%）	14人（2.66%）	5（0.95%）	
最新设置	D-Link	58	9人（15.52%）	17（29.31%）	32（55.17%）（10.34%）	6	39例（67.24%）	0（0.00%）	8人（13.79%）	12人（20.69%）	0（0.00%）（1.72%）	1	4人（6.90%）	3人（5.17%）	0（0.00%）	
	TP-Link	69	13人（18.84%）	22人（31.88%）	34（49.28%）（0.00%）	0	53例（76.81%）	16人（23.19%）	0（0.00%）	0（0.00%）	0（0.00%）（0.00%）	0	0（0.00%）	0（0.00%）	0（0.00%）	
	网络齿轮	101	32人（31.68%）	44人（43.56%）	25（24.75%）（0.99%）	1	46例（45.54%）	36例（35.64%）	19人（18.81%）	0（0.00%）	0（0.00%）（1.98%）	0（0.00%）	2	0（0.00%）（0.00%）	0	
	TRENDnet	106	18人（16.98%）	29人（27.36%）	59（55.66%）（0.00%）	0	28人（26.42%）	9人（8.49%）	13个（12.26%）	6人（5.66%）	0（0.00%）（10.38%）	11（10.38%）	11	3（2.83%）（19.81%）	21	
	华硕	107	28人（26.17%）	72（67.29%）	2（1.87%）（0.00%）	0	106例（99.07%）	0（0.00%）	0（0.00%）	51例（47.66%）	0（0.00%）（0.00%）	0（0.00%）	0	0（0.00%）（0.93%）	1	
	贝尔金	37	2人（5.41%）	20例（54.05%）	15（40.54%）（0.00%）	0	25例（67.57%）	0（0.00%）	11例（29.73%）	0（0.00%）	0（0.00%）（0.00%）	0（0.00%）	0	0（0.00%）（10.81%）	4	
	Linksys	55	15人（27.27%）	30例（54.55%）	10（18.18%）（1.82%）	1	23例（41.82%）	1人（1.82%）	6人（10.91%）	26例（47.27%）	0（0.00%）（0.00%）	0（0.00%）	0	0（0.00%）（0.00%）	0	
	合勤科技	20	5（25.00%）	10例（50.00%）	5（25.00%）（0.00%）	0	2（10.00%）	0（0.00%）	2（10.00%）	7人（35.00%）	0（0.00%）（25.00%）	2（10.00%）	5	0（0.00%）（15.00%）	3	
小计		553	122（22.06%）	244例（44.12%）	182（32.91%）（1.45%）	8	322例（58.23%）	62人（11.21%）	59人（10.67%）	102（18.44%）（0.00%）	0	14人（2.53%）	22人（3.98%）	6（1.08%）（5.24%）	29	
CamSet	D-Link	26	8人（30.77%）	15人（57.69%）	3（11.54%）（0.00%）	0	6人（23.08%）	0（0.00%）	0（0.00%）	0（0.00%）（50.00%）	13	0（0.00%）	0（0.00%）	0（0.00%）（23.08%）	6	
	TP-Link	6	6（100.00%）	0（0.00%）	0（0.00%）（0.00%）	0	0（0.00%）	0（0.00%）	0（0.00%）	0（0.00%）（0.00%）	0	0（0.00%）	0（0.00%）	0（0.00%）（100.00%）	6	
	趋势网	13	1个（7.69%）	10例（76.92%）	2（15.38%）（0.00%）	0	0（0.00%）	0（0.00%）	0（0.00%）	0（0.00%）（46.15%）	6	0（0.00%）	2人（15.38%）	0（0.00%）（15.38%）	2	
小计		45	15人（33.33%）	25人（55.56%）	5（11.11%）（0.00%）	0	6人（13.33%）	0（0.00%）	0（0.00%）	0（0.00%）（42.22%）	19	0（0.00%）	2人（4.44%）	0（0.00%）（31.11%）	14	
总数		**1124**	**274（24.38%）**	**422（37.54%）**	**423（37.63%）**	**8（0.71%）**	**619（55.07%）**	**148（13.17%）**	**128（11.39%）**	**116（10.32%）**	**58（5.16%）**	**24（2.14%）**	**29（2.58%）**	**20（1.78%）**	**48（4.27%）**	

表5：FirmAE删除每个仲裁的完整结果

							不带引导仲裁的FirmAE数量		无网络仲裁无NVRAM仲裁		无内核仲裁		无其他仲裁			
					数据集供应商	图片	网路	Web服务	网路	Web服务	网路	Web服务	网路	Web服务	网路	Web服务	网路	Web服务
AnalysisSet	D-Link	179	177例（98.88%）	167例（93.30%）	162（90.50%）（81.01%）	145	100（55.87%）	90（50.28%）	176例（98.32%）	129例（72.07%）	154例（86.03%）	144例（80.45%）	173例（96.65%）	146例（81.56%）		
	TP-Link	73	73（100.00%）	59例（80.82%）	53例（72.60%）（49.32%）	36例	27人（36.99%）	13人（17.81%）	73（100.00%）	56例（76.71%）	73（100.00%）	60（82.19%）	55人（75.34%）	31例（42.47%）		
	网络齿轮	274	259例（94.53%）	257例（93.80%）	110（40.15%）（40.15%）	110	191例（69.71%）	191例（69.71%）	239例（87.23%）	86（31.39%）	259例（94.53%）	250（91.24%）	252例（91.97%）	185例（67.52%）		
小计		526	509（96.77%）	483（91.83%）	325（61.79%）（55.32%）	291	318例（60.46%）	294例（55.89%）	488（92.78%）	271例（51.52%）	486（92.40%）	454例（86.31%）	480（91.25%）	362例（68.82%）		
最新设置	D-Link	58	54例（93.10%）	48例（82.76%）	46例（79.31%）	41例（70.69%）	19人（32.76%）	18人（31.03%）	54例（93.10%）	48例（82.76%）	54例（93.10%）	40例（68.97%）	51例（87.93%）	45（77.59%）		
	TP-Link	69	69（100.00%）	54例（78.26%）	54例（78.26%）	32例（46.38%）	39例（56.52%）	23人（33.33%）	69（100.00%）	53例（76.81%）	69（100.00%）	57例（82.61%）	57例（82.61%）	23人（33.33%）		
	网络齿轮	101	92（91.09%）	79（78.22%）	49人（48.51%）	41人（40.59%）	68例（67.33%）	60（59.41%）	92（91.09%）	25人（24.75%）	92（91.09%）	82例（81.19%）	87人（86.14%）	54（53.47%）		
	TRENDnet	106	91（85.85%）	63例（59.43%）	55人（51.89%）	41例（38.68%）	49人（46.23%）	37例（34.91%）	91（85.85%）	56例（52.83%）	87例（82.08%）	52例（49.06%）	84（79.25%）	44人（41.51%）		
	华硕	107	63例（58.88%）	62例（57.94%）	31例（28.97%）	31例（28.97%）	34人（31.78%）	32例（29.91%）	63例（58.88%）	45人（42.06%）	62例（57.94%）	61例（57.01%）	58人（54.21%）	25人（23.36%）		
	贝尔金	37	30例（81.08%）	22例（59.46%）	3人（8.11%）	3人（8.11%）	14人（37.84%）	14人（37.84%）	30例（81.08%）	19人（51.35%）	30例（81.08%）	22例（59.46%）	29例（78.38%）	5人（13.51%）		
	Linksys	55	48例（87.27%）	44（80.00%）	34例（61.82%）	34例（61.82%）	31例（56.36%）	31例（56.36%）	47例（85.45%）	42例（76.36%）	48例（87.27%）	44（80.00%）	44（80.00%）	27例（49.09%）		
	合勤科技	20	18（90.00%）	10例	7人	2	8例	2	18	6例	18	10例	18	1（5.00%）		

				（50.00%）	（35.00%）	（10.00%）	（40.00%）	（10.00%）	（90.00%）	（30.00%）	（90.00%）	（50.00%）	（90.00%）	
小计		553	465（84.09%）	382例（69.08%）	279例（50.45%）	225（40.69%）	262例（47.38%）	217（39.24%）	464例（83.91%）	294例（53.16%）	460（83.18%）	368例（66.55%）	428（77.40%）	224例（40.51%）
CamSet	D-Link	26	19例（73.08%）	17例（65.38%）	13例（50.00%）	12人（46.15%）	13例（50.00%）	11例（42.31%）	18例（69.23%）	3人（11.54%）	18例（69.23%）	16例（61.54%）	18例（69.23%）	14例（53.85%）
	TP-Link	6	6（100.00%）	0（0.00%）	0（0.00%）	0（0.00%）	0（0.00%）	0（0.00%）	6（100.00%）	0（0.00%）	6（100.00%）	0（0.00%）	6（100.00%）	0（0.00%）
	趋势网	13	10例（76.92%）	10例（76.92%）	10例（76.92%）	4人（30.77%）	10例（76.92%）	9例（69.23%）	10例（76.92%）	2人（15.38%）	10例（76.92%）	8例（61.54%）	10例（76.92%）	6人（46.15%）
小计		45	35例（77.78%）	27例（60.00%）	23人（51.11%）	16人（35.56%）	23人（51.11%）	20例（44.44%）	34例（75.56%）	5人（11.11%）	34例（75.56%）	24人（53.33%）	34例（75.56%）	20例（44.44%）
合计		**1124**	**1009（89.77%）**	**892（79.36%）**	**627（55.78%）**	**532（47.33%）**	**603（53.65%）**	**531（47.24%）**	**986（87.72%）**	**570（50.71%）**	**980（87.19%）**	**846（75.27%）**	**942（83.81%）**	**606（53.91%）**