交互很重要：混合IoT/OT蜜罐的综合分析和数据集

奥尔堡大学丹麦　　奥尔堡大学丹麦　　丹麦理工大学丹麦

摘要

物联网（IoT）和利用操作技术（OT）协议的关键基础设施如今是用于进一步传播恶意操作的常见攻击目标和/或攻击表面。已经为物联网和OT提出了诸如蜜罐之类的欺骗技术，但它们要么缺乏广泛的评估，要么受到指纹攻击。在本文中，我们扩展和评估RIoTPot，混合互动蜜罐，通过暴露它在互联网上的攻击，并进行了纵向研究，多个评估参数为三个月。此外，我们以数据集的形式发布了上述研究，可应要求提供给研究人员。我们利用RIoTPot的混合交互模型将其部署在三个交互变体中，其中六个协议部署在云和自托管的基础架构上，以研究和比较收集的攻击。一眼望去，我们收到了10。8700万次攻击事件源自22，518个唯一IP地址，涉及暴力攻击、中毒攻击、多级攻击和其他攻击。此外，我们对攻击者的IP地址进行指纹识别，以识别参与攻击的设备类型最后，我们的研究结果表明，蜜罐的互动水平有一个重要的作用，在吸引特定的攻击和扫描探针。

CCS概念

• **安全和隐私→网络安全**。

关键词

蜜罐，欺骗，操作技术，物联网

ACM参考格式：

Shreyas Srinivasa、Jens Myrup Pedersen和Emmanouil Vasilomanolakis。2022. 交互很重要：综合分析和混合IoT/OT蜜罐数据集。*在年度计算机安全应用会议（ACSAC），2022年12月5日至9日，美国德克萨斯州奥斯汀。ACM*，New York，NY，USA 14页。https://doi.org/10.1145/3564625.3564645

1　引言

针对关键任务基础设施的网络攻击数量稳步增加[20]。最近对美国殖民地管道[22]和佛罗里达水处理厂[35]等工业系统的攻击表明了对公众和环境的影响。

允许免费制作本作品的全部或部分的数字或硬拷贝，以供个人或课堂使用，前提是制作或分发副本的目的不是为了盈利或商业利益，并且副本的第一页上有本声明和完整的引用。必须尊重作者以外的其他人所拥有的本作品组成部分的著作权。允许使用学分进行摘要 以其他方式复制、重新发布、在服务器上发布或重新分发到列表，需要事先获得特定许可和/或付费。 请求权限请发邮件至permissions@acm.org。

ACSAC

©2022版权归所有者/作者所有。授权ACM出版权ACM ISBN 978-1-4503-9759-9/22/12。. . 15美元
https://doi.org/10.1145/3564625.3564645

政府。操作技术（OT）中的关键任务系统依赖于传感器和连接设备来自动化工业控制过程。然而，对最近攻击的研究揭示了这些设备缺乏安全性[50]。此外，物联网设备在消费者和工业应用中的广泛采用增加了攻击空间。研究表明，大量易受攻击的、配置错误的物联网设备暴露在互联网上，复杂的恶意软件可以利用它们[16，40]。此外，《2020年ENISA威胁形势报告》指出，恶意软件仍然是最具挑战性的攻击媒介，每天分发多达*230，000*个变种。

蜜罐是一种欺骗系统，充当攻击者的陷阱机制它们具有低误报，因为没有理由与蜜罐通信，因此，所有通信都可以被认为是可疑的。蜜罐根据它们提供给攻击者的交互能力分为低、中和高交互。多年来，已经针对各种协议或设备简档提出了许多honeypot众所周知的例子包括Cowrie [26]，Conpot [34]，HosTaGe [46]，Glastopf [33]，Dionaea [42]和T-Pot [29]。

由于缺乏交互、静态响应或维护不善蜜罐指纹识别是通过探测机制确定通信中的系统是否是蜜罐成功的指纹攻击会破坏蜜罐的价值，因为它们的身份会暴露。在最近的研究中已经提出了许多用于指纹化蜜罐的技术[23，39，47]。此外，许多开源蜜罐项目被放弃或依赖于没有积极维护的库;这导致缺乏可扩展性和范围缩小[39]。

在本文中，我们扩展了RIoTPot，这是一个模块化和混合交互的蜜罐，它解决了可扩展性和交互性之间的差距，以模拟IT，IoT和OT环境中使用的应用层协议[41]。本文的贡献如下。

- 我们扩展了RIoTPot，并提供了模拟Telnet、SSH、MQTT、Mod-bus、CoAP和HTTP协议的IoT和OT设备配置文件的源代码。
- 我们对RIoTPot进行了纵向研究，根据交互级别，模拟环境，部署基础设施和地理位置实施了许多评估参数。我们报告了我们的研究结果，并讨论了蜜罐操作的评估参数的影响此外，我们研究如何RIoTPot执行相比，另一个流行的国家的最先进的蜜罐。
- 我们向研究社区提供攻击数据集

本文其余部分的结构如下。第2节概述了使用蜜罐分析攻击趋势的相关工作。我们讨论RIoTPot，我们的扩展

实施和第3节中提供的数据集。在第4节中，我们概述了我们的研究方法和实验设置，并在第5节中评估我们的方法。我们将在第6节讨论我们的发现，并在第7节得出结论。

2 相关工作

蜜罐通常根据它们提供给攻击者的交互级别分为低交互、中交互和高交互低交互蜜罐提供了对协议或服务的有限模拟，并且易于管理。中等交互提供了对低交互的扩展交互级别，并且涉及仿真构成多个协议的设备。高交互蜜罐是真实的（或虚拟的）系统/设备，具有定制的日志记录、有限的出口流量规则和短暂的配置以防止误用。乍一看，各种众所周知的低/中交互蜜罐包括：Cowrie [26]，Conpot [34]，HosTaGe [46]，Glastopf [33]，Dionaea [42]和TPot [29]。此外，还有一些高交互性的蜜罐，如Honware[48]，Siphon [13]和Sarracenia [36]。所有上述蜜罐都以低或高交互模式操作，从而提供二进制交互能力。此外，它们在集成附加协议或针对特定环境提供多少可扩展性方面存在很大差异。

另一个需要考虑的重要因素是指纹识别;攻击者可能能够使用利用从目标系统获得的最小数据的技术来检测目标系统是否是蜜罐。一方面，最近的研究表明，虽然低/中交互蜜罐易于管理并且风险最小，但它们更容易受到指纹攻击[39，47]。另一方面，高交互的蜜罐冒着被破坏的环境的风险，该环境可以被用于需要更高维护的恶意活动。

Litchfield等人提出HoneyPhy来解决模拟网络物理环境的蜜罐的有限模拟问题[18]。作者认为，蜜罐无法模拟可能导致指纹的网络物理系统的实际行为。作者通过提出HoneyBot扩展了他们的工作;这是一种明确为网络机器人系统设计的混合交互蜜罐[15]，能够根据攻击请求切换交互。然而，HoneyBot仅限于特定环境，无法扩展到多样化的操作。

在研究中对蜜罐的评估旨在呈现攻击环境的概述，这并不罕见[1，5，9]。Dang等人采用硬件和软件蜜罐相结合的方法进行了为期12个月的纵向研究[7]。该实验涉及部署4个硬件和108个软件物联网蜜罐，以收集对物联网环境的攻击。软件/虚拟蜜罐部署在八个公共云提供商和各种地理位置上。作者提出了一个概述收到的蜜罐和影响的攻击。Minn等人提出了*IoTPOT*，一种模拟来自各种物联网设备的Telnet服务的蜜罐，以研究Telnet协议的攻击趋势[27]。*IoTPOT*蜜罐由一个低交互前端和一个高交互后端组成，该后端称为*IoTBOX*，模拟来自各种物联网设备配置文件的Telnet服务作者部署了39天的蜜罐，收集了43个不同的恶意软件样本。

Srinivasa等人在互联网上找到大量配置错误的物联网设备，并部署六个开源物联网蜜罐来研究攻击趋势[40]。Tabari等人通过观察现实世界的攻击，呈现一个多方面的物联网蜜罐生态系统，具有可扩展的复杂性[52]。作者为物联网摄像头开发了一个蜜罐，以观察它们周围的攻击环境。此外，作者还提出了*ProxyPot*，这是一种位于物联网设备和外部网络之间的蜜罐代理，用于观察入站和出站流量。物联网摄像头蜜罐部署了两年，观察到攻击数量增加

Vetterl等人提出*Honware*，一种高交互性的虚拟蜜罐框架，可以在没有硬件的情况下模拟各种物联网设备固件[48]。作者通过部署ADSL调制解调器的四个设备配置文件来评估蜜罐，并发现针对特定于仿真设备的漏洞的各种攻击。此外，Guarnizo et al.*Siphon*是一种可扩展的高交互蜜罐架构，它利用物理部署在地理位置并连接到互联网进行模拟的物联网设备[13]。作者在不同的位置部署了85个蜜罐实例，使用了5个物理摄像头，一个NVR和一个IP打印机。对攻击的分析显示，某些城市的蜜罐比其他城市收到更多的攻击流量Valeros等人 提供*Hornet 40*，这是一个地理位置放置的蜜罐网络数据集，用于研究地理位置的影响[44]。数据由118个特征组成，每个流包括480字节的内容。该数据集不包含交互式攻击流量，因为在研究中仅使用了被动蜜罐。攻击数据是从部署在八个地点的蜜罐中收集的，包含在40天内收集的470万个网络流量

Barron等人进行了一项为期四个月的研究，涉及102个中等交互蜜罐[3]。除了在多个位置部署蜜罐之外，作者还对参数进行了试验，如闯入和文件生成的难度他们观察这些参数的差异如何导致攻击者行为的偏差此外，作者还在黑客论坛和粘贴网站上泄露蜜罐的访问信息，以监视攻击者作者发现，引入的参数差异影响了基于人类的攻击者，并列出了实验中的关键要点我们认为这项工作最接近我们的方法，作者在研究中引入了特定的参数以及它们如何影响攻击者。

附录表6提供了相关工作中提出的蜜罐蜜罐比较基于其源代码的可用性，支持的协议，交互水平，操作环境和已知的指纹技术。大多数提出的蜜罐都是开源的，支持多种协议。然而，我们观察到，没有高交互蜜罐是可用的，可作为开源。此外，我们观察到，对于开源蜜罐，已经提出了某些指纹识别方法。大多数蜜罐被设计为部署为虚拟环境，除了在硬件上运行的IoTPot。虽然大多数蜜罐都是在二进制交互级别上运行的，即在低或中或高相互作用水平下，RIoTPot能够在低、高或混合相互作用水平下操作。

表1中列出的相关工作总结了涉及部署蜜罐以研究不同攻击的研究

表面。然而，没有一项研究通过部署具有不同交互级别和多个协议的操作环境的蜜罐来比较攻击。此外，没有公共数据集提供基于交互级别和协议的多样化数据我们的目标是通过部署多个蜜罐实例来解决这个差距，这些蜜罐实例具有不同的交互级别和地理分布的操作环境此外，我们还分析和比较了在不同的物联网和OT应用协议上模拟的多个交互级别上收到的攻击

学习	互动 水平	学习 周期	地理上 分布式	部署
Honeycloud[7] （2019）	中等	12个月	是的，是 的	硬件、云
[27]第27话：我爱你	低	39天	不知道	物理的
开放供出租[40] （2021）	低、中	一个月	不知道	物理的
多面的 [52]第二季第15集	低	两年	不知道	物理的
[2019]第48话	高	14天	不知道	物理的
[13]第13集	高	两个月	是的，是 的	物理的，云
大黄蜂40[44]（ 2021）	被动式	40天	是的，是 的	云
[2017]第三季第3集	中等	4个月	是的，是 的	物理的，云
RioTPot（2022）	**低、高、混合**	**3个月**	**是的，是 的**	**物理的，云**

表1：相关工作评价参数概述

3 扩大骚乱

RIoTPot旨在解决可扩展性，交互和操作环境的限制[41]。它遵循一个模块化的架构，并提供混合互动水平。 我们扩展了RIoTPot [1]以适应该纵向研究以及各种增强（也参见图1）。RIoTPot提供多种功能，如可扩展性、操作模式以及与各种部署环境的兼容性。

RIoTPot的动机是为管理员提供在其基础设施中部署蜜罐的便利性，特别是容器形式的高交互蜜罐。RIoTPot是开源的，可以在虚拟基础设施上运行，不像其他高交互蜜罐。此外，RIoTPot为管理员提供了根据资源更改交互级别的灵活性。RIoTPot能够开始部署为低交互，并在运行中切换到高交互RIoTPot的设计重点是模块化，以促进附加协议的集成和维护。许多开源项目经常由于缺乏可扩展性而被放弃模块化体系结构通过以模块的形式提供可扩展性来解决这个问题对于低交互模式，蜜罐管理员可以使用默认模板添加新的模拟，并轻松地将其与启动配置集成。类似地，对于高交互模式，提供容器图像的路径以用于模拟。通过模块化实施，管理员可以将任何相关场景添加到RIoTPot模拟产品组合中。许多蜜罐项目面临着指纹识别的威胁（见第2节）。指纹识别使对手能够根据通过精心制作的请求探测获得的基本信息来检测蜜罐。这对于经常利用特定库或可以被指纹识别的硬编码响应的低交互和中等交互蜜罐特别有害然而，这样的蜜罐具有低维护性和低风险，并且因此是有吸引力的欺骗机制。RIoTPot的混合交互特性提供了蜜罐管理员可以选择在低、高或混合交互中操作仿真Hybrid允许一些协议在低交互下运行，一些在高交互下运行。这允许防御者以类似面包屑的方法将攻击者引诱到特定的协议中。例如，管理员可以在高交互上运行他们感兴趣的协议，而在低交互上运行一些其他（攻击者所期望的）协议。然后，攻击者最终将大部分时间和精力花费在高交互协议上。

虽然许多蜜罐项目的开发重点是容器化部署，但RIoTPot还提供了一种混合部署方案，管理员可以选择在远程主机或本地基础设施上运行高交互容器。此功能在资源受限的环境中非常有用。例如，RIoTPot可以部署在树莓派上，而高交互容器可以部署在云环境中。此外，如果模拟简档需要特定硬件（例如，传感器），RIoTPot模拟容器可以部署在支持基础设施上。

[1] https://github.com/aau-network-security/riotpot

3.1 RIoTPot扩展架构

RIoTPot的架构遵循模块化，能够快速集成协议和仿真模块[41]。模块化软件架构是一种将软件组件构建为模块的结构方法，通过将程序的功能分离为独立的可互换模块，使得每个模块包含仅执行所需功能的一个方面所需的一切[21]。图1显示了从RIoTPot [41]改编的扩展架构。 除了快速集成附加模块之外，该架构还便于集成扩展以支持扩展分析或配置。该架构中的突出模块是*RIoTPot核心*模块、*低交互*模块、*高交互*模块和*攻击数据库*。数据*包捕获和噪声过滤器*模块用作扩展，以支持对攻击数据的进一步分析

3.2 扩展组件

3.2.1 RIoTPot核心。 *RIoTPot核心*由支持配置、管理和组织服务的基本组件组成。核心模块便于管理员配置参数，如指定用于仿真的协议、期望的交互级别以及在高交互模式的情况下用于加载图像的路径。启动配置允许用户选择所需的协议和交互级别。除了配置之外，核心还促进高交互模式中的容器之间的网络管理基于攻击所针对的模拟协议，将流量转发到容器此外，在基于云的部署的情况下，核心负责与远程容器的通信 我们扩展的配置模块的核心，通过增强的静态文件为基础的配置，以外壳为基础的交互式配置。基于shell的配置提供交互式提示，管理员可以在其中选择所需的启动配置。提示包括选择用于仿真的协议，选择交互级别作为操作模式，在高交互模式下操作的情况下提供远程数据库、*pcap*存储库和容器映像

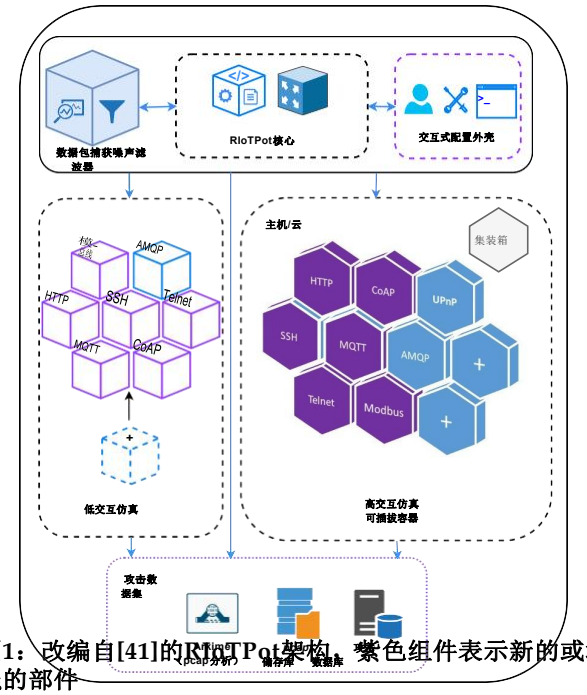


图1：改编自[41]的RIoTPot架构。紫色组件表示新的或增强的部件

3.2.2 *低交互模块*。低*交互模式*通过实现模拟各个协议的包来实现。RIoTPot是用Go语言实现的[11]，可以开发独立的包。RIoTPot的模块化架构是通过将协议开发为独立的包来实现的，这些包可以进一步集成为插件。例如，Telnet和SSH协议使用*fakeshell*包，该包模拟系统shell并响应命令列表。*fakeshell*包可以扩展为包含更多命令。RIoTPot提供了一个默认模板，可用于集成其他协议。默认情况下，RIoTPot支持七种协议的仿真，包括Telnet、SSH、HTTP、MQTT、AMQP、Modbus和CoAP。我们通过增强Telnet、SSH、HTTP、MQTT、Modbus和CoAP模块来扩展RIoTPot，以适应我们的研究。这些变化包括扩展fakeshell模块的shell仿真功能，以及增强Modbus、MQTT和CoAP协议的仿真功能。

3.2.3 *高互动模块*。RIoTPot通过将协议模拟为在容器映像上运行的服务，在高交互模式下运行。对于每个已配置的协议，管理员提供一个相关的映像，该映像将部署在容器上以进行仿真。由于协议作为容器上的完整服务操作，因此它们充当提供协议的完整实现的高交互模块，从而为攻击者提供高交互能力。通过在启动配置中指定协议和映像路径，可以集成其他协议。通过利用Docker提供的容器镜像，在这项工作中扩展了高交互模块Hub存储库[8]。使用Docker Hub的一个优点是，它为大多数镜像（例如，验证Apache基金会是httpd映像和busybox映像的BusyBox的发布者我们使用Busybox [4]，OpenSSH [17]，HTTPD [31]，Modbus-Server [25]，Eclipse-Mosquitto [30]和CoAP-Gateway [28]映像来实现协议的高度交互。

3.2.4*混合交互*。混合交互模式允许为选择性协议选择期望的交互水平。通过混合交互模式，像SSH这样的特定协议可以在低交互模式下运行，而另一个（例如，HTTP）可以在高交互模式下运行这有助于管理员设置设备配置文件，这些配置文件构成了资源需求较少的协议集合，并且可以在启动期间通过交互式外壳配置提示选择混合操作模式。

3.2.5 *噪声过滤器和数据包捕获*。RIoTPot有两个默认扩展-攻击捕获组件和噪声过滤器。攻击捕获组件使用*tcpdump*将RIoTPot上接收的所有流量存储为*pcap*文件，以便于进行全面分析。通过攻击捕获扩展，用户可以进一步指定数据包捕获所需的循环级别攻击捕获组件负责使用tcpdump将攻击数据包存储为*pcap*文件，该文件可用于详细分析（例如，深度分组检查）。噪声过滤器组件过滤掉从Shodan [38]和Censys [6]等互联网扫描器接收的流量。该组件可以过滤来自19个互联网扫描服务的流量。在攻击数据库中相应地标记攻击源。这有助于管理员通过消除此类服务产生的干扰来集中注意力于重要的攻击。

3.2.6 *攻击数据集。RIoTPot上接收的流量存储在攻击数据库中*。它被配置为一个独立的容器，以确保在发生崩溃或故障时不会中断日志记录。在低交互模式和高交互模式下接收到的攻击都存储在数据库中。可以从低交互容器和高交互容器访问数据库实例。为了促进这种纵向研究，我们扩展了RIoTPot，将所有攻击记录到云中的远程数据库，以确保可扩展性和简化的备份过程。我们通过集成*Arkime进一步增强了RIoTPot，Arkime*是一种开源索引数据包捕获和搜索工具[51]。我们利用Arkime搜索从蜜罐部署生成的*pcap*文件。Arkime从*pcap*存储库导入pcap文件，并将其存储在后端（Elasticsearch）中，以实现索引搜索功能。此外，Arkime支持基于属性的查询以及与Virustotal的集成[49]，这有助于识别恶意事件和来源。

4 方法论

我们的工作旨在捕获对物联网和OT环境的攻击，并通过利用其混合交互操作功能来评估RIoTPot此外，我们施加六个评价参数，以观察他们对我们的实验的影响，并比较结果。我们在以下章节中描述了我们的纵向研究的评价参数和实验设置。

4.1 评价参数

4.1.1 交互水平。 由于RIoTPot可以在低，高和混合交互水平下运行，因此我们研究并比较了每个交互水平上收到的攻击。 基于交互水平的分析将提供对在每个交互水平上使用的欺骗的有效性的洞察。

4.1.2 多个蜜罐实例。 通过部署多个实例的蜜罐，我们得到了更好的理解所观察到的攻击。例如，来自在所有蜜罐实例上识别的特定IP的攻击可以解释为攻击源是互联网范围的扫描服务或来自机器人的侦察过程的一部分。相反，在特定情况下识别的独特攻击源提供了对不同攻击类型和方法的洞察。

4.1.3 部署基础设施。 蜜罐的部署基础设施在欺骗层中起着重要的作用。 虽然一些应用协议在云环境中是开放的是常见的（例如，Telnet，SSH，MQTT，HTTP），在云上使用Modbus等协议是很奇怪的。包括用于模拟的容器、攻击数据库和*pcap*文件存储库的云基础设施在Digital Ocean云上被提供有限制出口流量的配置。 我们评估RIoTPot在自托管实验室以及云基础设施，研究在不同的设置部署蜜罐的影响。

4.1.4 地域分布。 有关于在不同地理位置部署蜜罐的影响的研究[45]，因此我们通过在不同大陆和国家部署蜜罐来考虑这一点。特别是，我们的实验在四个地理位置进行，即：纽约市（云），法兰克福（云），新加坡（云）和丹麦（实验室基础设施），以审查特定区域的攻击。这样，可以分析攻击数据以发现特定于区域或潜在特定于区域的恶意软件变体的攻击。

*4.1.5 协议仿真。*我们研究了六种应用协议，即：Telnet、SSH、HTTP、MQTT、Modbus和CoAP。 选择这些协议的原因是拥有一个混合仿真产品组合，其中包括自托管和云基础设施以及物联网和ICS环境中最常用的应用程序协议。协议被模拟为模块形式的低交互和专用的短暂容器中的高交互。对攻击的分析提供了由错误配置导致的特定于协议的威胁和攻击趋势。

*4.1.6 研究期间RIoTPot*的评估为期三个月，针对自托管和云环境，这些环境会产生在每个RIoTPot实例上捕获的大量攻击流量的数据集。随着时间的推移收集的攻击提供了对每个协议模拟的攻击趋势此外，我们运行Conpot蜜罐来比较从RIoTPot接收到的攻击 本研究于2021年12月10日至2022年3月10日进行。

4.2 实验装置

我们打算在不同的环境、交互模式、地理位置和模拟环境中部署RIoTPot，以全面了解袭击情况 实验分布在我们的实验室和云基础设施中，以便于评估和参数。我们以三级相互作用模式-低、高和混合相互作用部署RIoTPot以进行进一步评估。我们将在以下部分描述我们实验室的实验设置和云基础架构。

*4.2.1 实验室设置。我们实验室*的实验设置如附录图8所示。将RIoTPot部署在三个宿主Rl（高相互作用）、R2（低相互作用）和R3（混合相互作用）上。除了RIoTPot之外，Conpot [34]蜜罐也部署在主机C1上（中等交互）。所有四台主机都连接到Internet，并在未过滤的网络上配置了公共IP地址然而，主机配置有有限的出口流量，以避免滥用蜜罐。 由RI-oTPot产生的容器作为周期性地重新产生的短暂实例运行，以避免感染传播并从可用性崩溃中恢复。来自所有主机的攻击数据以分区、单独和循环文件的形式存储在远程文件存储库中，以便于进一步分析。所有的攻击流量都存储在攻击数据库中，便于查询和分析。在远程系统上提供攻击数据库和文件存储库，以避免在系统故障的情况下中断日志记录。 主机R3在混合交互模式下运行，其中SSH、MQTT、Modbus和CoAP在高交互模式下运行，Telnet和HTTP在低交互模式下运行。

*4.2.2 云设置。云基础设施*的实验设置如附录图9所示。与实验室设置类似，云实例作为Droplets在Digital Ocean上提供，并具有12个蜜罐实例（R4-C4）。12个蜜罐实例分布在三个地理位置-纽约市，法兰克福和新加坡，并相应地配置了公共IP地址。与实验设置类似，来自所有实例的攻击流量都存储为*Pcap*文件和在专用远程系统上运行的数据库中容器被定期地重新产生并且用静态配置文件重新供应来自所有容器的出口流量受到限制，以防止易受攻击环境的潜在误用。数据库配置有弹性模型，以支持从蜜罐实例收集的大量攻击流量数字海洋液滴监测有助于跟踪蜜罐实例的状态，这有助于识别任何故障情况[24]。

4.2.3 总结。 表2总结了评价的实验设置。 为了总结第4.1节中描述的纵向研究的评估参数，我们在三个交互级别（低，高，混合），两个部署环境（实验室，云），每个交互级别12个独立主机，四个地理位置（丹麦（实验室），纽约市，法兰克福和新加坡），六个应用协议仿真（Telnet，SSH，HTTP，MQTT，Modbus，CoAP），与中等交互中的一个蜜罐（Conpot）进行比较，评估期为三个月（2021年12月10日至2022年3月10日）。

4.3 数据集

来自研究的所有蜜罐实例上接收的流量存储在攻击数据库中，并作为*pcap*云中的*pcap*文件。输入

主机	环境方面	地理位置	交互层	仿真协议
R1	实验室	丹麦Name	高	Telnet、SSH、HTTP、MQTT、Modbus、CoAP
R2	实验室	丹麦Name	低	Telnet、SSH、HTTP、MQTT、Modbus、CoAP
R3	实验室	丹麦Name	混合动力	高-SSH、MQTT、Modbus、CoAP 低- Telnet、HTTP
C1	实验室	丹麦Name	中等	Telnet、SSH、HTTP、Modbus、S7
R4	云计算	纽约市	高	Telnet、SSH、HTTP、MQTT、Modbus、CoAP
R5	云计算	纽约市	低	Telnet、SSH、HTTP、MQTT、Modbus、CoAP
R6	云计算	纽约市	混合动力	高-SSH、MQTT、Modbus、CoAP 低- Telnet、HTTP
C2	云计算	纽约市	中等	Telnet、SSH、HTTP、Modbus、S7
R7	云计算	法兰克福	高	Telnet、SSH、HTTP、MQTT、Modbus、CoAP
R8	云计算	法兰克福	低	Telnet、SSH、HTTP、MQTT、Modbus、CoAP
R9	云计算	法兰克福	混合动力	高-SSH、MQTT、Modbus、CoAP 低- Telnet、HTTP
C3	云计算	法兰克福	中等	Telnet、SSH、HTTP、Modbus、S7
R10	云计算	新加坡Name	高	Telnet、SSH、HTTP、MQTT、Modbus、CoAP
R11	云计算	新加坡Name	低	Telnet、SSH、HTTP、MQTT、Modbus、CoAP
R12	云计算	新加坡Name	混合动力	高-SSH、MQTT、Modbus、CoAP 低- Telnet、HTTP
C4	云计算	新加坡Name	中等	Telnet、SSH、HTTP、Modbus、S7

表2：实验装置概述

在这项纵向研究中，我们在三个月的时间内从RIoTPot的12个蜜罐实例和Conpot的四个实例数据集是数据库转储和*pcap*文件的集合*pcap*文件捕获来自蜜罐实例的入口和出口流量。 数据集基于蜜罐实例、协议、地理位置和交互级别进行隔离。通过在数据库中标记业务来执行从入口业务过滤扫描服务。扫描服务的标记事件的标记数据集可以从攻击数据库中导出。目前，检查数据集的19个扫描服务。在*pcap*文件上捕获的流量是"数据包缓冲的"，因此输出在每个数据包的末尾而不是在每行的末尾写入pcap文件。管理流量被排除在*pcap*文件和攻击数据库之外。 *pcap*文件定期循环（每天）。该数据集将根据要求提供给学术研究人员，并遵守保密协议[2]。

5 评估

为了全面概述研究期间的发现，我们根据评估参数对结果进行了分解。以下各节讨论了调查结果

5.1 相互作用水平

我们将在以下章节中讨论基于交互水平的研究结果。

5.1.1 事件总数 图2显示了基于低、高和混合交互水平的所有RIoTPot实例上的事件总数。与低交互和混合交互相比，高交互水平接收到更高的事件。所有实例共收到1087万起事件，其中32%（3，487，877）来自低水平，35%（3，788，435）来自高水平，其余33%（3，600，823）来自混合相互作用。 总事件包括从互联网范围的扫描探测接收的探测业务（例如，[6]，[38]）。附录图10按交互水平显示了每天收到的事件百分比。从一开始，我们就看到，与低交互水平和混合交互水平相比，在高交互水平上接收到的事件有所增加。我们看到，2021年12月13日至15日，与2022年2月13日至20日的事件数量存在明显差异一个

[2]https://doi.org/10.11583/DTU.21088651

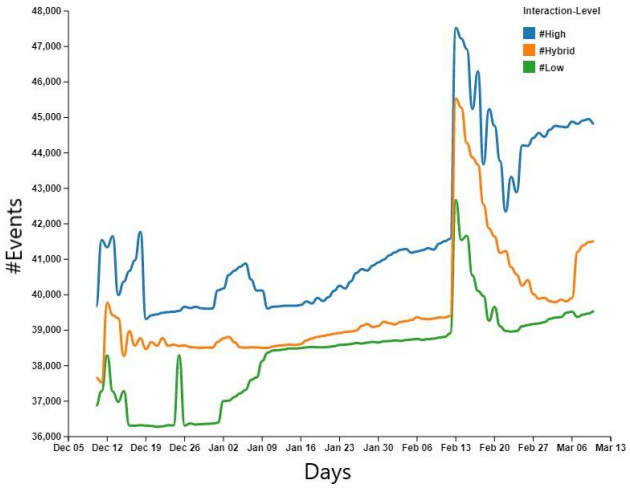


图2：按相互作用列出的事件总数

对这种不确定性的可能解释可以是混合交互级别涉及模拟协议上的低交互级别和高交互级别。 我们在第6.3节中讨论了偏离的可能原因。

5.1.2 事件分类。 一共10个。在所有RIoTPot部署上接收到8700万个事件表3按类型和相互作用水平总结了事件分类。在全部事件中，56%的流量是从互联网扫描服务（例如，Shodan [38]，Censys [6]）。我们认为来自扫描服务的流量是良性的，因为它们的扫描背后的意图是已知的。过滤掉良性流量简化了对具有恶意意图的流量进行聚焦的分析过程RIoTPot的噪声过滤器识别并标记了总共19个独特的扫描服务[3]，该噪声过滤器具有良性扫描服务的数据库。 虽然经常会观察到来自Shodan和Censys等扫描服务的重复流量，但一些扫描每天发生多次，而其他服务则遵循不同的模式，从几天到几周不等。请注意，我们在RIoTPot中模拟了六个协议，一些扫描针对特定端口范围，一些扫描针对具有自定义请求的特定端口[43]。最后，我们没有检测到任何偏差，在收到的扫描服务流量的基础上的互动。

未标记为扫描服务的流量将被分类
是恶意的恶意分类包括可疑流量和在请求或有效载荷中具有明显恶意意图的流量两者。 可疑流量包括来自未知源的探测流量和可能的反向散射噪声。由于蜜罐没有任何生产价值，我们认为任何通信，不包括上述扫描服务，对他们可疑。
根据我们的标准，有480万个事件被标记为恶意事件请注意，此处所述的恶意事件数量并不因攻击源而异。我们观察到多个攻击来自同一攻击源的流量。基于交互级别对收到的恶意流量的进一步分类如表所示

[3] https://github.com/aau-network-security/riotpot#12-Noise-Filter

至9日，德克萨斯州奥斯汀

交互层	偶型	计数
低交互作用	扫描服务	2.*第一章* 02个月
高互动	扫描服务	2.*第一章* 02个月
混杂交互作用	扫描服务	2.*第一章* 02个月
低交互作用	恶毒	1 .一、 46个月
高互动	恶毒	1 .一、 76海里
混杂交互作用	恶毒	1 .一、 57M
扫描服务事件总数		**6.07百万**
恶意事件		**4.8百万**
事件总数		**10.87百万**

表3：按类型和相互作用水平列出的事件总数

3. 我们观察到，高互动的情况下，收到更高的音量比低和混合互动水平。

*5.1.3 按交互级别划分的攻击源。*图3显示了在几天内从恶意流量中识别的唯一IP地址的数量和交互级别。我们观察到独立IP地址总数在几天内稳步增加，从2022年2月13日达到峰值，随后在接下来的四天内下降此外，我们观察到，高交互级别的实例收到的攻击，从更多的独特的来源比其他的互动水平。

互动 级别	#恶意 活动内容	#唯一 IP
高互动	1 763 395	十八*四百*三十一
混杂交互作用	1575807	12618
低交互作用	1 463 883	*八千*六百三十五
不同的IP 所有交互水平	22518	

表4：恶意事件和唯一IP的汇总

表4按交互级别总结了唯一源IP地址的不同累计总数。在高交互实例上检测到最大数量的唯一IP。我们在所有恶意事件中识别出22，518个唯一IP我们要强调的是，在我们的研究中，RIoTPot模拟了六种协议服务，表4中总结的独特攻击源基于通过这些协议模拟接收的流量。

5.2 部署基础设施

RIoTPot部署在实验室（自托管）和云基础设施进行评估。实验室基础设施托管三个实例，而云基础设施托管九个RIoTPot实例。图4显示了基于托管基础设施的RIoTPot实例上接收到的恶意事件的分布。云基础设施上的事件数量很高，因为与实验室（3个实例）相比，部署了更多的RIoTPot实例（9个实例）。此外，该图还显示了实验室和云基础架构中每个实例我们观察到云实例比实验室实例具有更高数量的恶意事件 这可能是因为任何可疑的扫描或特定于区域的恶意请求。

在图4中，总结了实验室和云基础设施上每个交互级别的恶意事件数量虽然流量上有偏差，但恶意事件数量

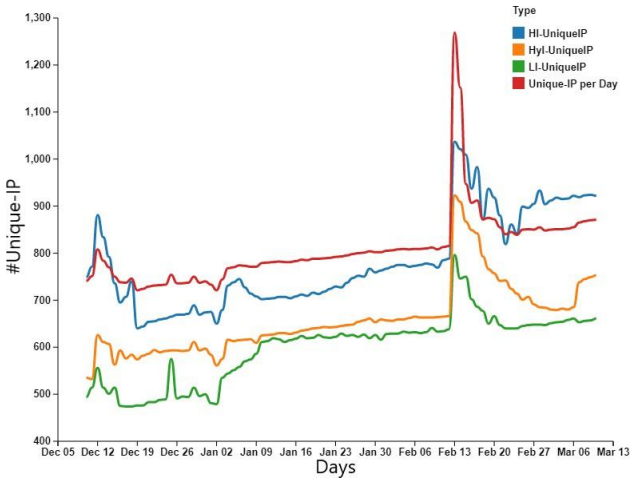


图3：一天内的唯一IP和相互作用

在所有的交互水平上都是随着时间而增加的高交互实例比低交互和混合交互水平收到更多的攻击。请注意，云上的恶意事件数量高于实验室中的恶意事件数量，因为与实验室环境相比，云基础设施上部署的RIoTPot实例更多。我们观察到两种操作环境中恶意事件趋势的微小变化。

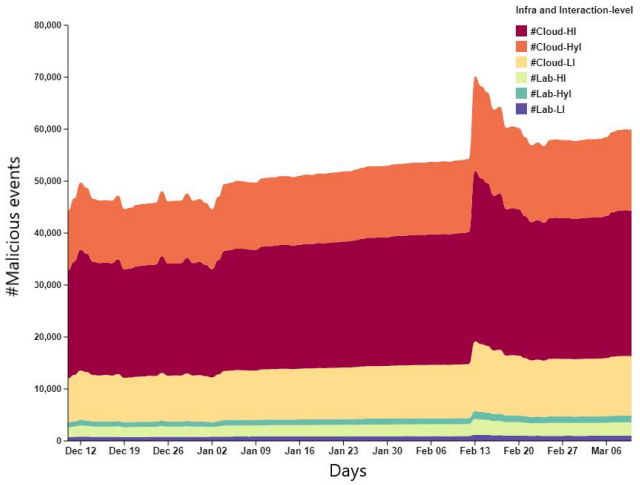


图4：按基础架构和交互划分的恶意事件总数比较

5.3 地理位置

为了研究地理位置和特定区域攻击分布的影响，我们在四个地点部署了RIoTPot-纽约市，法兰克福，新加坡和丹麦（实验室）。图5显示了每个位置和交互级别的恶意事件分布相互作用水平是用颜色编码的，实心球体表示通过相互作用的日常事件的数量。球体的半径与图例中表示的事件数成比例。

每天每次交互收到的最低攻击次数为743次，而最高为13，287次。 与云实例相比，实验室实例接收到的恶意事件显著更低。最初，纽约实例的流量较高;然而，法兰克福实例总体上接收了最高的流量。在整个评估期间，与其他云部署相比，部署在新加坡的实例报告的流量最低 我们观察到特定于该地区的可疑事件，这些事件在其他云实例中没有看到。可疑事件包括端口扫描、暴力破解尝试和特定于RIoTPot模拟的协议的我们发现特定区域的良性扫描，从已知的实体，如教育机构和政府援助的组织以外的恶意事件。在附录图14中，我们进一步讨论了位置和云与实验室部署与攻击次数有关

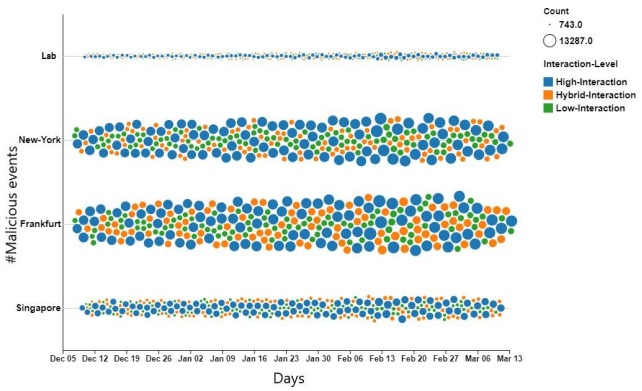


图5：恶意事件在实验室和云部署

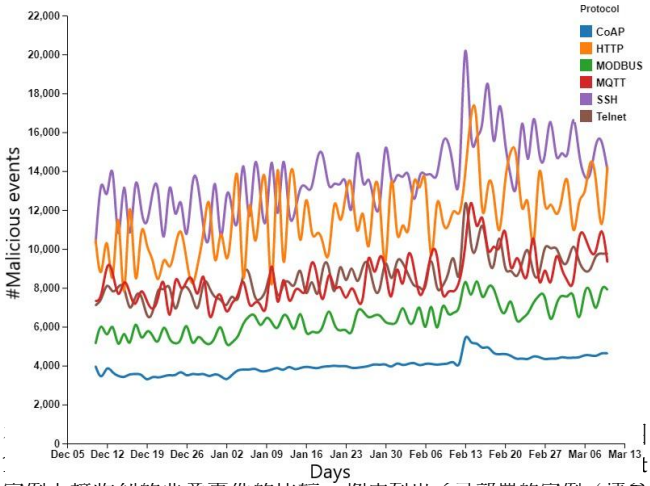
5.4 仿真协议

RIoTPot模拟六种协议-Telnet、SSH、HTTP、Modbus、MQTT和CoAP。协议在部署中的不同交互级别中进行仿真表2总结了在实例上仿真的各个协议的交互级别图6显示了每个协议上记录的恶意事件的数量 我们观察到SSH协议上的事件数量最多，其次是HTTP，Telnet，MQTT，Modbus和CoAP。请注意，每个源IP的事件数量不是唯一的，而是在所有相互作用水平上观察到的事件总数。

附录图13按交互级别总结了每个方案收到的恶意事件 我们观察到，在所有协议仿真中，恶意事件的最高数量是在高交互实例上接收的。 在Telnet、SSH、MQTT和Modbus等协议中，我们观察到低交互实例上的事件数量逐渐减少。观察到许多攻击类型，如蛮力攻击、中毒攻击、旋转攻击和反射攻击。 攻击类型将在第6.1节中进一步讨论。

5.5 与Compot

为了比较RIoTPot实例上收到的攻击，我们部署了Conpot [34]，这是一种中等交互的蜜罐，可以模拟SSH，Telnet，HTTP，Modbus和S7协议。我们部署康波特



实例上接收到的恶意事件的比较。图中列出了已部署的实例（请参见表2）和每个实例接收的恶意事件总数。Conpot实例模拟了四种协议（Telnet、SSH、HTTP和Modbus），可以与RIoTPot实例模拟的协议进行比较。与Conpot相比，我们观察到RIoTPot在高交互和混合交互实例上接收到更高数量的事件，并且在低交互水平实例下接收到类似数量的事件。该图还比较了在由托管基础设施、位置交互级和仿真协议部署的每个RIoTPot实例上观察到的恶意事件的数量 我们观察到，部署在法兰克福云基础设施（R7）上的实例收到的恶意事件数量最多。 实例R1、R2、R3、C1部署在实验室; R4、R5、R6、C2部署在纽约市; R7、R8、R9、C3部署在法兰克福; R10、R11、R12、C4部署在新加坡。我们怀疑，恶意事件的数量之间的差异，Conpot和RIoTPot可能是由于有限的交互能力的Conpot相比，RIoTPot。

6 讨论情况

本节讨论在评估过程中观察到的攻击类型，以及我们根据评估参数对收到的各种恶意事件的发现。 我们进一步陈述了我们方法的局限性，以及我们方法中的伦理考虑。

6.1 恶意事件

我们总共收到了480万个恶意事件。请注意，所有未标记为扫描服务的事件都被归类为恶意事件。恶意事件还包括请求或有效载荷中的具有恶意意图的流量。我们观察到不同的攻击类型在所有实例上收到的恶意流量以下各节将讨论评估期间观察到的攻击类型和独占攻击。

6.1.1 按交互级别划分的攻击类型 图7显示了在我们的评估过程中观察到的攻击类型的概述（按百分比-年龄和交互级别）。我们观察到的攻击类型包括暴力攻击、中毒攻击、反射攻击和来自未知扫描器的端口扫描 暴力攻击是所有实例中最常见的攻击，并且针对所有模拟协议。 仿真协议配置了弱访问控制和凭据，以捕获高级攻击类型。在所有的互动水平上观察到持续的暴力攻击量。此外，我们还看到来自所有交互级别的相同参与者（IP）的暴力攻击。然而，在一些特定情况下观察到一些区域性攻击，其中攻击来源似乎来自同一个大陆。中毒攻击集中在未经授权的访问后修改数据。例如，我们发现CoAP协议上的消息以修改数据。在高交互水平上观察到更大数量的中毒攻击此外，我们观察到，来自同一攻击源的攻击中断了低交互实例上的连接，而追求高交互实例上的连接。有了这个，我们需要威胁行为者使用来自会话的特定信息来确定攻击的追求。

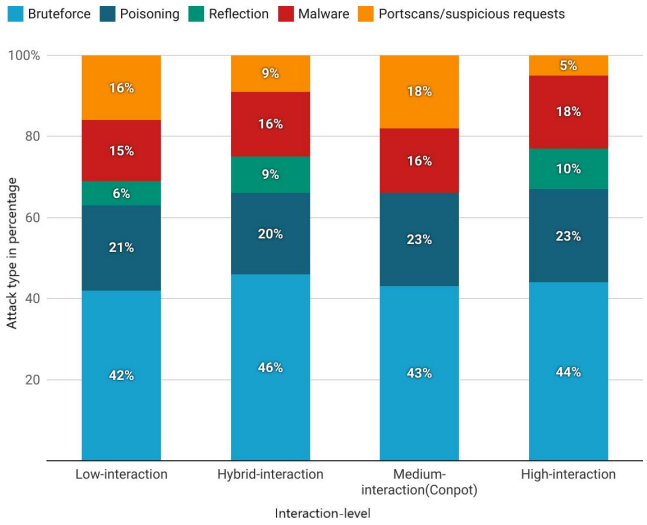


图7：按交互划分的

在CoAP协议上检测到反射攻击当目标地址端口为端口80，源端口为5832时，我们识别出反射攻击在高交互实例中再次观察到更大量的反射攻击。然而，我们承认反射攻击可能是反向散射流量的一部分。我们观察恶意软件注入攻击，攻击者试图从恶意链接下载恶意软件 通过分析pcap文件中记录的攻击 *来识别恶意链接*。在有效载荷中发现可疑链接后，我们使用Virustotal [49]检查链接以确定恶意性。我们观察到Mirai [2]恶意软件的多个变体以及LuaBot、Mozi
[19]和BrickerBot，其中。最后，我们观察到一个特定数量的非重复性端口扫描流量，这是不相同的
已知的扫描服务或攻击类型。 我们将此类流量分组为portscans和可疑请求。

6.1.2 按协议划分的攻击类型 附录图12显示了仿真协议收到的攻击类型的百分比。攻击流量以pcap文件和会话日志的形式存储在攻击数据库中。我们总结了在每个仿真协议上发现的攻击类型

Telnet 和SSH 。 Telnet和SSH协议受到了大量的暴力攻击。Telnet协议还在成功的暴力破解尝试中接收到某些恶意软件注入 在与Virustotal进行检查后，在Telnet攻击中观察到的恶意软件链接被检测到是Mirai家族的变体。类似地，在SSH协议上检测到Mirai的几个变体。除了普通的暴力攻击和恶意软件注入之外，还观察到许多端口扫描这意味着仍然有许多参与者在寻找易受攻击的Telnet和SSH实例。

HTTP 。 HTTP协议模拟了Siemens LOGO 230 RCEo Modbus控制器的静态登录页面。 该协议收到了大量的暴力尝试。 除了暴力攻击之外，HTTP协议还受到许多Log4j攻击，尽管Apache Web服务器[31]用于仿真。攻击者试图通过远程服务器的RMI（远程方法调用）调用进行注入攻击。最后，发现了大量的网络抓取器以及未知的扫描服务。

MQTT 。 MQTT协议是使用Eclipse-Mosquitto [30]镜像（用于高交互）和库（用于低交互）进行仿真的。尽管该协议被配置为允许匿名登录，但仍存在大量的暴力破解攻击。此外，在攻击者试图修改队列中的数据的情况下，检测到几次数据中毒尝试。此外，我们检测到一些攻击创建了新的主题，并试图专门访问*SYS$*主题。该协议主要由良性扫描服务扫描，然而，在法兰克福和新加坡部署的实例中检测到一些区域性可疑扫描。

Modbus 。 Modbus协议收到了大量的poi- soning攻击，以读取和修改来自寄存器和线圈的数据。据观察，攻击的目标是十九个功能代码中的三个，这些功能代码可用于获取设备、报告服务器和持有寄存器上的信息。此外，我们观察到，大多数的攻击使用无效的函数代码来访问寄存器中的数据。 这需要扫描搜索具有用于已知利用的特定功能代码的设备。

CoAP 。 CoAP协议被配置为在UDP端口5683上服务。检测到许多尝试访问CoAP服务的暴力攻击此外，我们确定数据中毒攻击，旨在通过发布消息修改值 除了数据中毒攻击之外，CoAP协议还出现了反射泛洪攻击，其中攻击者试图欺骗源包以将所有响应流量转移到受害者。 这种攻击是通过观察目的地港口来识别的。我们观察到27个受害IP，其中12个位于巴西，4个位于韩国，4个位于俄罗斯，3个位于中国，1个位于法国，1个位于德国。然而，我们发现受害者IP没有有效的域，并提供空白的HTML页面。

6.1.3 针对特定区域的攻击。 RIoTPot实例部署在四个地理位置。我们检测特定区域的目标实例的攻击和攻击源附录表7列出了仅在特定区域观察到的恶意流量来源的攻击类型和数量百分比。我们观察到来自特定攻击源的几种协议和攻击类型的攻击表中列出的唯一源IP表示专门针对该区域的攻击源虽然已知互联网扫描服务使用区域主机来扫描特定位置，但表7中列出的唯一IP来自恶意事件。为了过滤我们的结果，我们检查IP是Tor中继[32]还是来自VPN [14]，发现它们都不是。我们检查了互联网扫描服务（如Shodan [38]）上的IP源，发现主机的SSH端口是打开的。此外，在查找IP历史记录时，我们发现它们最近从ASN移动。

6.2 攻击源

从恶意事件中总共识别出22，518个唯一IP为了了解攻击源，我们尝试使用基于横幅的指纹技术来识别攻击源。我们通过使用Lift [4]（一种开源的低影响指纹识别工具）在IP上发送Telnet（端口23）上的连接尝试和端口80，8080和443上的HTTP请求。然后，我们检查横幅和响应，以查找攻击源的潜在标识符。我们注意在探测中发送最小数量的数据包，以限制与攻击源的通信量表5显示了通过横幅抓取检查识别的设备类型。通过横幅检查共识别出5264件（23%）器械。我们怀疑这些是导致互联网攻击的受损设备。除了受感染的设备，我们注意到绝大多数HTTP响应包含来自Apache，NGinX和Tengine Web服务器的默认测试网站。共确定了4218（19%）的此类响应。我们执行反向DNS查找以确定是否存在与IP地址范围相关联的任何域，并找到21个域。这些域名与一些通用顶级域名相关联，包括*.art*（5），*.games*（6），*.love*（3），*.website*（2）和*.webcam*（5）。最后，无法确定其余器械（58%）

设备类型	协议	计数
路由器	HTTP	1819
DVR	HTTP	1621
路由器	Telnet	721
IP电话	HTTP	311
开关	HTTP	287
开关	Telnet	211
IP打印机	HTTP	176
NAS	HTTP	118
合计		**5264**

表5：攻击源类型

6.3 我们的评估和发现表明，对于一些协议，对低交互蜜罐的攻击逐渐减少（见图10）。而高交互实例接收到更高数量的攻击事件。因此，我们的研究结果表明，互动水平在吸引特定的攻击中发挥着至关重要的作用我们对非重复扫描探针的逐渐减少的观察表明，现代扫描探针可以具有有助于表征被扫描系统是否是蜜罐[5]的检查。在混合交互模型上接收到的恶意事件表明，低交互仿真和高交互仿真的组合确实吸引了更多的攻击，并成功地欺骗了来自可疑扫描探针的检查总结交互级别的影响，低交互蜜罐在捕获扫描和bot流量方面仍然有效然而，我们建议部署高互动蜜罐有限的网络配置上的一些协议是更有效地实现更高的欺骗层。

[4]https://github.com/trylinux/lift

6.4 限制

我们承认我们的方法存在以下局限性首先，实验室基础设施仅限于一个位置，而云部署范围为三个位置。这种限制导致实验室部署和云部署之间的直接比较不均衡。如果部署的实例数量相同，则在实验室和云中部署的实例之间的比较将是描述性的。其次，我们在四个城市部署RIoTPot，将范围限制在三大洲。在所有大陆部署RIoTPot将提供更广泛的攻击前景。第三，我们将仿真协议的数量限制为六个。我们承认，更多的协议将为我们提供广泛的数据集进行分析。然而，这项工作的目的是可视化的影响，许多评估和设计参数，可以影响蜜罐的目的。第四，我们认为每个事件都是一个联系，这就需要在过度计算方面进行一些限制 由于连接项在协议中各不相同，因此我们按事件而不是连接进行计数。最后，数据集不将at-tack数据分组为Netflow格式。将数据存储为Netflow格式有助于与分析平台进行更广泛的集成

7 结束语

在这项工作中，我们扩展RIoTPot，一个模块化和混合互动的蜜罐，并促进纵向研究。为了确定蜜罐的交互级别等参数的影响，我们通过测量基于交互级别，部署的基础设施，地理位置和仿真协议等参数收集的恶意事件来对RIoTPot进行广泛的纵向评估。我们的研究结果表明，这些参数是必不可少的蜜罐研究，可以提供一个更广泛的概述的攻击景观。结果表明，高互动的蜜罐收到更复杂的攻击相比，低互动的蜜罐。此外，我们观察到特定于托管环境和地理位置的攻击。与Conpot相比，RIoTPot的高交互实例在所有评估参数上收到的恶意事件数量更高。我们观察到不同的攻击，如反射，数据中毒和恶意软件的蜜罐。最后，我们观察到大量的流量从扫描服务，可能会导致警报疲劳和误报。

[5]事实上，像Shodan这样的服务已经具备了这样的能力[37]。

致谢

这项研究得到了欧盟欧洲区域发展基金北海计划支持的区域间项目COM 3的支持。

参考文献

[1] P Dilsheer Ali和T.Gireesh Kumar2017年。Dionaea蜜罐中恶意软件的捕获与检测。在2017年*的电力和先进计算技术创新（i-PACT）*。IEEE，Vellore，India，1 https://doi.org/10.1109/IPACT。 2017.8245158

[2] Manos Antonakakis，Tim April，Michael Bailey，Matt Bernhard，Elie Bursztein，Jaime Cochran，Zakir Durumeric，J. Alex Halderman、Luca Invernizzi、Michalis Kallitsis、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（USENIX Security 17）*。USENIX协会，温哥华，BC，1093-1110。https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[3] 蒂莫西·巴伦和尼克·尼基福拉基斯。2017年。挑剔的攻击者：量化系统属性对入侵者行为的作用 在*第33届年度计算机安全应用会议*（Orlando，FL，USA）*（AC-SAC '17）的会议记录中*。计算机协会，美国纽约州纽约市，387https://doi.org/10.1145/3134600.3134614

[4] 爱管闲事 2022. *Busybox DockerHub*。BusyBox。https://hub.docker.com/_/busybox

[5] 沃伦·Z放大图片创作者：James F.Sikos和Samuel G.韦克林2021年。用于网络欺骗的Conpot及其BACnet特征分析在*安全、网络和物联网的进展*中，Kevin Daimi、Hamid R.Arabnia，Leonidas Dannidis，Min-Shiang Hwang，and Fernando G.Tinetti（编）.施普林格国际出版社，Cham，329-339。

[6] Censys 2021年。Censys搜索。2021年6月28日检索自https://censys.io/

[7] FanDang，Zhenhua Li，Yunhao Liu，Ennan Zhai，Qi Alfred Chen，Tianyin Xu，YanChen，and Jingyu Yang. 2019年。使用HoneyCloud了解对基于Linux的IoT设备的无文件攻击。*第17届移动系统、应用和服务国际年会*（韩国首尔）*（MobiSys '19）*。计算机协会，美国纽约州纽约市，482-493。https://doi.org/10.1145/3307334.3326083

[8] Docker 2022年*DockerHub*。Docker https://hub.docker.com/

[9] 迈克尔·多德森阿拉斯泰尔·R Beresford和Mikael Vingaard。2020.使用全球蜜罐网络检测目标ICS攻击。2020年*第12届网络冲突国际会议（CyCon）*，卷。一千三IEEE，爱沙尼亚，275https：//doi.org/10.23919/CyCon49761.2020.9131734

[10] 伊莉莎2020年。*ENISA威胁形势2020-恶意软件*。伊莉莎网址：//www.enisa.europa.eu/publications/malware

[11] Golang 2021年。去语言。2022年3月16日从https://golang.org/检索

[12] 灰噪2022. 灰噪 https://viz.greynoise.io/

[13] JuanDavid Guarnizo、Amit Tambe、Suman Sankar Bhunia、Martin Ochoa、Nils OleTippenhauer、Asaf Shabtai和Yuval Elovici。2017年。SIPHON：Towards Scalable High-Interaction Physical Honeypots（可扩展的高交互物理蜜罐）第*三届ACM网络物理系统安全研讨会*（阿拉伯联合酋长国阿布扎比）*（CPSS '17）的会议记录*。计算机协会，纽约，纽约，美国，57https：//doi.org/10.1145/3055186.3055192

[14] 伊芙布2022. *伊芙*布伊芙布https://iphub.info/

[15] Celine Irvene、David Formby、Samuel Litchfield和Raheem Beyah。2018年。HoneyBot：机器人系统的蜜罐。*Proc. IEEE*106，1（2018），61-70。https://doi.org/10.1109/JPROC.2017.2748421

[16] Xingbin Jiang，Michele Lora，and Sudipta Chattopadhyay.2020年。工业物联网设备安全漏洞的实验分析 *ACM Trans. 互联网技术20*，2，第16条（2020年5月），24页。https://doi.org/10.1145/3379542

[17] Linuxserver.io. 2022. *Openssh DockerHub*。OpenSSH。https://hub.docker.com/r/linuxserver/openssh-serve

[18] Samuel Litchfield，David Formby，Jonathan Rogers，Sakis Meliopoulos，and Ra- heem Beyah.2016年。海报：重新思考网络物理系统的蜜罐IEEE Symposium*on Security and Privacy（IEEE安全与隐私研讨会）*IEEE，San Jose，California.

[19] 微软。2021. *Mozi僵尸网络微软*。https://www.microsoft.com/security/blog/2021/08/19/how-to-proactively-defend-against-mozi-iot-botnet/

[20] 托马斯·米勒、亚历山大·斯塔维斯、萨姆·梅斯查克、米里亚姆·斯特迪和本杰明·格林。2021年。回顾过去，展望未来：从工业控制系统的网络攻击中吸取的教训。*国际j的。关键基础设施。波特35*，C（2021年12月），14页。https://doi.org/10.1016/j.ijcip.2021.100464

[21] Mawal Mohammed，Mahmoud Elish，and Abdallah Qusef.2016年。对设计模式背景的经验洞察：模块化分析。*2016*年第七届IEEE，安曼，乔丹1-6 https://doi.org/10.1109/CSIT.2016.7549474

[22] 丽莎·奥摩纳哥。2021年。*DAG摩纳哥在新闻发布会上就殖民地管道的黑暗面攻击发表讲话*。美国司法部。https://www.justice.gov/opa/speech/dag-monaco-delivers-remarks-press-会议-黑暗面-攻击-殖民地-管道

[23] Shun Morishita，Takuya Hoizumi，Wataru Ueno，Rui Tanabe，Carlos Gañán，Michel JG van Eeten，Katsunari Yoshioka，and Tsutomu Matsumoto. 2019年。如果你发现我。. .啊等等。全互联网范围内的自我暴露蜜罐。*2019*年IFIP/IEEE集成网络和服务管理（IM）研讨会。IEEE，IEEE，Washington DC，USA，134-143.

[24] 数字海洋2022. 数字海洋水滴监测。2022年3月16日从https://docs.digitalocean.com/products/monitoring/

[25] OITC。2022. *Modbus服务器DockerHub*。OITC。https://hub.docker.com/r/oitc/www.example.com

[26] 米歇尔·奥斯特霍夫2016年。Cowrie SSH/telnet蜜罐。https://github.com/micheloosterhof/cowrie

[27] Yin Minn 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。第*九届USENIX进攻性技术研讨会（WOOT15）*。USENIX协会，华盛顿特区https://www.usenix.org/conference/woot15/workshop-program/presentation/pa

[28] plgd 2022. *CoAP-Gateway*。plgd https://hub.docker.com/r/plgd/coap-gateway

[29] 德国电信公司蜜罐项目。2022年T-Pot：一个多蜜罐平台。

[30] Eclipse项目。2022. *Eclipse Mosquitto DockerHub*。Eclipse项目。https：//hub.docker.com/_/eclipse-mosquitto

[31] Apache HTTP服务器项目2022年*HTTPD DockerHub*Apache项目https://hub.docker.com/_/httpd

[32] Tor项目2022. ExoneraTor Tor项目https://metrics.torproject的网站。org/exonerator.html

[33] L Rist.2009年Glastopf项目。

[34] Lukas Rist、Johnny Vestergaard、Daniel Haslinger、A Pasquale和J Smith。2013年。Conpot ics/scada蜜罐。

[35] 弗朗西斯·罗伯斯和妮可·佩洛思。2021年。*“危险的东西”：黑客试图毒害佛罗里达镇的供水纽约时报*. https://www.nytimes。com/2021/02/08/us/oldsmar—florida—water—supply—hack.html

[36] Stewart Sentanoe、Benjamin Taubmann和Hans P.雷瑟2018年。瓶子草：使用虚拟机内省增强SSH蜜罐的性能和隐蔽性在*Secure IT Systems*中，Nils Gruschka（Ed.）。施普林格国际出版社，占，255-271。

[37] 秀丹2022年Honeypot or Not？https://honeyscore.shodan.io

[38] 秀丹2022. Shodan https://www.shodan.io/

[39] Shreyas Srinivasa、Jens Myrup Pedersen和Emmanouil Vasilomanolakis。2021. Gotta CatcharXiv：2109.10652[cs.CR]

[40] Shreyas Srinivasa、Jens Myrup Pedersen和Emmanouil Vasilomanolakis。2021. *公开招聘：物联网设备的攻击趋势和错误配置陷阱*。计算机协会，纽约，纽约，美国，195-215。https://doi.org/10.1145/3487552.3487833

[41] Shreyas Srinivasa、Jens Myrup Pedersen和Emmanouil Vasilomanolakis。2021. RIoTPot：一个模块化的混合交互IoT/OT蜜罐。2021年第*26届欧洲计算机安全研究大会（ESORICS）*。施普林格，施普林格，达姆施塔特，德国。

[42] 恐龙工具。2010. 网络蜜罐。 https://github.com/DinoTools/dionaea/

[43] R Trapkickin。2015年。谁在扫描互联网？

[44] 维罗妮卡·瓦莱罗斯和塞巴斯蒂安·加西亚。2022年Hornet 40：地理上放置的蜜罐的网络数据集*Data in Brief*40（2022），107795. 网址：//doi.org/10.1016/j.dib.2022.107795

[45] 维罗妮卡·瓦莱罗斯和塞巴斯蒂安·加西亚。2022年Hornet 40：地理上放置的蜜罐的网络数据集*Data in Brief*40（2022），107795. 网址：//doi.org/10.1016/j.dib.2022.107795

[46] Emmanouil Vasilomanolakis，Shankar Karuppayah，Max Mühlhäuser，andMath-ias Fischer. 2014年HosTaGe：一个用于协同防御的移动蜜罐 在*第七届信息和网络安全国际会议*（英国苏格兰格拉斯哥）*（SIN '14）的会议记录中*。计算机协会
New York，NY，USA，330 https://doi.org/10.1145/2659651.2659663

[47] Alexander Vetterl和Richard Clayton 2018.苦涩的收获：在互联网规模上系统地识别低交互和中等交互的蜜罐。第*12届USENIX进攻性技术研讨会（WOOT 18）*。USENIX Association，Baltimore，MD，9.https://www.usenix.org/conference/woot18/presentation/vetterl

[48] 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，Pittsburgh，PA，USA，1https：//doi.org/10.1109/eCrime47957.2019.9037501

[49] Virustotal。2022年Virustotal。https://www.virustotal.com

[50] 王建新，王明.Lim，Chao Wang，and Ming-Lang Tseng.2021年。物联网（IoT）在过去20年的发展 *计算机&工业工程*155（2021），107174。https://doi.org/10.1016/j.cie.2021的网站。107174

[51] 安迪·威克和社区2022年*Arkime*。Arkime。https://arkime.com/index#www.example.com

[52] Armin Ziaie Tabari和Xinming Ou。2020年。*多阶段多方面的物联网蜜罐生态系统*。计算机协会，纽约，纽约，美国，2121-2123。https://doi.org/10.1145/3372297.3420023

A 定性比较

表6提供了相关工作中提出 蜜罐比较基于他们的源代码的可用性，支持的协议，交互水平，操作环境和已知的指纹技术。大多数提出的蜜罐都是开源的，支持多种协议。然而，我们观察到，有没有高互动蜜罐可作为开源。

蜜罐	开放源码	支持协议	互动水平	虚拟与五金件	已知指纹图谱方法
对照[34]	是的，是的	SSH、Telnet、Modbus、BACNet、HTTP	中等的	虚拟的	是的，是的
[26]第二十六话	是的，是的	SSH、Telnet	中等的	虚拟的	是的，是的
格拉斯托夫[33]	是的，是的	HTTP、HTTPS	中等的	虚拟的	是的，是的
[27]第二十七话	是的，是的	不知道	低	硬件	不知道
Dionaea[42]	是的，是的	是的，是的	中等的	虚拟的	是的，是的
匈牙利[48]	不知道	基于图像	高的	虚拟的	不知道
[41]第四十一话	是的，是的	基于图像	低，混合。高的	虚拟的	不知道

表6：蜜罐的定性比较

B 附录：实验概述

B.1 实验室设置

我们的实验设置的实验室设置如图8所示。部署了RIoTPot R1、R2、R3的三个实例和ConpotC1的一个实例，并为每个实例分配了一个公共IP。从蜜罐接收和发送的所有流量都存储在远程文件系统中作为存储库，此外还将会话参数存储在攻击数据库中。

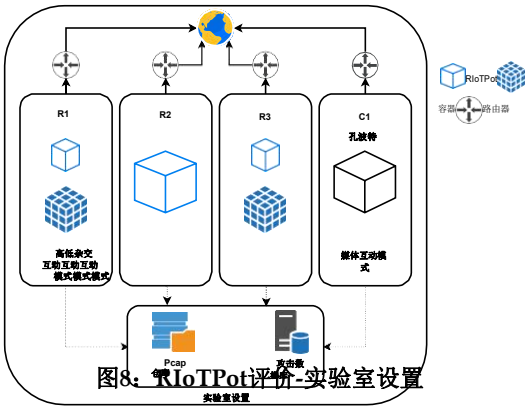


图8：RIoTPot评价-实验室设置

B.2 云设置

该方法的云设置如图9所示。云实例在三个地理节点（法兰克福、纽约市和新加坡）进行配置。

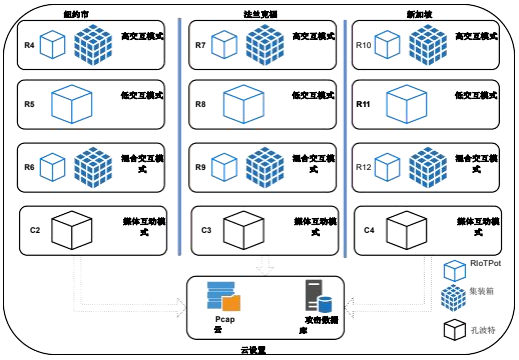


图9：RIoTPot评估-云设置

C 附录：补充结果

C.1 图10显示了基于交互水平的RIoTPot实例上接收的每日事件的百分比。我们观察到
当与高交互和混合交互实例上的事件相比时，低交互上的事件随时间的百分比我们怀疑这可能是因为有限的相互作用水平。

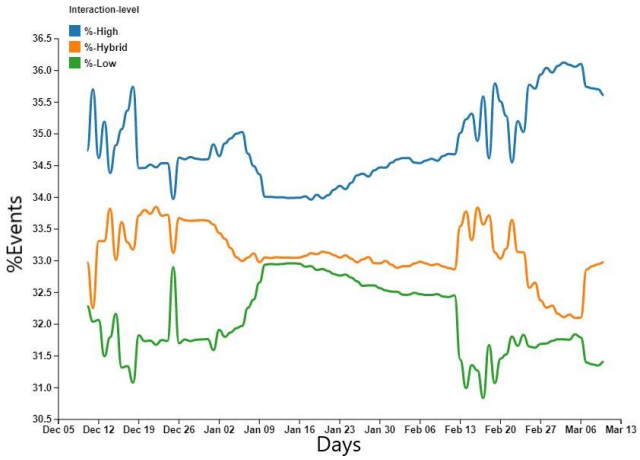


图10：按交互级别和百分比列出的事件百分比

C.2 通过交互级别、位置、蜜罐和仿真协议进行比较

图11示出了由RIoTPot（R）的蜜罐实例和具有Conpot（C）的仿真协议接收的攻击的数量的比较与其他部署相比，我们在RIoTPot的高交互实例上观察到大量的恶意事件。

C.3 按协议划分的攻击类型

图12显示了仿真协议上攻击类型的百分比。我们观察到多种攻击类型包括暴力攻击，

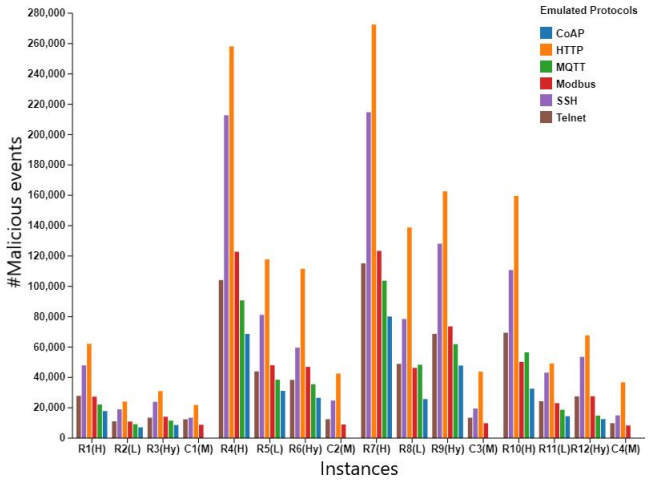


图11：按实例划分的恶意事件总数

中毒、反射、恶意软件和端口扫描。在所有协议模拟中观察到暴力攻击和端口扫描等攻击，在Telnet和SSH模拟中观察到恶意软件等攻击。

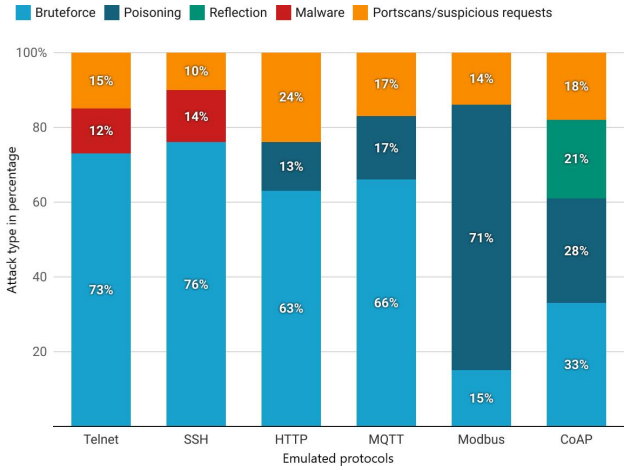


图12：按模拟协议划分的攻击类型百分比

C.4 按交互水平列出的按照方案接收的恶意事件

图13按交互级别总结了每个协议接收的恶意事件 我们观察到，在所有协议仿真的恶意事件的最高数量的高交互实例上接收。在Telnet、SSH、MQTT和Modbus等协议中，我们观察到低交互实例上的事件数量逐渐减少。观察到许多攻击类型，如蛮力攻击、中毒攻击、旋转和反射攻击（CoAP）

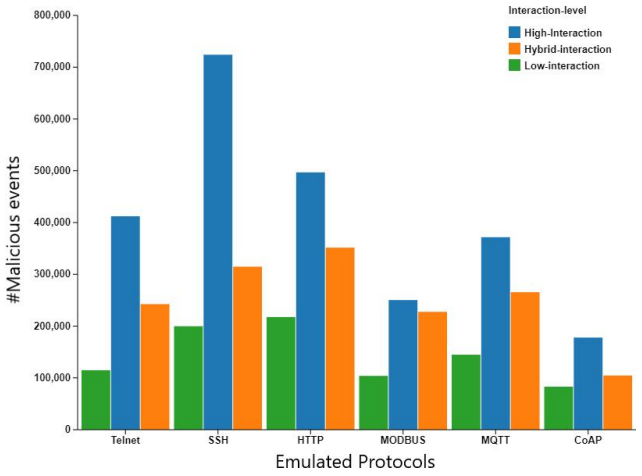


图13：按协议和交互划分的恶意事件总数

C.5 攻击和地理分布

图14显示了每个交互和城市获得的恶意事件的最大和最小数量的聚合。法兰克福市的实例在每个交互水平上记录了最大数量的事件，而实验室基础设施每天记录的事件数量最少。高交互实例接收到更多的恶意事件，无论基础设施或位置如何，其次是混合交互实例上的事件。

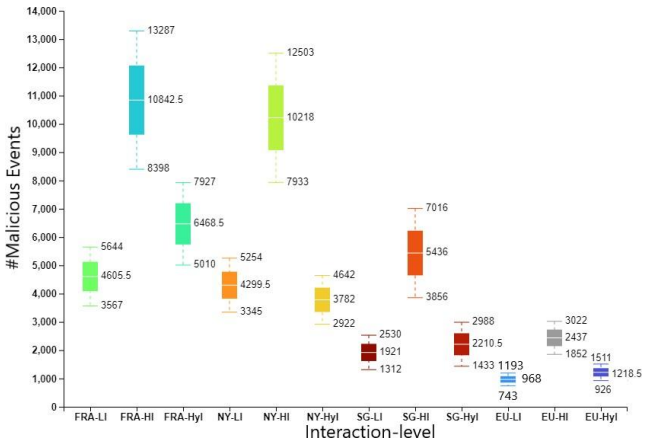


图14：按交互级别和城市划分的恶意事件总数：每天最低、平均和最高

C.6 区域特定攻击类型

表7列出了仅在特定区域观察到的攻击类型和攻击量占恶意流量

C.7 多阶段攻击

在上面讨论的攻击类型中，我们观察到对实例的多级攻击多阶段攻击是指

实例	地区	攻击型	协议	唯一攻击者IP	成交量
R1	丹麦（实验室）	蛮力	Telnet	19	百分之七
R4	纽约	蛮力	SSH	36	百分之十一
R7	法兰克福	蛮力	Telnet	27	百分之十四
R10	新加坡Name	蛮力	Modbus	7	百分之十四
R5	纽约	蛮力	HTTP	33	百分之十七
R7	法兰克福	中毒	MQTT	21	百分之十八
R10	新加坡Name	中毒	MQTT	13	百分之十二
R10	新加坡Name	反思	CoAP	6	百分之十六

表7：特定区域攻击类型汇总

从相同的对手，并顺序地针对目标系统上仿真的多个协议。在所有RIoTPot部署中，共检测到4786次多级攻击。图15显示了攻击者按顺序瞄准的协议。 起始节点表示首先被攻击的协议，节点step-2和step-3表示由相同对手对其他协议进行的顺序攻击。协议下面的数字表示在攻击中使用的协议上接收的请求量。虽然这种行为在扫描服务中是典型的，但是在这种情况下，当在请求中观察到恶意内容时，攻击被分类为多级攻击 大多数请求都是从Telnet和SSH协议发起的。MQTT协议被观察到已经接收到最高量的后续攻击。

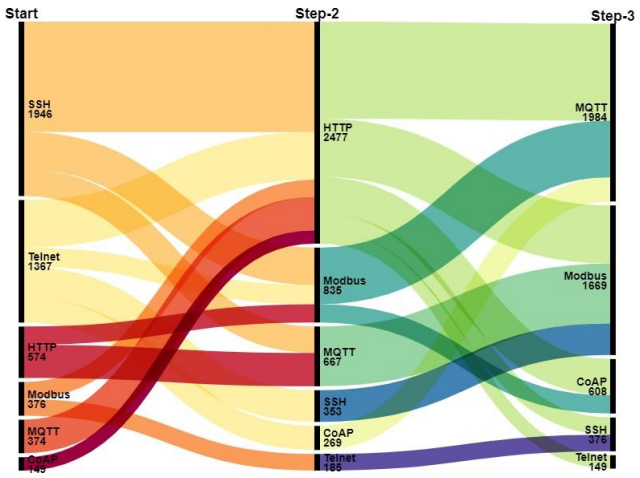


图15：多级攻击

D 标记良性交通

RIoTPot有一个已知的互联网范围扫描服务的数据库然而，它目前仅限于19个服务，因此可能会丢失良性服务。为了进一步对我们数据集中识别的唯一源IP进行分类，我们使用Greynoise API [12]对其进行检查Greynoise提供可疑IP的分类，无论它们是良性的，恶意的还是未知的。图16示出了从Greynoise检索的分类 在将Greynoise的分类与在我们的蜜罐实例上观察到的恶意流量的IP相关联时，我们发现所有的IP都被分类为恶意的或Greynoise未知的。

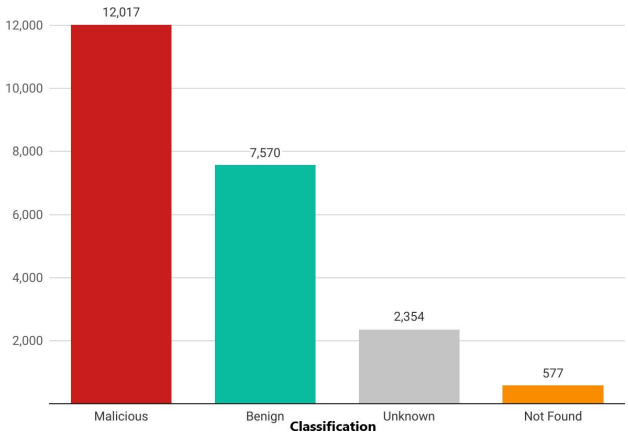


图16：灰噪声分类

E 附录： 补充讨论

E.1 伦理考虑

我们部署了12个实例的RIoTPot在不同的互动水平。 由于蜜罐被配置为显示为易受攻击的系统，因此它们很容易被用于在Internet上引起攻击。我们在所有部署上配置出口规则，以限制离开我们实例的流量，以防止此类滥用。此外，为了避免任何恶意软件攻击传播的感染，我们使用临时容器实例进行蜜罐部署。新的实例会定期产生，以避免任何感染的传播。