物联网蜜罐综述

游建舟[1,2], 吕世超[1,2*], 孙玉砚[1,2], 石志强[1,2], 孙利民[1,2]

[1] 中国科学院信息工程研究所 物联网信息安全技术北京市重点实验室, 北京 中国 100093
[2] 中国科学院大学 网络空间安全学院, 北京 中国 100049

摘要 物联网(The Internet of Things, 简称 IoT)是新一代信息技术的重要组成部分, 已广泛应用于经济社会发展的各个领域, 如工业控制系统、智能家居、智慧城市等。随着物联网应用的爆发式增长, 物联网设备被直接暴露在互联网中, 成为了黑客攻击的重点目标, 并引发了大量安全事件。在多源异构的物联网环境中, 传统的入侵检测、防火墙等安全防护工具存在易漏报和易误报的问题。蜜罐作为一种新兴的主动防御技术, 通过构建可控的诱饵环境, 主动引导黑客攻击, 能够捕获高质量的原始攻击数据, 从而低误报地发现攻击威胁。本文通过调研大量物联网蜜罐文献, 总结了物联网蜜罐的基本概念和技术发展主线, 重点介绍了重定向、识别与反识别和数据分析三种关键技术。此外, 本文提出了一种基于杀伤链模型的物联网蜜罐评估体系, 实现相关蜜罐工作的对比分析, 并讨论和展望了物联网蜜罐未来可能的研究方向。

关键词 物联网; 蜜罐; 工业控制系统; 信息物理系统
中图法分类号 TP393.08 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2020.07.09

A Survey on Honeypots of Internet of Things

YOU Jianzhou[1,2], LV Shichao[1,2], SUN Yuyan[1,2], SHI Zhiqiang[1,2], SUN Limin[1,2]

[1] Beijing Key Laboratory of IoT Information Security Technology, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract Internet of Things (IoT) is an important part of the new generation information technology. It has been widely infiltrated into the national economy and social development in various fields, such as industrial control systems, smart home, and smart city. With the explosive growth of IoT applications, IoT devices are exposed on the Internet directly. It has become an attractive target for hackers and caused lots of security issues. For conventional security tools like intrusion detection systems (IDS) and firewalls, it's prone to be high false alarm rate and hard to deploy in heterogeneous IoT environments. As a new initiative based on the defense network security technology, the honeypot builds a highly controlled environment to capture high-value primary data and discover threats with low false alarm rate. By analyzing relevant IoT honeypot systems and literature, this paper summarized some basic conception of IoT honeypots and the line of development in technology. Based on IoT honeypots, this paper introduced and discussed three technologies: redirection, recognition & counter-recognition and data analysis. Besides, this paper proposed a new IoT honeypot evaluation system based on the cyber kill chain to estimate related work and further discussed the research trend.

Key words internet of things; honeypot; industrial control system; cyber physical system

1 引言

蜜罐是一类没有实际业务用途的网络安全资源, 本质是一种对攻击方进行欺骗的技术, 其价值是吸引攻击者对它进行非法使用[1], 从而帮助网络安全研究人员发现、捕获和分析攻击行为。蜜罐系统由诱饵环境和监控模块两部分构成。诱饵环境是指用于吸引和迷惑攻击者的设备、网络服务或信息, 为攻击活动提供可利用的运行环境; 监控模块负责记录蜜罐中的所有攻击活动, 并限制攻击者的操作范围, 保证蜜罐系统自身的安全性, 防止攻击者利用蜜罐作为跳板对其他网络资源实施攻击。

近年来, 随着物联网逐渐得到广泛应用, 物联网安全问题也变得愈发严重, 广受学术界和产业界

通讯作者: 吕世超, 博士, 高级工程师, Email: lvshichao@iie.ac.cn。
本课题得到国家重点研发计划(No.2018YFB0803402), 国家自然科学基金重点项目(No.U1766215), 国家电网公司总部科技项目(No.522722180007)资助。
收稿日期: 2019-11-09; 修改日期: 2020-01-31; 定稿日期: 2020-07-09

关注。2017 年市场分析公司高德纳(Gartner)在报告中指出, 2020 年物联网设备数量将达到 200 亿[2]。同时, 《2017 年中国互联网网络安全报告》[3]表明, 2017 年国家信息安全漏洞共享平台收录的物联网设备安全漏洞数量较上年增长近 1.2 倍, 每日活跃的受控物联网设备 IP 地址已达 2.7 万个。由于物联网设备和系统广泛应用于智能医疗、安防、家居、交通或制造等与人身安全息息相关的领域, 相比于以窃密和破坏计算机为主的传统互联网攻击, 物联网攻击能够损害人身及资产安全, 甚至影响社会稳定及国家安全。然而, 由于设备厂商缺乏信息安全的研发和资金投入, 物联网设备在抵御网络攻击的能力上先天不足, 并引发了一系列针对物联网设备的网络攻击事件。2016 年 10 月 Mirai 僵尸网络通过控制大量的物联网设备对美国域名解析服务器提供商 Dyn 公司发动 DDoS 攻击, 造成美国东部大面积断网, 推特、亚马逊、华尔街日报等网站无法正常访问[4]。2017 年 4 月信息安全专家 Marc Goodman 研究发现, 植入式医疗设备(如心脏起搏器、胰岛素泵等)伴随着智能化和网络化的发展, 存在可通过网络利用的严重安全漏洞, 攻击者可以远程破坏或扰乱这些设备, 将极大增加网络犯罪的危害[5]。2019 年 3 月 7 日, 委内瑞拉全国发生大规模停电事件, 影响 23 个州中的 18 个州, 这场针对委内瑞拉的“电力战争”被归咎于网络犯罪。上述的安全事件都表明, 人们万物互联的生活正面临着越来越多的攻击威胁[6]。

相比于入侵检测、防火墙等被动防护手段, 蜜罐能够低误报、低干扰地实现部署运行和攻击数据捕获, 并完成攻击预警和潜在威胁发现。因此, 为了更好的研究分析新型和未知物联网攻击, 蜜罐技术已成为物联网安全的重要研究领域[7-8]。自 2004 年以来, 蜜罐技术在蜜网项目组[9]等团队推动发展的同时, 物联网蜜罐也在不断成熟完善, 并逐步融入物联网安全体系中[10]。

根据蜜罐的应用类型, 物联网蜜罐研究可以分为工控蜜罐、消费级物联网蜜罐和信息物理蜜罐三大类。工控蜜罐主要模拟联网的工业控制设备, 其中西门子[11]或施耐德[12]厂商的设备因其极高的市场份额, 成为工控蜜罐实现的最主要参考对象; 消费级物联网蜜罐以网络摄像头、打印机等网络实体设备为参考对象, 分析这些设备所遭受的攻击来源和行为特征; 信息物理系统蜜罐强调对计算资源和物理资源的协同仿真, 分析信息空间的网络攻击对物理空间业务操作之间的相互作用机理, 从而研究攻击者深层次的攻击意图。

在物联网蜜罐系统的发展演进过程中, 网络重定向、蜜罐识别与反识别和数据分析等关键技术的研究取得了阶段性进展。但是, 目前物联网蜜罐系统复杂多样, 该领域缺乏统一的蜜罐评估对比方法。因此, 本文在梳理相关工作后, 基于网络杀伤链模型, 分析了物联网攻击威胁特征, 并提出了一种物联网蜜罐评估体系, 对现有物联网蜜罐系统进行了对比分析。

本文的组织结构如下: 第 1 节简述物联网蜜罐的基本概念, 包括特点、组成、分类和形态; 第 2 节归纳物联网蜜罐技术的起源与发展历史; 第 3 节详细介绍重定向技术、识别与反识别技术和数据分析技术三种关键技术进展; 第 4 节提出一种基于杀伤链模型的蜜罐评估体系; 第 5 节讨论并展望未来的研究方向; 第 6 节对全文进行总结。

2 物联网蜜罐的基本概念

物联网蜜罐是指以物联网计算、网络、感知及执行等资源为诱饵, 用于物联网安全威胁发现、捕获和分析的网络欺骗技术。

2.1 物联网蜜罐的特点

与常规蜜罐相比, 物联网蜜罐的最大特点无疑体现在其诱饵环境的特殊性上, 主要包括:

1) 架构封闭性。物联网领域缺乏明确的架构标准, 导致通信协议、操作系统及应用程序的私有化程度较高, 并往往不提供开放的接口及程序库。因此, 对于大量的私有功能, 物联网蜜罐通常难以实现高真实度的诱饵环境。

2) 诱饵多样性。在物联网蜜罐中, 对外接口及协议标准种类繁多, 不同厂商或型号的操作系统和多源异构的硬件模块, 成为限制物联网蜜罐通用性的重要因素。同时, 多样化诱饵环境也对数据捕获和安全控制的能力提出了更高的要求。

3) 物理融合性。物联网蜜罐参考的业务通常与物理空间感知和操作相关, 如机械操作、图像传输、温度传感等。物联网诱饵环境在构造过程中需兼顾物理空间和信息空间的融合仿真, 以实现深度的交互能力。

2.2 物联网蜜罐的组成

物联网蜜罐组成结构如图 1 所示, 分为物联网诱饵环境和监控模块两部分。

诱饵环境是实现与攻击者交互的系统资源或环境, 是诱惑攻击者的主要战场, 分为以下三个层次:

1) 诱饵接口是攻击者侵入蜜罐系统的必要通信

入口, 包括 SSH、HTTP、Telnet、USB、蓝牙等常见物联网通信接口。

2) 执行环境是物联网正常服务或恶意代码执行的系统依赖环境, 包括应用环境、专用操作系统环境和底层硬件环境等。

3) 物理业务模型是表征物理规律及业务流程的约束条件或模型, 用于约束物联网设备的物理信息感知、执行器的物理操作和状态变化等。物理业务模型是实现高真实度诱饵环境的必要条件, 如以水处理业务为参考对象的蜜罐系统需要遵循箱体水位升降的物理规律和业务约束等。

监控模块是用于采集攻击者威胁数据和维护蜜罐安全的重要组成部分, 以确保攻击活动的可控性。该模块分为数据捕获和安全控制两部分:

1) 数据捕获是针对各类安全威胁的行为数据采集组件, 常见的数据类型有原始数据包、系统行为记录、二进制文件等。

2) 安全控制是保障蜜罐中攻击行为维持在可控范围内的防护组件, 常见的方式有蜜罐外联网络隔离、黑白名单等。

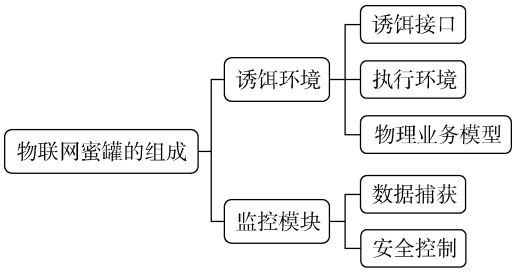

图 1 物联网蜜罐的组成

Figure 1 Composition of IoT Honeypot

2.3 物联网蜜罐的分类

结合物联网通信及硬件形态的特点, 本节分别从交互能力和资源类型两个角度对物联网蜜罐进行分类。

2.3.1 基于交互的分类

物联网蜜罐按照交互能力的强弱分为低交互蜜罐、中交互蜜罐和高交互蜜罐三种类型。

1) 低交互蜜罐是指仅实现物联网诱饵接口模拟的蜜罐。诱饵接口(如各类通信协议栈)是攻击者实施网络探测、目标识别、漏洞利用等攻击的基础。通过解析协议请求数据, 低交互蜜罐能够快速捕获攻击 IP、端口、协议、请求载荷等威胁信息。由于具有协议轻量、运行高效、易维护的优势, 低交互蜜罐的应用部署非常普遍, 是蜜罐应用产品的主要形式, 也通常作为安全威胁感知平台的数据采集工具。

2) 中交互蜜罐是能够实现设备系统执行环境模拟的蜜罐。物联网正常业务和恶意二进制文件的运行都依赖于系统执行环境, 它通常是捕获恶意样本和深度入侵的必备条件。一方面, 中交互蜜罐依托于完整的执行环境实现进程级的对外服务模拟, 另一方面, 执行环境可诱导攻击者完成主机代码注入、执行或潜伏行为, 捕获二进制恶意代码文件。

3) 高交互蜜罐是能够实现物联网信息空间与物理空间执行及操作环境模拟的蜜罐。大部分物联网攻击在中交互蜜罐即可完成所有攻击请求, 并不识别和操作物理空间业务, 而高持续性威胁(APT)[13]等高级别攻击者往往对操作业务有深入理解, 有很强的物理空间操作识别能力。因此, 高交互蜜罐在业务和物理特征模拟上, 通过对物理信号采集、执行、传输等过程建立数字状态模型, 迷惑攻击者对目标系统的业务认知, 实现对高级别攻击者威胁行为的深度捕获和解析。

2.3.2 基于资源类型的分类

根据诱饵资源类型的不同, 物联网蜜罐可分为实物蜜罐、虚拟蜜罐和半实物蜜罐三种。

1) 实物蜜罐是采用原装的软硬件设备作为物联网诱饵的蜜罐。实物蜜罐拥有真实的硬件、操作系统和应用服务, 具备对未知攻击较强的响应能力, 往往对攻击者有极大的引诱性。但实物蜜罐系统往往存在部署成本高、配置管理难度大和应用灵活性差等缺点。

2) 虚拟蜜罐是采用虚拟化等技术对参考目标进行软件实现的蜜罐。虚拟蜜罐通过预先设定的诱捕需求, 用软件模拟相关操作系统和业务, 诱导攻击者入侵隔离的虚拟诱饵环境, 实现高效的威胁捕获及控制。虽然虚拟蜜罐一般对硬件没有依赖, 能够进行灵活部署和配置, 但是虚拟化技术仍有一定局限, 仿真程度难以与实物媲美。

3) 半实物蜜罐是同时采用实物与软件仿真的蜜罐, 兼顾了虚拟蜜罐和实物蜜罐的优势。虚拟蜜罐具备快速构建、灵活部署的特点, 但其与攻击者的交互程度较低; 实物蜜罐的真实度有保障, 但其部署和管理上存在较大难度。半实物蜜罐中通常需要对流量进行调度和分配, 使得虚拟部分和实物部分形成“前端-后台”的架构模式, 充分保证蜜罐系统的灵活性和真实度。半实物蜜罐的核心是虚拟部分与实物部分的协同工作, 主要依赖于重定向技术, 这将在 3.1 节中进行详细阐述。

2.4 物联网蜜罐的形态

2003 年蜜网项目组创始人 Lance Spitzner 在蜜罐

基础上提出蜜网(Honeynet)[14]、蜜场(Honeyfarm)[15]和蜜标(Honeytoken)[16]的概念, 用以描述不同目标环境下的蜜罐应用形态。物联网蜜罐也在不同的安全场景中逐步应用了不同蜜罐形态, 本节通过剖析蜜罐形态进一步梳理了物联网蜜罐的结构特点。为避免混淆蜜罐术语, 在本节中“蜜罐”表示狭义的蜜罐形态, 广义的蜜罐技术用“蜜罐系统”一词表述。

2.4.1 蜜罐

如图 2 所示, 蜜罐是具有单个诱饵节点的蜜罐系统形态, 是蜜罐系统最原始的表现形态。物联网蜜罐往往参考单一设备进行实现, 如摄像头、打印机等。它可以部署在任意的网络位置, 通常用于收集到达特定网络节点的攻击情报, 并缓解相同网段的其他生产设备与资源受到的攻击。由于物联网蜜罐诱饵多样性的特点, 通常需要借助蜜罐形态易开发和扩展的优势, 进一步定制同一物联网设备的不同型号。以典型的 Conpot、Cowrie 物联网蜜罐系统为例, 皆以蜜罐形态为基础, 并分别实现了西门子工控设备和常用嵌入式设备的多种型号扩展。随着互联网中物联网设备安全问题凸显, 多数物联网蜜罐系统以蜜罐形态配合公网 IP 的方式部署在互联网中, 实现物联网设备探测、识别和渗透行为的发现和预警。虽然部署中蜜罐形态可扩展为分布式蜜罐[17]增大攻击面, 但蜜罐节点需独立开发, 对于异构多样的物联网设备, 蜜罐形态通常难以实现高交互。

图 2 蜜罐形态

Figure 2 “Honeypot” Form

2.4.2 蜜网

如图 3 所示, 蜜网是在同一网络中配置多个诱饵节点的蜜罐系统形态。在物联网环境中, 几乎不存在单个设备完成的业务, 例如, 工控设备需连接上位机, 摄像头设备需连接网络视频录像机 NVR(Network Video Recorder)等。物联网蜜网研究往往关注组网和通信接口的问题, 如 Zigbee 蜜网[41]、智能照明系统蜜网[46]。因此, 如何构建高定制化的诱饵网络环境成为物联网蜜网构建的最大挑战。此外, 由于蜜网为攻击者提供了在多个节点设备间横向传播的空间, 对研究攻击传播方式有较大帮助。

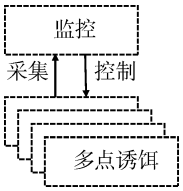

图 3 蜜网形态

Figure 3 “Honeynet” Form

2.4.3 蜜场

如图 4 所示, 蜜场是通过代理的方式扩展诱饵节点部署范围的形态。由于架构封闭的特点, 设备高交互能力的独立开发门槛高, 结合真实设备的蜜场已成为实现高交互物联网蜜罐系统的一种有效手段。在物联网蜜场中, 真实设备和监控模块往往被集中在一个固定的节点或网络中, 便于实现对实物设备的管理维护和数据集中分析。而轻量级的代理部署在任意网络节点中, 将网络攻击重定向至诱饵环境, 通过真实设备实现未知和难以模拟的真实业务, 如摄像头转动、视频图像控制等[49]。

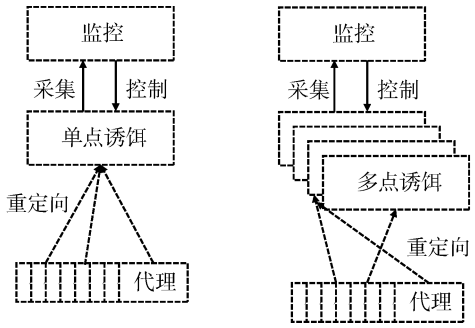

图 4 蜜场形态

Figure 4 “Honeyfarm” Form

2.4.4 蜜标

如图 5 所示, 蜜标是一种特殊的蜜罐诱饵, 它不是任何的主机节点, 而是一种带标记的数字实体。它被定义为不用于常规生产目的的任何存储资源, 例如文本文件、电子邮件消息或数据库记录等。目前, 虽然没有独立的物联网蜜标研究, 但已有相关工作[41]利用蜜标高灵活性、强针对性的特点, 作为物联网蜜罐系统诱捕的补充手段, 辅助捕获细粒度的攻击行为(如文件读取、传递和扩散等)。

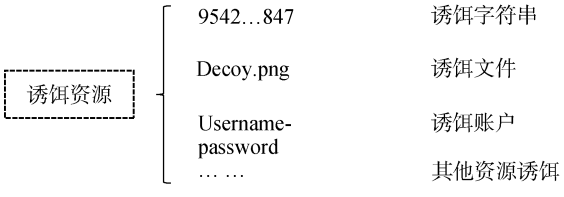

图 5 蜜标形态

Figure 5 “Honeytoken” Form

表 1　物联网蜜罐的发展

Table 1　Development of IoT Honeypot

序号	年份	文献	诱饵接口	系统环境		物理模型	数据捕获	安全控制	形态	交互程度	资源类型
				操作环境	底层硬件						
1	2004	Pothamsetty et al.[20]	Modbus; HTTP; FTP; Telnet	无	无	无	日志	无	蜜罐	低	虚拟
2	2006	Digtial Bond Inc.[22]	Modbus; HTTP; FTP; SNMP; Telnet; VxWorks Debugger	无	无	无	日志	蜜墙隔离	蜜罐	低	虚拟
3	2013	Conpot[28]	S7comm; Modbus; IEC104; Bacnet; ENIP; HTTP; FTP; SNMP	无	无	无	日志	无	蜜罐	低	虚拟
4	2013	Snap7[31]	S7Comm	无	无	无	日志	无	蜜罐	低	虚拟
5	2014	Cryplh[29]	S7comm; HTTP; SNMP	无	无	无	日志	未知	蜜罐	低	虚拟
6	2015	MiniCPS[59]	SSH; Modbus; ENIP	MiniNet	未知	水处理	日志	虚拟机隔离与控制	蜜网	高	虚拟
7	2015	IoTPOT[53]	Telnet	OpenWRT	MIPS、X86 等 8 种	无	日志; 流量; 二进制代码	虚拟机隔离控制	蜜罐	中	虚拟
8	2015	Kara et al.[42]	TR-069; HTTP	无	无	无	日志	未知	蜜网	低	虚拟
9	2016	Xpot[32]	S7Comm	未知	未知	未知	日志	未知	蜜罐	中	虚拟
10	2016	Chamotra et al.[43]	Telnet; SSH; SIP; HTTP	有限状态机	无	无	日志; 流量; 二进制文件	未知	蜜罐	中	虚拟
11	2016	Litchfield et al.[57]	REST	无	无	水处理	日志	未知	蜜网	高	虚拟
12	2017	Zhao et al.[33]	S7Comm; HTTP	无	无	无	日志	未知	蜜罐	低	虚拟
13	2017	S7commTrace[34]	S7Comm; HTTP	无	无	无	日志	未知	蜜罐	低	虚拟
14	2017	Iotcandyjar[48]	SSH; Telnet; HTTP; TR069; XMPP; MQTT; UPnP; CoAP; MS-RDP	无	无	无	流量	入侵检测与阻断	蜜罐	低	虚拟
15	2017	SIPHON[49]	Telnet; SSH	实物	实物	实物	流量	未知	蜜场	高	半实物
16	2017	Dowling et al.[41]	SSH; ZigBee; HTTP	Kippo 虚拟环境	未知	无	日志; 流量; 二进制文件	沙箱运行与调试	蜜网; 蜜标	中	虚拟
17	2017	Iskhakova et al.[44]	SSH; Telnet; HTTP	实物	实物	实物	流量	未知	蜜网	高	实物
18	2017	Krishnaprasad et al.[45]	Telnet; SSH; HTTP; TR-069	OpenWRT; Honeything	未知	无	日志; 流量	未知	蜜罐	低	虚拟
19	2018	Wang et al.[46]	XMPP/MQTT; REST	无	无	Philips Hue 智能照明系统	日志	未知	蜜网	低	虚拟
20	2018	U-PoT[47]	UPnP	无	无	无	日志	未知	蜜网	低	虚拟
21	2018	HIoTPOT[50]	Telnet	Raspberry Pi	Raspberry Pi	无	日志	未知	蜜场	高	实物
22	2018	HoneyBot[60]	未知	未知	未知	机器人	未知	虚假执行业务操作	蜜罐	高	半实物

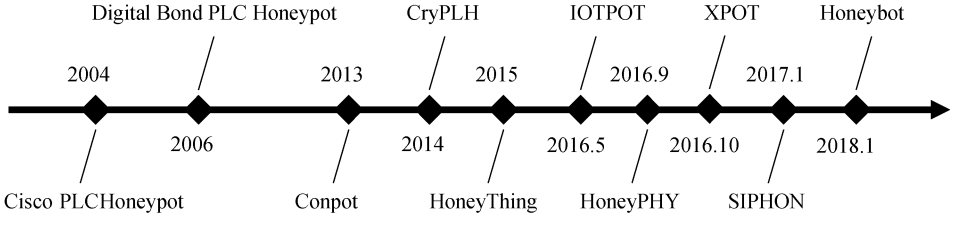

图 6　物联网蜜罐发展的时间轴

Figure 6　Timeline of the Development of IoT Honeypot

3 物联网蜜罐的发展过程

蜜罐的概念起源于 1990 年, 在电脑安全专家 Cliff Stoll 出版小说《The Cuckoo's Egg》[18]中首次次提及了“Honeypot(蜜罐)”一词, 小说描述了主人公如何通过部署一系列虚假数字文件追踪网络黑客的故事。随着物联网技术的不断成熟和应用普及, 从网络摄像头、打印机到楼宇控制器、电力设施等工业设备, 物联网实现了各类设备的互联互通。物联网蜜罐在常规蜜罐发展基础上也经历了十多年的发展。表 1 整理了本文涉及的物联网蜜罐工作, 相关内容见后续详述。如图 6 所示, 物联网蜜罐按时间顺序依次从工控蜜罐、消费级物联网蜜罐以及信息物理系统蜜罐三条主线开展研究。

3.1 工控蜜罐

工控系统往往涉及国家关键基础设施的安全, 容易成为国家间攻击的首选目标。工控系统的攻击者不仅熟练掌握了网络攻击手段, 还对工控系统业务流程有深入的研究, 发起的攻击具有很强的隐蔽性。依靠传统的防火墙、入侵检测等防护手段难以有效检测此类攻击。而依托于低扰低误报的特点, 蜜罐可以作为工控安全威胁发现有效工具。工控蜜罐研究通常从通信协议解析出发, 通过仿真工控协议服务并绑定工控设备默认端口, 等待攻击者的连接。

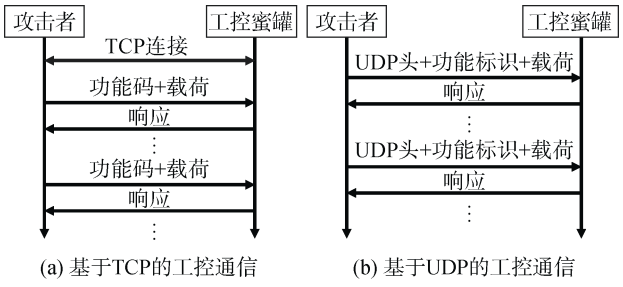

图 7 工控蜜罐的通信方式

Figure 7 Communication Mode of ICS Honeypot

工控协议是工控蜜罐研究的重要基础。工控协议通常规定设备在标准化 TCP/IP 协议的应用层通信中的报文标准和交互模式。除了基本的协议类型、长度、版本、校验和等字段, 工控协议最重要的是其中的应用协议数据单元。它通常以功能码(function code)的方式实现设备状态获取、设备控制和程序文件装载等操作。常见的工控协议有 S7comm、Modbus、DNP3、ICCP、IEC 60870-5-104、Ethernet/IP 和 FINS 等。其中, Modbus 和 S7comm 是工控蜜罐研究领域最流行的两种工控协议, 接下来围绕 Modbus 和 S7comm 协议对工控蜜罐的发展进行简要分析。

Modbus 协议[19]是 Modicon 公司(施耐德电气旗下的一个品牌)于 1979 年为可编程逻辑控制器(Programmable Logic Controller, PLC)发布的公开通信协议。由于 Modbus 是受业界普遍认可和采纳的工控协议, 以 Modbus 协议仿真为主的工控蜜罐(简称“Modbus 蜜罐”)已经成为研究典型。虽然 Mobdus 是公开的协议, 但功能码分为公共和私有两种。私有部分的功能探索也是工控蜜罐研究的首要任务。2004 年 Cisco 公司关键基础设施保障组(Critical Infrastructure Assurance Group , CIAG)的 Matthew Franz 等实现了第一个具有 Modbus 服务的工控蜜罐系统[20], 以传统互联网蜜罐工具 Honeyd[21]为基本框架, 添加了工控协议 Modbus-TCP 的仿真模块, 针对常用的写入和读取操作, 实现了对读取线圈(0x01)、诊断(0x08)、写多个寄存器(0x16)等常见的 Modbus 功能码响应。2006 年 Digital Bond 公司[22]结合蜜墙(Honeywall)工具[23](一种用于流量控制的网关软件)搭建了双主机的 Modbus 蜜罐, 将 Modbus 服务仿真与数据捕获系统分离, 提升了蜜罐的数据捕获与分析能力。协议栈通信是 Modbus 蜜罐研究的重要依赖, 经开源社区的努力, 逐渐形成了 Modbus-tk[24]、Pymodbus[25]、libmodbus[26]等 Modbus 通信库, 并被大多数 Modbus 蜜罐所采用。如表 2 所示, 本文通过对库源码分析, 对比了三种 Modbus 通信库的交互能

表 2 Modbus 协议功能码对比

Table 2 Comparison of Modbus Function Code

功能码	功能标识	开发库支持
01(0x01)	读线圈	[24-26]
02(0x02)	读离散量输入	[24-26]
03(0x03)	读保持寄存器	[24-26]
04(0x04)	读输入寄存器	[24-26]
05(0x05)	写单个线圈	[24-26]
06(0x06)	写单个寄存器	[24-26]
07(0x07)	读异常状态*	[24-26]
08(0x08)	诊断*	[24-25]
11(0x0B)	获得通信时间计数器*	[25]
12(0x0C)	获得通信时间记录*	[25]
15(0x0F)	写多个线圈	[24-26]
16(0x10)	写多个寄存器	[24-26]
17(0x11)	报告从站 ID*	[24-26]
20(0x14)	读文件记录	[25]
21(0x15)	写文件记录	[25]
22(0x16)	屏蔽写寄存器	[25-26]
23(0x17)	读/写多个寄存器	[24-26]
24(0x18)	读 FIFO 队列	[25]
43(0x2B)	读设备标识	[24-25]

*表示该功能码仅适用于串行链路。

力。结果表明, Pymodbus 的实现已覆盖所有公共功能码, Modbus-tk 和 libmodbus 库略有欠缺。

S7comm 协议[27]也称为 STEP7 协议, 是西门子公司基于 ISO TCP(RFC1006)标准实现的一种非公开控制设备通信协议。由于西门子公司工控市场占有率高, 以 S7comm 协议仿真为主的蜜罐(简称“S7 蜜罐”)也获得了广泛关注。2013 年蜜网项目组旗下发布的开源工控蜜罐框架 Conpot[28] 集成了 S7comm 协议仿真模块, 采用模板文件的形式管理工控端口与服务, 并结合 SNMP、HTTP 等服务可将蜜罐封装为 S7-300 、S7-400 等系列西门子 PLC 蜜罐实例。虽然 Conpot 为开发者提供了工控蜜罐便捷部署、管理、修改和配置的开发框架, 但它的仿真能力仍有所不足。因此, 2014 年布达佩斯技术经济学院 DI Buza 等提出了蜜罐 CryPLH[29] , 针对西门子 S7-300 PLC 增强了控制器各种功能服务的真实度, 在 HTTP 方面实现了设备 Web 页面克隆, SNMP 方面新增了设备历史操作跟踪和操作关联的模拟, S7comm 方面新增了与西门子官方的工控编程及设备管理软件 Simatic Step7[30] 的会话长连接功能。为了完善 S7comm 的通信流程, Davide Nardella 针对该软件开发了一款开源的通信库 SNAP7[31], 支持多平台部署、多编程语言开发, 能够灵活仿真西门子工控设备客户端和服务端。为了评估不同 S7 蜜罐的交互能力, 2016 年柏林自由大学 Stephan Lau 等提出了 S7 蜜罐交互标准, 并基于此标准实现了高交互 S7 蜜罐 XPOT[32]。XPOT 参考了西门子 S7-314C-2 PN/DP 型号 PLC, 通过 MC7 字节码编程实现了 PLC 程序执行及调试功能, 并采用底层虚拟机(Low Level Virtual Machine, LLVM)优化了 MC7 程序编译时间, 以逼近真实 PLC 程序编译用时。此外, 大量文献针对 S7 蜜罐的工控服务进行了强化, 但并未提供有效实验验证或数据证明。2017 年北京邮电大学 Chunhui Zhao 等[33]针对 S7-1200 设备实现了人机接口(Human Machine Interface, HMI), 并扩展了部分功能码。同年, 中国科学技术大学 Feng Xiao 等设计的 S7commTrace 蜜罐[34] 也进一步扩展了协议功能, 已支持全部 12 种一级功能码和大部分二级功能码。

S7comm 协议没有公开协议标准, 本文参考 Wireshark 2.6 版本源码[35]中 S7comm 协议解析方法对比分析了相关工作。如表 3、表 4 所示, S7comm 协议包含 12 个一级功能码和 7 组二级功能码, 其中二级功能码皆为“0x00 CPU services”一级功能码的扩展。对于已知的 S7comm 通信交互, 本文构造了 12 种一级功能码和 14 种二级功能码(已涵盖主要请求功能码)的测试数据集, 对相关蜜罐进行了测试。对于开源蜜罐软件, 本文选取了最新的发布版本。相关文献并未过多阐述功能码实现, 本文根据功能描述进行了人工匹配和标注。结果表明, Conpot 0.5.2 版本[36]仅实现了“0x4 Read SZL”1 种二级功能码, 而 Snap7 1.4.0 版本[37]支持 6 种一级功能码和 10 种二级功能码。

表 3 S7comm 协议一级功能码

Table 3 Function Codes of S7comm Protocol

功能码	功能标识	已支持蜜罐
0x00	CPU services*	[29, 32-34, 36-37]
0xf0	Setup communication*	[29, 32-34, 37]
0x04	Read Var*	[29, 32-34, 37]
0x05	Write Var*	[29, 32-34]
0x1a	Request download*	[29, 32, 37, 34]
0x1b	Download block*	[29, 32, 34]
0x1c	Download ended*	[29, 32, 34]
0x1d	Start upload*	[29, 32, 34]
0x1e	Upload*	[29, 32-34]
0x1f	End upload*	[29, 32-34]
0x28	PI-Service*	[29, 32-34, 37]
0x29	PLC Stop*	[29, 32-34, 37]

*表示具备该功能的测试用例。

表 4 S7comm 协议二级功能码

Table 4 Subfunction Codes of S7comm Protocol

组别	次级功能标识	已支持蜜罐
0x1	0x01	[34]
	0x02*	[34, 37]
	0x0c, 0x0e	无
	0x0f*	[37]
	0x10*	[37]
	0x13*	[37]
0x2	0x01*	[34]
	0x04*	无
	0x05	无
0x3	0x01*	[29, 32-34, 37]
	0x02*	[29, 32-34, 37]
	0x03*	[29, 32, 34, 37]
0x4	0x01	[29, 32-34, 37]
	0x02	[34, 37]
	0x03	[33]
	0x05-0x09	无
	0x0b-0x0e	
	0x11-0x13	
	0x16	
0x5	0x01*	[34]
0x6	None	[34]
0x7	0x01*	[32-34, 37]
	0x02*	[33-34]
	0x03	[34]
	0x04	[34]

*表示具备该功能的测试用例。

除了工控协议外, 工业控制设备中也有不少常规协议通信。对于工业控制设备, 工控协议服务主要负责工控业务操作, 常规协议服务主要完成设备网络管理、可视化和文件传输等通用功能。如表 5 表所示, 上述工控蜜罐通常采用 SNMP、HTTP 等常规协议服务辅助设备仿真。2015 年, Alexandru 等[38] 选取了 DNP3、ICCP、IEC104、Modbus 四种工控协议和 SNMP、TFTP、XMPP 三种常规协议, 分析了不同工控和常规协议组合对攻击者的相对吸引力。结果表明, SNMP 协议会增加攻击者对 DNP3 服务的交互次数, TFTP、XMPP 对四种工业协议没有明显影响。

表 5 工控蜜罐组成

Table 5 Composition of ICS Honeypot

蜜罐研究	诱饵接口	
	工控服务	非工控服务
Pothamsetty et al.[20]	Modbus	HTTP; FTP; Telnet
Digtial Bond Inc.[22]	Modbus	HTTP; FTP; SNMP; Telnet; VxWorks Debugger
Conpot[28]	S7comm; Modbus; IEC104; Bacnet; ENIP	HTTP; FTP; SNMP
Cryplh[29]	S7comm	HTTP; SNMP
Snap7[31]	S7comm	无
Xpot[32]	S7comm	无
Zhao et al.[33]	S7comm	SNMP; HTTP
S7commTrace[34]	S7comm	无

综上所述, 工控蜜罐主要体现私有封闭的特点, 不同厂商不同型号的设备仿真往往存在较大差异。这些封闭性和差异化也体现在执行环境、业务流程中。在这种情况下, 最理想的工控诱饵环境是使用实物工控设备。但设备成本高、型号多样, 导致工控实物蜜罐难以部署和扩展。因此, 现阶段的工控蜜罐主要以虚拟蜜罐为主。工控系统的长期封闭也导致工控系统软件架构及虚拟化研究相对薄弱, 已有高交互工控蜜罐研究并未详述其实现方法[32]或仅仅是有限条件的交互能力[29, 33-34], 主要通过诱饵接口的协议栈去模拟工控设备的正确交互, 与真正意义的高交互工控蜜罐还有较大差距。

3.2 消费级物联网蜜罐

随着物联网技术的快速发展, 网络摄像头、智能打印机等消费级物联网设备得到了大规模应用。由于缺乏有效的防御机制, 消费级物联网设备很快成为网络攻击的重灾区。物联网设备对外通信方式各异, 常见的有以太网、USB、蓝牙、无线、串口通信等。针对多样化的入侵接口, 以各类入侵介质为诱饵的蜜罐研究相继出现。2012 年德国波恩大学 S. Poeplau and J. Gassen 实现了基于 USB 的蜜罐[39], 并验证了其在 USB 病毒检测和捕获方面的能力。同年, 美国达科他州立大学 Ashley Podhradsky 等实现了一款面向蓝牙攻击的蜜罐工具 Bluepot[40], 提供图形化的界面监控蓝牙网络入侵行为, 支持 Blue Bugging、Blue Snarfing 等多种入侵行为检测。2017 年, 爱尔兰高威梅努斯理工学院 Seamus Dowling 等创建了一个模拟 ZigBee 网关的蜜罐[41], 通过伪造一些敏感医疗流量信息作为蜜标, 发现网络中专门针对 ZigBee 的攻击。2015 年, 土耳其加齐大学 Ö. Erdem 等研制了家庭网关蜜罐 HoneyThing[42], 除了实现基于 TR069 协议的远程管理功能外, 在 Http 服务中还植入了三个漏洞 (CVE-2014-9222 、CVE-2014-4019 、CVE-2013-6786), 以增强蜜罐可利用性。与工控领域不同, 消费级物联网蜜罐的各类通信协议已有不少成熟的开源或官方协议应用库(如 OpenSSH、Httplib 等), 且其应用执行环境架构明确, 涌现了大量基于各类通信接口的物联网蜜罐研究, 如 SIP[43]、SSH[44-45]、XMPP[46]、UPnP[47]。为了快速扩展多样化的诱饵接口, 2016 年美国黑客大会上 Tongbo Luo 等设计了一个通用的物联网蜜罐构造框架 IoTCandyJar[48], 通过流量重放的方式将蜜罐接收的探测流量转发给互联网中的其他物联网设备节点, 并通过“马尔科夫决策模型+Qlearning 算法”筛选响应数据, 从而完成各物联网服务的智能交互。

在诱饵接口的基础上, 消费级物联网蜜罐注重对执行环境的实现, 用于捕获恶意二进制文件。针对二进制文件传递、安装和执行的需求, 执行环境可划分为诱饵进程、操作系统和物理硬件三个层次。外联的诱饵进程通常是二进制文件网络传递的入侵接口(如 Telnet、SSH 进程等)。操作系统和物理硬件则是二进制文件安装和执行的系统及硬件依赖。

消费级物联网蜜罐的执行环境分为实物环境和虚拟环境两种。如图 8(a)所示, 实物环境是由真实物理设备上运行的原装操作系统和诱饵进程构成。它提供了真实的运行环境, 可以满足二进制文件的所有系统调用需求, 攻击者通常难以识别实物执行环境。但是, 实物环境存在以下两种限制条件。一方面, 硬件设备的高度依赖使得其难以远程部署, 异构设备多样的通信接口也增大了统一管理的成本; 另一方面, 消费级物联网设备出于成本考虑, 往往限制了系统执行环境计算、存储和通信的空间, 难以部署蜜罐所需的常规监控模块, 导致无法有效提取入侵

到实物环境中的二进制文件。针对前者, 2017 年新加坡科技与设计大学 Juan Guarnizo 等设计了物联网蜜场系统 SIPHON[49], 通过集中部署与云端代理结合的方式解决了实物环境的部署与运维的问题。而对于后者, 相关研究中并没有好的解决方案, SIPHON 和 HIotPOT[50]虽然都部署了实物环境, 但依旧无法捕获二进制文件, 仅能够从流量角度进行分析。

如图 8(b)所示, 虚拟环境通过虚拟化技术, 在虚拟化层(虚拟 CPU、内存等硬件)运行虚拟机, 装载与参考设备相同的操作系统, 为二进制文件的执行提供依赖环境。为了减少系统消耗和降低部署难度, 多数消费级物联网蜜罐采用开源工具 Kippo 或 Cowrie 实现虚拟交互环境。Kippo[51]是 2011 年开发的一款基于 Python 并支持多系统部署的 SSH 蜜罐工具, 模拟了 SSH 登陆后的 Shell 交互环境, 实现系统文件目录操作、命令行响应、敏感文件伪装等功能, 并支持捕获 SSH 口令爆破、自动化脚本攻击、僵尸网络病毒等恶意攻击。为解决 Kippo 多接口扩展难的系统性问题, 开源社区 2015 年开发了一个新的 Kippo 分支 Cowrie[52], 扩展了 SCP、SFTP 和 Telnet 协议, 并支持 SSH 和 Telnet 的登陆交互和命令运行, 支持恶意文件上传和恶意行为隔离, 以保障蜜罐系统的运行安全。但 Kippo 和 Cowrie 本质是通过文本交互实现的伪操作系统, 仅支持二进制文件的下载, 无法进行真实的进程调用。

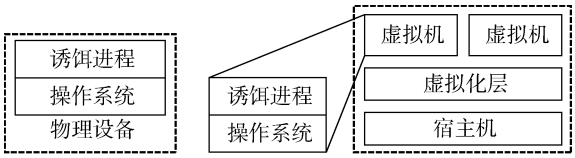

图 8 执行环境的两种形式

Figure 8 Two Forms of Execution Environment

物联网设备架构众多, 恶意软件针对不同的架构通常会编译出不同的二进制文件, 以保障在不同执行环境中的正常运行。为深入研究不同物联网恶意软件家族的威胁现状, 2016 年横滨国立大学 Yin Minn PA PA 等开发了蜜罐 IOTPOT[53], 支持 MIPS、MIPSEL、PPC 等 8 种物联网常用硬件架构执行环境, 装载嵌入式 Linux 操作系统, 并开放 Telnet 诱饵服务, 成功捕获了多种架构依赖的二进制文件。如图 9 所示, 在虚拟环境中的 IOTPOT 包含前端响应器、分析器、后端物联网沙箱、管理器和下载器五个模块。前端响应器负责响应探测阶段的攻击请求, 分析器支持协议选项、登陆提示符、账号/密码认证等 Telnet 常规通信内容配置。当传入未知请求命令时, 前端响应器则将命令转发后端物联网沙箱 IOTBOX。IOTBOX 是以 QEMU 虚拟机[54]搭载 OpenWrt 操作系统[55]方式实现的物联网沙箱, 支持 8 种 CPU 架构执行环境, 可以选择 8 种 CPU 架构的执行环境处理操作命令, 分析不同恶意软件家族的进程行为。下载器则通过提取攻击者的下载命令(诸如 wget、ftp 和 tftp 等)中的 URL, 隔离下载攻击过程所需的二进制文件。管理器用于蜜罐环境配置, 如设备环境、重定向配置等。值得关注的是, 前端响应器具有自动响应已知请求的功能, 它能够解析攻击者与 IOTBOX 之间的交互, 并更新响应配置文件, 以便提升前端的攻击响应能力, 减少对 IOTBOX 的依赖。

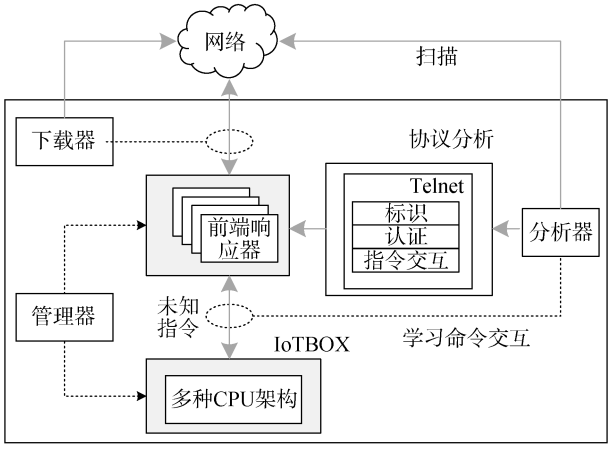

图 9 文献[53]中的物联网蜜罐架构

Figure 9 Architecture of IOTPOT Proposed in [53]

综上所述, 不论是诱饵接口还是执行环境, 消费级物联网蜜罐主要体现了诱饵多样性的特点。2018 年比利时鲁汶天主教大学 Lionel 等[7]使用 SURFnet 公司/15 网段的网络嗅探数据包和 15 个 IPV4 空间物联网蜜罐, 分析了 SSH、Telnet、HTTP 通用协议及 MQTT、CoAP、UPnP 等物联网专用协议安全威胁。结果表明, 虽然当前针对物联网设备的攻击仍以 Telnet 为主要入口, 但物联网的各种其他协议(如 MQTT、UPnP 和 CoAP 等)也捕获到了不同的连接尝试, 攻击者可利用这些协议进行入侵。针对不同架构的执行环境, 消费级物联网蜜罐同样难以采用实物环境实现二进制文件的捕获, 而虚拟化实现的执行环境对蜜罐研究者提出了更高的技术要求。此外, 虚拟化技术的对象依然是通用的计算设备, 针对物联网设备的虚拟化研究仍难以摆脱多样异构和私有封闭的限制。

3.3 信息物理系统蜜罐

物联网的核心是物与物、物与环境、物与人之

间的信息交互[56], 实现物理空间和信息空间的融合。信息物理系统(CPS)通过计算、通信和控制技术的深度协作, 实现大型工程系统的实时感知、动态控制和信息服务。在物联网中, 物理空间包含了各种物理世界的输入输出活动。输入部分是对物理域进行信息采集和获取, 典型设备包括 RFID 装置、各类传感器(如红外、温度、湿度、速度等)、图像捕捉装置(摄像头)、全球定位系统 (GPS)、激光扫描仪等, 其作用是监测网络的不同数据; 输出部分的任务是对物理域进行操作, 通过调节对象事务的状态及其变换方式, 使对象处于预期的运动状态。

信息物理系统蜜罐的研究是对物联网蜜罐本质的重新思考, 并经历了理论构建、概念验证和部署研究等过程。2016 年佐治亚理工大学 Litchfield S 等提出了信息物理蜜罐框架 HoneyPHY[57], 在工控蜜罐的基础上抽象出了信息空间和物理空间的具体含义, 并进行了概念验证。如图 10 所示, 该理论框架将蜜罐分为网络接口、设备模型、流程模型和内部通信四个模块。网络接口模块支持各类对外的诱饵服务, 保持与攻击者的连接并转发给设备仿真模块处理; 设备模型模块负责与攻击者进行完整的通信交互, 通过实时查询流程模型模块当前物理空间数据信息, 并将其封装在特定协议的通信报文中返回给攻击者; 流程模型模块采用数字模型模拟物理空间业务操作过程, 如电网中的电流传输模型、加热过程中的温度变化模型等; 内部通信模块负责将其他三个模块连通, 并与攻击者的通信环境隔离。

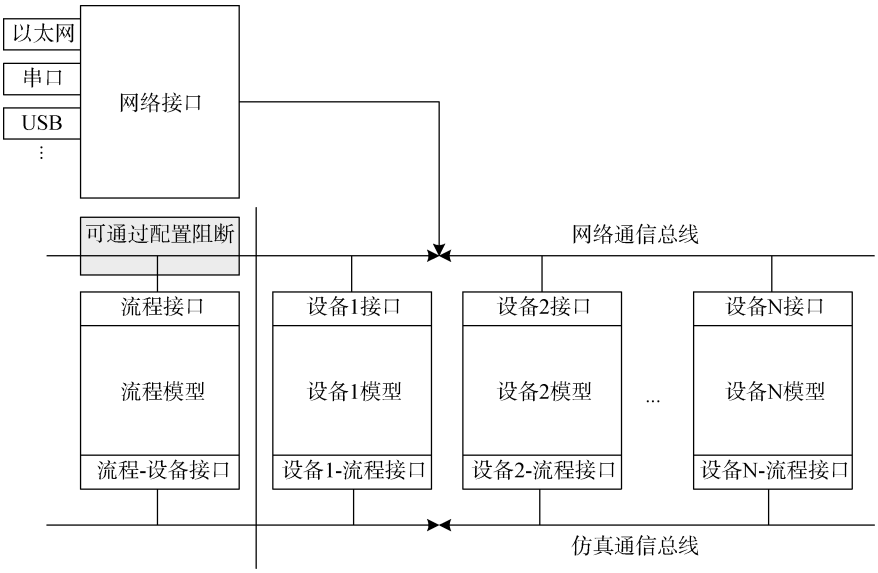

图 10 文献[57]中的信息物理蜜罐架构

Figure 10 Architecture of CPS Honeypot Proposed in [57]

除了描述信息物理蜜罐概念及构成, HoneyPHY 还实现了不同的业务场景的概念验证。如图 11 所示, 在暖通空调系统的加热模型中, 加热时间与温度变化必须满足客观的物理规律, 而多数蜜罐未重视物理空间行为的模拟, 往往采用随机值进行填充。

针对不同物理空间的应用场景, 2014 年华东理工大学周昆等采用经典的田纳西-伊斯曼过程[58]化工反应模型, 结合 Honeyd 框架和 Matlab/Simulink 的仿真软件, 构建了一种过程控制系统蜜罐。Honeyd 实现蜜罐基础服务和监控功能, 通过 Python 语言调用 Matlab/Simulink 仿真的实时生产数据, 实现生产过程的模拟与调控。2016 年新加坡科技设计大学 Daniele 等研制了面向信息物理系统的仿真工具箱 MiniCPS[59], 以水处理测试台为参考, 提供虚拟网络

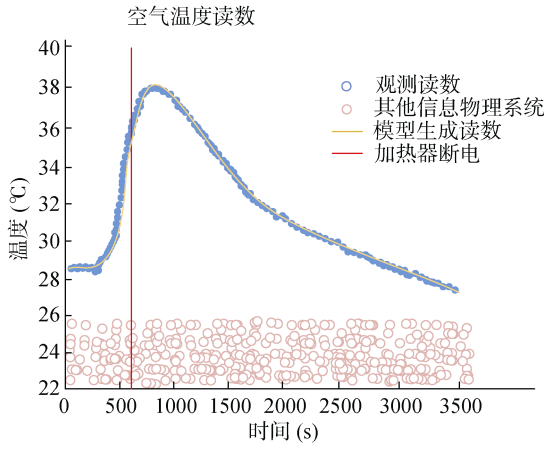

图 11 文献[57]中加热结果的对比分析

Figure 11 Contrastive Analysis of Heating Results Proposed in [57]

环境构建、典型 CPS 控制组件模拟和物理层仿真等功能。MiniCPS 以 SSH 和 VPN 服务为攻击入口, 实现从互联网侦察、利用, 到入侵控制等攻击路径的全程监测。2018 年佐治亚理工学院 Celine 等基于 HoneyPHY 设计了一个机器人蜜罐 Honeybot[60], 能够执行位移、捡起、放下等指令, 并针对攻击者的恶意操作响应虚假状态信息。

综上, 信息物理系统蜜罐的研究将物联网蜜罐的内涵提升到了一个新的高度。但当前物理信息系统蜜罐的研究仍处于概念研究阶段, 并未出现完整通用的信息物理蜜罐的方案。

4 物联网蜜罐的关键技术

上一节主要介绍了三种类型物联网蜜罐在攻击诱导上的方法和技术。除此之外, 物联网蜜罐的关键技术还包括部署、对抗和分析三个方面, 主要用于物联网应用中的大规模蜜罐部署、反蜜罐识别和安全威胁分析所面临的问题

4.1 重定向技术

物联网蜜罐的诱饵多样性表明, 物联网业务的实现往往涉及多种不同的诱饵类型, 需要根据攻击特征对攻击流进行切换。因此, 物联网蜜罐多点部署和流量调度的需求尤为突出。为了兼顾部署数量和部署成本的限制, 采用重定向技术调度多点攻击流量已成为物联网蜜罐设计和部署的常态。重定向技术通常运用在蜜场中的跨网段调度和混合蜜罐中的不同交互能力蜜罐转换中。通过连接不同地理空间或不同类别的诱饵环境, 进行统一的攻击诱导、流量分配和数据采集。

如图 12 所示, 重定向技术大致分为入侵检测、转发决策、流量调度和响应回复四个阶段。在入侵检测阶段, 蜜场需要对网络流量进行解析和分类, 为转发决策提供依据; 在转发决策阶段, 依据专家知识构建决策模型将不同流量进行区分, 常用的判断方式有正常与异常判断、低交互与高交互判断等; 在流量调度阶段, 通过 TCP 重放或代理的方式将攻击流导向至不同的目标。在请求响应阶段, 整合不同诱饵的响应进行筛选分析, 选择最佳响应返回给攻击者。

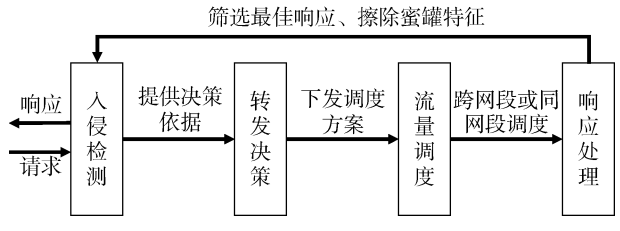

图 12 重定向技术的工作流程

Figure 12 Workflow of Redirection Technique

2009 年, 马里兰大学 Berthier 等提出了一种蜜罐工具 Honeybrid[61], 支持网络威胁的量化和网络流向的控制, 用于部署和管理具备不同交互能力的蜜罐。Honeybrid 由决策引擎和重定向引擎构成。决策引擎用于配置每个蜜罐节点转发策略信息, 分全拒绝、全接收、随机选择、载荷匹配、攻击源匹配五种转发模式。重定向引擎则读取决策信息, 通过 TCP 握手重放和序列号同步实现同网段的流量转发。为了增强决策依据和跨网段调度能力, 北京大学陆腾飞等提出了“策略路由+OpenVPN”的网络流重定向机制[62], 通过网络攻击检测匹配攻击流量, 并实现跨网段的流量调度。为了降低对入侵检测和转发决策阶段的技术依赖, 2016 年亚利桑那州立大学 Wonkyu Han 等提出了基于软件定义网络(SDN)的混合蜜罐架构 HoneyMix[63], 通过对组播后蜜罐响应的筛选, 实现对响应流量的细粒度控制并擦除可能暴露的蜜罐特征。2017 年, 马德里理工大学 Wenjun Fan 等提出了一种新的基于 SDN 的混合蜜罐框架[64], 采用了 Snort 入侵检测工具, 建立了以首个载荷数据包为依据的决策方法, 实现低高交互的流量的判别。但其并未制定详细的调度标准。同年, 亚利桑那州立大学 Sukwha 等提出了混合蜜罐架构 HoneyProxy[65], 以连接持续时间、探测攻击次数等信息为基础判别历史最佳的响应, 并以此为基础实现转发决策。为提高重定向和监测的效率, 2018 年弗吉尼亚联邦大学 Bahman 等提出了混合蜜罐架构 HoneyV[66], 建立了以信任级别为基础的评估机制, 实现对不同网络流量的差异化监测。

综上所述, 表 6 展示了相关文献在各阶段的贡献。可以看出, 重定向技术的重点在于转发决策和流量调度。决策模型的主要依赖于专家知识, 而缺乏对交互反馈的关注, 导致重定向决策的灵活性不足。此外, 相关文献对跨网段流量调度中的转发决策也缺乏深入研究。

4.2 蜜罐识别与反识别技术

虽然蜜罐诱饵环境日益完善, 但是攻击者对蜜罐识别的能力也在不断加强。攻防对抗是网络安全领域的常态, 物联网蜜罐作为一种主动防御技术, 其反蜜罐识别能力直接影响其自身的价值。一旦攻击者识别蜜罐, 蜜罐将失去意义, 甚至沦为攻击者的入侵跳板。蜜罐识别与反识别是攻防博弈的两面。蜜罐识别也称为反蜜罐(Anti-Honeypot), 其本质就是

分析蜜罐和真实设备之的差异, 即通过检测蜜罐各类指纹特征对网络环境中蜜罐进行区分。本节将以蜜罐识别作为切入点对相关工作进行梳理。

表 6 重定向技术

Table 6 Redirection Technique

相关文献	入侵检测	转发决策	流量调度	响应处理
Berthier et al.[61]	无	分全拒绝、全接收、随机选择、载荷匹配、攻击源匹配五种转发模式	同网段 TCP 握手重放和序列号同步	无
Lu et al.[62]	IDS 检测	按业务流和攻击流进行转发	跨网段的 OpenVPN 代理和策略路由	无
Han et al.[63]	无	按攻击目标端口进行转发	同网段的组播	筛选响应并擦除蜜罐特征
Fan et al.[64]	Snort 检测	按低交互和高交互进行转发	同网段 SDN 转发	无
Guarnizo et al.[49]	无	全转发	跨网段 SSH 隧道	无
Kyung et al.[65]	无	按历史最佳响应进行转发, 否则进行组播	同网段的 SDN 转发	(组播模式)筛选响应并擦除蜜罐特征
Rashidi et al.[66]	无	评估信任级别, 按信任等级转发	同网段的 SDN 转发	无

判别应用程序指纹特征差异已经有很长的历史, 并提出了多种以二进制分析和协议逆向为主的理论和原型方法[67-69]。2015 年, 密歇根大学 Chen 等[70]通过对 6900 个不同来源的恶意软件样本分析, 总结了硬件、环境、应用和行为四个抽象层次的反虚拟化和反调试的检测指纹, 并分析了检测方法的准确度、使用复杂度和反识别方法。同年, 印度斋浦尔马拉维亚国立技术学院 Gajrani J 等[71]针对安卓恶意软件检测问题, 总结了后台进程、性能、行为等 12 维沙箱检测方法。2017 年芬兰图尔库大学 Joni Uitto 等[72]剖析了 Chen 和 Gajrani J 工作, 并总结了时间、操作、硬件和环境四类的蜜罐指纹。上述工作都着重分析了恶意代码执行环境的蜜罐指纹, 对物联网蜜罐的识别方法缺乏细粒度的分析和评估。因此, 针对物联网蜜罐可能存在的识别特征, 接下来本文将对相关的蜜罐识别特征进行简单的阐述。

1) 时间的差异。由于蜜罐监控模块的额外开销, 不可避免地带来特定的网络和主机延迟特征。这两种延迟分别指网络外部通信延迟和主机 API 或指令级别的系统延时。2005 年德国亚琛工业大学 Holz 等[73]表明, 由于日志捕获和沙箱执行本身的额外开销, 攻击者命令的执行时间可能会明显延长。2006 年马萨诸塞大学洛厄尔分校 Fu 等[74]发现攻击者可以使用网络层的延迟来作为 Honeyd 蜜罐指纹。2007 年美国新墨西哥理工大学 Mukkamala 等[75] 实验证明了虚拟蜜罐对 ICMP 回应请求的响应确实比实际系统慢。对于物联网设备而言, 由于部分业务涉及对物理设备的操作, 其对网络通信或系统时延的实时性要求更高。因此, 如果时延处理不当, 物联网蜜罐很容易被攻击者识别。

2) 软硬件的差异。由于大部分蜜罐都采用硬件虚拟化的方式进行资源模拟, 与实际的业务系统的执行环境存在明显差异。软件实现上, 2007 年卡内基·梅隆大学 Brumley 等[67]表示, 通过从二进制文件中自动构建符号公式, 他们可以在 HTTP 和 NTP 的协议实现中找到蜜罐系统与真实系统之间的差异。2014 年德州农工大学 Xu 等通过二进制分析生成恶意 C&C 服务器的指纹[76]。关于协议逆向工程的识别, 2009 年维也纳科技大学 Comparetti 等开发了一种提取协议特征的系统 Prospex[68], 可用于识别协议偏差。2014 年 SANS 技术研究所[77]的报告指出攻击者可使用“file /sbin /init”命令的动态返回值识别 Kippo 蜜罐。在硬件特性上, 2016 年 Litchfield S 等指出, 大部分蜜罐对物理域采集与执行设备的实现不完善[57], 其感知数据违背物理规律, 如温度传递、水位感知等实现过程容易被攻击者识别。

3) 网络的差异。通过网络响应来识别漏洞和提取网络服务特征的基本研究方法, 常用的工具包括 Nmap [78]和 ZMap [79]。Censys.io 对所有主要协议每周进行扫描, 抓取所有互联网节点对外接口信息[80]。Shodan 提供了一个在线工具[81], 允许任何人检查特定主机是蜜罐还是真正的工业控制系统(ICS)。2005 年威斯康星大学 Bethencourt 等通过网络扫描和互联网公开的感知数据进行关联[82], 实现互联网中蜜罐的识别与发现。为了更全面的表征蜜罐网络指纹, 2016 年中国科学院信息工程研究所 Feng 等[83]进行了互联网范围的工业控制设备发现和识别, 通过训练概率模型和启发式算法来剔除工控蜜罐的干扰, 采用了包括开放端口数量和 HTTP 配置等四个特征。

4) 操作的差异。由于法律和道德要求, 网络管理员在知情条件下有义务阻止受感染的主机继续感染其他设备。这也为蜜罐识别提供了一个思路。2004 年 Krawetz 开发了一种名为 Honeypot Hunter[84]的工具来识别蜜罐。由于低交互蜜罐可能无法模拟高级

功能, 此工具通过测试受感染系统对外发送垃圾电子邮件是否成功来识别蜜罐。由于蜜罐参与真实攻击和执行恶意活动的能力有限, 2006 年中佛罗里达大学 Zou 和 Cunningham 通过检查受感染机器是否能够成功向攻击者的传感器发送未修改的恶意流量来判别蜜罐[85]。

物联网设备的接口多样、通信时效严苛和物理信息融合的特点, 都可以作为针对物联网蜜罐识别的有效特征, 如设备的接口组合指纹、通信时延或丢包指纹、物理域操作行为指纹等。蜜罐识别技术涉及网络设备发现、虚拟环境检测、软件指纹检测等方面的研究, 往往能够提取各类仿真工具的特征, 在蜜罐构造过程中应避免采用这些容易被识别的工具。

4.3 数据分析技术

蜜罐是调查和预防网络犯罪的重要工具, 其最终目的是通过观察攻击者在蜜罐中的活动分析出安全威胁情报。物联网设备出于成本和易用性的考虑, 往往缺乏严格的安全开发规范和安全测试流程, 大量设备直接暴露在互联网中, 为恶意软件的入侵和传播提供了巨大的空间, 这也对蜜罐数据分析提出了更高的要求。物联网蜜罐数据分析是整理威胁数据并评估威胁事件的过程, 往往采用各类统计指标佐证分析观点, 每个指标具有不同的分析角度和准确性。物联网蜜罐常用的分析方法可分为统计分析和深度分析。统计分析是针对原始数据进行的统计和查询等工程化分析, 通常包括攻击源、目标活动和二进制文件的分析; 深度分析是对攻击者的隐藏特征进行挖掘和推理。

在统计分析中, 攻击源的识别不依赖于蜜罐的体系结构或交互深度, 一旦攻击发生, 通过连接信息检索出各类攻击源信息。常见的攻击源描述信息包括 IP 地址、IP 前缀、自治系统号、域名、URL、国家、用户代理、操作系统等。2010 年, 基于 SIP 和 VoIP 协议所使用“user-agent”名称, 埃迪斯科文大学 Craig Valli 等在蜜罐分析中构建了攻击源网络指纹[86]。具有该协议标签的任何协议都可以采用这种方法进行攻击源攻击源识别, 但“user-agent”是可以被省略并且容易被篡改。为了推断攻击源的操作系统, 通常使用额外如 p0f 的被动操作系统指纹识别工具, 通过分析数据包的实现差异来识别攻击者的操作系统。目标活动分析是统计分析中的另一个重点, 分析攻击者在目标 IP 地址、端口和服务上的行为, 分析结果通常包括协议占比、端口分布和交互行为统计。大量蜜罐分析结果表明, 物联网攻击最常用的协议是 Telnet 协议[7], 并广泛存在针对物联网设备的暴力破解和字典攻击[41]。攻击频率分析主要用于衡量攻击强度, 常见的分析方法有首次攻击时间、单位时间窗口的攻击量、单位时间窗口的新攻击、单位时间窗口的数据包和单位时间窗口的数据包大小等。对于物联网蜜罐捕获的二进制文件, 一般是通过静态分析和动态分析解析攻击方式和意图[53]。但在蜜罐分析中往往采用 VirusTotal[87]在线分析结果[42-53]来快速评估二进制文件危害程度和蜜罐捕获未知威胁的能力。物联网恶意软件通常与僵尸网络关联(如 Mirai), 并通过远程控制进行加密货币挖掘[7]或发动的各类 DDoS 攻击[88]。

在深度分析中, 通常需要提出合理的分析框架对统计信息进行二次关联与聚合, 实现对攻击源、攻击方式和攻击意图的深入理解。如图 13 所示, 文献[51]在统计分析的基础上结合物联网僵尸网络中扫描器、入侵服务器、下载服务器、命令与控制服务器和 DNS 服务器的五种不同连接关系, 分析出了 8 种不同物联网僵尸网络架构, 并总结了各类恶意软件家族发动 Telnet 攻击时在入侵、感染、变现三个阶段的行为模式。2018 年, 中国科学院大学 Ke Li 等[89]针对工控蜜罐数据缺乏深度信息处理方法的问题, 提出了基于非参模型 DP-means 的攻击源识别算法。该算法将攻击行为模式具体划分为攻击目标、攻击频率和攻击方法三维量化向量, 实现攻击源身份的关联与区分。

5 物联网蜜罐的评估体系

物联网蜜罐的目的是发现攻击并研究物联网攻击特征。近年来, 物联网威胁的复杂性不断提高、攻击频率激增, 蜜罐诱饵的真实度和监控能力也在很大程度上获得了提升。为了客观分析不同蜜罐的能力, 结合物联网杀伤链模型, 本节量化评估了蜜罐诱饵环境和监控模块。

5.1 网络杀伤链模型

“杀伤链”的概念最初是指军事领域中对攻击目标从探测到破坏的一系列处理过程。此过程包括“发现-定位-跟踪-瞄准-打击-达成目标”六个环节, 主要用于评估攻击过程和制定各阶段防御措施。随着网络空间成为大国博弈的新战场, 2011 年美国国防承包商洛克希德·马丁公司正式提出网络空间杀伤链[90]。如图 14 所示, 杀伤链模型分为“侦察-武器化-投递-利用-部署-命令与控制-目标达成”七个过程, 详细描述了普遍适用的网络攻击流程与防御概念。

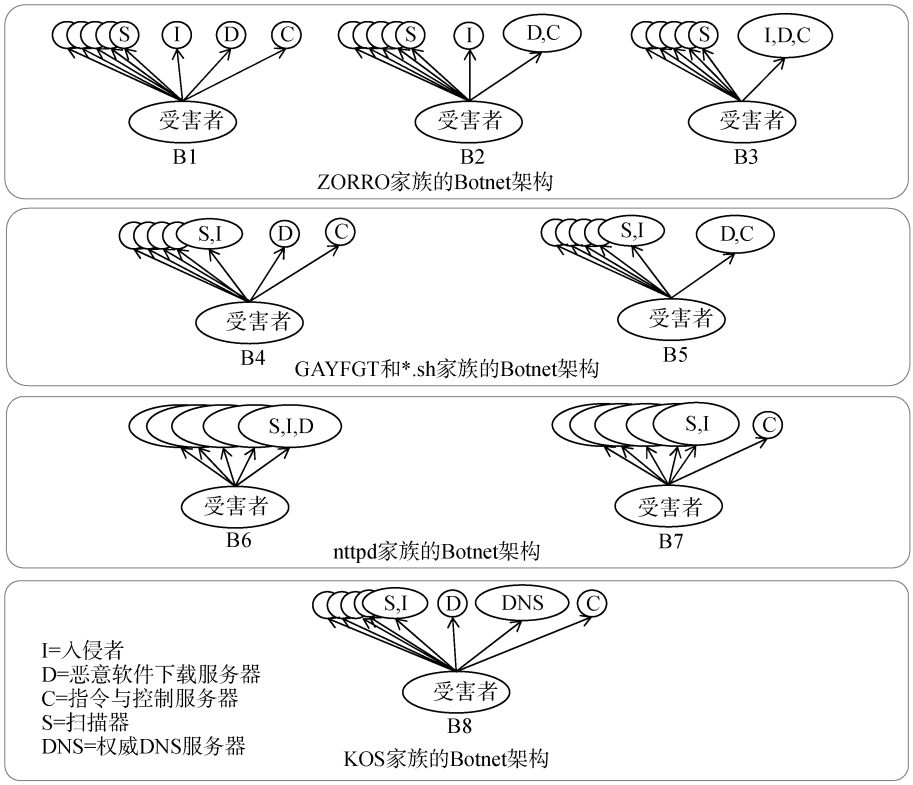

图 13 文献[53]中的僵尸网络架构

Figure 13 Botnet Architectures Proposed in [53]

物联网的多元服务环境导致相同阶段的内外网攻击方式及防护手段有所差异，2018 年韩国首尔延世大学研究生院信息学院 Hyeob Kim 等提出了针对物联网多元服务改进的网络杀伤链模型[91]，针对物联网领域服务多元化、设备异构和入侵媒介多样的特点，在七个杀伤链过程的基础上，强调了内外网的差异，如侦察的媒介、武器化的特征、投递的方式和利用的漏洞等。杀伤链模型准确地提取了网络攻击的七种关键要素，是衡量物联网攻防博弈的深度的有效方法，为蜜罐评估体系的建立奠定了基础。

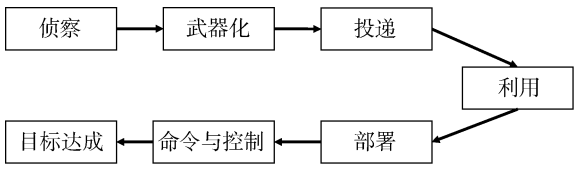

图 14 杀伤链模型

Figure 14 Model of “Kill Chain”

5.2 基于杀伤链模型的蜜罐评估体系

蜜罐的意义在于发现、捕获和分析攻击，理应从攻击角度评估蜜罐的效果。现有的蜜罐评价方法多从架构与实现原理角度考虑，鲜有针对攻防博弈特征要素入手开展分析工作。从 4.1 节的分析可得，杀伤链模型可以拆分恶意软件的每个攻击阶段，能够剖析蜜罐诱饵的交互深度和不同阶段的监控能力。根据物联网蜜罐诱导及监控深度，各蜜罐组成部分在杀伤链模型中具备新的内涵。如图 15 所示，基于物联网杀伤链模型，本文划分了诱饵环境和监控能力的深度层次，并赋予各层指标数值，提出了一种物联网蜜罐评估体系。

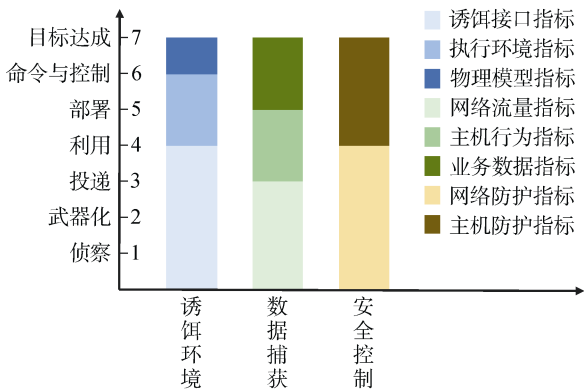

图 15 基于杀伤链的物联网蜜罐评估体系

Figure 15 IoT Honeypot Evaluative Criteria Based on Cyber Kill Chain Model

在诱饵环境方面，评估要素分为诱饵接口、执行环境和物理模型三个层次。诱饵接口用于完成杀伤

链侦察、武器化、投递和利用阶段的通信诱导。诱饵接口指标是衡量蜜罐对外通信深度和广度的指标，主要评定物联网蜜罐的漏洞部署能力和对外通信服务的完成度。执行环境用于完成杀伤链部署阶段的系统命令执行。执行环境指标是衡量为恶意软件启动或运行提供系统依赖环境深度的指标，主要评定物联网蜜罐对恶意软件执行的包容度和鲁棒性。物理模型用于完成杀伤链命令与控制、目标达成阶段的正确数据反馈。物理模型指标是指为攻击者提供真实业务场景能力的指标，主要评定物联网蜜罐为攻击者提供业务资源的真实度和完备性。

在数据捕获方面，评估要素分为网络流量捕获、主机行为捕获和业务数据捕获三个层次。网络流量捕获用于实现物联网杀伤链侦察、武器化和投递阶段行为数据的捕获。网络流量指标是衡量蜜罐采集网络攻击流量完整性和可靠性的指标，包括入站出站链接、数据包、包头信息、漏洞利用信息或有效载荷等指标；主机行为捕获用于实现对杀伤链利用和部署阶段的行为的捕获。主机行为捕获指标是指蜜罐捕获攻击者主机活动的指标，包括击键记录、系统调用、进程行为等指标；业务数据捕获用于实现对杀伤链命令与控制和目标达成阶段行为数据的捕获。业务数据捕获指标是指蜜罐捕获攻击者业务操作行为的指标，包括攻击意图、操作数据等指标。

在安全控制方面，评估要素分为网络防护和主机防护两个层次。网络防护是指在杀伤链侦察、武器化、投递和利用阶段保障蜜罐系统和正常生产业务的安全防护措施。网络防护指标是指蜜罐限制攻击网络行为的能力指标，包括网络阻断、转移和黏着等指标；主机防护是指在杀伤链部署、命令与控制和目标达成阶段保障蜜罐系统和正常的生产业务的安全防护措施。主机防护指标是指蜜罐对恶意软件在主机内实施破坏行为的防控指标。

如图 16，本文依据评估指标人工标注了相关物联网蜜罐的各项能力。在诱饵环境构建方面，由于物联网蜜罐封闭私有特点，“利用”阶段是明显的分界线，多数物联网诱饵环境仅支持恶意载荷的投递，无法支持攻击执行结果的仿真。诱饵环境能力值达到 7 的蜜罐中，除了实物蜜罐以外，信息物理蜜罐框架也能够吸引攻击者完成从侦察到目标达成的全部攻击流程。但这些框架多数仍处于概念验证阶段，诱捕对象有极大局限性，缺乏应用部署能力。在数据捕获方面，由于物联网设备难以部署主机检测工具，实物蜜罐存在明显劣势，通常采用网络流量嗅探的捕获方案。反之，虚拟物联网蜜罐以高度可操作的诱饵环境，可支持二进制代码及细粒度的捕获操作。安全控制方面，基于相关法律和伦理要求，对于已知的对外攻击蜜罐必须采取防护措施。特别地，由于物联网的封闭特性，蜜罐系统中的实物设备的状态往往难以全面感知，对安全控制有更高要求，但多数物联网蜜罐研究并没有提及相关的防护策略。

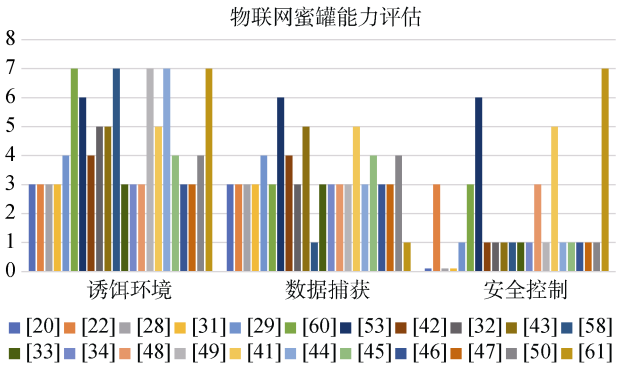

图 16　物联网蜜罐的对比

Figure 16　Contrastive Analysis of IoT Honeypot

6　未来研究方向展望

由以上对物联网蜜罐的发展和关键技术评估可知，目前尚不存在广泛适用的物联网蜜罐应用框架和明确的研究理论体系。多样化、私有化和物理融合化的特点导致物联网蜜罐统一架构缺乏、实物环境难以部署和虚拟环境技术门槛过高的问题。此外，物联网攻击正在不断演化，各种新兴的攻击方式也为蜜罐数据分析带来了挑战。

通过对以上问题的分析，本文认为未来物联网蜜罐的研究方向主要包括以下三个方面。

1) 虚实结合诱饵环境的构建。无论是设备密集型的实物环境还是技术密集型的虚拟环境都需要大量的人力物力投入，虚实结合的诱饵环境将成为主流。对于实物环境与虚拟环境的有效协作，一方面探寻灵活切换、时延稳定、决策准确的重定向机制，另一方面研究通过机器学习等手段在实物设备交互数据中自我提升的能力。

2) 物联网攻击流量的深度解析。网络流量是物联网蜜罐最常见的数据格式，针对无加密或弱加密的物联网专有协议，探寻如何提取流量中有效执行载荷的方法，从而缓解难以在物联网设备中安装监控工具的问题。

3) 针对新兴物联网威胁的蜜罐研究。借助规模化的物联网蜜罐，探寻对僵尸网络架构、来源等的深

入分析方法, 并思考如何利用物联网蜜罐进行 DDoS 攻击防护。此外, 随着加密货币领域的崛起, 利用蜜罐系统研究挖矿类物联网病毒也势在必行。

7 总结

物联网蜜罐是新的网络安全形势下的主动防御手段, 其架构封闭性、诱饵多样性和物理融合性的特点导致难以实现通用的高交互物联网蜜罐。结合物联网的特征, 本文提炼了工控蜜罐、消费级物联网蜜罐、信息物理系统蜜罐的技术发展主线。在分析物联网蜜罐关键技术的过程中, 为了科学地评估入侵接口构建能力, 本文提取了 Wireshark 源码中的功能码标准作为统一指标对相关工作进行测试验证和评估。此外, 本文总结了蜜罐识别与反识别的方法, 从时间、软硬件、网络和操作四个维度梳理了现有蜜罐识别工作。

最后, 基于网络杀伤链模型, 本文明确了物联网攻击的普遍攻击方式和阶段, 提出了一种基于杀伤链模型的物联网蜜罐评估体系, 阐述了蜜罐评估中的各项指标。通过对物联网蜜罐特性和关键技术的研究发现, 虽然目前的蜜罐研究已经描述了物联网蜜罐的物理特征, 并提出了相关的概念框架, 但是仍缺乏针对物理空间的完整实现和对物理空间攻击者模型的研究。

由于物联网僵尸网络、工控网络 APT 等攻击技术不断演进, 亟需采用物联网蜜罐对抗性思维进行博弈研究。蜜罐的交互程度越高, 被识别的可能性越大, 而物联网设备相较于普通的互联网节点具有更加独特的指纹特征。如何针对性地进行物联网蜜罐反识别也是蜜罐研究者应该重点关注的问题。物联网蜜罐技术将跟踪物联网安全威胁的发展与更新, 并有望成为物联网安全防护体系中的重要组成部分。

参考文献

[1] L. Spitzner. Honeypots: tracking hackers. Hacker, 2003.

[2] Leading the IOT. H. Mark, https://www.gartner.com/imagesrv/boo.ks/iot/iotEbook_digital.pdf, 2017.

[3] 2017 年中国互联网网络安全报告. CNCERT, http://www.cert.org.cn/publish/main/upload/File/2017annual(1).pdf, June. 2018.

[4] 2016 dyn cyberattack. Wikipedia, https://en.wikipedia.org/wiki/.2016_Dyn_cyberattack, July. 2018.

[5] Hacking the Human Heart. G. MARC. 2017.

[6] Venezuela hit by major power outage. The Mercury News, https://www.mercurynews.com/2019/03/07/venezuela-hit-by-major-power-outage/, 2019.

[7] Metongnon L, Sadre R. Beyond Telnet: Prevalence of IoT Protocols in Telescope and Honeypot Measurements[C]. *The 2018 Workshop on Traffic Measurements for Cybersecurity*, 2018: 21-26.

[8] Acien A, Nieto A, Fernandez G, et al. A Comprehensive Methodology for Deploying IoT Honeypots[M]. Trust, Privacy and Security in Digital Business. Cham: Springer International Publishing, 2018: 229-243.

[9] L. Spitzner. The honeynet project: trapping the hackers[C]. *IEEE Security and Privacy (SP'03)*, 2003: 15-23.

[10] W. Z. Zhang, B. S. Qu. Security architecture of the Internet of Things oriented to perceptual layer[C]. *International Journal on Computer, Consumer and Control (IJ3C'13)*, 2013: 37-45.

[11] Siemens. Wikipedia, https://en.wikipedia.org/wiki/Siemens, Nov. 2018.

[12] Schneider Electric. Wikipedia, https://en.wikipedia.org/wiki/Schn.eider_ElectricNov. 2018.

[13] Advanced persistent threat. Wikipedia. https://en.wikipedia.org/.wiki/Advanced_persistent_threat. Nov. 2018.

[14] Know Your Enemy: Defining Virtual Honeynets. Honeynet Project, http://project.honeynet.org/papers/index.html, Sep. 2002.

[15] Honeypot Farms. L. Spitzner, Aug. 2003. https://www.symantec.com/connect/articles/honeypot-farms

[16] L. Spizner. Honeytokens: The Other Honeypot. Security Focus Information, July 2003.

[17] C. Carella, J. Dike, N. Fox, et al. UML extensions for honeypots in the ISTS Distributed Honeypot Project[C]. *Information Assurance Workshop*, 2008:265-274.

[18] Stoll C, Connolly J W D. The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage[J]. *Physics Today*, 1990, 43(8): 75-76.

[19] Modbus. Wikipedia, https://zh.wikipedia.org/wiki/Modbus, May. 2018.

[20] SCADA HoneyNet Project: Building Honeypots for Industrial Networks. V. Pothamsetty and M. Franz, http://scadahoneynet.sourcefor.ge.net/. 2005.

[21] N. Provos. Honeyd: A Virtual Honeypot Daemon[C]. *The 12th USENIX Security Symposium (USENIX '03)*, 2003: 1-7.

[22] Installation Instructions Virtual PLC Honeynet. Digtial Bond Inc., http://www.digitalbond.com/wp-content/uploads/2011/02/Installation_Instructions.pdf, 2006.

[23] G. Chamales.The honeywall cd-rom[C]. *IEEE Security and Privacy (SP'04)*, 2004, 2(2): 77-79.

[24] ljean/modbus-tk. ljean, https://github.com/ljean/modbus-tk/blob/master/modbus_tk/defines.pyJun. 2017.

[25] PyModbus Documentation. Sanjay, 2https://media.readthedocs.org/pdf/pymodbus/latest/pymodbus.pdf. Oct. 2018.

[26] stephane/libmodbus. stephane, https://github.com/stephane/ libmodbus/ blob/master/src/modbus.h. Jun. 2018.

[27] S7 Communication (S7comm). Wireshark, . https://wiki.wireshark.org/S7commMay, 2016.

[28] Conpot. L. Rist, http://conpot.org/, May. 2013.

[29] Buza D I, Juhász F, Miru G, et al. CryPLH: Protecting Smart Energy Systems from Targeted Attacks with a PLC Honeypot[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014: 181-192.

[30] SIMATIC STEP 7 and WinCC V15 TRIAL Download. Simens Inc., https://support.industry.siemens.com/cs/document/109752566/simatic-step-7-and-wincc-v15-trial-download-?dti=0&lc=en-CN, Dec. 2017.

[31] Step7 Open Source Ethernet Communication Suite. D. Nardella, http://snap7.sourceforge.net/, Jun. 2018.

[32] Lau S, Klick J, Arndt S, et al. POSTER: Towards Highly Interactive Honeypots for Industrial Control Systems[C]. *The 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016: 1823-1825.

[33] C. H. Zhao, S. J. Qin. A research for high interactive honeypot based on industrial service[C]. *2017 3rd IEEE International Conference on Computer and Communications (ICCC'17)*, 2017:258-264.

[34] Xiao F, Chen E H, Xu Q. S7commTrace: A High Interactive Honeypot for Industrial Control System Based on S7 Protocol[M]. Information and Communications Security. Cham: Springer International Publishing, 2018: 412-423.

[35] wireshark/wireshark. Wireshark, https://github.com/wireshark/wireshark/blob/master/epan/dissectors/packet-s7comm.h. Feb. 2018

[36] mushorg/conpot, Release 0.5.2 (Python2 retirement imminent!). mushorg, https://github.com/mushorg/conpot/releases, Aug. 2018.

[37] Snap7. D. Nardella, https://sourceforge.net/projects/snap7/files/, Jun. 2018.

[38] V. Alexandru, O. Sebastian, Y. Deryeuan. ICS Threat Analysis Using a Large-Scale Honeynet[M]. *International Symposium for Ics & Scada Cyber Security Research*, British Computer Society, 2015.

[39] S. Poeplau, J. Gassen. A honeypot for arbitrary malware on usb storage devices[C]. *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS'12)*, 2012: 1-8.

[40] A. Podhradsky, C. Casey, P. Ceretti. The bluetooth honeypot project: Measuring and managing bluetooth risks in the workplace[J]. *International Journal of Interdisciplinary Telecommunications and Networking*, 2012, 4(3): 1-22.

[41] S. Dowling, M. Schukat, H. Melvin. A ZigBee honeypot to assess IoT cyberattack behavior[C]. *28th Irish Signals and Systems Conference (ISSC'17)*, 2017: 1-6.

[42] M. Kara, A. İkinci. HoneyThing: Nesnelerin İnterneti için Tuzak Sistem[C]. *8th International Conference on Information Security and Cryptology (ISCTurkey'15)*, 2015:258-264.

[43] Chamotra S, Sehgal R K, Ror S, et al. Honeypot Deployment in Broadband Networks[M]. Information Systems Security. Cham: Springer International Publishing, 2016: 479-488.

[44] Iskhakova A, Meshcheryakov R, Iskhakov A, et al. Analysis of the Vulnerabilities of the Embedded Information Systems of IoT-devices through the Honeypot Network Implementation[C]. Proceedings of the IV International research conference \"Information technologies in Science, Management, Social sphere and Medicine\" (ITSMSSM 2017), 2017: 268-274.

[45] P. Krishnaprasad. Capturing attacks on IoT devices with a multi-purpose IoT honeypot [D]. Indian Institute of Technology Kanpur, May. 2017.

[46] M. Wang, J. Santillan, F. Kuipers. Thingpot: an interactive internet-of-things honeypot[EB/OL]. arXiv:1807.04114 [cs.NI], 2018.

[47] M. A. Hakim, H. Aksu, A. S. Uluagac, et al. U-pot: a honeypot framework for upnp-based iot devices[EB/OL]. arXiv:1812.05558 [cs.CR], 2018.

[48] T. Luo, Z. Xu, X. Jin, et al. Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices[OL]. Black Hat, 2017.

[49] J. D. Guarnizo, A. Tambe, S. S. Bhunia, et al. SIPHON: Towards Scalable High-Interaction Physical Honeypots[C]. *3rd ACM Workshop on Cyber-Physical System Security (CPSS'17)*, 2017:456-462.

[50] U. D. Gandhi, P. M. Kumar, R. Varatharajan, et al. Hiotpot: surveillance on iot devices against recent threats[C]. *Wireless Personal Communications*, 2018: 458-492.

[51] Kippo—SSH Honeypot. http://code.google.com/p/kippo/, 2011.

[52] Cowrie-active kippo fork. M. Oosterhof, http://www.cowrie.org/, July. 2015.

[53] Y. M. P. Pa, S. Suzuki, K. Yoshioka, et al. IoTPOT: analysing the rise of IoT compromises[C]. *The 9th USENIX Conference on Offensive Technologies (WOOT'15)*, 2015: 9.

[54] QEMU. Wikipedia, https://en.wikipedia.org/wiki/QEMU, Oct. 2018.

[55] OpenWrt. Wikipedia, https://en.wikipedia.org/wiki/OpenWrt, Nov. 2018.

[56] Q. B. Sun, J. Liu, S. Li, et al. Internet of Things：Summarize on Concepts，Architecture and Key Technology Problem[J]. *Journal of Beijing University of Posts and Telecommunications*, 2010, 33(3): 1-9.（孙其博, 刘杰, 黎彝, 等. 物联网:概念、架构与关键技术研究综述[J]. *北京邮电大学学报*, 2010, 33(3), 1-9.）

[57] Litchfield S, Formby D, Rogers J, et al. Rethinking the Honeypot

for Cyber-Physical Systems[J]. *IEEE Internet Computing*, 2016, 20(5): 9-17.

[58] Zhou K. A Honeypot Process Control System Platform Based on Honeyd[D]. Shanghai: East China University of Science and Technology, 2015.(周昆. 一种基于 Honeyd 的过程控制蜜罐系统的平台搭建研究[D]. 上海: 华东理工大学, 2015.)

[59] Antonioli D, Tippenhauer N O. MiniCPS: A Toolkit for Security Research on CPS Networks[C]. *The First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaC*, 2015: 91-100.

[60] Irvene C, Formby D, Litchfield S, et al. HoneyBot: A Honeypot for Robotic Systems[J]. *Proceedings of the IEEE*, 2018, 106(1): 61-70.

[61] R. G. Berthier. Advanced honeypot architecture for network threats quantification[D]. 2009

[62] T. F. Lu, Z. J. Chen, J. W. Zhuge, et al. Research and Implementation of Network Attack Flow Redirection Mechanism in the Honeyfarm Environment[J]. *Journal of Nanjing University of Posts and Telecommunications (Natural Science)*, 2009, 29(3): 14-20. (陆腾飞, 陈志杰, 诸葛建伟, 等. 邹维面向蜜场环境的网络攻击流重定向机制的研究与实现[J]. *南京邮电大学学报(自然科学版)*, 2009, 29(3): 14-20.)

[63] W. Han, Z. Zhao, A. Doupé, et al. Honeymix: Toward sdn-based intelligent honeynet[C]. *The 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization (SDN-NFV Security'16),* 2016: 1-6.

[64] W. Fan, D. Fernandez, W. Fan, et al. A novel SDN based stealthy TCP connection handover mechanism for hybrid honeypot systems[C]. *IEEE Conference on Network Softwarization (NetSoft'17)*, 2017: 589-591.

[65] S. Kyung, W. Han, N. Tiwari, et al. Honeyproxy: Design and implementation of next-generation honeynet via sdn[C]. *2017 IEEE Conference on Communications and Network Security (CNS'17)*, 2017: 1-9.

[66] B. Rashidi, C. Fung, K.W. Hamlen, et al. HoneyV: A virtualized honeynet system based on network softwarization[C]. *IEEE/IFIP Network Operations and Management Symposium (NOMS'18)*, 2018: 1-5.

[67] D. Brumley, J. Caballero, Z. Liang, et al. Towards Automatic Discovery of Deviations in Binary Implementations with Applications to Error Detection and Fingerprint Generation[C]. *USENIX Security Symposium (USENIX '07)*, 2007: 486-492.

[68] P. M. Comparetti, G. Wondracek, C. Kruegel, et al. Prospex: Protocol specification extraction[C]. *30th IEEE Symposium on Security and Privacy (S&P '09)*, 2009: 110–125.

[69] G. Wondracek, P. M. Comparetti, C. Kruegel, et al. Automatic Network Protocol Analysis[C]. *The 16th Annual Network & Distributed System Security Symposium (NDSS '08)*, 2008 1-14.

[70] X. Chen, J. Andersen, Z. Morley, et al. Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware[C]. *IEEE/IFIP International Conference on Dependable Systems & Networks (DSN'08)*, 2008: 177-186.

[71] Gajrani J, Sarswat J, Tripathi M, et al. A Robust Dynamic Analysis System Preventing SandBox Detection by Android Malware[C]. *The 8th International Conference on Security of Information and Networks*, 2015: 290-295.

[72] Uitto J, Rauti S, Laurén S, et al. A Survey on Anti-honeypot and Anti-introspection Methods[M]. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2017: 125-134.

[73] T. Holz, F. Raynal. Detecting honeypots and other suspicious environments[C]. *The Sixth Annual IEEE SMC Information Assurance Workshop (IAW'05)*, 2005: 29-36.

[74] X. Fu, W. Yu, D. Cheng, et al. On recognizing virtual honeypots and countermeasures[C]. *The 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC '06)*, 2006: 211-218.

[75] S. Mukkamala, K. Yendrapalli, R. Basnet, et al. Detection of Virtual Environments and Low Interaction Honeypots[C]. *IEEE SMC Information Assurance & Security Workshop (IAW'07)*, 2007: 92-98.

[76] Xu Z Y, Nappa A, Baykov R, et al. AUTOPROBE: Towards Automatic Active Malicious Server Probing Using Dynamic Binary Analysis[C]. *The 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014: 179-190.

[77] Kippo Users Beware: Another Fingerprinting Trick. Sans Technology Institude, https://isc.sans.edu/forums/diary/Kippo+Users+Beware+Another+fingerprinting+trick/18119/, accessed September 16, 2019.

[78] G. F. Lyon. Nmap network scanning: The official Nmap project guide to network discovery and security scanning[C]. *Insecure*, 2009:154-162.

[79] Z. Durumeric, E. Wustrow, J. A. Halderman. ZMap: Fast Internet-wide scanning and its security applications[C]. *The 22nd USENIX Security Symposium (USENIX Security'13)*, 2013: 605-619.

[80] Durumeric Z, Adrian D, Mirian A, et al. A Search Engine Backed by Internet-Wide Scanning[C]. *The 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015: 542-553.

[81] Shodan. Honeyscore. SHODAN, https://honeyscore.shodan.io/, 2017.

[82] J. Bethencourt, J. Franklin, M. K. Vernon. Mapping Internet Sensors with Probe Response Attacks[C]. *The USENIX security symposium (USENIX'05)*, 2005: 193–208.

[83] X. Feng, Q. Li, H. Wang, et al. Characterizing industrial control system devices on the internet[C]. *2016 IEEE 24th International*

Conference on Network Protocols (ICNP'16), 2016: 1–10.

[84] Krawetz N. Anti-honeypot Technology[J]. *IEEE Security & Privacy Magazine*, 2004, 2(1): 76-79.

[85] C. C. Zou, R. Cunningham. Honeypot-aware advanced botnet construction and maintenance[C]. *IEEE International Conference on Dependable Systems and Networks (DSN'06)*, 2006: 199-208.

[86] C. Valli. An analysis of malfeasant activity directed at a voip honeypot[C]. *The 8th Australian Digital Forensics Conference (ADF'10)*, 2010: 258-264.

[87] Virustotal. Hispasec Sistemas, https://www.virustotal.com/, 2004.

[88] M. Anirudh, S. A. Thileeban, D. J. Nallathambi. Use of honeypots for mitigating DoS attacks targeted on IoT networks[C]. *IEEE 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP'17)*, 2017: 1-4.

[89] K. Li, J. You, H. Wen, et al. Collaborative Intelligence Analysis for Industrial Control Systems Threat Profiling[C]. *The Future Technologies Conference (FTC'19)*, Cham, 2018: 94-106.

[90] E. M. Hutchins, M. J. Cloppert, R. M. Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains[J]. *Leading Issues in Information Warfare & Security Research*, 2011, l(1): 80.

[91] Kim H, Kwon H, Kim K K. Modified Cyber Kill Chain Model for Multimedia Service Environments[J]. *Multimedia Tools and Applications*, 2019, 78(3): 3153-3170.

游建舟 于 2015 年在厦门大学电子信息工程专业获得学士学位。现在中国科学院大学通信与信息系统专业攻读博士学位。研究领域为工控安全、物联网安全。研究兴趣包括: 工控蜜罐设计、大数据分析。Email: youjianzhou@iie.ac.cn

孙玉砚 于2016年在中国科学院信息工程研究所获得博士学位。现任中国科学院信息工程研究所助理研究员, 研究领域工业控制系统的信息安全, 特别是电力工控安全。研究兴趣包括电网脆弱性分析、入侵检测。Email: sunyuyan@iie.ac.cn

孙利民 于 1998 年在国防科学技术大学计算机体系结构专业获得工学博士学位。现中国科学院信息工程研究所第四研究室研究员。物联网安全、工业控制系统安全。研究兴趣包括: 工控入侵诱捕、工控态势感知。Email: sunlimin@iie.ac.cn

吕世超 于 2018 年在中国科学院大学信息安全专业获得工学博士学位。现任中国科学院信息工程研究所第四研究室高级工程师。研究领域为物联网安全、工业控制系统安全。研究兴趣包括: 工控入侵诱捕、工控态势感知。Email: lvshichao@iie.ac.cn

石志强 于 2001 年在中国科学院软件研究所计算机应用技术专业获得工学博士学位。现任中国科学院信息工程研究所正研级高级工程师。研究领域为工业控制系统安全、嵌入式固件脆弱性分析。Email: shizhiqiang@iie.ac.cn