基于数据包分片的工控蜜罐识别方法

游建舟[1,2], 张悦阳[1,2], 吕世超[1,2*], 陈 新[1,2], 尹丽波[3], 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
[3] 国家工业信息安全发展中心, 北京 中国 100040

摘要 蜜罐是一种用于安全威胁发现与攻击特征提取的主动防御技术, 能够提供高价值且低误报率的攻击流量和样本。蜜罐的应用压缩了网络黑客的隐匿空间, 攻击者可通过蜜罐识别技术来发现和规避蜜罐。因此, 安全人员有必要从攻击者的角度深入研究蜜罐识别的方法, 以便优化蜜罐系统的设计与实现。本文从蜜罐的结构出发, 总结了 8 种蜜罐识别要素, 并评估了不同识别要素的准确性和隐蔽性。结合互联网蜜罐分布特点, 归纳了一种互联网中的蜜罐识别流程, 并基于 Conpot 工控蜜罐架构的固有缺陷, 提出了一种基于数据包分片的工控蜜罐识别方法。通过三次互联网扫描, 共发现 2432 个 Conpot 工控蜜罐, 并进一步分析了其分布特点。

关键词 蜜罐识别; 数据包分片; 蜜罐
中图法分类号 TP393.08 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2019.05.06

Method of ICS Honeypot Identification Based on Packet-Sharding

YOU Jianzhou[1,2], ZHANG Yueyang[1,2], LV Shichao[1,2*], CHEN Xin[1,2], YIN Libo[3], 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
[3] National Industrial Information Security Development Research Center, Beijing 100040, China

Abstract Honeypot is one kind of active defence technology for threat discovery and attack signature generation, providing highly valuable data and samples with low false alarm rate. With the development of honeypot technology, there is less and less space for attackers to disguise themselves. Attackers devote themselves to recognize honeypot and circumvent them via honeypot identification technology. Therefore, it is critical that researchers dive into the identification technology and point out the optimization direction of Honeypot. By assessing the honeypot architecture, this paper summarizes 8 identification factors for honeypot. After evaluating the identification accuracy and concealment of these factors, this paper proposes an anti-honeypot scheme on the Internet and a concrete method based on packet-sharding. The effectiveness of our proposed method is validated through case study of 3 detection experiments of Conpot (a typical honeypot for industrial control system) on the Internet. 2432 Conpot honeypots are found and their distribution is also derived.

Key words honeypot identification; packet sharding; honeypot

1 引言

随着工业化和信息化的深度融合, 工业控制系统网络安全事件频发, 2010年6月曝光的震网病毒[1]是专门定向攻击工业控制系统的病毒, 摧毁了伊朗大量工业离心机设备。从国家互联网应急中心[2]发布的《2017 年中国互联网网络安全报告》可知, 自2013 年以来, CNVD[3]收录的安全漏洞数量年平均增长率为 21.6%。2017 年较 2016 年的安全漏洞数量增长了 47.4%。面对高级持续性威胁(Advanced Persistent Threats, APT)和不断增长的漏洞威胁, 传统的被动安全防御手段已难以保证充分检测出工控攻击。

相比于入侵检测、防火墙等被动防护手段, 蜜罐是一种安全威胁的主动检测技术[4], 它通过模拟一

通讯作者: 吕世超, 博士, 助研, Email: lvshichao@iie.ac.cn。
本课题得到国家重点研发计划(No.2016YFB0800202), 国家自然科学基金重点项目(No.U1766215), 中国科学院战略性先导科技专项课题(No.XDC02020500), 中国科学院信息工程研究所国际合作项目(No.Y7Z0461104)资助。
收稿日期: 2019-02-16; 修改日期: 2019-05-10; 定稿日期: 2019-05-13

个或多个易受攻击的主机来吸引攻击者, 捕获攻击者的流量和样本, 发现网络威胁和提取威胁特征, 其价值在于被探测、攻击和损害[5]。由于蜜罐并没有向外界提供任何真实有价值的服务, 因此所有与蜜罐的交互都是可疑的。蜜罐采集的是高质量的攻击信息。此外, 蜜罐还能够拖延攻击者对真正目标的攻击, 让攻击者长时间停留在蜜罐系统中, 间接地保护真实业务系统。

蜜罐诱捕是一个与攻击者对抗的攻防博弈过程。随着蜜罐技术的不断发展成熟, 蜜罐识别的能力也不断提高。从攻击者的角度来研究蜜罐识别技术, 有助于安全研究人员不断完善蜜罐技术。通过相关文献整理和实验研究, 本文的主要贡献是:

1) 在分析现有蜜罐识别方法的基础上, 归纳了 8 种蜜罐识别要素, 分析了不同要素的准确性和隐蔽性, 并提出了一种互联网中的蜜罐识别流程。

2) 分析了 Conpot 蜜罐架构的固有缺陷, 提出了一种基于数据包分片的工控蜜罐识别方法。

3) 进行了互联网范围的 Conpot 蜜罐识别, 共发现蜜罐 2432 个, 并验证了识别方法的有效性。

本文的组织结构如下: 第 2 节简述蜜罐系统的基本工作原理和相关蜜罐工具; 第 3 节总结蜜罐的识别要素及相关工作进展, 基于准确性和隐蔽性分析提出了一种互联网蜜罐识别流程; 第 4 节根据 Conpot 蜜罐架构的固有缺陷, 提出了一种基于数据包分片的工控蜜罐识别方法; 第 5 节进行互联网 Conpot 蜜罐识别与分析; 第 6 节总结全文并展望未来蜜罐发展。

2 蜜罐系统与工具

蜜罐的概念起源于在电脑安全专家 Cliff Stoll 1990 年出版的小说《The Cuckoo’s Egg》[6], 小说描述了主人公如何通过部署一系列虚假数字文件来追踪网络黑客的故事。经历近三十年的发展, 蜜罐技术已经逐步成熟。本节将主要介绍蜜罐系统的结构和工作机理, 为蜜罐识别研究提供理论基础。

2.1 蜜罐系统

如图 1 所示, 蜜罐系统的设计原理可归纳为“两部分三模块”, “两部分”指面向攻击者设计的攻击者可见部分和面向研究人员设计的攻击者不可见部分, “三模块”指交互仿真、数据捕获和安全控制三个模块。交互仿真模块属于攻击者可见的部分, 数据捕获和安全控制模块属于攻击者不可见部分。交互仿真模块通过在网络中暴露自身的虚假服务或资源, 诱导攻击者进行网络探测、漏洞利用等恶意行为。数据捕获模块则通过对网络、系统和应用业务等方面的监测, 捕获网络连接记录、原始数据包、系统行为数据、恶意代码样本等高价值的威胁数据。安全控制模块通过阻断、隔离和转移攻击等手段, 确保蜜罐系统不被攻击方恶意利用, 防止引发蜜罐系统对外发起的恶意攻击。

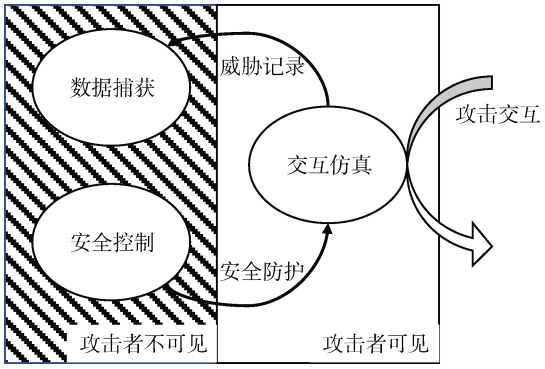

图 1 蜜罐系统“两部分三模块”的设计原理
Figure 1 The design principle of honeypot by “Two parts and three modules”

2.2 蜜罐工具

随着蜜罐技术的不断成熟, 针对“两部分三模块”的不同需求已经衍生出了各种蜜罐工具软件, 为蜜罐的快速部署和数据采集提供了保障, 包括虚拟网络拓扑、服务仿真、威胁捕获、安全控制、数据可视化分析等。下面结合几种常用的蜜罐工具进行简要介绍。

1) Honeyd[7]是 2004 年由密歇根大学 Provos 开发的一种通用的蜜罐开发工具, 主要用于构建虚拟网络和简单的仿真服务。它是一个小型守护进程, 可在单个主机上同时模拟数千个虚拟主机和 TCP/IP 堆栈, 并能够欺骗 Nmap 和 Xprobe 等操作系统指纹识别工具。Honeyd 可灵活配置任意开放端口上的服务, 并通过位于/var/run/honeyd.sock 的 UNIX 套接字, 实现自定义命令脚本与它的内部通信。

2) Conpot[8]是蜜网项目组旗下的一款工控蜜罐开发工具。通过不同工控协议的组合和特定工控负载响应时间的调节, 实现了对西门子 S7 系列 PLC、施耐德 Modicon PLC、Guardian AST 油箱监控器的仿真, 支持与人机界面(HMI)或真实控制设备互通互联, 并可利用容器技术在各类设备中快速部署。

3) tcpdump[9]是 Linux 中常用的一款网络数据采集分析工具, 支持对网络层、协议层、IP 和端口等的过滤和分析。由于其网络流量采集的强大功能, 常被部署于蜜罐中监控网络活动, 分析攻击者在交互中的所有通信细节。

4) p0f[10]是一款被动操作系统指纹识别工具, 通过捕获并分析目标主机发出的数据包来识别操作系统信息, 包括系统版本、软件版本和网络拓扑等信息。p0f 仅根据网络数据包即可快速识别操作系统信息, 为安全人员溯源分析提供了新的情报。

5) Sebek[11]是系统级的行为监控工具, 其组成包括客户端和服务器两个部分。客户端完全运行在蜜罐系统的内核空间中, 以可加载内核模块的方式执行, 用于记录攻击者进行系统调用时的所有数据, 并将数据隐蔽地发送给服务器。服务器则利用统一的标准格式收集所有客户端发送的数据, 并把数据记录到数据库中。

6)Kippo/Cowrie。Kippo[12]是一款基于 Python 并支持多操作系统的 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, 通常用于捕获 SSH 口令爆破、自动化脚本攻击、僵尸网络病毒等攻击活动。Cowrie[13]是 Kippo 的继承者, 扩展了对 SCP、SFTP 和 Telnet 协议的支持, 同时实现 SSH 和 Telnet 的登陆交互和命令执行, 诱导攻击者远程登录、渗透和下载恶意样本。

3 蜜罐识别

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, 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。一旦攻击者识别出蜜罐, 蜜罐将失去意义。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间的差异。

蜜罐识别与蜜罐构建是攻防博弈的两面。蜜罐对攻击者的欺骗遵循“木桶效应”, 攻击者可以利用任意蜜罐脆弱点进行蜜罐识别, 任何与真实设备的差异都可能成为蜜罐识别的特征。目前蜜罐研究没有通用的识别流程和识别理论指导, 并缺乏以攻击者角度对蜜罐系统识别要素的分析。因此, 本文在分析蜜罐系统设计原理的基础上, 提取了 8 种蜜罐识别要素, 总结了互联网中蜜罐识别的流程。

3.1 蜜罐的识别要素

由第 2.1 节可知, 在蜜罐系统设计原理中, 交互仿真对攻击者而言是一种显式的处理过程, 通过各种仿真手段去逼近真实设备的交互能力。而数据捕获和安全控制是隐式的处理过程, 通过隔离或混淆的方式隐藏各阶段对数据流的处理痕迹, 尽可能减少额外操作行为所导致的交互行为差异。因此, 如图 2 所示, 蜜罐的识别可分为基于交互仿真差异的识别和基于隐匿行为发现的识别两个方向。

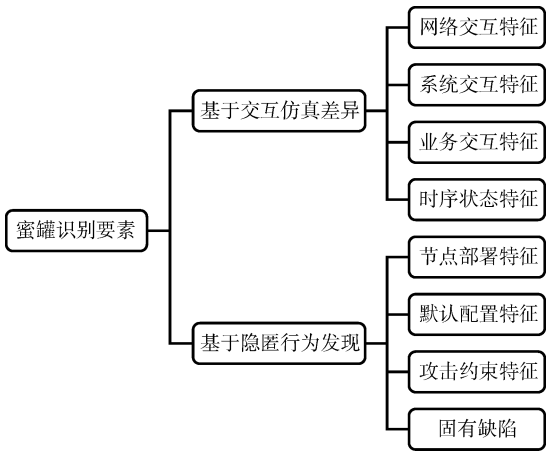

图 2 蜜罐的识别要素

Figure 2 The identification elements of honeypot

基于交互仿真差异的识别是攻击者与蜜罐设计者针对目标系统研究的正面对抗, 攻击者可以通过发现蜜罐未实现的功能来识别蜜罐, 主要包括协议交互、系统交互、业务交互和时序状态响应方面的特征。

1) 协议交互特征。该特征主要是针对通信协议的研究, 从协议实现的完整性角度鉴别设备是否为蜜罐。通过向目标发送其可能未实现的真实请求报文, 根据设备的响应信息来判断设备协议的完备性。由于数据请求并不能完全被解析, 蜜罐系统返回的响应差异将成为其识别特征。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[14], 采用协议逆向和分析方法判别协议差异。此外, 大量网络工具和平台能够支持对协议交互能力的检测, 也属于协议交互特征。网络探测工具(如 Nmap[15]和 Zmap[16])凭借其强大的探测能力, 可用于分析操作系统、端口信息等网络特征。Censys.io[17]面向互联网每周进行常规端口扫描, 抓取所有互联网节点对外接口信息。Shodan 提供了一个在线工具[18], 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 等通过将网络扫描与互联网公开的感知数据进行关联[19], 实现对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[20]进行了互联网范围的工业控制设备发现和识别, 采用了包括开放端口数量和 HTTP 配置等四个特征, 并通过训练概率模型和启发式算法来识别工控蜜罐并剔除干扰。

2) 系统交互特征。该特征一般用于识别主机系

统执行环境, 如虚拟机操作环境识别等。2014 年 SANS 技术研究所[21]的报告指出攻击者可使用系统命令“file /sbin /init”的动态返回值识别 Kippo 蜜罐。2015 年, 密歇根大学 Chen 等[22]通过对 6900 个不同来源的恶意软件样本进行分析, 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[23]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[24]剖析了 Chen 和 Gajrani J 工作, 并总结出了时间、操作、硬件和环境四类蜜罐指纹。

3) 业务交互特征。随着计算机技术的不断发展, 常规的计算设备承载着各种各样复杂的业务逻辑, 对蜜罐交互仿真带来了巨大挑战。特别是在具备物理空间操作的物联网业务中, 蜜罐如何正确处理物联网物与物、物与环境、物与人之间的信息交互显得尤为重要。各类传感器 (如红外、超声、温度、湿度、速度等)、图像捕捉装置(摄像头)、全球定位系统(GPS)、激光扫描仪等基础服务也提高了业务交互仿真的难度。常规蜜罐技术在设计业务逻辑时通常难以顾及物理环境与逻辑信息间的映射, 因此业务逻辑的不匹配易成为此类蜜罐的识别特征。蜜罐系统对设备状态信息的获取都是事先设定好或是随机产生的, 但是实际环境中的数值会在一定范围内呈现正态分布且具有一定的规律性, 尤其是线圈状态、寄存器状态、文件记录、温度、阀门状态等信息会随着系统当前工作的不同而改变。多数蜜罐在实现过程中, 状态信息都是固定不变或随机生成的, 这并不符合实际的业务逻辑。以 Modbus 为例, 通过频繁请求各线圈状态, 通过判断线圈状态变化是否复合实际规律可以鉴别蜜罐。2016 年 S. Litchfield 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[25], 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。

4) 时序状态特征。蜜罐系统大多实现静态的交互仿真, 并不真实地执行业务请求, 缺乏对目标系统执行状态变化的模拟, 如协议状态机、业务模型的状态转移等。蜜罐系统对于独立的网络、系统和业务交互的响应准确度普遍较高, 而对于有时序关联的交互缺乏完整的实现。基于时序状态特征的识别方法对真实设备可能会造成难以预知的破坏, 因此这类识别要素很难应用于现实的识别工作中。

基于隐匿行为发现的识别是通过对蜜罐系统威胁捕获、安全控制等非交互处理过程中的行为特征的提取来识别蜜罐系统。

1) 节点部署特征是一种静态特征, 主要由蜜罐系统在互联网部署运行过程中与真实设备存在的运行差异造成。该特征包括网络节点的 IP、端口等信息。IP 信息方面, 可以衍生出地理位置、ISP (Internet Service Provider,互联网服务提供商)信息和其他威胁情报信息。地理位置信息可通过 IP 定位服务获得, 能够评估目标系统所在的地理环境。ISP 信息通常能够反映出特定的组织集团, 如云服务器提供商、数据中心、网络公司等。特别是在全球 IPV4 地址稀缺的环境下, 为蜜罐系统单独提供 ISP 信息的伪装并不实际。除 ISP 信息以外, 还可以通过 IP 获取其他的威胁情报信息, 如域名变更历史、滥用记录等。端口信息方面, 通常包括端口开放情况和端口服务内容。通过将常见蜜罐默认端口的组合情况与真实设备常见的端口组合对比, 可以一定程度上识别蜜罐。此外, 多个蜜罐系统常以软件或虚拟机形态运行在同一个物理机器上, 系统资源上存在较大限制。当系统资源不足时, 将会发生资源争夺, 产生竞争现象。而对于真实设备而言, 往往是独立使用所有业务资源, 不会表现出资源争夺现象。因此, 若加重某种系统负载时, 通过比较响应速度的偏差值, 也可以识别出蜜罐系统。

2) 默认配置特征。为了节省时间和成本, 安全从业者往往采用各种成熟的蜜罐工具进行蜜罐构建。在这种前提下, 不仅考验研究者对目标设备的交互仿真能力, 同时还考验其对蜜罐工具运用的掌握程度。通常情况下, 蜜罐工具为了方便使用会为用户提供默认的配置和启动参数, 但用户实际的部署使用可能并未完全理解所有配置便直接部署上线, 从而导致遗留大量的蜜罐工具特征。Honeyd 蜜罐工具接收的数据包首先被中央包分配器处理, 它首先检查数据包的长度和校验和。在处理包之前, 分配器必须查询配置数据库来找到一个与目的 IP 相符合的蜜罐配置, 如果没有指定的配置存在, 将使用默认的模板。通过对默认应答的检查, 可以方便地识别蜜罐工具。Conpot 的初始模板文件中也存在默认配置, 在其模拟的 S7 系列设备中, 使用默认的模块序列码。攻击者可以根据特定的型号特征, 将交互对象识别为蜜罐。

3) 攻击约束特征。通常情况下, 基于伦理和法律的要求, 蜜罐所有者已知蜜罐中对外攻击发生而不阻止, 需要承担相关的责任。因此, 蜜罐通常交互仿真能力并不高, 也是为了防止攻击者利用蜜罐发

起对外攻击。这也是一种较为有效的蜜罐识别方法。2004 年 Krawetz 开发了一种名为 Honeypot Hunter[26] 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[27]。

4) 固有缺陷。固有缺陷可以认为是蜜罐框架上具备硬编码特性的底层行为特征, 如网络连接方式、时延特征等。各类蜜罐系统需要在正常的交互仿真处理中插入额外处理过程, 以采集和控制攻击数据, 导致蜜罐系统与真实系统必定是架构不同的, 必定存在运行机理上的固有差异。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[28]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[29]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[30]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。尤其对物联网设备而言, 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, 蜜罐很容易被攻击者识别。

3.2 识别要素的准确性

蜜罐识别要素的准确性指采用特定识别要素进行蜜罐识别的可靠程度。计算机技术发展半个多世纪以来, 各类计算设备、网络基础设施不断完善, 在网络中采用不同蜜罐识别要素的过程自然受到不同程度的环境因素干扰。

如表 1 所示, 定性地对比了蜜罐识别要素的准确性。由于网络安全工具类型多样, 真实业务场景也有攻击约束的需求, 通常无法直接断定攻击约束源于蜜罐的限制, 所以基于攻击约束特征的识别方法准确率较低。节点部署特征通常反映地理环境或网络运营商信息, 仅能根据真实业务部署的相关性来定性判断, 因此准确率一般较低。此外, 协议交互的完备性也受其他安全工具的影响, 未响应的协议交互不一定是蜜罐。默认配置特征由于无法识别该配置的真实设备, 识别准确率定性为中级别。其他四类特征往往提取了蜜罐系统与真实设备的系统级特征, 识别准确率较高。

表 1 识别要素的准确性对比

Table 1 Contrast of the accuracy of identification elements

准确性	蜜罐识别要素
低	攻击约束特征
	节点部署特征
中	协议交互特征
	默认配置特征
	固有缺陷
高	系统交互特征
	业务交互特征
	时序状态特征

3.3 识别要素的隐蔽性

隐蔽性指蜜罐识别要素的识别动作对蜜罐系统或真实设备的影响和危害级别, 识别动作影响范围越大或危害级别越高, 隐蔽性越低。因此, 蜜罐识别要素也可以分为交互和非交互两大特征, 非交互特征即不需要与目标系统进行通信的识别方法, 交互特征即需要通过发送测试报文并验证响应的蜜罐识别方法。

表 2 定性地对比了蜜罐识别要素的隐蔽性。节点部署特征通常采用第三方的威胁情报, 并未与蜜罐系统进行直接交互, 因此这类特征具有最高的隐蔽性。而固有缺陷特征, 就目前的研究情况而言, 多数都是对正常业务低干扰的识别方式, 通过极少的正常交互识别蜜罐, 也具备较高的隐蔽性。此外, 协议交互特征和默认配置特征的交互深度一般并不高, 属于中等的隐蔽性。其他五种蜜罐识别要素, 通常已深入蜜罐系统严格监视环境中, 对系统运行造成了深度的影响, 属于低隐蔽性。

表 2 蜜罐识别要素的隐蔽性对比

Table 2 The comparison of the concealment of identification elements

隐蔽性	蜜罐识别要素
低	系统交互特征
	业务交互特征
	攻击约束特征
	时序状态特征
中	默认配置特征
	协议交互特征
高	固有缺陷
	节点部署特征

3.4 互联网中的蜜罐识别流程

互联网是蜜罐主要部署地点, 理论上的蜜罐识别方法在实际互联网识别过程中增加了诸多限制, 如网络的不稳定性、探测攻击伦理要求等。而目前研究主要是在高度可靠网络连接和无攻击限制的条件下进行的[14, 21-23, 25], 尚未有针对互联网中蜜罐识别理论的体系研究。因此, 综合 2.2 节和 2.3 节的蜜罐识别要素优先级的分析, 本文提出了一种互联网中的蜜罐识别流程。

如图 3 所示, 互联网中的蜜罐识别流程主要分为下面三个步骤:

1) 蜜罐部署特征识别。通过查询目标 IP 相关的地理位置、ISP 信息、域名反查信息和威胁情报信息。地理位置通常可以反映目标的业务特点, ISP 和域名反查信息则反馈目标的组织信息, 而类似 Shodan、Threatbook[31]等网络安全情报提供者会对提供的 IP 进行综合分析、历史滥用记录查询、网络功能分析, 甚至是蜜罐可能性分析。因为内网 IP 无法查询到上述信息, 且用于特定保护功能的蜜罐往往与产品系统临近部署, 所以这种识别对内网蜜罐和企业内部蜜罐几乎不起作用。

2) 蜜罐默认配置识别。默认配置识别是针对已知的蜜罐工具识别方法进行筛选分析, 对相关已知的蜜罐特征进行“黑名单”筛选, 确定目标系统的蜜罐可能性。

3) 蜜罐固有缺陷挖掘。除了仿真交互的直接对抗方式, 蜜罐固有缺陷挖掘更加强调对蜜罐系统架构级的识别。通过 TCP/IP 特征分析、操作系统识别、物理地址识别、时延特征等基础分析测试, 挖掘蜜罐系统的架构级特征。

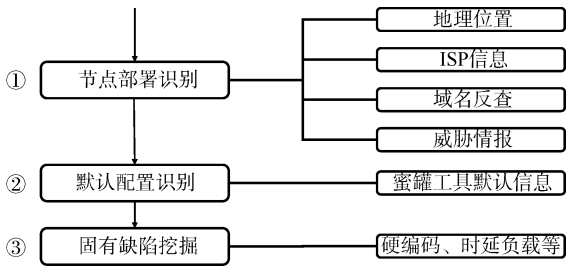

图 3 一种互联网中的蜜罐识别流程

Figure 3 Process of honeypot identification on the Internet

4 基于数据包分片的工控蜜罐识别方法

由第 3 节的分析可知, 利用蜜罐的固有缺陷进行识别具备较高的准确性和隐蔽性。本节根据工控蜜罐接收分片的请求数据时会产生响应异常这一特征, 提出了一种基于数据包分片的工控蜜罐识别方法, 并以 Conpot 为例进行了三次互联网识别实验。

由于工控协议的私有化特性, 工控蜜罐的交互仿真主要以开源的协议解析为基础实现, 如 Modbus-tk、Pymodbus 等。由于这些程序库是应用层进行的非完整协议处理, 与专用实时系统处理在兼容性和连贯性上有很大的差别, 成为工控蜜罐的固有缺陷。本节中将以 Conpot 为例, 分析其工控协议处理过程的固有缺陷, 并以此构建蜜罐识别方案。

Conpot 中常用的服务有 Modbus、S7comm 和 IEC104。Modbus 协议[32]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(PLC)发布的公开通信协议。S7comm 协议[33]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。IEC104 规约是一个广泛应用于电力、城市轨道交通等行业的国际标准。

4.1 框架级数据包分片方法

在 Conpot 仿真过程中, 为了避免一个连接长时间占用带宽, 设置了连接最长等待时延为 5 秒。在无通信状态下, 其保持连接的时长是 5 秒, 5 秒后由 Conpot 发送 fin_ack 请求断开连接。而真实设备的连接处理更加完善, 能够长时间保持连接的有效性和稳定性。因此, 将正常的请求报文进行包内分片, 并间隔 10 秒进行分次发送, 判别识别对象的响应。若目标为 Conpot 蜜罐, 则 5 秒时连接自动断开, 第二段报文将得到异常响应。相比之下, 正常设备并不受数据包分片的影响, 因此该识别过程能够有效地避免对真实设备的危害。

4.2 基于 Modbus 协议的分片方法

在 Modbus 协议解析中, 为了实现请求校验和检测分析, Conpot 会分段接收数据以判断其有效性, 非合规数据将直接丢弃。如图 4 所示, 数据包分片的关键在于第 7 个字节, 若蜜罐系统接收少于 7 字节, 会判定为无效报文并结束会话。因此, 将正常的 Modbus 协议报文进行包内分片, 保证第一部分长度小于 7 个字节, 发送给目标系统, 根据响应的异常情况来识别蜜罐。

4.3 基于 S7comm 协议的分片方法

如图 5 所示, 在 Conpot 中 S7comm 协议的解析大致分为 TPKT、COTP 和 Trailer 三个层次。蜜罐经过应用层握手后, 获得正确的控制载荷, 交互属于阻塞模式, 若接收缓冲区没有数据, 系统将永久阻塞。但 Conpot 的 S7comm 解析函数中只要数据字节数小于请求字节数, 接收函数 recv 就会阻塞。因此,

Conpot 能够正确解析了 TPKT 头部, 但是当解析到 COTP 部分时, 其协议解析引擎和 Modbus 解析函数存在同样的缺陷。如图 5 所示, 首先明确 COTP 段的长度, 并对该段报文进行多次分片, 发送给目标系统, 通过响应异常来识别蜜罐。

4.4 基于 IEC104 协议的分片方法

Conpot 中的 IEC104 协议解析实现得非常简单, 仅实现时钟同步和总召唤两种请求功能, 不对协议规范做检测处理, 且不会断开已经建立的连接。因此, IEC104 协议解析过程在接收异常包时会处于长时间的等待状态, 没有时间上限, 可以通过等待时间长度识别该服务。但是, 为了保证互联网探测的安全和有效性, 本文实验中采用 Conpot 未实现的功能码进行探测识别。通过对 IEC104 协议规范的了解, 最终采用协议报文“680443000000”。该数据包可以用于测试连通性, 并且对真实设备业务不会产生影响。

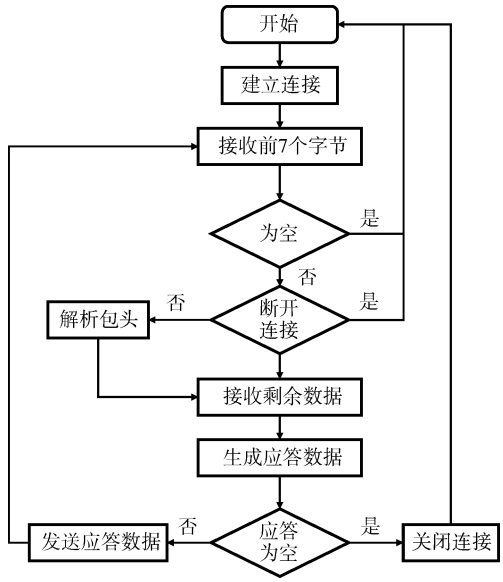

图 4 Conpot 中 Modbus 服务的运行逻辑

Figure 4 The operation of modbus service in Conpot

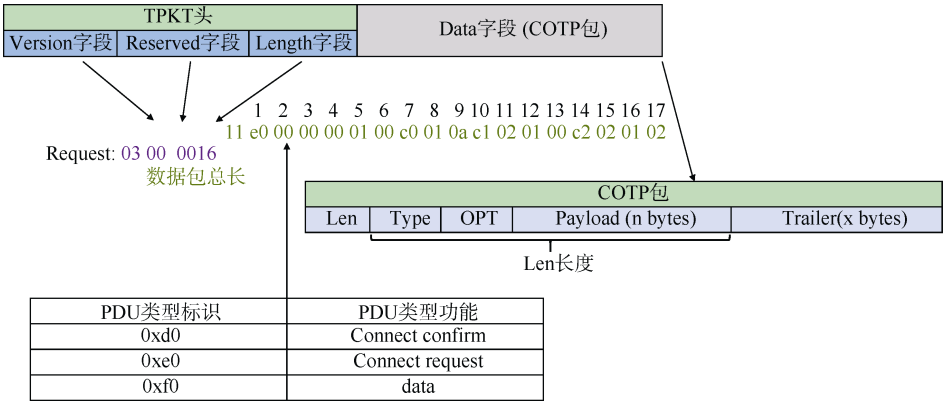

图 5 S7 协议解析

Figure 5 The analysis of S7 protocol

4.5 讨论

基于数据包分片的工控蜜罐识别方法是针对工控协议解析模块实现的固有缺陷而提出的, 协议模块是工控蜜罐的核心功能, 单纯修改配置参数不能够避免被识别, 需要完全修改源码中的协议处理过程部分。例如, 在现有的 Conpot 程序架构下, 改变 Modbus 协议处理的第 7 字节读取规则, 但同时也完全改变了源码中输入参数属性, 影响所有后续的参数处理过程, 相当于完全重写 Modbus 协议处理模块。此外, 若采用与真实设备相同的逐字节的读取和处理方式, 在设备未开源的前提下, 完整模拟这种处理方式的可能性非常小, 没有实际意义。

虽然 4.1 节提到的框架级的识别方法是一种没有协议制约的通用方法, 但应用于互联网中的蜜罐识别仍然存在连接可靠性不足的问题。因此, 在互联网中的 Conpot 蜜罐识别主要采用针对协议的识别方法, 并利用交叉验证的方式进一步核验了方法的准确率。

5 实验测试与结果分析

在进行蜜罐识别的测试前, 首先提取了 IPV4 空间中共计 8387 个标记开放 Modbus、S7comm 和 IEC104 服务的网络节点信息。本文总计进行了三次互联网规模的 Conpot 蜜罐识别, 统计结果如表 3 所示。

表 3 实验安排与识别结果

Table 3 The experimental arrangement and identification results

时间	IP 总数	Modbus 蜜罐	S7 蜜罐	IEC104 蜜罐	蜜罐总数
2018.12.3	*8387*	1381	1002	14	2397
2018.12.16	*8387*	1407	1017	24	2448
2019.1.8	*8387*	1400	1019	13	2432

5.1 实验过程

实验过程分析主要采用第三次互联网扫描的结果, 其中 2155 个开放 S7comm 服务、5870 个开放 Modbus 服务、362 个开放 IEC104 服务。如表 4 所示, 由于网络状态、带宽、安全防护等因素, 识别过程存在以下五种响应结果。

表 4 响应状态

Table 4 The response status

工控服务	数量	节点不存活	连接失败	拒绝交互	蜜罐	真实设备
Modbus	5870	2016	328	200	1400	1926
S7comm	2155	792	125	252	1019	17
IEC104	362	50	9	50	285	13

1) 节点不存活, 网络不通或 ICMP 服务关闭。

2) 连接失败, 节点存活但相关服务连接失败。

3) 拒绝交互, 节点存活但是相关服务请求被阻断。

4) 蜜罐, 判定节点为蜜罐。

5) 真实设备, 判定节点为真实设备。

对于真实设备而言, 往往处于安全防护环境中, 暴露在互联网的设备在连接过程中可能收到其他安全保护, 导致连接失败、拒绝交互等结果。保证网络连通性也是蜜罐捕获恶意流量的必要条件。因此, 节点不存活、连接失败和拒绝交互的节点有较大可能性是真实设备。

5.2 结果分析

如图 6 所示, 通过地理位置信息查询, 美国的 Conpot 蜜罐数量明显高于其他国家(603 个), 并且占该国暴露工控节点的 45.68%。除了美国, 另外还有四个国家 Conpot 蜜罐数量超过 100 个, 分别是德国(152 个)、意大利(130 个)、法国(101 个)、西班牙(114 个)、中国(131 个)、土耳其(128 个)。非洲地区几乎没有发现 Conpot 蜜罐, 主要原因是该地区网络设施并不发达, 暴露的工控节点数量较少。

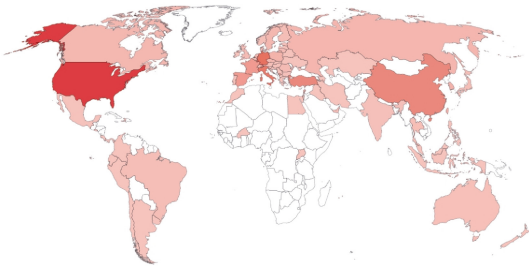

图 6 Conpot 蜜罐识别结果的国家分布

Figure 6 The country distribution of the Conpot identification results

如表 5 所示, 在开放工控节点数量前十的国家中, Conpot 蜜罐占比都大于或接近 20%, 蜜罐技术已成为一种研究互联工控攻击的普遍工具。

表 5 工控蜜罐与真实设备的占比

Table 5 The proportion of ICS honeypot and real devices

国家	工控节点	蜜罐数量	蜜罐占比(%)
美国	1320	603	45.68
德国	658	152	23.10
意大利	591	130	22.00
法国	495	101	20.40
西班牙	486	114	23.46
中国	445	131	29.44
土耳其	441	128	29.02
瑞典	294	55	18.71
韩国	276	77	27.90
加拿大	273	53	19.41

5.3 交叉验证

为了验证蜜罐识别方法的准确性, 我们采用 3.4 节中的通用识别流程人工标注了 1000 个互联网节点, 其类型分布如表 6 所示。

表 6 人工标注的网络节点类型

Table 6 The human-annotated node types

类型	数量/个
Conpot 蜜罐	300
真实工控节点	400
非工控节点	250
非活跃节点	50

通过对中高隐蔽性的蜜罐识别要素交叉验证, 计算并对比其精确率、召回率和 F1 值。如表 7 所示, 基于数据包分片的蜜罐识别方法在准确率和时间开销上都优于其他识别要素。

表 7 不同识别要素的评估

Table 7 The evaluation of different identification elements

识别要素	精确率(%)	召回率(%)	F-1 值	用时(min)
协议交互特征	0.87	0.46	0.602	147
默认配置特征	0.3	0.12	0.171	16
节点部署特征	0.85	0.36	0.506	77
数据包分片	0.98	0.95	0.964	39

6 结论与展望

作为网络安全中的一种主动防御技术, 蜜罐技

术已在众多信息领域中得到应用, 如企业内网、物联网、工控网等。在与攻击者的攻防博弈中, 蜜罐技术也在不断发展和演进, 而探索新兴的计算机技术与蜜罐的融合是未来发展的重要方向。在技术应用方面, 随着区块链研究迈向高潮, 安全研究社区正尝试采用蜜罐技术分析加密货币与网络智能合约等全新计算平台的安全威胁,如比特币[34-35]、以太坊[36-37]等。在架构方面, Luo 等[38]提出了“智能交互(intelligent-interaction)”蜜罐的理念, 通过机器学习技术实现蜜罐仿真的自动化和智能化, 从海量的物联网设备交互信息中自动筛选和构建攻击者所需的响应。机器学习方法主要用于增强蜜罐的动态响应能力, 通过训练学习实现不同状态下最佳响应的筛选。

蜜罐技术的发展催生了蜜罐识别技术, 而且蜜罐识别技术的研究同样促进蜜罐技术的发展。通过对 8 种蜜罐识别要素的分析, 本文提出了一种互联网中的蜜罐识别流程和一种基于数据包分片的蜜罐识别方法, 通过对请求数据包的分片, 验证蜜罐的响应过程中的流畅性差异, 从而准确识别蜜罐。以 Conpot 蜜罐为例进行了互联网范围识别实验, 指明了工控蜜罐协议解析的缺陷, 为蜜罐的改进和完善提供参考, 对蜜罐技术的发展具有重要意义。

参考文献

[1] R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon,” *IEEE Security & Privacy*, vol 9, no. 3, pp. 49-51, 2011.

[2] “CNCERT/CC,” National Internet Emergency Center, http://www.cert.org.cn/, 2018.

[3] “CNVD,” China national vulnerability database, http://www.cnvd.org.cn/

[4] G. Portokalidis, A. Slowinska, and H. Bos. “Argos: An Emulator for Fingerprinting Zero-Day Attacks for Advertised Honeypots with Automatic Signature Generation,” *Acm Sigops/eurosys European Conference on Computer Systems ACM* (EuroSys’06), 2006.

[5] L. Spitzner, “Honeypots: tracking hackers,” Hacker, 2003.

[6] C. Stoll, “The Cuckoo’s Egg: Tracking a Spy Through the Maze of Com-puter Espionage,” *Physics Today*, vol. 43, no. 8, pp. 75-76, 1989.

[7] N. Provos, “Honeyd: A Virtual Honeypot Daemon,” In Proc. of the *12th USENIX Security Symposium* (USENIX ‘03), pp. 1–7, 2003.

[8] “Conpot,” L. Rist, http://conpot.org/, May. 2013.

[9] “tcpdump,” Wikipedia, https://en.wikipedia.org/wiki/Tcpdump, Dec. 2018.

[10] “p0f v3(version 3.09b),” Michal Zalewski, http://lcamtuf.coredump.cx/p0f3/, 2014.

[11] “Sebek,” L. Spitzner, http://www.honeynet.org/project/sebek/, 2008.

[12] “Kippo-SSH Honeypot,” disaster, http://code.google.com/p/kippo/, 2011.

[13] “Cowrie – active kippo fork,” M. Oosterhof, http://www.micheloosterhof.com/cowrie/, July. 2015.

[14] P. M. Comparetti, G. Wondracek, C. Kruegel and E. Kirda, “Prospex: Protocol specification extraction,” In Proc. of the *30th IEEE Symposium on Security and Privacy* (S&P ‘09), pp. 110–125, 2009.

[15] G. F. LYON, “Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning,” *Insecure*, USA, 2009.

[16] Z. Durumeric, E. Wustrow and J. A. Halderman, “ZMap: Fast Internet-wide Scanning and Its Security Applications,” In Proc. of the *22nd USENIX Security Symposium* (USENIX ‘13), pp. 605–619, 2013.

[17] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey and J. A. Halderman, “A Search Engine Backed by InternetWide Scanning,” In Proc. of the *22nd ACM SIGSAC Conference on Computer and Communications Security* (CCS ‘15), pp. 542–553, 2015.

[18] “Honeypot Or Not?” Shodan, https://honeyscore.shodan.io/, 2017

[19] J. Bethencourt, J. Franklin and M. Vernon, “Mapping Internet Sensors with Probe Response Attacks,” In Proc. of the *14th USENIX Security Symposium* (USENIX’05), pp. 193–208, 2005.

[20] X. Feng, Q. Li and H. Wang, “Characterizing Industrial Control System Devices on the Internet,” In Proc. of the *24th IEEE International Conference on Network Protocols* (ICNP’16), pp. 1–10, 2016.

[21] “Kippo Users Beware: Another Fingerprinting Trick,” SANS TECHNOLOGY INSTITUTE, https://isc.sans.edu/forums/diary/Kippo+Users+Beware+Another+fingerprinting+trick/18119/, May. 2014.

[22] X. Chen, J. Andersen, Z. Morley, M. Michael and B. J. Nazario, “Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware,” *IEEE/IFIP International Conference on Dependable Systems & Networks* (DSN’15), 2015.

[23] J. Gajrani, J. Sarswat, M. Tripathi, V. Laxmi, M.S. Gaur, M. Conti, “A robust dynamic analysis system preventing sandbox detection by android malware,” In Proceedings of the *8th ACM International Conference on Security of Information and Networks*, pp. 290–295, 2015.

[24] J. Uitto, S. Rauti, S. Laurén and V. Leppänen, “A Survey on Anti-honeypot and Anti-introspection Methods,” *World Conference on Information Systems & Technologies. Springer*, 2017.

[25] S. Litchfield, D. Formby, J. Rogers, S. Meliopoulos and R. Beyah,

"Rethinking the Honeypot for Cyber-Physical Systems," *IEEE Internet Computing*, vol. 20, no. 5, pp. 9-17, 2016.

[26] N. Krawetz, "Anti-honeypot Technology," *IEEE Security & Privacy Magazine* vol. 2, no. 1, pp. 76–79, 2004.

[27] C. Zou and R. Cunningham, "Honeypot-Aware Advanced Botnet Construction and Maintenance," In Proc. of the *36th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (DSN '06) (2006), pp. 199–208, 2006.

[28] T. Holz and F. Raynal, "Detecting Honeypots and Other Suspicious Environments," In Proc. of the *6th Annual IEEE System, Man and Cybernetics Information Assurance Workshop* (SMC'05), pp. 29–36, 2005.

[29] X. Fu, W. Yu, D. Cheng, X. Tan, K. Streff and S. Graham, "On Recognizing Virtual Honeypots and Countermeasures," In Proc. of the 2nd IEEE International Symposium on Dependable, *Autonomic and Secure Computing* (DASC '06), pp. 211–218, 2006.

[30] S. Mukkamala, K. Yendrapalli, R. Basnet and M. K. Shankarapani, "Detection of Virtual Environments and Low Interaction Honeypots," *Information Assurance & Security Workshop*, IEEE, 2007.

[31] "Threatbook," ThreatBook. CN, https://x.threatbook.cn/ip/, 2018.

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

[33] "S7 Communication (S7comm)," Wireshark, https://wiki.wireshark.org/S7comm, May, 2016.

[34] T. Curran, D. Geist. Using the Bitcoin Blockchain as a Botnet Resilience Mechanism. 2016.

[35] S. Li, S. Wu. Your Device and Your Power, My Bitcoin, International Conference on Blockchain. Springer, Cham, 2018: 285-292.

[36] C. F. Torres, M. Steichen. The Art of The Scam: Demystifying Honeypots in Ethereum Smart Contracts. arXiv preprint arXiv:1902.06976, 2019.

[37] Z. Cheng, X. Hou, R. Li, Y. Zhou, X. Luo, J. Li and K. Ren. Towards a First Step to Understand the Cryptocurrency Stealing Attack on Ethereum. arXiv preprint arXiv:1904.01981, 2019.

[38] T. Luo, Z. Xu, X. Jin, Y. Jia and X. Ouyang. Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices. Black Hat, 2017.

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

张悦阳 于 2017 年在太原理工大学物联网工程专业获得学士学位。现在中国科学院大学计算机技术专业攻读硕士学位。研究领域为物联网安全、态势感知。研究兴趣包括: 自然语言处理、深度学习。Email: zhangyueyang@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助理研究员。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

陈新 于 2016 年在郑州大学控制工程专业获得硕士学位。现任中国科学院信息工程研究所工程师。研究领域为工控安全。研究兴趣包括: 工控网络入侵检测。Email: chenxin1990@iie.ac.cn

尹丽波 现任国家工业信息安全发展中心主任。研究领域为网络与信息安全理论与技术研究。研究兴趣包括: 信息安全标准、计算机内容安全、信息安全战略。

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