物联网安全威胁与安全模型

郑尧文[1], 文 辉[1], 程 凯[1,2], 李 红[1], 朱红松[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 随着物联网应用的发展和普及利用, 针对物联网的攻击事件日益增多且危害严重。目前面对物联网安全问题主要采用被动补救的方式, 缺乏对物联网安全的体系化思考和研究。本论文首先介绍物联网系统架构和各实体的发展, 然后分析物联网面临的多层次安全威胁, 包括各实体自身的安全威胁, 也包括跨域的安全威胁。其中, 实体自身安全威胁涉及到云平台、设备端、管道、云端交互。物联网跨域安全威胁包含 4 个方面: 多域级联攻击、物理域的冲突与叠加、信息域对物理域进行非预期的控制、信息域对物理域输入的理解不全面。在此基础上, 论文研究了基于 PDRR 网络安全体系的物联网安全模型,包含安全防护、安全检测、响应、恢复 4 个维度。安全防护包含认证、授权与访问控制、通信加密等技术, 需要考虑物联网种类繁多, 规模巨大, 异构等特点进行设计与实施。安全检测需要对各实体进行入侵检测、在线安全监测、脆弱性检测以及恶意代码检测。其中, 在线安全监测获取系统内部设备、应用程序的行为、状态、是否存在已知脆弱性等。脆弱性检测偏向于对未知脆弱性进行深度挖掘。在响应阶段, 除了配合相关部门机关完成安全行动资源配置、态势感知等响应工作外, 还需要进行入侵事件的分析与响应, 漏洞与恶意代码的公告与修复, 以及安全防护加固与检测规则的更新。在恢复阶段, 需要对关键数据进行恢复, 并对系统进行升级与恢复。最后论文进行总结并提出值得关注的研究方向。

关键词 物联网安全; 安全威胁; 安全模型; PDRR 模型
中图法分类号 TP309.1 DOI 号 10.19363/J.cnki.cn10-1380/tn.2023.09.06

IoT Security Threat and Security Model

ZHENG Yaowen[1], WEN Hui[1], CHENG Kai[1,2], LI Hong[1], ZHU Hongsong[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 With the development and widespread use of Internet of Things (IoT) applications, attacks on the IoT are becoming more frequent and more serious. Currently, the main approach to address IoT security issues is passive remediation, lacking systematic thinking and research on IoT security. Therefore, in this paper, we first introduce the IoT system architecture and the development of various entities. Then, we analyze the multi-level security threats faced by the IoT system, including the security threats of the entities themselves and cross-domain security threats. The security threats of the entities themselves involve cloud platforms, device ends, pipelines, and cloud-end interactions. The cross-domain security threats of the IoT include four aspects: multi-domain cascading attacks, conflicts and overlaps in physical domains, unexpected control of physical domains by cyber domains, and incomplete understanding of physical domains by cyber domains. Based on it, we present an IoT security model based on PDRR network security framework, including four dimensions: security protection, security detection, response, and recovery. Security protection includes technologies such as authentication, authorization and access control, and communication encryption. It requires considering the wide variety, large scale, and heterogeneity of the IoT in design and implementation. Security detection requires intrusion detection, online security monitoring, vulnerability detection, and malicious code detection for each entity. Online security monitoring obtains the behavior, status, and known vulnerabilities of internal devices and applications, while vulnerability detection focuses on in-depth exploration of unknown vulnerabilities. In the response phase, in addition to coordinating with relevant departments to complete security action resource allocation and situational awareness response work, it is also necessary to analyze and respond to intrusion events, announce and fix vulnerabilities and malicious code, and update security protection and detection rules. In the recovery phase, it is necessary to recover critical data and upgrade and restore the system. Finally, we summarize the paper and present some research directions worth attention.

Key words IoT security; security threat; security model; PDRR model

通讯作者: 文辉, 博士, 助理研究员, Email: wenhui@iie.ac.cn。
本课题得到广东省重点研发计划(No. 2019B010137004), 国家自然科学基金联合基金项目(No. U1766215)资助。
收稿日期: 2020-04-01; 修改日期: 2020-07-08; 定稿日期: 2022-12-23

1 引言

随着智慧城市、智能驾驶、智能家居、共享单车等物联网应用的发展, 针对物联网系统的攻击也日益增多。虽然物联网应用发展较快, 但设计者和开发者在实现物联网应用的过程中, 更关注于功能实现与使用的便捷性, 缺少对物联网系统中安全功能的设计与实现。在这样的应用环境下, 不仅物联网的系统、软件、设备暴露出很多安全问题, 同时, 这些安全问题还会反过来影响网络空间的安全。例如, 2016 年 Mirai 僵尸网络引起的攻击则是典型的影响网络空间安全的事件。由于数千万的物联网设备(网络摄像头等)在使用的过程中, 没有进行安全地运维, 仍然使用默认口令或弱口令, 导致攻击者可以轻易侵入这些设备, 组成僵尸网络, 对网络服务提供商发起分布式拒绝服务攻击(DDoS Attack), 导致大规模网络瘫痪。

针对物联网中的安全问题, 目前系统与设备的厂商仍然采用被动补救的方式进行解决。若发现物联网系统与软件中存在安全缺陷, 则厂商在特定系统组件上施加安全措施(打补丁修复, 添加安全模块等)。若物联网设备已经交付于用户使用, 则厂商发布公告让用户进行系统升级或者安全配置。总体来说, 面对物联网安全问题仍缺乏体系化的思考与研究, 导致各类安全问题层出不穷。

物联网系统由云、管、端、边缘实体构成。云管端的架构最初由华为在 2010 年提出。其中云指的是新一代数据中心和业务平台, 用于处理海量数据。端指的是智能化设备。管指的是介于云与端之间的网络基础设施, 用于传输云与端之间的海量数据。由于物联网的端可能存在计算和存储能力不足的问题, 物联网系统引入边缘实体来完成节点的计算任务。由于物联网是由云、管、端、边缘多层结构组成的新型网络系统, 相比于传统网络系统, 其面对的安全问题是不一样的。因此, 本论文将分析该物联网新型架构下的安全威胁。首先, 论文分析云、管、端、边缘各个实体内部存在的安全威胁。接着, 论文分析实体在交互过程中存在的安全威胁。在物联网系统中, 由于管道是云与端通信的“桥梁”, 边缘是端的“缓冲”, 云和端是真正进行“交流”的两个实体, 因此论文将重点分析云和端实体交互过程中存在的安全威胁。另外, 物联网系统本身属于信息域, 与物理世界紧密耦合。因此, 论文会分析物理域到信息域以及信息域到物理域的跨域威胁。

针对物联网系统存在的安全威胁, 本论文进一步提出相应的系统安全模型。在网络安全体系建立的过程中, 美国国防部提出 PDRR 模型[1]。该模型由防护(Protection)、检测(Detection)、响应(Response)、恢复(Recovery) 4 个阶段构成, 一方面实现系统全面的防御能力, 另一方面强调安全事件发生后系统的响应和恢复能力, 因而被安全技术团队沿用至今。针对物联网系统安全模型的建立, 本论文沿用 PDRR 模型, 根据物联网系统的特点和安全威胁, 分析出各阶段所需的安全措施。

论文的结构安排如下: 第二章论述物联网系统架构, 以及云、管、端、边缘等实体的功能和发展现状; 第三章介绍物联网信息域的安全威胁; 第四章将介绍物联网跨域的安全威胁; 第五章将介绍物联网的安全模型, 分别从防护、检测、响应、恢复展开; 第六章将总结全文并展望。

2 物联网系统架构

图 1 显示了物联网系统架构, 包含云、管、端、边缘等实体。其中, 云平台是云计算在物联网中的应用, 完成用户与业务数据的存储和处理。端分为传感器和控制器, 其中, 传感器负责用户与业务数据的

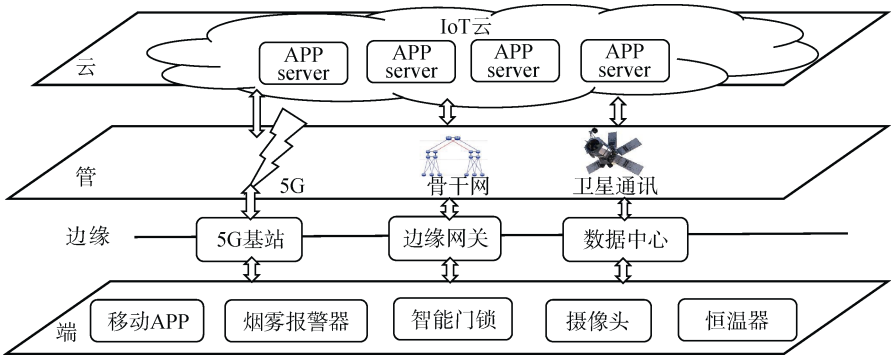

图 1 物联网系统架构

Figure 1 IoT system architecture

采集, 控制器负责对物理世界的对象进行控制。管道负责端数据的上传或者云平台控制命令的下放。边缘是边缘计算在物联网中的应用, 主要靠近端侧, 辅助端节点进行计算和存储, 进一步提升物联网服务的实时性和智能化, 同时也为端设备提供安全服务。下面介绍物联网各实体的发展现状。

云平台: 全球各大 IT 厂商(苹果、微软、三星、亚马逊、ARM)均开发了物联网云平台[2-6],为智能家居、工业控制等领域提供数据采集与处理、设备管控等服务。通过对各厂商的云平台架构进行分析, 云平台基本上由硬件、操作系统、资源虚拟化层、设备抽象层、服务管理层、接口层、上层应用组成。其中, 资源虚拟化保证各类物联网业务在共享物联网平台的情况下, 能够做到业务之间的独立和互不干扰。而设备抽象层实现了各类型设备在物联网平台上统一的数据查询与管控接口, 使得物联网设备与云平台的连接、通信更加兼容和智能化。

端: 主要分为两大类, 一类是随着移动互联网技术发展的智能化移动终端, 包括平板电脑、手机, 主要功能是实现用户对物联网设备数据的查询与设备控制。另一类则是各类型的物联网设备。随着万物互联时代的来临, 小到心脏起搏器、智能门锁、烟雾报警器、智能门锁、恒温器等, 大到共享单车, 智能汽车、信号灯等都成为了物联网设备端。

管道: 作为云平台和端的媒介, 低功率广域网(LPWA)[7,8]是目前构成物联网网络的重要形式, 在低成本条件下实现了大范围端通信的覆盖。其中, 包括窄带物联网(NB-IoT)[9]、Lora[9-10]等物联网管道技术。随着通信技术的进一步发展, 5G 网络、卫星通信等也将成为物联网管道的重要组成部分。

边缘: 物联网的边缘在云和端之间, 辅助云和端完成部分功能, 进一步提升物联网系统的高效性和可靠性。根据华为云对物联网边缘的论述[11], 总结出物联网边缘对整个系统的如下好处。(1)快速决策: 获取云平台的部分规则, 针对端数据进行快速决策, 降低任务处理的时延; (2)降低组件计算与通信的负担: 对端采集数据进行初步分析, 再上传至云平台, 或是辅助云平台进行任务处理, 减少云平台的计算负担, 同时降低对管道带宽的要求; (3)提高系统安全性: 端数据采集后直接由边缘处理, 并将决策结果反馈至其它端设备, 缩短业务传递和数据传输的链条, 减少不必要的安全风险; (4)提高兼容性: 在物联网边缘位置为不同协议、数据模型的端设备提供统一的云接入方式。

3 物联网信息域的安全威胁

物联网系统由云平台、管道、端(移动终端、物联网设备)、边缘构成。与传统的网络系统相比, 其信息域面临的安全威胁存在差异。首先, 本论文分析各实体内部存在的安全威胁。由于 Android、iOS 等移动终端的安全问题在移动安全领域已有深入研究, 同样地, 边缘节点的安全威胁在边缘计算中有了深入的讨论。因此本论文主要分析除移动终端与边缘之外实体面临的安全威胁。此外, 论文将分析实体交互(主要是云和端)过程中存在的安全威胁。

在介绍各种安全威胁时, 论文将引入物联网传感模型简化图(未包含边缘实体)进行分析, 如图 2-5。根据图中所示, 物联网系统分为信息域和物理域。其中, 传感过程将物理域的物理对象作为起点, 由信息域的传感器感知, 通过传输通道, 将数据传输给云平台进行处理, 最终根据处理结果下放控制命令, 经由传输通道, 由控制器完成对物理域被控部件的控制。后续论文在传感模型简化图中对各类安全威胁进行深入探究。

3.1 物联网云平台的安全威胁

物联网云平台由多层功能结构构成, 若系统存在任意一层功能设计、实现、配置、运行的漏洞, 攻击者可以利用这些漏洞对云平台发起攻击, 对系统进行渗透、非法控制、数据窃取等。下面具体介绍云平台中存在的虚拟机逃逸、数据泄漏与丢失, 认证劫持与绕过的安全威胁。

物联网云平台中的虚拟机需要对 CPU、内存、I/O 硬件资源进行虚拟化。相比于 CPU、内存等通用硬件, I/O 硬件的种类繁多, 完整功能的虚拟化实现复杂, 且 I/O 虚拟化代码量大且不统一, 因而容易引入较多的安全漏洞。而攻击者可以利用这些漏洞跨虚拟机对宿主机上的内存进行越权读写, 导致宿主机资源的非法访问。通过对 VMware[12]、VirtualBox[13]、QEMU[14]等虚拟化应用的 I/O 虚拟化漏洞进行分析, 发现漏洞数量相对于整个软件的漏洞数量占比较大。因此, 云平台因 I/O 虚拟化漏洞带来的虚拟机逃逸的安全威胁较大。

其次, 云平台存在数据泄漏与丢失的安全威胁。物联网云平台中的应用服务存放大量端的数据(移动终端用户数据、物联网设备状态与任务数据)。若云平台配置不当或 APP 存在脆弱性, 则攻击者可以获取到数据的访问与操作权限, 对数据进行盗取或破坏。因此云平台存在数据泄漏与丢失的安全威胁。

此外, 云平台存在账户被劫持、认证被绕过的安

全威胁。云平台对用户口令、配置信息的管理不够严格, 导致攻击者轻易获取认证信息从而登陆云平台, 获取重要数据。同样的, 若口令设置过于简单, 易被攻击者破解, 认证机制也会被绕过。

图 2 显示了物联网云平台的安全威胁。攻击者利用云平台的脆弱性成功入侵云平台。之后, 攻击者可以通过传输通道对控制器进行操作, 从而按照其攻击意图影响物理世界。

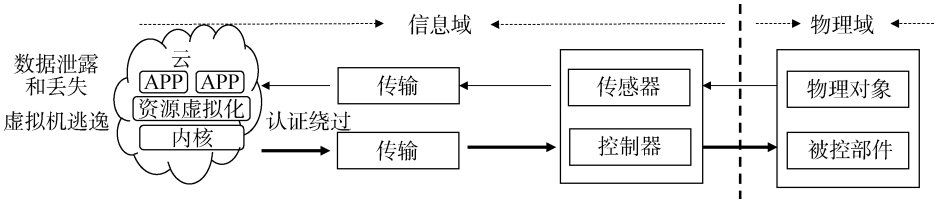

图 2 云平台安全威胁

Figure 2 IoT cloud security threat

3.2 物联网设备端的安全威胁

相比于通用信息系统, 物联网设备一旦运行, 其软硬件通常难以进行更改。由于这个特点, 一方面, 很多设备运行需要的信息(认证信息)或调试方法(后门)需提前预置在设备中, 因此存在认证信息窃取和后门利用的安全威胁。另一方面, 若设备中存在漏洞, 其也将会长时间的存在于设备中, 导致安全漏洞持续可利用的安全威胁。此外, 由于设备计算和存储能力有限, 安全性检测功能缺乏, 因此存在固件篡改的安全威胁。图 3 显示了物联网设备的安全威胁。攻击者利用控制器持续可利用的安全漏洞、易泄漏的认证信息、潜在后门、篡改固件等方式, 直接操纵控制器, 按照其攻击意图影响物理世界。对于其余的设备数据泄漏等信息系统中通用的安全威胁, 论文将不再赘述。

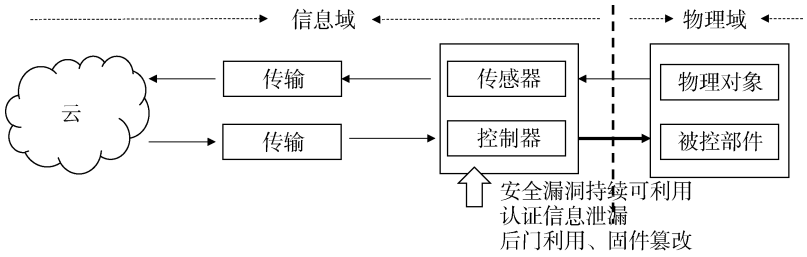

图 3 设备端安全威胁

Figure 3 IoT device security threat

3.2.1 安全漏洞持续可利用

由于不同类型的物联网设备需求大, 很多厂商在定制设备时会使用第三方开源组件。因而设备软件模块存在大量同构, 一旦第三方开源组件存在漏洞, 则大量物联网设备将同样存在漏洞。文献[15]大规模地分析了此类问题。然而, 由于信息的不对称, 安全分析人员很难确定所有存在开源组件漏洞的物联网设备。同时, 设备一旦上线后, 其使用权和管理权分散到各个用户。虽然第三方组件的漏洞信息开源, 但是否修复以及修复时长仍有很多不确定因素。综上, 大量开源组件漏洞可能长期存在于物联网设备中, 并且持续可被攻击者利用。

3.2.2 认证信息泄漏

物联网设备在使用之前需要与用户账户绑定, 并通过云平台的认证连入云平台。然而, 由于物联网设备海量, 认证模式不统一, 强度强弱不一, 导致大量设备认证存在安全威胁, 易被攻击者劫持从而伪造成设备。文献[16]分析了如下认证信息泄漏的安全威胁。在各类物联网云平台对设备认证过程中, 会利用一些辅助信息用于认证。然而, 这些认证信息存在以下可能泄漏的风险。1)设备认证信息可公开获取。例如: 阿里的 Alink 平台将物联网设备认证相关的凭证公开在开源网站上, 攻击者可以直接获取。2)认证信息易被猜解。例如: Alink 平台将 MAC 地址作设备标识参与认证。然而, 对于特定厂商、型号的物联网设备, 其 MAC 地址的所在地址空间也是确定的, 因而容易被推断出来。3)设备认证的信息硬编码在设备上。由于设备标识硬编码在设备中, 攻击者可以在局域网被动监听或者通过日志文件获取该标识。由于设备标识是不会变化的, 一旦攻击者获取设备标识, 则设备将永久处于不安全状态。

3.2.3 后门利用

物联网设备在开发过程中, 为了实现设备上线

后的调试, 开发者在设备固件中有意或无意地预留了后门。这些后门通常是一些特权操作, 当输入满足一定条件时, 即可执行如重启、固件上传下载、功能开启关闭等操作。而这些条件通常是输入一些口令信息, 而这些信息硬编码在设备固件中, 通过静态分析固件即可获得。由于物联网设备无人值守, 这些安全后门可以在物联网设备部署之后被攻击者利用。为了深入分析这类安全威胁, 文献[17]利用静态程序分析、符号执行技术、机器学习分类等方法, 分析了认证信息硬编码、隐藏认证接口、命令注入等类型的后门, 同时发现物联网设备各类应用服务中的后门。

3.2.4 固件篡改

物联网设备固件需要不断升级, 从而满足设备功能的更新换代, 或者设备漏洞的修复。由于物联网设备缺乏对固件来源的验证和固件完整性校验, 导致攻击者可以直接使用包含恶意代码的固件替代原始固件, 从而控制设备并进一步传播恶意代码。文献[18]介绍了如何通过打印机的特定端口下载包含木马的固件到设备中去。文献[19]研究了针对工控可编程控制器(PLC)固件的修改。由于设备没有对更新固件进行深度校验, 使得攻击非常容易实现。

3.3 物联网管道的安全威胁

物联网管道作为云和端、边缘之间的媒介, 用于传送来自于端的感知数据, 以及来自于云平台的控制命令。它由各种网络设备组成, 可以是各种带宽、覆盖率的网络形式。除了传统的网络设备脆弱性带来的安全威胁, 无线通信的特点也带来了安全威胁。图 4 显示了物联网管道的安全威胁。攻击者劫持传输通道, 截获私密数据, 或者更改数据, 致使云平台作出错误决策, 反过来影响控制器做出错误操作。根据文献[20-21], 物联网管道中的无线传输信道存在如下安全威胁。

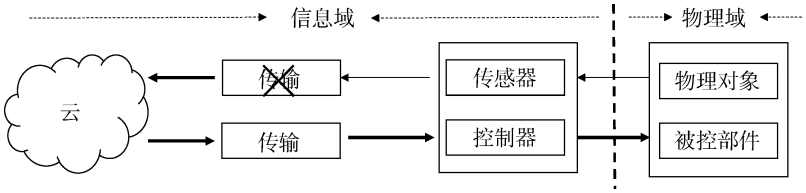

图 4 管道安全威胁

Figure 4 IoT pipeline security threat

(1) 信道数据易被监听。例如在共享单车应用中, 共享单车每次使用完, 需要由云平台通过管道网络(如 NB-IoT)进行新密码的下放。攻击者可以通过中间人攻击, 监听到密码, 从而实现免费骑行。

(2) 信道数据篡改后不被发现。在物联网管道的无线通信信道中, 通常含有两个通道: 命令通道和数据通道。命令通道用于向云平台发送心跳报文证明设备仍处于连接状态。数据通道用于传输设备(传感器)测量的数据。在该机制下, 攻击者可以劫持数据通道, 而不攻击命令通道。之后, 云平台在收到设备正常心跳报文, 而没收到数据通道的数据时, 则认为传感器的数据没有发生变化。攻击者可以在这一时间段改变物理域的事件而不被发现。

3.4 物联网云、端交互中的安全威胁

除了物联网云平台、管道、端自身存在的安全威胁, 云与端在交互过程中, 同样存在安全威胁。在云、端交互过程中, 至少存在云平台、物联网设备、用户移动终端三个实体。在交互的过程中, 可能存在如下状态破坏[16]、越权操作[22]的安全威胁, 导致攻击者可以执行非授权操作。

对于状态破坏的安全威胁, 主要是因为云平台未对设备端进行严格的状态保护。无论是云平台还是设备端, 都维护着相应的状态机。其中, 包括设备注册、设备绑定等状态。在不同的状态下, 设备可以进行相应的操作。然而, 由于云平台对设备状态没有严格的保护, 导致云平台在某些状态下可以进行其它状态才允许的操作, 致使安全隐患的存在。例如, 若云平台存在状态同步缺陷, 攻击者可以在云平台与烟雾传感器已连接的状态下, 不经过认证直接绑定一台新的烟雾传感器, 并伪造烟雾传感器发送烟雾消息, 从而使得云平台的门锁管理 APP 打开门锁, 保证安全逃生。事实上, 并没有烟雾产生而门锁却被打开, 攻击者可以未授权地进入房间。

对于越权操作的安全威胁, 主要是因为设备端会接收来自局域网移动终端、云平台的多方控制而导致的。由于在设备固件代码中, 没有对来自不同源的同一命令进行分离处理, 使攻击者可以绕过云平台, 在局域网内直接使用移动终端发送相应的功能操作, 从而实现非授权功能的执行。例如, 设备与用户的解绑操作需要通过云平台操作。若固件处理解绑的逻辑没有做源校验, 则攻击者可以在局域网内使用移动终端, 直接完成设备与云平台的解绑。

4 物联网跨域的安全威胁

由于物联网系统与物理世界紧密相连, 除了云、管、端、边缘信息域存在的安全威胁, 物联网系统还存在跨物理域的安全威胁。一方面, 系统存在来自于物理域的跨域安全威胁, 包括多域级联攻击、信息域对物理域输入的理解不全面带来的安全威胁。另一方面, 系统存在信息域对物理域带来的安全威胁, 包括物理域的冲突与叠加、信息域对物理域进行非预期控制的安全威胁。图 5 显示物联网典型的跨域的安全威胁。由于物联网信息域(如云平台)存在缺陷, 则攻击者可以构造特定输入, 影响信息域系统做出有冲突或错误的决策, 从而在反馈到物理域时生成错误的输出。

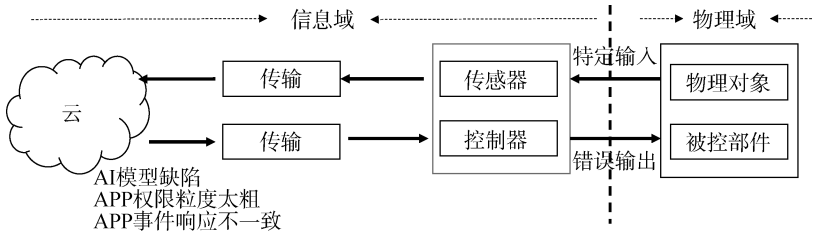

图 5 物联网跨域安全威胁

Figure 5 IoT cross-domain security threat

4.1 多域级联攻击的安全威胁

对于物联网系统, 存在一套物理世界和信息世界的联动规则。攻击者可以利用物理世界与信息系统不断交织的特点, 生成新型级联攻击。具体来说, 若两个物联网系统存在物理域->信息域->物理域->信息域->物理域的传播规则, 前一个物联网系统的物理域输出是后一个物联网系统的物理域输入。攻击者通过构造前一个系统的物理域输入, 并根据级联效应影响到另一个系统的物理域输出。例如, 家庭 AI 智能产品可以接受用户的语音指令, 进行智能家居控制、音频播放、网上购物等, 而门锁控制器在听到指定分贝的烟雾报警声时会打开门锁。如果攻击者在门外通过语音控制让家庭 AI 智能产品播放烟雾报警声的音频并调大音量, 则门锁控制器会误以为是烟雾报警器发出的报警, 从而打开门锁, 让攻击者可以未经授权进入他人家门。该攻击能够成功, 单从家庭 AI 智能系统和门锁控制器系统看来没有任何缺陷。然而, 由于门锁控制器对物理输入理解不全面(未料想到 AI 智能系统可以播放烟雾报警声音频), 导致可以发起跨系统的级联攻击, 即语音指令(物理域)->家庭 AI 智能系统(信息域)->高分贝烟雾报警音频(物理域)->智能门锁系统(信息域)->打开门锁(物理域)。然而, 对于这类安全威胁, 仅依靠单系统的防护方案已不足以保障安全。

4.2 物理域的冲突与叠加

物联网信息系统在对物理域进行操作和控制时, 仅按照物理域的输入制定的规则, 通常缺少对物理输出导致的真实情况的判断, 导致物理域的变化存在冲突和叠加。具体来说, 云平台中的 APP 存在事件响应逻辑, 即对物理域发生的特定事件进行相应的处理, 指示特定控制器完成相应动作。通常情况下, 云平台中多类型 APP 处理不同的任务, 然而, 攻击者会恶意构造 APP, 使 APP 之间的事件响应逻辑相违背, 从而影响物理端设备, 对物理世界造成危害。根据文献[23-24], 总结出如下安全威胁。

(1) 物理域的缺失: 云 APP 本身没有对特定事件进行相应处理, 导致发生特定事件后没有响应, 使得物联网设备处于不安全状态。例如: 当探测到烟雾时, 若烟雾监控 APP 未对该事件进行响应, 发送报警控制命令, 则家庭可能会发生火灾危险。

(2) 物理域的冲突: 物联网系统在处理相同事件或不同事件时, 会造成物理域的冲突。一方面, 云平台中不同 APP 在处理相同事件时会将同一控制器设置为不同的值, 当同一事件发生时, 设备的状态将处于矛盾状态。例如, 云 APP A 在检测到家里无人时打开窗户进行通风, 而云 APP B 在检测到家里无人时关闭窗户来保证安全。这样窗户在家里无人时将处于矛盾状态。另一方面, 云平台中不同 APP 在处理不同事件时也会将同一控制器设置为不同的值, 设备的状态也处于矛盾状态。例如, 若云 APP A 在检测到家里无人时打开窗户进行通风, 而云 APP C 在检测到屋外风力达到 5 级时关窗户。当家里无人且同时屋外风力达到 5 级时, 窗户将处于冲突状态。

(3) 物理域的叠加: 当多套信息域系统控制统一控制器时, 会出现物理世界的叠加, 致使物理世界处于不安全状态。例如: 当室外温度为零下时, 某云 APP 将调高恒温器温度 *n* 度。若有多个 APP 均有此事件响应逻辑, 恒温器将持续调高多个 *n* 度。

4.3 信息域对物理域进行非预期的控制

物联网系统对信息域如何控制物理域设定了规则。具体来说, 物联网的云平台与通用移动终端操作系统一样, 存在权限机制, 在云 APP 安装与运行过程中, 赋予其对特定控制器相应的操作权限, 用于实现对云 APP 操纵物理世界的访问控制。然而, 若云 APP 在设计与实现存在缺陷, 导致信息域对物理域的控制与其描述的不一致, 或与用户预期的不一致, 则存在云 APP 对物理域进行非预期操作的安全威胁。其中非预期的控制主要分为以下两个方面。

(1) 云 APP 的对端设备的权限使用情况与其描述的不一致, 导致云 APP 在用户不知情的情况下获得额外的设备操作权限, 导致端设备不安全。

(2) 云 APP 获得额外的非必要权限, 可以执行用户非预期的操作[25]。例如, 门锁打开和关闭属于不同权限, 但某些家庭安全监控的 APP 在获得门锁关闭的权限时, 同时获取了门锁打开的权限, 导致该家庭安全监控 APP 可以在让用户不在家的时候打开门锁, 让家处于不安全的状态。

4.4 信息域对物理域输入的理解不全面

物联网系统的信息域需要对物理域的输入进行分析, 并作出正确的决策, 进一步控制物理域的输出。若物联网系统对物理域的输入存在理解不全面的情况, 则攻击者可以在物理域中构造对抗输入, 导致信息域作出错误决策, 最终将反馈至物理域危害物理世界。

当物联网系统对物理域输入中的语义信息存在理解不全面的情况, 攻击者可以发起类似于语音蹲的攻击[26]。在语音蹲攻击中, 由于人类语言存在发音和语义上的歧义性, 导致攻击者可以进行特殊的发音, 通过物联网管道传输到云平台进行处理。由于云 APP 的 AI 模型存在缺陷, 攻击者构造的有歧义语音指令可以使云 APP 启动非预期的服务。

当物联网系统对物理域输入信道信息存在理解不全面的情况, 攻击者可以发起类似于海豚音的攻击[27]。在海豚音攻击中, 攻击者恶意编码信息到非人耳感受到的频段。由于物联网系统未对输入语音频段进行分析, 因此攻击者可以利用非人耳感受频段, 在不被用户察觉的情况下, 让云 APP 下发指令给控制器执行非预期操作, 影响物理世界。

5 物联网的安全模型

针对物联网系统存在的安全威胁, 论文提出基于传统网络安全体系 PDRR 的物联网安全模型, 同样涉及安全防护、检测、响应、恢复 4 个阶段, 如表 1 所示, 全方位地保护物联网系统。其中, 在安全防护阶段, 由于物联网设备存在大规模、异构、跨平台接入等特点, 相比于传统网络, 需进一步在云、端位置加强认证以及授权与访问控制。另外, 端设备的计算和存储能力有限, 需实现轻量级的加密机制保障通信安全。在安全检测阶段, 云、端应加强恶意代码与脆弱性的检测, 尽早修复自身存在的缺陷, 防范于未然。在管道部分, 应加强对异常流量的检测, 防止如 Mirai 僵尸网络发起的 DDoS 攻击。在安全响应阶段, 云平台应加强与管、端的联动, 在安全事件发生后进行决策, 采取相应的措施, 如将不安全的端进行隔离等。在安全恢复阶段, 与传统网络系统一致, 需保留数据备份与恢复、系统的升级与恢复等功能, 而端设备的恢复功能需尽量精简。后续将介绍各阶段的具体策略和措施。

表 1 物联网安全模型

Table 1 IoT security model

	P: 安全保护	D: 安全检测	R: 响应	R: 恢复
云	加强: 认证、授权与访问控制	加强: 脆弱性检测、恶意代码检测	加强: 云、管、端联动	保留
管	保留	加强: 基于流量的安全检测	保留	保留
边缘		基于边缘计算实现安全功能上移或前置		
端	加强: 认证、授权与访问控制 轻量级的通信加密	加强: 脆弱性检测、恶意代码检测	隔离	精简

5.1 安全防护

针对物联网的安全防护, 思科安全提出认证、授权、强制性的安全策略、安全分析逐层递进的安全框架[28]。其中网络强制策略要求所有元素(数据、控制以及管理)需安全地进行传输和路由。在本论文中, 安全防护部分将按照通信加密来论述。因此, 本论文

的安全防护策略将包含认证、授权与访问控制, 通信加密等技术。

5.1.1 认证

认证是整个安全防护体系中最核心的措施。物联网中所有实体进行交互之前均需要完成双方的认证。由于不同类型的物联网系统存在架构、功能上的差异性, 因此认证机制存在一定的区别。文献[29]分析了智能家居、车联网、智能电网、无线传感网等典型物联网系统的身份认证机制。文献[30]综述了机器对机器(M2M)、车联网(IoV)、能源网络(IoE)、传感器网络(IoS)的安全认证协议。

在物联网系统中, 物联网设备相比于云平台、管道、边缘、移动终端等实体, 其具有种类繁多, 规模巨大, 异构等特点。因此, 在认证之前如何有效定义物联网设备的标识是重要的科学问题。文献[31]基于当前对物联网设备没有严格定义的情况, 提出将“继承”、“联系”、“知识”、“环境上下文”组合信息作为设备标识, 并建议使用基于属性的设备认证机制。文献[29]从认证因子、认证过程、认证架构、硬件特征使用等方面对物联网的认证机制进行了分类归纳。根据文献[32], 物联网的安全防护措施应该按照强、弱终端进行区别对待, 其中的认证机制同样需要如此。对于计算能力和存储资源充足的云平台、边缘、以及部分物联网设备, 如智能车辆、摄像头、路由器等, 通过公钥基础设施(如 X.509 证书机制)完成实体之间的认证。为了更安全的进行身份认证, 可搭建基于安全芯片的可信执行环境, 并在此基础上实现认证机制。对于大规模部署的计算和存储能力较弱的设备节点和穿戴式小型或微型设备, 通常采用轻量级的认证方式, 如使用共享密钥、MAC 物理地址、设备硬件特征作为标识来实现认证。

在跨平台的物联网设备认证方面, 由于物联网设备并不与特定厂商的云平台关联, 在使用过程中, 会部署到不同厂商的云平台下, 并且频繁跨域。因此, 需要进一步探究无平台归属的物联网设备身份认证机制, 保证物联网设备认证的有效性和安全性。同时, 物联网设备在移动过程中, 会与新的平台进行绑定。因此, 需保证设备迁移过程中的安全性, 保证与原平台的正确解绑和新平台的正确绑定。此外, 可研究多平台下设备身份映射机制, 实现物联网设备多平台迁移的高效性。

5.1.2 授权与访问控制

授权与访问控制用于实体之间特定信息的交换和功能的控制。该安全功能建立在认证之上, 需要利用认证过程中的实体信息。一旦认证机制存在缺陷或不安全, 授权与访问控制也将失去意义。文献[33]总结通用访问控制技术在物联网中的使用, 涉及基于角色的访问控制、基于属性的访问控制、基于权能的访问控制、基于信任的访问控制等技术。由于授权与访问控制仅在物联网云平台上实现, 容易导致单点失效的问题。因此, 文献[34]提出分布式的基于权能的物联网访问控制技术。然而, 由于物联网设备性能较低且容易被攻破导致不可信, 从而造成分布式访问控制技术的失效, 文献[35]利用区块链技术, 研究基于智能合约的物联网访问控制技术。

另一方面, 文献[36]对物联网的访问控制进行分析, 发现授权由以设备为中心逐渐转变为以能力为中心。虽然, 目前广泛使用的物联网云平台系统与智能终端设备具有相似的工作机制(事件触发-动作响应), 但由于物联网访问控制机制实现的分散化, 导致物联网系统的访问控制仍存在挑战。文献[37]分析出以下 3 个挑战: (1)物联网系统场景(如家庭网络)的访问控制通常由多个包含不同软硬件资源的异构框架实现; (2)事件的获取通常通过不同平台架构下的传感器或 API 获取; (3)访问控制的实施主体与物联网云平台相互独立。针对这些挑战, 文献[37]提出情景式的访问控制, 在各类异构云平台功能和云 APP 之间添加中间层, 该中间层由环境的情景式单元构成, 每一单元代表特定的感知事件, 满足最少权限原则, 从而统一所有事件描述。

由于物联网云平台逐渐开放, 可以兼容不同厂商的设备, 因而物联网云平台允许各类 APP 的开发来控制各种设备。然而, 若云 APP 拥有的过多的物联网设备权限, 或者与其应用描述的权限不一致, 会给物联网系统引入额外的安全风险。文献[38]提出了云 APP 实际权限与描述不一致的检测技术, 一方面通过代码分析提取出云 APP 的实际权限, 另一方面利用自然语言处理技术获取云 APP 描述的权限, 通过两种分析结果的一致性检查, 并在用户使用时给用户相应的提示以用于决策, 减少不安全应用带来的风险。考虑到云 APP 的权限有时是在特定事件条件下触发, 为了解决该类权限过度使用的问题, 文献[39]通过代码补丁的方式, 使得云 APP 在第一次执行某动作时, 向用户展示当前触发事件的上下文, 由用户来决策是否授权, 并记录下云 APP 在这一事件条件下的权限, 而后将默认执行。

此外, 物联网边缘具备一定的计算和存储能力, 可以成为端侧安全技术部署的集中地, 同时也可以屏蔽平台的差异性。因此, 授权与访问控制的手段可以更多地上移或前置到物联网边缘中去实现。

5.1.3 通信加密

通信加密包括对数据、控制命令、管理消息进行安全加密, 防止被攻击者窃取。对于物联网系统, 由于物联网设备的计算和存储能力有限, 需要采用轻量级的加密方法。当数据采用 TCP 协议进行传输, 可采用 TLS 协议来进行加密。当数据采用 UPD 协议进行传输, 应采用 DTLS 协议来进行加密。目前, 根据文献[40]的总结, 苹果 HomeKit 与微软的 Azure IoT 物联网平台采用 TLS 与 DTLS 来实现通信加密。三星 SmartThings 与亚马逊的 AWS IoT 等平台采用 TLS 实现通信加密。此外, ARM 的 Mbed 平台采用定制化的 Mbed TLS 实现通信加密。

为了进一步降低通信加密的损耗, 文献[21]提出 DTLS+技术, 在物联网设备端结束休眠后, 利用 connection id 继续使用原有安全信道, 无需重新通过一系列握手建立安全信道, 最终降低 40%的功耗。

5.1.4 其他安全防护措施

在物联网中的边缘、云平台、管道的网关位置, 需要部署防火墙技术。其中, 通用系统防火墙技术对特定端口的流量进行分析和过滤, 在一定程度上防止拒绝服务、恶意扫描等攻击。对于物联网设备而言, 因移动性较强, 无法使用 IP 技术与设备进行通信。因此, 文献[41]提出了针对物联网应用层级的双向通信方式, 能够在任意防火墙技术下实现设备的实时控制与维护。

同时, 为了增强物联网设备终端的抗攻击能力, 文献[21]提出远程安全升级管理服务(FOTA)、安全启动功能、支持轻量级可信计算(DICE)芯片以及轻量级的系统安全(LiteOS)等安全措施。为了保护云平台的数据, 以及保障平台功能的正确执行, 除了采取 Web 相关的安全防护措施, 需进一步采取代码加固以及数据隐私保护技术。

综上文献所述, 在安全防护阶段, 物联网实体间的认证应根据实体的计算与存储能力部署相应的方案。若计算与存储资源充足, 可基于安全芯片搭建可信执行环境, 并采用公钥基础设施完成认证。若计算与存储能力较弱, 则利用 MAC 物理地址、设备硬件等私有特征实现认证。对于物联网的授权与访问控制, 为了实现不同厂商云平台与设备间的授权与访问控制, 需统一感知事件与操作实施的描述。同时, 需校验实际获得权限与声明或应获得权限的一致性。对于实体间通信的安全性, 应采用 TLS、DTLS、DTLS+等技术实现通行数据加密。

5.2 安全检测

在物联网安全检测策略中, 需要对各实体进行入侵检测、在线安全监测、脆弱性检测以及恶意代码检测。其中, 入侵检测对来自系统外部的报文进行分析, 判断是否存在入侵攻击行为。而在线安全监测获取系统内部设备、应用程序的行为、状态、是否存在已知脆弱性等。

5.2.1 入侵检测

在物联网系统中, 需要对云、管、端、边缘均进行入侵检测。文献[42]对物联网的入侵检测技术发展进行综述。首先, 该文献将物联网划分为物理域、网络域、而应用域。而对于物联网入侵检测检测技术, 按照检测部署位置、检测技术、检测攻击类型、检测技术验证进行分类[42]。在检测部署位置方面, 由于物联网系统是分布式的系统, 检测系统的部署位置可以是中心化(部署在边界路由器)、分布式(部署在网络节点上)或者混合式(部署在汇聚节点上)的[42]。根据本论文对物联网系统的云、管、端定义, 物联网系统应采用混合式的部署方案, 将入侵检测系统部署在物联网边缘上, 实现对端设备以及边缘自身的攻击入侵检测, 减轻端设备节点的计算和运行开销。对于云平台和管道的入侵检测, 一方面, 应部署在出入口位置, 防止外部攻击。同时, 对于实体内部重要节点(数据服务器等), 若具备较强的计算和存储能力, 可采取基于主机的入侵检测方式, 防止实体内部被攻陷从而发起的入侵。

在检测技术方面, 主要分为特征检测、异常检测、规范检测、以及混合检测[42]。其中, 特征检测技术也称为误用检测, 通过生成攻击特征库, 从而识别已知攻击。异常检测技术与特征检测相反, 通过机器学习的方法生成正常行为模式, 将与正常行为模式差距超过一定阈值的行为判定为入侵攻击, 从而有效识别未知攻击。相比于异常检测技术, 基于规范的检测技术根据专家经验生成合法的行为模式, 能够进一步降低检测的误报率[42]。混合检测技术充分结合前三种技术的优势进行入侵检测[42]。在物联网系统中, 若端设备的功能较少, 合法行为模式有限且明确, 可采用规范检测技术。例如: 文献[43]通过在 SmartThings 与 IFTTT 平台上对设备状态进行收集并动态建模, 从而生成物联网系统中的正确安全策略, 并阻断异常状态。相反的, 若针对端设备的攻击事件有限, 可采用特征检测技术。一旦有新攻击被发现, 可添加为新的攻击特征。对于云平台、管道(数据中心)、边缘等实体, 由于与外界传输的样本数量较大, 具备合法行为模式训练的条件, 可在这些实体出入口位置采用异常检测技术。而对于实体内部的各类型服务器, 可采用规范检测技术。

在检测攻击类型方面, 主要分为传统攻击、路由的攻击、中间人攻击以及拒绝服务攻击[42]。其中, 传统攻击主要是针对系统与软件, 如利用其内部的溢出类漏洞、命令注入、目录遍历类漏洞发起入侵攻击。路由攻击指的是针对路由层发起的攻击, 包含槽洞攻击、虫洞攻击、女巫攻击、选择性转发攻击等[42]。中间人攻击和拒绝服务攻击在传统网络系统中很常见, 由于物联网设备安全管理较弱, 导致这类攻击在物联网世界更加频发。在物联网系统中, 需对所有实体内部的主机、设备进行传统攻击、中间人攻击的入侵检测。例如, 文献[44]提出基于自然语言处理的漏洞报告利用脚本提取技术, 成功生成针对物联网设备传统攻击的入侵特征库。而在云平台中, 由于 Web 服务的存在, 应部署网站应用级入侵防御系统, 防止针对云平台 Web 漏洞的入侵。针对云平台、管道, 需进一步进行路由攻击的检测。针对管道网络, 应在出入口位置进行拒绝服务攻击检测, 对异常流量进行阻断。总体来说, 针对基于物联网设备形成的僵尸网络发起的分布式拒绝服务攻击的检测, 需联动系统中的所有实体入侵检测组件进行抵御。

在检测技术验证方面, 主要包括假设类、实证类、仿真类、理论证明类[42]。其中, 实证类是搭建实体软硬件的仿真环境进行技术验证, 而仿真类是是使用网络仿真工具来进行技术验证[42]。在物联网系统中, 传统攻击针对系统与软件真实缺陷, 需要采用实证类的验证方法。路由攻击与中间人攻击检测需关注网络拓扑, 多采用仿真类的方法。而对于拒绝服务攻击检测, 两种验证方法均可采用。

5.2.2 在线安全监测

在线安全监测对物联网系统中的实体状态、行为、脆弱性进行实施观测, 并将结果反馈至云平台进行分析。对于端设备, 采用轻量级的监测方式, 对设备的关键信息和状态进行监控, 防止未知攻击对这些关键数据的篡改。根据现有的物联网安全监测平台[45], 监测内容应包含物联网设备接入监测、设备状态监测以及设备漏洞实时监测等方面。其中, 设备接入监测通过硬件指纹比对来判断是否存在设备伪造、非法接入等情况。设备状态监测需要将设备的异常状态及时上传给云平台, 从而分析是否存在攻击入侵的发生。设备漏洞实时监测需生成物联网设备已知漏洞库(包含漏洞基本信息与漏洞指纹), 并通过扫描设备以及漏洞指纹比对判断设备是否存在已知漏洞。

对于边缘实体, 需实时监测云平台与端的通信, 包括提取上传的感知数据与下放的控制命令, 并将这些数据和命令进行备份, 以用于后续的取证与安全分析。对于边缘与端侧设备构成的局域网, 需进行局域网通信质量的实时监测与评估, 从而判断局域网是否受到拒绝服务等攻击。

物联网云平台安全监测部署在云平台上, 对云平台系统状态、漏洞以及云 APP 的漏洞、行为、更新升级进行实时监测。主要包含以下方面: (1)云平台系统状态监测: 对 CPU、内存占用率以及进程等系统状态进行实时监测; (2)云平台系统漏洞监测: 生成系统内核、设备驱动、虚拟机、管理组件等服务的已知漏洞指纹, 以探测系统内存在的已知漏洞; (3)云 APP 的漏洞监测: 通过建立云 APP 及其使用第三方库的已知漏洞指纹, 在系统层实施探测从而发现漏洞; (4)云 APP 行为监测: 对云 APP 的权限操作进行监测, 并在后端进行实施分析, 判断是否有越权或恶意操作; (5)云 APP 更新升级检测: 监测云 APP 的更新来源是否可信, 并对更新包的完整性进行检测。

5.2.3 脆弱性检测

相比于通用网络系统与软件, 由于物联网中实体多交互、智能化特点, 暴露出更多脆弱性, 因此需要对这些脆弱性进行检测。脆弱性的检测主要分为云平台的检测和端设备的检测。由于边缘和管道中的系统与设备属于通用系统或嵌入式端设备范畴, 因此不单独说明。后续将论述云、端中的脆弱性类型和检测的技术手段。

在云平台系统漏洞检测中, 需要分析与挖掘以下特有的漏洞类型。(1)云平台的系统漏洞。通过检测云平台虚拟化漏洞, 防止虚拟机逃逸、跨虚拟机内存读写等安全攻击; 同时, 检测云平台内核、云平台管理组件等漏洞, 防止系统被渗透。(2)云平台用户接口的安全性。可通过模糊测试的方法测试云平台提供的 API, 从而分析系统接口的安全性。

在云平台 APP 漏洞检测中, 需要分析与挖掘以下漏洞类型。(1)APP 通用漏洞类型, 包括内存溢出类、命令注入类等通用漏洞, 可采用移动 APP 的检测技术, 如符号执行[46]、模糊测试[47]等; (2)云 APP 存在的逻辑漏洞, 如多 APP 权限设计与实现存在的不一致, 导致同一事件发生时会产生不同的决策, 从而对对物理域造成危害。这种情况下, 需要对 APP 的权限进行建模, 并在全局视角进行逻辑分析, 分析出相违背的设计与实现。(3)第三方组件中存在的漏洞, 可以通过已有的漏洞信息库(CVE、CNVD 等)中组件版本号的比对, 分析云 APP 是否使用了有脆弱性的组件; (4)APP AI 模型的缺陷。随着人工智能的发展, 各厂商提供的云平台服务采用 AI 模型来进

行决策。对于 AI 模型的脆弱性, 华为总结了若干攻击方式[48]。其中, 若 AI 模型训练样本不充足, 会导致 AI 模型预测准确性降低, 攻击者可以产生对抗输入从而造成 AI 模型决策错误。对于这种 AI 模型的脆弱性, 可以通过生成的对抗样本, 或是进行 DNN 模型验证等方法来检测。另一方面, 开发者可以在 AI 模型中嵌入后门, 一旦 AI 模型中的后门被攻击者发现, 同样造成很严重的结果。对于 AI 后门的检测, 文献[48]仅总结出防御方法, 能够有效减少后门触发的几率, 但没有提出有效检测后门的方法。

对于端设备的检测, 由于固件是物联网设备的软件形式, 包含完整的操作系统、文件系统、服务程序。对固件的漏洞分析足以完全挖掘设备软件层面的漏洞。而固件的漏洞类型有以下几个方面: (1)固件通用漏洞。系统与软件的通用漏洞类型, 如内存破坏类(栈溢出、堆溢出、指针二次释放、空指针引用等)、输入验证类(命令注入类、SQL 注入等)、敏感信息泄漏(目录遍历、弱口令等)漏洞。(2)固件后门漏洞。为了方便物联网设备在上线之后的管理和调试, 开发人员会在固件中预留后门(口令硬编码等), 导致如 3.2.3 节提到的后门利用的安全威胁。在文献[17]中, 通过静态程序分析以及符号执行等技术发现后门漏洞, 并分析出触发后门漏洞的输入。(3)固件供应链上的脆弱性。由于物联网技术发展迅速, 物联网设备中很多功能并非重新定制设计开发, 而是沿用通用系统和软件的组件。因此, 若通用软件、库中存在脆弱性, 则引入它的物联网设备固件中同样存在该问题, 导致如 3.2.1 节提到的安全威胁。所以需要对固件的供应链进行分析和识别, 并深入安全检测。(4)固件内敏感信息(如私钥、配置文件)易泄漏。由于物联网设备的计算和存储能力有限, 导致对这些敏感信息未进行有效地安全保护, 从而容易被攻击者窃取。(5)固件逻辑漏洞。在物联网系统云、管、端的特定架构下, 物联网设备会与云, 端(移动 APP, 其他设备)进行通信。若各实体之前的通信或状态转移逻辑存在缺陷, 则会被攻击者利用从而入侵系统。例如 3.4 节, 物联网设备未对云平台和移动端的命令进行权限分离的逻辑漏洞。

对于设备的漏洞挖掘方法, 可以采用非在线的方式, 即在设备非真实部署的运行环境下, 对固件进行静态程序分析或动态运行分析的方法。在静态方法中, 将现有的通用漏洞分析方法通过改进, 运用到物联网设备固件的分析, 如数据流分析[49-51]、污点分析、符号执行[52-53]技术等, 可以有效分析出固件中的通用漏洞、后门漏洞等。在动态分析方法中, 研究者们关注设备固件仿真技术[54-58], 在仿真成功的基础上, 研究灰盒测试[59-67]、污点分析等动态分析方法, 有助于更精准地挖掘固件漏洞。物联网固件的漏洞关联[15,68-70]是近几年由于物联网设备固件频繁复用开源组件引入的研究方法, 可以有效挖掘固件供应链引入的脆弱性。对于感知信息处理逻辑的脆弱性分析, 需要对物理域存在的威胁充分理解, 从而分析是否存在脆弱性。如海豚音攻击, 通过理解非人耳感知频段攻击输入的存在, 从而分析语音信号处理存在的脆弱性。

5.2.4 恶意代码检测

在物联网恶意代码的检测中, 对于端设备更多的采用非在线的动静态分析方法。对于云平台, 则多采用在线的分析方法。首先, 在云平台恶意代码检测方面, 一是对云 APP 的权限使用与声明的一致性进行检测[38], 从而判断云 APP 是否存在额外的非预期权限功能。同时, 分析云 APP 的事件响应逻辑。若存在逻辑缺失或不一致性的问题, 则不允许该云 APP 上线云平台。此外, 采用沙箱技术, 搭建与云平台一致的分析系统, 捕获恶意代码的功能操作。

在端恶意代码检测方面, 首先进行移动 APP 的恶意代码检测, 分析 APP 的实际权限是否与声明不一致, 是否存在越权操作等。对于设备端的恶意代码检测, 分析设备内部是否存在木马、僵尸程序等恶意代码(如 Mirai 等)。文献[71-72]分析了物联网设备恶意代码的种类, 尤其对具有 DDoS 能力, 能对网络空间产生严重威胁的恶意代码进行分析。文献[71]综述性地介绍了物联网恶意代码的检测方法, 而文献[73-75]则提出了具体的检测技术, 通过蜜罐技术捕获恶意代码并使用沙箱技术进行分析。

综上文献所述, 在入侵检测、在线安全监测、脆弱性检测、恶意代码检测方面, 物联网系统各实体根据自身特征采取不同的检测技术。其中, 具体应采用的技术需要根据实际物联网系统的应用场景确定。此外, 由于云平台、边缘具备更强的计算能力与资源, 其检测技术应结合先进的人工智能技术, 从而提高检测的准确率。

5.3 响应

在响应策略中, 需要对检测出的安全事件进行响应。在原有的网络安全模型中, 响应阶段由多个部门机构完成, 包括国土安全、司法、国防、情报等机构, 完成安全行动资源配置、态势感知、安全事件影响评估、犯罪调查等响应工作[76]。对于新型物联网的安全事件响应, 同样需要相应机构健全机制并协同合作来完成安全响应工作。此外, 对于物联网云平

台、管道设施、边缘设备、端设备的厂商, 需要采取以下响应措施。(1)入侵事件的分析与响应; (2)漏洞与恶意代码发现的公告与修复; (3)安全防护加固与检测规则的更新。

5.3.1 入侵事件分析与响应

首先, 对入侵事件给物联网各实体造成的影响进行分析, 包含系统检测与运维日志分析手段。在系统检测方面, 检测是否有非必要端口开放, 可疑进程启动, 可执行程序与配置文件被篡改等情况。在运维日志分析方面, 分析系统与安全产品生成的日志文件, 进一步推断入侵行为并对攻击进行溯源。在明确入侵事件给物联网实体造成的影响后, 将在恢复阶段进行相应系统与软件的恢复。另一方面, 云平台、管道、边缘、端设备应对入侵事件进行快速联动响应, 若检测出端设备存在入侵事件, 云平台、管道和边缘应对不安全设备进行隔离与告警。若云平台发生入侵事件, 管道、边缘应对入侵告警进行响应, 立即停止云平台对下层设备的不安全的功能操作, 直到云平台功能恢复。

5.3.2 漏洞与恶意代码公告与修复

当物联网的任一系统与软件发现漏洞, 应及时公开相关信息(组件厂商、型号、版本、漏洞描述、POC 等), 保证物联网系统各部分能够先通过其他方式阻止攻击的发生。接着, 应及时分析漏洞成因, 发布漏洞补丁, 并告知用户尽快修复。

当在物联网的任意实体中发现恶意代码时, 应及时公开相关信息, 保证物联网系统其他实体能够通过其他方式阻止恶意代码的进一步传播和破坏。接着, 应及时分析恶意代码的功能、家族等信息。

5.3.3 安全防护加固与检测规则的更新

为了抵御新的攻击入侵, 应在安全防护阶段进行加固, 或在检测阶段进行入侵检测规则的更新。例如: 针对认证或授权协议缺陷发起的攻击, 应在防护阶段修复认证协议或授权协议存在的缺陷。针对系统特定端口, 或携带特定攻击载荷的攻击入侵, 可通过在防火墙或者入侵检测系统中添加相应的规则来阻断攻击。

5.4 恢复

在恢复策略中, 一方面需要对关键数据进行恢复, 另一方面需要对系统进行升级与恢复。

5.4.1 数据备份和恢复

物联网云平台、移动 APP、边缘数据中心保存着大量用户的数据、设备状态数据等信息, 其数据的安全防护尤为重要。一旦这些数据因攻击发生丢失或篡改, 将会对物联网系统造成极其严重的危害。因此物联网系统应具备数据备份和数据恢复功能。

在数据备份方面, 移动 APP 的数据需要在云平台上进行备份, 并进行加密保护, 保证不同 APP 的用户数据不会被非法获取。同时, 物联网云平台数据需要在自身平台上进行备份, 并进行加密保护。此外, 边缘的数据和服务中心需在自身平台上或云上对关键数据进行备份, 并加密存储。

在数据恢复方面, 数据恢复为数据备份的逆过程, 一旦数据发生不可逆转的破坏, 进行数据恢复。

5.4.2 系统升级与恢复

系统恢复过程包括系统升级、软件升级、后门修复几个方面。

在系统升级方面, 若物联网的云平台操作系统、移动端操作系统、管道中网络设备系统、物联网设备固件系统、边缘网关与数据中心的操作系统存在安全缺陷, 需及时对系统进行漏洞补丁操作, 或升级系统。升级过程中, 需对系统升级源进行严格校验, 并对升级包进行完整性的验证。

在软件升级方面, 若物联网的云 APP、移动端 APP、物联网设备服务程序、边缘服务、数据中心应用存在安全缺陷, 需及时对软件或服务进行漏洞补丁操作。升级过程中, 需对软件升级源进行严格校验, 并对升级包的完整性进行验证。

在后门修复方面, 若物联网云平台、管道、移动端、设备端、边缘中存在系统与软件的后门, 应及时关闭或修复, 防止被攻击者发现和利用。后门修复的方式应采用系统或软件升级的方式完成。

6 总结与展望

本论文针对物联网系统安全问题被动补救方式低效的现状, 提出需系统性研究物联网安全问题的思路。其中, 论文分析了物联网面临的安全威胁, 并提出基于 PDRR 的物联网安全模型。在安全威胁分析部分, 一方面, 论文分析了物联网信息系统自身存在的安全威胁, 包括物联网内部各实体(云、管、端)的安全威胁, 以及云、端交互存在的威胁。另一方面, 论文分析了物联网信息域与物理世界交互存在的跨域威胁。在安全模型的构建中, 提出了安全防护、检测、响应、恢复各阶段需采取的措施。其中, 在安全防护阶段, 由于海量物联网设备节点的存在以及频繁跨域的需求, 身份认证和授权与访问控制需进一步加强。其中, 可通过区块链等新兴技术实现分布式的访问控制。在安全检测阶段, 由于物联网云平台和设备的智能化, 入侵检测、恶意代码与脆弱性检测技术相比于通用系统有了新的提升, 通过引入人

工智能等新兴技术, 提高安全检测的准确性与安全性。在响应和恢复中, 需要完成 PDRR 模型中的基本的功能和策略。

当前, 物联网安全模型中的防护和检测方面有了深入的研究。但对于响应和恢复方面, 仅存在一定策略和方案, 仍然缺乏对关键问题与核心技术的研究。事实上, 由于海量物联网设备的存在以及其高移动的特性, 安全事件的响应和系统软件与数据的恢复应更具有挑战性。后续应加强这方面的研究。

参考文献

[1] Wang Y, Sun D G, Lu D. American Network Security Architecture[J]. *Journal of Information Security Research*, 2019, 5(7): 582-585.
(王妍, 孙德刚, 卢丹. 美国网络安全体系架构[J]. *信息安全研究*, 2019, 5(7): 582-585.)

[2] SmartThings. Add a Little Smartness to Your Things. SmartThings Inc. https://www.smartthings.com.

[3] AWS IoT Applications & Solution. Amazon Web Services Inc. https://aws.amazon.com/cn/iot.

[4] HomeKit - Apple Developer. Apple Inc. https://developer.apple.com/homekit.

[5] Azure IoT. Microsoft. https://azure.microsoft.com/en-us/overview/iot.

[6] Mbed, IoT Device Development. Arm Limited. https://www.mbed.com/en.

[7] Raza U, Kulkarni P, Sooriyabandara M. Low Power Wide Area Networks: An Overview[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(2): 855-873.

[8] Finnegan J, Brown S. A Comparative Survey of LPWA Networking[EB/OL]. 2018: arXiv: 1802.04222. https://arxiv.org/abs/1802.04222

[9] Sinha R S, Wei Y Q, Hwang S H. A Survey on LPWA Technology: LoRa and NB-IoT[J]. *ICT Express*, 2017, 3(1): 14-21.

[10] Sarker V K, Queralta J P, Gia T N, et al. A survey on LoRa for IoT: Integrating edge computing[C]. *2019 Fourth International Conference on Fog and Mobile Edge Computing*, 2019: 295-300.

[11] IoT edge. Huawei cloud. https://www.huaweicloud.com/product/iotedge.html.
(IoT 边缘. 华为云. https://www.huaweicloud.com/product/iotedge.html.)

[12] VMware - Official Site. VMware Inc. https://www.vmware.com.

[13] Oracle VM VirtualBox. Oracle. https://www.virtualbox.org.

[14] Bellard F. QEMU, a Fast and Portable Dynamic Translator[C]. *The annual conference on USENIX Annual Technical Conference*, 2005: 41.

[15] Costin A, Zaddach J, Francillon A, et al. A Large-Scale Analysis of the Security of Embedded Firmwares[C]. *The 23rd USENIX conference on Security Symposium*, 2014: 95-110.

[16] Zhou W, Jia Y, Yao Y, et al. Discovering and Understanding the Security Hazards in the Interactions between IoT Devices, Mobile Apps, and Clouds on Smart Home Platforms[C]. *The 28th USENIX Conference on Security Symposium*, 2019: 1133-1150.

[17] Shoshitaishvili Y, Wang R Y, Hauser C, et al. Firmalice - automatic detection of authentication bypass vulnerabilities in binary firmware[C]. *Proceedings 2015 Network and Distributed System Security Symposium*, 2015.

[18] Cui A, Costello M, Stolfo S. When Firmware Modifications Attack: A Case Study of Embedded Exploitation[C]. *Network and Distributed System Security Symposium*, 2013.

[19] Basnight Z, Butts J, Lopez J Jr, et al. Firmware Modification Attacks on Programmable Logic Controllers[J]. *International Journal of Critical Infrastructure Protection*, 2013, 6(2): 76-84.

[20] OConnor T, Enck W, Reaves B. Blinded and Confused: Uncovering Systemic Flaws in Device Telemetry for Smart-Home Internet of Things[C]. *The 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019: 140-150.

[21] IoT security White Paper – Evolving Security Architecture. Huawei. https://www.huawei.com/minisite/iot/img/iot_security_white_paper_2018_v2_cn.pdf, 2018.
(物联网安全技术白皮书 – 安全架构的不断演进. 华为. https://www.huawei.com/minisite/iot/img/iot_security_white_paper_2018_v2_cn.pdf, 2018.)

[22] Yao Y, Zhou W, Jia Y, et al. Identifying Privilege Separation Vulnerabilities in IoT Firmware with Symbolic Execution[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019: 638-657.

[23] Celik Z B, McDaniel P, Tan G. SOTERIA: Automated IoT Safety and Security Analysis[C]. *The 2018 USENIX Conference on Usenix Annual Technical Conference*, 2018: 147-158.

[24] Wang Q, Datta P, Yang W, et al. Charting the Attack Surface of Trigger-Action IoT Platforms[C]. *The 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019: 1439-1453.

[25] Fernandes E, Jung J, Prakash A, et al. Security analysis of emerging smart home applications[C]. *2016 IEEE Symposium on Security and Privacy*, 2016: 636-654.

[26] Zhang N, Mi X H, Feng X, et al. Dangerous skills: understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems[C]. *2019 IEEE Symposium on Security and Privacy*, 2019: 1381-1396.

[27] Zhang G M, Yan C, Ji X Y, et al. DolphinAttack: Inaudible Voice Commands[C]. *The 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017: 103-117.

[28] Securing the Internet of Things: A Proposed Framework. Cisco Security. https://tools.cisco.com/security/center/resources/secure_iot_proposed_framework.

[29] El-Hajj M, Fadlallah A, Chamoun M, et al. A Survey of Internet of Things (IoT) Authentication Schemes[J]. *Sensors* (*Basel*, *Switzerland*), 2019, 19(5): 1141.

[30] Ferrag M A, Maglaras L A, Janicke H, et al. Authentication Protocols for Internet of Things: A Comprehensive Survey[J]. *Security and Communication Networks*, 2017, 2017: 1-41.

[31] Lam K Y, Chi C H. Identity in the Internet-of-Things (IoT): New Challenges and Opportunities[M]. Information and Communications Security. Cham: Springer International Publishing, 2016: 18-26.

[32] Xiaojun Wang, Junhua Yu. Build a Defense-in-Depth Framework to Secure IoT Security. Huawei Technology, vol. 79. https://www. huawei.com/cn/about-huawei/publications/communicate/79/iot-network-security-cn.
(王小军, 余俊华. 构建纵深防御体系, 保障物联网安全. 华为技术, 第 79 期. https://www.huawei.com/cn/about-huawei/publications/communicate/79/iot-network-security-cn.)

[33] Zhang Y, Wu X Q. Access Control in Internet of Things: A Survey[J]. *DEStech Transactions on Engineering and Technology Research*, 2017(apetc).

[34] Hussein D, Bertin E, Frey V. A Community-Driven Access Control Approach in Distributed IoT Environments[J]. *IEEE Communications Magazine*, 2017, 55(3): 146-153.

[35] Zhang Y Y, Kasahara S, Shen Y L, et al. Smart Contract-Based Access Control for the Internet of Things[J]. *IEEE Internet of Things Journal*, 2019, 6(2): 1594-1605.

[36] He W J, Golla M, Padhi R, et al. Rethinking Access Control and Authentication for the Home Internet of Things (IoT)[C]. *The 27th USENIX Conference on Security Symposium*, 2018: 255-272.

[37] Schuster R, Shmatikov V, Tromer E. Situational Access Control in the Internet of Things[C]. *The 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018: 1056-1073.

[38] Tian Y, Zhang N, Lin Y H, et al. Smartauth: User-Centered Authorization for the Internet of Things[C]. *The 26th USENIX Conference on Security Symposium*, 2017: 361-378.

[39] Jia Y J, Chen Q A, Wang S Q, et al. ContexIoT: towards providing contextual integrity to appified IoT platforms[C]. *Proceedings 2017 Network and Distributed System Security Symposium*, 2017.

[40] Ammar M, Russello G, Crispo B. Internet of Things: A Survey on the Security of IoT Frameworks[J]. *Journal of Information Security and Applications*, 2018, 38: 8-27.

[41] Kubler S, Främling K, Buda A. A Standardized Approach to Deal with Firewall and Mobility Policies in the IoT[J]. *Pervasive and Mobile Computing*, 2015, 20: 100-114.

[42] Zarpelão B B, Miani R S, Kawakani C T, et al. A Survey of Intrusion Detection in Internet of Things[J]. *Journal of Network and Computer Applications*, 2017, 84: 25-37.

[43] Celik Z B, Tan G, McDaniel P. IoTGuard: dynamic enforcement of security and safety policy in commodity IoT[C]. *Proceedings 2019 Network and Distributed System Security Symposium*, 2019.

[44] Feng X, Liao X J, Wang X F, et al. Understanding and Securing Device Vulnerabilities through Automated Bug Report Analysis[C]. *The 28th USENIX Conference on Security Symposium*, 2019: 887-903.

[45] IoT Security Monitor Platform. DBAPPSecurity. https://www.dbappsecurity.com.cn/show-57-12-1.html.
(物联网安全监测平台. 安恒信息.https://www.dbappsecurity.com.cn/show-57-12-1.html.)

[46] Mirzaei N, Malek S, Păsăreanu C S, et al. Testing Android Apps through Symbolic Execution[J]. *ACM SIGSOFT Software Engineering Notes*, 2012, 37(6): 1-5.

[47] Ye H, Cheng S Y, Zhang L B, et al. DroidFuzzer: fuzzing the android apps with intent-filter tag[C]. *Proceedings of International Conference on Advances in Mobile Computing & Multimedia – MoMM'13*, 2013: 68-74.

[48] AI Security White Paper. Huawei. https://www-file.huawei.com/-/media/corporate/pdf/cyber-security/ai-security-white-paper-cn.pdf, Oct. 2018.
(AI 安全白皮书. 华为. https://www-file.huawei.com/-/media/corporate/pdf/cyber-security/ai-security-white-paper-cn.pdf, 2018 年 10 月.)

[49] Cheng K, Li Q, Wang L, et al. DTaint: detecting the taint-style vulnerability in embedded device firmware[C]. *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (*DSN*), 2018: 430-441.

[50] Cojocar L, Zaddach J, Verdult R, et al. PIE: parser identification in embedded systems[C]. *The 31st Annual Computer Security Applications Conference*, 2015.

[51] Zheng Y W, Cheng K, Li Z, et al. A Lightweight Method for Accelerating Discovery of Taint-Style Vulnerabilities in Embedded Systems[M]. Information and Communications Security. Cham: Springer International Publishing, 2016: 27-36.

[52] Davidson D, Moench B, Jha S, et al. FIE on Firmware: Finding Vulnerabilities in Embedded Systems Using Symbolic Execution[C]. *The 22nd USENIX conference on Security*, 2013: 463-478.

[53] Corteggiani N, Camurati G, Francillon A. Inception: System-Wide Security Testing of Real-World Embedded Systems Software[C]. *The 27th USENIX Conference on Security Symposium*, 2018: 309-326.

[54] Zaddach J, Bruno L, Francillon A, et al. Avatar: A framework to support dynamic security analysis of embedded systems' firmwares[C]. *Proceedings 2014 Network and Distributed System Security Symposium*, 2014.

[55] Koscher K, Kohno T, Molnar D. SURROGATES: Enabling Near-Real-Time Dynamic Analyses of Embedded Systems[C]. *The 9th USENIX Conference on Offensive Technologies*, 2015: 7.

[56] Chen D D, Egele M, Woo M, et al. Towards automated dynamic analysis for linux-based embedded firmware[C]. *Proceedings 2016 Network and Distributed System Security Symposium*, 2016.

[57] Costin A, Zarras A, Francillon A. Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces[C]. *The 11th ACM on Asia Conference on Computer and Communications Security*, 2016: 437-448.

[58] Marius Muench, Aurélien Francillon, Davide Balzarotti. Avatar2: A Multi-Target Orchestration Platform[C]. *Workshop on Binary Analysis Research*, 2018.

[59] Alimi V, Vernois S, Rosenberger C, et al. Analysis of embedded applications by evolutionary fuzzing[C]. *2014 International Conference on High Performance Computing & Simulation*, 2014: 551-557.

[60] Lee H, Choi K, Chung K, et al. Fuzzing CAN packets into automobiles[C]. *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, 2015: 817-821.

[61] Sim K Y, -C Kuo F, Merkel R. Fuzzing the Out-of-Memory Killer on Embedded Linux: An Adaptive Random Approach[C]. *The 2011 ACM Symposium on Applied Computing*, 2011: 387-392.

[62] Gauthier A, Mazin C, Iguchi-Cartigny J, et al. Enhancing Fuzzing Technique for OKL4 Syscalls Testing[C]. *The 2011 Sixth International Conference on Availability, Reliability and Security*, 2011: 728-733.

[63] TriforceAFL. NCC-Group. https://github.com/nccgroup/TriforceAFL. 2017.

[64] Zheng Y W, Davanian A, Yin H, et al. FIRM-AFL: High-Throughput Greybox Fuzzing of Iot Firmware via Augmented Process Emulation[C]. *The 28th USENIX Conference on Security Symposium*, 2019: 1099-1114.

[65] Zheng Y W, Song Z W, Sun Y Y, et al. An efficient greybox fuzzing scheme for linux-based IoT programs through binary static analysis[C]. *2019 IEEE 38th International Performance Computing and Communications Conference*, 2020: 1-8.

[66] Chen J Y, Diao W R, Zhao Q C, et al. IoTFuzzer: discovering memory corruptions in IoT through app-based fuzzing[C]. *Proceedings 2018 Network and Distributed System Security Symposium*, 2018.

[67] Muench M, Stijohann J, Kargl F, et al. What You corrupt is not what You crash: Challenges in fuzzing embedded devices[C]. *Proceedings 2018 Network and Distributed System Security Symposium*, 2018.

[68] Feng Q, Zhou R D, Xu C C, et al. Scalable Graph-Based Bug Search for Firmware Images[C]. *The 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016: 480-491.

[69] Xu X J, Liu C, Feng Q, et al. Neural Network-Based Graph Embedding for Cross-Platform Binary Code Similarity Detection[C]. *The 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017: 363-376.

[70] Chang Q, Liu Z J, Wang M T, et al. VDNS: An Algorithm for Cross-Platform Vulnerability Searching in Binary Firmware[J]. *Journal of Computer Research and Development*, 2016, 53(10): 2288-2298.
(常青, 刘中金, 王猛涛, 等. VDNS: 一种跨平台的固件漏洞关联算法[J]. *计算机研究与发展*, 2016, 53(10): 2288-2298.)

[71] Karanja E M, Masupe S, Mandu J. Internet of Things Malware: A Survey[J]. *International Journal of Computer Science & Engineering Survey*, 2017, 8(3): 1-20.

[72] Spognardi A, De Donno M, Dragoni N, et al. Analysis of DDoS-capable IoT malwares[C]. *The 2017 Federated Conference on Computer Science and Information Systems", "Annals of Computer Science and Information Systems*, 2017: 807-816.

[73] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, et al. IoTPOT: Analysing the Rise of IoT Compromises[C]. *USENIX Workshop on Offensive Technologies*, 2015.

[74] Andrei Costin, Jonas Zaddach. IoT Malware: Comprehensive Survey, Analysis Framework and Case Studies[C]. *BlackHat 2018 USA*, August 2018.

[75] Sun H, Wang X F, Buyya R, et al. CloudEyes: Cloud-Based Malware Detection with Reversible Sketch for Resource-Constrained Internet of Things (IoT) Devices[J]. *Software*: *Practice and Experience*, 2017, 47(3): 421-441.

[76] Feng Liu, Dongdai Lin. American Cyberspace Security Architecture[M]. Science Press, 2015.
(刘峰, 林东岱. 美国网络空间安全体系[M]. 科学出版社, 2015.)

郑尧文 于 2020 年在中国科学院大学网络空间安全专业获得博士学位。研究领域为 IoT 安全, 嵌入式设备安全。研究兴趣包括二进制分析, 模糊测试, 漏洞分析与利用。Email: zhengyaowen@iie.ac.cn

文辉 于 2016 年在中国科学院大学信息安全专业获得博士学位。现任中国科学院信息工程研究所助理研究员。研究领域为信息安全、数据挖掘。研究兴趣包括: 物联网安全、恶意代码检测与分析、数据关联与挖掘。Email: wenhui@iie.ac.cn

程凯 于 2014 年在西安电子科技大学计算机科学与技术专业获得学士学位。现在中国科学院信息工程研究所网络空间安全专业攻读博士学位。研究领域为 IoT 安全、嵌入式设备安全。研究兴趣包括: 二进制逆向、固件安全分析、漏洞挖掘。Email: chengkai@iie.ac.cn

李红 于 2017 年在中国科学院大学网络空间安全专业获得博士学位。现任中国科学院信息工程研究所副研究员。研究兴趣包括: 物联网安全、隐私保护、区块链。Email: lihong@iie.ac.cn

朱红松 于 2009 年在中国科学院计算技术研究所计算机体系结构专业获得博士学位。现任中国科学院信息工程研究所研究员。研究领域为网络空间安全。研究兴趣包括: 物联网安全、网络对抗、智能攻防, 网络空间安全测量和威胁态势感知等。Email: zhuhongsong@iie.ac.cn

孙利民 于 1998 年在国防科技大学计算机科学与技术专业获得博士学位。现任中国科学院信息工程研究所研究员。研究领域为物联网安全、工控安全。研究兴趣包括: 工控系统漏洞挖掘与关联、在线设备发现与识别、工控系统入侵诱捕与行为分析、工控系统入侵检测与监管。Email: sunlimin@iie.ac.cn