http://bhxb.buaa.edu.cn jbuaa@buaa.edu.cn
DOI: 10.13700/j.bh.1001-5965.2022.0592

一种自动实时的物联网在野漏洞攻击检测方法

何清林[1,2]，王丽宏[2,*]，陈艳姣[3]，王星[4]

(1. 国家互联网应急中心，北京 102299；2. 北京航空航天大学 计算机学院，北京 100191；3. 浙江大学 电气工程学院，杭州 310007；4. 西北工业大学 网络空间安全学院，西安 710072)

摘 要：与互联网相连的海量物联网（IoT）设备容易被黑客攻击和利用，进而造成关键IoT应用的瘫痪。漏洞利用是一种常用的针对IoT设备的攻击方式，然而由于在野的漏洞利用形式多样、变异性和伪装性强，如何快速自动识别针对IoT设备的在野漏洞攻击极具挑战。为此，提出一种基于混合深度学习判别和开源情报关联的IoT漏洞攻击检测方法，所提检测方法可以实时判别网络流量中的IoT在野漏洞攻击行为，并且能够精准识别漏洞攻击行为的具体类别。实验结果表明：所提检测方法在大规模数据集上的判别准确率超过99.99%。所提检测方法在真实场景中应用效果显著，在不到1个月时间内发现了13种新的在野漏洞攻击。

关 键 词：物联网；在野漏洞利用；攻击检测；混合深度学习；开源情报

中图分类号：TP393.4; TP312

文献标志码：A **文章编号**：1001-5965(2024)07-2195-11

与传统设备相比，物联网(internet of things, IoT)设备更容易成为攻击的目标，主要有以下原因：①IoT设备如摄像头、小型网关路由器等，由于成本限制很少考虑安全防护功能，导致其设备存在大量漏洞，根据绿盟科技发布的《物联网2020年安全年报》[1]显示，网络空间中每天发现的针对各类IoT设备的漏洞攻击次数达数亿次之多；②互联网上存在大量IoT开源漏洞信息平台，如通用漏洞披露（common vulnerabilities and exposures, CVE）[2]、Exploits-DB[3]和Packet-Storm[4]等，这些信息很容易被黑客利用，迅速集成到攻击工具和恶意样本中，增加IoT设备遭受攻击的风险。

攻击者大量使用漏洞攻击方式，远程入侵IoT设备，进一步获取设备控制权限、植入木马或窃取敏感信息等。如何及时发现和精准识别各类IoT设备的漏洞攻击行为，成为IoT安全研究的一个迫切问题。传统的检测方法主要依靠专家知识，根据不同的漏洞攻击详细信息，人工提炼形成snort[5]或yara[6]等恶意行为检测规则，部署到各类流量入侵检测设备中。这种方法对已知漏洞的识别准确率高，但是无法应对快速变异的漏洞攻击方式，具有一定的滞后性。因此，急需一种不依赖专家人工经验的漏洞攻击检测方法，能够实时、自动识别网络流量中的漏洞攻击行为，并且能够精准定位漏洞攻击的具体类别。

为解决该问题，本文提出一种基于混合深度学习判别和开源情报关联的IoT漏洞攻击检测方法，该方法设计了一种新的基于混合深度学习的判定算法，并融合了开源情报提取关联技术，能够从实时的网络流量中自动识别和精准定位各种针对IoT设备的在野漏洞攻击行为。本文的主要贡献如下：

1）标注了一套面向IoT漏洞攻击检测的报文数据集。对常见的在野IoT漏洞网络攻击行为模式进行系统性梳理，通过人工标记形成通用的检测规则，基于该检测规则从真实的网络流量中萃取疑

收稿日期：2022-07-05；录用日期：2022-09-02；网络出版时间：2023-03-29 13:00
网络出版地址：link.cnki.net/urlid/11.2625.V.20230329.1041.004
*通信作者. E-mail：hql@cert.org.cn

引用格式：何清林，王丽宏，陈艳姣，等. 一种自动实时的物联网在野漏洞攻击检测方法[J]. 北京航空航天大学学报，2024，50(7)：2195-2205. HE Q L，WANG L H，CHEN Y J，et al. An automatic and real-time detection method of IoT in-the-wild vulnerability attack[J]. Journal of Beijing University of Aeronautics and Astronautics，2024，50(7)：2195-2205 (in Chinese).

似攻击报文，并通过专家经验验证将这些报文标记为正样本（真正的漏洞攻击报文）和负样本（正常非恶意报文）。该数据集包括 5×10^5 条攻击报文和 4×10^6 条正常报文。

2）针对漏洞攻击自动识别难题，提出了基于深度自注意力变换网络和卷积神经网络（convolutional neural networks, CNN）的混合深度学习模型。该模型将疑似攻击报文输入 Transformer 网络进行语义学习和表征，再将关键表征结果输入 CNN 模型进行分类。该模型在测试数据集上的 F_1 值达到了 99.98%，证明其可以有效自动判别实时输入的疑似攻击报文是否真正的漏洞攻击报文。

3）针对漏洞攻击精准定位难题，提出基于攻击向量回归的开源漏洞情报关联方法。对 Exploits-DB[3] 等 10 个主流开源漏洞情报平台的信息进行了实时提取和分析，并与贡献 2 中本文学习模型自动判别出来的漏洞进行自动关联，实现了对漏洞攻击类别的精准定位。

4）基于本文检测方法研发了一套 IoT 漏洞实时检测系统 IoT_Exploits_Founder，并在真实环境中进行了部署，系统运行效果显著。在一个月内，IoT_Exploits_Founder 系统从流量中实时检测和精准定位出了 13 类新的 IoT 在野漏洞攻击方式，识别出数万个漏洞攻击报文。整个检测过程不依赖于领域专家的人工参与，以自动实时的方式输出结果，为解决 IoT 漏洞入侵检测问题提供了新的有效手段和工具。

1 攻击检测相关工作

IoT 在野漏洞攻击检测本质上属于一种网络入侵检测，即从网络流量中检测出针对 IoT 的漏洞攻击流量，而针对网络入侵检测的研究历史可以追溯到 20 世纪 90 年代。1999 年知识发现与数据挖掘（knowledge discovery and data mining, KDD）会议发布了网络入侵检测领域的标准数据集 KDD99[7]，其覆盖了 Probe、DoS、R2L、U2R 和 Data5 大类 58 种典型的网络攻击方式，是目前引用率较高的入侵检测数据集。该数据集的发布开启了基于异常的网络入侵检测研究时代，与传统的基于已知规则的入侵检测技术相比，基于异常的入侵检测不需要人工规则，可实现自动的网络恶意入侵行为检测。

早期的基于异常的入侵检测研究[8-12] 主要集中在如何提取流量的关键特征，如统计行为特征和传输控制协议特征等，将特征进行向量化表征后采用传统机器学习方法，如层次聚类、支持向量机（support vector machine, SVM）和朴素贝叶斯等方法，进行入侵检测行为判定。Khan 等[11] 提出了将 SVM 和层次聚类相结合的网络流量入侵异常行为检测，采用动态生长的自组织树（dynamical growing self-organizing tree, DGSOT）算法实现更有效的聚类，这在处理大数据集时非常有效。

随着移动互联网和 IoT 技术的发展，网络流量变得更为丰富多样，网络入侵行为也变得更为复杂，文献 [13-14] 对基于异常的入侵检测方法的局限性进行了总结，指出了其中存在的数据集缺乏、不平衡分类、效果难以评估等问题。入侵检测研究开始由通用场景转向针对特定场景的异常行为检测研究，如恶意统一资源地址（uniform resource locator, URL）检测、恶意木马流量检测和恶意超文本传输协议（hypertext transfer protocol, HTTP）流量检测等[15-19]。在这些研究中，除了流量本身的特征外，还会利用更多外部关联的信息，如域名查询信息和网络地址信息等，在特征提取和表征方面，也用到了主动学习、深度学习和混合模型等方法。文献 [18] 使用 CNN 与多层感知机结合的混合结构深度神经网络，分别处理文本与统计信息，从而检测恶意 HTTP 流量，并标注了一套基于 HTTP 的流量数据集，包含 4.5×10^5 条以上恶意流量和 2×10^8 条以上非恶意流量。

近年来，针对 IoT 的入侵检测，开始成为一个研究热点。Thamilarasu 和 Chawla[20] 提出了一个实时处理 IoT 交通数据的集中系统，该系统首先用无监督方式训练深度置信网络（deep belief networks, DBN），然后，使用节点神经元构建标记数据，用来训练深度神经网络（deep neural networks, DNN），从而缩短模型的整体训练时间，该系统在包含 5 类攻击流量和正常流量的模拟合成数据集上的 F_1 得分为 99%。Al-Hawawreh[21] 等提出了一种基于自动编码器和前馈神经网络的工业 IoT 入侵检测模型，通过 TCP/IP 数据包进行可疑活动检测和分类，在标注数据集上该系统的准确率分别为 98.6% 和 92.4%。文献 [22] 针对 IoT 交通数据中存在的入侵检测问题，提出了一种分层半监督训练方法，该方法利用训练过程中的顺序特征，并在公开数据集中进行了实证评估。文献 [23] 对基于深度学习的 IoT 入侵检测方法进行了综述，指出 IoT 是技术和工程的下一个前沿领域，网络安全是 IoT 面临的主要挑战，而深度学习可能是解决该问题的最佳解决方案。

深度学习在各领域的快速发展和成功应用为解决 IoT 入侵检测问题指出了一个新的方向。但是该研究领域仍面临标准数据集缺乏、入侵场景复杂多变等问题，目前仅有 KDD99[7]、ISCX 2012[8] 和

CIDDS-001[9] 等少数可用验证检测方法有效性的标准数据集, 很多研究工作仍使用模拟数据进行验证, 已有网络入侵检测研究工作仅进行恶意行为和正常行为的二元判别, 不能对恶意行为的具体类别进行判定。针对这些问题, 本文从实际场景出发, 通过对真实 IoT 漏洞攻击进行人工标记的办法, 形成了一套面向 IoT 漏洞攻击检测的报文数据集, 为业界和学术界验证相关方法提供了标准数据集, 进一步基于本文开源漏洞情报关联方法, 实现了漏洞攻击类型的精准定位。

2　检测方法概述

2.1　物联网设备漏洞攻击现状分析

漏洞攻击是一种利用设备软硬件存在的缺陷发起的网络入侵行为, 其主要目标是获得设备控制权限、植入样本、窃取数据等。漏洞攻击已成为网络入侵的主要攻击手法。与传统网络设备相比, IoT 设备更容易成为漏洞攻击的目标, 其原因主要有 2 点。首先, IoT 设备大量采用各种开源组件和服务, 本身存在大量的漏洞; 其次, IoT 设备一般都是规模化部署, 具有同类型漏洞的设备在互联网上大量存在, 因而很容易成为各类黑产人员的目标。

例如, 针对 NetLink GPon 设备的漏洞, 2020 年 3 月 18 日在 Exploits-DB[3] 平台上披露, 2020 年 3 月 19 日已由黑客组织集成到 Gafgyt 僵尸网络工具中; 针对路由器组件的 CVE-2021-20 090 漏洞[24], 2021 年 8 月 3 日由 Tenable 公司安全研究员首次披露, 2021 年 8 月 14 日已由黑客组织集成到 Mirai 僵尸网络工具中, 如图 1 所示。从上述案例可以看出, 漏洞信息披露时间和漏洞被大规模利用时间之差越来越短。根据公开发布的报告显示, 有漏洞的设备一旦被部署到互联网上, 其平均被攻陷的时间已经从以天为单位缩短到以小时为单位。

随着网络攻击对抗的日趋激烈, 时效性和准确性成为 IoT 设备漏洞攻击检测方法的关键技术指标。而传统的依靠专家经验进行人工分析和总结提炼规则进行识别的检测方法已经无法满足实际防御场景的需求。

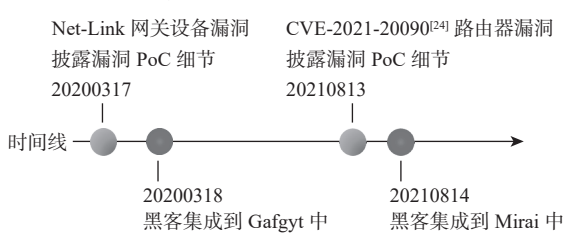


图 1　IoT 漏洞信息披露和在野利用发现

Fig. 1　IoT vulnerability information disclosure and inofficial exploits discovery

在漏洞攻击中, 攻击行为从发现到解决一般有 3 个关键的时间节点, 分别是: ①漏洞攻击行为第 1 次被人发现的时刻 T_0; ②漏洞攻击的细节被公开披露的时刻 T_1; ③漏洞被厂商修复的时刻 T_2。若漏洞攻击发生在 T_0 到 T_1 时刻之间, 则将该攻击行为称之为 0Day 漏洞攻击, 这个时候在网络上还没有任何关于此漏洞的公开信息; 若漏洞攻击发生在 T_1 和 T_2 时刻之间, 则称为 1Day 漏洞攻击; 若漏洞攻击发生在 T_2 时刻之后, 则称为 NDay 漏洞攻击。

本文主要解决如何从海量的网络流量中自动和实时识别各类 1Day 和 NDay 漏洞攻击问题。而 0Day 漏洞攻击则无法通过系统自动进行识别, 需要领域专家对漏洞进行人工验证和复现等一系列复杂的步骤。

2.2　本文检测方法框架

本文检测方法能够从大规模网络流量中检测并识别各类针对 IoT 设备的漏洞攻击行为。该方法融合了深度混合学习模型和开源情报提取关联技术, 在不依赖安全专家领域知识的条件下, 可以准确快速识别漏洞攻击。

本文检测方法主要由以下 3 个部分组成: 报文预处理、混合深度学习判定和开源情报关联, 如图 2 所示。

1) 报文预处理。根据某骨干网络真实流量测算, IoT 漏洞利用攻击报文在实际互联网流量中占比极低(小于 0.01%), 漏洞攻击检测是一个极端的不平衡分类学习问题。如果直接将原始未经处理的网络流量数据输入学习模型进行分类, 将会造成很高的漏报率。首先, 详细梳理 IoT 漏洞攻击常用范式并形成一套通用检测规则; 其次, 基于该检测规则从海量流量中萃取疑似漏洞攻击报文, 以疑似报文作为输入, 解决不平衡分类问题; 最后, 对疑似报文的格式与内容进行预处理, 以满足下一步学习任务和关联任务的需求。

2) 混合深度学习判定。传统基于一般特征提取和简单机器学习模型的检测方法, 无法准确提取漏洞攻击报文的关键特征, 无法有效学习攻击特征随时间不停变换的特性。因此, 本文提出了基于深度自注意力变换网络和 CNN 网络的混合深度学习模型, 该模型根据注意力机制提取报文的关键表征信息, 在此基础上实现快速自动的攻击报文识别。在本文模型训练阶段, 使用包含 4×10^6 个正常报文和 5×10^5 个 IoT 漏洞攻击报文的训练数据集进行训练; 在模型判定阶段, 将经过数据预处理后的疑似攻击报文输入该模型中, 来预测该报文是否是真正的漏洞攻击报文; 在模型更新阶段, 将后续正确识别出的新的漏洞攻击报文补充到训练数据集中, 使

用离线的方式来更新训练本文模型。

3)开源情报关联。经过前面判定后的疑似攻击报文,虽然能够明确是一种IoT漏洞攻击行为,但无法定位出具体的漏洞类型,因此,需要使用开源威胁情报辅助漏洞攻击的精准识别。本文跟踪了10个主流开源漏洞平台,这些平台基本覆盖了所有已公开的漏洞概念验证(proof of concept, PoC)信息,通过实时抓取这些平台上的漏洞PoC信息,对其进行范式处理后,与判定为正的攻击报文进行回归关联。若关联成功则确定为1Day或Nday漏洞攻击;否则可能是疑似0Day攻击,需要进一步的人工分析和确认。

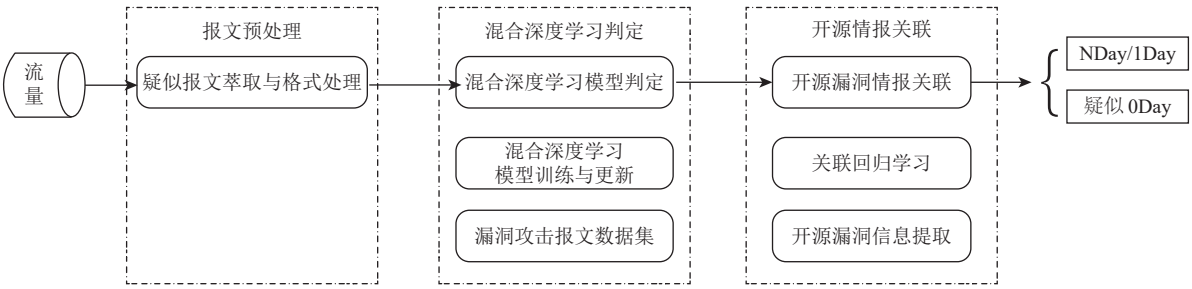


图2 本文检测方法框架

Fig. 2 Framework of the proposed detection method

综上,本文实现了一个端到端的IoT漏洞攻击识别方法,能够自动从背景流量中检测、识别和判定各种针对物联网的漏洞攻击行为,并能进一步精确地识别1Day/NDay漏洞攻击的具体类型,以及发现疑似0Day攻击行为。该方法同时加强了漏洞攻击识别的时效性和准确性,而且不依赖于人工专家经验的干预。最后,根据本文检测方法实现了一套IoT漏洞攻击识别系统IoT_Exploits_Founder,并且在真实环境中实际运转1个月,发现了13种新的在野漏洞攻击,识别效果显著。

3 数据预处理

3.1 攻击报文萃取

本文要从实时的网络流量中检测IoT漏洞攻击报文,然而在真实的网络流量中,真正的IoT漏洞攻击报文所占的比例极少。如果直接从流量中通过学习算法训练和判定哪些是真正的漏洞攻击报文,将是一个极端不平衡的学习分类问题,在实际场景中无法实现。因此,需要先进行数据预处理,将所有可疑的IoT漏洞攻击报文从流量萃取出来;然后,再利用学习算法训练和判定,将极端不平衡学习分类问题转换成一个普通的学习分类问题。经过数据预处理后,大量的正常报文将从流量中被剔除出去,留下的是疑似攻击报文。作为正例的漏洞攻击报文和作为负例的正常报文,其比例在预处理之前低于$1/10^5$,经过预处理后,变为1/10,这极大地缓解了数据集的不平衡问题。

首先,本文对大量已知IoT漏洞攻击报文的格式类型进行梳理,发现标准HTTP请求是目前已知IoT漏洞攻击类型的唯一格式,通过向IoT设备发起精心构造的HTTP请求报文,达到控制和利用设备的目的,对可疑攻击报文萃取时本文仅仅针对标准HTTP格式的报文进行萃取。其次,对各类IoT漏洞攻击报文的恶意行为分析后发现,针对不同IoT设备的漏洞攻击行为,一般都具有通用的恶意行为特征和范式:①大部分有远程命令执行、恶意样本植入、行为隐藏等恶意行为,而这些通用的恶意行为特征,都会在漏洞攻击的流量报文中出现;②这些具有漏洞的IoT设备操作系统,大部分基于linux kernal系统。因此,这些通用的恶意行为具有相同的范式,比如“wget xxx; ./xxx; rm xxx”就是一种典型的漏洞攻击行为,包括了样本植入下载、样本执行和痕迹删除等操作。最后,本文整理总结出整套通用的IoT漏洞利用行为特征规则,可以从大规模网络流量中萃取出所有可疑的IoT漏洞攻击报文,大大减少了漏洞攻击识别的范围。但预处理后的报文误报率仍然太高,且无法精准定位漏洞攻击的类别,因此,需要进行深度学习判定和开源情报关联。

3.2 报文预处理

经过萃取后的可疑漏洞攻击报文,还需要进行进一步的报文格式预处理。首先,对报文的字符集进行处理,将报文中出现的各种转义字符和计算机编码后的字符进行统一的转换处理,转换成标准可见字符集形式的文本信息;其次,对报文的格式进行处理,本文中的攻击报文仅限于标准的HTTP格式报文,一般具有如图3所示的格式。

请求行。在目前已知的漏洞攻击报文中,请求方法有HTTP GET和HTTP POST这2类。URL代表被攻击设备的访问路径,其标记了设备上的程序或组件的访问入口,是漏洞攻击中最关键的字段之

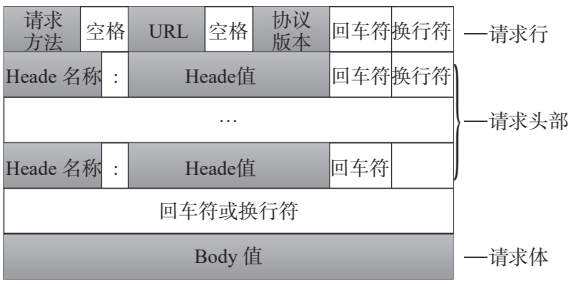


图 3　HTTP 请求报文典型格式

Fig. 3　Typical format of HTTP request messages

CVE_2021_20090
POST/images/../apply_abstract.cgi HTTP/1.1
Connection: keep-alive
User-Agent: Dark
action=start_ping&submit_button=ping.html&action_params=blink_time=5&ARC_ping_ipaddress=212.192.241.7%0AARC_SYS_TelnetdEnable=1&%0AARC_SYS_=cd+/tmp;wget xxxx;curl-O xxx;chmod+777+lolol.sh;sh + lolol.sh&ARC_ping_status=0&TMP_Ping_Type=4

CVE_2020_13872
GET/portal/_ajax_explorer.sgi?action=umnt&path=path&where=here&en=;cd+/tmp+rm +−rf+*;wget+1.1.1.1;chmod+777+fetch.sh;sh+fetch.sh;HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept:∧
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0

图 4　2 个 IoT 漏洞攻击的报文示例

Fig. 4　Request message examples of two IoT exploits

一。如果请求方法是 HTTP GET, 则 URL 中可以带参数, 因此, 很多漏洞攻击会选择 URL 里的参数入口, 如图 3 中的 CVE_2020_13872 漏洞攻击, 就利用了 URL 的参数调用植入了恶意命令; 如果请求方法是 HTTP POST, 则 URL 中一般不带参数。在对漏洞攻击报文进行预处理时, 先提取出 URL, 再对一些可变的参数值进行统一处理。

请求头部。正常的 HTTP 请求, 一般由浏览器发出, 其请求头部及属性值一般也是由浏览器自动填入; 而漏洞攻击报文, 则一般是由攻击者通过编写攻击程序构造并发送, 其攻击报文中的请求头部值也是由攻击者填写。常见的请求头部类别一般不超过 10 个, 而且大部分都与具体漏洞利用关联性很小。因此, 本文在做数据预处理时, 便将大部分的请求头部字段直接丢弃。值得注意的是, 有一些攻击者在填充请求头部信息时, 会加入一些特殊字符串, 如图 4 中的 CVE_2021_20090 漏洞攻击中, 攻击者在“User-Agent”填入了特殊的“Dark”字符信息, 这些信息可以用于追踪和定位攻击者信息。

请求体。该部分是 HTTP 报文的主要内容, 大部分漏洞攻击的命令都会在 Body 植入, 如图 4 所示: CVE_2021_20090 漏洞攻击的命令植入请求体中, 请求体的内容直接被提取出来, 没有其他的预处理步骤。

报文预处理这一步的主要任务是将报文中的请求方法、处理后的 URL 字段、部分请求头部字段及请求体字段分别提取出来, 直接进行拼接后, 得到可疑的攻击报文文本 X, 作为后续学习判定和关联识别的输入。

4　混合深度学习检测判定

4.1　漏洞攻击报文数据集介绍

本文中的 IoT 漏洞攻击数据集, 全部是从某互联网出入口捕获的真实报文, 报文的捕获时间为 2021 年 4 月～10 月, 均为标准的 HTTP 格式报文, 且都经过了第 3 节的数据预处理操作, 其中包含正例样本和负例样本。

正例样本。总数为 5×10^5 个不同的 HTTP 报文, 是通过专家编写的 yara 格式的漏洞攻击检测规则, 从网络流量中检测和标记的真实 IoT 漏洞攻击报文, 覆盖了 197 种 IoT 漏洞攻击行为, 表 1 列出了最近 3 年的部分在野利用 IoT 漏洞类别。这些漏洞均为远程命令注入、命令执行等能够控制设备的高

表 1　最近 3 年部分在野利用 IoT 漏洞示例

Table 1　Some inofficial IoT exploit examples in past three years

CVE编号	漏洞名称	针对设备
CVE_2021_33544	UDP_Technology_Geutebruck_IP_Cameras_Command_Injection	IP摄像头
CVE_2021_33514	HTTP_Router_NetgearGC108P	Netgear路由器
CVE_2021_31755	Tenda_AC11_Router_Stack_Buffer_Overflow_Vulnerability	腾达路由器
CVE_2021_28799on	QNAP_NAS_Hybrid_Backup_Sync_Command_Injecti	QNAP NAS设备
CVE_2021_20090	Arcadyan_Buffalo_Routers_Configuration_File_Injection	Buffalo组件的路由器
CVE_2021_1497	Cisco_HyperFlex_HX_Data_Platform_Command_Execution	Cisco管理平台
CVE_2020_9054_	Zyxel_NAS_RCE_Attempt_Inbound	Zyxel NAS设备
CVE_2020_8949	Gocloud_Router_Remote_Code_Execution	Gocloud路由器
CVE_2020_8515	DrayTek_Vigor_RCE	Vigor路由器
CVE_2020_35713	Linksys_RE6500_1_0_11_001_Remote_Code_Execution	Linksys路由器
CVE_2020_35576	TpLink_TLWR841N_Command_Injection	Tplink路由器
CVE_2020_17456	Seowon_Route	Seowon路由器
CVE_2020_13872	DLink_DIR_865L_Ax120B01_Command_Injection	Dlink路由器
CVE_2020_10987	Tenda_AC15_AC1900_goform_setUsbUnload_RCE	腾达路由器

危类型漏洞，漏洞覆盖的设备类型主要为各种路由器设备、摄像头设备，及相关设备的通用组件等，其中含有CVE编号的漏洞有117类，剩下的80类漏洞均在Exploits-DB[3]网站上有披露，其完整漏洞类别信息已经发布到GitHub网站[25]。

负例样本。总数为4×10^6个不同的HTTP报文，先通过流量萃取规则从网络流量中筛选出可疑漏洞攻击报文，再通过白名单机制将疑似攻击报文中的正常报文标记出来，作为数据集中的负例样本。

4.2 混合深度学习模型

文献[26]综合比较了目前各种分类任务中用到的分类学习模型，总结出表现最为优异的3个分类学习模型分别是CNN[27]、Transformer[28]和多层感知器(multilayer perceptron, MLP)模型，本文模型是一个Bert(bidirectional encoder representation from transformers)[29]和CNN[27]的混合深度学习模型，如图5所示，该模型用于对漏洞攻击报文的分类实验，其中，C为文本的上下文信息，T为文本的单个元素信息。

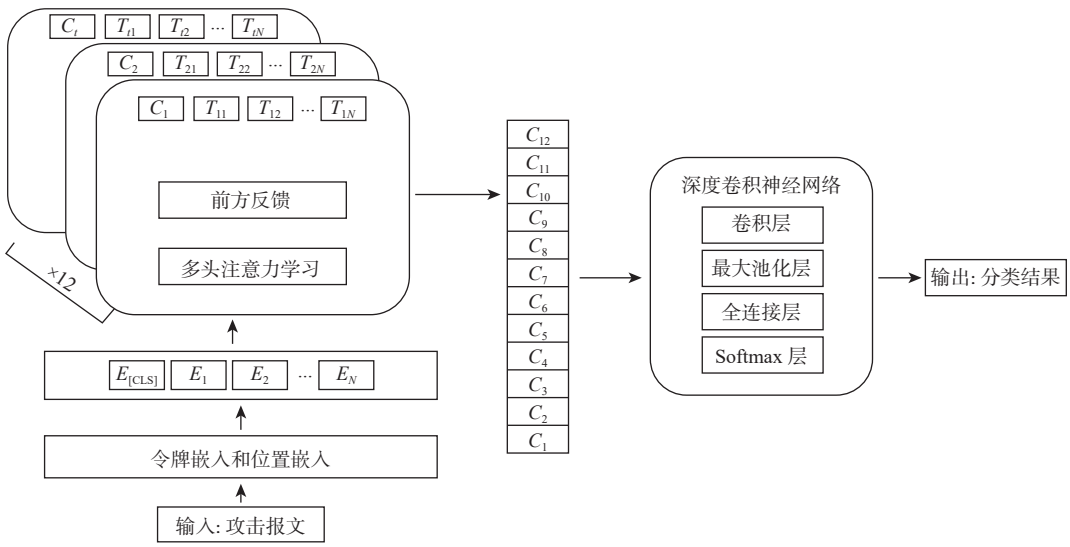


图5 本文模型结构

Fig. 5 Structure of the proposed model

Google团队于2017年提出了Transformer[28]模型，该模型与传统深度学习模型CNN[27]不同，其直接采用了注意力机制处理文本，避免了在文本摘要编码过程中按照一定顺序生成向量，因此，该模型有更强的并行计算能力。2018年，Google团队在Transformer[28]模型的基础上提出了Bert[29]模型，采用掩码语言模型(masked language model, MLM)对Transformer[28]模型进行预训练，生成了更深层次的双向表征，该模型在多项自然语言处理(neuro-linguistic programming, NLP)任务中均取得了优异的成绩。

在将文本信息向量输入Bert[29]模型时，会在首部加一个标志位，最后一层相应位置的向量可以用作整段文字的语意表示，用于分类任务。Bert[29]模型的self-attention部分会用文本中的其他词来增强目标词的语意表示，但是目标词本身的语意还是会占主要部分。但是额外增加的标志位本身没有语义，所以相比原有的字词，标志位可以更加公平地融合文本各个字词的语义信息，更好地表征语义。

1) CNN模型[27]。CNN起初主要用来处理图像任务，Kim[27]在2014年提出了TextCNN模型，即将CNN[27]模型应用到文本分类任务中。本文模型中使用CNN[27]模型结构，其具体结构如图5右侧所示，最大输入分别经过卷积层、最大池化层、全连接层、softmax层，最后输出分类结果。在CNN[27]模型的优化方面，激活函数选用Relu，优化器选用Adam，为更好地捕捉文本的局部相关性，提取文本的关键信息，卷积核大小分别选择了3、4、5这3种尺寸。

2) 损失函数。交叉熵主要用来判断实际输出和期望输出的接近程度，交叉熵越小说明实际输出与期望输出的差距越小。假设概率分布q为实际输出，概率分布p为期望输出，则交叉熵为

$$H(p,q)=-\sum p(x)\ln q(x) \tag{1}$$

3) 本文模型。考虑到漏洞攻击报文的文本属性特点，设计本文模型来判定可疑攻击报文是否漏洞攻击报文，本文模型结构如图5所示。首先，将每个预处理后的可疑漏洞报文的文本信息X作为

一个单独的句子, 进行 Token Embedding 和 Position Embedding 后输入 Bert[29] 模型; 其次, 将 12 个隐藏层的 [CLS] 向量拼接后, 作为 CNN[27] 模型的输入, 通过卷积层、最大池化层、全连接层 softmax 层; 最后, 输出分类结果。

4.3　评价指标

按照分类算法评价指标的惯例做法, 使用精确率 P、召回率 R、准确率 A_{CC} 和 F_1 值作为混合深度学习模型的分类效果的评价指标, 其表达式分别为

$$R = T_P/(T_P + F_N) \tag{2}$$

$$P = T_P/(T_P + F_P) \tag{3}$$

$$A_{cc} = (T_P + T_N)/(T_P + T_N + F_P + F_N) \tag{4}$$

$$F_1 = 2PR/(P + R) \tag{5}$$

式中: T_P 为被正确标记的正例样本; F_P 为被错误标记的正例样本; F_N 为被错误标记的负例样本; T_N 为被正确标记的负例样本。

4.4　分类实验结果及分析

经过报文预处理得到的可疑漏洞攻击报文文本 X, 将作为学习分类模型的输入, 以确定哪些是真正的 IoT 漏洞攻击报文。根据实际数据统计结果, 在可疑漏洞攻击报文中真正的漏洞攻击报文比例在 0.1～1 之间, 因此, 选择 1∶1、1∶2、1∶4、1∶8 这 4 种正负样本的比例, 来进行不同的分类实验。在分类实验同时对比使用本文模型和 Bert[29] 模型进行分类验证, 并将每组的训练集和测试集划分为 0.8 和 0.2。

根据不同的正负样本比例和不同模型的选择, 测试集上的分类结果分别如表 2 和图 6 所示。实验结果表明, 本文模型用于漏洞报文检测具有优越的检测性能。首先, 2 个模型的精确率接近 1, 在实际场景中, 精确率 P 是一个更有实用性的指标, 即被标记的漏洞攻击报文中有多少个是真正的攻击报文。因为本文检测方法分类仅仅是其中的一个步骤, 通过分类算法识别出的攻击报文会继续下一步流程, 进行关联和线索发现。如果精确率不高, 则会影响后续的流程; 其次, 2 种不同分类模型的准确率 A_{CC} 和 F_1 值均超过了 99.9%, 特别是在正负样本比例 1∶2 的场景下, 本文模型的准确率超过了 99.99%, 这意味着平均每检测 10^5 个可疑报文, 误报的报文数量仅为个位数, 具备较强的实用价值。最后, 由表 2 可知, 本文模型 F_1 比单个 Bert[29] 模型平均提高了 0.02%, 本文研究的问题是在大规模流量中检测漏洞攻击行为, 而在真实系统中每天发现的疑似攻击漏洞攻击报文数量超过 10^8 个。如果 F_1 值提升 0.01%, 即意味着有 10^5 个报文被正确识别, 反之则有 10^5 个报文被错误识别, 这对于后续发现 NDay 攻击和 0Day 漏洞攻击线索非常重要。所以 0.02% 的提升对实际工作的影响是显著的, 能大大减少误报量从而增强检测系统的实用性。

表 2　F_1 和 A_{CC} 实验结果

Table 2　Experimental results of F_1 and A_{CC}

正负样本比例	F_1/%		A_{CC}/%	
	Bert[29]模型	本文模型	Bert[29]模型	本文模型
1∶1(总数10^6)	99.93	99.94	99.95	99.96
1∶2(总数1.5×10^6)	99.97	99.97	99.99	99.99
1∶4(总数2.5×10^6)	99.93	99.96	99.98	99.98
1∶8(总数4.5×10^6)	99.96	99.97	99.96	99.97

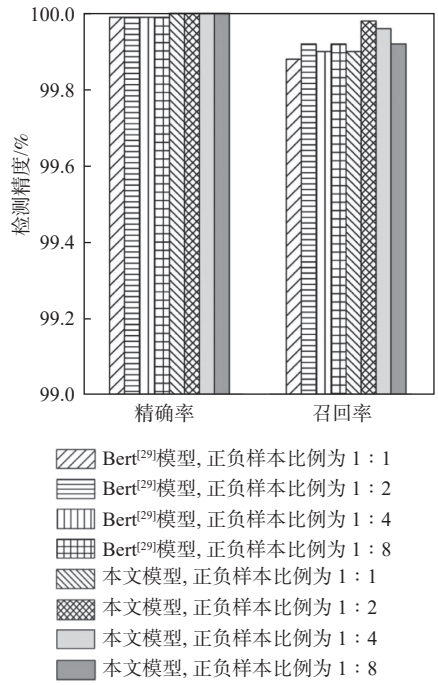


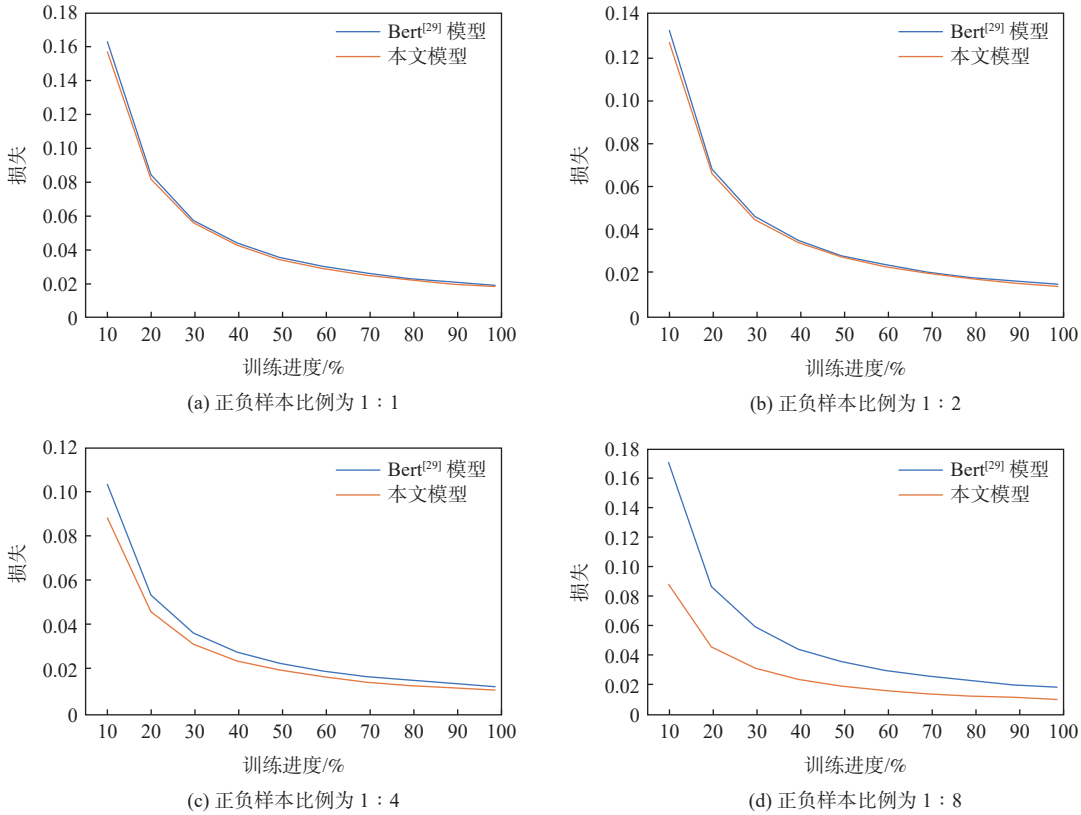
图 6　精准率和召回率实验结果

Fig. 6　Experimental results of precision and recall rate

另外, 由图 7 可知, 本文模型在训练时的损失值收敛效果比单个 Bert[29] 模型更快更好。在模型训练时硬件采用了 GPU 4 000 加速卡进行模型训练。

每一轮训练耗时对比如表 3 所示, 2 种模型的训练耗时都根据样本个数呈现线性变化, 耗时 60～150 min。

通过本文模型对可疑漏洞攻击报文进行分类判定后, 虽然能够确定某个可疑漏洞攻击报文是否真正的漏洞攻击报文, 但仍无法明确该漏洞攻击报文的攻击类型, 需要进一步使用后续的开源情报关联, 来进行漏洞的精确识别。



(a) 正负样本比例为 1∶1　(b) 正负样本比例为 1∶2

(c) 正负样本比例为 1∶4　(d) 正负样本比例为 1∶8

图 7　2 种模型损失值

Fig. 7　Loss of two models

表 3　模型每轮训练时间

Table 3　Model training time in each round

样本数量/10^6	训练时间/min	
	Bert[29]模型	本文模型
1	62	71
1.5	89	102
2.5	127	146
4.5	146	167

5　开源信息关联识别

5.1　开源漏洞信息提取

近年来基于开源威胁情报的网络狩猎研究成果[30]表明，开源威胁情报可应用于支撑威胁行为狩猎。本文关注的开源信息是漏洞利用信息 Exploit（EXP）是如何利用漏洞的详细说明或漏洞攻击代码，详细阐述了漏洞的机制及使用方法。在各类开源漏洞 EXP 信息平台中，Exploit-DB[3]的影响力最大，其是一个面向全世界黑客的漏洞提交平台，及时发布各种最新的漏洞 EXP 信息。除了 Exploit-DB[3]之外，还有 Packet-Storm[4]等知名开源漏洞情报平台，这些平台上也会及时披露各类漏洞 EXP 信息。而在针对各类 IoT 设备发起的漏洞攻击报文中，会具体利用到漏洞 EXP 信息中披露的方法，安全分析人员也需借助这些开源情报网站的漏洞 EXP 信息，来确定某个漏洞攻击报文是不是 1Day 或 NDay 漏洞攻击。

从这些开源网站上爬取的开源漏洞信息，本身是一种开源的威胁情报，但是与传统威胁情报不同的是其并没有标准的描述格式。除了与漏洞相关的描述信息，如发布时间和漏洞对象以外，最关键的信息是 EXP 内容，也就是具体的漏洞利用脚本，一般有 shell、python、php、java 等不同语言的脚本形式。本文对 10 个主流的漏洞 EXP 开源网站进行了实时跟踪，对在这些网站上发布的漏洞 EXP 信息及关联信息进行了实时抓取和存储。截止到 2021 年 10 月，从这 10 个开源情报网站上抓取的各类漏洞信息数量超过 2.5×10^5 条，覆盖了不同的漏洞类型。但是其中真正能用来发动网络攻击、进行命令注入、远程执行的漏洞，只有 1Day 或 NDay 漏洞（0Day 漏洞信息不会公布），均占比较低。

将第 4 节检测出的漏洞攻击报文与这些开源的漏洞信息进一步比对和关联，就能够确认漏洞攻击报文的类别。与以往依赖安全人员人工比对和关联不同，本文提出基于攻击向量回归的开源漏洞情报关联方法，从而对漏洞攻击报文精准识别。

5.2　基于攻击向量回归的漏洞关联识别

不管是从开源漏洞平台爬取的漏洞 EXP 信息，

还是网络流量中的漏洞攻击报文, 与自然语言短文本相比都存在较大差异, 已有的短文本处理工具, 如 NLTK、scikit-learn 主要处理英文及西欧语言, HanLP 则处理中文文本, 这些工具中的大部分函数不能直接应用于处理攻击报文或 EXP。因为攻击报文没有词库的概念, 某些标点符号、单个字符等也是代码执行的一部分, 不能直接根据空格及标点进行分词。因此, 本文设计和编写了专门的命名实体识别模块, 从攻击报文和 EXP 中分别提取关键词表。

对漏洞攻击报文进行信息抽取, 提取出报文中与漏洞利用有关的攻击向量字段, 形成如下结构: $[x_1:\text{attack_target};x_2:\text{attack_para};x_3:\text{attack_command}]$。其中, 攻击目标指漏洞攻击报文请求的具体地址, 即设备被漏洞攻击时触发的具体路径, 一般代表设备上某个组件或服务对外开放的服务接口, 展现设备可以被外部访问的位置; 攻击参数则代表漏洞攻击时触发具体使用的参数。表 4 列出了表 1 中提到的漏洞攻击报文的攻击目标和攻击参数。

表 4　IoT 漏洞攻击中的攻击目标及攻击参数

Table 4　Attack targets and parameters in IoT exploits

漏洞编号	攻击目标	攻击参数
CVE_2021_33544	/uapi-cgi/certmngr.cgi	action,createselfcert,local
CVE_2021_33514	cgi/setup.cgi	token
CVE_2021_31755	/goform/setmac	mac,wifien,wifissid
CVE_2021_28799	/cgi-bin/backup/hbs_mgnt.cgi	run_cmd,jisoosocoolhbsmgnt
CVE_2021_20090	/images/../apply_abstract.cgi	action,submit_button,action_params,arc_ping_ipaddress
CVE_2021_1497	/storfs-asup	action,token
CVE_2020_9054	/cgi-bin/weblogin.cgi	adv,username,password
CVE_2020_8949	/cgi-bin/webui/admin/tools/app_ping/diag_ping/	None
CVE_2020_8515	cgi-bin/mainfunction.cgi	action,keypath,loginPwd
CVE_2020_35713	/goform/setSysAdm	AuthTimeout, admpasshint
CVE_2020_35576	/cgi?	maxhopcount, numberoftries, host
CVE_2020_17456	/cgi-bin/system_log.cgi	Command, traceMode
CVE_2020_13872	/portal/__ajax_explorer.sgi	Action/path/where/en
CVE_2020_10987	/goform/setUsbUnload	setUsbUnload

攻击指令代表了具体的攻击行为, 如样本植入、隐藏痕迹、删除痕迹等, 根据对大量已知漏洞攻击的统计, IoT 漏洞攻击中常见的攻击指令如表 5 所示。

表 5　IoT 漏洞攻击中常见的攻击指令

Table 5　Common attack commands in IoT exploits

攻击指令类型	关键词
样本植入	wget,curl,tftp,fetch
反弹回连	bash-i, tmp/socat exec
探测	Nc,echo,dnslog
隐藏痕迹	Rm,mv,base64,decodeHex
命令执行	chmod, system,md5sum, busybox ,exec

使用攻击报文测试集中的已知漏洞攻击报文和对应的开源漏洞 EXP 信息进行回归计算, 得到关联系数w_1、w_2、w_3、α的值, 计算方式如下:

$$f(x,y)=w_1\text{sim}(x_1,y)+w_2\text{sim}(x_2,y)+w_3\text{sim}(x_3,y)+\alpha \tag{6}$$

$$f(x,y)=\begin{cases}>1 & x,y\text{对应}\\ <0 & \text{其他}\end{cases} \tag{7}$$

式中: x为漏洞攻击报文文本; y为开源漏洞情报中的漏洞利用信息文本; x_1、x_2、x_3为 3 类攻击, 直接从攻击报文中提取; w_1、w_2、w_3、α为关联系数, 通过将攻击报文数据集和开源漏洞信息进行回归计算得到; sim()函数表示某类攻击在报文中和开源漏洞信息中的相似值, 取值范围为 (0～1)。计算方式如下: 攻击报文中的某一类攻击中的关键词列表在开源漏洞信息中出现的比例。以 CVE-2021-20090 所示的漏洞攻击报文为例, 其中攻击目标为: “/images/../apply_abstract.cgi”, 按照路径‘/’拆分成不同的字符串, 如果所有字符串在y中全部出现, 则$\text{sim}(x_1,y)$的值为 1, 如果所有字符串在y中均没有出现, 则$\text{sim}(x_1,y)$为 0。$\text{sim}(x_2,y)$和$\text{sim}(x_3,y)$的计算方法类似。

通过真实数据回归计算得到w_1、w_2、w_3、α, 即可获得攻击报文和开源漏洞信息的关联函数$f(x,y)$。在实际应用中, 当一个攻击报文x需要进行关联时, 其会与所有已知的开源漏洞信息进行关联。判断关联是否成功的依据如下: 首先, 选$f(x,y)$取值最大的那个漏洞类型y; 其次, 检查y对应

的$f(x,y)$是否超过某个阈值，如果超过，则将攻击报文成功关联到漏洞类型y。该阈值一般由专家根据实际场景调整。

6 结 论

1）本文通过深度学习和开源情报关联，进行IoT漏洞攻击检测判定和识别，在测试集上达到了99.99%的检测准确率。根据本文进一步开发出一套实际的在野漏洞攻击检测系统IoT_Exploits_Founder，该系统在真实环境中实际运转了1个月，新发现并识别了13类新的NDay漏洞攻击。

2）本文检测方法对IoT漏洞攻击检测是有效的，由于其针对各类IoT设备的攻击行为呈现出自动化和规模化的特征，整个攻击过程不需要人的参与，因此，可以使用数据驱动的学习和关联方法。

未来的工作将继续探索其他场景的漏洞攻击检测，并将自动学习关联和专家经验相结合，以精确识别各种攻击行为，特别是0Day漏洞攻击。

参考文献（References）

[1] 绿盟科技. 2020物联网安全年报[EB/OL]. (2021-01-08)[2022-05-28]. https://www.nsfocus.com.cn/html/2021/92_0118/147.html.
NSFOCUS. 2020 IoT Security annual report[EB/OL]. (2021-01-08)[2022-05-28]. https://www.nsfocus.com.cn/html/2021/92_0118/147.html(in Chinese).

[2] CVE-CVE [EB/OL]. (2022-05-28) [2022-05-29]. https://cve.mitre.org/.

[3] Exploit-DB - exploits for penetration testers[EB/OL]. (2022-05-28) [2022-05-29]. https://www.exploit-db.com/.

[4] Packet storm-exploits the possibilities[EB/OL]. (2022-05-29) [2022-05-30]. https://packetstormsecurity.com/.

[5] Snort - network intrusion detection & prevention system[EB/OL]. (2022-05-01) [2022-05-30]. https://www.snort.org/.

[6] Yara-the pattern matching swiss knife for malware researchers[EB/OL]. (2022-05-01) [2022-05-30]. https://virustotal.github.io/yara/.

[7] KDD cup 1999 data [EB/OL]. (2000-09-18) [2022-05-30]. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[8] SHIRAVI A, SHIRAVI H, TAVALLAEE M, et al. Toward developing a systematic approach to generate benchmark datasets for intrusion detection[J]. Computers and Security, 2012, 31(3): 357-374.

[9] RING M, WUNDERLICH S, GRUEDL D, et al. Technical report cidds-001 data set[EB/OL]. (2017-04-28) [2022-05-30]. https://www.hs-coburg.de/fileadmin/hscoburg/Forschung/WISENT_cidds_Technical_Report.pdf.

[10] LEE W K, STOLFO S J. Data mining approaches for intrusion detection[C]//Proceedings of the Conference on USENIX Security Symposium . New York: ACM, 1998: 6.

[11] KHAN L, AWAD M, THURAISINGHAM B. A new intrusion detection system using support vector machines and hierarchical clustering[J]. The VLDB Journal, 2007, 16(4): 507-521.

[12] NGUYEN TTT, ARMITAGE G. A survey of techniques for Internet traffic classification using machine learning[J]. IEEE Communications Surveys & Tutorials, 2008, 10(4): 56-76.

[13] SOMMER R, PAXSON V. Outside the closed world: On using machine learning for network intrusion detection[C]//Proceedings of the IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 2010: 305-316.

[14] SUTHAHARAN S. Big data classification[C]//Proceedings of the Measurement and Modeling of Computer Systems. New York: ACM, 2014, 41(4): 70-73.

[15] MA J, SAUL L K, SAVAGE S, et al. Identifying suspicious URLs: An application of large-scale online learning[C]//Proceedings of the Annual International Conference on Machine Learning. Montreal : ICML , 2009: 681-688.

[16] MA J, SAUL L K, SAVAGE S, et al. Byond blacklists: Learning to detect malicious web sites from suspicious URLs[C]//Proceedings of the Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2009: 1245-1254.

[17] ZHAO P L, HOI S C H. Cost-sensitive online active learning with application to malicious URL detection[C]//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2013: 919-927.

[18] 李佳, 云晓春, 李书豪, 等. 基于混合结构深度神经网络的HTTP恶意流量检测方法[J]. 通信学报, 2019, 40(1): 24-33.
LI J, YUN X C, LI S H, et al. HTTP malicious traffic detection method based on hybrid structure deep neural network[J]. Journal on Communications, 2019, 40(1): 24-33(in Chinese).

[19] HODO E, BELLEKENS X, HAMILTON A, et al. Threat analysis of IoT networks using artificial neural network intrusion detection system[C]//Proceedings of the International Symposium on Networks, Computers and Communications. Piscataway: IEEE Press, 2016: 1-6.

[20] THAMILARASU G, CHAWLA S. Towards deep-learning-driven intrusion detection for the Internet of Things[J]. Sensors, 2019, 19(9): 1977.

[21] AL-HAWAWREH M, MOUSTAFA N, SITNIKOVA E. Identification of malicious activities in industrial Internet of Things based on deep learning models[J]. Journal of Information Security and Applications, 2018, 41: 1-11.

[22] ABDEL-BASSET M, HAWASH H, CHAKRABORTTY R K, et al. Semi-supervised spatiotemporal deep learning for intrusions detection in IoT networks[J]. IEEE Internet of Things Journal, 2021, 8(15): 12251-12265.

[23] TSIMENIDIS S, LAGKAS T, RANTOS K. Deep learning in IoT intrusion detection[J]. Journal of Network and Systems Management, 2021, 30(1): 8.

[24] CVE-2021-20090[EB/OL]. (2022-06-14) [2022-06-15]. https://medium.com/tenable-teblog/bypassing-authentication-on-arcad-an-routers-with-cve-2021-20090-and-rooting-some-buffalo-ea1dd30980c2.

[25] IoT_Exploits_Founder[EB/OL]. (2022-1-12) [2022-06-15]. https://github.com/bennyhee/IoT_Exploits_Founder.git.

[26] ZHAO Y C, WANG G T, TANG C X, et al. A battle of network structures: an empirical study of CNN, transformer, and MLP

[EB/OL]. (2021-08-30) [2022-06-17]. http://arxiv.org/abs/2108.13002.

[27] KIM Y. Convolutional neural networks for sentence classification[EB/OL]. (2014-08-25) [2022-06-15]. http://arxiv.org/abs/1408.5882.

[28] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[EB/OL]. (2017-06-12) [2022-06-15]. http://arxiv.org/abs/1706.03762.

[29] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[EB/OL]. (2018-10-11) [2022-06-15]. http://arxiv.org/abs/1810.04805.

[30] 崔琳, 杨黎斌, 何清林, 等. 基于开源信息平台的威胁情报挖掘综述[J]. 信息安全学报, 2022, 7(1): 1-26.
CUI L, YANG L B, HE Q L, et al. Survey of cyber threat intelligence mining based on open source information platform[J]. Journal of Cyber Security, 2022, 7(1): 1-26(in Chinese).

An automatic and real-time detection method of IoT in-the-wild vulnerability attack

HE Qinglin[1, 2], WANG Lihong[2, *], CHEN Yanjiao[3], WANG Xing[4]

(1. CNCERT/CC, Beijing 102299, China;
2. School of Computer Science and Engineering, Beihang University, Beijing 100191, China;
3. College of Electrical Engineering, Zhejiang University, Hangzhou 310007, China;
4. School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: The vast number of Internet-connected internet of things (IoT) devices are susceptible to hacking and exploitation, which can lead to the paralysis of critical IoT applications. Vulnerability exploitation is a common method of attack on IoT devices; however, due to the diverse, mutable, and highly disguised forms of in-the-wild vulnerability exploitations, it is extremely challenging to quickly and automatically identify ongoing vulnerability attacks targeting IoT devices. To address this, a detection method for IoT vulnerability attacks based on a hybrid deep learning discrimination and open-source intelligence correlation is proposed. This detection method can identify IoT in-the-wild vulnerability attack behaviors in network traffic in real-time and accurately identify the specific categories of vulnerability attack behaviors. Experimental results show that the proposed detection method achieves an accuracy rate of over 99.99% on large-scale datasets. The application of the proposed detection method in real-world scenarios has been significant, discovering 13 new in-the-wild vulnerability attacks within less than a month.

Keywords: internet of things; in-the-wild vulnerability exploitation; attack detection; hybrid deep learning; open-source intelligence

Received: 2022-07-05; **Accepted**: 2022-09-02; **Published Online**: 2023-03-29 13:00
URL: link.cnki.net/urlid/11.2625.V.20230329.1041.004

* **Corresponding author**. E-mail: hql@cert.org.cn