计算机科学
COMPUTER SCIENCE

基于时延特征的网络设备异常检测

崔竞松, 张童桐, 郭迟, 郭文飞

引用本文

崔竞松, 张童桐, 郭迟, 郭文飞.基于时延特征的网络设备异常检测[J]. 计算机科学, 2023, 50(3): 371-379.

CUI Jingsong, ZHANG Tongtong, GUO Chi, GUO Wenfei. Network Equipment Anomaly Detection Based on Time Delay Feature [J]. Computer Science, 2023, 50(3): 371-379.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.211200280

基于时延特征的网络设备异常检测

崔竞松[1]　张童桐[1]　郭　迟[2]　郭文飞[2]
1 空天信息安全与可信计算教育部重点实验室(武汉大学国家网络安全学院)　武汉 430079
2 武汉大学卫星导航定位技术研究中心　武汉 430079
(jscui@whu.deu.cn)

摘　要　随着互联网的飞速发展,网络设备的安全问题受到了广泛关注。针对现有的网络设备异常检测技术存在破坏性强、检测难度大的问题,文中以网络设备传输处理数据包所花费的时延作为检测依据,提出了一种基于时延特征的异常检测方案。所提方案采用了侧信道分析的方法,无须对网络设备进行升级改造,具有非侵入、易实施、广域性等特点。首先,使用高精度授时技术时戳机采集家庭路由器传输数据包时的时延变化信息,采用遗传算法提取时延分布的峰值位置特征;然后,针对数据集不平衡的问题,使用一类支持向量机算法构建异常检测算法;最后,通过搭建实验平台验证了检测方案的有效性,并对实验结果进行了评估。实验结果表明,所提方法具备可行性和有效性。

关键词:异常检测;时延;网络设备;一类支持向量机;峰值位置
中图法分类号　TP181

Network Equipment Anomaly Detection Based on Time Delay Feature

CUI Jingsong[1],ZHANG Tongtong[1],GUO Chi[2] and GUO Wenfei[2]
1 Key Laboratory of Aerospace Information Security and Trusted Computing,Ministry of Education,School of Cyber Science and Engineering,Wuhan University,Wuhan 430079,China
2 GNSS Research Center,Wuhan University,Wuhan 430079,China

Abstract　With the rapid development of the Internet,the security of network equipment has received extensive attention. Aiming at the problems of that the existing network equipment anomaly detection technology is destructive and difficult to detect,the paper uses the packets delay spent by the network equipment to transmit and process data packets as the detection basis,and proposes an anomaly detection scheme based on delay characteristics. The proposed scheme adopts side channel analysis,and it does not need to upgrade the equipment's software or hardware. It has the characteristics of non-intrusive and easy to implement. Firstly,the method uses the high-precision timing technology time stamp machine to collect the time delay information,and uses the genetic algorithm to extract the peak position feature of the delay distribution. Secondly,to solve the problem of the imbalance of data set,the method uses one-class support vector machine algorithm to construct anomaly detection algorithm. Finally,the validity of the method is verified by building an experimental platform,and the experimental results are evaluated. Experimental results show that the proposed method is feasible and effective.

Keywords　Anomaly detection,Delay,Network equipment,One-class support vector machine,Peak position

1　引言

随着互联网的飞速发展,网络设备逐渐成为现代人类生活中不可或缺的部分。国家互联网应急中心(CNCERT)在2021年发布的《2020年我国互联网网络安全态势综述》中[1]指出,目前许多恶意程序带来了大量安全威胁和风险,主要包括用户信息和设备数据泄露、硬件设备遭控制和破坏、攻击路由器等网络设备以窃取用户上网数据等。如何对网络设备进行保护以防止其受到恶意程序的攻击成为了保障国家和人民信息安全的重要课题之一。

目前,针对网络设备进行异常检测主要采用的是入侵检测技术,其工作原理是对网络设备的数据包流量进行分析以提取出异常特征[2-4]。然而,这些方法往往会陷入待检数据量庞大、传输协议复杂、数据内容加密等困境,无法有效检测隐蔽的异常威胁。除此之外,由于无法获得网络设备的源代码,研究人员采用基于逆向分析的异常检测技术[5],即先将硬件和固件逆向,然后分析其中是否有恶意代码。但此类方法存在3个致命缺陷:1)此类分析方法可能会因为设备硬件和

到稿日期:2021-12-27　返修日期:2022-04-03
基金项目:十三五重点研发计划项目(2016YFB0501801)
This work was supported by the National Key R&D Project of China During the 13th Five-Year Plan Period(2016YFB0501801).
通信作者:郭迟(guochi@whu.edu.cn)

固件复杂度过高而无法分析;2)此类分析方法属于静态分析方法,如果网络设备具有在线升级能力,则难以确保其升级后的设备安全性;3)此类分析方法需要对设备进行拆解,使得分析后的设备无法正常上线工作,只能对一批设备进行抽样检测,而对于没有检测到的设备,无法确保其安全性。因此,亟需一种能够在实用环境下非侵入式高效检测网络设备异常的方法。

为了解决上述问题,本文提出了一种基于时延特征的异常检测方法来检测网络设备的异常。网络设备中出现异常工作状态时,例如恶意代码被激活后,网络设备的数据包处理流程发生变化,从而表现出不同的时延信息,且时延信息难以被掩盖。但由于传统时延度量方法存在度量精度不够、工作场景无法应用于广域环境的不足,本文利用 GNSS(Global Navigation Satellite System)高精度授时技术构建了纳秒级信号时戳生成器(Nanosecond Precision Signal Timestamp Generator,NPSTG),简称信号时戳生成器或者 GNSS 纳秒级时戳机。NPSTG 通过测量网络设备传输处理数据包所耗费的时延,实现了高精度、广域性、易实施的传输时延度量。在得到网络设备的时延后,我们利用侧信道方法分析网络设备处于异常工作状态时的时延分布,用数据包时延的变化来评估检测网络设备的异常,形成了完整的检测和评估体系。

本文方案的主要贡献如下:

(1)研制了一种基于 GNSS 的纳秒级信号时戳生成器 NPSTG,其精度可高达纳秒级,容易在大规模广域的设备中进行部署。

(2)提出了一种基于时延特征的网络设备异常检测方法,该方法利用网络设备传输处理数据包时延的难以掩盖性,通过分析数据包时延分布,判断其是否处于异常工作状态。本文方法不需要侵入网络设备进行分析,其具备普适性,不依赖于复杂流量分析。

本文第 2 节主要介绍了与网络设备相关的异常检测技术以及时延测量方法;第 3 节介绍了本文方案所针对的威胁模型;第 4 节介绍了本文方案的检测模型构建过程;第 5 节通过搭建实验平台验证了检测的效果,并对实验结果进行了分析;最后总结全文并展望未来。

2 相关工作

2.1 网络设备异常检测方法

常见的网络设备异常检测方法根据检测方式主要可以分为外部流量检测、侧信道信息检测和设备系统软件检测 3 种。

在流量检测方面,主要体现在入侵检测技术方面[2-4]。Eskandari 等[6]提出了一种基于异常的物联网网络设备入侵检测系统,其工作重点是捕获网络设备的输入和输出流量信息,并构建系统正常工作时的流量模型。通过训练入侵检测模型以发现多种网络异常行为。Yan 等[7]提出了一种基于无线路由器的 IoT 设备轻量级防御框架 WRGuardian,利用家用无线路由器在网络流量的掌控能力和拓扑结构优势,从被动防御和主动防御两个方面入手,及时监测并阻断目前针对 IoT 设备的主要攻击行为,同时定期扫描检测安全问题并进行修复。

在侧信道信息检测方面,Dunlap 等[8]提出了一种对工业设备进行异常检测的方法,该方法使用基于时序的侧信道分析方法来建立唯一的设备指纹,以帮助检测对设备的未经授权的修改。Ni 等[9]提出了一种基于芯片辐射的物联网设备异常检测方法,该方法使用芯片辐射的电磁波作为检测依据,可以检测出设备的异常状况。Yang 等[10]直接在实际网络环境中部署网络管理及检测监控软件,以评估路由器面临的安全风险,通过实时分析路由器的 CPU 使用率及带宽占用率来达到在线监测的目的。

在设备系统软件检测方面,Adithyan 等[5]讨论了针对网络设备固件逆向工程的流程与工具,有几个开源工具可用于嵌入式固件逆向工程,其中包括 Binwalk[11] 和 Firmware-mod-kit(FMK)[12]。这些工具可以极易用于分析、逆向工程和提取嵌入式固件,包括固件头、Linux 内核、引导程序和文件系统。然而,如果没有所需的技术经验来识别嵌入式设备固件中的不同文件系统签名,这些工具就不能用来完全逆向固件。Firmalice[13]使用静态程序分析生成固件的程序依赖图,获取一个从入口点到特权程序点的认证切片,再通过符号执行判断路径的约束中是否具有确定性的约束,如果存在,则可以认定二进制固件中存在后门。Hu 等[14]提出了一种基于嵌入式固件的库函数识别方法,用于对无文件系统固件恶意代码进行检测。

本文方法属于侧信道检测,任何网络设备传输处理数据包都需要花费一定的时间,且该时间不可掩盖,因此具备普适性。本文方案不需要侵入网络设备分析,不依赖于复杂流量分析和固件分析,检测人员实施相对容易。

2.2 网络设备时延度量方法

本文利用网络设备工作过程中产生的时延信息的难以掩盖性,构建时延分布并提取特征,然后使用机器学习技术来检测异常。国内外对网络设备时延的测量方法的研究很多,但尚未有将时延应用到异常检测的研究。Angrisani 等[15]提出了一种对软件路由器单跳时延的测量方法,该方法通过在路由器操作系统内核程序中插入钩子来测量数据包的处理时延。Breuer 等[16]提出了一种高精度的数据包时延测量方法,使用 IEEE 1588 标准[17]进行时间同步,测量了数据包在 LAN 网线中的传播时延以及交换机的数据包处理时延,精度可以达到纳秒级。Chen 等[18]提出了一种分析数据包包时延和吞吐量以监测多核路由器处理器运行状态的方法,即分析多核路由器中正常工作的处理器的数量。该方法中数据包处理时延的测量是通过软件路由器中任务的处理时间和数据流的吞吐率计算而得的,测量精度依赖于处理器主频,因此测量精度较低。

当被测设备距离检测人员较远或检测人员不易接近待测设备时,此时急需一种能够在远距离、广域环境下有效检测设备异常的方法。由于 GNSS 时间同步技术易于在广域环境下应用,因此本文使用 GNSS 技术测量本地设备时延,并且可以轻易地扩展到远程测量环境中。本文构建了 GNSS 纳秒级时戳机来测量网络设备的处理时延,可以得到纳秒级精度的测量结果,因此可以观测到设备内部微小变化导致的时延改变。

3 威胁模型

考虑到实际针对网络设备的攻击场景,我们将攻击范围

限制在以下几个方面。

(1)攻击者可以非法侵入网络设备，然后通过篡改系统固件程序[19]代码来实现恶意操纵数据包的行为。

(2)攻击者实现的恶意行为主要是窃取敏感信息。在数据包报文未被加密的情况下，攻击者可以检索报文信息以窃取秘密信息；当数据包报文被加密时，攻击者可以检索包头信息，如 IP 地址、协议类型等实现用户隐私的窃取。

由于网络设备被攻击者非法修改，因此网络设备的数据包处理流程会发生改变。恶意程序一般会增加对数据包的额外操作，如检索数据包内容、篡改数据包内容或者复制转发数据包，这些行为会造成网络设备转发处理数据包的时延发生变化。同时，数据包时延的特性是难以被攻击者掩盖，即攻击者难以消除网络设备异常行为造成的数据包时延变化，因此本文选择利用时延分析网络设备的异常。

4 基于时延特征的网络设备异常检测方法构建

本文方法基于网络设备传输处理数据包的时延特征，使用 OCSVM(One-Class Support Vector Machine)算法[20]构建异常检测模型。首先使用构建的 GNSS 纳秒级时戳机采集时延数据，然后利用遗传算法提取时延分布特征，最后使用 OCSVM 构建异常检测模型，从而有效识别网络设备异常行为。

4.1 方法原理

本文所研究的处理时延指网络设备传输处理数据包所需的时间，对于数据包 n，将其到达设备输入端口的时间表示为 $t_{in}(n)$，离开输出端口的时间表示为 $t_{out}(n)$。因此，处理时延 $d(n)$ 的计算式如式(1)所示：

$$d(n)=t_{out}(n)-t_{in}(n) \tag{1}$$

当网络设备发生异常时，被修改的程序尝试窃取敏感信息，该行为需要检索数据包以匹配敏感信息字符串，从而导致数据包在网络设备内部传输处理所花费的时间增加。在发生异常的情况下，若将数据包 n' 到达设备输入端口的时间表示为 $t_{in}(n')$，离开输出端口的时间表示为 $t_{out}(n')$，则处理时延 $d(n')$ 的计算式如下：

$$d(n')=t_{out}(n')-t_{in}(n') \tag{2}$$

其中，$d(n')>d(n)$，网络设备异常行为会造成传输处理数据包的时延发生变化，使其成为恶意代码难以掩盖的行迹。因此，时延的不易掩藏性使其成为了对网络设备行为安全状态进行分析的重要视角，而将网络设备处理网络数据包所产生的处理时延作为检测网络设备异常的特征是一种有效的侧信道分析方法。

基于上述分析，如果能准确地采集数据包经过网络设备的时延分布，并且其度量精度达到能够分辨数据包经过网络设备正常工作时的时延分布与经过网络设备异常工作时的时延分布的显著差异，就可以通过比对数据包时延分布来判定当前网络设备是否处于安全工作状态。

4.2 整体框架

基于时延特征的网络设备异常检测框架如图 1 所示。

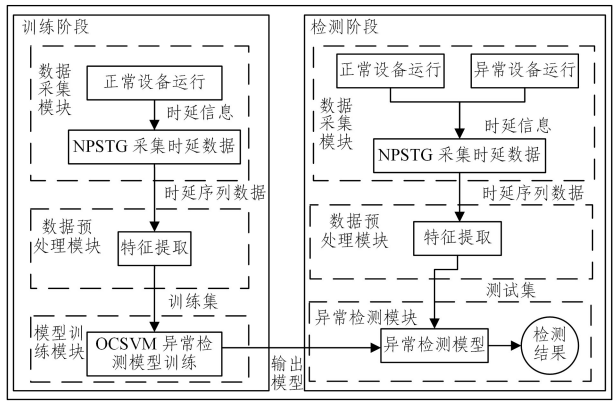

图 1 异常检测框架

Fig. 1 Anomaly detection framework

检测框架主要分为训练阶段与检测阶段。训练阶段包括数据采集模块、数据预处理模块和模型训练模块；检测阶段包括数据采集模块、数据预处理模块和异常检测模块。

(1)数据采集模块

通过 GNSS 纳秒级时戳机采集路由器运行过程中的数据包处理时延数据。

(2)数据预处理模块

数据预处理模块使用遗传算法提取路由器时延信息的特征向量，构建样本数据集。

(3)模型训练模块

模型训练模块构建基于 OCSVM 算法的异常检测模型，使用只包含网络设备正常行为时延数据的数据集训练模型。

(4)异常检测模块

将测试样本数据集输入到异常检测模型中，模型输出检测结果，根据检测结果判断网络设备是否存在异常。

4.3 时延采集

4.3.1 纳秒级信号时间戳生成器

为了将 GNSS 授时的纳秒级精度时间应用到路由器时延的测量中，本文设计研发了一种基于 GNSS 纳秒级精度授时的时戳硬件装置，命名为纳秒级信号时戳生成器(NPSTG)。NPSTG 的实现主要分为 5 个模块：输入模块、输出模块、以太网帧头检测模块、GNSS 纳秒级授时模块和时戳数据帧生成模块，其时戳生成原理如图 2 所示。

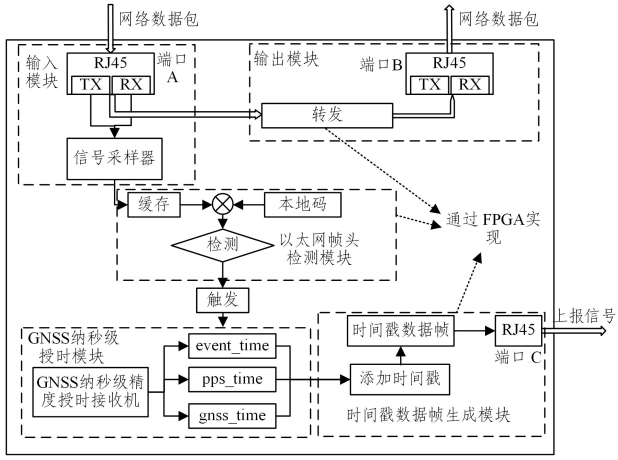

图 2 纳秒级信号时间戳生成器的设计原理图

Fig. 2 Design schematic diagram of NPSTG

首先，当端口 A 检测到网络信号时，会通过 FPGA 转发器直接将信号从端口 A 转发到端口 B，从而不会对原始网络信号造成任何干扰；同时，触发信号采样器对网络信号进行采样，产生数字脉冲，送到现场可编程门阵列（Field Programmable Gate Array，FPGA）进行以太网帧头检测。如果检测结果为真，即检测到以太网帧头时，则由 FPGA 实现的计时模块负责接收 GNSS 纳秒级精度授时接收机提供的整秒时间 gnss_time 和 FPGA 触发时整秒内的时间 event_time，并将其传送给时间戳数据帧生成模块，构造时间戳并将其添加到新生成的时间戳数据帧中，然后通过端口 C 将时间戳数据帧发送到计算机用于分析。

（1）输入模块

输入模块负责接收到达端口 A 的网络信号，同时触发信号采样器对网络信号进行采样。

输入模块由一个 RJ45 接口和信号采样器组成。RJ45 接口负责接收网络信号；信号采样器负责对网络信号采样，生成数字脉冲发送给以太帧头检测模块。

（2）输出模块

输出模块负责将网络信号从端口 B 发送出去，不对原始网络信号做任何改变。

（3）以太网帧头检测模块

以太网帧头检测模块主要负责检测以太网帧头以确定网络信号到达 GNSS 纳秒级时戳机的准确时间，触发计时模块。

以太网帧检测模块如图 2 所示，输入为信号采样器量化的数字信号，输出为帧头检测结果，缓存中存储的缓冲序列是对输入信号进行缓冲得到的序列编码，本地码代表用于比较的以太网帧前导符（见图 3）。本地码是一个固定序列，包括 7 个字节的前导码（帧同步码）以及 1 个字节的帧起始定界符。前导码由“1 0”交替的 7 个字节组成，用于信号同步，而帧起始定界符为连续 6 位交替的“1 0”及末位 2 个“1”，末位连续 2 个“1”表示一帧即将开始。FPGA 通过对缓冲序列和本地码序列进行异或运算来判断缓冲序列和本地码序列是否一致，从而确定是否开始接收以太网帧信号。因此，其在确认以太网帧开始时输出“1”（高电平信号），否则输出“0”（低电平信号）。

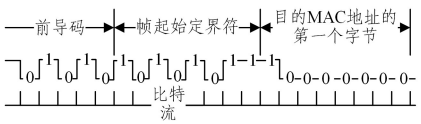

图 3 时间获取位置

Fig. 3 Location of determined time

（4）GNSS 纳秒级授时模块

为了获得纳秒精度时间戳信息，GNSS 纳秒级精度授时接收机（简称 GNSS 授时接收机）被集成到 GNSS 纳秒级时戳机中。GNSS 授时接收机外接信号天线，负责接收 GNSS 信号，并输出 1PPS 整秒脉冲以及协调世界时（Universal Time Coordinated，UTC）计数。

注意，本文使用的 GNSS 授时接收机的晶体频率为 12 MHz，倍频增加到 60 MHz，被用作本地时间计数器的信号时钟频率。由于本地晶振分辨率为 16.7 ns，因此本信号时戳生成器的时间精度为 16.7 ns。

GNSS 纳秒级授时模块是用 FPGA 实现的，如图 2 所示，输入为 GNSS 授时接收机提供的 GNSS 信号，输出为 gnss_time，pps_time 和 event_time。当 FPGA 检测到以太网帧头时，在物理层的 PHY 接口触发输出脉冲信号，将计数器的值存储在相应的寄存器中。与时间戳信息相关的 3 个部分的定义如下。

1）gnss_time：GNSS 授时接收机提供的 UTC 时间计数，其中 UTC+8 是准确的本地整秒时间。

2）pps_time：GNSS 授时接收机生成的秒脉冲信号，从一个 PPS 秒脉冲到达下一个 PPS 秒脉冲这一过程中，本地晶振振荡的计数。

3）event_time：从 PPS 秒脉冲开始计数到检测到以太网帧头的本地晶振振荡次数。当 gnss_time 跳变时该计数值置零，重新开始计数。

（5）时间戳数据帧生成模块

时间戳数据帧生成模块负责将计数模块中的时间信息封装成时间戳数据帧（包含监测到以太帧信号到达的准确时间信息的数据帧），通过上报端口将其发送到外部数据包处理程序。外部数据包处理程序通过解析时间戳数据帧的特定字段，来获得以太帧信号到达时间戳生成器的精确时间并对其进行分析。

如图 2 所示，时间戳数据帧生成模块由 FPGA 实现，输入是来自计时模块的 3 个时间计数，输出是带有时间信息的时间戳数据帧。FPGA 从对应的寄存器中获取 3 个时间计数，将 MAC 地址、IP 数据报头、UDP 数据报头及这 3 个时间计数依次填入数据帧的相应字段中，并计算帧校验序列（FCS），形成如图 4 所示的时间戳数据帧。

物理层（以太网帧前导码）	数据链路层（MAC首部）	网络层（IP首部）	传输层（UDP伪首部）	应用层（pps_time,event_time_gnss_time）	FCS校验

图 4 时间戳数据帧

Fig. 4 Timestamp data frame

（6）硬件实现

信号时戳生成器的实物如图 5 所示，其硬件信息如表 1 所列。

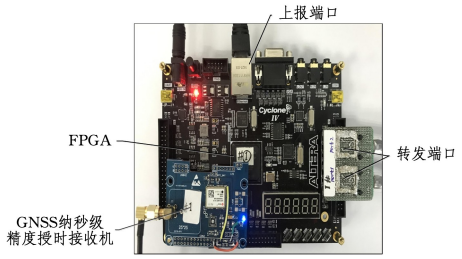

图 5 信号时戳生成器实物图

Fig. 5 Physical photo of signal timestamp generator

表 1 信号时戳生成器的硬件信息

Table 1 Hardware information of NPSTG

硬件	介绍
开发板	AX359，集成现场可编程门阵列（FPGA）芯片和多个网络端口
FPGA 板	XILINX A7-35T 型，检测以太网帧头，触发时间计数模块，转发以太网帧，并生成时间戳数据帧
GNSS 授时接收机	提供纳秒精度的秒脉冲信号和 UTC 串行端口时间

4.3.2 采集模型

如图 6 所示，我们建立了路由器时延采集模型。采集模型包含两个 GNSS 纳秒级时戳机，一个数据包发送节点，一个数据包接收节点，一个上位机和一个要检测的网络设备。两个 GNSS 纳秒级时戳机检测网络信号，并生成带有时间戳信息的时间戳数据帧。数据包处理程序安装在处理器（另一台独立 PC）中，它从两个 GNSS 纳秒级时戳机接收时间戳数据帧 T_{in} 与 T_{out}，根据式(1)计算处理时延。

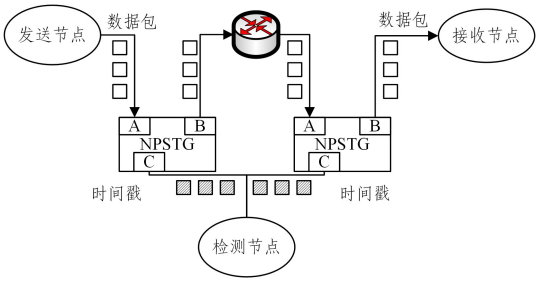

图 6 时延数据采集示意图

Fig. 6 Diagram of delay data collection

4.4 数据预处理

4.4.1 特征选择

假设根据上述采集方法在一段时间里采集到的路由器时延数据序列为 $\boldsymbol{D}=\{d_1,d_2,\cdots,d_i,\cdots,d_n\}$，其中 n 表示采集到的路由器传输处理数据包的总量，d_i 表示第 i 个数据包的处理时延，则可以根据式(3)和式(4)构造该序列的时延-频数分布图，即时延分布图。

$$F_j=\sum_{i=1}^{n} f_j(d_i) \tag{3}$$

$$f_j(d_i)=\begin{cases}1, & d_i\in[t_j,t_{j+1}]\\ 0, & \text{else}\end{cases} \tag{4}$$

其中，$F_j(j=1,2,\cdots,m)$ 表示序列 $\boldsymbol{D}$ 中出现在时延区间 $[t_j,t_{j+1}]$ 中的时延频次，$f_j(d_i)$ 是判定特定时延数据 d_i 是否属于时延区间 $[t_i,t_{i+1}]$ 的判定函数。当 d_i 属于时延区间 $[t_i,t_{i+1}]$ 时，取值为 1，否则为 0。

图 7 给出了一次性采集一定数量的经过某特定路由器的数据包所得到的处理时延分布图。可见，路由器传输处理数据包的时延本身具有波动性，无法直接将其作为路由器异常检测的依据。

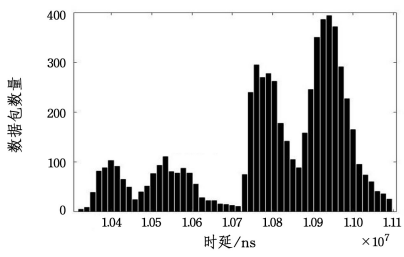

图 7 路由器时延分布图

Fig. 7 Router delay distribution diagram

通过多次实验，对比正常路由器和异常路由器数据包传输处理时延分布图的差异，发现存在以下现象。

现象 1 虽然时延序列本身具有波动性，但是对于特定型号的路由器而言，异常行为会使其时延分布图的峰值所处的时间区间发生偏移，但峰值个数和峰位置相对稳定。

此现象可以被解释为：由于路由器中的异常行为引入了数据筛选、转发，导致传输处理数据包所需要的时间增加，从而表现为时延分布峰值的整体非正常右移。

因此，我们最终选取表示多个主峰的 n 元峰位置作为路由器时延分布图的特征向量。

$$Feature=\begin{bmatrix}b_1\\ b_2\\ \cdots\\ b_k\end{bmatrix} \tag{5}$$

其中，$b_i(i=1,\cdots,k)$ 表示第 i 个主峰的位置。

现象 2 虽然时延序列本身具有波动性，但是对于特定型号的路由器而言，其正常状态下的时延分布图可以看作由多个高斯分布组合而成。

因此，我们采用多高斯拟合对时延分布进行特征提取。

4.4.2 特征提取

多高斯拟合[21]是使用高斯函数对数据点进行函数逼近拟合的方法，如式(6)所示：

$$G(x)=\sum_{i=1}^{k} A_i\exp\left(-\frac{(x-b_i)^2}{c_i^2}\right) \tag{6}$$

其中，待定系数 A_i，b_i 和 c_i 分别表示高斯曲线的峰值、峰值位置和半宽度，k 为待拟合的高斯分布的数量。因此，特征提取算法的目标是将多个高斯分布的峰位置计算出来，即求多个高斯分布的峰位置 b。

为了得到多个高斯分布的期望，我们采用遗传算法[22]优化目标函数的方式拟合高斯分布的参数，其提取算法的步骤如图 8 所示。

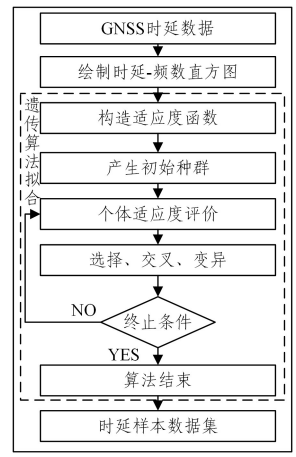

图 8 特征提取算法的流程图

Fig. 8 Flow chart of feature extraction algorithm

假设时延分布符合多高斯分布，则频数可用高斯函数描述为：

$$\boldsymbol{F}_j=\sum_{i=1}^{k} A_i\exp\left(-\frac{(t_j-b_i)^2}{c_i^2}\right) \tag{7}$$

其中，$\boldsymbol{F}_j(j=1,2,\cdots,m)$ 表示序列 D 出现在时延区间 $[t_j,t_{j+1}]$ 的时延频次。

(1)编码个体

向量$[A_1,b_1,c_1,A_2,b_2,c_2,\cdots,A_k,b_k,c_k]$表示一个由 k 个高斯分布组成的个体，由高斯曲线的峰值 A_i、峰值位置 b_i 和半宽度 c_i 组成，k 为待拟合的高斯分布的数量。

(2)产生初始种群

种群是个体的集合，采用随机方法生成包含 N_{InitSize} 个个体的初始种群。

(3)评估个体

适应度函数用于度量个体对环境的适应程度，个体的适应度越强，表示它的存活能力越强，越容易被遗传至下一代。构造的适应度函数如式(8)所示：

$$Fitness=\sum_{i=1}^{k}\sum_{j=1}^{m}(F_j-A_i\exp\left(-\frac{(t_j-b_i)^2}{c_i^2}\right)^2 \tag{8}$$

根据适应度函数可以得知，个体适应度值越小，表示时延分布越接近该个体所构造的多高斯分布，个体越容易被遗传至下一代。

(4)选择

选择操作指定遗传算法如何为下一代选择父代。每个父代对应于一段与其适应度值成比例的长度直线。算法沿着直线以相同的步长移动。在每个步骤中，该算法根据其所在的部分分配父节点。第一步是一个小于步长的均匀随机数。

(5)交叉

根据给定的概率进行单点交叉，生成新的个体。随机选择交叉点的位置，将两个个体交叉点后的基因互换。

(6)变异

根据给定的概率对个体进行变异操作生成新的个体，随机产生变异点，将个体上该点位的基因取反。

(7)设置终止条件

终止条件设置为：当连续 N_{maxGen} 代适应度函数加权平均值变化小于 1×10^{-6} 时，算法停止。

4.4.3 训练样本集生成

经过对一段时间内得到的多个时延分布进行特征提取，最终得到训练样本集 $\boldsymbol{T}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$，其中 $\boldsymbol{x}_i=\{b_1,b_2,\cdots,b_k\}$ 表示某时延序列对应的时延分布特征向量(n 表示峰值位置数量，即时延特征向量的维数)，$i=1,2,\cdots,N$，N 表示训练样本集的个数。

4.5 OCSVM 异常检测算法

基于 SVM[23] 的超平面分类和最大分类区间的思想，Schölkopf 等提出了一种 OCSVM 算法[20]，由于其训练样本只需要一类数据，因此非常适用于其他类别数据未知或难以获得的两分类场景，如异常检测场景。为此，针对网络设备异常检测中正样本容易采集、异常样本难以获得且即使获得也难有代表性的情况，采用 OCSVM 算法构建异常检测模型，实现对网络设备异常的检测。

OCSVM 的原理是通过核函数将样本映射到高维空间，在高维空间中构造线性判别函数，从而实现对样本的分类决策。OCSVM 具有运算时间较短、在小样本数据集上表现较好等优点。基于 OCSVM 的异常检测方案通过学习正样本求解出可以代表这类数据特征的模型，进而在检测过程中正确判定样本类别。

具体而言，即在高维空间中，构造一个超平面 $\boldsymbol{\omega}\varphi(\boldsymbol{x})-\rho=0$，将正样本从原点分离出来，其中，$\boldsymbol{\omega}$ 是超平面的法向量，ρ 是超平面的截距，$\varphi(x)$ 为映射函数。

$$\min\left(\frac{1}{2}\|\boldsymbol{\omega}\|^2+\frac{1}{vN}\sum_{i=1}^{N}\xi_i-\rho\right) \tag{9}$$

$$\text{s.t.}(\boldsymbol{\omega}\cdot\varphi(\boldsymbol{x}_i))\geqslant\rho-\xi_i,\xi_i\geqslant0 \tag{10}$$

式(9)中，N 为正样本数量，ξ_i 为松弛因子，$v\in(0,1)$ 为百分比参数，用于控制支持向量在训练样本中的比例。引入的径向基核函数(Radial Basis Function，RBF) 具有良好的泛化能力和快速收敛性，且需要较少的参数，可以降低模型的复杂性，同时提高异常的有效检测率。因此，本文使用 RBF 作为核函数，如式(11)所示：

$$K(x_i,x_j)=exp(-\gamma\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2) \tag{11}$$

利用拉格朗日公式将式(11)转化为对偶函数，最后最优分类超平面函数为：

$$f(x)=\text{sgn}(\sum_{i=1}^{n}\alpha_iK(\boldsymbol{x}_i,\boldsymbol{x}_i)-\rho) \tag{12}$$

其中，α_i 是拉格朗日乘数，$K(\boldsymbol{x}_i,\boldsymbol{x})$ 是核函数。对于训练样本 $\boldsymbol{x}_i$，$f(\boldsymbol{x}_i)$ 表示 $\boldsymbol{x}_i$ 在高维空间中位于超平面的正负方向，$f(\boldsymbol{x}_i)$ 大于 0 表示分类为正样本，$f(\boldsymbol{x}_i)$ 小于 0 表示分类为负样本。

4.6 评估指标

为了评估检测模型的检测能力，本文使用混淆矩阵计算相应评价指标。根据样本标签，可以将检测结果分为 4 类，如表 2 所列。本文将检出率(TNR)、准确率(ACC)与召回率(TPR)作为评估指标，检出率(TNR)是被模型检测为负类样本的数量占实际为负类样本的数量的比例，准确率(ACC)是被模型正确检测的样本数量占测试集数量的比例，召回率(TPR)是被模型检测为正类样本的数量占实际正类样本数量的比例。

表 2 混淆矩阵

Table 2 Confusion matrix

样本标签	预测为正类	预测为负类
正类	真正类(TP)	假负类(FN)
负类	假正类(FP)	真负类(TN)

检出率 TNR、准确率 ACC 与召回率 TPR 的计算式如式(13)－式(15)所示：

$$TNR=\frac{TN}{FP+TN} \tag{13}$$

$$ACC=\frac{TN+TP}{TP+FN+FP+TN} \tag{14}$$

$$TPR=\frac{TP}{TP+FN} \tag{15}$$

5 实验与分析

5.1 环境搭建

为了对本文方法进行验证和评估，我们搭建了检测的测试环境。我们选取家用路由器作为检测目标设备，本文的研究重点是网络设备传输处理数据包的时延，由于网络设备的主要功能是传输处理数据包，因此本文的实验结论也适用于其他网络设备。

实验的测试环境由发送端 Sender、接收端 Reciver、攻击

端 Attacker、待网络设备 Router、上位分析机 Decision 以及两台纳秒级信号时间戳生成器（NPSTG）组成，具体的网络拓扑环境如图 9 所示。当有数据包经过时戳机所在的链路时，时间戳设备会向上位分析机 Decision 上报数据包经过时对应的时间戳，通过计算即可得到数据包在路由器中的处理时延。

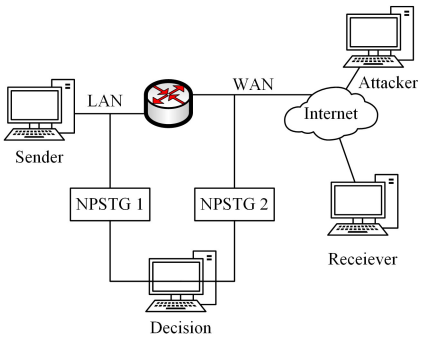

图 9 测试环境

Fig. 9 Experimental environment

异常检测模型部署在分析机 Decision 上。Decision 使用的操作系统为 Windows 10（64 位），处理器为 Intel（R） Core（TM） i3-7350K 4.2 GHz，内存 8 GB。实验设备配置信息如表 3 所列。

表 3 实验设备信息

Table 3 Experimental equipment information

实验设备	型号
网络设备	GL.iNet 品牌 GL-MIFI 家用智能路由器
发送主机	台式机
接收主机	笔记本
攻击主机	笔记本
时戳机 1	—
时戳机 2	—
分析机	台式机

通过采集路由器木马程序被激活前后的时延数据来构建数据集，并训练异常检测模型，最后使用准确率、精确率、召回率等指标来评估模型的性能。攻击主机在某一时刻激活路由器中的木马以影响数据包处理流程。木马程序被通过刷固件的方式预先植入路由器，路由器开机后自动运行。

木马程序主要有两个状态：监听状态与工作状态。

（1）监听状态

被激活前，木马程序处于监听状态。木马程序等待攻击者发送控制命令，控制命令分为激活、停止工作。当攻击主机发送激活命令时，路由器开始工作；当攻击主机发送停止工作命令时，路由器回到监听状态。

（2）工作状态

被激活后，木马程序处于工作状态。恶意程序检索数据包内容中是否匹配敏感词汇表中的内容，敏感词汇表中的内容为判断是否复制转发数据包的依据，即当发往接收主机数据包中包含敏感词汇表中的内容时，路由器将数据包复制转发到攻击主机。敏感词汇表如表 4 所列。在检索敏感词汇表内容的同时，木马程序会等待停止工作命令的下达，以便切换到激活前的状态。

表 4 敏感词汇

Table 4 Sensitive words

敏感词	是否忽略大小写
password	否
username	否

5.2 样本构建

测量实验中，发送主机 Sender 向接收主机 Receiver 持续发送 UDP 数据包产生测量流量，以监测路由器时延特征。发送主机的数据包发送周期为 15 ms，数据包长度固定为 60 字节，分析机以每 20000 个数据包为一组提取一个时延分布特征向量。在实际网络流量中，包含敏感信息的数据包数量比不包含敏感信息的数据包数量少，因此在每一组测试流量中，根据敏感词汇表的大小，设置包含敏感信息的数据包占比为 1%。

根据上述方法，构建了 170 个正常样本（木马未被激活）作为训练集样本，构建了 40 个正常样本、40 个异常样本（木马被激活）作为测试集样本。训练集和测试集的数据分布如表 5、表 6 所列。

表 5 训练集数据分布

Table 5 Training set data distribution

类别	数量	标记
正类	100	1

表 6 测试集数据分布

Table 6 Test set data distribution

类别	数量	标记
正类	40	1
负类	40	−1

5.3 实验结果

图 10 给出了一次实验中分别采集路由器正常状态和异常状态的时延分布图。图中，X 轴表示数据包处理时延，Y 轴表示在该次实验中数据包处理时延处于 X 轴数值的数据包个数。黑柱表示木马程序未被激活前的时延分布图，白柱表示木马程序被激活后处于异常状态的时延分布图，可以观察到木马程序被激活后会对时延分布带来微小的改变，使得激活前的分布整体向右平移。

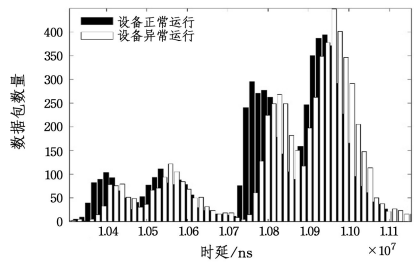

图 10 设备分别处于两种状态的时延分布图

Fig. 10 Time delay distribution diagram of equipment in two states

5.3.1 特征向量维度确定

在获取时延数据后，对数据进行预处理，得到正、负样本集。为了确定特征向量的维度，分别对特征向量的维度为 1，2，3，4，5，6 的情况作了检测准确率的对比分析，实验结果如图 11 所示。

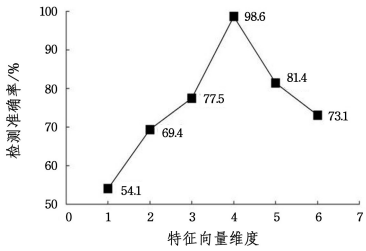

图 11 检测准确率随特征向量维度的变化

Fig. 11 Variations of detection accuracy with dimension of eigenvector

图 11 中，横轴表示样本特征向量的维度，即多高斯分布峰值位置的数量，纵轴表示检测的准确率。从图中可以看出，当特征向量维度达到 4 时，检测准确率达到 98.6%，说明使用遗传算法从时延-频数直方图中提取 4 个峰值位置时，检测效果最好，也符合图 10 所示的观测特征。因此，将特征向量的维度确定为 4。

5.3.2 测试集样本特征向量特征分析

图 12 给出了时延分布特征提取之后的结果。X 轴上的值表示样本序号，Y 轴上的值表示时延分布图中多峰峰值位置值。对于任意一个给定的样本 x，其对应的 4 个峰值位置值会对应 4 个 y 值。正方形的点是木马未被激活的时延特征向量，三角形的点是木马被激活后的时延特征向量。

从图中可以观察到，时延特征本身存在一定的波动性，但木马被激活前后的时延特征差异大于时延特征的波动，因此本文方法的实验结果可以感知木马被激活前后的时延变化。

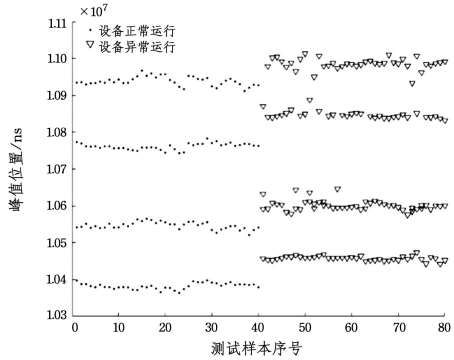

图 12 特征提取结果

Fig. 12 Feature extraction results

5.3.3 异常检测结果

在 OCSVM 模型训练中，经过多次实验，取经验参数 $\nu=0.025$，$\gamma=1\times10^{-12}$，其检测结果如表 7 所列。

表 7 模型评估结果

Table 7 Model evaluation results

（单位：%）

	TNR	*TPR*	*ACC*
结果	99.7	97.5	98.6

由表 7 可知，异常检测模型的准确率可达 98.6%，TNR 在 99%以上，TPR 在 97%以上。实验结果表明，该方法具有很高的检测准确率和极高的异常检出率。

5.4 相关工作比较

本文将传统网络设备异常检测方法中基于流量分析的入侵检测与软件逆向分析作为对比对象，从多个方面分析了本文方法的优缺点。三者的异同如表 8 所列。

表 8 相关工作比较

Table 8 Related work comparision

检测条件	逆向分析	入侵检测	本文方法
获取固件	需要	不需要	不需要
分析流量	不需要	需要	不需要
侵入设备	需要	不影响	不影响

经过分析可以得出，相比传统的异常检测方法，首先本文方法不需要提前获得被测路由器的固件系统，无须对固件中的代码进行逆向分析，因此在不容易获得固件情况下也无须拆解设备以获得固件程序。本文方法对测试人员的代码分析能力要求较低，容易上手开展检测工作，其检测难度大幅降低，具有广泛推广使用的可能。其次，本文方法具有非侵入式的特点，不需要详细分析数据包流量内容。最后，得益于使用的 GNSS 技术，本文方法可应用于远程检测环境。

本文方法检测准确率、异常检出率均很高，无须获取待检设备固件，无须对代码进行分析，且容易被扩展至远程测量环境中，因此其具备一定的应用前景。

结束语 本文提出了一种基于时延特征的网络设备异常检测方法。该方法包括一个 NPSTG、一种基于遗传算法的时延分布特征提取算法和基于 OCSVM 的异常算法。本文方法可以检测网络设备中的异常，而无须对流量、固件进行分析，具有非侵入式的特点。实验结果验证了该方法的有效性。本文方法需要提前获取网络设备的正常训练样本，这在一定程度上是有局限性的。此外，本文方法现阶段使用的测试流量是固定包长、固定协议的，如果有多种不同的流量混杂在测试流量中，则会对测试结果造成影响。因此，未来尝试升级可区分不同流量的时戳机以提高检测能力。

参 考 文 献

[1] CNCERT. Summary of China's Internet Network Security Situation in 2020[EB/OL]. (2021-05-26)[2021-12-02]. http://www.cac.gov.cn/2021-05/26/c_1623610314656045.htm.

[2] LIU H, LANG B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A survey [J]. Applied Sciences, 2019, 9(20): 4396-4420.

[3] KHRAISAT A, GONDAL I, VAMPLEW P, et al. Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges [J]. Cybersecurity, 2019, 2(1): 1-22.

[4] CHOUDHARY S, KESSWANI N. A Survey: Intrusion Detection Techniques for Internet of Things [J]. International Journal of Information Security and Privacy (IJISP), 2019, 13(1): 86-105.

[5] ADITHYAN A, NAGENDRAN K, CHETHANA R, et al. Reverse Engineering and Backdooring Router Firmwares [C]// 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2020: 189-193.

[6] ESKANDARI M, JANJUA Z H, VECCHIO M, et al. Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices [J]. IEEE Internet of Things Journal, 2020, 7(8): 6882-6897.

[7] YAN Z T,FANG B X,LIU Q X,et al. A Wireless Router-Based Lightweight Defense Framework for IoT Devices[J]. Journal of University of Chinese Academy of Sciences,2017,34(6):759-770.

[8] DUNLAP S,BUTTS J,LOPEZ J,et al. Using Timing-Based Side Channels for Anomaly Detection in Industrial Control Systems [J]. International Journal of Critical Infrastructure Protection,2016(15):12-26.

[9] NI M T,ZHAO B,WU F S,et al. CREBAD:Chip Radio Emission Based Anomaly Detection Scheme of IoT Devices[J]. Journal of Computer Research and Development,2018,55(7):1451-1461.

[10] YANG J G,LIANG L,LIU G J,et al. Method for Router Online Security Risk Assessment Quantification[J]. Journal on Communications,2013,34(11):59-70.

[11] HEFFNER C. Binwalk-Firmware Analysis Tool [EB/OL]. (2021-09-11)[2021-12-12]. https://github. com/ReFirmLabs/binwalk.

[12] COLLAKE J,HEFFNER C. Firmware modification kit [EB/OL]. (2021-05-20) [2021-12-12]. https://github. com/rampageX/firmware-mod-kit.

[13] SHOSHITAISHVILI Y,WANG R,HAUSER C,et al. Firmalice-Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware[C]// NDSS. 2015:1. 1-8. 1.

[14] HU C J,XUE Y B,ZHAO L,et al. Backdoor Detection in Embedded System Firmware without File System[J]. Journal on Communications,2013,34(8):140-145.

[15] ANGRISANI L,VENTRE G,PELUSO L,et al. Measurement of Processing and Queuing Delays Introduced by an Open-Source Router in a Single-Hop Network [J]. IEEE transactions on instrumentation and measurement,2006,55(4):1065-1076.

[16] BREUER J,VIGNER V,ROZTOČIL J. Precise Packet Delay Measurement in an Ethernet Network [J]. Measurement,2014(54):215-221.

[17] EIDSON J C,FISCHER M,WHITE J. IEEE-1588™ Stanard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems[C]// Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting. Reston,Virginia,2002:243-254.

[18] CHEN X,CHASAKI D,WOLF T. External Monitoring of Highly Parallel Network Processors[C]// Proceedings of the 2013 IEEE 14th International Conference on High Performance Switching and Routing(HPSR). IEEE,2013:197-204.

[19] BASNIGHT Z,BUTTS J,LOPEZ JR J,et al. Firmware Modification Attacks on Programmable Logic Controllers [J]. International Journal of Critical Infrastructure Protection,2013,6(2):76-84.

[20] SCHÖLKOPF B,PLATT J C,SHAWE-TAYLOR J,et al. Estimating The Support of a High-Dimensional Distribution[J]. Neural Computation,2001,13(7):1443-1471.

[21] MATJELO N J,MOKHOMO M. Gaussian Mixture Model Fitting Using Differential Linear Regression[J/OL]. International Research Journal of Engineering and Technology (IRJET),2021,8(7). https://www. irjet. net/archives/V8/i7/IRJET-V8I7253. pdf.

[22] KATOCH S,CHAUHAN S S,KUMAR V. A Review on Genetic Algorithm: Past, Present, and Future [J]. Multimedia Tools and Applications,2021,80(5):8091-8126.

[23] VAPNIK V N. An Overview of Statistical Learning Theory [J]. IEEE Transactions on Neural Networks,1999,10(5):988-99.

CUI Jingsong, born in 1975, Ph.D, associate professor, master supervisor, is a member of China Computer Federation. His main research interests include information security, cloud security and chip security.

GUO Chi, born in 1983, Ph.D, professor, Ph.D supervisor, is a senior member of China Computer Federation. His main research interests include Beidou application, unmanned system navigation and location-based service.

(责任编辑:杨雪敏)