毕业设计（论文）

课题名称 物联网设备

入侵检测系统研究

摘 要

随着人工智能、大数据、5G、物联网等新兴技术的发展，物联网设备进入到千家万户里。相关数据显示，在2015年至2025这十年间，网络攻击引发的全球潜在经济损失可能高达2940亿美元，而预计在2025年，中国市场的物联网支出也才3000亿美元。国家互联网应急中心从恶意程序、漏洞隐患、移动互联网安全、网站安全以及云平台安全、工业系统安全、互联网金融安全等方面，对我国互联网网络安全环境开展宏观监测。数据显示，与 2018 年上半年数据比较，2019 年上半年我国境内通用型“零日”漏洞收录数量，涉及关键信息基础设施的事件型漏洞通报数量，遭篡改、植入后门、仿冒网站数量等有所上升。

为了对入侵行为进行积极地响应，本文主要针对基于Linux操作系统的物联网设备，研究了基于网络的入侵检测系统。重点研究了对网络数据包的捕获，入侵检测的实现，使用PyQt开发一个用户友好的GUI来展示捕获的流量数据以及入侵检测结果。通过预先设计的实验，本文评估了入侵检测系统在检测某些类型的攻击中的可用性，如常见的扫描攻击。结果表明，该入侵检测系统能够以高度的可靠性和效率准确地检测到攻击并发出预警。

关键词：物联网设备 入侵检测 流量捕获 PyQt

**Research on IoT Device Intrusion Detection System**

**ABSTRACT**

With the development of artificial intelligence, big data, 5G, Internet of things and other emerging technologies, Internet of things equipment into the home. According to the data, the potential global economic damage caused by cyber attacks could reach $294 billion between 2015 and 2025, and it is expected that in 2025, the Chinese market only spends $300bn on the Internet of things. National Internet emergency response center from malicious programs, vulnerability, mobile Internet Security, website security and Cloud Platform Security, industrial system security, internet financial security, etc. , to Our Country Internet network security environment carries on the macroscopic monitoring. The data show that, compared with the first half of 2018, the number of generic “Zero-day” vulnerabilities in our country in the first half of 2019, and the number of incident-type vulnerabilities involving critical information infrastructure notifications, tampering, implanting a back door, the number of counterfeit sites has risen. In order to respond positively to the intrusion, this paper mainly studies the network-based intrusion detection system for the IOT devices based on Linux operating system. It focuses on the capture of network packets and the implementation of intrusion detection, and uses PyQt to develop a user-friendly GUI to display the captured traffic data and the results of intrusion detection. Through pre-designed experiments, this paper evaluates the availability of intrusion detection systems in detecting certain types of attacks, such as common scanning attacks. The results show that the intrusion detection system can detect attacks and give early warning with high reliability and efficiency.

**Keywords:** IoT device; Intrusion detection; Traffic capture; PyQt

目 录

[摘 要 I](#_Toc19299)

[ABSTRACT II](#_Toc29852)

[第一章 绪论 1](#_Toc24964)

[1.1 研究背景 1](#_Toc14534)

[1.2 研究现状 2](#_Toc4611)

[1.3 主要研究内容和论文组织结构 3](#_Toc32368)

[第二章 入侵检测系统概述 4](#_Toc20779)

[2.1 入侵检测系统定义 4](#_Toc20394)

[2.2 入侵技术 4](#_Toc29708)

[2.3 入侵检测技术 6](#_Toc23415)

[2.4 入侵检测系统模型 6](#_Toc19184)

[2.5 入侵检测系统分类 7](#_Toc24840)

[2.6 本章小结 10](#_Toc12327)

[第三章 入侵检测系统设计 11](#_Toc4550)

[3.1 入侵检测系统设计原理 11](#_Toc13927)

[3.2 系统设计要求 11](#_Toc2231)

[3.3 系统模块化设计与分析 12](#_Toc15137)

[3.4 系统开发环境 13](#_Toc10754)

[3.5 本章小结 14](#_Toc1138)

[第四章 入侵检测系统实现与测试 15](#_Toc5022)

[4.1 网络数据包捕获模块实现 15](#_Toc3988)

[4.2 入侵规则模块实现 17](#_Toc20594)

[4.3 入侵检测模块实现 17](#_Toc7175)

[4.4 响应模块实现 19](#_Toc18796)

[4.5 数据包存储模块实现 20](#_Toc8388)

[4.6 界面管理模块实现 21](#_Toc3160)

[4.7 入侵检测系统测试 23](#_Toc1075)

[4.8 本章小结 28](#_Toc32393)

[第五章 总结与展望 29](#_Toc29895)

[参考文献 30](#_Toc386)

[附 录 31](#_Toc28511)

[摘 要 33](#_Toc588)

[致 谢 35](#_Toc10729)

# 绪论

## 研究背景

在大数据信息技术的迅猛发展下，物联网技术的发展也同样突飞猛进。物联网技术作为互联网技术的延伸，已经成为世界信息产业的第三次浪潮。根据IDC数据显示，2021年全球物联网支出将达到7542.8亿美元，并有望在2025年达到1.2万亿美元，其中，中国市场规模将在2025年超过3000亿美元，全球占比约26.1%[1]。而据GSMA预测，全球的物联网设备将在2025年达到252亿台，物联网市场发展前景十分巨大。

但是，物联网技术的发展具有两面性：一方面给人类社会的发展带来便捷，另一方面也为攻击者提供了多样的攻击手段和攻击场所。相对于传统的PC设备，新兴的物联网设备性能较为低下，但它们给人们带来的威胁和资产损失不容忽视。比如基于Mirai的僵尸网络正慢慢地将物联网设备纳入其僵尸主机的范围，并将物联网设备用于DDoS攻击，其攻击频率以及攻击峰值越来越高[2]。在2018年的重大安全事件中，攻击者通过漏洞编写了恶意软件感染大量的物联网设备，在暗网进行攻击服务买卖，肆意对物联网设备进行攻击和破坏，以此勒索个人或者企业。全球爆火的家庭安防硬件产品Ring被爆出安全漏洞，黑客不仅能暴露用户的WiFi密码，更能通过摄像头监控用户家庭，大量用户的私生活照被黑客发到网上，给用户带来了严重的困扰和威胁。这些对物联网设备发动的攻击或由处于僵尸网络的物联网设备发动的攻击的行为，对国家重要基础设施、大小型企业和个人的安全构成了严重的威胁，物联网安全总体态势十分严峻，缓解或解决物联网威胁的道路仍十分遥远。

为了保证设备系统和网络资源的安全，更有效地发现入侵行为，专业组织和安全研究机构正致力于提出并完善安全防护方案，如增强访问控制、设置防火墙、使用VPN和病毒防护等。虽然这些网络安全技术能对网络安全提供重要的功防护能，但是这些技术也存在不少的适用缺陷。例如，身份认证技术就难以抵御字典攻击、弱口令攻击、特洛伊木马攻击等攻击方式。防火墙技术虽然拥有强大的身份认证技术和访问控制技术，但是来源于防火墙内部的攻击却不能被防火墙防御，未经更新的防火墙难以防备新出现的攻击方式，也无法阻断绕过防火墙的攻击[3]。VPN技术可以保证网络传输中的信息安全，但不能防御Dos攻击和缓冲区溢出等攻击。总的来说，上述技术属于静态安全技术范畴，它们只能消极地抵御入侵，不能主动地对入侵行为进行检测和分析。

网络入侵攻击的现状突出了优先考虑网络安全、采取积极措施保护信息和防止数据泄露的重要性。网络数据包是网络通信的基本单位，它包含了通信的所有信息，包括源地址、目标地址、协议、数据等[4]。详细分析网络数据包可以帮助人们了解网络通信的具体情况，包括网络流量、协议使用情况、数据传输速度、网络拓扑结构等方面的信息。

基于网络的入侵检测系统就是需要通过捕获网络传输中的网络数据包，对网络数据包进行详细分析，找出潜在的入侵行为，及时通知安全管理员，做出进一步的防范措施，保障目标设备的安全。同时，还可以帮助网络管理人员进行网络性能的优化，如检测网络拥塞情况、分析网络流量等，从而提高网络的质量和可靠性。

通过对网络数据包的详细分析，可以帮助网络管理员发现网络安全威胁，并及时采取措施进行应对，保障网络的安全性。同时，还可以帮助网络管理人员进行网络性能的优化，如进行网络拥塞情况分析、网络流量分析等，从而提高网络的质量和可靠性。

## 研究现状

入侵检测的研究最早可追溯到James Anderson在1980年的工作，他首先提出了入侵检测的概念。Anderson提出可以通过审计追踪来检测对文件的非授权访问，并给出了一些基本术语的定义，包括威胁、攻击、渗透和脆弱性等[5]。在1988年，为保障美国空军基地的计算机安全，在Lawrence Livermore实验室的努力研究下诞生了第一个IDS系统[6]。1980年到1990年是主机入侵检测系统奠基的十年，而1990年之后进入了快速发展的时代。1997年Agent第一个被应用于系统中，使得数据收集和数据判断可以在不同的主机上进行。2000年入侵检测系统IntruderAlert被Axent公司研发出来，并实现商业化[7]。而随着计算机和网络结构变得越来越复杂，已有的入侵检测系统难以满足安全防护需求。

随着近十年机器学习的不断发展，机器学习被应用于入侵检测之中。林琳等人通过对BP神经网络和日志分析进行结合得到了两层过滤的机制，增加检测的成功率，并利用该机制设计了主机入侵检测系统[8]。Badgujar等人在入侵检测中使用朴素贝叶斯分类方法，取得了不错的成果[9]。Morgan等人在入侵检测模型中使用人工神经网络方法，在检测率和误报率方面取得了斐然的成绩[10]。

并且，网络数据包分析通常是网络入侵攻击检测的关键组成部分，近期国内外在该领域有不少研究和发展。研究的重点是利用机器学习和人工智能进行数据包分析，开发数据包分类和识别的新算法，以及提高数据包分析工具和技术的性能等课题。在中国，已经有许多研究项目和倡议，旨在开发更先进的数据包分析方法，包括开发新的数据包捕获和分析的硬件和软件解决方案。其研究工作的重点是开发更有效的方法来检测和缓解网络安全威胁，如网络入侵攻击和数据泄露。在国际上，已经有大量的研究集中在应用机器学习和人工智能进行数据包分析，包括开发网络流量分类和识别的自动系统。同时，机器学习和人工智能在推进网络攻击检测方面发挥了重要作用，其能对网络流量和行为模式进行更先进和自动化的分析。且随着网络流量继续呈指数级增长，人们也越来越关注开发更多可扩展和高效的数据包分析工具和方法。

## 主要研究内容和论文组织结构

由于当前物联网技术的快速发展和物联网设备数量的不断增加，且市面上较多数的物联网设备使用的是Linux操作系统，该操作系统的物联网设备在配置入侵检测系统时，需要繁琐且复杂的系统配置过程，且人机交互方式多由命令行进行，十分不利于使用人员的操作。本论文探讨了基于网络的物联网设备入侵检测系统，试图通过捕获网络传输中的网络数据包，并对网络数据包进行分析以发现潜在的入侵行为，并在PyQt和Python库的基础上开发一个入侵检测系统，旨在为网络入侵攻击的实时检测和分析提供更强的能力，其友好的界面可以让安全分析人员快速识别和分析可疑的网络攻击数据，并对入侵或攻击行为进行报警。

本论文对课题的研究分为五大部分：

第一章，主要分析了课题的研究背景和国内外的研究现状，以及课题的研究意义。

第二章，主要讨论了入侵技术、入侵检测技术和各类型的入侵检测系统。

第三章，主要讨论了入侵检测系统原理、入侵检测系统要求、入侵检测系统各个模块的设计分析和入侵检测系统的开发环境。

第四章，主要讨论入侵检测系统的实现步骤与入侵检测系统的测试。

第五章，对实现的入侵检测系统进行总结，并对未来的入侵检测系统提出展望。

# 入侵检测系统概述

## 入侵检测系统定义

任何企图损毁目标资源的完整性、有效性和保密性的行为以及违背目标的各种安全策略和手段的行为都被称为入侵。而入侵检测则是对操作系统、网络数据包、应用程序、系统程序等信息进行收集，并检测和监视潜在的违背安全策略的行为。

入侵检测系统（IDS）是集多种功能于一身的动态防护技术。入侵检测系统的主要功能有[11]：

1. 监视与分析用户及系统的活动；
2. 目标系统的构造和日志的审计；
3. 识别反应已知的进攻模式并向系统管理员报警；
4. 异常行为模式的统计与分析；
5. 评估重要系统和相关数据的完整性；
6. 对操作系统进行审计跟踪与识别用户违反安全策略的行为。

## 入侵技术

随着越来越多物联网产品的产生以及入侵技术的开源，黑客的入侵行为也日渐猖獗，让人防不胜防。如果能了解黑客采用的入侵技术和惯用手段，将极大地加强我们对入侵检测技术的研究深度，完善入侵检测系统。目前，黑客常用的入侵技术有：

1. 端口扫描与漏洞扫描

由于TCP/UDP与应用程序是通过端口进行连接的，所以服务端可以预先设定固定的端口以等待客户端发送连接请求。当设备的端口开启等待连接时，黑客可以向目标设备发送一组端口扫描消息，以此了解设备的弱点以及可攻击方式。此时开放的端口如同一个入侵通道，成为黑客入侵的帮手。

1. 缓冲区溢出攻击

当应用程序向缓冲区发送的数据大小超过了缓冲区预设的容量大小时，溢出来的数据将会覆盖其他合法的数据[12]。缓冲区溢出类型中最危险的是堆栈溢出，黑客可以利用堆栈溢出，在函数返回时改变返回程序的地址，当函数跳转到任意地址时，会导致程序崩溃，拒绝服务，当函数跳转到一段恶意代码，比如得到了shell，黑客可以为所欲为，严重危害设备安全。

1. 拒绝服务攻击（DoS攻击）

Denial of Service被称为拒绝服务攻击，该攻击方式是通过发送大量的数据给用户，使得该用户大量的系统共享资源被占用，而系统的共享资源是有限的，其他用户无法得到系统的共享资源。典型的DoS攻击有SYN Flooding，其攻击原理是：黑客向服务器的各个端口发送大量SYN数据包，服务器收到了这些无用的数据包，然而这些数据包的地址是伪造的，当服务器想要回复SYN/ACK数据包时，无法找到发送的目标用户，也不能通过向黑客的客户端发送RST数据包来关闭连接。在超时之前，黑客会发送重复的SYN请求给服务端，使得服务器几乎完全忙于处理黑客的连接请求，无法处理正常的SYN数据包，合法用户无法正常使用服务器的共享资源。

1. 分布式拒绝服务攻击（DDoS攻击）

Distributed Denial of Service Attack被称为分布式拒绝服务攻击，它借助客户/服务器技术，把多个设备组合起来作为一个攻击系统，对一个乃至多个目标发动拒绝服务攻击，使得原来单一的拒绝服务攻击的威力成倍地提升[13]。当黑客想要进行攻击时，会在一个设备安装DDoS主控程序，该主控程序在一个设定好的时间与大量被感染的设备通讯，黑客发送攻击指令后，被感染的设备就会向目标设备发动攻击。而利用客户/服务器技术，该主控程序可以在短时间内多次激活被感染的设备，从而对目标设备进行攻击。

1. 特洛伊木马攻击

特洛伊木马是一种后门程序，一个完整的特洛伊木马程序有两个部分：控制端和服务端。当木马程序被植入服务端后，其会在系统启动或者宿主程序被执行时自主启动，不被使用者察觉。当黑客使用控制端向服务端特定端口发送数据时，木马程序会将这些数据解释为控制命令，并在目标系统执行某一些操作，以达到窃取用户隐私或进入目标系统进行操作等目的。

1. Sniffer攻击

Sniffer通常运行在路由器或者拥有路由功能的主机上，以便于监听大量的流量。当发现符合条件的数据包，就会把它们存到一个log文件，黑客通常是为了得到用户的账号和密码，或者是重要的金融信息等等。为了保证Sniffer能正常执行，黑客还会获得系统的控制权，以便留下再入侵的后门。

1. 欺骗攻击

欺骗攻击通常分为ARP欺骗、IP地址欺骗、DNS欺骗、电子邮件欺骗和Web欺骗，如IP地址欺骗，黑客通过伪造一种数据包，将数据包的源IP地址设置为目标主机信任的IP地址，从而获得目标主机的许可，黑客便可以访问到目标主机上的资源，并且该攻击是难以溯源的。

## 入侵检测技术

1. 基于模式匹配的入侵检测

模式匹配对已公开的入侵方式和入侵规则做出了明确的描述。当对系统日志、应用程序日志、审计日志或者网络数据包分析后，得到的数据模式与预设的入侵事件模式相匹配时，产生响应信号。目前各类系统对入侵事件的描述规则非常细致，应用该入侵检测技术的入侵检测系统比较广泛，且对已知入侵事件的检测率非常高，但无法检测出未知的入侵事件，有一定的局限性。

1. 基于专家系统的入侵检测

专家系统是对模式匹配功能多样性的完善。它通过if-then结构对相关的入侵事件、入侵规则、入侵处理进行分类，if的结构内容为入侵事件和入侵规则，then的结构内容为相应的处理措施[14]。专家系统的效用性取决于知识库的完备性和时效性与处理措施的可行性，只有及时更新知识库才行更好地检测入侵事件并根据相应措施进行处理。

1. 基于神经网络的入侵检测[15]

神经网络技术以典型入侵事件的样本作为学习库，应用模式识别算法进行训练，对每类的入侵特征模型进行加权，对捕获的网络数据进行处理，得到判别结果。还可以对未知的入侵事件进行学习，将未知的入侵事件加入到样本库中并不断优化加权，完善入侵检测模型。但此技术需要大量的入侵事件样本，不断优化训练模型才能取得较好的效果。

1. 基于协议认证的入侵检测[16]

协议认证技术通过预设一个严格的协议标准来检查入侵事件，防止黑客使用不正常的协议攻击系统，但是因为各类协议的实现方法不同，极大地影响了预设协议标准的可复用性，可能会导致误报率提高。

1. 基于统计模型的入侵检测

该方法是应用得最早的一种入侵检测方法。它以系统用户的使用行为作为统计模型，根据统计量的偏差来检测入侵行为。但是该方法有较高的误报率，当系统用户改变使用习惯后，原本生成的统计模型就会失效，需要重新建立统计模型。且该方法无法应用在其它的主机上，复用率十分低。

## 入侵检测系统模型

1987年，Dorothy Denning提出了一个通用入侵检测系统模型，虽然该系统模型与实际系统和输入无关，但是对之后研发的入侵检测系统的借鉴价值意义重大[17]。 图2-1描述了该模型的实现结构。

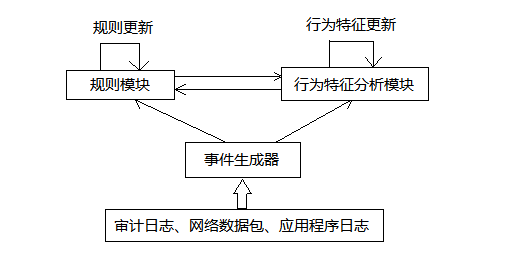


图2-1 通用入侵检测系统模型

在此模型中，由系统产生的审计日志、网络中的数据包、应用程序产生的日志可作为事件产生器，这些数据是入侵检测分析的基础。由系统的安全策略、入侵规则等模型组成规则模块，不仅能够为入侵事件提供规则参考，还能够记录事件的各种状态并更新规则模块和其他模块的内容。行为特征分析模块是该系统模型的核心，它对系统用户和系统本身产生的所有行为特征进行记录与分析，特征行为分析则是根据具体的系统环境、统计方法以及分析方法而确定的，并且根据特征行为分析更新该模块和其他模块。虽然在实际实现时，这两个模块的形式与方式不太一样，但总的来说，规则模块是对知识库的分析，行为特征分析模块是对行为库的分析。

## 入侵检测系统分类

### 基于网络的入侵检测系统

对基于网络的入侵检测系统（NIDS）来说，它的事件产生器数据来源就是网络数据包[18]。NIDS通过协议分析、网络流量分析得到数据，利用如特征匹配或者异常识别技术来检测入侵攻击，并产生响应信息。

NIDS的系统结构如图2-2所示。

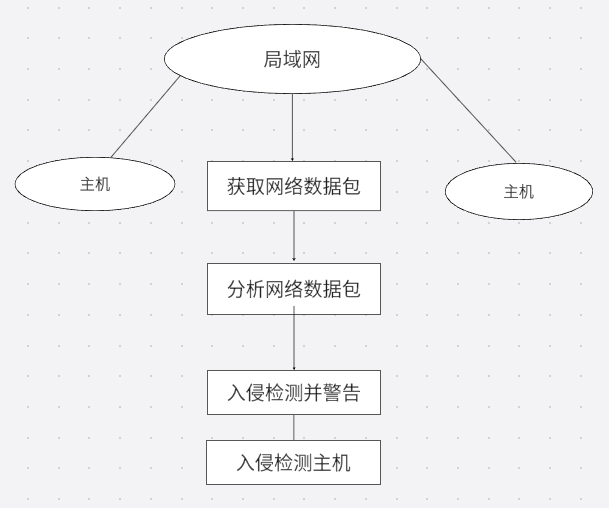


图2-2 基于网络的入侵检测系统

NIDS的优点：可以检测来自非法访问和网络的攻击。对主机的性能影响比较小，不需要改变主机的配置。

NIDS的缺点：采用如特征检测、协议分析等检测方法，漏报率与误报率可能较高。没有主动的抵御手段，系统的检测规则的更新总是落后于攻击手段的更新。系统不能识别数据的来源，不能定位IP地址。

### 基于主机的入侵检测系统

基于主机的入侵检测系统（HIDS）作为主机系统的监视器与分析器，并不能作用到外部的接口，而是作用于系统内部，它根据主机系统的系统日志和审计日志对发生的入侵事件进行检测分析，并且在事后得出分析结果阻止接下来可能的入侵事件[19]。

HIDS的系统结构如图2-3所示。

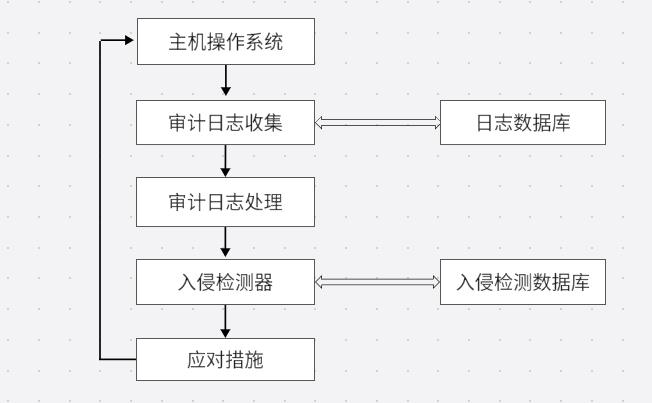


图2-3 基于主机的入侵检测系统

HIDS的优点：有些攻击并不通过网络传输，而是在本地进行，只能使用HIDS。可以根据已有的入侵事件信息判断攻击是否成功。能够监视系统的特定行为，即使是用户的登录和退出以及其中包含的所有操作。

HIDS的缺点：监视工作只能发生在已部署的主机系统中。主机系统数量越多，工作量越大。对系统的要求较高，并且需要配置好系统日志功能。

### 分布式入侵检测系统

分布式入侵检测系统（DIDS）采用C/S结构模式，通过配置服务器端，实现对局域网中所有的客户机的控制[20]。该系统整合了NIDS和HIDS的优点，策略定义与管理都统一由服务器来控制，极大地解决集中式入侵检测系统存在的问题。DIDS的结构如图2-4所示。

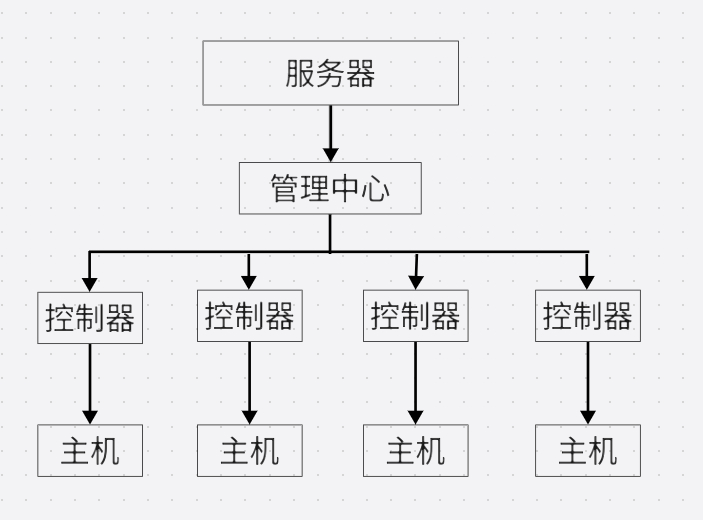


图2-4 分布式入侵检测系统

## 本章小结

本章论述了入侵检测系统的定义和系统功能，分析了常用的入侵技术、入侵检测技术以及入侵检测系统的三种类型与模型。就物联网设备而言，考虑其系统体量较小，使用的技术不应该对系统的性能有较大的影响，所以适用于基于网络的入侵检测系统，为课题的设计提供了研究方向。

# 入侵检测系统设计

## 入侵检测系统设计原理

随着物联网技术的不断发展，物联网设备越来越多，而物联网设备处于边缘区域，基于主机的入侵检测系统不仅需要日志审计功能，还需要对每个设备单独进行配置，工作量大，将极大地占用设备系统的效能。所以基于该现状，本文提出将基于网络的入侵检测系统应用到物联网设备中，对原本系统体量小的设备性能影响小，可以抵御大多数的网络入侵行为，工作量小。应用在物联网设备的基于网络的入侵检测系统原理如图3-1所示。

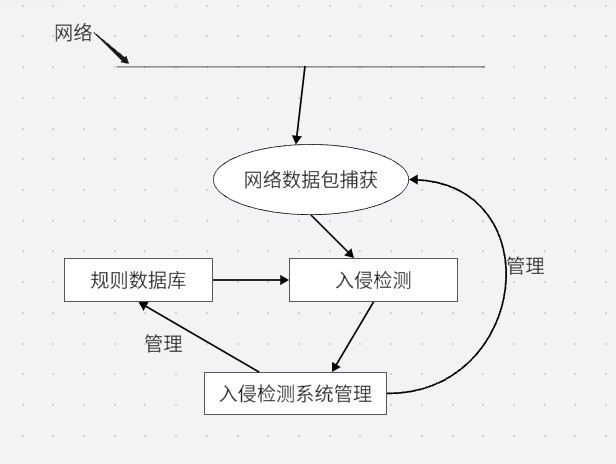


图3-1 基于网络的入侵检测系统原理

## 系统设计要求

对该物联网设备入侵检测系统而言，它需要满足以下的几个要求：

准确性要求：系统需要对入侵事件有一个准确的判断，误报率和漏报率低。

实时性要求：系统需要实时或者近乎实时地对入侵事件做出响应。

安全性要求：系统的运行不易成为黑客攻击对象，能持续地对网络进行监控。

拓展性要求：系统需要预设一种可以描述入侵事件的规则，该规则库是动态的，且根据攻击手段的复杂性和迭代性，不断地更新规则库中的内容，以对新类型入侵事件做出响应。

能效性要求：由于系统的体量小，小部署体量不会对系统的能效产生较大影响。

## 系统模块化设计与分析

### 入侵检测系统的总体结构

基于网络的入侵检测系统从总体结构上讲，主要分为三个部分：数据收集、数据处理和结果响应。从实现的功能模块上讲，主要分为六个模块：网络数据包捕获模块、入侵规则模块、入侵检测模块、响应模块、数据包存储模块、界面管理模块。功能模块如图3-2所示。

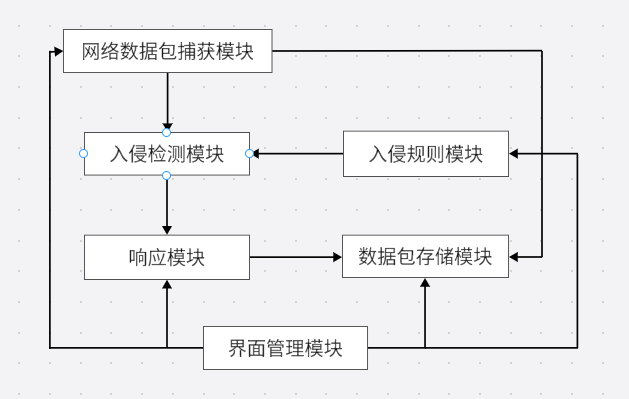


图3-2 系统功能模块

### 系统各模块分析

总体结构如图，3-3所示。

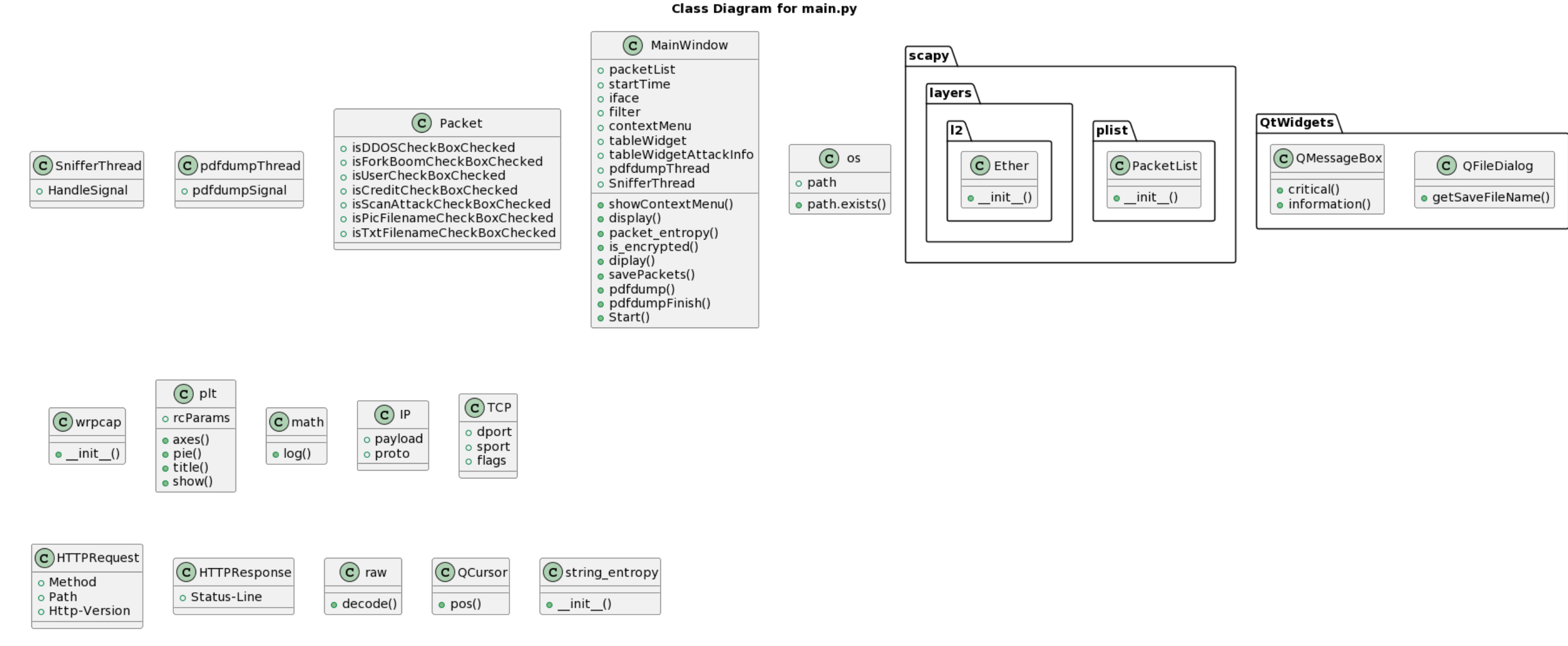


图3-3 系统功能模块

1. 网络数据包捕获模块

网络数据包捕获模块要设置好对应的网卡，通过监听网卡把网络上传输的数据包收集起来，然后传递给入侵检测模块，入侵检测模块通过分析操作判断该数据包是否为入侵数据包。所以网络数据包捕获模块是整个入侵检测系统的数据来源与分析前提。

1. 入侵规则模块

入侵规则模块是整个入侵检测系统的重点，其中的入侵规则库是入侵检测判断的知识来源。入侵规则库中包含了多种入侵事件，并对这些事件进行了规则表述，本文以入侵特征关键字作为入侵规则，当需要进行入侵检测时，就将规则库中的特征关键字读出。

入侵规则模块中最主要就是建立入侵规则库，且入侵规则库的完备性是入侵检测系统能否对多样的入侵行为进行检测的基础。

1. 入侵检测模块

入侵检测模块是入侵检测系统的重点，该模块是通过预设好的入侵规则库，对入侵行为进行检测，判断该事件是否符合入侵规则库。该模块主要实现的功能是特征匹配功能，首先要从网络数据包中提取到关键信息，形成一个信息字符串，在从入侵规则库中提取入侵规则关键字字符串，使用字符串匹配方法进行匹配。当数据包的特征字符串与入侵规则关键字相匹配时，通过分析该入侵规则关键字所属的入侵事件，得到该入侵事件对应的入侵行为，并向响应模块发送信号。

1. 响应模块

响应模块是一个入侵检测系统必不可少的部分，没有该模块，入侵检测系统就失去了它应有的价值。它需要入侵检测系统完成入侵检测时做出的警告反应，及时提醒管理员目标设备受到了入侵攻击，需要采取措施进行新一轮的防御与入侵检测，减少出现重大损失的可能性，极大地保护了设备的运行安全。

1. 数据包存储模块

数据包管理模块是入侵检测系统的一种后手准备。在入侵检测系统完成入侵检测时，将分析得到的数据包进行储存，及时保护现场，使得安全管理员能够在这些捕获到的入侵数据包中分析出更为有用的信息，为可能的入侵行为变种做好预先准备，是一种积极的入侵行为应对方法。

1. 界面管理模块

界面管理模块是上述五个模块的整合与管理，可以通过简单操作实现网络数据包的获取，并能显示网络数据包的主要信息，如数据包的源地址、目的地址、协议信息、原始数据等等。通过友好的图形界面对入侵规则进行描述，更新入侵规则库。并且可以在界面中展示分析的入侵数据包，得知其有关的入侵行为，并产生警报。

该模块的实现将极大地便于操作人员的理解与使用，是对入侵检测系统设计的完善。

## 系统开发环境

### 系统开发语言及工具

Python是一种高级编程语言，由Guido van Rossum于1989年创建[21]。Python具有许多强大的标准库，在网络编程、多线程编程、图形用户界面编程、正则表达式和加密等方面提供了丰富的支持，且语言具有跨平台的特性，可以在多种操作系统下运行，包括Windows、Linux和macOS等，还支持多种解释器，包括CPython、Jython和IronPython等。

PyQt 是流行的Qt应用程序框架的一组Python绑定。它提供了范围广泛的类和工具，使图形用户界面的编程更加容易和高效，并具有一系列可自定义的小部件，包括按钮、表单、菜单和对话框，以及对动画、图形和多媒体、数据库和线程等高级功能的支持。

PySide6是一个跨平台GUI工具包Qt6的Python绑定。它为Python开发者提供了对完整的Qt6 API的访问，包括部件、图形、多媒体、网络等，使他们能够创建强大而复杂的桌面和移动应用程序。

Scapy是一个基于Python的数据包操作和分析库，允许用户以编程方式捕获、解码和发送网络数据包[22]。它支持广泛的网络协议，常用于网络侦察、调试和安全测试，也可用于开发自定义网络工具和应用程序，如数据包嗅探器、扫描器和嗅探引擎等。且其主要的优势之一是其灵活性和易用性。它提供了一个高层次的API，允许用户使用Python代码构建和操作网络数据包，同时也提供了一个低层次的API，让用户在数据包层面完全控制网络数据包。

### 系统环境

本系统的硬件配置如表3-1所示，软件配置如表3-2所示。

表3-1 硬件配置

|  |  |
| --- | --- |
| 硬件 | 配置 |
| 内存 | 16GB |
| 显卡 | 3070 |
| CPU | Intel CORE i7 |

表3-2 软件环境

|  |  |  |
| --- | --- | --- |
|  | 名称 | 版本 |
| 运行操作系统 | Ubuntu | 22.04 |
| 代码编辑器 | Vim/vscode | 21 |
| 框架 | PyQt | 5 |

## 本章小结

本章论述了入侵检测系统的设计原理与系统的功能性要求，拆分系统的各个功能为模块，对每个模块进行分析，通过模块化的设计完成功能的设计，并就使用到的技术系统设计环境做一个简单的说明。

# 入侵检测系统实现与测试

**由上一章节对入侵检测系统各模块的设计思路，本章节将实现网络数据包捕获模块、入侵规则模块、入侵检测模块、响应拨快、数据包存储模块、界面管理模块，实现对各个模块功能的整合，并对整个入侵检测系统进行功能测试。**

## 网络数据包捕获模块实现

**网络数据包捕获模块是整个入侵检测系统的前提，能否正常捕获网络传输中的数据包，并处理其中的数据，将关乎相关模块功能的正常与否。**

网络数据包捕获模块的实现有多个步骤，实现流程如图4-1所示。

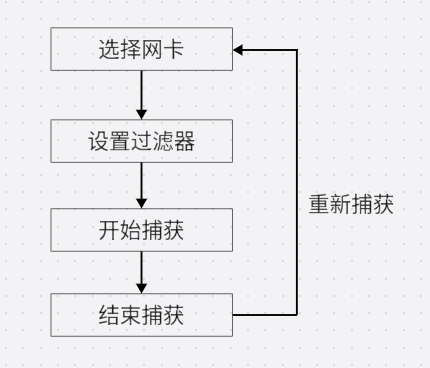


图4-1 数据包捕获流程

由数据包的捕获流程可得，该模块的实现具体步骤如下：

1. 网络数据包结构体

首先需要定义网络数据包结构体，用于存储数据包信息，主要代码如下：

class Packet:

srcIP = ''

protocol = ''

rawData = ''

flags = 0x00

def \_\_init\_\_(self, in\_packet):

self.srcIP = str(in\_packet[IP].src)

self.protocol = int(in\_packet.proto)

self.rawData = str(in\_packet)

self.flags = in\_packet[TCP].flags

其中srcIP、protocol、rawData、flags、in\_packet分别表示IP地址、协议类型、原始数据、标志和捕获到的数据包。

1. 选择网卡

对网络中传输的数据包进行捕获，首先要选择合适的网卡，该方法会根据路径寻找系统文件，找到网卡信息并读取出来。主要代码如下：

def LookupIface(self):

netcards = os.listdir('/sys/class/net/')

self.comboBoxIface.addItems(netcards)

其中addItems函数是为了将系统上搜索到网卡显示到页面中。

1. 数据包过滤器

**数据包捕获模块可以将所有经过该设备的网络数据包截获，其中也包括有着不同协议和不同端口的网络数据包。但是在系统真正投入使用时，某些数据是用户不需要的，或者是可以忽略的，它们被统称为垃圾数据。垃圾数据在流量捕获过程中占据着极大的比重，捕获这些垃圾数据将极为影响系统的运行性能。因此，一个可行且效率高的数据包过滤机制是数据包捕获模块的重要部分，它可以让用户指定一个特定的IP地址，只获取该IP地址的有关数据包，或者一个特定的端口，只获取该端口的相关数据包，从而能提升入侵检测系统的检测效率。**

PreFilter函数是数据包过滤器，可以指定源IP地址、目的IP地址、源端口、目的端口、协议类型、也可以自定义过滤规则，其中实现的代码如下：

**def setPreFilter(self,filter):**

**self.filter = filter**

def PreFilter(self):

list = ["指定源IP地址","指定目的IP地址", "指定源端口","指定目的端口","指定协议类型","自定义规则"]

item, ok = QInputDialog.getItem(self, "选项","规则列表", list, 1, False)

if ok:

if item=="指定源IP地址":

filter,ok\_1 = QInputDialog.getText(self, "标题","请输入指定源IP地址:",QLineEdit.Normal, "\*.\*.\*.\*")

rule = "src host "+filter

elif item =="指定目的IP地址" :

filter,ok\_2 = QInputDialog.getText(self, "标题","请输入指定目的IP地址:",QLineEdit.Normal, "\*.\*.\*.\*")

rule= "dst host "+filter

**elif item =="指定源端口":**

**filter,ok\_3 = QInputDialog.getInt(self, "标题","请输入指定源端口:",80, 0, 65535)**

**rule="src port "+str(filter)**

**elif item =="指定目的端口":**

**filter,ok\_4 = QInputDialog.getInt(self, "标题","请输入指定目的端口:",80, 0, 65535)**

**rule ="dst port "+str(filter)**

**elif item =="指定协议类型" :**

**filter,ok\_2 = QInputDialog.getText(self, "标题","请输入指定协议类型:",QLineEdit.Normal, "icmp")**

**rule =filter**

**elif item =="自定义规则":**

**filter,ok\_2 = QInputDialog.getText(self, "标题","请输入规则:",QLineEdit.Normal, "host 202.120.2.1")**

**rule=filter**

**rule=rule.lower()**

**self.setPreFilter(rule)**

1. 网络数据包捕获过程

定义一个嗅探线程类，包含的关键信息为过滤规则以及网卡信息，主要代码如下：

class SnifferThread(QtCore.QThread):

HandleSignal = QtCore.pyqtSignal(scapy.layers.l2.Ether)

def \_\_init\_\_(self,filter,iface):

super().\_\_init\_\_()

self.filter = filter

self.iface = iface

def run(self):

sniff(filter=self.filter,iface=self.iface,prn=lambda x:self.HandleSignal.emit(x))

创建一个数据包捕获线程，即SnifferThread函数，需要传入的参数是过滤规则和网卡信息，主要代码如下：

def Start(self):

global count

count = 0

global display

display = 0

self.packetList = []

self.startTime = time.time()

self.iface = self.comboBoxIface.currentText()

self.tableWidget.setRowCount(0)

self.tableWidget.removeRow(0)

self.SnifferThread = SnifferThread(self.filter,self.iface)

self.SnifferThread.HandleSignal.connect(self.display)

self.SnifferThread.start()

使用terminate函数销毁进程，停止捕获，主要代码如下：

def Stop(self):

self.SnifferThread.terminate()

1. **显示过滤数据包**

**def PostFilter(self):**

**filter,ok = QInputDialog.getText(self, "过滤前需要停止抓包","请输入需要搜索的字段:",QLineEdit.Normal, "http")**

**if ok:**

**global display**

**display = 0**

**if filter == None:**

**for row in range(self.tableWidget.rowCount()):**

**self.tableWidget.setRowHidden(row,True)**

**display += 1**

**else:**

**for row in range(self.tableWidget.rowCount()):**

**if(self.tableWidget.item(row, 7))!=None:**

**p = self.tableWidget.item(row,7).text()**

**packet = scapy.layers.l2.Ether(p.encode('Windows-1252'))**

**f = open('search.tmp','w')**

**old = sys.stdout**

**sys.stdout = f**

**packet.show()**

**sys.stdout = old**

**f.close()**

**f = open('search.tmp','r')**

**data = f.read()**

**f.close()**

**os.remove('search.tmp')**

**obj= re.search(filter.lower(),data.lower())**

**if obj is None:**

**self.tableWidget.setRowHidden(row,True)**

**display += 1**

**else:**

**self.tableWidget.setRowHidden(row,False)**

**其中，sys.stout将当前系统输出存储到临时变量中，再将输出重定向到文件。**

## 入侵规则模块实现

**入侵规则模块是整个入侵检测系统的重点，该模块的完备与否将决定着入侵检测行为完整与否，其中主要的就是对入侵特征关键字的构建，部分代码如下：**

NULL\_FLAG = 0b00000000

FIN\_FLAG = 0b00000001

XMAS\_FLAG = 0b00101001

NIKTO\_KEYWORDS = ["Nikto", "nikto", "NIKTO"]

SHOCK\_KEYWORDS = ["() { :; };", "(){:;};", "() { :;};", "() { : ; } ;", "() {:; };"]

**USER\_KEYWORDS = ["mac", "log", "login", "user", "user\_name", "pseudo", "username", "\_username", "userid", "loginname", "login\_id", "loginid", "id", "user\_id", "uname", "account", "member", "login\_username"]**

**PASS\_KEYWORDS = ["pass", "pass password", "\_password passwd", "session\_password", "login\_password", "loginpassword", "userpassword", "pwd", "upassword", "login\_password"]**

其中，FLAG标志代表的是nmap扫描关键字，NIKTO\_KEYWORDS代表nikto扫描关键字，SHOCK\_KEYWORDS代表fork boom攻击关键字，**USER\_KEYWORDS代表有关用户名的关键字，PASS\_KEYWORDS代表有关用户密码的关键字。如果需要新建入侵特征关键字，则需要在rules文件夹中添加新的文本，其中，文本名代表入侵事件，文本内容代表入侵事件的有关特征关键字。**

## 入侵检测模块实现

代码的入侵检测部分在main.py文件中的sniff\_packet（）函数中实现。此函数接收一个数据包，并检查NULL、FIN或XMAS nmap隐形扫描、Nikto扫描、Shellshock攻击、清除中发送的凭据和清除中发送信用卡号的痕迹。如果检测到其中任何一个，则会调用print\_alert（）函数来打印包含检测到的入侵相关信息的警报。Packet类用于保存数据包中的信息，以便不需要传递Scapy数据包的实例。

以下函数可以获取入侵规则库中的入侵特征关键字，获取nikto扫描、fork boom攻击、用户名、用户密码、自定义规则关键字的主要代码如下：

def load\_rules(filename):

with open(filename, 'r') as f:

lines = f.readlines()

for line in lines:

rule = line.strip()

if "nickto" in filename or "NICKTO" in filename:

NIKTO\_KEYWORDS.append(rule)

if "shock" in filename or "SHOCK" in filename:

NIKTO\_KEYWORDS.append(rule)

**if "user" in filename or "USER" in filename:**

**USER\_KEYWORDS.append(rule)**

**if "pass" in filename or "PASS" in filename:**

**PASS\_KEYWORDS.append(rule)**

**def btn\_choose\_rule\_dir\_clicked(self):**

**self.sig\_file\_path = QFileDialog.getExistingDirectory(None, "选择特征库文件目录", "")**

**print(self.sig\_file\_path)**

**box = QMessageBox()**

**box.setText("特征库选择成功->%s" % self.sig\_file\_path)**

**box.exec()**

**for file in os.listdir(self.sig\_file\_path):**

**box = QMessageBox()**

**box.setText("特征库成功->%s" % file)**

**box.exec()**

**load\_rules(self.sig\_file\_path + "/" + file)**

**如果使用暴力匹配法进行特征字符串匹配，将极大地影响系统的性能，所以一个好的匹配算法是实现该功能必不可少的一部分。该系统使用KMP算法作为字符串匹配算法，其中text为需要分析的数据，pattern为入侵规则特征字，具体代码如下：**

def kmp(text, pattern):

n = len(text)

m = len(pattern)

if m == 0:

return 0

fail = [0] \* m

j = 0

for i in range(1, m):

while j > 0 and pattern[j] != pattern[i]:

j = fail[j-1]

if pattern[j] == pattern[i]:

j += 1

fail[i] = j

j = 0

for i in range(n):

while j > 0 and pattern[j] != text[i]:

j = fail[j-1]

if pattern[j] == text[i]:

j += 1

if j == m:

return i - m + 1

return -1

该模块可以通过读取多个规则来判断是否有入侵行为，**以下是判断shock入侵行为以及获取用户名的主要代码：**

def get\_shock\_script(packet\_data):

shellshock\_line = ""

data = packet\_data.splitlines()

for line in data:

for keyword in SHOCK\_KEYWORDS:

if kmp(line, keyword) != -1:

shellshock\_line = line

break

return shellshock\_line

**def get\_username(raw\_data):**

**words = str(raw\_data).split()**

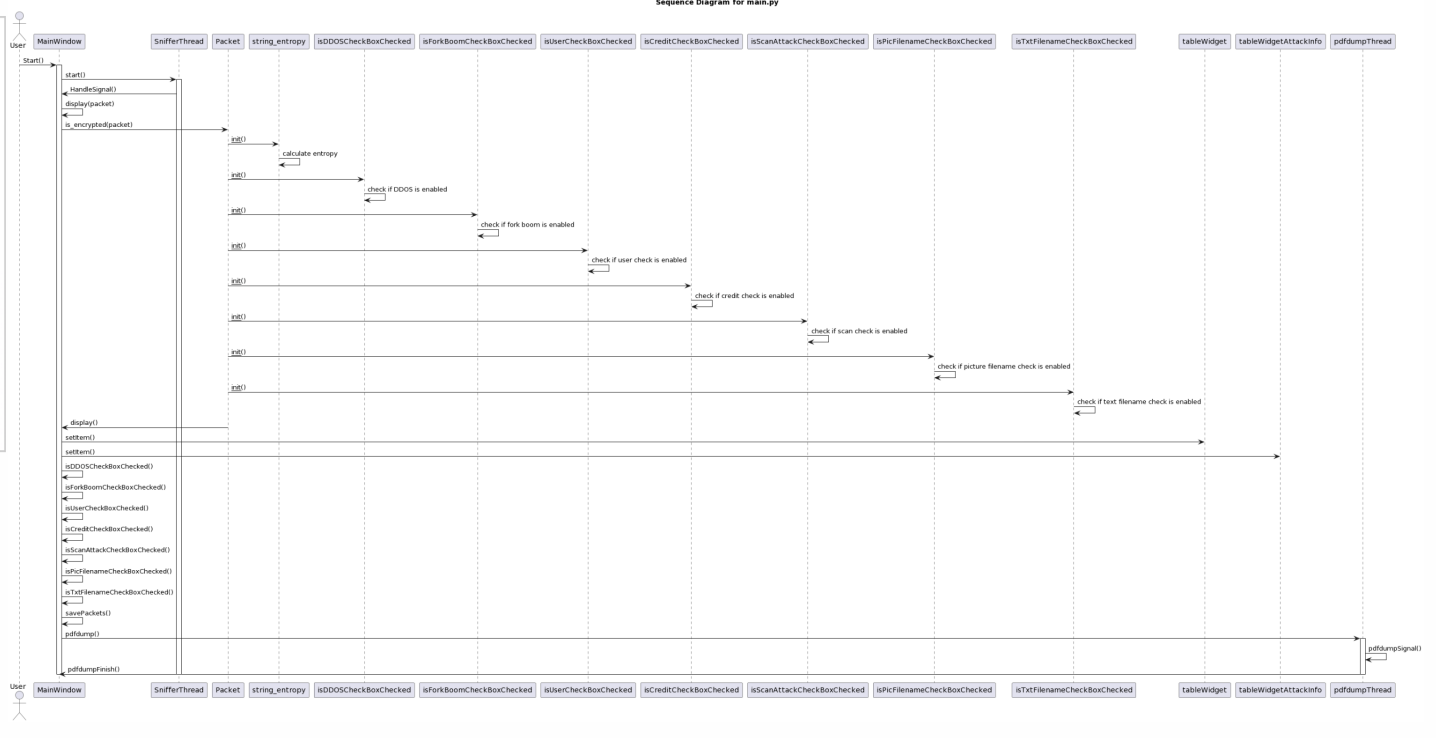
**for i in range(len(words)):**

**for keyword in USER\_KEYWORDS:**

**if kmp(words[i], keyword) != -1:**

**return words[i]**

**如图4-1所示，**

****

Start（）函数通过创建SnifferThread对象并启动它来初始化数据包捕获过程。SnifferThread对象捕获数据包并发出HandleSignal信号，该信号连接到MainWindow类中的display（）函数。display（）函数使用有关捕获的数据包的信息更新GUI，包括源和目的地IP地址、数据包长度以及数据包是否加密。is\_encrypted（）函数用于确定数据包是否加密。它使用string\_entropy（）函数计算数据包有效载荷的熵，并将其与阈值进行比较。如果熵高于阈值，则认为数据包已加密，并且函数返回True。否则，函数将返回False。函数的作用是将选定的数据包保存为PDF文件。它首先从tableWidget对象中检索数据包数据，并创建一个pdfdumpThread对象来处理PDF创建过程。当PDF创建完成时，pdfdumpThread对象会发出一个pdfdumpSignal信号，该信号连接到MainWindow类中的pdfdumpFinish（）函数。pdfdumpFinish（）函数显示一个消息框，指示PDF创建是否成功。总体而言，main.py中的代码提供了一个基本的数据包捕获和分析工具，并具有一些附加功能，例如将数据包保存为PDF文件。is\_encrypted（）函数可以通过使用更复杂的算法来检测加密来进行改进，但它为进一步开发提供了一个很好的起点。

## 响应模块实现

本模块是入侵检测系统中必不可少的一个模块，当没有该模块时，如果检测到入侵行为，却没有任何的响应与警告，使用人员将无法探查该系统是否正常工作，亦或者该系统对入侵行为没有检测能力，所以需要该模块来响应相关事件的发生。而响应模块有两种实现方法：一种是主动响应，主动响应就是在入侵检测系统检测出入侵行为时，主动对目标系统或者目标设备做出改动，防止入侵行为继续发生，比如及时断开网络连接，但是该响应行为有极大的弊端，如果入侵行为危害很小时，断开网络连接反而会影响到目标设备的正常工作，产生更大的损失。另一种是被动响应，被动响应是在入侵检测系统检测出入侵行为时，产生预警信息或者或者安全日志，提醒安全管理员处理入侵行为。比如发送报警信息到控制台，由安全管理员对该入侵行为进行处理。

本模块在会在控制台发送警告，并且将警告信息存入日志中，主要代码如下：

def print\_alert(scan\_type, src, proto, payload):

global ALERT\_COUNTER

ALERT\_COUNTER += 1

if payload == "":

print("ALERT #%d: %s is detected from %s (%s)!" % (ALERT\_COUNTER, scan\_type, src, PROTOCOLS[proto]))

logging.info("ALERT #%d: %s is detected from %s (%s)!" % (ALERT\_COUNTER, scan\_type, src, PROTOCOLS[proto]))

else:

print("ALERT #%d: %s from %s (%s) !" % (ALERT\_COUNTER, scan\_type, src, PROTOCOLS[proto] ))

logging.info("ALERT #%d: %s from %s (%s)!" % (ALERT\_COUNTER, scan\_type, src, PROTOCOLS[proto]))

## 数据包存储模块实现

此模块最主要的作用就是存储捕获到的网络数据包信息。因为在某一时间段内，捕获到的网络数据包数量十分庞大，所以必须通过一个方法把它们储存到文件中。而且存储数据包有许多益处，最主要的就是供日后对可疑的数据包或者检测到的入侵数据包进行分析，便于安全管理员发现入侵攻击变种，逐步完善入侵规则库，数据包存储的流程如图4-2所示。

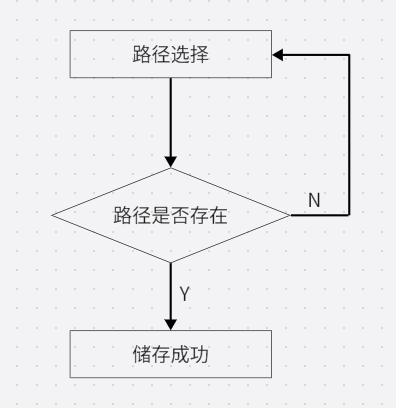


图4-2 数据包存储流程

数据包的获取是通过scapy中内置的函数实现的，通过os.path.existe函数判断存储路径是否存在。数据包存储模块的主要代码如下：

def savePackets(self):

path, filetype = QtWidgets.QFileDialog.getSaveFileName(None,"选择保存路径",

"./","pcap文件(\*.cap);;全部(\*)")

if path == "":

return

if os.path.exists(os.path.dirname(path)) == False:

QtWidgets.QMessageBox.critical(None,"错误","路径不存在")

return

packets = scapy.plist.PacketList(self.packetList)

wrpcap(path,packets)

QtWidgets.QMessageBox.information(None,"成功","保存成功")

## 界面管理模块实现

如何管理其他模块的交互与功能显示是界面管理模块的主要功能，它能将系统完成的功能与分析结果以行之有效的方式告知安全管理人员。

界面管理模块能提供操作简易、使用友好的图形用户界面。该模块可以使得用户通过简单的鼠标点击和简易的键盘输入即可完成对入侵检测系统的设置与操作，数据包的十六进制信息、协议类型、源IP地址、目的IP地址、原始数据、入侵事件的显示等等，都可以直观地显示出来，非专业人员在经过简单的培训后也能极好地设置、使用以及管理入侵检测系统。

它定义了数据包嗅探器应用程序的图形用户界面的布局和小部件。

Ui\_MainWindow类包含一个setupUi方法，用于设置应用程序的主窗口。该窗口的固定大小为1440x872像素，包含多个窗口小部件，包括一个表格窗口小部件、一个带有多个复选框的组框、一个文本浏览器、一个树窗口小部件和另一个表格小部件。

表格小部件显示捕获的数据包信息，包括数据包编号、时间、源和目的地地址、协议、长度以及数据包是否加密。组框包含用于过滤网络攻击的复选框，例如DDOS、用户名、信用卡信息、扫描攻击和分叉炸弹。

文本浏览器显示所选数据包的原始数据，而其下方的文本浏览器以更可读的格式显示数据包。树小部件以分层树结构显示数据包。最后一个表格小部件显示有关任何检测到的网络攻击的信息。

每次捕获数据包时都会调用显示功能，它会用有关数据包的信息更新GUI中的表，例如源和目的地IP地址、使用的协议和数据包的大小。

packet\_entropy函数计算IP分组的有效载荷的熵，并且is\_encrypted函数检查熵值是否高于特定阈值以确定分组是否被加密。

该脚本还包括几个功能，用于检查各种类型的网络攻击，如DDOS、shell shock和用户/通行证泄漏。对每个捕获的数据包调用这些函数，并使用有关任何检测到的攻击的信息更新GUI中的单独表。

最后，该脚本包括将捕获的数据包保存到文件的功能。启动功能开始在选定的网络接口上捕获数据包。

plt模块用于创建饼图，显示使用每个传输层协议（TCP、UDP、ICMP和IGMP）捕获的数据包的百分比。当用户单击GUI中的“传输层统计信息”按钮时，图表将显示在一个单独的窗口中。

使用槽函数实现了该模块对其他模块的管理功能，部分功能的主要代码如下：

def setSlot(self):

self.tableWidget.itemClicked.connect(self.clickInfo)

self.tableWidget.setContextMenuPolicy(Qt.CustomContextMenu)

self.tableWidget.customContextMenuRequested.connect(self.showContextMenu)

self.contextMenu = QMenu(self.tableWidget)

self.comboBoxIface = QComboBox()

self.toolBar.addWidget(self.comboBoxIface)

self.LookupIface()

startAction = QAction(QIcon('./img/start'),'&开始捕获(Ctrl+E)',self)

startAction.setShortcut('Ctrl+E')

startAction.triggered.connect(self.Start)

self.toolBar.addAction(startAction)

stopAction = QAction(QIcon('./img/stop'),'&停止捕获(Ctrl+Q)',self)

stopAction.setShortcut('Ctrl+Q')

stopAction.triggered.connect(self.Stop)

self.toolBar.addAction(stopAction)

其中，clickInfo代表的操作是鼠标左键点击，tableWidget用于创建主菜单，comboBoxIface用于调试出网卡信息，startAction代表的是开始捕获的操作，stopAction代表的是停止捕获操作，setShortcut函数用于设置操作的快捷键。

该界面使用的是PyQt6/PySide6框架，通过调用和个模块接口进行功能的实现，主程序通过将按钮和\_clicked()系列函数进行绑定，从而实现不同按钮和对应事件的触发。MainWindows UML如图4-3所示。

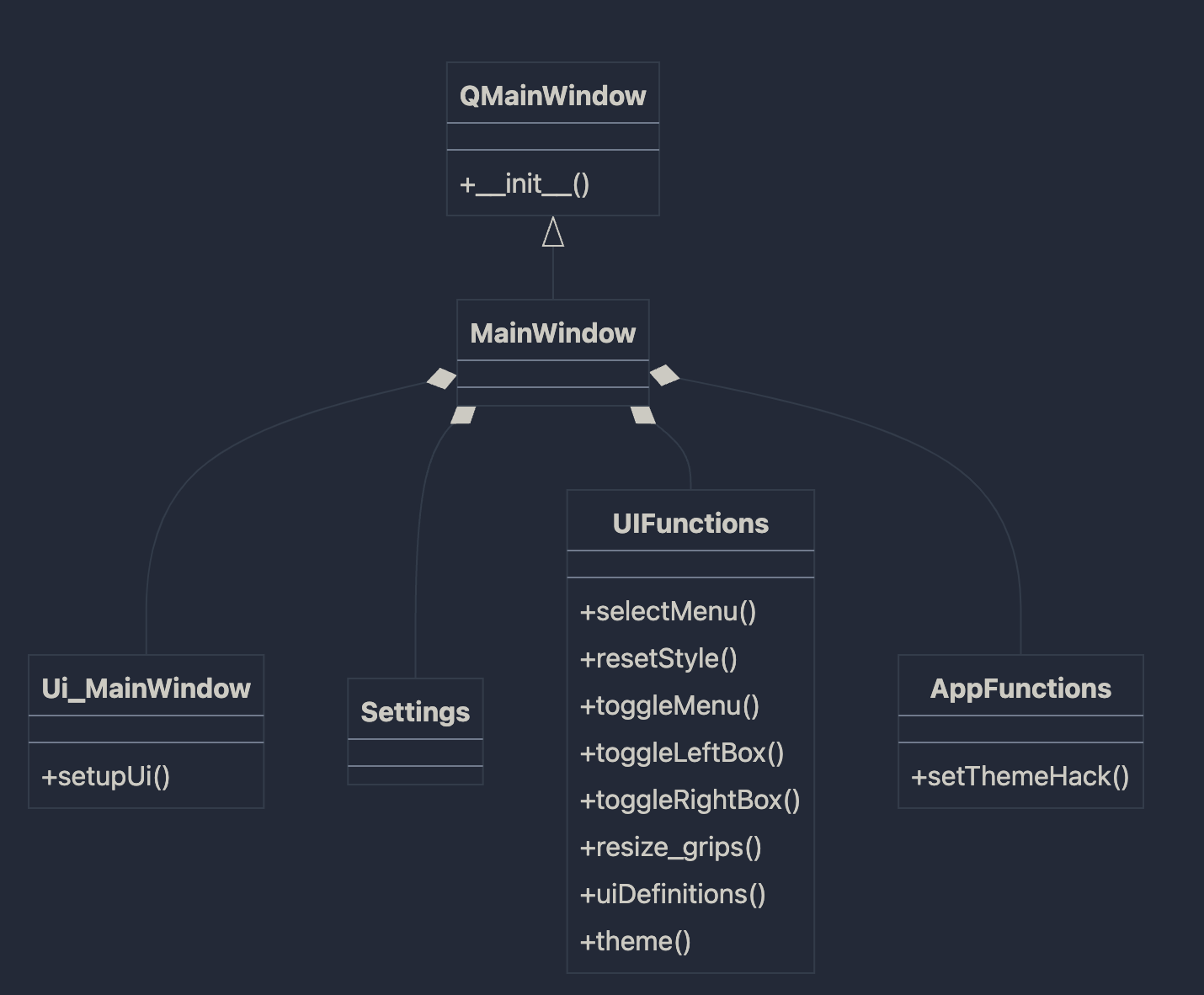


图4-3 MainWindows UML

整体界面的布局如图4-4所示。

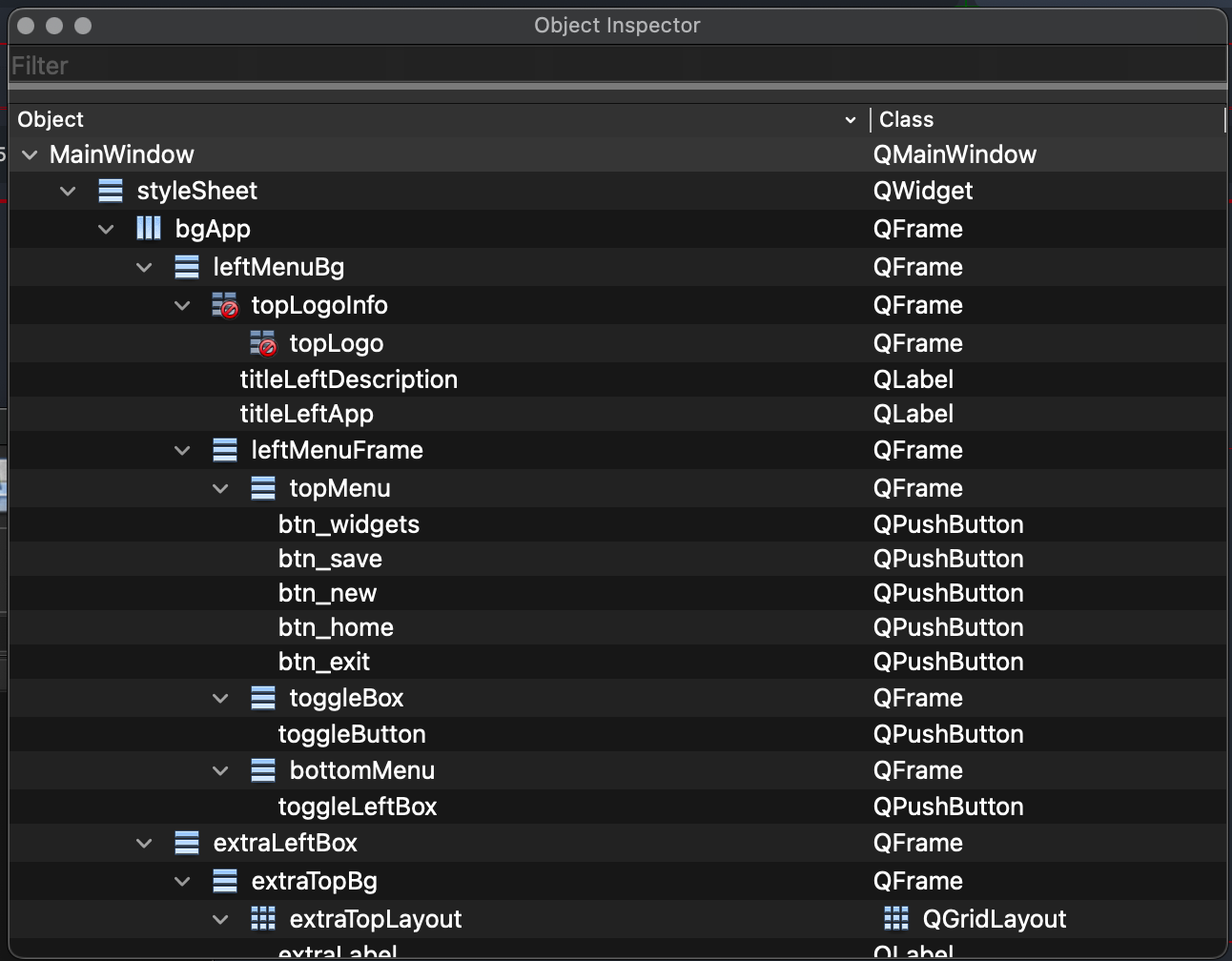


图4-4 布局图

使用Qtdesigner进行图形界面的设计如图4-5所示。

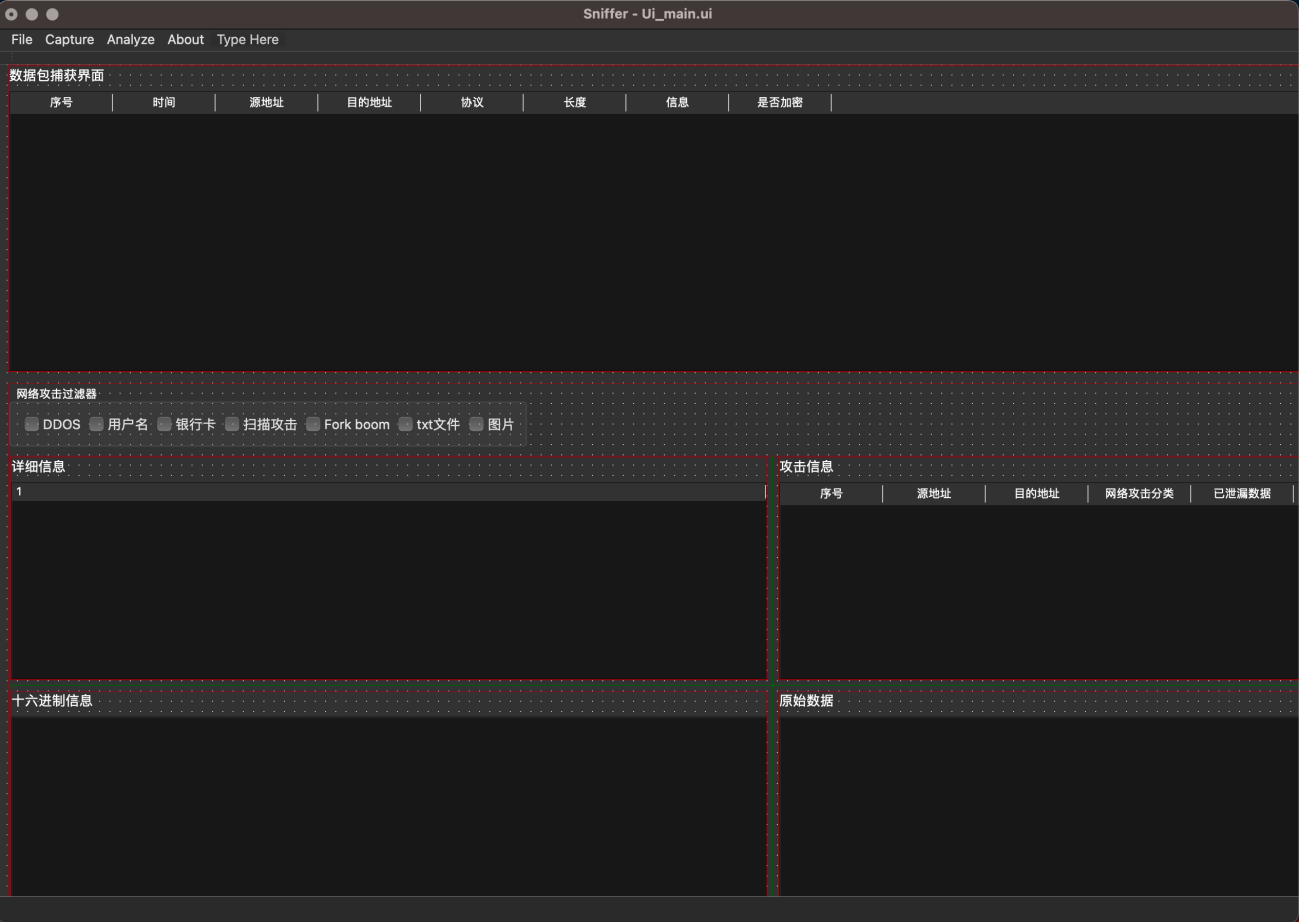


图4-5 图形界面

## 入侵检测系统测试

在文件根目录下打开控制台中，输入启动代码python3 main.py即可运行系统，实际运行界面如图4-6所示。

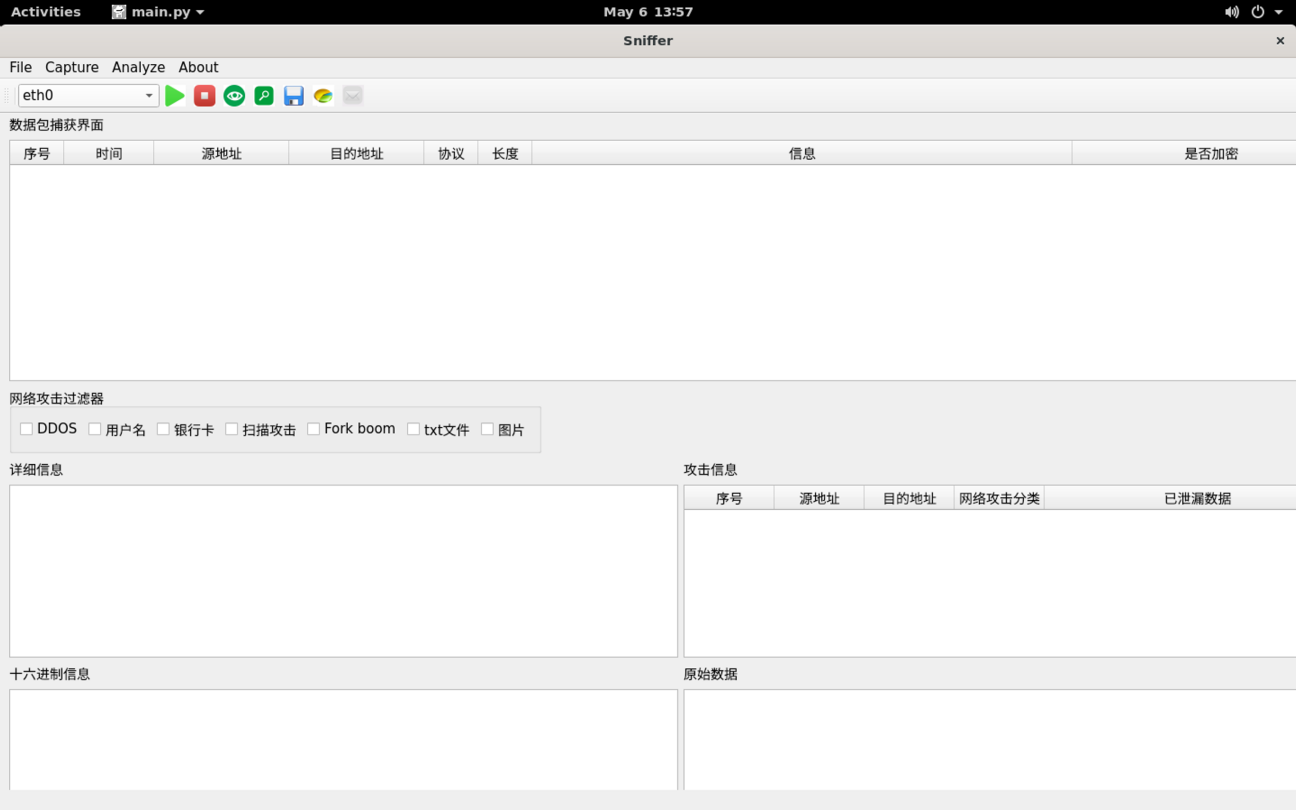


图4-6 系统运行界面

主界面可以让用户选择不同网卡，并按需求通过**数据包**过滤器实现网络数据包的过滤，可以实现对IP地址、端口、网络协议过滤或者可以自定义**数据包**过滤器，如图4-7、图4-8、图4-9所示。

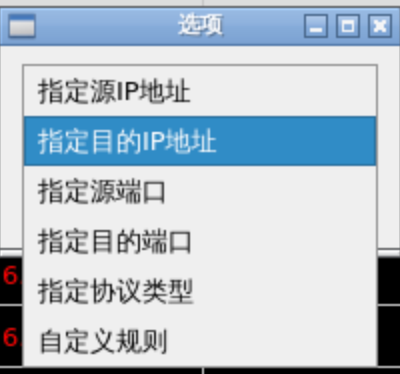


图4-7 IP地址过滤

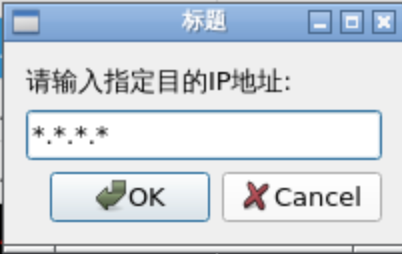


图4-8 指定IP地址

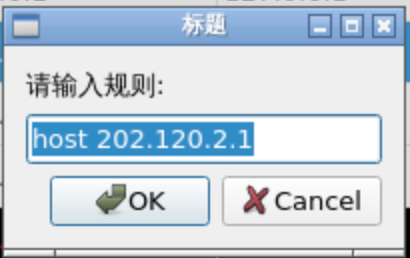


图4-9 指定过滤规则

点击捕获按钮开始捕获数据包，数据包包含了从数据链路层->网络层->传输层等信息，使用eth0网卡捕获到的网络传输中的数据包如图4-10所示。

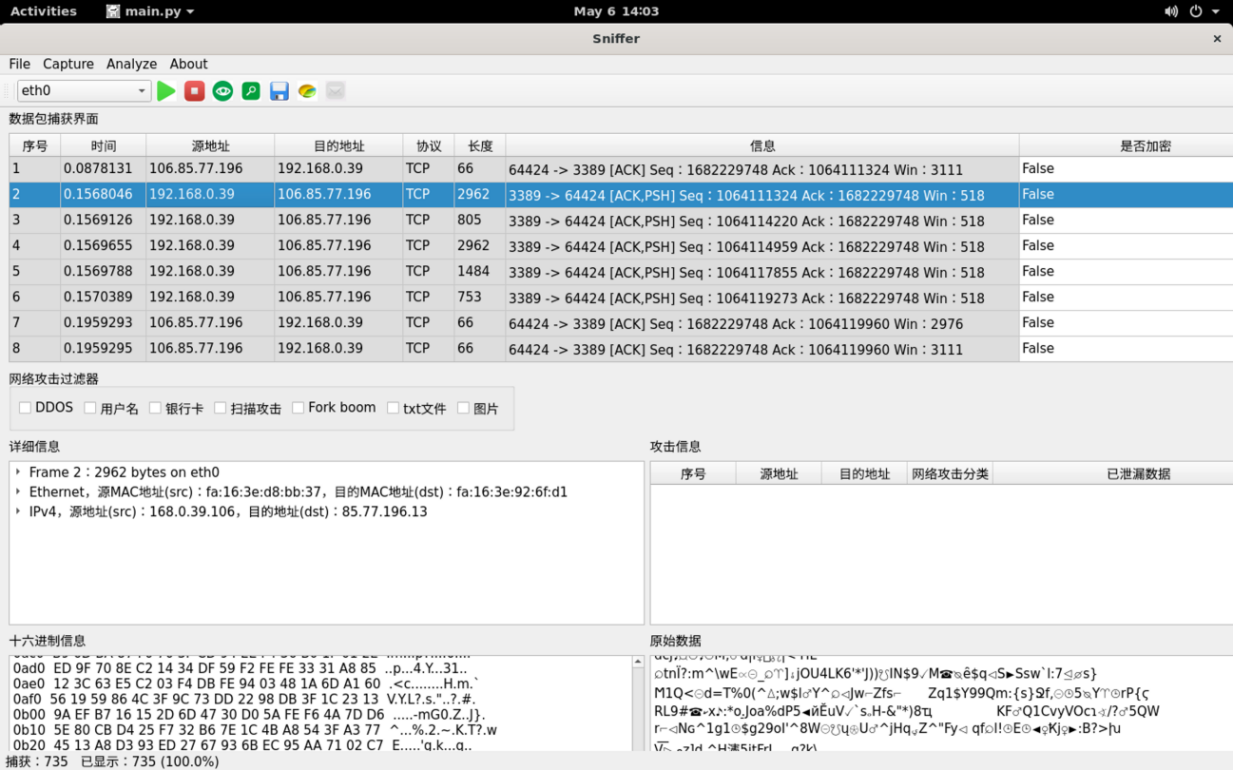


图4-10 捕获的数据包

当启动入侵检测系统时，需要在网络攻击过滤器中选择可以检测入侵行为的类型，如图4-11所示。

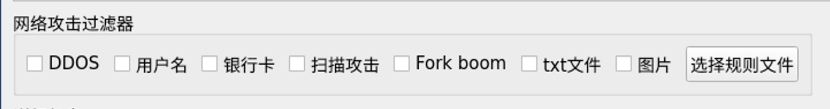


图4-11 网络攻击过滤器

在命令行中输入nmap -A localhost命令，使用nmap工具模拟nmap扫描攻击，如图4-12所示。

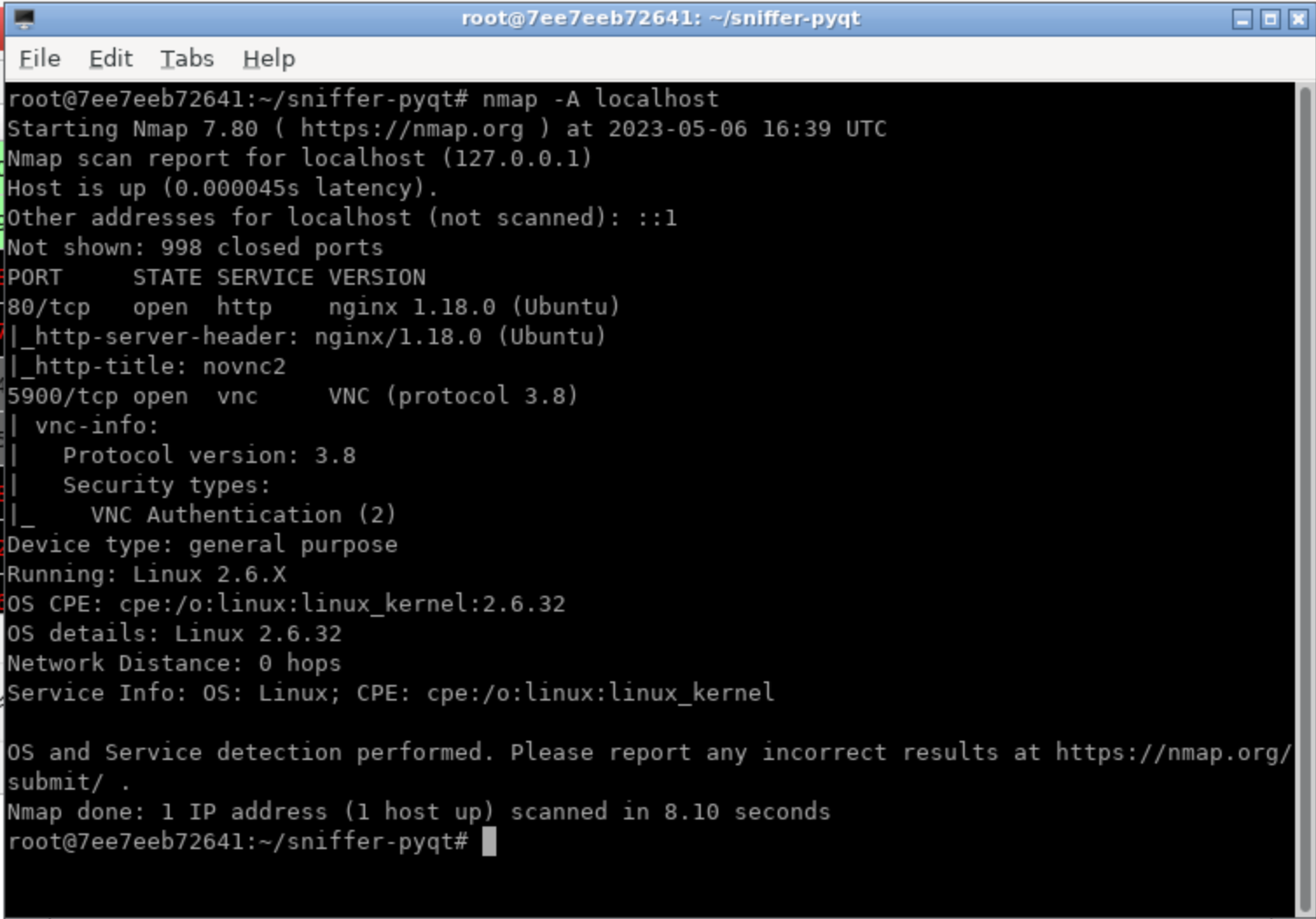


图4-12 nmap模拟扫描攻击

nmap -A参数会发送大量特征数据包，可以看到系统成功检测出扫描攻击，并显示攻击分类，如图4-13所示。



图4-13 检测到扫描攻击

检测到扫描攻击，发送扫描攻击的预警，提醒管理员采取相应措施，如图4-14所示。

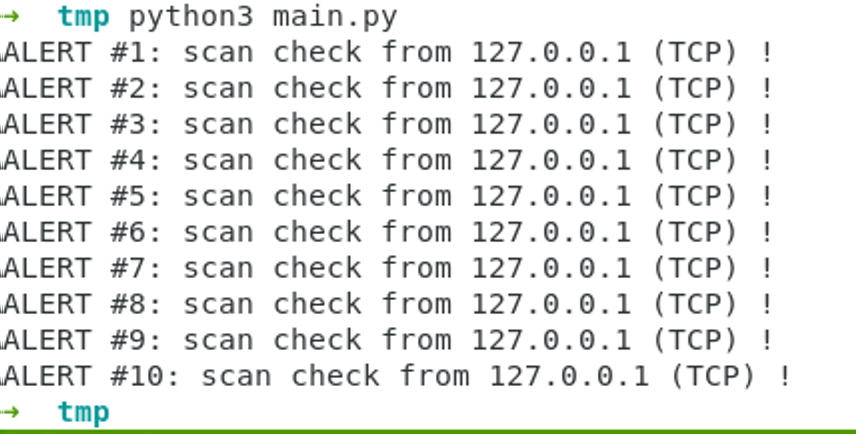


图4-14 扫描攻击预警

使用nc命令进行明文传输模拟，入侵检测系统服务端使用nc -lp 3333开始监听3333端口等待攻击客户端连接，如图4-15所示。攻击客户端使用nc localhost 3333发起连接，其中传输的数据包含fork boom攻击关键字，如图4-16所示。系统成功检测出fork boom攻击，如图4-17所示。

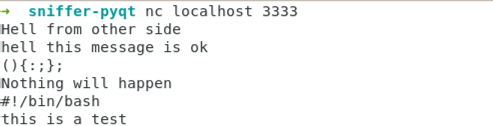


图4-15 收到带攻击命令的消息

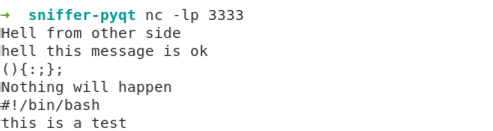


图4-16 发送带攻击命令的消息



图4-17 检测到fork boom攻击

检测到fork boom攻击并预警，提醒管理员采取相应措施，如图4-18所示。

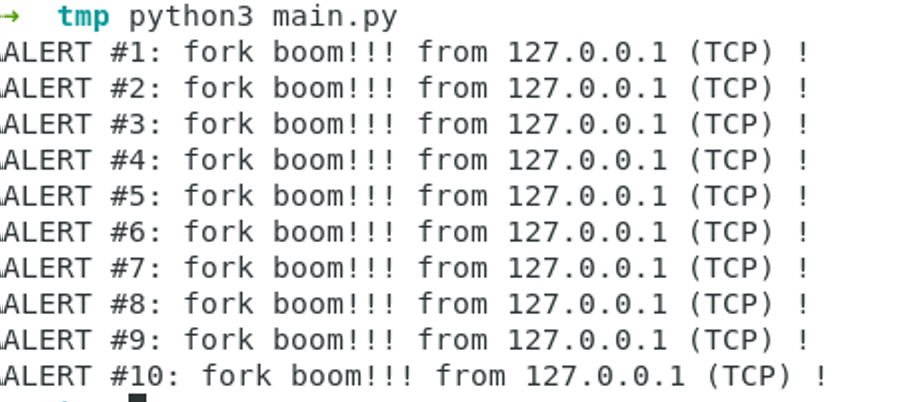


图4-18 fork boom攻击预警

通过点击保存按钮进行数据包的保存，方便保留现场，同时支持使用其他的数据包分析软件进行分析，保存路径如图4-19所示。

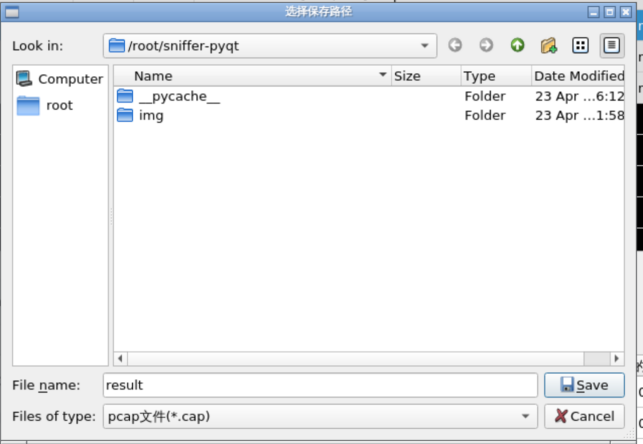


图4-19 保存路径

## 本章小结

本章主要论述了入侵检测系统网络数据包捕获模块、入侵规则模块、入侵检测模块、响应模块、存储模块、界面管理模块的的具体设计与实现细节，并进行攻击模拟。系统测试表明，该系统可以捕获到网络传输中的网络数据包，能对已知的入侵行为做出良好的检测反应，基本满足入侵检测系统要求。

# 总结与展望

防火墙是抵御网络入侵的第一道屏障，而入侵检测系统称得上是抵御网络入侵的第二道屏障，在对目标系统的性能影响较小的前提下，基于网络的入侵检测系统是非常好的选择。本文提出了一种用于Linux操作系统物联网设备、使用PyQt设计的入侵检测系统，该方法具有较高的可行性和实用性。实验结果表明，该入侵检测系统可以将网络数据包进行捕获和分析，并且根据匹配规则发现入侵数据包，也可以及时发现隐私数据泄露，有较好的实用性。

自互联网和物联网诞生以来，安全问题一直是困扰企业乃至普通用户的重大问题，为了应对不断进阶的入侵手段，人们研究了新的检测方法，如数据融合技术，免疫学原理的应用等等。为了应对数量庞大的物联网设备，日后的入侵检测系统研究工作可以在分布式入侵检测系统上，再通过融入先进的机器学习和深度学习技术进行恶意流量的分析和检测，通过结合各种自动化工具进行配合实现更加高效的网络实时监控。

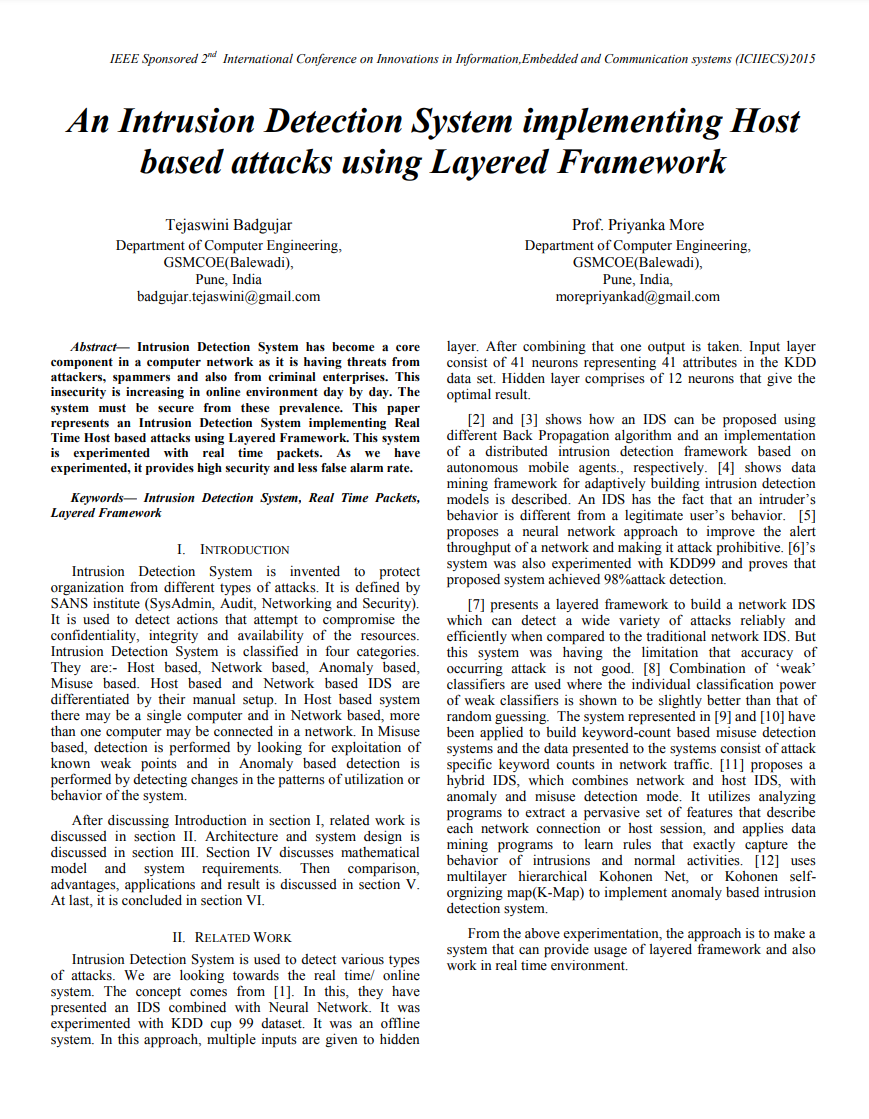
对物联网设备的防护，不能单单依靠入侵检测系统来实现，更需要多种防护方法来保障设备的正常运行。同时，对入侵技术的主动研究或许会更有利于推动防护技术的更新。

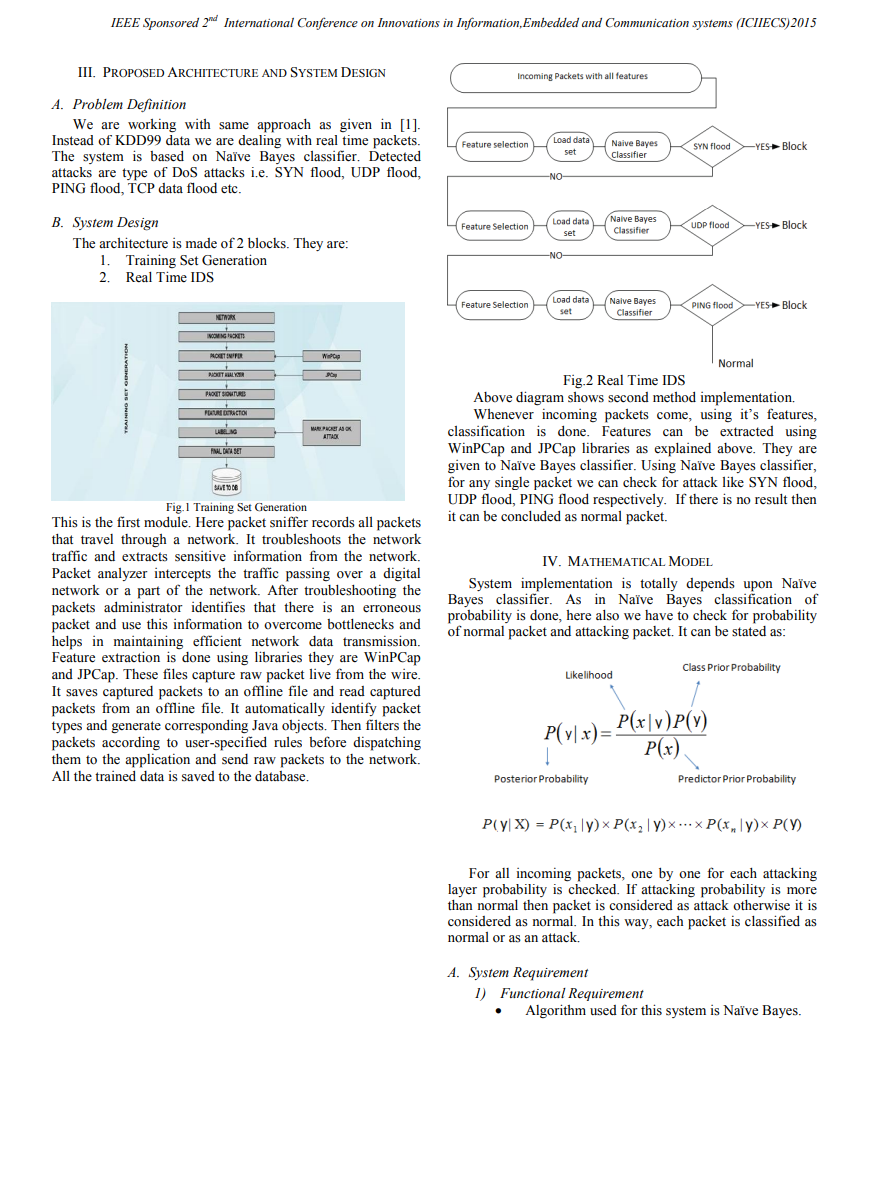
参考文献

1. IDC发布2021下半年全球物联网支出指南[J].传感器世界,2021,27(12):35.
2. 钱劼. 基于蜜罐技术的Mirai僵尸网络检测技术研究[D].北京邮电大学,2019.
3. 杨忠铭.计算机网络信息安全及其防火墙技术应用[J].数字通信世界,2023(01):126-128.
4. 尹锁强.网络数据包中的分析程序设计与开发[J].无线互联科技,2022,19(09):59-61.
5. 刘毅.入侵检测技术发展简述[J].现代计算机(专业版),2012(18):31-33+36.
6. 涂保东.入侵检测系统的发展历史[J].计算机安全,2003(09):18-21.
7. 王贵珍,曲天光.入侵检测系统研究与发展概述[J].保密科学技术,2019(02):30-35.
8. Ying L, Zhang Y, Ou Y J. The Design and Implementation of Host-Based Intrusion Detection System[C]. In: 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jinggangshan, China, 2010: 595-598.
9. T Badgujar, P More. An Intrusion Detection System implementing Host based attacks using Layered Framework[C]. In: 2015 International Conference on Innovations in Information,Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2015: 1-4.
10. P Lotfallahtabrizi, Y Morgan. A novel host intrusion detection system using neural network[C]. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CWCC), Las Vegas, America, NV, 2018: 124-130.
11. 邱晓理.防范入侵全接触[J].计算机安全,2005(07):26-27.
12. 袁连海,李湘文,徐晶.缓冲区溢出攻击研究[J].舰船电子工程,2019,39(04):88-93.
13. 张鹏骞.分布式拒绝服务攻击及防御[J].衡水学院学报,2018,20(03):1-4.
14. 杨慧.基于专家系统的入侵检测技术[J].中国科技信息,2008(15):81-82+84.
15. 郭三田. 基于神经网络模型的入侵检测技术研究[D].广东工业大学,2022.
16. 戴庭. 无线传感器网络广播认证协议及入侵检测研究[D].南京邮电大学,2014.
17. 赵俊忠,游林,徐茂智,等.入侵检测系统中检测技术的研究[J].计算机工程与应用,2005(02):11-13+89.
18. 代威.入侵检测技术在网络安全中的应用[J].重庆理工大学学报(自然科学),2018,32(04):156-160+185.
19. 许大卫. 基于主机入侵检测的先进智能方法研究[D].江南大学,2010.
20. 陈晓安.计算机网络入侵检测系统的研究[J].电子测试,2021(18):76-77+73.
21. 靳志成,王静雅.Linux操作系统加固防护技术的探究[J].信息记录材料,2022,23(05):94-97.
22. 陈光辉.Python在物联网中的应用与发展综述[J].信息通信,2019(04):123-125.

# 附 录

1. 外文参考文献原文





1. 外文参考文献部分译文

使用分层框架实现基于主机的攻击的入侵防御系统

摘 要

入侵预防系统已成为电脑网络的核心组成部分，因为它受到来自攻击者、垃圾邮件发送者和犯罪企业的威胁。这种不安全感在网络环境中日益增加。系统必须保证不受这些流行病的影响。本文介绍了一个利用分层框架实现基于实时主机的攻击的入侵预防系统。该系统采用实时数据包进行实验。实验表明，该方法具有较高的安全性和较低的误报率。

关键词：入侵预防系统 实时数据包 分层框架

Ⅰ. 引言

入侵预防系统是为了保护组织免受不同类型的攻击而发明的。它是由SANS协会（系统管理、审计、网络和安全）定义的。它用于检测试图破坏资源的机密性、完整性和可用性的操作。入侵预防系统分为四类。它们是:-基于主机，基于网络，基于异常，基于滥用。基于主机和基于网络的入侵检测系统的区别在于它们的手动设置。在基于主机的系统中可能有一台计算机，而在基于网络的系统中，可能有多台计算机连接在一个网络中。在基于误用的检测中，检测是通过寻找已知的弱点来完成的，而在基于异常的检测中，检测是通过检测系统利用模式或行为的变化来完成的。

在第一节讨论了导言之后，第二节讨论了相关的工作。第三部分讨论了体系结构和系统设计。第四节讨论数学模型和系统要求。第五部分对比、优势、应用及结果进行了讨论。最后，在第六部分进行了总结。

Ⅱ. 相关工作

入侵预防系统是用来侦测不同类型的攻击。我们期待实时/在线系统。这个概念来自[1]。在此基础上，提出了一种结合神经网络的入侵检测系统。它是实验与KDD CUP 99数据集。那是个离线系统。在这种方法中，对隐藏层给出了多个输入。合并后的一个输出采取。输入层由代表KDD数据集中41个属性的41个神经元组成。隐层由12个神经元组成，给出了最佳结果。

[2]和[3]分别展示了如何使用不同的反向传播算法和基于自主移动代理的分布式入侵检测框架的实现来提出入侵检测系统。描述了自适应构建入侵检测模型的数据挖掘框架。IDS的事实是，入侵者的行为不同于合法用户的行为。[5]提出了一种神经网络方法，以提高网络的警报吞吐量，使其攻击禁止。系统还用KDD99进行了实验，证明该系统可以实现98%的攻击检测。

[7]与传统的网络入侵检测系统相比，本文提出了一种分层的网络入侵检测系统的构建框架。但是该系统具有发生攻击的准确性不好的局限性。[8]在弱分类器的个体分类能力略强于随机猜测的情况下，使用“弱”分类器的组合。在[9]和[10]中所表示的系统已被应用于构建基于关键字计数的误用检测系统，提供给系统的数据包括网络流量中特定于攻击的关键字计数。[11]提出了一种混合入侵检测系统，它结合了网络和主机入侵检测系统，具有异常和误用检测模式。它利用分析程序来提取描述每个网络连接或主机会话的普遍特征集，并应用数据挖掘程序来学习准确捕获入侵行为和正常活动的规则。[12]使用多层分层的 Kohonen网络，或Kohonen自我识别映射（k-Map）来实现基于异常的入侵预防系统。

致 谢

行文至此，思绪万千，二十载求学之路将尽，觉行之愈笃，知之益明。

落其实者思其树，学其成时念吾师。未及弱冠之年，幸得刘容娟等恩师传道受业解惑，更习得温、良、恭、俭、诚等品行，无以为报，揖礼还授。今蒙受师杨锴点拨指导之恩，方能作得此文。经师易遇，人师难遇，微微寸心难报之。

年年遇人，人遇去去，遇遇至散散。每忆同行挚友，轻重自在心头。思年少青春岁月，念故人陈年旧事，畅聊彻夜至天明，共赏人间山与河。俱怀逸兴壮思飞，欲上青天揽明月。而今留此片语只言托尺素，纸短情长聊表相思。愿有岁月来日再聚首，故人相聚也自有方。

幸与良人一春秋，胜得黄金九万两。聘聘袅袅十三余，豆蔻梢头二月初。春风十里扬州路，卷上珠帘总不如。我住长江头，君住长江尾。日日思君不见君，共饮长江水。此水几时休，此恨几时已。只愿君心似我心，定不负相思意。

生我者，父母也。养我者，亦如是。我挚爱者，我至亲也。每念及父之肝疾，母之辛劳，胸中块垒难平，涕泪如雨，感怀交加，夜不能寐。求学之路，家中艰苦之多，如世间沙石，困扰其左右，然致余安于学业，不闻他事。望学有所成，以宽父母心。

而今天涯去，故园仍在桃李灼灼春风里，余自当博学而日参省乎己，力求知明而行无过，以报恩师家长情。

文毕，且祝诸君平安喜乐，万事顺意。