基于机器学习的网络流量分析审计系统
需求分析文档

郑宇森 **王鑫** **赵鸿宇**

韩志鹏 **姜来** **喻路稀**

1 引言

随着互联网的快速发展和信息交换的日益增加，企业、组织和个人面临着越来越复杂的网络安全威胁。网络攻击的形式和手段也变得更加隐蔽和高级，传统的安全措施已经不再足够应对这些威胁。此外，随着云计算和大数据技术的不断发展，网络流量数据量的增加和复杂性的提高也给网络流量分析带来了新的挑战。

传统的网络安全解决方案通常依赖于特征和规则的匹配，但这种方法往往无法及时应对新型的网络攻击。因此，基于机器学习的网络流量分析审计系统应运而生。

机器学习技术可以通过对大量网络流量数据的学习和分析，从中发现潜在的威胁和异常行为。通过训练算法，该系统可以识别出正常的网络流量模式，并检测出与正常行为不符的异常流量。这种方法可以更加准确地识别出潜在的网络攻击，提高网络安全性能，并及时采取相应的防御措施。

本需求文档旨在定义一种基于机器学习的网络流量分析审计系统，以满足市场对于更高效、更准确、更全面的网络安全解决方案的需求。该系统将结合人工智能的先进算法和网络安全领域的专业知识，实现对网络流量的实时嗅探、数据存储、流量分析以及分析结果的呈现，从而帮助用户对网络安全状况的深入了解和及时响应。

在市场需求和现行系统分析的基础上，我们将探讨该系统的技术可行性，并明确功能需求，包括网络流量嗅探、数据存储、流量分析和分析结果呈现。此外，我们还将介绍数据建模的重要性，并提供测试和维护需求以确保系统的可靠性和稳定性。最后，我们将概述系统的开发计划，以实现系统的高效开发和部署。

2 市场需求与现行系统分析

2.1 市场需求分析

网络安全已经成为全球范围内企业、政府和个人所关注的重要议题。随着网络攻击手段的不断演进，市场对于更高效、更准确的网络安全解决方案的需求也在不断增长。在流量分析与监控领域，市场的主要需求包括以下方面：

- **流量监控与威胁检测**：用户需要能够监控网络流量，并快速检测出异常行为和潜在的网络攻击威胁，以便及时采取措施应对。
- **自动化分析与决策支持**：用户需要一个自动化的系统，能够对大量的网络流量数据进行分析和处理，并提供有效的决策支持，帮助用户更好地应对网络安全挑战。
- **分析结果可视化呈现**：用户需要系统能够将复杂的网络流量数据以直观、易于理解的方式进行可视化呈现，帮助用户深入了解网络安全状况和发现潜在的风险。

2.2 现行系统分析

目前，在网络流量分析审计领域已经存在一些现行系统，它们尝试满足市场对于网络安全的需求。然而，这些系统存在一些局限性和挑战：

- **手动规则配置**：许多现行系统需要手动配置规则和模式来识别网络攻击行为，这对于快速变化和复杂的网络攻击形式来说可能不够灵活和高效。
- **有限的机器学习应用**：一些系统尝试使用机器学习算法进行网络流量分析，但在实际应用中，其应用范围有限，缺乏对多样性和复杂性网络攻击的全面覆盖。
- **数据处理和存储能力**：随着网络流量的不断增长，现行系统在数据处理和存储能力方面可能面临挑战，无法满足大规模数据的高效处理和长期存储需求。
- **用户体验和交互性**：一些系统在用户界面设计和交互性方面存在不足，缺乏直观性和易用性，使得用户难以快速获取所需信息和进行操作。

基于上述市场需求和现行系统的局限性，我们计划开发一种基于机器学习的网络流量分析审计系统，可以弥补现有系统的不足，并提供更高效、更准确、更全面的网络安全解决方案。

3 技术可行性分析

针对系统中的各个功能模块，进行技术可行性分析如下：

3.1 网络流量嗅探

使用基于 Qt 平台和 C++ 语言编写的网络嗅探器软件，底层使用 libpcap 库捕获原始网络数据包。这种选择提供了成熟的网络数据包捕获和处理能力。libpcap 库广泛应用于网络分析工具中，能够满足实时捕获和解析网络流量的需求。

技术可行性：高。libpcap 库稳定且功能强大，已经经过广泛应用和验证。

3.2 数据存储

系统使用开源对象存储库 MinIO 作为数据库，用于存放嗅探器抓取到的原始流量文件、预处理生成的特征文件以及模型输出的标签文件。MinIO 提供了可靠的分布式对象存储解决方案，并具备良好的可伸缩性和可用性。

技术可行性：高。MinIO 是一个成熟的对象存储库，具备高性能、高可靠性和良好的可扩展性。

3.3 流量分析

系统中的流量分析部分包括两个关键模块：

3.3.1 特征提取器

该模块由 C++ 语言编写，负责从抓取的 pcap 文件中提取与网络流量相关的特征。特征提取器使用自定义算法和规则，能够快速高效地提取出特征，并进行处理和编码。

技术可行性：高。自定义的特征提取器能够满足系统对网络流量特征的需求，并具备快速高效的处理能力。

3.3.2 NTML (Network Traffic Machine Learning) 模型

该模块基于 PyTorch 的 nn 库，使用 KDD Cup99 数据集训练网络流量分析神经网络模型。通过该模型，预期实现对 20 余种不同的恶意流量的分析，并对网络流量进行分类和标记。

技术可行性：高。PyTorch 提供了强大的深度学习框架，能够支持神经网络模型的训练和推断。KDD Cup99 数据集是公认的网络流量分析数据集，为模型的训练提供了基础。

3.4 分析结果呈现

系统计划使用基于 streamlit 框架的网页模块来呈现分析结果，提供直观的可视化和动态交互。这种基于网页的呈现方式使用户能够方便地访问和查看分析结果，并进行交互操作。

技术可行性：高。基于网页的可视化呈现已经成为现代应用程序中常用的方式之一，相关工具与技术非常成熟。

3.5 总结

综合来看，系统中各个功能模块所采用的技术方案具备较高的可行性。网络嗅探、数据存储、流量分析和分析结果呈现的关键组件都选择了成熟的技术，并具备所需的功能和性能。这将为系统的开发和实施提供坚实的技术基础，确保系统能够有效地满足市场需求并提供可靠的基于机器学习的网络流量分析审计功能。

4 功能需求

在基于机器学习的网络流量分析审计系统中，我们需要实现以下功能：

4.1 网络流量嗅探功能

- 捕获和记录网络流量数据包。
- 解析和提取数据包中的关键信息，如源地址、目标地址、端口号、协议等。
- 支持数据包筛选，以便用户根据指定条件对已捕获的数据包进行选择性查看。
- 支持 GUI 和 CLI 两种模式，用户可以根据部署环境选择合适的模式进行流量嗅探。
- 支持将捕获的网络流量导出为 pcap 文件格式，以便后续存储和分析。

4.2 数据存储功能

- 将嗅探器抓取到的原始流量文件进行存储，并确保访问的稳定性。
- 存储预处理生成的特征文件和模型输出的标签文件。
- 提供数据库管理接口，支持文件上传、下载和统一审计的操作。

4.3 流量分析功能

- 使用 KDD99 特征提取器模块从原始流量文件中提取与网络流量相关的特征。
- 对提取的特征进行处理和编码，以便输入到 NTML 模型进行流量分析。
- 使用 NTML 模型进行流量分类和标记，识别出不同类型的恶意流量。
- 提供可定制的规则和策略，以支持针对特定威胁或流量行为的检测和分析。

4.4 分析结果呈现功能

- 使用 Dash Board 网页模块呈现系统的分析结果。
- 提供流量分析的时序图，展示流量的变化趋势和重要事件。
- 显示流量属性表，列出各个特征的统计数据和相关指标。
- 提供 IP 拓扑关联图谱，展示不同 IP 地址之间的关联关系。
- 统计并显示端口和协议的计数情况，帮助用户了解流量的分布和特征。
- 标记和分类流量，以便用户可以快速识别和区分特定类型或特定标签的流量。

以上是系统中的主要功能需求，通过这些功能，用户可以实现对网络流量的全面审计和分析，识别潜在的网络攻击和异常行为，提高网络安全性和监控能力。

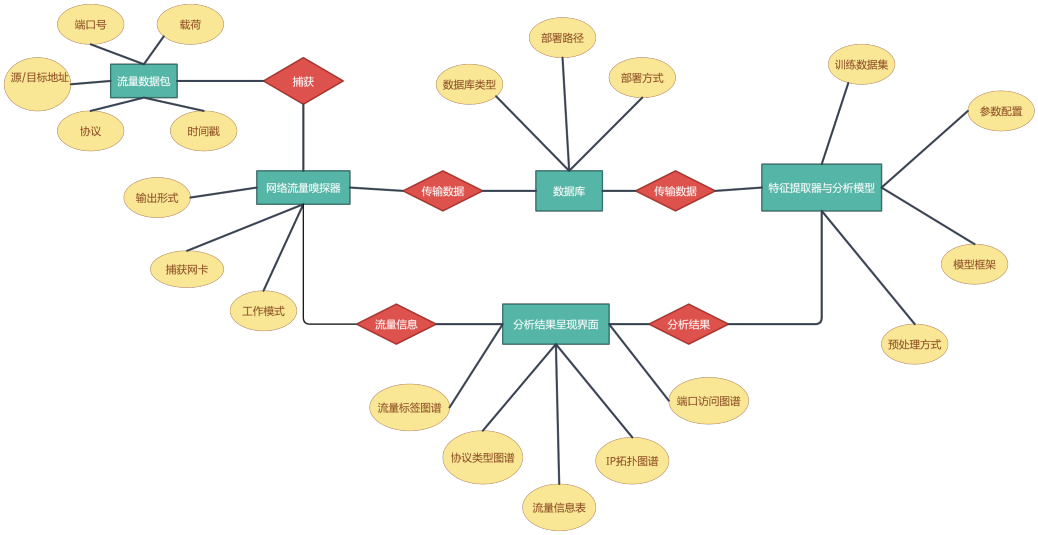

Figure 1: 系统 ER 图

5 数据建模

在该网络流量分析系统的需求分析中，数据建模是关键的一部分，它有助于开发者理解和组织系统中的结构和关系。在进行数据建模时，ER 图是一种常用的工具，用于描述系统中的实体（Entity）和实体之间的关系（Relationship），本系统的数据建模 ER 图如图 1所示。

ER 图 1描述了每个实体的名称和属性，以及实体之间的联系。数据建模的目的是为了确保系统中的结构和关系能够满足系统需求，并提供一个清晰的数据模型来指导系统的开发和实现。

6 测试和维护需求

6.1 测试需求

在开发网络流量分析审计系统时，需要进行全面的测试以确保系统的功能和性能符合预期。测试需求包括以下方面：

- **单元测试**：对系统中的各个模块进行单元测试，验证其功能的正确性和稳定性。
- **集成测试**：将各个模块整合在一起进行测试，确保它们之间的协同工作正常。
- **性能测试**：对系统进行负载和压力测试，评估系统在高负载情况下的性能表现和稳定性。
- **用户验收测试**：与目标用户合作进行测试，确保系统满足他们的需求并易于使用。

6.2 维护需求

一旦网络流量分析审计系统投入使用，需要进行持续的维护以确保其正常运行和性能优化。以下是维护需求的一些重点：

- **故障排除和修复**：及时响应和解决系统中的故障、错误和异常情况，确保系统的稳定性和可靠性。
- **安全更新和修补程序**：监测和修复系统中的安全漏洞，及时更新和升级系统以保护系统免受新的威胁。
- **数据库维护**：定期备份和优化 MinIO 数据库，确保数据的完整性和性能。
- **性能监控和优化**：监测系统的性能指标，识别性能瓶颈并进行优化，以提升系统的响应速度和吞吐量。

7 开发计划

为了有效地完成网络流量分析审计系统的开发，以下是开发计划的概述：

7.1 需求分析和设计（3 天）

- **收集用户需求**：进行调研，了解当前市场需求和期望，明确系统的功能和性能要求。
- **系统设计**：基于用户需求，进行系统架构设计、数据库设计和界面设计，明确各个模块的功能和交互关系。

7.2 系统开发（5 天）

- **网络嗅探器开发**：使用 C++ 和 Qt 平台，开发网络嗅探器软件，实现数据包捕获、解析和过滤功能，并支持导出为 pcap 文件。
- **MinIO 数据库集成**：集成开源对象存储库 MinIO，用于存储嗅探器抓取的原始流量文件、预处理的提取器文件和模型输出的标签文件。
- **KDD99 特征提取器开发**：使用 C++ 编程语言，开发特征提取器，从捕获的 pcap 文件中提取与网络流量相关的特征，并进行处理和编码。
- **NTML 模型开发**：使用 PyTorch 的 nn 库，基于 KDD Cup99 数据集，训练网络流量分析的机器学习模型，以识别恶意流量。
- **Dash Board 网页开发**：使用前端和后端技术，开发网页界面，用于呈现系统的分析结果，提供直观的可视化和交互功能。

7.3 测试和部署（4 天）

- **单元测试和集成测试**：对各个模块进行单元测试和集成测试，确保系统的功能和模块之间的协同工作正常。
- **性能测试**：进行系统的负载和压力测试，评估系统在高负载情况下的性能和稳定性。
- **故障排除和修复**：解决测试过程中发现的故障、错误和异常情况，确保系统的稳定性和可靠性。
- **部署和打包**：将系统部署到目标环境中，配置必要的网络设置和权限，确保系统正常运行并提供服务。

7.4 任务分工

- **郑宇森**：项目整体组织，网络嗅探器开发，网页开发，单元测试，设计文档，测试文档
- **王鑫**：网络嗅探器开发，网页开发，单元测试，集成测试，设计文档，测试文档，部署文档
- **赵鸿宇**：NTML 模型构建与训练，单元测试，设计文档，测试文档
- **韩志鹏**：KDD99 特征提取器开发，单元测试，设计文档，测试文档
- **姜来**：数据库开发，单元测试，Docker 封装，需求文档，设计文档，测试文档，部署文档
- **喻路稀**：数据库开发，单元测试，设计文档，测试文档，PPT 制作