软件定义网络中的基于深度强化学习的攻击检测

Introduction

在现代网络环境中，网络拓扑和流量的动态变化，以及不断增加的网络攻击数量，对现有的防御系统带来了巨大挑战。入侵检测是通过监控网络或计算机系统活动来识别和检测潜在的入侵行为或恶意活动。入侵检测方式可以分为两类：基于特征的入侵检测和基于异常的入侵检测。基于特征的入侵检测（Signature-based Detection）和基于异常的入侵检测（Anomaly-based Detection）都可以利用机器学习算法来训练模型以区分正常和异常活动，从而实现入侵检测的高性能。传统机器学习方法能够从输入数据中提取有用的特征，例如网络流量统计信息、协议字段、传输特征等。这些方法如决策树、朴素贝叶斯、支持向量机等在已知攻击类型检测中具有良好性能和效果。然而，传统机器学习方法对于特征选择和提取高度依赖领域知识和人工设计，需要专业人员进行特征工程，耗费大量时间和精力。此外，传统机器学习方法构建的模型是基于静态的训练数据，在动态的网络环境中，入侵行为和攻击方式不断演变和变化，包括新攻击的出现和新服务的利用，网络行为可能随时间发生显著变化。这使得传统的机器学习模型很快过时，需要频繁且昂贵的模型更新。因此，提出具有自适应性的入侵检测模型成为网络安全的重要任务。

软件定义网络 (SDN) 是一种创新的网络模型，它将控制从数据平面中分离出来。具体来说，SDN控制器掌握全局网络视图并执行转发决策，而SDN交换机仅在数据平面中进行流量转发。此外，控制平面和数据平面之间的通信通过 OpenFlow 协议进行管理，该协议允许 SDN 交换机使用流表执行流量转发任务，其中 SDN控制器 可以动态添加/删除/更新流规则。因此，利用SDN架构实现了网络安全等众多复杂的网络控制和管理任务。然而，基于SDN的网络中存在一些安全漏洞，并且由于暴露的网络层出现新的攻击面而导致异常问题的处理具有挑战性。

本文中，我们提出了一种新的基于强化学习的入侵检测系统，在数据平面中利用P4语言实现强化学习算法对流量进行特征提取，然后将最优的特征向量发送至控制平面进行攻击类型检测，检测结果通过控制器下发流表的方式发送至数据平面，更改交换机中恶意流量的转发规则，对后续的恶意流量在数据平面实现线速处理。

本文的其余部分安排如下。第二部分介绍了用于入侵检测的机器和强化学习技术。第三节回顾相关工作。第四节描述了我们提出的用于入侵检测的强化学习模型，而第五节对其进行了评估。第六节提出了本文的最后结论。

Preliminary

1. SDN Dataplane



SDN网络架构图

1. Reinforcement Learning

强化学习（Reinforcement Learning）是一种机器学习方法，通过智能体（Agent）与环境的交互学习如何在某个特定环境中做出最优的决策。在强化学习中，智能体通过尝试不同的行动（动作），观察环境的反馈（奖励或惩罚），并根据这些反馈来调整自己的行为策略，从而逐步学习并改进其决策能力。

强化学习的核心概念包括以下要素：

1. 环境（Environment）：智能体所处的外部环境，可以是真实世界中的物理环境，也可以是模拟环境或虚拟环境。

2. 状态（State）：描述环境的某个特定时刻的情况或特征，用于确定智能体在该状态下应该采取的行动。

3. 动作（Action）：智能体在特定状态下可以执行的操作或决策。

4. 奖励（Reward）：环境根据智能体的动作提供的反馈信号，用于评估智能体的行为质量。奖励可以是正值（鼓励）或负值（惩罚）。

5. 策略（Policy）：智能体根据当前状态选择动作的策略，可以是确定性策略（根据状态选择一个确定的动作）或随机策略（根据状态选择动作的概率分布）。

6. 值函数（Value Function）：衡量智能体在某个状态或状态-动作对上的长期回报期望值，用于评估策略的好坏。

7. 强化学习算法：用于根据智能体与环境的交互数据进行学习和决策优化的算法，如Q-learning、Deep Q-Networks（DQN）、Proximal Policy Optimization（PPO）等。

1. P4

P4（Programming Protocol-Independent Packet Processors）是一种用于编写可编程网络数据包处理器的领域专用语言。它的作用是提供一种灵活的方式来定义网络数据包的处理行为，使网络设备能够根据特定协议或应用需求对数据包进行定制化的处理。

Related work

【1】中，提出了使用深度 Q 学习 (AE-DQN) 算法的对抗/多代理强化学习来实现针对基于异常的 NIDS 。利用DQN解决5 标签分类问题，在特定的攻击类型（DoS和Probe）中表现出很强的性能，当样本数据很少时，该算法表现不佳。

【2】利用P4语言在数据平面（支持 P4 的物联网网关）对数据包头进行解析，判断其是否为恶意流量然后选择将其正常转发或发到控制器进行分类识别，控制器将根据分类器的结果来定义一组特定的数据包字节子串作为标头字段，网关将使用这些标头字段来安装流规则。然后，匹配的数据包将直接由网关处理，而不需要转发到SDN控制器或其他远程防火墙功能，从而实现数据包线速处理。这种方式利用可编程网络的特点直接在数据平面进行恶意流量的初步识别，但对于恶意流量，数据平面需要将其全部发送至控制平面，将带来转发时延的问题，而且对于新的攻击类型，现有的流规则无法及时将其正确处理，可能会带来一定的误差，给网络安全带了很大的风险。

【3】将基于Q-learning的强化学习与深度前馈神经网络方法相结合进行网络入侵检测。文中提出的深度Q学习（DQL）模型为网络环境提供了持续的自动学习能力，可以使用自动试错方法检测不同类型的网络入侵，并不断增强其检测能力。

Architecture design

RL设计

因为强化学习的逻辑是被训练的主题从初始状态开始，每进行一次动作就有一次状态的变更，而这个状态可能是当前最优的，最后得出一个整体最优的结果，所以可以类比从一组数中选出和最大的子数组这个算法一样，从原始特征中选出最优的子特征组。

每次状态对应一个特征组，最后选出10个特征（暂定）

FeatureList1

Action1

FeatureList2

Action2

FeatureList3

Action3

FeatureList4

Action4

FeatureList5

Action5

FeatureList6

Action6

FeatureList7

Action7

FeatureList8

Action8

FeatureList9

Action9

FeatureList10

**一共79次动作，10个状态**

**Q表设计**

**动作空间：79个动作，每次都有79种选择**

**Q大小：10\*79（10个状态，79个动作）**

**Reward设计？**

所提出的框架包括三层：学习平面、控制平面和数据平面。在学习平面中，我们用样本数据训练强化学习模型，对每个数据包有三种动作：捕获、丢弃、转发，每种动作对应一个状态，将其与样本中的标签作对比，

在数据平面中使用强化学习对数据包进行模糊决策，对于不明类型的数据包将其捕获并发送至控制平面，交由入侵检测分类器进行准确分类，然后下发流表至数据平面中，并将分类结果反馈至RL代理，对已知的攻击类型选择直接丢弃，对已知的良性流量选择将其根据MA表进行转发。

****

**系统架构图**

1. Learning

Q-learning是一种基于值迭代的强化学习算法，它用于解决马尔可夫决策过程（Markov Decision Process，MDP）问题。在Q-learning中，我们使用一个称为Q表的数据结构来表示每个状态和动作对应的Q值，这些Q值用于指导智能体在环境中做出决策。

Q表是一个二维表格，行表示状态，列表示动作，每个表格单元格(i, j)表示在状态i下采取动作j所对应的Q值。初始时，Q表可以是随机初始化的，或者初始化为某个固定值，具体取决于问题的性质和问题域。

Q表设计：

1. 确定状态空间（State Space）：首先，你需要明确定义智能体可能会遇到的所有状态。状态可以是离散的，也可以是连续的。如果状态空间较大，可以考虑使用函数逼近方法来近似Q值，而不是使用Q表。

2. 确定动作空间（Action Space）：定义智能体可能采取的所有动作。与状态空间类似，动作可以是离散的或连续的。

3. 初始化Q表：创建一个二维数组，行数等于状态空间大小，列数等于动作空间大小。将Q表中的所有元素初始化为一个较小的随机值或者一个固定的初始值。

4. 定义奖励函数（Reward Function）：奖励函数用于给智能体在执行动作后提供即时反馈。通过奖励函数，我们可以衡量每一步的好坏，并帮助智能体逐渐学会最优策略。奖励函数可以根据问题的需求来定义。

5. 更新Q值：在Q-learning算法的训练过程中，智能体会根据当前的状态和奖励来更新Q表的值。更新Q值的公式如下：

Q(s, a) = Q(s, a) + α \* (r + γ \* max(Q(s', a')) - Q(s, a))

其中，Q(s, a)表示在状态s下采取动作a所对应的Q值，α是学习率（通常是一个小于1的常数），r是智能体在执行动作a后获得的奖励，γ是折扣因子（通常是0到1之间的值），s'表示智能体执行动作a后转移到的新状态，a'表示在新状态s'下智能体可能采取的动作。

6. 探索与利用（Exploration and Exploitation）：在Q-learning中，智能体需要在探索（exploration）和利用（exploitation）之间进行权衡。探索是指尝试新的动作以发现更多有价值的信息，而利用是指选择已知效果较好的动作。为了平衡这两者，通常会使用ε-贪婪策略，即以概率ε随机选择一个动作，而以概率(1-ε)选择当前Q值最大的动作。

7. 训练过程：在训练过程中，智能体通过与环境的交互不断更新Q表，直到Q值收敛或达到预设的训练轮次。

需要注意的是，对于具有大型状态空间的问题，Q表的大小可能会非常庞大，导致算法的计算和存储成本过高。在这种情况下，可以考虑使用函数逼近方法，如深度强化学习中的神经网络来近似Q值，以减少存储空间和计算复杂度。

Qlearning是一种强化学习算法，采取了一种非政策方法，旨在根据其当前状态确定最佳操作，可以通过制定自己的一套规则或偏离规定的政策来实现这一点。由于 Q 学习可能偏离给定的策略，因此不需要定义的策略。非策略方法是使用 Q 值（也称为行动值）实现的。Q 值是操作的预期未来值，存储在 Q 表中。

Q Learning模型在迭代过程中运行，该过程涉及多个组件协同工作以帮助训练模型。迭代过程涉及代理通过探索环境并在探索继续时更新模型来学习。本文中Q-Learning的多个组成部分设计如下：

- 代理。代理是在环境中执行和操作的实体。本文中指数据平面中交换机的P4程序

- 状态。 状态是标识代理环境中当前位置的变量。有两种状态，识别成功or失败

- 行动。该操作是代理处于特定状态时的操作。本设计中包括三个动作，捕获、丢弃和转发

- 奖励。强化学习中的基本概念是为代理的行为提供积极或消极响应的概念。当前数据包的攻击类型识别成功则加分，否则减分

- 情节。一集是指代理无法再采取新操作并最终终止。

- Q 值。Q 值是用于测量特定状态下的操作的指标。

以下是确定 Q 值的两种方法：

a.时间差异。时间差分公式通过将当前状态和动作的差值与先前状态和动作进行比较，合并当前状态和动作的值来计算 Q 值。

B.贝尔曼方程。数学家理查德·贝尔曼（Richard Bellman）在1957年发明了这个方程，作为最优决策的递归公式。在qLearning环境中，贝尔曼方程用于帮助计算给定状态的值并评估其相对位置。具有最高值的状态被视为最佳状态。

Q Learning模型通过试错体验来学习任务的最佳行为。其过程涉及通过学习最佳操作值函数或 q 函数来建模最佳行为。此函数表示状态 s 中操作 a 的最佳长期值，随后在每个后续状态中遵循最佳行为。

Q(s, a) = Q(s, a) + \alpha \* (r + \gamma \* \max\_{a'} Q(s', a') - Q(s, a))

Experiment

1. Dataset

目前用于异常检测的数据集比较常见的有KDD Cup 99、NSL-KDD、CICIDS2017和CICIDS2019等，但大多数数据集已经过时且不可靠。其中一些数据集缺乏流量多样性和流量，一些数据集没有涵盖各种已知的攻击，而另一些数据集对数据包有效负载数据进行了匿名处理，无法反映当前的趋势。有些还缺乏功能集和元数据。在本文中，我们选取CICIDS2017作为实验数据集。CICIDS2017数据集包含良性且最新的常见攻击，类似于真实的现实世界数据（PCAP）。它还包括使用 CICFlowMeter 进行网络流量分析的结果，以及基于时间戳、源和目标 IP、源和目标端口、协议和攻击（CSV 文件）的标记流。还可以使用提取的特征定义。 该数据集基于 HTTP、HTTPS、FTP、SSH 和电子邮件协议构建了 25 个用户的抽象行为。实施的攻击包括暴力 FTP、暴力 SSH、DoS、Heartbleed、Web 攻击、渗透、僵尸网络和 DDoS。

1. Q-learning学习过程

Q-learning学习过程是一种交互式方法，代理通过探索环境并根据收到的奖励更新 Q 表来学习，涉及的步骤包括：

- Q 表初始化。第一步是创建 Q 表作为跟踪每个状态下的每个操作和相关进度的位置。每个数据包作为一个输入，初始状态都为良性。

- 观察。代理需要观察环境的当前状态。

- 行动。代理选择在环境中执行操作。操作完成后，模型将观察操作是否对环境有益。

- 更新。执行操作后，就可以使用结果更新 Q 表了。

- 重复。重复步骤 2-4，直到模型达到所需目标的终止状态。



Qlearning算法步骤

References

【1】Suwannalai E, Polprasert C. Network intrusion detection systems using adversarial reinforcement learning with deep Q-network[C]//2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE). IEEE, 2020: 1-7.

【2】 Qin Q, Poularakis K, Tassiulas L. A learning approach with programmable data plane towards IoT security[C]//2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2020: 410-420.

【3】 Alavizadeh H, Alavizadeh H, Jang-Jaccard J. Deep Q-learning based reinforcement learning approach for network intrusion detection[J]. Computers, 2022, 11(3): 41.