

基于深度强化学习的恶意ELF文件检测对抗方法

孙贺¹, 张博成², 耿嘉炫³, 吴迪³, 王俊峰³, 方智阳²

(1. 试验物理与计算数学重点实验室, 北京 100094;

2. 四川大学网络空间安全学院, 四川 成都 610207; 3. 四川大学计算机学院, 四川 成都 610065)

摘要: 近年来, 基于深度学习的恶意可执行与可链接格式 (executable and linkable format, ELF) 文件检测研究取得了显著进展。同时, 关于模型对抗性攻击的研究也得到了广泛关注, 攻击者可以通过生成对抗样本误导神经网络, 使恶意软件被错误归类为良性软件, 从而逃逸检测。尽管已有多种生成对抗样本的方法被提出, 但它们通常不适合对ELF文件进行修改, 或缺乏在不同检测模型之间迁移的能力。为克服现有方法的局限性, 提出了一种基于深度强化学习的恶意ELF文件检测对抗方法, 该方法通过构造目标检测模型的最优扰动字节序列, 能够在保留ELF文件原有功能的前提下生成对抗样本, 而无须依赖目标模型的内部实现细节。实验结果表明, 该方法生成的对抗样本针对目标检测模型的逃逸成功率达到76.80%, 并能通过对抗训练提升目标检测模型的鲁棒性。

关键词: 深度强化学习; 恶意软件检测; 对抗样本; ELF文件

中图分类号: TP309.5

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2024074

Adversarial method for malicious ELF file detection based on deep reinforcement learning

SUN He¹, ZHANG Bochong², GENG Jiaxuan³, WU Di³, WANG Junfeng³, FANG Zhiyang²

1. Key Laboratory of Experimental Physics and Computational Mathematics, Beijing 100094, China

2. School of Cyber Science and Engineering, Sichuan University, Chengdu 610207, China

3. College of Computer Science, Sichuan University, Chengdu 610065, China

Abstract: In recent years, research on detecting malicious executable and linkable format (ELF) files based on deep learning had made significant progress. At the same time, adversarial attacks on models had also gained widespread attention. Attackers could generate adversarial examples to mislead neural networks, causing malicious software to be misclassified as benign, thereby evading detection. Although various methods for generating adversarial examples had been proposed, they were often not suitable for modifying ELF files or lacked the ability to transfer

收稿日期: 2024-04-30; 修回日期: 2024-07-15

通信作者: 方智阳, fangzhiyang@scu.edu.cn

基金项目: 国家自然科学基金 (U2133208); 四川省科技厅重点研发项目 (2023YFG0290)

Foundation Items: The National Natural Science Foundation of China (U2133208), The Key Research and Development Program of Science and Technology Department of Sichuan Province (2023YFG0290)

引用格式: 孙贺, 张博成, 耿嘉炫, 等. 基于深度强化学习的恶意ELF文件检测对抗方法[J]. 网络与信息安全学报, 2024, 10(5): 152-162.

Citation Format: SUN H, ZHANG B C, GENG J X, et al. Adversarial method for malicious ELF file detection based on deep reinforcement learning[J]. Chinese Journal of Network and Information Security, 2024, 10(5): 152-162.

across different detection models. To overcome the limitations of existing methods, an adversarial example generation method based on deep reinforcement learning was proposed. This method generated adversarial examples by constructing optimal perturbation bytes for the target detection model while preserving the original functionality of the ELF files, without relying on the internal details of the target model. The experimental results showed that the adversarial examples generated by this method achieved a 76.80% success rate in evading the target detection model, and could enhance the robustness of the model through adversarial training.

Keywords: deep reinforcement learning, malware detection, adversarial samples, ELF file

0 引言

云计算和物联网技术的迅猛发展, 为人们的日常生活和工作带来了极大的便利, 然而, Linux 系统安全也因此面临着前所未有的挑战。作为 Linux 系统的默认可执行文件, ELF (executable and linkable format) 文件已成为网络攻击者常用的攻击载体。恶意 ELF 文件主要分为漏洞利用工具 (exploits)、病毒 (virus) 和后门 (backdoor) 3 种类型, 攻击者通过执行这些恶意文件完成权限提升和信息窃取等非法目的^[1]。

传统的基于签名的恶意软件检测方法根据已知恶意软件样本的特定模式来创建一个特征库, 以此识别潜在的恶意软件。尽管这种方法易于实现且效率高, 但它的泛化能力较弱, 需要持续手动更新恶意特征库, 并且对新型恶意软件的识别效果较差。

近年来, 深度学习已被广泛应用于图像分类^[2]、语音识别^[3]、自然语言处理^[4]等领域, 鉴于其强大的特征处理与识别能力, 研究人员开始将其应用于各种恶意文件的检测任务^[5-7]。在端到端深度学习恶意软件检测模型^[8]中, 神经网络能够自动提取原始二进制字节的特征, 无须执行耗时的特征工程便可取得良好的分类效果。

尽管深度学习模型在恶意软件检测领域有着出色的表现, 但最近的研究表明^[9-10], 它在逃逸攻击面前是十分脆弱的。攻击者可以通过精心设计的扰动, 生成对抗样本来误导深度学习模型, 使其将恶意软件错误地分类为良性软件, 从而规避检测。这种对抗性攻击严重威胁着基于深度学习的恶意软件检测系统的安全性和可靠性。

目前, 已有多种生成对抗样本的方法被提出, 如向固定位置插入扰动字节^[11]、使用优化算法结合多种扰动插入方法^[12]等。但这些方法在应用于 ELF 文件的对抗样本生成中存在一些局限

性, 如依赖于目标模型的具体实现细节、难以在保留原有功能的前提下修改 ELF 文件等。

为了克服现有方法的局限性, 本文提出了一种基于深度强化学习的恶意 ELF 文件检测对抗方法。该方法通过深度强化学习构造目标检测模型的最优扰动字节序列, 在保留 ELF 文件原有功能的前提下, 能够生成有效误导检测模型的对抗样本。

本文的主要贡献如下。

1) 提出了一种基于深度强化学习的对抗样本生成方法, 并基于该方法实现了对抗样本生成模型, 该模型能在不损害原始恶意 ELF 文件功能的前提下, 高效地生成目标检测模型的对抗样本。

2) 优化了扰动生成过程, 通过智能学习算法自动确定最优的扰动字节序列, 而不依赖于人工预定义的特征或扰动模式, 从而提高了对抗样本的生成效率。

3) 设计了针对原始检测器性能、攻击效果、文件功能保留及模型鲁棒性提升等方面的多组实验, 实验结果证明本文提出方法的有效性和实用性。

2 背景知识和相关工作

本节主要介绍相关的知识与工作, 包括 ELF 文件格式、恶意软件检测模型、深度强化学习方法和对抗样本生成方法。

2.1 ELF 文件格式

ELF 文件格式是一种广泛使用的标准文件格式, 用于定义程序、库 (如共享库)、核心转储和其他在类 Unix 操作系统中执行的文件类型。ELF 格式由 UNIX 系统实验室 (USL) 作为应用程序二进制接口 (ABI) 的一部分开发, 并被许多不同的操作系统采用, 包括 Linux、Solaris、IRIX 和 FreeBSD 等。

在未被实际加载到内存中执行时, ELF 文件结构如图 1 所示, 主要分为 4 个部分。

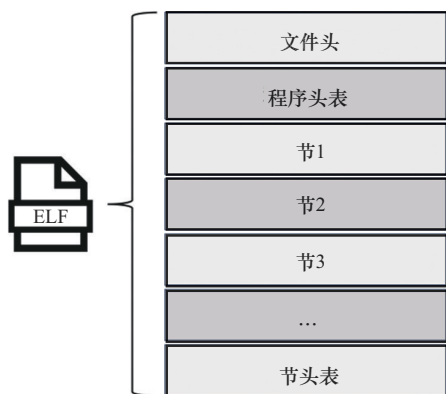


图1 ELF 文件结构

Figure 1 ELF File Structure

1) 文件头: 描述了整个文件的基本属性, 如 ELF 文件的类型、目标机器类型 (如 x86 或 ARM)、入口点地址 (程序开始执行的位置) 等。

2) 程序头表: 指定了创建进程映像所需的段或其他信息。程序头表告诉系统如何创建进程映像, 包含了各个段的大小、位置、对齐等信息。

3) 节: ELF 文件中的数据被组织为多个节, 每个节包含了不同类型的数据。例如, `.text` 节通常包含程序的执行代码, `.data` 节包含初始化的全局变量和静态变量, 而 `.bss` 节包含未初始化的数据。

4) 节头表: 描述了文件中的各个节, 每个节包含不同类型的数据, 如程序代码、数据、符号表、重定位信息等。节头表中的每个条目指定了节的名称、大小、位置等属性。

因 ELF 文件格式具有灵活性、可扩展性和支

持不同字节序格式的特点, 其成为 Unix 和类 Unix 操作系统中二进制文件的事实标准。

2.2 恶意软件检测模型

在数字安全领域, 面对日益增长的恶意软件的威胁, 研究人员已经开发出多种基于深度学习的检测模型以提高检测效率和准确性。端到端的恶意软件检测模型和基于图像的文件分类模型因其高效准确的特点和跨平台的能力成为比较热门的研究方向。因此, 本文在这两个方向上各选了一个具有代表性的模型作为实验的基线检测器。

(1) MalConv

MalConv^[13]是一种基于卷积神经网络的恶意软件检测模型, 其模型结构如图2所示。模型以填充或截断为 d byte 的二进制文件作为输入, 经过嵌入层后每个字节被映射为一个 8D 向量作为卷积层的 8 个输入通道。将 8 个通道分为两个 4D 通道后分别输入两个卷积层 (其中一个带有 Sigmoid 激活函数), 再将结果经过门控法组合后得到 128 通道的特征图。特征图进行最大池化后, 经过带有 ReLU 激活函数的全连接层, 得到该二进制文件为恶意样本的概率。

(2) ResNet18

残差神经网络^[2] (residual neural network, ResNet) 是一种使用了跨层数据线路的深度卷积神经网络。来自上一层的数据可以不经卷积层 (或只经过 1×1 卷积层) 直接通过残差块, 使输入更快地向前传播, 因此残差网络更容易拟合出恒等映射, 极大地消除了“退化”现象。

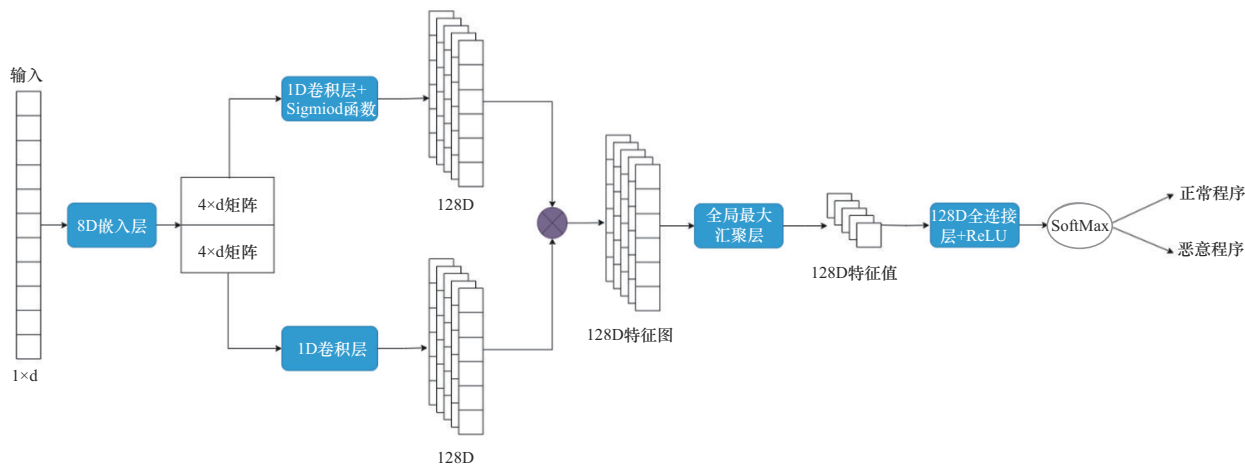


图2 Malconv 模型结构

Figure 2 MalConv model structure

ResNet18检测流程如图3所示, 首先将二进制文件通过一定方式转化为灰度图像, 经过卷积层、批量规范化层、ReLU以及最大汇聚层初步提取特征后, 输入连续的3个残差块中, 每个残差块中将卷积结果与输入相加, 以此降低学习恒等映射的难度。最后, 深度卷积结果经过全局平均汇聚以及全连接层后, 通过SoftMax函数得到文件为良性及恶性的概率。

2.3 深度强化学习方法

深度强化学习是一种将深度学习与强化学习相结合的机器学习方法。在强化学习中, 智能体通过与环境交互, 学习如何采取行动以最大化长期累积奖励。传统的强化学习在处理高维度感知输入时面临困难, 而深度学习恰好擅长处理这类数据。深度强化学习利用深度神经网络来逼近强化学习中的关键函数, 如价值函数或策略函数, 使智能体能够直接从原始高维度数据中学习到最优策略, 无须人工提取特征。

深度Q网络(deep Q-network, DQN)作为深度强化学习的代表性算法之一, 由DeepMind在2013年^[14]和2015年^[15]的两篇论文中提出。DQN使用卷积神经网络来逼近Q函数, 即在给

定状态下采取特定动作的期望长期回报。DQN的创新之处在于引入了两个关键技术: 经验回放和目标网络。经验回放通过存储智能体的历史交互数据, 从中随机抽取样本进行训练, 从而打破了数据间的相关性, 提高了训练效率和稳定性。目标网络是一个与Q网络结构相同但参数更新频率较低的网络, 用于生成训练标签, 减少了因目标值不稳定而导致的训练振荡问题。

2.4 对抗样本生成方法

对抗样本的概念由Szegedy等^[16]首次提出, 是一类在机器学习模型, 特别是在深度学习模型中, 用来欺骗模型的输入样本。这些样本通过在原始输入数据中添加经过精心设计的微小扰动来生成, 这种扰动虽然对人类的感知影响很小, 但却能导致机器学习模型做出错误的预测或分类。

快速梯度符号方法^[17](fast gradient sign method, FGSM)是一种经典的对抗样本生成方法, 它利用了输入数据对损失函数的梯度, 通过添加一个小的、符号与梯度相同的扰动来生成对抗样本。这种方法旨在以最小的改动最大化模型的输出误差, 从而导致模型做出错误的预测或分类。

在FGSM的基础上, Al-Dujaili等^[18]提出了

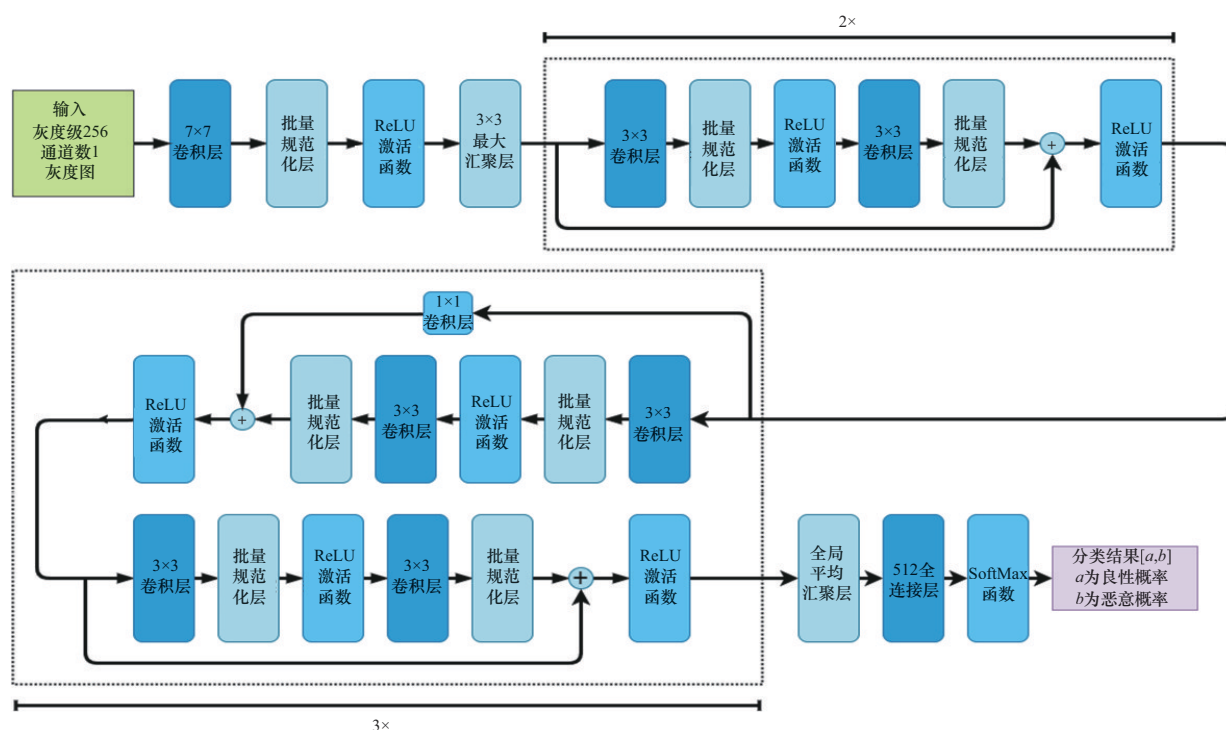


图3 ResNet18检测流程

Figure 3 ResNet18 detection process

针对二进制领域中恶意软件的对抗样本生成方法。具体的方法是通过在使损失函数最大化的方向上移动或扰动输入样本以生成对抗样本,但这种方法依赖于激活函数的单调性,在实际使用中有很大的局限性。

Kolosnjaji 等^[19]沿用了基于梯度的攻击思想,首先计算出模型输出的损失函数关于输入样本的梯度,再基于梯度信息,选择性地修改那些对模型输出影响最大的字节,这通常仅涉及恶意软件样本末尾几个字节的修改,能够尽可能地避免对恶意软件功能的影响。

为了在不损坏文件原有功能的同时获取足够的扰动空间,Kreuk 等^[11]提出了一种单独构造扰动序列再插入原文件的方法。具体来说,该方法利用梯度信息生成了一段具有足够对抗能力的扰动字节,然后在 PE 文件的可松弛区域或末端空间插入这段字节。

除了利用梯度信息生成扰动字节外,Ding 等^[20]利用特征采样技术,从良性文件中提取目标检测模型特征层计算出的权重较高的部分组成一个扰动集合,生成对抗样本时从中选择效果较好的进行注入。

Demetrio 等^[12]在多种保留文件功能的字节注入操作的基础上,提出了一种基于优化的对抗样本生成框架。该框架以目标检测模型的恶意评分为优化目标,使用遗传算法学习扰动注入的过程,最后实现自动化生成对抗样本的目标。这种方法不依赖于模型的详细结构信息,有很强的通用性。

尽管针对 PE 文件的对抗样本生成方法已经非常成熟,Kosikowski 等^[21]指出,由于 ELF 文件的结构复杂性和各个部分之间的依赖性,针对 PE 文件的插入和修改方法并不能完全适用于 ELF 文件。该研究测试了 5 种修改方法在 ELF 文件上的逃逸攻击效果,最终发现填充改变、末尾附录和动态扩展 3 种操作是实际有效的。

本文结合了良性文件采样和优化框架构建两种方法的优势,使用一个深度强化学习框架来构建目标检测模型的扰动字节序列,并采用“末尾附录”操作进行扰动填充,在目标检测模型上取得了良好的逃逸攻击效果。

3 对抗样本生成

本节主要介绍对抗样本生成的方法和具体过程,包括深度强化学习模型架构和内部实现细节。

3.1 模型架构

本文的对抗样本生成方法以智能体的训练为核心,通过智能体自动地构造目标检测模型的扰动字节序列,并将其填充到原始恶意 ELF 文件的末尾空间中,对抗样本生成系统架构如图 4 所示。

3.2 环境

作为深度强化学习框架的一个非常重要的部分,环境定义了智能体的观测空间,并计算给定动作实施后的奖励。本文方法的环境主要包含 4 个部分:原始恶意文件、生成的扰动序列、特征提取器和目标检测模型。观测空间被定义为与目标

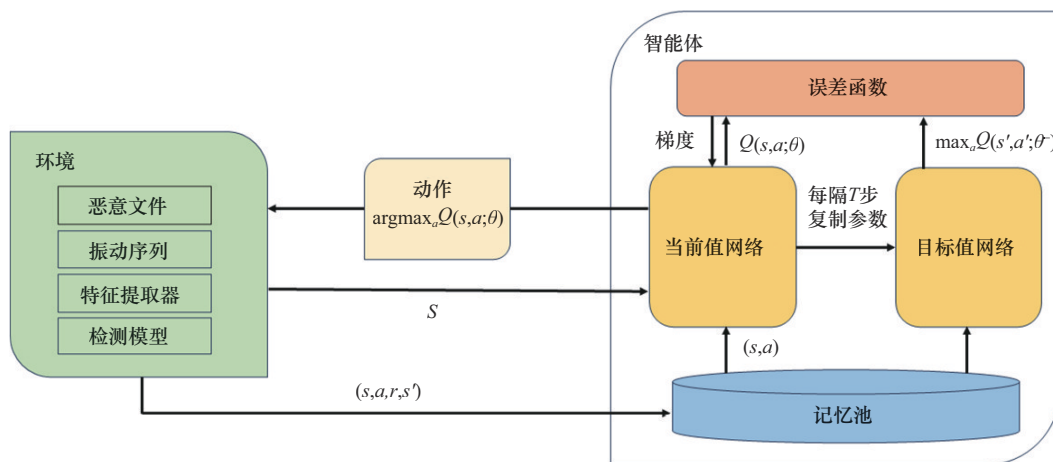


图 4 对抗样本生成系统架构

Figure 4 Adversarial example generation system architecture

检测模型输入空间大小相同的一维数组，数据类型为8位无符号整数，正好对应文件的一个字节。

在交互过程中，环境首先接收由智能体给定动作生成的新的扰动字节序列，然后在确保原始恶意文件功能完整的前提下将扰动字节序列填充至尾部的可扩展空间以形成临时的对抗样本。目标检测模型会对该样本进行检测并给出恶意评分，由此评分可计算出本轮获得的奖励，之后再由特征提取器将样本转换为与观测空间相同的数据类型，最后即可将所有相关数据送回智能体完成本轮的训练。

3.3 动作空间

动作空间定义了智能体可以选择执行的所有可能动作的集合，因此，其设计是系统构建中的一个至关重要的环节。过大的动作空间会导致学习过程复杂化和训练时间延长，增加计算负担和资源消耗；过小的动作空间可能限制智能体的决策能力，阻碍达到最优策略并降低模型的整体效能。

本文方法首先使用目标检测模型对所有数据集中的样本进行检测评分，然后选择了恶意评分最低的20个样本组成一个良性样本集以供后续操作使用。

完整动作空间包括以下4种操作。

- 1) 序列重构：从良性样本集中抽取一个样本，随机选择该样本的一个合法区间，用其覆盖整个扰动字节序列。
- 2) 序列插入：从良性样本集中抽取一个样本，随机选择该样本的一个合法区间，将其插入扰动字节序列的随机位置。
- 3) 序列删除：随机选择扰动字节序列的一个合法区间进行删除。
- 4) 序列倍长：将当前扰动字节序列复制一份拼接在其末尾。

3.4 奖励

在DQN及其他强化学习方法中，奖励是驱动学习过程的关键因素。奖励机制定义了一个环境如何根据智能体的行为给予反馈，在每个时间步，智能体根据当前的策略选择行动，执行行动后，环境提供下一个状态和一个奖励，DQN使

用这个奖励来更新Q值，更新公式为

$$Q_{\text{new}}(s,a) = Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \quad (1)$$

其中， $Q(s,a)$ 表示在状态 s 下采取动作 a 所获得的预期回报； α 是学习率，用于控制学习过程中新信息的影响程度； γ 是折扣因子，用于控制未来回报对当前动作选择的影响； r 是当前动作采取后从环境中获得的即时奖励，在本系统中，如果当前对抗样本成功逃逸目标检测模型的检测，则获得一个较大的固定值作为奖励，否则获得当前恶意评分与先前恶意评分之差作为奖励。

3.5 智能体

智能体是DQN算法的核心部分，它负责与环境交互，做出决策，并不断学习以提高性能。如图4所示，DQN的智能体包括一个记忆池和两个神经网络。记忆池用于存储智能体的历史经验，以（状态，动作，奖励，新状态）的四元组形式存储。通过随机抽取这些历史经验进行训练，DQN可以打破数据间的相关性，并提高样本利用的效率。当前值网络和目标值网络是两个结构完全相同但参数不同的网络，当前值网络的参数 θ 根据损失函数实时更新，目标网络参数在每 T （ T 为常量）轮迭代后从当前值网络中复制得到。第 i 轮迭代的损失函数更新如下。

$$L(\theta_i) = E_{s,a,r,s' \sim U(D)} [(r + \gamma \max_{a'} Q(s',a'; \theta_i^-) - Q(s,a; \theta_i))^2] \quad (2)$$

其中， $(s,a,r,s') \sim U(D)$ 为记忆池中抽取的经验样本。

3.6 完整过程

对抗样本生成完整过程如图5所示，对抗样本生成系统首先从数据集中抽取一个恶意ELF文件作为原始二进制样本，其状态表示为（原始二进制序列+初始扰动序列），该状态送往智能体后，如果处于推理模式，智能体直接通过目标值网络输出该状态下的最优动作，并使用相关组件修改扰动序列以形成对抗样本；如果处于训练模式，则通过当前值网络输出可能的最优动作，在生成新的对抗样本后，基于目标分类器的评分获取对应的奖励，并将其与新的状态一同送回当前值网络进行训练，在经过一定轮次的迭代后将当前值网络的参数复制到目标值网络中。

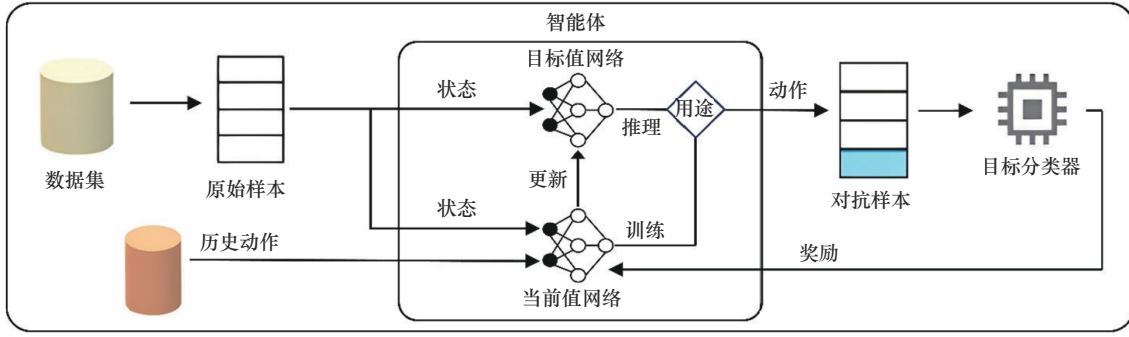


图5 对抗样本生成完整过程

Figure 5 Complete process of adversarial sample generation

算法1描述了DQN模型训练的具体过程。

算法1 DQN模型训练

- 1) 初始化容量为 N 的经验回放存储器 D
- 2) 初始化当前值网络 Q 和目标值网络 Q' , 并随机初始化 Q 网络的参数 θ
- 3) 将目标网络 Q' 的参数 θ' 设为 Q 网络的参数 θ
- 4) 初始化包含多个恶意ELF文件的训练集 $P = \{p_1, p_2, \dots, p_n\}$
- 5) 初始化目标分类器 C
- 6) **for** episode = 1 to M **do**
- 7) 从 P 中随机抽取一个恶意ELF文件 p_1
- 8) 初始化二进制序列 $x_1 = \text{binary}(p_1)$
- 9) 初始化扰动序列 $g_1 = \text{Null}$
- 10) 初始化状态 $s_1 = \{x_1 + g_1\}$
- 11) 初始化恶意评分 $\text{score}_1 = C(x_1)$
- 12) **for** $t = 1$ to T **do**
- 13) 以 ϵ 概率选择随机动作 a_t
- 14) 否则选择 $a_t = \arg\max_a Q(s_t, a; \theta)$
- 15) 执行动作 a_t 使 $g_{t+1} = a_t(g_t)$
- 16) $\text{score}_{t+1} = C(x_1 + g_{t+1})$
- 17) **if** $\text{score}_{t+1} < \text{threshold}$ **then**
- 18) $r_t = R, s_{t+1} = \text{END}$
- 19) **else**
- 20) $r_t = \text{score}_t - \text{score}_{t+1}$
- 21) $s_{t+1} = \{x_1 + g_{t+1}\}$
- 22) **end if**
- 23) 将经验 (s_t, a_t, r_t, s_{t+1}) 存入 D
- 24) 从 D 中随机采样一小批经验 (s_j, a_j, r_j, s_{j+1})
- 25) 对于每一个采样的经验

(s_j, a_j, r_j, s_{j+1})

26) **if** $s_{j+1} = \text{END}$ **then**

27) $y_j = r_j$

28) **else**

29) $y_j = r_j + \gamma \max_{a'} Q'(s_{j+1}, a'; \theta')$

30) **end if**

31) 计算损失函数

$L(\theta) = (y_j - Q(s_j, a_j; \theta))^2$

32) 使用梯度下降法最小化损失函数

$L(\theta)$, 更新 Q 网络的参数 θ

33) 每 S 步将目标网络 Q' 的参数 θ' 设为 Q 网络的参数 θ

34) **end for**

35) **end for**

3.7 优势分析

与传统的优化方法相比, 基于深度强化学习的对抗样本生成方法不依赖于目标模型的内部细节, 而是基于观察到的输入和输出构造对抗扰动。这不仅增强了方法的普适性和在不同检测模型之间的迁移能力, 还使其能够动态适应检测模型的变化, 因此更适用于频繁更新迭代的真实环境。此外, 该方法利用深度强化学习构造针对目标检测模型的最优扰动字节, 从而最大化地利用ELF文件中固定的可填充区域, 无须人工查找特定ELF文件中的可修改或填充位置, 实现了自动化添加扰动的同时确保原始ELF文件的功能不受影响。

4 实验与分析

本节首先介绍了实验使用的数据集、评估指标和计算环境, 然后通过实验证明了本文提出方法的可行性和实用性。

4.1 数据集

本文中使用的数据集包括 11 240 个恶意 ELF 文件和 11 240 个良性 ELF 文件，其中恶意 ELF 文件主要来自 VirusShare 的公开样本集，良性 ELF 文件主要来自 Github Labeled-Elfs 项目。图 6 为数据集的样本分布。

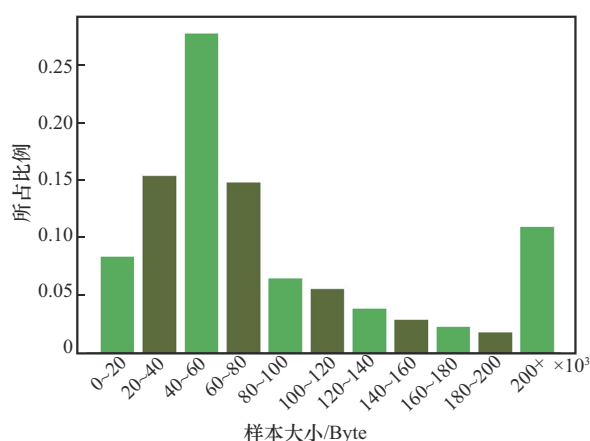


图6 数据集的样本分布

Figure 6 Sample distribution of the dataset

该数据集中的恶意 ELF 文件均源自 Unix 和各种类 Unix 系统，包含了多种常见的恶意文件类型，恶意文件类型统计如表 1 所示。

表1 恶意文件类型统计

Table 1 Malicious File Type Statistics

恶意文件类型	数量/个
木马	5 260
恶意软件	3 009
下载器	873
工具	258
其他	1 840

整个数据集被划分为两个部分，用于不同的实验。在进行基线检测模型的性能测试时，实验首先从总数据集中随机选取了 10 000 个恶意 ELF 文件和 10 000 个良性 ELF 文件。这些文件随后按照 8:1:1 的比例分配到训练集、验证集和测试集中。最后，这些文件被分别标记为良性或恶意，并保存在 CSV 文件中，以形成一个完整的数据集。

在逃逸攻击实验中，剩余的 1 240 个恶意 ELF 文件按照 8:2 的比例分为训练集和测试集。

在这个过程中，模型直接利用这些原始的二进制文件进行训练。

在对抗训练实验中，将逃逸攻击实验生成的对抗样本加入训练集中，并重新进行上述实验。

4.2 评估指标

(1) 模型性能评估

本文使用以下指标对检测模型的性能进行评估。

假阳性 (false positive, FP): 模型预测结果为恶意文件，但实际为良性文件的样本数量。

假阴性 (false negative, FN): 模型预测结果为良性文件，但实际为恶意文件的样本数量。

真阳性 (true positive, TP): 模型预测结果为恶意文件，且实际为恶意文件的样本数量。

真阴性 (true negative, TN): 模型预测结果为良性文件，且实际为良性文件的样本数量。

准确率 (accuracy, ACC): 模型预测结果正确的样本占总样本的比例。

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

精确率 (precision, P): 模型预测结果为恶意文件的样本中实际恶意文件的占比。

$$P = \frac{TP}{TP + FP} \quad (4)$$

召回率 (recall, R): 模型正确识别出的恶意文件占有所有实际恶意文件的比例。

$$R = \frac{TP}{TP + FN} \quad (5)$$

F1 分数: 精确率和召回率的调和平均，能够更加均衡地反映出模型的检测性能。

$$F1 = \frac{2 \times P \times R}{P + R} \quad (6)$$

曲线下面积 (area under curve, AUC): 接收者操作特征曲线 (ROC Curve) 下的面积，该曲线展示了在不同阈值下模型的真阳性率和假阳性率，用以衡量分类模型在所有分类阈值中的性能。

(2) 逃逸攻击效果评估

在逃逸攻击实验中，主要关注的是生成的对抗样本对目标恶意 ELF 文件检测模型的逃逸成功率。这一比率指的是能够通过强化学习模型生成相应对抗样本的原始恶意样本在整个攻击实验数据集的占比。此外，实验还引入了平均步长作为衡量对抗样本生成过程复杂度的一个指标。

4.3 实验环境

本文的主要实验环境基于 64 位 Windows 11 专业版，硬件环境为 Intel Core i9-13900HX (CPU)，32 GB (RAM)，NVIDIA GeForce RTX 4080 Laptop (GPU)，基于 PyTorch 深度学习框架，使用 Python 完成编程。同时使用一台辅助计算机，硬件环境为 Intel Core i7-1260P (CPU)，16 GB (RAM)，运行 64 位 Ubuntu 22.04 LTS 以验证生成样本的功能完整性。

4.4 实验结果与分析

(1) 基线检测器性能测试

根据图6的数据，超过70%的样本小于1 MB，其余大部分样本则集中分布在1 MB到2 MB之间。因此，本文实验将 Malconv 模型的输入大小设置为 100 kB 和 200 kB，这样既保证了模型的分类性能，又提高了训练效率。ResNet的现有版本采用了自适应全局平均汇聚层，这使得任意规模的输入图像经过深度卷积后都能通过汇聚层汇集特征送入最后的全连接层，无须额外设置输入大小。基线检测器训练结果如表2所示。

表2 基线检测器训练结果

Table 2 Training results of the baseline detector

检测模型	ACC	精确率	召回率	F1	AUC
Malconv_100k	98.6%	97.5%	99.6%	98.6%	98.6%
Malconv_200k	98.7%	97.9%	99.4%	98.7%	98.7%
ResNet18	99.6%	100.0%	99.2%	99.5%	99.6%

总体来说，这些基线检测模型在恶意 ELF 文件检测任务上都具有良好的表现。由于输入数据的截断处理，具有更大输入空间的 Malconv 模型往往能够达到更高的检测准确度，但模型的训练和检测时间也会相应增加。

(2) 基于强化学习的逃逸攻击实验

为了避免过多无效尝试带来较差的训练效果和过长的训练时间，实验对单个恶意 ELF 文件样本的最大修改步数限制为 50 步。

以 Malconv_100k 为例，图7展示了在 15 000 步学习过程中 Malconv 对抗模型探索率的变化趋势。

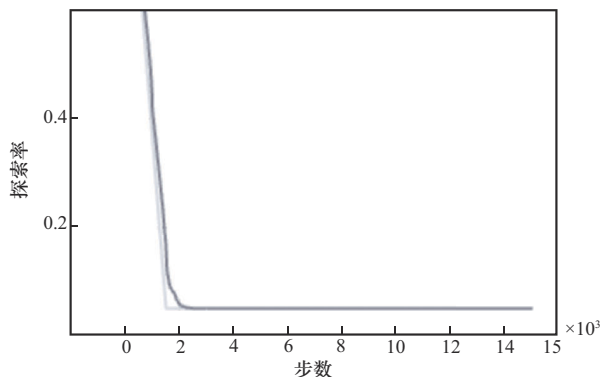


图7 Malconv 对抗模型探索率的变化趋势

Figure 7 Trend of exploration rate changes in the Malconv adversarial model

可以观察到，模型的探索率在大约 1 600 步时降低到了最小值 0.05，这说明模型已经从探索新策略的阶段转向利用已知有效策略的阶段。实验在模型行为模式逐步稳定后，每隔 1 000 步记录一次历史模型，并从中挑选出最优的生成模型。随后，从总数据集中随机抽取 1 000 个恶意 ELF 文件，对选定的模型进行性能评估。各检测模型对应的对抗样本生成模型表现如表3所示。

表3 对抗样本生成模型表现

Table 3 Performance of the adversarial example generation model

目标模型	逃逸成功率	平均步长
Malconv_100k	41.00%	2.27
Malconv_200k	48.20%	8.39
ResNet18	76.80%	3.12

实验结果表明，这些基于深度强化学习的生成模型对目标检测器具有一定的对抗样本逃逸成功率，并且平均修改步数较少，这表明这些模型能够在较短的时间内有效地生成对抗样本。此外，实验还发现，具有较大输入空间的模型更容易受到攻击，这可能是因为更大的输入空间间接提供了更大的扰动空间，使对抗模型能够在原始恶意文件中添加更多有效的扰动字节，从而误导检测器的判断，平均步长变长也侧面印证了这个猜想。

在相同检测模型上进行逃逸攻击实验，深度强化学习方法与文献[21]中验证的3种基础方法的逃逸攻击结果比较，如表4所示。

表4 逃逸攻击结果比较

Table 4 Comparison of evading attack outcomes

方法	逃逸成功率	
	Malconv_100k	Malconv_200k
动态扩展	8.1%	4.8%
填充更改	8.4%	5.3%
末尾附录	7.3%	4.5%
深度强化学习	41.0%	48.2%

从表4可以看出,利用深度强化学习构造扰动序列能够显著提高对抗样本的逃逸成功率。此外,当Malconv模型的输入空间增大时,使用固定字节作为扰动序列的3种基础方法的逃逸成功率有所降低,而深度强化学习方法生成的对抗样本逃逸成功率反而有所提高。

但需要注意的是,使用深度强化学习构建扰动序列会消耗更多的时间。实验数据显示,针对Malconv_100k模型,平均生成一个对抗样本需要47.2 ms,而使用基础方法只需7.5 ms。这表明尽管深度强化学习方法在对抗样本逃逸率上具有显著优势,但其生成效率仍有待提升,未来可以通过优化中间过程和模型选择来提高整体性能。

(3) 原有功能保留验证实验

为了验证本文提出的对抗样本生成方法不会影响ELF文件的原有功能,实验在本地Linux环境中运行了若干个具有代表性功能的ELF文件对应的生成样本,并测试它们的功能完整性。ELF文件功能测试结果如表5所示。

表5 ELF文件功能测试结果

Table 5 Function test results for ELF files

文件名称	功能描述	运行结果
cd	改变所在目录	正常
touch	创建文件	正常
rm	删除文件	正常
top	显示进程信息	正常
ping	测试网络连通性	正常

实验结果证明,本文方法中涉及ELF文件修改的扰动字节序列插入操作并不会损害文件的原有功能。

(4) 对抗训练实验

为了增强检测模型的鲁棒性,将逃逸攻击实验中生成的1 000个对抗样本加入检测模型的训练

集,在进行了相同数量的训练周期之后,另抽取了1 000个原始恶意文件进行逃逸攻击测试,可得对抗训练后的对抗样本生成模型表现如表6所示。

表6 对抗训练后的对抗样本生成模型表现

Table 6 Performance of the adversarial example generation model after adversarial training

目标模型	逃逸成功率	平均步长
Malconv_100k	11.20%	2.64
Malconv_200k	17.50%	10.14
ResNet18	44.10%	7.52

通过对比训练前后的数据观察到,在面对经过对抗训练增强的检测模型时,对抗样本生成模型的性能显著下降。具体而言,对抗样本的逃逸成功率明显降低,同时生成对抗样本所需的平均步长也有所增加。这些变化表明检测模型的鲁棒性得到了显著提升,其稳定性、可靠性和泛化能力都有所增强,能够更加有效地应对复杂多变的安全威胁。

5 结束语

本文提出了一种基于深度强化学习的恶意ELF文件检测对抗方法,并基于该方法实现了对抗样本生成模型。实验结果表明,该方法能够高效地生成目标检测模型的对抗样本,生成的对抗样本也能够保留其原始功能。同时,这些对抗样本参与的对抗训练有效地提升了检测模型的鲁棒性。

在未来的工作中,将进一步优化对抗样本的生成效率,并通过生成的对抗样本探索更多恶意文件检测模型构建与强化的方法,提升基于深度学习的恶意软件检测系统的安全性和可靠性。

参考文献:

- [1] GALANTE L, BOTACIN M, GRÉGIO A, et al. Malicious Linux binaries: a landscape[C]//Proceedings of the Anais Estendidos do XVIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSEG Estendido 2018). 2018: 213-222.
- [2] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2016: 770-778.
- [3] AMODEI D, ANUBHAI R, BATTENBERG E, et al. Deep speech 2: end-to-end speech recognition in English and mandarin[EB]. arXiv preprint arxiv: 1512.02595, 2015.
- [4] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep

- bidirectional transformers for language understanding[EB]. arXiv preprint arXiv: 1810.04805, 2018.
- [5] JEON J, JEONG B, BAEK S, et al. Hybrid malware detection based on Bi-LSTM and SPP-net for smart IoT[J]. IEEE Transactions on Industrial Informatics, 2022, 18(7): 4830-4837.
- [6] WANG H R, HE H, ZHANG W Z. Demadroid: object reference graph-based malware detection in android[J]. Security and Communication Networks, 2018: 7064131.
- [7] CORUM A, JENKINS D, ZHENG J. Robust PDF malware detection with image visualization and processing techniques[C]//Proceedings of the 2019 2nd International Conference on Data Intelligence and Security (ICDIS). Piscataway: IEEE Press, 2019: 108-114.
- [8] COULL S E, GARDNER C. Activation analysis of a byte-based deep neural network for malware classification[C]//Proceedings of the 2019 IEEE Security and Privacy Workshops (SPW). Piscataway: IEEE Press, 2019: 21-27.
- [9] KOLOSINJAJI B, DEMONTIS A, BIGGIO B, et al. Adversarial malware binaries: evading deep learning for malware detection in executables[C]//Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO). Piscataway: IEEE Press, 2018: 533-537.
- [10] DEMETRIO L, BIGGIO B, LAGORIO G, et al. Explaining vulnerabilities of deep learning to adversarial malware binaries[EB]. arXiv preprint arXiv: 1901.03583, 2019.
- [11] KREUK F, BARAK A, AVIV-REUVEN S, et al. Deceiving end-to-end deep learning malware detectors using adversarial examples[EB]. arXiv preprint arXiv:1802.04528, 2018.
- [12] DEMETRIO L, COULL S E, BIGGIO B, et al. Adversarial examples[J]. ACM Transactions on Privacy and Security, 2021, 24(4): 1-31.
- [13] RAFF E, BARKER J, SYLVESTER J, et al. Malware detection by eating a whole exe[C]//Proceedings of the Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
- [14] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[EB]. arXiv preprint arXiv: 1312.5602, 2013.
- [15] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518 (7540): 529-533.
- [16] SZEGEDY C, ZAREMBA W, SUTSKEVER I, et al. Intriguing properties of neural networks[EB]. arXiv preprint arXiv: 1312.6199, 2013.
- [17] GOODFELLOW I J, SHLENS J, SZEGEDY C, et al. Explaining and harnessing adversarial examples[J]. arXiv preprint arXiv: 1412.6572, 2014.
- [18] AL-DUJAILI A, HUANG A, HEMBERG E, et al. Adversarial deep learning for robust detection of binary encoded malware[C]//Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW). Piscataway: IEEE Press, 2018: 76-82.
- [19] KOLOSINJAJI B, DEMONTIS A, BIGGIO B, et al. Adversarial malware binaries: evading deep learning for malware detection in executables[C]//Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO). Piscataway: IEEE Press, 2018: 533-537.
- [20] DING Y X, SHAO M M, NIE C, et al. An efficient method for generating adversarial malware samples[J]. Electronics, 2022, 11 (1): 154.
- [21] KOSIKOWSKI A, CHO D, NINAN M, et al. EvilELF: evasion attacks on deep-learning malware detection over ELF files[C]//Proceedings of the 2023 International Conference on Machine Learn-

ing and Applications (ICMLA). Piscataway: IEEE Press, 2023: 1702-1709.

[作者简介]



孙贺 (1990-), 男, 博士, 试验物理与计算数学重点实验室助理研究员, 主要研究方向为网络安全、二进制代码分析、漏洞挖掘。



张博成 (2003-), 男, 四川蓬安人, 主要研究方向为恶意代码分析。



耿嘉炫 (1994-), 男, 福建福州人, 四川大学博士生, 主要研究方向为恶意软件分析、网络攻防对抗、机器学习安全。



吴迪 (1999-), 男, 四川广安人, 四川大学博士生, 主要研究方向为软件逆向分析、软件安全、恶意软件分析。



王俊峰 (1976-), 男, 安徽芜湖人, 四川大学研究员、博士生导师, 主要研究方向为空间信息网络、网络与信息安全。



方智阳 (1986-), 男, 浙江永康人, 四川大学副教授, 主要研究方向为二进制安全、固件漏洞挖掘。