注：此处无需更改。

阅后删除此文本框。



本科生毕业设计(论文)

基于强化学习的对抗性PE可执行程序恶意样本生成

**Reinforcement Learning-Based Adversarial PE Executable Malicious Sample Generation**

|  |  |
| --- | --- |
| 学 院： | 计算机学院 |
| 专 业： | 计算机科学与技术 |
| 班 级： | 07112105 |
| 学生姓名： | 李函书 |
| 学 号： | 1120213139 |
| 指导教师： | 田东海 |

2025 年 月 日

原创性声明

本人郑重声明：所呈交的毕业设计（论文），是本人在指导老师的指导下独立进行研究所取得的成果。除文中已经注明引用的内容外，本文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

特此申明。

本人签名： 日 期： 年 月 日

关于使用授权的声明

本人完全了解北京理工大学有关保管、使用毕业设计（论文）的规定，其中包括：①学校有权保管、并向有关部门送交本毕业设计（论文）的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存本毕业设计（论文）；③学校可允许本毕业设计（论文）被查阅或借阅；④学校可以学术交流为目的,复制赠送和交换本毕业设计（论文）；⑤学校可以公布本毕业设计（论文）的全部或部分内容。

本人签名： 日 期： 年 月 日

指导老师签名： 日 期： 年 月 日

北京理工大学本科生毕业设计（论文）题目

注：此处是中文题目，居中，字体：黑体，加黑，字号：小二，行距：单倍行距，间距：段前、段后均为1行，取消网格对齐选项。阅后删除此文本框。

摘　要

注：此处无需更改。阅后删除此文本框。

近年来，随着勒索病毒和恶意PE可执行文件木马程序的流行，目前存在的一些反病毒软件的查杀引擎，例如360杀毒的QVM2反病毒引擎，对于恶意PE可执行文件的检测仍然有较多的漏报、误报问题，仍需要用户主动向反病毒软件厂商提交样本漏报和误报的样本。同时，部分反病毒软件过度依赖动态检测来对抗恶意软件，静态检测能力仍然有不足。本文提出了一种基于强化学习的Sarsa算法的静态对抗型样本生成框架，输入已有的PE可执行程序恶意样本来生成能够规避部分反病毒软件静态检测的对抗性样本。本实验的行为表定义了对PE文件执行扰动的操作，这些操作基于Python的lief库和pefile库，样本集基于国内反病毒软件的官方论坛提供的样本以及Virus Share的样本， 某个样本集经处理前后，Clam AV对其查杀率从27.8%下降到了21.9% 360杀毒对其的查杀率从99.3%下降到了66.5%。本实验中强化学习模型生成的对抗性样本能有一定程度的静态检测对抗性，虽然对于火绒杀毒的查杀率下降影响并不显著。总而言之，使用本文中提出的方法生成的对抗样本集对杀毒软件进行训练，可以训练基于深度学习和机器学习的反病毒引擎和反病毒软件的对于恶意软件的检测能力，对恶意PE可执行文件的对抗性样本生成领域有一定的贡献。

关键词：强化学习；静态分析规避；Sarsa算法；PE恶意可执行文件

注：页脚内容为页码，宋体、五号，居中排列。阅后删除此文本框。

Reinforcement Learning-Based Adversarial PE Executable Malicious Sample Generation

注：正文设置成每段落首行缩进2字符，字体：Times New Roman，字号：小四，行距：固定值22磅，间距：段前、段后均为0行。阅后删除此文本框。

注：此处是英文题目，居中，字体：Times New Roman，加黑，字号：三号，行距：单倍行距，间距：段前、段后均为1行，取消网格对齐选项。阅后删除此文本框。

Abstract

In recent years, with the prevalence of ransomware and malicious PE executable trojans, existing antivirus detection engines — such as 360 Antivirus's QVM2 engine — still exhibit significant false negatives and false positives in detecting malicious PE executables. This necessitates users to proactively submit samples of missed or misclassified threats to antivirus vendors. Furthermore, some antivirus solutions over-rely on dynamic analysis to counter malware, revealing deficiencies in static detection capabilities.

This paper proposes a static adversarial sample generation framework based on the reinforcement learning SARSA algorithm. By inputting existing malicious PE executable samples, the framework generates adversarial variants capable of evading static detection by certain antivirus software. The experiment defines perturbation operations for PE files using Python's *lief* and *pefile* libraries. The dataset combines samples from Chinese antivirus vendors' official forums and Virus Share. After processing, the detection rates declined as follows: Clam AV dropped from 27.8% to 21.9%, while 360 Antivirus fell sharply from 99.3% to 66.5%. Although the adversarial samples showed limited impact on Huo rong Antivirus's detection rate, the reinforcement learning model demonstrated measurable success in evading static analysis.

In conclusion, training antivirus engines (particularly those leveraging deep learning or machine learning) with adversarial samples generated by this method can enhance malware detection capabilities. This work contributes to the domain of adversarial sample generation for malicious PE executables.

Key Words: Reinforcement learning; Static analysis evasion; Sarsa algorithm; PE malicious executable

注：Key Words与摘要正文之间空一行。Key Words与中文“关键词”一致（3-8个）。词间用分号间隔，末尾不加标点；Times New Roman，小四，加粗。阅后删除此文本框。

目　录

注：此处无需更改。阅后删除此文本框。

[摘　要 I](#_Toc196122499)

[Abstract II](#_Toc196122500)

[第1章 引言 1](#_Toc196122501)

[1.1 目前恶意软件的威胁 1](#_Toc196122502)

[1.2对抗性样本对于反病毒软件带来的挑战 1](#_Toc196122503)

[1.3关于过往的对抗性样本生成工作 2](#_Toc196122504)

[1.4 关于本实验的创新点以及主要贡献 3](#_Toc196122505)

[1.4.1 创新点 3](#_Toc196122506)

[1.4.2 主要贡献 4](#_Toc196122507)

[第2章 预备知识与环境配置 4](#_Toc196122508)

[2.1实验环境配置 4](#_Toc196122509)

[2.2实验预备知识 5](#_Toc196122510)

[2.2.1 关于PE文件的预备知识 5](#_Toc196122511)

[2.2.1 关于强化学习的预备知识 6](#_Toc196122512)

[第3章 实验方案及过程 10](#_Toc196122513)

[3.1样本收集与处理前查杀率测试 10](#_Toc196122514)

[结　论 14](#_Toc196122515)

[参考文献 15](#_Toc196122516)

[附　录 17](#_Toc196122517)

[致　谢 18](#_Toc196122518)

注：在目录页面中点击鼠标右键，选择“更新域”，在弹出窗口中选择“更新整个目录”，确定即可自动生成目录。章、节标题和页码，字体：宋体，字号：小四，不加粗。阅后删除此文本框。

第1章 引言

1.1 目前恶意软件的威胁

恶意软件是经过精心设计，被用来攻击计算机系统或计算机网络并且造成损害的软件，危害不仅限于感染型病毒和蠕虫的自身复制耗尽系统资源，破坏操作系统、后门软件开放系统端口供黑客连接从而形成僵尸网络对服务器发起分布式拒绝服务攻击、木马伪装成正常程序，实则窃取用户的敏感数据和破坏系统，为黑客提供后门、勒索病毒加密文件要求用户支付赎金等。根据统计数据，仅2022年在全球大约发生了55亿次恶意软件攻击事件。[a1]

在早期，恶意软件的代码较为简单，容易被反病毒软件检测到特征值从而处理清除。但是多年以来，恶意软件的复杂性不断发展，传统的基于黑名单哈希值的恶意软件检测技术难以应对当今恶意软件复杂的混淆策略，尽管这种方法速度很快，但是难以识别新一代的恶意软件以及一些0day恶意软件。[a2]

1.2对抗性样本对于反病毒软件带来的挑战

目前，为了应对恶意软件带来的威胁，许多开源以及商业杀毒软件厂商不断升级病毒库，更替杀毒软件版本。目前针对恶意软件的识别主要分为静态分析、动态分析以及混合分析（同时结合了静态分析和动态分析）这三种。而本实验研究的强化学习模型，是生成针对反病毒软件的静态特征分析的PE可执行程序对抗性样本，目的是用于反病毒软件厂商的机器学习和深度学习反病毒引擎模型训练和逆向分析工作者们学习研究，本实验未研究动态特征分析对抗性样本生成。静态特征分析包括文件Hash（如MD5 SHA256）匹配，这在很多病毒样本分析网站，例如VirusTotal，VirSCAN等被使用，如果上传的文件和病毒库里面存在的样本的Hash值匹配，则反病毒软件判断该文件是病毒。此外，静态分析还包括资源节分析，时间戳检查，数字签名检查，函数导入表检查，特征字符串匹配，DEBUG信息检查等。

在过去的十几年中，学术界出现了大量利用机器学习和强化学习的恶意软件查杀模型判断恶意软件的研究成果，甚至可以运用启发式杀毒规则，检测分析软件的代码行为，来判断从未出现过的新型恶意软件。[a3][a4][a5][a6][a7][a8]。

但不幸的是，某些基于静态特征分析的反病毒软件在某些情况下有着很高的误报率，可能因为机器学习模型自身的问题，误判一些正常的软件的行为，认为这些正常软件是恶意软件，例如QVM反病毒引擎误报Microsoft Visual Studio Complier、Clang等C/C++语言编译器，以及LLVM语法分析器等软件。此外，基于机器学习的反病毒软件也存在严重的漏洞，攻击者只需要对恶意软件进行修改，甚至有时只需要增加一个资源文件改变恶意软件的Hash值，就能绕过反病毒软件的检测，恶意软件的编写者的规避技术对于反病毒软件的识别带来了极大的挑战，这促使攻防对抗的双方不断采取更先进的措施。

1.3关于过往的对抗性样本生成工作

在过往的PE对抗性样本生成实验中，许多已有的对抗性模型生成，有些采用值函数的强化学习算法，例如使用深度学习神经网络的DQN。[a9]

有些使用遗传编程算法，基于适应度进行选择，交叉，编译。[a10]

有些采用蒙特卡洛搜索树方法，将对抗样本转化为路径搜索问题。[a11]

这些现有的方法已经展现了一定有效性，但时序差分算法，例如Sarsa Q-learning的研究却很稀少，因此本实验的主要研究目标是使用时序差分算法Sarsa构建强化学习对抗性样本生成模型，来生成能够规避部分静态检测的PE可执行程序对抗样本，使部分样本逃逸反病毒软件的查杀。

基于强化学习的方法是通过构建动作（Action），状态（State），奖励（reward），来使智能体与环境交互，通过多次操作获取经验，挑选一个相对更好的策略来修改已有的PE恶意程序来达到逃逸反病毒软件检测。

在2017年，基于强化学习生成静态对抗性样本绕过黑盒的反病毒软件的GYM-Malware框架被提出。Anderson等人定义了10种不会影响恶意软件和功能的扰动操作，例如利用签名漏洞来更改恶意软件的签名，修改恶意软件的调试（Debug）信息，修改可选首部检验码（checksum），修改现有的节的名称。这些操作与DQN强化学习算法结合和病毒检测软件交互以指导选择的扰动操作[a13][a14]。

但遗憾的是，一部分使用强化学习模型生成的对抗性样本无法在虚拟机中正常执行，有一部分恶意软件经过某些操作后被破坏了。尽管某些操作修改的部分似乎与恶意软件的代码部分无关，但还是影响到了恶意程序的功能。[a8]

但GYM-Malware模型仍然对后续的研究有着很大的作用，许多后续的研究基于其工作来进行。其中封装的一部分动作（Action），被拆成了函数用于其他的研究中，例如Mab-malware。

有一些研究[a15]，关注了GYM-Malware中的一些造成恶意软件功能损坏无法正常执行的操作。Mab-Malware认为是Python的LIEF库导致的，并且对其进行修复，将使用LIEF库的一些对PE可执行程序的操作改为使用Python的Pefile库，以减少损坏的恶意程序数量。也有一些研究[a16]，直接删除了可能导致恶意程序遭到破坏的操作，引入随机化操作来缩小动作空间，限制强化学习智能体可执行的动作次数，鼓励强化学习智能体寻找更优秀的Action集合。

而由Labaca-Castro等人开展的研究[a17]，则考虑修正奖励函数，对奖励函数添加惩罚因子，为了进一步优化强化学习智能体的操作。对奖励函数的修正鼓励智能体尽可能用更少的步骤对恶意PE程序进行修改以逃逸反病毒软件的检测。

类似的，Gibert et al.等人使用空操作[a18]，即插入大量NOP指令来修改恶意软件，表明使用插入无意义空操作的方法对于绕过MalConv等反病毒软件一样是有效的。

同时，目前也存在基于梯度的对抗性样本生成，例如FGSM、 Carlini和Wagner创建的C&W、以及deepfool模型。[a19][a20][a21]

且存在一些研究能够在已知梯度信息的情况下，通过基于梯度的方法对恶意软件的字节或者外观表现形式进行修改来规避静态特征检测[a22][a23][a24][a25][a26][a27]。

1.4 关于本实验的创新点以及主要贡献

1.4.1 创新点

本实验采用的强化学习算法是基于时序查分算法的Sarsa算法。关注PE恶意程序的原因是Windows操作系统在个人电脑（PC），服务器等操作系统中占有72%左右的份额，因此超过70%的恶意软件将Windows操作系统作为攻击目标[a28][a29]。相比已有的工作，本实验额外考虑了UPX加壳操作和sigthief造成的假数字签名，resourceHacker调用导致的图标类资源添加等行为，对静态检测的对抗性样本生成可能带来的影响。同时，本实验生成的部分样本能逃逸已有的反病毒软件，为相关研究提供了参考，为基于机器学习的反病毒引擎提供了可选的训练样本生成器。

1.4.2 主要贡献

本文的主要贡献如下：

（1）研究了时序差分强化学习算法Sarsa对于对抗性样本的生成。

（2）能够生成高概率逃逸360杀毒云查杀的对抗性样本并且保证恶意程序的功能，生成的恶意程序样本即使开启了允许上传可疑文件，仍然需要经过多次扫描后上传分析才得以查杀。

（3）预测了未来可能出现的高危害性感染型病毒变种和结合了静态分析的挂马网站，以及部分木马下载器变种，并且建议反病毒软件厂商们加以防范，尽管动态分析技术仍然可以针对感染型病毒变种，但该类型病毒仍具有一定威胁性。这种感染型病毒变种会在感染新文件时对于新产生的病毒自带进行自动地静态检测规避处理，同时被感染的文件相较于传统的感染型病毒难以恢复。

（4）action更加倾向于模块化，减少了模块之间的耦合程度，便于后续的研究人员从项目中直接提取函数，而无需修改大量内容，对于某些扰动函数只需要传入原恶意程序文件的绝对路径，和用户期望生成的文件的绝对路径，就能实现对抗型扰动操作。相比原有框架PSP-Mal[a12]，本实验项目中的扰动函数与行为（Action）相互分离，并且额外加入日志系统，扰动函数自身不需要传递过多的参数。而原框架PSP-Mal对于恶意软件的操作相对混乱，难以单独从一个行为（Action中）抽象出修改函数，对恶意软件的扰动行为和智能体之间耦合度过高，导致修改较为困难。

第2章 预备知识与环境配置

2.1实验环境配置

本实验使用如下开发环境：

实体机操作系统及版本号：Windows10 22H2

虚拟机软件，虚拟机操作系统：VMWare Workstation 16.1.2 build-17966106

Windows7 旗舰版

集成开发环境：PyCharm 2024.3.5 (Professional Edition)

Anaconda 24.11.3 AnacondaNavigator2.6.5

对抗性样本测试：主机使用Clam AV，并且手动更新病毒库到最新版本。

虚拟机采用火绒杀毒和360杀毒。

项目需要的依赖已经导出在项目根目录下的requirements.txt文件中，读者可以使用anaconda直接导入requirements.txt文件来创建Python虚拟环境。

实体机存储：

内存空间 32GB

磁盘空间 512GB

2.2实验预备知识

2.2.1 关于PE文件的预备知识

PE文件是Windows操作系统下的可执行文件形式，包括EXE（可执行文件），DLL（动态链接库），SYS（系统文件）等类型。

而本实验的对抗性样本生成操作，需要修改PE文件某些节的信息。

PE文件中包含PE文件头（IMAGE\_NT\_HEADERS），其中的COFF文件头（IMAGE\_FILE\_HEADER）和可选首部（IMAGE\_OPTIONAL\_HEADER）中的有些部分是我们需要修改的目标。

COFF文件头（IMAGE\_FILE\_HEADER）包含如下关键字段：

Machine（目标CPU架构，指明了能运行这个程序的机器码，可以指明支持程序运行的机器架构是x86、x64、PowerPC、ARM等）

NumberOfSections：指明所有节区的数量。

TimeDateStamp：时间戳，指明了这个文件被编译生成的时间。

SizeOfOptionalHeader：可选首部的大小。

Characteristics：文件的类型，是动态链接库，还是可执行文件等类型。

可选首部（IMAGE\_OPTIONAL\_HEADER）包含如下字段，可选首部指明了当PE程序被载入内存后的一些情况：

Magic：魔法位，包含PE信息，一定要和COFF文件头中的Machine对应，否则就会报错导致程序无法启动。。

AddressOfEntryPoint：程序入口点，即相对虚拟地址。

ImageBase：加载机制。

SectionAlignment：内存中节区的对齐粒度，不建议修改，否则程序可能无法启动。

FileAlignment：文件中节区的对齐粒度，不建议修改，否则程序将无法启动

SizeOfImage：加载到内存后的总大小。

SubSystem：子系统类型。

DataDirectory：数据目录表，记录某些数据的位置及其大小。

PE文件中的节表（Section table）描述了每个节的属性，由多个IMAGE\_SECTION\_HEADER组成，每个条目对应一个节区，包括下列关键属性：

Name：节区的名字。

VirtualAddress：虚拟地址中的起始相对位置。

SizeOfRawData：节区中数据的大小。

PointerToRawData：节区偏移量。

Characteristics：节区属性。

节区数据正常情况下应包含如下内容：

.text：代码段，存放该PE程序执行的指令。

.data：已经完成初始化的某些数据。

.rdata：只读（Read Only）的数据。

.rsrc：资源节，存放PE文件的图标等信息。

.reloc：重定位信息，可以用于动态链接库的装载过程。

.idata：import data，即导入函数信息。

数据目录表中包含的内容如下：

Import Table：导入表，用于存放该PE文件依赖的动态链接库和调用的某些函数。

Export Table：导出表，可以用于存放这个PE文件封装好的函数，多半运用于动态链接库封装函数供其余PE程序调用使用。

Relocation Table：重定位表，修复地址偏移相关问题。

TLS：线程存储表，与多线程程序有关，存储线程初始化数据。

Debug Directory：存放该PE程序的调试信息。

本实验对抗性模型生成只使用了PE可执行文件（EXE），并未使用动态链接库（DLL）和系统文件（SYS）。

2.2.1 关于强化学习的预备知识

基本概念相关：

1.智能体：在强化学习中决策，行动，学习。智能体是一个感知者，能感知并且理解当前的状态，智能体是一个决策者，能够知道在一个状态下应该采取什么行动，智能体是一个执行者，通过改变状态从而获取奖励。

2.状态：描述了智能体与环境的相对状况。

3.状态空间：所有状态的集合。

4.动作：智能体在某一状态下能选择的操作。

5.动作空间：所有动作的集合。

6.状态转移：当执行一个动作时，智能体可能从一个状态转移到另一个状态的过程。

7.策略：智能体在每一个状态下应该采取什么样的动作，允许分为确定性策略和随机性策略。8.奖励：作为人机交互的一个重要手段，可以设置合适的奖励来引导智能体按照我们的预期选择正确的决策，正数奖励表明我们鼓励智能体执行该行动，负数奖励表明我们不鼓励智能体执行该行动。

9.回合/尝试：智能体执行一个策略与环境交互的过程中，智能体从开始状态到终止状态停止的过程被称为一个回合或尝试，一般用英文episode来表示

10.折扣因子：用于调整智能体对于近期奖励和远期奖励的重视程度，可以记作折扣因子为γ，γ在(0,1)的范围，且折扣因子的引入允许了无限长的轨迹

11.状态值：表达式为式（2-1）

（2-1）

状态值说明智能体在一个状态之下，最终能获取到的回报。首先，需要了解基于时序差分策略的方法可被用于估计状态值：

时序差分方法的表达式为式（2-2）：

（2-2）

(

代表t时刻对于的估计，是t状态下对于状态的学习率。

在t时刻，只有当时正在被访问的状态的估计值会被更新。

在式（2-2）中：

作为时序差分方法的目标。

作为时序差分方法的误差。

为在t时刻对于状态的学习率。

为新的t+1时刻对于状态值估计值。

为t时刻对于状态的状态值的估计值。

Sarsa（state action reward action state action）是基于时序查分方法的强化学习算法，但是该方法不是估计状态值而是估计动作值[a30][a31]。

首先要引入动作值的概念：

对于一个状态-动作配对，动作值定义表达式为式（2-3）

= （2-3）

动作值表示在一个状态采取一个动作之后获得回报的期望值。

这里将的估计值记作。

给定一个策略π，需要估计其动作值，可以从π的经验样本中，使用Sarsa算法来估计动作值，其表达式为式（2-4）：

（2-4）

式（2-4），学习率为

在t时刻，只有的动作值被更新，其它的动作值保持不变。

Sarsa算法主要是用于求解Bellman方程近似算法，近似算法的表达式如式（2-5）所示：

 （2-5）

式（2-5）是一个基于动作值而不是状态值的Bellman方程。

运用Sarsa算法学习最优策略的伪代码如下：

目标：学习最优策略，使智能体能从给定状态到达目标状态。

初始化函数：

Init():

for all values and all t values:

select =

var

set init value as

训练学习函数：

Train():

for episode in episodes:

if s == :

from let a =

if !=target\_state:

collect experience samples(, , , )

run action

generate ,

generate from

update value:

update strategy of :

if a == :

else if a!= :

let =

let =

Sarsa算法描述如下：

算法初始化：

对于所有数值和所有t数值，选取学习率因子 *= ，*贪婪因子，并且设定所有的初始值为，从获取初始贪婪策略。

算法目标：学习最优策略，使智能体能从给定的状态出发，到达目标终止状态。

对于每个回合：

如果当前状态是根据选取策略 a =

在时刻t，如果不是目标状态

收集经验样本(, , , )：在通过与环境交互生成，。

再根据生成

随后更新(, 的值

Sarsa算法也存在一些推广，例如Expected Sarsa算法、n-step Sarsa算法等，尽管本实验中并未涉及到这些推广算法，但是读者在构建强化学习对抗性样本生成模型上也可以尝试使用它们。

其中Expected Sarsa算法类似Sarsa算法，但是它们的时序差分方法目标上不同。具体实现流程上二者相似。

已知策略π，则Expected Sarsa算法对于该策略π的动作值估计公式如式（2-6）所示：

（2-6）

在式（2-6）中：

（2-7）

式（2-7）是在策略下的期望值。

不同于Sarsa算法的目标是求出，

Expected Sarsa算法的目标是求出。

引入期望值会增加计算复杂度，但是会减小所求目标的方差，Expected Sarsa的涉及的随机变量是{}，相比Sarsa算法的{}，减少了一项，因此对于减少估计方差是有正面作用的[a31][a32][a33]。

第3章 实验方案及过程

3.1样本收集与处理前查杀率测试

首先，需要收集足够量的恶意样本以供强化学习模型使用，因此，笔者从国外Virus Share的开放恶意样本库，以及部分国内安全软件论坛，例如火绒杀毒，360杀毒等官方论坛，收集并且下载了大约3000个恶意样本。

但需要注意的是，有些样本可能不是PE文件，如图所示，笔者在手动筛选样本时，发现了其中的部分样本包包含其他类型的恶意程序，如图3-1所示：

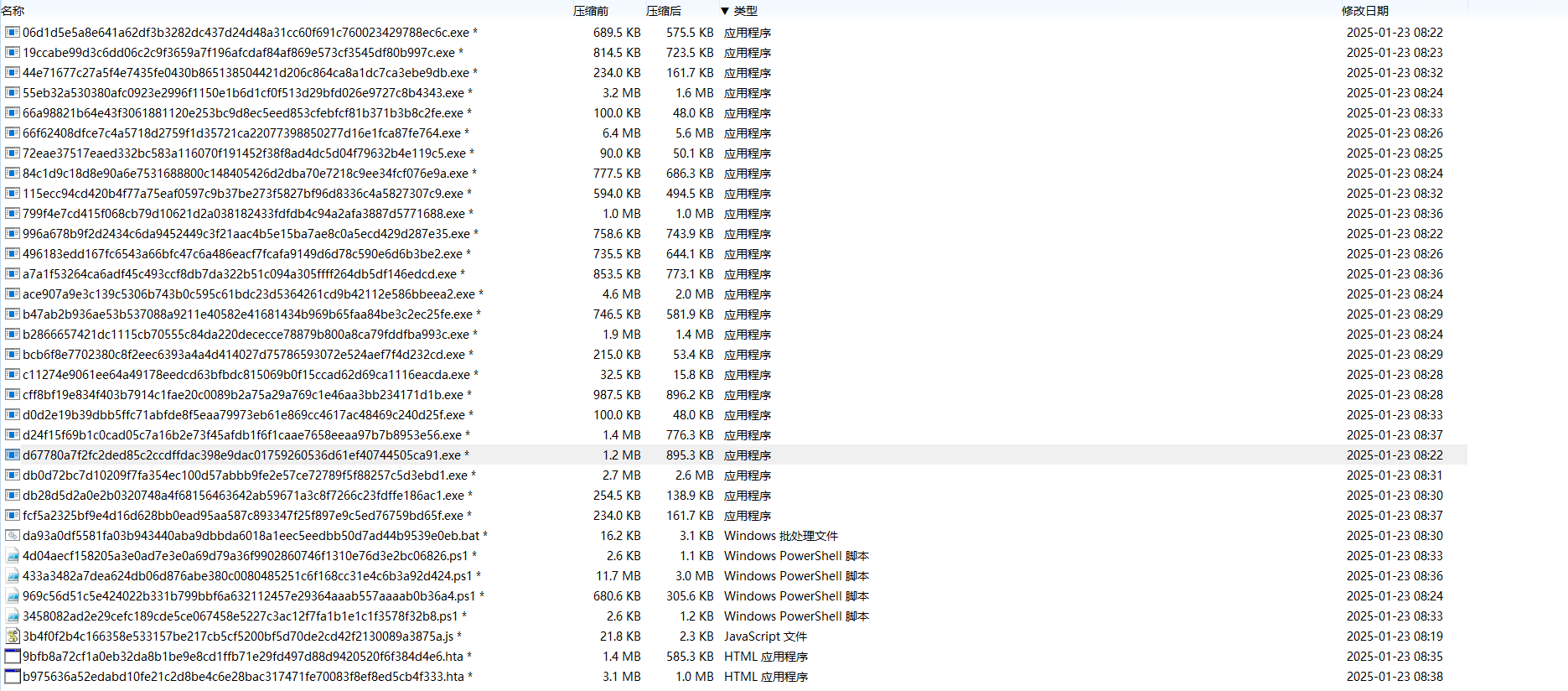
图3-1 样本包

图3-1表示了其中一个样本包，不难看出里面包含了恶意Windows Power Shell脚本和HTML恶意应用程序，恶意JavaScript脚本，恶意Visual Basic Script等非PE恶意程序。而非PE恶意程序的对抗性样本生成不是本研究的目标，所以对于样本的处理，需要删除掉这些非PE文件。其次，本实验并没有对恶意动态链接库（DLL）和SYS文件进行对抗性样本生成，只是对恶意PE可执行文件（EXE）进行了对抗性样本生成，是因为反病毒软件厂商提供的样本不包含动态链接库和SYS文件，且检验动态链接库在对抗性样本生成后是否因为对抗性扰动导致损坏，相比检验PE可执行文件是否损坏更困难。

本实验在筛选样本这一步操作时，前四组样本集，笔者选择了在Windows7虚拟机中手动筛选，利用Windows文件系统的相关功能，设置类型排序，然后删除后缀文件名非exe的文件，只保留PE可执行程序样本。然而，这有着比较大的操作隐患，很容易误操作，意外执行恶意程序导致可能遭受到恶意软件攻击。在后续的第五组样本集中，笔者将筛选操作自动化封装成了一个Python函数，只需要传入样本集所在的绝对路径即可自动化完成操作，同时也规避掉了手动筛选误操作的风险。

然而，上述的筛选看似合理，但是在实验中会存在一些比较严重的问题，例如有些是伪装的PE可执行文件，这些文件使用了结构化欺骗，实际上这些文件自身并不含有COFF文件头等内容，如果封装的对抗性扰动函数认为它们是PE可执行文件并且对其操作，就会因为PE格式解析异常报错，进而Python主程序将会抛出异常，没有被处理的异常会导致整个智能体的训练被迫终止，幸运的是，这可以通过在do\_action( )函数中设置异常处理来解决这个问题，即使遇到了不是正常的PE可执行文件，程序仍然因为不会抛出异常，导致强制终止整个程序。另外还存在一些PE可执行文件恶意软件，自身大小非常大，大约64MB到数百MB，可能是这些文件本身已经经过了一些对抗性操作导致的，笔者在对对抗性扰动的函数进行黑盒测试中发现，生成对抗性样本的扰动函数在处理它们时极其缓慢，有些甚至需要数分钟的时间，这无疑严重拖累了强化学习智能体自身运行的速度，因为每个回合（episode）是不能多线程并发执行的，必须要按照单线程，进行顺序执行。其次，这些较大的PE可执行程序读取和写入操作对磁盘的损害也是巨大的，极有可能在运行中发生系统蓝屏死机的问题，甚至可能会导致数据丢失。笔者的计算机在运行第五组样本时已经发生过一次因为硬盘故障导致的Windows蓝屏问题，幸运的是，重启后固态硬盘自检后成功修复。因此，我们还需要对样本进行第二次筛选，在本实验中，样本中大多数大于4096字节的PE可执行程序在挑选样本时需要被删除，以避免智能体出现故障导致程序强制终止。

当筛选出大约400~600个PE可执行程序恶意软件时，将这些样本归为一个样本集，以备后续使用。同时，我们需要对原始样本使用反病毒软件扫描。这一步笔者是在Windows10的实体机上配置了Clam AV，使用最新的病毒库，并且在系统服务中需要手动开启Clam AV的服务。对于Clam AV的病毒扫描服务，笔者选用的是3310端口号，需要在clamd.conf中修改TCPSocket属性的值来指定服务端口号，读者在这一步环境配置时需要注意端口号冲突问题，否则服务将无法启动，同时这个端口号也是Python调用Clam AV服务所需要的。图3-2中所示是Clam AV服务成功启动的状况。

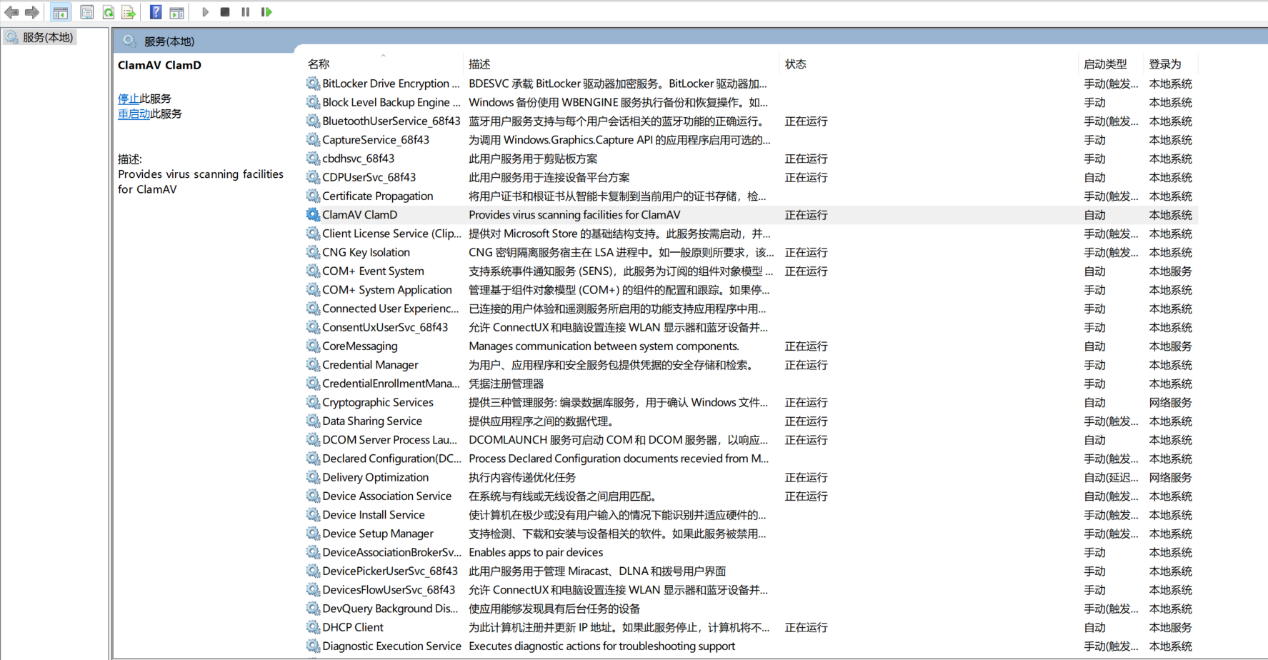


图3-2 Clam AV服务

但是在使用360杀毒，火绒杀毒等具有云查杀功能的反病毒软件对原始样本进行扫描时，需要注意样本污染问题，对于360杀毒，需要关闭自动上传发现的可疑文件，如图3-2所示，否则可能导致有些样本被360杀毒上传到云查杀服务器数据库中，导致后续结果查杀率偏高的问题。

图3-2 关闭360杀毒的云查杀上传可疑文件

结　论

注：此处无需更改。阅后删除此文本框。

本文结论……。

结论作为毕业设计（论文）正文的最后部分单独排写，但不加章号。结论是对整个论文主要结果的总结。在结论中应明确指出本研究的创新点，对其应用前景和社会、经济价值等加以预测和评价，并指出今后进一步在本研究方向进行研究工作的展望与设想。结论部分的撰写应简明扼要，突出创新性。【阅后删除此段】

结论正文样式与文章正文相同：宋体、小四；行距：22磅；间距段前段后均为0行。【阅后删除此段】

参考文献

注：此部分蓝色字体为注释，阅后可删除；黑色字体为具体示例。

阅后删除此文本框。

注：此处无需更改。阅后删除此文本框。

参考文献书写规范

参考国家标准《信息与文献参考文献著录规则》【GB/T 7714—2015】，参考文献书写规范如下：

1. 文献类型和标识代码

普通图书：M 会议录：C 汇编：G 报纸：N

期刊：J 学位论文：D 报告：R 标准：S

专利：P 数据库：DB 计算机程序：CP 电子公告：EB

档案：A 舆图：CM 数据集：DS 其他：Z

2. 不同类别文献书写规范要求

**期刊**

[序号] 主要责任者. 文献题名[J]. 刊名, 出版年份, 卷号(期号): 起止页码.

[1] 余雄庆. 飞机总体多学科设计优化的现状与发展方向[J]. 南京航空航天大学学报, 2008, 40(4): 417-426.

[2] Hajela P, Bloebaumj C L, Sobieszczanski-Sobieski J. Application of Global Sensitivity Equations in Multidisciplinary Aircraft Synthesis[J]. Journal of Aircraft, 1990, 27(12): 1002-110.

**普通图书**

[序号] 主要责任者. 文献题名[M]. 出版地: 出版者, 出版年: 起止页码.

[3] 张伯伟. 全唐五代诗格会考[M]. 南京: 江苏古籍出版社, 2002: 288.

[4] O’BRIEN J A. Introduction to information systems[M]. 7th ed. Burr Ridge, III: Irwin, 1994.

**会议论文集**

[序号] 主要责任者．题名:其他题名信息[C]. 出版地: 出版者, 出版年.

[5] 雷光春. 综合湿地管理: 综合湿地管理国际研讨会论文集[C]. 北京: 海洋出版社, 2012.

**专著中析出的文献**

[序号] 析出文献主要责任者. 析出题名[M]//专著主要责任者. 专著题名. 出版地: 出版者, 出版年: 起止页码.

[6] 白书农. 植物开花研究[M]//李承森. 植物科学进展. 北京: 高等教育出版社, 1998: 146-163.

**学位论文**

[序号] 主要责任者. 文献题名[D]. 保存地: 保存单位, 年份.

[7] 张和生. 嵌入式单片机系统设计[D]. 北京: 北京理工大学, 1998.

[8] Sobieski I P. Multidisciplinary Design Using Collaborative Optimization[D]. United States -- California: Stanford University, 1998.

**报告**

[序号] 主要责任者. 文献题名[R]. 报告地: 报告会主办单位, 年份.

[9] 冯西桥. 核反应堆压力容器的LBB分析[R]. 北京: 清华大学核能技术设计研究院, 1997.

[10] Sobieszczanski-Sobieski J. Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems[R]. NASA CP-3031, 1989.

**专利文献**

[序号] 专利所有者. 专利题名:专利号[P]. 公告日期或公开日期[引用日期]. 获取和访问路径. 数字对象唯一标识符.

[11] 姜锡洲. 一种温热外敷药制备方案: 881056078 [P]. 1983-08-12.

**国际、国家标准**

[序号] 主要责任人. 题名: 其他题名信息[S]. 出版地: 出版者, 出版年: 引文页码.

[12] 全国信息与文献标准化技术委员会. 文献著录: 第4部分 非书资料: GB/T 3792.4-2009[S]. 北京: 中国标准出版社, 2010: 3.

**报纸文章**

[序号] 主要责任者. 文献题名[N]. 报纸名, 年(期): 页码.

[13] 谢希德. 创造学习的思路[N]. 人民日报, 1998-12-25(10).

**电子文献**

[序号] 主要责任者. 电子文献题名[文献类型/载体类型]. (发表或更新日期) [引用日期]. 获取和访问路径. 数字对象唯一标识符.

[14] 姚伯元. 毕业设计(论文)规范化管理与培养学生综合素质[EB/OL]. [2005-02-02]. 中国高等教育网教学研究.

关于参考文献的未尽事项可参考国家标准《信息与文献参考文献著录规则》（GB/T 7714—2015）

附　录

注：此处无需更改。阅后删除此文本框。

附录相关内容…

附录是毕业设计（论文）主体的补充项目，为了体现整篇文章的完整性，写入正文又可能有损于论文的条理性、逻辑性和精炼性，这些材料可以写入附录段，但对于每一篇文章并不是必须的。附录依次用大写正体英文字母A、B、C……编序号，如附录A、附录B。【阅后删除此段】

附录正文样式与文章正文相同：宋体、小四；行距：22磅；间距段前段后均为0行。【阅后删除此段】

致　谢

注：此处无需更改。阅后删除此文本框。

值此论文完成之际，首先向我的导师……

致谢正文样式与文章正文相同：宋体、小四；行距：22磅；间距段前段后均为0行。【阅后删除此段】