信息安全研究 第8卷 第8期 2022年8月
Journal of Information Security Research Vol.8 No.8 Aug. 2022
DOI:10.12379/j.issn.2096-1057.2022.08.01

基于 LSTM 的 Linux 系统下 APT 攻击检测研究

时 林 时绍森 文伟平
(北京大学软件与微电子学院 北京 102600)
(shilin@stu.pku.edu.cn)

Research on APT Attack Detection Based on LSTM in Linux System

Shi Lin, Shi Shaosen, and Wen Weiping
(*School of Software and Microelectronics, Peking University, Beijing* 102600)

Abstract As people's daily life is covered by the network, the security of cyberspace has been paid more and more attention. There are many kinds of attack methods in the network. The APT attack is one of the more complex and harmful. It has the characteristics of strong sustainability and the attack process runs through the outside and inside of the system, and it is difficult to detect and thoroughly defend. This paper proposes a scheme of APT attack detection under a Linux system based on LSTM, constructs an analysis sandbox LAnalysis of malicious Linux ELF files based on kernel instrumentation to capture malicious behaviors in APT attacks, and constructs APT attack dataset by using LAnalysis analysis result dataset and network attack traffic dataset NSL-KDD according to attack timing characteristics, This paper solves the problem of lacking APT attack dataset under the current Linux system in the industry. Finally, the timing of APT attacks is introduced into the detection, and APT attacks are detected based on LSTM. The experimental results show that the APT attack detection model constructed in this paper has a good application effect.

Key words APT attack; Linux sandbox; long short-term memory; APT attack dataset; ELF file

摘 要 随着人们的日常生活被网络所覆盖,网络空间的安全问题也逐渐被重视起来.网络中的攻击手段多种多样,高级持续威胁(advanced persistent threat, APT)攻击是其中较为复杂并且危害性较高的一种,具有攻击过程贯穿系统外部与系统内部且持续性强的特点,并且难以检测与彻底防御.提出一种基于 LSTM(long short-term memory)的 Linux 系统下 APT 攻击检测方案,构建了一款基于内核插桩的分析恶意 Linux ELF 文件的沙箱 LAnalysis 来捕获 APT 攻击中的恶意行为;通过使用 LAnalysis 分析结果数据集结合网络攻击流量数据集 NSL-KDD,依据攻击时序特征构建了 APT

收稿日期:2022-06-09
基金项目:国家自然科学基金项目(61872011)
通信作者:文伟平(weipingwen@pku.edu.cn)
引用格式:时林, 时绍森, 文伟平. 基于 LSTM 的 Linux 系统下 APT 攻击检测研究[J]. 信息安全研究, 2022, 8(8): 736-750

攻击数据集，解决了当前业内 Linux 系统下 APT 攻击数据集较为缺乏的问题；最后将 APT 攻击中的时序性引入检测，基于 LSTM 进行 APT 攻击的检测.实验结果表明，构建的 APT 攻击检测模型具有良好的应用效果.

关键词　APT 攻击；Linux 沙箱；长短期记忆网络；APT 攻击数据集；ELF 文件

中图法分类号　TP309

随着网络逐渐渗透到人们生活的方方面面，网络空间的安全问题也逐渐被重视起来.网络中的攻击手段多种多样，高级持续威胁(advanced persistent threat，APT)攻击为其中较为复杂并且危害性较高的一种.APT 攻击过程贯穿系统外部与系统内部，且持续性很强，因此难以进行检测与彻底防御，需要得到更多的关注.近年来，全球 APT 组织持续增多，攻击涉及金融、政府、教育、科研等重点行业[1-2].到 2021 年上半年[3]，APT 攻击整体形势严峻，发现和披露的 APT 攻击活动较 2020 年同期大幅增加.

APT 攻击周期一般较长，在长期且持续的攻击下，攻击者会将多种攻击方式进行组合并调整，导致防护系统无法对经过精心处理的攻击特征完成规则匹配，最终导致目标机器遭到入侵.目前，大多数 IDS 系统仅具有单步攻击的检测能力，并没有将持续性的攻击联系起来，且检测内容往往局限于网络流量，如鱼叉攻击、XSS、SQL 注入等.一些 APT 攻击监控系统使用 IDS 的告警日志进行攻击判定.报警日志往往具有一定比例的误报，判定的攻击行为并不准确.更重要的是，仅检测流量往往会忽略完整的 APT 攻击过程，APT 攻击攻入系统后，会进一步释放控制机器的恶意代码，导致主机系统发生大量攻击.因此，将主机内的攻击行为与网络流量中的攻击行为相结合，共同作为 APT 攻击的判断条件是十分必要的，并且对于 APT 攻击检测而言，主机内的恶意行为非常重要.

目前，针对 Windows 系统的入侵检测系统与沙箱系统较多，而针对 Linux 系统攻击的分析与防御措施较为薄弱.很多恶意代码检测分析工具与防火墙都是针对 Windows 系统的，侧重于 Linux 系统的恶意文件检测手段较少.例如，开源的 Cuckoo 沙箱、商用的腾讯微步沙箱、奇安信文件分析平台等对 Windows 恶意文件的检测较为成熟，但是对于 Linux 平台上的恶意文件缺乏检测能力.因此针对 Linux 系统的 APT 攻击检测亟待解决.此外，目前业内普遍用于研究的攻击数据也只局限于网络流量数据或某些 APT 攻击释放的恶意文件，并不具有很强的关联性.APT 攻击持续时间长、攻击步骤繁多的特点导致 APT 攻击样本并不丰富，目前还没有形成一套科学的数据集供广大安全从业人员进行研究.

针对以上问题和研究现状，本文提出并实现了一种基于 LSTM(long short-term memory)的 Linux 系统下 APT 攻击检测方案.该方案综合了主机侧与网络侧的双侧行为特征，将特征数据集依据 APT 攻击的生命周期进行建模重构，进而使用 LSTM 进行训练，得到了检测效果良好的 APT 攻击检测模型.

本文主要贡献如下：

1) 捕获恶意 Linux ELF 文件行为的 LAnalysis 沙箱.

构建了一款能力较强的分析 Linux ELF 文件的 LAnalysis 沙箱，通过对相关内核函数以及系统调用函数的针对性内核插桩，LAnalysis 可以获取恶意代码的持久化、隐藏与伪装、权限提升、进程注入等 10 类共 16 种不同的恶意行为.

2) 符合 APT 攻击生命周期的数据集.

使用 LAnalyise 沙箱分析了 500 个恶意家族的共 4 101 个恶意样本.获取了恶意样本的主机侧攻击行为特征，构建了 Linux 主机侧的攻击数据集，并结合网络侧数据集 NSL-KDD 按照 APT 攻击生命周期，构建了一套兼具主机行为和网络行为特征的 APT 攻击数据集.

3) 基于 LSTM 的 APT 攻击检测模型.

将符合 APT 生命周期的数据集放入注重时序性特征的 LSTM 进行训练，其中包含网络与主机双侧特征，得到了可以检测 APT 攻击的深度学习模型，并取得了良好的应用效果.

1 相关工作

APT攻击生命周期较长，各种攻击行为之间具有一定的关联性，这给检测带来很大挑战.目前的检测手段主要分为侧重于主机侧APT攻击部署的恶意代码的检测、侧重于网络侧恶意流量分析的检测以及将多步攻击相结合的注重攻击关联性分析的检测等.

主机侧恶意代码检测主要是对APT攻击释放的木马[4]或后门等文件进行检测.当前针对可疑恶意程序的分析方法主要为动态分析与静态分析[5].冯学伟等人[6]利用恶意代码中使用的IP地址之间的联系进行聚类；霍彦宇[7]将分析恶意代码时产生的行为警报信息处理为特征，使用聚类的方法进行分类识别；Sharma等人[8]提出一种入侵检测框架，使用6个监视器监视系统中的行为，统计4天中各个文件的更改情况以及进程数据作为正常情况后续进行状态检测，若文件与进程数据出现异常就会发出威胁告警；Moon等人[9]提出一种基于主机中发生行为的攻击检测方法，通过捕获主机中39种特定行为的发生作为特征对APT攻击进行检测；孙增等人[10]提出基于沙箱回避对抗的相关检测方法，统计了常见沙箱中使用系统的各种特征，在代码运行前查找所运行系统的相关特征，进而判别当前软件是否在沙箱中运行.

网络侧恶意流量分析检测是对流量中的信息进行特征提取[11]，利用这些特征通过规则匹配或机器学习与深度学习训练模型等方式判定是否为异常流量.攻击者通过系统中运行的Web服务进行入侵，或从已经完成入侵的系统中横向移动至其他系统，在此期间都会产生大量的异常流量.因此检测相关攻击可以从异常流量入手.戴震等人[12]通过流量分析发现恶意软件的远端控制服务器对其进行指令发送的过程具有一致性，进而通过解析报文的通信特征对攻击进行判定；Chuan等人[13]通过结合机器学习模型形成了一种集成学习器对URL中的特征进行分析与提取，对具有恶意风险的网站进行识别；Liu等人[14]通过对数据集NSL-KDD进行处理形成了一套新的网络攻击数据集，使用DBN网络降维后通过SVD模型对可疑数据进行识别与分类.APT攻击流量检测中还有一部分是通过域名检测来判定恶意流量的.Vinayakumar等人[15]收集了网上公开的恶意域名数据集，并在系统中收集了DNS日志，在合并处理后采用LSTM进行检测；Niu等人[16]针对移动端的DNS日志进行C2域名检测.

攻击关联性分析检测更加注重对APT攻击之间关联性的分析与建模.Bahrami等人[17]利用杀伤链模型对APT攻击场景进行建模，该模型将APT攻击分解为40多项子活动，并确定了APT攻击中的行为特征，进而进行攻击检测；Kim等人[18]对杀伤链模型进行了改进与细化，主要用于对IOT网络的APT攻击进行检测；Zhou等人[19]通过对移动目标进行防护来处理APT攻击中的路径突变问题；Jasiul等人[20]和杜镇宇等人[21]设计了不同的基于Petri网的攻击检测模型，文献[20]使用主机内的系统特征与文件特征生成了有色Petri网，利用其对恶意软件中的恶意行为进行建模，文献[21]中的模型通过匹配攻击路径，并根据收集到的报警信息对攻击行为进行预测；Ghafir等人[22]将属于一个完整APT攻击的不同子攻击活动的检测结果进行关联，通过HMM对其解码，确定最有可能的攻击序列；Niu等人[23]使用动态步骤图对APT攻击进行映射，建立网络攻击模型捕获APT攻击因素；孙文新[24]提出了因果场景生成算法，将相关流量以及对应的攻击步骤匹配至杀伤链模型中，发掘流量数据之间的相关性.

通过对以上APT攻击检测方法的研究发现，目前的检测方法存在若干问题，包括对APT攻击中单个攻击之间的时序性结合关注度不高；针对Linux系统中恶意软件的检测与行为捕获工具较少，且已有工具效果较差；由于APT攻击周期过长导致业内对于APT攻击的高质量数据集较少等.

2 Linux系统下APT攻击原始数据集的构建

高质量的APT攻击数据集是构建APT攻击检测模型的关键.本文构建的原始数据集融合了APT攻击的主机侧与网络侧双侧特征，为后续生成APT攻击数据集提供了良好基础.构建Linux

主机攻击行为捕获沙箱 LAnalysis，利用其分析 Linux ELF 恶意文件生成主机侧初始数据集，对网络公开数据集 NSL-KDD 进行处理，生成网络侧初始数据集.

2.1 沙箱 LAnalysis

目前常用的开源沙箱以及众多商用和在线文件分析平台针对 Linux ELF 文件的恶意代码样本分析能力较弱，以致无法获取恶意样本的全部恶意行为.本文构建了一款能力较强的分析 Linux ELF 文件的沙箱 LAnalysis，针对 APT 攻击中使用的恶意代码进行分析，进而构建 APT 攻击中的主机行为数据集.

LAnalysis 为 C/S 架构，分为服务端（监控端）与客户端（被监控端）2 部分.检测方式为服务端将样本发送至客户端，对目标样本进行检测，检测完后将检测报告进行回传，服务端将检测报告进行分类处理，形成不同类恶意行为的特征文件.系统结构如图 1 所示：

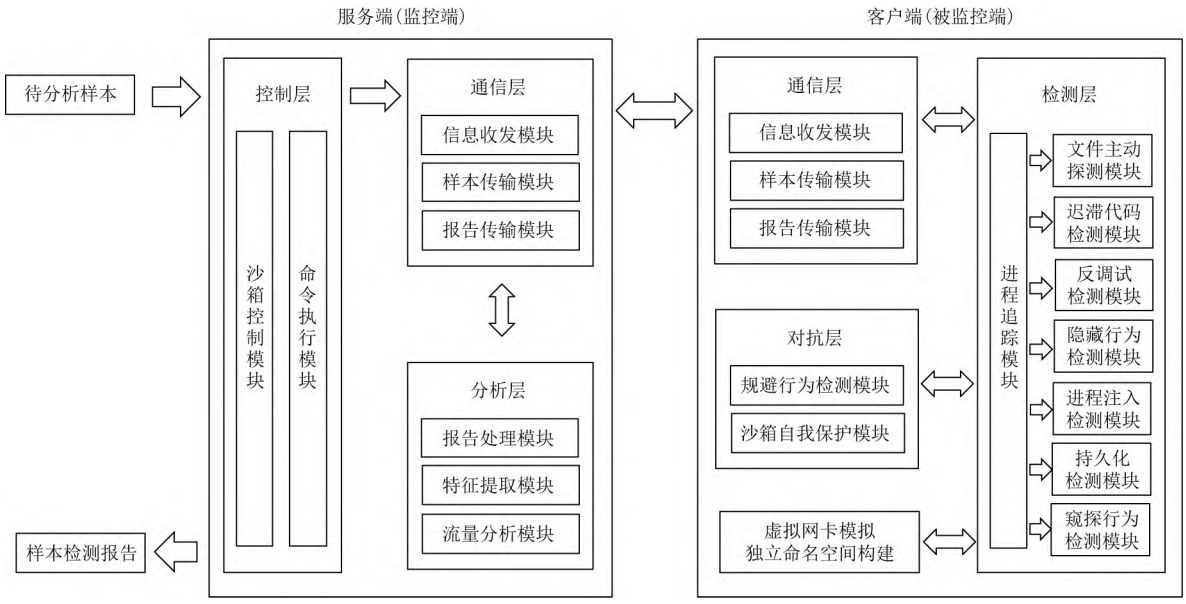

图 1 LAnalysis 架构

服务端分为通信层、分析层、控制层共 3 层.控制层依据用户传递过来的指令启动沙箱工作，并且通过命令执行的方式对沙箱进行控制；通信层负责收集和处理另一端发来的消息并进行下一步流程的推进；分析层对客户端发来的分析样本的行为报告以及流量进行处理与分析，进一步提取出该样本的恶意行为特征.

客户端由一个沙箱虚拟机组成，沙箱分为通信层、检测层与对抗层共 3 层.检测层负责部署沙箱检测模块，对目标样本进行行为分析；通信层负责与服务端的沟通并将分析结果回传给服务端；对抗层负责部署沙箱自我保护模块与规避行为检测模块，对恶意样本进行检测并与破坏沙箱的行为进行对抗.

LAnalysis 获取恶意代码的 10 类共 16 种不同的恶意行为，在检测与分析 APT 攻击中部署的恶意 Linux ELF 文件的恶意行为上表现更为出色.

2.2 Linux ELF 恶意文件行为捕获与检测

LAnalysis 对恶意代码的 10 类恶意行为进行捕获，用于构建主机行为数据集.具体为反调试行为、迟滞代码行为、持久化行为、文件隐藏行为、网络隐藏行为、进程隐藏行为、网络行为、权限提升行为、进程注入行为、对系统的窥探行为.捕获行为的选择来自攻击框架 ATT&CK 中常用的攻击行为.

恶意行为的捕获与检测主要依赖于 LAnalysis 的检测层，它运行在沙箱内部，部署沙箱监控的各个模块以检测目标样本的各类恶意行为，是整个沙箱系统的核心检测层，由 SystemTap 编写的 8 个恶意行为检测模块构成，分别为文件主动探测

模块、迟滞代码检测模块、反调试检测模块、隐藏行为检测模块、进程注入检测模块、持久化检测模块、窥探行为检测模块、进程追踪模块. 除此之外还有为了防止沙箱被破坏所构建的沙箱自我保护模块.另外,还有一些恶意行为的检测不是通过SystemTap部署内核探针完成的,而是通过恶意代码运行前后系统发生的变化进行检测.

2.3 APT攻击数据集构建

在APT攻击中,主机攻击指攻击者通过各种方式入侵受害者的主机系统,在主机系统中植入恶意代码,进而产生对受害者主机的控制与破坏行为.网络侧攻击指在进入主机前在网络空间中进行的流量攻击或建立C2通道远程操控进行攻击的手段.

2.3.1 APT攻击主机行为数据集的构建

通过对大量不同恶意家族中恶意样本的分析,获取其在Linux主机上产生的恶意行为,将其构建为APT攻击的主机行为数据集.通过对无害良性样本进行分析,提取与恶意样本同样的特征,作为正常主机行为特征,为构建非APT攻击数据集提供原数据基础.另外,为了方便生成APT攻击样本数据,需要将主机行为数据集进行标签化处理.

2.3.2 APT攻击流量数据集的处理

本文选用的流量数据集为NSL-KDD. NSL-KDD是入侵检测领域的一个经典数据集,其每条数据均由41种特征组合而成,每个网络连接被标记为normal或attack.本文使用NSL-KDD作为APT攻击网络攻击数据集的原始数据集,其中attack表示攻击数据,共有4大类,这4大类又被细分为22种不同的攻击,共有125 973条数据.为了适应深度学习模型的训练,通过one_hot对数据进行编码.对数据集进行维度处理后,还需要将数据和标签进行分离,同时将同一类标签的数据进行归类,以生成APT攻击流量数据集.

3 基于LSTM的APT攻击检测方案

3.1 总体设计

APT攻击检测方案分为3个部分,分别为原始数据集生成模块、APT攻击数据集生成模块、模型构建模块.总体设计图如图2所示.

1) 原始数据集生成模块.

在原始数据集生成模块中,构建了沙箱LAnalysis并对采集的恶意样本与良性样本进行分析,形成APT攻击的主机行为数据集;同时使用NSL-KDD作为APT攻击的网络流量初始数据集.

2) APT攻击数据集生成模块.

APT攻击数据集生成模块用于将网络行为数据集与主机行为数据集根据APT攻击的攻击流程进行合并重构,将各种单独的攻击方式组合成具有前后上下文关联的APT攻击,生成APT攻击数据集.同时,还需要生成非APT攻击数据集作为深度学习模型训练的负样本.

3) LSTM模型构建模块.

利用生成的APT攻击数据集与非APT攻击数据集构建基于LSTM的APT攻击检测模型.

3.2 APT攻击数据的生成

在APT攻击数据生成前首先要针对APT攻击过程进行攻击步骤拆分与建模,之后使用网络行为数据集与主机行为数据集中的子攻击标签组成APT攻击标签序列,形成符合APT攻击生命周期的攻击数据.

3.2.1 攻击过程建模

根据Hutchins等人[25]提出的网络攻击杀伤链分析模型(Cyber Kill Chain)可知,1次APT攻击分多个具体的攻击步骤.因此将网络侧行为数据与主机侧行为数据依据APT攻击过程进行建模是生成APT攻击数据的关键.由于APT攻击方式多种多样,每次攻击都可能有不同的战术变化,本文只选用最经典的3种APT攻击方式进行模拟.

这3种最经典的APT攻击方式是:钓鱼攻击、利用Web漏洞与操作系统漏洞入侵、利用线下移动设备入侵.3种攻击方式都包含相似的必要攻击步骤,但是在感染主机和感染后的行为上略有不同.例如,钓鱼攻击通常没有针对Web的攻击行为,但会有可疑的网络流量出现,若被攻击者成功执行钓鱼程序,则其可以直接以被攻击者的权限运行,有时不需要进一步提升权限.利用Web漏洞与操作系统漏洞的攻击方式往往有较多的网络攻击流量,且攻入系统后有权限提升行为.利用线下移动设备入侵没有任何网络攻击流量,直接利用U盘、移动硬盘等通过植入恶意代码进行攻击,

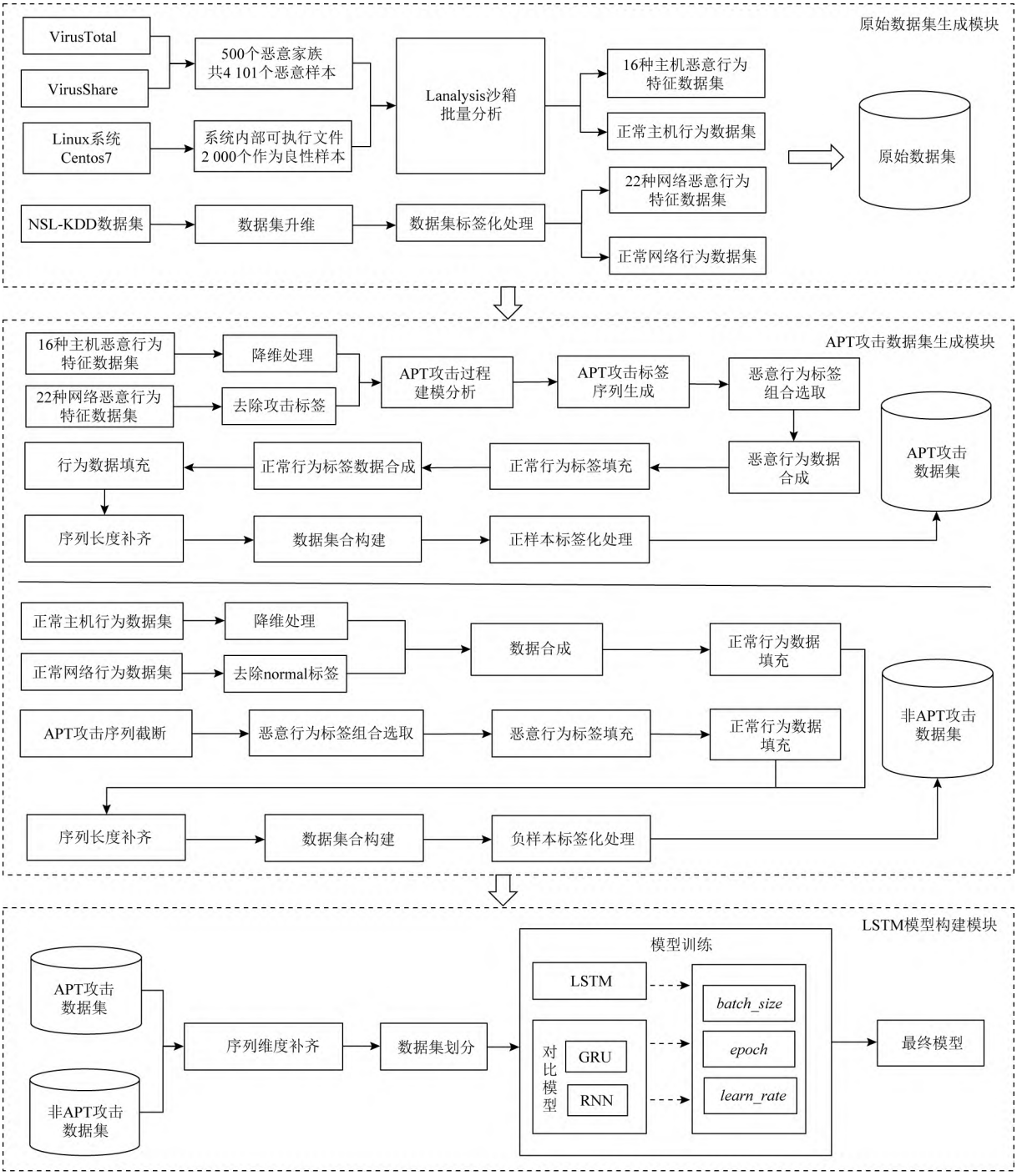

图 2　APT 攻击检测方案总体设计图

但是在入侵后具有与控制服务器交互的行为，会产生网络流量.

1）通过网页或邮件等进行钓鱼攻击.

由于被攻击者安全意识不强，误执行了攻击者发送的邮件或网页链接的不可信内容，以致主机被感染后遭受到一系列攻击行为.该种攻击方式的攻击过程如表 1 所示.

2）利用 Web 漏洞与操作系统漏洞进行入侵.

利用运行的 Web 漏洞或开放的高危端口对应的系统漏洞，通过网络入侵进入主机后，进一步部署恶意代码进行针对主机的攻击.该种攻击方式的攻击过程如表 2 所示.

表1 钓鱼攻击流程

攻击步骤	攻击操作
1	扫描目标IP以及开放的端口
2	初始探测攻击
3	对被攻击者进行钓鱼攻击
4	成功部署恶意软件
5	恶意软件利用root进程
6	获取被攻击者账户
7	恶意软件针对主机进行持久化攻击
8	隐藏攻击者信息
9	横向移动与DoS攻击

表2 利用Web漏洞与操作系统漏洞进行入侵

攻击步骤	攻击操作
1	对被攻击者的IP地址和端口进行扫描并监视
2	探测运行的Web服务和高危系统服务
3	利用Web漏洞或高危端口入侵上传Webshell或木马等
4	攻击者进行提权操作
5	下载并安装恶意软件
6	恶意软件窥探攻击
7	恶意软件对主机进行持久化攻击
8	隐藏攻击者信息
9	横向移动与DoS攻击

3）利用线下移动设备入侵.

以便携式的移动设备进行攻击，如移动硬盘、U盘、手机等设备接入主机，可以直接进行植入式攻击.该种攻击方式的攻击过程如表3所示：

表3 利用线下移动设备入侵

攻击步骤	攻击操作
1	恶意软件线下植入
2	恶意软件窥探攻击
3	恶意软件部署
4	攻击者获得本地账户及密码
5	权限提升
6	恶意软件对主机进行持久化攻击
7	隐藏攻击者信息
8	横向移动与DoS攻击

3.2.2 攻击序列标签生成

在对APT攻击行为进行建模后，使用网络行为数据标签与主机行为数据标签针对这3类攻击的攻击过程进行对应，形成APT攻击的标签序列可以通过变更标签中的不同样本，生成大量不同的APT攻击样本.表4～6分别为3种APT攻击方式对应的攻击标签.

在生成APT攻击数据时，由于真正的攻击步骤在时间上不可能是完全连续的，各个步骤之间会有不可估计的时间间隔，因此需要在表4～6中的步骤之间穿插一系列的非攻击操作.在实际生成APT攻击数据集时，在表4～6中的各个步骤间，选择在随机位置插入随机数量的标签为normal的数据，表示网络行为中的正常行为与主机行为中的正常行为.同时每个标签都对应较多不同样本，保证了APT攻击数据集中数据的丰富性.

表4 钓鱼攻击流程的攻击标签

攻击步骤	网络行为数据标签	主机行为数据标签
1	ipsweep,portsweep	normal
2	perl,nmap,satan,spy	normal
3	ftp_write,guess_passwd	Endurance_file&code
4	waremaster,multihop	Endurance_file&code,Endurance_rootkit
5	loadmodule,buffer_overflow,imap	privilege,Endurance_shell
6	imap,guess_passwd	Endurance_passwd
7	rootkit,phf,warzeclient	Endurance_file&code,Endurance_rootkit,Endurance_passwd,Endurance_shell
8	rookit	process_hide,network_hide,file_hide,process_inject
9	back,land,pod,smurf,teardrop,neptune,ipsweep	spy_process,spy_syshard,spy_syssoft,net_invasion,net_dos

表 5　利用 Web 漏洞与操作系统漏洞进行入侵的攻击标签

攻击步骤	网络行为数据标签	主机行为数据标签
1	ipsweep,portsweep	normal
2	nmap,portsweep,phf	normal
3	guess_passwd,imap,satan,multihop	Endurance_file&code
4	loadmodule,buffer_overflow,imap	privilege
5	warezmaster	Endurance_file&code,Endurance_rootkit
6	rootkit,spy	anti_debug,anti_time,spy_process,spy_syshard,spy_syssoft
7	rootkit,phf,warzeclient	Endurance_file&code,Endurance_rootkit,Endurance_passwd,Endurance_shell
8	spy,multihop	process_hide,network_hide,file_hide,process_inject
9	back,land,pod,smurf,teardrop,neptune,ipsweep	spy_process,spy_syshard,spy_syssoft,net_invasion,net_dos

表 6　利用线下移动设备入侵的攻击标签

攻击步骤	网络行为数据标签	主机行为数据标签
1	normal	Endurance_file&code
2	ipsweep,portsweep,spy,multihop	anti_debug,anti_time,spy_process,spy_syshard
3	warezclient	Endurance_file&code,Endurance_rootkit
4	guess_passwd,imap	Endurance_passwd
5	loadmodule,buffer_overflow,imap	privilege
6	rootkit,phf,warezclient	Endurance_file&code,Endurance_rootkit,Endurance_passwd,Endurance_shell
7	rootkit	process_hide,network_hide,file_hide,process_inject
8	back,land,pod,smurf,teardrop,neptune,ipsweep	spy_process,spy_syshard,spy_syssoft,net_invasion,net_dos

3.2.3　依据标签生成 APT 攻击数据

在生成的 APT 攻击数据中每个步骤都会有主机行为和网络行为与之对应，APT 攻击的每个时间步的数据由网络行为与主机行为合并而成，表示当前步骤中的网络状态与主机状态.特征处理方面，表示网络状态的特征为 122 维，即 1 个 NSL-KDD 数据标签表示当前的网络情况.主机行为特征由 1 个或多个当前步骤中的不同主机攻击行为随机组合而成.由于每种不同的主机攻击行为在 89 维中都有自己固定的位置，因此组合后不会造成互相的覆盖，而是会将各自的特征保留下来，没有融合进来的其余位置都用 0 进行补齐，表示当前步骤中没有类似的行为.因此 1 个或多个主机攻击行为标签最终会组合成 89 维的主机行为特征.1 个步骤有网络行为数据集的 122 维加上主机行为数据集的 89 维共 211 维特征.而 1 条 APT 攻击数据就是由多个时间步的行为按照攻击方式对应的标签顺序排列而成.

1 条 APT 攻击数据的生成过程如图 3 所示，每条数据都由若干个时间步构成，表示每次攻击

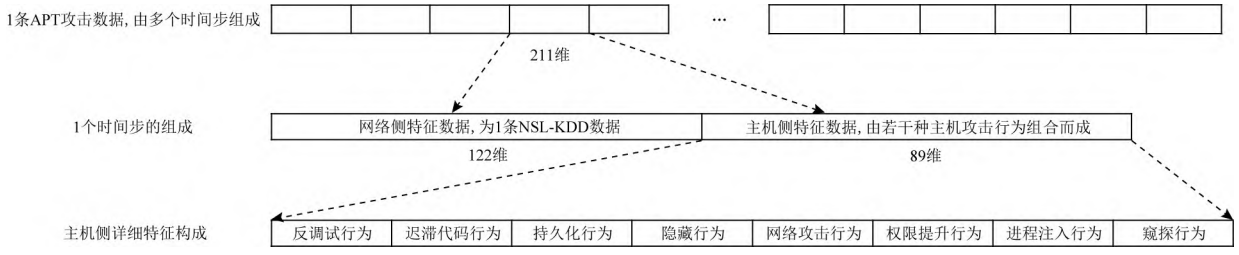

图 3　1 条 APT 攻击数据构成

中的攻击步骤，而每个时间步有 211 维特征，由网络侧特征与主机侧特征组合而成.利用线下移动设备入侵行为中的第 7 个攻击步骤的 1 个时间步构成如图 4 所示，其网络行为数据标签为 rootkit，主机侧特征由恶意行为特征集 file_hide 和 process_inject 中随机选取 1 种组合而成.

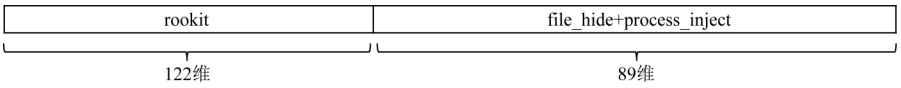

图 4　1 个时间步构成示例

3.3　非 APT 攻击数据的生成

APT 攻击数据集在最终模型的数据集中使用标签 1 进行对应，表示正样本集，还需生成非 APT 攻击数据作为负样本集，用标签 0 与之对应.网络特征部分使用 NSL-KDD 中标签 normal 的样本，总共为 122 维特征.主机行为的正常行为样本在创建时就没有按攻击行为进行拆分，整个 89 维的标签 normal 的样本就是当前时刻正常的主机行为，因此从中随机选取进行利用即可.将主机行为数据和网络行为数据进行组合形成 211 维的当前步骤下的单步非 APT 攻击数据.

使用单步非 APT 攻击数据生成 1 条完整的非 APT 攻击数据的方式共有 2 种：第 1 种由标签为 normal 的网络单步数据与主机单步数据组成，这样得到的数据为完全正常的行为序列；第 2 种是在 APT 攻击的 3 种入侵方式中只生成某步骤之前的攻击序列，代表攻击的中止与失败，该条数据也为非 APT 攻击数据，为了与攻击数据有一定的区分度，截取的步骤为整体步骤的前 2/3 处，不会从接近攻击结束的地方进行截取.

3.4　基于 LSTM 的 APT 攻击检测模型构建

基于构建出的 APT 攻击数据集与非 APT 攻击数据集，使用 LSTM 进行训练，训练过程分为以下几个步骤：数据集划分与处理、模型初步训练、调参优化.生成参数最优的 APT 攻击检测模型.整体流程如图 5 所示：

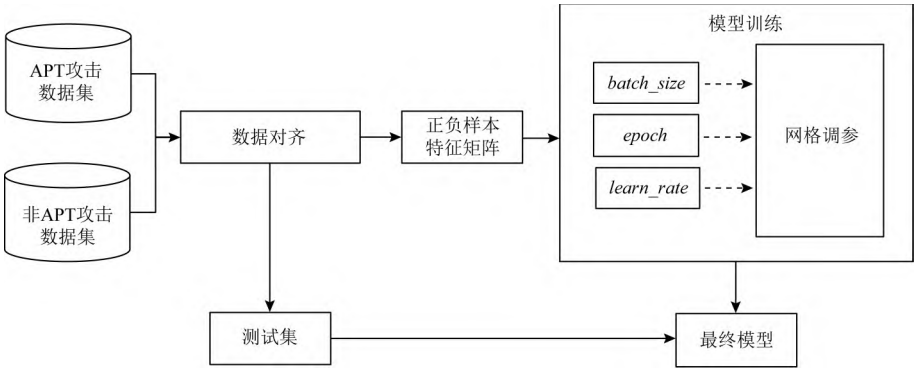

图 5　APT 攻击检测模型训练过程

1）数据集划分与处理.

首先对输入模型的数据进行时序划分处理，使用 LSTM 进行模型训练时，3 个较为关键的可以体现时序特征的参数为单个数据维度、时间步长度与最大数据长度.本文单个数据维度为 122＋89＝211 维，表示当前步骤的网络状态与主机状态，最长数据由 40 个时间步构成，因此最大数据长度为 40×211＝8 440.为了输入数据的长短一致，将所有数据的长度补齐至 8 440.

本文共生成 16 000 条 APT 攻击数据与 16 000 条非 APT 攻击数据，将训练集与测试集按照 3∶1 进行划分，则训练集共有 12 000 条正样本数据与 12 000 条负样本数据，测试集共有 4 000 条正样本数据与 4 000 条负样本数据.

2）模型的调参优化.

针对 LSTM 的 *learn_rate*，*batch_size* 以及 *epoch* 这 3 个参数进行优化，比较取何值时模型的表现最好.*batch_size* 是单批次训练数据的大小，影响训练时间与训练速度.*epoch* 为训练的轮数，值越大训练的时间就越长，随着 *epoch* 的增大损

失函数会逐渐收敛，此时模型趋于稳定，因此需要找到合适的 *epoch* 值.*learn_rate* 是 1 个相对重要的超参数，其取值会影响模型最终的准确率.

3）模型评价标准.

APT 攻击检测是一个二分类问题，模型的预测出的结果只有 1 和 0 这 2 种取值.本文采用准确率、精确率、召回率和 *F*1 值这 4 个深度学习与机器学习中较为常用的指标衡量模型的优劣程度.

4 测试与结果分析

4.1 测试环境

4.1.1 LAnalysis 实现环境

LAnalysis 分为服务端与客户端，都是基于 Python3 进行开发，同时还使用了 shell 脚本、SystemTap 脚本进行相关功能的实现.服务端与客户端采用虚拟机嵌套的方式运行，服务端使用主机系统中的 VMware 创建，客户端使用服务端系统中的 VirtualBox 创建.具体环境参数如表 7 所示：

表 7 LAnalysis 环境参数

沙箱结构	参数名称	具体参数
服务端	操作系统	Linux Centos7 64bit
	内存/GB	10
	CPU	6 核
	软件版本	VirtualBox 6.1.16
	开发语言	Python3.6.8
客户端	操作系统	Linux Centos7 64bit
	内存/GB	4
	CPU	2 核
	开发语言	Python3.6.8，Shell，SystemTap
服务端虚拟机主机	操作系统	Macos Mojave 10.14.6
	处理器	2.6 GHz Intel Core i7
	内存信息	16 GB 2400 MHz DDR4
	软件版本	VMWare Fusion 10.1.6，Pycharm

4.1.2 LSTM 构建环境

LSTM 的构建环境为 Linux 操作系统，主要使用 Python 进行开发.神经网络模型开发框架使用开源的 Keras 框架.具体环境参数如表 8 所示.

4.2 LAnalysis 行为检测效果与分析

将本文自主构建的沙箱 LAnalysis 与开源沙箱 Cuckoo、业内较常用的微步云沙箱和腾讯哈勃沙箱以及奇安信文件分析平台进行对比，从 Virustotal 与 VirusShare 2019—2020 ELF 样本集中随机选取 20 个恶意样本，对比统计分析出的恶意行为种类及个数.

表 8 LSTM 环境参数

参数名称	具体参数
操作系统	Macos Mojave 10.14.6
处理器	2.6 GHz Intel Core i7
内存信息	16 GB 2400 MHz DDR4
开发语言	Python3.6.8，Shell
开发框架	Keras2.2.5

20 个恶意样本的用例编号如表 9 所示，根据恶意样本用例编号可以直接在 Virustotal 和 VirusShare 中找到该恶意样本，这是每个恶意样本的唯一性标识.

由于 20 个恶意样本并不能完全覆盖 10 类的 16 种恶意行为，因此将恶意行为概括为对抗行为、隐藏行为、持久化行为、窥探行为以及其他恶意行为 5 种.由于 20 个样本一一列举篇幅占用过多，因此只列出其中 2 个样本的详细对比结果与 20 个样本的总体分析结果.

1）0dbcc464a0dc0463bc9969f755e853d8.

该样本为盖茨家族的恶意样本，各沙箱分析结果如表 10 所示.

2）0A9BBC90CAB339F37D5BDD0B906F1A9C.

该样本为 Skeeyah 家族的恶意样本，各沙箱分析结果如表 11 所示.

3）20 个样本.

各沙箱所检测出来的 20 个恶意样本的恶意行为总量如表 12 所示.

由对比结果可知，虽然 Cuckoo、腾讯哈勃、奇安信文件分析平台以及微步在 Windows 文件分析上较为成熟，在业内也较为常用，但是对 Linux ELF 文件的分析能力十分薄弱.

以上结果证明了本文构建 Linux ELF 文件分析沙箱的合理性，同时也证明了 LAnalysis 在分析 Linux ELF 文件上的功能十分强大.LAnalysis 较好地分析了众多恶意家族的恶意样本，因此基于 LAnalysis 构建的 APT 攻击主机行为数据集较为全面，利用这些数据构成的 APT 攻击样本的合理性与丰富性也因此得到了保障.

表 9　20 个恶意样本的用例编号

序号	用例编号
1	0dbcc464a0dc0463bc9969f755e853d8
2	VirusShare_320adee47e53823albe8a335e4beb246
3	0e40acd896e32f404c207c5836a0b514f316b4dc68cead03a3cc4b740b812ddc
4	VirusShare_7e21d4fbd6d6ef12deb0f8abeeaa942d
5	VirusShare_7ccd91eble2e56dde3d1072c8ae200c4
6	VirusShare_7bca36debd5bf1bcc4493637705348d3
7	VirusShare_6d011884c22644ea7ded66d15e457dcc
8	VirusShare_06cb62ecc781040b7df6d388f1f94408
9	VirusShare_5d726a5c571f5d7289b5ab19e58e7005
10	0A9BBC90CAB339F37D5BDD0B906F1A9C
11	1fa323999c7a973e27c4857b4756211d8d80fb7f551a55e0ceb0ed12c6e60874
12	4a2c13c9cb54652980e1b81aea74e50bc00fd97e03de78cc94a843e51d9dd78d
13	4f591948f1701b7d656ed0e03af82588497c66a82c8ae7 bebdb23850c9fde247
14	6cae84657a7b399f4f008bd7022bdd43d0a45e970c9642c7436677db181c925
15	17679226b5f0d33eef6f0841cca9c36d
16	877592648db7fc47b448a9eb1d5b8289747bb946254569aef3a79a37f5b4e1f4
17	6fae784d131090e7c7ae75b646d39e3c920eef10363a73207deab8d21ef40940
18	8b9bd0b7b43ce0cf829c9e50cbf47a6107e30f27209c2713daa3c0749d9b0eee
19	VirusShare_5f9164ec0d758de70bee4aad2f9c08a7
20	VirusShare_5e488aa38ebdb176d11fdbfa4291982f

表 10　0dbcc464a0dc0463bc9969f755e853d8 的沙箱分析结果对比

沙箱	对抗行为	隐藏行为	持久化行为	窥探行为	其他恶意行为
LAnalysis（本文）	反调试行为、迟滞代码行为 2 项	进程隐藏行为、文件隐藏行为 2 项	代码驻留与后门部署 8 项	proc 目录探测、配置文件探测、虚拟机探测 3 项	可疑 IP 与端口访问 2 项
奇安信文件分析平台	无	无	代码驻留与后门部署 1 项	配置文件探测 1 项	可疑 IP 与端口访问 2 项
腾讯哈勃	无	无	代码驻留与后门部署 1 项	无	可疑 IP 访问 1 项
微步	无	无	无	无	无
Cuckoo	无	无	无	无	无

表 11　0A9BBC90CAB339F37D5BDD0B906F1A9C 的沙箱分析结果对比

沙箱	对抗行为	隐藏行为	持久化行为	窥探行为	其他恶意行为
LAnalysis（本文）	反调试行为、迟滞代码行为、敏感系统命令 3 项	文件隐藏行为 1 项	代码驻留与后门部署 4 项	proc 目录探测、配置文件探测、sys 目录探测 3 项	可疑 IP 与端口访问 2 项、权限提升行为 2 项
奇安信文件分析平台	敏感系统命令 1 项	无	无	无	可疑 IP 与端口访问 2 项、权限提升行为 1 项
腾讯哈勃	无	无	代码驻留与后门部署 3 项	无	无
微步	无	无	无	无	无
Cuckoo	无	无	无	无	无

表 12　20 个恶意样本的沙箱分析结果对比　　项

沙箱	对抗行为	隐藏行为	持久化行为	窥探行为	其他恶意行为
LAnalysis(本文)	41	18	91	49	40
奇安信文件分析平台	5	2	5	3	19
腾讯哈勃	0	0	5	0	4
微步	0	0	1	0	0
Cuckoo	0	0	0	0	0

4.3　模型检测效果实验及对比分析

4.3.1　LSTM 模型训练与优化

将 3.4 节所给出的训练集与测试集的正样本与负样本形成 2 维数组输入模型，标签为 x_train，y_train 与 x_test，y_test.

首先设置 $batch\_size=32$，$learn\_rate=0.0001$，$epoch=10$.将初始参数设定为较小的值可以提高调参优化的效率.此时的评价指标如表 13 所示：

表 13　初始状态的评价指标

准确率	精确率	召回率	$F1$
0.8639	0.8840	0.8533	0.8684

1）改变 $batch\_size$.

$batch\_size$ 是训练 1 次所使用数据量的大小，对训练速度与时间具有较大影响，因此先对 $batch\_size$ 进行调整有利于为后续 $epoch$ 和 $learn\_rate$ 的调整节省时间.$batch\_size$ 受总内存值的影响，取值从 32 开始翻倍增长，当 $batch\_size$ 增大到 256 时程序会明显变慢，出现卡顿，因此 $batch\_size$ 最合适的大小应设置为 128.后续的调参过程中 $batch\_size$ 均设定为 128，此时训练速度是最快的.

2）改变 $epoch$.

$epoch$ 为训练的轮数，初始状态 $epoch=10$，此时损失函数的变化曲线如图 6 所示.通过图 6 可知损失函数尚未收敛，说明模型仍在优化的过程中，因此应该增大 $epoch$ 的值进行再次训练.

当 $epoch=50$ 时，损失函数变化曲线如图 7 所示，此时可以看到模型随着训练轮数的增多逐渐收敛，当 $epoch=40$ 时可以看到模型基本收敛，因此后续的调参过程将 $epoch$ 设定为 40.

3）改变 $learn\_rate$.

通常 $learn\_rate$ 的取值范围为[0.0001,1]，但是若直接取值，大概率会选择到[0.1,1]的范围

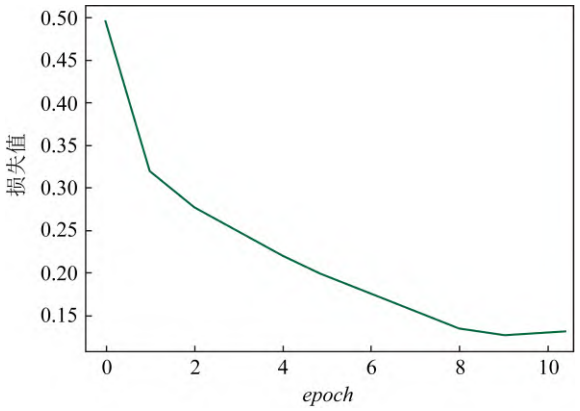

图 6　$epoch=10$ 时的损失函数曲线

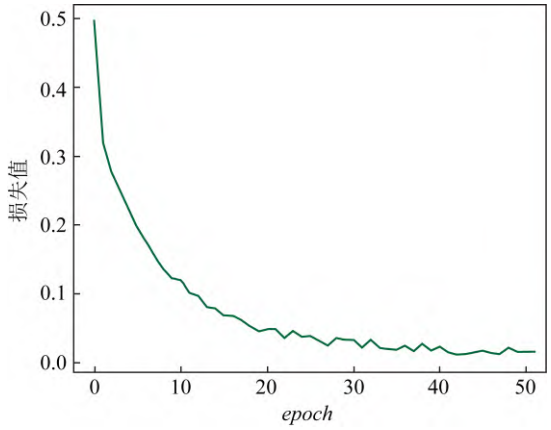

图 7　$epoch=50$ 时的损失函数曲线

内，此时 $learn\_rate$ 的值难以在较大范围内变化.因此采用对数取值的方式，先选定 4 个区间，即[0.0001,0.001]，[0.001,0.01]，[0.01,0.1]，[0.1,1]，在这 4 个区间内再进行平均取值.本文在 $batch\_size=128$，$epoch=40$ 的情况下不断改变 $learn\_rate$ 进行训练，不同 $learn\_rate$ 下的评价指标对比情况如表 14 所示.

由表 14 可知，$learn\_rate=0.002$ 时模型的 $F1$ 值是最好的，因此模型的最终指标于 $batch\_size=128$，$epoch=40$，$learn\_rate=0.002$ 时取

得，如表 15 所示.

表 14　不同 *learn_rate* 下的评价指标对比情况

learn_rate	准确率	精确率	召回率	*F*1
0.0001	0.918 6	0.926 3	0.921 7	0.924 0
0.001	0.925 2	0.920 2	0.931 3	0.925 7
0.002	0.921 7	0.916 5	0.937 1	0.926 7
0.01	0.906 0	0.899 4	0.938 4	0.918 5
0.02	0.909 5	0.897 0	0.939 3	0.917 7
0.1	0.704 9	0.708 7	0.718 9	0.713 8
0.2	0.605 8	0.616 8	0.596 2	0.606 3

表 15　模型最终指标

准确率	精确率	召回率	*F*1
0.921 7	0.916 5	0.937 1	0.926 7

4.3.2　不同检测模型和方案的对比与分析

1）不同时序处理模型的比较.

本文方案的模型准确率为 92.17%，*F*1 值为 0.926 7.为确定最优的时序处理模型，训练了 RNN 模型、GRU 模型与 LSTM 模型进行对比，实验结果如图 8 所示.从图 8 可以看出 LSTM 在处理 APT 攻击时序数据时效果要好于 RNN 与 GRU 模型.

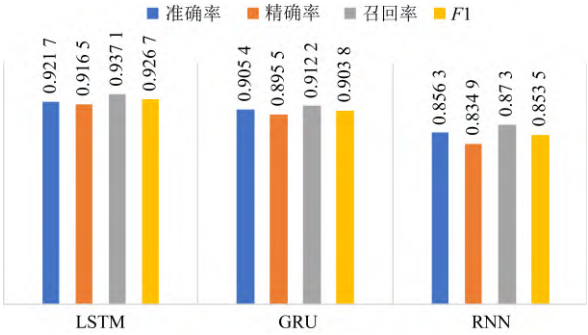

图 8　不同时序处理模型的比对情况

2）不同 APT 攻击检测方案的比较与分析.

目前业内有众多利用机器学习与深度学习进行 APT 攻击检测的方法.本文选择了较有代表性的 4 种[9,14,22,26]与本文构建的模型从模型准确率、特征全面度以及 APT 攻击建模等方面进行比较，结果如表 16 所示：

表 16　APT 攻击检测模型对比

检测模型	网络侧特征	网络侧特征来源	主机侧特征	主机侧特征来源	时序性建模	模型准确率
本文	122 维-23 种标签	NSL-KDD	89 维-16 种标签	4 101 个恶意代码-自建沙箱分析	是	0.921 7
文献[22]	8 种标签	自主模拟生成网络告警 4 900 条	无	无	是	0.848 0
文献[9]	无	无	39 种标签	3 133 个恶意代码-沙箱 Cuckoo 分析	否	0.922 0
文献[14]	122 维-5 种标签	NSL-KDD	无	无	否	0.931 7
文献[26]	122 维-5 种标签	KDD99	无	无	是	0.849 6

由表 16 可知，对比的 4 种方法没有完全兼顾网络侧特征、主机侧特征以及 APT 攻击的时序性建模这 3 个 APT 攻击检测的重要方面，模型都存在不足之处.

文献[22]使用自主构建的网络告警数据进行模型训练，没有在大量网络数据集中进行验证与复现，具有一定的片面性.此外，虽然其考虑到了 APT 攻击中对子攻击的时序性建模，但是没有将主机侧的行为数据放入模型，对 APT 攻击的检测具有片面性与不合理性.

文献[9]使用沙箱 Cuckoo 对恶意代码进行分析，由 4.2 节可知 Cuckoo 对恶意代码的分析能力并不出色，因此基于 Cuckoo 进行恶意样本分析以检测 APT 攻击具有很大的提升空间.另外文献[9]并没有考虑 APT 攻击的时序性，仍局限于单种恶意代码的检测，并不能作为有效的 APT 攻击检测模型.

文献[14]在数据处理时没有使用 NSL-KDD 中的恶意行为细分标签，而是将攻击行为局限于大类，导致对攻击的刻画不够具体.另外其没有将 APT 攻击与普通的入侵检测进行区别，仍是针对单种攻击进行检测，没有将 APT 攻击的时序性特征考虑在内.虽然其准确率为 93.7%，但却不能视为较好的 APT 攻击检测效果.

文献[26]在实验中使用的数据集为 KDD99，该数据集相较于 NSL-KDD 有明显不足，存在大量冗余数据，目前已很少被业内使用.同时文献[26]也没有使用 KDD99 中更为细分的攻击小类，而是使用大类进行检测，对攻击的刻画不够具体.另外，文献[26]没有使用任何主机数据构建 APT 攻击数据集，仅用 KDD99 中包含的网络侧特征数据，而仅依据 KDD99 构建的 APT 攻击数据集与 APT 攻击检测模型是不合理的.

5 总　　结

针对业内检测方案对 APT 攻击中单个攻击之间的时序性结合关注度不高、对于 Linux 系统中的恶意代码攻击检测工具较少、APT 攻击周期过长导致 APT 攻击研究样本较少等问题，本文提出一种基于 LSTM 的 Linux 系统下 APT 攻击检测方案，并与其他方案进行了对比.实验证明，本文方案能够较好地对 APT 攻击进行检测，并且能够构建一套兼具主机行为和网络行为特征的 APT 攻击数据集，较好解决了当前业内缺乏高质量的 APT 攻击数据集的问题，为后续研究工作打下了良好的基础.

参 考 文 献

[1] 恒安嘉新(北京)科技股份公司. 2020 年网络安全态势报告[J]. 信息安全研究，2021，7(3)：198—206

[2] 张博，崔佳巍，屈肃，等. 高级持续性威胁及其重构研究进展与挑战[J]. 信息安全研究，2021，7(6)：512—519

[3] 360TI 威胁情报中心. 2021 年上半年全球高级持续性威胁(APT)研究报告[EB/OL]. 2022 [2022-04-14]. http://pub-shbt.s3.360.cn/cert-public-file/2021APT-1015.pdf

[4] Bist A S，Jalal S. Identification of metamorphic viruses [C] // Proc of the 2014 IEEE Int Advance Computing Conference (IACC). Piscataway，NJ：IEEE，2014：1163—1168

[5] Stojanović B，Hofer-Schmitz K，Kleb U. APT datasets and attack modeling for automated detection methods：A review [J]. Computers & Security，2020，92：101734

[6] 冯学伟，王东霞，黄敏桓，等. 一种基于马尔可夫性质的因果知识挖掘方法[J]. 计算机研究与发展，2014，51(11)：2493—2504

[7] 霍彦宇. 基于杀伤链和模糊聚类的 APT 攻击场景生成方法的研究与设计[D]. 北京：北京邮电大学，2018

[8] Sharma P，Joshi A，Finin T. Detecting data exfiltration by integrating information across layers [C] //Proc of the 14th IEEE Int Conf on Information Reuse & Integration (IRI). Piscataway，NJ：IEEE，2013：309—316

[9] Moon D，Lee H，Kim I. Host based feature description method for detecting APT attack [J]. Journal of the Korea Institute of Information Security & Cryptology，2014，24(5)：839—850

[10] 孙增，施勇，薛质. 基于沙箱回避的 APT 研究[J]. 信息安全与通信保密，2015 (3)：92—96

[11] Marchetti M，Pierazzi F，Guido A，et al. Countering advanced persistent threats through security intelligence and big data analytics [C] //Proc of the 8th Int Conf on Cyber Conflict (CyCon). Piscataway，NJ：IEEE，2016：243—261

[12] 戴震，程光. 基于通信特征的 APT 攻击检测方法[J]. 计算机工程与应用，2017，53(18)：77—83

[13] Chuan B L J，Singh M M，Shariff A R M. APTGuard：Advanced persistent threat (APT) detections and predictions using Android smartphone [G] //LNEE 481：Computational Science and Technology. Singapore：Springer，2018：545—555

[14] Liu F，Li Y，Xia F，et al. A method of APT attack detection based on DBN-SVDD [J]. Compute Science Applied，2017，7(11)：1146—1155

[15] Vinayakumar R，Soman K P，Poornachandran P. Detecting malicious domain names using deep learning approaches at scale [J]. Journal of Intelligent & Fuzzy Systems，2018，34(3)：1355—1367

[16] Niu W，Zhang X，Yang G W，et al. Identifying APT malware domain based on mobile DNS logging [J]. Mathematical Problems in Engineering，2017，2017：4916953

[17] Bahrami P N，Dehghantanha A，Dargahi T，et al. Cyber kill chain-based taxonomy of advanced persistent threat actors：Analogy of tactics，techniques，and procedures [J]. Journal of Information Processing Systems，2019，15(4)：865—889

[18] Kim H，Kwon H J，Kim K K. Modified cyber kill chain model for multimedia service environments [J]. Multimedia Tools and Applications，2019，78(3)：3153—3170

[19] Zhou Z，Xu C，Kuang X，et al. An efficient and agile spatio-temporal route mutation moving target defense mechanism [C] //Proc of IEEE Int Conf on Communications (ICC). Piscataway，NJ：IEEE，2019：1—6

[20] Jasiul B，Szpyrka M，Śliwa J. Detection and modeling of cyber attacks with petri nets [J]. Entropy，2014，16(12)：6602—6623

[21] 杜镇宇，刘方正，李翼宏. 基于 Petri 网的 APT 攻击模型生成方法[J]. 计算机应用研究，2019，36(7)：2134—2142

[22] Ghafir I, Kyriakopoulos K G, Lambotharan S, et al. Hidden Markov models and alert correlations for the prediction of advanced persistent threats [J]. IEEE Access, 2019, 7: 99508—99520

[23] Niu W, Zhang X, Yang G, et al. Modeling attack process of advanced persistent threat using network evolution [J]. IEICE Trans on Information and Systems, 2017, 100(10): 2275—2286

[24] 孙文新. 基于机器学习的 APT 流量检测研究[D]. 北京: 北京邮电大学, 2020

[25] Hutchins E M, Cloppert M J, Amin R M. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains [J/OL]. Leading Issues in Information Warfare & Security Research, 2011, 1(1) [2022-07-30]. https://www.researchgate.net/publication/266038451_Intelligence-Driven_Computer_Network_Defense_Informed_by_Analysis_of_Adversary_Campaigns_and_Intrusion_Kill_Chains

[26] 刘海波, 武天博, 沈晶, 等. 基于 GAN-LSTM 的 APT 攻击检测[J]. 计算机科学, 2020, 47(1): 281—286

时　林

硕士研究生.主要研究方向为漏洞挖掘、软件安全防护.

shilin@stu.pku.edu.cn

时绍森

硕士.主要研究方向为恶意代码检测、漏洞挖掘、Web 攻防.

511306747@qq.com

文伟平

教授,博士生导师.主要研究方向为系统与网络安全、大数据与云安全、智能计算安全.

weipingwen@pku.edu.cn