基于 OpenStack 云平台的 Docker 容器安全监测方法研究*

崔 轲，燕 玮，刘子健，张慕榕，贾星威，许凤凯
（华北计算机系统工程研究所，北京 100083）

摘 要： 随着虚拟化技术和容器技术的兴起，容器安全问题引起了社会和企业的广泛重视。针对传统的监控方式对 Docker 容器信息监控不全面、易产生监控黑洞等问题，提出一种针对 OpenStack 云平台下的 Docker 容器安全监测方法，该方法针对性强，资源占用率小，除了实现传统监测功能外，通过采用 Logistic－ARMA 预警模型和 BERT 序列标注，还可以实现对 DoS 攻击、容器逃逸等恶意攻击的有效监测，且根据容器规模不同可实现自定义的预警功能。经过实验验证，该方法在大规模容器网络中威胁预测准确率可达85%以上。

关键词： Docker 容器；Logistic－ARMA；Bert 序列标注；大规模容器网络
中图分类号： TP391　　**文献标识码：** A　　**DOI：** 10.19358/j.issn.2096－5133.2022.04.010
引用格式： 崔轲，燕玮，刘子健，等. 基于 OpenStack 云平台的 Docker 容器安全监测方法研究[J].信息技术与网络安全，2022，41(4)：65－70.

Research on security monitoring method of Docker container engine based on OpenStack cloud platform

Cui Ke，Yan Wei，Liu Zijian，Zhang Murong，Jia Xingwei，Xu Fengkai
（National Computer System Engineering Research Institute of China，Beijing 100083，China）

Abstract： With the rise of virtualization technology and container technology，container security has attracted extensive attention of society and enterprises. In view of the problems that the traditional monitoring method does not fully monitor the Docker container information and is easy to produce monitoring black holes，this paper proposes a Docker container security monitoring method under the OpenStack cloud platform. This method has strong pertinence and low resource occupancy. In addition to realizing the traditional monitoring function，this methocl can effectively monitor the malicious attacks such as DoS attack and container escape by using the Logistic－ARMA warning model and BERT sequence annotation，and realize the customized early warning functions according to different container sizes. Experimental results show that the accuracy of threat prediction in large－scale container networks can reach more than 85%.

Key words： Docker container；Logistic－ARMA；Bert sequence annotation；large scale container network

0 引言

随着虚拟化技术的兴起，Docker 容器技术凭借自身资源占用低、不易受环境因素影响等特点被各大企业广泛使用。例如京东、天猫等电商企业在大型销售活动中均采用 Docker 作为关键业务的支撑，招行、浦发和法国兴业等国内外银行通过 Docker 搭建容器及服务的金融云架构平台，为千万级用户提供个人理财、快速交付等服务。但是，在容器技术被各大企业广泛使用的同时，容器逃逸、资源非法占用、容器 DoS 攻击等安全问题也愈演愈烈，容器本身及其运行环境的安全问题亟待解决。目前针对容器安全监测的方法很多，例如 Zabbix、Scout 插件等，但是，Zabbix 脱离集中数据存储且复杂度较高，Scout 监控起来要收取大量的费用，并且存在监控不全面，易产生监控黑洞等问题，因此，针对容器安全监测问题，本文提出了一种新型的安全监测方法，该方法可对容器的整个生命周期进行监控，并且对容器遭受逃逸和 DoS 攻击等可以进行有效监测[1]。

* 基金项目：国防基础科研计划(JCKY2020211B005)

1 理论基础

1.1 轻量级容器引擎

轻量级容器引擎是一款自动化的管理系统，主要包括镜像、镜像仓库和容器三个部分[2]。其中镜像是用来构建容器的基础，镜像仓库是用来存储镜像和分发镜像，容器是由镜像生成的，是镜像运行的实际表现。

轻量级容器引擎可对少量的 Docker 容器进行创建、删除和管理。针对构建的 Docker 容器网络集群进行统一视图管理则需要通过专用的容器管理平台。

1.2 容器管理平台

目前主流的容器管理平台有 Kubernetes、OpenStack 等。Kubernetes 和 OpenStack 这两种平台目前都十分受到广大用户的青睐，但是，去年 12 月份，Kubernetes 宣布自 v1.20 起放弃对 Docker 容器的支持，这将导致使用 Kubernetes 的大量用户将转向应用 OpenStack 进行 Docker 容器的管理[3-4]。

OpenStack 是一个开源的云计算容器管理平台，内部的核心组件有 Nova、Neutron 和 Heat 等，其中，Nova 通过 Nova Docker driver 可以实现对 Docker 容器的管理；Neutron 对容器提供虚拟网络；Heat 可对云平台上的 Docker 容器进行大规模创建、修改和删除[5-6]。

1.3 安全风险分析

Docker 容器常见的安全风险问题有用户非法提权、DoS 攻击等。例如 Linux 内核在 3.16 以前的版本存在内核溢出的漏洞，导致宿主机和 Docker 容器崩溃；国外某公司通过上传包含钓鱼程序的镜像来获取用户信息等[7-8]。为保证基于 OpenStack 云管理平台构建的 Docker 容器集群安全运行，本文从内核、Docker 镜像和网络等方面对 Docker 容器面临的安全风险进行分析[9]。

内核安全风险：namespace 命名空间保证各容器进程存在于不同的运行空间。cgroups 保证各容器占用的资源彼此独立。但是，由于容器共享物理机内核，恶意用户将以此为攻击点对容器发起内核攻击[10-12]。

Docker 镜像安全风险：Docker 镜像大多数来自于镜像仓库，仓库中存在用来钓鱼的恶意镜像，这些镜像一旦被用户下载使用，将会对整个项目产生致命的危害[13]。

网络安全风险：Docker 容器网络默认使用 bridge 网络模式，攻击者可以利用物理机的内网发起 ARP 欺骗、网络嗅探和广播风暴等攻击[14]。

针对以上安全风险，本文从以下两方面来保障容器的安全运行：

(1)安全加固策略

主要针对内核、镜像、网络、容器等方面进行安全加固。①内核：及时更新物理机内核，采用 SELinux、AppArmor 或 GRSEC 控制文件访问权限[15]。②镜像：采用官方认证过的镜像文件，并且不定期对镜像文件进行扫检。③容器：禁止在容器内开启远程调用接口，避免以特权模式启动容器[16]。

(2)专用安全组件

主要针对威胁发现与处置、安全扫描、安全监测方面进行安全加固。①威胁发现与处置类：部署入侵防御工具对容器的攻击性行为进行中断和禁止。②安全扫描类：利用扫描工具对容器内的漏洞、木马、病毒和恶意软件等进行扫描分析。③安全监测类：利用监测组件对容器整个生命周期的数据进行安全监测。因此在容器遭受 DoS 攻击、容器逃逸以及容器自身资源异常预测方面可以通过实时监控，对数据进行分析和预测从而保证容器的安全运行。

2 安全监测方法

针对 OpenStack 云平台上 Docker 容器集群的安全运行，本文采用 prometheus+influxdb+cadvisor 架构，通过采集代理将容器中当前的数据进行筛选和聚合分类，结合用户制定的规则库对该数据进行匹配，若匹配成功则触发报警机制并分级响应显示给用户查看，若匹配失败则不做任何处理。其次，由于采集的是以时间序列为主的数据，因此，本文进一步选取 Logistic-ARMA 时间序列预警模型对该类数据进行分析和处理，获得容器数据在时间维度上的关联性，从而预测容器在未来一段时间内的资源使用情况，最终实现对容器的资源进行实时预测和监测。再结合目前较为流行且准确率较高的 BERT 序列标注算法，对用户手动输入的文本进行文本标注，进行安全特征库的更新[17-18]。该监测方法由五部分组成，分别为数据获取、数据预处理、数据监测、数据处置和监测管理，原理图如图 1 所示。

(1)数据获取

数据获取中存在数据采集和数据收集，数据采

集主要通过服务端对云平台上的数据进行拉取，数据收集利用监控代理数据进行收集，保存在数据库中，以此获得容器信息及运行状态。

(2)数据预处理

主要对数据进行预处理操作，包括缺失值、异常值剔除等，通过数据筛选器对数据进行筛选，作为数据监测部分的数据输入。

(3)数据监测

数据监测部分由容器安全数据监测和威胁态势感知算法模型组成。预处理后的数据同时流入安

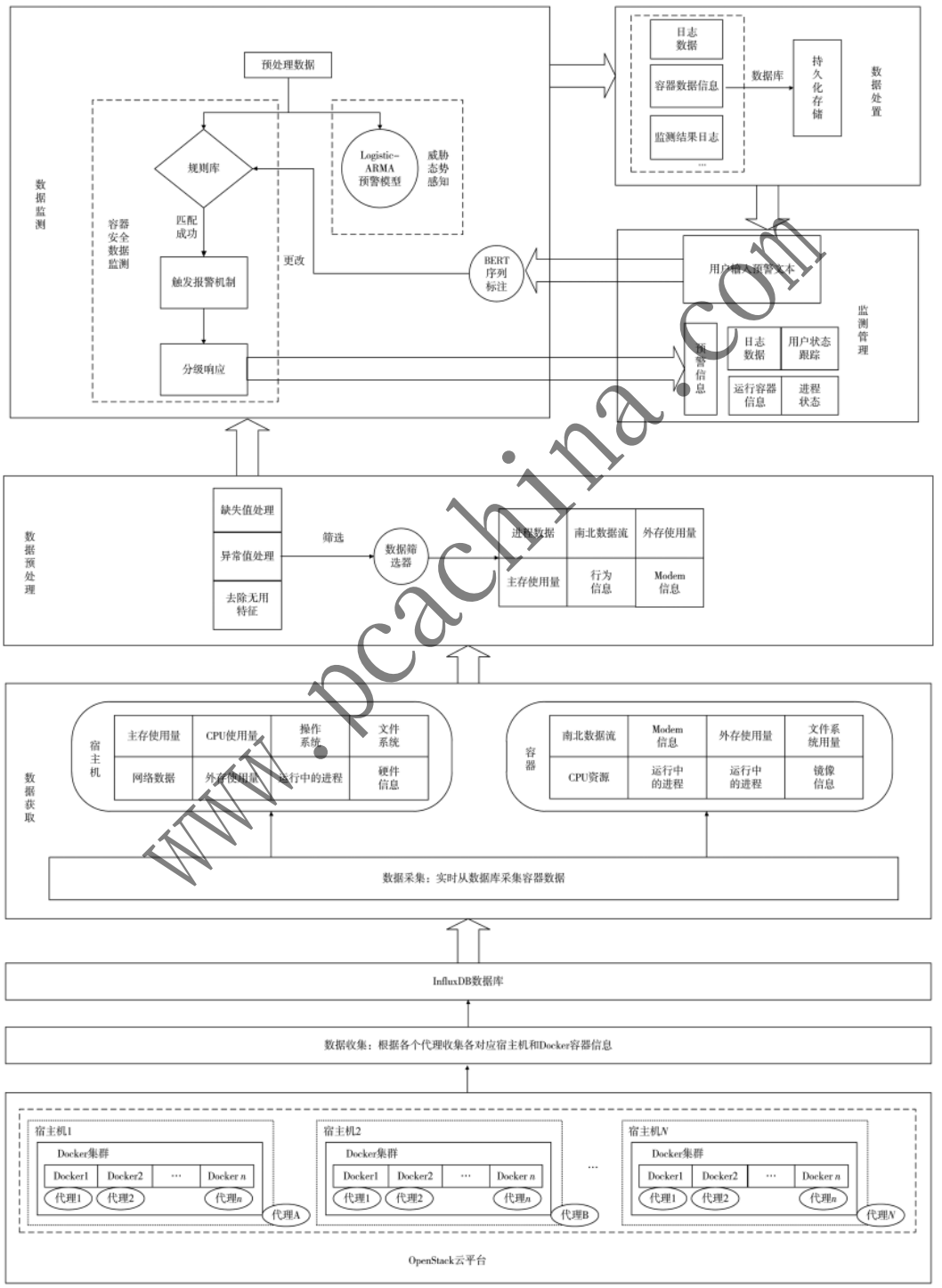

图 1 基于云平台容器的安全监测方法原理

全数据监测和威胁态势感知中，在安全数据监测中将预处理后的数据与自定义的报警规则进行匹配，若匹配成功则触发报警机制，并且进行分级响应，以供管理员采取应急响应措施。在威胁态势感知中，利用数据挖掘、文本分析和流量分析技术，将处理后的数据经过 Logistic－ARMA 预警模型进行数据趋势感知，通过对系统日志的分析和处理，获得容器数据在时间维度上的关联性，从而预测容器在未来一段时间内的资源使用情况，最终实现对容器的资源进行实时预测和监测。在告警规则中，用户可以通过输入预警文本，预警文本通过 BERT 序列模型标注出关键字，从而实现对报警规则的动态更新。数据监测部分的所有数据都将流入数据处置和监测管理部分。

(4)数据处置

数据存储主要对安全分析部分产生的数据以及整个系统日志中的数据进行存储，利用数据库对容器数据进行持久化存储，这样有利于历史的回溯，防止采集到的数据发生丢失现象。

(5)监测管理

监测管理主要采用了数据可视化工具库实现，立体地对安全威胁态势进行综合展示以及对基于云平台上运行的容器状态进行可视化，以供管理员查看各个容器的安全状态，包括运行容器的基本信息、报警信息、资源预测、用户状态跟踪和各个进程状态信息等，实现从攻击预警、攻击识别到分析取证的综合能力。

安全监测方法流程如图 2 所示。启动服务后，

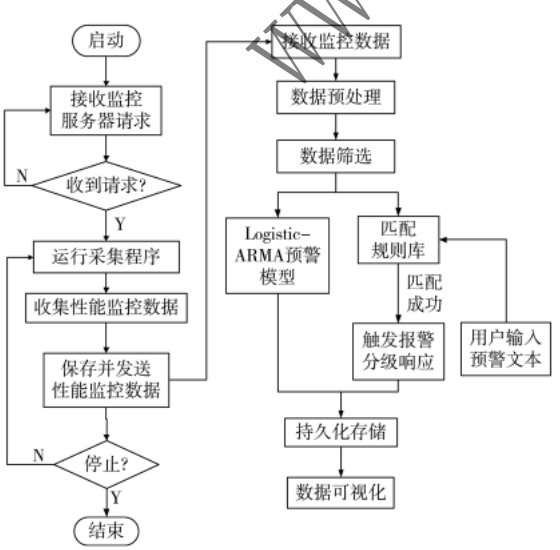

图 2 安全监测方法流程图

接收监控服务器请求，监控代理开始采集主机上运行容器的信息，并将信息发送给数据库进行保存，服务端从数据库中拉取数据并将数据传入到数据预处理模块，数据预处理模块对数据进行缺失值、异常值剔除并将主机和容器指标之外的无用特征去除，将处理后的数据通过数据筛选器筛选出可用特征信息，之后一方面传入报警规则中，与报警规则进行匹配查看是否为异常数据，另外一方面传入 Logistic－ARMA 模型中进行训练预测，并实时将预测的数据与报警规则进行匹配。在该阶段产生的数据均将传入到数据库和可视化界面，进行数据的保存与显示。

3 实验分析

本文采用实验室搭建的私有云平台环境，将其中一台计算节点服务器作为 Docker 容器集群管理节点，并在 OpenStack 云平台上创建 1 000 个 Docker 容器组成轻量级容器集群网络，Docker 容器采用 Alpine 基础镜像，并在该基础镜像上部署了探针、SSH 等服务，将其中一台 Docker 作为测试机，一台作为攻击机，一台作为监测终端进行攻击实验，攻击机网段为：192.168.100.0/24，测试终端网段为：192.168.100.0/24，Docker 集群网段为：172.16.0.0/16，并且在测试机上装有探针去实时采集数据。在 OpenStack 各个计算节点上部署 Docker 服务和各个功能模块后，进入监控界面，如图 3 所示。

3.1 容器逃逸监测

对 1 000 个 Docker 容器通过监测其进程名字空间测试容器逃逸，如图 4 所示，从监测管理图上可以直观看到指定容器内所有进程的 Namespace 状态的实时监测值，横轴对应采集时间序列，纵轴对应进程的 Namespace 状态值。正常行为的名字空间状态值(Nsid)为 1。

在容器内发起逃逸攻击后，通过监测管理中的进程管理可以成功监测出逃逸行为，图 5 所示为容器内各进程的最新 Namespace 状态值，可以看到 Pid 为 6464 的进程属于容器值为 9 时触发触发器，随后发出报警。

3.2 DoS 攻击监控效果

在没有合理划分容器使用内存的情况下，恶意用户通过无限制申请内存可造成容器运行缓慢，甚至造成宿主机内存耗尽发生宕机。通过在名字为 dockermon 容器内运行不断申请动态内存的进程，

图 3 容器集群整体状态监控图

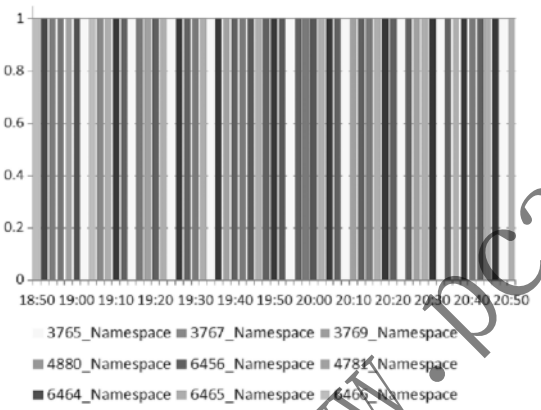

图 4 容器内进程 Namespace 状态监控图

	状态值
4880_Namespace	1
6456_Namespace	1
6464_Namespace	9
6465_Namespace	1
6466_Namespace	1
触发器：Docker容器逃逸	[=9]

图 5 容器逃逸监测图

发起资源耗尽型拒绝服务攻击，该容器的 CPU 监控实时状态如图 6 所示，横轴为时间序列，纵轴为实时 CPU 时间片值。可以看到 2:21:15PM 时刻 CPU 使用骤然上升，并超过触发器设置的阈值。

3.3 威胁预警功能测试

将大量的训练集和测试集数据流入 Logistic – ARMA 预警模型中进行训练，预警功能用于在应用数据出现异常时，或数据即将发生异常时，根据用户自定义编写的容器触发报警规则向用户发送警报，并且对攻击类型进行有效判别。实验结果如表1所示，本次实验共分为 8 组，其中有 3 组 DoS 攻击，3 组容器逃逸，2 组正常操作，结果表明有 7 组预判正确，1 组预判失败，该方法威胁预测准确率可达

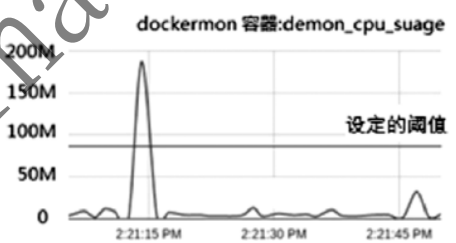

图 6 容器 CPU 使用量监控图

表 1 威胁预判记录

攻击程序编号	实际攻击类型	预测攻击类型	预测有效性
1	DoS 攻击	DoS 攻击	√
2	容器逃逸	容器逃逸	√
3	/	DoS 攻击	×
4	容器逃逸	容器逃逸	√
5	DoS 攻击	DoS 攻击	√
6	容器逃逸	容器逃逸	√
7	/	/	√
8	DoS 攻击	DoS 攻击	√

85%。其中判断错误的一组是由于网络使用的带宽接近设计的阈值，在应用预警模型的时候进行了误判处理。因此，后续可对该模型添加 SVM 机制进行准确率的提升。

4 结论

本文以目前较流行的 OpenStack 云平台为基础，提出了基于 OpenStack 云平台的 Docker 容器引擎安全监测方法，并且利用 BERT 序列标注和 Logistic-ARMA 预警模型对容器资源进行威胁态势感知，一旦当容器遭到了 DoS 攻击或者容器逃逸时，该方法能对其进行有效识别并且进行显示。当容器在一段时间内资源使用率过高或者负载率过高的时候，该方法能对其进行未来一段时间内的资源预测，一旦预测资源会超出瓶颈时，立即触发报警功能，有利于管理人员及时进行维护，从而满足企业对于 Docker 容器引擎的监控，弥补了 Docker 容器监控在此方面的不足。

参考文献

[1] 毕玉蓉.基于 Zabbix 的软件监测告警系统的设计与开发[D].南京：东南大学，2019.

[2] 汪恺，张功萱，周秀敏.基于容器虚拟化技术研究[J].计算机技术与发展，2015，25(8)：138-141.

[3] IT 老男孩.了解 Kubernetes 1.20[EB/OL].(2021-02-15)[2022-01-10].https://www.xtplayer.cn/kubernetes/about-kubernetes-1.20.

[4] Li Haifeng，Zhou Huachun，Zhang Hongke，et al.An openstack-based DTN network emulation platform(extended version)[J].Mobile Information Systems，2016，2016(5).DOI：10.1155/2016/6540207.

[5] 李佳曦.基于容器技术的云化平台安全风险与应对分析[J].信息通信技术，2020，14(6)：26-31，38.

[6] Zhang Yongmin，Lan Xiaolong，Ren Ju，et al.Efficient computing resource sharing for mobile edge-cloud computing networks[J].IEEE/ACM Transactions on Networking，2020，28(3)：1227-1240.

[7] CHELLADHURAI J，CHELLIAH P R，KUMAR S A.Securing Docker containers from Denial of Service(DoS) attacks[C]//2016 IEEE International Conference on Services Computing(SCC)，2016：856-859.

[8] LAWRENCE G.Dirty COW(CVE-2016-5195)-Docker container escape[EB/OL].[2022-01-10].https://blog.paranoidsoftware.com/dirty-cow-cve-2016-5195-docker-container-escape/.

[9] DoSec 安全团队.Docker 容器最佳安全实践白皮书[R].2018.

[10] The Linux Kernel Archives.Cgroups[EB/OL].(2018-xx-xx).https://kernel.org/doc/Documentation/cgroup-v1/cgroups.txt.

[11] 孙建波.Docker 背后的内核知识——cgroups 资源限制[EB/OL].(2015-04-20).https://www.infoq.cn/news/docker-kernel-knowledge-cgroups-resource-isolation.

[12] 苟兴昊.虚拟化容器的安全隔离技术的研究与实现[D].成都：电子科技大学，2021.

[13] HUANG D，CUI H，WEN S，et al.Security analysis and threats detection techniques on Docker container[C]//2019 IEEE 5th International Conference on Computer and Communications(ICCC)，Chengdu，China，2019：1214-1220.

[14] GUPTA R R，MISHRA G，KATARA S，et al.Data storage security in cloud computing using container clustering[C]//2016 IEEE 7th Annual Ubiquitous Computing，Electronics & Mobile Communication Conference，New York，USA，2016：1-7.

[15] MA B，MU D J，FAN W，et al.Improvements the sec-comp sandbox based on PBE theory[C]//International Conference on Advanced Information Networking and Applications Workshops，2013：323-328.

[16] 任兰芳，庄小君，付俊.Docker 容器安全防护技术研究[J].电信工程技术与标准化，2020，33(3)：73-78.

[17] 郑强清.基于 LSTM 的容器云资源预测与配置的研究[D].桂林：桂林理工大学，2019.

[18] 姜猛.基于深度学习的中文信息抽取研究[D].贵阳：贵州大学，2019.

(收稿日期：2022-01-21)

作者简介：

崔轲(1996-)，男，硕士研究生，主要研究方向：云安全。

燕玮(1990-)，男，硕士，工程师，主要研究方向：云安全、工控安全、行业系统仿真。

刘子健(1991-)，男，本科，助理工程师，主要研究方向：云计算。

版权声明

经作者授权，本论文版权和信息网络传播权归属于《信息技术与网络安全》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《信息技术与网络安全》编辑部

中国电子信息产业集团有限公司第六研究所