文章编号:1007-5321(2018)02-0050-06 **DOI**:10.13190/j.jbupt.2017-192

一种基于虚拟化的文件杀毒实现方法

尹学渊[1], 陈兴蜀[2], 李 辉[1], 陈 林[1]
(1. 四川大学 计算机学院, 成都 610065; 2. 四川大学 网络空间安全研究院, 成都 610065)

摘要: 针对云计算环境下因虚拟机杀毒和病毒库更新等引发的性能开销及资源占用问题,提出了一种基于虚拟化的云杀毒实现框架——HyperAV,HyperAV 能够在较低性能开销的情况下对虚拟机文件进行杀毒,并提供扇区级别的访问控制和隔离机制. 通过获取虚拟机运行过程中的扇区更改信息,HyperAV 优化了文件病毒的扫描方式,对虚拟机杀毒过程起到了显著加速效果. HyperAV 实现了控制—杀毒相互分离的前后端框架,杀毒所需数据可以导入专用杀毒服务集群,从而避免了病毒库的重复更新,解决了因杀毒而使虚拟机运行缓慢的问题. 在基于内核的虚拟机虚拟化平台下实现了原型系统. 实验结果表明,HyperAV 能够在虚拟机较低负载开销的情况下为虚拟机文件提供病毒扫描防护能力.

关 键 词: 云计算; 虚拟化; 虚拟机; 杀毒
中图分类号: TP316 **文献标志码**: A

A Method to Implement File Antivirus Based on Virtualization

YIN Xue-yuan[1], CHEN Xing-shu[2], LI Hui[1], CHEN Lin[1]
(1. College of Computer Science, Sichuan University, Chengdu 610065, China;
2. Cybersecurity Research Institute, Sichuan University, Chengdu 610065, China)

Abstract: To solve the performance overhead and resource consumption brought by an antivirus software when performing operations of virus scanning and virus database updating, an antivirus framework named HyperAV based on virtualization was proposed. HyperAV was able to provide antivirus capability for virtual machine files with low performance overhead, a mechanism of access control and isolation at the granularity of sector level was also provided. The process of virus scanning was optimized by monitoring the sector change information of a running virtual machine, which had a significant acceleration effect to the virus scanning process of virtual machines. HyperAV was constructed by a front and a rear end with each used as a controller and an antivirus worker, the data needed by antivirus software was redirected to server clusters so that duplications of virus database updating could be avoided, and performance overload brought by antivirus software running inside virtual machines was avoided. A prototype system based on kernel-based virtual machine virtualization platform was realized, the results showed that HyperAV was able to provide antivirus capability with low performance overload for virtual machines.

Key words: cloud computing; virtualization; virtual machine; antivirus

随着云计算技术的快速发展与应用,其安全问题也备受关注. Islam 等[1-3]对云计算安全问题进行了分类和整理,从网络安全[4]、虚拟化安全等方面对云计算安全威胁和风险进行了概述. 俞能海

收稿日期: 2017-09-20
基金项目: 国家科技支撑计划项目(2012BAH18B05);国家自然科学基金项目(61272447)
作者简介: 尹学渊(1988—), 男, 博士生; 陈兴蜀(1968—), 女, 教授, 博士生导师, E-mail: chenxsh@scu.edu.cn.

等[5]介绍了当前云计算的主要安全技术，涉及数据加密与存储、身份认证、可信云计算[6-7]等，同时指出杀毒服务应该作为安全即服务之一提供给云用户使用.

在云杀毒方面，Oberheide 等[8]提出了一种基于网络的前后端杀毒机制，前端获取被杀毒文件，后端通过网络为前端提供杀毒服务. Zheng 等[9]和 Oberheide 等[10]对上述方法做了进一步拓展，虚拟机中的前端首先计算出被杀毒文件的唯一标识符(UID, unique identifier)数值(MD5)，比较 UID 与白名单列表，判断文件是否需要发送给后端进行病毒查杀. Nachenberg[11]针对病毒扫描过程提出了一种加速方法，将文件拆分成了不同扇区. 当首次对某个文件进行病毒扫描时，文件包含的每个扇区都被分别计算 Hash 值. 当再次对该文件进行杀毒时，该方法重新计算每个扇区的 Hash 值，只有当有扇区内容变化时才会再次对文件进行病毒扫描. McAfee[12]将防病毒软件与虚拟化基础设施相整合，将病毒扫描工作转移到专用服务器进行.

CloudAV[10]需要在虚拟机内部实现一个代理，通过代理程序计算并提取文件的 UID 数值来确定文件的安全性，该过程会消耗虚拟机的 IO 和 CPU 资源. 同样，Nachenberg[11]提出的病毒扫描加速方法需要读取虚拟机扇区数据并计算 Hash 值来判断该文件内容是否发生变化，效率仍然不够高效. 针对上述问题，笔者提出并实现了一套基于虚拟化技术的杀毒框架——HyperAV. HyperAV 具有如下特点：

1) 更加轻量级的客户端代理，与 CloudAV 不同，代理程序不需要读取被杀毒文件的数据内容并计算 UID 数值；

2) 杀毒与虚拟机监视器相结合，通过底层虚拟化平台来监控虚拟机运行过程中的扇区修改信息；

3) 基于底层虚拟化技术提供扇区级别的访问控制，为用户文件提供保护与隔离机制；

4) 在虚拟机外直接重构文件内容，减少了通用的从虚拟机内部读取文件内容的 IO 开销；

5) 多引擎同时扫描，HyperAV 后端可以接入多个杀毒引擎，采用集群的工作方式能够提升查杀率以及查杀速度.

1　基于内核的虚拟机 IO 虚拟化

基于内核的虚拟机(KVM, kernel-based virtual machine)[13]模块基于 Linux 内核和硬件辅助虚拟化技术实现，x86 平台下利用了 Intel 的 VT[14](virtualization technology)技术以及 QEMU[15]模拟器，KVM 作为可执行模块可以动态插入到 Linux 内核中. QEMU 运行在宿主机的应用层，为 KVM 提供了各种设备模型的模拟以及用户态接口，两者通过 ioctl 系统调用交互.

在 KVM 虚拟化平台上，virtio_blk 作为磁盘驱动的前端，安装在虚拟机中负责与 QEMU 模拟器中的后端交互. 当虚拟机中产生磁盘 IO 时，触发 VM-Exit 使执行流程到达宿主机内核中的 KVM 模块. 磁盘 IO 模拟需要借助于 QEMU，针对不同镜像格式，QEMU 完成不同的 IO 模拟操作. 当前，QEMU 已经实现了对主流镜像格式的支持，其中较为常用的包括 RAW 和 qcow2[16]格式.

2　HyperAV 实现

2.1　系统架构

如图 1 所示，HyperAV 主要由运行在虚拟机中的 HyperAV Tools、宿主机中的 QEMU 前端以及提供杀毒服务的后端组成. 其中，HyperAV Tools 用于获取被杀毒文件的物理扇区信息并发送给后端.

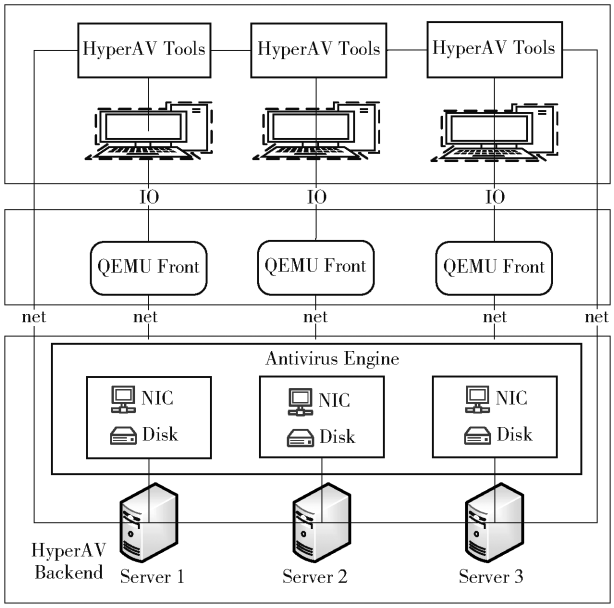

图 1　HyperAV 系统架构

HyperAV 后端在收到 HyperAV Tools 发送的杀毒请求后，判断文件所在扇区内容自上次杀毒之后是否发生了变化. 若扇区数据未被更改，则直接返回上次杀毒结果给 HyperAV Tools，否则将通知 QEMU 前端读取扇区数据并进行病毒扫描. QEMU 前

端在收到 HyperAV 后端的数据读取请求后,负责读取扇区内容并发送给后端,后端调用相应杀毒引擎进行杀毒.

2.2 云杀毒实现

为加速虚拟机中的病毒扫描过程,HyperAV 在每次杀毒时,只对那些从上次杀毒之后扇区内容发生变化的文件进行杀毒,因而需要跟踪并记录虚拟机中的扇区变化情况.

当 QEMU 初始化时,线程 NetAV 被创建,NetAV 负责从 QEMU 主线程接收磁盘更改信息并传递给 HyperAV 后端中的 NetAV Server,如图 2 所示. 虚拟机的 IO 数据经过 KVM 到达 QEMU 后,由 QEMU 主线程负责实现虚拟机 IO 模拟. 虚拟机的每次 IO 写入操作可使用数据结构 sector_ops:{start_sector, sector_num, ops_size}定义,用以描述虚拟机每次 IO 写入时的扇区号以及扇区数量.

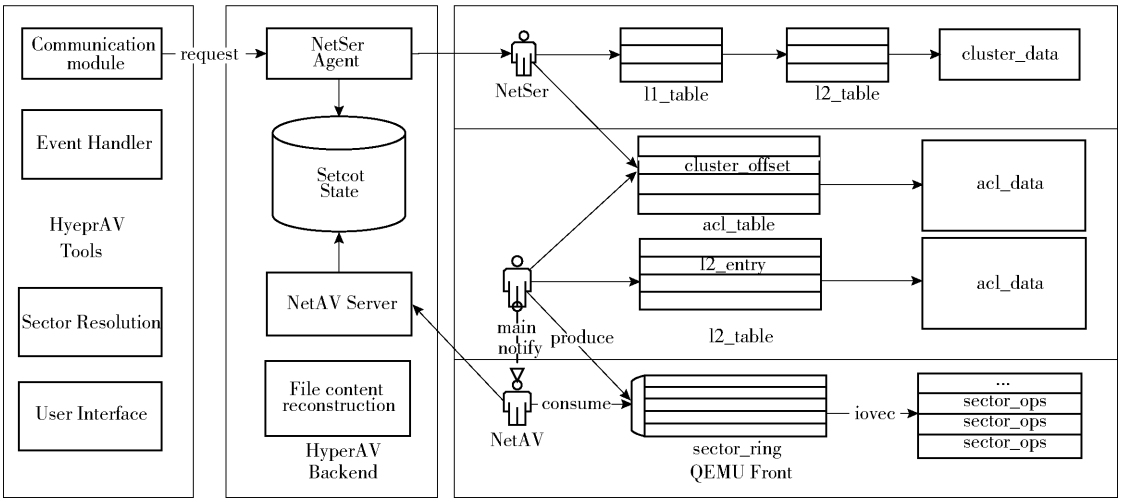

图 2 HyperAV 实现原理

QEMU 主线程使用环形队列 sector_ring 向 NetAV 线程传递数据,当虚拟机 IO 操作到达主线程后,主线程判断若为写操作,则将当前虚拟机的 IO 操作抽象成 sector_ops 结构并存入队列中,通知 NetAV 线程处理. NetAV 线程从队列中获取扇区变化信息并发送给后端杀毒服务器.

为标识虚拟机中的扇区内容自上次杀毒后是否被更改,HyperAV 后端采用位图的方式记录每个扇区的状态. 当杀毒请求从 HyperAV Tools 发出后,杀毒后端可以直接根据扇区状态判断所述文件内容自上次杀毒后是否发生了变化.

2.3 扇区隔离

针对有安全风险的文件,HyperAV 提供隔离机制,用于将所述文件从虚拟机中隔离出来,防止被虚拟机恶意程序读写与执行. HyperAV 定义的 qcow2 镜像格式如图 3 所示.

虚拟机在读取文件数据时,文件偏移经过虚拟机内核转换为磁盘扇区偏移,扇区偏移按照 qcow2 格式转换为相对于 qcow2 镜像文件的内部偏移. 该过程需要遍历 qcow2 的一级表(l1_table)和二级表(l2_table),根据二级表表项内容 l2_entry 得到虚拟机磁盘数据所在簇偏移,默认情况下扇区偏移低 16 位被用于索引簇内偏移得到对应数据.

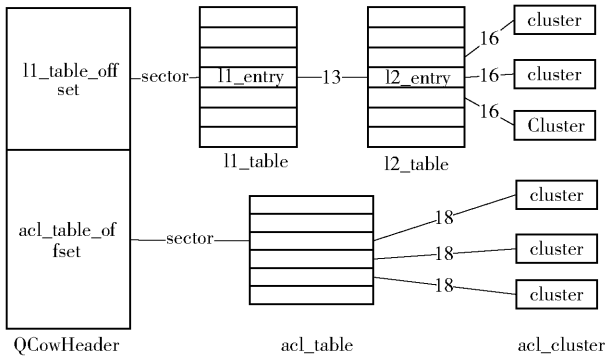

图 3 HyperAV 定义的 qcow2 镜像格式

HyperAV 定义的 qcow2 二级表格式如图 4 所示,qcow2 镜像二级表表项第 9 ~ 54 位表示簇偏移量,其余位用作标志位. 除开 qcow2 自带的标志位外,HyperAV 新添加了 DENY_WRITE 和 DENY_READ 标志位用以控制虚拟机对该簇的访问控制权限. 默认情况下,每个簇可以容纳 128 个扇区,二级表中的访问控制标志只能实现以簇为粒度的访问控制,而虚拟机需要实现以扇区为粒度的访问控制机

制. 因而镜像中新增了 acl_table,用以标志虚拟机每个扇区的访问控制权限.

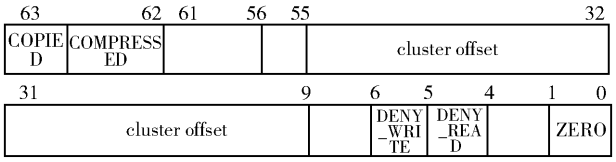

图 4　HyperAV 定义的 qcow2 二级表格式

2.4　客户代理

HyperAV Tools 作为客户代理工具被安装在虚拟机中,主要用于解析虚拟机中的文件所在物理扇区信息以及接收系统文件相关的事件通知. HyperAV Tools 还提供用户接口,当用户对指定文件杀毒时,HyperAV Tools 首先把文件缓存刷新到磁盘,随后读取文件扇区信息与 HyperAV 后端交互.

后端在收到 HyperAV Tools 发送的扇区信息后,首先将所述扇区信息按序合并成一段连续的区域;然后向 QEMU 前端 NetSer 发起扇区内容读取请求. 通过该方法,后端能够在服务端重构出虚拟机的文件内容,避免了从虚拟机内部读取文件内容的过程.

通过利用操作系统提供的事件通知机制,HyperAV 能够获取虚拟机中指定目录下的文件新建、更改以及属性变更等信息,以此提供对指定文件的实时病毒扫描. 当系统关键位置中的文件内容发生变化时,HyperAV Tools 接收到系统通知后,获取被更改文件的物理扇区信息并传递给 HyperAV 后端. 通过文件内容重构技术,后端可以对系统关键位置中的文件提供实时病毒查杀.

3　实验测试与分析

所提出的方法在 KVM 虚拟化平台下进行了实现. 其中,测试虚拟机操作系统版本为 Windows 7 企业版 Service Pack 1,配置 2 vCPU、4 GB 内存. 宿主机物理 CPU 型号 Intel(R) Xeon(R) CPU E5-2630 v3,2.40 GHz,128 GB 内存,磁盘转速为 7 200 r/min,磁盘容量为 2 TB. 宿主机操作系统版本为 CentOS 7.2,内核版本为 3.10.0 – 514.2.2.el7,QEMU 版本为 2.3.0. 测试例中使用的杀毒引擎为 ClamAV[17],一款可在 Linux、Windows、Mac OS 环境下使用的开源高性能杀毒引擎.

3.1　文件内容重构

通过文件内容重构,HyperAV 能够在虚拟机外部重新构造出虚拟机中的文件内容. 图 5 给出了 HyperAV Tools 获取到文件 test.com 的扇区位置信息,每项分别对应逻辑扇区、物理扇区和扇区有效数据大小. 上述信息被发送给后端后,后端经过文件重构方法,向 QEMU 前端发起文件读取请求并将文件重构为 test.com1,如图 6 所示. 2 个文件的 MD5 值相同. 同样,文件 PS-0352C.COM 以上述相同方式被重构,与虚拟机中原来的文件 MD5 值相同.

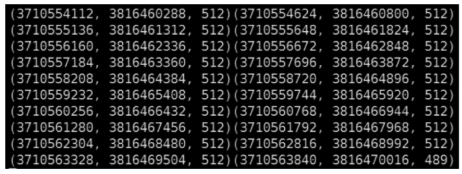

图 5　文件 test.com 对应的扇区位置信息

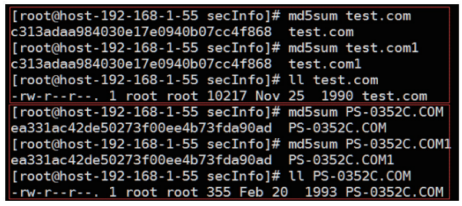

图 6　杀毒后端重构文件内容

3.2　实时扫描

HyperAV 能够对虚拟机提供实时病毒扫描,通过 HyperAV Tools 设置虚拟机中被监控的目录为 C:\test. 当每次在该目录下修改 test.txt 文件时,如图 7 所示,此时 test.txt 文件内容在后端被实时重现,如图 8 所示.

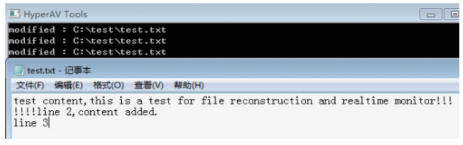

图 7　监控到虚拟机 C:\test 目录下文件被修改

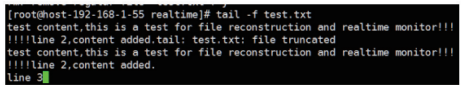

图 8　后端实时重构被修改的虚拟机文件内容

图 9 示出了往 C:\test 目录下复制病毒样本文件 ClamAVSample.txt 后,HyperAV 后端重构出文件内容并调用 ClamAV 杀毒引擎扫描后的结果.

当 ClamAV 扫描到 ClamAVSample.txt 包含病毒后,HyperAV 默认对该文件进行隔离. 如图 10 所示,在虚拟机中打开该文件时,系统提示找不到文件.

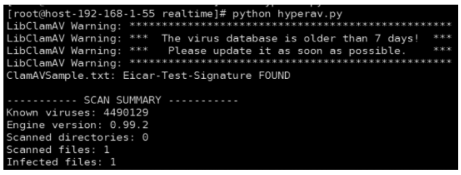

图9 HyperAV 后端实时病毒扫描测试

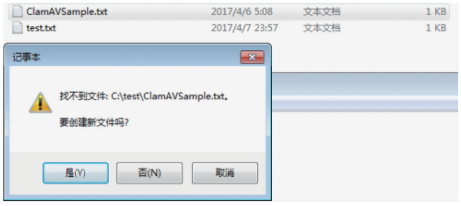

图10 文件扇区隔离测试

3.3 性能测试

图11给出了HyperAV后端获取到的虚拟机扇区修改信息,虚拟机的IO数据到达QEMU后,QEMU会进行IO合并重组,QEMU每次下发到底层存储的IO请求都是一段连续的扇区操作. 主线程捕获该数据后,通过NetAV线程将该数据发送给HyperAV后端并由后端进行存储.

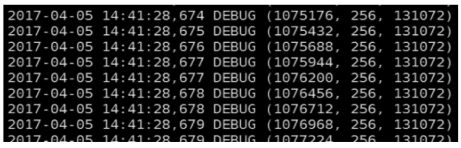

图11 HyperAV 获取的虚拟机扇区修改信息

表1给出了因QEMU前端监控虚拟机运行过程中的扇区修改信息所带来的性能开销和损耗. 监控过程只会影响虚拟机的IO写入性能. 针对表中每行,性能损失的计算方式为(normal-HyperAV)/normal,从表中可以看出,扇区信息监控对虚拟机的IO写入性能影响基本小于1%.

表1 IO写入监控性能测试对比

测试参数		带宽/(KB·s^{-1})		IO操作次数/s	
rw	bs/k	normal	HyperAV	normal	HyperAV
wr	4	13 960	13 504	3 490	3 376
wr	32	84 732	81 867	2 647	2 558
wr	128	138 121	137 269	1 079	1 072
rdw	4	1 664	1 644	416	411
rdw	32	12 417	12 383	388	386
rdw	128	39 884	39 594	311	309

表2对当前杀毒软件的主流处理方式进行了对比. 针对不同文件内容大小,测试给出了调用杀毒引擎ClamAV以及计算文件MD5值的时间开销,最后再对比使用HyperAV云杀毒的时间开销. 第5列(M/H)给出了每行中MD5时间开销除以HyperAV时间开销后的结果,可以看出,HyperAV的时间开销要比MD5值计算节约10倍以上的时间.

表2 通用杀毒处理方式时间对比 10 μs

大小\操作/MB	ClamAV	MD5	HyperAV	M/H
100	1 762 181	274 085	18 796	14.58
50	926 236	136 295	10 254	13.29
10	895 595	32 509	2 841	11.44
1	843 912	3 143	313	10.04
0.500	816 426	1 723	155	11.12
0.100	786 880	615	39	15.76

4 结束语

提出了一种针对云环境下的文件杀毒实现框架HyperAV,避免了云环境下每个虚拟机中都安装杀毒引擎以及病毒库重复更新等问题. 相比于传统杀毒引擎以及其他同类型研究,HyperAV提供更加轻量级的客户端工具,减少了读取文件内容并计算UID数值来确认文件内容是否发生变化的过程,对杀毒时间有显著加速效果. 同时,HyperAV还提供基于扇区级别的访问控制机制来隔离病毒文件和保护用户指定文件内容不被恶意更改.

参考文献:

[1] Islam T, Manivannan D, Zeadally S. A classification and characterization of security threats in cloud computing [J]. International Journal of Next-Generation Computing, 2016, 7(1): 1-17.

[2] Khari M, Gupta S, Kumar M. Security outlook for cloud computing: a proposed architectural-based security classification for cloud computing [C] // 2016 International Conference on Computing for Sustainable Global Development (INDIACom). NJ: IEEE, 2016: 2153-2158.

[3] Khurana R, Gupta H. A hybrid model on cloud security [C] // 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions). NJ: IEEE, 2016: 347-352.

[4] Chen Lin, Chen Xingshu, Jiang Junfang, et al. Research and practice of dynamic network security architecture for IaaS platforms [J]. Tsinghua Science and Technology, 2014, 19(5): 496-507.

[5] 俞能海, 郝卓, 徐甲甲, 等. 云安全研究进展综述

[J]. 电子学报, 2013, 41(2): 371-381.
Yu Nenghai, Hao Zhuo, Xu Jiajia, et al. Review of cloud computing security[J]. Acta Electronica Sinica, 2013, 41(2): 371-381.

[6] 金鑫, 陈兴蜀. 可信链跨物理主机迁移及快速恢复方法[J]. 武汉大学学报(理学版), 2016 (2): 103-109.
Jin Xin, Chen Xingshu. Rapid restoration of migrated trusted chain between physical machines[J]. Journal of Wuhan University (Natural Science Edition), 2016 (2): 103-109.

[7] Zhang Lei, Chen Xingshu , Liu Liang, et al. Trusted domain hierarchical model based on noninterference theory[J]. The Journal of China Universities of Posts and Telecommunications, 2015, 22(4): 7-16.

[8] Oberheide J, Cooke E, Jahanian F. Rethinking antivirus: executable analysis in the network cloud[C]// 2nd USENIX Workshop on Hot Topics in Security (HotSec 2007). Berkeley: USENIX, 2007: 1-5.

[9] Zheng Xufei, Fang Yonghui. An ais-based cloud security model[C]//2010 International Conference on Intelligent Control and Information Processing. NJ: IEEE, 2010: 153-158.

[10] Oberheide J, Cooke E, Jahanian F. CloudAV: *N*-version antivirus in the network cloud[C]//USENIX Security Symposium. Berkeley: USENIX, 2008: 91-106.

[11] Nachenberg C. Antivirus accelerator: U. S, 6021510 [P]. 2000-02-01.

[12] McAfee. McAfee management for optimized virtual environments AntiVirus [EB/OL]. [2017-03-24]. https://www.mcafee.com/cn/resources/data-sheets/ds-move-anti-virus.pdf.

[13] Kivity A, Kamay Y, Laor D, et al. KVM: the Linux virtual machine monitor[C]//Proceedings of the Linux Symposium. Ottawa: USENIX, 2007: 225-230.

[14] Uhlig R, Neiger G, Rodgers D, et al. Intel virtualization technology[J]. Computer, 2005, 38(5): 48-56.

[15] Bellard F. QEMU, a fast and portable dynamic translator [C] // USENIX Annual Technical Conference, FREENIX Track. Berkeley: USENIX, 2005: 41-46.

[16] McLoughlin M. The QCOW2 image format[EB/OL]. Orinda: GNOME Foundation, 2008[2017-03-24]. https://people.gnome.org/~markmc/qcow-image-format.html.

[17] ClamAV. ClamAV official website [EB/OL]. [2018-1-6]. https://www.clamav.net/.