天翼云终端杀毒

技术白皮书

地址：北京市东城区青龙胡同
邮编：100007

● 版权声明

修订记录

版本	状态	修订理由和内容摘要	修订人	批准人	修订日期

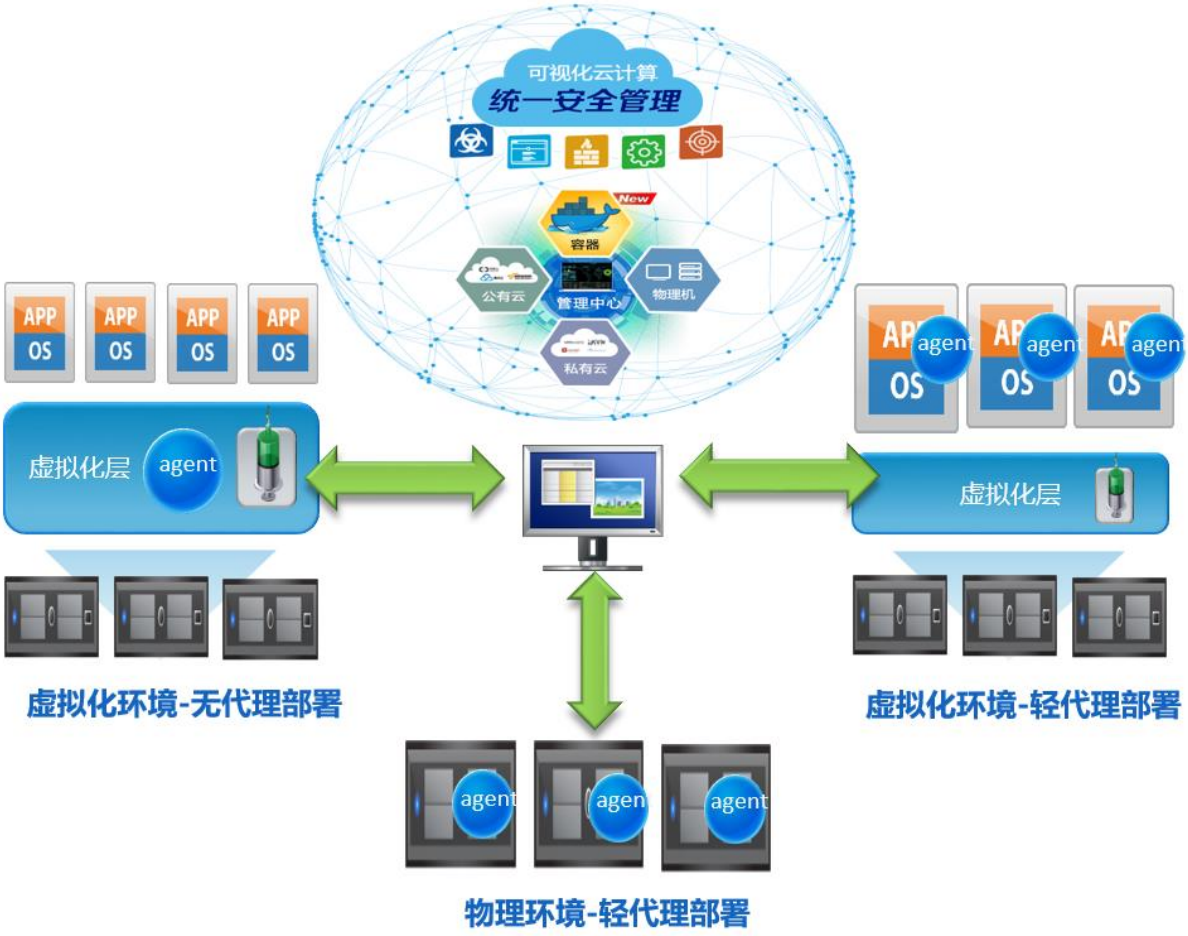
状态：C-创建，A-增加，M-修改，D-删除

目　录

1 主要核心技术

1.1 多种部署方式

随着虚拟化场景的多样化，传统的部署方式需要在各个主机上安装代理客户端，对主机资源的占用，容易出现效率问题。天翼云终端杀毒针对多种虚拟化平台，提供了有代理和无代理的部署方式。



有代理部署即在每个虚拟机安装杀毒引擎，和传统的物理环境部署类似。

无代理部署即在宿主机或者安全虚机部署杀毒引擎，在虚拟机上不需要部署或者只需要部署很轻量的消息中心。减少对主机资源的占用，不仅提升了运行效率，同时能够有效保证主机的安全。

1.2 多引擎扫描技术

随着黑客攻击手段的不断进化，新兴病毒样本成指数倍增长，传统杀毒引擎已疲于应对。天翼云依靠多年在杀毒领域的技术积累，自主研发了云查杀引擎、QOWL 查杀引擎、QDE 深度学习引擎，可对各种新兴变种病毒进行有效查杀与隔离。

其中，云查引擎依托于天翼云云端超过 200 亿条样本，进行家族类可视化分析，可对未知变种病毒实现精准的查杀与隔离；云查杀引擎的使用还可以根据企业的网络环境自由选择，终端无法连接天翼云云的情况下也可以选择通过管理中心或者独立的代理服务器代理到天翼云云，企业内部网络和互联网完全隔离的情况下还可以使用专用的私有云安全鉴定中心来保障安全。

QOWL 是天翼云自研的本地查杀引擎，除了对木马类病毒、脚本病毒、感染型病毒、宏病毒等多种病毒的查杀能力外，还具备丰富的格式识别和解析功能、脱壳功能，以及 CVE 病毒检测和启发式检测功能。QOWL 引擎资源占用极低，内存平均占用 30MB，单文件扫描平均毫秒级即可完成，同时病毒库支持细粒度更新，可最小化升级的带宽占用。QOWL 引擎同时支持 Windows、Linux、MacOS、国产化等终端平台， 在天翼云用户的千万级终端上经过了验证。

QDE 深度学习引擎基于天翼云安全大数据平台对样本数据进行预处理和统计分析，形成科学的模型评估和设计过程，具有海量的、带优质标签样本数据集为深度学习系统赋能，无需频繁升级特征库，就能快速识别鉴定多种未知病毒和变种病毒。QDE 引擎基于自主建构的 AI 计算平台进行模型的训练和调优，使用多范式的样本向量化抽取算法和混合机器学习模型处理多源恶意软件，具备优异的性能检测指标。

这些查杀引擎在运行时可进行数据交互，对终端进行扫描结果缓存共享，在整个数据中心进行增量扫描从而提高扫描效率。

1.3 多维化安全管理

天翼云终端杀毒对操作系统中关键点实时监控，结合上下文对具有可疑行为的操

作主体进行智能判断，来有效发现威胁，判断过程可以综合使用多种杀毒引擎，有效发现未知威胁。安全管理系统包含：

防恶意软件：对操作系统中进程、文件、注册表、驱动程序等关键位置的防护，以及根据勒索病毒特征对勒索病毒的防护。

入侵防御：操作系统或应用程序不能及时打补丁的主机或者还没有对应的安全补丁，经常会面临病毒等恶意软件的攻击，但大量终端的补丁管理很难做到一步到位，并且新发布的补丁与业务系统的兼容性也需要验证的时间；入侵防御模块提供针对这种入侵进行防御功能，避免恶意软件的威胁。

1.4 文件缓存技术

防恶意软件相关功能都需要对文件进行读取，如病毒扫描时需要计算文件的全文哈希或根据特征描述对文件的各种偏移位置进行扫描，主动防御还需要获取文件的数字签名、资源属性等信息，对这些信息的重复获取会造成大量磁盘 IO，天翼云终端杀毒开发了文件缓存技术来解决这种问题。磁盘上的每个文件第一次被检测时，计算上述所需的信息并保存到本地，下次再扫描同一个文件时，根据文件的关键特征判断文件是否发生变化，如果未发生变化则直接返回缓存的属性信息，否则才重新计算，磁盘文件发生异常变化时自动将缓存置为无效。由于大多数环境的文件变化并不频繁，使用缓存技术后病毒扫描和实时防护的性能可达到显著提升。

2 产品主要功能模块说明

2.1 防病毒模块

2.1.1 功能简介

随着黑客攻击手段的不断进化，新兴病毒样本不断增长，以及攻击类型的不断变化，传统的病毒查杀引擎已经疲于应对。

天翼云终端杀毒支持对病毒以及网络攻击进行防护，集成云查杀引擎、QOWL-启发式恶意代码查杀引擎、QDE-人工智能引擎、BitDefender 杀毒引擎，可以对各种变形病毒、网络漏洞攻击等威胁进行有效的查杀和防护。

天翼云终端杀毒防恶意软件组件，通过在终端机器上安装终端安全代理软件或在平台上安装无代理组件，能够有效的对终端进行安全加固，提供防护，抵御外来的攻击。

管理员可以通过管理中心对终端进行统一的病毒查杀管理，定制定时查杀任务，辅助实时防护，主动防御以确保终端安全。

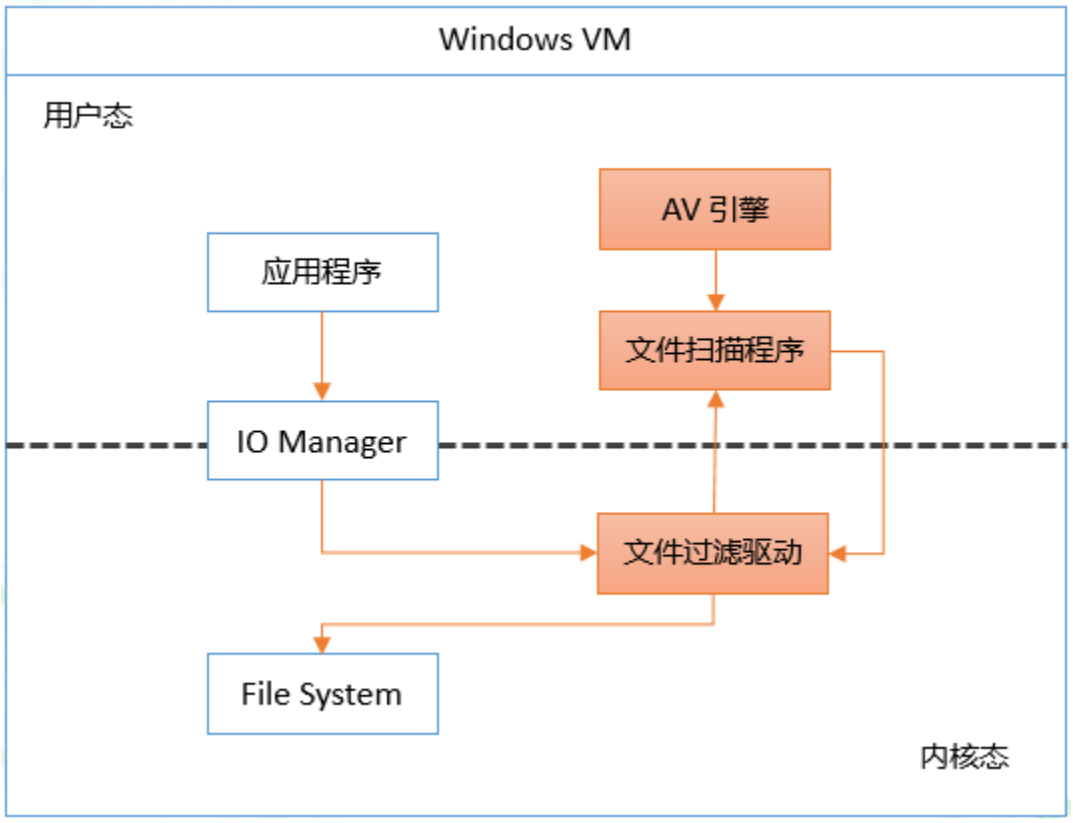
防恶意软件的安全防护分为两个方面：病毒扫描、实时防护。

病毒扫描：通过客户端的程序对终端的文件进行扫描，对威胁文件进行鉴定识别处理。在扫描过程中，可以同时启用云查杀引擎、QOWL-启发式恶意代码检测引擎、QDE-人工智能引擎、BitDefender 杀毒引擎，全方位扫描文件，从而达到全面查杀终端威胁的作用。同时会定期对系统中的 powershell 计划任务进行检测，查杀可能已经存在的 powershell 病毒。扫描时可以配置终端资源占用率，平衡型（默认）表明扫描时候调动 CPU 使用率，将 CPU 占用率控制在 50%以内。若企业终端配置较低可以选择低资源，保障扫描时限制 CPU 的使用率最高不超过 25%。若企业为中高配机越高，则扫描的速度越快。

实时防护：在文件被访问时对文件进行扫描，如果发现病毒，则会锁定文件并及时通过弹出提示窗口告警用户，迅速处理。如果设置默认的处理模式，则会安装设置的处理规则对病毒进行处理，并通知用户。通过设置诱饵文件，当诱饵文件被修改时，检测修改的进程，从而能够及时发现勒索软件对文件的加密操作，终止勒索进程，并查杀勒索病毒，防止客户的文件被篡改，保护文件数据的安全。并对系统注册表的关键位置进行监控，如果遇到恶意软件的操作，则对其进行拦截。

2.1.2 实现原理

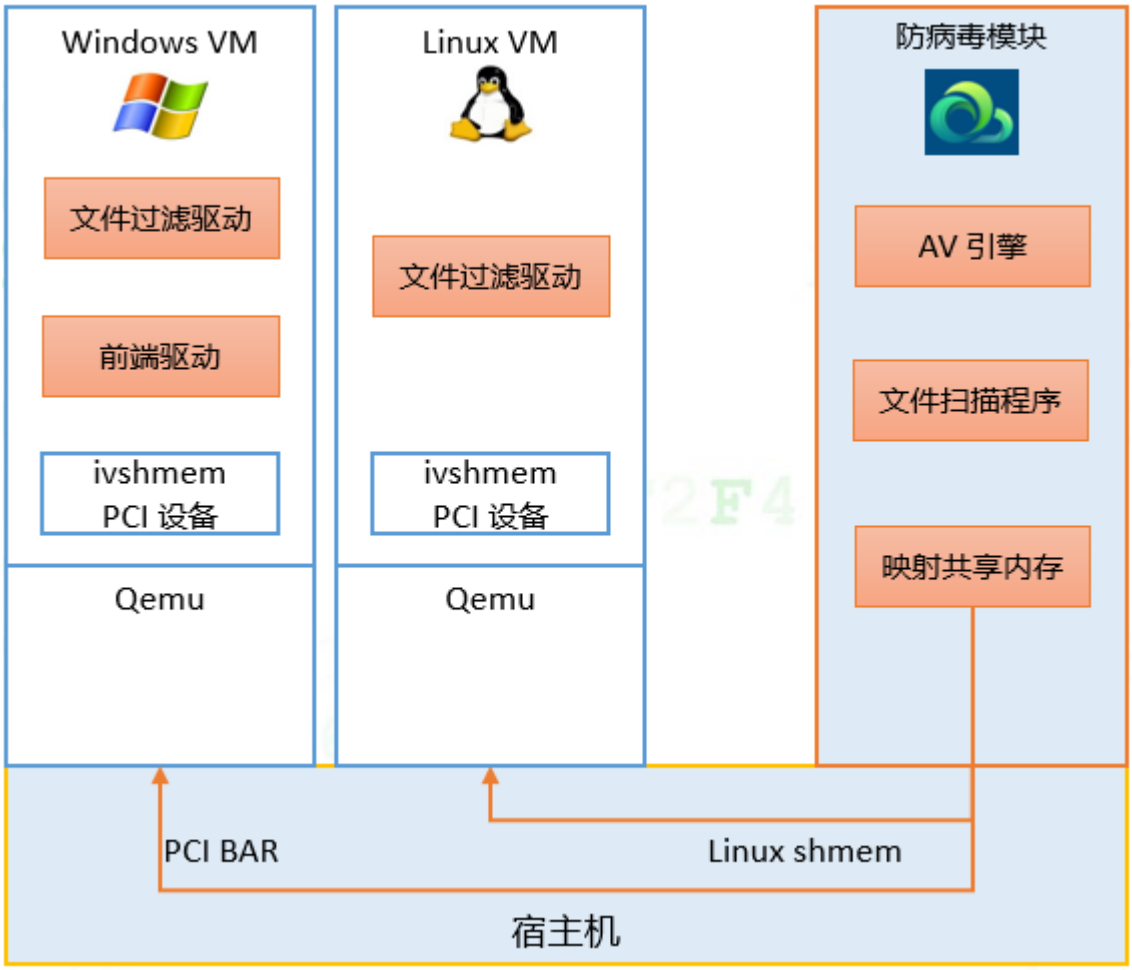
2.1.2.1 有代理实时防护



Windows 和 linux 实时防护的原理类似，都是通过内核拦截文件操作进行转发，再进行检测。

这里详细介绍下 windows 实时防护原理：用户态应用程序操作文件时，通过 IO 管理器将请求发送给文件系统，防恶意软件模块的文件过滤驱动在 IO 请求到达文件系统前捕获应用程序的该请求，并将该文件信息发送给用户态的文件扫描程序，文件扫描程序获得文件后调用杀毒引擎对文件进行检测，如果是病毒则将病毒文件按照管理中心设置的规则对病毒进行处理，处理完成后将数据返回给文件过滤驱动，文件过滤驱动按照返回的数据对文件拦截或放行。在文件执行前对文件进行处理，从而有效维护了系统的安全。

2.1.2.2 无代理实时防护



无代理实时防护原理：在虚拟机主机上，文件过滤驱动获取到系统的文件操作，通过 PCI 将文件数据写入共享内存，宿主机通过共享内存获取文件数据，并通过映射共享内存将文件数据传递给文件扫描程序，文件扫描程序获取文件信息后，调用杀毒引擎对文件进行检测处理，并将检测后的数据通过共享内存返回给文件过滤驱动，文件驱动再根据检测结果对文件拦截或放行。

2.2 入侵防御模块

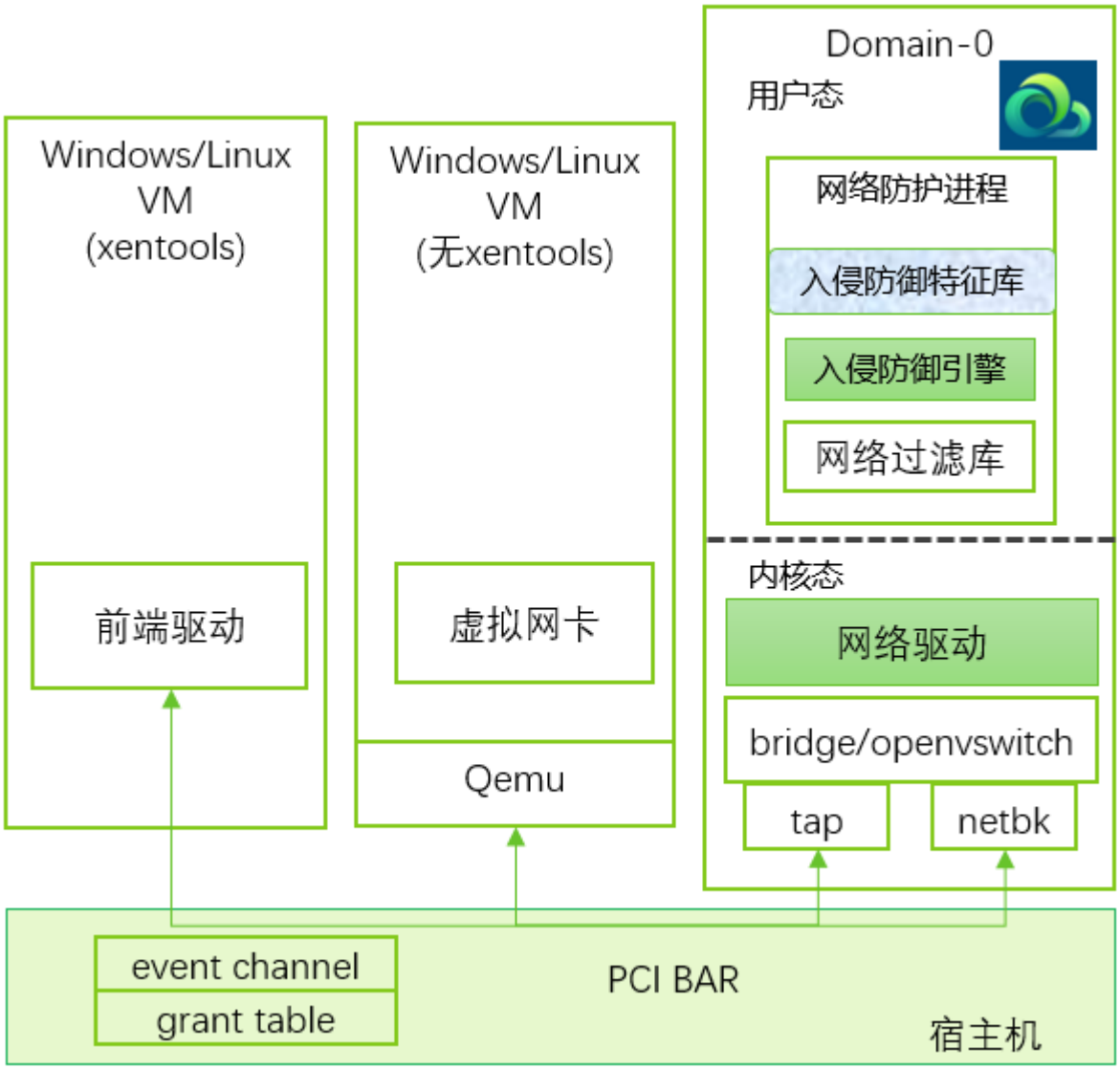
2.2.1 功能简介

操作系统或应用程序不能及时打补丁的主机或者还没有对应的安全补丁，经常会面临病毒等恶意软件的攻击，但大量终端的补丁管理很难做到一步到位，并且新发布

的补丁与业务系统的兼容性也需要验证的时间，此时系统就存在一定的安全风险。终端杀毒的入侵防御模块提供针对这种入侵进行防御功能，避免恶意软件的威胁。

2.2.2 实现原理

入侵防御模块通过网络驱动获取网络数据，并调用入侵防御引擎进行检测。入侵防御引擎将获取到的流量数据与入侵防御特征库进行匹配。如果发现存在威胁，则将该流量拦截，从而保护服务器的安全弱点不被不法分子所利用。



3 产品安全性说明

3.1 管理中心

3.1.1 网络通信的安全性设计

3.2 客户端

作为一个安全产品，天翼云终端杀毒客户端在设计之初就考虑了安全问题，主要表现在：

3.2.1 客户端的自我保护能力

天翼云终端杀毒有自我保护驱动，用于保护客户端不被木马或别的恶意程序破坏。并拥有防卸载保护功能，能够保护客户端被卸载。目前拥有以下自我保护能力：

1. 受保护的进程，必须是位于天翼云终端杀毒安装目录下，并且 EXE 的天翼云 S3 签名有效；
2. 保护天翼云终端杀毒安装目录下的文件不被外部程序删除、修改，只能被受保护进程修改；
3. 保护天翼云终端杀毒注册表不被外部程序修改、删除，只能被受保护的进程修改、删除；
4. 防止保护的进程被其他程序挂起线程、注入、强制结束；
5. 防卸载功能可通过管理中心设置卸载校验码，防止客户端被随意卸载；

3.2.2 内核对象保护

对于一些重要的内核对象，例如天翼云终端杀毒客户端中使用的一些互斥量、事件等内核对象，名称里会加上特殊的前缀，自我保护驱动会对这些名称的内核对象做保护，第三方进程无法创建出这些被自我保护驱动保护的内核对象。

4 产品可靠性说明

4.1 管理中心

4.1.1 微服务架构的可靠性设计

4.2 客户端

4.2.1 升级

升级功能在客户端可靠性保证中起非常大的作用，万一产品由于 Bug 导致不稳定，或者有安全缺陷等问题，都是通过升级的方式把修复好的文件分发到客户端上。

4.2.2 日志

客户端会保存一些运行过程中产生的日志，日志里一般会记录一些程序运行过程中的值，供定位问题。

4.2.3 重要模块使用独立进程

对于一些重要模块，例如安全防护模块，使用独立进程启动，进程内只做安全防护这一件事，避免由于别的模块缺陷导致整个进程崩溃。