



虚拟化数据中心的安全问题

序言

应用和操作系统虚拟化所包含的概念并非新概念，它们在服务器设备和桌面PC得到广泛普及的很长一段时间之前就已经出现。然而在过去几年内，虚拟化普及的速度（尤其是软件操作系统虚拟化）迅速提高。据EMC®公司CEO Joe Tucci称，大部分VMware®客户已经计划在未来三年内实现其50%的IT基础设施的虚拟化²。虚拟机最终盛行起来，并且正迅速进入企业数据中心，成为各个地方的IT部门内所有人员和小组的通用工具。

那么，到底什么是虚拟机呢？VMware对虚拟化的定义是“将物理硬件与操作系统分离的一个抽象层...”³ 当前，我们通常会认为虚拟机属于运行多个软件操作系统的硬件平台的范畴。最常见的是，这个概念通过一个硬件设备（主机平台）上一个操作系统而实现，即在虚拟机平台（客户端）上按顺序运行多个独立操作系统。平台虚拟化一般依赖于全面的硬件分区：允许每个客户端平台接入物理主机硬件的特定部分，而不会与主机平台产生冲突，或者对主机平台造成影响。例如，即使客户端操作系统与主机系统需要一样的CPU和RAM存取，客户端将使用与主机不同的硬件位置和地址。这允许主机和客户端依次运行，而不会互相冲突。

平台虚拟化主要有两种类型：透明和主机感知（通常称为半虚拟化）。对于透明虚拟化的实施，客户端不知道它在虚拟化状态下运行。客户端消耗资源，就如同在本地运行于硬件平台上，有一点非常明确，即它由额外的一个组件进行管理，叫做VMM（虚拟机监视器），也叫Hypervisor。当前，更多标准形式的虚拟化（例如VMware的系统）实施透明的Hypervisor系统。这些系统可以被认为是代理：Hypervisor将透明地代理客户端和主机硬件之间的所有通信，向客户端隐藏其存在，这样，客户端就会认为它是唯一运行于该硬件上的系统。

主机感知的实施有所不同，即客户端的内核中嵌入了某种类型的虚拟化知识；这些可以被视为“虚拟自我感知”环境。客户端操作系统内核的某些部分知道Hypervisor的存在，并与其直接通信。客户端操作系统并非透明地代理所有通信，而是直接调用Hypervisor，并管理对硬件的通信。Xen（发音是'zen'）作为针对Linux的常用虚拟化实施系统，它采用主机感知架构，要求特殊的Hypervisor命令代码主动地运行在主机以及所有正在运行的虚拟化客户端中。每种虚拟化类型各有优缺点，但同样有效。透明系统对客户端来说最便携，但牺牲了速度，而且一般围绕更重的Hypervisor而设计；主机感知系统速度更快，重量更轻，但需要对客户端进行修改，而且可能引入透明系统所没有的安全性问题。

推动虚拟化普及的一个因素是硬件对VMM硬件平台支持的开放性特点，它运行并管理主要的主机操作系统，而且VMM不是专业的设备或装置。虚拟主机平台可以是当前所用的任何类型的硬件：单CPU台式机、笔记本电脑、x86服务器、SPARC服务器、机架安装式设备等。在笔记本电脑上运行Microsoft® Windows® XP Professional 的普通用户可以运行其它通信的多个虚拟化事例—例如Linux、BSD或Windows Vista— 可以使用任意数量的免费VMM软件。这种灵活性（虚拟化软件向常用硬件的转移）使每个人都能以经济的方式直接接入，以运行虚拟化环境。尽管这种接入方式最初是提供给专业技术人员，例如需要将Windows作为基础操作系统运行的Unix用户，但很快成为了IT经理谈论的话题。平台虚拟化为充分扩展服务器组和数据中心提供了一种经济的机制。虚拟化允许一个公司采购一台高端硬件设备运行20个虚拟操作系统，而不必采购20台商用的低端设备，分别运行每个操作平台。

虚拟化的威胁

虚拟化的好处显而易见：为您的人员提供更大的能力。但凡事各有利弊，虚拟化也不例外。有利的方面是主要的，为多个项目和环境提供更快的速度和灵活性。但不利的方面不很明显。用一台服务器的价钱运行20台服务器有什么不好？尽管目前尚未被视为重大的威胁，但虚拟机和环境的安全一般并未被考虑，并不是因为这些实施项目的安全性有技术难度，而是因为安全问题是实施大范围虚拟化的小组所未知的领域。换句话说，实施虚拟化时一般没有考虑它所带来的新安全风险。从表面上讲，虚拟化的BSD客户端与真正的单个设备具有同样的安全威胁和问题，这一点毋庸置疑。然而，主要的区别还在于额外的管理层：Hypervisor。

“数据不会在虚拟机之间泄露，而且应用只能通过配置的网络连接进行通信。”

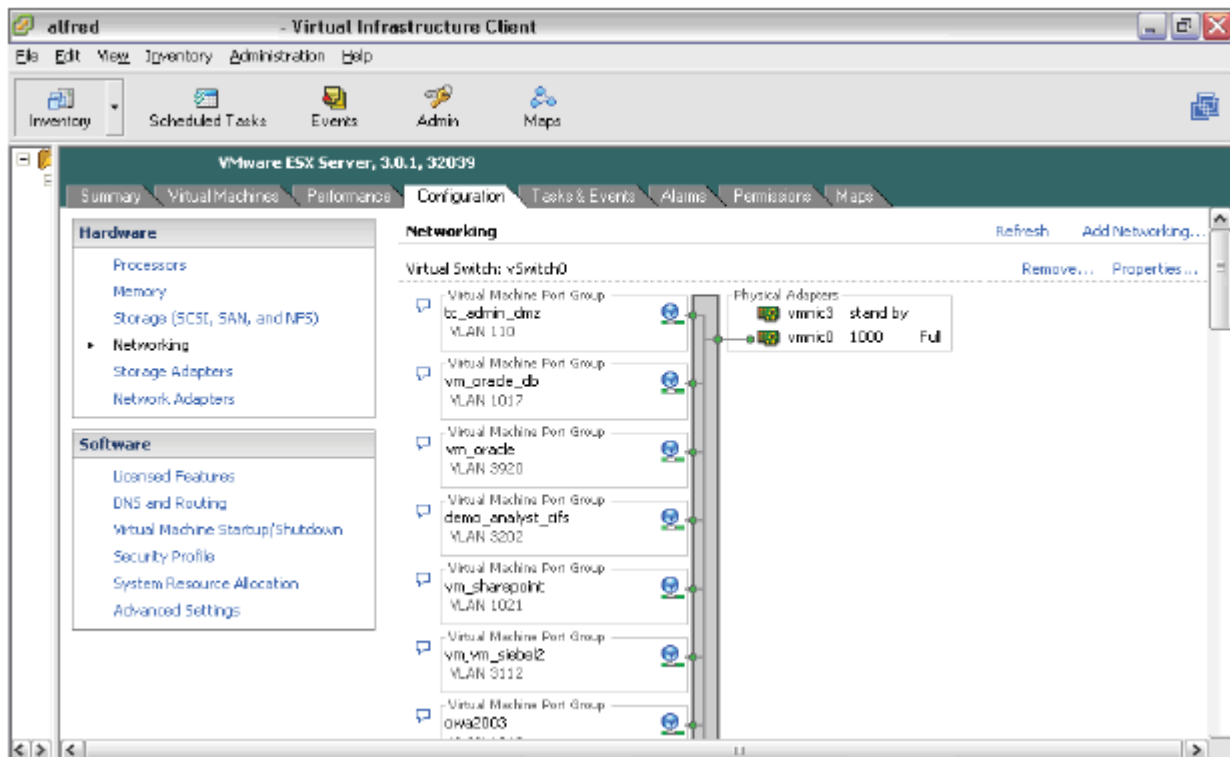
- vmware.com¹

虚拟化数据中心的安全问题

Hypervisor实际上是另一个操作系统，它管理主机OS和客户端OS之间的通信。管理员不用担心单个设备上的单个BSD，而是必须关注三个操作系统的安全。例如，如果您保证虚拟BSD客户端安装的安全，但未考虑主机和VMM的安全，则会忽略关键的组件。如果主机受到安全威胁，客户端设备上的安全则与此无关。

尽管Hypervisor是虚拟化的“主控制器程序”，但它不是具有安全风险的唯一虚拟化抽象层。对于任何虚拟化系统，另一个关键方面是网络层。对于不同的虚拟化软件平台，构建和处理客户端与主机间的联网的方法也不同，包括从Qemu⁴最基本的通用网络仿真到VMware极为复杂的专用软件交换平台。虚拟化联网一般被认为“仅能运行”，应考虑虚拟机和环境安全研究的一个最重要方面。与硬件网络设备不同，基于软件的网络带来了在硬件中一般看不到的安全问题。例如，最近，VMware的客户端联网子系统在“仅集线器”模式下运行，即在同一个软件网络域内，同一个物理主机上的多个客户端实例可以公开地自由访问该域内共享的所有网络数据。它是通过软件实施的标准集线器。如果两台客户端都是同一个仅主机网络中的成员，并共享同一个虚拟接口，两台机器可以看到主机和客户端之间的所有流量。对于VMware的桥接模式配置，目前仍是这种情况，即所有客户端映射到同一个物理网络设备。尽管在更健壮的配置中设置了屏障，例如采用多个VMware软件联网设备（即vmnet0、vmnet1、vmnet2等），但整个网络管理子系统仍存在于软件中。采用多个虚拟网络接口分割流量相当于将两台机器插入同一个客交换机中，分别具有唯一的子网，但都没有被VLAN分割。

VMware也是软件网络安全迅速发展的例子。VMware在最新的企业版本中创建了极为健壮的软件交换网络，正在向虚拟交换（从2层VLAN到3层路由）的仅软件分区模式发展。图1表示具有多个虚拟VLAN分区的VMware ESX虚拟机交换机（vSwitch0）。如图所示，'vm_oracle'和'vm_sharepoint'在不同的VLAN上，但两个网络都是同一个虚拟交换机的一部分，而且其流量要通过同一个物理网卡。但即使这些进步都考虑了安全性，软件交换仍然比硬件的风险大。在真正的网络环境中，相邻距离通常是攻击的一个重要屏障，与此不同的是，对于一台主机上的所有客户端，其它所有客户端和主机之间共享同一个软件堆栈。网络堆栈共享是虚拟化安全的一个主要问题；如果所有客户端和主机共享一个网络堆栈，则攻击者只需攻击一台机器即可接入整个堆栈。例如，在VMware网络子系统中的一个简单攻击方法是：在主机软件交换网络驱动器处理VLAN标签和2层帧大小的过程中有一个漏洞—可允许客户端上的攻击者看到所有网络数据从客户端->客户端、客户端->主机传送。即使在主机和物理网络之间交换的数据也可能暴露于风险中，无论在多个客户端之间是否有软件分区。





虚拟化数据中心的安全问题

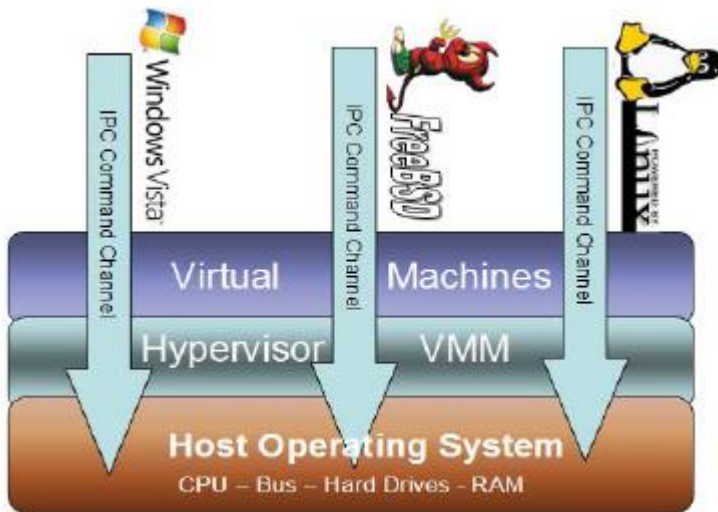
最重要的是，平台虚拟化的范例需要我们将我们引导到新类型的管理员：系统、网络和安全管理员应进一步扩展他们所掌握的知识，了解虚拟化带来的新概念。网络管理员花费多年时间积累知识，但这些知识都是针对现实世界，即固态的交换机、电缆、CAM表以及受邻近性限制的VLAN。在虚拟世界中，不仅所有这些概念从硬件转向软件（许多情况下是从软件内核领域向用户领域转移），而且这些概念一直处于混乱和“封闭状态”，不适用于传统接入方法。管理虚拟交换网络的系统管理员再也不能使用简单的工具进行监控和故障排除。管理员不能走近虚拟交换机，插入笔记本电脑，添加一个网络分路器，可靠地对一个端口进行映射，或者查看虚拟设备的统计信息。所有这些能力和知识已经超出专业管理员的能力范围，而且对软件控制器、管理GUI、专用内核模块和二进制系统隐藏，最重要的是，已经进入了只有设计人员和开发人员知道如何真正管理设备的时代。设想一下如果一个人需要对10位架构师和设计师构建的CheckPoint® 防火墙或Cisco® 交换机进行故障排除。这是一种可怕的情况。

新的安全法令

我们已经知道，在安全方面，我们不能对虚拟化想当然，而且应该比物理和专用操作系统及设备的日常威胁更加警惕。但是，“实现绿色化”有哪些需要特别关注的方面？下面通过几个实例说明虚拟安全的现实影响：

攻击管理接口：主机

所有虚拟操作系统环境都有某种形式的软件管理控制接口负责管理客户端和主机间的IPC（进程间通信）呼叫。这一般通过主机操作系统上的本地进程管理，由Hypervisor控制。这些管理应用允许主机上的进程监控客户端，并与客户端直接交互；否则，主机将无法管理（或控制）客户端。这种架构的最佳例子是VMware的vmware-tools和Xen的客户端内核模块，两者都允许各自的主机应用通过Hypervisor控制客户端和环境。



曾运行VMware的任何人都使用过并了解vmware-tools。这个应用使VMware能够直接从主机通过内核级IPC应用和服务接入客户端。一般情况下，vmware-tools用于从主机到客户端的非仿真硬件接入，例如从主机向客户端装载CD-ROM（或ISO文件），并为USB设备和鼠标等外设提供更快的硬件接入。VMM/hypervisor也采用同样的通信模式管理主机和客户端之间的用户交互，例如控制鼠标在管理GUI中的虚拟事例上的移动。

我们看一下使用vmware-tools从主机向客户端上的虚拟CD加载ISO文件的情况。首先，主机上的VMware进程必须能够接入ISO文件。这一般不是问题，因为VMware进程作为一个高级用户运行，具有较高的权限（Linux机器中的“根用户”，或者Windows机器上的“系统用户”）。接下来，客户端上必须有一个进程知道如何与主机上的进程通信。这要求在客户端上安装软件，通过Hypervisor进行命令调用而与主机VMM通信。客户端管理子系统向主机发起呼叫，一般通过指定的管道发起，要求主机代替客户端启动硬件调用，“欺骗”客户端认为它正在通过物理驱动器接入物理CD。这是VMware通过实施vmware-tools从主机向客户端加载CD的基本例子。这个过程很简单，但对安全性极为重要，因为我们创建了一个在主机和客户端上作为超级用户运行的未检查的系统，可以被恶意攻击者操纵和利用。

再看一下现实世界的例子：运行10个Windows 2003虚拟事例的Linux VMM主机。每个虚拟主机都是运行财务部会计Web应用的冗余HTTP服务器组的一分子。每个虚拟客户端上的内容和数据在虚拟驱动器之间复制，而且始终保持完全一样。无论最终用户接入哪个HTTP服务器，后端数据始终保持一样。用户不知道自己在任何特定时间接入多个服务器，也不知道这些Web服务器都是虚拟服务器。例子中的这家公司没有为这些数据构建一个完全冗余的后端数据块，而是构建了一个极为低价的虚拟基础设施来管理这些财务数据的分发。



虚拟化数据中心的安全问题

每星期，这家公司的Linux主机上的计划任务加载一个新CD，其中包含这星期的财务报表。每个Windows虚拟客户端上都设定了计划任务，以拷贝虚拟加载的CD上的这些财务报表（记住，通过vmware-tools加载），允许在财务服务器之间复制数据。Alice，恶意攻击者，采用弱密码攻击和她最爱用的本地权限提升漏洞——ELF 'uselib'漏洞——去攻击Linux主机，以获得对该主机的全面根用户级远程接入权限。虽然她的主要目的是窃取每周的财务数据进行勒索，但她很快认识到，她有机会威胁数据和包含数据的应用。

Alice有大量的攻击和选项来攻击客户端系统。例如，她可以用包含AutoRun功能的一个文件替换ISO文件，迫使虚拟CD上的内容在Windows客户端加载时立即运行。然后，她只需等待ISO在指定时间加载到客户端上，在客户端上运行她所希望的任何恶意应用程序，例如攻击工具、病毒、窃听程序，或者可能是netcat程序。利用netcat，她可以在任何时候直接接入客户端系统（远程接入）。这种延迟接入使她能够偷偷地在客户端Web服务器上实施破坏性报复行动，窃取有权登录到服务器请求财务报表的所有主管人员的密码，然后将证书数据传回位于西弗吉尼亚的森林中的隐身之处。最初简单的数据拦截攻击可能很快演变为对整个虚拟化数据基础设施的全面攻击。

通过客户端攻击Hypervisor

避开虚拟机

以上面的架构场景为例。但这次，Alice攻击的是VMware客户端，而不是主机。即使没有做到，Alice也攻击了一个虚拟系统，而不是物理系统。从表面上讲，攻击产生的影响非常相似。但对于客户端HTTP服务器，Alice能够访问她正在寻找的财务数据，在这种情况下，她的攻击仅限于这一个客户端；她并未认识到自己在一个虚拟服务器组中的一个虚拟机上。在她能够对整个虚拟基础设施发起攻击之前，她1) 需要认识到自己位于由主机VMM管理的虚拟机中，而且2) 避开该虚拟机。一旦她避开了虚拟机，就可以接入基础设施的其它部分，继续实施她的破坏计划。

要避开虚拟机，需要依次迂回进入硬件和操作系统设计。在基于x86的CPU，ring 0（通常指内核模式）是CPU的一部分，在这里管理内核级进程。同样，ring 3（或用户模式）是对用户级进程分配处理空间的地方。虚拟化操作系统要求ring 0接入CPU，就如同真正的本地安装的操作系统。物理和虚拟操作系统都需要一定数量的内核代码（例如中断表格和视图）在ring 0中运行，并且始终知道这些代码在RAM中位于何处。与物理操作系统不同的是，虚拟客户端不可能像独立的机器那样，将中断表格放置在RAM中的同一个位置；主机的中断表格已经占用了内存中的该位置。因此，尽管客户端认为其中断表格可能位于其虚拟RAM中的0x0000ffff处（表格始终存储在这个特定操作系统中），但主机实际上已经通过透明的Hypervisor将这个位置映射到物理RAM中的0x1234abcd位置，对虚拟系统隐藏了实际位置。每次当虚拟客户端操作系统需要参考0x0000ffff处的中断表格时，Hypervisor透明地将呼叫转换为0x1234abcd处的实际位置，然后在指令结果返回到客户端时再转换回来。客户端不知道Hypervisor已经处理了中断表格的位置。然而，这是一个内核级空间，因此，即使主机Hypervisor已经针对客户端处理了中断表格的实际位置，虚拟表格仍必须从实际CPU上有权限的ring0运行（并驻留在其中）。

大多数情况下，任何针对客户端虚拟CPU的攻击很有可能造成客户端瘫痪，这实质上是虚拟DoS。这与通过主机操作系统利用物理CPU中的CPU微代码并没有太大不同，但直接向CPU注册表读写信息并不简单。对于虚拟化环境，情况有些不同。现在，不仅CPU寄存器本身实现了虚拟化，而且还包括更多的方面。如上文所述，这些方面的大部分都对用户隐藏。例如，VMware不向vmware-tools或Hypervisor公开其源代码，因此，一般用户也无法知道当客户端向Hypervisor请求其中断表格中的数据时，在VMware的虚拟ring 0中发生了什么。我们知道，VMM翻译该呼叫，并通过IPC将该呼叫交付给运行在主机上的超级用户，但问题就在于此。如果攻击者找到利用虚拟微代码的方式，则他们可能会操纵主机内核和CPU。这叫做虚拟机躲避：跳出虚拟环境的限制，并进入物理环境。

虚拟机检测

在攻击者发起对虚拟环境的攻击之前，攻击者必须知道虚拟机在何处，以及他目前是否在一个虚拟机上。如果可以从本地接入平台，例如通过SSH接入控制终端，就会有信号指出机器已被虚拟化。MAC地址、进程列表、机器上安装的文件、驱动程序等容易检测到的项目，只需敲击几下键盘即可访问。然而，除了这些基本项目之外，有许多攻击可帮助攻击者检测机器是否运行在虚拟环境中，有时甚至返回关键信息，例如客户端的中断表格的当前位置。两种常用的虚拟检测工具是redpill⁵和scoopy do⁶，每种工具都可以通过CLI在后台默默地运行，就好像不会制造怀疑，然后利用netcat等工具通过网络隧道交回攻击者。攻击者可在他们破坏的每台机器上利用这些工具，创建一个已知虚拟环境的列表，以发起有针对性的攻击。

虚拟化数据中心的安全问题

在知道机器是否有虚拟CPU之前，没必要浪费时间利用具有破坏性的微代码尝试攻击一个虚拟CPU。redpill等工具是全面虚拟机攻击的第一步。

攻击客户端和避开虚拟机已经真正成为针对虚拟环境发起攻击的制胜法宝。如果或者当针对虚拟机的攻击随时可以发起时，攻击者只需花时间攻击一台虚拟机，这样就可以破坏一个闭合网络中的其它虚拟机，并最终避开虚拟VMM环境，接入主机。记住：虚拟机必须考虑其内核运行在CPU和RAM中受保护并且有权限的空间内，而且必须由Hypervisor授予接入主机上这些物理位置的权限。但虚拟化并不是一个内核运行在一个CPU平台上，而是要求多个内核共享接入并相互交互，即使虚拟化并没有实现这一点。如果攻击者的目的是攻击尽可能多的机器，而且攻击者知道某个系统组全是虚拟系统，则基于Hypervisor的攻击将提供最有利的攻击池。这种多个虚拟内核之间受保护的接入共享可能为所有类型的攻击打开了方便之门，例如：

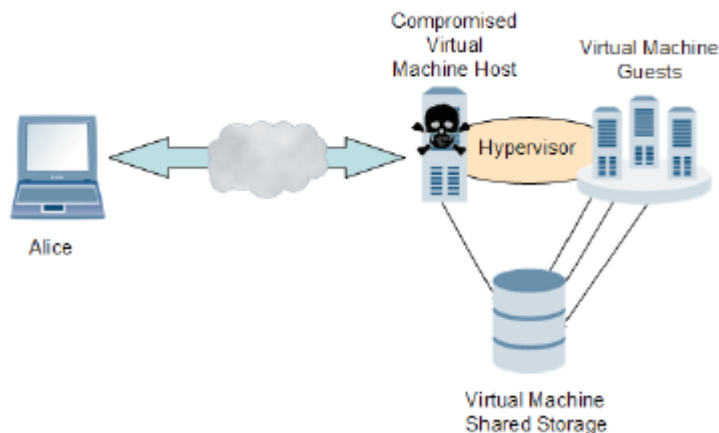
- 将破坏性的微代码从虚拟CPU传送到物理CPU
- 向虚拟机添加特洛伊木马，从Hypervisor传送到主机，在主机的ring 0中安装并运行resident"
- 操纵虚拟机管理接口本身，越过客户端和主机之间的验证
- 采用Hypervisor接入软件网络子系统，并且使客户端和主机网络接口进入混乱模式，以通过客户端透明地探查主机网络
- 利用虚拟硬件驱动器操纵主机上的驱动器
- 可能最具破坏性且最有利可图的就是攻击和操纵Hypervisor本身。如果攻击者可以操纵并控制多平台虚拟化主机上的Hypervisor，则攻击者就可以控制每个接入Hypervisor的客户端的所有软硬件命令。这种攻击相当于拥有了数据中心内的每一台服务器，可深入到CPU和总线级别。它是整个数据中心上的核心工具包，可以在主机和每台客户端之间转换核心内核操作。

现实世界攻击举例

尽管所有这些情况可能听起来有些极端、复杂，而且太过于理论化，但任何虚拟化系统所具备的最基本的安全威胁是大多数法则。在典型的攻击场景中，攻击者必须一次针对一台机器进行攻击，而无论其意图是什么。当然，攻击者可以利用僵尸网络或机器人网络同时攻击多台机器，但这些攻击仍是一对一的关系。攻击一台机器对该机器进行破坏。虚拟化环境消除了这种限制，并且产生了一对多的攻击情况：通过攻击主机拥有客户端的权限。甚至是攻击一台客户端，然后可能拥有全部客户端的权限。

从追攻击虚拟磁盘

我们使用前面的例子：攻击者Alice破坏了一台拥有10个Windows 2003虚拟客户端并运行VMware的Linux主机。Alice的最终目标是单个Linux主机系统中所有虚拟Web服务器上存储的破坏财务数据。Alice知道，她需要破坏每个虚拟机中的这些数据，原因是：正如前面的例子提到的那样，每个虚拟客户端写入自己的本地存储器，然后将数据分发给每个冗余的虚拟存储设备。Alice需要移除客户端中的共享存储器和本地虚拟存储设备，以清除关键财务数据的所有痕迹。在典型的单设备环境中，Alice必须分别接入每台设备，以通过电子化方式破坏每个本地数据分区，这意味着发起繁杂的1:1设备攻击。由于Alice此次攻击的所有目标是一个物理主机上的虚拟机和主机，她可以利用这个虚拟基础设施。所有客户端都使用虚拟硬盘：从每台客户端和主机中访问到的文件相同。



此刻，您可能会想到“当然！在Linux主机上，Alice有权接入vmnet*，它包含管理接口和用于HTTP服务器的VLAN，这样，她能够从虚拟主机网络攻击客户端，或者更简单的方法是在SMB管理员密码从主机上的eth 1向客户端上的vmnet3传送时对密码进行拦截！”，可能您是正确的。但是，Alice的意图只是破坏数据，因此，她会选择阻力最小的途径。她已经拥有了Linux主机的根用户访问权限，因此可以简单地指定cron工作在下一个工作日的2:57 a.m. 运行，命令是：

```
[root@vmhost:/] # for vmdisk in $(find . -name "*.vmdk"~); do dd bs=1024 count=10 if=/dev/zero of=$vmdisk; done
```



虚拟化数据中心的安全问题

一切就是这么简单。在几秒之内，Alice将改写每个VMDK文件（与VMware在虚拟物理硬盘上使用的文件相同）中的前10k，使其变为乱码。在物理磁盘中，磁盘的各部分被分配给引导区和主引导记录，仅删除这些内容一般不会造成磁盘其它部分中的数据无法访问，而VMware一般没有这样的弹性。如果不能理解文件虚拟驱动器的开头，则文件的其余部分就会只是随机数据，而且会永远丢失。

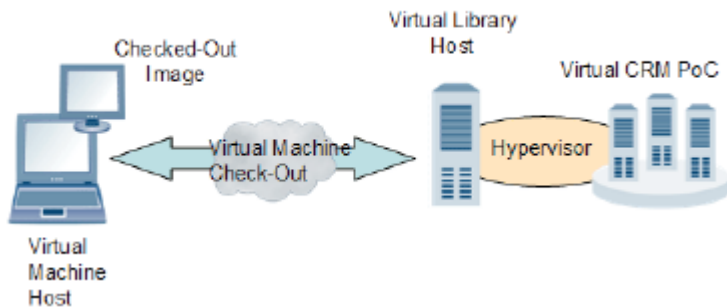
Alice能够通过Linux主机发起一次攻击而破坏10台关键Windows Web服务器中的物理数据。她不必花时间分别攻击每台机器，而且不必知道如何攻击Windows系统。例子中的这家公司自己的虚拟基础设施环境允许Alice通过一次攻击完成她的破坏目的。

种下攻击的种子：虚拟库查验

恶意数据存储攻击是虚拟化本身为聪明攻击者引入的新攻击类型的一个例子，但标准的病毒爆发是什么样，病毒被无意引入到虚拟网络中会发生什么情况？这种匿名工具更为常见，而且是企业网络每天都要面对的。虚拟网络容易给人一种安全风险较低的错误理解，因为大多数管理员认为这些环境被锁定和隔离。我们讨论这样一种情况：不会再伤害某公司，但也不会有帮助。在这个例子中，整个网络在虚拟环境中定义和创建，目的是帮助阻止这些攻击。

某公司正在为未来的CRM系统构建概念证明 (PoC) 数据库环境。一旦完成以后，生产环境将包含多个数据库平台、允许访问数据的Web应用前端以及在全球各地部署的孤立的客户端系统，所有这些都实现虚拟化。在公司花费数千万美元构建整个全球系统之前，公司希望看到一个正在运行的小型虚拟系统。为了模拟这个生产网络，需要在分阶段开发网络中创建10个虚拟系统。公司雇用了一家咨询公司来到现场，建立了虚拟化的数据库PoC实验室。

在几星期内，工作按照预期进展，顾问将笔记本电脑带入并带出隔离的企业网络。在这个CRM PoC环境中，顾问例行地从虚拟基础设施库中“取出”他们处理的镜像，并且在场外对这些镜像进行数天和数周的处理。当他们返回到企业办公室时，承包商将把更新的虚拟镜像重新“放入”基础设施库，从而节省公司现场咨询和差旅费用。这些移动镜像仅允许接入他们所构建的虚拟库PoC环境和网络，而该环境和网络将通过/27网络分区上的分区VLAN进一步从企业网络中移除。公司的IT部门不信任顾问的笔记本电脑——的确是——而且未授权他们（或这些虚拟镜像）接入专用企业网络的任何部分。最后，虚拟化PoC数据库实验室建立完成，可随时交付给IT部门并开始测试。



几天过去了，公司对虚拟PoC库感到非常满意，能够以更低的成本全面测试整个CRM系统，并且消除了曾经影响实际生产环境的缺陷和物流问题。但是，这种喜悦心情并没有长久。在周末，全部新的虚拟镜像意外破坏，使整个虚拟化实验室无法运行。整个星期一，IT人员都试图对镜像进行备份，但收效甚微；大部分镜像（只有一个除外）完全损坏而无法启动；虚拟驱动器文件被破坏。故障排除的时间比预期时间要长，因为无法对硬盘和文件系统采用标准的鉴定；这些系统都实现了虚拟化，而且硬盘上现在只有大的二进制破坏文件。

然而，唯一剩下的镜像处于杂乱状态，尽管可以启动并识别出加载程序，但操作系统无法启动，而且应用程序也不能运行。值得庆幸的是，公司有一位常驻的虚拟机专家，

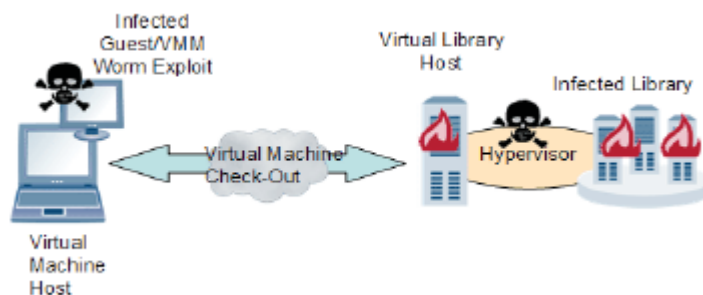
system

她能够将破坏的文件系统加载到新镜像中，对其进行修复，以供进行基础的调查

工作。她还能够确定这个镜像是在完成任务之前放入的最后一个镜像。现实是残酷的。这个镜像含有一个定时炸弹：其中安装并且编程了一些恶意代码，会在以后爆发。一旦自动起爆，这个炸弹将通过自身复制的蠕虫（包含大量的有效负载）从一台机器感染另一台机器。有效负载包含专门查看虚拟环境的工具，并攻击非常著名的虚拟管理机器。这个特定的攻击通过受感染的客户端镜像而攻击主机Hypervisor，使蠕虫能够通过客户端找到并破坏Hypervisor，并使整个库立即陷入瘫痪，破坏Hypervisor，并破坏虚拟基础设施。这个蠕虫的唯一缺陷是不能移除其自身以及携带该蠕虫的镜像，不能完全移除作为其载体的镜像，因为二进制文件必须驻留才能执行攻击。

虚拟化数据中心的安全问题

IT部门认识到，由于网络被完全分区，并与企业网隔离，风险并不是那么高。最后，主机和整个客户端网络必须从头开始构建，迫使项目重新开始，使预算增加一倍。虚拟基础设施对于快速构建孤立测试环境来说获得了成功，但IT部门未能认识到与移动操作系统和虚拟客户端相关的风险。客户端允许在没有任何屏蔽措施的情况下带回家之后再带回来。IT部门有例行程序禁止员工将个人机器带入企业网，将企业机器带回家，所有这些都是出于安全考虑。为什么需要以不同的方式处理虚拟机？遗憾的是，这通常是因为对移动平台可能造成的破坏的了解。与通常被视为应用程序的虚拟设备相比，我们可以更容易地认识到物理移动设备的这些危险，例如笔记本电脑。



结束语

随着企业级虚拟化软件在市场上流行，例如VMware和Microsoft（分别是ESX Virtual Infrastructure和Virtual PC），许多著名公司和支持资金为虚拟数据中心的移植提供了支持。Microsoft的Longhorn平台将配备一个本地Hypervisor，允许通过操作系统直接实现虚拟化，而不需要安装第三方VMM。CPU制造商正迅速实施基于硬件的Hypervisor，例如AMD的Pacifica项目。现在，企业IT部门用一台容纳20个独立操作系统的4U机器全面取代容纳单个操作系统的专用1U机器的现象非常普遍。我们承认，操作系统存在安全缺陷，因此，我们对在网络中部署未打补丁的系统非常谨慎，而我们又盲目地认为VMM及其客户端操作系统是安全的。虚拟操作环境与物理操作环境同样安全的概念可能是代价极为高昂而且极具破坏性的谬论。

这种虚拟化范例的变化引出了一套新的安全问题和风险。安全管理员对“一种产品针对一种应用”环境中的“加固操作系统”、“围墙花园”和“网络分区”等术语比较熟悉，但他们如何将这些概念应用到尚不了解的虚拟数据中心？我们如何在尚不了解的环境中保护自己？当前的系统和安全管理员需要开始关注虚拟安全，为应对分布式和针对性攻击的新威胁做好准备。

虚拟基础设施管理员应了解并且准备应对许多安全风险和考虑因素，其中许多并未在本文中讨论。在向完全虚拟化的环境演进之前，仍有许多问题需要解决，例如：

- 我们当前的分析、调试和调查工具如何适应虚拟化环境？
- 在所有虚拟化平台之间，安全管理员需要掌握哪些新工具？
- 补丁管理如何影响客户端、主机和管理子系统的虚拟基础设施？
- 新的安全工具（例如嵌入CPU中的硬件虚拟化）是否有助于通过移出软件而保护Hypervisor？
- 在实现全面虚拟化时，已知的安全最佳实践（例如非可执行堆栈）如何提供帮助？硬件虚拟化是否能成为真正的安全WMM铺平道路？
- 虚拟化和共享存储：如果我们对iSCSI传输层进行了虚拟化会怎样？我们是否为避开内置的SAN安全打开了方便之门？

这些问题都需要在企业全面向虚拟化环境演进之前给出答案。更重要的是，我们现在就应考虑虚拟化安全会将我们带向何处。我们都承认，虚拟化是为了更好地开展业务并坚持下来，但安全管理员需要保证比威胁先行一步，并在攻击者进行编码之前考虑虚拟化的威胁。