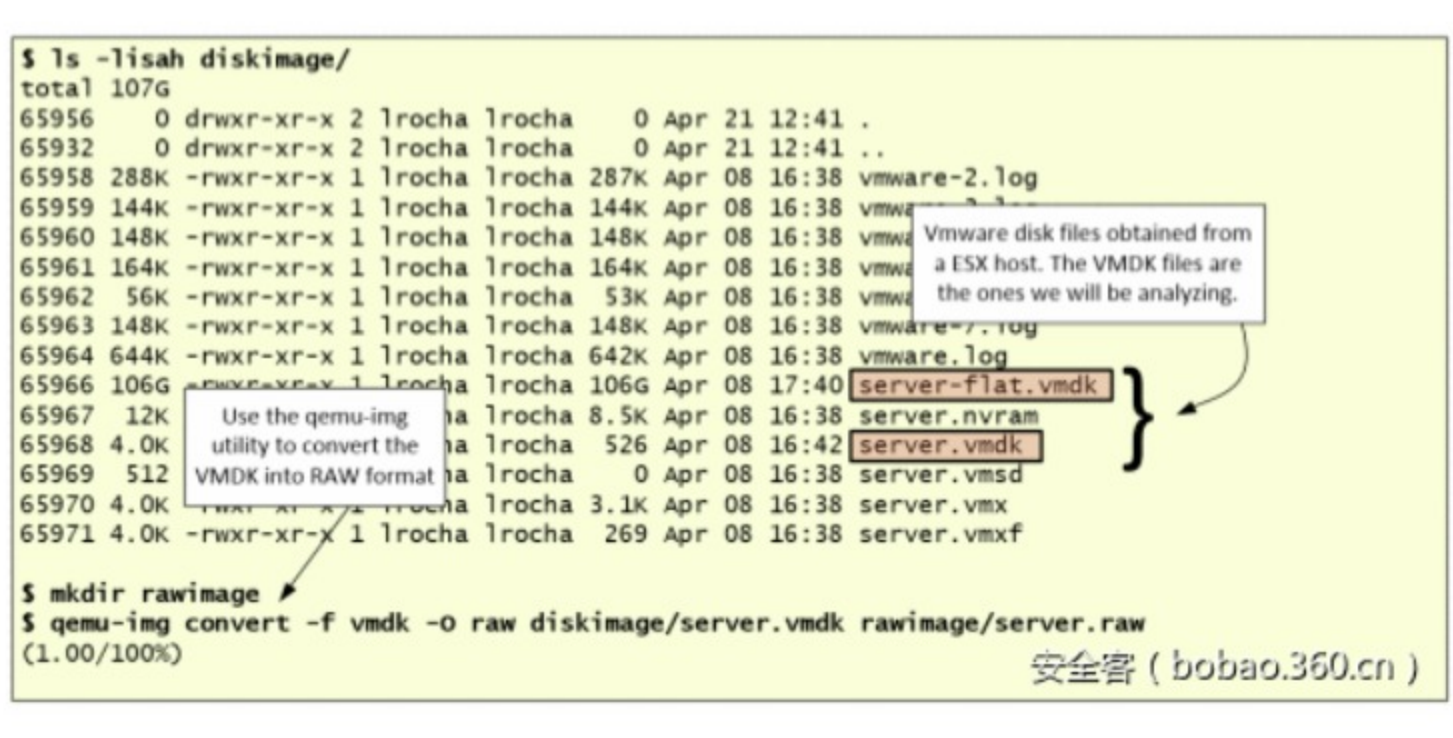
前言

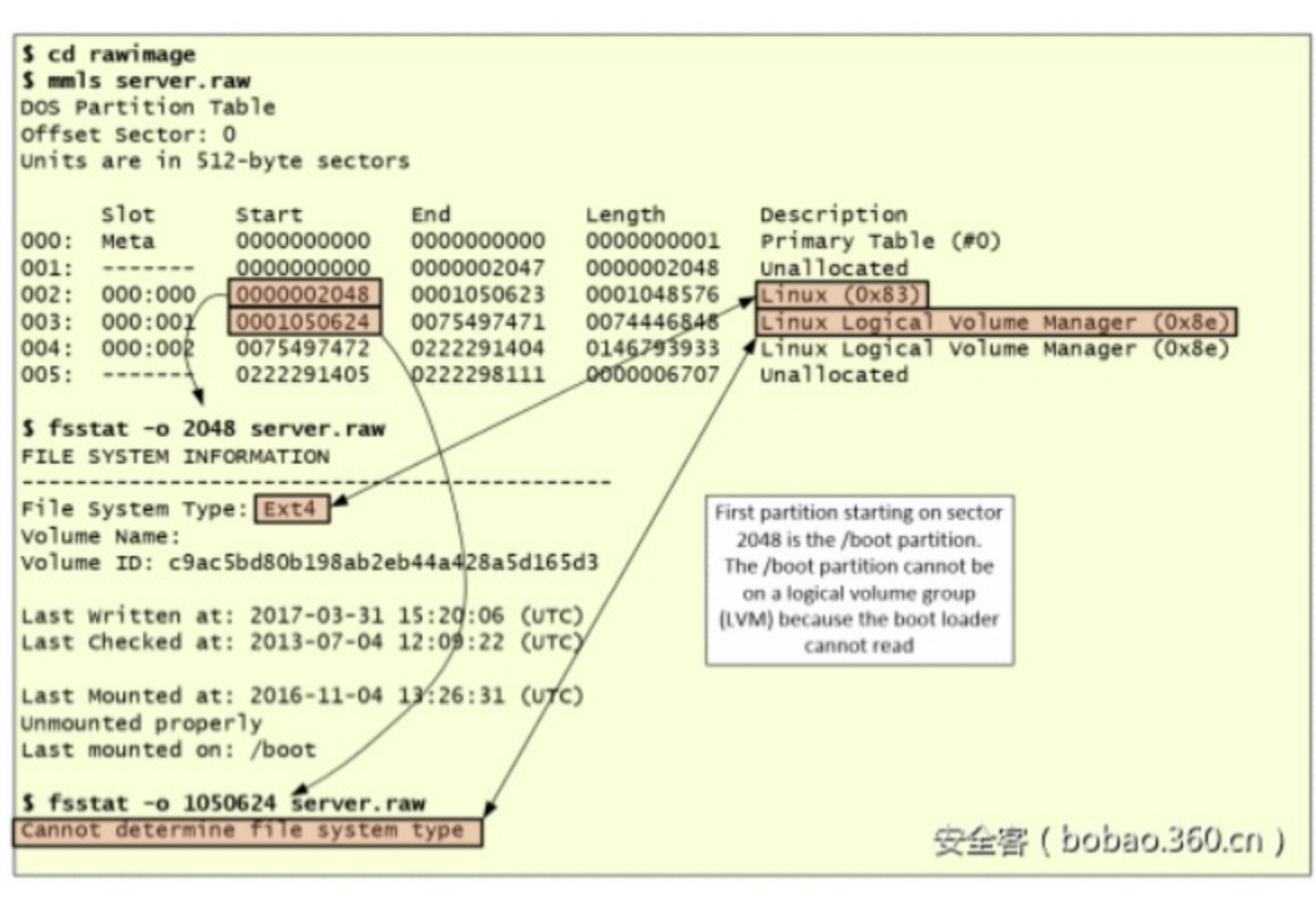
本文将介绍Linux取证技术并予以实践。我将执行一系列步骤，以分析从运行红帽操作系统的受损系统获取的磁盘。我首先是识别文件系统、挂载不同的分区、创建一个超级时间轴和一个文件系统时间轴。我还快速查看了工件，然后解挂了不同的分区。我将跳过如何获取磁盘的过程，不过可点击此处了解如何从VMware ESX主机获取磁盘镜像。

Linux取证技术实践

从ESX主机获取不同的磁盘文件时，你需要VMDK文件。然后，你将其移到你的实验室，这可能很简单，，因为你的笔记本电脑使用SIFT工作站运行虚拟机。要分析VMDK文件，你可以使用"libvmdk-utils"软件包，该软件包包含用于访问存储在VMDK文件中的数据的工具。不过另一种方法是将VMDK文件格式转换为RAW格式。如果采用后一种方法，运行不同工具将会更容易，比如Sleuth Kit中的工具（将针对镜像大量使用）。要执行转换，可以使用QEMU磁盘镜像实用程序。步骤如下图所示。

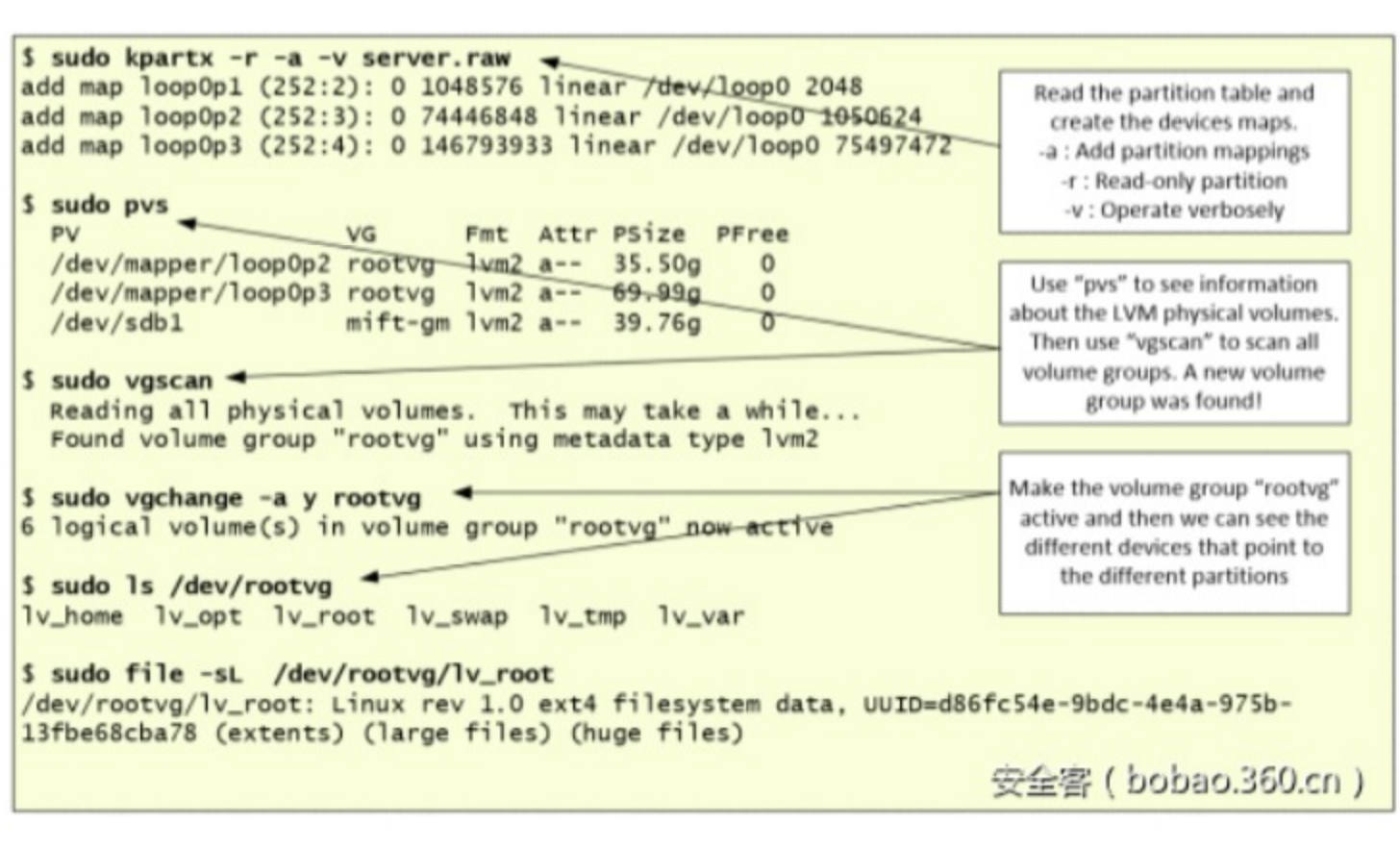


之后，你可以从磁盘镜像列出分区表，并使用"mmls"实用程序获取有关每个分区起始位置（扇区）的信息。然后，使用起始扇区，并使用"fsstat"实用程序查询与文件系统相关的详细信息。从下图可以看到，"mmls"和"fsstat"实用程序能够识别第一个分区"/boot"，类型为0x83 (ext4)。但是，"fsstat"无法识别从扇区1050624起始的第二个分区。

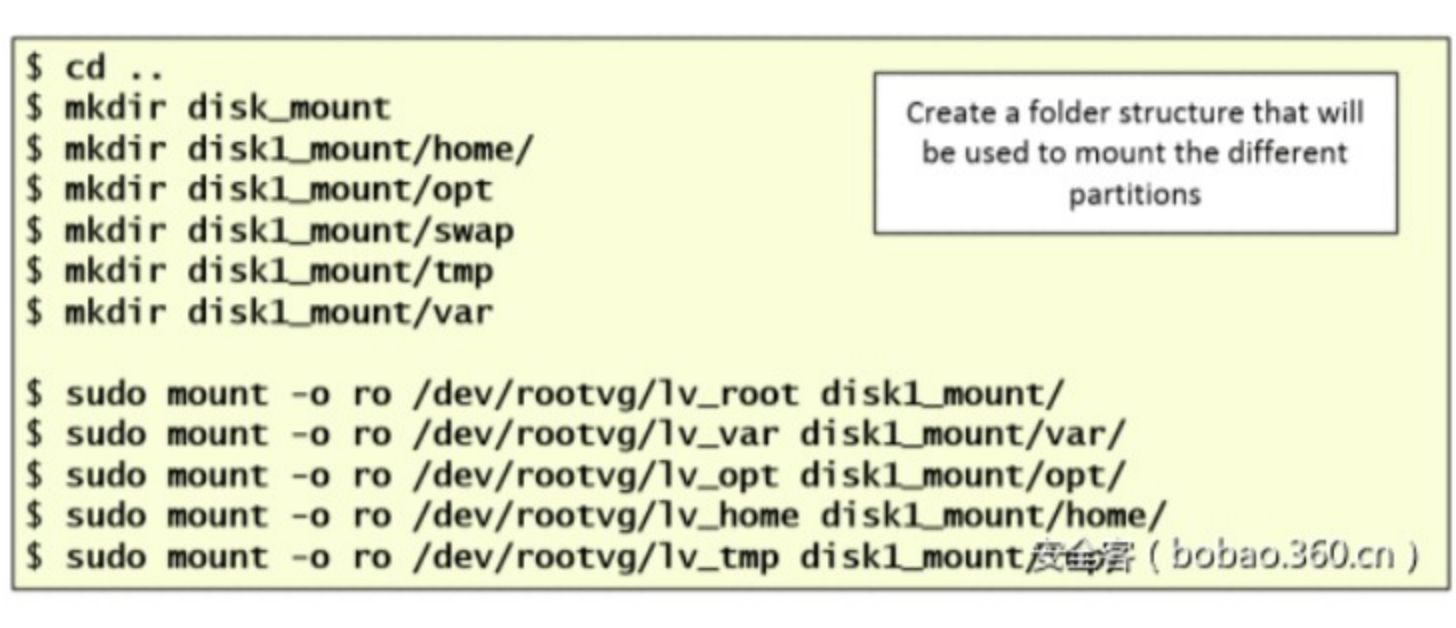


这是因为此分区的类型为0x8e（逻辑卷管理器）。如今，许多Linux发行版默认使用LVM（逻辑卷管理器）方案。LVM使用允许将硬盘驱动器或一组硬盘驱动器分配给物理卷的抽象层。物理卷组合成逻辑卷组，逻辑卷组可以分为具有挂载点和ext4等文件系统类型的逻辑卷。

使用"dd"实用程序，你可以轻松看到LVM2卷的存在。为了使其可用于我们不同的取证工具，我们需要从LVM分区表创建设备映射，其将通过创建环回设备并进行映射来自动创建分区设备。然后，我们使用管理LVM卷的不同实用程序，比如"pvs"、"vgscan"及"vgchange"。下图显示了执行此操作的必要步骤。



在激活LVM卷组之后，我们有六个设备映射到六个挂载点，这些挂载点生成了该磁盘的文件系统结构。下一步是将不同的卷作为只读挂载，因为我们将挂载正常设备进行取证分析。因为创建一个匹配分区方案的文件夹结构非常重要。



挂载磁盘后，通常可以通过创建时间轴来开始进行取证分析和调查。这是一个非常有用的关键步骤，因为其包含有关以人可读格式修改、访问、更改及创建的文件的信息，称为MAC时间证据（修改的、访问的、更改的）。此活动有助于查找事件发生的特定时间和顺序。

在我们创建我们的时间轴之前，值得注意的是，在Linux文件系统（如ext2和ext3）中，没有关于文件创建/生成时间的时间戳。只有3个时间戳。ext4中引入了创建时间戳。Dan Farmer和Wietse Venema的"The Forensic Discovery 1st Edition"一书概述了不同的时间戳：

最后修改时间。对于目录，是指最后一次添加、重命名或删除条目的时间。对于其他文件类型，是指最后一次写入文件的时间。

最后访问（读取）时间。对于目录，是指最后一次被搜索的时间。对于其他文件类型，是指最后一次文件被读取的时间。

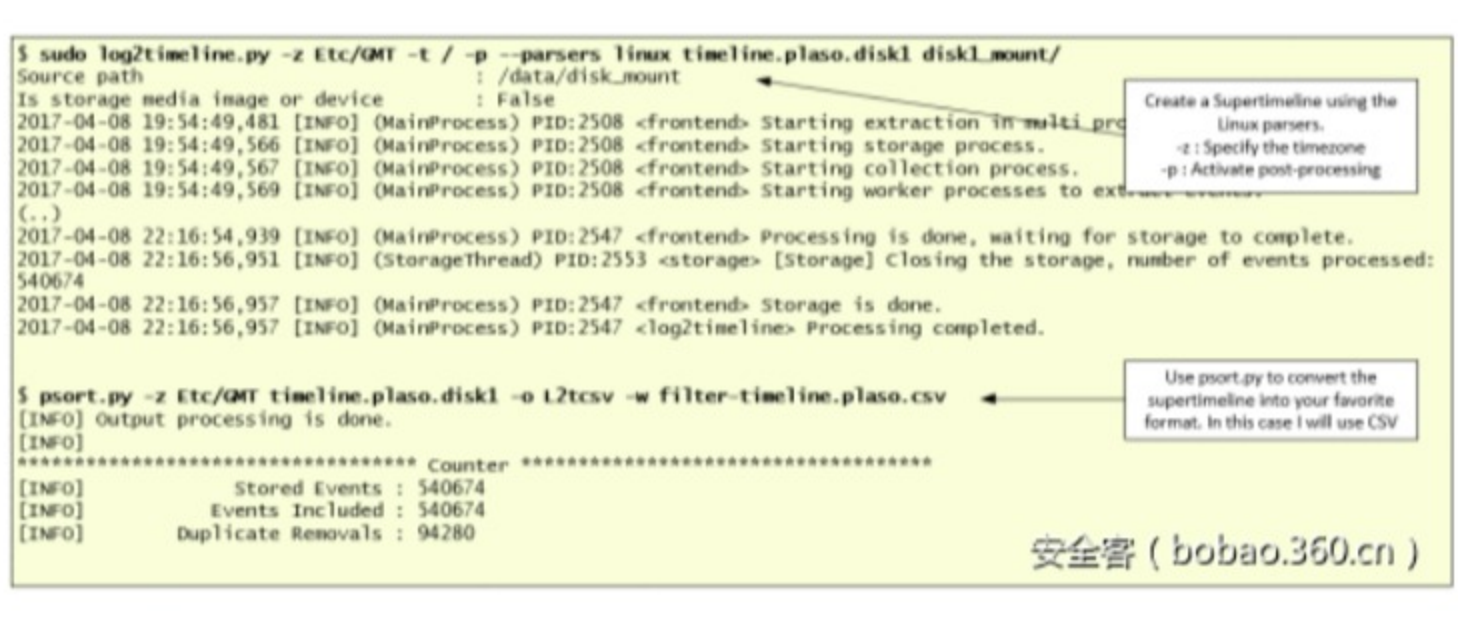
最后状态更改。状态更改的例子有：所有者的变更、访问许可的更改、硬链接计数的更改或任何MAC时间的显式更改。

删除时间。Ext2fs和Ext3fs在dtime时间戳中记录文件被删除的时间，但并不是所有的工具都支持它。

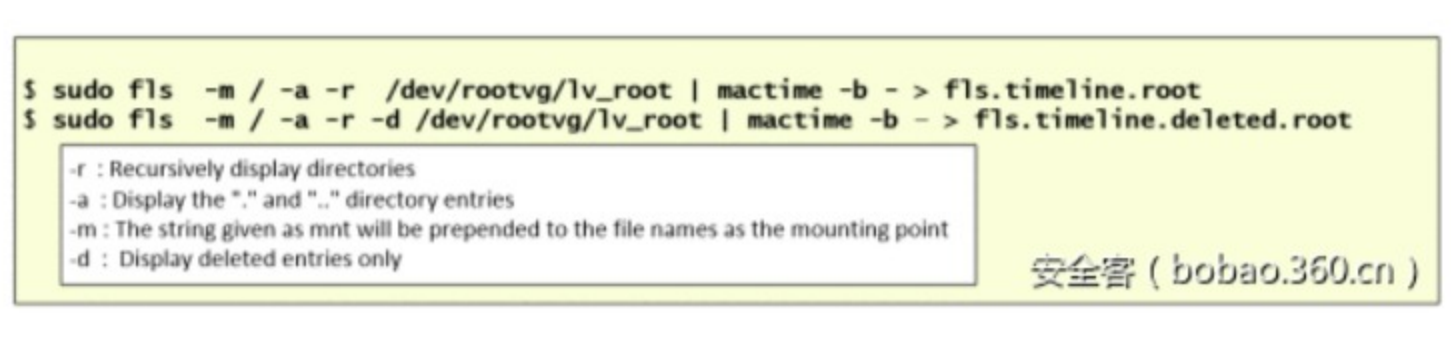
创建时间：Ext4fs在crtime时间戳中记录文件被创建的时间，但并不是所有的工具都支持它。

不同的时间戳存储在包含在inode中的元数据中。Inode（索引节点）相当于Windows中的MFT条目号。在Linux系统中读取文件元数据的一种方法是，首先使用，比如"ls -i file"命令，获取inode号，然后针对分区设备使用"istat"，并指定inode号。这将显示不同的元数据属性，包括时间戳、文件大小、所有者组和用户标识、权限及包含实际数据的块。

好的，我们先来创建一个超级时间轴。我们将使用Plaso来创建。Plaso是基于Perl的log2timeline的基于Python的重写。超级时间轴的创建是一个简单的过程，其适用于不同的操作系统。但是，解释很难。最后一个版本的Plaso引擎能够解析EXT 4，还能解析不同类型的工件，比如syslog消息、审计、utmp，等等。为创建超级时间轴，我们将针对已挂载的磁盘文件夹启动log2timeline并使用Linux解析器。这个过程将需要一些时间，当完成后，你将获得plaso数据库格式的带有不同工件的时间轴。然后，你可以使用"psort.py"实用程序将它们转换为CSV格式。下图概述了执行此操作所需的步骤。



在开始查看结合了不同工件的超级时间轴之前，你还可以为ext文件系统层（包含有关已分配和已删除的文件及未分配的inode的数据）创建传统时间轴。这分两步完成。首先，使用TSK中的"fls"工具生成body文件。然后，使用"mactime"对其内容进行排序，并以人可读格式呈现结果。您可以对使用"kpartx"创建的每个设备映射执行此操作。为简洁起见，下图仅显示了"/"分区的这一步。你需要为每个其他映射设备执行此操作。

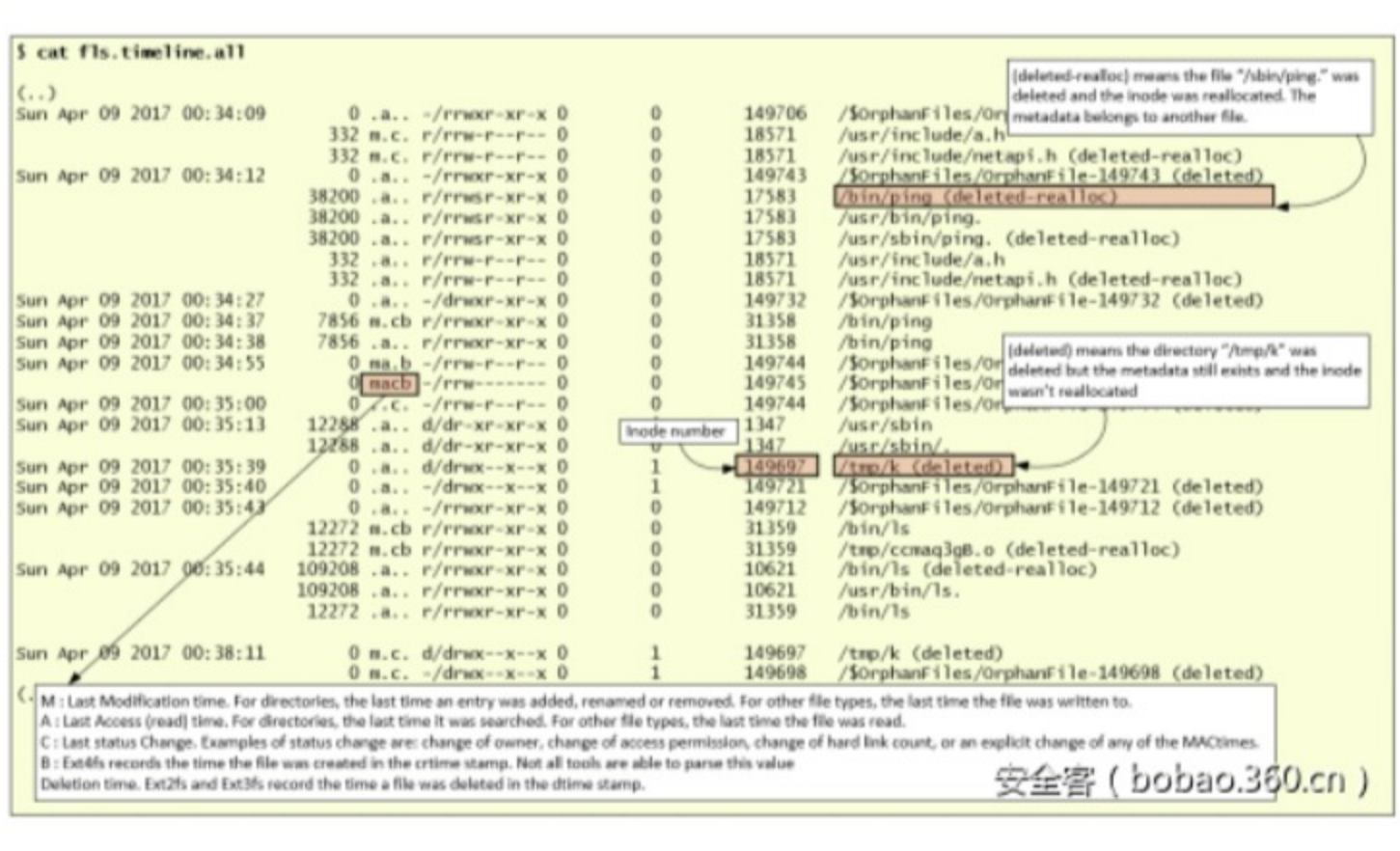


在我们开始分析之前，值得一提的是，在Linux系统中存在与调查相关的大量文件和日志。可用于收集和调查的数据量可能因配置的设置以及系统执行的功能/角色的不同而各异。另外，不同的Linux操作系统遵循一种文件系统结构——以共同标准排列不同的文件和目录。这称为文件系统层次结构标准(FHS)。熟悉这种结构有助于发现异常。要查看的东西很多，但其中要做的一件事是针对挂载的磁盘运行"chkrootkit"工具。Chrootkit是由Nelson Murilo和Klaus Steding-Jessen创建的脚本集合，可让您检查磁盘是否存在任何已知的内核模式和用户模式rootkit。

现在已生成超级时间轴和时间轴，我们可以开始分析了。在这种情况下，我们直接进入时间轴分析，在此提示，在四月的头几天可能发生了一些事。

在分析过程中，做到细致、耐心很有帮助，拥有全面的文件系统和操作系统工件知识也有裨益。有助于分析（超级）时间轴的一件事是拥有一定的有关事件确实发生的时间的引领。在这种情况下，在此提示，在4月初可能发生了一些事情。有了这一信息，我们开始缩小（超级）时间轴的时间范围。本质上，我们将寻找与日期有时间接近的相关的工件。目标是能够根据不同的工件重新创建所发生的事情。

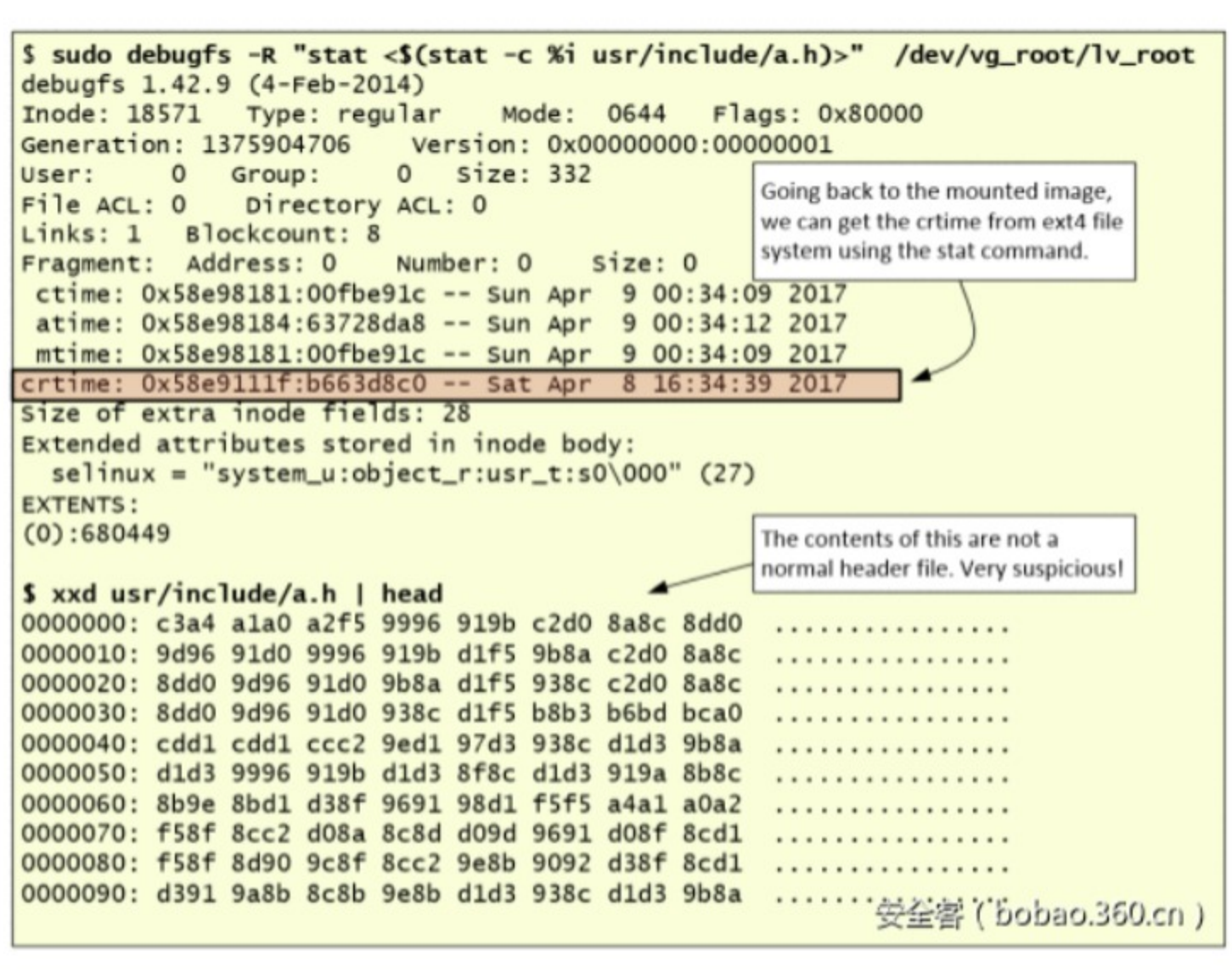
在分析时间轴后，我们发现了一些可疑活动。下图展示了使用"fls"和"mactime"生成的时间轴输出。有人删除了一个名为"/tmp/k"的文件夹，并重命名了"ping"和"ls"等常用二进制文件，并将相同名称的文件放在了"/usr/bin"文件夹中。



这需要进一步查看。查看时间轴后可以看到，"fls"的输出显示该条目已被删除。因为inode没有重新分配，所以我们可以尝试查看文件的备份是否仍然驻留在日志中。日志概念是在ext3文件系统中引入的。在ext4中，默认情况下日志功能为启用状态，并使用"data = ordered"模式。在这种情况下，我们也可以检查用于挂载文件系统的选项。为此，要查看"/etc/fstab"。我们可以看到使用的是默认值。这意味着，如果目录被删除与映像获取之间的时间间隔很短，那么我们可能有机会从被删除的文件恢复数据。尝试恢复被删除的数据的一种方法是使用"extundelete"工具。下图显示了该步骤。

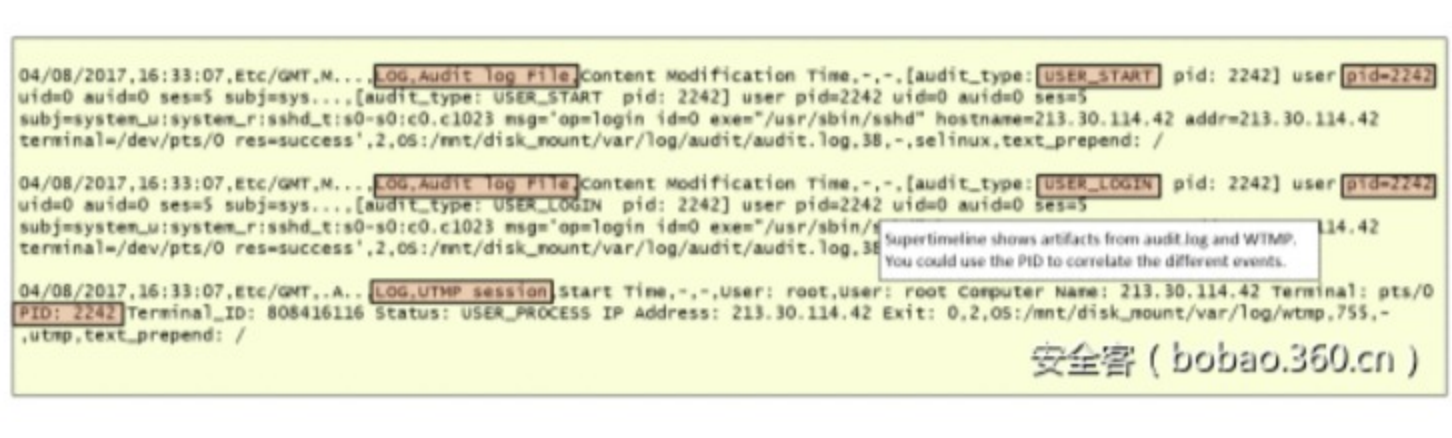
恢复的文件对更多了解发生了什么非常有用，可进一步帮助调查。我们可以计算文件MD5，验证其内容以及其是否是NSLR数据库或Virustotal已知的。如果其是一个二进制文件，那我们可以使用"objdump"和"readelf"等工具针对该二进制文件算出字符串并推导出功能。我们还获取并查看了在时间轴中看到的在"/usr/sbin"上创建的不同文件。检查其MD5，我们发现其是与Red Hat一起分发的合法操作系统文件。但是，"/bin"文件夹中的文件，比如"ping"和"ls"，不是合法文件，这些文件匹配从"/tmp/k"恢复的文件的MD5。因为某些文件是ELF二进制文件，所以我们将这些文件复制到了一个隔离的系统中，以便执行快速分析。我们可以使用"ltrace -i"和"strace -i"轻松启动二进制文件，其将拦截和记录不同的函数/系统调用。查看输出后可以很容易发现有错误之处。这个二进制文件看起来不是正常的"ping"命令，它调用fopen()函数来读取文件"/usr/include/a.h"，并写入/tmp文件夹中的一个文件，其中文件名使用tmpnam()生成。最后，其产生一个分段错误。下图显示了该行为。

提供这些信息后，我们返回查看，发现文件"/usr/include/a.h"在文件"ping"被移动/删除前一刻被修改了。所以，我们可以检查"a.h"文件的创建时间——ext4文件系统的新时间戳——使用"stat"命令。默认情况下，"stat"不显示crtime时间戳，但你可以将其与"debugfs"结合使用来获取。我们还检查了这个奇怪文件的内容是否是乱码。



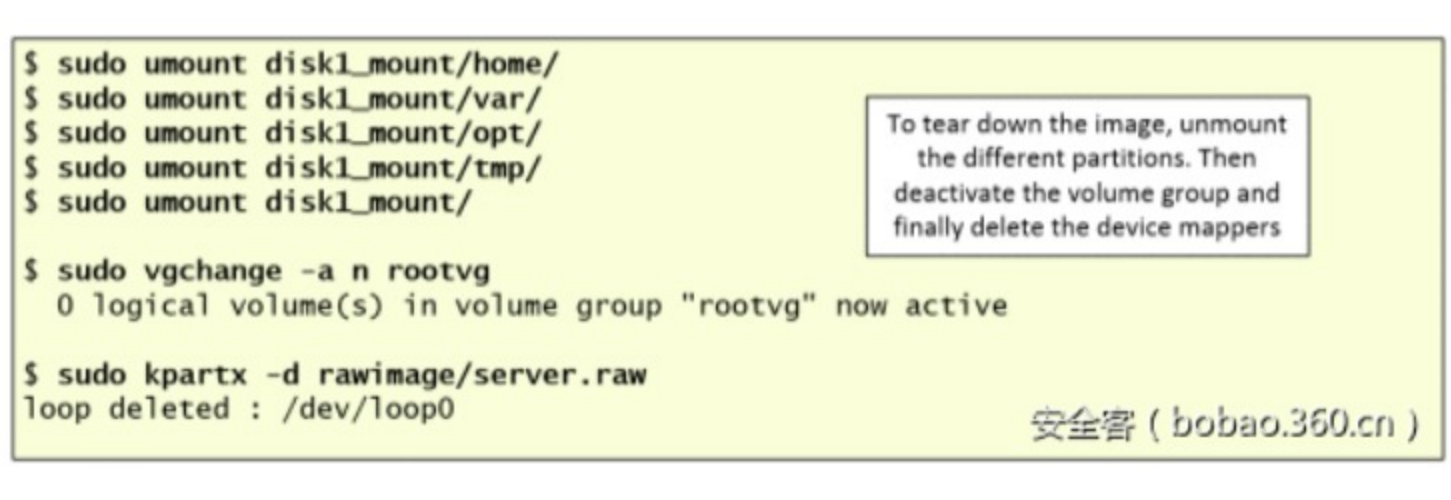
现在我们知道有人在2017年4月8日16:34创建了这个"a.h"文件，我们能够恢复被删除的其他几个文件。此外，我们发现一些系统二进制文件似乎放错了，至少"ping"命令预期从"a.h"文件中读取了一些东西。有了这些信息，我们回头看看超级时间轴，以便找到这个时候可能发生的其他事件。正如我提到的，超级时间轴能够解析来自Linux操作系统的不同工件。在这种情况下，经过一些清理，我们可以看到，在相关时间内我们有来自audit.log和WTMP的工件。Linux audit.log跟踪有关红帽系统的安全相关信息。基于预先配置的规则，Audit守护进程生成日志条目，以尽可能多地记录有关系统上正在发生的事件的信息。WTMP记录有关登录和登出系统的信息。

这些日志显示，在文件"a.h"被创建及"ping"和"ls" 二进制文件被错放前一刻，某人使用根证书从从IP 213.30.114.42（假IP）登录了系统。



现在我们有了网络指标。下一步，我们应开始查看我们的代理和防火墙日志，以获取关于IP地址的痕迹。同时，我们可以继续我们的时间轴分析，以查找有关的其他工件，并对找到的文件进行深入的二进制分析，创建IOC，比如Yara签名，其有助于找到环境中更多的受损系统。

完成分析和取证工作后，你可以解挂分区，取消激活卷组并删除设备映射。下图显示了这些步骤。



后记

Linux取证与微软Windows取证相比有很大不同，也很有趣。有趣的部分（调查）是熟悉Linux系统工件。安装原始的Linux系统，获取磁盘并查看不同的工件。然后利用一些工具/方法攻破机器，获得磁盘并再次进行分析。你可以通过这种方式反复练习自己的Linux取证技能。