计算机取证技术及其发展趋势*

王 玲[1,2,3+], 钱华林[1]

[1](中国科学院 计算机网络信息中心,北京 100080)
[2](中国科学院 计算技术研究所,北京 100080)
[3](中国科学院 研究生院,北京 100039)

Computer Forensics and Its Future Trend

WANG Ling[1,2,3+], QIAN Hua-Lin[1]

[1](Computer Network Information Center, The Chinese Academy of Sciences, Beijing 100080, China)
[2](Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, China)
[3](Graduate School, The Chinese Academy of Sciences, Beijing 100039, China)
+ Corresponding author: Phn: 86-10-62555113, E-mail: lwang@cnnic.net.cn
http://www.cnnic.net.cn
Received 2003-03-07; Accepted 2003-04-30

Wang L, Qian HL. Computer forensics and its future trend. *Journal of Software*, 2003,14(9):1635~1644.
http://www.jos.org.cn/1000-9825/14/1635.htm

Abstract: Computer forensics is the technology field that attempts to prove thorough, efficient, and secure means to investigate computer crime. Computer evidence must be authentic, accurate, complete and convincing to juries. In this paper, the stages of computer forensics are presented, and the theories and the realization of the forensics software are described. An example about forensic practice is also given. The deficiency of computer forensics technique and anti-forensics are also discussed. The result comes out that it is as the improvement of computer science technology, the forensics technique will become more integrated and thorough.
Key words: computer crime; computer security; computer forensic; anti-forensics

摘 要: 计算机取证研究的是如何为调查计算机犯罪提供彻底、有效和安全的技术.其关键是确保证据的真实性、可靠性、完整性和符合法律规定.介绍了计算机取证的过程以及取证软件的原理和实现,并且给出完整的取证实例.从理论和实现两个方面讨论了现有取证技术的局限性和面临的挑战,并展望其未来的发展方向.由于计算机犯罪手段的变化和其他技术的引入,现有的取证工作将向着深入和综合的方向发展.
关键词: 计算机犯罪;计算机安全;计算机取证;反取证
中图法分类号: TP309 **文献标识码**: A

计算机犯罪使公众蒙受重大损失,而打击计算机犯罪的关键是找到充分、可靠、有说服力的电子证据,因

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.2001AA112040, 2001AA112136 (国家高技术研究发展计划(863))

第一作者简介: 王玲(1973－),女,河北秦皇岛人,博士生,主要研究领域为计算机网络管理,网络安全.

此,计算机和法学的交叉学科——计算机取证(computer forensics)受到了越来越多的关注,甚至连续几年成为FIRST(forum of incident response and security teams)安全年会的热点.

与此同时,计算机取证技术得到了广泛应用.现在美国至少有 70%的法律部门拥有自己的计算机取证实验室.取证专家在实验室内分析从犯罪现场获取的计算机(和外设),试图找出谁、在什么时间、从哪里、怎样地进行了什么非法活动.

本文第 1 节介绍计算机取证的过程.第 2 节介绍当前计算机取证软件的原理和实现方法.第 3 节给出一个完整的取证实例.第 4 节从理论和实践的角度,分析现有取证技术的局限性,并且介绍反取证的方法.第 5 节对计算机取证技术的发展趋势作出预测.

1 什么是计算机取证

计算机在相关的犯罪案例中可以扮演黑客入侵的目标、作案的工具和犯罪信息的存储器这 3 种角色[1].无论作为哪种角色,计算机(连同它的外设)中都会留下大量与犯罪有关的数据.计算机取证就是对计算机犯罪的证据进行获取、保存、分析和出示,它实质上是一个详细扫描计算机系统以及重建入侵事件的过程.

计算机取证包括物理证据获取和信息发现两个阶段.物理证据获取是指调查人员来到计算机犯罪或入侵的现场,寻找并扣留相关的计算机硬件;信息发现是指从原始数据(包括文件,日志等)中寻找可以用来证明或者反驳什么的证据[2].与其他证据一样,电子证据必须是真实、可靠、完整和符合法律规定的[3].

1.1 物理证据获取

物理证据获取是全部取证工作的基础,在获取物理证据时最重要的工作是保证存到的原始证据不受任何破坏.无论在任何情况下,调查者都必须牢记[2]:

(1) 不要改变原始记录;

(2) 不要在作为证据的计算机上执行无关的程序;

(3) 不要给犯罪者销毁证据的机会;

(4) 详细记录所有的取证活动;

(5) 妥善保存得到的物证.

若现场的计算机正处于工作状态,取证人员还应该设法保存尽可能多的犯罪信息.由于犯罪的证据可能存在于系统日志、数据文件、寄存器、交换区、隐藏文件、空闲的磁盘空间、打印机缓存、网络数据区和记数器、用户进程存储区、堆栈、文件缓冲区、文件系统本身等不同的位置[2],要收集到所有的数据是非常困难的,在关键的时候要有所取舍.如果现场的计算机是黑客正在入侵的目标,为了防止犯罪者销毁证据文件,最佳的选择也许是马上关掉电源[4];而如果计算机是作案的工具或相关信息的存储器,则应该尽量保存缓存中的数据[2].

1.2 信息发现

取得了物理证据后,下一个重要的工作就是信息发现.不同的案例对信息发现的要求是不一样的.在有些情况下,只需找到关键的文件、图片或邮件就可以了,在其他时候则可能要求重现计算机在过去工作的细节(比如入侵取证).

为了保护原始数据,除非有特殊的需要,所有的信息发现工作都是对原始证据的物理拷贝进行的.物理复制的工作可以使用 Unix 系统的 dd 命令或使用专用设备进行.一般情况下,取证专家还要用 MD5 对原始证据上的数据做摘要,然后把原始证据和摘要信息及相关文档妥善保存.

由于包含着犯罪证据的文件可能已经被删除了,所以要通过数据恢复找回关键的文件、通信记录和其他的线索.事实上,现在的取证软件已经具有了很好的数据恢复能力,同时它们还可以做一些基本的文件属性获取和档案处理工作.

数据恢复以后,取证专家还要仔细进行关键字查询、分析文件属性和数字摘要、搜索系统日志、解密文件、评估 Windows 交换区等工作.由于现在缺乏对计算机上的所有数据进行综合分析的工具,所以信息发现的结果在很大程度上还依赖于取证专家的经验和智慧.这就要求一个合格的取证人员必须对信息系统有深刻的了解,

掌握计算机组成、操作系统、分布式计算、数据库、网络体系和协议等多方面的知识.

最后,取证专家会就计算机信息发现的结果作出完整的报告.这个报告将成为打击犯罪者的依据.

2 当前计算机取证软件的原理和实现

2.1 当前计算机取证软件的原理

当前的计算机取证软件的主要功能是文件信息获取和数据恢复.为了更好地理解它们的原理和实现方法,下面将以 Unix 为例介绍基本的文件系统理论和元数据的知识(不针对任何特定的实现).

组成一个 Unix 文件系统的数据分为两类:关于文件的信息和文件的内容.关于文件的结构和账号的信息称为元数据(包括超级块、索引节点和目录文件),而文件的内容则被简单地称为数据.

文件中的数据存储在硬盘上的数据块中,数据块的状态(空闲/已分配)信息存储在数据块位图中.借助于索引节点、目录文件等元数据,文件系统才能将一个或多个数据块中的数据组合成抽象的文件.

Unix 系统使用索引节点来记录文件信息,每一个普通的文件和目录都有惟一的索引节点与之对应.在索引节点中记录着文件的 UID、GID、大小、MAC(修改/存取/属性变更)时间、链接计数等信息.在每个索引节点中都存在一个数组,用来记录文件内容所在的数据块的直接地址(如果文件过大还会用到间接数据块指针).所有的索引节点都存储在索引节点表中,索引节点的状态信息(空闲/已分配)存放在索引节点位图中.索引节点的结构如图 1 所示.

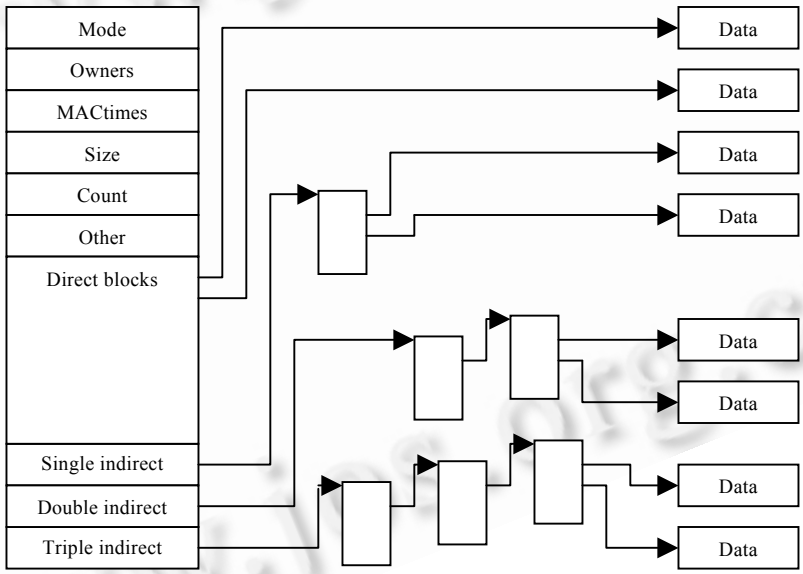

Fig 1. Structure of inode

图 1 索引节点的结构

为了能够通过文件名找到文件,Unix 文件系统还用到了目录文件.目录文件的逻辑结构为一棵以根目录开始的树.它实际上是文件名和索引节点的对应关系——目录项(directory entry)的列表.在目录项中记录文件名、索引节点号等信息.另外,由于目录项的长度不是固定的,所以目录项中还有专门的变量表示自身的长度.每建立一个文件就会在目录文件中就增加一个新的目录项.目录文件和目录项的结构如图 2 所示.

在 Unix 环境下删除一个文件的过程很简单,即首先把索引节点和文件所占用的数据块的状态信息标记为空闲,然后增大目录文件中相应的目录项的前一项中记录项长度的值,绕过对被删除目录项的访问.有关文件的数据和元数据都没有被彻底清除.图 3 表示的是删除一个文件对应的目录项之后的情形.

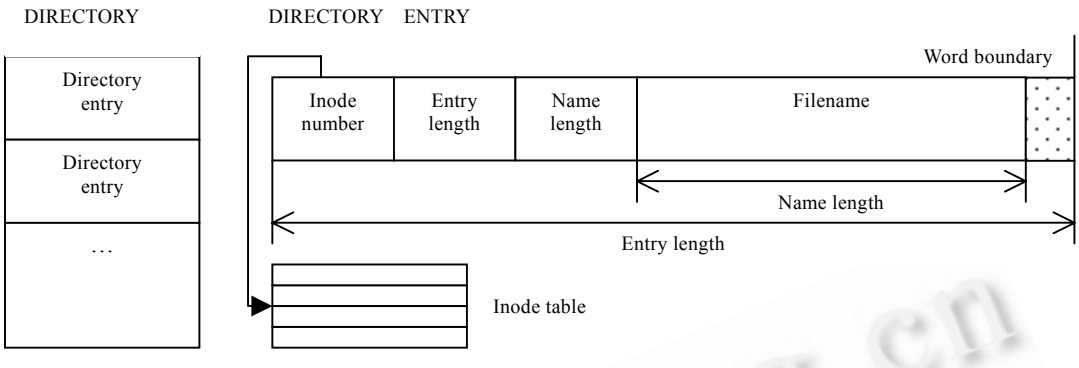

Fig.2 Directory and directory entry
图 2 目录和目录项结构

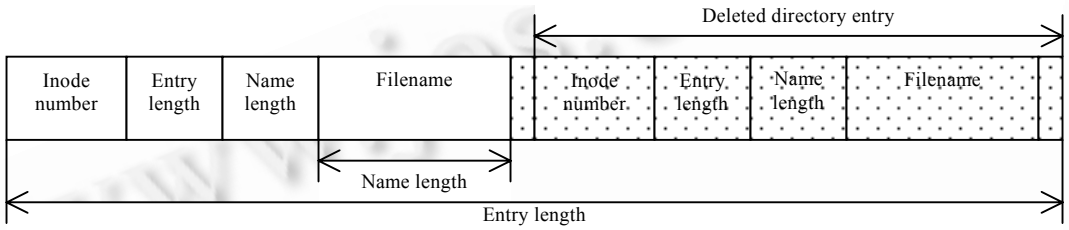

Fig.3 Delete a directory entry
图 3 删除一个目录项

正是由于文件系统的上述特点,所以在计算机的硬盘中会留下大量记录着过去曾经发生过的文件操作的信息.当前取证软件的作用就是收集和分析这些残存信息.取证软件可以从目录文件中找到被删除的目录项,从中恢复出过去存在过的文件的名字,还可以从索引节点中得到有关文件的信息,如果被删除的数据块没有被重新使用,它就能把文件从硬盘上找回来;即使找不回文件数据,取证者也可以从元数据中了解到文件系统中曾经发生过什么[5].

要了解文件系统过去发生的情况要用到索引节点中的“MAC 时间”.“MAC 时间”是 Unix,NT 和其他操作系统中文件和目录的时间属性——mtime(最近内容修改时间)、atime(最近访问时间)和 ctime(最近属性变更时间)的缩写(在微软的文件中把它们分别叫做 LastWriteTime,LastAccessTime 和 CreationTime).读一个文件会改变它的 atime,修改文件的内容会改变它的 mtime,而当文件的属性(所有者、组、权限)被改变时,它的 ctime 也会改变.在没有 dtime(删除时间)的系统中,文件被删除后可以认为 ctime 是它的删除时间——ctime 实际指的是对文件的最后一个引用被删除的时间[6].

“MAC 时间”对于调查计算机入侵事件和了解用户习惯非常重要.现在的取证软件可以读取索引节点中的数据得到文件的修改/存取/属性变更的时间,使调查人员了解系统启动的顺序,并可以通过 passwd 的修改时间、木马程序的生成时间来判断入侵发生的时间[6].

2.2 计算机取证软件实例

当前取证专家普遍看好的取证软件有 TCT(the coronor’s toolkit)和 Encase 等,下面以 TCT 为例,介绍取证软件的功能和实现方法.TCT 是 Earthlink 网络的 Dan Farmer 和 IBM 公司 Wietse Venema 研究员为了协助计算机取证而设计的工具包,它可以用来找回被删除的数据和取得系统中所有文件的属性信息.它有 4 个主要组成部分:grave-robber,unrm&lazarus,mactime 和一组小工具(ils,icat,pcat,file 等)[7].

2.2.1 缺省取证程序 grave-robber

Grave-Robber 是 TCT 中支持其他程序运行的框架.它通过调用其他子程序来找到文件、程序和网络信息等有用的数据,同时避免对原始数据的破坏,并且能保存和输出取证的结果.

2.2.2 数据恢复和阅读工具 unrm&lazarus

Unrm&lazarus 用来恢复被删除的数据.unrm 程序通过把所有处于空闲状态的数据块的内容都按字节拷贝到取证软件的数据空间来防止对原始数据的破坏.

Lazarus 的作用是整理由 unrm 找回的未知结构的数据,以方便用户阅读和操作.它有两个基本功能:取得原始数据和对这些数据进行分割、分析.它只是把原始数据划分成小的数据片,而不对它们进行任何改变.lazarus 对数据的分割基于两条最基本的假设:

(1) Unix 文件系统总是从数据块边缘开始写数据.所以只要按照数据块的大小对原始数据进行分割,就不会把不同文件的内容分在同一个数据片中;

(2) Unix 文件系统总是尽量把同一个文件放在连续的数据块上.

此外,lazarus 还能通过检查数据片中最初 10%的字节是否为可打印的字符来确认该数据的类型——文本或二进制.如果是文本数据,它甚至还会对照一系列常用的格式确定更精确的细节.比如,它会把有“From:foo@bar.com”的文本确认为邮件.

2.2.3 获取文件 MAC 时间的工具 mactime

Mactime 用于读取并且报告系统中所有文件的 MAC 时间,为取证人员了解程序的调用和敏感文件的访问和改变时间提供帮助.

2.2.4 其他小工具

TCT 工具包中提供了一些有用的小工具,这些小工具可以由 grave-robber 调用执行,也可以单独使用.比如,file 的作用是用来确定文件是文本文件还是二进制程序;ils 是用来显示被删除的索引节点的原始资料;icat 用于取得特定的索引节点对应的文件的内容等等.

3 完整的取证过程实例

上面我们介绍了计算机取证软件的原理和实现,为了让读者对计算机取证(特别是信息发现)的过程有一个完整的概念,下面让我们看一看华盛顿大学计算机安全服务小组的资深专家 Dave Dittrich 对被入侵计算机进行取证的全过程[4].

Dave Dittrich 在有证据显示计算机被入侵时马上切断了系统电源.接下来他所做的事是对原始硬盘进行物理拷贝,用 MD5 对原始硬盘上的数据做摘要,保存原始证据和摘要信息.相关的注意事项在前面的内容中已作过介绍,在此不再赘述.

下面就要进行信息发现过程的操作了.Dave Dittrich 对此给出了一条非常有用的忠告:“准备一个正规的笔记本,仔细记下在调查过程中所有的发现”.因为计算机取证专家的工作就是在硬盘、日志文件、软盘等原始证据中寻找线索以重建犯罪发生时的情景,而重要的证据可能来自于系统中的任何数据碎片,所以保存和收集证据的过程都应该是小心谨慎地进行,对取证中的每一个步骤和发现都进行详细的记录.

Dave Dittrich 将物理复制的磁盘以只读方式安装在取证系统上.首先他使用标准的 Unix 工具进行分析.他检查 passwd 文件并发现了 3 个可疑的账号 r00t 和 *x*,*y*,于是决定以此为突破口.他分别检查了这 3 个可疑账户的 home 目录,从中发现了一些奇怪的文件,进一步分析这些文件得到黑客安装 rootkit 工具的证据,并且找到了被安装的黑客工具.

为了了解更详细的信息,Dave Dittrich 接下来使用了 TCT 工具.他使用 mactime 对系统中所有文件的 MAC 时间进行排序,并且使用 unrm 恢复出所有被删除的文件.随后,他开始从大量的数据中寻找入侵线索.最后,Dave Dittrich 把所有的发现汇总起来形成了最终的报告.

在这份报告中,Dave Dittrich 根据一些敏感文件被修改的时间确认了入侵者首次成功进入系统的时间以及以后每次登录的时间和进行的操作.从 unrm 恢复出来的日志文件里,他找到了入侵者登录时使用的 IP 地址以及入侵者编译和安装黑客软件的证据.最让人兴奋的是,Dave Dittrich 发现这名入侵者把一些敏感数据发往一个特定的邮箱.这些发现为确认入侵者的身份提供了依据.

以下的内容摘自这份报告[4]:

……

在 XXX 04 之前,没有发现异常情况.在 XXX 04 当天,Berkeley 的远程登录程序被改写了.经过检查,证明改写后的程序是一个系统后门:

(mactime 的输出,格式为:日期 时间 文件大小 操作(MAC) 权限 所有者 组 文件名)

```
------------------------------------------------------------------------------
XXX 04 XX 23:42:21     23421 m.. -rwxr-xr-x root         root         /x/usr/sbin/in.rlogind
------------------------------------------------------------------------------
```

8 天以后,这个文件的属性又发生了变化,同一时间“chown”程序被执行:

```
------------------------------------------------------------------------------
XXX 12 XX 11:04:10     23421 ..c -rwxr-xr-x root         root         /x/usr/sbin/in.rlogind
XXX 12 XX 11:04:11      8156 .a. -rwxr-xr-x root         bin          /x/bin/chown
------------------------------------------------------------------------------
```

半个小时以后,一个 sniffer 源程序被拷贝到隐含目录/etc/中(这是一个精心伪造的目录名,它实际上是“/etc/.. ”,即在“/etc/”的后面还有两个“.”和 3 个连续的空格).源程序被编译,可执行文件放在系统目录“/usr/sbin/telnetd”中.

4 分钟以后,有人通过 ftp(wu.ftpd)登录:

```
------------------------------------------------------------------------------
XXX 12 XX 11:36:59     5127 m.c -rw-r--r-- root     root     /x/etc/..___/linsniff.c
XXX 12 XX 11:37:08     4967 .a. -rw-r--r-- root     root     /x/usr/src/linuxelf-1.2.13/include/linux/if.h
                       3143 .a. -rw-r--r-- root     root     /x/usr/src/linuxelf-1.2.13/include/linux/if_arp.h
                       3145 .a. -rw-r--r-- root     root     /x/usr/src/linuxelf-1.2.13/include/linux/if_ether.h
                       1910 .a. -rw-r--r-- root     root     /x/usr/src/linuxelf-1.2.13/include/linux/ip.h
                       2234 .a. -rw-r--r-- root     root     /x/usr/src/linuxelf-1.2.13/include/linux/route.h
                       1381 .a. -rw-r--r-- root     root     /x/usr/src/linuxelf-1.2.13/include/linux/tcp.h
XXX 12 XX 11:37:10     2048 ..c drwxr-xr-x root     bin      /x/usr/sbin
XXX 12 XX 11:37:14     2048 m.. drwxr-xr-x root     bin      /x/usr/sbin
XXX 12 XX 11:37:15     8179 m.c -rwxr-xr-x root     root     /x/usr/sbin/telnetd
XXX 12 XX 11:37:48     8179 .a. -rwxr-xr-x root     root     /x/usr/sbin/telnetd
XXX 12 XX 11:41:52     77476 .a. -rwxr-xr-x root    bin      /x/usr/sbin/wu.ftpd
XXX 12 XX 11:42:08     4096 mac -rw-r--r-- root     root     /x/var/pid/ftp.pids-remote
------------------------------------------------------------------------------
```

此人的活动可以通过被恢复的日志得到:

```
------------------------------------------------------------------------------
XXX 12 11:33:05 XXXX in.telnetd[1290]: connect from AAAAAA.XXXXXX.XXX
XXX 12 11:33:16 XXXX login: 1 LOGIN FAILURE FROM AAAAAA.XXXXXX.XXX, XXX
XXX 12 11:33:21 XXXX login: 2 LOGIN FAILURES FROM AAAAAA.XXXXXX.XXX, XXX
 . . .
XXX 12 11:34:02 XXXX su: XXXXX on /dev/ttyp1
XXX 12 11:41:52 XXXX wu.ftpd[1327]: connect from BBBBBBB.XXXXXX.XXX
XXX 12 11:41:57 XXXX ftpd[1327]: USER XXXXX
XXX 12 11:41:59 XXXX ftpd[1327]: PASS password
XXX 12 11:42:00 XXXX ftpd[1327]: SYST
XXX 12 11:42:01 XXXX ftpd[1327]: CWD /tmp
XXX 12 11:42:06 XXXX ftpd[1327]: TYPE Image
XXX 12 11:42:06 XXXX ftpd[1327]: PORT
XXX 12 11:42:06 XXXX ftpd[1327]: STOR mountd
XXX 12 11:42:08 XXXX ftpd[1327]: QUIT
XXX 12 11:42:08 XXXX ftpd[1327]: FTP session closed
XXX 12 12:00:25 XXXX in.telnetd[1342]: connect from AAAAAA.XXXXXX.XXX
XXX 12 12:00:25 XXXX telnetd[1342]: ttloop:    peer died: Try again
```

记录显示入侵者可能是上载了一个叫做“mountd”的缓冲区溢出程序,并通过这个程序进入了系统.同时还表明入侵者在 11:33:05 到 12:00:25(太平洋标准时间)内使用过主机 AAAAAA.XXXXXX.XXX [XXX.XXX.XXX.XX].

在“/usr/sbin/telnetd”的信息表明 sniffer 程序被执行,日志保存在“tcp.log”中:

```
 . . .
cant get SOCK_PACKET socket
cant get flags
cant set promiscuous mode
----- [CAPLEN Exceeded]
----- [Timed Out]
----- [RST]
----- [FIN]
%s =>
%s [%d]
eth0
tcp.log
cant open log
Exiting...
 . . .
```

.....

以下是 14 日的 telnet 会话的记录.记录显示有人把 sniffer 的结果发送到一个电子邮件内:

```
XXXXXXXXXXXXX.washington.edu => XXXXXXX.washington.edu [23]
!'"%W#$ 38400,38400vt100bdoor
password
w
su r00t
cd /etvc
cd ".. "
ls
cat /etc/".. "/tcp.log | mail hackeraccount@hotmail.com
cat /etc/".. "/tcp.log | mail hackeraccount@hotmail.com
ncftp -u ls
cp tcp.log 1
ls
ncftp -y XXX.XXX
[A[D[D[D[D[D[D[D[D[Du
----- [Timed Out]
```

……

4 当前计算机取证技术的局限和反取证技术

计算机取证的理论和软件是近年来计算机安全领域内取得的重大成就.然而从对计算机取证理论和软件实现过程的分析中我们都可以发现,当前的计算机取证技术还存在着很大的局限性.

从理论上讲,计算机取证人员能否找到犯罪的证据取决于以下 3 个条件:首先,有关犯罪的电子证据必须没有被覆盖;其次,取证软件必须能够找到这些数据;,再次,取证人员还要能够知道文件的内容,并且能够证明它们

和犯罪有关.从当前软件的实现情况来看,许多所谓的“取证分析”软件还仅仅是可以恢复使用 rm 或 strip 命令删除的文件,要用它们对付老奸巨猾的犯罪者还相差甚远.

正是这些问题让一些计算机犯罪者感觉到有机可乘,所以在计算机取证技术蓬勃发展的同时,一种叫做反取证的技术悄悄地出现了.反取证就是删除或者隐藏证据使取证调查无效[5].总之,现在的反取证技术分为 3 类:数据擦除、数据隐藏和数据加密.这些技术还可以结合起来使用,让取证工作的效果大打折扣.

数据擦除是最有效的反取证方法.它是指清除所有可能的证据(索引节点、目录文件和数据块中的原始数据).原始数据不存在了,取证自然就无法进行.反取证工具包 TDT(The Defiler's Toolkit)专门设计了两款用于数据擦除的工具软件 Necrofile 和 Klismafile.Necrofile 用于擦除文件的信息和数据,它直接将 TCT 工具包中检查索引节点状态的工具 ils 据为己用,它把所有 TCT 可以找到的索引节点的内容用特定的数据覆盖,同时它还会用随机数重写相应的数据块;Klismafile 用于擦除目录中的残存信息,它从目录文件的入口开始寻找所有被删除的目录项,然后用零覆盖满足特定条件的目录项内容.Klismafile 不是一个完美的解决工具,因为被它修改后的目录文件中会出现目录项大小不正常的情况,当然现在还没有工具做这项检查[5].

为了逃避取证,计算机犯罪者还会把暂时还不能被删除的文件伪装成其他类型(例如库文件)或者把它们隐藏在图形或音乐文件中;也有人把数据文件藏在磁盘上的隐藏空间中,比如,反取证工具 Runefs 就利用 TCT 工具包不检查磁盘坏块的特点,把存放敏感文件的数据块标记为坏块来逃避取证[5].这类技术统称为数据隐藏.

数据隐藏仅仅在取证者不知道到哪里寻找证据时才有效,所以它仅适用于短期保存数据.为了长期保存数据,必须把数据隐藏和其他技术联合使用,比如使用别人不知道的文件格式或加密(包括对数据文件的加密和对可执行文件的加密).

加密数据文件的作用已经为我们所熟知了.而对可执行文件加密是因为在被入侵的主机上执行的黑客程序无法被隐藏,而黑客又不想让取证人员反向分析出这些程序的作用.尽管对可执行文件加密的具体方法随处理器的能力和操作系统的不同而发生变化,但基本思想是相同的:运行时先执行一个文本解密程序来解密被加密的代码,而被解密的代码可能是黑客程序,也可能是另一个解密程序[8].

除此之外,黑客还可以利用 Root Kit(系统后门、木马程序等),绕开系统日志或者利用窃取的密码冒充其他用户登录[9],使取证调查变得更加困难.

5 计算机取证的发展趋势

由于自身的局限性和计算机犯罪手段的变化(特别是反取证软件的出现),现有的取证技术已经不能满足打击犯罪的要求.另外,由于当前取证软件的功能集中在磁盘分析上,而其他工作全部依赖于取证专家人工进行,几乎造成计算机取证软件等同于磁盘分析软件的错觉.这些情况必将随着对计算机取证研究工作的深入和新的取证软件的开发而得到改善.此外,计算机取证技术还会受到其他计算机理论和技术的影响.总之,未来的计算机取证技术将会向着以下几个方向发展.

5.1 取证的领域扩大,取证工具向着专业化和自动化方向发展

现在的计算机犯罪已经达到了无孔不入的地步,除台式机外,大量的移动设备如便携式计算机、掌上电脑、手机都可能成为犯罪的目标.而犯罪的证据也会以各种不同的形式分布在计算机、便携式设备、路由器、入侵检测系统等不同设备上.要找到这些证据就需要针对不同的硬件和信息格式做出相应的取证工具.在 2002 年 FIRST 年会上,Joe Grand 介绍的对手持式操作系统设备进行取证的工具 pdd 就是这一类工具的代表[10].相信在不久的将来会有更多针对特定硬件、操作系统数据结构的取证工具出现,可弥补现有取证工具的匮乏.

另外,计算机取证科学是一门综合性的学科,涉及到磁盘分析、加密、图形和音频文件的研究、日志信息发掘、数据库技术、媒介的物理性质等许多方面的知识.现在,很多工作都依赖于人工实现的情况大大降低了取证的速度和取证结果的可靠性.相信未来的取证软件会加入更多的信息(系统数据、日志、数据库等)分析和自动证据发现的功能,以代替大部分人工操作.

5.2 融合其他理论和技术

吸收计算机领域内其他的理论和技术有助于更好地打击计算机犯罪.对下列领域的研究成就有可能帮助计算机取证技术克服当前的局限性:

5.2.1 磁盘数据恢复

磁盘是利用它表面介质的磁性方向表示数据的.在将数据写入磁盘时,磁头产生的磁场会使存储数据的介质朝着某个方向磁化.值得注意的是,在写入新数据时,介质所具有的磁性强度不能完全摆脱其原始状态的影响.通俗地说,假设我们认为“1”被写到磁盘上时介质的磁力强度应该是 1.但事实上,我们把这个“1”写在原来为“0”的地方得到的磁力的强度大约是 0.95,而写在原来是“1”的地方就是 1.05.普通的磁盘电路会把这两个值都认为是 1,但是使用磁力显微镜(magnetic force microscope,简称 MFM)这样的专门工具,人们完全可以恢复出磁盘上的上一层甚至上两层数据.另外,由于新的数据很难精确地写在原有数据的位置上,即使经过多次随机覆盖之后,原来的数据还是可能被找出来[11].这一结论为取证专家提供了新的思路,使得恢复被覆盖了的数据成为可能.

5.2.2 反向工程

分析被入侵主机上可疑程序的作用是计算机取证工作的一部分.现在对 Unix 系统上二进制程序进行分析的工具软件屈指可数,它们更适合于对程序进行调试而不是反向工程,特别是可执行程序压缩和加密方法的使用,使得反向工程变得更加困难.为了分析计算机犯罪者所使用的软件的作用,需要专业的反向分析工程师的帮助[8].

5.2.3 解密技术

由于越来越多的计算机犯罪者使用加密技术保存关键文件,为了取得最终的证据,需要取证人员将文件中的内容进行解密.另外,在调查被加密的可执行文件时,也需要用到解密技术.

5.2.4 更安全的操作系统

当前,计算机取证软件的功能很大程度上取决于操作系统的支持.如何提高系统的安全性和更好地保存证据也是一个值得研究的问题.

5.3 取证的工具和过程标准化

由于计算机取证倍受关注,很多组织和机构都投入了人力对这个领域进行研究,并且已经开发出大量的取证工具.目前除 TCT 和 EnCase 以外,被大量使用的取证工具还有 DiskSearch 32,DiskSig,DM,DRIVESPY,FileCNVT,ForensiX,GetSlack 等等.因为没有统一的标准和规范,软件的使用者很难对这些工具的有效性和可靠性进行比较.另外,到现在为止,还没有任何机构对计算机取证机构和工作人员的资质进行认证,使得取证结果的权威性受到质疑.为了能让计算机取证工作向着更好的方向发展,制定取证工具的评价标准、取证机构和从业人员的资质审核办法以及取证工作的操作规范是非常必要的.

6 结束语

计算机取证科学是一个迅速成长的研究领域,它在国家安全、消费者保护和犯罪调查方面有着重要的应用前景.本文从技术的角度探讨了计算机取证的原理和未来的发展趋势.相信在这一领域的研究会向着深入和综合这两个方向不断前进,使取证科学发挥更大的作用.

致谢 在此,我们向对本文的工作给予支持和建议的同行,尤其是中国科学院计算机网络信息中心的同学们表示感谢.

References:

[1] Oseles L. Computer forensics: The key to solving the crime. 2001. http://faculty.ed.umuc.edu/~meinkej/inss690/oseles_2.pdf.

[2] Parra M. Computer forensics. 2002. http://www.giac.org/practical/Moroni_Parra_GSEC.doc.

[3] Sommer P. Computer forensics: An introduction. In: Proceedings of the Compsec'92—the 9th World Conference on Computer Security Audit and Control. London: Elsevier Advanced Technology, 1992. 89~96. http:// www.virtualcity.co.uk/vcaforens.htm.

[4] Dittrich D. Basic steps in forensic analysis of Unix systems. 2000. http://staff.washington.edu/dittrich/misc/forensics/.

[5] grugq. Defeating forensic analysis on Unix. Phrack #59 article6. 2002. http://www.phrack.org/show.php?p=59&a=6.

[6] Farmer D. What are MACtimes? Dr. Dobb's Journal. 2000,10. http://www.ddj.com/documents/s=880/ddj0010f/0010f.htm.

[7] Farmer D, Venema W. The coroner's toolkit (TCT). Dan Farmer & Wietse Venema. 2002. http://www.fish.com/tct/.

[8] grugq, scut. Armouring the ELF: Binary encryption on the UNIX platform. Phrack #58 article5. 2001. http://www.phrack.org/show.php?p=58&a=5.

[9] Grand J. pdd: memory imaging and forensic analysis of palm OS devices. In: Proceedings of the 14th Annual Computer Security Incident Handling Conference. Waikoloa. Hawaii: Forum of Incident Response and Security Teams, 2002. http://www.mindspring.com/~jgrand/pdd/pdd-palm-forensics.pdf.

[10] Gutmann P. Secure deletion of data from magnetic and solid-state memory. In: Proceedings of the 6th USENIX Security Symposium. San Jose, California: USENIX, 1996. 77~90. http://www.usenix.org/publications/library/proceedings/sec96/full_papers/gutmann/.

Call for Papers for GCC 2003

December 7~10 , 2003, Shanghai, China

http://www.cs.sjtu.edu.cn/gcc2003/index.htm

The Second International Workshop on Grid and Cooperative Computing (GCC2003) is to be held from December 7~10, 2003 in Shanghai, China. GCC2003 is the follow-up of the highly successful GCC2002 Workshop held in SanYa, HaiNan. It will serve as a forum to present current and future work as well as to exchange research ideas by researchers, developers, practitioners, and users in Grid computing, Web services and cooperative computing.

TOPICS OF INTEREST

The main topics of interest include, but not limited to:

Grid Computing and Grid Security

Grid Information Services

Grid Middleware and Toolkits

Grid Monitoring, Management and Organization Tools

Grid Applications

Information Grid and Knowledge Grid

Advance Resource Reservation and Scheduling

Performance Evaluation and Modeling

Web Services and Web Security

P2P Computing

Cooperative Middleware

Software Integration Technologies

Software Engineering Support for Cooperative Computing

Computer-Supported Cooperative Work

SUBMISSION

GCC 2003 invites authors to submit original and unpublished work. Papers should not exceed 10 pages of text using 10 point size type on 8.5×11 inch paper. Papers will be refereed and accepted on the basis of their scientific merit and relevance to the conference topics. Submission of full technical papers is called. Papers should be written in English and submitted in PDF format. All papers are to be electronically submitted to (Please also include the address, telephone, FAX, and email of the primary contact person in a separate sheet): gcc2003@cs.sjtu.edu.cn

IMPORTANT DATES

Electronic submissions due:	September 15, 2003
Acceptance notification due:	October 15, 2003
Camera-ready due:	November 5, 2003
GCC2003 Workshop:	December 7~10, 2003