电子数据证据收集系统保护机制及其发展趋势*)

孙 波 纪建敏 孙玉芳 梁 彬
(中国科学院软件研究所 北京 100080)

摘 要 随着计算机犯罪的不断增加,电子数据取证技术(Digital Forensic Technologies)越来越受到人们的重视。目前对计算机取证技术的研究主要集中于证据提取及证据分析等方面,然而对取证机制本身的安全没有考虑,这使得电子数据证据的完整性得不到充分的保障。本文对取证机制保护的相关研究工作进行了分析,并且讨论了该领域尚存的问题及今后的研究方向。试图为今后进一步研究取证系统保护机制提供理论及实践的依据。

关键词 电子数据取证,电子数据证据收集系统,完整性,真实性

The Protection Mechanism for Digital Evidence Collecting System and its Future Trend

SUN Bo JI Jian-Min SUN Yu-Fang LIANG Bin
(Institute of Software, Chinese Academy of Sciences, Beijing 100080)

Abstract Research regarding Digital Forensic Technologies has become more active with the recent increases in illegal accesses to computer system. Many researchers focus only on the techniques or mechanisms for evidence detecting and evidence analyzing, without considering the security of forensic mechanisms themselves. In this situation, we can't protect the digital evidence completely. Based on the analysis of relative researches, this paper summarizes the existing proposals, and offers suggestions to enhance the protection of Digital Evidence Collecting System.

Keywords Digital forensics, Digital evidences collecting system, Integrity, Fidelity

1 引言

近些年来,随着人们对信息技术的依赖性越来越强,计算机犯罪不断增加,此类犯罪造成的后果已经越来越严重[1]。为了侦破计算机犯罪以及查清系统中被利用的安全漏洞,计算机犯罪的调查取证显得越发重要[2,3]。对于计算机犯罪的取证,就是对计算机数据的取证。因此在对计算机犯罪进行侦察时,完整地从系统中提取电子数据证据以及对其进行固定与保全就显得更加重要。

电子数据证据不同于传统的犯罪证据,它们更加容易消失和被破坏[4]。这使得调查人员很难获得充分的电子数据证据。即使取得了一些证据,其完整性(integrity)和真实性(fidelity)也很难得到证明[5]。这些电子数据证据的固有特性严重阻碍了计算机取证技术的发展。为了获得完整可信的电子数据证据,应在敏感主机中预先安装设置电子数据证据收集系统(Electronic Digital Evidence Collecting System, EDECS),用来收集相关的电子数据证据,尤其是那些易被损坏和易于消失的数据。这样就保证了 EDECS 的明确性、单一性以及可证明性,从而可以依据需要取得较为完备的电子数据证据,确保被收集的电子数据证据的完整性和真实性[6]。

然而,EDECS 的某些模块往往存在于被攻击系统之中,因此其系统本身的安全性是首先应考虑的问题。也就是说,只有保证了整个 EDECS 的完整性,才能依据此系统所取得的数据进行后续的取证分析工作。本文在对取证机制保护之相关研究工作分析的基础上,讨论了该领域尚存的问题及今后的研究方向。

2 现存的证据收集系统保护机制的分析

2.1 EDECS 整体框架

为了更清楚的说明 I-LOMAC 保护机制的作用,首先概要地介绍一下 EDECS。此类系统一般多为 Client/Server 模式。客户端基本都包含了主机收集模块和网络收集模块,它们分别被安装在主机中和网络的入口处,以便收集主机以及网络信息。服务器端大体有与客户端进行通讯的数据采集模块,有依据相应规则对收集到的电子数据证据进行分类的分类模块,以及用于取证信息备份的备份模块等。为了支持后续的取证分析工作,还应有取证模块,它提供了一个与取证分析工具的接口,以便取证分析工具在取证库中查找所需的电子数据证据。图 1 描述了 EDECS 的整体框架。

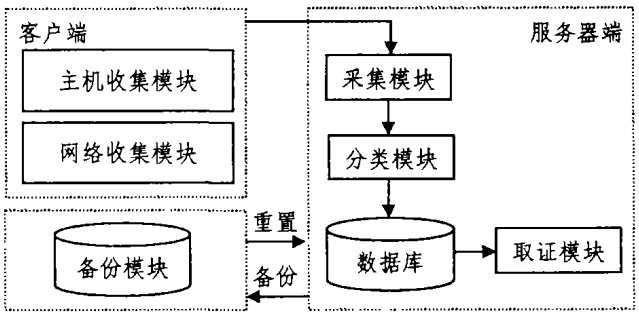

图 1 EDECS 的整体框架

2.2 客户端电子数据证据收集机制概述

客户端电子数据证据收集机制包括两个方面:主机收集模块和网络收集模块。

主机收集模块被安装在主机上。通过此模块收集的信息包括易丢失数据和描述系统运行状态的数据。易丢失数据主要包括临时数据(即关机就会消失的数据,如打开的网络连

*)本文研究得到国家自然科学基金(No. 60073022)、国家 863 高科技项目基金(No. 863-306-ZD12-14-2)、中国科学院知识创新工程基金(No. KGCX 1-09)、国家自然科学基金(No. 60373054)资助。孙 波 博士生,主要研究领域为系统软件安全性。纪建敏 中级工程师。孙玉芳 研究员,博士生导师,主要研究领域为系统软件和中文信息处理。梁 彬 博士生,主要研究领域为系统软件安全性。

接、内存常驻程序等),易变数据(即存储于硬盘但会被轻易改变的数据,如上次访问的时间戳)等。描述系统运行状态的数据大体分为:进程相关数据、文件系统相关数据、第三方安全系统(防火墙,IDS 等)的日志文件。

网络收集模块可被安装于一台独立主机之上,此主机应位于局域网的入口处。此模块截获过往的数据流,并依据不同的协议特征进行简单的分类。

诚然,以上框架并不一定具有绝对的普遍性。但无论何种证据收集框架,其证据收集模块必须存在于被检测的环境中,这也正是最有可能遭到攻击的环境。因此,对于证据收集模块的保护至关重要。本文力图通过对以上框架保护机制的分析及讨论,为今后进一步研究取证系统保护机制提供理论及实践的依据。

3 现存技术分析

3.1 威胁模型

EDECS 的保护机制使用了一个威胁模型,它包含了整个的攻击过程。最初,我们假设系统是可信的,并且正确配置了证据收集机制。经过一段时间之后,攻击者掌握了足够的系统相关信息,并对系统进行攻击。攻击者成功地获取了系统管理员的权限,并迅速攻破了系统中包括内核在内的所有软件,阅读和修改系统中的所有数据(包括存留在系统中的取证数据)。为了避免攻击被发现,攻击者往往使得各种服务正常运行以避免被察觉。而且这种逃避的时间越长,系统所遭受的损害越大。此时,系统管理员会通过某种方式发现此攻击。同时为了避免攻击者对系统造成更大的破坏,将会使系统停止运行。之后,通过获得的电子数据证据查找线索,分析攻击方法、攻击意图和造成的损失。

3.2 几种典型保护机制的分析

尽管文[6]提出了预先在系统中做好证据收集工作的重要性,但并没给出具体的实现。目前还没有可实际应用的 EDECS,更没有用于保护 EDECS 的机制。但我们发现日志系统保护机制中的很多思想和方法可以引用到 EDECS 的保护机制中去,并且对日志系统保护机制的研究也比较深入。因此本节将对现存的日志系统保护机制进行分析,作为 EDECS 保护机制可借鉴的理论基础。

3.2.1 Forward Integrity 保护机制 对于日志系统的保护最早来源于对安全时间戳的研究。一串时间戳顺序排列,任意时间戳的验证基于上一个时间戳[7]。这一思想被 Bellare 和 Yee 提炼为一种特性,称之为"Forward Integrity(FI)"特性。此特性保证了当系统被攻破后,系统日志的完整性不会被破坏[8]。

FI 将日志信息流视为一个个具有时间序列的日志信息体$\{m_1,m_2,\cdots,m_l\}$,所以日志文件可分解为一个个具有时间序列的日志记录体$\{e_1,e_2,\cdots,e_l\}$。对日志系统的攻击也就是对 e_i 的攻击,其中 $1\leqslant i\leqslant l$。因而将日志系统的安全问题转换为每个日志记录体的安全问题。FI 对每个日志记录体进行加密,并且每次加密所使用的密钥均不相同,即对于某时间段 j 有 $k_j=H(k_{j-1})$,其中 H 是某种单向函数。一旦 k_j 被生成,k_{j-1}即刻被删除。而基本密钥 k_0 将被保存在另一台可信主机中。因而即便在某时间段 i 内的 k_i 被泄漏,攻击者也无法获得当 $j<i$ 时的 k_j。从而攻击者无法更改时间段 j 内生成的任何信息。每个日志体中不但包含了加密的日志信息,还拥有信息认证码(MACs)。这是 FI 机制的重要组成部分:指定 m_i 被日志系统记录的时间为 t_i。将时间分成序列时段 ε_j,其中 $j=1,\cdots,n$;某一时间段 ε_j 的起始点标识为 T_j,因而 $\varepsilon_j=\{t:T_j\leqslant t\leqslant T_{j+1}\}$。边界时间 T_j 没有相关的日志记录。对于每个时段 ε_j 内的日志信息 m_i,日志系统生成日志记录体 $\log\_fn_j(m_i)$。对于每个日志记录体 $\log\_f\ n_j(m_i)=(m_i,FIMAC_j(m_i))$,其中 $FIMAC_j(m_i)$是附加在 m_i 上的认证码,它是依据 k_j 产生的。对 $\log_fn_j(m_i)$的验证可描述为:

$$valid(m,j,x)=\begin{cases}true\ \text{if}\ x=FIMAC_j(m)\\ false\qquad\text{otherwise}\end{cases}$$

以上描述了 FI 机制的基本原理,其设计思想主要体现在:即使在时间段 ε_j(其中 $j=1,\cdots,n$)的密钥 k_j 被攻破,所有 k_i(其中 $i<j$) 也不会被泄漏,从而保证了日志记录体 $\log_fn_i(\cdot)$的完整性不被破坏。但此机制也有其无法避免的安全缺陷。首先此机制需要一个可信系统来存储 FI 的基本密钥 k_0,以保护 k_i(其中 $i>0$)不被泄漏以及对 $\log_fn_i(\cdot)$的验证。但此机制无法保护日志系统的进程,对于日志文件的恶意删除也无法禁止。而且可信系统本身就包含了不可信的因素。

3.2.2 S&K 保护机制 这种由 Schneier 和 Kelsey 提出的机制与 FI 机制有许多相似之处,他们都将日志生成的时间分成连续的时间段 ε_j(其中 $j=1,\cdots,n$),对于每个 ε_j 都有唯一的认证密钥 k_j 与其对应。此密钥的生成也是基于某一单向函数($k_{j+1}=H(k_j)$)。但在他们的设计中对 FI 进行了改进[9]。图 2 显示了一个日志记录体生成的过程。其中,D_j 是将加入第 j 个日志记录体的日志信息;W_j 是第 j 个日志记录体的类型,它用作生成此记录体的验证许可掩码(permissions mask for verification)的输入条件之一;A_j 是第 j 个日志记录体的验证密钥;K_j 是用于对第 j 个日志记录体加密的密钥;Y_j 是哈希链中的第 j 个哈希值,用于验证第 j 个日志记录体的完整性。值得注意的是当 A_{j+1}和 K_j 被生成时,A_j 和 K_{j-1}立即被删除。

由以上描述可知 S&K 保护机制使用了与 FI 保护机制相似的方法来对日志信息记录体进行校验。然而其安全性在以下几方面有所改进:

(1)S&K 保护机制中的密钥 K_j 的生成不像在 FI 保护机制中那样直接生成,而是需要其它域的参与($K_j=H(W_j,A_j)$)。

(2)Y_j 是用于验证第 j 个日志记录体完整性的校验值。由于 $Y_j=H(Y_{j-1},E_{K_j}(D_j),W_j)$使得校验者看不到日志记录明文的情况下也可对其进行完整性验证。因此即便校验者只是一位半可信主体(semi-trusted party)也可对日志记录进行校验。

(3)在 S&K 保护机制中,Y_j 域提供了相对于 FI 保护机制更强的序列号结构,能更好地检测对日志记录体的删除和插入。

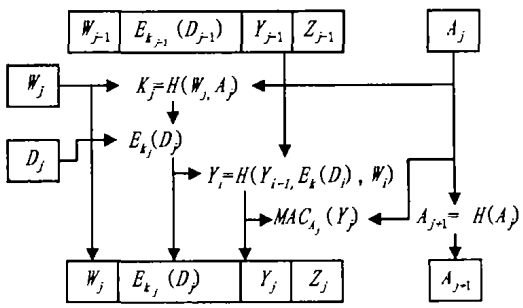

图 2 日志记录体的生成

然而,此机制也有其固有的弱点。首先,此机制虽然可以检验序列中是否有日志记录体被恶意删除,但对于整个日志文件的恶意删除还是无法保护。其次,无法防止对于日志系统

进程的恶意操纵。最后，此机制也需要一个可信系统来存储它的基本验证密钥 A_0，这本身也包含了不可信的因素。

3.2.3 NIGELOG 保护机制 NIGELOG 主要通过隐藏文件的方法来对日志信息实施保护[10]。此机制将日志信息多重备份，并将它们隐藏在随机选定的目录中。并且，当原始的日志信息被更改或删除时 NIGELOG 可以自动通过备份文件将原始信息恢复。此方法虽较好地保护了日志文件，但无法防止对日志系统进程的恶意操作。

3.2.4 Syslog-Sign 保护机制 前面几种方法集中于在当前主机中保护日志信息的安全，而此方法则侧重于将日志信息传输到另一台可信主机中来保障日志信息的安全[11]。此方法还提供了一些原始信息认证、信息完整性等功能，以防止信息在传递过程中被删除或更改。但此方法对于 Denial of Service 等攻击方法却无能为力，而且也无法保证日志系统的进程不受损害。

3.2.5 访问控制保护机制 此方法使用访问控制来限制一般用户对日志系统的访问[12]。然而，文[12]需要操作系统提供大量特殊的附加功能，与原系统的兼容性很差，并不适合一般情况。

4 尚存的问题及今后的工作

4.1 尚存的问题

在上一节中，我们对几种典型的日志系统保护机制进行了分析，它们的主要思想完全适用于 EDECS 的保护。然而，每种机制都存在着一些不安全、不实用的因素，这在上节中已有论述，此处不再重复。下面我们从攻击者的角度来说明现存技术中尚存的问题。

4.1.1 DOS 攻击对 EDECS 的影响 拒绝服务攻击(Denial of Service)指使用大量的无效信息充塞系统资源，引起系统服务中断。此类攻击十分普遍，它不但可用于攻击 Web 服务器，而且也可用于攻击 EDECS，使之失效。其对 EDECS 的攻击可分为两种，一种是直接攻击，另一种是间接攻击。直接攻击指通过产生海量的无效证据充塞 EDECS，使得之陷于瘫痪。间接攻击指攻击 EDECS 的客户端与服务器端的通讯，使之失效，这样 EDECS 位于服务器端的数据库就无法获得位于客户端的收集模块传来的取证信息。上一节讨论的保护策略均无法防止两种攻击方式。

4.1.2 日志截断攻击对 EDECS 的影响 此类攻击需与 DOS 攻击共同执行，现通过 DOS 攻击使得 EDECS 的服务器端与客户端失去通讯，然后攻入主机，截断或删除存于客户端的取证信息。这中攻击方式可避免服务器端生成取证信息完整性验证码，从而无法验证取证信息的完整性。

4.1.3 取证进程的特洛伊木马对 EDECS 的影响 特洛伊木马(Trojan horse)是一种基于远程控制的黑客工具，具有隐蔽性和非授权性的特点。所谓隐蔽性是指木马的设计者为了防止木马被发现，会采用多种手段隐藏木马；所谓非授权性是指一旦控制端与服务端连接后，控制端将享有服务端的大部分操作权限，而这些权力并不是服务端赋予的，而是通过木马程序窃取的。

攻击者取得系统管理员权限后，使得伪造的取证进程替代 EDECS 客户端的取证进程，并通过特洛伊木马非授权性的特点对取证信息进行控制。使得服务器端收集到的信息虽符合完整性的校验，但实际上是无效的。

4.2 保护需求

综合以上分析，为了保障 EDECS 的完整性和有效性，我们需实施以下保护：

- 对证据收集系统中的所有文件实施保护；
- 对证据收集系统中的所有进程实施保护；
- 对客户端与服务器端的通讯链路实施保护。

4.3 今后的工作

基于以上的保护需求，本文认为对 EDECS 的保护应从以下几方面去考虑：

安全隔离环境的提出。通过以上描述，我们可以看到日志系统本身就可包含自我保护的机制，此机制作为日志系统的一部分与它一起运行。然而此类机制无法对整个日志系统实施保护。对于 EDECS 也面临着同样的问题，即无法对运行于客户端的电子数据证据收集进程实施保护。

基于以上分析，我们提出了对 EDECS 实施保护的主要思想：提供一个安全隔离环境，使得 EDECS 运行于其中；也就是说 EDECS 将运行于一个隔离的环境中，其它进程无法对此环境进行访问，从而其它进程无法对 EDECS 的进程和文件进行访问。

整合入侵检测系统。借助于 IDS 来侦测包括系统管理员在内的系统用户之异常行为，来保护 EDECS 免受攻击。IDS 也可通过取证信息，来获得更大范围的系统运行状态。

支持分布式系统。大多数安全管理员都将取证信息收集到某一台可信主机中，这不但便于信息的收集，而且也使得后续的分析工作易于执行。然而，这也带来了诸如 DOS 攻击等安全威胁。虽 Syslog-sign IETF 可解决一些链路连接方面的问题，但也存在一些问题[11]。因此，将分布式系统应用于 EDECS 的保护是很重要的。

结论 本文对几种典型的可用于保护 EDECS 的安全方案进行了分析，并通过一些攻击方法说明几种典型的可用于 EDECS 保护之技术的局限性。最后依据分析结果提出了将安全隔离环境等技术应用于 EDECS 的必要性。计算机取证是一项正在发展的技术[5]，对 EDECS 的保护之研究更是刚刚起步[6]。如何对 EDECS 中的所有文件、所有进程和客户端与服务器端的通讯链路实施全面的、有效的保护，有待进一步的研究。

参 考 文 献

1 Hof R D. A New Era of Bright Hopes and Terrible Fears. Business Week, Oct. 1999. 50～56
2 Stallard B. Automated Analysis for Digital Forensic Science: [Masters Thesis]. Dec. 2002
3 Patel A. The Impact of Forensic Computing on Telecommunications. IEEE Communications Magazine, Nov. 2000. 64～67
4 Kornblum J. Preservation of Fragile Digital Evidence by First Responders. In: Digital Forensics Research Workshop, Aug. 2002
5 Palmer G. A Road Map for Digital Forensic Research. In: Digital Forensics Research Workshop, Aug. 2002
6 Tan J. Forensic Readiness. http://www.atstake.com, July 2001
7 Haber S, Stornetta W S. How to Time Stamp a Digital Document. Advances in Cryptology-Crypto'90, Springer-Verlag, 1991. 437～455
8 Bellare M, Yee B S. Forward Integrity For Secure Audit Logs. 1997 University of California, San Diego
9 Schneier B, Kellsey J. Secure Audit Logs to Support Computer Forensics. ACM Transaction on Information and System Security, 1999, ,2(2): 159～176
10 Takada T, Koike H. NIGELOG: Protecting Logging Information by Hiding Multiple Backups in Directories. In: Intl. Workshop on Electronic Commerce and Security (in conjunction with DEXA'99), IEEE CS Press, Sep. 1999. 874～878
11 Kelsey J, Callas J. Syslog-Sign Protocol DRAFT. Network Working Group. June 2002
12 Hewlett-Packard Company: HP Praesidium/VirtualVault White Paper, 1998. http://www.hp.com/security/products
13 U.S. Department of Defense, Computer Security Center. Trusted computer system Evaluation criteria, Dec. 1985