Revisionssichere Protokollierung in Standardbetriebssystemen

Stephen D. Wolthusen

Protokolldateien lassen sich sehr einfach erzeugen, aber auch verändern, unterdrücken und löschen. Die in Stadard-Betriebssystemen und Anwendungsprogrammen vorhandenen Schutzmechanismen sind unzureichend und müssen ergänzt und erweitert werden, um den wachsenden Anforderungen und der Kritikalität der Protokolldaten Rechnung zu tragen. Der Autor betrachtet die technischen und organisatorischen Anforderungen an eine revisionssichere und manipulationsgesicherte Sammlung von Protokolldaten und weist auf die Rahmenbedingungen hin, die bei der Realisierung der Anforderungen in Protokollarchitekturen für Standardbetriebssystemen zu berücksichtigen sind.

Prof. Stephen Wolthusen
Lecturer in der Information Security Group, Royal Holloway, University of London und Associate Professor an dem Gjøvik University College, Norwegen. Senior-Wissenschaftler am Fraunhofer-Institut für Graphische Datenverarbeitung in Darmstadt.
E-Mail: stephen.wolthusen@rhul.ac.uk

Einleitung

Die Erfassung, Speicherung und Verarbeitung von Protokolldaten stellt ein kritisches Element jeder IT-Sicherheitsarchitektur dar. Darüber hinaus führt die wachsende Bedeutung der IT-Unterstützung für Verwaltungs- und Geschäftsprozesse dazu, dass der Nachweis über den korrekten Ablauf dieser Prozesse verstärkt aus IT-Protokolldaten erbracht werden muss. Mithin ergeben sich eine Reihe stringenter Anforderungen an die Güte, Vertraulichkeit, Zuverlässigkeit, Integrität und Vollständigkeit von Protokoll- und Revisionsdaten, die auf mehreren Abstraktionsebenen vorliegen und von den rein technischen Abläufen bis hin zur Semantik von Geschäftsprozessen reichen [7].

Diesen Anforderungen – obgleich seit längerem in ihren Grundzügen bekannt [8] – stehen nur sehr ungenügende technische Mechanismen gegenüber, die in ihrem Einsatzbereich mit einer erheblichen Anzahl von Problemen belastet sind. Im Folgenden sollen daher sowohl die technischen und organisatorischen Anforderungen an eine revisionssichere und manipulationsgesicherte Sammlung von Protokolldaten betrachtet als auch die bei der Realisierung von Protokollarchitekturen auf Standardbetriebssystemen zu berücksichtigenden Randbedingungen beleuchtet werden.

1 Anforderungen

Die Anforderungen an revisionssichere Protokollierung ergeben sich einerseits mittelbar aus den Anforderungen der zu protokollierenden Prozesse und Abläufe, die ein breites Spektrum von der Erfassung atomarer technischer Vorgänge bis hin zu komplexen Geschäftsprozessen abdecken, andererseits auch unmittelbar aus den Bedrohungen, denen Protokolldaten von ihrer Entstehung bis hin zu einer analytischen Bewertung und Zusammenfassung ausgesetzt sind. Es ist daher zweckmäßig, zunächst diese beiden Hauptstränge der Anforderungen zu betrachten, bevor organisatorische und konkrete technische Anforderungen und Problemstellungen untersucht werden.

1.1 Prozessbasierte Anforderungen

Anforderungen an Zuverlässigkeit und Betriebssicherheit bedingen eine ständige Überwachung von IT-Systemen und Netzen, um Anomalien und Probleme in Abläufen erkennen und z.B. auch Angriffe auf die IT-Systeme erfassen zu können. Hierbei fallen auf sehr niedriger Abstraktionsebene erhebliche Volumina an Daten an, die zumeist ohne die Hilfe weiterer Analysewerkzeuge wie *Intrusion Detection Systems* zur Reduktion der Daten-Volumina sowie zur Erkennung von Angriffssignaturen und Anomalien nur begrenzt verwendbar sind. Dennoch müssen auch diese umfangreichen Datensätze in Rohdatenform vorgehalten werden, um verschiedene nachgeordnete und tiefergehende Analysen oder auch Beweisdatensicherung betreiben zu können. Diese Anforderungen können jedoch teilweise erst mit erheblicher Verzögerung bekannt werden, sodass eine langfristige Archivierung der Protokolldaten zu fordern ist.

Problematisch sind dabei jedoch nicht nur die Kosten einer langfristigen Archivierung, sondern auch die Tatsache, dass diese Daten ebenfalls personenbezogene Analysen zulassen und somit einerseits nicht unbegrenzt vorgehalten werden dürfen und andererseits auch der zweckgebundene Zugriff selbst nur auf vordefinierte Bedarfsträger beschränkt sein muss.

Für eine Vielzahl von Abläufen in IT-Systemen sind jedoch diese elementaren Protokollierungsmechanismen unangemessen, da sie auf einer zu niedrigen Abstraktionsstufe ansetzen und daher die Protokollierung der eigentlich gewünschten Prozesse

und Abläufe nur unzureichend wiedergegeben oder eine ungenaue Rekonstruktion aus partiellen Daten erfordern. Oft müssen diese Daten auch über mehrere Netz-Knoten hinweg korreliert werden. Dies kann jedoch nur ungenau gelingen, weil nicht zuletzt die Uhren der einzelnen Netzknoten nicht absolut präzise synchronisiert sein werden und damit die Kausalität nicht immer feststellbar ist, so dass sich auch hier Überwachungslücken auftun.

Es ist daher sinnvoll und notwendig, wo immer möglich eine Protokollierung auch auf höheren semantischen Abstraktionsebenen durchzuführen, um alle relevanten Aspekte der beteiligten Akteure und der erfassten Operationen selbst möglichst vollständig aufnehmen zu können.

Dies stößt jedoch zur Zeit noch auf Probleme in der praktischen Umsetzung, da – anders als bei den vorgenannten Basis-Revisionsdaten – hier kaum Standards existieren und daher unterschiedliche Formate der jeweiligen Anwendungen und Umgebungen separat behandelt werden müssen, was in komplexen Umgebungen mit einer größeren Anzahl von Anwendungsprogrammen einen erheblichen Mehraufwand bewirken kann [4].

Eine Vereinheitlichung und Standardisierung von Protokollierungs-Verfahren, welche das vorgenannte Spektrum von atomaren Ereignissen auf Betriebssystem- und Netzwerkebene bis hin zu relevanten Vorgängen in Geschäftsprozessen abdecken, ist daher angebracht. Um diese Daten erschließen zu können, ist es zudem erforderlich, das Format der erfassten Daten zu vereinheitlichen oder derart mit Metadaten aufzubereiten, dass eine anschließende automatische Verarbeitung effizient möglich ist.

Da Art und Umfang der zu protokollierenden Daten zumindest aus Sicht der technischen Analyse a priori keinen Beschränkungen unterliegt und diese Anforderungen sich mit der Zeit ändern, ist es von herausgehobener Bedeutung, dass eine derartige Protokollierungs-Architektur flexibel erweiterbar und an neue Anforderungen anpasst werden kann.

Neben diesen funktionalen Anforderungen müssen jedoch auch kritische Randbedingungen der Verarbeitungsgeschwindigkeit und der durch eine derartige Aufbereitung entstehende Erhöhung des zu verarbeitenden Datenvolumens beachtet werden, da beide Faktoren beträchtliche Mehrkosten verursachen können.

Welche durch eine exakte und detaillierte Erfassung von Protokoll- und Revisionsdaten verursachten Kosten und Aufwände hingenommen werden können, ist anwendungsabhängig. So gilt in kritischen Anwendungsbereichen bei Mainframe-Systemen ein Mehraufwand in der Größenordnung von 10 - 15% als durchaus akzeptabel.

1.2 Sicherheitsbezogene Anforderungen

Grundlage jeder weiteren Verarbeitung von Protokoll- und Revisionsdaten muss deren Vertrauenswürdigkeit und Zuverlässigkeit sein. Hierbei stellen sich vielfältige Anforderungen zum einen an die Prozesskette der Erfassung, Archivierung Übertragung und Verarbeitung der Protokolldaten, zum anderen an die Protokolldaten selbst [5].

Diese Anforderungen beginnen bei der Identifikation und Authentisierung der Quelle der Protokolldaten. Hierzu muss zweifelsfrei das System (z.B. Computer, aber auch die Netzwerk-Überwachungseinheiten) identifiziert werden können und ein Nachweis über die angegebene Identität (d.h. ein Authentisierungsvorgang) erbracht werden. Dabei ist es nicht ausreichend, allein das betroffene Gerät zu identifizieren, sondern es muss möglichst die gesamte Protokollierungsinstanz zusammen mit ihren logischen Abhängigkeiten erfasst werden.

Dies ist einerseits der Möglichkeit zur Kompromittierung oder Modifikation der Protokollierungsarchitektur geschuldet, zum anderen ist die Annahme einer einzelnen Instanz zur Protokollierung (oder auch nur einer einzelnen Betriebssysteminstanz) mittlerweile nicht mehr haltbar, da auf einem physischen Computer im Wege einer Virtualisierung mehrere unterschiedliche Betriebssystem-Instanzen mit unterschiedlichen Protokollierungs-Architekturen angesiedelt sein können.

Die Protokollierungs-Instanz muss hingegen in der Lage sein, die Quelle der Protokolldaten zweifelsfrei zu identifizieren und zu authentisieren (z.B. eine Netzwerkschnittstelle, aber auch ein Anwendungsprogramm oder -prozess) und die erforderlichen Metadaten (z.B. Zeitstempel oder auch verwendete Nutzerkonten) manipulationssicher erfassen zu können.

Für diese Datenquelle gilt jedoch, dass die Protokollierungs-Instanz keine unmittelbare Prüfung der semantischen Korrektheit der Daten durchführen kann. Es liegt daher in jedem Fall eine implizite Vertrauensbeziehung zur Datenquelle vor. Ist die Datenquelle selbst kompromittiert oder liefert sie aus anderen Gründen (z.B. aufgrund eines technischen Fehlers) fehlerhafte Daten, so kann dies nur nachträglich unter der Voraussetzung entsprechender Möglichkeiten zum Abgleich und zur Plausibilitätsprüfung anhand anderer Daten erkannt werden.

Aus diesem Grund muss der Kanal zwischen Datenquelle und Protokollierungs-Instanz vor externen Einflüssen geschützt sein. Ist diese Annahme nicht unmittelbar (z.B. durch Ansiedlung beider Komponenten im Betriebssystemkern) gegeben, muss neben der Identifikation und Authentisierung der beteiligten Kommunikationsparteien auch der eigentliche Kommunikationskanal gesichert werden, um Angriffe wie das wiederholte Einspielen von Daten, eine Manipulation der Datenübertragung oder deren Unterdrückung oder auch das Einspielen falscher Daten auszuschließen. Neben diesen Maßnahmen zur Sicherung der Integrität des Kommunikationsweges kann es ebenfalls erforderlich sein, die Vertraulichkeit des Kanals zu gewährleisten.

Die vorgenannten Eigenschaften (siehe auch Abschnitt 2) sind jedoch von den bestehenden Protokollierungs-Architekturen nicht oder nur unzureichend abgedeckt. Dies gilt sowohl für lokale Lösungen als auch insbesondere für netzwerkbasierte Ansätze (z.B. Syslog, SNMP, Microsoft Windows Event Log).

Analog zur Absicherung des Kommunikationspfades zur Protokollierungs-Instanz müssen dieselben Forderungen auch für den Transport und die Lagerung der Protokolldaten erhoben werden, ungeachtet der unmittelbaren Bestimmung der einzelnen Datensätze. Insbesondere die Lagerung der Datensätze verdient dabei besondere Aufmerksamkeit [9]. Sofern diese zulässig ist, besteht die Gefahr, dass die vorliegenden Daten manipuliert oder gelöscht werden. In Abhängigkeit vom Typ der gesicherten Protokolldaten muss darüber hinaus auch die Vertraulichkeit der Daten sichergestellt werden, damit nur berechtigte Bedarfsträger einen Zugriff auf die Protokolldaten erhalten.

Selbst bei der Verwendung kryptographischer Verfahren ergeben sich eine Reihe von Herausforderungen. Einerseits müssen bei langfristiger Lagerung Verfah-

ren eingerichtet werden, mit deren Hilfe die verwendeten kryptographischen Schlüssel oder Verfahren während der Lagerfrist gewechselt werden können, falls diese kompromittiert werden oder aus technischen oder organisatorischen Gründen nicht mehr als sicher angesehen werden können.

Darüber hinaus müssen die verwendeten kryptographischen Verfahren aber auch besonderen Anforderungen genügen. Aufgrund der potentiell großen Datenvolumina und der oft gegebenen Gleichförmigkeit der protokollierten Datensätze sind einige kryptologische Angriffe in diesem Zusammenhang besonders Erfolg versprechend. So müssen die verwendeten Verfahren zur Vertraulichkeits- und Integritätssicherung auch gegen gezielte Veränderungen am Chiffrat (*malleable ciphertext*) gesichert sein. Es muss verhindert werden, dass ein Angreifer ohne Kenntnis des Schlüsselmaterials für ihn günstige Änderungen des Klartextes allein durch Manipulation des Chiffretextes erzielen kann.

1.3 Organisatorische Anforderungen

Der Verwaltung der Protokollierungsmechanismen im täglichen Betrieb kommt eine entscheidende Rolle zu, da durch Veränderungen an der Konfiguration der Protokollierungs-Architektur oder anderen administrativen Tätigkeiten ähnliche und zum Teil deutlich schlechter verfolgbare Angriffsmöglichkeiten entstehen, als dies für externe Angreifer der Fall wäre.

Für die Verwaltung des Protokollierungs-Systems sowie des täglichen Betriebes (z.B. Archivierung älterer Datenbestände) ist daher eine präzise Trennung der Rollen erforderlich, die in sensiblen Bereichen bis hin zu einer Erzwingung des Vier-Augen-Prinzips für kritische Änderungen wie etwa der Löschung von Beständen reichen sollten.

Dabei muss vor allem sichergestellt sein, dass keine Rolle mit einem möglichen Interessenskonflikt (reguläre Nutzer des IT-Systems, vor allem aber auch sonstige administrative Rollen) einen uneingeschränkten Zugriff auf die Protokolldaten erlangen kann. Die entsprechenden Anforderungen lassen sich hierbei etwa durch ein erweitertes rollenbasiertes Zugriffsmodell (z.B. *RBAC with dynamic separation of duty*) abbilden [1].

Darüber hinaus ergeben sich indirekt aus den im folgenden Abschnitt beschriebenen Problemstellungen eine Reihe von Anforderungen, insbesondere auch an die physische Absicherung aller an der Protokollierungs-Architektur beteiligter Komponenten.

2 Probleme der Realisierung

Mit Ausnahmen weniger Großrechner-Systeme sind die vorgenannten Anforderungen an Protokollierungs-Architekturen nur schwer zu erfüllen. Möglichkeiten zur Manipulation lassen sich hierbei in hard- und softwareseitige Fragestellungen unterteilen:

2.1 Physische Absicherung und Hardware

In Abhängigkeit der einem Angreifer zur Verfügung stehenden Ressourcen und Qualifikation (d.h. etwa die Dauer des Zugangs zu einem zu untersuchenden Gerät, die zur Verfügung stehenden Werkzeuge und eventuell die Möglichkeit, ähnliche Geräte auch destruktiv zu analysieren) ist grundsätzlich festzuhalten, dass die Wirksamkeit selbst von gegen Manipulation gesicherter Geräte (*tamper resistant devices*) nur in der Erhöhung der zu betreibenden Aufwände besteht [2]. Dies ist insbesondere deswegen problematisch, weil Schutzmechanismen nach einem einmaligen Aufwand im Wiederholungsfall deutlich leichter zu umgehen sind. Ein Problem, das vor allem im Bereich der technischen Durchsetzung von Urheberrechten gravierend ist.

Physikalische Schutzmechanismen (z.B. versiegelte Gehäuse mit automatischen Selbst-Löschungsmechanismen bei der Erkennung von Manipulationsversuchen) sind für die im kommerziellen Bereich eingesetzten Rechnersysteme aufgrund der damit verbundenen erheblichen Kosten de facto nicht im Einsatz.

Ein potentieller Angreifer kann daher bei einem physischen Zugriff mit geeigneten Werkzeugen stets in den laufenden Betrieb des Rechnersystems eingreifen und so etwa die logischen Schutzmechanismen eines Betriebssystems auf verschiedene Arten umgehen. Dies kann z.B. durch das Starten eines Drittbetriebssystems und der Installation eines Trojanischen Pferdes auf dem Ziel-System von einem Wechseldatenträger erfolgen, dem Entfernen und Manipulieren der System-Laufwerke oder der Manipulation von Speicherschnittstellen zum Auslesen von Speicherinhalten oder der Kommunikation mit Peripheriegeräten. Derartige, teilweise trivial durchzuführende, Angriffe verletzen jedoch wesentliche Annahmen, die oftmals Sicherheits- und Protokollierungs-Architekturen zugrunde liegen.

Die hier genannten Einschränkungen gelten insbesondere auch für partielle Schutzmechanismen wie sie z.B. durch *Trusted Platform Modules* (TPM) realisiert werden. Da aus Kostengründen auch in diesen Fällen noch eine Reihe interner Schnittstellen ungesichert verbleiben, bestehen nach wie vor Verwundbarkeiten, die von hinreichend gut ausgestatteten Angreifern (z.B. mittels *in circuit emulators*) ausgenutzt werden können.

2.2 Einschränkungen auf Software-Seite

Auch auf Software-Seite bestehen zumal bei Einsatz von Standard-Betriebssystemen eine Reihe von Einschränkungen, welche die Erfüllung der vorgenannten Anforderungen an eine Protokollierungsarchitektur zunächst deutlich erschweren.

Bezüglich der verfügbaren Leistungsmerkmale ist zunächst die bei vielen Standard-Betriebssystemen nur unzureichend ausgeprägte Möglichkeit zur systematischen und korrekten Verwendung fortgeschrittener Zugriffskontrollmodelle (*Mandatory Access Control*, MAC) zu nennen. Insbesondere die in den Abschnitten 2 und 3 genannten Anforderungen lassen sich nur mittels MAC-Mechanismen, zumeist rollenbasierten Systemen effektiv realisieren [6]. Diese existieren für verschiedene Betriebssystem-Plattformen wie z/OS, OpenVMS und in einer Vielzahl von Ausprägungen für diverse Unix-Derivate und Linux, jedoch nicht für die Windows-Familie von Betriebssystemen.

In diesen Fällen sowie in der Mehrzahl der betriebenen Konfigurationen von IT-Systemen ist eine klare Trennung zwischen den funktionalen Rollen der allgemeinen Systemverwaltung und der Verwaltung der Protokollierungs-Instanzen nicht möglich, was vielfältige Möglichkeiten kaum nachzuweisender und nachzuvollziehender Manipulationen eröffnet.

Vielfach können jedoch auch durch den Einsatz legitimer Mittel Grundannahmen für den Betrieb von sicheren Protokollierungs-Architekturen in Frage gestellt werden. Durch den verstärkten Einsatz von Virtualisierungs-Plattformen wie VMWare

oder Xen [3] sind unter anderem die Kommunikationspfade der Protokollierungs-Infrastruktur gegebenenfalls von außen einsehbar und somit auch manipulierbar.

Hinzu kommt, dass gerade Endgeräte, die von regulären Nutzern verwendet werden, weiteren Risiken einer Kompromittierung von außen, etwa durch Trojanische Pferde oder Rootkits [4] ausgesetzt sind. Eine der Kernfunktionen dieser Schadsoftware ist insbesondere die Deaktivierung oder anderweitige Beeinträchtigung der Revisions-Subsysteme.

Neben den vielfältig vorliegenden Bedrohungen durch Verwundbarkeiten sowohl aufgrund von Entwurfs- als auch Implementierungsmängeln sind vor allem auch mangelnde einheitliche Schnittstellen für die Sammlung von Revisionsdaten problematisch. Daraus ergibt sich, dass eine Protokollierung an einer Vielzahl verschiedener Schnittstellen ansetzen muss, was nicht nur mit einem beträchtlichen Aufwand verbunden ist, der grundsätzlich bei Hinzunahme weiterer Anwendungsprogramme erneut überprüft werden muss, sondern bewirkt auch, dass die zur Erfassung der Protokolldaten erforderlichen Code-Pfade und das Gesamtvolumen des zu betrachtenden Codes massiv anwächst. Dies wiederum bedingt eine deutliche Abnahme der erreichbaren Gesamt-Vertrauenswürdigkeit einer Protokollierungs-Architektur.

Auch und gerade bei Erweiterungen von Standard-Systemen lässt sich daher effektiv nicht gewährleisten, dass sämtliche Zugriffspfade durch adäquate Sicherheitsmechanismen abgedeckt sind.

3 Fazit

Die Möglichkeiten, auf der Grundlage verfügbarer Systeme, insbesondere in heterogenen verteilten Systemen eine Protokollierungs-Architektur aufzubauen, welche den Anforderungen an Vertrauenswürdigkeit, Integrität, Vertraulichkeit und Authentizität sowie dem Schutz personenbezogener Daten genügen, sind eingeschränkt. Zwar existieren im Bereich der Unix-Derivate verschiedene technische Möglichkeiten, auf Software-Seite grundlegende Absicherungsmaßnahmen vorzunehmen. Die verfügbaren Protokollierungs- und Revisions-Systeme genügen jedoch nicht den zu stellenden Anforderungen, da diese zumeist Daten ungesichert übertragen und speichern.

Neue und bestehende Anforderung auf Grund rechtlicher Auflagen und Regulierungen, aber auch der Entwicklungen der zu überwachenden Systeme selbst lassen die Erarbeitung einer konsistenten Protokollierungs-Architektur und deren Einsatz im Feld aufgrund der bestehenden Risiken zunehmend als dringlich erscheinen. Zwar bestehen seit längerem entsprechende Möglichkeiten insbesondere im wehrtechnischen Bereich, da sichere Protokollierungsarchitekturen seit langem zu den Kernanforderungen von Prüfkriterien z.B. für sichere Betriebssysteme gehören [10], doch sind diese bislang nicht im zivilen Einsatz. Dies ist nicht zuletzt den erheblichen Kosten der Aufrechterhaltung der Protokollierungs- und Rechtearchitektur geschuldet, die eine streng hierarchische Organisationsstruktur voraussetzt. Es besteht daher erheblicher Forschungsbedarf, äquivalente Sicherheits- und Protokollierungsarchitekturen auch für den zivilen Bereich zu entwickeln.

Literatur

[1] AHN, G.-J., AND SANDHU, R. S. The RSL99 Language for Role-Based Separation of Duty Constraints. In *Proceedings of the Fourth ACM Workshop on Role-Based Access Control (RBAC 1999)* (Fairfax, VA, USA, Oct. 1999), ACM Press, pp. 43–54.

[2] ARNOLD, M., SCHMUCKER, M., AND WOLTHUSEN, S. D. *Techniques and Applications of Digital Watermarking and Content Protection*. The Artech House Computer Security Series. Artech House, Norwood, MA, USA, 2003.

[3] BARHAM, P., DRAGOVIC, B., FRASER, K., HAND, S., HARRIS, T., HO, A., NEUGEBAUER, R., PRATT, I., AND WARFIELD, A. Xen and the Art of Virtualization. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles* (Bolton Landing, NY, USA, Oct. 2003), ACM Press, pp. 164–177.

[4] BUSCH, C., AND WOLTHUSEN, S. D. *Netzwerksicherheit*. Spektrum Akademischer Verlag, Heidelberg, 2002.

[5] COMMON CRITERIA IMPLEMENTATION AND MAINTENANCE BOARD. *Common Criteria for Information Technology Security Evaluation: Part 2: Security Functional Components*. Common Criteria Implementation and Maintenance Board, Cheltenham, Glos, UK, 2005. Version 3.0, Revision 2, CCMB document 2005-07-002.

[6] FERRAIOLO, D. F., KUHN, D. R., AND CHANDRAMOULI, R. *Role-Based Access Control*. The Artech House Computer Security Series. Artech House, Norwood, MA, USA, 2003.

[7] MERCURI, R. T. On Auditing Audit Trails. *Communications of the Association for Computing Machinery 46*, 1 (Jan. 2003), 17–20.

[8] PICCIOTTO, J. The Design of an Effective Auditing Subsystem. In *Proceedings of the 1987 IEEE Symposium on Security and Privacy* (Oakland, CA, USA, Apr. 1987), IEEE Press, pp. 13–22.

[9] SCHNEIER, B., AND KELSEY, J. Secure Audit Logs to Support Computer Forensics. *ACM Transactions on Information and System Security 2*, 2 (May 1999), 159–176.

[10] UNITED STATES DEPARTMENT OF DEFENSE DoD 5200.28-STD: Department of Defense Trusted Computer System Evaluation Criteria (1985).