FREIE UNIVERSITÄT BERLIN
FACHBEREICH MATHEMATIK UND INFORMATIK – INSTITUT FÜR INFORMATIK

Konzept zum anonymen Audit unter Linux

Konrad Rieck
Herbst 2003

Zusammenfassung

Dieses Konzept beschreibt die Erhebung, Anonymisierung und Verarbeitung von Audit-Daten zu Forschungszwecken. Schwerpunkte sind die Qualität der faktischen Anonymisierung personenbezogener Daten sowie eine gute Gewährleistung der Verfügbarkeit der überwachten Rechnersysteme.

1 Einleitung

Im Rahmen einer Diplomarbeit sollen anonymisierte Daten aus dem Rechnerbetrieb des Instituts für Informatik an der Freien Universität Berlin zu Forschungszwecken an den Studenten Konrad Rieck ausgegeben werden. Die Diplomarbeit trägt den Titel *'Maschinelles Lernen in hostbasierten Intrusion-Detection-Systemen (IDS)'* und wird von Prof. Dr. Hannes Federrath von der Universität Regensburg betreut. Zielsetzung der Arbeit ist die Analyse verschiedener Methoden des maschinellen Lernens zur automatischen Angriffserkennung und die Implementierung intelligenter IDS-Module.

Die aus den Systemen des Rechnerbetriebs gewonnen Daten sollen zur Evaluation dieser intelligenten IDS-Modulen eingesetzt werden. Maßgeblich für die Qualität einer solchen Evaluation sind authentische Daten von realen Rechnersystemen, da nur so der direkte Rückschluss auf das Verhalten unter realen Bedingungen möglich ist [MHL+03].

Die folgenden Abschnitte beschreiben im Detail die einzelnen Schritte der Datenerhebung, Anonymisierung und Verarbeitung auf Linux-Rechnersystemen. Sie sollen dem Rechnerbetrieb und Studenten als Referenz dienen und aufzeigen, welche Maßnahmen zum Schutz personenbezogener Daten und der Verfügbarkeit des Rechnerbetriebs eingesetzt werden.

Die vorgestellten Verfahren wurden mit Ingrid Pahlen-Brandt, der Datenschutzbeauftragten der Freien Universität Berlin, und Ingmar Camphausen vom Rechnerbetrieb des Fachbereichs für Mathematik und Informatik abgestimmt.

Die überwachten Rechnersysteme sollen mittels Schildern an den Monitoren und durch einen Eintrag in der „Message of the Day“ gekennzeichnet werden, so dass die Nutzer Kenntnis von der Erhebung, Anonymisierung und Weiterverarbeitung der Daten erlangen.

Ferner sollen nicht alle Rechnersysteme eines Typs überwacht werden, so dass eine Nutzung eines überwachten Systems nicht zwingend nötig ist.

2 Datenerhebung

Die zur Evaluation benötigten Daten sollen mittels *Betriebssystemaudit* auf Linux-Rechnersystemen erhoben werden. Diese Form des Audit erlaubt das Protokollieren von erfolgreichen und fehlgeschlagenen Zugriffsversuchen auf Ressourcen des Betriebssystems [Sob99]. Die Zugriffsversuche werden hierbei über die vom Betriebssystem als Schnittstelle zu den Ressourcen zur Verfügung gestellten *Systemaufrufe* ermittelt.

Für die Erhebung soll das *Secure Auditing for Linux (SAL)* System eingesetzt werden, das vom amerikanischen Verteidigungsministerium entwickelt wird. Das System wird vollständig in den Dokumenten [WG02, WG03] spezifiziert und diskutiert. SAL bietet die folgenden Vorteile gegenüber anderen Systemen wie SNARE [PC$^+$02], dem Linux BSM [Ban01] oder dem Linux Trace Toolkit [Y$^+$02]:

1. Audit entsprechend der internationalen Common Criteria [CC99]
2. Guter Schutz der Verfügbarkeit durch hohe Leistungsfähigkeit
3. Audit von allen Systemaufrufen innerhalb des Systemkerns

Während ein Programm auf einem überwachten Rechnersystem ausgeführt wird, protokolliert SAL zu jedem Systemaufruf die in Tabelle 1 gezeigten Datenfelder. Es ist entscheidend, dass SAL nur die Systemaufrufe selbst und nicht die von den Systemaufrufen verarbeiteten Daten aufzeichnet. So wird z. B. der Systemaufruf `open` protokolliert, nicht jedoch welche Datei geöffnet wird und welche Daten ein- oder ausgelesen werden.

Die in Tabelle 1 dargestellten Felder werden nach § 3 Abs. 1 des Bundesdatenschutzgesetzes (BDSG) als personenbezogenen gewertet und unterliegen daher dem Datenschutz. Der Personenbezug lässt sich durch drei Aspekte in der folgenden Aussage zusammenfassen:

> Die von SAL erhobenen Audit-Daten protokollieren *welcher* Nutzer *welches* Programm zu *welcher* Zeit nutzt.

Zudem wird mit den Systemaufrufen und deren Status auch protokolliert, *wie* sich ein Programm verhält. Das Verhalten eines Programms ohne die vom Programm verarbeiteten Daten erlaubt keine Rückschlüsse auf den Nutzer und bedarf daher keiner weiteren Betrachtung.

Feld	Inhalt des Feldes
`syscall`	Name (bzw. Nummer) des Systemaufrufs
`status`	Status des Systemaufrufs (erfolgreich oder fehlgeschlagen)
`time`	Uhrzeit des Systemaufrufs in Mikrosekunden
`program`	Name (bzw. Pfad) des aktiven Programms
`pid`	Kennung des aktiven Programms
`uid`	Kennung des aktiven Nutzers

Tabelle 1: Von SAL pro Systemaufruf erhobene Daten

3 Anonymisierung

Die erhobenen Daten sollen vor einer Speicherung ein Anonymisierungsverfahren durchlaufen, welches als Teil des in der Diplomarbeit vorgestellten IDS entwickelt wird. Da eine absolute Anonymisierung in der Realität gar nicht oder äußerst schwer zu erreichen ist, sollen die Daten *faktisch* entsprechend des § 3 Abs. 6 BDSG anonymisiert werden:

> „Anonymisieren ist das Verändern personenbezogener Daten derart, dass die Einzelangaben über persönliche oder sachliche Verhältnisse nicht mehr oder nur mit einem unverhältnismäßig großen Aufwand an Zeit, Kosten und Arbeitskraft einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können.“

3.1 Verwendetes Anonymisierungsverfahren

Als personenbezogene Informationen sind in den von SAL erhobenen Audit-Daten wie bereits beschrieben die drei folgenden Angaben sowie Kombinationen aus diesen zu werten:

1. Der Name des aktiven Programms
2. Die Kennung des aktiven Nutzers
3. Die Uhrzeit eines Datensatzes

Es kann festgestellt werden, dass der Name des ausgeführten Programms maßgeblich für die beabsichtige Evaluation von IDS-Modulen ist, da nur der Name die Zuordnung von einem Programm zu dessen Verhalten ermöglicht. Eine Anonymisierung kann folglich nur an Punkt 2 und 3 erfolgen.

Die folgende Aufstellung beschreibt, wie die einzelnen Felder der SAL Audit-Daten modifiziert werden, um eine Anonymisierung zu erreichen. Wenn möglich werden die Daten vollständig gelöscht, in anderen Fällen werden sie stark reduziert.

– `time` – Uhrzeit des Systemaufrufs in Mikrosekunden
Die Uhrzeit eines Systemaufrufs wird auf einen Bereich von 60 Sekunden reduziert. Weitere Informationen wie Minuten, Stunden, Tage, Monate, usw. werden gelöscht.

– `program` – Namen des aktiven Programms
Es werden nur Datensätze von Systemprogrammen protokolliert. Datensätze über von Nutzern selbst installierte oder implementierte Programme, die persönliche Informationen enthalten können, werden entfernt.

– `uid` – Kennung des aktiven Nutzers
Die Kennung des Nutzers wird vollständig gelöscht. Sowohl effektive als auch reale Kennungen werden auf den Wert 0 gesetzt. Welcher Nutzer ein Programm ausgeführt hat, ist für die Evaluation von Programmverhalten in IDS irrelevant.

– `pid` – Kennung des aktiven Programms
Die Kennung wird mittels der XOR-Funktion mit einem zufälligen Schlüssel chiffriert, der alle 24 Stunden gelöscht und neu gewählt wird. Eine Entschlüsselung ohne diesen Schlüssel ist informationstheoretisch unmöglich [Sch96].

Feld	**Art der Anonymisierung**
`syscall`	*keine Anonymisierung*
`status`	*keine Anonymisierung*
`time`	Reduktion auf Bereich von 60 Sekunden
`program`	Reduktion auf Systemprogramme
`uid`	Löschung von allen Nutzer- und Gruppenkennungen
`pid`	XOR-Verschlüsselung mit zufälligem Schlüssel

Tabelle 2: Anonymisierung der einzelnen Felder

Tabelle 2 und 3 zeigen zusammenfassend wie die einzelnen Felder der von SAL erhobenen Audit-Daten anonymisiert werden und welche Anonymisierungsmethoden zum Einsatz kommen.

Verfahren / Feld	`time`	`program`	`uid`	`pid`
Datenreduktion	○	○		
Vollständige Löschung			○	
Irreversible Verschlüsselung				○

Tabelle 3: Eingesetzte Anonymisierungsmethoden

Die folgenden Abbildung verdeutlichen noch einmal visuell wie sowohl die Kennung des aktiven Nutzers als auch die Uhrzeit der Datensätze anonymisiert wird.

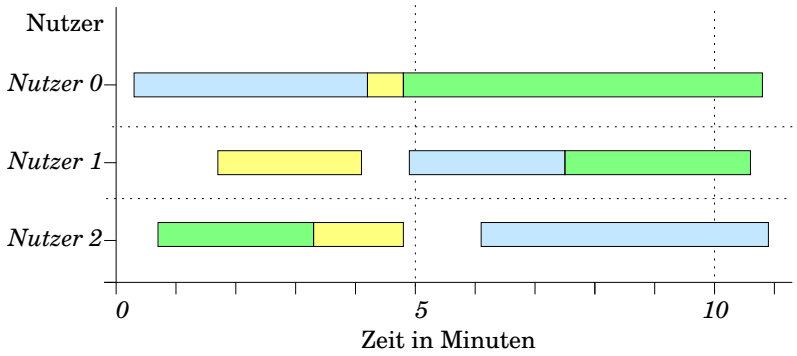

Abbildung 1: Visuelle Darstellung der ursprünglichen Audit-Daten

Abbildung 1 zeigt drei Nutzer, die über einen Zeitraum von 10 Minuten verschiedene Programme ausführen. Anhand der Abbildung lässt sich eindeutig entscheiden welcher Nutzer welches Programm zur welcher Zeit ausführt.

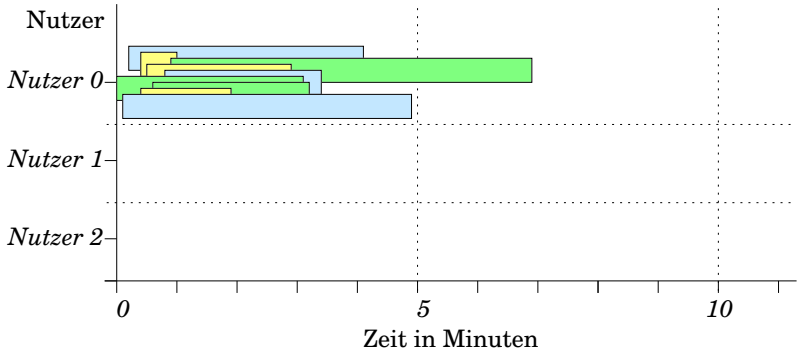

Abbildung 2: Visuelle Darstellung der anonymisierten Audit-Daten

Abbildung 2 zeigt die anonymisierten Audit-Daten. Obwohl die Reihenfolge der Systemaufrufe erhalten bleibt, lassen sich die Programme sowohl zu den Nutzern als auch zur Zeit nicht mehr eindeutig zuordnen.

Es ist zu bemerken, dass die Abbildung 2 nicht die zusätzliche Verschlüsselung der Kennungen von aktiven Programmen und die Reduktion der Namen der Programme darstellt.

3.2 Einsatz des Anonymisierungsverfahrens

Das beschriebene Anonymisierungsverfahren wird sofort nach Erhebung der Audit-Daten eingesetzt, so dass auf den überwachten Rechnersystemen zu keinem Zeitpunkt personenbezogene Daten vorgehalten werden.

Ferner kann eine zweite Anonymisierung auf einem internen System des Rechnerbetriebs erfolgen, um eventuelle Manipulation oder Fehlfunktionen der ersten Anonymisierungsstufe zu korrigieren.

3.3 Angriffe gegen die Anonymisierung

Als Angriff gegen eine Anonymisierung gilt jedes Verfahren, das es erlaubt, die anonymisierten Daten vollständig oder teilweise zu *deanonymisieren*. Generell kann zwischen *zeitversetzten* und *zeitgleichen* Angriffen unterschieden werden.

3.3.1 Zeitversetzter Angriff

Ein um mehr als 24 Stunden zeitversetzter Angriff auf Audit-Daten, die entsprechend dem vorherigen Abschnitt anonymisiert wurden, ist unmöglich. Die Audit-Daten können in diesem Fall als absolut anonym gewertet werden.

Grund hierfür ist die Tatsache, dass keines der anonymisierten Felder deanonymisiert werden kann. Die Kennung des aktiven Nutzers ist vollständig gelöscht, die Uhrzeit kann nicht mehr rekonstruiert werden, und der zufällige Schlüssel für die Kennungen des aktiven Programms wird im 24-Stundenrhythmus gelöscht und neu gewählt.

3.3.2 Zeitgleicher Angriff

Bei einem zeitgleichen Angriff auf die Anonymisierung kann ein Angreifer selbst Daten erheben und diese für eine Rekonstruktion nutzen. Um das beschriebene Anonymisierungsverfahren anzugreifen, ist es notwendig, sämtliche anonymisierten Felder zu deanonymisieren. Eine solcher Angriff ist möglich, wenn der Angreifer folgende Aktionen ausführt:

1. Kompromittierung des überwachten Systems
2. Minütliche Überwachung der aktiven Prozesse und aktiven Nutzer
3. Minütliche Protokollierung der Uhrzeit
4. Rekonstruktion der Uhrzeiten
5. Rekonstruktion der Programmkennungen

Ein zeitgleiche Angriff ist allerdings äußerst unwahrscheinlich. Zum einen wird ein sehr großer Aufwand zur Erhebung der zusätzlichen Daten nötig, der schwer vor dem Rechnerbetrieb zu verbergen ist, zum anderen enthalten die zusätzlich erhobenen Daten bereits nahezu alle Informationen der anonymisierten Audit-Daten, so dass eine Deanonymisierung eigentlich nutzlos ist.

3.3.3 Bewertung der Angriffe

Nach § 3 Abs. 6 des BDSG sind die Audit-Daten anonym, da sowohl für den zeitversetzten Angriff als auch für den zeitgleichen Angriff keine Deanonymisierung oder eine solche nur mit unverhältnismäßig hohem Aufwand und geringem Informationsgewinn möglich ist.

Nach [MW00] kann geschlossen werden, dass das Risiko einer Deanonymisierung der Audit-Daten infolge des Inhalts und Verwendungskontexts so weit gemindert ist, dass den Nutzern der Rechnersysteme dieses Risiko zugemutet werden kann.

4 Datenverarbeitung

Während der vorherige Abschnitt den Schutz personenbezogener Daten mittels Anonymisierung beschreibt, soll nun die auf die Anonymisierung folgende Datenverarbeitung dokumentiert werden. Im Vordergrund steht hierbei der Schutz der Verfügbarkeit der Rechenleistung und des Hintergrundspeichers der überwachten Systeme. Die für die Datenverarbeitung nötige Software wird im Rahmen der Diplomarbeit entwickelt und nutzt die Schnittstellen des SAL Systems.

Die anonymisierten Audit-Daten werden über einen Zeitraum von 12 Stunden erzeugt und für diese Periode auf den überwachten Systemen zwischengespeichert. Am Ende eines solchen Zeitraums werden die Daten von den überwachten auf ein internes Rechnersystem des Rechnerbetriebs übertragen.

Während der Zwischenspeicherung sorgen die folgenden Maßnahmen für einen Schutz der Verfügbarkeit des Hintergrundspeichers:

1. Alle Daten werden mittels der Z-Kompressionsbibliothek [GA03] vor dem Speichervorgang komprimiert und sind sowohl auf dem Hintergrundspeicher als auch beim Transfer über das Netzwerk auf unter 5% der ursprünglichen Größe reduziert.

2. Die für die Datenverarbeitung eingesetzte Software überwacht in regelmäßigen Abständen die verbleibende Kapazität des Hintergrundspeichers und bricht die Datenerhebung ab, sobald ein Schwellwert von 95% Auslastung erreicht ist.

Die folgenden Maßnahmen dienen dem Schutz der Verfügbarkeit der Rechenleistung der überwachten Systeme:

1. Die für Datenerhebung und Anonymisierung entwickelte Software beendet die Verarbeitung, so bald eine übermäßige Rechenlast auf dem System auftritt. Eine solche Last konnte in Experimenten nur durch böswillige und zielgerichtete Angriffe erreicht werden.

2. Der Erhebungs-, Anonymisierungs- und Kompressionsvorgang wird auf mehrere Ausführungsstränge (Threads) verteilt, so dass der gesamte Vorgang beschleunigt ausgeführt werden kann.

3. Die Anonymisierung wird vollständig ohne aufwendige kryptographische Funktionen realisiert, so dass sie sehr effizient arbeitet.

Die anonymisierten Daten werden nach ihrer Sammlung an den Studenten Konrad Rieck ausgegeben und ausschließlich zu Forschungszwecken genutzt.

5 Schlussbetrachtung

Das vorgestellte Konzept zum anonymen Audit unter Linux beschreibt Verfahren, die einen hohen Schutz personenbezogener Daten durch eine Anonymisierung gewährleisten. Die erzeugten Daten sind nach einer Betrachtung potentieller Angriffe im Sinne des Bundesdatenschutzgesetzes anonym.

Ferner gewährleisten die vorgestellten Verfahren zur Erhebung und Datenverarbeitung den Schutz der Verfügbarkeit der überwachten Rechnersysteme. Verschiedene Softwaremaßnahmen verhindern, dass die Verfügbarkeit der Rechenleistung und des Hintergrundspeichers stark beeinträchtigt wird.

Die Nutzung der überwachten Systeme geschieht freiwillig, da die Nutzer in Kenntnis gesetzt werden und Rechnersysteme ohne Überwachung zur Verfügung stehen.

Fragen bezüglich weiterer Details dieses Konzepts sind direkt an Konrad Rieck unter der folgenden Adresse zu richten.

Konrad Rieck
Email: `rieck@inf.fu-berlin.de`

Fragen bezüglich des Rechnerbetriebs sind an die Technik des Instituts für Informatik der Freien Universität zu richten.

Rechnerbetrieb der Informatik
Email: `staff@inf.fu-berlin.de`

6 Literaturverzeichnis

[CC99] *Common Criteria for Information Technology Security Evaluation – Part 2*. NIST, NSA (USA), CSE (Kanada), BSI (Deutschland), NNCSA (Niederlande), CESG (England), SCSSI (Frankreich), August 1999.

[MHL+03] Peter Mell, Vincent Hu, Richard Lippmann et al. *An Overview of Issues in Testing Intrusion Detection Systems*. National Institute of Standards and Technology, 2003. *(http://csrc.nist.gov/publications/nistir/nistir-7007.pdf)*

[MW00] Rainer Metschke, Rita Wellbrock. *Datenschutz in Wissenschaft und Forschung*. Berliner Beauftragter für Datenschutz und das Recht auf Akteneinsicht, Hessischer Datenschutzbeauftragter, Berlin, Deutschland, November 2000.

[Sch96] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, New York, USA, 2. Edition, 1996. ISBN 0-47-111709-9

[Sob99] Michael Sobirey. *Datenschutzorientiertes Intrusion Detection*. Datenschutz und Datensicherheit Fachbeiträge. Vieweg-Verlag, Wiesbaden, Deutschland, 1999. ISBN 3-528-05704-1

[WG02] Wiliam Wolfe, Javier Godinez. *Secure Auditing for Linux (SAL) Software Requirements Specification (SRS)*. Space and Naval Warfare Systems Center, San Diego, USA, Juli 2002. *(http://secureaudit.sourceforge.net/docs/LINUX-SRS-new-v22.pdf)*

[WG03] Wiliam Wolfe, Javier Godinez. *Secure Auditing for Linux (SAL) Software Design Document*. Space and Naval Warfare Systems Center, San Diego, USA, Februar 2003. *(http://secureaudit.sourceforge.net/docs/SAL_SDD_1.pdf)*

7 Internetreferenzen

[Ban01] Jeremy Banford. *The Linux Basic Security Module Project*. University of California at Davis, USA, 2001. *(http://linuxbsm.sourceforge.net)*

[GA03] Jean-Loup Gailly, Mark Adler. *zLib – Compression Library*, 2003. *(http://www.gzip.org/zlib)*

[PC+02] Leigh Purdie, George Cora et al. *SNARE – System Intrusion Analysis and Reporting Environment*. Interselect Alliance Pty Ltd, Canberra, Australia, 2002. *(http://www.intersectalliance.com/projects/Snare)*

[Y+02] Karim Yaghmour et al. *Linux Trace Toolkit*. Opersys Inc., Quebec, Kanada, 2002. *(http://www.opersys.com/LTT)*