Ein Ansatz zur visuellen Analyse von Audit-Daten

Sebastian Schmerl

Brandenburgische Technische Universität Cottbus
Lehrstuhl Rechnernetze und Kommunikationssysteme
Postfach 10 13 44, 03013 Cottbus
sbs{at}informatik.tu-cottbus.de

Abstract: Intrusion-Detection-Systeme (IDS) haben sich als wichtiges Instrument für den Schutz informationstechnischer Systeme erwiesen. Die meisten heute eingesetzten IDS realisieren eine Signaturanalyse. Fehlalarme sind bei dieser Analysetechnik per se ausgeschlossen, jedoch zeichnet die Realität ein anderes Bild. Ursache dafür sind unscharfe Signaturen. Diese resultieren aus der Komplexität des Signaturentwurfs. Besonders die Ableitung von Signaturen aus Exploits erweist sich als schwierig. Für diesen Vorgang bildet die manuelle Audit-Daten-Analyse die Basis. Dieser Beitrag zeigt einen alternativen Ansatz zur Audit-Daten-Repräsentation mit dem Ziel, den Analyse-Prozess für den Signaturprogrammierer zu vereinfachen. Dazu werden Audit-Daten sowie vorhandene Zusammenhänge grafisch dargestellt. Anhand einer prototypischen Implementierung dieses Ansatzes konnten die Stärken dieser Darstellungsform verdeutlicht werden.

1 Motivation

Intrusion Detection Systeme (IDS) haben sich als ein wichtiges Instrument für den Schutz informationstechnischer Ressourcen erwiesen. Es existieren zwei grundlegende Strategien, die Anomalieerkennung und die Signaturanalyse. Erstere bietet den Vorteil, unbekannte Sicherheitsverletzungen erkennen zu können, führt aber aufgrund geringer Erkennungsgenauigkeit häufig zu unakzeptabel hohen Fehlalarmraten. Dagegen untersuchen signaturbasierte Analyseverfahren Protokoll- bzw. Audit-Daten nach Mustern bekannter Sicherheitsverletzungen, den Signaturen. Die Wirksamkeit dieses Verfahrens hängt entscheidend von der Genauigkeit der verwendeten Signaturen ab. Ungenaue Signaturen schränken die Erkennungsmächtigkeit eines IDS stark ein. Die Ursachen der Erkennungsunsicherheit sind hauptsächlich auf der Ebene der Signaturableitung zu suchen. Besonders die Ableitung aus Exploits erweist sich als komplex. Hierbei müssen Spuren, die ein Angriff in einem System hinterlässt, identifiziert und charakteristische Zusammenhänge bestimmt werden. Dies setzt eine manuelle Analyse der Audit-Daten voraus. Ferner müssen zur Signaturverifikation erkannte Angriffe bzw. IDS-Fehlalarme durch eine manuelle Analyse der Audit-Daten untersucht werden. Diese grundlegende Tätigkeit ist zeitaufwendig, schwierig und mühsam. Hauptgründe dafür sind die sehr feingranular auf mehrere Quellen/Stellen verteilte Informationsflut und die nicht ergonomische Betrachtung der Audit-Daten. Typische Werkzeuge bieten nur eine Sicht auf die Datenfülle bzw. keine Abstraktionsmöglichkeiten. Gerade Abstraktion, also das zielgerichtete Hervorheben von relevanten Zusammenhängen zwischen Audit-Ereignissen und

das gleichzeitige Verbergen irrelevanter Daten bildet einen entscheidenden Aspekt, um den Signaturprogrammierer bei der Analyse zu unterstützen. Ein weiterer Aspekt ist die Darstellung der Audit-Daten. Typisch ist eine textuelle Repräsentation, diese kann aber Zusammenhänge zwischen Audit-Ereignissen nur ungenügend abbilden. Somit ist diese Darstellungsform als suboptimal anzusehen, da sie aufgrund der Unübersichtlichkeit irritieren kann. Oft kommt es somit zu Fehleinschätzungen und in deren Konsequenz zu falschen Signaturen. Diese Nachteile könnten durch eine grafische Präsentation im mehrdimensionalen Raum behoben werden.

Es existieren mehrere Ansätze zur Visualisierung von Netzwerkverhalten. Diese stellen zeitnah veränderliche Parameter wie z.B. Verbindungen, Durchsatz, verlorene Pakete und IDS-Alarme dar. Als Vertreter können *SilentRunner* [3], *NIVA* [4] und *DNVS* [9] genannt werden. Zur graphischen Darstellung für forensische Untersuchungen bzw. für die Analyse von Exploits existieren nur wenige Ansätze. *Visual Audit Browser* [2] ist ein Versuch Audit-Daten graphisch für den Review-Prozess aufzuarbeiten. Allerdings ist keine Nutzerinteraktion möglich und die Darstellung im 2D-Raum ist eher suboptimal.

Dieser Beitrag skizziert ein Werkzeug zur 3D-Darstellung von Audit-Daten mit der Möglichkeit, diese Darstellungen interaktiv zu explorieren. Der Nutzer kann beliebige Sichten auf die Daten generieren, sowie Zusammenhänge bzw. Abhängigkeiten erforschen und visualisieren. Abschnitt 2 skizziert zunächst einen allgemeinen Ansatz zur Visualisierung von Audit-Daten, bevor Abschnitt 3 eine konkrete Realisierung vorstellt. Im Anschluss wird im Abschnitt 4 beispielhaft ein Einsatzszenario des Werkzeuges bei der Audit-Daten-Analyse erläutert. Der Beitrag schließt mit einer Zusammenfassung.

2 Erstellung virtueller Audit-Datenräume

Grundsätzlich kann die Erstellung eines virtuellen Informationsraumes in drei Phasen gegliedert werden: *Erschließung des Datenraumes*, *Analyse der Objektähnlichkeiten* und *Transformation der Objektrelationen*. Diese drei Arbeitsschritte werden im Zusammenhang mit der Verarbeitung von Audit-Daten im Folgenden näher beschrieben.

Inhaltliche Erschließung: Um den Datenraum inhaltlich zu erschließen, müssen die zu analysierenden Audit-Daten aus ihren möglicherweise unterschiedlichen Quellen ausgelesen und klassifiziert und anschließend einheitlich abgelegt werden. Audit-Ereignisse des gleichen Typs werden dabei in identischen Strukturen abgelegt, so dass in den folgenden Phasen eine einheitliche Interpretation ermöglicht wird.

Quantitative Analyse von Objektähnlichkeiten: In dieser Phase werden Abhängigkeiten bzw. Ähnlichkeiten auf Grundlage von Metriken zwischen *Objekten* festgelegt. Im Kontext der Audit-Analyse bilden die Audit-Ereignisse die darzustellenden Objekte. Eine *Metrik* weist jedem Paar von Audit-Ereignissen eine Distanz zu. Diese Distanz bildet ein Maß für *Relationen* zwischen Audit-Ereignissen, die Ähnlichkeiten, Abhängigkeiten bzw. Zusammenhänge beschreibt. Charakteristische Metriktypen in diesem Kontext berechnen Abstände abhängig von zeitlichen, inhaltlichen oder semantischen Zusammenhängen. Eine konkrete Metrik kann z.B. die Distanz zwischen konträren Ereignissen (wie Fork-/Exit-Ereignisse) beeinflussen.

Raumbezogene Transformation: Um die in der quantitativen Analyse definierten Relationen zu visualisieren, werden die Audit-Daten in den Präsentationsraum transformiert. Jedes Audit-Ereignis wird als Objekt im Präsentationsraum dargstellt. Um die berechneten Distanzen bzw. Relationen zwischen den Audit-Daten darzustellen, können *Darstellungsparameter* wie räumliche Nähe, verbindende Kanten, Objektfarbe, -form oder -größe genutzt werden. Da diese Darstellungsparameter gleichzeitig visualisierbar sind, ist auch die Darstellung unabhängiger Relationen möglich.

3 ADO: Ein Werkzeug zur Darstellung und Exploration von Audit-Daten

Zur automatisierten Realisierung der im Abschnitt 2 beschriebenen Phasen wurde ein Werkzeug (**A**udit**D**aten**O**kular) implementiert. Anhand dieses Werkzeuges sollen die Vorzüge dieser Darstellung der Audit-Daten empirisch evaluiert werden. In dieser konkreten Realisierung werden BSM-Audit-Logs [6] als Eingabedaten genutzt. Der Prototyp besteht aus den folgenden drei Komponenten: dem *BSM-Sensor*, der Analyse- und Transformationskomponente *Cosmos* und der Präsentationskomponente *3DO*. Im Folgenden werden die Komponenten *Cosmos* und *3DO* näher beschrieben.

Analyse und Transformations-Komponente *Cosmos*: Sie dient der Definition von Metriken bzw. Relationen über Audit-Ereignissen. Es können GUI-unterstützt boolesche Bedingungen definiert und gewichtet werden. Paarweise werden dann die Audit-Ereignisse mit den Bedingungen evaluiert. Ist eine Bedingung erfüllt, so wird die Distanz zwischen diesen Audit-Ereignissen um den zugehörigen Gewichtungswert erhöht. Die Bedingungen können zwischen beliebigen Merkmalen bzw. Attributen von Audit-Ereignissen definiert werden. Nachdem die Distanzberechnungen abgeschlossen sind, erfolgt die Transformation in den 3D-Raum. Hierbei wird der Abstand zwischen zwei Einträgen als Grad ihrer Zusammengehörigkeit interpretiert, d.h. ein Ereignispaar, das einen hohen Abstand besitzt, erfüllt starke oder mehrere gering gewichtete Bedingungen. Zusammenhänge zwischen Audit-Ereignissen werden durch räumliche Nähe dargestellt. Dazu wird ein Cluster-Algorithmus [7] genutzt. Die Funktionsweise des Algorithmus kann abstrahiert als Energieminimierung in einem Federsystem angesehen werden. Die zahlreichen Definitionsmöglichkeiten von Bedingungen, die Abhängigkeiten beschreiben, und die zugehörigen Einflusswerte erlauben die Erstellung beliebiger Sichten auf die Audit-Daten. Neben der räumlichen Nähe können Kanten genutzt werden, um Relationen zwischen einem einzelnen Audit-Ereignis und allen anderen darzustellen. Dafür werden die Relationen nur in Verbindung mit einem selektierten Objekt evaluiert. Metriken können alternativ durch Objektfarbe, -form oder -größe visualisiert werden.

Visualisierungskomponente *3DO*: *3DO* stellt die von *Cosmos* generierten dreidimensionalen Audit-Daten-Welten dar. Dem Nutzer stehen zahlreiche Navigationsfunktionen und Selektionsfunktionen zur Daten-Explorierung zur Verfügung [8]. Bei der Selektion von Objekten werden weitergehende Informationen zu den zugehörigen Audit-Ereignissen dargestellt. Wird ein einzelnes Objekt selektiert, besteht die Möglichkeit zusätzliche Bedingungen im Zusammenhang mit diesem Audit-Ereignis zu definieren. Das ermöglicht die Anpassung der aktuellen Sicht an das aktuelle Analyseziel.

4 Einsatzszenario / Anwendungsbeispiel

Um einen Eindruck der Eignung von virtuellen Informationsräumen bzw. dem vorgestellten Werkzeug *ADO* zur Audit-Daten-Analyse zu erhalten, wird exemplarisch die Analyse einer BSM-Trail mit 109 Ereignissen erläutert. In dieser Audit-Trail wurde von einem signaturbasierten IDS eine Login-Attacke erkannt. Die zugehörige Signatur löst einen Alarm aus, falls fünf fehlerhafte Anmeldeversuche eines Nutzers binnen von einer Minute protokolliert werden. Bei der Analyse soll das Systemverhalten innerhalb dieses Zeitabschnitts näher untersucht werden. Dazu wird folgendermaßen vorgegangen:

Schritt 1: Um einen Überblick über die Nutzer und ihre Aktivitäten zu erhalten, werden zunächst Ereignisse bzgl. der zugehörigen Nutzerkennung (User-ID) in Zusammenhang gesetzt. Dazu wird eine Metrik *A* definiert, die Audit-Ereignisse auf gleiche User-IDs überprüft. Die Metrik ist mit dem Faktor 10 gewichtet und wird durch räumliche Nähe (Clusterung) und durch Kanten visualisiert. Abb. 1 zeigt die resultierende Darstellung.
Interpretation: Erkennbar ist eine große Menge von Ereignissen die bzgl. der User-ID homogenen sind. Diese werden durch den Cluster *A* in der Mitte symbolisiert. Derartige Cluster entsprechen typischen Ereignisfolgen, die aus normalen Nutzerverhalten resultieren. Ferner ist eine geringe Anzahl von Ereignissen verschiedener Nutzer außerhalb des großen Clusters erkennbar. Das ist ein eher ungewöhnliches Nutzerverhalten.

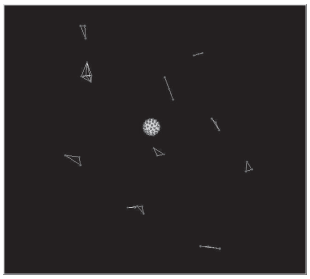

Abb. 1: Metrik A

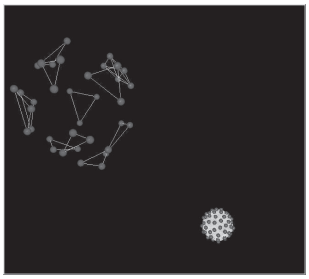

Abb. 2: Metrik A und B

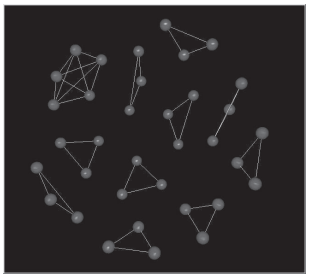

Abb. 3: Metrik A über Anmeldeereignisse

Schritt 2: Um diese anormalen Ereignisse näher zu untersuchen, wird zusätzlich zu der Metrik *A* aus Schritt 1 eine weitere Metrik *B* definiert. Diese bewertet Typübereinstimmungen zwischen Audit-Ereignissen mit dem Ziel nach Ereignistypen Cluster zu bilden. Damit Metrik *A* dominierend bleibt, wird Metrik *B* lediglich mit dem Faktor 1 gewichtet. Somit beeinflusst sie die Gruppierung nur zu einem Zehntel. Ferner wird für Metrik *B* nur die räumliche Nähe zur Visualisierung genutzt. Abb. 2 zeigt die entstehende Szene.
Interpretation: Der Cluster *A* wird durch die verschiedenen Ereignistypen, die er vereint, nur leicht aufgefächert, da die Metrik A die Sicht auf die Daten dominiert. Ferner rücken die bzgl. Nutzerkennung heterogenen Ereignisse näher zusammen. Ursache dafür ist die Übereinstimmung dieser Ereignisse bzgl. des Typs.

Schritt 3: Durch Selektieren der Objekte, wobei alle weiteren Ereignisinformationen angezeigt werden, kann erkannt werden, dass der Cluster *B* (links oben in Abb. 2) nur Ereignisse vom Typ fehlerhafter Anmeldeversuch enthält. Da Cluster *B* somit die Ereignisse des eigentlichen Analyseziels gruppiert, werden zur Abstraktion die Ereignisse des

Clusters *A* selektiert und von der weiteren Analyse ausgeschlossen. Eine erneute Analyse der Metrik *A* auf der reduzierten Ereignismenge erzeugt das in Abb. 3 dargestellt Bild.
Interpretation: Die Anmeldeereignisse mit gleicher User-ID sind durch Linien verbunden. Deutlich sichtbar sind die fünf fehlerhaften Anmeldeversuche (links oben), die vom IDS als Alarm gemeldet wurden. Ferner ist erkennbar, dass 30 weitere fehlerhafte Anmeldeversuche vorliegen. Es wurde versucht, sich jeweils dreimal unter zehn verschiedenen Nutzernamen anzumelden.

Bei der Audit-Trail-Analyse ist neben dem IDS-Alarm weiteres verdächtiges Nutzerverhalten identifiziert worden. Bei einer anschließenden tiefgründigeren Untersuchung der Trail könnte nun der Ursprung dieser fehlerhaften Anmeldeversuche exploriert werden.

5 Schlussbemerkungen

Die manuelle Analyse von Audit-Daten stellt eine grundlegende Tätigkeit beim Signaturentwurf und forensischen Untersuchungen dar. Um diesen Vorgang zu unterstützen, wurde eine alternative Repräsentation von Audit-Daten im 3d-Raum vorgestellt. Ausgehend von den Vorteilen dieser Art der Datenrepräsentation wurde ein allgemeingültiger Ansatz zur Erstellung von virtuellen Informationsräumen für Audit-Daten beschrieben. Dieser Ansatz wurde unter Verwendung einer erstellten Implementierung experimentell evaluiert. Es wurde gezeigt, dass sich Zusammenhänge bzw. Abhängigkeiten zwischen Audit-Ereignissen gut untersuchen und visualisieren lassen. Um diesen viel versprechenden Ansatz weiter auszubauen, ist eine stärkere Kopplung zwischen der Audit-Daten-Repräsentation und der Signatur wünschenswert. Zum einen könnten aus identifizierten Zusammenhängen automatisch Signaturen oder Teile davon abgeleitet werden. Zum anderen ist für die forensische Untersuchung von IDS-Fehlalarmen eine Visualisierung der Signatur im Zusammenhang mit den Audit-Daten denkbar.

6 Literaturverzeichnis

[2] J. Hoagland; C. Wee; K. Levitt: Audit Log Analysis Using the Visual Audit Browser Toolkit. UC Davis Computer Science Technical Report CSE-95-11; 1995.

[3] A. Lawson: eTrust Network Forensics. Computer Associates International Inc.; April 2004.

[4] K. Nyarko; T. Capers; C Scott; K Ladeji-Osias: Network Intrusion Visualisation with NIVA, an Intrusion Detection Visual Analyser with Haptic Integration. Proc. of the 10th Symposium on Haptic Interfaces; 2002.

[6] Sun Microsystems, Inc.: SunSHIELD Basic Security Module Guide, Solaris 2.6 System Administrator Collection Volume 1, Mountain View, CA, 1997.

[7] A. Noack: Energy Models for Drawing Clustered Small-World Graphs. Computer Science Report 07/03, BTU Cottbus, Mai 2003.

[8] M. Schulz; S. Schmerl: Visualisierungs-Frontend für dreidimensionale virtuelle Informationsräume. Studienarbeit, BTU Cottbus, 2002.

[9] I. Onut; B. Zhu; A. Ghorbani: A novel visualization technique for network anomaly detection. Proc. of PST 04 Canada, Oct. 2004.