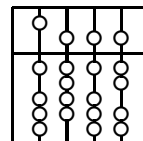


TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK



System zur Anomalieerkennung in IP-Netzwerken: Design und Realisierung

Diplomarbeit

Tobias Sandhaas

Aufgabenstellerin: Prof. Anja Feldmann, Ph.D.

Betreuer: Dipl.-Inf. Robin Sommer

Abgabedatum: 6. November 2003

Ich versichere, dass ich diese Diplomarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 6. November 2003

Tobias Sandhaas

Inhaltsverzeichnis

Inhaltsverzeichnis	i
Abbildungsverzeichnis	iii
Tabellenverzeichnis	iv
1 Einleitung	1
1.1 Gliederung der Arbeit	2
2 Grundlagen	4
2.1 Intrusion Detection	4
2.1.1 Klassifikation von Intrusion Detection Systemen	6
2.1.2 Das Intrusion Detection System BRO	13
2.2 Statistische Verfahren zur Bewertung von Anomaliekenngößen	15
2.2.1 Grafische Analysetechniken	16
2.2.2 Quantitative Analysetechniken	19
2.2.3 Robuste Lage- und Streumaße	29
3 Entwurf eines Anomalie-basierten Intrusion Detection Systems	32
3.1 Anforderungen	32
3.2 Lösungsansatz zur Effektivitätssteigerung	33
3.3 Konzept	35
3.4 Einordnung in die Klassifikation	39
3.5 Komponenten der Anomaliebestimmung	41

Inhaltsverzeichnis

3.5.1	Anomaliekenngrößen	41
3.5.2	Anomaliebewertung	47
3.5.3	Eskalationsbewertung	54
4	Realisierung	61
4.1	Struktureller Aufbau	61
4.2	Implementation	64
4.2.1	Datenlieferant: BRO	64
4.2.2	Datenorganisation	68
4.2.3	Anomaliebestimmung	75
5	Erfahrungen mit dem Prototypen	79
5.1	Anomaliekenngrößen	79
5.2	Anomaliebewertung	81
5.3	Eskalationsbewertung	89
6	Zusammenfassung	92
	Literaturverzeichnis	96

Abbildungsverzeichnis

2.1	Beispiel eines Boxplots	17
2.2	Beispiel eines Quantile-Quantile Plots	18
2.3	Beispiel verschiedener Lagemaße	21
2.4	Beispiel von kumulativen Verteilungsfunktionen	27
3.1	Vorgang der Anomaliebestimmung	35
3.2	Exemplarischer Verlauf der Datentransferrate innerhalb einer Woche	42
3.3	Veranschaulichung des standardized residuals als Boxplot	53
3.4	Prinzip der Eskalationsbewertung	56
3.5	Die Entscheidungsfindung bei der Eskalation	60
4.1	Struktureller Aufbau	63
5.1	Boxplot einer Z-Score Berechnung stark schwankender Datenpunkte	82
5.2	Veränderter Boxplot bei Vergrößerung des Betrachtungszeitraums	83
5.3	Histogramm des übertragenen Datenvolumens des Port 110 auf Tagesbasis	84
5.4	Boxplot der Z-Score Berechnung einer Flash Crowd	84
5.5	Histogramm des übertragenen Datenvolumens des Port 80 auf Tagesbasis	84
5.6	Eine typische Chi-square Verteilung	86
5.7	Exemplarische Liste von Eskalation eines 60-tägigen Zeitraums	90

Tabellenverzeichnis

4.1	Die möglichen Zustände einer Verbindung unter BRO	66
4.2	Verwendung von Zwischenwerten bei den Anomaliekenngrößen	67
4.3	Der Datenbankschlüssel zum Zugriff auf die Anomaliekenngrößen	71
4.4	Die Datenbanktabellen zur Zuweisung der IPs zu den Regeln/Profilen	73

1 Einleitung

Die Kommunikation gewinnt aufgrund zunehmender Globalisierung stetig an Bedeutung. Daher ist die Weiterentwicklung von Kommunikationstechniken und ihre Verbreitung ein Ansporn für Forschung und Wirtschaft. Durch Computersysteme gestützte Kommunikation ist heutzutage in der Praxis nicht mehr wegzudenken und ersetzt zunehmend traditionelle Kommunikationssysteme. Das Internet ist ein Kommunikationssystem, welches in den letzten Jahren stetig an Bedeutung gewann und auch von der Wirtschaft eingesetzt wird. Die einfache und schnelle Transaktion von Barvermögen über „Online Banking“ sei nur exemplarisch genannt. Die Vorteile der Kommunikation über das Internet führen letztendlich zu einem explosionsartigen Anstieg der Anzahl an dem Netz beteiligter Rechner. Durch den zunehmenden Gebrauch des Internets in vielen Bereichen des Lebens steigt natürlich auch der Bedarf an Sicherheit. Der Schutz persönlicher Daten, wie beispielsweise Kennwörter, ist für die Sicherheit des Einzelnen von grundlegender Bedeutung. Tatsächlich steigt die Anzahl von Angriffen auf einzelne Rechner jährlich gravierend an [1], sodass die Überprüfung der Datenflüsse zunehmend wichtiger wird. Die Kontrolle kann jedoch oftmals nur automatisiert erfolgen, da durch den Anstieg der Anzahl beteiligter Rechner die Datenmenge zu groß ist, als dass sie manuell geprüft werden könnte.

Es existieren verschiedene Typen von Intrusion Detection Systemen, die sich mit dieser Problematik beschäftigen. Jedoch benötigen die meisten in der Praxis eingesetzten Systeme sehr spezifisches Wissen über die Angriffe, um sie dann auch lückenlos und fehlerfrei zu erkennen. Diese Information ist im Vorfeld aber oftmals nicht bekannt und kann sich zudem leicht durch geringe Modifikationen des Angriffs ändern. Anomalie-basierte Intrusion Detection Systeme versuchen mittels statistischen Analysen des Datenflusses auf ähnliche Ergebnisse zu schließen, wobei in diesem Fall kein oder nur sehr wenig Hintergrundwissen über die Angriffe bekannt sein muss. Die statistischen Analysen vergleichen hierbei üblicherweise aktuelle Charakteristika von Datenflüssen, wie beispielsweise das derzeitig übertragene Datenvolumen, mit den aus Vergangenheitswerten berechneten zu erwartenden Charakteristika. Anomalie-basierte Intrusion Detection Systeme werden allerdings derzeit nur selten produktiv eingesetzt, da sie oftmals sehr viele Fehlalarme melden und die Nachbearbeitung der Alarme daher sehr zeitaufwendig ist.

Aufgrund dessen ist das Ziel dieser Arbeit, ein Anomalie-basiertes Intrusion Detection Sy-

stem zu entwickeln, welches auch im realen Einsatz praktikabel ist und möglichst nicht nur von Sicherheitsfachleuten sinnvoll eingesetzt werden kann. Um das zu erreichen, muss als primäres Ziel die Anzahl der Fehllarme minimiert werden. Dies ist denkbar schwierig, da in der Regel nur wenig Informationen über die Natur der Datenflüsse innerhalb des Netzwerks vorliegen. Zum einen muss daher mittels robuster Heuristiken versucht werden, den Fehler bei der Berechnung von Haus aus zu minimieren. Zum anderen muss aber zusätzlich eine Möglichkeit geschaffen werden, auf der Menge der gefundenen Alarme sinnvoll selektieren zu können. Axelsson [2] meint hierzu, dass aufgrund der Vernachlässigung der Basisrate (*base-rate-fallacy*) die Effektivität Anomalie-basierter Intrusion Detection Systeme hauptsächlich durch die Möglichkeiten zur Minimierung der Fehllarme bestimmt wird, statt durch die Vollständigkeit der Erkennung von Angriffen. Die Reduktion der Alarme mittels Selektion setzt ein Maß voraus, mit Hilfe dessen die gravierenden Alarme von den unwesentlichen unterschieden werden können. Bei der Selektion soll – je nach individueller Konfiguration – nach der Schwere der Alarme selektiert werden, indem ausgehend von Alarmen, die auf minimalen Abweichungen von dem Erwartungswert beruhen, zunehmend gravierendere Alarme ignoriert werden. Da im Allgemeinen gilt, dass die Alarme, die auf geringeren Abweichungen beruhen, eher Fehllarme darstellen als die gravierenden Alarme, ist dieses Maß ein adäquates Mittel zur Reduktion von Fehllarmen. Der nachträgliche Aufwand bei der Analyse der Alarme minimiert sich zum einen, da durch die Reduktion potentieller Fehllarme die Gesamtzahl gemeldeter Alarme verringert wird. Zum anderen sind die restlichen Alarme üblicherweise leichter zu analysieren, da sie eher einen tatsächlichen Angriff widerspiegeln. Letztendlich werden dem Administrator je nach individueller Konfiguration nur noch bestimmte Alarme präsentiert. Nachteilig ist, dass die Vollständigkeit der Erkennung von Angriffen dabei verringert wird, weil die Möglichkeit besteht, dass auch tatsächlich stattgefundene Alarme gefiltert werden. Durch die individuelle Konfiguration kann der Administrator jedoch einen Kompromiss zwischen dem nachträglichen Aufwand bei der Analyse und der Vollständigkeit der Erkennung finden.

Die Auswahl der robusten Heuristiken und die Selektion auf der Menge der gefundenen Alarme sind daher zwei wichtige Maßnahmen, um dem Ziel ein in der Praxis taugliches Anomalie-basiertes Intrusion Detection Systems zu entwerfen, Rechnung zu tragen.

1.1 Gliederung der Arbeit

Kapitel 2 beschreibt wichtige Grundlagen, die für dieses Thema relevant sind. Zum einen wird eine Taxonomie für Intrusion Detection Systeme näher erläutert und zum anderen statistische Verfahren zur Analyse zweier Datenreihen vorgestellt.

Kapitel 3 stellt den Entwurf eines in dieser Arbeit entwickelten Anomalie-basierten Intrusion Detection Systems vor. Es werden Vor- und Nachteile des Systems beschrieben und wichtige Eigenschaften des Systems exakt spezifiziert.

Kapitel 4 beschäftigt sich mit Einzelheiten der Realisierung eines Prototypen und erwähnt einige resultierende Probleme.

Kapitel 5 beschreibt einige gewonnene Erfahrungen bei der Anwendung des Prototypen und seiner Algorithmen unter realen Bedingungen.

Kapitel 6 liefert eine Zusammenfassung der Diplomarbeit und Ausblick über das weitere Vorgehen zur Realisierung eines voll funktionsfähigen Anomalie-basierten Intrusion Detection Systems.

2 Grundlagen

In diesem Kapitel werden grundlegende Informationen, die zum Verständnis der Arbeit beitragen, angeführt. Zunächst wird ein Überblick über die verschiedenen Arten von Intrusion Detection Systemen vermittelt, um eine exakte Klassifizierung des neu zu entwickelnden Systems zu ermöglichen. Daraufhin wird das von Vern Paxson entwickelte Intrusion Detection System BRO vorgestellt, welches am Lehrstuhl VIII der Technischen Universität München mitentwickelt wird. Zuletzt folgt eine detaillierte Aufstellung von möglichen statistischen Methoden, die in Intrusion Detection Systemen eingesetzt werden können.

2.1 Intrusion Detection

Aufgrund des steigendem Bedarfs an Kommunikation zwischen Menschen entstehen maschinengestützte Kommunikationssysteme wie beispielsweise das Internet. Die Anzahl der an das Internet angeschlossenen Rechnersysteme steigt weltweit seit Jahren explosionsartig an.

Mitunter aufgrund dieser Tendenz steigt auch der Bedarf an Systemen zur Kontrolle dieser Datenflüsse stetig an. Durch sehr einfach zu bedienende Tools, die voll automatisch Programmschwachstellen ausnützen können ohne dass der Angreifer hierbei viel Wissen benötigt, gewinnt die Sicherung persönlicher Informationen zunehmend an Bedeutung.

Es werden Systeme entwickelt, die den Datenfluss beobachten und gegebenenfalls auch unterbinden können, um die eigenen Rechner vor unbefugten Zugriffen zu schützen. In der Praxis sehr häufig eingesetzte Systeme sind unter anderem Firewalls, die nach Vorgabe einer vorher definierten Sicherheitsrichtlinie explizit definierte Datenflüsse blocken oder durchlassen. Andere häufig eingesetzte Systeme sind Intrusion Detection Systeme (IDS), die den Datenfluss beobachten und verdächtige Aktionen erkennen. Intrusion Detection ist definiert als der Prozess der Erkennung böswilliger oder auch ungewollter Aktivitäten innerhalb der elektronischen Datenverarbeitung. Die Definition ungewollter Aktivitäten bestimmt sich hierbei unmittelbar aus der Sicherheitsrichtlinie der jeweiligen Einrichtung. Daher ist es erforderlich im vornherein diese Sicherheitsrichtlinie exakt zu definieren, um dann die Intrusion Detection Systeme auf diese Bedürfnisse möglichst genau und lückenlos einzustellen. Eine unvollständige Liste häufig auftretender Angriffe und dessen Auswirkungen ist:

- Spionage von privaten Daten

Durch Spionage erhalten nicht autorisierte Personen Einblick in private, potentiell sensible Daten Dritter. Dieser Zugriff kann an verschiedenen Stellen des Datenflusses passieren. Üblicherweise wird hierzu entweder der Rechner auf dem die Daten liegen direkt komprimiert oder die Datenverbindung zwischen zwei Rechnern abgehört. Schlecht geschützte, vernetzte Systeme können Tür und Tor für Wirtschaftsspionage öffnen. Durch Verschlüsselung der Daten sowohl im Langzeitspeicher als auch während der Datenübertragung, besteht die Möglichkeit den Schaden zu reduzieren. Beispielsweise wird hierzu oft ein Programm installiert, welches unerkannt und automatisch alle Aktionen der Benutzer des Rechners mitschneidet und sammelt (*Backdoor*). Zusätzlich ermöglichen diese Programme oft auch einen einfachen und unaufmerksamen Zutritt des Angreifers zu späteren Zeitpunkten. Bleibt dieses Programm längere Zeit unerkannt, ist davon ausgehen, dass der Angreifer zusätzlich zu den im Langzeitspeicher gespeicherten Daten des Systems, auch die flüchtigen, beispielsweise durch den Benutzer mittels Tastatur eingetippten Daten, ausspioniert hat.

- Denial of Service

Hierbei wird letztendlich ein bestimmter Dienst eines Rechners oder aber auch der gesamte Rechner angegriffen, sodass er seinen gestellten Aufgaben nicht mehr nachgehen kann. Die Art der Angriffe erfolgt üblicherweise über das Netzwerk und kann zum Teil mit einfachen Mitteln bewerkstelligt werden. Denial of Service (DOS) Angriffe sind eine sehr häufige Angriffsform im Internet, da zur Durchführung wenig Know-How und Ressourcen benötigt werden und zudem nur schwer die Quelle des Angriffs zurückverfolgt werden kann. Die Anonymität der Angreifer wird oft durch mangelndes Sicherheitsbewusstsein des Betreibers des jeweiligen Rechnernetzes unterstützt, indem dem Angreifer ermöglicht wird seine Absenderadresse (Quell-IP) beliebig zu fälschen¹. Filterregeln an den betreffenden Edge-Routern der Teilnetze können hierbei Abhilfe schaffen und eine Rückverfolgung der Angreifer stark vereinfachen.

- Würmer

Würmer sind automatisch agierende Programme, die sich unter Ausnutzung von Sicherheitslöchern in Programmen über das Rechnernetz fortpflanzen (*exploit*). Das auf diese Art befallene System wird von dem Wurm daraufhin verwendet um weitere Systeme zu infizieren [3]. Ist das System befallen, hat der Wurm oftmals Zugriff auf die Daten des Rechners und kann unter Umständen beliebige Aktionen durchführen. Oftmals wird zum Ausschnüffeln von Daten, wie bereits beschrieben, eine Backdoor installiert. Die Datenmenge die bei der Fortpflanzung verwendet wird, kann so klein sein, dass diese im normalen Datenverkehr des Systems untergeht, oder aber auch, wie im Beispiel des „W32/SQLSlammer“ [4] Wurms die gesamte zur Verfügung stehende

¹spoofing

Bandbreite des Systems in Anspruch nehmen. Dies kann dann unmittelbar zu einem Zusammenbruch der Netzinfrastruktur und somit zu einem Denial of Service führen.

- **Missbrauch von bestehenden Diensten eines Rechners**
Oftmals können die vom Administrator installierten Dienste eines Rechners, wenn dieser Dienst nicht umsichtig konfiguriert wurde, hohe unerwartete zusätzliche Kosten verursachen oder sogar den gesamten Betrieb des Rechnernetzes gefährden. Nach eigenen Erfahrungen werden im Internet mehrmals täglich typische Dienste aller angeschlossenen Rechner geprüft, um neue Ziele, die missbraucht werden können, auszumachen. Wird beispielsweise ein anonymer FTP-Server fälschlicherweise mit Schreib- und Leserechten auf ein Verzeichnis konfiguriert, wird dieser höchstwahrscheinlich innerhalb von kurzer Zeit als Austauschpunkt illegal kopierter Software verwendet. Das in diesem Fall transferierte Datenvolumen kann dann für den Betreiber unerwartet hohe Kosten verursachen.
- **Installation eines neuen Dienstes**
Ein von einem Angreifer erfolgreich gehackter Rechner, wird oftmals von diesem missbraucht um eigene Dienste zu starten. Dieser neue Dienst wird oftmals auf einen für ihn untypischen Port gestartet, um den Dienst bestmöglichst zu verschleiern.

Intrusion Detection Systeme überwachen den Datenfluss eines Rechnernetzes und durchsuchen ihn nach Sicherheitsverstößen. Die Systeme erleichtern die Arbeit des Administrators ungemein, da oftmals die Anzahl der Datenflüsse zu groß sind um diese manuell zu kontrollieren. Eine korrekte Bewertung ist jedoch manchmal problematisch, da aufgrund der Vielfältigkeit der Datenflüsse oftmals nicht die wahre Natur des Angriffs festgestellt werden kann. Daher erfordert eine nachträgliche Bewertung der Vorfälle durch den Administrator in diesen Fällen ein umfassendes Wissen. Letztendlich muss oftmals viel Zeit investiert werden, um die Ergebnisse der IDS nachträglich zu verifizieren. Wurde ein Angriff erkannt, kann der Administrator die nötigen Schritte eingeleitet.

Die verschiedenen Arten von Intrusion Detection Systeme und dessen Methoden zur Erkennung von Angriffen werden nun zusammengefasst.

2.1.1 Klassifikation von Intrusion Detection Systemen

Um zu verstehen, wie die unterschiedlichen Intrusion Detection Systeme arbeiten und wie sie sich unterscheiden, werden in diesem Kapitel verschiedene Eigenschaften dieser Systeme untersucht. Eine Klassifizierung ist zudem auch nötig, um ein neu zu entwickelndes Intrusion Detection System in die der bestehenden Systeme einordnen zu können und vor allem die Ziele des neuen Systems klar definieren und differenzieren zu können. Die Klassifikation gibt zudem einen Hinweis auf die zu erwarteten Stärken und Schwächen des Systems.

Leider ist die Terminologie in diesem Bereich nicht eindeutig [5]. Daher werden in dieser Arbeit die durch [6, 7, 5] vorgegebenen Begriffe verwendet. Die Klassifizierung der Intrusion Detection Systeme ist an [6, 7] angelehnt. Im Folgenden werden wichtige Eigenschaften von Intrusion Detection Systemen vorgestellt:

Datenursprung

Die Funktionalität der Systeme richtet sich essentiell an die Datensätze, die den Systemen zur Auswertung zur Verfügung stehen. Hierbei wird zwischen zwei Grundtypen unterschieden:

- Host-basierte Intrusion Detection Systeme (HIDS)

Diese Systeme arbeiten mit Datensätzen, die direkt auf dem jeweiligen Rechner gesammelt wurden. Beispielsweise können durch Modifikationen des Betriebssystems mitunter sehr detaillierte Informationen über den Datenfluss, die Aktionen eines Anwenders oder auch den Zustand der Prozesse gesammelt werden. Zudem besteht oft die Möglichkeit die Protokolldateien, die von der Applikation direkt geliefert werden, mit in die Datenauswertung aufzunehmen. Jedoch ist oftmals eine ausführliche Informationsgewinnung durch die Anwendung nicht möglich oder die Menge an Informationen ist sehr groß und erschwert daher den Einsatz in der Praxis. Bei zu extensiver Informationsgewinnung besteht die Gefahr, dass die Speicherung und Analyse der riesigen Datenmenge schon zu einem Denial of Service führt. Andererseits sind häufig genaue Informationen für die Bewertung von Angriffen unerlässlich. Viele moderne IDS korrelieren die Protokolldateien mehrerer Rechner miteinander um eine bessere Aussage über den Angriff treffen zu können.

Wichtige Vorteile *Host-basierter Intrusion Detection Systeme* sind:

- Die Nähe zum Ort des Geschehens erleichtert wesentlich eine detaillierte Informationsbeschaffung.
- Sehr detaillierte Informationen ermöglichen oftmals bessere Aussagen.
- Unzulässige Programme innerhalb des Systems zum Beispiel Würmer oder Trojaner können frühzeitig festgestellt werden.

Wesentliche Nachteile *Host-basierter Intrusion Detection Systeme* sind:

- Die Analyse der Informationen in den Protokolldateien ist aufgrund der komplexen Abläufe in Betriebssystemen und Applikationen oftmals sehr aufwendig.
- Eine lückenlose Überwachung des Rechnersystems kann aus ähnlichen Gründen nicht gewährleistet werden.
- Durch die Datensammlung und Bearbeitung werden die Ressourcen des Rechnersystems (CPU, Speicher, ...) stark beansprucht und kann das Laufzeitverhalten negativ beeinflussen.

- Die Daten sind zwar detailliert aber in der Regel auf das eigene Rechnersystem begrenzt.
 - Oftmals liefern Betriebssysteme und Applikationen keine detaillierte Ausgaben, sodass eine Auswertung dann nicht möglich ist.
 - Netzwerkangriffe, beispielsweise „TCP hijacking“, können gar nicht oder nur schlecht festgestellt werden, da benötigte Informationen häufig nicht vorliegen.
- Netzwerk-basierte Intrusion Detection Systeme (NIDS)

Netzwerk-basierten Intrusion Detection Systeme haben in den letzten Jahren unter anderem aufgrund zunehmender Vernetzung an Bedeutung gewonnen. Diese Systeme werden in der Regel an einem zentralen Punkt des Rechnernetzes platziert. Üblicherweise liegt dieser Knotenpunkt am Rand eines zu schützenden Rechnernetzes, das normalerweise mehrere Rechner umfasst. Da die Kommunikation passiv mitgeschnitten wird, kann sie dort unauffällig kontrolliert werden, ohne dass der Angreifer es merkt oder die zu schützenden Rechner dadurch beeinflusst werden. Die auf diese Weise gesammelten Daten umfassen dementsprechend nur den Datenfluss von und zu den zu überwachenden Rechnern. Detaillierte Informationen, beispielsweise über den Prozesszustand, liegen im Gegensatz zu den *Host-basierten Intrusion Detection Systemen* in der Regel nicht vor. Aufgrund dieses Sachverhalts ist es oftmals schwer die Datenflüsse lückenlos und fehlerfrei zu bewerten.

Wichtige Vorteile *Netzwerk-basierter Intrusion Detection Systeme* sind:

- Ein NIDS kann mehrere Rechner des Rechnernetzes überwachen ohne das Modifikationen an den einzelnen Rechnern notwendig sind.
- Auf den zu untersuchenden Rechnern entsteht keine zusätzliche Last.
- Eine Korrelation von Angriffen auf mehrere Rechner ist möglich.
- Es besteht die Möglichkeit einen unerwünschten Datenfluss, noch bevor dieser Schaden anrichten kann, aktiv zu unterbinden.
- Da NIDS für den Angreifer nicht erkennbar sind, ist ein unauffälliges Umgehen erschwert.

Wichtige Nachteile *Netzwerk-basierter Intrusion Detection Systeme* sind:

- Die Datensätze beschränken sich auf die aus dem Netzwerkverkehr extrahierten Informationen und geben im allgemeinen keinen Aufschluss über den genauen Zustand der Prozesse der Rechner wieder.
- Aus ähnlichen Gründen können Angriffe, die innerhalb des überwachten Rechnernetzes stattgefunden haben, nicht erkannt werden.
- Aufgrund lückenhafter und mangelhafter Informationen ist eine Bewertung der Datenflüsse schwierig.

- Durch einen gezielten Angriff auf das IDS kann die Funktionalität oftmals so stark eingeschränkt werden, dass Angriffe unbemerkt am System vorbeigeschleust werden können.
- Die Hardwareanforderungen an das NIDS steigen mit der Menge des zu untersuchenden Datenverkehrs. Zudem spielt die Anzahl und Komplexität der Analysen eine entscheidende Rolle.
- Aufgrund zunehmender Verschlüsselung des Datentransfers ist eine umfassende Analyse eventuell nicht möglich.

Die in den letzten Jahrzehnten entwickelten Intrusion Detection Systeme lassen sich in der Regel gut nach diesem Schema kategorisieren. Die Stärken beider Typen sind in Hybriden Intrusion Detection Systemen, wie beispielsweise DIDS [8], vereint. Dieser Ansatz wird jedoch in der Praxis selten verfolgt.

Strategien zur Erkennung von Angriffen

Die Ansätze um Angriffe feststellen zu können, sind grob in die drei Hauptstrategien nämlich der „Missbrauch-basierten Erkennung“, der „Anomalie-basierten Erkennung“ und der „Spezifikations-basierten Erkennung“ untergliedert [9]. Zunächst werden diese Strategien vorgestellt und später auf die Methoden zur Analyse eingegangen. Weitere Informationen sind unter anderem in [10, 11, 12, 2, 5, 13, 14, 15] zu finden.

- **Missbrauch-basierte Erkennung (*misuse detection*)**

Intrusion Detection Systeme die nach der Strategie der Missbrauch-basierten Erkennung arbeiten, benötigen extensive Information über spezifische Angriffe und Schwachstellen von Rechnersystemen. Mit Hilfe dieser spezifischen und teilweise auch eindeutigen Signaturen wird der Datenfluss nach deren Auftreten untersucht. Sobald eine Signatur im Datenfluss erkannt wird, wird ein Alarm veranlasst. Alle anderen Datenflüsse werden explizit als erlaubt deklariert und passieren das IDS. Theoretisch ist die Fehlerfreiheit bei der Erkennung sehr hoch und minimiert aufgrund dieser Strategie in der Regel den Aufwand an die Administration erheblich. Beitragen tut hierbei zudem, dass der Typ des Angriffs aufgrund der Signatur bekannt ist und oft der Kontext eines Alarms existiert. Eine korrekte Analyse der Vorfälle ist dennoch meistens sehr zeitaufwendig und benötigt viel Know-How.

Zweifelhaft ist jedoch die Vollständigkeit der Erkennung von Angriffen. Denn dem System mangelt es prinzipiell an der Möglichkeit unbekannte Angriffe zu erkennen. Es ist daher direkt von der Aktualität der Informationen über spezifische Angriffe und Schwachstellen der Rechnersysteme abhängig. Die Pflege dieser Daten ist sehr zeitaufwendig und zudem sind diese Informationen stark abhängig von den überwachten

Systemen und Applikationen. Da in der Regel nur begrenzt viel Rechenzeit für die Analyse zur Verfügung steht, muss man sich häufig bei der Menge der zu überprüfenden Signaturen auf die unbedingt erforderlichen beschränken. Dabei besteht die Gefahr einige Angriffe nicht zu bemerken.

- **Anomalie-basierte Erkennung (*anomaly detection*)**

Anomalie-basierte Intrusion Detection Systeme (ADS) vergleichen aktuelle Charakteristika der Datenflüsse mit den zu erwartenden. Daraus berechnet sich ein Maß für den Unterschied der jeweiligen Eigenschaften. Dieses Maß ist ein direktes Indiz für die Ungewöhnlichkeit des Datenflusses und wird in der Regel unmittelbar für die Generierung eines Alarms verwendet. Die zu erwartenden Werte der Charakteristika werden bei diesen Intrusion Detection Systemen mit Hilfe von Algorithmen, die in Kapitel 2.2 noch genannt werden, automatisch ermittelt. Dazu werden bereits aufgetretene Datenflüsse als Datengrundlage verwendet. Da diese Systeme das Normverhalten aus realen Datenflüssen herleiten und diese Ströme üblicherweise nicht das ganze Spektrum der legitimen Datenflüsse repräsentieren, leiden diese Systeme prinzipiell unter vielen Fehlalarmen (*False-Positives*). Diese Anzahl kann in Abhängigkeit des Systems und der Situation mehrere tausend Alarme am Tag mit einer Fehlerwahrscheinlichkeit von ungefähr 99% überschreiten [16]. Die nachträgliche Analyse der Alarme durch den Administrator ist dadurch sehr zeitaufwendig und fehleranfällig und minimiert den Nutzen des Systems erheblich. Da der Kontext, der zu einem Alarm führt oftmals nicht mehr zur Verfügung steht ist eine korrekte Analyse erschwert [16, 7]. Da diese Systeme die erwarteten Charakteristika aus den real auftretenden Daten herleiten, besteht prinzipiell die Gefahr, dass ein potentieller Angreifer diese Eigenschaft ausnützt, um das System über längere Zeit hinweg zu manipulieren bis letztendlich die eigentliche Attacke vom System als legitim eingestuft wird. Im Gegensatz dazu haben die ADS aufgrund der empirisch ermittelten Charakteristika aber einen besseren Anspruch auf Vollständigkeit bei der Erkennung von Angriffen als anderen Strategien. Die Struktur dieser Systeme ermöglicht es zumindest prinzipiell bis dato unbekannte Angriffe festzustellen, solange diese nicht in dem als legitim eingestuften Verhalten untergehen. Zudem sind die ADS im Gegensatz zu den Missbrauch-basierten Systemen nicht ganz so spezifisch von den überwachten Betriebssystemen und Applikationen abhängig und benötigen in dieser Hinsicht weniger Pflege.

- **Spezifikations-basierte Erkennung (*specification based detection*)**

Spezifikations-basierte Intrusion Detection Systeme stellen eine Mischform zwischen den Missbrauch-basierten und Anomalie-basierten Systemen dar, mit dem Ziel die Stärken beider Systeme zu vereinen. Ähnlich wie bei Anomalie-basierten IDS wird die Diskrepanz zwischen den aktuellen und den erwarteten Charakteristika der Datenflüsse

bestimmt und dann je nach Größe der Abweichung ein Alarm ausgerufen. Im Unterschied zu den ADS werden hierbei die erwarteten Charakteristika und damit die erlaubten Handlungen nicht automatisch berechnet, sondern ähnlich wie bei den Missbrauch-basierten Systemen manuell festgelegt. Leider ist das Spezifizieren von legitimen Charakteristika, die das Spektrum aller möglichen Angriffe abdecken, üblicherweise kompliziert und individuell von der Sicherheitsrichtlinie abhängig. Jedoch kann aufgrund der exakten Spezifikation der legitimen Charakteristika ein Verstoß eindeutig erkannt werden und erhöht somit die Korrektheit der Alarme auf ein ähnlich hohes Niveau wie bei den Missbrauch-basierten Intrusion Detection Systemen. Die Vollständigkeit der Erkennung von Angriffen liegt prinzipiell zwischen dem der anderen beiden Strategien und ist, wie erwähnt, direkt von der Sicherheitsrichtlinie und der daraus erfolgten Spezifikation erlaubter Handlungen abhängig.

Diese drei Strategien werden durch verschiedene Ansätze verfolgt. Im Folgenden werden die Wichtigsten kurz vorstellen:

- **Signatur Analyse**

Die Signatur Analyse wird hauptsächlich von Missbrauch-basierten Systemen eingesetzt. Bei dieser Methode benutzt das System Signaturen (Regeln), die im Vorfeld erzeugt werden, indem aus einer bekannten semantischen Beschreibung eines Angriffs spezifische Eigenschaften herausgezogen werden, die direkt in den zu untersuchenden Datensätzen zu finden sind. Solche statische Signaturen beschreiben die Muster, die dann in der Mustererkennung durch das IDS, in den Daten gesucht werden. Eine solche Signatur kann beispielsweise konkreten Programmcode zur Ausnutzung eines Sicherheitslochs beinhalten, wie zum Beispiel einen Wurm. Bekannte Vertreter der Signatur-basierten IDS Systeme, die diese Technik einsetzen können, sind SNORT [17] und BRO [18].

- **Expertensysteme**

In Expertensystemen werden verschiedene Tatsachen aus den zu untersuchenden Datenflüssen extrahiert und mit im Vorfeld erzeugten logischen Regeln verknüpft. Durch die Verknüpfung wird ein Kentniss erlangt, die ein Mensch mit dem gleichen Informationen auch gewonnen hätte. Die beim Erkenntnisprozeß benötigten Regeln sind bei der Missbrauch-, Spezifikations- und Anomalie-basierten Erkennung vom Administrator üblicherweise fest vorgegeben. Bei der Anomalie-basierten Erkennung gibt es jedoch Systeme, die diese Regeln während Trainingsprozeduren automatisch bestimmen können [19]. Werden zur Analyse einfache Regeln und Logiken verwendet, werden diese Systeme oftmals auch **Regel-basierten Systemen** genannt. Zur Verknüpfung extrahierter Tatsachen existieren zudem weitere Methoden, die beispielsweise auf grafischen Auswertungen (GrIDS [20]), Petrinetzen (IDIOT [21]) oder Zustandsübergangs-

Analysen (USTAT [22]) beruhen. Ein häufig eingesetztes Expertensystem ist P-BEST [23], welches im Zuge der Entwicklung des IDS Emerald [24] entworfen wurde.

- **Statistische Methoden**

Statistische Methoden werden oftmals in Anomalie-basierten Intrusion Detection Systemen eingesetzt, um Charakteristika des Vergangenenverhalten eines Rechnersystems zu extrahieren. Die daraus gewonnenen Informationen werden in Profilen abgelegt. Im Unterschied zu den Regeln der Expertensysteme, passen sich die Profile dem jeweiligen Normalverhalten über den Betrachtungszeitraum an. Die Informationen in den Profilen werden dann mittels statistischer Methoden mit den aktuellen Charakteristika des Datenflusses verglichen und ein Maß der Abnormalität berechnet. Je anomaler der Vergleich ist, umso wahrscheinlicher ist ein Angriff. Statistische Methoden werden oftmals eingesetzt, wenn wenige Hintergrundinformationen über die Datensätze zur Verfügung stehen und die Natur der Daten unbekannt ist. Aufgrund dessen kann die Sicherheit der gewonnenen Ergebnisse nicht garantiert werden und führt dazu, dass viele auf statistische Methoden basierte IDS unter einer großen Anzahl an Fehlalarmen leiden. Bekannte Intrusion Detection Systeme, die unter anderem statistische Methoden verwenden, sind NIDES [25] und Emerald [24]. Ein einfaches Beispiel für die Verwendung von statistischen Methoden ist beispielsweise der Vergleich der Anzahl erfolgloser Passworteingaben mit einem Schwellwert. Wird dieser Vergangenheitswert überschritten, wird ein Alarm ausgerufen.

- **Andere Ansätze**

Es existieren viele weitere Ideen und Ansätze um Angriffe festzustellen. Zu diesen gehören unter anderem Data Mining (MADAM ID [26]), Neurale Netze (NNID [27, 28]) und Immunologie [29, 30]. Diese sind für diese Arbeit nicht relevant.

Die derzeitige Generation kommerzieller Intrusion Detection Systeme sind hauptsächlich Netzwerk-basiert und verwenden meistens die Strategie der Missbrauch-basierten Erkennung [6]. In der Forschung werden alle drei Ansätze verfolgt, wobei Spezifikations-basierte IDS derzeit noch selten zu finden ist. Manchmal werden innerhalb eines IDS parallel mehrere Strategien und Ansätze verfolgt, um Ergebnisse korrelieren zu können und dadurch bessere Aussagen zu fällen. Emerald beispielsweise kann sowohl Missbrauch als auch Anomalie Erkennung durchführen und ermöglicht mit Hilfe eines ausgeklügelten Expertensystems die Verknüpfung der Ergebnisse der statistischen Methoden und der Signatur Analyse.

Reaktionen des Systems

Ist ein Angriff festgestellt worden, können Intrusion Detection Systeme verschieden reagieren. Passive Systeme weisen den Administrator ausschließlich auf einen Verstoß hin, wo hingegen aktive Systeme automatisch auf Angriffe reagieren. Bei den aktiven Systemen wird einerseits

zwischen den Systemen, die sich aktiv wehren und den ursprünglichen Angreifer selber angreifen und andererseits den Systemen, die lediglich den eigenen betroffenen Service schützen, unterschieden. Der Schutz des eigenen Services könnte beispielsweise durch das Deaktivieren des Services oder aber auch durch das Sperren der Netzwerkverbindungen zu diesem Service erfolgen. Es besteht jedoch die Gefahr, dass Angreifer durch geschickte Aktionsfolgen indirekt mit Hilfe des IDS den Betrieb der Dienste stören und dadurch die legitime Nutzung behindert.

Entdeckungszeit und Granularität der Datenverarbeitung

Intrusion Detection Systeme unterscheiden sich prinzipiell auch in der Entdeckungszeit. Es wird hierbei zwischen den Systemen, die die Angriffe in nahezu Echtzeit erkennen und den Nicht-Echtzeitsystemen, bei denen die Auswertungen mit zeitlichen Verzögerungen gemeldet werden, unterschieden. Prinzipiell stellt ein Echtzeitsystem daher erhöhte Anforderungen an die Ausführungseffizienz.

Zudem können die Systeme nach der Art der Datenverarbeitung unterschieden werden. Die Systeme können die Datensätze entweder kontinuierlich verarbeiten oder die Daten in Schübe (*batches*) einteilen, die dann der Reihe nach abgearbeitet werden.

Positionierung des IDS

Die Funktionalität eines Intrusion Detection Systems kann prinzipiell in zwei Komponenten untergliedert werden. Zunächst werden die Daten gesammelt und dann im zweiten Schritt verarbeitet. Bei Host-basierten Systemen kann die Datenerfassung aus einer Quelle herrühren oder auch mittels mehreren Prozessen von verschiedenen Rechnern zeitgleich. Ein Netzwerk-basiertes System untersucht den Datenfluss eines Rechnernetzes, wobei es auch hier sinnvoll sein kann aufgrund der Topologie der Netze mehrere Systeme zeitgleich einzusetzen. Die Position dieser Meßpunkte ist hierbei entscheidend.

Ebenso kann die Datenverarbeitung zentral und somit nur auf einem Rechner getätigt oder auf mehrere Rechner verteilt werden. Durch die Aufspaltung der Aufgaben in Prozesse und ihrer Verteilung im Rechnernetz wächst die Komplexität der Systeme stark an. Dies ist mitunter ein Grund, weshalb viele bestehende Systeme keine Verteilung ermöglichen.

2.1.2 Das Intrusion Detection System BRO

BRO ist ein von Vern Paxson entwickeltes Netzwerk-basiertes Intrusion Detection System und wird zudem am Lehrstuhl VIII der Technischen Universität München mitentwickelt und dort auch eingesetzt. Im Gegensatz zu vielen anderen IDS arbeitet dieses System mit Ereignissen (*Events*), die in Echtzeit aus den Datenflüssen extrahiert werden. Solch ein Ereignis ist

beispielsweise ein FTP-Login oder HTTP-Request. Diese Ereignisse werden dann mittels vorgegebenen Regeln korreliert und daraus Aussagen geformt. Eine Besonderheit des Systems ist die von Paxson entwickelte Sprache, die eine flexible und fehlertolerante Definition der Regeln ermöglicht. Um diese Eigenschaften möglichst optimal zur Geltung zu bringen, ist eines der Hauptziele des Systems die Trennung der eigentlichen Regel von dessen zugrunde liegenden Ereignisgewinnung.

BRO lässt sich nicht eindeutig einer der oben beschriebenen Strategien zur Erkennung von Angriffen zuweisen. Genauer betrachtet wird durch dieses System sowohl die Missbrauch-basierte Erkennung, als auch die Spezifikations-basierte Erkennung ermöglicht. Die Missbrauch-basierte Erkennung wird durch die einfache Mustererkennung von Signaturen ermöglicht. Spezifikations-basiert ist das System deshalb, da durch die mächtige Sprache auch komplizierte, legitime Handlungen abgebildet werden und Abweichungen von diesen dann erkannt werden können. Die Frage, ob das System auch eine Anomalie-basierte Erkennung ermöglicht ist strittig. Meiner Meinung nach ist dies derzeit aus folgenden Gründen nicht der Fall: Nach der in Kapitel 2.1.1 auf Seite 6 beschriebenen Taxonomie werden bei dem Anomalie-basierten Ansatz die aktuellen Charakteristika eines Datenflusses mit den zu erwartenden verglichen. Hierbei ist es essentiell, dass die zu erwartenden Charakteristika von dem System selber hergeleitet werden. Dies ist meiner Meinung nach jedoch nur möglich, wenn die erwartenden Charakteristika im Langzeitspeicher abgelegt werden können. Denn nur durch die Persistenz dieser Daten ist eine sinnvolle Auswertung auch über größere Zeiträume möglich. BRO ermöglicht dies derzeit noch nicht, allerdings wird seit kurzem daran gearbeitet. Zudem ist eines der Probleme mit denen derzeit gekämpft wird die Laufzeitoptimierung. Durch den Ereignis-basierten Ansatz und dessen ohnehin aufwendige Bearbeitung ist die Durchführung von größeren Berechnungen innerhalb BRO daher derzeit nicht sinnvoll.

Bei der Analyse verwendet BRO zwei Techniken. Einerseits ermöglicht das System eine Signatur Analyse um feste Bytefolgen innerhalb des Datenflusses zu erkennen. Andererseits verwendet BRO aber auch das bereits erwähnte Regel-basierte System, um auf Basis beliebiger Ereignisse eine Auswertung durchzuführen.

Das System arbeitet kontinuierlich die zu untersuchenden Daten ab und meldet in Echtzeit falls es zu Unstimmigkeiten kommt. Hierbei beschränkt sich das System auf die passive Meldung dieser und führt keine aktiven Gegenmaßnahmen durch.

Genauere Informationen sind unter [18, 31] zu finden.

2.2 Statistische Verfahren zur Bewertung von Anomaliekenngrößen

Wie bereits beschrieben, werden bei der Anomalie-basierten Erkennung aktuelle Charakteristika mit den zu erwartenden Charakteristika der Datenflüsse des Netzwerks verglichen. Die Charakteristika beschreiben Eigenschaften, wie die Verbindungsdauer, Datentransfervolumen und Anzahl an Verbindungen, und werden durch die in dieser Arbeit genannten Anomaliekenngrößen (vgl. Kapitel 3.5.1 auf Seite 41) abgebildet. Da diese Werte statistischer Natur sind, werden diese im Folgenden durch Zufallsvariablen modelliert, das heißt die einzelnen Werte bezüglich eines Charakteristikums (z.B. Verbindungsdauer) als Realisierung einer Stichprobenvariable angesehen und durch X_1, X_2, \dots, X_n notiert. Im weiteren sei durch x_1, x_2, \dots, x_n eine Stichprobe von n Realisierungen der Stichprobenvariablen X_1, X_2, \dots, X_n bezeichnet. Im Folgenden werden statistische Verfahren beschrieben, die aus Stichproben oder hypothetisierten Zufallsvariablen statistische Parameter, wie Lage und Streumaße, ermittelt. Die statistischen Parameter können mitunter durch vorgestellte Tests verglichen werden. Für konkrete Problemstellungen gibt es oftmals mehrere Statistiken, die teilweise auch gleiche Annahmen an die Zufallsvariable stellen.

Sobald statistischen Verfahren zur Erkenntniserweiterung beitragen, aber die Sicherheit der gewonnenen Erkenntnisse nicht begründet werden kann, werden sie *Heuristiken* genannt [32].

Im Allgemeinen gilt, dass ein Verfahren, das viele Annahmen stellt, bessere Ergebnisse liefert als ein allgemein gültigerer, da ersterer in der Regel eine spezifischere Bewertung ermöglicht. Dies setzt jedoch voraus, dass diese Annahmen auch erfüllt sind. Ein Verfahren wird als *nicht-parametrisch* bezeichnet, wenn dieser keine konkreten Parameter, wie zum Beispiel Mittelwert und Varianz, zur Beschreibung der zugrunde liegenden Verteilung voraussetzt (vice versa). Somit sind diese Verfahren üblicherweise nicht so mächtig wie ihre parametrischen Gegenüber. Eine zweifelsfreie Befriedigung der Annahmen ist aber manchmal nur schwer sicherzustellen. So besagt der *Zentrale Grenzwertsatz*, der oft als Begründung für die Normalverteilung von Zufallsvariablen verwendet wird, dass die durch Summenbildung einer genügend großen Anzahl an identisch verteilten und unabhängigen Zufallsvariablen gebildete Zufallsvariable gegen eine Normalverteilung konvergiert. Dieser Satz gibt aber keinen Aufschluss darüber wie groß diese Anzahl wirklich sein muss. Es ist ersichtlich, dass die Anzahl bei sehr schiefen und stark gewölbten Verteilungen größer ist als bei von Haus aus schon näherungsweise normalverteilten Zufallsvariablen. Die Theorie der robusten Statistik, beschrieben in Kapitel 2.2.3, beschäftigt sich mit Verfahren, die fehlertolerant auf Verstöße gegen ihre Annahmen reagieren.

Aufgrund der Vielfältigkeit der statistischen Methoden und noch offener Fragestellungen, zum Beispiel bei den ab Seite 25 beschriebenen Anpassungstests (*goodness-of-fit*) [33], ist die Wahl eines korrekten Tests oftmals schwer. Zudem sind einige Tests auf spezielle Fälle zugeschnitten und liefern nur für diese gute Ergebnisse. Sind die Charakteristika einer Verteilung im vornherein unbekannt oder schwanken die Stichprobenwerte stark, ist die Auswahl eines optimalen Anpassungstests oftmals nicht eindeutig möglich [33].

Statistische Verfahren können grob in zwei Analysetechniken unterteilt werden. Zum einen können sie auf rein grafische Verfahren beruhen und ermöglichen somit eine an visuellen Eigenschaften angelehnte Auswertung. Oder sie beruhen auf mathematischen Formeln mit Hilfe deren die quantitative Eigenschaft einer Stichprobe überprüft werden können. Oftmals kann es sinnvoll sein, beide Ansätze parallel zu verfolgen, da sie sich gut ergänzen und somit eine detaillierte und vor allem weniger fehleranfällige Auswertung ermöglicht wird. Um eine Auswahl passender Algorithmen für Intrusion Detection Systemen zu erleichtern, wird zunächst ein grober Überblick über die statistischen Methoden in Abhängigkeit dieser Analysetechniken geliefert. Wenn nicht anders erwähnt, sind die jeweiligen Zufallsvariablen univariat (eindimensional), unabhängig und identisch normalverteilt.

2.2.1 Grafische Analysetechniken

Diese Verfahren werden oftmals vom Statistiker im Vorfeld einer Analyse verwendet, um ein Gefühl für das „Aussehen“ der jeweiligen Zufallsvariable zu bekommen und sind oftmals essentiell um den richtigen Algorithmus für weitergehende Analysen bestimmen zu können. Zudem hilft eine Visualisierung oftmals um das jeweilige Ergebnis eines Algorithmus bewerten zu können [34]. Gerade für die in der Anomalieerkennung eingesetzten Heuristiken können daher die grafische Analyse ein wichtiger Teil zur Bewertung der Ergebnisse darstellen. Eine Auswertung rein grafischer Aspekte ist in vielen Fällen aber nur schwer automatisiert durchführbar (siehe Quantile-Quantile Plots auf Seite 18). Weitere Informationen sind unter anderem in [35, 34, 36] zu finden. Es folgt eine Auswahl grafischer Analysetechniken, die sich im Bereich der Intrusion Detection anbieten:

Blob plots ist eine einfache grafische Methode. Bei diesem wird lediglich eine horizontale Skala gezeichnet und die jeweiligen univariaten Stichprobe an dieser Achse entlang aufgetragen. „Blob plots“ liefern auf einfache Art und Weise einen schnellen Überblick über die Verteilung der Werte relativ zueinander. Dadurch dass jeder Stichprobenwert auf der Achse aufgetragen wird, eignet sich dieses Verfahren nicht für eine große Anzahl von Stichprobenwerten.

Histogramme sind zweidimensionale Grafiken, bei dem für disjunkte Intervalle auf der X-Achse jeweils die Häufigkeit der Stichprobenwerte in den jeweiligen Intervallen auf der Y-Achse aufgetragen wird. Sie eignen sich gut zur Veranschaulichung der Verteilung

der aufgetragenen Zufallsvariable. Folgende Eigenschaften der Stichprobe sind in der Regel erkennbar:

- Das Zentrum (*center*)
- Die Streuung (*spread*)
- Die Schiefe (*skewness*)
- Die Wölbung (*kurtosis*)
- Die Ausreißer (*outliers*)
- Die Häufungspunkte

Bei diskreten Zufallsvariablen werden brauchbare Histogramme erzeugt, indem der Wertebereich in Intervalle unterteilt wird und die Häufigkeiten innerhalb der Intervalle aufgetragen werden. Bei dieser Methode wird die Menge an Stichprobenwerten erheblich minimiert, weshalb Histogramme auch sehr gut zur Visualisierung größerer Stichproben verwendet werden können.

Boxplots bestimmen zunächst den Median der Stichprobe und verwenden diesen dann als Zentrum einer Box, dessen Größe sich nach dem unteren und oberen *Quantil* bestimmt. Das untere Quartil ist laut Definition das 0.25-*Quantil* und das obere Quartil das 0.75-Quantil. Ein p -Quantil ist der Wert, die eine der Größe nach geordnete Zahlenreihe auf die Art teilt, sodass $p * 100$ Prozent der Stichprobenwerte kleiner sind. Quantile teilen also grafisch betrachtet eine Dichtefunktion so, dass die Fläche zu seiner Linken gleich dem vorgegebenen Prozentsatz der Gesamtfläche darstellt. Der Abstand des unteren zum oberen Quartil wird auch als Interquartilabstand bezeichnet. Der Abstand des un-

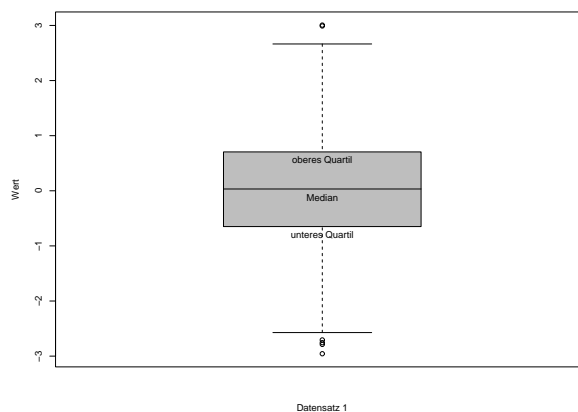


Abbildung 2.1: Beispiel eines Boxplots

teren zum oberen Quartil wird auch als Interquartilabstand bezeichnet und wird in der Abbildung durch die Höhe der Box veranschaulicht. Abbildung 2.1 stellt einen solchen Boxplot dar.

Boxplots stellen auf sehr effiziente Art das Lage- und Streumaß einer oder mehrerer Zufallsvariablen dar und eignen sich daher sehr gut, um Unterschiede zwischen ihnen festzustellen. Stichproben, die aus vielen Stichprobenwerten bestehen, können sehr gut abgebildet werden und werden durch Boxplots anschaulich. Oftmals wird dieses Verfahren auch verwendet um *Ausreißer* innerhalb einer Stichprobe zu finden. Hierzu werden in der Regel alle Stichprobenwerte, die jenseits des 1.5 fachen Interquartilabstands außerhalb der Box liegen, als Ausreißer bewertet. Zusammengefasst werden folgende Metriken veranschaulicht:

- Der Median der Stichprobe
- Das untere und obere Quartil (Begrenzung der Box)
- Der Interquartilabstand (Höhe der Box)
- Die Ausreißer und ihre Lage (weit außerhalb der Box)
- Der Mittelwert (optional)

Wie in Abbildung 3.3 auf Seite 53 dargestellt, wird in dieser Arbeit eine alternative Darstellung der Boxplots verwendet. Hierbei wird zur Erkennung von Ausreißern nicht eine einzige Schranke verwendet, sondern unter Verwendung mehrerer Schranken Ausreißer sensibler bewertet.

Quantile-Quantile Plots prüfen, ob zwei Stichproben von einer gemeinsamen Verteilung stammen. Hierzu werden möglichst viele Quantil-Paare beider Stichproben gebildet und diese Paare dann durch Koordinaten in einer Grafik ausgedrückt. Auf der X-Achse wird der Wert des Paares der ersten Stichprobe und auf der Y-Achse der der zweiten Stichprobe verwendet. Basieren die zwei Stichproben auf einer gemeinsamen Verteilung dann sollten diese Punkte mehr oder weniger alle auf einer 45 Grad Linie liegen. Abbildung 2.2 veranschaulicht diesen Sachverhalt. Der Quantile-Quantile Plot

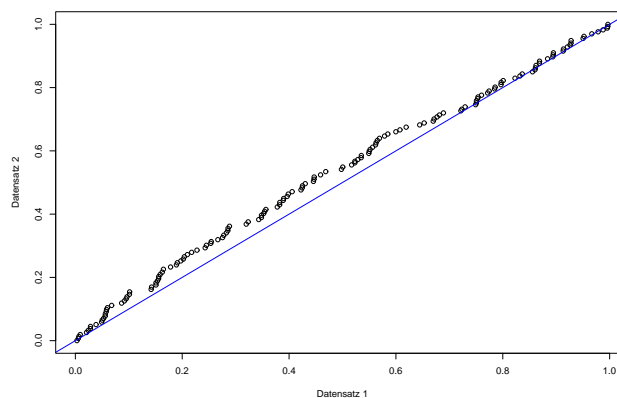


Abbildung 2.2: Beispiel eines Quantile-Quantile Plots

benötigt für diese Analyse keine gleich großen Stichproben; gegebenenfalls werden fehlende Stichprobenwerte durch die nächstliegenden interpoliert. Durch die Natur dieses Verfahrens werden gleichzeitig mehrere Aussagen überprüft:

- Basieren die Stichproben auf einer gemeinsamen Verteilung?
- Haben sie ein ähnliches Lagemaß und Ausmaß (*scale*)?
- Haben sie eine gleiche Form (*shape*)?
- Haben sie ein gleiches Verhalten an den Ausläufen?

Quantile-Quantile Plots ermöglichen teilweise ein besseres Verständnis der Unterschiede zwischen Stichproben als andere Analysetechniken.

2.2.2 Quantitative Analysetechniken

Die quantitativen Analysetechniken versuchen Schlüsse anhand der rechnerischen Auswertung der zu untersuchenden Stichproben zu ziehen und können grob in zwei Kategorien eingeteilt werden:

In der **Intervallabschätzung** wird zunächst, basierend auf der erhobenen Stichprobe, eine bis dato unbekannter statistischer Parameter, wie beispielsweise der Stichprobenmittelwert oder die Stichprobenvarianz, bestimmt. Dieser statistische Parameter stellt eine Annäherung an die echten Zufallsvariable dar, die sich ergeben würde, wenn der gesamte Stichprobenraum bekannt wäre. Da dieses Wissen üblicherweise nicht vorliegt oder praktisch zu umständlich ist, arbeitet die Statistik mit dieser Annäherung. Der beobachtete statistische Parameter (z.B. Stichprobenmittelwert) ist für jede Stichprobe unterschiedlich und führt zu einer Unsicherheit, die durch ein Konfidenzintervall beschrieben wird. Dieses Konfidenzintervall wird durch die Wahrscheinlichkeit α charakterisiert mit der der realisierte statistische Parameter innerhalb des Intervalls liegt. Große Konfidenzintervalle reduzieren die Aussagekraft eines Tests.

Die **Hypothesentests** beschäftigen sich ebenfalls mit dieser Problematik. Jedoch anstatt ein Konfidenzintervall vorzugeben, wird eine Behauptung (Nullhypothese) aufgestellt und versucht diese dann zu widerlegen. Wenn sie widerlegt wird, ergibt sich unmittelbar eine Erkenntnis. Kann sie aber nicht widerlegt werden, dann bedeutet dies nicht, dass die Behauptung zutrifft! In diesem Fall wurde in der betreffenden Stichprobe nur kein Gegenbeweis gefunden. Es ist jedoch nicht auszuschließen, dass dieser in einer weiteren Stichprobe vorkommt. „Bei der Wahl der Hypothese sollte man bedenken, dass sie auch mit sehr hoher Wahrscheinlichkeit angenommen wird, wenn sie „nur“ ein bisschen falsch ist“ [37]. Die Behauptung, die der Hypothese widerspricht, wird auch Alternative genannt. Ein Beispiel einer Hypothese wäre: „Der Mittelwert zweier Stichproben ist gleich“. Die entsprechende Alternative kann dann lauten „Der Mittelwert zweier

Stichproben ist nicht gleich“. Es existieren eine Reihe von Hypothesentests. Es muss ein Test gewählt werden, zu dem die Nullhypothese passt und dessen spezifischen Annahmen durch die Testumgebung erfüllt sind. Einige Testverfahren werden in diesem Kapitel vorgestellt.

Es können zwei Arten von Fehlern auftreten.

- *Fehler der 1. Art*, wenn die Nullhypothese tatsächlich wahr ist, aber irrtümlich vermutet wurde, dass sie falsch sei.
- *Fehler der 2. Art*, wenn die Nullhypothese tatsächlich nicht gilt, aber sie irrtümlich angenommen wird.

Im Allgemeinen gilt allerdings, dass die Minimierung der Fehler 1. Art und der Fehler 2. Art gegenläufige Ziele sind und ein vernünftiger Ausgleich zwischen beiden Fehlern gefunden werden muss [37]. Die Wahrscheinlichkeit, dass ein Fehler 1. Art auftritt wird als Signifikanzniveau α bezeichnet und kann frei gewählt werden. Üblicherweise werden α Werte von 1% oder 5% gewählt. In der Regel werden in Abhängigkeit des Signifikanzniveaus aus bestehenden Tabellen ein Ablehnungsbereich bestimmt. Ist der jeweilig berechnete Wert des Tests innerhalb des Ablehnungsbereich gilt die Nullhypothese als abgelehnt. Der Punkt an dem der Ablehnungsbereich beginnt, wird auch „kritischer Wert“ genannt.

Eine Stichprobe kann durch Charakteristika beschrieben werden. Häufig untersuchte Charakteristika sind das Lagemaß, Streumaß, Formmaß, Verteilungsmaß und Ausreißer. Im Folgenden werden nun diese Charakteristika näher beschrieben und einige Verfahren zu ihrer Untersuchung genannt.

Lagemaße

Das Lagemaß beschreibt das „Zentrum“ der Stichprobe. Viele Berechnungen setzen diese Maßzahl ein. Für univariate Stichproben werden hier drei Möglichkeiten das Zentrum zu charakterisieren genannt.

- **Arithmetrischer Mittelwert**

Der Mittelwert \bar{x} ist die Summe der Stichprobenwerte geteilt durch den Stichprobenumfang n .

$$\bar{x} = \frac{1}{n} * \sum_{i=1}^n x_i$$

Durch diese Definition fließt jeder Stichprobenwert zu gleichen Teilen in das Ergebnis ein. Ist das nicht erwünscht, kann beispielsweise den Median verwendet werden.

- Median

Der Median \tilde{m} beschreibt den mittleren Wert einer der Größe nach sortierten Stichprobenwerte. Links vom Median liegen genauso viele Stichprobenwerte wie rechts. Besteht die Stichprobe aus einer geraden Anzahl n von Stichprobenwerten wird ein Zwischenwert interpoliert.

$$x_1, x_2, x_3, x_4, \dots, x_n \text{ sei die der Größe nach sortierte Liste der Stichprobenwerte.}$$

$$\tilde{m} = x_{\frac{n+1}{2}} \quad , \text{ wenn } n \text{ ungerade ist}$$

$$\tilde{m} = \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2} \quad , \text{ wenn } n \text{ gerade ist}$$

Der Median betrachtet tatsächlich nicht die einzelnen Stichprobenwerte, sondern nimmt sich einfach das mittlere Element der sortierten Liste heraus. Hierdurch kann das Ergebnis grundlegend verschieden zu dem des arithmetischen Mittelwerts sein, wenn die Stichprobe beispielsweise einige sehr hohe Stichprobenwerte beinhaltet.

- Häufigster Wert (*Mode*)

Der Modus ist der Wert, der am häufigsten oder wahrscheinlichsten in einer Stichprobe vorkommt. Eingesetzt wird er, um qualitative Aussagen über Verteilungen treffen zu können.

Die genannten Lagemaße können in Abhängigkeit der realen Bedingungen der Testumgebung unterschiedliche Ergebnisse liefern. Es hängt dennoch vom Einzelfall ab, welche der Lagemaße tatsächlich sinnvoll sind. Ein Beispiel mit sehr unterschiedlichen Ergebnissen wird in Abbildung 2.3 dargestellt.

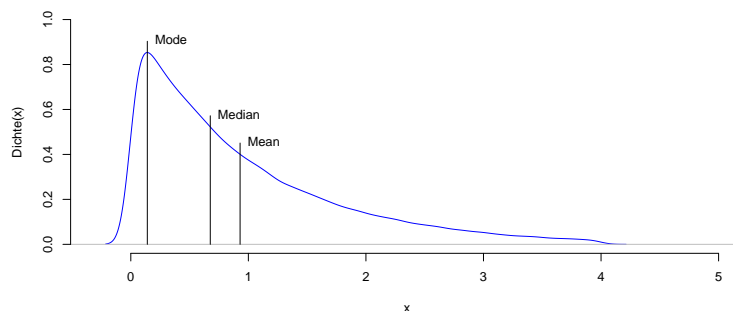


Abbildung 2.3: Beispiel verschiedener Lagemaße

Es existieren Hypothesentests, die das Lagemaß untersuchen. Eine Reihe möglicher Tests sind der „Zweistichproben t-Test“, „Konfidenzgrenzwert und Einstichproben t-Test zur Prüfung des Mittelwerts“, „One Factor Analysis of Variance“ und „Multi-Factor Analysis

of Variance“. Die „One-Factor Analysis of Variance“ ist hierbei eine Verallgemeinerung des „Zweistichproben t-Test zur Prüfung des Mittelwerts“, in dem mehr als zwei Stichproben verglichen werden können. Weitere Details können in [37, 38, 36] nachgeschlagen werden.

Streuemaße

Das Streumaß ist ein weiteres sehr wichtiges Charakteristikum von Stichproben und ist ein Maß für die Streuung der Stichprobenwerte um das Zentrum. Bei der Streuung kann zwischen der Ausbreitung um den Mittelwert und der Ausbreitung an den Ausläufen unterschieden werden. Die einzelnen Streumaße betonen diese zwei Komponenten unterschiedlich stark. Einige Streumaße werden nun vorgestellt:

- **Varianz**

Die Varianz s^2 einer Stichprobe $x_1, x_2, x_3, \dots, x_n$ mit dem arithmetrischen Mittelwert \bar{x} ist definiert durch

$$s^2 = \frac{1}{n-1} * \sum_{i=1}^n (x_i - \bar{x})^2.$$

Durch die Quadrierung des Abstands werden größere Abweichungen stärker betont als kleinere. Sowohl der Mittelwert als auch die Varianz werden negativ durch Ausreißer beeinflusst.

- **Standardabweichung**

Die Standardabweichung s ist die primitive Wurzel der Varianz und stellt daher die ursprüngliche Einheit der beobachteten Zufallsvariable wieder her. Die Standardabweichung wird ebenso stark von Ausreißern beeinflusst wie die Varianz.

$$s = \sqrt{\frac{1}{n-1} * \sum_{i=1}^n (x_i - \bar{x})^2}$$

- **Spannweite**

Die Spannweite R ist die Differenz des minimalen von dem maximalen Stichprobenwerts. Ausreißer könnten diesen Wert maßgeblich beeinflussen. Aussagen über die Ausbreitung der Stichprobenwerte um den arithmetrischen Mittelwert werden nicht geliefert.

$$R = \text{Max}(x_i) - \text{Min}(x_i)$$

- **Mittlere absolute Abweichung vom Mittelwert**

Die „mittlere absolute Abweichung vom Mittelwert“ (AAD) ähnelt der Varianz, wobei auf die Quadrierung des Abstands verzichtet wird. Ausreißer verfälschen aufgrund der Nutzung des arithmetrischen Mittelwerts, als auch durch die Mittelwertbildung über die

Differenzen, dieses Streumaß stark. Wird bei der Differenz anstatt dem Mittelwert der Stichprobe der Median verwendet, so ändert sich dieser Umstand nicht.

$$AAD = \frac{1}{n} * \sum_{i=1}^n |x_i - \bar{x}|$$

- Median der absoluten Abweichung vom Median

Der „Median der absoluten Abweichung vom Median“ (MAD) berechnet im Gegensatz zum AAD nicht das arithmetische Mittel der Differenzen, sondern verwendet dessen Median. Dadurch ist dieses Verfahren wesentlich weniger anfällig gegen Ausreißern als der AAD. Wird auf das Ergebnis der Faktor 1.4826 dazu multipliziert, so kann bei normalverteilten Zufallsvariablen das neue Ergebnis wie die Standardabweichung behandelt werden [39].

$$MAD = m(|x_i - \tilde{x}|)$$

- Interquartilabstand

Der Interquartilabstand (IQR) wird definiert durch den Abstand des unteren zum oberen Quartil. Hierdurch werden die 25% kleinsten und 25% größten Stichprobenwerte weggeschnitten und erhält somit ein Streumaß der restlichen Werte.

$$IQR = \text{oberes Quartil} - \text{unteres Quartil}$$

Es gibt noch weitere Hypothesentests, die sich mit Streumaßen beschäftigen. Zur Untersuchung der Gleichheit von Varianzen kann beispielsweise der „F-Test zur Prüfung der Gleichheit zweier Standardabweichungen“ oder der „Bartlett’s Test“ eingesetzt werden. Weitere Informationen sind mitunter in [37] zu finden.

Formmaße

Bei univariaten Stichproben lassen sich die Formmaße ihrer Verteilungsfunktion in die zwei Typen „Schiefe“ und „Wölbung“ unterscheiden. Da beide mit der Standardabweichung s und dem Mittelwert \bar{x} rechnen, sind sie fehleranfällig für Ausreißer. Außerdem sind bei kleinem Stichprobenumfang die Formmaße recht unzuverlässig. Histogramme eignen sich sehr gut zur Darstellung der Formmaße.

Die Schiefe sk ist ein Maß für die Symmetrie der Stichprobenwerte um den arithmetischen Mittelwert. Ist das Ergebnis negativ, gleich oder positiv, so ist die Stichprobe respektive linksschief, symmetrisch oder rechtschief.

$$sk = \frac{\sum_{i=1}^n (x_i - \bar{x})^3}{(n - 1)s^3}$$

Die Wölbung ku ist ein Maß für die Häufung von Stichprobenwerten. Der statistische Parameter ist groß, wenn sich die Stichprobenwerte um die Mitte der Verteilung scharen oder klein, wenn sie sich gleichmäßig bis zu den Ausläufen verteilen.

$$ku = \frac{\sum_{i=1}^n (x_i - \bar{x})^4}{(n - 1)s^4}$$

Ausreißer

Diese Verfahren suchen innerhalb einer Stichprobe nach Ausreißern. Ausreißer sind im Allgemeinen die Stichprobenwerte, die sich untypisch von der Mehrheit der sonstige Stichprobenwerte abgrenzen, indem sie sehr groß oder sehr klein sind. Die Verfahren können grob in zwei Arten unterteilt werden. Entweder erlaubt das Verfahren lediglich das Prüfen eines Stichproben Wertes, sodass das Verfahren mehrfach iterativ über alle Stichprobenwerte durchgeführt werden muß. Hierbei wird ein erkannter Ausreißer im nächstfolgenden Durchgang weggelassen. Oder es wird nach Ausreißerpaaren gesucht, wobei sich solch ein Paar beispielsweise aus einem sehr großen und einem sehr kleinen Stichprobenwert bildet.

Der „Grubbs Test“ wird beispielhaft genannt, da er sehr leicht durchzuführen ist. Dieser Algorithmus untersucht, wieviele Standardabweichungen ein zu untersuchender Stichprobenwert von dem Mittelwert entfernt ist. Dieses Verfahren arbeitet auf einer univariaten Stichprobe und setzt die Normalverteilung voraus.

Grubbs Test	
Annahmen:	x_1, x_2, \dots, x_n seien unabhängig, identisch und $\mathcal{N}(\mu_1, \sigma_1)$ -verteilt. Die Parameter der theoretischen Normalverteilungen μ_1, σ_1 sind nicht bekannt.
Nullhypothese:	Ein Stichprobenwert x_i (i aus $1, \dots, n$) ist ein Ausreißer.
Alternative:	x_i ist kein Ausreißer.
Testgröße:	$G = \frac{ x_i - \bar{x} }{s}$, wobei s die Standardabweichung ist.
Signifikanzniveau:	α .
Ablehnungsbereich:	$G > \frac{n-1}{\sqrt{n}} * \sqrt{\frac{t_{(\frac{\alpha}{2n}, n-2)}^2}{n-2+t_{(\frac{\alpha}{2n}, n-2)}^2}}$ $t_{(x,y)}$ ist der kritische Wert.

Die Korrektheit der Ergebnisse der Ausreißertests steigt mit dem Stichprobenumfang, denn durch einen größeren Stichprobenumfang wird die zugrunde liegende Verteilungsfunktion besser widerspiegelt. Wird ein Ausreißer durch diese Verfahren gefunden, sollte dieser erst nach reichlicher Überlegung wirklich als solcher deklariert und entfernt werden. Durch eine Automatisierung des Prozesses werden oftmals (auch auf längere Sicht) falsche Schlüsse gezogen. Genauere Informationen können unter anderem in [39] nachgelesen werden.

Verteilungsmaße

Um Verteilungen zu vergleichen, können Anpassungstests eingesetzt werden. Ein Vorteil der Anpassungstests ist ihre einfache und flexible Anwendung in vielen Szenarien. Bei diesen Verfahren wird überprüft wie ähnlich eine Stichprobe einer zugrunde liegenden Verteilung ist. Da es eine Reihe von Verfahren gibt, die sich im Einsatzgebiet überschneiden und zudem in der Realität die Alternative oftmals nur vage definiert werden kann, ist die Auswahl des besten Tests oftmals nicht eindeutig [33]. Ist die Verteilungsfunktion nicht exakt spezifiziert, müssen nicht-parametrische Verfahren eingesetzt werden. Der Chi-square Test ist ein sehr häufig eingesetzter nicht-parametrisches Test. Eine Ausprägung der Chi-square Test ist der Pearson Chi-square Anpassungstest, der nun genauer erklärt wird. Weitere Informationen zu den Anpassungstests sind in [40, 33, 41] zu finden.

Pearson Chi-square Anpassungstest χ^2

Der Pearson Chi-square Anpassungstest vergleicht die empirisch beobachteten Häufigkeiten von disjunkten Ereignissen mit den erwarteten und leitet daraus ein Maß für die Diskrepanz her. Hierzu wird in der Regel der Wertebereich einer Kenngröße in disjunkte Intervalle unterteilt und die Häufigkeiten der Treffer in diesen Intervallen untersucht. Der Algorithmus kann auf allen univariaten Verteilungen eingesetzt werden, die diskret oder stetig sind. Bei diskreten Verteilungen ist der Wertebereich Ω endlich und damit abzählbar. Bei stetigen Verteilungen ist dieser nicht abzählbar, da die Zufallsvariable beliebige Werte annehmen kann, also auf ganz \mathbb{R} definiert ist. Dieser Test kann unter anderem deshalb oft eingesetzt werden, da fast jeder Wertebereich einer Zufallsvariable in disjunkte Intervalle unterteilt werden kann. Durch die Kategorisierung in Intervalle gehen jedoch Informationen der Stichprobe verloren, die ursprünglich zur Verfügung standen. Dadurch kann das χ^2 Verfahren prinzipiell nicht so mächtig sein wie andere Verfahren, die nicht auf Aggregation beruhen [33]. Das Problem des Informationsverlusts bei Aggregation ist auch in anderen Methoden, die darauf beruhen, allgegenwärtig. Andererseits reduziert jede Form der Aggregation die zu speichernde Datenmenge, und ermöglicht bei sehr großen Stichprobenumfang eine performante Bearbeitung. Ein Problem dieses Tests ist die Wahl optimaler Intervallgrenzen. Hierbei sollen die Intervallgrenzen den Verlauf der erwarteten Verteilung möglichst gut beschreiben. Um das Ergebnis nicht zu verfälschen, sollten daher möglichst Intervalle gebildet werden, die gleiche Trefferwahrscheinlichkeiten haben! Dieser Sachverhalt wird oftmals nicht erwähnt. Zudem gibt es eine grobe Richtlinie wie viele Intervalle M gebildet werden sollen. Dieser ist von dem Stichprobenumfang abhängig:

$$M = 2n^{(2/5)}$$

Wenn beide Regeln erfüllt sind, entspricht das Ergebnis des Algorithmus näherungsweise

optimal der χ^2 -Verteilung [33]. In der Realität ist dies aber nur schwer zu gewährleisten, weshalb oftmals die Intervallbildung in Abhängigkeit der erwarteten Verteilungsfunktion durchgeführt wird. Selbst dann soll sichergestellt sein, dass keins der Intervalle keine oder im Vergleich zu den anderen Intervallen außerordentlich wenig Treffer hat oder 20% der Intervalle weniger als 5 Treffer besitzen. Gegebenenfalls sollen diese „seltenen“ Intervalle erneut zusammengefasst werden. Zur Darstellung der Häufigkeiten in den Intervallen eignen sich Histogramme sehr gut.

Pearson Chi-square Anpassungstest	
Annahmen:	X_1, X_2, \dots, X_{n_1} seien unabhängig und identisch verteilt mit gemeinsamer zugrunde liegender Verteilungsfunktion F . Man wähle r Intervalle mittels $a_0 = -\infty < a_1 < \dots < a_{r-1} < a_r = +\infty$ und berechne $E_j = n * (F(a_j) - F(a_{j-1})), 1 \leq j \leq r.$ E_j beschreibt die erwartete Häufigkeit der Treffer im j . Intervalls. Die Variable O_j gebe die Anzahl der X_i mit $X_i \in I_j$ an.
Nullhypothese:	$F_X = F$.
Alternative:	$F_X \neq F$.
Testgröße:	$\chi^2 = \sum_{j=1}^r \frac{(O_j - E_j)^2}{E_j}$
Signifikanzniveau:	α .
Ablehnungsbereich:	$\chi^2 > \chi^2_{(\alpha, r-c)}$ Wenn $n_1 = n_2$, gilt i.a. $c = 0$, ansonsten $c = 1$. $\chi^2_{(\alpha, r-c)}$ beschreibt den kritischen Wert.

Es existieren parallel zu dem genannten Pearson Chi-square Anpassungstest noch eine Reihe weiterer wie beispielsweise der „likelihood ratio test“, der „Freemann-Tukey chi-squared test“ oder der „Rao-Robson Test“. Der „Rao-Robson Test“ ist laut [33] der mächtigste aller Chi-square basierten Tests, benötigt jedoch eine sehr aufwendige Berechnung. Auch wurde bei kleinem Stichprobenumfang festgestellt, dass der „Pearson chi-square“ bessere Ergebnisse liefert als der „likelihood ratio test“ und „Freemann-Tukey chi-squared test“ [40]. Aufgrund der Flexibilität bei der Anwendung und der einfachen Berechnungsformel des Pearson Chi-square Anpassungstests, wird dieser heutzutage am häufigsten eingesetzt.

Bei stetigen Verteilungen kann auch der „Kolmogorov-Smirnov-D Test“ oder der „Anderson-Darling Test“ angewandt werden, solange deren Annahmen erfüllt sind. Jedoch sind diese nicht so universal einsetzbar wie die nicht-parametrischen χ^2 Verfahren. Der „Anderson-Darling Test“ ermöglicht beispielsweise nur Vergleiche mit bestimmten Verteilungen. Beide parametrische Verfahren werden nun noch vorgestellt.

Kolmogorov-Smirnov-D Test

Der „Kolmogorov-Smirnov-D Test“ arbeitet auf stetigen Zufallsvariablen. Dieser Test bildet zunächst aus der zu untersuchenden Stichprobe eine empirische kumulative Verteilungsfunktion „EDF“ und vergleicht diese dann mit einer stetigen, voll spezifizierten kumulativen Verteilungsfunktion „CDF“. Eine kumulative Verteilungsfunktion beschreibt die Wahrscheinlichkeit, dass eine Zufallsvariable kleiner oder gleich x ist:

$$F(x) = Pr[X \leq x], -\infty < x < \infty$$

Die empirische kumulative Verteilungsfunktion ist bei n Stichprobenwerten x_1, x_2, \dots, x_n definiert durch:

$$F_n(x) = \frac{(\text{Anzahl der } X_i < x)}{n}, -\infty < x < \infty$$

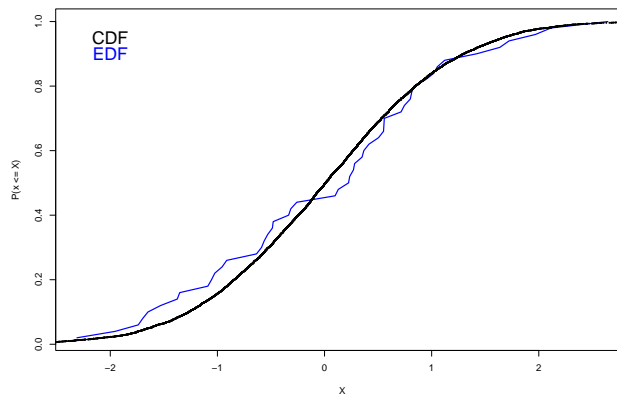


Abbildung 2.4: Beispiel von kumulativen Verteilungsfunktionen

Da sich die kumulativen Verteilungsfunktionen bei $x \rightarrow -\infty$ der 0 und $x \rightarrow +\infty$ der 1 annähern, haben sie dort eine Gemeinsamkeit. Für den Vergleich von Verteilungen ist daher das Verhalten zwischen diesen zwei Enden von Bedeutung. Der „Kolmogorov-Smirnov-D Test“ nimmt als Maß der Diskrepanz den maximalen Abstand dieser Verteilungen. Dieser Test ist durch die Verwendung von kumulativen Verteilungsfunktionen im Gegensatz zu den χ^2 Verfahren auch bei kleinem Stichprobenumfang einsetzbar. Prinzipiell kann dieser Test für Vergleiche mit beliebigen Verteilungen herangezogen werden, solange diese bekannt, stetig und voll spezifiziert sind, das heißt es darf kein Parameter geschätzt worden sein. Ist das nicht erfüllt, liefert dieser Test falsche Ergebnisse. Dieser Test reagiert sensitiver auf Schwankungen nahe am Zentrum als an den Ausläufen.

Kolmogorov-Smirnov-D Test	
Annahmen:	Die Stichprobenvariablen X_1, X_2, \dots, X_n seien der Größe nach geordnet, unabhängig und identisch verteilt und durch die empirische kumulative Verteilungsfunktion F_n darstellbar. Die Verteilungsfunktion F ist stetig und voll spezifiziert.
Nullhypothese:	Die Stichprobenvariablen gehorcht der Verteilungsfunktion F .
Alternative:	Die Stichprobenvariablen gehorcht nicht der Verteilungsfunktion F .
Testgröße:	$\mathcal{D} = \sup_x F_n(x) - F(x) $
Signifikanzniveau:	α .
Ablehnungsbereich:	$\mathcal{D} > D_n$ D_n beschreibt den kritischen Wert

Anderson-Darling Test

Der „Anderson-Darling Test“ erwartet im Gegensatz zu dem „Kolmogorov-Smirnov Test“ nicht eine voll spezifizierte Verteilungsfunktion, sondern funktioniert nur für bestimmte bekannte Verteilungen. Der Ablehnungsbereich muss bekannt sind. Diese Tabellen existieren für Normal, Lognormal, Exponentiell, Weibull, Logistic und „Extremwert Typ 1“-Verteilungen. Durch diese zusätzliche Annahme kann dieser Test sensitiver prüfen, sodass er mächtiger als der „Kolmogorov-Smirnov-D Test“ ist und teilweise bessere Ergebnisse liefert. Zudem ist dieser Test auch an den Ausläufen genauer. Sind die statistischen Parameter der zugrunde liegenden Verteilung nicht bekannt, können trotzdem basierend auf geschätzten Maße Ergebnisse berechnet werden, da bei größeren Stichprobenumfang sich das Ergebnis bestimmten Punkten asymptotisch annähert. Nähere Informationen sind in [33] zu finden. Das Ergebnis ist dann natürlich mit erhöhter Vorsicht zu genießen.

Anderson-Darling Test	
Annahmen:	Die Stichprobenvariablen X_1, X_2, \dots, X_n seien der Größe nach geordnet, unabhängig und identisch verteilt. Die Verteilungsfunktion F ist stetig und bekannt.
Nullhypothese:	Die Stichprobenvariablen gehorcht der Verteilungsfunktion F .
Alternative:	Die Stichprobenvariablen gehorcht nicht der Verteilungsfunktion F .
Testgröße:	$\mathcal{A}^2 = -n - \sum_{i=1}^n \frac{(2i-1)}{n} [\ln F(X_i) + \ln (1 - F(X_{n+1-i}))]$
Signifikanzniveau:	α .
Ablehnungsbereich:	$\mathcal{A}^2 > k * A_F$ A_F beschreibt den kritischen Wert. k ist ein Faktor, der sich aus diversen Größen ergibt (n,..). Nähere Details in [33].

In vielen Fällen ist der „Anderson-Darling Test“ mächtiger als der „Kolmogorov-Smirnov-D

Test“. Dies ist vor allem in Situationen der Fall in dem Ausläufe für die Ergebnisse entscheidend sind. Für umfassendere Kriterien zur Auswahl eines Tests sei auf die einschlägige Literatur verwiesen [33]. Es gibt zudem noch eine Fülle weiterer Tests, die auf kumulativen Verteilungsfunktionen basieren. Prinzipiell gilt, dass die parametrischen Tests, da sie mehr Annahmen an das Szenario stellen, mächtiger sind als die nicht-parametrischen. Wichtig ist aber, dass die jeweiligen Annahmen auch erfüllt sein müssen. Die Theorie der robusten Statistik beschäftigt sich mit diesem Sachverhalt und wird im nächsten Kapitel vorgestellt.

2.2.3 Robuste Lage- und Streumaße

Die Robuste Statistik beschäftigt sich mit der Tatsache, dass viele der üblichen Annahmen in der Statistik, wie beispielsweise Normalverteilung, Linearität und Unabhängigkeit von Zufallsvariablen, nur eine Näherung an die Realität darstellen.

A tacit hope in ignoring deviations from ideal models was that they would not matter; that statistical procedures which were optimal under the strict model would still be approximately optimal under the approximate model. Unfortunately, it turned out that this hope was often drastically wrong; even mild deviations often have much larger effects than were anticipated by most statisticians. [42]

Selbst wenn es theoretisch optimale Tests für die jeweilige Problemstellung gibt, geben diese selten Auskunft über die Qualität des Ergebnisses, wenn die Annahmen auf die sie beruhen, in der Praxis nicht ganz oder gar nicht zutreffen. Ein wichtiger und oft angewandter Satz in der Statistik ist der „Zentrale Grenzwertsatz“, der besagt, dass die durch Summenbildung einer genügend großen Anzahl identisch verteilter und unabhängiger Zufallsvariablen gebildete Zufallsvariable gegen eine Normalverteilung konvergiert. Aber wie gut die Realität durch diese Zufallsvariablen abgebildet wird, wird oftmals nicht untersucht. Die Ziele der Robusten Statistik sind:

1. Auswahl statistischer Parameter, die in Abhängigkeit der Problemstellung die Stichprobe am besten beschreibt.
2. Identifikation und Behandlung von der Erwartung abweichender Stichprobenwerte (Ausreißer).
3. Bewertung der Qualität der Ergebnisse von Algorithmen, wenn deren Annahmen nicht oder nur teilweise erfüllt sind.

Robuste Statistiken werden beispielsweise sinnvoll eingesetzt, wenn eine Unempfindlichkeit gegenüber kleinen Abweichungen von dem idealisierten Modell benötigt wird. Dies bedeutet

aber letztendlich auch, dass für den Fall, dass keine Abweichungen in der Stichprobe zu finden sind, oftmals die robusten Statistiken nicht so präzise Ergebnisse liefern wie andere Tests. Daher kann es je nach Situation sinnvoll sein nicht-parametrische, parametrische oder auch robuste Verfahren einzusetzen. Prinzipiell sollte unterschieden werden zwischen qualitativ hochwertigen Stichproben mit keinen gravierenden Ausreißern und den in der Praxis typischen Stichproben mit vielen Ausreißern. In diesem Fall dürfen die Ausreißer natürlich nicht unterschlagen werden. Detailliertere Informationen über robuste Statistik finden sich in [43].

Diese Arbeit betrachtet nur einige einfache, univariate, robuste Tests, die Lagemaße und Streumaße beschreiben. Diese werden hier lediglich intuitiv verglichen; für die mathematisch korrekte Beweisführung wird auf [43] verwiesen.

Robuste Lagemaße

Einige Lagemaße, wie arithmetischer Mittelwert und Median, wurden bereits in Kapitel 2.2.2 auf Seite 20 vorgestellt. Überlicherweise ist der arithmetrische Mittelwert ein guter Schätzer des „Zentrums“ einer symmetrisch-verteilten Verteilungsfunktionen. Ist diese Bedingung aber nicht erfüllt oder existieren Ausreißer in der Stichprobe, dann beschreibt er nur äußerst ungenau das wahre „Zentrum“ der zugrunde liegenden Verteilungsfunktion. Dies liegt daran, dass jeder Stichprobenwert zu gleichen Teilen in den Mittelwert einfließt, unabhängig davon wie weit entfernt vom wahren „Zentrum“ er liegt. Die Stichprobe 7, 8, 9, 10, 11, 12, 13 beispielsweise streut sehr schön um den Mittelwert 10. Wird aber nun lediglich der Punkt 50, der einen vermeintlichen Ausreißer darstellt, hinzugefügt, so verschiebt sich der Mittelwert gravierend zu 15. Dies verdeutlicht, dass der arithmetrische Mittelwert kein robuster Schätzer für das „Zentrum“ ist. Im Gegensatz dazu ändert sich der Median hierbei nur geringfügig von 10 auf 10.5. Der Median stellt in diesem Fall, wenn von einer symmetrisch-verteilten zugrunde liegenden Verteilungsfunktion ausgegangen wird, das wahre „Zentrum“ besser wieder und ist daher wesentlich robuster gegenüber Ausreißern als der arithmetrische Mittelwert. Da der Median ein robuster Schätzer ist, ist wie bereits erwähnt die statistische Präzision jedoch schlechter als die des arithmetrischen Mittelwerts. Dies wird in diesem Beispiel aber nicht veranschaulicht.

Es existieren noch eine Reihe weiterer robuster Verfahren zur Bestimmung der Lage einer Stichprobe:

- **Mid-Mean**

Bei diesem Verfahren wird der arithmetrische Mittelwert der Stichprobe zwischen dem 0.25-Quantil und dem 0.75-Quantil gebildet. Durch die Quantile werden jeweils die 25% kleinsten und größten Werte entfernt und fließen daher nicht in die durch Ausreißer

anfällige Mittelwertsberechnung ein. Sobald nach der Verkürzung aber immer noch Ausreißer in der Stichprobe vorzufinden sind, wird das Ergebnis dennoch verfälscht. Da oftmals in der Praxis nicht bekannt ist, wieviele Ausreißer vorkommen können, kann eine zu enge Wahl der Schranken zu Fehlern führen.

- **Trimmed Mean**

Der „Trimmed Mean“ ist die Verallgemeinerung des „Mid-Mean“ bei dem jedoch beliebige Quantile benützt werden dürfen. Daher unterliegt es den gleichen Beschränkungen.

- **Winsorized Mean**

Dieses Verfahren arbeitet ähnlich zu dem der „Trimmed Mean“. Die Stichprobe wird aber nicht gekürzt, sondern alle Stichprobenwerte außerhalb der Quantilgrenzen auf eben den Wert an der jeweiligen oberen oder unteren Grenze gesetzt. Durch dieses Vorgehen werden die Stichprobenwerte an den Ausläufen nicht einfach ignoriert, sondern fließen in ihrem Wert abgeschwächt immer noch in das Ergebnis ein.

Die Ergebnisse dieser drei robusteren Verfahren sind bei kleinen Abweichungen von dem idealisierten Modell aufgrund der Verwendung des arithmetischen Mittelwert auf den bereinigten Stichproben präziser als der Median. Jedoch muss beachtet werden, dass durch Ausreißer stärker „verschmutzte“ Stichproben, die Ergebnisse eher von den Ausreißern negativ beeinflusst werden können als beim Median.

Robuste Streumaße

Einige Streumaße wurden bereits in Kapitel 2.2.2 auf Seite 22 vorgestellt. Die Varianz und Standardabweichung sind gute Streumaße, wenn von einer symmetrischen Verteilung der zugrunde liegenden Verteilungsfunktion ausgegangen wird und die Stichproben keine gravierenden Ausreißer beinhaltet. Ist dies nicht gegeben, empfiehlt sich der AAD und MAD, da diese Abweichungen an den Ausläufen weniger stark bewerten. Der MAD ist durch die Verwendung des Medians hierbei besonders tolerant. Ebenfalls kann aber auch der IQR eingesetzt werden, der aber extreme Werte in den Ausläufen vollends ignoriert. Die Spannweite ist leicht zu berechnen, eignet sich aber nur selten als gutes Streumaß. Da er lediglich den Abstand zwischen dem kleinsten und größten Extremwert erfasst, ist er nicht robust. Es kann aber sinnvoll sein, dieses Verfahren in Verbindung mit einem anderen, zum Beispiel der Varianz, einzusetzen, um eine bessere Indikation für die Streuung an den Ausläufen zu erhalten.

Die Wahl der richtigen Lage- und Streumaße ist, wie bereits schon öfters erwähnt, stark von dem Experiment abhängig und bedarf oftmals Erfahrung über die Natur der zugrunde liegenden Verteilungsfunktion. Bestehen die Stichproben aus vielen Ausreißern, sollten die Annahmen an den Versuch oder die Qualität der Stichprobe in Frage gestellt werden [39].

3 Entwurf eines Anomalie-basierten Intrusion Detection Systems

Nachdem in Kapitel 2 sowohl die verschiedenen Ausprägungen von Intrusion Detection Systemen vorgestellt wurden, als auch ein grober Überblick über die mathematischen Möglichkeiten zur Auswertung von Stichproben vermittelt wurden, werden nun die Designentscheidungen für das entwickelte Intrusion Detection Systems „SNADS“ vorgestellt. SNADS steht für „Statistical Network Anomaly Detection System“ und versucht Anomalien auf Netzwerkebene unter Verwendung von statistischen Algorithmen zu erkennen. Dabei wird *Anomalie* in dieser Arbeit als Abweichung von der Norm definiert.

3.1 Anforderungen

Das grundlegende Problem bei Anomalie-basierten IDS ist die schlechte *Effektivität*. Sie beschreibt in diesem Zusammenhang die Praxistauglichkeit des Systems und bestimmt sich unmittelbar durch den Aufwand bei der nachträglichen Bewertung der Anomalien. Aufgrund dieser Problematik finden sich derzeit kaum Anomalie-basierte IDS, die erfolgreich in der Praxis eingesetzt werden können. Das Ziel dieser Arbeit ist es daher, ein effektives Anomalie-basiertes Intrusion Detection System zu entwerfen.

Wie bereits in Kapitel 2.1.1 auf Seite 10 beschrieben, vergleichen Anomalie-basierte Intrusion Detection Systeme (ADS) aktuelle Charakteristika mit den zu erwartenden. Daraus berechnet sich ein Maß für den Unterschied der jeweiligen Eigenschaften. Typische Charakteristika beschreiben Eigenschaften wie die Verbindungsdauer, das Datentransfervolumen und die Anzahl an Verbindungen. Da diese Systeme die zu erwartenden Charakteristika jedoch aus realen Datenflüssen herleiten und diese Ströme üblicherweise nicht das ganze Spektrum des legitimen Verhaltens repräsentieren, leiden sie prinzipiell unter vielen Fehllarmen. Eine große Anzahl an Fehllarmen führt jedoch zu einem enormen Anstieg des administrativen Aufwands bei der nachträglichen Bewertung der Anomalien. Die Effektivität des Systems wird dadurch erheblich negativ beeinflusst. Daher ist die sinnvolle Reduzierung der Fehllarme eine wichtige Anforderung an das System.

Ebenfalls erforderlich ist es, das System individuell auf die Bedürfnisse des Betreibers einzustellen. Denn nur dann können die Sicherheitsrichtlinien ordnungsgemäß abgebildet und das ADS den Gegebenheiten des zu prüfenden Rechnernetze angepasst werden. Die Sicherheitsrichtlinie gibt dem System daher unmittelbar vor, was als Anomalie verstanden werden soll. Zusammenfassend werden folgenden Anforderungen an das System gestellt:

- Geringer administrativer Aufwand bei der nachträglichen Bewertung der Anomalien
- Reduzierung der Fehlalarme
- Individuelle Konfiguration

3.2 Lösungsansatz zur Effektivitätssteigerung

Wie in Kapitel 3.1 beschrieben hängt die Effektivität der ADS von mehreren Faktoren ab. Aus den beschriebenen Anforderungen resultieren folgende Lösungsansätze:

- Einsatz robuster Statistiken zur Minimierung der Fehlalarme
- Sinnvolle Selektion nach der Schwere der Anomalien
- Automatische Ursachenbestimmung
- Flexible Parametrisierung von
 - Algorithmen
 - Betrachtungszeiträumen
 - Arbeitsanweisungen
- Bereitstellung von Detailinformationen zu den Anomalien

Die Lösungsansätze tragen insgesamt zu einem wesentlich geringeren administrativen Aufwand bei der nachträglichen Bewertung der Anomalien bei und werden nun der Reihe nach beschrieben.

Entscheidend zur Reduzierung der Fehlalarme sind unter anderem die Qualität der Daten und die eingesetzten Verfahren. Die Qualität der Daten ist im Vorfeld der Berechnung oftmals nicht bekannt, jedoch sind sie üblicherweise mit Ausreißern gespickt und neigen öfters zu starken Schwankungen. Aufgrund dessen werden zur Bewertung der Diskrepanz zwischen aktuellen und zu erwartenden Charakteristika robuste Statistiken eingesetzt, da diese, wie in Kapitel 2.2.3 auf Seite 29 beschrieben, wesentlich toleranter gegenüber Schwankungen sind. Der administrative Aufwand kann zusätzlich durch nachträgliche Selektion innerhalb der berechneten Anomalien verringert werden [16]. Die Selektion orientiert sich hierbei an der Schwere der Diskrepanz und kann über den Grad der Selektion individuell festgelegt werden. Dieser gibt vor, welche Anomalien unterdrückt werden, wobei bei den weniger gravierenden

Anomalien angefangen wird. Im Allgemeinen gilt, dass Anomalien, die auf geringeren Abweichungen beruhen, eher Fehlalarme darstellen als gravierende Anomalien, sodass durch das Verfahren der Selektion eher Fehlalarme als tatsächliche Alarme unterdrückt werden. Zudem ist das Interesse an den gravierenden Anomalien in der Praxis üblicherweise größer und die aufwendige Analyse der großen Anzahl kleiner Anomalien oftmals aufgrund zeitlicher Gesichtspunkte nicht möglich. Durch die Selektion besteht allerdings die Gefahr, dass tatsächliche Angriffe nicht erkannt werden, womit sich die Vollständigkeit der Erkennung reduziert. Axelsson [2] bekräftigt das Vorgehen indem er erklärt, dass aufgrund der „Vernachlässigung der Basisrate“ (*base-rate-fallacy*) die Effektivität Anomalie-basierter Intrusion Detection Systeme hauptsächlich durch die Möglichkeiten zur Reduzierung der Fehlalarme bestimmt wird, statt durch die vollständige Erkennung von Angriffen. In unserem System bleibt es jedoch dem Administrator überlassen, welcher Kompromiss zwischen nachträglichem Aufwand und Vollständigkeit gewählt wird. Dem Administrator wird daher je nach seinen Wünschen nur noch die Spitze des Eisberges präsentiert.

Eine Arbeitserleichterung kann zusätzlich erreicht werden, wenn den Anomalien typische Ursachen zugewiesen werden. Es ist jedoch zu erwarten, dass aufgrund mangelhafter Informationen (vgl. Kapitel 2.1.1 auf Seite 8) eine zu spezifische Ursachenforschung oftmals nicht korrekt durchgeführt werden kann. Daher untersucht SNADS zunächst nur relativ grundlegender Ursachen von Anomalien. Die individuelle Konfiguration wird in vielen Bereichen ebenfalls gefordert, da sie wie beschrieben eine fehlerfreie Bewertung fördert. Sie zeigt sich unter anderem bei der Wahl der Parameter der eingesetzten Verfahren. Die Sensitivität der Auswertung kann dadurch vom Administrator frei festgelegt werden, sodass die Genauigkeit der Analyse den Bedürfnissen der Sicherheitsrichtlinien angepasst werden können [44]. Aufgrund dessen werden Fehlalarme vermieden, die den Betreiber ohnehin nicht interessieren, da sie laut seiner Sicherheitsrichtlinie nicht relevant sind. Ebenfalls sehr wichtig ist die flexible Wahl des Betrachtungszeitraums aus dem die Werte der zu erwartenden Charakteristiken gebildet werden. Denn es gilt im Allgemeinen, dass aufgrund des Zentralen Grenzwertsatzes gravierende Schwankungen bei längeren Betrachtungszeiträumen weniger stark ins Gewicht fallen als bei kürzeren (vgl. Kapitel 3.5.2 auf Seite 47). Dieser Sachverhalt kann ausgenutzt werden, um die Qualität der Daten und somit der Bewertung der Anomalie zu steigern. Ebenso flexibel sind die Arbeitsanweisungen (Regeln), die für jede IP-Adresse des Rechnernetzes exakt beschreiben, welche Anomaliekenngrößen geprüft werden sollen. Dies hat einerseits den Vorteil, dass das SNADS in seiner Rechenzeit wirklich auch nur das vorgegebene erledigt, andererseits reduziert die flexible Gestaltung der Regeln auch die Fehlerrate, indem die realen Charakteristika eines jeden Rechners exakter nachgebildet werden können. Vorstellbar ist beispielsweise ein spezielles Ereignis besonders zu behandeln, wie etwa ein täglich in der Früh laufendes Backup einer Festplatte. So können die daraus resultierenden potentiellen Anomalien schon im Vorfeld unterdrückt werden.

Zur Reduzierung des nachträglichen Arbeitsaufwands ist es zudem sinnvoll, möglichst viel des Kontextes der betreffenden Anomalie zu konservieren und sie dem Administrator für seine eigenen Recherchen zu präsentieren. Denn die bei der Berechnung verwendeten Informationen reichen alleine oftmals nicht aus, um eine nachträgliche Bewertung des Alarms effektiv und effizient zu tätigen.

3.3 Konzept

Das grundsätzliche Ziel eines Anomalie-basierten Intrusion Detection Systems ist die Bestimmung einer Anomalie. Dies wird in dem ADS „SNADS“ in mehrere Schritte unterteilt. Zunächst nimmt „SNADS“ die verbindungsabhängigen Datensätze entgegen, bearbeitet sie je nach Definition der Regeln auf und legt die einzelnen Datenpunkte der Anomaliekenngrößen in einer Datenbank ab. Im nächsten Schritt wird basierend auf diesen Datenpunkten unter Verwendung von robusten Heuristiken nach Anomalien gesucht und diese dann quantitativ bewertet. Dieser Vorgang wird *Anomaliebewertung* genannt. Sämtliche zur Verfügung stehenden Datenpunkte sind empirisch gemessen worden und sind daher Realisierungen einer Stichprobenvariable und werden in den nächsten Kapiteln auch als solche behandelt. Der letzte Schritt ist die *Eskalationsbewertung*, die die Schwere der Anomalien unter zeitlichen Gesichtspunkten näher untersucht und zudem versucht die Ursache des Vorfalls zu konkretisieren. Mit *Anomaliebestimmung* wird daher der Vorgang der Anomaliebewertung mitsamt Eskalationsbewertung bezeichnet. Der Vorgang der Anomaliebestimmung ist in Abbildung 3.1 dargestellt. Die einzelnen Schritte werden nun präzisiert und in Kapitel 3.5 detailliert beschrieben.

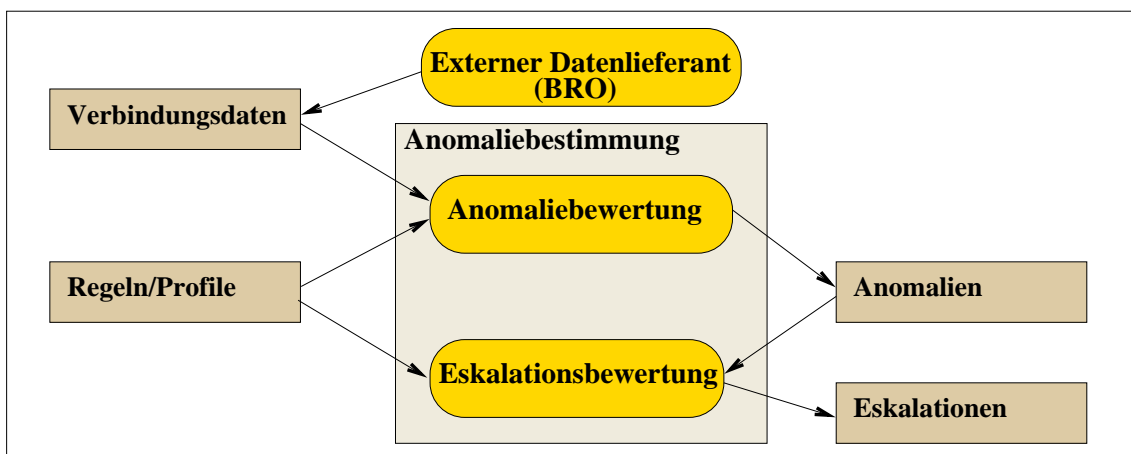


Abbildung 3.1: Vorgang der Anomaliebestimmung

Zur Anomaliebewertung benötigt SNADS Informationen über die Datenflüsse. Um den Aufwand bei der Realisierung des Systems zu reduzieren, werden zunächst Daten aus Drittquellen herangezogen. Vor allem wird dadurch das Abbilden der Protokollsemantik durch aufwendige Logiken eingespart. Das in Kapitel 2.1.2 auf Seite 13 beschriebene IDS BRO erkennt beispielsweise bei TCP Verbindungen, ob ein Three-Way-Handshake zur Verbindungsaufnahme stattgefunden hat, oder ob die bestehende Verbindung ordnungsgemäß beendet wurde. Aufgrund der strikten Trennung von Analyseregeln und der Ereignisgewinnung (vgl. Seite 13) ist BRO auch sehr leicht an verschiedene Bedürfnisse anpassbar. Die Anpassung wird durch BRO's flexible Sprache weiter gestützt. Da BRO mit Ereignissen arbeitet und sie später mittels Logiken analysiert werden, ist es auch vorstellbar die von SNADS ermittelten Anomalien als Ereignisse dem System wieder zurückzuführen. Dies würde dann eine Korrelation der Ereignisse verschiedener Analysetechniken ermöglichen und zu einer exakteren Aussage beitragen. Dadurch dass BRO an der Universität eingesetzt und auch weiterentwickelt wird, wurden hier bereits viel Erfahrung mit dem System gesammelt, was den Einsatz zudem erleichtert. Als Anomalie-basiertes IDS verwendet SNADS bei der Entscheidungsfindung nur Informationen, die aus dem Header oder aus den Eigenarten des Paketflusses (z.B. Verbindungsdauer) extrahiert werden und bedarf daher keiner Informationen aus dem Datenteil der Internetpakete (*pay-load*). Da BRO die Datenflüsse in einer verbindungsartige Struktur (*Flows*) speichert, werden diese Informationen in dieser Form direkt an SNADS übergeben. Die verbindungsabhängigen Datensätze zeichnen sich durch eine sehr starke Kompression des ursprünglichen Datenflusses aus und minimieren den erforderlichen Datenaustausch zwischen BRO und SNADS enorm. Nachteilig ist jedoch, dass natürlich nur noch die übermittelten Informationen sichtbar sind und weitere Details verloren gehen.

Eine Anomalie-basierte Erkennung von Angriffen zeichnet sich, wie bereits in Kapitel 2.1.1 auf Seite 10 beschrieben, durch den Vergleich zwischen dem derzeitigen und dem erwarteten Charakteristika der Datenflüsse aus. *Profile* beschreiben hierbei, wie aus abgelegten Datensätzen die zu erwarteten Eigenschaften gebildet werden. SNADS ermöglicht außerdem eine einfache Spezifikations-basierte Analyse auf den gesammelten Daten. *Regeln* bestimmen hierbei, welche Handlungen als legitim eingestuft werden. Bei den Regeln gibt es noch die Abstufung der *Preprozessor-Regeln*. Der Unterschied zwischen ihnen ist, dass sie ohne aufwendige Berechnung auskommen und direkt eine Anomalie ausrufen können. Beispielsweise kann eine Preprozessor-Regel lauten, dass eine bestimmte IP-Adresse auf einen gewissen Port keinen Datenverkehr aufweisen darf.

In Abhängigkeit, ob ein Profile oder eine Regel bearbeitet wird, ändert sich der Aufwand der Anomaliebewertung. Da die Regeln fest kodiert sind und die Vergleichskriterien, in diesem Fall Schranken, nicht extra berechnet werden müssen, ist der Rechenaufwand im Vergleich zur Bearbeitung von Profilen wesentlich geringer. Denn hierbei muss mit Hilfe von Logiken, erst auf Basis gesammelter Informationen, die erwarteten Charakteristika abgeleitet werden, um

danach die Anomaliebewertung durchführen zu können. Profile und Regeln sind für einzelne IP-Adressen des zu überwachenden Netzes definiert, können aber, um die Administration zu erleichtern und den Überblick zu fördern, zusätzlich für Gruppen von IP-Adressen festgelegt werden. Dies ist beispielsweise vorteilhaft, um die Charakteristika eines ganzen Subnetzes zu prüfen. Ebenso förderlich ist es, wenn die gleichen Profile und Regeln auch zeitgleich für mehrere IP-Adressen eingesetzt werden können. Dadurch ist es möglich Arbeitsanweisungen einmal zu definieren und sie dann für beliebig viele IP-Adressen zu nutzen. Beispielsweise könnten Profile für Server- oder Desktop Systeme definiert werden und dann für ähnliche Systeme wiederverwendet werden. Änderungen an diesen Profilen würden sich dann auf alle IP-Adressen der Gruppe auswirken. Dies minimiert den Aufwand für die Pflege der Regeln und Profile.

Überlicherweise ist die Anzahl an Alarmen bei Anomalie-basierten Systemen sehr hoch, weshalb in SNADS eine Selektion auf der Menge an Alarme durchgeführt wird. Die Eskalationsbewertung orientiert sich bei der Entscheidungsfindung an das menschliche Denkmuster und ist in zwei Phasen unterteilt. Im ersten Schritt werden die Anomalität der Datensätze für definierte Zeitintervalle bestimmt und abgelegt. Im zweiten Schritt werden die gefundenen Anomalien nach dem zeitlichen Auftreten geordnet, um eine Tendenz für die Schwere der Anomalien zu bestimmen. Nur wenn die Tendenz eine Schranke überschreitet, wird letztendlich eskaliert, also die Anomalien dem Administrator präsentiert. Diese Methode erlaubt dem System eine Ordnung der Angriffe nach der Heftigkeit ihres Auftretens. Durch diese Methode können zeitlich sehr kurze Änderungen der Charakteristika herabgestuft werden. Das System bietet zusätzlich die Möglichkeit die Anomaliebewertungen des ersten Schrittes einzusehen. Dadurch werden den Administrator grundlegende Informationen zur Verfügung gestellt und ermöglicht das Ziehen von eigenen Schlüssen. Die Parameter der Heuristiken und die Kriterien, die die Eskalationsbewertung näher beschreiben, sind in den Kapiteln 3.5.2 und 3.5.3 zu finden.

Eines der grundlegenden Probleme mit denen SNADS umgehen muss, ist die *Datenmenge*. Es gilt, dass desto mehr Datenverkehr kontrolliert wird, auch mehr Daten zu verarbeiten sind und gespeichert werden müssen. An dem Knotenpunkt der Münchner Universitäten (inklusive einiger anderer Forschungseinrichtungen) werden an einem Tag zur Mittagszeit grob geschätzt 800 neue Verbindungen pro Sekunde gemessen. Zudem umfasst das Netz einige tausend IP-Adressen. Durch die enorme Menge an Rohdaten, die in diesem Netz anfallen, wird die Notwendigkeit der flexiblen Regeldefinition ersichtlich. Für die Verbindungen müssen nun jeweils die Anomaliewerte von über ein Dutzend Anomaliekenngrößen (vgl. Kapitel 3.5.1 auf Seite 41) gebildet werden, ohne dass die Verarbeitung der Rohdaten sich anstauen dürfen. Dies könnte zu erheblichen Verzögerungen in der Anomaliebestimmung führen und dadurch den Nutzen des ADS reduzieren. Die Arbeitsschritte der Anomalie- und die Eskalationsbewertung sollten möglichst ebenfalls ohne Verzögerungen durchgeführt werden. Der Rechen-

aufwand und die Menge an speichernden Datenpunkte sind in diesem Fall sehr groß. In Anbetracht dessen ist die Skalierbarkeit des Systems und die effiziente Ausführung von besonderer Bedeutung. Da die Informationen üblicherweise aus dem Header der Pakete extrahiert werden, stellen sie in der Regel nur einfache statistische Werte (Anzahl, Größe, ...) dar. Diese Datenpunkte können mit wenig Aufwand aggregiert werden. Das grundlegende Prinzip zur Datenreduktion in SNADS ist die *Aggregation* über die Zeit. Hierbei werden die zu untersuchenden Anomaliekenngrößen für bestimmte Zeitintervalle (*Aggregationsstufen*) gebildet und abgelegt. Werden Datenpunkte für größere Zeitintervalle benötigt, reicht es in der Regel die Datenpunkte der kleineren Zeitintervalle auf zu summieren. Wird beispielsweise die Anomaliekenngröße „Anzahl an Verbindungen“ einer IP-Adresse gesammelt, so werden Datenpunkte für jede „Minute“¹ des Tages gespeichert. Aus diesen können dann die Datenpunkte für die nächst größere Zeiteinheit „Stunden“ gebildet werden. Dieser Vorgang wird nun für „Tage“ und „Wochen“ fortgesetzt. Die einzelnen „Minuten“ werden natürlich auch irgendwann wieder gelöscht, um den Speicherbedarf zu reduzieren. Wann die genannten Datensätze verworfen werden müssen, ist abhängig von den Speicherressourcen und der Datenmenge. Ein Mindestmaß an Datenpunkten sollte den eingesetzten Heuristiken jedoch zur Verfügung stehen, um das Vergangenheitsverhalten möglichst gut qualifizieren zu können. Bei jeder Aggregationsstufe reduziert sich der Speicherbedarf, jedoch gehen auch Informationen verloren, die gegebenenfalls für eine umfassende Auswertung notwendig sind. Die Heuristiken in SNADS arbeiten nur noch auf den Datenpunkten der Aggregationsstufen und müssen daher in der Regel nur die jeweiligen Datenpunkte aus dem Langzeitspeicher auslesen. Auf die große Menge an ursprünglichen Verbindungsdaten wird nicht mehr zugegriffen und dadurch eine effiziente Bearbeitung gefördert. Selbst wenn zur Bewertung andere als die definierten Zeitintervalle herangezogen werden, bedarf es lediglich der Summenbildung einiger Datenpunkte. Andere ADS, wie beispielsweise NIDES, verwenden fließende Zeitfenster, um die Datenmenge zu reduzieren. Allerdings können in diesem Fall die Länge des Zeitfensters als auch die eingesetzten Algorithmen nicht problemlos verändert werden, ohne Fehler bei der Berechnung in Kauf zu nehmen oder die bereits durchgeführten Berechnungen zu verwerfen. Im Gegensatz dazu ermöglicht das in SNADS verwendete Aggregationsprinzip eine nachträgliche und exakte Analyse beliebiger vergangenen Zeitfenster, die aus den Aggregationsstufen gebildet werden können. Ebenso können die hierbei eingesetzten Algorithmen nachträglich leicht geändert werden.

Üblicherweise benötigen Anomalie-basierte Systeme eine *Trainingsphase* in denen sie mit von Anomalien bereinigten Datensätzen gefüttert werden. Eine Trainingsphase ist aber nur bei Charakteristika, die sich über die Zeit nicht ändern zweckmäßig, da nur in diesem Fall das trainierte Modell auch später sinnvoll einsetzbar ist. Bei schwankenden Daten einer Charakteristika muss regelmäßig neu trainiert werden. Alternativ kann auf die Trainingsphase

¹Die Wahl der Aggregationsstufen sind hier nur exemplarische gewählt.

verzichtet werden und stattdessen auf Basis der tatsächlichen, unsauberen Datensätze ohne Unterbrechung adaptiv gelernt werden. Um jetzt gute Ergebnisse erzielen zu können, müssten die eingesetzten Algorithmen dies beachten. Ansonsten müssen die unsauberen Datensätze des IDS zunächst automatisch bereinigt werden. SNADS verzichtet auf eine explizite Trainingsphase, da es äußerst schwer ist für jede IP-Adresse des Netzwerks saubere Datensätze zu bekommen. Zudem ist nicht bekannt, ob alle realen Ausprägungen der Charakteristika der IP-Adresse auch tatsächlich in den jeweiligen Datensätzen enthalten sind [45]. Durch den Einsatz von robusten Statistiken wird versucht das Problem der unbereinigten Datensätze in den Griff zu bekommen. Die Voraussetzungen an die robuste Statistik, vor allem dass die Qualität der Daten nicht zu schlecht ist, müssen jedoch erfüllt sein (vgl. Kapitel 2.2.3 auf Seite 29). Entsprechend erkennt SNADS ein von der Vergangenheit abweichendes Verhalten. Es kann jedoch keine Aussagen liefern, ob das alte oder das neue Verhalten auf einem unerwünschten Sachverhalt beruht! Die letztendliche Bewertung der Anomalien ist daher trotz umfassender Hilfestellung dem Administrator vorbehalten.

3.4 Einordnung in die Klassifikation

Das System SNADS wird nun nach der in Kapitel 2.1.1 auf Seite 6 beschriebenen Taxonomie für Intrusion Detection Systeme eingeordnet und wichtige Vor- und Nachteile nochmals kurz genannt. SNADS arbeitet auf Daten, die von BRO gesammelt werden, und ist daher, wie auch BRO, ein *Netzwerk-basiertes IDS*. Als Netzwerk-basiertes IDS kann es ohne viel Aufwand zur Überwachung vieler Rechner eines Rechnernetzes eingesetzt werden. Aufgrund der vom Ursprungsort entlegenen Datenerfassung sind die Informationen jedoch oftmals lückenhaft und nicht immer ausreichend gesichert.

Wie der Name des Systems schon vermuten lässt, beruht es auf *statistischen Verfahren* zur Analyse von Anomalien. Die *Anomalie-basierte Analyse* wird mittels so genannter Profile durchgeführt und dient zur Erkennung von Änderungen der Charakteristika in den Datenflüssen eines Netzwerks. Die Erkennung von Angriffen beschränkt sich nicht nur auf bereits bekannte Angriffe und benötigt daher keine aufwendige Sammlung an Informationen. Da die auf diese Art ermittelten Anomalien nicht zwangsläufig auch einem tatsächlichem Angriff entsprechen muss, ist der Nachbearbeitungsaufwand in der Regel sehr hoch. SNADS versucht den Aufwand über die in Kapitel 3.2 auf Seite 33 beschriebenen Ansätze zu minimieren. Bei der Analyse werden hauptsächlich robuste Heuristiken eingesetzt, die wenn ihre Annahmen nicht erfüllt sind, weniger fehlerträchtig sind. Zudem können sie dadurch in möglichst vielen Situationen eingesetzt werden, zum Beispiel, wenn die Datensätze Ausreißern beinhalten. Die statistischen Verfahren erfassen Veränderungen des Datenflusses über Zeiträume hinweg und werden deshalb bei der Entscheidungsfindung in Betracht gezogen. Die bei der Anomaliebe-

wertung eingesetzten Algorithmen haben den Nachteil keine gesicherten Aussagen zu liefern. Daher sind sie stets mit etwas Skepsis zu betrachten. Zusätzlich zu der Anomalie-basierten Erkennung ermöglicht SNADS einen *Spezifikations-basierten Ansatz*. Die Regeln geben hierbei ein festes Normverhalten per IP-Adresse vor, nachdem sich die tatsächlichen Charakteristika der IP-Adressen richten müssen.

Da unerlaubtes Verhalten per IP definiert ist und die berechneten Anomalien nur heuristisch erkannt werden, ist es nicht zweckmäßig aktiv gegen Angreifer vorzugehen. SNADS sieht daher lediglich eine *visuelle Meldung* an den Administrator vor.

SNADS ist kein *Echtzeitsystem*. Prinzipiell werden die Rohdaten von BRO zwar in Echtzeit übergeben und könnten unter großem Rechenaufwand mit minimaler Verzögerung analysiert werden. Jedoch neigt, aufgrund der natürlichen Schwankungen der Charakteristika der Datenflüsse eine Anomaliebewertung über sehr kurze Zeiträume, zu Fehlern. Eine Aggregation der Charakteristika über größere Zeiträume ist daher sehr zu empfehlen. Zudem können gravierende Verzögerungen bei der Auswertung entstehen, wenn das Ende einer Verbindung nicht erkannt wird (vgl. Timeout Problematik auf Seite 67). Um den Bedarf an Rechenzeit zu reduzieren, werden die Charakteristika über konkrete Zeiträume (Aggregationsstufen), wie beispielsweise Minute, Stunde und Tag, aggregiert und erst dann die Bewertung durch die Heuristiken durchgeführt. Durch die Definition der jeweiligen Profile, Regeln und Parametern der Algorithmen kann die *Entdeckungszeit* beeinflusst werden. Im Allgemeinen gilt, dass je weiter die Eskalation in die Vergangenheit reicht, desto fehlerfreier ist sie. Nur durch eine frühere Erkennung von Anomalien kann rechtzeitig auf einen Angriff reagiert werden. Jedoch führt dies dann mitunter zu einer höheren Fehlerrate. Aufgrund der potentiell großen Datenmenge und dem Design von BRO ist eine kontinuierliche Datenausgabe ohne größere Modifikationen ohne weiteres jedoch nicht möglich, sodass in zeitlich sehr eng aufeinanderfolgende *Schüben* kommuniziert wird. Daher übergibt die Verbindungsdaten in Schüben, die dann in der Regel auch unmittelbar von SNADS abgearbeitet werden. SNADS ist aber so konzipiert, dass es die Datensätze auch kontinuierlich bearbeiten kann.

SNADS ist ebenso wie BRO *kein über Rechengrenzen hinweg verteiltes System* und findet sich netzwerktopografisch zwischen den zu schützenden Rechnern und den potentiellen Angreifern. SNADS kann Multiprozessor-Architekturen ausnützen, indem es die Rechenlast auf *mehrere Prozesse* eines Rechners verteilt. Die Kommunikation der Prozesse untereinander (IPC) wird beispielsweise durch „Shared Memory“ getätigt. Durch diese Aufteilung kann die Effizienz bei der Ausführung unter Multiprozessor-Architekturen erheblich verbessert werden.

3.5 Komponenten der Anomaliebestimmung

Nachdem das Konzept des neuen Anomalie-basierten Intrusion Detection Systems SNADS vorgestellt wurde, werden in diesem Kapitel wichtige Einzelheiten konkretisiert. Zunächst wird die Auswahl der Anomaliekenngrößen, die von dem System untersucht werden, näher beschrieben. Das nächste Unterkapitel beschäftigt sich mit ihrer Bewertung über relativ kurze Zeiträume. Hierzu werden einige zweckmäßige Algorithmen ausgewählt. Das letzten Unterkapitel beschreibt die Eskalationsbewertung näher. Hierbei werden die ermittelten Anomalien unter Betrachtung der Zeit korreliert, um eine Tendenz für das Ausmaß und der Ernsthaftigkeit der Anomalien festzustellen.

3.5.1 Anomaliekenngrößen

Entscheidend für eine umfassende und lückenlose Analyse eines Rechnernetzes ist die exakte Definition der zu prüfenden Charakteristika der Datenflüsse. Die Charakteristika beschreiben Eigenschaften der Datenflüsse näher und werden in die fünf Untersuchungskriterien, Datentransfervolumen, Anzahl an Paketen, Anzahl an Verbindungen, verwendete Dienste und Verbindungsdauer, unterteilt. Zu jedem Untersuchungskriterium werden im Folgenden markante Eigenschaften identifiziert, die Anomaliekenngrößen genannt werden. Die Anomaliekenngrößen sind für jeweils eine IP-Adressen beschrieben, und repräsentieren die Eigenschaften für einen bestimmten Zeitraum. Der Einfachheit halber wird dies aber nicht jedesmal erwähnt. Als Datengrundlage dienen die von BRO übergebenen Verbindungen.

1. Datentransfervolumen

Diese Gruppe umfasst Anomalienkenngrößen, die auf einer Änderung des Datentransfervolumens beruhen. Zu beachten ist, dass BRO nur die Nutzdaten von etablierten Verbindungen weiterreicht. Dies hat den Nebeneffekt, dass unsere Statistiken durch „Denial of Service“, „Port Scans“, etc. nicht negativ beeinflusst werden. Dies erleichtert die Analyse zumindest teilweise (vgl. Kapitel 3.5.3 auf Seite 54). Andererseits fehlen Informationen, die für bestimmte Situationen sinnvoll sind. Die von BRO übergebenen Verbindungen sind durch eine Flussrichtung und den gesendeten und empfangenen Daten bestimmt. Die Flussrichtung ist entweder ein- oder ausgehend und gibt an wer die Verbindung initiiert hat. Diese Informationen werden unter anderem zur Aggregation benötigt.

Das Datenvolumen ist eine sehr wichtige Größe, da es ein Kostenfaktor für den Betreiber des Netzes ist. Zudem wirkt sich eine stark ausgelastete Leitung negativ auf die Qualität sämtlicher Datenübertragungen aus. Vorteilhaft ist, dass ein über einen längeren Zeitraum

überhöhtes Datentransfervolumen oftmals ein eindeutiges Indiz für einen Angriff ist. In der Praxis ist der Missbrauch bestehender Dienste eines Rechners oftmals die Ursache für ein schwankendes Datentransfervolumen. Der klassische Fall ist ein vom Administrator installierter FTP-Server mit Lese- und Schreibrechten auf Verzeichnisse. Sobald ein Angreifer diese Schwachstelle entdeckt, ist zunächst ein kurzfristiger Anstieg der Datentransferrate zu dem Server hin festzustellen, der gefolgt wird von Zugriffen verschiedener IP-Adressen, die die dort deponierten, unerwünschten Daten dann herunterladen. Dies sorgt dafür, dass der ausgehende Datentransfer rapide ansteigt und dann über einen längeren Zeitraum anhält bis es der Administrator merkt.

Normalerweise sollte das Datenvolumen beim Vergleich verschiedener Tage relativ gleich sein. Lediglich beim Vergleich sehr kurzer Zeiträume dieser Tage kann es zu größeren Abweichungen kommen. Starken Schwankungen ist das Datenvolumen innerhalb eines Tages unterworfen. Es gilt für Firmen üblicherweise, dass das Transfervolumen zu Beginn der Arbeitszeit stetig ansteigt, ihren Höhepunkt zur späten Mittagszeit erreicht und dann bis zum Arbeitsende wieder abfällt. An den sonstigen Zeiten ist der Datentransfer vergleichsweise gering. Natürlich ist dieses Schema nicht für alle Szenarien einsetzbar, da der Verlauf von vielen anderen Faktoren, wie beispielsweise der Arbeitszeit der Mitarbeiter oder dem Serviceport abhängt. Es zeigt jedoch, dass das System bei der Anomaliebestimmung die Tageszeiten beachten sollte. Um diesen Verlauf zu verdeutlichen, ist die Datentransferrate eines Router, der mehrere Dutzend DSL-Leitungen bedient, in Abbildung 3.2 zu sehen. An den Arbeits-

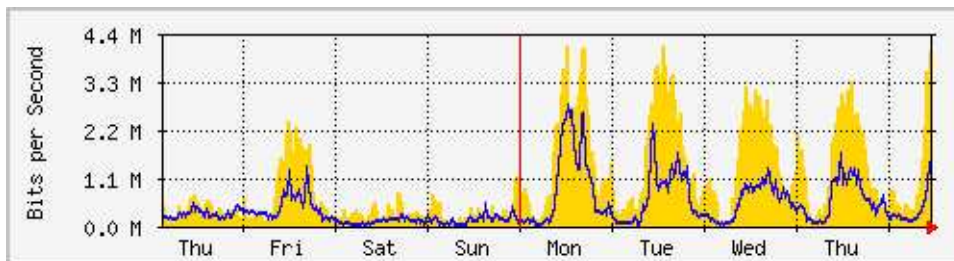


Abbildung 3.2: Exemplarischer Verlauf der Datentransferrate innerhalb einer Woche

tagen von Montag bis Donnerstag kann das beschriebene Schema grob erkannt werden. An den sonstigen Tagen ist die Transferrate deutlich geringer. Am Donnerstag und Freitag der Vorwoche ergibt sich wieder erwarten eine relativ geringe Transferrate. Nach näherer Betrachtung ist dies erklärbar, da der Donnerstag ein gesetzlicher Feiertag in Deutschland war. Der Freitag wurde vermutlich von vielen Arbeitnehmern freigenommen, um ein verlängertes Wochenende zu gewinnen. Das Beispiel verdeutlicht, dass gesetzliche Feiertage, Wochenenden, etc. die Anomaliekenngrößen stark beeinflussen können. SNADS untersucht folgende Anomaliekenngrößen, die das Datentransfervolumen betreffen:

- **Transfermenge**

Bei dieser Anomaliekenngröße wird das übertragende Volumen einer IP betrachtet. Hierbei wird in Abhängigkeit der Flussrichtung eine eigene Anomaliekenngröße verwendet. Die beiden Anomaliekenngrößen unterscheiden zwischen gesendeten und empfangenen Daten und legen sie in eigenen Feldern ab.

- **Transfermenge eines spezifischen Serviceports**

Diese Anomaliekenngröße arbeitet analog zur "Transfermenge". Allerdings wird nicht das gesammte Transfervolumen betrachtet, sondern nur den des spezifizierten Serviceports.

- **Transfermenge aufgeschlüsselt nach Serviceports**

Diese Anomaliekenngröße betrachtet das gesammte Datenvolumen einer IP-Adresse, wobei die Daten in Serviceport-abhängige Felder abgelegt werden. Als Kriterium für den Serviceport gilt stets nur der Ziel-Port. Für jeden Serviceport wird nur noch zwischen gesendeten und empfangenen Daten unterschieden.

- **Transfermenge der Verbindungen eines spezifischen Serviceports**

Bei dieser Anomaliekenngröße wird das transferierte Datenvolumen jeder beendeten Verbindung eines zuvor spezifizierten Serviceports einzeln abgelegt. Dadurch werden Veränderungen der übertragenen Datenvolumen der einzelnen Verbindungen deutlich und im Gegensatz zu den bisherigen nicht nur die aufsummierte Datenvolumen betrachtet. Mit ihrer Hilfe könnte eventuell der Missbrauch eines Dienstes erkannt werden, beispielsweise das Tunneln größerer Datenmengen über eine Dienst. Die Ursachen solcher Änderungen können vielfältig sein und lassen sich nur schwer ergründen.

Die Unterteilung in mehrere Anomaliekenngrößen ist aus folgenden Gründen sinnvoll. Zunächst gewinnt das IDS dadurch an Flexibilität, da die Möglichkeit geboten wird genau zu spezifizieren was das System untersuchen soll. Um die Ergebnisse der Anomaliebestimmung nicht zu verfälschen, kann es vorteilhaft sein bei einer IP-Adresse, die sowohl als HTTP-Server als auch als FTP-Server fungiert, aufgrund des unregelmäßigen Gebrauchs des FTP-Servers lediglich den HTTP-Server zu kontrollieren. Zudem kann die Effizienz der Auswertung des Systems durch die Flexibilität erhöht werden. Des weiteren wird die Kontrolle der Anomaliekenngrößen teilweise unterschiedlich durchgeführt und bedarf einer anderen Datenhaltung. So ermöglichen die zwei erst genannten Anomaliekenngrößen zusätzlich zu der Unterscheidung nach gesendeten und empfangenen Datenverkehr, eine weitere sehr spezifische Auswertung der Datensätze nach der Flussrichtung. So lässt sich beispielsweise bei einer eingehenden HTTP-Verbindung, dessen Anfrage 100 Kilobyte groß war und ein Megabyte an gesendeten Daten produzierte, jedes dieser Felder getrennt bewerten. Die restlichen zwei Anomaliekenngrößen erlauben diese Unterscheidung nicht.

2. Anzahl an Paketen

Durch Pakete werden Daten über das Rechnernetz transportiert. Daher ist die Anzahl der Pakete ein sinnvolles Mittel zur Analyse der Datenflüsse. Außerdem kann es zweckmäßig sein, spezielle ICMP-Pakete, die Kontrollinformationen über den Zustand des Netzes transportieren, zu beachten und ebenfalls gesondert auszuwerten. Änderungen an diesen Anomaliekenngrößen treten oft gekoppelt mit denen der Datentransfervolumen auf, da sich eine Änderung am übertragenen Datentransfervolumen üblicherweise auch in der Anzahl der Pakete niederschlägt. Daher sind diese Anomaliekenngrößen für spätere Korrelation geeignet und können zur besseren Entscheidungsfindung beitragen. Die Anzahl der Pakete einer IP-Adresse ist ebenso wie das Datentransfervolumen stark von der Zeit abhängig und muss mit den gleichen Restriktionen auskommen. Folgende Anomaliekenngrößen werden untersucht:

- **Anzahl an Paketen**

Diese Anomaliekenngröße betrachtet die Gesamtanzahl der Pakete (üblicherweise eines bestimmten Protokolltyps) zur IP-Adresse hin und auch ebenso von ihr weg. Besonders vorteilhaft kann beispielsweise die Untersuchung der ICMP-Pakete sein. Das Vorkommen sehr vieler ICMP „host unreachable“ Pakete ist ein gutes Indiz für einen Scan des Netzwerkes.

- **Anzahl an Paketen eines spezifischen Serviceports**

Hier wird bei der Datenerfassung nur ein Serviceport betrachtet.

- **Anzahl an Paketen von Verbindungen eines spezifischen Serviceports**

Da jede Verbindung aus einer Anzahl von Paketen besteht, bietet sich diese Anomaliekenngröße an. Auch hier gilt, dass die wahre Ursache von Schwankungen bei dieser Anomaliekenngröße nur schwer festzustellen sind und der Einsatz daher höchstwahrscheinlich nicht praktikabel ist.

3. Anzahl an Verbindungen

Die dritte Kategorie sind die Anzahl an Verbindungen. Tatsächlich wird die Anzahl beendeter Verbindungen innerhalb eines Zeitraums gezählt, da das Ergebnis bei dem wichtigen Schritt der Aggregation mehrerer Zeiträume nur dann korrekt ist, wenn jede Verbindung nur einmal gezählt wurde. Ansonsten wäre das Ergebnis verfälscht. Bei dieser Kategorie wird zwischen eingehenden und ausgehenden Verbindungen (Flussrichtung) unterschieden. Die Anzahl an Verbindungen stellt ein gutes Maß für die tatsächlichen Anfragen an und durch die IP-Adresse dar. Sie spiegelt daher sehr gut die Auslastung der IP-Adresse wider. Aufgrund dessen ist auch bei diesen Anomaliekenngrößen der Betrachtungszeitraum von grundlegender Bedeutung. Dadurch dass BRO sowohl die etablierten Verbindungen als auch die Verbindungsversuche aufzeigt, lassen sich diese Anomaliekenngrößen sehr gut für die Erkennung von Scans und

DOS-Attacken einsetzen. Beispielsweise kann ein vermehrtes Vorkommen von Verbindungen, die nicht etabliert wurden auf einen SYN-Flood hindeuten. Bei einem SYN-Flood schickt ein Angreifer viele Pakete zu einer IP-Adresse, ohne überhaupt die Verbindung vollends aufbauen zu wollen. Das Ziel der Attacke ist ein „Denial of Service“. Folgende Anomaliekenngrößen sind definiert:

- **Anzahl an Verbindungen**

Sie erfasst die Anzahl an Verbindungen, wobei unterschieden wird, ob sie etabliert wurden oder nur Verbindungsversuche waren.

- **Anzahl an Verbindungen eines spezifischen Serviceports**

Die Anzahl an Verbindungen für vordefinierte Serviceports.

- **Anzahl an Verbindungen mit besonderen Zuständen**

Diese Anomaliekenngröße fasst nur Verbindungen mit speziellen Zustände zusammen.

- **Anzahl der Verbindungen einzigartiger IP-Adressen eines spezifischen Serviceports**

Diese Anomaliekenngröße arbeitet analog wie die „Anzahl an Verbindungen eines spezifischen Serviceports“. Jedoch wird ausgeschlossen, dass ein Teilnehmer (IP-Adressen) mehrfach gezählt wird.

In bestimmten Situationen ist es vorteilhaft zusätzlich die Gesamtzahl einzigartiger Teilnehmer zu erfassen, um die gefundenen Anomalien besser bewerten zu können. Die Anomaliekenngröße „Anzahl der Verbindungen einzigartiger IP-Adressen eines spezifischen Serviceports“ ermöglicht dies, wogegen die anderen drei Anomaliekenngrößen diese Unterscheidung nicht tätigen. Die Information kann beispielsweise ein Indiz für eine „Flash Crowd“ sein und zu einer verbesserten Bewertung führen.

4. Verwendete Dienste

Bei diesen Anomaliekenngrößen werden die verwendeten Dienste der IP-Adresse untersucht und daher nur etablierte Verbindungen betrachtet. Der Dienst wird anhand des Ziel-Ports bestimmt und ist, weil die Ports beliebig festgelegt werden können, nicht immer zwangsläufig korrekt. Da häufig aber nur interessant ist, ob und welche Serviceports neu verwendet wurden, beschränkt sich das System auf folgende Anomaliekenngröße:

- **Verwendete Serviceports**

Diese Anomaliekenngröße bestimmt, welche Serviceports in einem Zeitraum verwendet wurden.

Diese Anomaliekenngröße eignet sich sehr gut um fremd-installierte Dienste, beispielsweise Backdoors, festzustellen und ist zudem ein sehr eindeutiges Merkmal, das wenig Nachbear-

beitung erfordert.

5. Verbindungsdauer

Die Dauer der Verbindungen zu einem Dienst ist ebenfalls ein interessantes Kriterium und kann auf Störungen des Netzwerkes oder auch des jeweiligen Rechners hindeuten. Längere Verbindungszeiten können mitunter durch hohe Auslastung des Rechners entstehen oder auch durch den Austausch des Dienstes auf dem Port. Üblicherweise entstehen Schwankungen allein schon aufgrund der unterschiedlichen Anfragen an den Dienst und des unterschiedlichen Aufwands bei der Beantwortung. Die Leitungsqualität des Teilnehmers ist ebenfalls von grundlegender Bedeutung. In der Regel sollten sich diese Anomaliekenngrößen jedoch unabhängig von dem Betrachtungszeitraum verhalten, da die Eigenschaften der Verbindungen im Allgemeinen relativ unabhängig von der Zeit sind. Die Anomaliekenngrößen sind stark vom Dienst des Ziel-Ports abhängig und müssen daher für die einzelnen Ports getrennt betrachtet werden. SNADS untersucht in diesem Kriterium nur folgende Anomaliegröße:

- **Verbindungsdauer eines spezifischen Serviceports**

Diese Anomaliekenngröße betrachtet die Verbindungsdauer zu einem zuvor festgelegten Serviceports. Es wird unterschieden zwischen ein- und ausgehenden Verbindungen.

Sicherlich gibt es noch eine Reihe weiterer Anomalienkenngrößen. In dieser Arbeit werden aber nur obige Anomaliekenngrößen betrachtet.

Besondere Eigenschaften der Anomaliekenngrößen

Die ersten drei Untersuchungskriterien sind wie beschrieben relativ stark von der Intensität der Verwendung abhängig. Es ist zu erwarten, dass die Datenpunkte der Anomaliekenngrößen, die auf vielen Verbindungen fussen, sich her Glätten als die, die auf weniger Verbindungen beruhen. Je nach dem Schwanken die Datenpunkte der Anomaliekenngrößen nicht so stark. Aufgrund dessen ist zu erwarten, dass das ADS umso bessere Ergebnisse liefert, je aktiver die jeweilige IP-Adresse ist [46]. Sehr problematisch sind Desktop Systeme, die sich in der Regel durch sehr sporadisches Verhalten auszeichnen. Die Schwankungen können allerdings minimiert werden, wenn der genannte Sachverhalt ausgenützt wird und längere Betrachtungszeiträume gewählt werden. Denn durch die Vergrößerung der Betrachtungszeiträume steigt die Anzahl der Verbindungen an und das Ergebnis ist besser geglättet. Die einzelnen Datenpunkte werden letztendlich durch diese Form der Aggregation robuster [43]. Dies hat aber auch mehrere Einschränkungen zur Folge. Einerseits wird der Rechenaufwand größer, andererseits sind die Ergebnisse durch die längeren Zeitfenster ungenauer. In einem gewissen Maße ist das sinnvoll, liegen jedoch so wenig Daten vor, dass sehr große Zeitfenster gewählt werden müssen, sollte vielleicht besser auf die spezifische Kontrolle der entsprechenden IP-Adresse

verzichtet werden. Zudem entfernt sich durch die Wahl längerer Betrachtungszeiträume der Zeitpunkt der Bewertung einer Anomaliekenngröße immer weiter von dem tatsächlichen Ereignis und kann daher zu erheblichen Verzögerungen in der Auswertung resultieren. SNADS bietet alternativ die Möglichkeit mehrere IP-Adressen in Gruppen zusammenzufassen und dadurch die Daten zusammengefasst als eine Anomaliekenngröße zu behandeln. Dies kann aufgrund einer größeren Zahl an Verbindungen die Schwankungen ebenfalls reduzieren. Allerdings sollte die Zuordnung einer IP-Adresse zu einer Gruppe sinnvoll sein, sodass wenn möglich nur Rechner gleichen Typs, beispielsweise Desktop Systeme, zusammengefasst werden. Es wird erwartet, dass unter Einbeziehung längerer Zeitfenster die Anomaliegrößen dem tatsächlichen Verlauf gut genug widerspiegeln, um mit Hilfe von robusten Statistiken zu guten Ergebnissen zu gelangen.

3.5.2 Anomaliebewertung

Für jedes der in Kapitel 3.5.1 vorgestellten Anomaliekenngrößen bestimmt SNADS einen Anomaliewert, der ein Maß für die Anomalität der Daten darstellt. Nur durch ein vergleichbares Maß lassen sich die einzelnen Anomalien untereinander vergleichen. Die Bewertung erfolgt mittels Heuristiken, da über die Qualität der Anomalien keine Aussage gemacht werden kann. Hierbei können vielfältige Algorithmen eingesetzt werden, eine Auswahl darunter ist in Kapitel 2.2 zu finden. Wie beschrieben gibt es in vielen Fällen keine optimale Wahl und auch wenn es sie gäbe, wäre sie oftmals äußerst schwer zu treffen. Einen Sonderfall stellt die Anomaliekenngröße „verwendete Serviceports“ dar, da hier lediglich zwei Listen verglichen werden müssen und deshalb kein kompliziertes Verfahren nötig ist.

Aufgrund der unbekanntenen Natur der zur Verfügung stehenden Daten, sollte sich das ADS auf robuste Verfahren beschränken. So könnte die Unabhängigkeit der Datenpunkte untereinander zwar angenommen werden, aber feststehen ist es nicht. Von einer Normalverteilung der Datensätze kann ebenfalls nicht ausgegangen werden. Allerdings gilt nach dem „Zentralen Grenzwertsatz“, dass durch Summenbildung einer genügend großen Anzahl identisch verteilter und unabhängiger Zufallsvariablen die resultierende Zufallsvariable einer Normalverteilung folgt. Mitunter ist deshalb die Länge des Betrachtungszeitraums bei der Anomaliebewertung von essentieller Bedeutung. Durch Anwendung robuster Statistiken sollte auch bei mangelhafter Erfüllung der Annahmen der Algorithmen das Ergebnis gut genug sein, um es zur Entscheidungsfindung heranziehen zu können. Details zu den robusten Statistiken sind in Kapitel 2.2.3 auf Seite 29 zu finden.

Die beschriebenen Anomaliekenngrößen können in zwei disjunkte Gruppen unterteilt werden. Einerseits gibt es Anomaliekenngrößen, bei denen in relativ kurzer Zeit sehr viele Datenpunkte vorliegen. In der Regel wird hierbei für *jede* Verbindung ein Datenpunkt extrahiert

und kann dann direkt mit anderen verglichen werden. Beispielhaft sind die Anomaliekenngrößen „Verbindungsdauer eines spezifischen Serviceports“ und „Anzahl von Verbindungen eines spezifischen Serviceports“ genannt. Andererseits gibt es Anomaliekenngrößen, die bereits auf aggregierten Datenpunkten basieren. Bei diesen repräsentiert ein Datenpunkt eine Mehrzahl von Verbindungen, die in einem vergangenen Zeitraum stattgefunden haben. Daher stehen dieser Gruppe in dem gleichen Betrachtungszeitraum wesentlich weniger Datenpunkte zur Verfügung. Beispielehaft sind die Anomaliekenngrößen „Transfermenge“, „Anzahl an Paketen eines spezifischen Serviceports“ und „Anzahl an Verbindungen“ genannt. In der ersten Gruppe fallen so viele Datenpunkte an, dass eine kompakte Datenspeicherung nötig ist. In der anderen können so wenig Datenpunkte zur Verfügung stehen, dass gegebenenfalls das tatsächliche Charakteristikum nur schlecht repräsentiert wird. Je nachdem sind auch die Anforderungen an die Heuristiken grundlegend verschieden. Letztendlich werden vier Algorithmen eingesetzt, die jeweils auf eine Teilmenge der Anomaliekenngrößen anwendbar sind.

Pearson Chi-square Anpassungstest

Betrachtet wird zunächst die erst genannte Gruppe für die viele Datenpunkte vorliegen. Ob viele Datenpunkte vorliegen, ist stark von der Aktivität der jeweiligen IP-Adresse abhängig. Ein mäßig ausgelasteter Webserver zum Beispiel kann mehrere Tausend Anfragen in der Stunde erzeugen, jedoch sind nach oben hin kaum Grenzen gesetzt. Das bekannte Wörterbuch LEO² läuft beispielsweise auf einem Webserver der TU-München und bearbeitet grob 2,2 Millionen Verbindungen am Tag. Da für jede Verbindung ein Datenpunkt der Anomaliekenngröße erzeugt wird und die Daten über einen längeren Zeitraum zur Verfügung stehen sollten, skaliert das System bei einfacher Datenspeicherung nicht ausreichend. Eine sehr gute Möglichkeit zur Reduzierung der Datenmenge ist die Kategorisierung. Hierzu wird der Wertebereich in Intervalle unterteilt und lediglich die Häufigkeit der Datenpunkte innerhalb der entsprechenden Intervalle festgehalten. Die Aufteilung lässt sich auch sehr gut mittels Histogrammen abbilden. Durch diese Form der Aggregation hält sich der Informationsverlust in Grenzen und fördert die Robustheit der Analyse [43]. Um die Verteilungen zweier Datensätze zu vergleichen eignen sich Anpassungstests sehr gut. Bei gegebener Häufigkeitsverteilung ist der „Pearson Chi-square Anpassungstest“ χ^2 das allgemein akzeptierte Prüfverfahren [47]. Auch in bereits bestehenden Intrusion Detection Systemen, wie beispielsweise NIDES, findet sich dieser Test in ähnlicher Form wieder [25, 48, 49]. Die Besonderheit dieses Hypothesentests ist, dass er mit verhältnismäßig wenig Annahmen auskommt. Gerade bezüglich der zugrunde liegenden Verteilungsfunktion, sprich dem zu erwartenden Charakteristikum, begnügt sich dieser nicht-parametrische Algorithmus mit wenig Informationen. Daher kann die Verteilung bei diesem Prüfverfahren beliebig aussehen; entscheidend ist nur, dass die empirischen

²<http://dict.leo.org>

Datensätze möglichst gut den erwarteten Verlauf folgen. Der parametrische „Kolmogorov-Smirnov-D Test“ verlangt im Gegensatz dazu eine voll spezifizierte Verteilungsfunktion, in der kein Parameter geschätzt sein darf. Da die wahre Natur der Daten aber unbekannt ist, lässt sich die Verteilungsfunktion jedoch nicht eindeutig festlegen. Der „Anderson-Darling Test“ geht noch einen Schritt weiter und verlangt, dass die Verteilungsfunktion einer bestimmten Verteilung entspricht. Da aber im Vorherrein in der Regel keine Aussagen über die tatsächliche Verteilung gemacht werden kann, lässt er sich ebenfalls nicht einsetzen. Alle drei Tests verlangen zudem noch die Unabhängigkeit der Datenpunkte. Ob diese Bedingung erfüllt ist, ist schwer nachzuweisen, zunächst wird davon aber ausgegangen. Der Pearson Chi-square Test ist zwar nicht ganz so mächtig wie die Anderen, aber dafür ist es wahrscheinlicher, dass die Voraussetzungen erfüllt sind. dafür lässt er sich ordnungsgemäß einsetzen.

Das grundlegende Problem bei dem χ^2 Test ist, dass er sehr stark von der Wahl der Intervalle abhängt. Um Fehler in der Berechnung zu reduzieren, sollten die Trefferwahrscheinlichkeit aller Intervalle möglichst gleich sein. Da eine optimale Intervalleinteilung aber unbekannt ist und sich mit der Zeit ändern kann, ist eine dynamische Anpassung der Grenzen in Abhängigkeit der Daten nötig. Eine zweckmäßige Methode um dies zu bewerkstelligen, ist die Intervallbreite am Anfang möglichst klein zu wählen und sie durch eine Häufigkeitsverteilung abzuspeichern. Bei der späteren Berechnung sind dann lediglich auf Basis dieser relativ feinen Abstufungen neue Intervallgrenzen zu wählen, sodass eben die Häufigkeit der Treffer in den Intervallen annähernd gleich sind. Die Methode erhöht zwar den Speicherbedarf, da wesentlich mehr Intervalle gespeichert werden müssen, jedoch sollte sich der Aufwand aufgrund besserer Ergebnisse lohnen.

Besonders schlecht für die Berechnung sind Intervalle in denen kein Treffer liegt oder die Anzahl der Treffer verhältnismäßig klein ist. Diese Intervalle sollten daher so gut es geht mit den benachbarten Intervallen zusammengefasst werden. Allerdings sollte, wenn möglich, die Anzahl an Intervalle nicht allzustark von der optimalen Anzahl M (vgl. Seite 25) abweichen, da auch in diesem Fall das Ergebnis negativ beeinflusst wird.

Damit das Ergebnis des Pearson Chi-square Tests näherungsweise der χ^2 -Verteilung folgt, ist eine ausreichende Anzahl von Datenpunkten erforderlich. Die Anzahl der Datenpunkte lässt sich in SNADS durch eine Vergrößerung des Betrachtungszeitraums problemlos vergrößern. Schwankungen innerhalb der Datensätze können auf diese Art auch geglättet werden. Auf das Problem des Informationsverlustes durch die Aggregation wurde bereits hingewiesen.

Sind diese Annahmen erfüllt, wird die Diskrepanz durch Vergleich des aktuellen Verlaufs mit dem zu erwartenden berechnet. Sind zu wenig Datenpunkte in den jeweiligen Verteilungen vorhanden, wird der Betrachtungszeitraum vergrößert. Das Ergebnis des Pearson Chi-square Tests folgt dann eher der χ^2 -Verteilung. Im Allgemeinen gilt, dass je größer der berechnete χ^2 -Wert ist, desto größer ist auch die Diskrepanz zwischen den zwei Verteilungen. Üblicherweise wird nun der Ablehnungsbereich anhand des vorgegebenen Signifikanzniveau und der

Freiheitsgrade ermittelt und geprüft, ob der χ^2 -Wert innerhalb liegt. Anstatt den Ablehnungsbereich anhand einer groben Tabelle zu prüfen, wird nun unter Verwendung des χ^2 -Werts und der Freiheitsgrade des Tests das Signifikanzniveau berechnet, unter dem die Nullhypothese gerade eben noch erfüllt ist. Das Signifikanzniveau α ist also ein direktes Maß für die Anomalität und es gilt, dass je kleiner es ist, umso unwahrscheinlicher ist es, dass sich die beiden Verteilungen ähneln. Letztendlich wird nun das berechnete Signifikanzniveau mit einer Prozentschranke verglichen. Sollte es kleiner als diese Schranke sein, ist die Annahme, dass die zwei Verteilungen gleich sind nicht erfüllt. Die Schranke ist die Wahrscheinlichkeit des Fehlers 1. Art, nämlich dass die zwei Verteilungen gleich sind, aber der Test es irrtümlich abgelehnt.

Da ein einfaches Maß für die Anomalität benötigt wird, wird zunächst davon ausgegangen, dass solange das Signifikanzniveau kleiner als 0,001 ist, mit Sicherheit von einer Anomalie ausgegangen werden kann. Die Schranke wird so klein gewählt, um auch wirklich nur gravierende Abweichungen zu melden. Aufgrund des berechneten Signifikanzniveaus können mehrere, beliebig feine Schranken verwendet werden, um eine Abstufung in Abhängigkeit der Schwere der Diskrepanz zu ermöglichen. Es wird folgendes Schema benutzt:

Signifikanzniveau	Anomaliewert
$\alpha < 0,001$	1,0
$\alpha < 0,01$	0,7
$\alpha < 0,03$	0,5
$\alpha < 0,05$	0,3
$\alpha \geq 0,05$	0,0

Das Maß der Anomalität wird im Folgenden Anomaliewert genannt und ist definiert für den Wertebereich zwischen $[-1; +1]$. Ein Wert von 0 bedeutet, dass keine Anomalie vermutet wird. Zwischen 0 und ± 1 steigt die Anomaliewahrscheinlichkeit stetig an. An den Rändern ± 1 ist höchstwahrscheinlich mit einer Anomalie zu rechnen. Das Vorzeichen wird in diesem Fall nicht benötigt, weshalb der Anomaliewert nur positive Werte annehmen kann. Der Wert spiegelt also den „Anomaliegehalt“ eines Datensatzes wieder und ist untereinander vergleichbar. Es ergeben sich folgende Abhängigkeiten, die zur Übersichtlichkeit nochmal dargestellt werden:

Je größer die Diskrepanz, umso größer ist der berechnete χ^2 -Wert.
 Je größer der berechnete χ^2 -Wert, umso kleiner wird das Signifikanzniveau.
 Je kleiner das Signifikanzniveau, umso wahrscheinlicher ist eine Anomalie.
 Je wahrscheinlicher eine Anomalie, umso größer ist der Anomaliewert.

Standardized residual

Die zweite Gruppe der Anomaliekenngößen hat, wie bereits erwähnt, wesentlich weniger Datenpunkte für die Bewertung zur Verfügung. In der Regel gilt, dass für jeden Betrachtungszeitraum genau ein Datenpunkt gebildet wird, der sich aus der Menge an Verbindungen innerhalb des Betrachtungszeitraums ergibt. Letztendlich wird ein Algorithmus gesucht, der einen Datenpunkt mit einer relativ kleinen Menge von bereits berechneten Datenpunkten, die von anderen Betrachtungszeiträumen herrühren, vergleicht und seine Anomalität bewertet. Beispielsweise könnte es sinnvoll sein, den Datenpunkt der Anomaliekenngöße „Transfermenge“ eines Samstags mit den vergangenen Datenpunkten der Samstage der Vorwochen zu vergleichen. Da aber aufgrund begrenzter Rechenressourcen nur endlich viele Datensätze der Vergangenheit gespeichert sein können, ist die Anzahl der Datenpunkte begrenzt. In Abhängigkeit der Vergleiche sollte sichergestellt werden, dass stets genügend Datensätze aus der Vergangenheit zur Verfügung stehen. Es gilt hierbei natürlich, dass die Genauigkeit des Ergebnisses direkt von der Anzahl der Datenpunkte und deren Qualität abhängt. Eine große Anzahl von Datenpunkten kann den tatsächlichen Verlauf der Anomaliekenngöße besser darstellen als wenige. So kann beispielsweise eine Mittelwertbildung von nur drei Datenpunkten den tatsächlichen Verlauf der zugrunde liegenden Verteilung nur sehr schlecht abbilden. Die Qualität der Daten ist wiederum von dem Verhalten der IP-Adresse abhängig. Es gilt, dass die Schwankungen in der Auslastung der IP-Adresse die Anomaliekenngößen stark beeinflussen können. Daher kann es erforderlich sein, die Schwankungen durch Vergrößerung des Betrachtungszeitraums zu minimieren. Aufgrund dieser Unsicherheiten sollten auch hier robuste Statistiken bevorzugt eingesetzt und die Parameter entsprechend angepasst werden.

Üblichen Tests, die prüfen, wie wahrscheinlich ein Datenpunkt Teil eines bestimmten Datensatzes ist, sind Ausreißertests (vgl. Seite 24). Die eingesetzten Algorithmen sollten sowohl das Lagemaß als auch das Streumaß mitbeachten, um einen möglichst umfassenden Einblick in die Natur der Daten zu erhalten. Der „Grubbs Test“ ist so ein Verfahren. Er bestimmt wie weit der zu untersuchende Datenpunkt vom Mittelwert der Datenreihe entfernt ist. Hierbei wird als Maß die Standardabweichung verwendet. Zudem existiert auch eine kritische Tabelle, die den Ablehnungsbereich des Verfahrens vorgibt und daher eine genaue Bewertung ermöglicht. Leider basiert der Test auf dem Mittelwert und der Standardabweichung und ist daher wie in Kapitel 2.2.3 auf Seite 29 beschrieben stark von Ausreißern beeinflussbar und daher nicht robust. Vorteilhaft ist, dass bei ihm sowohl Lagemaße als auch Streumaße berücksichtigt werden. Da die Verteilung der Daten jedoch unbekannt ist und zudem auch keine Aussage über die Qualität der Datensätze gemacht werden können, wird der Test modifiziert, sodass robuste Maße verwendet werden. Anstatt dem Mittelwert wird der Median und anstatt der Standardabweichung der MAD eingesetzt. Das resultierende Ergebnis dieses Tests wird als „*standardized residual*“ oder „*Z-Score*“ [50] bezeichnet. Aufgrund dieser Änderungen gilt natürlich die kriti-

sche Tabelle des Grubbs Tests nicht mehr und es müssen eigene Ablehnungsbereiche definiert werden. Der robuste Z-Score Algorithmus prüft ebenso wie der Grubbs Test, inwiefern der Datenpunkt Y auf Basis des Datensatzes X einen Ausreißer darstellt und ist folgendermaßen definiert:

$$Z = \frac{|Y - \text{MEDIAN}(X)|}{\text{MAD}(X)}$$

Der Z-Score ist äußerst robust, und funktioniert auch gut bei schwankenden Datensätzen, solange mindestens 50% der Datenpunkte den tatsächlichen Verlauf gut nachbilden, denn die bei der Berechnung verwendeten Größen MAD und Median werden direkt aus diesen Werten gebildet. Liegen viele Ausreißer vor, sollte aber die Versuchsanordnung oder die Qualität der Datenpunkte in Frage gestellt werden. Zur Visualisierung der Punkte eignen sich Boxplots sehr gut und können daher zur nachträglichen Validierung benutzt werden. Die Abweichung Z ist ein direktes Maß für die Anomalität des Datenpunktes und lässt sich sehr gut mittels Schranken prüfen. Üblicherweise wird ein Punkt als Ausreißer deklariert, wenn dieser grob $Z \geq 3$ MADs vom Median entfernt ist. Es ist aber prinzipiell sinnvoll feinere Abstufungen zu ermöglichen, um die Ernsthaftigkeit des Ausreißers exakter bestimmen zu können [43]. Bei normalverteilten Datensätzen befinden sich innerhalb des ersten MADs vom Median 75% der Datenpunkte. Innerhalb der ersten zwei MADs 91% und innerhalb der ersten drei MADs 97% der Daten. Aufgrund dieser Fakten werden Datenpunkte innerhalb der ersten zwei MADs als „normal“ eingestuft. Die verbleibenden 9% der Datenpunkte werden letztendlich nach dem folgendem Schema dem Anomaliewert zugewiesen.

Z	Anomaliewert
$Z > 5$	1,0
$Z > 3$	0,7
$Z > 2,5$	0,5
$Z > 2$	0,3
$-2 \geq Z \leq 2$	0,0
$Z > -2$	-0,3
$Z > -2,5$	-0,5
$Z > -3$	-0,7
$Z > -5$	-1,0

Die Abbildung 3.3 veranschaulicht diesen Sachverhalt anhand eines Boxplots. Das Schema lässt sich natürlich jederzeit verfeinern. Da die Annahme über die Normalverteilung nicht gesichert ist und die Datensätze sehr klein sein können, sind größere Schwankungen bei den Ergebnissen nicht auszuschließen. Daher kann es sinnvoll sein, die beschriebenen Schranken nach oben hin anzupassen. Da Anomaliekenngrößen aus vielfältigsten Gründen beeinflusst

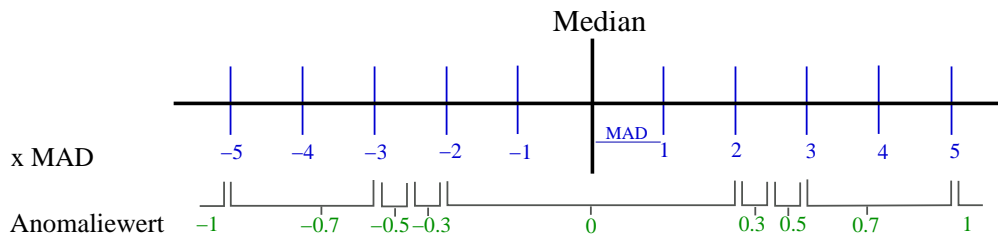


Abbildung 3.3: Veranschaulichung des standardized residuals als Boxplot

werden können, ist zu erwarten, dass in Abhängigkeit der IP-Adresse individuelle Schemas mit unterschiedlichen Schranken eingesetzt werden müssen.

Aufgrund der beschriebenen Probleme durch zu kleine Datensätze und ihre potentiell mangelhafte Qualität, ist es zweckmäßig eine Mindestanzahl an Datenpunkten bei der Berechnung vorzugeben. Das System setzt mindestens zehn Datenpunkte voraus bevor die Berechnung getätigt wird. Gegebenfalls ist diese Anzahl für die einzelnen Anomaliekenngrößen getrennt festzulegen. Durch diese Vorgabe wird auch die zeitliche Dauer der Lernphase festgelegt, also die zeitliche Dauer in der SNADS noch keine Auswertungen tätigt. Anzumerken ist, dass die Lernphase unter SNADS nicht einmalig erfolgt, sondern aktuelle Datensätze kontinuierlich abgespeichert und dann bei späteren Vergleichen auch herangezogen werden.

Durch das kontinuierliche Lernen finden sich in den Vergangenheitswerten Anomalien, sprich Ausreißer wieder. Das automatische Bereinigen von Ausreißern ist denkbar schwierig, da dem System zur Bewertung lediglich statistische Methoden zur Verfügung stehen. Nur aufgrund von statistischen Methoden sollte aber kein Ausreißer gelöscht werden, da dadurch auch ungewollt legitime Datenpunkte gelöscht werden könnten [39]. Die Verteilung der Datensätze könnte in diesem Fall der wahren Verteilung nicht folgen. Üblicherweise ist eine Löschung des Datenpunktes nur dann gerechtfertigt, wenn die Ursache des Ausreißers bekannt ist oder persönliche Erfahrungswerte dafür sprechen. Dies ist in der Regel nicht gegeben, weshalb die als Ausreißer deklarierten Datenpunkte dennoch für spätere Berechnungen herangezogen werden. Aufgrund des Sachverhalts wird die Entscheidung robuste Statistiken einzusetzen gestärkt. Bei der Auswahl der Algorithmen muss jedoch darauf geachtet werden, solche auszuwählen, die auch mit möglichst vielen Ausreißern klarkommen. Das Z-Score Prüfverfahren ist aufgrund der Verwendung der MAD nicht vor Ausreißern anfällig und ist daher auch in diesem Zusammenhang zu empfehlen.

Der Vollständigkeit halber wird erwähnt, dass Anpassungstests für diese Gruppe der Anomaliekenngrößen nicht verwendet werden können. Denn der Pearson Chi-square Hypothesentest verlangt relativ große Datensätze zur Berechnung und bei den parametrischen Anpassungstests sollte die zugrunde liegende Verteilungsfunktion bekannt sein. Beide Bedingungen sind nur mangelhaft erfüllt.

Schwellwertrechnung

Für die auf Seite 36 beschriebenen Regeln werden lediglich einfache Verfahren benötigt. Die Verfahren werden in zwei Arten unterteilt. Zunächst gibt es Regeln, die Schwellwerte vorgeben. Bei ihnen wird auf einfache Art und Weise geprüft, ob ein Datenpunkt den Schwellwert über- oder unterschreitet. Optional können auch mehrere Datenpunkte durch den Schwellwert getestet werden. In diesem Fall kann noch ein Faktor mitangegeben werden, der angibt wie oft der Schwellwert missachtet werden darf. Wird die vorgegebene Regel letztendlich gebrochen, wird direkt eine Anomalie mit einem Anomaliewert von 1 ausgerufen. Die zweite Art der Verfahren arbeitet ähnlich zu den bereits genannten, jedoch wird nicht ein Schwellwert geprüft, sondern sichergestellt, dass die Datenpunkte innerhalb einer prozentuale Abweichung um einen vorgegebenen Wert liegen. Die genannten Verfahren basieren auf fest vorgegebenen Schranken und benötigen daher im Gegensatz zu den bereits beschriebenen Algorithmen keine Feinabstufung für die Schwere der Anomalie. Wird zu oft fälschlicherweise eine Anomalie erkennen, sollten die Schranken der Regel angepasst werden.

Vergleich zweier Zahlenreihen

Zur Bewertung der Anomaliekenngröße „verwendeten Serviceports“ wird eine Methode benötigt, die lediglich auf einfache Art und Weise zwei Listen, bestehend aus numerischen Werten der Serviceports, miteinander vergleicht. Jede Liste besteht aus den verwendeten Diensten eines Zeitraumes. Die Methode kann hierbei entweder neue verwendete Dienste, nicht mehr verwendete Dienste oder beides untersuchen. Bei Prüfung neuer verwendete Dienste ist der Zeitraum, der das vergangene Nutzerverhalten beschreibt, nicht zu klein zu wählen. Unregelmäßig genutzte Dienste führen dann nicht zu einer mehrmaligen Meldung eines Alarms. Beispielsweise ist es sinnvoll bei einer stündlichen Prüfung der Dienste, die eine Liste zeitlich auf die derzeit letzte Stunde zu begrenzen und die andere auf die entsprechenden 24 Stunden zuvor. Wurde ein Verstoß entdeckt, wird eine Anomalie mit einem Anomaliewert von 1 gemeldet. Auf eine feine Abstufung des Anomaliewerts wird vorerst verzichtet, kann aber bei unregelmäßiger Nutzung von Diensten vorteilhaft sein.

3.5.3 Eskalationsbewertung

Wie bereits erklärt werden Heuristiken eingesetzt, um den Anomaliewert, also das Maß der Diskrepanz zwischen Datensätzen, zu bewerten. Die Anomaliebewertung kann daher fehlerbehaftet sein. Um die Anzahl an Fehlalarmen einzuschränken und dem Administrator die Arbeit zu erleichtern, wird eine automatische Selektion auf den berechneten Anomalien durchgeführt, sodass dem Administrator nach individuell definierten Parametern nur noch gravierende Anomalien vorgelegt werden. Der Vorgang in dem SNADS dem Administrator aus einer Reihe

von gefundenen Anomalien die gravierenden selektiert und die Ursache der Anomalie näher bestimmt, wird Eskalationsbewertung genannt. Das Ergebnis wird folglich als Eskalation betitelt.

Die Eskalationsbewertung korreliert hierbei den Anomaliewert eines konkreten Zeitraums mit den zeitlich umliegenden Anomaliewerten der selben Anomaliekenngröße. Sie ist hierbei an dem menschlich intuitiven Vorgehen zur Bewertung der Diskrepanz zwischen zwei Kurven angelehnt. Zunächst sucht man nach offensichtlich gravierenden Abweichungen. Findet man nichts, wird nach kleineren Abweichungen geprüft, die dann aber langanhaltender sein müssen. Schritt für Schritt wird nun der Betrachtungszeitraum vergrößert und gleichzeitig die für eine Eskalation erforderliche Abweichung reduziert. Die Abweichungen können sukzessive verkleinert werden, da im Allgemeinen lang anhaltende Abweichungen eher auf eine gravierende Anomalie hindeuten.

SNADS berechnet bei der Eskalationsbewertung die Diskrepanz allerdings nicht anhand von Kurven, sondern verwendet hierzu die zuvor berechneten Anomaliewerte. Die Anomaliewerte werden, wie in Kapitel 3.5.2 auf Seite 47 beschrieben, bei dem Vorgang der Anomaliebewertung berechnet und drücken eben diese Diskrepanz aus. Dies minimiert den Rechenaufwand und den Bedarf an Speicherressourcen enorm. Aufgrund dessen wird beim sukzessiven Vergrößern der Zeitfenster kein neuer repräsentativer Anomaliewert berechnet, sondern die bereits vorhandenen Anomaliewerte, die das Zeitfenster umfassen, herangezogen. Zur Bewertung der Abweichungen werden einfache Schranken verwendet, die sich nach der Länge des Betrachtungszeitraums richten. Bei jedem Schritt wird also überprüft, ob die einzelnen Anomaliewerte allesamt die Schranke durchbrechen. Ist dies nicht für jeden Anomaliewert des jeweiligen Betrachtungszeitraums der Fall, wird für diesen Schritt nicht von einer Eskalation ausgegangen.

In der Praxis lässt sich der Zusammenhang gut in einer zweidimensionalen Grafik darstellen, in dem die X-Achse die Zeitachse repräsentiert und die Y-Achse, die Anomaliewerte darstellt. Veranschaulicht wird diese Prozedur anhand Abbildung 3.4. Wie zu erkennen ist, reduziert sich bei jedem Schritt die für eine Eskalation erforderliche Schranke. Der Betrachtungszeitraum wird schrittweise vergrößert bis letztendlich im dritten Schritt eine Eskalation erkannt wird.

Bei der Eskalationsbewertung ist lediglich der kleinste Anomaliewert und die Länge des Betrachtungszeitraums relevant, da nur die beiden Größen ein Maß für die Schwere der Eskalation darstellen. Durch die Korrelation an der Zeitachse wird versucht langfristige Anomalien zu erkennen. Hierzu ist der „normalste Datenpunkt“ üblicherweise der wichtigste, da er gravierende Eskalationen untereinander zeitlich trennt. Aufgrund der Bedeutung des „normalsten Datenpunktes“ werden die einzelnen Anomaliewerte auch nicht einfach addiert und mit einer Schranke verglichen.

Die verwendete Schranke ist wie beschrieben von der Länge des Betrachtungszeitraums ab-

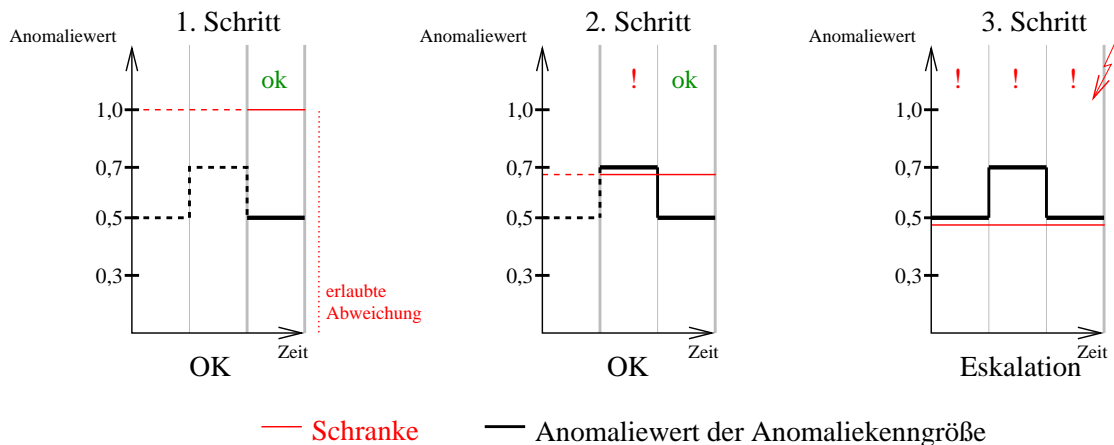


Abbildung 3.4: Prinzip der Eskalationsbewertung

hängig, kann aber zudem durch übergebene Parameter beeinflusst werden. Die Eskalationsregeln beschreiben hierbei – analog wie die Regeln der Anomaliebewertung – detailliert, was wie zu untersuchen ist und ermöglichen durch individuelle Einstellung der Parameter bessere Ergebnisse. Beispielsweise könnte es sinnvoll sein bei IP-Adressen, die sehr schwankendes Verhalten vorweisen, größere Schranken zu definieren. Aufgrund der Definition der Anomaliewerte wird ein einzelner Anomaliewert von ± 1 direkt eskaliert. Falls nicht anders definiert, wird ansonsten folgendes Schema für die Schranken verwendet.

Zeitfenster	Schwellwert für eine Eskalation
1 faches	$ Anomaliewert == 1,0$
2 faches	$ Anomaliewert \geq 0,7$
3 faches	$ Anomaliewert \geq 0,5$
4 faches	$ Anomaliewert \geq 0,3$

Kann durch eine weitere Vergrößerung des Zeitfensters einer bereits gemeldeten Eskalation die Schwere des Vorfalls gesteigert werden, so wird es auch getan. Dadurch umfassen die Eskalationen stets das größt mögliche Zeitfenster. Damit der Administrator die Eskalationen auf einen Blick leichter bewerten kann, wird aber zudem die einfache Summe der Anomaliewerte des Zeitraums zur Verfügung gestellt.

Das Ziel ist es, in Abhängigkeit von den Parametern der Heuristiken aus einer Menge von Anomalien nur noch die Spitze des Eisberges zu sehen. Die Spitze besteht, wie bereits erklärt, aus den gravierenderen Anomalien und ist weniger fehlerträchtig.

Das Prinzip der Eskalationsbewertung kann nun für jegliche in Kapitel 3.5.1 definierten Anomaliekenngrößen verwendet werden und ist daher universell sinnvoll einsetzbar. Die nachträgliche Analyse der Eskalationen durch den Administrator ist jedoch oftmals aufwendig und kann noch zusätzlich vereinfacht werden. Hierzu werden ein paar typische Ursachen von An-

omalien definiert, die sich relativ leicht erkennen lassen. Prinzipiell gilt, dass die Intrusion Detection Systeme die wahren Ursachen der Anomalien aufgrund der genannten Gründen nicht kennen können und daher ist die Ursachenforschung mit Skepsis zu betrachten. Eine eindeutige und korrekte Bewertung kann nur in seltenen Fällen vorgenommen werden. Folgende relativ primitive Eskalationsursachen werden automatisch von SNADS erkannt:

- **Port Scans**

Die „Port Scans“ werden von Angreifern eingesetzt, um sich ein Bild der laufenden Dienste einer IP-Adresse zu machen und kündigen daher oftmals einen Angriff an. Die gesuchte Information ist deshalb wertvoll, da jeder Dienst, der über Rechnergrenze hinweg erreichbar ist, potentiell ein Sicherheitsrisiko darstellt und angegriffen werden kann.

- **Denial of Service (DOS)**

Die DOS Attacke wurde bereits in Kapitel 2.1 auf Seite 5 vorgestellt. Dieser Angriff kann in zwei Fälle unterteilt werden. Einerseits kann der Angriff alle Serviceports der IP-Adresse betreffen, oder andererseits ist es auch möglich, dass nur einen einzelnen Serviceport attackiert wird. Das Ziel ist es den Rechner, einen spezifischen Serviceport oder auch die Datenleitung zu beeinträchtigen, um die Funktionalität eines Dienstes zu reduzieren.

- **Umfangreiche Nutzung**

Diese Eskalationsursache ist relativ allgemein gehalten und deutet daher lediglich auf Änderungen im Nutzerverhalten einer IP-Adresse oder Dienstes hin. Erkannt werden können beispielsweise Dienste, die plötzlich unüblich viel Daten transferieren.

Flash Crowds

Die Eskalationsursache ist nur eine besondere Ausprägung der „Umfangreichen Nutzung“ und zeichnet sich durch einen schnellen Anstieg des transferierten Datenvolumens eines Dienstes aus, der dann langsam aber stetig wieder auf Normalwert abfällt. Charakteristisch ist die sehr große Anzahl an verschiedenen IP-Adressen (Teilnehmern), die bei diesem Vorgang beteiligt sind. Die Ursache einer „Flash Crowd“ kann beispielsweise ein Veröffentlichung eines sehr gefragten Softwarepakets sein.

- **Neue verwendete Dienste**

Dieses Ereignis ist bereits in Kapitel 3.5.1 auf Seite 45 beschrieben und eignet sich beispielsweise sehr gut um Backdoors oder Würmer zu erkennen.

Wirklich sinnvoll ist diese weitere Hilfestellung jedoch nur, wenn die Eskalationsursachen eindeutig einem tatsächlichen Sachverhalt zugewiesen werden können. Eine exakte Abgrenzung ist aufgrund der unbekanntenen Ursache der Angriffe nicht fehlerfrei, weshalb nur in recht allgemeine Eskalationsursachen unterschieden wird. Die Vielfältigkeit der Angriffe erschwert

die Zuweisung zudem. Das Ziel ist eine möglichst exakte Erkennung und Abgrenzung der Eskalationsursachen unter Verwendung von möglichst wenig Anomaliekenngrößen.

Da die *Port Scans* sämtliche Serviceports der IP-Adresse prüfen und üblicherweise die absolute Mehrheit der Ports der IP-Adresse nicht verwendet werden, ist von einer starken Zunahme der Anzahl an nicht etablierten Verbindungen³ auszugehen. Auffällig ist, dass bei diesen Verbindungen der jeweilige Ziel-Port sich stets ändern sollte. Die zusätzliche Untersuchung der Verbindungen ist in der Praxis jedoch sehr aufwendig, wäre aber zur Abgrenzung von den DOS Attacken sinnvoll.

Der *Denial of Service* ist wie beschrieben in mehrere Fällen unterscheidbar. Für den Fall, dass der Angriff auf alle Serviceports der IP-Adresse stattfindet, ist eine Abgrenzung von den „Port Scans“ nur schwer möglich. Denn der einzige Unterschied besteht in der absoluten „Anzahl an nicht etablierten Verbindungen“. Ansonsten ist die Ausprägung identisch. Aufgrund dessen wird ein konfigurierbarer Schwellwert definiert, sodass wenn die „Anzahl nicht etablierter Verbindungen“ innerhalb einer Minute unterhalb von 5000 liegt, ein Port Scan gemeldet wird und ansonsten ein DOS vorliegt. Da eine DOS Attacke in der Regel nur bei einer außerordentlichen Flut an Verbindungen ihre Wirkung zeigt, kann dieser Schwellwert sehr hoch gelegt werden. Bei anderen Anomaliekenngrößen, die beispielsweise auf der Paketrate beruhen, müsste ebenfalls einen Schwellwert eingeführt werden, weshalb bei der Untersuchung der DOS Attacke lediglich die eine Anomaliekenngröße betrachtet wird. Bei dem zweiten Fall einer DOS Attacke, bei dem nur ein spezifischer Serviceport bombardiert wird, muss zudem unterschieden werden, ob auf dem Port ein Dienst arbeitet und die Verbindung durch den Three-Way-Handshake korrekt initiiert wurde. Wird keine korrekte Verbindung initiiert, könnte als eindeutiges Kriterium für die Abgrenzung zu den „Port Scans“ die Anzahl verschiedener Serviceports der Verbindungen herangezogen werden. Wie bereits erwähnt ist diese Information nur unter erheblichen Aufwand zu beschaffen, weshalb SNADS in dem Fall ebenfalls die oben genannte Schranke von 5000 Verbindungen pro Minute verwendet und nur bei Verstoß einen DOS meldet. Ist jedoch die Verbindung korrekt initiiert worden, ändert sich die Anomaliekenngröße „Anzahl an nicht etablierten Verbindungen“ nicht, stattdessen steigt die „Anzahl an etablierten Verbindungen“ an. Diese Anzahl muss nicht zwangsläufig auf eine DOS Attacke hindeutet, da eine „Umfangreiche Nutzung“ der IP-Adresse den gleichen Effekt bewirken kann. Daher wird die Klassifizierung mit Hilfe der Anomaliekenngröße „Transfermenge“ durchgeführt. Falls die Transfermenge in dem gleichen Zeitraum unverändert blieb, wird eine DOS Attacke und ansonsten eine „Umfangreiche Nutzung“ vermutet. In jedem Fall wird aber stets nur ein Vorfall gemeldet. Problematisch sind die DOS Attacken, die auf großen Pakete beruhen. Denn sie werden in diesem Fall einer falschen Ursache zugewiesen. Aufgrund mangelhafter Informationen ist eine korrekte Unterscheidung aber sehr aufwendig. Die Eskalationsursache *umfangreiche Nutzung* wird anhand der „Transfermenge“ untersucht

³diese Größe fällt unter die Anomaliekenngröße „Verbindungen mit besonderen Zuständen“

und ist ein relativ eindeutiges Kriterium. Da BRO die Transfermenge nur bei etablierten Verbindungen zur Verfügung stellt, kann die Eskalationsursache leicht von den „Port Scans“ und DOS Attacken abgegrenzt werden. Mit Hilfe der Anomaliekenngröße „Anzahl der Verbindungen einzigartiger IP-Adressen“ lässt sie sich auch von den *Flash Crowds* unterscheiden. Wurden außerordentlich viele verschiedene IP-Adressen gemessen, so wird ein „Flash Crowd“ gemeldet, ansonsten bleibt es bei der „Umfangreichen Nutzung“.

Die Eskalationsursache *Neue verwendete Dienste* verwendet die Anomaliekenngröße „Verwendete Serviceports“. Ursprünglich umfasst die Anomaliekenngröße beide Flussrichtungen und zeigt somit alle bei Fremdrechnern angeforderten Dienste und auch die lokal verwendeten Dienste an. Dies ist bei Server Systemen sinnvoll, da ein umfassender Blick auf die verwendeten Dienste ermöglicht wird. Bei Desktop Systemen kann es aufgrund des üblicherweise sehr sporadischen Anwendungsverhalten zweckmäßiger sein, nur die eingehende Flussrichtung zu kontrollieren.

Nun ist es aber immer noch möglich, dass ein Angriff mehrere Eskalationen hervorruft. Wird beispielsweise ein neuer Dienst gestartet und über ihn dann Unmengen an Daten transferiert, dann wird sowohl die Eskalationsursache „Neue verwendete Dienste“ als auch „Umfangreiche Nutzung“ gemeldet. Um dies zu verhindern, wird nach der Eskalationsbewertung geprüft, ob eine gemeinsame Ursache zugrunde liegt, und wenn mit hoher Wahrscheinlichkeit davon auszugehen ist, werden die jeweiligen Eskalationen dann zusammengefasst. Ein zusätzlicher Informationsverlust ist nicht erwünscht, sodass es an dieser Stelle schon vorkommen kann, dass mehrere Eskalationen mit gleicher Ursache gemeldet werden. Bei der Bewertung der Zusammenhänge ist zunächst strikt zwischen den verschiedenen Anomaliekenngrößen zu unterscheiden. Es gibt Serviceport- und Host-abhängige Größen, sodass es sinnvoll ist nur gleiche Typen zusammenzufassen. Wird die Eskalationsursache „Umfangreiche Nutzung“ sowohl für die gesamte IP-Adresse als auch für einen spezifischen Serviceport erkannt, so muss nicht zwangsläufig der Anstieg der Transfermenge an dem Port alleine für den Anstieg der gesamten IP-Adresse verantwortlich sein. Weitere Serviceports können ebenfalls beteiligt sein, wurden aber vielleicht aufgrund des Aufwands bei der Datenerfassung gar nicht erst kontrolliert. Andersrum gilt dies natürlich auch, da bei Eskalation eines spezifischen Serviceports nicht unbedingt auch auf der gesamten IP-Adresse eine Eskalation festgestellt sein muss, denn der zusätzliche Datenverkehr kann möglicherweise einfach nur in der Gesamtmenge der Daten untergehen. Aufgrund solcher Überlegung sollten nur Eskalationen gleicher Anomalietypen zusammengefasst werden. Problematisch sind die verschiedenen Betrachtungszeiträume der Eskalationen. Prinzipiell sollten nur Eskalationen identischer Zeiträume zusammengefasst werden. Eine Zuweisung mehrerer Eskalationen zu einer gemeinsamen Ursache ist ohnehin schon aufgrund der Verzögerungen in der Auswertung fehlerbehaftet. Bei lang anhaltenden

3 Entwurf eines Anomalie-basierten Intrusion Detection Systems

Anomalien kann der Fehler vernachlässigt werden, da die Dauer der Überschneidung zwischen den Eskalationen größer werden sollte. Daher könnte auch hier eine Zusammenfassung erfolgen, wenn ein Großteil der Zeiträume der Eskalationen sich überlappen. Eine Überblick über die Entscheidungsfindung bei der Eskalationsbewertung und der nachträglichen Zusammenfassung ist in Abbildung 3.5 zu sehen. Aufgrund dass die Eskalationsursachen „Neue

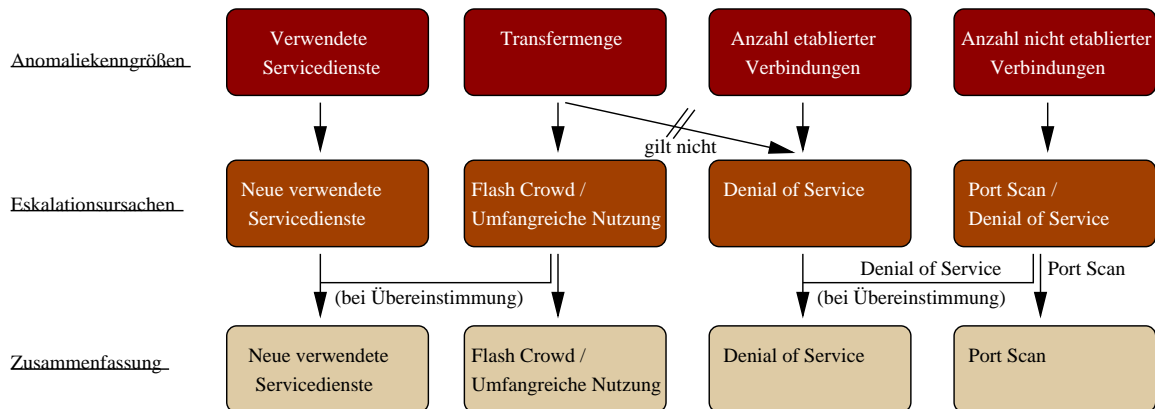


Abbildung 3.5: Die Entscheidungsfindung bei der Eskalation

verwendete Dienste“ und „Denial of Service“ nicht unmittelbar in Bezug zueinander stehen, vereinfacht sich das Schaubild bei der Zusammenfassung von Eskalationen erheblich. Kompliziertere Logiken, beispielsweise Data Mining Techniken, könnten bei der Entscheidungsfindung und der nachträglichen Zusammenfassung noch bessere Ergebnisse liefern. Jedoch ist aufgrund sehr begrenzter Informationen auch hier mit Schwierigkeiten zu rechnen.

4 Realisierung

Nachdem das Konzept des Anomalie-basierten Intrusion Detection Systems SNADS in Kapitel 3 dargelegt wurde, werden nun einige Details der Realisierung beschrieben. Da für das Projekt nur begrenzt viel Zeit zur Verfügung steht, wurden lediglich die wesentlichen Komponenten umgesetzt. Der Prototyp ist jedoch in diesen Teilen voll funktionsfähig und veranschaulicht das vorgestellte Konzept. Deshalb betrachtet der Prototyp SNADS zunächst nur das TCP-Protokoll und die zwei wichtigen Eskalationsursachen „Umfangreiche Nutzung“, „Neue verwendete Dienste“, sowie das Ereignis „Schwankungen bei der Verbindungsdauer“. Die hierzu nötigen Anomaliekenngrößen wurden ebenfalls realisiert. Zunächst wird ein Überblick über den strukturellen Aufbau des Systems geliefert und danach Details der Implementation erläutert.

4.1 Struktureller Aufbau

Der Aufwand bei der Anomaliebestimmung ist stark von den Regeln und Profilen, aber auch von der zu untersuchenden Datenmenge abhängig. Daher sollte die eingesetzte Hardware möglichst optimal ausgenutzt werden. Die Trennung der zeitkritischen Datenbeschaffung von den anderen Tätigkeiten erlaubt uns Multiprozessor-Architekturen zu nutzen. Hierzu werden die Tätigkeiten auf mehrere eigenständige Prozesse aufgeteilt. Die Speicherung der vielfältigen und großen Datenmengen ist mit Hilfe von Datenbanken realisiert. Datenbanken sind im Allgemeinen leicht einzusetzen und ermöglichen einen effektiven Zugriff. Es wurden folgende drei Prozesse definiert:

- **inputSNADS**

Dieser Prozess ist für die Datenbeschaffung zuständig. Er nimmt Verbindungen von BRO zunächst entgegen. Hierbei ist der Aufwand bei der Bearbeitung abhängig von der zu untersuchenden Datenmenge. Sollte der Prozess die Daten nicht rechtzeitig verarbeiten können, ist mit gravierenden Verzögerungen in dem gesamten System zu rechnen, da sich letztendlich alle weiteren Schritte hinauszögern. Datenverlust tritt in diesem Fall jedoch nicht auf, da durch aufwendige Synchronisation zwischen den Prozessen die weiteren Arbeitsschritte losgelöst sind von der Echtzeit. Da BRO die Verbindungssätze in

persistenten Dateien ablegt, können hier Informationen auch nicht verloren werden.

Für jede Anomaliegröße extrahiert der Prozess Daten aus den Verbindungen. Hier wird unnötige Arbeit vermieden, indem die definierten Regeln und Profile vorgeben, welche Daten überhaupt benötigt werden. Alle anderen Informationen werden einfach verworfen. Preprozessor-Regeln werden hier geprüft und können gegebenenfalls direkt zu einer Eskalation führen. Hierzu benötigt der Prozess Zugriff auf die Datenbank, in der die Daten abgelegt werden.

Im letzten Schritt des Prozesses werden die extrahierten Werte zusammengefasst und als Datenpunkte der Anomaliekenngroße in der Datenbank abgespeichert. Für jede Anomaliekenngroße wird eine eigene Tabelle verwendet. Die Datenpunkte werden hierbei nach dem in den Regeln und Profilen vorgegebenen *initialen Betrachtungszeitraum* abgelegt. Für die Aggregation in größere Zeiträume ist der Prozess nicht zuständig.

- **logicSNADS**

Dieser Prozess übernimmt die Aufgaben bis hin zur Eskalation. Er arbeitet ausschließlich auf den in der Datenbank abgelegten Anomaliekenngroßen und besitzt keine direkte Möglichkeit, die ursprünglichen Datensätze einzusehen. Einerseits trägt das zu einer klaren Struktur des Systems bei, andererseits reduziert sich dadurch die Möglichkeiten bei der Auswertung. Ausschlaggebend ist jedoch, dass ein nachträglicher Einblick in die potentiell sehr große Datenmenge sehr aufwendig ist. Zudem ist dieser Schritt auch nur möglich, falls diese Datenmenge noch zur Verfügung steht und das ist eben aufgrund des großen Datenvolumens nicht immer gewährleistet. Daher sollten alle für die Analyse benötigten Informationen in der Datenbank abgelegt sein.

Die Bewertung der Anomalien und der Eskalationen ist der aufwendigste Arbeitsschritt des Systems. Die Regeln und Profile definieren was zu tun ist und welche Heuristiken eingesetzt werden sollen. Die Ergebnisse werden dann erneut in einer entsprechenden Datenbanktabelle abgelegt.

Zudem ist der Prozess aber auch für die weitere Aggregation, der von inputSNADS übergebenen Datenpunkte über die initialen Betrachtungszeiträume zuständig. Die Datenpunkte größerer Betrachtungszeiträume werden wiederum in spezielle Tabellen abgelegt, sodass der Prozess später problemlos darauf zugreifen kann (vgl. Seite 38).

- **analyseSNADS**

Nachdem die Anomalien und Eskalationen in der Datenbank vorliegen, können problemlos weitere Prozesse diese Daten einsehen. Der Prozess analyseSNADS ermöglicht eben den Einblick und bereitet die Ergebnisse auf eine leicht verständliche und anschauliche Art auf. Beispielsweise können die Histogramme des „Pearson Chi-square“ Algorithmus und die Z-Score Berechnung anhand von Boxplots zusätzlich grafisch dargestellt werden.

Durch die Aufteilung in mehrere Prozesse ist jedoch die Realisierung des Systems komplizierter. Ein wesentlicher Nachteil ist, dass nun eine Kommunikation unter den Prozessen nötig ist. Dies schließt sowohl die Übergabe von Metainformationen als auch eine Signalisierung zwischen den Prozessen ein. Metainformationen sind die Informationen, die für die Bearbeitung nötig sind, wie beispielsweise Regeln und Profile. Eine spezielle Signalisierung zur Übergabe der Datensätze ist zum einen zwischen BRO und inputSNADS und zum anderen zwischen inputSNADS und logicSNADS erforderlich. Genauere Informationen hierzu sind im folgendem Unterkapitel 4.2 beschrieben. Aufgrund der genannten Einteilung ergibt sich der in Abbildung 4.1 dargestellte Aufbau des Intrusion Detection Systems SNADS. Der Prozess analyseSNADS wurde hierbei nicht eingezeichnet, da er lediglich die Daten der eingezeichneten Tabellen visualisiert und daher nur das Schaubild unnötig verkomplizieren würde.

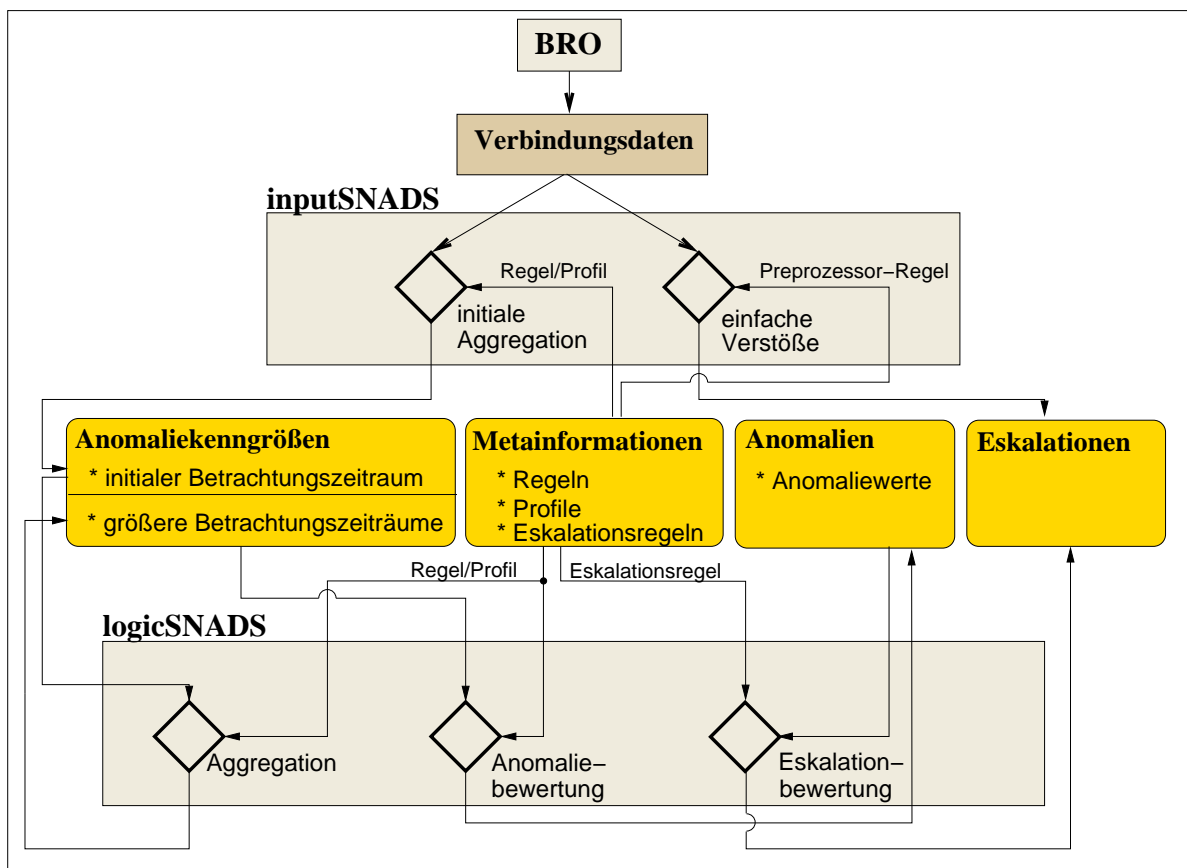


Abbildung 4.1: Struktureller Aufbau

4.2 Implementation

Das Anomalie-basierte Intrusion Detection System SNADS kann in drei Phasen aufgeteilt werden. In der ersten Phase werden die benötigten Datensätze dem System zur Verfügung gestellt. Die zweite Phase beschäftigt sich ausschließlich mit der weiteren Aufbereitung und Abspeicherung der Daten. Die dritte Phase führt die Anomaliebewertung auf Basis der abgespeicherten Datensätze durch und startet danach die Eskalationsbewertung. Die genannten Phasen werden nun der Reihe nach vorgestellt.

4.2.1 Datenlieferant: BRO

Das Intrusion Detection System BRO wurde bereits in Kapitel 2.1.2 vorgestellt und einige Vorteile des Systems auf Seite 36 beschrieben. Es wird eingesetzt, um einzelne Pakete semantisch korrekt zu Verbindungen zusammenzufassen und diese dann an SNADS weiter zu reichen. BRO stellt von Haus aus folgende Informationen zur Verfügung:

- **Startzeitpunkt und Dauer der Verbindung**

BRO's Startzeitpunkt ist der Zeitpunkt an dem das erste Paket einer Verbindung prozessiert wurde. Die Dauer der Verbindung ist der Zeitraum zwischen dem Ersten und dem letzt gesehenen Paket. BRO gibt die Verbindungen nach ihrem Ende aus, sodass diese zeitlich geordnet sind. Die übergebenen Daten stellen eine Momentaufnahme bis zu dem Zeitpunkt des jüngsten Endes der Verbindungen dar, weshalb sich das Ende der Verbindung als Kriterium für die Datenverarbeitung und Datenablage anbietet. Es kann aus der einfachen Addition von dem Startzeitpunkt und der Dauer der Verbindung berechnet werden. Allerdings sind die Verbindung aufgrund der Timeout Problematik (vgl. Seite 67) nicht entsprechend ihrem Ende sortiert. Dies kann zu einer Falschablage der Daten führen. Der Fehler ist jedoch relativ gering, da sie durch üblicherweise relativ kurze Timeouts verursacht werden. Zudem reduziert sich dieser unter Anwendung größerer Betrachtungszeiträume zunehmend. Eine nachträgliche Korrektur wird aus Effizienz Gründen unterlassen.

- **Quell-IP und Ziel-IP**

Die IP-Adresse ist das wichtigste Kriterium bei der Bearbeitung. Sie dient sowohl als Schlüssel für die Datenablage, wird aber auch von Regeln und Profile direkt referenziert. In Abhängigkeit, welcher Partner die Verbindung initial gestartet hat, ist entweder die Quell-IP oder Ziel-IP zu wählen. Bei eingehenden Verbindungen, das sind die, bei dem ein Fremdrechner eine Verbindung zu dem zu schützenden Rechnernetz öffnet, muss die Ziel-IP genommen werden. Bei ausgehenden Verbindungen die Quell-IP. Verbindungen

innerhalb des Rechnernetzes werden in der Regel nicht von SNADS gesehn, da SNADS üblicherweise an den Grenzen des eigenen Netzes analysiert.

- **Verwendete Serviceports**

Ebenso wie bei den IP-Adressen besteht jede Verbindung aus einem Paar aus Ports. Interessant ist in der Regel nur der Ziel-Port, da der Quell-Port von den meisten Betriebssystemen und Anwendungen nicht-deterministisch gewählt wird. Der Ziel-Port liefert einen ersten direkten Aufschluss darüber, welcher Service angesprochen wurde. Er stellt aber kein eindeutiges Indiz dar, da Servicedienste einen beliebigen Port verwenden können. Im Gegensatz zu TCP und UDP verwendet ICMP keine Ports, weshalb diese Informationen in den Datensätzen nicht vorhanden sind.

- **Übertragenes Datenvolumen**

Für jede Verbindung liefert BRO das übermittelte Datenvolumen. Hierbei werden jedoch nur die Nutzdaten der Pakete zusammengefasst. Unterschieden wird zwischen dem gesendeten und empfangenen Datenteil.

- **Anzahl der Pakete**

Jede Verbindung besteht aus einer Menge an Paketen. BRO unterscheidet auch hier zwischen der Anzahl an verschickter und empfangener Paketen.

- **Zustand der Verbindung**

Der Zustand einer Verbindung beschreibt die Semantik des Datenflusses näher. Die von BRO definierten Zustände sind in Tabelle 4.1 beschrieben.

- **Flags der Verbindung**

Mittels einiger Felder wird zudem die Flussrichtung der Verbindung als auch das eingesetzte Übertragungsprotokoll beschrieben.

Die zur Auswertung einiger Anomaliekenngrößen essentiell benötigte Information über die „Anzahl an Paketen“ fehlt jedoch. Des weitern übergibt BRO von Haus aus nicht das beteiligte Serviceport-Paar zu jeder Verbindung, sondern liefert nur den Ziel-Port in ASCII-Form. Diese Erweiterungen konnten problemlos durchgeführt werden, da lediglich eine bestehende Regel in BRO angepasst werden musste.

Ursprünglich leitet BRO, wie bereits beschrieben, nur Verbindungen weiter für die keine weiteren Pakete zu erwarten sind. Um die Genauigkeit der Auswertung zu erhöhen, wurde BRO jedoch erweitert, sodass auch Zwischenwerte lang anhaltender Verbindungen ausgegeben werden. Denn eben solche Verbindungen könnten, sobald sie beendet sind, im Extremfall zu einem sprunghaften Anstieg der Anomaliekenngrößen führen. Da BRO Verbindungen ursprünglich erst nach ihrer Beendigung meldet, würde beispielsweise ein über mehrere Stunden aktiver FTP-Transfer plötzlich zu einem starken Anstieg der Transfermenge führen. Zudem werden durch häufigere Übergaben von Zwischenwerten die einzelnen Datensätze runder, da

BRO Zustand	Erklärung
S0	Verbindungsversuch stattgefunden. Aber es gab keine Antwort von der Ziel-IP.
S1	Verbindung festgestellt, wurde aber nicht beendet.
S2	Verbindung festgestellt. Die Quell-IP wollte sie schließen, es gab aber keine Antwort von der Ziel-IP.
S3	Verbindung festgestellt. Die Ziel-IP wollte sie schließen, es gab aber keine Antwort von der Quell-IP.
SF	Verbindung festgestellt und ordnungsgemäß beendet.
REJ	Verbindungsversuch wurde von der Ziel-IP abgelehnt.
RSTO	Verbindung festgestellt. Die Quell-IP brach aber ab (RST).
RSTR	Verbindung festgestellt. Die Ziel-IP brach aber ab (RST).
RSTOSO	Die Quell-IP schickte ein SYN, gefolgt von einem ACK ohne dass ein SYN-ACK von der Ziel-IP gesehn worden ist.
RSTRH	Die Ziel-IP schickte ein SYN-ACK, gefolgt von einem RST ohne dass ein SYN von der Quell-IP gesehen worden ist.
SH	Die Quell-IP schickte ein SYN, gefolgt von einem FIN ohne dass ein SYN-ACK von der Ziel-IP gesehen worden ist.
SHR	Die Ziel-IP schickte ein SYN-ACK, gefolgt von einem FIN ohne dass ein SYN von der Quell-IP gesehen worden ist.
OTH	Es wurde kein SYN gesehen, dennoch fließen in beiden Richtungen Daten.

Tabelle 4.1: Die möglichen Zustände einer Verbindung unter BRO

sie die tatsächliche Nutzung zeitlich betrachtet präziser widerspiegeln. Da eine Verbindung folglich durch mehrere dargestellt wird, führt dies zu einem enormen Anstieg der zu übertragenen Verbindungsmenge. Daher ist eine Übergabe der Verbindungsdaten in Echtzeit nicht praktikabel und würde die Ressourcen des Rechners nur unnötig strapazieren. Die zu übertragene Anzahl an Verbindungen lässt sich jedoch in der Regel gravierend reduzieren, indem in *Schüben* nur noch die Verbindungen übertragen werden, die länger als eine bestimmte Zeit aktiv waren. Denn die meisten Verbindungen im Netz sind innerhalb weniger Sekunden beendet und würden nur unnötig mehrfach gemeldet werden. Aufgrund dessen wurde BRO dahingehend modifiziert, dass jede Minute alle Verbindungen erneut gemeldet werden, die länger als eine Minute bestehen und innerhalb der letzten Minute aktiv waren. Somit ist jede lang anhaltende Verbindung praktisch in mehrere Teilverbindungen aufgespaltet. Die Länge der Zeitdauer kann frei gewählt werden, jedoch erhöht sich bei kürzeren Zeiten der Aufwand und bei längeren leidet die Genauigkeit. Für die Datenerfassung sind dann für die meisten Anomaliekenngrößen nicht mehr die Daten, die die gesamte Verbindung repräsentieren von Interesse,

sondern lediglich die Änderung seit der vorausgegangenen Meldung. Die Tabelle 4.2 stellt diesen Sachverhalt für die in Kapitel 3.5.1 auf Seite 41 vorgestellten Anomaliekenngößen dar. Die Übergabe der Verbindungsdaten erfolgt über Dateien, da diese persistent sind und im Zuge

Anomaliekenngöße	Verwendung von Zwischenwerten
Transfermenge	
~ des Rechners	Ja
~ eines spezifischen Serviceports	Ja
~ aufgeschlüsselt nach Serviceports	Ja
~ der Verbindungen eines spezifischen Serviceports	Nein
Anzahl an Paketen	
~ des Rechners	Ja
~ eines spezifischen Serviceports	Ja
~ von Verbindungen eines spezifischen Serviceports	Nein
Anzahl beendeter Verbindungen	
~ des Rechners	Nein
~ eines spezifischen Serviceports	Nein
~ mit besonderen Zuständen	Nein
~ einzigartiger IP-Adressen eines spezifischen Serviceports	Nein
Verwendete Serviceports	Ja
Verbindungsdauer eines spezifischen Serviceports	Nein

Tabelle 4.2: Verwendung von Zwischenwerten bei den Anomaliekenngößen

der Entwicklung des Systems auch mehrmals wiederverwendet werden können. Hierbei kann SNADS über ein SIGUSR1 Signal zur Auswertung einer Datei angeregt werden. Das Auslesen der Datenfelder auf den ASCII-basierten Datensätzen ist jedoch eine sehr aufwendige Angelegenheit und bietet kaum Platz für Optimierung. Allerdings könnte die Datenübergabe effizienter mittels einem IPC-Mechanismus, beispielsweise einem Shared Memory Bereichs erfolgen, jedoch würde das größere Eingriffe in BRO bedingen und in einer zu starken Abhängigkeit zwischen den zwei Systemen resultieren.

Problematisch ist die Anwendung der Definition von Verbindungen auf den diversen Netzwerkprotokollen. Verbindungen unter *TCP* sind in der Regel eindeutig durch ein Anfang und ein Ende definiert. Sollte das Ende des verbindungsorientierten Datenflusses aber aus irgendwelchen Gründen nicht erkannt werden, sind Probleme zu erwarten. Hinderlich ist hierbei, dass das TCP-Protokoll laut der Spezifikation keine Keep-Alive Pakete verschicken muss und daher der wahre Status einer Verbindung nicht immer korrekt durch das bloße Abhö-

ren der Verbindung ermittelt werden kann. Beispielsweise kann bei einer interaktiven SSH-Verbindung die Verbindung teilweise über mehrere Tage hinweg offen gehalten werden, obwohl in diesem Zeitraum kein einziges Paket fließt. Da die Intrusion Detection Systeme aber aufgrund begrenzter Rechnerressourcen solche Verbindungen nicht ewig im Speicher halten können, werden sie als beendet angesehen, wenn die Idlezeit gewisse Schranken überschreitet. Die Schranken sind je nach Umfang des Datenverkehrs zu wählen, wobei sie erfahrungsgemäß erheblich unter einer Stunde liegen. Aufgrund dessen ergeben sich unmittelbar folgende Nachteile: Zum einen können bei langen Verbindungen wichtige Informationen verloren gehen, die für das Ergebnis einer Analyse sehr wichtig sein könnten. Des Weiteren könnte ein Angreifer diese Schwachstelle ausnutzen, um unerkannt Daten an dem IDS vorbei zu schieben. Da *ICMP* und *UDP* verbindungslose Protokolle sind, werden UDP- und ICMP-Pakete durch Heuristiken in eine verbindungsartige Struktur (*Flows*) gebracht. Hierbei besteht die Gefahr, dass beim Zusammenfassen der Pakete Fehler zustande kommen, indem Pakete zusammengefasst werden, die kausal gar nicht voneinander abhängen. Zudem entstehen Fehler, wenn nicht alle Pakete mit der gleichen Ursache korrekt einer Verbindung zugewiesen werden, da beispielsweise zuvor ein interner Timeout abgelaufen ist.

4.2.2 Datenorganisation

Die Speicherung der Datenbestände ist von grundlegender Bedeutung, denn die Genauigkeit der Heuristiken und die Aussagekraft der Ergebnisse hängt stark von der Menge zur Verfügung stehender Informationen ab. Es gilt wie bereits erwähnt, dass die Qualität der Auswertung steigt je genauer das Vergangenheitsverhalten einer Anomaliekenngröße beschrieben werden kann. Aufgrund dessen ist eine der wichtigen Forderungen an die Datenspeicherung, dass sie mit großen Datenmengen umgehen kann und auf die Daten natürlich auch möglichst effizient zugegriffen werden kann. Da zudem Datensätze für verschiedene Betrachtungszeiträume verarbeitet werden müssen und sie teilweise Wochen in die Vergangenheit zeigen können, ist ein ressourcen-schonender Zugriff erforderlich. Häufig verwendete Datensätze sollten um einen schnellen Zugriff zu gewährleisten im Speicher gehalten und die anderen im Langzeitspeicher ausgelagert werden. Dieser Schritt sollte automatisch durchgeführt werden und mittels einer universellen Funktion ein transparenter Zugriff auf die Datensätze möglich sein.

Ein weiterer Aspekt ist die Datensicherheit. Würden die gesamten Daten nur im Speicher gehalten werden, dann wäre bei einem unerwartetem Ausfall des Systems unmittelbar ein gravierender Datenverlust die Folge. Aufgrund dieser Überlegungen wird das Datenbanksystem „Berkeley DB“ der Sleepycat Software [51] zur Speicherung der Daten verwendet, welches für diverse Anwendungsfälle maßgeschneiderte Datenbanktypen realisiert. Für den Fall, dass die Fehlersicherheit der Daten nicht allzu wichtig ist und mehr Wert auf die Performanz der

Bearbeitung gelegt wird, bietet sich der Einsatz einer ausschließlich im Hauptspeicher liegenden Datenbank an. In der Regel beschreitet SNADS jedoch einen Mittelweg zwischen performanter Ausführung und Fehlertoleranz, indem eine im Hauptspeicher liegende Datenbank eingesetzt wird, dessen Daten regelmäßig mit der Festplatte abgeglichen werden. Ein weiterer sehr wichtiger Vorteil ist, dass diese Datenbank zeitgleich von mehreren Prozessen verwendet werden können und sich daher auch sehr gut eignen um gemeinsame Datenstrukturen Prozessübergreifend anzubieten. Dies ist beispielsweise für die vom inputSNADS initial aggregierten Anomaliewerte äußerst zweckmäßig, da sie gleich in die entsprechende Datenbank abgelegt werden, ohne das hierbei aufwendige IPC-Mechanismen realisiert werden müssen.

Wie auf Seite 38 beschrieben ist die Aggregation der Datensätze nach der Zeit das zugrunde liegende Prinzip zur Datenreduktion. Das bedeutet letztendlich, dass die Datenpunkte jeder Anomaliekenngröße für verschiedene Betrachtungszeiträume abgespeichert werden müssen und nebenbei nach jeder Aggregationsstufe die Datenmenge abnimmt. Um die Flexibilität des Systems zu erhöhen, kann der initiale Betrachtungszeitraum zwischen einer und sechzig Minuten frei gewählt werden. Jedoch muss ein Vielfaches des Zeitraums sechzig ergeben, da ansonsten aufgrund der auf Seite 70 beschriebenen Zugriffsmethode einige Datensätze verloren gingen. Wie bereits erwähnt gilt wieder, dass sich bei größeren Betrachtungszeiträumen der Rechenaufwand und die Datenmenge reduziert, aber zugleich auch die Genauigkeit der aggregierten Datenpunkte leidet. Bei größeren Betrachtungszeiträumen glätten sich zudem die aggregierten Daten mehr. Um die Performanz von SNADS zu optimieren, erfolgt die Aggregation innerhalb des initialen Betrachtungszeitraums nicht in einer Datenbank, sondern im Prozessspeicher. Eine im Speicher liegende Hash-Tabelle, die ebenfalls mit der „Berkeley DB“ realisiert wird, speichert die Zeiger auf diese Daten und ermöglicht somit einen effizienten Zugriff. Die nachfolgenden Aggregationsstufen vergrößern die Betrachtungszeiträume auf Stunden, Tage und Wochen Basis. Zusätzlich zu der Abnahme der Datenmenge ermöglicht die Aggregation einen effizienten Zugriff auf längere Betrachtungszeiträume, ohne dass die Datenpunkte jedesmal aufwendig neu berechnet werden müssen. Die Betrachtungszeiträume sind hierbei willkürlich gewählt. Andere Betrachtungszeiträume können jedoch jederzeit daraus abgeleitet werden, solange die Daten noch vorhanden sind. Für jeden der festgelegten Betrachtungszeiträume wird jeweils eine eigene Tabelle verwendet, die intern mit B-Trees arbeitet. Durch die Aufspaltung wird die Anzahl der Einträge in den jeweiligen Tabellen reduziert und der Zugriff kann schneller erfolgen. Zudem können die einzelnen Tabellen für sich individuell optimiert werden, wie beispielsweise die Größe des Caches. Des Weiteren kann die Administration über externe Tools spezifischer erfolgen. Aus den gleichen Gründen wird jede Anomaliekenngröße ebenfalls in eigenen Tabellen abgelegt. Hierdurch kann beispielsweise auf einfache Art die aggregierten Datensätze einer bestimmten Anomaliekenngröße gelöscht werden, ohne dass die anderen Anomaliekenngrößen beeinflusst werden.

Die in der Datenbanktabellen abgelegten Daten, wie beispielsweise die Datenpunkte der Ano-

maliekenngrößen, sind direkt referenzierbar, sodass bei jedem Zugriff genau ein Datenpunkt zurückgeliefert wird. Ein aufwendiges Suchen und Extrahieren der benötigten Informationen wird dadurch vermieden. Alternativ könnten mehrere Anomaliekenngrößen bei einem Zugriff zurückgeliefert werden. Bei dieser Methode wäre die Menge der zu speichernden Informationen für jeden Datenbankschlüssel sehr unterschiedlich und würde aufgrund der Implementierung der „Berkeley DB“ zu größeren Lücken in der Speicherausnutzung führen [52]. Der Zugriff könnte sich dadurch eventuell verlangsamen, was aber nicht näher untersucht wurde. Da wie beschrieben eine Datenbank verwendet wird, muss ein Datenbankschlüssel, der einen eindeutigen Zugriff auf die Daten ermöglicht, definiert werden. Bei den Anomalien und Eskalationen gestaltet sich der Datenbankschlüssel denkbar einfach, da lediglich eine einzigartige Identifikationsnummer als Indiz verwendet werden muss. Bei den aggregierten Datenpunkten der Anomaliekenngrößen werden jedoch einige zusätzliche Informationen benötigt: Zunächst muss bekannt sein, welche Anomaliekenngröße gebraucht wird und in Abhängigkeit des Betrachtungszeitraums, welche Datenbanktabelle überhaupt herangezogen werden soll. Beide Informationen sind unmittelbar aus der Definition der Regeln und Profile zu ziehen. Nähere Informationen zu den Regeln und Profilen sind auf Seite 72 zu finden. Des Weiteren muss spezifiziert sein, wessen Daten gesucht werden. In der Regel wird hierzu eine IP-Adresse benötigt. Alternativ könnten aber auch Datenpunkte einer Gruppe von IP-Adressen gesucht werden. Hierzu werden Gruppen-Identifikationsnummern eingeführt, die eindeutig beschreiben, welche Menge an IP-Adressen darin miterfasst werden. Normalerweise müsste der Datenbankschlüssel, um einen eindeutigen Zugriff zu ermöglichen, Felder für die IP-Adresse als auch der Gruppen-Identifikationsnummer beinhalten. Die betreffende Vergleichsfunktion würde dann aber komplexer sein und könnte langsamer ablaufen, weshalb die zwei Informationen in einem Feld zusammengefasst werden. Aufgrund der Definition der Netze nach RFC1700 ist der IP-Bereich von 0.0.0.0 bis 0.255.255.255 dem lokalen Netzwerk vorbehalten und sollte problemlos für die Gruppen-Identifikationsnummer verwendet werden können, da er im Allgemeinen nicht an Knotenpunkten des Netzwerkes Verwendung findet. Ist das System jedoch Teil einer geschichteten Umgebung, in der weitere Rechner stehen, ist jedoch das Vorkommen dieses IP-Bereichs nicht auszuschließen.

Der zweite wichtige Faktor zur eindeutigen Referenzierung ist die genaue Spezifikation der Zeitraums. Die aggregierten Datenpunkte des Systems repräsentieren wie beschrieben den Datenfluss innerhalb eines Zeitintervalls und sind daher durch eine Startzeit des Intervalls und ihrer Länge festgelegt. Die Länge des Intervalls ist wiederum entweder in den Regeln oder den Profilen vorgegeben oder kann direkt aus dem Betrachtungszeitraums der Datenbanktabelle, sprich Stunde, Tage, Woche, abgeleitet werden. Für die Startzeit des Intervalls wird jedoch eine Konvention benötigt. Daher wird definiert, dass die Startzeit stets am Anfang des nächst größeren Betrachtungszeitraums beginnt und dann nur ein Vielfaches der Länge des Betrachtungszeitraums einnehmen kann. Das bedeutet exemplarisch, dass wenn der aggregierte

Stunden-Wert des 6.1.2003 von 15:31 Uhr gesucht wird, dieser in der Datenbank eindeutig bestimmt werden kann durch das Datum des Tages (6.1.2003) und dem Vielfachen der Stunde, also 15. Der Tages-Wert wird analog dazu festgelegt durch den Anfang des Monats (1.1.2003) und dem Vielfachen eines Tages, also in dem Fall 6. Da diese Betrachtungszeiträume nicht verändert werden können, kann jeder aggregierte Datenpunkt eindeutig bestimmt werden. Ein spezielle Behandlung muss jedoch bei den vom Administrator frei wählbaren initialen Betrachtungszeiträume erfolgen. Denn sie dürfen nur zu Beginn einer Stunde verändert werden, da ansonsten bei Veränderung der Intervalllänge innerhalb einer Stunde der Zugriff nicht mehr eindeutig wären. Aufgrund dieser Überlegungen ergibt sich der in Abbildung 4.3 abgebildete Datenbankschlüssel zum Zugriff auf die aggregierten Datenpunkte der Anomaliekenngrößen.

$$\text{Datenbank Schlüssel} = \underbrace{\text{IP/GrpID}}_{\text{Wer?}} + \underbrace{\text{Anfang des Zeitraums} + \text{Offset}}_{\text{Wann?}}$$

Tabelle 4.3: Der Datenbankschlüssel zum Zugriff auf die Anomaliekenngrößen

Die abzuspeichernden Datenpunkte der Anomaliekenngrößen werden von dem System je nach Anforderung unterschieden. So existieren wie beschrieben Anomaliekenngrößen, die mit Intervallen arbeiten, und dementsprechend die Werte einordnen, aber auch andere die lediglich aus wenigen aggregierten Werten bestehen. Bei den auf Intervallen basierten Anomaliekenngrößen, wie beispielsweise „Verbindungsdauer“, muss jedoch die Anzahl der Intervalle fest vorgegeben werden, da eine spätere Aggregation von Datensätzen mit anderer Intervalleinteilung gar nicht möglich ist oder nur mit größerer Ungenauigkeit erfolgen kann. Da der verwendete „Pearson Chi-square Algorithmus“ jedoch stark von der Einteilung der Intervalle abhängt, ist eine starre Intervalleinteilung entweder ungenau, wenn sie zu grob ist oder die Ressourcen werden unnötig verschwendet und der Aufwand maximiert, wenn sie zu fein sind. Es wird daher die auf Seite 49 beschriebene Methode verwendet, um ausgehend von einer relativ feinen Einteilung der Intervalle eine möglichst optimale Einteilung zu erzeugen. Zunächst werden maximal 128 Intervalle für die Einteilung zugelassen.

Bei den anderen Anomaliekenngrößen, die auf wenigen Werten beruhen, gibt es aber auch welche, die zur Abspeicherung eine zweite Dimension benötigen. So ist ursprünglich nur ein Datenpunkt innerhalb eines Zeitraums abzulegen, jedoch müsste beispielsweise bei der Anomaliekenngröße „Transfermenge aufgeschlüsselt nach Serviceports“, zusätzlich noch die theoretisch möglichen 65536 Serviceports beachtet werden. Bei diesen komplexeren Anomaliekenngrößen wird die Anzahl der unterschiedlichen Datenpunkte auf 100 beschränkt, um zu

verhindern, dass die Datenmenge der Anomaliekenngrößen außer Kontrolle gerät. Der Zahlenwert sollte so groß gewählt werden, dass unter Normalbedingungen keine Informationen verloren gehen. Sollte dieser Schwellwert erreicht werden, wird einmalig ein Hinweis ausgegeben und die restlichen Dienste ignoriert.

Ein weiterer äußerst wichtiger Teil bei der Datenverarbeitung sind die *Regeln* und *Profile*. Beide beschreiben exakt das Aufgabenspektrum des ADS. Dadurch können schon bei der initialen Aufbereitung der von BRO übergebenen Verbindungsdaten unnötige Arbeitsschritte vermieden und die Performanz gesteigert werden. Der Aufwand bei der Datenverarbeitung wird jedoch dadurch erheblich komplexer. Sämtliche Regeln und Profile werden derzeit lediglich in einer Datei fest kodiert definiert. Zukünftig sollten Änderungen an den Regeln und Profilen auch während der Ausführung möglich sein. Optional könnte dann auch mittels Lex und Yacc die syntaktische und semantische Korrektheit sichergestellt werden.

Jede Regel und Profil besteht hauptsächlich aus folgenden wichtigen Komponenten:

- Welche Anomaliekenngröße zu untersuchen ist.
- Welcher Algorithmus hierbei verwendet werden soll.
- Zu welchem Zeitpunkt der Vergleich getätigt werden soll.
- Welche Zeiträume zu vergleichen sind.

Grundsätzlich wird in dem System nicht eine „große“ Regel definiert, in der alle nötigen Schritte bis zur Erkenntnisgewinnung festgelegt werden, sondern der Prozess der Datenaggregation wird von der Analyse getrennt und in eigenständigen Regeln beschrieben. Das bedeutet, dass Regeln existieren, die genau beschreiben, welche Anomaliekenngrößen zu sammeln sind und andere vorgeben, was mit den Daten überhaupt zu tun ist. Für jeden der Schritte kann der Administrator den exakten Zeitpunkt festlegen, zum Beispiel „Jeden Tag um 0:10 Uhr“. Die Aufspaltung erhöht zum einen die Flexibilität des Systems, da noch genauer definiert werden kann, was wann zu erledigen ist und zum anderen vereinfacht sich der Programmieraufwand. Der Aufwand ergibt sich vor allem durch die starken Abhängigkeiten der Regeln untereinander, denn die Analyse der Anomalien kann natürlich nur stattfinden, wenn die betreffenden Datenpunkte bereits aggregiert wurden. Dies bedeutet, dass zeitlich vor jeder Regel, die eine Analyse beschreibt, eine entsprechende Regel zur Aggregation der Datenpunkte dieser Anomaliekenngröße aktiviert werden muss. Da die aggregierten Datenpunkte von mehreren Regeln verwendet werden können, sind Misskonfigurationen nicht unmittelbar ersichtlich. Zudem erschwert sich eine fehlerfreie Konfiguration dadurch, dass angestrebt wird die Rechenlast möglichst gleichmäßig auf die Laufzeit zu verteilen. Durch Logiken hätte dies umgesetzt werden können, jedoch wird zunächst darauf verzichtet und stattdessen die Reihenfolge manuell festgelegt.

Die zu vergleichenden Zeiträume bestimmen sich analog wie die des Datenbankschlüssels aus dem Anfang des Zeitraums und der Länge. Der Anfang bestimmt sich, wie bereits erklärt, anhand des des nächst größeren in der Regel definierten Betrachtungszeitraums. Die Länge des Zeitraums wird hierbei durch ein Vielfaches eines festgelegten Zeitintervalls bestimmt. Als Zeitintervalle ist der jeweilige in den Regeln vorgegebene Betrachtungszeitraum sowie Stunde, Tage und Woche erlaubt. Wie bereits genannt, sind durch Kombinationen beliebige Zeiträume erzeugbar. Der Prototypen beschränkt sich jedoch zunächst nur auf Aggregationen gleicher Betrachtungszeiträume.

Auf Seite 37 wurde bereits erwähnt, dass es Vorteile mit sich bringt, wenn die definierten Regeln für mehrere IP-Adressen und Gruppen verwendet werden können. Eines der Vorteile ist, dass auf diese Weise mit minimalen Aufwand für bestimmte Rechnertypen, zum Beispiel Server- oder Desktop Systeme, Einheitsregeln definiert werden können und Änderungen an ihnen dann auch wie gewünscht sich sofort auf alle beteiligten IP-Adressen niederschlagen. Natürlich erhöht sich durch die Erweiterung auch der Aufwand bei der Bearbeitung, da zusätzlich Informationen mitbeachtet werden müssen. SNADS verwendet zur Speicherung der Regeln und ihrer Metainformationen drei Tabellen. In der einen Tabelle stehen in Abhängigkeit eines eindeutigen Identifiers, sämtliche definierten Regeln und Profile. In den anderen beiden Tabellen wird dieser Identifier mit den IP-Adressen und Gruppen in Relation gebracht. Prinzipiell muss das System nach beiden Richtungen auflösen können. Das heißt, bei Eingabe einer IP-Adresse (Gruppe) sollen alle Identifier und bei Eingabe eines Identifiers sollen alle IP-Adressen (Gruppen) zurückgeliefert werden. Abbildung 4.4 veranschaulicht den Zusammenhang. Alternativ könnten die zwei zuletzt genannten Tabellen zusammengefasst werden,

Tabelle 1	ID	->	Regel/Profil
Tabelle 2	IP/Gruppe	->	ID
Tabelle 3	ID	->	IP/Gruppe

Tabelle 4.4: Die Datenbanktabellen zur Zuweisung der IPs zu den Regeln/Profilen

denn prinzipiell ist eine eindeutige Zuweisung auch mittels sekundär Indize möglich. Da es aber vorkommen kann, dass zu einer IP-Adresse (Gruppe) mehrere Identifier gehören und ein Identifier auch für mehrere IP-Adressen (Gruppen) zugehörig ist, existieren Duplikate in der Datenbanktabelle, welche den Einsatz von sekundär Indize nicht ermöglicht. Als Ausweg könnte ein weiterer eindeutiger Identifier dienen, mit dessen Hilfe Duplikate verhindert werden könnten. Im Zuge der Arbeit wurde nicht überprüft, welche Methode die schnellere ist und die Dokumentation [52] lässt keine eindeutigen Schlüsse zu.

Aufgrund der Aufteilung in mehrere Prozesse müssen Möglichkeiten geschaffen werden, um

die Daten und Metainformationen zwischen ihnen auszutauschen. Interessant ist vor allem die Übergabe der initial aggregierten Datenpunkte von dem inputSNADS zu dem logicSNADS Prozess. Als Medium bietet sich ein Shared Memory Bereich an, auf den dann die zwei Prozesse zugreifen können. Prinzipiell muss dann der Zugriff auf den Speicherbereich ausschließlich erfolgen, um keine Fehler durch gleichzeitige Aktionen auf dem gleichen Speicherbereich hervorzurufen. Der ausschließende Zugriff müsste dann mittels Semaphoren sichergestellt werden. Dies hätte allerdings zur Folge, dass es Situationen gibt in der der zeitkritische inputSNADS auf die Freigabe der Ressource wartet und daher in seiner Ausführung stockt. Bei großen Datenmengen könnte der Bearbeitungsfluss dadurch erheblich beeinflusst werden. Nachdem die Daten in den Shared Memory kopiert wurden, müssten sie jedoch noch von dem logicSNADS in die Datenbanktabellen eingetragen werden, um sie späteren Berechnungen zur Verfügung zu stellen. Die „Berkeley DB“ ermöglicht allerdings den zeitgleichen Zugriff von mehreren Prozessen auf die entsprechenden Datenbanktabellen und stellt daher eine relativ leicht einzusetzende Alternative dar. Vorteilhaft ist zudem, dass die Speicherbereiche, in denen zeitgleich nur ein Zugriff erlaubt ist, bereits in der Art optimiert sind, dass nur ein möglichst kleiner Speicherbereich mittels Semaphoren geschützt wird und daher der andere Prozess zur gleichen Zeit schon auf andere Teile der Daten zugreifen kann. Beispielsweise wird daher bei einem Schreibzugriff nicht die gesamte Datenbanktabelle für weitere Schreiber gesperrt, sondern lediglich ein relativ kleiner Speicherbereich um die betreffenden Speicheradresse.

Ein Shared Memory Bereich wird dennoch zur Übertragung von Metainformationen eingesetzt. Denn bei Verwendung des soeben beschriebenen Ansatzes muss logicSNADS zumindest mitgeteilt werden, welche Datensätze in der Datenbank kürzlich abgespeichert wurden und zur weiteren Analyse bereit stehen. Zudem sollte SNADS ebenfalls erkennen, falls eine IP-Adresse überhaupt kein Daten mehr verschickt. Um diese Probleme zu lösen, könnte der logicSNADS Prozess entweder aufwendig über alle IP-Adressen iterieren oder es wird ein festes Schema vorgegeben durch das genau definiert wird, wann welche Daten vorliegen sollten. Der Einsatz eines *Ablaufplans* wird aufgrund der großen Anzahl an IP-Adressen bevorzugt. Der Ablaufplan umfasst sechzig logische Einheiten, die einmalig initialisiert werden und sich üblicherweise zur Laufzeit nicht ändern. Jede Einheit steht für eine Minute einer Stunde und besteht aus einer Menge an Regeln und Profilen. Hierdurch definiert sich die kleinste Zeiteinheit des Systems und beschreibt, was das System in dieser Minute zu bearbeiten hat. Die Wahl einer kleineren Zeiteinheit als das Ein-Minuten-Raster macht nur Sinn, wenn es tatsächlich Regeln oder Profile gibt, die einen initialen Betrachtungszeitraum von unter einer Minute verwenden. Ein zu grobes Raster könnte hingegen zu Verzögerungen und größeren Pausen in der Bearbeitung führen. Die Anzahl der Einheiten als auch das Ein-Minuten-Raster wurde so festgelegt, dass es für den Administrator intuitiv vorstellbar ist und zugleich eine ausreichend feine Verteilung der Arbeitsanweisungen ermöglicht. Der Ablaufplan wird anhand der Regeln

und Profile gebildet und muss auch nur falls diese sich ändern modifiziert werden. Beiden Prozessen liegt der gleiche Ablaufplan vor, sodass dem logicSNADS nur noch mitgeteilt werden muss, welcher Einschub von inputSNADS zuletzt bearbeitet wurde. Diese Methode ist äußerst effektiv und ist zudem tolerant gegenüber Verzögerungen bei der Bearbeitung. Da die Bearbeitung anhand des Ablaufplans synchronisiert wird, ist sie von der Echtzeit unabhängig. Zudem gehen keine Informationen verloren, wenn logicSNADS mit der Anomaliebestimmung nicht schnell genug vorankommt, da es in diesem Fall lediglich zu einer zeitlichen Verzögerung bei der Anomalie- und Eskalationsbewertung kommt.

4.2.3 Anomaliebestimmung

Der Prototyp realisiert bereits die wichtigsten Anomaliekenngößen; davon sind vor allem „Transfermenge aufgeschlüsselt nach Serviceports“, „Anzahl von Verbindungen“, „Verwendete Serviceports“ und „Verbindungsdauer eines spezifischen Serviceports“ interessant. Da alle nötigen Strukturen schon definiert sind und verwendet werden, ist der Aufwand, um die restlichen Anomaliekenngößen hinzuzufügen minimal.

Die in Kapitel 3.5.2 auf Seite 47 vorgestellte Algorithmen zur Anomaliebewertung sind auch alle realisiert. Die auf Schwellwerten basierenden Algorithmen erlauben wie beschrieben feste oder auch durch Prozente angegebene Schranken und können auch zur Prüfung mehrerer Datenpunkte verwendet werden. Für Untersuchungen, wie beispielsweise „Verbindungsdauer eines spezifischen Serviceports“, ist der Pearson Chi-square Anpassungstest heranzuziehen. Es wurde beschrieben, dass die Korrektheit des Ergebnisses direkt von der Wahl der Intervallgrenzen abhängt und es das Ziel sein sollte, möglichst Intervalle mit gleicher Trefferwahrscheinlichkeit zu bilden. Inwiefern das Zusammenfassen der initialen Intervalle in gleichwahrscheinliche Intervalle funktioniert, hängt unmittelbar von der initialen Intervalleinteilung und der zugrunde liegenden Verteilung ab. Die näherungsweise optimale Anzahl der Intervalle M kann wie auf Seite 25 beschrieben, berechnet werden. Der Algorithmus zur Bewertung der Anomaliekenngöße „verwendeten Serviceports“ ist einfach zu realisieren, da er lediglich zwei Listen aus Zahlenwerten miteinander vergleichen muss. Wie bereits beschrieben stellt jede Änderung eine gravierende Anomalie dar und wird daher auch direkt eskaliert. Das Prüfverfahren zur Berechnung und Bewertung des Z-Score wurde ebenfalls implementiert, jedoch muss bei komplexeren Anomaliekenngößen mit zwei Dimensionen jede Informationen einzeln extrahiert und behandelt werden. Bei der Anomaliekenngöße „Transfermenge aufgeschlüsselt nach Serviceports“ muss beispielsweise nach dem Serviceport unterschieden werden, wobei hier wie beschrieben maximal 100 verschiedene Serviceports betrachtet werden. Eine Zerstückelung der voneinander abhängigen Informationen ist nicht erwünscht, weshalb die Ergebnisse der Anomaliebewertung wiederum unter einem Datenbankschlüssel abgespei-

chert werden. Da die Anzahl der dabei zusammengefassten Anomaliewerte variieren kann und ein effektiver Zugriff auf sie möglich sein soll, werden sie zusätzlich in ASCII-basierten Komma-getrennten Listen abgelegt. Die Listen ermöglichen einen relativ schnellen Zugriff auf die Daten. Später werden die Listen von der Eskalationsbewertung ausgewertet. Bei den anderen Anomalien wird das in diesem Fall konkrete Ergebnis in vorgegebene Felder abgelegt und ermöglichen daher einen noch schneller Zugriff, weil auf das Extrahieren der Ergebnisse in den Komma-getrennten Listen verzichtet werden kann. Der Z-Score tätigt die Berechnung nur, wenn mindestens zehn Datenpunkte zur Verfügung stehen. Weniger Datenpunkte führen aufgrund der Verwendung des Medians zu sehr schwankenden Ergebnissen und mehr Datenpunkte verlängern die Dauer der Lernphase erheblich. Es ist auch hier ein Mittelweg zu bestreiten. Der Algorithmus beschränkt sich bei der Anomaliebewertung vorerst auf maximal 20 Datenpunkte, um die Zugriffe auf die Datenbank zu minimieren. Würden mehr Datenpunkte beachtet werden, vergrößert sich der Betrachtungszeitraum und das berechnete Ergebnis könnte genauer sein. Es besteht jedoch dann die Gefahr, dass das Ergebnis der Anomaliebewertung verfälscht wird, da aufgrund von grundlegenden Änderungen der Datenflüsse ältere Datenpunkte das tatsächliche Verhalten mittlerweile nicht mehr korrekt widerspiegeln. Hintergrundinformationen zu diesem Sachverhalt wurden bereits in Kapitel 3.5.2 auf Seite 51 genannt. Die drei Algorithmen zur Anomaliebewertung sind so ausgelegt, dass sie mit möglichst vielen Anomaliekenngrößen arbeiten können. Nur in den Fällen in denen sich die Struktur der gespeicherten Daten unterscheidet ist die Bewertung nicht möglich. Beispielsweise kann der Pearson Chi-square nicht auf die aggregierte Anomaliekenngröße „Transfermenge“ einer IP-Adresse durchgeführt werden. Es wäre nur möglich, wenn die Anomaliekenngröße in Intervalle abgespeichert würde. Bei den Algorithmen, die die Eskalationsbewertung durchführen, wurde lediglich die bereits genannten Eskalationsursachen „Neue verwendete Serviceports“ und „Umfangreiche Nutzung“ vollständig implementiert. Die Eskalationsursache „Denial of Service“ und „Port Scan“ realisieren noch nicht die genannten Schwellwerte und liefern daher teilweise noch nicht wünschenswerte Ergebnisse. Aufgrund dessen wird bei einer Anomalie der Anomaliekenngröße „Anzahl nicht etablierter Verbindungen“ derzeit auch nicht zwischen „Denial of Service“ und „Port Scan“ unterschieden, sondern vorläufig einfach ein „Port Scan“ gemeldet.

Bei der Abspeicherung der Anomalien und Eskalationen sollten grundlegende Informationen, wie beispielsweise Zeit, Dauer und die, bei der Berechnung verwendeten Datenpunkte, mit abgelegt werden. Denn bei der nachträglichen Analyse kann aufgrund der großen Datenmenge nicht sichergestellt sein, dass die Bewertung nachträglich erneut durchgeführt werden kann. Der Zugriff auf die Datenbank sollte zudem nach diesen grundlegende Informationen möglich sein, um ein gezieltes Selektieren auf der Menge der Anomalien und Eskalationen zu ermög-

lichen. Da eine Reihe von zusätzlicher Informationen existieren, bietet sich letztendlich nur der Weg über weitere sekundär Indize an. Daher wird für jede interessante Information ein weiterer sekundär Schlüssel verwendet. Der Einsatz eines sekundär Schlüssels ist aber wie beschrieben nur möglich, wenn die Einträge in der Datenbank eindeutig sind, das heißt keine Duplikate existieren. Gewährleistet wird dies mit Hilfe einer eindeutigen Identifikationsnummer, die jeden Eintrag eindeutig referenziert. Die noch nicht verwendeten Identifikationsnummern sind in einem Bereich des Shared-Memory abgelegt, damit mehrere Prozesse daraus eine eindeutige Identifikationsnummer erhalten können. Der ausschließende Zugriff wird auch hier mittels Semaphoren gewährleistet. Der Zugriff auf den Bereich sollte problemlos erfolgen, da er nur von kurzer Dauer ist und nur statt findet wenn eine neue Anomalie oder Eskalations abgespeichert werden muss.

Problematisch ist die Suche von Einträgen eines bestimmten Zeitraums, da die „Berkeley DB“ lediglich Gleichheitsvergleiche ermöglicht. Dies bedeutet, dass kein Zeitraum bei der Selektion gewählt werden kann, sondern stattdessen direkt ein konkretes Datum angegeben werden muss. Da die Anomalien und Eskalationen zu verschiedenen Zeitpunkten gemeldet werden, wurde ein Raster über die Zeitachse gelegt, welches analog wie bei der Definition des Datenbankschlüssels aus einem eindeutigem Startzeitpunkt und der Länge der Zeitfenster besteht. Der Startzeitpunkt ist hierbei der eindeutige Schlüssel für den Datenbankzugriff und umfasst alle Alarme oder Eskalationen innerhalb des darauf folgenden Zeitfensters. Bei den Anomalien wurde eine Länge des Zeitfensters von 30 Minuten und bei den weniger häufigen Eskalationen eine Länge von 60 Minuten festgelegt. Die Längen der Zeitfenster kann beliebig gewählt werden, jedoch sollten die Längen aufgrund der Häufigkeit der Einträge ein gutes Maß darstellen.

Das nachträgliche Zusammenfassen von Eskalationen ist auf Seite 59 beschrieben. Das jede Eskalation letztendlich aufgrund von einer eindeutigen Regel (Profil) erzeugt wurde, sollte nicht missachtet werden. Dadurch erschwert sich allerdings das Zusammenfassen, da die zusammengefasste Eskalation dann mindestens aus zwei Regeln (Profilen) erzeugt wurde und daher der Zugriff auf die Datenbank nicht mehr eindeutig erfolgen kann. Anstatt einen weiteren sekundär Schlüssel zu definieren und den gesamten Sachverhalt zu verkomplizieren, löscht das System die Eskalation, die in eine andere eingeflossen ist, nicht einfach, sondern stuft sie lediglich als unwichtig und bereits gemeldet ein. Die andere Eskalation wird um die Informationen der herabgestuften Eskalation erweitert. Dies hat zudem den Vorteil, dass nachträglich aufgrund der immer noch eindeutigen Zuweisung zwischen Regel (Profil) und Eskalation weitere Schritte zur Filterung angebracht werden können und die nachträgliche manuelle Kontrolle aufgrund dessen, dass keine Eskalationen gelöscht wurde, erleichtert wird.

Die Problematik der zeitlichen Reihenfolge bei der Anomalienbewertung anhand der Regeln, und Profile wurde bereits auf Seite 74 ausführlich erklärt und ein Lösungsvorschlag geliefert.

Analog hierzu wird die korrekte zeitliche Ausführung der Eskalationsbewertung ebenfalls anhand eines Ablaufplans gewährleistet.

Zur Darstellung der Ergebnisse und der nachträglichen Bearbeitung kann der ProzessanalyseSNADS verwendet werden. Daher braucht er vollen Zugriff auf die Datenbank. Um die Ergebnisse nachträglich besser bewerten zu können, werden sie zudem noch mittels Grafiken visualisiert. Bei dem Pearson Chi-square Anpassungstest wird die Intervalleinteilung mit samt Ist- und Soll-Verteilung und bei dem Z-Score der dementsprechende Boxplot mit den jeweiligen Schranken dargestellt. Bei der Schwellwertrechnung werden einfache Histogramme mit dem entsprechenden vorgegebenen Schranken gebildet. Die libgd [53] Bibliothek stellt rudimentäre Funktionen zum Zeichnen von Grafiken zur Verfügung und wird hierbei eingesetzt. Als Schnittstelle zum Administrator wird das weit verbreitete HTTP-Protokoll eingesetzt, da es problemlos über Rechengrenzen hinweg verwendet werden kann. Zudem ermöglicht es auf sehr einfache Art die grafische Darstellung der Grafiken. Anstatt den Apache-Webserver [54] als HTTP-Server zu verwenden, wurde im Zuge der Arbeit ein eigener programmiert, da sich dadurch einige Nachteile verhindern ließen. Da der Zugriff auf die Datenbank gewährleistet sein muss und auch nachträglich über den analyseSNADS Prozess nochmals Berechnungen durchgeführt werden sollen, hätte entweder die Skriptsprache PHP eingesetzt werden oder ein Modul für den Apache programmiert werden müssen. Da SNADS in C programmiert ist, hätten Teile der implementierten Funktionen zu PHP konvertiert werden müssen. PHP ist zwar relativ flexibel und definiert eine Reihe von praktischen Funktionen, jedoch bietet C hier weitaus mehr Möglichkeiten. Letztendlich führe die Einfachheit des Ansatzes und die besseren Möglichkeiten bei der Programmierung zu diesem Entschluss. Typische Ausgaben des analyseSNADS Prozesses sind in Kapitel 5 zu finden.

Das Anomalie-basierte Intrusion Detection System SNADS wurde auf Linux und Solaris portiert und läuft auf beiden Systemen wie gewünscht. Das Projekt steht unter der GNU General Public License [55] und umfasst derzeit über 15500 Programmzeilen.

5 Erfahrungen mit dem Prototypen

In diesem Kapitel werden einige Erfahrungen, die beim Einsatz des Prototypen SNADS an dem Backbone der netplace Telematic GmbH [56] festgestellt wurden, vorgestellt. Im Speziellen werden nochmals einige wichtige Designentscheidungen herangezogen und ihre Sinnhaftigkeit subjektiv bewertet.

Prinzipiell gilt, dass in dem Bereich der Anomalieerkennung insbesondere bei dem Einsatz eines Anomalie-basierten Intrusion Detection Systems, jahrelange Erfahrung mit der Materie von grundlegender Bedeutung ist. Die Natur und typische Gegebenheiten der Datenflüsse im Netzwerk sollten möglichst bekannt sein und beachtet werden, denn nur dann können sie zweckmäßig verarbeitet und letztendlich korrekt bewertet werden.

Das entwickelte System SNADS stellt ein erstes System zur Anomalieerkennung dar und lässt sich sicherlich in vielen Teilbereichen noch verbessern. Aufgrund der vielfältigen Aggregationsstufen der Datensätze wird jedoch stets ein nachträglicher Einblick in die gesammelten Daten ermöglicht und bietet daher dem Anwender die Möglichkeit, einige wichtige Aspekte der Datenflüsse nachträglich zu begreifen. Die somit gewonnenen Erfahrungen können direkt zur Verbesserung des bestehenden Systems herangezogen werden. Letztendlich ist das Sammeln von Erfahrungen in dem Gebiet der Anomalieerkennung aber ein langjähriger Prozess und Bedarf einiges an Arbeitsaufwand.

5.1 Anomaliekenngößen

Das ADS SNADS bewertet wie beschrieben Anomalien, indem die Diskrepanz zwischen zwei Datensätzen unter Verwendung von Heuristiken bestimmt wird. Der berechnete Unterschied ist mathematisch korrekt, er muss jedoch nicht zwangsläufig auch einem wahren Angriff entsprechen. Bei der Bewertung, der auf diese Art berechneten Anomalien, ist die Frage nach der Ursache der Diskrepanz von grundlegender Bedeutung. Da das System sich prinzipiell nur mit statistischen Eigenschaften beschäftigt, ist eine weitergehende Untersuchung der übermittelten Nutzdaten der Pakete sicherlich oftmals hilfreich. Daher kann es beispielsweise vorteilhaft sein den aus Signatur-basierte IDS gewonnenen Wissenstand der betreffenden Datenflüsse zur

Auswertung ebenfalls heranzuziehen.

Die Aussagekraft der in Kapitel 3.5.1 auf Seite 41 definierten Anomaliekenngrößen ist zudem prinzipiell sehr unterschiedlich. Alle erlauben zwar die Erkennung von Anomalien und können daher zweckmäßig sein. Jedoch können einige Datenpunkte der Anomaliekenngrößen schon auf natürliche Weise schwanken, auch wenn kein Angriff vorliegt und lassen sich daher nur sehr schwer nachträglich korrekt bewerten. Entsprechend meiner Erfahrungen ist die aussagekräftigste Anomaliekenngröße „verwendete Serviceports“, da sie unmittelbar zur Erkenntnisgewinnung beiträgt. Denn neue Serviceports deuten direkt auf den Einsatz eines neuen Dienstes hin und das sollte, wenn das Nutzerverhalten zuvor korrekt festgelegt wurde, nur selten oder gar nicht eintreten.

Die Anomaliekenngrößen, die das Datentransfervolumen betrachten, eignen sich ebenfalls sehr gut zur Auswertung und sind oftmals ein eindeutiges Indiz eines Angriffs. Im Allgemeinen sollte bei der Bewertung des übertragenden Datenvolumens die Unterscheidung in ein- und ausgehende Flussrichtung genügen, sodass der Großteil der spezifischeren Anomaliekenngrößen wahrscheinlich nur selten eingesetzt werden. Die erste Wahl ist deshalb wohl oftmals die Anomaliekenngröße „Transfermenge aufgeschlüsselt nach Serviceports“. Die Anomaliekenngröße „Transfermenge der Verbindungen eines spezifischen Serviceports“ wird mittels Pearson Chi-square analysiert und leidet unter dem Problem, dass nur begrenzt viele Intervalle definiert werden können und daher der Wertebereich stark eingeschränkt wird. Als Ausweg bleibt nur eine relativ starke Vergrößerung der Intervalle, was sich aber negativ auf die korrekte Bewertung des Algorithmus auswirken kann. Das gleiche gilt übrigens auch für die Anomaliekenngröße „Verbindungsdauer eines spezifischen Serviceports“. Die Aussagekraft ist zudem bei dieser Kenngröße von Haus aus gering, da Schwankungen auch hier einen natürlichen Ursprung haben können.

Ebenfalls sehr zweckmäßig sind die Anomaliekenngrößen, die die Anzahl an Verbindungen betrachten. Die Anzahl an etablierter Verbindungen ist ein gutes Indiz für die Auslastung des Rechners und eignet sich ebenfalls zur Untersuchung von DOS-Attacken. Die Anomaliekenngröße „Anzahl der Verbindungen einzigartiger IP-Adressen eines spezifischen Serviceports“ wurde noch nicht implementiert, sodass die Information wieviele einzigartige Teilnehmer bei einer Anomalie beteiligt waren, noch nicht zur Verfügung steht. Aufgrund dessen ist die Eskalationsbewertung der „Flash Crowds“ derzeit nicht möglich. Die Anzahl nicht-etablierter Verbindungen eignet sich, wie erwartet, sehr gut zur Erkennung von Scans.

Die Anomaliekenngrößen, die sich mit der Anzahl an Paketen beschäftigt, wurden in dem Prototypen SNADS noch nicht implementiert. Dies liegt daran, dass diese Anomaliekenngrößen im Allgemeinen eng verbunden sind mit denen der Datentransfervolumen und daher der Informationsgewinn derzeit gering ist. Es ist zu erwarten, dass beide in der Regel synchron fallen und steigen. Zudem können bei der Bewertung eines „Denial of Service“ oder

eines „Port Scans“ anstatt dessen auch die Anomaliekenngrößen „Anzahl an nicht-etablierter Verbindungen“ und „Anzahl an etablierter Verbindungen“ herangezogen werden.

Die Anomaliekenngröße „Verbindungsdauer eines spezifischen Serviceports“ ist wie beschrieben von vielen Faktoren abhängig und liefert daher bei der Erkennung einer Anomalie oftmals keine unmittelbare Aussage. Aufgrund dessen ist der Einsatz der Anomaliekenngröße in vielen Szenarien nicht zweckmäßig.

5.2 Anomaliebewertung

Ein weitere äußerst wichtige Entscheidung war die Wahl der Algorithmen, die zur Bewertung von Anomalien verwendet werden. Sie wurden in Kapitel 3.5.2 auf Seite 47 vorgestellt und werden nun nochmal subjektiv bewertet. Für die Bewertung wurden zwei verschiedene Datensätze verwendet, die über einen Zeitraum von zwei Monaten an dem Backbone der netplace Telematic GmbH gesammelt wurden. Zum einen wurde ein relativ stark frequentierter Server beobachtet, auf dem mehrere Dutzend virtuelle Webserver liefen. Aufgrund der großen Datenmenge konnte hier nur das HTTP-Protokoll betrachtet werden. Zum anderen wurde ein weiteres Class-C Netzsegment beobachtet, auf dem mehrere Serverrechner und auch einige Desktop Systeme liefen. Die Auswertung erfolgte mittels gemeinsamen, relativ allgemein gültigen Regeln und Profilen. Einige Erkenntnisse bezüglich der Anomaliebestimmung werden im Folgenden dargestellt und mit realen Beispielen unterlegt.

Standardized residual

Der vorgestellte Z-Score Algorithmus ist ein sehr robuster Algorithmus, der in vielen Situationen eingesetzt werden kann. Er funktioniert auch gut bei Anomaliekenngrößen, deren Einflussgröße stark schwanken, solange mindestens 50% der aggregierten Datenpunkte den tatsächlichen Verlauf gut nachbilden. Denn aus diesen Datenpunkten ergibt sich der MAD, welches das zur Bewertung zugrunde liegende Maß darstellt. Ein Beispiel, in dem das Prinzip nicht funktioniert ist der vereinfachte Boxplot in Abbildung 5.1. Er zeigt die von SNADS erzeugte Verteilung des Z-Score der Anomaliekenngröße „Transfermenge aufgeschlüsselt nach Serviceports“ des POP3-Ports (110) eines Servers. Insgesamt wurde 37 auf Stundenbasis aggregierte Datenpunkte zwischen 0 Byte und 35 Megabyte gemessen. Der berechnete MAD von 206 Kilobyte ist in der Abbildung in Intervalle um den Median (216 Kilobyte) eingezeichnet und bezeichnet wie beschrieben den jeweiligen Anomaliewert, in dem der zu untersuchende Datenpunkt liegt. Bei nachträglicher Analyse erkennt man, dass – nach dem vorgegebenen Schema der Schranken – 11 Datenpunkte eine Anomaliewert größer als 0 haben, und davon 10 außerhalb der fünften Schranke liegen und daher direkt

einen maximalen Anomaliewert zugewiesen bekommen. Die relativ große Anzahl der Anomalien mit maximalen Anomaliewert ergeben sich aufgrund einer nur sporadischen, aber wenn dann intensiven Nutzung des POP3-Dienstes und führt letztendlich dazu, dass die Datenpunkte ungünstig um den Median liegen und eine Normalverteilung daher nur schlecht wiedergespiegelt wird. Aufgrund dieser Schwankungen ist die Aussagekraft des Algorithmus in diese Fall ohnehin zu bezweifeln. Der Sachverhalt wird auch deutlich, da sich viele Datenpunkte bei den Ausläufen um 0 und dem positiven Extrempunkt wiederfinden.

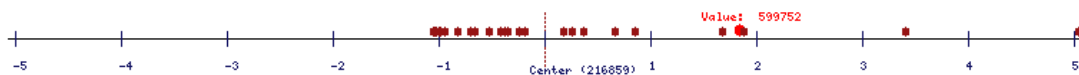


Abbildung 5.1: Boxplot einer Z-Score Berechnung stark schwankender Datenpunkte

Ein weiteres Problem kaum genutzter Anomaliekenngrößen wird ebenfalls deutlich. Denn oftmals können die Datenpunkte aufgrund des Typs nur positive Werte annehmen und führt bei sehr geringen Werten zu einer „Stauchung“ bei 0. Als Folge ist die Berechnung in einzelnen Fällen mangelhaft. Relativ kleine Werte neigen zudem bei Gebrauch zu verhältnismäßig stärkeren Schwankungen und führen daher oftmals zu sehr sensiblen Bewertungen. Daher könnte es sinnvoll sein, die zur Bewertung verwendeten Schranken des Schemas in Abhängigkeit der Größe der Werte zu wählen. Fraglich ist jedoch, ob es überhaupt sinnvoll ist, einen Dienst zu überwachen, der kaum genutzt wird und ob für diesen Fall nicht besser mit einer Regel auf Basis der Schwellwertrechnung (vgl. Seite 54) zu arbeiten ist. Insbesondere stellt sich die Frage bei den für die Eskalationsursachen „Denial of Service“ und „Port Scan“ benötigten Anomaliekenngrößen. Es zeigt sich in der Praxis, dass die Anomaliekenngrößen bei schwach ausgelasteten Rechnern verhältnismäßig stark schwanken können und daher oftmals fälschlicherweise eine Anomalie vermutet wird. Bei der Implementation der Eskalationsursache „Denial of Service“ fehlt derzeit noch die beschriebene Minimal-Schranke von 5000, sodass noch mit einer Verbesserung zu rechnen ist. Eventuell sollte bei der Erkennung der „Port Scans“ und anderer Eskalationsursachen ebenfalls eine Minimal-Schranke definiert werden, um auch hier unnötige Eskalationen zu vermeiden. Ein Anpassen der Parameter des Z-Score Algorithmus ist bei diesem Problem oftmals nicht zu empfehlen, da die Schwankungen oftmals zu gewaltig sind und die erforderliche Anpassung die Sensitivität des Algorithmus zu stark beeinträchtigen würde. Daher bliebe nur noch eine spätere Selektion mittels der Eskalationsbewertung übrig. Der Einsatz der Schwellwertrechnung ist in diesem Fall zu bevorzugen. Eine weitere Alternative gegen zu starke Schwankungen besteht darin, einen größeren Betrachtungszeitraum zu wählen. Durch die Aggregation werden die Schwankungen gleichmäßiger und die Berechnung weniger fehlerträchtig. Hierbei ist aber zu beachten, dass die Sensitivität des Algorithmus leidet, das heißt das mehr Angriffe nicht erkannt werden. Zudem kann

es zu größeren Verzögerungen zwischen dem eigentlichen Vorfall und der Auswertung kommen, denn bei der Prüfung von Datenpunkten auf Basis ganzer Tage, lässt sich die Berechnung auch erst am Ende des Tages durchführen. Das System SNADS kann zwar wie beschrieben auch on-the-fly Datensätze aggregieren, dies würde aber zu einem erheblichen Mehraufwand führen und wird daher derzeit vermieden. Letztendlich ist bei der Vergrößerung des Betrachtungszeitraums die Zweckmäßigkeit der Berechnung zu beurteilen. Das obige Beispiel des POP3-Dienstes wird zum Vergleich nochmals betrachtet, wobei aber nun auf Tage aggregierte Datenpunkte betrachtet werden. Der Boxplot 5.2 beinhaltet auch hier 33 Datenpunkte, die folglich 33 Tage repräsentieren und Werte zwischen rund 80 Kilobyte und 76 Megabyte einnehmen. Der Median ist nun 12,3 Megabyte und der MAD beträgt 10,2 Megabyte. Durch die Vergrößerung des Betrachtungszeitraums haben nun nur noch vier Datenpunkte einen Anomaliewert größer als 0 und nur einer wurde direkt als Anomalie eingestuft.

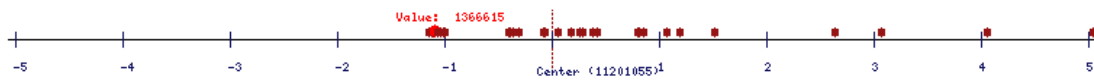


Abbildung 5.2: Veränderter Boxplot bei Vergrößerung des Betrachtungszeitraums

Die Vollständigkeit bei der Erkennung von Angriffen minimiert sich wie eben beschrieben bei Vergrößerung des Zeitraums. Die subjektive Bewertung kann hierbei jedoch erheblich erleichtert werden, weshalb SNADS für beliebige Zeiträume Grafiken zur Verfügung stellen kann. In Abbildung 5.3 ist ein Ausschnitt der aggregierten Tageswerte des transferierten Datenvolumens dieses Dienstes zu sehen. Die Paare aufeinanderfolgender niedriger Datenpunkte stellt hierbei stets ein Wochenende dar und verdeutlicht nochmal wie sinnvoll die Unterscheidung der Wochentage bei der Analyse ist. Der Ausschlag in dem 19. Intervall ist der in Abbildung 5.2 als direkte Anomalie klassifizierte Ausreißer. Die zwei der restlichen drei als anomal eingestuft Datenpunkte befinden sich in der Vorwoche und sind ebenfalls sichtbar. Der vierte Datenpunkt wird in dieser Grafik nicht gezeigt.

Bei dem Vergleich verschiedener IP-Adressen wird deutlich, dass die Auslastung der IP-Adresse direkt die Anzahl der Anomalien beeinflusst. Es gilt, dass je häufiger ein Dienst genutzt wird, umso regelmäßiger stellt sich auch die Benutzung dar und umso besser folgen die einzelnen Datenpunkte einer Normalverteilung. Bei der Betrachtung, der auf Stunden basierten Datenpunkte von über 60 Tagen des stark frequentierten Webservers, sind die Anzahl der vermuteten Anomalien im Vergleich zu der anderen Datenquelle äußerst gering. Hierbei wurde für die Anomaliekenngröße „Transfermenge aufgeschlüsselt nach Serviceport“ in dem Zeitraum nur 37 Anomalien mit einem Anomaliewert von 1 berechnet, wobei 26 davon direkt hintereinander eintraten und, wie sich herausstellte, eine gemeinsame Ursache

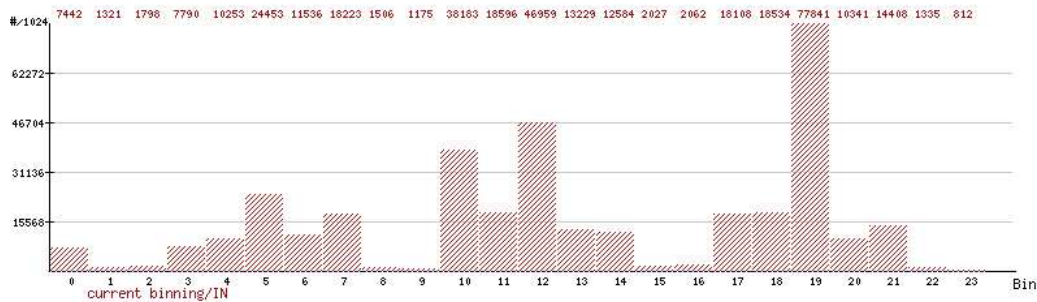


Abbildung 5.3: Histogramm des übertragenen Datenvolumens des Port 110 auf Tagesbasis

hatten. Die Ursache der über einen ganzen Tag hinweg stattgefundenen extremen Auslastung war die Veröffentlichung einer neueren Version des bekannten CD-Brennprogramms X-CD-Roast [57]. Die restlichen elf der als Anomalie eingestufteten Stundenwerte der betrachteten 60 Tage traten wahllos zu verschiedenen Zeiten auf und deuten auf eine umfangreiche Nutzung des HTTP-Ports hin. Sie können im Schritt der Eskalationsbewertung näher betrachtet und gegebenenfalls „gefiltert“ werden. Exemplarisch ist der Boxplot einer Z-Score Berechnung in Abbildung 5.4 zu sehen und verdeutlicht nochmal die Qualität des Datensatzes und die Lage des unüblichen Ausreißers.

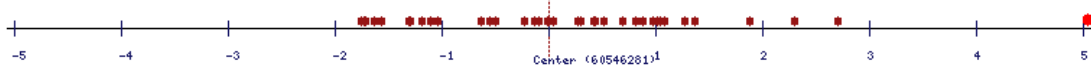


Abbildung 5.4: Boxplot der Z-Score Berechnung einer Flash Crowd

Die Abbildung 5.5 beschreibt die übertragene Datenmenge des Webservers auf Tagesbasis und verdeutlicht nochmal die Relevanz der Anomalie. Der rapide Anstieg am Tag der Veröffentlichung und der kontinuierliche Abfall der übertragenden Datenmenge sind deutlich zu erkennen.

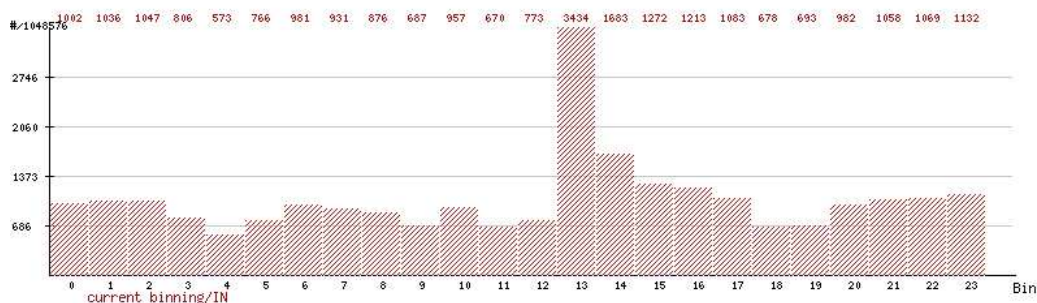


Abbildung 5.5: Histogramm des übertragenen Datenvolumens des Port 80 auf Tagesbasis

Dies ist der klassische Fall einer „Flash Crowd“, die aber weil die betreffende Anomaliekenngröße noch nicht implementiert ist, noch als „Umfangreiche Nutzung“ deklariert wurde.

Damit die aggregierten Datenpunkte möglichst gut den tatsächlichen Verlauf der jeweiligen Anomaliekenngröße nachbilden, wird die Berechnung nur durchgeführt, wenn wie beschrieben mindestens zehn Datenpunkte aus der Vergangenheit zur Verfügung stehen. Die Anzahl scheint im Einsatz praktikabel zu sein, da dadurch einerseits die Lernphase klein ist und die Ergebnisse in der Regel gut sind. Bei sehr stetigen Datenwerten könnte die Anzahl sogar noch ein wenig reduziert werden.

Zusammengefasst eignet sich der Z-Score gut zur Anomalieerkennung. Die meisten Probleme beruhen auf eine ungleichmäßige Auslastung der jeweiligen IP-Adresse und lassen sich oftmals durch die vorgestellten Methoden, wie Vergrößerung des Betrachtungszeitraums, reduzieren.

Pearson Chi-square Anpassungstest

Der Pearson Chi-square Anpassungstest wird für zweidimensionale, in Intervalle unterteilte Anomaliekenngrößen wie beispielsweise der „Verbindungsdauer eines spezifischen Serviceports“ und „Transfermenge der Verbindungen eines spezifischen Serviceports“ eingesetzt. In beiden Fällen stellt sich der Algorithmus als nicht geeignet heraus. Das zentrale Problem ist, dass die bei der Berechnung verwendeten Datensätze zu stark schwanken und sich selbst bei Betrachtung eines relativ großen Zeitraumes von einer Woche nur schlecht dem berechneten theoretischen Erwartungswerten der Intervalle nähern. Es ist nicht unüblich, dass bei dem Vergleich zweier Wochen die Anzahl der Verbindungen innerhalb der Intervalle, trotz Normierung auf die gleiche Gesamtzahl, um mehr als 10% schwanken. Der Pearson Chi-square scheint bei ausgeprägten Schwankungen sehr empfindlich zu sein, sodass unser System fast immer auf eine Anomalie hinweist. Abbildung 5.6 stellt die Verteilung der Intervallwerte der Anomaliekenngröße „Transfermenge der Verbindungen eines spezifischen Serviceports“ dar und verdeutlicht die Problematik exemplarisch. Auf den ersten Blick scheinen die zwei Verteilung bis auf zwei Intervalle identisch, tatsächlich werden jedoch nach Pearson Chi-square lediglich drei Intervallen als relativ ähnlich klassifiziert.

Ein weiteres schwerwiegendes Problem wird ebenfalls in Abbildung 5.6 verdeutlicht. Denn die χ^2 -Verteilung ist nur dann näherungsweise exakt, wenn die Treffer in den Intervalle annähernd gleichverteilt sind und möglichst eine bestimmte Anzahl an Intervalle verwendet werden (vgl. Seite 25 und 49). In diesem Beispiel sollten in jedem Intervall ungefähr 660 Treffer landen, sodass aufgrund der dargestellten Verteilung das Ergebnis ohnehin stark

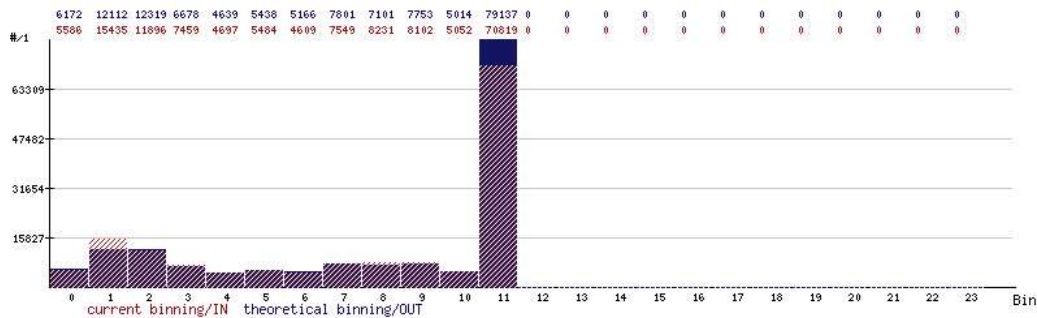


Abbildung 5.6: Eine typische Chi-square Verteilung

bezweifelt werden muss. Prinzipiell gilt jedoch, dass das χ^2 Ergebnis des Pearson Chi-square Algorithmus unter einer noch feineren Einteilung der Intervalle nur größer werden kann und daher unter gleichen Bedingungen noch deutlicher eine Anomalie vermutet wird. Dieser Sachverhalt wird deutlich, wenn beispielsweise der Erwartungswert 1200 ist und für die Differenz $(O_i - E_i)$ 80 angenommen wird. Der berechnete χ^2 Wert des Intervalls beträgt dann 5,3 und kann unter keiner Aufspaltung des Intervalls jemals unterschritten werden. In der Praxis ergibt sich daher unter Vergrößerung der Anzahl der Intervalle und der Verkleinerung der Intervallbreite keine nennenswerten Änderungen der Ergebnisse. Im Allgemeinen wird das berechnete χ^2 Ergebnis im Gegenteil sogar aufgrund erheblicher Schwankungen der Datenpunkte ein wenig größer und deutet daher eher auf eine Anomalie hin.

Eine Ursache dieses Problems ist, dass die Datenpunkte der Anomaliekenngrößen an den Enden des Wertebereichs oftmals Extremas aufweisen und sie derzeit nicht adäquat in gleichhäufige Intervalle unterteilt werden können. Denn SNADS verarbeitet beispielsweise derzeit für die Untersuchung der Verbindungsdauer nur Werte mit einer Kommastelle und da verhältnismäßig viele Verbindungen des HTTP-Ports sehr kurz sind, finden sich die meisten Treffer auch nur in den wenigen Intervallen, die kurze Verbindungen repräsentieren, wieder. In Abhängigkeit der Anomaliekenngröße und der realen Datensätze können sich aber auch andere Häufungspunkte ergeben (siehe auch Abbildung 5.6).

Es ist zu erwarten, dass die Intervallbreite sehr klein gewählt werden muss, um die Anzahl der Treffer auf ein akzeptables Maß runterzudrücken und die Annahmen des Algorithmus vollständig zu erfüllen. Die Aussagekraft bei sehr kleinen Intervallbreiten ist aber ohnehin fraglich, wenn die minimalen Schwankungen auf natürliche Weise innerhalb des Netzes auftreten können. Denn eine Anomalie, die auf einer Verschiebung der Häufigkeiten der Verbindungsdauer von 0,1 auf 0,2 Sekunden beruht, dürfte in der Praxis irrelevant sein. Ein zusätzliches Problem ist, dass bei einer Verkleinerung der Intervallbreite die Gesamtanzahl der Intervalle zunehmen muss, um den gleichen Wertebereich abzudecken. Die maximale Anzahl der Intervalle ist aber, um eine performantere Bearbeitung zu ermöglichen, ursprünglich von dem

System auf 128 begrenzt. Alternativ wäre es vorteilhaft anstatt konstante Intervallbreiten zu verwenden, die Intervallbreiten lediglich an den Rändern des Wertebereichs feiner zu gestalten und ansonsten ein gröberes Raster zu verwenden. Der Algorithmus könnte dann ohne Zunahme der Gesamtanzahl der Intervalle auf Basis dieser feineren Einteilung exaktere Ergebnisse liefern.

Letztendlich ist der Pearson Chi-square Anpassungstest in dieser Form aufgrund enormer Schwankungen bei den Anomaliekenngrößen nicht geeignet. Die Verbindungsdauer ist zu stark von der jeweiligen Leitungsqualität abhängig, als dass bei der Anomaliekenngröße „Verbindungsdauer eines spezifischen Serviceports“ ein allgemeingültiger „Erwartungswert“ bestimmt werden kann. Das übertragene Datenvolumen der einzelnen Verbindungen ist ebenfalls stark von der Nutzung des jeweiligen Dienstes abhängig, sodass bei dem Vergleich von auf Wochen basierten Datenpunkte auch bei der Anomaliekenngröße „Transfermenge der Verbindungen eines spezifischen Serviceports“ kein zuverlässiger Erwartungswert berechnet werden kann. In anderen Szenarien oder auch für andere Anomaliekenngrößen könnte dennoch der Pearson Chi-square Algorithmus praktikabel sein. Ein robusterer Algorithmus, der nicht so sensibel gegenüber Schwankungen ist, sollte jedoch bevorzugt werden. Dadurch wird dann aber auch die Genauigkeit des Ergebnisses maßgeblich beeinflusst. Es sei auf [43] verwiesen.

Beispielsweise könnte ein an Pearson Chi-square angelehnter Algorithmus verwendet werden, der aber bei der Berechnung benachbarte Intervalle mitbetrachtet und daher eventuell nicht so sensibel auf Schwankungen innerhalb eines Intervalls reagiert. Um die Hypothese zu bewerten, müsste dann eine empirische Verteilungsfunktion über die neuen „ χ^2 Ergebnisse“ gebildet werden, um dann dementsprechend die Wahrscheinlichkeit des Auftretens der einzelnen Datenpunkte bestimmen zu können. Eine Beurteilung der Anomalität ist jedoch unter anderem nur dann akkurat, wenn die einzelnen berechneten Anomaliewerte eindeutig der Schwere der Diskrepanz zugewiesen werden können.

Schwellwertrechnung

Die Algorithmen für die Schwellwertrechnung sind gänzlich realisiert worden und funktionieren den Parametern entsprechend einwandfrei. Das große Problem ist aber eben die Wahl korrekter Parameter in der Art, dass auch tatsächlich nicht unnötig Anomalien gemeldet werden und zugleich aber auch keine Anomalien unerkannt bleiben. Das Ziel ist die Festlegung einer Schranke, ab der der Administrator von einer Anomalie ausgeht. Die Regeln der Schwellwertrechnung beinhalten entweder feste absolute Schranken oder können prozentuale Abweichung um ein konkreten Wert abdecken. In beiden Fällen ist für die Wahl angemessener Parameter

Erfahrung mit der Anomaliekenngröße nötig. Aufgrund der Unterschiede in der Auslastung diverser IP-Adressen ist es oftmals auch erforderlich für jede IP-Adresse eigene Parameter zu definieren. Das steigert jedoch den Aufwand zusätzlich. Einmal definierte Parameter müssen aber zudem dem möglicherweise veränderten Nutzungsprofil angepasst werden.

Da ein Verstoß gegen die Schranken laut Definition schon eine Anomalie darstellt, ist die Aussagekraft maximiert. Im Allgemeinen ist der Aufwand zur Pflege der Parameter immer dann relativ groß, wenn die Anomaliekenngrößen relativ stark schwanken und daher die Parameter oft nachgebessert werden müssen. Zusammengefasst ist die Schwellwertrechnung eine effektive, aber aufwendige Methode um Anomaliekenngrößen zu überwachen.

Vergleich zweier Zahlenreihen

Der Algorithmus zum Vergleich zweier Zahlenreihen ist relativ primitiv, da keine Feinabstufung der Anomaliewerte nötig ist. Eine Diskrepanz ist wie beschrieben stets eine Anomalie und kann näher untersucht werden. Im realen Einsatz erkennt der Algorithmus mit der zugehörigen Anomaliekenngröße „verwendete Serviceports“ nur sehr selten eine Anomalie und ist durch die gute Aussagekraft sehr zweckmäßig. Lediglich bei sehr unregelmäßig genutzten IP-Adressen und ihrer Dienste werden unnötig viele Anomalien ausgerufen. Durch eine Vergrößerung des Betrachtungszeitraums kann dem entgegen gewirkt werden, jedoch ist die Analyse dann nicht so sensibel. Wird beispielsweise die vergangene Woche als Grundlage der verwendeten Dienste gewählt, so würde bei einer Reinfektion eines bereits gemeldeten und danach entfernten Backdoors, dieser innerhalb der Woche nicht erneut gemeldet werden und somit unerkannt bleiben.

Eine Erweiterung des Algorithmus erlaubt eine Untersuchung nach „nicht mehr verwendeten Serviceports“. Damit ein illegaler, kurzfristig aktiver Dienst nicht mehrfach zu Anomalien führt, muss der vergangene Betrachtungszeitraum klein gewählt werden. Bei der Untersuchung der Stundenwerte sollte er beispielsweise nur eine Stunde umfassen. Auch dies funktioniert wunderbar, solange es keine Dienste der IP-Adresse gibt, die nur sehr sporadisch verwendet werden. In der derzeitigen Implementation ist die Erweiterung des Algorithmus dann nicht sinnvoll einzusetzen, da zuviele Anomalien gemeldet werden. Durch das Einfügen einer Logik, die beispielsweise bereits gemeldete Dienste filtert, kann das Problem aber gelöst werden.

Aufgrund der unterschiedlichen Anforderungen an die Parameter des Algorithmus ist der gekoppelte Einsatz beider Untersuchungsformen in einer Regel oder Profil nicht sinnvoll.

5.3 Eskalationsbewertung

Die Eskalationsbewertung wurde bereits in Kapitel 3.5.3 auf Seite 54 vorgestellt. Hierzu wurden einige konkrete Eskalationsursachen festgelegt, aber nur zum Teil in dem Prototypen realisiert. SNADS erkennt derzeit die „Umfangreiche Nutzung“, den „Denial of Service“ als auch den „Port Scan“ über das Eskalationsprinzip. Die Eskalationsursache „Neue verwendete Servicedienste“ wird derzeit nicht betrachtet, da sie im Anomaliefall direkt auf eine Anomalie hinweist und daher nicht mit den vergangenen Datenpunkten korreliert werden muss. Die Eskalationsursache „Flash Crowd“ ist im derzeitigen Stand der Implementierung nicht durchführbar. Das Eskalationsprinzip ist aber wie erwähnt ebenso für den Z-Score und damit für alle weiteren aggregierten Anomaliekenngrößen ebenso geeignet. Die Ergebnisse der Schwellwertrechnung als auch des Pearson Chi-square können derzeit nicht über Eskalationsbewertung ausgewertet werden. Die Zusammenfassung einzelner Eskalationsursachen ist noch nicht realisiert, weshalb eine Bewertung nicht möglich ist. Die einzelnen Eskalationsursachen werden korrekt erkannt, ihre Qualität hängt jedoch unmittelbar von den Ergebnissen der Anomaliebewertung ab. Daher hat die Eskalationsbewertung ebenfalls mit dem Problem durch selten genutzte Dienste zu kämpfen. Sind die Ergebnisse der Eskalationsbewertung zu schlecht, sollte daher zunächst die Anomaliebewertung der jeweiligen Anomaliekenngröße geprüft und optimiert werden. Erst danach sollten eventuell die Parameter der Eskalationsbewertung angepasst werden. Bei sehr schwankenden Datenpunkten der Anomaliekenngrößen kann aber selbst durch Änderung der Parameter der Heuristiken oftmals immer noch kein befriedigendes Ergebnis bei der Anomaliebewertung festgestellt werden. Beispielsweise kommt es bei sehr sporadischer, aber wenn dann intensiver Nutzung eines POP3-Servers auch bei optimaler Wahl der Parameter zu sehr schwankenden Anomaliewerten (vgl. Abbildung 5.1 auf Seite 82). Stark schwankenden Datenpunkte der Anomaliekenngrößen können zu fälschlicherweise gemeldeten Eskalationen führen. Eine nachträgliche Filterung auf der Menge berechneter Eskalationen bietet sich dann an. Als Kriterium der Filterung wird die Länge des Auftretens der Eskalation herangezogen, denn es gilt im Allgemeinen, dass je länger eine Anomalie anhält, umso relevanter ist sie dann auch. Im realen Einsatz zeigte sich, dass durch die Filterung der Eskalationen mit einem einfachen Zeitfenster die Menge der gemeldeten Eskalationen bei relativ stark schwankenden Datenpunkten einer Anomaliekenngrößen erheblich reduziert werden kann. Wirklich sinnvoll ist diese Methode jedoch nur bei sehr schwankenden Datenpunkten, bei denen die Anzahl an Fehlalarmen, die in diesem Fall hauptsächlich bei sehr kurzen Zeiträumen auftreten, reduziert werden soll. Es gilt natürlich auch hier, dass durch diese Filterung Informationen verloren gehen und eventuell tatsächliche Anomalien dann nicht mehr erkannt werden. Für die regelmäßig genutzten IP-Adressen ist dieser Schritt nicht nötig, da die Qualität der Eskalation von Haus aus zufrieden stellend ist.

Prinzipiell ist die Eskalationsbewertung eine Selektion der zuvor berechneten Anomalien nach ihrer Schwere, die sich einerseits aufgrund der Heftigkeit und andererseits auch aufgrund der zeitlichen Dauer der Anomalie bestimmt. Zusätzlich wird versucht den Anomalien typische Ursachen zuzuweisen. Durch die Eskalationsbewertung wird eine effizientere Bearbeitung der Anomalien ermöglicht und führt letztendlich zu einer Arbeitserleichterung. Abbildung 5.7 zeigt die typische Ausgabe der auf Stunden basierten Eskalationen eines Zeitraums von rund 60 Tagen, wobei hier sechs Eskalationen mit einem einfachen Zeitfenster bereits nach obiger Methode gefiltert wurden.

ID	IP_GrpID	ERuleID	EscalationTyp	EValue	Timeframe	Port	EscalationTime
4	213.183.1.128	1	BANDWIDTH	1.7	2	80	Wed May 7 01:00:00 2003
8	213.183.1.128	1	BANDWIDTH	-1.6	4	80	Sat May 17 15:00:00 2003
10	213.183.1.128	1	BANDWIDTH	1.9	5	80	Tue May 20 01:00:00 2003
12	213.183.1.128	1	BANDWIDTH	-1.2	4	80	Sat May 24 15:00:00 2003
16	213.183.1.128	1	BANDWIDTH	1.7	2	80	Wed May 28 08:00:00 2003
17	213.183.1.128	1	BANDWIDTH	-1.5	5	80	Thu May 29 18:00:00 2003
18	213.183.1.128	1	BANDWIDTH	1.7	2	80	Sat May 31 05:00:00 2003
19	213.183.1.128	1	BANDWIDTH	1.4	2	80	Mon Jun 2 05:00:00 2003
21	213.183.1.128	1	BANDWIDTH	31.2	34	80	Tue Jun 3 16:00:00 2003
23	213.183.1.128	1	BANDWIDTH	1.4	2	80	Thu Jun 5 05:00:00 2003
24	213.183.1.128	1	BANDWIDTH	2.7	4	80	Fri Jun 6 00:00:00 2003
25	213.183.1.128	1	BANDWIDTH	2.6	5	80	Fri Jun 6 08:00:00 2003

Abbildung 5.7: Exemplarische Liste von Eskalation eines 60-tägigen Zeitraums

Die Beschriftung der Spalten ist zum Großteil selbsterklärend. Timeframe beschreibt die zeitliche Dauer der Eskalation und ist in diesem Fall ein Vielfaches einer Stunde. EValue ist stets die Summe der einzelnen Anomaliewerte dieses Zeitraums.

Die Eskalation mit der ID 21 ist die oben beschriebene „Flash Crowd“, die durch die Veröffentlichung einer neuen X-CD-Roast Version stattfand. Die Schwere dieser „Umfangreichen Nutzung“ (hier: Bandwidth) wird anhand der zeitlichen Dauer von 34 Stunden deutlich und ist für den Administrator in dieser Tabelle sofort ersichtlich.

Um den Sinn des Eskalationsprinzips der Anpassung der Schranken zu verdeutlichen, wird exemplarisch noch die Eskalation mit der ID 10 angeführt. Die anderen Eskalationen wurden

zwar ebenfalls untersucht, sind aber nicht weiter interessant. Wie sich durch weitere Analysen dieser Eskalation herausstellte, wurde in diesem Zeitraum anscheinend ein kompletter virtueller Webserver gespiegelt und verursachte daher diese Eskalation. Die fünf an der Eskalation beteiligten Anomalien fallen aufgrund des recht großen „Hintergrundrauschens“ der anderen virtuellen Webserver nicht entscheidend ins Gewicht, sodass letztendlich erst durch die zeitliche Korrelation der Anomaliewerte in Verbindung mit der stetigen Reduzierung der Schranken diese Anomalie erkannt wurde. Die fünf berechneten Anomaliewerte der fünf Stunden sind zeitlich geordnet 0,7/0,3/0,3/0,3/0,3 und auf diese folgt ein quasi abschließender Anomaliewert von 0. Nach dem auf Seite 56 beschriebenen Schema wird eine Eskalation erst nach dem vierten Anomaliewert ausgerufen, denn erst dann Überschreiten alle Anomaliewerte die vorgegebene Schranke von 0,3. Der fünfte und letzte Anomaliewerte wird von SNADS richtig der bereits bestehenden Eskalation zugeordnet und trägt zu einer Zunahme der Schwere der bereits gemeldeten Eskalation bei.

Das Eskalationsprinzip zur zeitlichen Korrelation der Datenpunkte funktioniert in der Praxis sehr gut und führt wie oben beschrieben tatsächlich zu einer enormen Arbeitserleichterung. Hierbei wurden bei einem groben Überblick über die gemeldeten Eskalationen keine eindeutigen Anomalien übersehen. Über die Vollständigkeit der Erkennung wird jedoch in dieser Arbeit keine Aussage gemacht.

6 Zusammenfassung

Das Ziel dieser Arbeit war es, ein in der Praxis einsetzbares Anomalie-basiertes Intrusion Detection System zu entwickeln, welches die Datenflüsse innerhalb eines Netzwerks automatisch kontrolliert. Die Praxistauglichkeit bestimmt sich hierbei direkt nach der Qualität der Erkenntnisse und dem Aufwand, die der Administrator bei der nachträglichen Analyse aufbringen muss.

Um diese Ziele zu erfüllen und ein funktionierendes System zu entwerfen, war es daher zunächst erforderlich, die Stärken und Schwächen von Anomalie-basierten Intrusion Detection Systemen zu ergründen. Die Anomalie-basierten Intrusion Detection Systeme vergleichen unter Verwendung von Heuristiken die aktuellen mit den zu erwartenden Charakteristika der Datenflüsse und leiten daraus ein Maß der Diskrepanz her. Die erwarteten Charakteristika werden hierbei aus den bereits aufgetretenen Datenflüssen berechnet. Typische Charakteristika beschreiben Eigenschaften, wie die Verbindungsdauer, das Datentransfervolumen und die Anzahl an Verbindungen, und werden zu Teilen durch die in dieser Arbeit benutzten Anomaliekenngrößen abgebildet. Das zentrale Problem der Anomalie-basierten Intrusion Detection Systeme ist die große Anzahl an Fehlalarmen, da zum einen das Normverhalten aus realen Datenflüssen abgeleitet wird und diese üblicherweise nicht das ganze Spektrum der legitimen Datenflüsse repräsentieren, und zum anderen die Sicherheit der durch Heuristiken gewonnenen Erkenntnisse nicht begründet werden kann.

In der Literatur stehen eine Vielzahl an Algorithmen zur Bewertung einer Anomalie zur Verfügung, weshalb im Folgenden ein Überblick einiger häufig eingesetzter Techniken geliefert wurde. Hierbei wurde in grafische und quantitative Analysetechniken unterschieden, mit dem besonderen Augenmerk auf Robustheit. Die Theorie der robusten Statistik beschäftigt sich mit der Qualität der Ergebnisse von Algorithmen, wenn deren Annahmen, wie beispielsweise die Normalverteilung, nicht gänzlich erfüllt sind oder wenn die Datenreihe von Ausreißern kontaminiert ist. Außerdem liefert sie Aussagen über sinnvollere Techniken, um den Fehler zu reduzieren. Der Einsatz von robusten Algorithmen ist aufgrund unbekannter Charakteristika der Datenflüsse bei der Bewertung von Anomalien von grundlegender Bedeutung. Jedoch leidet im Allgemeinen die Sensitivität der Analyse, sodass die Vollständigkeit der Erkennung bei dem Einsatz von robusten Algorithmen reduziert wird.

Erst nachdem diese Grundlagen untersucht wurden, wurde das Anomalie-basierte Intrusion

Detection System „SNADS“ entworfen. Zunächst wurden Ziele des Systems exakt definiert und das grundlegende Konzept zur Bestimmung einer Anomalie vorgestellt. Der Vorgang lässt sich in mehrere Schritte unterteilen. Im ersten Schritt werden die verbindungsabhängigen Datensätze entgegen genommen, die von BRO – einem Netzwerk basierendem IDS – aus den Datenflüssen extrahiert wurden. Unter Beachtung der Arbeitsanweisungen, die in Regeln und Profilen festgelegt werden, werden die verbindungsabhängigen Datensätze aufbereitet und in Form von Anomaliekenngrößen in Datenbanktabellen abgelegt. Im zweiten Schritt wird basierend auf den Daten unter Verwendung von robusten Heuristiken nach Anomalien gesucht und diese quantitativ bewertet. Der Vorgang wird in der Arbeit Anomaliebewertung genannt. Der letzte Schritt ist die Eskalationsbewertung, die die Schwere der Anomalien unter zeitlichen Gesichtspunkten näher untersucht und zudem versucht, die Ursache des Vorfalls zu konkretisieren. Sie stellt eine Selektion auf den bereits gefundenen Anomalien dar. Mit Anomaliebestimmung wird daher der Vorgang der Anomaliebewertung mitsamt Eskalationsbewertung definiert.

Abschließend wurde das Anomalie-basierte Intrusion Detection System SNADS über zwei Monate an dem Backbone des Internet Service Providers netplace Telematic GmbH betrieben, um Erfahrungen mit dem System zu sammeln und die Sinnhaftigkeit des Konzepts in realer Situation zu bewerten. SNADS setzte zur Anomaliebewertung vier verschiedene Algorithmen ein. In Abhängigkeit der jeweiligen Anomaliekenngröße wurde entweder der „standardized residual“, der „Pearson Chi-square Algorithmus“, die einfache „Schwellwertrechnung“ oder der Algorithmus zum „Vergleich zweier Zahlenreihen“ verwendet werden.

Der Algorithmus zum „Vergleich zweier Zahlenreihen“ wurde lediglich für die Anomaliekenngröße „verwendete Serviceports“ verwendet und vergleicht zwei Listen miteinander, in denen die verwendeten Servicedienste verschiedener Zeiträume stehen. Dieser Algorithmus ist denkbar einfach und lieferte sehr aussagekräftige Informationen über beispielsweise neu installierte Dienste zurück.

Bei der Analyse der Anomaliekenngrößen „Verbindungsdauer eines spezifischen Serviceports“ und „Transfermenge der Verbindungen eines spezifischen Serviceports“ müssen sehr viele Datenpunkte betrachtet werden, weshalb eine Reduktion der Datenmenge erforderlich ist. Der „Pearson Chi-square Anpassungstest“ arbeitet auf in Intervalle eingeteilten Größen, weshalb sich hier die Reduktion durch Kategorisierung in Intervalle anbot. Letztendlich stellte sich jedoch heraus, dass der „Pearson Chi-square Anpassungstest“ in dieser Form aufgrund enormer Schwankungen bei den Anomaliekenngrößen nicht geeignet ist, da in beiden Fällen kein adäquater „Erwartungswert“ bestimmt werden konnte. Die Verbindungsdauer ist hierbei zu stark von der Leitungsqualität abhängig und das übertragene Datenvolumen der einzelnen Verbindungen stark von der Nutzung des jeweiligen Dienstes. Problematisch ist zudem, dass diese Anomaliekenngrößen Häufungspunkte an den Enden des Wertebereichs aufwiesen und

daher die Annahme des „Pearson Chi-square Algorithmus“, eine Gleichverteilung der Trefferanzahl in den Intervallen anzustreben, nur äußerst schlecht erfüllt war. Da in diesem Fall der „Pearson Chi-square“-Wert der χ^2 -Verteilung nicht folgt, sollten die Erkenntnisse des Tests ohnehin mit Skepsis betrachtet werden. In anderen Szenarien oder auch für weitere Anomaliekenngrößen gleichen Typs könnte der Pearson Chi-square Algorithmus dennoch praktikabel sein.

Für die anderen Anomaliekenngrößen, wie beispielsweise „Anzahl an Verbindungen“, wurde der sehr robuste „standardized residual“ eingesetzt und zeigte sich bei der Anomaliebewertung als sehr gut geeignet. Er ermöglicht einen Vergleich von Datenreihen auch dann, wenn sie von Ausreißern gespickt sind. Die Sensitivität kann dabei je nachdem durch die Wahl der Parameter des Algorithmus festgelegt werden und ermöglicht eine individuelle Analyse. Zudem lieferte er aufgrund seiner robusten Eigenschaften auch bei schwankenden Datenreihen gute Ergebnisse. Defizite zeigten sich lediglich bei sehr sporadischer Nutzung eines Dienstes. Dieses Problem bei der Analyse konnte jedoch durch die Wahl eines größeren Betrachtungszeitraums erheblich reduziert werden. Die Vollständigkeit bei der Erkennung von Anomalien reduziert sich dann aber ebenfalls.

Ist die Vergrößerung des Betrachtungszeitraums nicht gewünscht, bietet sich als Alternative auch die ebenfalls gänzlich realisierte Schwellwertrechnung an. Bei dieser einfachen Methode definiert der Administrator feste Schranken, die für den Fall, dass sie unter- oder überschritten werden, eine Anomalie vermuten lassen. Die Methode ist, da der Administrator die Schranken selber bestimmen muss, wesentlich aufwendiger in der Administration, liefert aber dafür stets die gewünschten Ergebnisse.

Prinzipiell ist die abschließende Eskalationsbewertung eine Selektion der zuvor berechneten Anomalien nach ihrer Schwere, die sich aus der Heftigkeit und der zeitlichen Dauer der Anomalie bestimmt. Zudem wird den Anomalien gegebenenfalls eine Eskalationsursache, wie „Denial of Service“, „Port Scan“ und „Umfangreiche Nutzung“ zugewiesen. Durch die Eskalationsbewertung wurde eine wesentlich effizientere Analyse der Anomalien ermöglicht, und das führte letztendlich zu einer enormen Arbeitserleichterung bei der nachträglichen Analyse. Hierbei darf jedoch nicht unterschlagen werden, dass durch diesen Schritt die Vollständigkeit der Erkennung von Angriffen reduziert wird. Es bleibt jedoch dem Administrator überlassen, welcher Kompromiss zwischen nachträglichem Aufwand und Vollständigkeit der Erkennung gewählt wird. Die Zuweisung der Eskalationsursachen funktioniert ebenfalls relativ gut, obwohl das Konzept in diesem Bereich noch nicht gänzlich realisiert wurde. Die wenigen Fehler sollten jedoch bei vollständiger Realisierung nicht mehr auftreten.

Indem SNADS die Daten in einer Datenbank ablegt, können Einzelheiten eines Angriffs relativ leicht im nachhinein geprüft werden. Somit trägt das System auch zum direkten Erfahrungsgewinn bei und erlaubt die weitere Optimierung des Systems anhand der Erfahrungen.

Der funktionsfähige Prototyp SNADS demonstriert, dass auch Anomalie-basierte Intrusion Detection Systeme durch geeignete Mittel, wie die vorgestellten robusten Statistiken und die Eskalationsbewertung, in der Praxis sinnvoll einsetzbar sind.

Die Weiterentwicklung des Anomalie-basierenden Intrusion Detection Systems SNADS wird nach der Diplomarbeit fortgesetzt mit dem genannten Ziel, ein in der Praxis recht allgemein einsetzbares Anomalie-basiertes System zu entwickeln. Allen voran sollen Lösungen zu den genannten Problemen gesucht und die einzelnen Schritte der Anomaliebewertung verbessert werden.

Literaturverzeichnis

- [1] *CERT Coordination Center*. http://www.cert.org/stats/cert_stats.html.
- [2] AXELSSON, STEFAN: *The base-rate fallacy and the difficulty of intrusion detection*. ACM Transactions on Information and System Security, 3(3):186–205, August 2000.
- [3] DONN, SEELY: *A tour of the worm*. IEEE/ACM Transactions on Networking, November 1991.
- [4] *W32/SQLSlammer*. http://vil.nai.com/vil/content/v_99992.htm.
- [5] AXELSSON, STEFAN: *Intrusion Detection Systems: A Survey and Taxonomy*. Technischer Bericht 99-15, Department of Computer Engineering, Chalmers University of Technology, Sweden, 2000.
- [6] NOEL S., WIJESKERA D., YOU MAN C.: *Modern Intrusion Detection, Data Mining, and Degrees of Attack Guilt, in Applications of Data Mining in Computer Security*. Technischer Bericht, George Mason University, Fairfax, USA, 2002.
- [7] HERVÉ DEBAR, MARC DACIER und ANDREAS WESPI: *A Revised Taxonomy for Intrusion Detection Systems*. Annales des Telecommunications, 55(7-8):361–378, 2000.
- [8] STEVEN R. SNAPP, JAMES BRENTANO, GIHAN V. DIAS TERRANCE L. GOAN L. TODD HEBERLEIN CHE-LIN HO KARL N. LEVITT BISWANATH MUKHERJEE STEPHEN E. SMAHA TIM GRANCE DANIEL M. TEAL und DOUG MANSUR: *DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and an early Prototype*. In: *Proceedings of the 14th National Computer Security Conference*, Seiten 167–176, Washington, DC, 1991.
- [9] KO, CALVIN, MANFRED RUSCHITZKA und KARL LEVITT: *Execution Monitoring of Security-critical Programs in Distributed Systems: A Specification-based Approach*. In: *Proceedings of the 1997 IEEE Symposium on Security and Privacy (SSP)*, Seiten 175–187, Oakland, CA, Mai 1997. IEEE Computer Society Press.

- [10] JACKSON, KATHLEEN: *Intrusion Detection System (IDS) Product Survey*. Technischer Bericht LA-UR-99-3883, Los Alamos National Laboratory, Juni 1999.
- [11] ALLEN, JULIA, ALAN CHRISTIE, WILLIAM FITHEN, JOHN MCHUGH, JED PICKEL und ED STONER: *State of the Practice of Intrusion Detection Technologies*. Technischer Bericht CMU/SEI-99-TR-028, Carnegie Mellon University, Januar 2000.
- [12] AXELSSON, STEFAN: *Intrusion Detection Systems: A Survey*. Technischer Bericht 98-17, Department of Computer Engineering, Chalmers University of Technology, Sweden, 1999.
- [13] KVARNSTRÖM, H.: *A survey of commercial tools for intrusion detection*. Technischer Bericht 99-8, Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden, Oktober 1999.
- [14] L., LAPADULA: *State of the art in anomaly detection and reaction*. Technischer Bericht, The MITRE Corporation, Bedford, MA, 1999.
- [15] L., LAPADULA: *Compendium of anomaly detection and reaction tools and projects*. Technischer Bericht, The MITRE Corporation, Bedford, MA, 2000.
- [16] JULISCH, KLAUS und MARC DACIER: *Mining Intrusion Detection Alarms for Actionable Knowledge*. In: HAND, DAVID, DANIEL KEIM und RAYMOND NG (Herausgeber): *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*, Seiten 366–375. ACM Press, Juli 23–26.
- [17] ROESCH, MARTIN: *Snort – Lightweight Intrusion Detection for Networks*. In: *Proceedings of the 13th Conference on Systems Administration (LISA) 1999*, Seiten 229–238, Berkeley, CA, November 7–12 1999. USENIX Association.
- [18] PAXSON, VERN: *Bro: A System for Detecting Network Intruders in Real-Time*. *Computer Networks*, 31(23–24):2435–2463, 1999.
- [19] LIEPINS G. E., VACCARO H. S.: *Anomaly detection purpose and framework*. In: *Proc. of the 12th National Computer Security Conference, Baltimore, MD*, Seiten 495–504, Oktober 1989.
- [20] STANIFORD-CHEN, S., S. CHEUNG, R. CRAWFORD, M. DILGER, J. FRANK, J. HOAGLAND, K. LEVITT, C. WEE, R. YIP und D. ZERKLE: *GrIDS – A Graph-based Intrusion Detection System for Large Networks*. In: *Proceedings of the 19th NIST-NCSC National Information Systems Security Conference*, 1996.
- [21] MARK CROSBIE, BRYN DOLE, TODD ELLIS IVAN KRSUL und EUGENE SPAFFORD: *IDIOT - User Guide*. Technischer Bericht TR-96-050, Purdue University, West Lafayette, IN, US, September 1996.

- [22] KORAL ILGUN, RICHARD A. KEMMERER und PHILLIP A. PORRAS: *State Transition Analysis: A Rule-Based Intrusion Detection Approach*. Software Engineering, 21(3):181–199, 1995.
- [23] LINDQVIST, ULF und PHILLIP A. PORRAS: *Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST)*. In: *Proceedings of the 1999 IEEE Symposium on Security and Privacy (SSP)*, Seiten 146–161, Oakland, California, Mai 1999. IEEE Computer Society Press.
- [24] PORRAS, PHILLIP A. und PETER G. NEUMANN: *EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances*. In: *Proceedings of the 20th NIST-NCSC National Information Systems Security Conference*, Baltimore, MD, Oktober 1997.
- [25] JAVITZ, HAROLD S. und ALFONSO VALDES: *The NIDES Statistical Component: Description and Justification*. Technischer Bericht, SRI International, März 1993.
- [26] WENKE LEE, SALVATORE J. STOLFO, KUI W. MOK: *Adaptive Intrusion Detection: A Data Mining Approach*. Artificial Intelligence Review, 14(6):533–567, 2000.
- [27] GHOSH A., SCHWARTZBARD A.: *A study in using neural networks for anomaly and misuse detection*. In: *Proceedings of the 8th USENIX Security Symposium*, Seiten 141–151. USENIX Association, 1999.
- [28] ZHANG, YIN: *A Hierarchical Anomaly Network Intrusion Detection System Using Neural Network Classification*. In: *Proc. of the 2001 WSES International Conference on: Neural Networks and Applications*, 2001.
- [29] FORREST S., HOFMEYR S., SOMAYAJI A. und LONGSTAFF T.: *A sense of self for unix processes*. In: *IEEE Symposium on Security and Privacy*, Seiten 120–128. IEEE Computer Society Press, 1996.
- [30] ANIL SOMAYAJI, STEVEN HOFMEYR und STEPHANIE FORREST: *Principles of a Computer Immune System*. In: *Meeting on New Security Paradigms, 23-26 Sept. 1997, Langdale, UK*, Seiten 75–82. New York, NY, USA : ACM, 1998.
- [31] SOMMER, ROBIN: *Bro: An Open Source Network Intrusion Detection System*. In: *17. DFN-Arbeitstagung über Kommunikationsnetze*, 2003.
- [32] SCHEPERS, HEINRICH: *Historisches Wörterbuch der Philosophie, Band 3*, 1974.
- [33] D’AGOSTINO R. B., STEPHENS M. A.: *Goodness-of-fit techniques*, 1986.
- [34] SHAUN, BURKE: *Understanding the Structure of Scientific Data*. LCGC Europe, Januar 2001.

- [35] CHAMBERS J., CLEVELAND W., KLEINER B. AND TUKEY P.: *Graphical Methods for Data Analysis*, 1983.
- [36] *e-Handbook of Statistical Methods*. <http://www.itl.nist.gov/div898/handbook/>.
- [37] STEGER ANGELIKA, SCHICKINGER THOMAS: *Skriptum zur Vorlesung: Diskrete Strukturen II*. April 2000.
- [38] SHAUN, BURKE: *Analysis of Variance*. LCGC Europe, Januar 2001.
- [39] SHAUN, BURKE: *Missing Values, Outliers, Robust Statistics and Non-parametric Methods*. LCGC Europe, Januar 2001.
- [40] LARNTZ, KINLEY: *Small-sample Comparisons of Exact Levels for Chi-squared Goodness-of-fit Statistics*. Journal of the American Statistical Association, 73:253–263, 1978.
- [41] ASLAN, B. und G. ZECH: *Comparison of different goodness-of-fit tests*. 2002.
- [42] J. W. TUKEY: *A survey of sampling from contaminated distributions*. Contributions to Probability and Statistics, Seiten 448–485, 1960.
- [43] F.R. HAMPEL, E.M. RONCHETTI, P.J. ROUSSEEUW, AND W.A. STAHEL: *Robust Statistics: The Approach Based on Influence Functions*, 1986.
- [44] RANUM, MARCUS J.: *Experiences Benchmarking Intrusion Detection Systems*. Technischer Bericht, NFR Security, Inc., Dezember 2001.
- [45] ESKIN, ELEAZAR: *Anomaly Detection over Noisy Data using Learned Probability Distributions*. In: *Proc. 17th International Conf. on Machine Learning*, Seiten 255–262. Morgan Kaufmann, San Francisco, CA, 2000.
- [46] NEUMANN, PETER G. und PHILLIP A. PORRAS: *Experience with EMERALD to Date*. In: *Proceedings of the Workshop on Intrusion Detection and Network Monitoring*, Seiten 73–80, Santa Clara, California, April 1999.
- [47] W. H. PRESS, B. P. FLANNERY, S. A. TEUKOLSKY und W. T. VETTERLING: *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1993.
- [48] ZHENG ZHANG, JUN LI, C.N. MANIKOPOULOS, JAY JORGENSON AND JOSE UCLES: *HIDE: a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification*. In: *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security* , Seiten 16–28, West Point, NY, 2001.

- [49] *Architecture of Generalized Network Service Anomaly and Fault Thresholds*. In: ZHENG ZHANG, CONSTANTINE N. MANIKOPOULOS, JAY JORGENSON (Herausgeber): *4th IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, Band 2216 der Reihe *Lecture Notes in Computer Science*, Seiten 241–255, Chicago, IL, USA, 2001. Springer.
- [50] IGLEWICZ, B. und D. HOAGLIN: *How to Detect and Handle Outliers*, 1993.
- [51] *Sleepycat Software: Berkeley DB*. <http://www.sleepycat.com/>.
- [52] *Berkeley DB: Dokumentation*. <http://www.sleepycat.com/docs/index.html>.
- [53] *GD Graphics Library*. <http://www.boutell.com/gd/>.
- [54] *Apache Software Foundation*. <http://www.apache.org/>.
- [55] *The GNU Project and the Free Software Foundation*. <http://www.gnu.org/>.
- [56] *netplace Telematic GmbH*. <http://www.netplace.de/>.
- [57] *X-CD-Roast*. <http://www.xcdroast.de/>.
- [58] GREINER, M. und G. TINHOFER: *Stochastik für Studienanfänger der Informatik*, 1996.
- [59] KUDYACHETE: *Intrusion Detection*, 2002.
- [60] SHAUN, BURKE: *Regression and Calibration*. LCGC Europe, Januar 2001.
- [61] WILLIAM S. CLEVELAND AND DON X. SUN: *Internet Traffic Data*. *Journal of the American Statistical Association*, 95:979–985, 2000.
- [62] FRANK, JEREMY: *Artificial Intelligence and Intrusion Detection: Current and Future Directions*. In: *Proceedings of the 17th National Computer Security Conference*, Baltimore, MD, 1994.
- [63] GONZALEZ, F. und D. DASGUPTA: *An Immunogenetic Technique to Detect Anomalies in Network Traffic*. In: *Proceedings of the International Conference Genetic and Evolutionary Computation*, New York, 2002.
- [64] DUNIGAN, TOM und GEORGE OSTROUCHOV: *Flow Characterization for Intrusion Detection*. Technischer Bericht 11/27/00, Oak Ridge National Laboratory, November 2000.
- [65] MAXION, ROY A. und KYMIE M.C. TAN: *Benchmarking Anomaly-Based Detection Systems*. In: *Proceedings of the 1st International Conference on Dependable Systems & Networks 2000*, 2000.

- [66] MICHAEL MIROLD: *Systementwicklungsprojekt: Experiences with Bro*, Oktober 2002.
- [67] PORRAS, PHILLIP A. und ALFONSO VALDES: *Live Traffic Analysis of TCP/IP Gateways*. In: *Proceedings of the ISOC Symposium on Network and Distributed System Security 1998*, 1998.
- [68] ANDERSON, DEBRA, TERESA LUNT, HAROLD JAVITZ, ANN TAMARU und ALFONSO VALDES: *Safeguard Final Report: Detecting Unusual Program Behavior Using the NIDES Statistical Component*. Technischer Bericht, SRI International, 1995.
- [69] JAVITZ, H. S. und A. VALDES: *The SRI IDES Statistical Anomaly Detector*. In: *Proc. IEEE Symposium on Research in Security and Privacy*, Seiten 316–326, 1991.
- [70] LUNDIN, EMILIE und ERLAND JONSSON: *Some Practical and Fundamental Problems with Anomaly Detection*. In: *Proceedings of the 4th Nordic Workshop on Secure IT systems (NORDSEC)*, 1999.
- [71] TOELLE, J. und O. NIGGEMANN: *Supporting Intrusion Detection by Graph Clustering and Graph Drawing*. In: *Third International Workshop on the Recent Advances in Intrusion Detection*, 2000.
- [72] LICHODZIJEWSKI P., ZINCIR-HEYWOOD A.N. und HEYWOOD M.I.: *Dynamic Intrusion Detection Using Self Organizing Maps*. In: *14th Annual Canadian Information Technology Security Symposium*, Mai 2002.
- [73] MAHONEY, MATTHEW V. und PHILLIP K. CHAN: *Detecting Novel Attacks by Identifying Anomalous Network Packet Headers*. Technischer Bericht CS-2001-2, Florida Institute of Technology, Melbourne, 2001.
- [74] DICKERSON, J.E. und J.A. DICKERSON: *Fuzzy Network Profiling for Intrusion Detection*. In: *Proceedings of NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society*, Atlanta, Juli 2000.
- [75] BARFORD, PAUL und DAVE PLONKA: *Characteristics of Network Traffic Flow Anomalies*. In: *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW) 2001*, 2001.
- [76] LEMONNIER, ERWAN: *Protocol Anomaly Detection in Network-based IDSs*. In: *Defence Communications 2001*, Stockholm, Sweden, 2001.
- [77] ESKIN, ELEAZAR: *Anomaly Detection over Noisy Data using Learned Probability Distributions*. In: *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

- [78] NONG YE, QIANG CHEN, SYED MASUM EMRAN und KYUTAE NOH: *Chi-square Statistical Profiling for Anomaly Detection*. In: *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop June 6-7, 2000 at West Point, New York*, Seiten 187–193, Juni 2000.
- [79] NONG YE, QIANG CHEN, SYED MASUM EMRAN und SEAN VILBERT: *Multivariate Profiling For Hotelling's T*. In: *Proceedings of the 2000 IEEE Workshop on Information Assurance and Security*, 2000.
- [80] LUCA DERI, STEFANO SUIN und GAIA MASELLI: *Design and Implementation of an Anomaly Detection System: an Empirical Approach*, Mai 2003.
- [81] ABRAHAM, TAMAS: *IDDM: Intrusion Detection using Data Mining Techniques*. 2000.
- [82] WENKE LEE, SALVATORE J. STOLFO und KUI W. MOK: *A Data Mining Framework for Building Intrusion Detection Models*. In: *IEEE Symposium on Security and Privacy*, Seiten 120–132, 1999.
- [83] GEORGE E. NOEL, STEVEN C. GUSTAFSON, GREGG H. GUNSCH: *Network-Based Anomaly Detection Using Discriminant Analysis*. *Journal of Information Warfare*, 2001.
- [84] GIACINTO, G. und F. ROLI: *Intrusion Detection in Computer Networks by Multiple Classifier Systems*. In: *16th International Conference on Pattern Recognition*, 2002.
- [85] PORRAS, PHILLIP A. und PETER G. NEUMANN: *EMERALD: Conceptual Overview Statement*. Technischer Bericht, SRI International, Dezember 1996.
- [86] VALDES, ALFONSO und KEITH SKINNER: *Probabilistic Alert Correlation*. In: *Recent Advances in Intrusion Detection*, Nummer 2212 in *Lecture Notes in Computer Science*. Springer-Verlag, 2001.
- [87] LEE, WENKE und DONG XIANG: *Information-Theoretic Measures for Anomaly Detection*. In: *Proceedings of the 2001 IEEE Symposium on Security and Privacy (SSP)*, Mai 2001.
- [88] YE, NONG: *A markov chain model of temporal behavior for anomaly detection*. In: *Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, 2000*, Seiten 171–174, 2000.
- [89] MINGMING, NONG YE: *Probabilistic Networks with Undirected Links for Anomaly Detection*. In: *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop June 6-7, 2000 at West Point, New York*, Seiten 175–179, Juni 2000.

- [90] NONG YE, XIANGYANG LI und SYED MASUM EMRAN: *Decision Tree for Signature Recognition and State Classification*. In: *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop June 6-7, 2000 at West Point, New York*, Seiten 194–199, Juni 2000.
- [91] YE, NONG und Q. CHEN: *An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems*. In: *Quality and Reliability Engineering International*, 2001.
- [92] MEHTA, C.R. und N.R. PATEL: *Exact inference for categorical data*. *Encyclopedia of Biostatistics*. Vol. 2, 1998.