Ein föderiertes Intrusion Detection System für Grids

Dissertation

an der

Fakultät für Mathematik, Informatik und Statistik
der
Ludwig-Maximilians-Universität München

vorgelegt von

Nils Otto vor dem gentschen Felde

Tag der Einreichung: 12. November 2008

Ein föderiertes Intrusion Detection System für Grids

Dissertation

an der

Fakultät für Mathematik, Informatik und Statistik
der
Ludwig-Maximilians-Universität München

vorgelegt von

Nils Otto vor dem gentschen Felde

Tag der Einreichung: 12. November 2008
Tag des Rigorosums: 17. Dezember 2008

1. Berichterstatter: **Prof. Dr. Heinz-Gerd Hegering**, Universität München
2. Berichterstatter: **Prof. Dr. Uwe Baumgarten**, Technische Universität München

„Ich bin bereit überall hinzugehen,
wenn es nur vorwärts ist."
David Livingstone
(britischer Forschungsreisender, 1813-1873)

Danksagung

Im Zuge der Anfertigung meiner Dissertation standen mir eine Menge Freunde und Kollegen stets hilfreich zur Seite, sei es in inhaltlicher oder motivierender Sache. All diesen Menschen möchte ich an dieser Stelle meinen herzlichen Dank ausdrücken! Mein ganz besonderer Dank gilt meinem Doktorvater, Prof. Dr. Heinz-Gerd Hegering. Mit großem persönlichen Einsatz stand er mir zu jeder Tages- und Nachtzeit mit Rat und Tat zur Seite und unterstützte mich mit bemerkenswerter Hingabe. Auch für die Unterstützung durch meinen zweiten Berichterstatter, Prof. Dr. Uwe Baumgarten, möchte ich mich ganz herzlich bedanken.

Diese Dissertation basiert maßgeblich auf den Gedanken und Diskussionsergebnissen einer jahrelangen Zusammenarbeit in einem großartigen Team von Wissenschaftlern. Insbesondere die Einbettung in ein wissenschaftlich hervorragendes Umfeld als Mitglied des MNM-Teams, das durch Professor Hegering geleitet wird, hat entscheidende Beiträge zum Erfolg dieses Werkes geleistet. So möchte ich mich ganz herzlich für die Geduld, Möglichkeit zur Diskussion und auch Motivation zum Durchhalten bei allen (ehemaligen) Kollegen und Mitglieder des MNM-Teams bedanken. Ohne eure Anteilnahme und Hilfestellung wäre mir diese Leistung nicht möglich gewesen. Besonders hervorzuheben sind dabei die Beiträge von Michael Schiffers, Vitalian Danciu und Helmut Reiser, die stets ein offenes Ohr und zielführende Anregungen für mich hatten.

Schlussendlich gilt mein Dank meiner Familie und all meinen Freunden, die mir hilfreich in den letzten Jahren zur Seite standen. Auch eure Unterstützung hat entscheidend zum Erfolg dieser Arbeit beigetragen.

November 2008, München

Zusammenfassung

Durch die bisher vorwiegend wissenschaftliche, und deswegen meist freizügige Nutzung von Grid-Infrastrukturen sind vor allem Aspekte der Überwachung, des Reportings und der Auditierbarkeit von Angriffen oder anderen Sicherheitsverletzungen in Grids vernachlässigt worden. Im Rahmen dieser Arbeit wird deswegen ein föderiertes Intrusion Detection System für Grids (GIDS) entwickelt, um eine weiterreichende Nutzung, auch im wirtschaftlichen und industriellen Umfeld, zu ermöglichen.

Eine große Herausforderung im Grid-Umfeld ist die Kooperation zwischen autonomen Partnern, die gemeinsam ein Sicherheitswerkzeug, hier in Form eines Intrusion Detection Systems, betreiben möchten. Die Kooperationspartner sind dabei in der Regel nicht nur organisatorisch autonom in Form von verschiedenen realen Organisationen, sie sind auch häufig geographisch getrennt, verfolgen aber dennoch ein kooperatives Koordinationsmuster.

Dadurch und durch neue durch das Grid bedingte Objekte, die bei konventionellen vernetzten Systemen in der Regel nicht in vergleichbarer Weise existierten, wie zum Beispiel Virtuelle Organisationen (VOs), erwachsen zahlreiche bisher ungelöste Probleme hinsichtlich der Klassifikation von Angriffsszenarien in Grids, der Kriterien zur Bestimmung der Eignung eines IDS für den Einsatz in Grids, der Adäquatheit von Architekturen eines Grid-Frühwarnsystems oder dessen Skalierbarkeit.

Zur Lösung dieser und weiterer Randfragestellungen wird im Rahmen der Arbeit eine Klassifikation von Angriffsszenarien in Grids als Teildisziplin des Security Engineerings vorgenommen. Damit wird die Erstellung eines Kriterienkatalogs für die Bewertung und Auswahl von IDS für Grids möglich. Die Anwendbarkeit des Kriterienkatalogs wird am Beispiel D-Grid demonstriert. Auf der Basis des Katalogs sowie Erkenntnissen, die aus bestehenden, themenverwandten Forschungsansätzen zu IDS für Grids abgeleitet werden können, wird eine generische Architektur für ein GIDS konzipiert. Abschließend wird eine prototypische Implementierung des entwickelten GIDS vorgestellt und ein Tragfähigkeitsnachweis in Form von Labormessungen und einem Testeinsatz im Münchner Wissenschafts-Netz (MWN) geführt. Ein Abgleich des Prototypen mit dem zuvor erhobenen Kriterienkatalog beschließt diesen Teil, bevor eine Zusammenfassung und ein Ausblick auf weiterführende Forschungsfragestellungen die Arbeit beschließen.

Abstract

Due to the mostly scientific and thus generous use of Grid-infrastructures, aspects of surveillance and reporting on security as well as auditing capabilities in Grids have been neglected up to now. Therefore, this work aims at designing a federated Intrusion Detection System for Grids (GIDS), e.g. to support a wider application of Grids in commercial and industrial environments.

But the cooperation of autonomous partners providing a cooperative security measure in form of an Intrusion Detection System poses great challenges, especially in Grids. The cooperating partners are usually not only autonomous with respect to different real organizations, but frequently also geographically separated. Anyways, they pursuit a common goal to form a cooperative GIDS.

Numerous, but yet unsolved challenges arise due to these facts and due to newly introduced, Grid-specific objects, that do not exist in conventionally networked environments, e.g. Virtual Organizations (VOs) to name one. Up to date research lacks a classification of attack scenarios in Grids, criteria to judge the applicability of an IDS for the Grid as well as the adequacy of architectures of IDS for Grids and their scalability.

To cope with these and other challenges, first a classification of attack scenarios with respect to the Grid is conducted in accordance to security engineering procedures. The results lead to the composition of a catalogue of criteria for the evaluation and selection of IDS for their use in the Grid. The catalogue's applicability is shown on the basis of the D-Grid. A generic architecture of an GIDS is then composed according to the before mentioned catalogue of criteria and the knowledge about related work within the area of IDS, especially those for Grids. Following, a prototypical implementation is presented to investigate on the skills of the concept. Measurements under lab conditions as well as a testing application in the Munich network of sciences (Münchener Wissenschafts-Netz, MWN) are conducted on basis of this implementation. Matching the prototypical implementation with the catalogue of criteria yields completion to the work, before a summary and outlook to further scientific challenges in this area concludes the work.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Hinter dem Begriff des Grid-Computing steckt die zukunftsweisende Idee, bekannte Konzepte des verteilten und Hochleistungsrechnens zu erweitern, mit dem Ziel, eine neue Infrastruktur bereitzustellen, die Anwendern einen einfachen, transparenten Zugriff auf einen Pool von weltweit über viele Organisationen verteilten Ressourcen ermöglicht. Die Nutzung von Rechen- und Speicherkapazitäten soll dabei ähnlich einfach möglich sein wie etwa der Bezug von Elektrizität aus dem Stromnetz. In diesem Zusammenhang ist auch der Begriff *Grid* geprägt worden, da eine Analogie zu einem Stromverbreitungsnetz gesucht wurde. Zumeist steht dabei der Gedanke des Beziehens von Rechen- und Speicherkapazitäten „aus der Steckdose“ im Vordergrund. Inzwischen wird der Grid-Ressourcenbegriff erweitert auch auf Experimentgeräte, Daten, Anwendungen und IT-Services angewendet.

Ein komplexes Beispiel für ein Grid-Computing Szenario bietet das D-Grid. Innerhalb des D-Grids existieren verschiedene sogenannte *Communities*. Jede der Communities stellt dem Grid eine Vielzahl an realen und virtuellen Ressourcen zur Verfügung. Das kann sowohl Rechenkapazität auf diversen Rechnerarchitekturen wie auch Speicherplatz auf verschiedensten Speichersystemen oder die Bereitstellung unterschiedlicher Dienste sein. Zusätzlich stellt das Kern D-Grid, bestehend aus einer Vielzahl beteiligter Parteien inklusive der größten Rechenzentren Deutschlands, diverse Ressourcen bereit. In den einzelnen Communities sind verschiedene Projekte angesiedelt, in denen jeweils eine Vielzahl an Projektteilnehmern arbeiten, die sowohl die vorhandenen Ressourcen der eigenen Community als auch die zur Verfügung gestellten Ressourcen anderer Communities in Anspruch nehmen wollen.

Im Umfeld von Grids ergeben sich im Vergleich zu konventionellen vernetzten Systemen eine Reihe bisher ungelöster Probleme: So begegnet man u.a. einem sehr dynamischen Umfeld. Dabei ist eine Dynamik unter verschiedenen Gesichtspunkten festzustellen, wie z.B. eine hohe Dynamik an verfügbaren Ressourcen oder auch hoch dynamische Nutzergruppen bzw. *Virtuelle Organisationen* (VO). Dies erfordert individuelle, dynamische Nutzersichten, die sich in den Kontext einer VO einbetten und deren individuellen Bedürfnissen nachkommen. Weiter ergibt sich ein Grid-typisch heterogenes Umfeld. Auch dieses existiert auf mehreren Ebenen und ist u.a. auch im Bereich der Ressourcen, der eingesetzten Grid-Middleware oder der verwendeten Grid-Monitoring Komponenten zu beobachten. Nicht zuletzt sind die zum Teil bereits von den jeweilig beteiligten realen und autonom agieren-

den Organisationen eingesetzten Sicherheitskomponenten und -werkzeuge zur Erkennung und Prävention von Angriffen unterschiedlichster Art.

Im Kontext von Grids existieren bereits bekannte Mechanismen zur Gewährleistung autorisierter Zugriffe, der Überprüfung von Identitäten (Authentifizierung), der Zugriffskontrolle, von Vertraulichkeit und Datenintegrität; teilweise werden diese zurzeit entwickelt und verwirklicht. Eine Lücke tut sich jedoch bei der Überwachung von Grids auf Sicherheitsangriffe und bei der Berichterstattung zur Sicherheit im laufenden Betrieb auf. Ohne die Gewährleistung aller Teildisziplinen des Sicherheitsmanagements ist das Bilden einer Vertrauensbasis in eine Grid-Infrastruktur jedoch nicht möglich. Aus diesem Grund zielt diese Arbeit darauf ab, *ein Frühwarnsystem für Grid-Umgebungen* zu entwickeln und in einem Grid zu etablieren.

Unter einem Frühwarnsystem oder auch *Intrusion Detection System* (IDS) wird die aktive und passive Überwachung von Computersystemen sowie Computernetzen bzw. Netzsegmenten mit dem Ziel der Erkennung von Angriffen und Missbrauch bezeichnet [BSI, 2002]. Das Ziel besteht dabei darin, aus den anfallenden Ereignissen und dazu aufgezeichneten Informationen diejenigen herauszufiltern bzw. angemessen zu aggregieren und korrelieren, die geeignet sind um Hinweise auf Angriffe, Missbräuche und/oder Sicherheitsverletzungen zu erhalten. Sämtliche Meldungen sollen dabei zeitnah geschehen. Die Frühwarnung im Umfeld von Computernetzen und verteilten Systemen ist als Prozess zu betrachten. Es bedarf einer geeigneten organisatorischen Einbindung sowie der technischen Unterstützung durch geeignete Management-Werkzeuge.

Zurzeit gibt es kein Gesamtkonzept für ein kooperatives Grid-weites Frühwarnsystem mit entsprechenden Reporting-Komponenten. Aus diesem Grund soll in dieser Arbeit ein Konzept für ein System zur Überwachung des Grids auf Sicherheitsangriffe sowie zur Berichterstattung entwickelt und prototypisch implementiert werden. Als Lösungsansatz wird die Idee der kooperativen Nutzung von lokalen Sicherheitssystemen und der Austausch von Angriffsdaten verfolgt. Dabei ist die datenschutzkonforme Aufarbeitung und Bereinigung von Log-Dateien unter Wahrung individuell bestehender Sicherheits- und Informationsverbreitungsrichtlinien mit entscheidend für den Erfolg des entwickelten Konzepts. In einem kooperativen Frühwarnsystem besteht die Möglichkeit Angriffe schneller zu erkennen als dies mit unabhängigen lokalen Sicherheitssystemen möglich ist. Somit kann eine Verkürzung der Reaktionszeit der beteiligten Parteien erzielt werden. Weiter können Vorwarnungen an zum Zeitpunkt der Erkennung eines Angriffs noch nicht betroffene Parteien ausgerufen sowie ggf. präventive Gegenmaßnahmen ergriffen werden.

Die weitere Struktur dieser Arbeit ergibt sich aus den Teilfragestellungen der Problematik und gliedert sich wie auch in Abbildung 1.1 dargestellt. Im folgenden Kapitel 2 wird die für die Arbeit entscheidende Begriffsbildung vorgenommen sowie in Kapitel 3 eine Anforderungsanalyse zur Gewinnung eines Kriterienkatalogs für die Bewertung und Auswahl von IDS im Grid-Umfeld durchgeführt. In Kapitel 4 werden anschließend bestehende Systeme sowie Forschungs- und Industrieansätze näher beleuchtet und auf ihre Defizite hin untersucht. In Kapitel 5 wird dann eine Architektur für ein Frühwarnsystem in

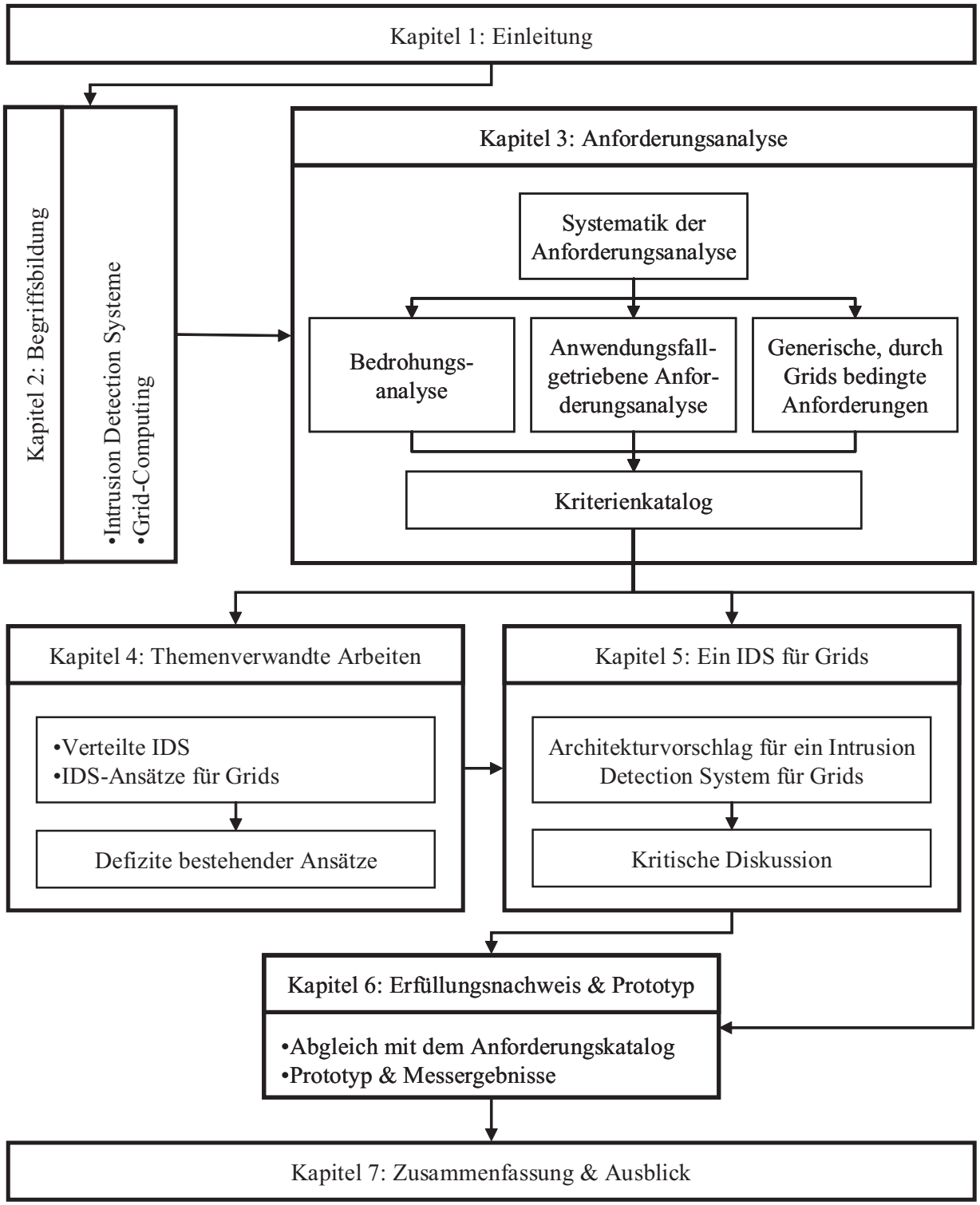


Abbildung 1.1: Vorgehensmodell der Arbeit

Grid-Umgebungen vorgeschlagen. Die Architektur wird zum einen formal beschrieben, zum anderen werden die erforderlichen Dienste und Komponenten detailliert vorgestellt. Eine kritische Diskussion und ein Abgleich mit dem in Kapitel 3 erarbeiteten Anforderungskatalog erfolgt in Kapitel 6. Zudem wird eine prototypische Implementierung vorgestellt und ein Tragfähigkeits- und Leistungsfähigkeitsnachweis des vorgeschlagenen Konzepts durchgeführt. Abschließend fasst Kapitel 7 die Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick auf mögliche weitere Anschlussarbeiten sowie wissenschaftliche Fragestellungen, die zur Laufzeit dieser Arbeit zu Tage getreten sind.

Kapitel 2

Begriffsbildung

Inhalt des Kapitels

In diesem Kapitel soll zuerst in den Abschnitten 2.1 und 2.2 die für die Ableitung von Anforderungen an und das Verständnis für Frühwarnsysteme in Grid-Umgebungen notwendige Begriffsbildung vorgenommen werden. Dazu wird Abschnitt 2.1 den Begriff der klassischen *Intrusion Detection Systeme* genauer beleuchten und Abschnitt 2.2 die Konzepte und Begriffe im Rahmen des *Grid-Computing* erörtern. Darauf aufbauend wird in

dem folgenden Kapitel 3 eine Anforderungsanalyse zur Bestimmung von Anforderungen an Frühwarnsysteme in Grid-Umgebungen durchgeführt, die schlussendlich in einem in Abschnitt 3.4 zusammengefassten Kriterienkatalog für die Bewertung und Auswahl von IDS im Grid-Umfeld mündet.

2.1 Intrusion Detection Systeme

Intrusion Detection Systeme (IDS) oder auch *Frühwarnsysteme*[1] sind eine Möglichkeit, die Sicherheit eines Computersystems zusätzlich zu anderen Sicherheitsmechanismen wie zum Beispiel Firewalls zu erhöhen und sind heutzutage ein wichtiger Bestandteil moderner Netze. Ein IDS hat die Aufgabe, das zu beobachtende und somit zu schützende System auf Eindringlinge oder Hinweise auf andere Gefahrensituationen hin zu untersuchen und diese mit möglichst detaillierter Beschreibung zu melden. Gegebenenfalls ist es auch erwünscht, dass entsprechende Gegenmaßnahmen zur Verbesserung der Situation automatisiert ergriffen werden. In einem solchen Fall spricht man zumeist von *Intrusion Prevention Systemen* (IPS), *Intrusion Response Systemen* (IRS) oder auch der Kombination *Intrusion Prevention and Response Systemen*.

Im Wesentlichen gliedert sich die Funktionsweise eines jeden IDS in drei Abschnitte, wie auch Abbildung 2.1 durch jeweils eine Komponente darstellt:

- Beobachtung ausgewählter Parameter des zu überwachenden Systems und Nachhalten der entsprechenden Daten (*Sensor/Agent*),
- Analyse der zuvor gesammelten Daten (*Analyseeinheit*) und
- Meldung einer erkannten Auffälligkeit (*Reporting*) bzw. Ergreifen geeigneter Gegenmaßnahmen, falls dies erwünscht und möglich sein sollte.

Zum Zwecke der Datensammlung ist die Beobachtung verschiedenster Parameter praktikabel. Von Eigenschaften einzelner Prozesse oder Dateien eines Computers bis hin zur Beobachtung des gesamten Netzverkehrs einer einzelnen oder sogar mehrerer Domänen gleichzeitig ist alles denkbar. Im Zuge einer solchen Datenakquise spricht man im Allgemeinen von *Sensoren* oder *Agenten* als informationssammelnde Einheiten. Dabei wird in [BSI, 2002] ein Sensor als eine Datenquelle bezeichnet, die die beobachteten Parameter bereits vorverarbeitet. Dadurch reduziert ein solcher Sensor das Datenaufkommen für eine (Netz-)Übertragung und meldet Auffälligkeiten in Form von *Alarmen*. Es gilt darauf zu achten, dass die daraus resultierenden Informationen auf der einen Seite detailliert genug sind, auf der anderen Seite aber die Flut an Meldungen nicht überhand nimmt, so dass das nachfolgend analysierende System nicht überlastet wird.

[1] Die Begriffe *Intrusion Detection System* und *Frühwarnsystem* werden im weiteren Verlauf der Arbeit synonym verwendet.

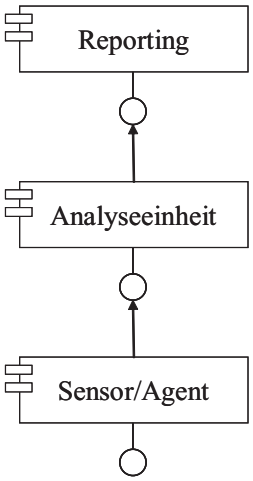


Abbildung 2.1: Generische Komponenten eines Frühwarnsystems

Bei dieser Begriffsdefinition ist zu beachten, dass der Begriff des Sensors meist synonym mit dem Begriff des Agenten verwendet wird. In der Literatur herrscht eine nur sehr schwache Differenzierung zwischen Agenten und Sensoren vor, in den meisten Fällen wird eine äquivalente Nutzung der Begrifflichkeiten für sowohl ausschließlich beobachtende Einheiten (Sensoren) wie auch bereits vorverarbeitende Einheiten (Agenten) verwendet. In beiden Fällen gilt jedoch, dass die entsprechende Beobachtungskomponente sogenannte *Ereignismeldungen* zur weiteren Analyse bereitstellt und diese an die Analyseeinheit weiterleitet. Bei der Datenkollektion eines IDS unterscheidet man in der Regel zwischen host- und netzbasierten Systemen. Mit den Unterschieden beschäftigt sich Abschnitt 2.1.1 genauer. Die eigentliche Analyse der anfallenden Informationen unterteilt man in Missbrauchs- und Anomalieerkennung. In Abschnitt 2.1.2 wird genauer auf Details dieser beiden Herangehensweisen eingegangen.

Ziel eines Intrusion Detection Systems ist es, möglichst exakt einen Angriff erkennen und melden zu können. Bei einer Vielzahl der eingesetzten Strategien besteht jedoch die Gefahr der sogenannten *False Positives* und *False Negatives*. Spricht man von False Positives, so ist die Meldung eines Angriffs gemeint, der in der Realität nicht ausgeübt wurde. Es handelt sich also um einen Fehlalarm. Ein False Negative hingegen bezeichnet das Fehlen einer Meldung eines in der Tat verübten Angriffs auf das zu beobachtende System. In einem solchen Fall konnte also ein Angriff erfolgen, ohne dass das IDS diesen bemerkt hat. Bei der Entwicklung eines IDS ist darauf zu achten, dass die Anzahl der False Positives und False Negatives minimiert wird. Eine zu große Anzahl an False Positives lässt in den meisten Fällen die Aufmerksamkeit, die den Meldungen des entsprechenden IDS geschenkt wird, sinken. Damit besteht die große Gefahr, einen tatsächlichen Angriff, der auch erfolgreich erkannt werden konnte, zu übersehen. Bei einer zu hohen Anzahl an False Negatives sollte die Methodik des entwickelten IDS überdacht werden, da es offensichtlich seine Aufgabe nur unzureichend verrichtet.

2.1.1 Host- und netzbasierte Intrusion Detection Systeme

Bei der Kollektion von Meldungen zur weiteren Analyse unterteilen sich existierende Ansätze im Wesentlichen in zwei Klassen. Es wird eine Unterscheidung zwischen den host- und den netzbasierten Intrusion Detection Systemen gemacht.

2.1.1.1 Hostbasierte Intrusion Detection Systeme

Zur Geburtsstunde von Intrusion Detection Systemen waren *hostbasierte IDS* (HIDS) die ersten verfügbaren Systeme. Sie wurden ursprünglich in den 80er-Jahren vom Militär entwickelt. Die Namensgebung solcher Systeme deutet bereits darauf hin, dass ein HIDS auf dem jeweils zu schützenden Host ausgeführt wird. Diese Variante eines IDS bringt sowohl Vor- wie auch Nachteile mit sich.

Vorteile:

- Da das HIDS auf dem zu überwachenden System ausgeführt wird, ist es möglich, sehr präzise Aussagen über den aktuellen Zustand des jeweiligen Hosts zu tätigen. Dadurch ist eine umfassende Sammlung detaillierter Systeminformationen realisierbar.
- Es können in den meisten Fällen sehr spezifische Aussagen über einen erkannten Angriff auf das Computersystem gemacht werden.

Nachteile:

- Das HIDS benötigt Systemressourcen des zu überwachenden Hosts.
- Das HIDS kann selber Ziel eines Angriffs sein und somit eventuell dem zu schützenden Rechner schaden.
- Im Falle eines Ausfalls des Hostsystems fällt das darauf ausgeführte IDS mit aus.
- Jedes HIDS muss für das zu schützende System individuell entwickelt worden sein. Dabei sind Umstände wie zum Beispiel die zugrunde liegende Systemarchitektur oder das eingesetzte Betriebssystem des Hosts zu beachten.
- Bei kommerziellen Anwendungen sind für jedes zu schützende System Lizenzkosten zu entrichten.

Das *Bundesamt für Sicherheit in der Informationstechnik* (BSI) unterteilt die HIDS generell in vier verschiedene Überwachungstypen [BSI, 2002]:

Systemüberwachung. Die Systemüberwachung erfolgt auf Prozessebene. Datei- und Anwendungszugriffe sowie Benutzeranmeldungen werden überwacht. Zum einen kann eine Überwachung durch Auswertung von Log-Dateien geschehen. Zum anderen können spezielle Sensoren zur regelmäßigen Integritätsprüfung, Prozessüberwachung etc. verwendet werden. Ein Vorteil dieser Art der Überwachung ist, dass in den meisten Fällen die jeweilig erzeugte Meldung einem Nutzer zugeordnet werden kann.

Applikationsüberwachung. Für die Applikationsüberwachung werden für die jeweils zu überwachende Anwendung speziell entwickelte Sensoren eingesetzt. In der Regel arbeiten diese durch die Auswertung der vom zu überwachenden Programm erzeugten Log-Dateien.

Integritätsüberwachung. Die Integritätsprüfung richtet sich an die Überwachung ausgewählter Dateien. Typischerweise erfolgt eine Integritätsprüfung über den Vergleich von Prüfsummen. Gängige Vertreter der Algorithmen zur Erstellung von Prüfsummen sind *md5* [RFC1321, 1992] oder auch *SHA-1* [RFC3174, 2001].

Problematisch gestaltet sich die Integritätsüberwachung im Falle einer berechtigten Änderung der zu überwachenden Dateien. Im Regelfall hat eine Änderung einer Datei die Änderung des Werts der erstellten Prüfsumme zur Folge, was bei einer Integritätsprüfung zum Auslösen eines Alarms führen würde. Dieser Tatsache muss man sich im Praxiseinsatz bewusst sein und ihr muss in geeigneter Weise Rechnung getragen werden.

Ein Nachteil der Integritätsprüfung ist, dass eine Meldung durch einen solchen Sensor immer auf einen bereits erfolgten Angriff hinweist. Eine Frühwarnung ist durch einen Sensor dieser Kategorie nicht zu realisieren.

Überwachung des hostspezifischen Netzverkehrs. Hierbei kann der Netzverkehr des zu überwachenden Systems auf sämtlichen Protokollebenen auf eventuelle Auffälligkeiten hin analysiert werden. Da allerdings eine Überwachung nur für das Hostsystem geschieht, kann typischerweise ein verteilter Angriff nicht erkannt werden.

Beispiele für Vertreter der Kategorie der hostbasierten Intrusion Detection Systeme sind *Tripwire* [Tripwire] bzw. das aktuelle Nachfolgeprojekt *Samhain* [Samhain] oder auch *AIDE* [AIDE].

2.1.1.2 Netzbasierte Intrusion Detection Systeme

Netzbasierte Intrusion Detection Systeme (NIDS) überwachen den Netzverkehr eines Rechners, eines Teilnetzes oder sogar einer gesamten Domäne. Durch diese Art der Konzeption ist es möglich, ein NIDS auf einem eigens für den Zweck der Intrusion Detection dedizierten System auszuführen. Auch die Architektur eines netzbasierten IDS bringt Vor- und Nachteile mit sich.

Vorteile:

- Ein einziger Sensor ist in der Lage, ein gesamtes Netz zu überwachen.
- Durch die Installation eines dedizierten Systems zur Intrusion Detection werden die Ressourcen der zu schützenden Systeme geschont.
- Durch geschickte Konfiguration des Netzes kann das NIDS nach außen hin vollkommen unsichtbar erscheinen und somit nicht Ziel eines Angriffs werden.

- Trotz eines erfolgreichen Angriffs auf einen Teilnehmer des zu schützenden Netzes oder einen Teil des Netzes, der dessen Ausfall zur Folge hat, kann das NIDS weiter seinen Dienst verrichten.
- NIDS sind prinzipiell in der Lage, verteilte Angriffe zu erkennen.

Nachteile:

- Beim Einsatz eines NIDS in einem Kommunikationsnetz, das mit Switches arbeitet, besteht die Herausforderung der lückenlosen Überwachung des Netzverkehrs. Ein Lösungsansatz in einem solchen Fall kann der Anschluss des NIDS an den Mirror-Port der eingesetzten Switches, an die der gesamte Datenverkehr zusätzlich weitergeleitet wird, sein. Dabei ist auf die Dimensionierung der Anschlüsse des NIDS an das Netz zu achten, da hier sehr schnell ein erhebliches Datenaufkommen anfällt und es dadurch gegebenenfalls zu Überlastsituationen kommen kann.
- Durch immer schneller werdenden Netzverbindungen und das immer höher werdende Datenaufkommen kann es unter Umständen zu einer Überlast der Systemressourcen des NIDS kommen. Es gilt jedoch in den meisten Fällen, die anfallenden Informationen möglichst in Echtzeit zu analysieren.
- NIDS sind prinzipiell nicht in der Lage verschlüsselte Kommunikationsdaten zu analysieren. Abhilfe können jedoch Einrichtungen wie zum Beispiel SSL-Proxys in Kombination mit geschicktem Routing schaffen. Dabei gilt es den verschlüsselten Datenverkehr gezielt über einen entsprechenden Proxy zu routen, der sich das Prinzip eines Man-in-the-Middle Angriffs zu nutze macht. Der Proxy baut für den Anwender transparent eine verschlüsselte Verbindung mit dem gewünschten Kommunikationspartner sowie eine separate verschlüsselte Verbindung mit dem Anwender selbst auf. Der anfallende Datenverkehr wird vom Anwender verschlüsselt an den Proxy versendet, der dann die Daten entschlüsselt und über die zweite verschlüsselte Verbindung an den Kommunikationspartner weiterleitet. So ist es dem Proxy möglich die verschlüsselten Daten einzusehen und eine Analyse vorzunehmen.
- NIDS können, durch ihre Konzeption bedingt, nur Auffälligkeiten erkennen, die sich im Netzverkehr äußern.

Ein typisches Beispiel für einen Vertreter der netzbasierten IDS ist u.a. *Snort* [Snort].

2.1.2 Missbrauchs- und Anomalieerkennung

Bei der Analyse der anfallenden Meldungen durch ein Intrusion Detection System unterscheidet man die Herangehensweisen in Missbrauchs- und Anomalieerkennung. Auf beide konzeptionell unterschiedliche Varianten der Datenanalyse wird im Folgenden näher eingegangen.

2.1.2.1 Missbrauchserkennung

Die Missbrauchserkennung sucht in den zur Verfügung stehenden Ereignismeldungen nach sogenannten *Signaturen* verdächtiger Verhaltensmuster. Diese Signaturen sind zuvor bekannt und spiegeln das typische Vorgehen des entsprechenden Angriffs auf ein System wider. Zum Zwecke der Suche nach diesen Signaturen im Meldungsstrom kommen Methoden des Patternmatching sowie Expertensysteme oder auch Zustandsautomaten zum Einsatz. Ein typischer Vertreter dieser Art der IDS ist das Open Source Projekt *Snort* [Snort].

Ein großer Vorteil einer Missbrauchserkennung ist die Zuverlässigkeit, mit der sie arbeitet. Alle aufgezeichneten Angriffsmuster können zuverlässig erkannt werden. Jedoch gilt es die vorhandenen Signaturen nach Möglichkeit so zu gestalten, dass leichte Variationen eines bekannten Angriffs ebenfalls erkannt werden. Dies ist eine mögliche Schwachstelle des Verfahrens, der man sich bewusst sein sollte und der durch geschickte Konzeption des Missbrauchserkennungssystems Rechnung getragen werden muss.

Ein Nachteil des Verfahrens ist die mangelnde Flexibilität gegenüber neuen oder unbekannten Angriffen. Ist keine Signatur oder, wie zuvor beschrieben, nur eine unzulängliche Signatur eines Angriffs vorhanden, so kann eine Missbrauchserkennung diese Bedrohung nicht entdecken. Daraus kann eine hohe Anzahl an False Negatives resultieren.

2.1.2.2 Anomalieerkennung

Bei einem Anomalieerkennungsverfahren „erlernt“ das IDS zu Beginn seines Einsatzes ein „Normalverhalten“ des zu beobachtenden Systems. Dies geschieht in der Regel rundenbasiert. Das bedeutet, dass die anfallenden Ereignismeldungen aus zuvor festgelegten Intervallen ausgewertet und die Charakteristika für das Netz extrahiert werden. Nach einer gewissen Zeit der Beobachtung des Netzes nimmt die Anomalieerkennung ihren eigentlichen Betrieb auf. Dabei wird das jeweils aktuelle Verhalten des Netzes mit dem zuvor erlernten Normalverhalten verglichen. Sollten bei diesem Vergleich zu große Abweichungen von Normalverhalten festgestellt werden, so deutet dies auf eine Anomalie in Netz hin. Bei einer solchen Vorgehensweise ist es wichtig, dass die Daten, aus denen zuvor das Normalverhalten abgeleitet wurde, keine Anomalien enthalten. Es besteht ansonsten die Gefahr, dass ein auffälliges Verhalten im Netz im Nachhinein nicht als Anomalie gedeutet wird.

Weiter werden die Anomalieerkennungsverfahren in statische und adaptive Verfahren gegliedert. Bei einem statischen Verfahren wird ein einmal erlerntes Normalverhalten und das damit assoziierte Profil nicht mehr verändert, während ein adaptives Verfahren versucht, sich den aktuellen Gegebenheiten im Netz anzupassen. Bei einem adaptiven Verfahren besteht die Gefahr, dass sich das erlernte Normalverhalten langsam an eine Anomalie anpasst. Dies könnte sich ein Angreifer zu Nutze machen und das IDS langsam an seinen Angriff gewöhnen, um diesen dann schlussendlich unbemerkt ausführen zu können. Vorteilhaft hingegen ist, dass adaptive Verfahren auf gewollte Änderungen im Netz flexibel reagieren und diese Änderungen in ihr Profil des Normalverhaltens integrieren.

Bei sämtlichen Anomalieerkennungsverfahren besteht die Gefahr einer hohen Rate an False Positives, die es einzudämmen gilt. Auf der anderen Seite ist die Rate der False Negatives durch die Flexibilität des Verfahrens in der Erkennung deutlich geringer im Vergleich zu Missbrauchserkennungsverfahren.

2.1.3 Hybride Erkennungsverfahren

Es gibt sowohl Intrusion Detection Systeme, die ausschließlich eine Missbrauchserkennung durchführen, sowie IDS, die sich auf eine Anomalieerkennung beschränken. Alternativ ist die Entwicklung eines Systems, das eine Mischung aus beiden Analyseverfahren bzw. beide Methoden der Datenanalyse gleichzeitig betreibt, realisierbar.

Die Kombination von Missbrauchs- und Anomalieerkennung in einem Intrusion Detection System bietet entscheidende Vorteile. Auf der einen Seite ist eine zuverlässige Erkennung bereits bekannter Angriffsmuster durch die integrierte Missbrauchserkennung möglich. Auf der anderen Seite ist durch die Integration einer Anomalieerkennung in das IDS die Möglichkeit gegeben, unbekannte Angriffe zu entdecken.

2.2 Grid-Computing

Ian Foster und seine Forschergruppe sind im Bereich des Grid-Computings die wohl anerkanntesten Vorreiter in Forschung und Entwicklung. Daher sind in der Literatur ihre Definitionen weitgehend als allgemeingültig anerkannt. In [Foster u. Kesselman, 1998] wird ein (Compute-)Grid wie folgt definiert:

> „*A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.*“

Im weiteren Verlauf dieses Abschnitts werden grundlegende Konzepte und Architekturen von Grids behandelt. Dabei stehen Dienste und Komponenten im Vordergrund. Ein Schwerpunkt liegt auf Sicherheit und die unterstützenden Mechanismen und Implementierungen in Grids.

2.2.1 Konzepte und Architektur

Hinter dem Begriff des Grid-Computing steckt die zukunftsweisende Idee bekannte Konzepte des verteilten und Hochleistungsrechnens zu erweitern, um die mit der Zeit gestiegenen Anforderungen vorwiegend aus dem Bereich des Höchstleistungsrechnens erfüllen zu können. Ziel ist es, eine neue Infrastruktur bereitzustellen, die Anwendern einen einfachen, transparenten Zugriff auf einen Pool von weltweit über viele Organisationen verteilten Ressourcen ermöglicht. Die Nutzung von Rechen- und Speicherkapazitäten soll dabei genauso

einfach möglich sein wie etwa der Bezug von Elektrizität aus dem Stromnetz [Chetty u. Buyya, 2002]. In diesem Zusammenhang ist auch der Begriff *Grid* geprägt worden, da eine Analogie zu einem Stromverbreitungsnetz gesucht wurde. Zumeist steht dabei der Gedanke des Beziehens von Rechen- und Speicherkapazitäten „aus der Steckdose“ im Vordergrund. Inzwischen wird der Grid-Ressourcenbegriff erweitert auch auf Experimentgerät, Daten, Anwendungen und IT-Services angewendet.

Zur Unterscheidung ähnlicher Ideen sollen die folgenden Charakteristika von Grids helfen, die Ian Foster in [Foster, 2002] aufstellt:

Dezentrale Organisation. Die im Grid verbundenen Ressourcen können sich über viele juristisch, organisatorisch und administrativ autonome und geografisch verteilte Unternehmen erstrecken. Grids unterliegen generell keiner zentralen Kontrolle, was insbesondere das Management erschwert.

Heterogenität. Als Folge der dezentralen Organisation sind Grids oftmals sehr heterogen aufgebaut. Die eingebrachten Ressourcen können sich stark in Hardware, Software und Netzanbindung unterscheiden.

Verwendung standardisierter und offener Protokolle. Nur die Nutzung standardisierter, offener und breit unterstützter Protokolle sowie Schnittstellen als Basis von Grid-Technologien stellt die Interoperabilität innerhalb von komplexen, verteilten Strukturen nachhaltig sicher. Dies dient letztlich auch der Vermeidung von Abhängigkeiten.

Hohe Leistungsfähigkeit. Der koordinierte Zugriff auf integrierte Ressourcen innerhalb von Grid-Verbünden soll zu einem im Vergleich zur Summe der Einzelsysteme signifikant größeren Nutzen und erhöhter Qualität des Gesamtsystems beispielsweise hinsichtlich Antwortzeit, Durchsatz, Speicherplatz oder Rechenkapazität führen.

Grids kommen zurzeit mehrheitlich zum Einsatz, um entweder Speicherkapazitäten (sogenannte *Data Grids* oder auch *Storage Grids*) oder um Rechenkapazitäten (sogenannte *Compute Grids*) durch die Föderation bestehender Kapazitäten zu erhöhen.

Damit soll vor allem neuen Herausforderungen des Höchstleistungsrechnens begegnet werden. In vielen Bereichen der Forschung (Hochenergiephysik, Astrophysik, Meteorologie, etc.) sieht man sich inzwischen mit Datenmengen konfrontiert, die auf herkömmliche Art und Weise nicht länger zu speichern und verarbeiten sind. Durch intelligente Verbindung vieler dezentral organisierter Rechner in einem Grid kann dem Problem der Speicherung und dem Austausch von großen Datenmengen begegnet werden. Ebenso lassen sich aufwendige Berechnungen zu anspruchsvollen Aufgaben effizient durchführen, die die Leistungsfähigkeit herkömmlicher Systeme sonst weit übersteigen würden.

Die Architektur

Wie zuvor bereits festgestellt, sieht man sich auf dem Gebiet des Grid-Computings mit gewaltigen Herausforderungen konfrontiert. Im Zuge der Entwicklung von Grids wurden

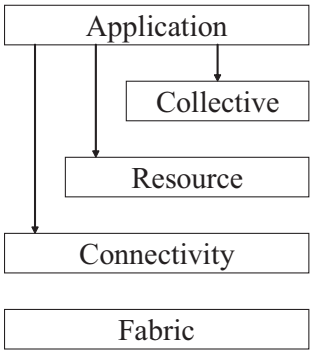


Abbildung 2.2**:** Layered Grid Protocol Architecture [Foster u. a., 2001]

offene, standardisierte Schnittstellen und Protokolle zur Kommunikation und Steuerung entworfen, die vor allem die Interoperabilität zwischen verschiedenen, meist sehr heterogenen Grid-Systemen sicherstellen sollen. Diese Schnittstellen und Protokolle lassen sich gemäß ihrer Funktion in mehrere Schichten aufteilen. Komponenten einer Schicht besitzen ähnliche Eigenschaften und können auf die von niedrigeren Schichten angebotenen Dienste aufbauen. Abbildung 2.2 veranschaulicht das Schichtenkonzept nach [Foster u. a., 2001].

Ziel der *Layered Grid Protocol Architecture* (GPA) ist es nicht, sämtliche benötigte Protokolle aufzuzählen, sondern Klassen von Komponenten zu identifizieren und deren grundlegende Funktion und Interaktion festzulegen. Das Modell besagt, dass die mittigen Kernschichten nur die nötigsten Funktionen besitzen, während möglichst viel an den Enden der Architektur geleistet werden soll. Dies erleichtert die Portabilität zwischen unterschiedlichen Plattformen, da möglichst wenig neu implementiert werden muss.

Application. An der Spitze der Protokollarchitektur nach [Foster u. a., 2001] stehen die eigentlichen Anwendungen, die unter Zuhilfenahme der Protokolle und Dienste der verschiedenen unteren Schichten koordiniert auf die Ressourcen im Grid zugreifen.

Collective. Die Collective-Schicht enthält Protokolle und Dienste, die sich auf mehrere Ressourcen beziehen. Typischer Weise werden dabei vielfältige Funktionalitäten wie Verzeichnisdienste, Sicherheits-, Policy- und Abrechnungsmechanismen bereitgestellt.

Resource. In die Resource-Schicht fällt die Initiierung, die Überwachung (*Monitoring*), die Kontrolle und die Abrechnung (*Accounting & Billing*) von einzelnen Zugriffen auf Ressourcen.

Connectivity. Auf Connectivity-Ebene sind die zentralen Protokolle zur (sicheren) Kommunikation und Authentisierung angesiedelt. Durch diese Schicht soll eine sichere Interaktion im Grid gewährleistet werden. Dazu gehören unter anderem auch Mechanismen für den sicheren Datentransport und das Routing im Grid.

Fabric. Die Fabric-Schicht organisiert den lokalen, jeweils spezifischen Zugriff auf die sehr unterschiedlichen Betriebsmittel wie Rechner, Speichersysteme, Kataloge, Netze oder

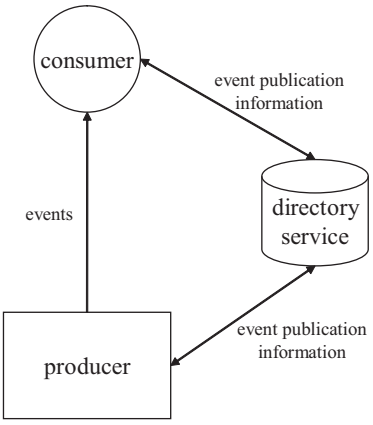


Abbildung 2.3**:** A Grid Monitoring Architecture [Tierney u. a., 2000]

Sensoren. Eine der primären Aufgaben dieser Schicht ist der Umgang mit der großen Heterogenität in Grids.

Neben einer Protokollarchitektur werden unter anderem in [Foster u. Kesselman, 2003] und [Li u. Baker, 2005] einige Basiskomponenten und -dienste beschrieben, die in einem Grid von Nöten sind. Diese sind insbesondere:

Grid-Security. Sicherheitsaspekte in Grids werden in fast jeder einschlägigen Arbeit kurz angesprochen. Jedoch findet sich in keiner Arbeit eine erschöpfende Abdeckung aller Teildisziplinen des Sicherheitsmanagements, sondern wird vielmehr in einer Großzahl der Arbeiten der Fokus auf die Überprüfen von Autorisierungen, das Feststellen einer Identität (Authentifizierung, Signaturen, Notarisierung bzw. Zertifizierung), das Durchführen der Zugriffskontrolle, die Sicherstellung der Vertraulichkeit (Verschlüsselung) sowie die Sicherstellung der Datenintegrität gelegt. Die Aspekte der Überwachung des Systems auf Sicherheitsangriffe und der Berichterstattung zur Sicherheit werden zumeist nur kurz erwähnt und deren Wichtigkeit angedeutet.

Auf Grund der großen Relevanz des Themenkomplexes für diese Arbeit, befasst sich Abschnitt 2.2.3 detaillierter mit bestehenden Standards und deren Einsatz in Grids.

Grid-Monitoring. Die *Performance Working Group* des *Global Grid Forums* (GGF) hat eine weit verbreitete und offene Architektur für das Monitoring in Grids entwickelt, die *Grid Monitoring Architecture* (GMA) [Tierney u. a., 2000]. Abbildung 2.3 illustriert den Aufbau der GMA, die aus den drei Komponenten *Producer*, *Consumer* und *Directory Service* und ihren Kommunikationsbeziehungen untereinander besteht.

Der sogenannte *Producer* der GMA ist eine Software-Komponente, die verfügbare Monitoring-Informationen an einen *Consumer* oder zu Deutsch Konsumenten weiterleitet. Als Kommunikationsmodell ist dabei sowohl ein Push- wie auch ein Pull-Modell vorgesehen. Der Datentransfer kann entweder als durchgehender Datenstrom (engl.

stream) oder als Aggregat aller angefragten Informationen (engl. *single response per request*) stattfinden.

Ein jedes Programm im Grid, das Monitoring-Informationen verarbeitet ist im Sinne der GMA ein *Consumer*. Es sind verschiedene Typen von Konsumenten vorgesehen, die sich in ihrem Verhalten voneinander abheben. So ist zum Beispiel ein archivierender Konsument denkbar, der einzig Monitoring-Daten für die spätere Auswertung oder Analyse speichert. Auch ist ein Echtzeit-Konsument (engl. *real time consumer*) realisierbar, der Monitoring-Informationen mit Echtzeitansprüchen benötigt. Solche Typen von Konsumenten nutzen zumeist die Möglichkeit Datenströme (*streams*) der Produzenten abzugreifen. In wenigen Fällen wird auf Datenaggregate zurückgegriffen um eine bessere Performanz unter Einbußen in Bezug auf Echtzeitansprüche zu gewährleisten. Ein weiterer interessanter Typ an Konsument ist der sogenannte *overview consumer*, der Daten mehrerer Produzenten sammelt um mit Hilfe eines globaleren Überblicks differenziertere Aussagen treffen und Entscheidungen fällen zu können [Tierney u. a., 2000].

Der Verzeichnisdienst der GMA (engl. *Directory Service*) stellt eine Informationssammlung zu verfügbaren Produzenten und Konsumenten im Rahmen des Monitoring zur Verfügung. Alle Produzenten und Konsumenten registrieren sich bei ihrem Start beim Verzeichnisdienst. Dort hinterlegen sie typischer Weise auch, welche Art von Information sie bereitstellen bzw. konsumieren. Die GMA sieht vor, dass in einem Grid entweder genau ein solcher Verzeichnisdienst existiert oder dass mehrere Verzeichnisdienste durch einen übergeordneten oder privilegierten Verzeichnisdienst verwaltet und koordiniert werden.

Im Grid-Umfeld gibt es eine Vielzahl an Monitoring-Systemen und -Komponenten, die hier erschöpfend aufzulisten schier unmöglich scheint. Einige prominente Vertreter dieser Gattung sind zum Beispiel die *Monitoring and Discovery Services* (MDS) des Globus Toolkit (siehe auch Abschnitt 2.2.2.2), die *Relational Grid Monitoring Architecture* (R-GMA) von gLite (siehe auch Abschnitt 2.2.2.3) oder auch *GridICE* (`http://gridice.forge.cnaf.infn.it/`).

Grid-Scheduling and Resource Management. Scheduling im Grid bezeichnet den Prozess der Abbildung von Grid-Jobs auf verfügbare Ressourcen im Grid über verschiedene administrative Domänen hinweg. Dabei kann ein Grid-Job in mehrere kleinere Teile aufgeteilt und so auf verschiedene Ressourcen verteilt werden. Der Scheduler ist dafür verantwortlich, dass die Ausführung eines Jobs stets unter der Beachtung der durch den Anwender spezifizierten Randbedingungen vollzogen wird. Insbesondere sind dies die Zeit, die bis zum Abschluss des Jobs verstrichen ist, sowie die Kosten, die durch die Ressourcennutzung für den Anwender entstanden sind.

In einem Grid sind typischer Weise mehrere Instanzen eines Schedulers oder auch mehrere verschiedene Scheduler im Einsatz. Prominente Implementierungen sind unter anderem *Condor* (`http://www.cs.wisc.edu/condor/`), *Sun Grid Engine* (SGE,

http://gridengine.sunsource.net/), *the Portable Batch System* (PBS, http://www.pbsgridworks.com/) oder auch *Nimrod/G* (http://www.csse.monash.edu.au/~davida/nimrod/nimrodg.htm).

Grid Workflow Management. Heutzutage stellt die *Open Grid Services Architecture* (OGSA) [OGSA] den de-facto-Standard für die Gestaltung Dienst-orientierter Grids dar. Das Hauptziel der OGSA ist dabei zu ihr konforme Grid-Dienste untereinander interoperabel zu gestalten. Hierbei kommen vor allem Web Services zum Einsatz, die SOAP [SOAP] als Protokoll für den Nachrichtenaustausch nutzen.

Ein Workflow wird von der *Workflow Management Coalition* (WfMC, http://www.wfmc.org/) in [Hollingsworth, 1995] wie folgt definiert:

> „Definition – Workflow
> The computerised facilitation or automation of a business process, in whole or part."

Ein Workflow besteht aus einer Menge an Aktivitäten, die parallel und/oder sequentiell abgearbeitet werden um ein übergeordnetes Geschäftsziel (engl. *business goal*) zu erreichen. Ein Workflow wird in einer Prozessbeschreibungssprache definiert und kann dann durch ein *Workflow Management System* (WFMS) abgearbeitet werden. Ein WFMS wird von der WfMC in [Hollingsworth, 1995] wie folgt definiert:

> „Definition – Workflow Management System
> A system that completely defines, manages and executes "workflows" through the execution of software whose order of execution is driven by a computer representation of the workflow logic."

Einige prominente Vertreter von Workflow-Beschreibungssprachen sind unter anderem die *Web Services Flow Language* (WSFL), die *Business Process Modelling Language* (BPML) oder auch die *Business Process Execution Language for Web Services* (BPEL4WS).

Grid-Portals. Ein Grid-Portal ist ein Web-basierter Zugriffspunkt für die Endanwender eines Grids. Das Portal bietet einen transparenten Zugriff auf eine Vielzahl an im Grid verfügbaren Ressourcen. In der Regel bietet ein Grid-Portal eine anwenderspezifische Sicht auf verfügbare Ressourcen, so dass der Endanwender eine an seine Problemstellung angepasste Repräsentation vorfindet.

In ihrer ursprünglichen Form unterstützen Grid-Portale eine Nutzerauthentifizierung (meist unter Nutzung von Nutzername und Passwort) zur nachfolgenden Autorisierung der Nutzung von Grid-Ressourcen, Job-Management (starten, stoppen und beobachten von Grid-Jobs), Datentransfer zu und von Grid-Ressourcen (z.B. hochladen von ausführbarem Programmcode oder herunterladen von Ergebnissen) sowie einen Informationsdienst, der die Suche nach verfügbaren und geeigneten Grid-Ressourcen für eine bestimmte Aufgabe unterstützt.

In einer Evolutionsstufe der Grid-Portale kamen die sogenannten *Portlets* auf. Aus Sicht des Anwenders erscheint ein Portlet wie ein Fenster, das einen bestimmten Dienst zur Verfügung stellt. Aus Anwendungssicht handelt es sich bei einem Portlet um eine Softwarekomponente, die in Java geschrieben ist. Sie verarbeitet Nutzeranfragen und generiert in Abhängigkeit von der Anfrage dynamischen Inhalt. Ein jedes Portlet benötigt zur Ausführung eine Laufzeitumgebung, einen sogenannten *portlet container*.

Ein typischer Vertreter für ein Portlet-basiertes Grid-Portal ist das *GridSphere portal framework* (`http://www.gridsphere.org`) oder auch IBMs *WebSphere* (`http://www.ibm.com/software/websphere/`).

Virtual Organizations. Ein zentraler Bestandteil von Grids ist das Konzept der *Virtuellen Organisation* (VO). In der Literatur findet sich eine Vielzahl verschiedener, jeweils auf den speziellen Anwendungsfall hin zugeschnittene Definitionen. Für diese Arbeit wird daher im Folgenden die Definition einer VO nach [Schiffers, 2007] herangezogen:

> „Definition (Virtuelle Organisation):
> Eine Virtuelle Organisation ist eine zeitlich begrenzte koordinierte Kooperation von Elementen in Form von Individuen, Gruppen von Individuen, Organisationseinheiten oder ganzer Organisationen, die Teile ihrer physischen oder logischen Ressourcen oder Dienste auf diesen, ihre Kenntnisse und Fähigkeiten sowie Teile ihrer Informationsbasis in Form virtueller Ressourcen und Dienste über eine Grid-Infrastruktur derart zur Verfügung stellen, dass die gemeinsam vereinbarten Ziele unter Berücksichtigung lokaler und globaler Policies erreicht werden können.“

Insbesondere das Management dynamischer, interorganisationaler Virtuellen Organisationen stellt immense Herausforderungen dar. Um diesen zu begegnen, bedarf es neuer Ideen und Technologien, die u.a. in [Schiffers, 2007] behandelt werden. In der heutigen Praxis wird zumeist im Zusammenhang mit dem Management von VOs nur eine Infrastruktur zur Authentifizierung und Autorisierung von Nutzern (eine sogenannte *AA-Infrastruktur* oder auch einfach nur *AAI*) beschrieben. Im Grid-Umfeld kommt hierzu zum Beispiel *Shibboleth* (`http://shibboleth.internet2.edu/`) zum Einsatz, dessen Konzepte und Komponenten nachfolgend kurz exemplarisch beschrieben werden und in Abbildung 2.4 illustriert sind.

Im Wesentlichen besteht eine AA-Infrastruktur (hier beispielhaft Shibboleth) aus drei zentralen Komponenten:

- Der *Identity Provider* (IdP) wird von jeder an einer Föderation teilnehmenden Einrichtung benötigt. In der Regel besteht ein solcher IdP aus einer *Attribute Authority*, dem *Handle Service*, dem Einrichtungs-lokalen *Identity Management* (meist ein Verzeichnisdienst oder eine Datenbank) und einem lokalen *Single Sign-On* Dienst (SSO).

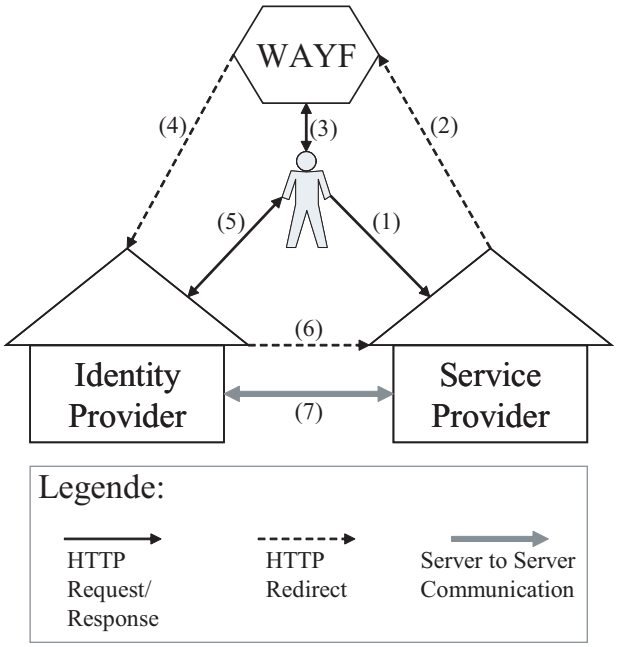


Abbildung 2.4**:** Aufbau und Funktionsweise von Shibboleth

- Jede Einrichtung, die einen Dienst in einer Föderation bereitstellen möchte, benötigt einen *Service Provider*, der aus einem *Assertion Consumer Service*, dem *Attribute Requestor* und dem *Resource Manager* besteht.
- Der *Where-are-you-from* Dienst (WAYF) ist die zentrale Stelle einer Föderation, die alle teilnehmenden IdPs kennt und einem Nutzer ermöglicht seinen IdP zu identifizieren.

Abbildung 2.4 stellt schematisch den Vorgang dar, der bei einem Browser-basierten Zugriff auf eine Ressource unter Nutzung von Shibboleth durchgeführt wird:

1. Ein Nutzer, der Zugriff auf eine durch Shibboleth verwaltete Ressource erhalten möchte, wendet sich an den zuständigen Dienstanbieter (*Service Provider*).
2. Der Anwender wird an den Where-are-you-from Dienst (*WAYF*) umgeleitet, der ...
3. ... den Nutzer zur Auswahl seines *Identity Provider*s auffordert.
4. Daraufhin leitet der *WAYF* Dienst die Nutzeranfrage an den zuständigen bzw. zuvor ausgewählten IdP weiter, und ...
5. ... der Nutzer authentifiziert sich auf dem ihm vertrauten Wege bei seinem *Identity Provider*.
6. Es wird eine eindeutige ID (der sogenannte *Handle*) erzeugt und zum Assertion Consumer Service des *Service Provider*s weitergeleitet.
7. Dieses *Handle* wird dazu verwendet, um Attribute des Nutzers bei seinem *Identity Provider* anzufragen. Anhand der zurückgelieferten Attribute kann schließlich

der Dienstanbieter (*Service Provider*) eine Entscheidung über die Gewährung des Zugriffs (Autorisierung) sowie ggf. die Art des Zugangs zum angebotenen Dienst fällen.

Für die Integration des ursprünglich nicht für Grids entwickelten Shibboleth in Grid-Umgebungen existieren eine Reihe an Projekten und Entwicklungen. Unter anderem ist dazu *GridShib* (`http://gridshib.globus.org/`) für die Integration von Shibboleth in das Globus Toolkit zu nennen. Weiterführende Informationen zu AA-Infrastrukturen in Grid-Middleware finden sich auch in [Dussa u. a., 2006].

2.2.2 Grid-Middleware Implementierungen

Im Laufe der Zeit sind eine Vielzahl an Grid-Middleware Implementierungen entstanden, jeweils mit für sie spezifische Vor- und Nachteilen. Nachfolgend wird eine kleine Auswahl prominenter Grid-Middleware Lösungen kurz umrissen, namentlich *UNICORE* (**UN**iform **I**nterface to **CO**mputing **RE**sources) [UNICORE], das *Globus Toolkit* [Globus Toolkit 4] und *gLite* [gLITE]. Im Rahmen dieser Arbeit werden die Implementierungen nur in sehr knapper Form behandelt, für eine umfangreichere Darstellung sei ausdrücklich auf die Originalquellen verwiesen. Die Auswahl genau dieser Grid-Middleware Lösungen begründet sich dadurch, dass alle drei Middleware-Konzepte im Rahmen des D-Grid Projekts (siehe hierzu auch Abschnitt 3.2.1), das im weiteren Verlauf der Arbeit als Anwendungsfall zur Analyse von Anforderungen an ein föderiertes Intrusion Detection System für Grids dient, parallel zum Einsatz kommen.

Eines der wichtigsten Ziele einer Grid-Middleware ist es, über viele Organisationen verteilte Ressourcen für eine bestimmte Nutzergruppe transparent verfügbar zu machen. Eine der Hauptaufgaben besteht also darin, den Endanwendern einen einheitlichen Dienstzugriffspunkt auf die dem Grid zugrundeliegenden heterogenen Ressourcen zur Verfügung zu stellen. Dabei gilt es die Komplexität von Grids soweit als möglich zu verbergen, was im Wesentlichen einer Vereinfachung in der Entwicklung von Grid-Anwendungen mit entsprechendem Zugriff auf die angebotenen Dienste und Betriebsmittel dienen soll.

Auch wenn es prinzipielle Unterschiede bei Architektur und angewendetem Implementierungsmodell gibt, so überlappen sich die Funktionalitäten der einzelnen Systeme doch in weiten Teilen. Als grundlegende Aufgaben einer Grid-Middleware identifizieren von Laszewski et. al. [von Laszewski u. Amin, 2004] im wesentlichen vier Aspekte:

- Eine zentrale Funktion von Grid-Middlewares ist die Ausführung von Programmen (engl. *Jobs*). Die Middleware entscheidet dabei anhand der momentanen Auslastung und der individuellen Anforderungen wie, d.h. an welchem Ort, in welcher Reihenfolge usw., der Auftrag unter Nutzung der zur Verfügung stehenden Ressourcen am besten abgearbeitet werden kann. Abhängigkeiten zwischen den Aufträgen müssen hierbei berücksichtigt und eventuelle Konflikte aufgelöst werden. Es muss außerdem immer sichergestellt sein, dass nur berechtigten Nutzern der Zugriff gestattet wird.

- Eine große Bedeutung kommt der Gewährleistung der Sicherheit im Grid zu. Stabile Mechanismen zur Authentisierung und Autorisierung von Anwendern sind essentiell und sollten Ressourcen und gegebenenfalls vertrauliche und/oder wertvolle Daten schützen. Eine Verschlüsselung der Kommunikation soll verhindern, dass Daten auf dem Transportweg ausgespäht werden. Gegebenenfalls kann auf etablierte Public Key Infrastrukturen (PKI) oder Kerberos zurückgegriffen werden. Die Größe von Grids erfordert zudem in den meisten Fällen Single Sign-On Dienste, um die Inanspruchnahme vieler Ressourcen zu vereinfachen.
- Von großer Wichtigkeit ist ebenfalls die Bereitstellung von Informationen über am Grid beteiligte Ressourcen und Dienste. Dabei kann es sich um eher statische Informationen wie Art und Umfang der Komponenten oder auch dynamische Aspekte wie zum Beispiel die momentane Auslastung, verfügbarer Speicherplatz oder ähnliches handeln. Die Daten sollten optimaler Weise mit Metainformationen angereichert und in einem sich selbst beschreibenden Format bereitgestellt werden. Der Vorgang einer solchen Informationsbeschaffung und Aufbereitung wird im Allgemeinen mit dem Begriff des *Monitoring* beschrieben. Monitoring-Informationen dienen zum einen einigen Grid-Systemen (z.B. einem Grid-Scheduler oder Broker) als Entscheidungsgrundlage für ihre Diensterbringung, zum anderen sind sie auch für den Grid-Nutzer zur persönlichen Information sinnvoll und werden deswegen meistens in einem Web-Portal übersichtlich dargestellt.
- Eine weitere Anforderung an eine Grid-Middleware ist die Übertragung von Daten, die den Austausch auch großer Datenmengen zwischen unterschiedlichen Standorten und hoch performanten Massenspeichersystemen erlaubt. Zusätzlich werden Replikationstechniken benötigt, um Daten aus Gründen der Leistungssteigerung oder zu Zwecken der redundanten Datensicherung an unterschiedlichen Orten benötigt.

2.2.2.1 UNICORE

Bereits in den 90er-Jahren initiierte das Bundesministerium für Bildung und Forschung (BMBF) die Arbeit an dem ***UN****iform* ***I****nterface to* ***CO****mputing* ***RE****sources* (UNICORE) [UNICORE], um die Zusammenarbeit und die gemeinsame Nutzung von Ressourcen der Zentren für *High Performance Computing* (HPC) in Deutschland zu fördern. In den folgenden Jahren wuchs UNICORE zu einer stabilen Grid-Middleware Lösung, die heute ihren Einsatz an einigen Rechenzentren weltweit im täglichen Betrieb findet. UNICORE ist frei unter der BSD-Lizenz verfügbar und wird bis zum heutigen Tage stets weiterentwickelt. Zum zurzeit aktuellen UNICORE 6 haben unter anderem namhafte Hersteller wie zum Beispiel die Intel Software and Solution Group in Brühl, die Fujitsu Laboratories of Europe in London oder auch das Forschungszentrum in Jülich beigetragen.

Mit UNICORE wird der Zugriff auf Rechenressourcen vereinheitlicht. Es wird ermöglicht für den Anwender transparent auf Ressourcen innerhalb einer (organisatorisch) verteilten Umgebung zuzugreifen. Dabei werden verschiedene Hardwarearchitekturen, herstellerspezifische Betriebssysteme, untereinander inkompatible Batch-Systeme, unterschiedliche

Anwendungsumgebungen oder Sicherheitsrichtlinien vor dem Endanwender verborgen und durch einen einheitlichen Dienstzugriffspunkt abstrahiert. UNICORE zeichnet sich insbesondere durch eine leicht bedienbare, grafische Oberfläche aus. Bei der Entwicklung der Middleware wurde auf ausgereifte sicherheitskritische Mechanismen, wie etwa zur Authentisierung von Anwendern, Servern und Software oder zur Verschlüsselung der Kommunikation, und deren Integration in administrative Prozesse geachtet, um einen reibungslosen Betrieb zu gewährleisten [Riedel u. Mallmann, 2006; Streit u. a., 2005].

Das UNICORE-Paket basiert auf einem sogenannten vertikal integrierten Software-Stack. Demnach sind von der Netzebene bis hin zur Benutzerschnittstelle alle Ebenen der Kommunikation in einem einzigen Programm zusammengefasst. Durch diesen monolithisch anmaßenden Ansatz müssen auf der einen Seite Einbußen in Bezug auf die Flexibilität in Kauf genommen werden, gleichzeitig können jedoch der Umfang und die Mächtigkeit des Clients anwachsen.

Die Vorteile und Funktionen von UNICORE werden nach Asadzadeh et. al. wie folgt zusammengefasst [Asadzadeh u. a., 2006]:

Direkte Erzeugung und Übermittlung von Jobs durch den Anwender. Eine grafische Oberfläche unterstützt den Anwender bei der Generierung komplexer und voneinander abhängiger Jobs, die auf jedem UNICORE-System ohne weitere Änderungen ausgeführt werden können.

Job-Management. Ein eigenes Job Management System ermöglicht Anwendern die Kontrolle über Jobs und Daten.

Daten-Management Während der Erzeugung von neuen Jobs kann der Anwender festlegen, welche Daten zur Ausführung des Jobs importiert oder exportiert werden.

Erweiterbarkeit. Das User-Interface lässt sich über Plugins erweitern, um speziellen Anforderungen zumeist wissenschaftlicher Anwendungen zu berücksichtigen.

Flusskontrolle. Ein Job kann als Menge gerichteter und azyklischer Graphen beschrieben werden.

Single Sign-On. UNICORE bietet einen *Single Sign-On* mittels X.509-Zertifikaten.

Unterstützung von Legacy-Jobs. UNICORE unterstützt den traditionellen Batch-Betrieb, indem Anwender auch ihre alten Job-Beschreibungen zu neuen Jobs hinzufügen können.

Ressourcen-Management. Anwender können Zielsysteme auswählen und benötigte Ressourcen spezifizieren. Der UNICORE-Client überprüft die Korrektheit des Jobs und fordert den Anwender gegebenenfalls zur Korrektur auf.

Einsatz findet UNICORE in einer Vielzahl von Grid-Projekten. Unter anderem wird es vom *EUROGRID* (`http://www.eurogrid.org/`) und dessen Anwendungen *Bio GRID*, *Meteo GRID*, *CAE GRID* und dem *HPC Research GRID*, vom *Grid Interoperability*

Project (GRIP, `http://www.grid-interoperability.eu/`), vom *OpenMolGrid* (`http://www.openmolgrid.org/`) und von der japanischen *National Research Grid Initiative* (NAREGI, `http://www.naregi.org/index_e.html`) als Middleware verwendet.

2.2.2.2 Globus Toolkit

Anfänglich waren Universitäten und Forschungsinstitute federführend an dem in den späten 1990er-Jahren initiierten Projekt zur Entwicklung des *Globus Toolkit* [Globus Toolkit 4] beteiligt. Mittlerweile beteiligen sich jedoch auch Industriepartner wie IBM, Microsoft oder Sun an der Weiterentwicklung, die von der *Globus Alliance* (`http://www.globus.org/`) vorangetrieben wird

Anders als UNICORE stellt das frei verfügbare Globus Toolkit eine reichhaltige Sammlung von grundlegenden Werkzeugen (engl. *Bag of Services*) für Grids bereit, die den einheitlichen Zugriff auf Ressourcen und Dienste im Grid ermöglichen. Bestandteil des Pakets sind unter anderem Komponenten betreffend die Sicherheit, das Management von Ressourcen und Daten, die Kommunikation, die Fehlerbehandlung und die Portabilität (siehe auch Abbildung 2.5). Das Globus Toolkit baut auf der *Grid Security Infrastructure* (GSI) auf, die ein breites Spektrum an Sicherheitsmechanismen für den Zugriff auf verteilte Ressourcen bereitstellt. Mit den enthaltenen Werkzeugen lässt sich so eine Infrastruktur schaffen, die Anwendern eine sichere Ausführung ihrer Rechenaufträge in einem verteilten und heterogenen Umfeld ermöglicht. Da die Grid Security Infrastructure als Sicherheitsplattform auch in anderen Grid-Middlewares, wie unter anderem bei *gLite*, sowie unabhängig davon auch in weiteren Anwendungen eingesetzt wird, geht Abschnitt 2.2.3.1 etwas genauer darauf ein.

Abbildung 2.5 stellt die fünf Säulen der Funktionsbereiche des Globus Toolkit v4 [Foster, 2007] dar.

Security. Die Säule der Sicherheitsfunktionalitäten des Globus Toolkit v4 besteht aus den vier Bausteinen *Authentication/Authorization*, *Community Authorization*, *Delegation* und *Credential Management*. Somit stellt die Summe der vier Funktionsblöcke die notwendige Funktionalität zur Authentifizierung und Autorisierung der Grid-Nutzer bereit, inklusive der Delegation und dem Management von Rechten (*Delegation and Credential Management*). Dem Begriff der Sicherheit zuwider stellt dieser Bereich insbesondere keine Funktionen zur Verschlüsselung von Daten- und Kommunikationsverbindungen oder zur Gewährleistung von Datenintegrität bereit.

Data Management. Unter Datenmanagement wird beim Globus Toolkit die Einheit der fünf Bausteine *GridFTP*, *Reliable File Transfer* (RFT), *Data Access & Integration* (DAI), *Replica Location* und *Data Replication* gefasst. Die Hauptaufgabe dieser Funktionsblöcke ist das Bereitstellen von Protokollen und Diensten zur Datenspeicherung und -übertragung, sowie deren Replikation für performanteren Zugriff durch redundante Speicherung, was auch zu Datensicherungszwecken verwendet werden kann.

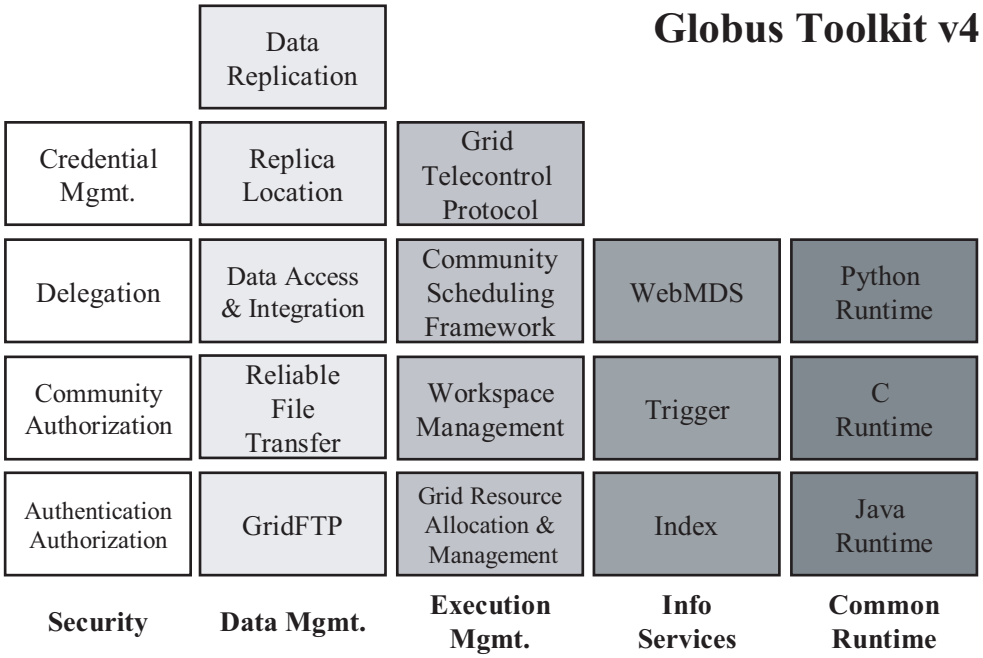


Abbildung 2.5**:** Aufbau des Globus Toolkit v4 nach [Foster, 2007]

Zusätzlich wird durch die DAI-Dienste eine Möglichkeit zur Anfrage, dem Update, der Transformation und der Auslieferung verfügbarer Daten bereitgestellt.

Execution Management. Dieser Funktionsbereich des Globus Toolkit 4 besteht aus Komponenten, die sich im Wesentlichen um die Initiierung, das Monitoring, Management, Scheduling und der Koordination im Grid ausführbarer Programme, die für gewöhnlich in diesem Kontext als *Job* bezeichnet werden, kümmern. Dazu stehen die vier Teilkomponenten *Grid Resource Allocation & Management* (GRAM), *Workspace Management*, *Community Scheduling Framework* und *Grid Telecontrol Protocol* bereit, die die zuvor genannten Funktionalitäten bereitstellen. Eine zentrale Komponente stellt dabei der GRAM dar, der unter anderem für die Job-Submittierung, eine grundlegende Ressourcenallokation für einen Grid-Job und die Möglichkeit zur Abfrage des Status eines Jobs zur Verfügung stellt. Zur Beschreibung notwendiger Ressourcen zur Ausführung eines Grid-Jobs kommt die *Resource Description Language* (RDL) zum Einsatz.

Information Services. Die Informationsdienste *Trigger*, *Index* und *WebMDS* stellen Statusinformationen zum Grid (z.B. Monitoring-Informationen zu Grid-Ressourcen) zum einen für Grid-Dienste wie dem Grid-Accounting und einem Grid-Scheduler zur Verfügung. Zum anderen bietet sich ebenfalls die Möglichkeit für den Endanwender relevante Informationen übersichtlich graphisch zu repräsentieren wie es u.a. das WebMDS tut.

Common Runtime. Mit den drei Blöcken des *Java Runtime*, *C Runtime* und *Python Runtime* stellt das Globus Toolkit eine Laufzeitumgebung für die drei Programmiersprachen Java, C und Python bereit. Somit können Programme für das Grid entwickelt werden und später als Grid-Jobs in einer Globus Toolkit Infrastruktur zur Ausführung gebracht werden.

Zusammenfassend lässt sich festhalten, dass das Globus Toolkit in der Version 4 eine umfassende Grid-Middleware bietet, die nicht ohne Grund eine recht weite Verbreitung in produktiven Grid-Umgebungen gefunden hat. Alleine im Bereich der Sicherheit lassen sich starke Defizite identifizieren. Bei der Entwicklung ist wohl Wert auf eine AA-Infrastruktur gelegt worden, auch ist darauf geachtet worden, dass Kommunikationsverbindungen stets verschlüsselt aufgebaut werden. Jedoch insbesondere in Bezug auf die Datenintegrität gespeicherter Informationen und der Sicherheitsüberwachung und Berichterstattung dazu sind keine Funktionalitäten vorgesehen.

2.2.2.3 gLite

Die Grid-Middleware *gLite* [gLITE] hat sich in Europa gerade im wissenschaftlichen Bereich einen Namen gemacht und durchaus etabliert. gLite ist aus dem von der Europäischen Union geförderten Projekt *European DataGrid* (EDG, `http://eu-datagrid.web.cern.ch/`) und besonders aus dem späteren Nachfolgeprojekt *Enabling Grids for E-sciencE* (EGEE, `http://www.eu-egee.org/`) hervorgegangen, das unter Einsatz großer finanzieller Mittel erstmals mehr als 70 Institutionen in 27 europäischen Ländern zur Schaffung und Weiterentwicklung einer konsistenten, robusten und sicheren Grid-Infrastruktur verbindet.

Ursprünglich basiert das System auf dem damals schon existenten *LHC Computing Grid* (LCG, `http://lcg.web.cern.ch/LCG/`), das zur Bewältigung der zu erwartenden Datenmengen des aufkommenden *Large Hadron Collider* (LHC, `http://lhc.web.cern.ch/lhc/`) am CERN in Genf installiert wurde und zu diesem Zweck weit verstreute Ressourcen aus wissenschaftlichen Einrichtungen der Hochenergiephysik zusammenführte. gLite erweitert diesen Ansatz nun hin zu einer neu entwickelten, integrierten Grid-Middleware, die eine große Bandbreite an grundlegenden Grid-spezifischen und interoperablen Diensten bereitstellt. gLite versteht sich dabei als leichtgewichtige Middleware, die eine Vielzahl an anerkannten Bausteinen bestehender Plattformen in sich vereint, um hierdurch eine umfassende und modulare Sammlung an Werkzeugen für alle Bereiche der Nutzung von verteilten Ressourcen im Grid zu bieten. Neben Ansätzen aus Projekten wie etwa dem *LHC Computing Grid* oder *Condor* (`http://www.cs.wisc.edu/condor/`) sind in großen Teilen auch Konzepte und Funktionalitäten aus dem zuvor vorgestellten Globus Toolkit in seiner Version 2 (unter anderem die Grid Security Infrastructure) eingeflossen. gLite kombiniert auf diese Weise eine Kern-Middleware auf niedriger Ebene mit einer Reihe an komponierbaren Diensten auf höherem Niveau.

Ähnlich wie beim Globus Toolkit sind Benutzer nicht gezwungen das System als Ganzes

zu nutzen. Stattdessen können auch nur die Teile der Middleware Anwendung finden, die gerade benötigt werden. Daneben lässt sich das System auch an individuelle Anforderungen anpassen. bei der Entwicklung ist vor allem größter Wert auf eine verbesserte Sicherheit und Fortschritte im Management der gesamten Infrastruktur gelegt worden.

Der Einsatz von gLite wird zunehmend auch außerhalb des EGEE-Projekts ernsthaft in Erwägung gezogen, wie etwa im Falle von *DILIGENT* (`http://www.diligentproject.org/`), wo die Middleware bereits produktiv läuft, oder auch der französischen Raumfahrtbehörde *CNES* (`http://www.cnes.fr/`).

2.2.3 Implementierungen zur Sicherheit im Grid

Aufgrund der Komplexität typischer Grid-Szenarien stellt die Entwicklung von Sicherheitslösungen eine große Herausforderung dar. Ressourcen und Anwender sind nicht nur geografisch, sondern auch über Organisationsgrenzen hinweg verteilt, zudem ist ihre Zusammensetzung sehr dynamisch und laufenden Änderungen unterworfen. Eine zusätzliche Anforderung im Grid ist die Delegation von Prozessen, was auch das Starten weiterer Prozesse und das Vererben von Rechten beinhaltet.

2.2.3.1 Grid Security Infrastructure (GSI)

Die *Grid Security Infrastructure* (ehemals die *Globus Security Infrastructure*) ist eine Sammlung von Sicherheitsprimitiven, Protokollen und APIs, die Mechanismen für einige Sicherheitsanforderungen von Grids bieten [Tuecke, 2001]. Sie lässt sich nach ihren Aufgaben grob in drei logische Teile gliedern [Filipovic u. Straub, 2006]:

1. Sichere Kommunikation (d.h. hier vertraulich, authentisch und integer) wird mit etablierten Verfahren der symmetrischen und asymmetrischen Kryptografie unterstützt, wobei dies sowohl kanal- als auch nachrichtenbasiert geschehen kann.

2. Eine *Public Key Infrastructure* (PKI), bestehend aus unabhängigen Zertifizierungsstellen (engl. *Certificate Authorities*), garantiert die Identität der Grid-Teilnehmer (Anwender, Systeme und Dienste).

3. Die temporäre Delegation von Identitäts- und Berechtigungsnachweisen (engl. *Credentials*) sowie ein benutzerfreundliches *Single Sign-On* an Grid-Diensten wird unter Verwendung sogenannter *Proxy-Zertifikate* und darauf aufbauender Werkzeuge realisiert.

Für eine sichere Kommunikation stellt die GSI drei verschiedene Schemata bereit. Zum einen ist dies eine auf TLS (*Transport Layer Security*) basierende Möglichkeit der Kommunikation. TLS bietet während des Verbindungsaufbaus zwischen den Kommunikationspartnern die Möglichkeit flexibel die einzusetzenden kryptographischen Verfahren und ihre Parameter auszuhandeln, so dass verschiedene Sicherheitsniveaus erreicht werden können. Auch besteht die Möglichkeit der starken Authentifizierung. Dieses erste Schema wird als *GSI Transport* bezeichnet und stellt einen sicheren Kommunikationskanal zur Verfügung.

	GSI Secure Conversation	**GSI Secure Message**	**GSI Transport**
Basis	Nachrichten	Nachrichten	Kanal
Technologie	WS-SecureConversation	WS-Security	TLS
Vertraulichkeit	Ja	Ja	Ja
Integritätsschutz	Ja	Ja	Ja
Delegation	Ja	Nein	Nein
Performanz	gut bei vielen Nachrichten	gut bei wenigen Nachrichten	am besten

Tabelle 2.1**:** Vergleich der drei GSI Protection Schemata [Filipovic u. Straub, 2006]

Die beiden weiteren Schemata der GSI basieren jeweils auf dem Schutz der Kommunikation mit Hilfe von Web Services. Zum einen kommt *WS-SecureConversation* im Profil *GSI Secure Conversation* zum Einsatz, zum anderen findet *WS-Security* im Schema *GSI Secure Message* Verwendung. In beiden Fällen wird die Kommunikation auf Basis einzelner Nachrichten abgesichert. Dabei wird ausschließlich der Nutzdatenanteil (engl. *Payload*) der Nachricht verschlüsselt. Nachteilig hierbei ist, dass für jede Nachricht erneut asymmetrische kryptographische Berechnungen vorgenommen werden müssen und durch den Einsatz von Web Services ein zusätzlicher Overhead entsteht. Auf weitere Details zu Sicherheit bei Web Services geht der nachfolgende Unterabschnitt ein. Tabelle 2.1 fasst nochmal die Eigenschaften der drei GSI Schemata zusammen, wie auch in [Filipovic u. Straub, 2006] zu finden.

Im Kontext von Grids existiert eine einheitliche Zertifizierungsstruktur, koordiniert von der *International Grid Trust Federation* (IGTF, `http://www.gridpma.org/`) mit der *Grid Policy Management Authority* (GridPMA) als oberste Instanz. Nationale Zertifizierungsstellen stellen im Sinne der Zuständigkeits- und Aufgabenteilung entweder Personen- oder Maschinen-Zertifikate basierend auf dem X.509 Standard [X.509, 2000] aus. Zertifikatsteilnehmer selber verfügen über sogenannte *End Entity Zertifikate*, so dass sie selber nicht wieder als Zertifizierungsstelle auftreten können. Um trotz dieses Fakts weitere Zertifikate ausstellen zu können, kommen die sogenannten *Proxy-Zertifikate* zum Einsatz. Zum Beispiel das Programm *myproxy*, das unter anderem auch im Globus Toolkit seinen Einsatz findet, erlaubt es Proxy-Zertifikate bereitzustellen. Eine genauere Behandlung der Zertifizierungsketten und Rechtedelegation im Grid ist für diese Arbeit nicht notwendig, weitere Informationen sind in der einschlägigen Literatur nachschlagbar.

Auf Basis der Grid Security Infrastructure lassen sich weitere Mechanismen zur Unterstützung der Sicherheit im Grid entwickeln. Dafür steht das *Generic Security Service Application Programming Interface* (GSS-API) [RFC2743, 2000; RFC2744, 2000] für die einfache Programmierung von Sicherheitsmechanismen zur Verfügung. Die GSI erweitert die GSS-API geringfügig wie in [Meder u. a., 2001] nachzulesen. Beispielhafte Anwendungen, die auf der GSS-API aufsetzen und die GSI unterstützen, sind *GridFTP* als Gridtaugliche Variante des FTP oder auch *GridCVS*.

Abbildung 2.6**:** Web Services Security Specifications

2.2.3.2 Sicherheit mit Web Services

Ein Web Service ist ein Software-System, das die Interoperabilität der Zwischen-Maschineninteraktion gewährleisten soll. In [WebServices, 2004] werden Web Services wie folgt definiert:

> „Definition:
> A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.“

Im Zuge der Standardisierungsbemühungen sind eine Vielzahl an Profilen hervorgebracht worden. Unter anderem ist dabei eine Spezifikation zur Sicherheit mit Web Services, die auch im Kontext von Grids ihre Anwendung findet. Abbildung 2.6 veranschaulicht die wesentlichen Bestandteile dieser Spezifikation, die nachfolgend einzeln kurz erläutert werden.

WS-Security. *WS-Security* (WSS) ist ein Kommunikationsprotokoll, das Mittel für eine sichere Kommunikation von Web Services bereitstellt. Dabei wird ein Rahmen definiert, wie Benutzer- und Sicherheitsinformationen in eine SOAP-Nachricht eingebettet werden können. Es ist möglich sowohl die Vertraulichkeit als auch die Integrität von in einem SOAP-Korpus Informationen zu gewährleisten. Dies geschieht auf einer nachrichtenbasierten Ebene und garantiert eine Ende-zu-Ende Sicherheit. Das WSS Protokoll enthält zusätzlich Informationen zur Nutzung der *Security Assertion Markup*

Language (SAML), Kerberos Tickets und verschiedenen Zertifikaten wie zum Beispiel X.509.

WS-SecurityPolicy. Die WS-SecurityPolicy definiert, wie Policies beschrieben werden können, die im Kontext der in WS-Security spezifizierten Eigenschaften wirksam werden.

WS-Security: SOAP Message Security. Eine Möglichkeit zur Gewährleistung von Vertraulichkeit und Integrität ausgetauschter SOAP-Nachrichten steht hier im Vordergrund. Insbesondere wird eine Unterstützung für den Umgang mit mehreren *Security Token* Formaten, *trust domains*, Signaturformaten und Verschlüsselungsverfahren angeboten.

WS-Security: Username Token Profile. Ein *username token* kann durch Angabe von Nutzername und optional auch einem Passwort ausgestellt werden. Unter Nutzung dieses Tokens kann sich so ein Anwender gegenüber einem Web Service authentifizieren.

WS-Security: Kerberos Binding. Das Anhängen von Kerberos Tickets an SOAP-Nachrichten ist Bestandteil des WS-Security: Kerberos Binding. Außerdem wird das Hinzufügen von Signaturen und Verschlüsselung im Einklang mit WS-Security zu SOAP-Nachrichten spezifiziert.

WS-Federation. Das Management und die Vermittlung von Vertrauensbeziehungen in heterogenen föderierten Umfeldern ist Inhalt der WS-Federation.

WS-Security: SAML Token Profile. Hiermit wird die Nutzung von *Security Assertion Markup Language* (SAML) Assertions im Kontext der WS-Security: SOAP Message Security spezifiziert. Dies umfasst auch den Zweck der Absicherung von SOAP-Nachrichten.

WS-Trust. WS-Trust beschreibt ein Rahmenwerk für Vertrauensmodelle, die es Web Services erlauben sicher zu interoperieren. Es nutzt die Basismechanismen von WS-Security und spezifiziert zusätzliche Primitive und Erweiterungen um Sicherheits-Token auszutauschen.

WS-Security: X.509 Certificate Token Profile. Hierbei handelt es sich um die Spezifikation zur Nutzung von X.509 Zertifikaten zur Authentifizierung im Rahmen der WS-Security: SOAP Message Security Spezifikation.

WS-SecureConversation. WS-Security stellt Basismechanismen für den sicheren Nachrichtenaustausch unter Nutzung von Web Services bereit, auf die WS-SecureConversation aufsetzt. In Abgrenzung zu WSS wird ein Sicherheitskontext spezifiziert, der es ermöglicht Sitzungsschlüssel oder auch andere/weitere Schlüsselpaare auszutauschen und dadurch die Performanz für den Austausch mehrerer Nachrichten erhöht. Hierzu ist das sogenannte *Security Context Token* (SCT) spezifiziert. Genau wie WS-Security gewährleistet WS-SecureConversation die Vertraulichkeit und Integrität der ausgetauschten Daten nachrichtenbasiert.

Kapitel 3

Anforderungsanalyse

Inhalt des Kapitels

Ziel der vorliegenden Arbeit ist der Entwurf eines Frühwarnsystems für Grid-Umgebungen. Dabei steht wie bereits angedeutet die Idee der kooperativen Nutzung existierender Sicherheits- und Logging-Mechanismen aller an einem Grid-basierten Intrusion Detection System (GIDS) beteiligter Partner im Vordergrund. In den nachfolgenden Abschnitten werden Anforderungen an ein solches Grid-Frühwarnsystem erhoben, um schlussendlich in Abschnitt 3.4 einen Kriterienkatalog zur Konzeption und Bewertung von Grid-Frühwarnsystemen aufzustellen.

Bei der Konzeption eines Sicherheitssystems steht an erster Stelle stets eine Bedrohungsanalyse wie unter anderem auch in [Eckert, 2007; Oberhaitzinger u. a., 2004] näher beschrieben. Eine solche Bedrohungsanalyse im Falle von Grids wird in Abschnitt 3.1 vorgenommen und orientiert sich an den Ausführungen in [Oberhaitzinger u. a., 2004]. Dabei geht Abschnitt 3.1.1 auf die *Angriffsziele und Risiken* im Grid ein. Dies passiert insbesondere auf Basis der in Abschnitt 2.2.1 erarbeiteten Erkenntnisse zu Konzepten und Architekturen im Grid-Umfeld. Zur besseren Veranschaulichung können die in Abschnitt 2.2.2 näher beschriebenen Implementierungen von Grid-Middleware Lösungen herangezogen werden. In den Abschnitten 3.1.2 und 3.1.3 wird eine Klassifikation der potentiellen Angreifer und die daraus resultierenden *Angriffsmuster* inklusive ihres *Ausgangsorts* (örtliche und organisatorische Positionierung des tatsächlichen Angreifers) sowie der *Angriffstypen* in Grids vorgenommen. Der abschließende Teil der Bedrohungsanalyse diskutiert in Abschnitt 3.1.4 die *Schutzzieldefinition* für ein Grid-Frühwarnsystem.

Abschnitt 3.2 setzt sich anschließend mit der Analyse der *Nutzergruppen* bzw. des *Kunden* eines GIDS auseinander. Zusätzlich, da der grundlegende Gedanke bei der Konzeption eines solchen GIDS die kooperative Nutzung bereits verfügbarer Informationsquellen ist, geht Abschnitt 3.2.3 näher auf mögliche *Informationsanbieter* für ein Frühwarnsystem im Grid-Szenario ein. Beiden zuvor genannten Gesichtspunkten wird durch eine Anwendungsfall-getriebene Anforderungsanalyse Rechnung getragen (für genauere Informationen zum Vorgehen siehe auch den einleitenden Teil zu Abschnitt 3.2 auf Seite 42). Als Anwendungsfall dient hierbei das D-Grid. Als tatsächlich existierendes Grid-Szenario erhebt dieses einen Anspruch auf Realitätsnähe und bietet somit eine gute Ausgangsbasis für das weitere Vorgehen.

Ein abschließender Teil setzt sich in Abschnitt 3.3 mit der Diskussion generischer, durch Grids bedingte Anforderungen auseinander. Dazu werden zuerst technische und organisatorische Anforderungen, die im Grid für eine jede Komponente gelten (siehe auch Abschnitt 2.2.1), für ein Frühwarnsystem in Grid-Umgebungen instanziiert. Da wie eingangs

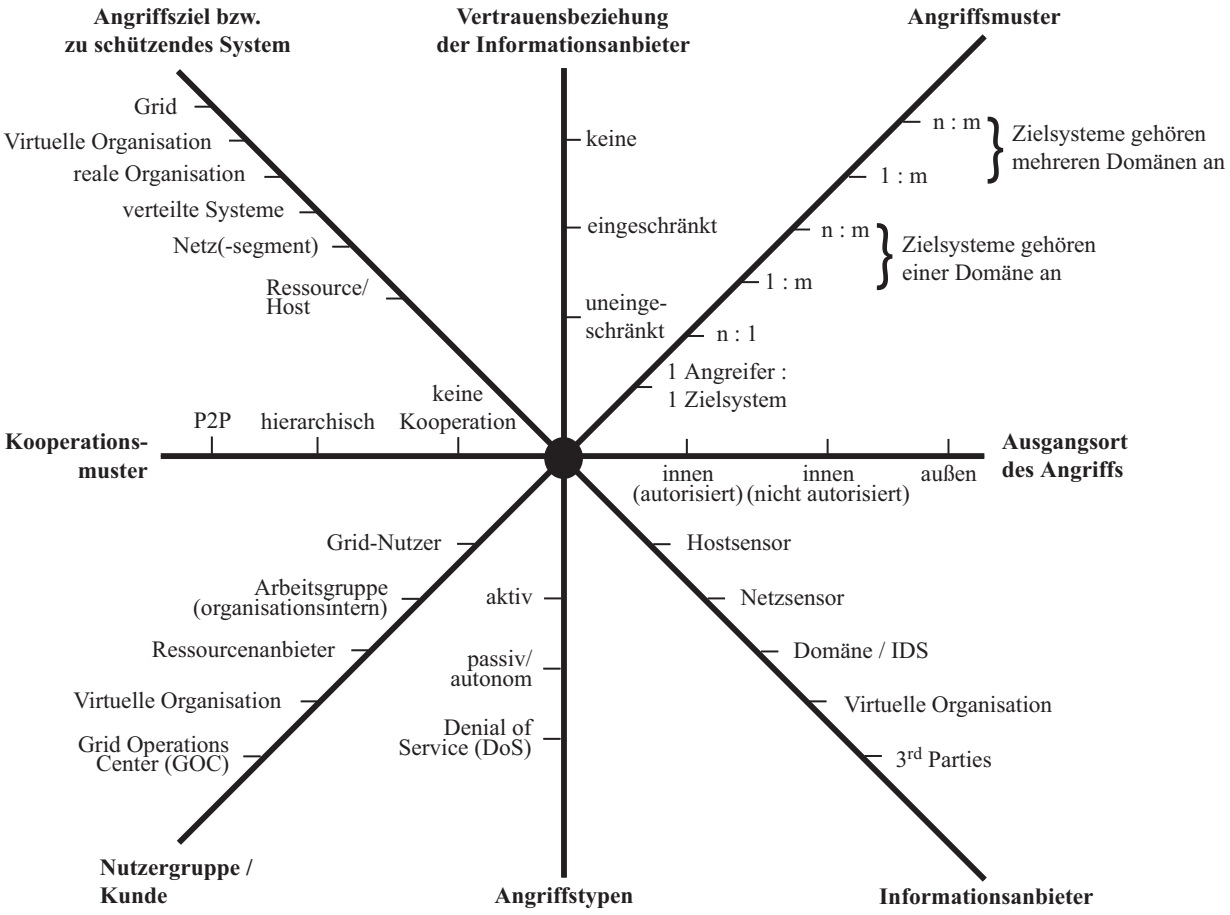


Abbildung 3.1**:** Dimensionen der Problemstellung

erwähnt der kooperative Zusammenschluss bestehender Sicherheitskomponenten zu einem Gesamtsystem im Vordergrund des zu erarbeitenden Konzepts stehen soll, gehen die Abschnitte 3.3.2 und 3.3.3 genauer auf die denkbaren *Kooperationsmuster* unter Informationsanbietern sowie deren *Vertrauensbeziehungen* untereinander ein.

Abbildung 3.1 stellt den Dimensionenraum der Problemstellung nochmals zusammenfassend dar, der sich nun aus den obigen Ausführungen erschließt. Die Dimensionen entsprechen dabei den zuvor *kursiv* gedruckten Begriffen. Die einzelnen Achsen des Problemraums werden in den Abschnitten 3.1 bis 3.3 einzeln bearbeitet, was in Abschnitt 3.4 schlussendlich zur Ableitung eines erwünschten Kriterienkatalogs zur Konzeption und Bewertung von Grid-Frühwarnsystemen führt.

3.1 Bedrohungsanalyse

Die nachfolgende Bedrohungsanalyse orientiert sich in der Vorgehensweise an [Oberhaitzinger u. a., 2004]. Dabei wird untersucht, welche möglichen Gefahren und Angriffe eine Ressource gefährden und welche Auswirkung ein erfolgreicher Angriff haben kann. Eine darauf folgende Risikobewertung im Sinne des Security Engineerings[1] wird nicht durchgeführt. Die entscheidenden Faktoren, um eine Risikobewertung vornehmen zu können, hängen von Szenario-spezifischen Faktoren ab, die auf dieser Ebene nicht weiter bewertet werden können. So ist z.B. die Eintrittswahrscheinlichkeit eines erfolgreichen Angriffs essentiell von den jeweiligen Sicherheitsvorkehrungen (Firewalls, Security Policies, Software Patch-Level etc.) und den Interessen potentieller Angreifer (wie interessant ist das Grid für einen Angreifer?) abhängig. Auch der durch einen erfolgreichen Angriff angerichtete Schaden ist vom jeweiligen Einzelfall abhängig.

3.1.1 Angriffsziele & Risiken im Grid

In Abschnitt 2.2.1 sind die Dienste und Komponenten, die ein Grid auszeichnen, identifiziert worden. Eine jede Komponente im Grid bzw. jeder erbrachte Dienst kann potentiell das Ziel eines Angriffs sein, weswegen nachfolgend die einzelnen potentiellen Angriffsziele und mögliche Folgen, die aus erfolgreich durchgeführten Angriffen resultieren können, diskutiert werden.

Grid-Security. Im Bereich der Sicherheitsmechanismen ist es sehr selten, dass Konzepte und Algorithmen zum Beispiel zur Verschlüsselung, Gewährleistung von Nachrichtenintegrität oder auch der Signierung angegriffen werden. In den meisten Fällen handelt es sich bei Angriffen in diesem Bereich um Angriffe auf Schwachstellen in den Implementierungen [Bleichenbacher, 1998; Jonsson u. Kaliski, 2002; Klíma u. a., 2003].

Verschlüsselung. Der Schaden, der durch einen erfolgreichen Angriff auf die Vertraulichkeit und Integrität von Nachrichten und Informationen im Grid entsteht, ist erheblich von der Anwendung abhängig. Dabei erscheinen Fälle wie zum Beispiel medizinische Forschung oder auch Industriespionage deutlich kritischer als beispielsweise das Kompromittieren bzw. Abhören wissenschaftlicher Ergebnisse, die ohnehin in der Regel offen gelegt werden. Für beide Szenarien bietet u.a. das D-Grid Projekt Beispiele wie auch in Abschnitt 3.2 detailliert wird.

Wie bereits einleitend erwähnt, wird im Bereich der Nachrichtenverschlüsselung und Gewährleistung der Nachrichtenintegrität selten ein Konzept, sondern es werden vielmehr die Implementierungen angegriffen. Damit erscheint ein aktueller Stand der Softwareprodukte zur Gewährleistung dieser Sicherheitsaspekte essentiell.

[1] $R(x) = E(x) * S(x)$; mit $R(x)$: Risiko, das durch einen Angriff x besteht, $E(x)$: Eintrittswahrscheinlichkeit eines erfolgreichen Angriffs x, $S(x)$: Schadenshöhe, die durch einen erfolgreichen Angriff x entsteht

Public Key Infrastructure. Das Risiko eines Außerbetriebnehmens einer *Public Key Infrastructure* (PKI) inklusive ihrer Zertifizierungsstellen im Grid-Umfeld ist wohl ärgerlich, bedingt durch das Konzept des Ausstellens und Signierens von Zertifikaten, aber nicht kritisch für den Betrieb der sonstigen Grid-Infrastruktur[2]. Da für die Verifikation der Signatur beispielsweise eines Zertifikats ausschließlich der öffentliche Schlüssel (*engl. public key*) der ausstellenden Einrichtung bekannt sein muss, ist deren Funktionstüchtigkeit zum Zeitpunkt der Überprüfung eines Zertifikats oder einer Signatur nicht von Nöten. Die öffentlichen Schlüssel der Zertifizierungsstellen im Grid sind im Vorhinein unter großen Vorkehrungen zur Gewährleistung der Informationsintegrität bereits ausgetauscht worden und somit lokal bei jedem Grid-Teilnehmer vorhanden.

Deutlich kritischer hingegen ist die Erlangung des privaten Schlüssels (*engl. private key*) einer Zertifizierungsstelle. Dies kann zum einen durch den erfolgreichen Einbruch in die Systeme der Zertifizierungsstelle oder auch einen Brute-Force Angriff [3] geschehen. Durch das Erlangen des privaten Schlüssels einer Zertifizierungsstelle wäre ein Angreifer in der Lage sich oder anderen beispielsweise unerlaubten Zugriff auf Grid-Ressourcen und -Dienste in fremdem Nutzerkontext zu gewähren, ohne dass dies durch die vorgesehenen Prüfverfahren auffallen würde. Weiter könnte er zusätzliche Ressourcen oder auch Nutzer durch das Ausstellen entsprechender Ressourcen- oder Nutzerzertifikate nicht autorisiert in das Grid einbringen und dadurch beispielsweise an ihm sonst unzugängliche Informationen oder Dienste gelangen.

Für den Fall, dass der private Schlüssel einer Zertifizierungsstelle in fremde Hände gelangt, sollte dringlichst ein Prozess spezifiziert werden. Hierzu empfiehlt sich die Anlehnung an ein Prozessrahmenwerk wie beispielsweise die *IT Infrastructure Library* (ITIL) [Office of Government and Commerce (OGC), 2007] oder auch die ISO/IEC 20000 [ISO/IEC 20000-1, 2005; ISO/IEC 20000-2, 2005]. Eine genauere Beschreibung der ITIL sowie Vorschläge zur Werkzeugunterstützung führt u.a. Michael Brenner in seiner Dissertation aus [Brenner, 2007].

Das widerrechtliche Erlangen eines privaten Schlüssels einer Ressource oder eines Nutzers hingegen ist weniger kritisch. Sobald ein solcher Diebstahl bekannt wird, bieten PKIs die wenig aufwendige Möglichkeit des Revozierens eines Zertifikats unter Zuhilfenahme der sogenannten *Certificate Revocation List* (CRL).

Grid-Monitoring. Angriffe auf das Grid-Monitoring erscheinen sehr interessant, da das Monitoring im Grid die Informationsbasis insbesondere für die nachgeschalteten Prozesse des Accountings und Billings bereitstellt. Ein Denial-of-Service Angriff auf das

[2] Auf das genaue Vorgehen zur Ausstellung von Zertifikaten und Signierungsvorgänge wird an dieser Stelle nicht weiter eingegangen, vielmehr wird ein grundlegendes Verständnis und Wissen vorausgesetzt. Hintergrundinformationen und detaillierte Beschreibungen liefern insbesondere Standardwerke wie z.B. [Eckert, 2007].

[3] Das systematische Berechnen des privaten Schlüssels durch zum Beispiel erschöpfendes Ausprobieren aller in Frage kommenden Schlüssel wird in diesem Zusammenhang als Brute-Force Angriff bezeichnet.

Monitoring ist dabei der eher uninteressante Fall, da dieser sehr schnell auffallen und dadurch angerichteter Schaden bekannt ist und somit ggf. wieder behoben werden kann. Für einen Angreifer deutlich interessanter ist die erfolgreiche Kompromittierung eines Monitoring-Systems um so zum Beispiel eine falsche Rechnungsstellung unbemerkt zu erwirken. Auch ein Dienstanbieter könnte ein Interesse an verfälschten Monitoring-Informationen haben um beispielsweise die Verletzung eines Service Level Agreements (SLA) zu verschleiern und daraus resultierende Pönalen zu umgehen.

Producer & Consumer. Angriffe auf sowohl Producer als auch Consumer eines Monitoring-Systems haben sehr ähnliche Effekte. Denkbare Angriffe auf ein Monitoring-System können beispielsweise das zeitweise Blockieren oder auch Außerbetriebnehmen (Denial-of-Service) von (Teil-)Bereichen des Monitorings sein um somit z.B. eine widerrechtliche oder auch unbemerkte Nutzung von Grid-Ressourcen zu verschleiern. Angriffe auf die Korrektheit von Monitoring-Informationen (sowohl an der Stelle des Producers als auch des Consumers) hingegen können auch für einen Dienstanbieter attraktiv werden um potentielle Vertragsbrüche zu vertuschen.

Directory Service. Da im Sinne der Grid Monitoring Architecture (GMA, siehe auch Abschnitt 2.2.1) ein Verzeichnisdienst „nur“ eine Liste verfügbarer Producer und Consumer und der von ihnen bereitgestellten bzw. verarbeiteten Informationen anbietet, ist der Directory Service für ein Monitoring-System im Betrieb weniger kritisch. Ein längerer Ausfall sollte jedoch dennoch dringlichst vermieden werden, gerade da durch die GMA in der Regel genau ein Verzeichnisdienst gefordert wird. Im Falle mehrerer solcher Verzeichnisdienste ist eine hierarchische Organisation selbiger vorgesehen, so dass sich auch in diesem Fall ein singulärer Angriffspunkt auf den Directory Service bietet.

Grid-Scheduler. Im Bereich des Schedulings ist zwischen zwei unterschiedlichen Verfahrensweisen, die beide in zurzeit produktiven Umgebungen gebräuchlich sind, zu unterscheiden. Zum einen kann ein zentraler *Broker* zum Einsatz kommen, an den ein jeder Grid-Job abgesetzt werden muss. Der Broker vermittelt die Jobs zur Verarbeitung weiter an eine geeignete Grid-Ressource. Alternativ können auch lokale Scheduler zum Einsatz kommen, die die eingehenden Jobs entsprechend ihrer Richtlinien verarbeiten. In ersterem Fall stellt selbstverständlich ein zentraler Broker auch eine für den Betrieb des gesamten Grids entscheidende Komponente dar, die sehr schützenswert ist. In zweiterem Fall, der Fall mehrerer (Site-lokaler) Scheduler, hingegen ist durch die Verteilung eine Vielzahl erfolgreicher Angriffe zur Störung des Grid-Betriebs notwendig.

Denkbare Effekte, die erfolgreich ausgeführte Angriffe nach sich ziehen können, ist zum einen das Außerbetriebnehmen einzelner oder auch aller Grid-Ressourcen oder auch das unrechtmäßige bevorzugte Behandeln einzelner Grid-Jobs.

Grid-Portale und Portlets. Grid-Portale sowie Portlets stellen nur ein vergleichsweise geringes Risiko im Falle ihrer Kompromittierung dar. Kam insbesondere bei Grid-Portalen der ersten Generation noch oftmals eine Nutzerauthentifizierungen per Eingabe von Nutzername und Passwort zum Einsatz, so sind die meisten aktuellen Portale und Portlets voll in die Grid-weite Nutzerverwaltung und somit eine zertifikatsbasierte Authentifizierung eingebunden. Zudem dienen Portale vielmals rein zur Information, eine steuernde Funktionalität und somit eine Einflussnahme auf den Betrieb des Grids ist nur in manchen Fällen vorgesehen.

VO-Management. Ein Angriff auf das VO-Management bedeutet im Grid in den meisten Fällen ein Angriff auf die etablierte AA-Infrastruktur. Dabei sind insbesondere die *Dienstanbieter und ihre Dienste*, der vermittelnde *Where-are-you-from Dienst* inklusive der beteiligten *Identity Provider* involviert.

Dienstanbieter und seine Dienste. Ein Angriff gegen einen Dienstanbieter und seine erbrachten Dienste inklusive der zur Erbringung notwendigen Ressourcen (z.B. Hintergrundspeicher, Rechensysteme, Netz- und Infrastrukturkomponenten) entspricht einem eher klassischen Angriff, wie er von konventionellen (verteilten) Systemen her bekannt ist.

Eine zusätzliche Herausforderung, die für Angriffsziele im Grid und insbesondere die angebotenen Dienste besteht, resultiert aus der Eigenschaft der dezentralen Organisation eines Grids. In Abschnitt 2.2.1 ist nach [Foster, 2002] festgestellt worden, dass die im Grid verbundenen Ressourcen sich über viele juristisch, organisatorisch und administrativ autonome und geografisch verteilte Unternehmen erstrecken können. Insbesondere bedeutet dies, dass ein Angriff gegen einen im Grid erbrachten Dienst oder eine Komponente im Grid in aller Regel gleichzeitig auf Zielsysteme, die mehreren verschiedenen Domänen angehören, gerichtet sein kann, ob implizit durch das Angreifen eines verteilt erbrachten Dienstes oder verteilt installierter Komponenten oder auch explizit.

Where-are-you-from Dienst. Der WAYF Dienst stellt eine kritische Komponente für die Funktionsfähigkeit eines Grids dar. Wie in Abbildung 2.4 auf Seite 19 dargestellt ist, stellt der WAYF Dienst eine zentrale Instanz bei der Authentifizierung der Nutzer im Grid dar. Ohne diesen Dienst ist vom ursprünglichen Gedankengang her kein Dienstanbieter in der Lage auf Grund fehlender Authentifizierung über die Zulässigkeit und die Modalitäten der Nutzung des angebotenen Dienstes zu entscheiden. Folglich muss eine Dienstnutzung vom Dienstanbieter fehlerbedingt untersagt werden.

Eine erfolgreiche Außerbetriebnahme des WAYF Dienstes oder auch eine falsche Abbildung von Nutzern zu ihren Identity Providern kann somit eine Betriebsunterbrechung des gesamten Grids zur Folge haben. Bereits authentifizierte und autorisierte Nutzer könnten zwar weiterhin ihre in Verwendung befindlichen Dienste

ausführen, neue Anfragen hingegen können in so einem Fall nicht korrekt verarbeitet werden.

Identity Provider. Ein besonders interessantes Angriffsziel bei einem Identity Provider sind die zumeist zur Nutzerverwaltung angeschlossenen Verzeichnisdienste. Im Gegensatz zum WAYF Dienst führt eine Betriebsstörung im Sinne eines Dienstausfalls „nur“ zur Beeinträchtigung der von diesem IdP verwalteten Nutzer, jedoch drohen durch die Manipulation der Verzeichnisdienste weitere Folgen. Da ein Identity Provider die tatsächlichen Anwenderdaten zur Authentifizierung verwaltet und nicht wie der WAYF Dienst nur vermittelt, können Folgen einer Kompromittierung beispielsweise das unrechtmäßige Einbringen neuer oder das Löschen bestehender Anwender sein oder auch das Abändern von beim IdP gespeicherten Attributswerten eines Nutzers, was die Entscheidungsfindung eines Dienstanbieters zur Gewährung des Zugangs zu einem Dienst beeinflussen kann.

3.1.2 Klassifikation der Angreifer in Grids

Die Klassifikation von Angreifern fußt vorrangig auf zwei Aspekten. Zum einen sind dies die Qualifikation und Mittel (sachlich und zeitlich), die einem Angreifer zur Verfügung stehen. Zum anderen ist die örtliche und organisatorische Positionierung von Bedeutung.

3.1.2.1 Voraussetzungen eines Angreifers

In [Oberhaitzinger u. a., 2004] wird eine Unterteilung potentieller Angreifer auf eine IT-Infrastruktur nach den drei Kriterien *Ressourceneinsatz*, *Know-How* und *Zeiteinsatz* vorgenommen. Diese Charakterisierung mündet in einem aufgestellten Angreifermodell, das in 3.2 dargestellt ist. Hierin wird insbesondere die Unabhängigkeit der drei Dimensionen der Angreifercharakterisierung verdeutlicht. Dieses Modell dient im weiteren Verlauf im Rahmen dieser Arbeit als Orientierung zur Klassifikation der Angreifer in Grids.

Ressourceneinsatz. In Bezug auf die Ressourcen, die einem Angreifer für seine Zwecke zur Verfügung stehen, ist zum einen nach der Leistungsfähigkeit und auf der anderen Seite nach der Anzahl der Ressourcen zu differenzieren. Dabei ermöglicht eine hohe Anzahl verfügbarer Systeme die Durchführung massiv verteilter Angriffe oder auch die Maskierung eines Angriffs durch die zufällige Nutzung unterschiedlicher Systeme. Zusätzlich kann mit einer Vielzahl an Endsystemen durch die Bündelung ihrer Kapazitäten ein ähnlicher Effekt wie durch ein sehr leistungsfähiges System erzielt werden. Diese Kapazitäten können insbesondere zur Unterstützung von Denial-of-Service Angriffen durch künstliche Erzeugung von Überlastsituationen beim Angriffsziel oder zum Brechen kryptographischer Verfahren missbräuchlich eingesetzt werden.

Durch den Ressourceneinsatz bedingt kann ein Angriff verteilt von mehreren Systemen ausgehen. Zusätzlich, durch verteilte Systeme bedingt (also insbesondere auch Grids), kann auch das Angriffsziel verteilt sein, woraus die in Abbildung 3.1 auf Seite 33 illustrierte Achse der unterschiedlichen Angriffsmuster resultiert.

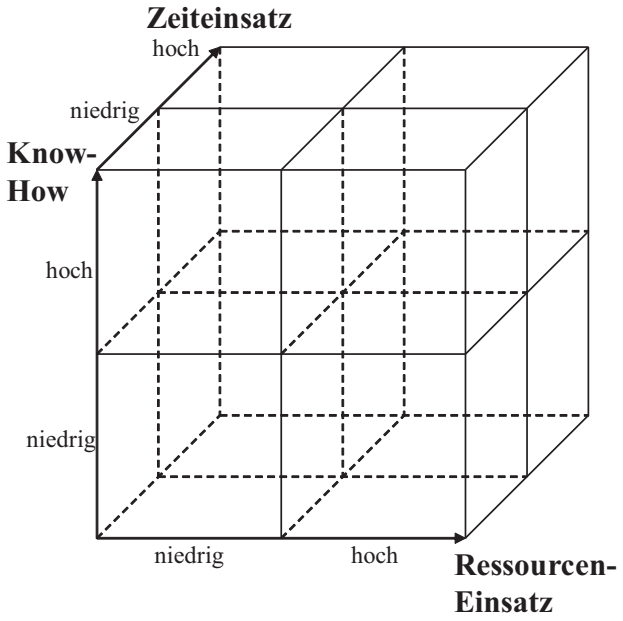


Abbildung 3.2**:** Angreifermodell nach [Oberhaitzinger u. a., 2004]

Know-How. In Bezug auf das Vorwissen möglicher Angreifer ist auch im Falle von Grids von der gesamten Spannbreite auszugehen. Gerade da Grids in der Regel große Angriffsziele mit einer hohen Attraktivität für Angreifer durch ihre enormen Kapazitäten bieten, sind sowohl Angriffe durch meist unwissende *Script Kiddies* bis hin zu (professionellen) hoch qualifizierten Angreifern denkbar.

Script Kiddies stellen die größte Gruppe unter allen Angreifern dar. Ihr Wissen ist dabei gering und sie setzen leicht bedienbare, vorgefertigte Programme zum Zwecke ihrer Angriffe ein, meist ohne weiteres Bewusstsein über die Art und Weise ihres Angriffs oder dessen Folgen. Da sie relativ weit verbreitete Programme verwenden um ihre Angriffe durchzuführen, sind diese Angriffsmuster vergleichsweise einfach zu erkennen und Script Kiddies sind somit in eine geringe Gefährdungsklasse einzuordnen. Eine vielfach eingesetzte Methode zur Erkennung von Angriffen durch Script Kiddies ist der Einsatz von signaturbasierten Erkennungsmechanismen (siehe auch Abschnitt 2.1.2), der sich sehr durch das vielfache Auftreten bekannter Muster bewährt. Somit ergibt sich durch Script Kiddies in jedem Fall die Notwendigkeit der sinnvollen Sensorplatzierung eines IDS um Angriffe vollständig beobachten zu können. Außerdem ist das Etablieren eines Prozesses (sinnvoller Weise automatisiert oder zumindest durch Tools unterstützt), um Angriffssignaturen bzw. Erkennungsmerkmale verbreiteter Schadprogramme möglichst zeitnah systemweit verfügbar zu machen, notwendig, da die Aktualität der Angriffssignaturen einen kritischen Faktor bei der Bewertung der Erkennungsleistung eines signaturbasierten Frühwarnsystems darstellt.

Eine deutlich kleinere Gruppe an Angreifern verfügt über eine sehr hohe Qualifikation

und meist sehr genaue Kenntnisse über ihr Angriffsziel. In den Reihen solch professioneller Angreifer kommen weit verbreitete Schadprogramme wie im Falle von Script Kiddies eher selten zum Einsatz. Oftmals werden hingegen vom Szenario abhängige Angriffsstrategien geplant und gezielt durchgeführt. Dadurch ist die Erkennung solcher Angriffe sehr schwer, was schlussendlich diese vergleichsweise kleine Gruppe an Angreifern in eine hohe Gefährdungsklasse einordnet. Eine Möglichkeit ihre Angriffe zu erkennen stellt z.B. der Einsatz von Anomalieerkennungsverfahren dar (siehe auch Abschnitt 2.1.2).

Zeiteinsatz. Insbesondere Angriffe, die über einen sehr langen Zeitraum sehr langsam ausgeführt werden, sind für ein jedes Frühwarnsystem kritisch zu erkennen. Dies liegt insbesondere an der Tatsache, dass sich Angriffsmuster über einen sehr großen Zeitraum und daraus folgend auf einen sehr großen Datenbestand an durch ein IDS beobachteten Ereignissen erstrecken. Zur korrekten und erfolgreichen Erkennung solcher Angriffe ist folglich auch die zusammenhängende Analyse des gesamten Datenbestands notwendig, so dass eine Vielzahl (zumindest relevanter) Daten stets für das Frühwarnsystem zur Verfügung stehen sollten. Welche Zeiträume und welcher Detailgrad der Informationen notwendig ist, ist im Einzelfall zu entscheiden und vom Schutzbedarf sowie der Notwendigkeit der Nachvollziehbarkeit und Berichterstattung zur Sicherheit abhängig.

3.1.2.2 Ausgangsort eines Angriffs und Rechte des Angreifers

Zur Klassifikation von Angreifern gehört genauso wie ihre Qualifikation auch ihre örtliche und organisatorische Positionierung, die sie zur Ausübung ihres Angriffs einnehmen. Dabei wird die örtliche Position unterschieden nach einem dem Grid angehörigen System im Gegensatz zu einem dem Grid nicht zugehörigen System, von dem aus ein Angriff ausgeführt wird. Im weiteren Verlauf wird von einem Angriff von „innen“ gesprochen, falls der Ausgangsort des Angriffs eine im Grid autorisierte Ressource ist. Ist dies nicht der Fall, so wird von einem Angriff von „außen“ oder auch „extern“ gesprochen.

Eine zusätzliche Unterscheidung eines Angreifers muss bezüglich seiner organisatorischen Position vorgenommen werden. D.h. es muss unterschieden werden, ob dieser autorisiert oder nicht autorisiert für den Zugriff auf Grid-Ressourcen ist. Ist er nicht autorisiert, so impliziert dies einen externen Angriff, da er keinen Zugriff auf ein System im Grid haben kann. Sollte er dennoch den Status eines autorisierten Grid-Nutzers widerrechtlich oder missbräuchlich erlangen und für seine Zwecke ausnutzen, so ist dies implizit ein Angriff von innen (autorisiert), da der Angreifer nun scheinbar legitimer Grid-Nutzer und somit Teil des Grids ist.

Somit ergeben sich im Wesentlichen drei Kategorien von möglichen Ausgangsorten für Angriffe: von innen (autorisiert), von innen (nicht autorisiert) und von außen.

3.1.3 Klassifikation der Angriffe in Grids

Zur Klassifikation von Angriffen in einem jeden System finden sich in der Literatur mehrere Taxonomien. Unter anderem sind auch in [Oberhaitzinger u. a., 2004] und [Eckert, 2007] solche Taxonomien aufgeführt, die naturgemäß auch auf Grids anwendbar sind.

In [Oberhaitzinger u. a., 2004] wird eine Unterteilung von Angriffen im Wesentlichen in drei Kategorien vorgenommen.

Aktive Angriffe. Ein vorsätzlicher und kontrollierter Angriff auf ein System wird als aktiver Angriff beschrieben. Bei einem solchen Angriff ist es das Ziel nach erfolgreicher Durchführung Zugriff auf das angegriffene System zu haben, nicht jedoch das System außer Betrieb zu setzen. Beispiele für aktive Angriffe sind unter anderem Spoofing-Angriffe (also die Vorgabe falscher Tatsachen) wie IP-Spoofing, Angriffe auf Passwörter oder auch Rootkits und Backdoors.

Passive Angriffe. Passive Angriffe werden oftmals auch als autonome Angriffe bezeichnet. In Abgrenzung zu aktiven Angriffen werden passive Angriffe durch eigenständige Programme oder Programmteile durchgeführt, die in aller Regel in der Lage sind selbstständig und ohne notwendige Nutzerinteraktion weitere Systeme anzugreifen. Prominente Beispiele dieser Angriffsklasse sind insbesondere Würmer und Viren.

Denial-of-Service Angriffe. Ein Denial-of-Service (DoS) Angriff verfolgt das Ziel des Blockierens oder Außerbetriebsetzens eines Dienstes oder eines gesamten Zielsystems. Dazu werden in vielen Fällen Fehler oder fehlerhafte Implementierungen in Programmen und Betriebssystemen ausgenutzt. Eine Variante des DoS Angriffs ist der verteilte oder auch Distributed Denial-of-Service (DDoS) Angriff, bei dem eine Vielzahl verteilt positionierter Angreifer koordiniert das gleiche System angreifen. Ältere, aber prominente Beispiele für DoS Angriffe sind unter anderem SYN-Flooding Angriffe oder auch Flood Pings.

In [Eckert, 2007] wird in der Definition eines Angriffs die Unterscheidung in aktive und passive Angriffe vorgenommen. Die Belegung der Begrifflichkeit ist wie bereits zuvor dargelegt, jedoch wird ein Angriff gegen die Verfügbarkeit eines Systems (also in hiesiger Nomenklatur ein Denial-of-Service Angriff) als aktiver Angriff eingestuft.

Für den weiteren Verlauf der Arbeit wird die feinergranulare Unterteilung in aktive, passive und DoS Angriffe nach [Oberhaitzinger u. a., 2004] herangezogen.

3.1.4 Schutzzieldefinition

Im Rahmen einer Risikoanalyse und bei der Erstellung eines Sicherheitskonzepts gilt es die Schutzziele im Form von Daten, Ressourcen und Diensten sowie das Schadenspotential, das durch ihren Verlust oder Ausfall bedingt ist, zu beziffern. Als notwendige Schutzziele im Grid ergeben sich entsprechend alle Grid-Ressourcen und -Dienste. Dies impliziert direkt, dass alle am Grid beteiligten Organisationen im Fokus des Schutzes stehen und somit

insbesondere die beteiligten Ressourcenanbieter mit unter den Schutz des GIDS fallen. Zusätzlich zu den Anwendungsdaten und weiteren Informationen, die im Grid hinterlegt sind, ist natürlich das zu entwickelnde GIDS selbst als potentielles Angriffsziel ebenfalls schutzbedürftig. Dies ist jedoch bereits durch das Postulat alle Grid-Ressourcen und -Dienste zu schützen mit eingeschlossen, da das GIDS selber nun wahrlich als ein Grid-Dienst einzustufen ist.

An dieser Stelle kann und soll keine weitere Risikoanalyse im Sinne des Security Engineerings vorgenommen werden. Das Schadenspotential der jeweilig verfügbaren Daten, Ressourcen und Dienste sowie die Eintrittswahrscheinlichkeit eines erfolgreichen Angriffs ist inhärent vom jeweiligen Einsatzszenario abhängig und somit im einzelnen zu klassifizieren. Im Rahmen dieser Arbeit ist es hinlänglich sich der potentiellen Angriffsziele und somit des Schutzbedarfs bewusst zu sein.

3.2 Anwendungsfall-getriebene Analyse von Anforderungen an GIDS

Die in den folgenden Unterabschnitten durchgeführte Anforderungsanalyse zur Erhebung von Anforderungen an Frühwarnsysteme in Grid-Umgebungen soll Anwendungsfall-getrieben erfolgen. Die dabei angewandte Methodik folgt der Vorgehensweise der objektorientierten Analyse von Anwendungsfällen (sogenannte *use cases*) wie auch u.a. in [Hennicker, 2006] und [Bruegge u. Dutoit, 2003] näher beschrieben. Der Ausgangspunkt der Anwendungsfallanalyse ist die knappe und informelle Beschreibung ausgewählter Szenarien, die aktuellen Grid-Projekten entspringen und somit einen Anspruch auf Praxisnähe erheben. Abbildung 3.1 zeigte zur Strukturierung des Vorgehens die unterschiedlichen Dimensionen des Problemraums für Frühwarnsysteme in Grid-Umgebungen auf. So gilt es insbesondere bei der Auswahl von Anwendungsfällen die Aspekte der Nutzergruppe bzw. des Kunden eines zu entwickelnden Systems (siehe Unterabschnitt 3.2.2) sowie der verfügbaren Informationsanbieter (siehe Unterabschnitt 3.2.3) hinreichend zu analysieren wie bereits in Kapitel 3 diskutiert.

Bei der Anwendungsfallanalyse wird zuerst eine informelle Beschreibung eines Anwendungsfalls, der aus einem aktuellen Grid-Projekt entspringt, gegeben. Mittels Abstraktion und Generalisierung wird daraus ein sogenanntes *Use Case-Modell* generiert, welches aus *Aktoren* und *Anwendungsfällen* (den *Use Cases*) besteht und eine externe Sicht auf das System beschreibt. Dieses Use Case-Modell modelliert in diesem speziellen Fall das zu beobachtende System (hier also das Grid) mit Fokus auf bzw. aus Sicht des beobachtenden Systems (hier also das Frühwarnsystem).

Ein *Aktor* ist in diesem Modell eine Entität, die von außen Informationen mit dem beschriebenen System austauscht. Aktoren werden dabei durch ihre Rolle, die sie gegenüber dem System einnehmen, charakterisiert und im nachfolgenden jeweils (zusätzlich zu ihrer

ausführlichen textuellen Beschreibung) durch eine Tabelle wie z.B. Tabelle 3.1 zusammengefasst.

Aktor *beispielhafter Aktor*	
Bezeichner	Aktor:ID
Kurzbeschreibung	An dieser Stelle steht eine Kurzbeschreibung des Aktors
Assoziierte Anwendungsfälle	Hier stehen die assoziierten Anwendungsfälle, in die der beschriebene Aktor involviert ist.

Tabelle 3.1: Zusammenfassung des Aktors *beispielhafter Aktor*

Der *Anwendungsfall* oder *Use Case* hingegen beschreibt auf der einen Seite funktionale Anforderungen an ein System, auf der anderen Seite beschreibt er die Interaktion zwischen Aktor(en) und dem System bei der Bearbeitung einer bestimmten Aufgabe. Nach [Hennicker, 2006] besteht eine Beschreibung eines Anwendungsfalls aus folgenden Elementen:

- Name des Anwendungsfalls
- Kurzbeschreibung
- Vorbedingung
  Die Vorbedingung ist die Voraussetzung für eine erfolgreiche Ausführung des zu beschreibenden Anwendungsfalls.
- Nachbedingung
  Die Nachbedingung ist der Zustand nach erfolgreicher Durchführung des Anwendungsfalls.
- Primärszenario
  Das Primärszenario oder auch Standardablauf bezeichnet die Schritte und Interaktionen, die im fehlerfreien Fall bei der Durchführung des Anwendungsfalls durchlebt werden.
- Sekundärszenarien
  Bei einem Anwendungsfall können mehrere Sekundärszenarien oder auch Alternativabläufe zum Tragen kommen. Dabei handelt es sich um Fehlerfälle während der Ausführung eines Anwendungsfalls und eventuell vorhandenen Optionen in einem solchen Fall.

In den nachfolgenden Unterkapiteln werden nach diesem Schema einzelne Anwendungsfälle aus dem D-Grid Projekt detailliert dargestellt und die funktionalen und nicht-funktionalen Anforderungen an ein Frühwarnsystem in Grid-Umgebungen davon abgeleitet. Zusätzlich zur ausführlichen Darstellung eines jeden Anwendungsfalls wird dieser in einer Tabelle Form wie z.B. Tabelle 3.2 zusammengefasst.

	Anwendungsfall *beispielhafter Anwendungsfall*
Bezeichner	UseCase:ID
Kurzbeschreibung	An dieser Stelle steht eine Kurzbeschreibung des Anwendungsfalls
Vorbedingung	Die Vorbedingung beschreibt die erforderlichen Voraussetzungen für eine erfolgreiche Ausführung des Anwendungsfalls.
Nachbedingung	Die Nachbedingung ist der Zustand nach erfolgreicher Durchführung des Anwendungsfalls.

Tabelle 3.2**:** Zusammenfassung des Anwendungsfalls *beispielhafter Anwendungsfall*

3.2.1 Allgemeine Beschreibung des „D-Grid“ Projekts

Der Grundstein für das D-Grid Projekt (`http://www.d-grid.de/`) wurde durch die Gründung der D-Grid Initiative (DGI) im Jahr 2003 gelegt. Eine Gruppe deutscher Wissenschaftler veröffentlichte ein richtungweisendes Strategiepapier [Hegering u. a., 2003], in welchem die aktuelle Situation des deutschen Forschungsumfelds analysiert und der zu erwartende Einfluss des Grid-Computings auf die Forschung untersucht wird. Aufgrund der elementaren Bedeutung wurde als Ergebnis der vorausgegangenen umfangreichen Arbeit ein langfristig ausgerichtetes, strategisches Forschungs- und Entwicklungsprogramm empfohlen – das D-Grid.

Diesem Vorschlag folgend begründete das Bundesministerium für Bildung und Forschung (BMBF, `http://www.bmbf.de/`) wenig später die deutsche D-Grid Initiative. Zur Förderung von Projekten aus den Bereichen e-Learning, Wissensmanagement und Grid-Computing in Form des anzustrebenden D-Grids sollten in einem Zeitraum von fünf Jahren bis zu 100 Millionen Euro bereitgestellt werden. Unter Zuhilfenahme der aufzubauenden Grid-Infrastruktur wird angestrebt, e-Science-Methoden in der deutschen Wissenschaft zu etablieren, wobei damit ein neuer Ansatz des netzbasierten wissenschaftlichen Arbeitens verfolgt wird. Auf der Basis neuester Netztechnologien und in konsequenter Nutzung der Informations- und Wissenstechnologien werden Forschungsprozesse erleichtert, verbessert und intensiviert, indem integrierte Forschungsnetze mit hochleistungsfähigen, verteilten Rechnerressourcen und darauf aufbauenden Diensten innerhalb eines Grids bereitgestellt werden.

Im September 2005 sind neben dem Kern-D-Grid bzw. dem D-Grid Integrationsprojekt sechs weitere, sogenannte Community-Projekte am Aufbau einer neuartigen, auf Nachhaltigkeit ausgelegten Grid-Infrastruktur beteiligt worden. Das Integrationsprojekt nimmt dabei die zentrale Aufgabe wahr, die vielfältigen Entwicklungen der untereinander sehr verschiedenen Community-Projekte in eine gemeinsame Plattform zu integrieren und der deutschen Wissenschaftsgemeinde als Dienste im Kern-D-Grid zugänglich zu machen. Zu den seit Anbeginn beteiligten Communities zählen dabei:

- AstroGrid-D – German Astronomy Community Grid (GACG)
- C3-Grid – Collaborative Climate Community Data and Processing Grid
- HEP-Grid – Grid-Infrastruktur für die Hochenergiephysik
- InGrid – Innovative Grid-Entwicklungen für ingenieurwissenschaftliche Anwendungen
- MediGRID – Grid-Computing für die Medizin und die Lebenswissenschaften
- TextGrid – Community-Grid für die Geisteswissenschaften

Mittlerweile ist eine Vielzahl an Communities in das D-Grid eingebunden, deren genaue Auflistung unter `http://www.d-grid.de/` verfügbar ist. In einer zweiten Entwicklungsphase (D-Grid 2, Jahre 2007-2010) kommen erweiterte IT-Dienste für Wissenschaft und Industrie hinzu, die auf der sogenannten Integrationsschicht des D-Grid aufbauen. Für die Zukunft sind noch weitere Schritte zum Ausbau der D-Grid-Infrastruktur geplant.

Eine technische Besonderheit des D-Grids ist, dass insgesamt drei verschiedene Grid-Middleware Lösungen parallel zum produktiven Einsatz kommen. So sind alle Rechnersysteme des DGI in der aktuellen Version des Globus Toolkit, LCG/gLite und UNICORE ansprechbar, es gibt dabei keine statische Systempartitionierung. Grundsätzlich stehen alle Systeme, die im D-Grid zum Einsatz kommen, jedem Teilnehmer am D-Grid zur Verfügung. Eine Ausnahme bilden dabei Sicherheitsprobleme oder Missbrauch im Einzelfall. An die Nutzung der Ressourcen im D-Grid sind keine Vorbedingungen geknüpft, die über die allgemeinen Vorbedingungen (Beantragung und Besitz gültiger Zertifikate) zur Teilnahme am Grid hinausgehen. Dies impliziert, dass sogar Nutzer außerhalb von D-Grid diese Ressourcen verwenden dürfen, jedoch haben sie eine geringere Priorität als jeder D-Grid Teilnehmer [D-Grid Referenzinstallation, 2007].

3.2.2 Nutzergruppen und Kunden eines GIDS

Betrachtet man im Kontext Grid-basierter Frühwarnsysteme die potentielle Nutzergruppe bzw. den Kundenkreis, so ergeben sich insbesondere einzelne Grid-Nutzer, VOs, Ressourcenanbieter sowie das Grid Operations Center (GOC) als Kandidatenmenge. Diese Menge ist abgeleitet aus den nachstehenden Anwendungsfällen, nachfolgend findet sich eine kurze Darstellung der beteiligten Aktoren.

VO als Kunde. Neben den einzelnen Grid-Nutzern zählen unzweifelhaft ebenfalls Virtuelle Organisationen bzw. deren jeweilige Mitglieder oder Sub-VOs in verschiedenen Rollenausprägungen zum Kundenkreis eines Grid-basierten Frühwarnsystems. Sie erwarten insbesondere eine angepasste Berichterstattung zu von der VO genutzten Ressourcen und Diensten. Des weiteren interagieren sie aber unter Umständen auch mit dem System, um möglicherweise auch Verfügbarkeiten und SLAs mit Hilfe von historischen Daten zu überprüfen. Tabelle 3.3 fasst den Aktor „VO“ als Kunde eines Grid-basierten IDS nochmals zusammen.

Ressourcenanbieter als Kunde. Ressourcenanbieter sind direkt an einer Grid-basierten Einbruchserkennung beteiligt. Sie stellen Kontingente ihrer lokalen Res-

Aktor *VO als Kunde*	
Bezeichner	Aktor:VO:Customer
Kurzbeschreibung	Eine Virtuelle Organisation, die Informationen zur Sicherheit der von ihr genutzten Ressourcen und Dienste wünscht
Assoziierte Anwendungsfälle	*UseCase:VO:Customer* (siehe Tabelle 3.9 auf Seite 50) *UseCase:Forensik* (siehe Tabelle 3.12 auf Seite 54) *UseCase:Privacy* (siehe Tabelle 3.13 auf Seite 55)

Tabelle 3.3**:** Zusammenfassung des Aktors *VO als Kunde*

sourcen und Dienste für eine Nutzung durch das Grid zur Verfügung, die somit auch bedroht sind. Ressourcenanbieter betreiben zumeist bereits lokale IDS Instanzen, können aber maßgeblich auch von der Analyse übergreifender Informationen (wie sie z.B. im Grid gesammelt und verarbeitet werden können) profitieren. Dabei liegt das Hauptaugenmerk auf dem aktuellen Sicherheitsstatus der eigenen Ressourcen. Zusätzlich verspricht sich ein Ressourcenanbieter von der Teilnahme an einem kooperativen Frühwarnsystem eben eine **Früh**warnung. Informationen zu gerade erfolgten Angriffen (ob erfolgreich oder nicht) können einen enormen Mehrwert für einen Betreiber bieten, als dass er daraus ggf. präventive Maßnahmen zu seinem eigenen Schutz ableiten und umsetzen kann. Tabelle 3.4 fasst einen Ressourcenanbieter als Kunde nochmals zusammen.

Aktor *Ressourcenanbieter als Kunde*	
Bezeichner	Aktor:ResProv:Customer
Kurzbeschreibung	Ein Ressourcenanbieter, der Informationen zur Sicherheit der von ihm für das Grid bereitgestellten Ressourcen und Dienste wünscht
Assoziierte Anwendungsfälle	*UseCase:Integration* (siehe Tabelle 3.8 auf Seite 49) *UseCase:ResProv:Customer* (siehe Tabelle 3.10 auf Seite 51) *UseCase:Forensik* (siehe Tabelle 3.12 auf Seite 54)

Tabelle 3.4**:** Zusammenfassung des Aktors *Ressourcenanbieter als Kunde*

Grid Operations Center. In größeren Grid-Umgebungen wie z.B. dem D-Grid wird eine zentrale Anlaufstelle, nicht zuletzt auch zur Information zur Sicherheit, immer häufiger gefordert und auch ins Leben gerufen [Gürich, 2007] – ein Grid Operations Center (GOC). Prinzipiell gehört zum Tätigkeitsbereich eines GOC u.a. auch die fortwährende Überwachung aller wichtigen Ressourcen und Dienste des Grids, speziell

also auch mit Hilfe eines Grid-basierten Intrusion Detection Systems sowie auch die Überwachung des IDS selbst. Kommt es zu Ausfällen oder Fehlern (sowohl im Grid wie auch in Bezug auf ein Grid-basiertes Frühwarnsystem), koordiniert und kontrolliert das GOC die zeitnahe Wiederherstellung betroffener Komponenten bzw. leistet Hilfestellung dazu. Zusätzlich kann eine zeitlich durchgehende Unterstützung von Nutzern, Virtuellen Organisationen und Ressourcenanbietern bei anstehenden Problemen und Fragen gefordert sein. Tabelle 3.5 fasst die Rolle eines GOC als Nutzer eines Grid-basierten IDS nochmals kurz zusammen.

Aktor *Grid Operations Center*	
Bezeichner	Aktor:GOC
Kurzbeschreibung	Das Grid Operations Center (GOC), das Informationen zum Status der Sicherheit im gesamten Grid benötigt
Assoziierte Anwendungsfälle	*UseCase:Integration* (siehe Tabelle 3.8 auf Seite 49) *UseCase:GOC* (siehe Tabelle 3.11 auf Seite 53)

Tabelle 3.5**:** Zusammenfassung des Aktors *Grid Operations Center*

Betreiber des GIDS. Auch für ein GIDS bedarf es eines Betreibers. Natürlich ist nicht auszuschließen, dass mehrere Instanzen eines GIDS an organisatorisch und lokal unterschiedlichen Stellen des Grids betrieben werden, jedoch wird mindestens eine Instanz benötigt, die als informierende Einheit Berichte zur Sicherheit für den gesamten Kundenkreis zur Verfügung stellt. Organisatorisch bietet sich als Betreiber zum Beispiel das Grid Operations Center an, dessen Aufgaben u.a. eine Unterstützung von Nutzern, VOs und Ressourcenanbietern bei Problemen und Fragen sind, wozu Informationen des GIDS mitunter von Nöten bzw. hilfreich sein können.

Management-Plattform. In großen Systemen erwachsen in vielen Fällen eine Vielzahl an Management-Plattformen und Insellösungen für den Betrieb des Gesamtsystems. Bei der Konzeption einer weiteren Komponente stellt sich somit die Herausforderung, das neu zu entwickelnde System möglichst gut in die bereits vorhandenen Plattformen zu integrieren, falls dies möglich ist. Da im Grid-Umfeld diverse unterschiedliche Management-Lösungen bei den einzelnen Grid-Teilnehmern zu finden sind, ergeben sich durch diese bedingt eine Menge Anforderungen, denen ein GIDS bereits in der Konzeptionsphase begegnen muss.

3.2.2.1 Anwendungsfall *Integration eines GIDS*

Ressourcenanbieter A sowie die Sicherheitsbeauftragten des Grid Operations Center möchten das GIDS in ihre (Site-lokal) bereits bestehenden Systeme zur Überwachung des

Aktor *Betreiber des GIDS*	
Bezeichner	Aktor:GIDS:Provider
Kurzbeschreibung	Der Betreiber des GIDS und somit die zentrale Anlaufstelle für alle Teilnehmer des Grids, die eine Berichterstattung zur Sicherheit wünschen.
Assoziierte Anwendungsfälle	*UseCase:VO:Customer* (siehe Tabelle 3.9 auf Seite 50) *UseCase:ResProv:Customer* (siehe Tabelle 3.10 auf Seite 51) *UseCase:GOC* (siehe Tabelle 3.11 auf Seite 53) *UseCase:Forensik* (siehe Tabelle 3.12 auf Seite 54) *UseCase:3rdParties* (siehe Tabelle 3.18 auf Seite 59)

Tabelle 3.6**:** Zusammenfassung des Aktors *Betreiber des GIDS*

Aktor *Management-Plattform*	
Bezeichner	Aktor:MonSys
Kurzbeschreibung	Der Aktor „Management-Plattform“ ist im Grid-Kontext meist in mehrfacher und unterschiedlicher Instanziierung bei organisatorisch verschiedenen Parteien vorzufinden. Eine maximale Integration eines GIDS in möglichst viele Management-Plattformen stellt eine große Herausforderung dar.
Assoziierte Anwendungsfälle	*UseCase:Integration* (siehe Tabelle 3.8 auf Seite 49)

Tabelle 3.7**:** Zusammenfassung des Aktors *Management-Plattform*

Grids (z.B. Monitoring-Systeme) integrieren. Beide verwenden jedoch dazu unterschiedliche Plattformen und möchten diese nicht aufgeben, um u.a. die gewohnte Umgebung für ihr Personal beibehalten zu können und keine weiteren Anwendungen zu etablieren.

Primärszenario. Die von Ressourcenanbieter A sowie dem GOC eingesetzte Management-Plattform authentifiziert sich gegenüber dem GIDS und erhält von diesem für den jeweiligen Nutzer relevante Sicherheitsberichte. Gegebenenfalls ist neben der reinen Darstellung von Informationen auch eine gezielte Anfrage an das GIDS möglich, so dass vom Nutzer gewünschte Daten bereitgestellt werden können. Sämtliche Daten werden unter Gewährleistung der Datenintegrität und Vertraulichkeit ausgeliefert.

Anwendungsfall *Integration eines GIDS*	
Bezeichner	UseCase:Integration
Kurzbeschreibung	Kunden des GIDS möchten die bereitgestellten Sicherheitsberichte in ihre bestehenden Management-Plattformen integrieren und darüber ggf. individuelle Anfragen an das GIDS richten können
Vorbedingung	Der Kunde kann sich dem GIDS gegenüber authentifizieren und ist zur Informationsabfrage autorisiert
Nachbedingung	Berichte des GIDS werden innerhalb der Management-Anwendungen des Kunden dargestellt

Tabelle 3.8: Zusammenfassung des Anwendungsfalls *Integration eines GIDS*

Abgeleitete Anforderungen.

- Integrierbarkeit in bestehende Management-Werkzeuge
- Interoperabilität
- Unterstützung etablierter Standards
- Einheitliche Schnittstellen
- Gesicherter Informationsaustausch (Integrität, Authentizität und Vertraulichkeit von Nachrichten)
- Nutzung standardisierter und einheitlicher Daten(austausch)formate
- Mandantenfähigkeit, nutzergruppenabhängige Berichterstattungen und Sichten
- Einbindung bestehender AA-Mechanismen

Beteiligte Aktoren.

- Management-Plattform
- Grid Operations Center
- Ressourcenanbieter als Kunde

Tabelle 3.8 fasst den Anwendungsfall nochmals kurz zusammen.

3.2.2.2 Anwendungsfall *Zugriff einer VO als Nutzer eines GIDS*

Die Teilnehmer des Community-Projekts A benötigen stets Informationen zur Sicherheit der von ihnen genutzten Ressourcen und Dienste im Grid, da sie vertrauliche Daten im Grid verarbeiten. Im Falle von Unregelmäßigkeiten werden genauere Informationen, die den entsprechenden Sicherheitsbericht ausmachen, benötigt. In einem solchen Fall ist eine proaktive Benachrichtigung eines zuvor festgelegten Ansprechpartners des Community-Projekts wünschenswert.

Anwendungsfall *Zugriff einer VO als Nutzer eines GIDS*	
Bezeichner	UseCase:VO:Customer
Kurzbeschreibung	Eine VO wünscht einen Bericht zur Sicherheit der von ihr genutzten Ressourcen und Dienste im Grid
Vorbedingung	Ein authentifizierbares und autorisiertes Mitglied der VO möchte auf die Benutzerschnittstelle des GIDS zugreifen.
Nachbedingung	Ein Bericht zur Sicherheit liegt der VO vor.

Tabelle 3.9**:** Zusammenfassung des Anwendungsfalls *Zugriff einer VO als Nutzer eines GIDS*

Primärszenario. Ein Mitglied einer (Sub-)VO des Community-Projekts A authentifiziert sich gegenüber einer Benutzeroberfläche, die die aufbereiteten Berichte des GIDS darstellt. Auch der Zugriff auf historische Berichte sollte dabei möglich sein.

Sekundärszenario. Sollte ein Bericht zum Status der Sicherheit Auffälligkeiten beinhalten, so ist ein proaktiver Hinweis durch das Frühwarnsystem sinnvoll. Daraufhin sollte ein Mitglied einer (Sub-)VO des Community-Projekts A die Möglichkeit haben weitere Informationen bezüglich der erkannten Unregelmäßigkeit und der sie ausmachenden Ursachen zu erhalten. Dazu ist unter Umständen ebenfalls ein Abgleich mit historischen Berichten und eine Recherche detaillierterer Daten notwendig, nicht zuletzt um eine Verletzung potentiell bestehende SLAs mit Ressourcenanbietern auch von Nutzerseite her überprüfen zu können und somit Transparenz zu schaffen.

Abgeleitete Anforderungen.

- Verwendung einer einheitlichen und übersichtlichen Benutzerschnittstelle (z.B. in Form eines Web-basierten Portals)
- Unterstützung Virtueller Organisationen und daraus folgend Einbindung in VO-Managementsysteme
- Mandantenfähigkeit, nutzergruppenabhängige Berichterstattungen und Sichten
- Einbindung bestehender AA-Mechanismen
- Nachvollziehbarkeit durchgeführter Anfragen
- Aussagekräftige Informationsaufbereitung
- Proaktive Benachrichtigung zuvor festgelegter Ansprechpartner über erkannte Unregelmäßigkeiten
- Nachhalten historischer Berichte
- Verschiedene Granularitätsstufen bei der Berichterstattung
- Gesicherter Informationsaustausch (Integrität, Authentizität und Vertraulichkeit von Nachrichten)
- Push- und Pull-Mechanismen für den Datenzugriff

Beteiligte Aktoren.

- Virtuelle Organisation
- Grid-globaler Betreiber des GIDS

Tabelle 3.9 fasst den Anwendungsfall nochmals kurz zusammen.

3.2.2.3 Anwendungsfall *Ressourcenanbieter als Anwender*

Ressourcenanbieter A stellt eine Reihe Ressourcen und Dienste zur Nutzung im Grid bereit und partizipiert an einem Grid-weiten Frühwarnsystem. Die Sicherheitsbeauftragten des Ressourcenanbieters wünschen eine proaktive Alarmierung über im Grid erkannte Unregelmäßigkeiten sowohl in Bezug auf die von Ihnen angebotenen Ressourcen und Dienste wie auch die von anderen Ressourcenanbietern bereitgestellten, um so ggf. präventiv auf anstehende Bedrohungen reagieren zu können. Dazu werden möglichst detaillierte und präzise Benachrichtigungen sowie die Möglichkeit aktiv in beim GIDS anfallenden Informationen durchsuchen zu können benötigt.

Anwendungsfall *Ressourcenanbieter als Anwender*	
Bezeichner	UseCase:ResProv:Customer
Kurzbeschreibung	Ein Ressourcenanbieter möchte den Sicherheitsstatus der von ihm bereitgestellten Ressourcen und Dienste im Grid erfragen, ggf. proaktiv über Vorkommnisse alarmiert werden und aktiv die anfallenden IDS Daten durchsuchen können.
Vorbedingung	Ein authentifizierbarer und autorisierter Administrator eines Ressourcenanbieters möchte auf die Benutzerschnittstelle des GIDS zugreifen.
Nachbedingung	Ein Bericht zur Sicherheit liegt dem Ressourcenanbieter vor, möglicher Weise notwendige Nachforschungen können aktiv vorgenommen werden.

Tabelle 3.10**:** Zusammenfassung des Anwendungsfalls *Ressourcenanbieter als Anwender*

Primärszenario. Ein Administrator oder Sicherheitsbeauftragter des Ressourcenanbieters A wird über durch das GIDS erkannte Unregelmäßigkeiten proaktiv benachrichtigt. Es besteht die Möglichkeit, dass sich ein Vertreter des Ressourcenanbieters A jederzeit gegenüber einer Benutzeroberfläche des GIDS authentifiziert und die aufbereiteten Berichte für das ganze Grid einsieht. Auch der Zugriff auf historische Berichte ist möglich.

Sekundärszenario. Sollten sicherheitsrelevante Vorkommnisse aufgetreten sein oder vermutet werden, so können die beim GIDS vorliegenden Sensordaten (aktuell und historisch in verschiedenen Detailgraden) aktiv untersucht werden.

Abgeleitete Anforderungen.

- Verwendung einer einheitlichen und übersichtlichen Benutzerschnittstelle (z.B. in Form eines Web-basierten Portals)
- Mandantenfähigkeit, nutzergruppenabhängige Berichterstattungen und Sichten
- Einbindung bestehender AA-Mechanismen
- Aussagekräftige Informationsaufbereitung
- Nachhalten historischer Berichte
- Verschiedene Granularitätsstufen bei der Berichterstattung
- Proaktive Benachrichtigung zuvor festgelegter Ansprechpartner über erkannte Unregelmäßigkeiten
- Zugriffsmöglichkeit auf Sensordaten
- Archivierung von Sensordaten, u.U. in verschiedenen Detailgraden
- Gesicherter Informationsaustausch (Integrität, Authentizität und Vertraulichkeit von Nachrichten)
- Push- und Pull-Mechanismen für den Datenzugriff

Beteiligte Aktoren.

- Ressourcenanbieter als Kunde
- Grid-globaler Betreiber des GIDS

Tabelle 3.10 fasst den Anwendungsfall nochmals kurz zusammen.

3.2.2.4 Anwendungsfall *Grid Operations Center*

Das Grid Operations Center dient als zentraler Ansprechpartner bei Fragen zum Grid. Zudem koordiniert das GOC die Wiederherstellung von Diensten im Grid und leistet hierzu ggf. Hilfestellung. Dazu benötigen die Mitglieder des GOC für den Bereich Sicherheit jeweils aktuelle Statusberichte, die in ihrem Detailgrad je nach Anfrage variieren und angepasst werden können müssen.

Primärszenario. Ein authentifizierbares und autorisiertes Mitglied des GOC greift auf die Benutzeroberfläche des GIDS zu und kann die aufbereiteten Berichte für das ganze Grid einsehen.

Sekundärszenario. Sollten Unregelmäßigkeiten berichtet werden, so können detailliertere Berichte angefordert und eingesehen werden. Außerdem können spezifische Anfragen formuliert und durch das GIDS bearbeitet werden.

Abgeleitete Anforderungen.

- Verwendung einer einheitlichen und übersichtlichen Benutzerschnittstelle (z.B. in Form eines Web-basierten Portals)
- Mandantenfähigkeit, nutzergruppenabhängige Berichterstattungen und Sichten

Anwendungsfall *Grid Operations Center*	
Bezeichner	UseCase:GOC
Kurzbeschreibung	Das GOC möchte sich ein Bild zur Sicherheitslage des Grids verschaffen. Als zentraler Ansprechpartner werden dazu hinreichen detaillierte Informationen benötigt, jedoch dennoch auf hoher Abstraktionsebene.
Vorbedingung	Ein authentifizierbares und autorisiertes Mitglied des GOC möchte auf die Benutzerschnittstelle des GIDS zugreifen.
Nachbedingung	Ein Bericht zur Sicherheitslage des gesamten Grids liegt vor.

Tabelle 3.11**:** Zusammenfassung des Anwendungsfalls *Grid Operations Center*

- Einbindung bestehender AA-Mechanismen
- Aussagekräftige Informationsaufbereitung
- Verschiedene Granularitätsstufen bei der Berichterstattung
- Gesicherter Informationsaustausch (Integrität, Authentizität und Vertraulichkeit von Nachrichten)

Beteiligte Aktoren.

- Grid-globaler Betreiber des GIDS
- Grid Operations Center

Tabelle 3.11 fasst den Anwendungsfall nochmals kurz zusammen.

3.2.2.5 Anwendungsfall *Beweissicherung & Forensik*

Ressourcenanbieter A und Projektgruppe B, die im Grid als VO modelliert ist, streiten um die Verletzung eines abgeschlossenen SLAs in Bezug auf einzuhaltende Sicherheitsmerkmale. B behauptet, dass A den geschlossenen SLA gebrochen hat und verlangt die im SLA verbrieften monetären Pönalen[4].

Primärszenario. Die vom GIDS erzeugten Sicherheitsberichte, die die vermeintliche Verletzung des SLAs bezeugen, können sowohl von Ressourcenanbieter A wie auch VO B eingesehen werden. Die Berichte können unter Umständen auch vor längerer Zeit in der Vergangenheit erzeugt worden sein.

Sekundärszenario. Zur genaueren Klärung des Vorfalls können die Sensordaten, aus denen die generierten Berichte entstanden sind, eingesehen werden. Je nach Alter der Daten variiert dabei der Detailgrad der vorhandenen Informationen.

[4]SLAs im eigentlichen Sinne werden zurzeit im D-Grid nicht geschlossen. Es ist jedoch auch eine entgeltliche Nutzung der Grid-Infrastruktur angedacht, so dass dieser Anwendungsfall dann zum Tragen kommen könnte.

Anwendungsfall *Beweissicherung & Forensik*	
Bezeichner	UseCase:Forensik
Kurzbeschreibung	Ein Ressourcenanbieter und eine Projektgruppe streiten um die Verletzung eines abgeschlossenen SLAs in Bezug auf einzuhaltende Sicherheitsniveaus.
Vorbedingung	Ein SLA mit sicherheitsrelevanten und überprüfbaren Merkmalen wurde abgeschlossen.
Nachbedingung	Die Überprüfung des SLAs konnte durch die vom GIDS gelieferten Informationen erfolgen.

Tabelle 3.12**:** Zusammenfassung des Anwendungsfalls *Beweissicherung & Forensik*

Abgeleitete Anforderungen.

- Nachvollziehbarkeit durchgeführter Anfragen
- Aussagekräftige Informationsaufbereitung
- Nachhalten historischer Berichte
- Zugriffsmöglichkeit auf Sensordaten
- Archivierung von Sensordaten, u.U. in verschiedenen Detailgraden
- Gewährleistung der Integrität des Datenbestands (Berichte und Sensordaten)
- Push- und Pull-Mechanismen für den Datenzugriff

Beteiligte Aktoren.

- Ressourcenanbieter als Kunde
- Grid-globaler Betreiber des GIDS
- Virtuelle Organisation

Tabelle 3.12 fasst den Anwendungsfall nochmals kurz zusammen.

3.2.2.6 Anwendungsfall *Datenschutz & Vertraulichkeit*

VO A (z.B. medizinische Forscher aus dem MediGrid) möchten sensible Patientendaten im Rahmen einer Studie im Grid auswerten und analysieren. Juristische Randbedingungen fordern Garantien zur Vertraulichkeit der Daten und der Einhaltung des Datenschutzes.

Primärszenario. Ein Beauftragter der VO A formuliert (optimaler Weise in einer maschinenlesbaren Form) die für das GIDS relevanten Randbedingungen, die für eine Verarbeitung der sensiblen Patientendaten garantiert werden müssen. Diese werden durch entsprechende Mechanismen des GIDS durchgesetzt, so dass eine Datenverarbeitung im Grid ermöglicht werden kann.

Anwendungsfall *Datenschutz & Vertraulichkeit*	
Bezeichner	UseCase:Privacy
Kurzbeschreibung	Eine medizinische VO möchte sensible Patientendaten im Grid verarbeiten.
Vorbedingung	Juristische Randbedingungen mit Relevanz für das GIDS können formuliert und technisch durchgesetzt werden.
Nachbedingung	Eine Datenverarbeitung im Grid kann für diesen Fall ermöglicht werden.

Tabelle 3.13**:** Zusammenfassung des Anwendungsfalls *Datenschutz & Vertraulichkeit*

Sekundärszenario. Sollte entweder eine technische Gewährleistung oder die Formulierung der Randbedingungen nicht möglich sein, so kann eine Verarbeitung der Daten im Grid nicht vorgenommen werden.

Abgeleitete Anforderungen.

- Möglichkeiten der Zugriffsbeschränkung auf jegliche Informationen im GIDS
- Anonymisierungs- oder Pseudonymisierungsmöglichkeiten inkl. einer notwendigen (maschinenlesbaren) Beschreibungsmöglichkeit
- Mandantenfähigkeit, nutzergruppenabhängige Berichterstattungen und Sichten
- Einbindung bestehender AA-Mechanismen
- Gesicherter Informationsaustausch (Integrität, Authentizität und Vertraulichkeit von Nachrichten)

Beteiligte Aktoren.

- Virtuelle Organisation

Tabelle 3.13 fasst den Anwendungsfall nochmals kurz zusammen.

3.2.3 Informationsanbieter eines GIDS

Auch als Informationsanbieter sind im Rahmen eines Grid-basierten IDS eine Menge Aktoren denkbar. Insbesondere ergeben sich aus den nachgestellten Anwendungsfällen bereits vorhandene Host- und Netzsensorik, der Ressourcenanbieter (diesmal in der Rolle des Informationslieferanten), VOs (hier ebenfalls in der Rolle des Informationslieferanten) sowie verschiedene Drittanbieter, die ansonsten nicht weiter in die Aktivitäten des Grids eingebunden sein müssen. Nachfolgend findet sich ein kurzer Überblick über die einzelnen Aktoren.

Ressourcenanbieter als Informationsanbieter. Der Erfolg und sicherheitstechnische Nutzen eines Grid-basierten Frühwarnsystems, unter der Annahme bereits vorhandene Informationen zur Sicherheit zu föderieren, hängt in hohem Maße davon ab, ob und

in welcher Qualität relevante Informationen von den einzelnen eingebundenen Partnern einbezogen werden können. Insbesondere dienen in diesem Umfeld natürlich die Ressourcenanbieter auch als Informationsanbieter für ein Grid-basiertes IDS. Jedoch erheben sie gleichfalls eine Reihe Randbedingungen, die für ihre Teilnahme erfüllt sein müssen. Tabelle 3.14 fasst nochmals kurz den Ressourcenanbieter in diesem Kontext in Abgrenzung zu Unterabschnitt 3.2.2 als Informationsanbieter zusammen.

In realen Grid-Umgebungen sind Ressourcenanbieter als Informationsanbieter für ein Grid-basiertes Frühwarnsystem am häufigsten zu finden, was insbesondere durch Randbedingungen wie Datenschutzbestimmungen und die Gewährleistung der Autonomie an einem GIDS beteiligter Parteien bedingt ist. In den meisten Fällen sind Informationen von Site-spezifischen Host- und Netzsensoren zu feingranular, was meist zu Problemen in Bezug auf die Skalierbarkeit führt. Informationen, die VOs hingegen beitragen können, gehen ohnehin aus dem Aggregat der Daten aller an einem GIDS partizipierenden Parteien hervor, so dass in vielen Fällen hierdurch kein essentieller Mehrwert dargestellt werden kann.

Aktor *Ressourcenanbieter als Informationsanbieter*	
Bezeichner	Aktor:ResProv:InfoProv
Kurzbeschreibung	Ein Ressourcenanbieter, der seine Informationen über den Status der Sicherheit seiner Ressourcen (z.B. IDS Daten, Firewall Logdateien etc.) einem GIDS zur Verfügung stellt
Assoziierte Anwendungsfälle	*UseCase:Autonomie* (siehe Tabelle 3.16 auf Seite 57) *UseCase:ISP* (siehe Tabelle 3.17 auf Seite 58)

Tabelle 3.14**:** Zusammenfassung des Aktors *Ressourcenanbieter als Informationsanbieter*

Betreiber des GIDS. *siehe Abschnitt 3.2.2 ab Seite 45*

3^{rd} **Parties.** Auch Informationen Dritter können einen entscheidenden Einfluss auf ein Grid-basiertes IDS haben. Denkbar sind diese vor allem als externe Dienstleister. Zum Beispiel können Berichte von Computer Emergency Response Teams (CERT) zu bestehenden Sicherheitslücken und diese ausnutzende Schadprogramme sehr hilfreich bei der Einschätzung wie schwerwiegend eine Sicherheitsverletzung ist oder der Erkennung neuartiger Angriffe sein. Ein weiteres Beispiel ist der Einfluss eines Dienstleisters, der Sicherheitsprodukte, die im Grid zum Einsatz kommen, mit Aktualisierungen versorgt. Diese können sich sowohl auf die Aktualisierung der Software-Komponenten an sich (engl. *Patches*) oder auch der Aktualisierung von z.B. Signaturdatenbanken eines Virenscanners oder auch IDS beziehen. Tabelle 3.15 fasst den Aktor „dritte Parteien“ nochmals tabellarisch zusammen.

Aktor 3^{rd} *Parties*	
Bezeichner	Aktor:3rdParties
Kurzbeschreibung	Nicht am Grid beteiligte Aktoren (3^{rd} *Parties*), die relevante Informationen zur Sicherheit des Grids liefern; z.B. CERTs, die Implementierungsschwächen und diese ausnutzende Schadprogramme melden
Assoziierte Anwendungsfälle	*UseCase:3rdParties* (siehe Tabelle 3.18 auf Seite 59)

Tabelle 3.15**:** Zusammenfassung des Aktors 3^{rd} *Parties*

3.2.3.1 Anwendungsfall *Autonomie beteiligter Organisationen*

Ressourcenanbieter A möchte gerne verschiedene Ressourcen und Dienste im Grid zur Verfügung stellen. Dazu wird jedoch gefordert, dass A auch als Informationsanbieter für Daten zur Analyse im Rahmen des im Grid etablierten IDS auftritt. A setzt bereits verschiedene Sicherheitsmechanismen, u.a. auch lokale IDS Instanzen, ein. Weitere (redundante) Installationen sind nicht durchsetzbar, vielmehr sollen bestehende Komponenten und Dienste verwendet werden.

Anwendungsfall *Autonomie beteiligter Organisationen*	
Bezeichner	UseCase:Autonomie
Kurzbeschreibung	Ein Ressourcenanbieter möchte gerne an der Datensammlung für ein GIDS teilnehmen, dafür sollen jedoch bestehende Systeme genutzt und keine neuen installiert werden.
Vorbedingung	Informationen von Sicherheitsmechanismen eines Ressourcenanbieters sollen in das GIDS integriert werden.
Nachbedingung	Der Ressourcenanbieter nimmt als Informationsanbieter am GIDS teil.

Tabelle 3.16**:** Zusammenfassung des Anwendungsfalls *Autonomie beteiligter Organisationen*

Primärszenario. Es werden Komponenten vom Grid angeboten, die die bei Ressourcenanbieter A existierenden Informationsquellen auslesen, die Daten für das GIDS semantisch und syntaktisch aufarbeiten und in die Analyse des GIDS einbringen. Diese werden Site-lokal bei A installiert und an das Grid-weite Frühwarnsystem angebunden.

Sekundärszenario. Sollten keine geeigneten Adapter für die Systeme von A vorliegen,

so besteht die Möglichkeit solche zu implementieren, da das GIDS entsprechende Erweiterungsmechanismen vorsieht.

Abgeleitete Anforderungen.

- Nutzung standardisierter und einheitlicher Daten(austausch)formate
- Unterstützung heterogener Informationsquellen
- Einheitliche Schnittstellen
- Portabilität
- Wiederverwendbarkeit bestehender Komponenten
- Interoperabilität
- Erweiterbarkeit des GIDS um bisher ungenutzte Informationsquellen
- Einbringen zusätzlicher Informationsquellen in das GIDS während des Betriebs

Beteiligte Aktoren.

- Ressourcenanbieter als Informationsanbieter

Tabelle 3.16 fasst den Anwendungsfall nochmals kurz zusammen.

3.2.3.2 Anwendungsfall *Information Sharing Policies*

Ein Rechenzentrum tritt in der Rolle des Ressourcenanbieters A als Informationsanbieter für ein Grid-weites Frühwarnsystem auf. Rechtliche Randbedingungen zwingen A, z.B. alle personenbezogenen Informationen nicht an Dritte und somit insbesondere nicht an das GIDS weiterzugeben. Außerdem äußert der Sicherheitsbeauftragte erhebliche Bedenken sämtliche Informationen ungefiltert dem Grid zur Verfügung zu stellen.

Anwendungsfall *Information Sharing Policies*	
Bezeichner	UseCase:ISP
Kurzbeschreibung	Durch gewisse organisatorische und rechtliche Randbedingungen kann ein Ressourcenanbieter nicht alle ihm vorliegenden Informationen zur Auswertung an ein GIDS ungefiltert weitergeben.
Vorbedingung	Ein Ressourcenanbieter möchte zum GIDS beitragen, unterliegt jedoch Auflagen bezüglich der Informationsweitergabe.
Nachbedingung	Der Ressourcenanbieter kann am GIDS partizipieren, da seine Randbedingungen technisch durchgesetzt werden können.

Tabelle 3.17: Zusammenfassung des Anwendungsfalls *Information Sharing Policies*

Primärszenario. Ressourcenanbieter A kann ein unter seiner eigenen Administration stehendes System dazu nutzen alle Daten, die er zur Auswertung an das GIDS weiterreichen könnte, zu filtern. Dabei ist sowohl das Aussortieren bzw. Löschen als auch das Anonymisieren bzw. Pseudonymisieren von Informationen möglich. Eine entsprechende Sprache zur Formulierung der Filterbedingungen wird dazu angeboten.

Sekundärszenario. Sollte Ressourcenanbieter A seine Vorgaben nicht durchsetzen können, da z.B. die technischen Möglichkeiten nicht existieren, so wird er nicht am GIDS teilnehmen können.

Abgeleitete Anforderungen.

- Anonymisierungs- oder Pseudonymisierungsmöglichkeiten inkl. einer notwendigen (maschinenlesbaren) Beschreibungsmöglichkeit
- Gewährleistung der Autonomie beteiligter Informationsanbieter
- (Technische) Durchsetzung des Datenschutzes

Beteiligte Aktoren.

- Ressourcenanbieter als Informationsanbieter

Tabelle 3.17 fasst den Anwendungsfall nochmals kurz zusammen.

3.2.3.3 Anwendungsfall 3^{rd} *Parties als Informationsanbieter*

Zusätzlich zu den Informationen der IDS-Sensorik im Grid sollen Daten Dritter mit in die Analyse des GIDS einfließen. Insbesondere sollen Informationen eines *Computer Emergency Response Teams* (CERT) zu bestehenden Sicherheitslücken in verwendeten Softwarekomponenten im Kontext des GIDS publik gemacht werden und Signaturen zur Erkennung von Angriffsmustern, die diese Lücken missbräuchlich ausnutzen, von einem externen Dienstleister in Form von Updates einer etablierten Signaturdatenbank übernommen werden.

Anwendungsfall 3^{rd} *Parties als Informationsanbieter*	
Bezeichner	UseCase:3rdParties
Kurzbeschreibung	Als zusätzliche Informationsquellen sollen Daten von Dritten (z.B. CERTs) in die Analyse des GIDS einfließen.
Vorbedingung	Vorbedingung
Nachbedingung	Nachbedingung

Tabelle 3.18: Zusammenfassung des Anwendungsfalls 3^{rd} *Parties als Informationsanbieter*

Primärszenario. Es steht eine Anlaufstelle im Grid bereit, die Informationen zu Sicherheitslücken im Grid und Signaturen zur Erkennung derer Ausnutzung für die eingesetzten Sicherheitssysteme bereithält. Die Teilnehmer des GIDS gleichen ihre Signaturdatenbanken regelmäßig mit den angebotenen Signaturen ab.

Abgeleitete Anforderungen.

- Nutzung standardisierter und einheitlicher Daten(austausch)formate
- Unterstützung heterogener Informationsquellen
- Einheitliche Schnittstellen
- Wiederverwendbarkeit bestehender Komponenten
- Interoperabilität
- Erweiterbarkeit des GIDS um bisher ungenutzte Informationsquellen
- Prozessspezifikation für (Signatur-)Updates der Site-spezifischen GIDS-Komponenten
- Gesicherter Informationsaustausch (Integrität, Authentizität und Vertraulichkeit von Nachrichten)
- Etablierung einer vertrauenswürdigen Koordinationseinheit im GIDS

Beteiligte Aktoren.

- Grid-globaler Betreiber des GIDS
- 3^{rd} Parties

Tabelle 3.18 fasst den Anwendungsfall nochmals kurz zusammen.

3.2.4 Zusammenfassung der beteiligten Aktoren und Anforderungen

Die Anwendungsfall-getriebenen Anforderungsanalyse, die in diesem Abschnitt auf den Nutzerkreis und die Informationsanbieter eines GIDS fokussiert, hat insbesondere die nachfolgend aufgelisteten beteiligte Aktoren hervorgebracht:

- VO als Kunde, siehe Tabelle 3.3 auf Seite 46
- Ressourcenanbieter als Kunde, siehe Tabelle 3.4 auf Seite 46
- Grid Operations Center, siehe Tabelle 3.5 auf Seite 47
- Betreiber des GIDS, siehe Tabelle 3.6 auf Seite 48
- Management-Plattform, siehe Tabelle 3.7 auf Seite 48
- Ressourcenanbieter als Informationsanbieter, siehe Tabelle 3.14 auf Seite 56
- 3^{rd} Parties, siehe Tabelle 3.15 auf Seite 57

Die Abbildungen 3.3 und 3.4 bringen nochmal zusammenfassend die jeweiligen Aktoren mit den sie betreffenden Anwendungsfällen in Verbindung und stellen diese in einem

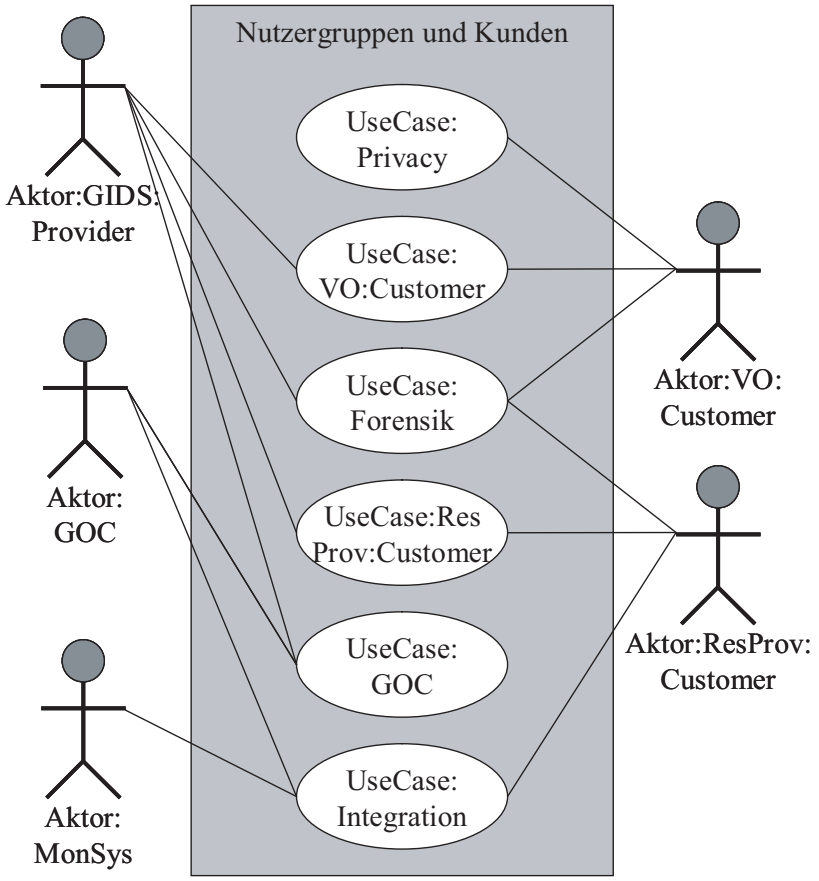


Abbildung 3.3: Übersicht der kundenspezifischen Anwendungsfälle

UML Use-Case Diagramm dar. Abbildung 3.3 bezieht sich dabei auf die kundenspezifischen Anwendungsfälle wie sie in Abschnitt 3.2.2 zu finden sind, Abbildung 3.4 fasst die Informationsanbieter-spezifischen Anwendungsfälle aus Abschnitt 3.2.3 nochmals zusammen.

Aus der Anwendungsfall-getriebenen Anforderungsanalyse ergeben sich eine Vielzahl Anforderungen, die durch den Anwender eines GIDS sowie die Informationsanbieter bedingt sind. Die vorangehend abgeleiteten Anforderungen sind nachfolgend nochmals in einer Liste zusammengefasst aufgeführt. Die Reihenfolge der Anforderungen ist dabei alphabetisch und soll in keiner Weise eine Gewichtung oder Wertung antizipieren!

1. Anonymisierungs- oder Pseudonymisierungsmöglichkeiten inkl. einer notwendigen (maschinenlesbaren) Beschreibungsmöglichkeit
2. Archivierung von Sensordaten, u.U. in verschiedenen Detailgraden
3. Aussagekräftige Informationsaufbereitung
4. Einbindung bestehender AA-Mechanismen
5. Einbringen zusätzlicher Informationsquellen in das GIDS während des Betriebs

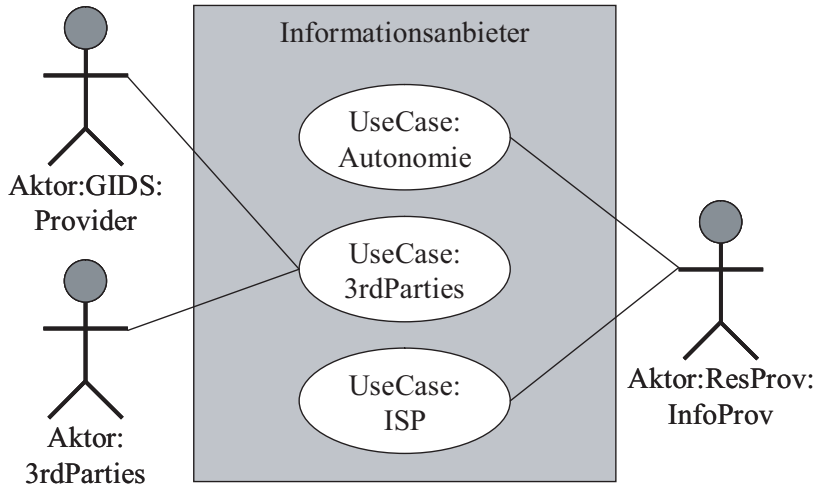


Abbildung 3.4**:** Übersicht der Informationsanbieter-spezifischen Anwendungsfälle

6. Einheitliche Schnittstellen
7. Erweiterbarkeit des GIDS um bisher ungenutzte Informationsquellen
8. Etablierung einer vertrauenswürdigen Koordinationseinheit im GIDS
9. Gesicherter Informationsaustausch (Integrität, Authentizität und Vertraulichkeit von Nachrichten)
10. Gewährleistung der Autonomie beteiligter Informationsanbieter
11. Gewährleistung der Integrität des Datenbestands (Berichte und Sensordaten)
12. Integrierbarkeit in bestehende Management-Werkzeuge
13. Interoperabilität
14. Mandantenfähigkeit, nutzergruppenabhängige Berichterstattungen und Sichten
15. Möglichkeiten der Zugriffsbeschränkung auf jegliche Informationen im GIDS
16. Nachhalten historischer Berichte
17. Nachvollziehbarkeit durchgeführter Anfragen
18. Nutzung standardisierter und einheitlicher Daten(austausch)formate
19. Portabilität
20. Proaktive Benachrichtigung zuvor festgelegter Ansprechpartner über erkannte Unregelmäßigkeiten
21. Prozessspezifikation für (Signatur-)Updates der Site-spezifischen GIDS-Komponenten
22. Push- und Pull-Mechanismen für den Datenzugriff

23. (Technische) Durchsetzung des Datenschutzes
24. Unterstützung etablierter Standards
25. Unterstützung heterogener Informationsquellen
26. Unterstützung Virtueller Organisationen und daraus folgend Einbindung in VO-Managementsysteme
27. Verschiedene Granularitätsstufen bei der Berichterstattung
28. Verwendung einer einheitlichen und übersichtlichen Benutzerschnittstelle (z.B. in Form eines Web-basierten Portals)
29. Wiederverwendbarkeit bestehender Komponenten
30. Zugriffsmöglichkeit auf Sensordaten

Tabelle 3.19 stellt nochmal in einer Übersicht die je Anwendungsfall abgeleiteten Anforderungen dar. Die jeweilige Anforderungsnummer entspricht dabei der Nummerierung vorstehender Auflistung der Anforderungen, die „Nummer“ des jeweiligen Anwendungsfalls referenziert hingegen den Unterabschnitt, in dem der Anwendungsfall detailliert beschrieben ist.

3.3 Generische Anforderungen an ein GIDS

Für das Konzept eines jeden (Grid-)Systems sind eine Menge generischer Anforderungen zu beachten, die im Nachfolgenden kurz erörtert werden. Zudem ist zur Verfolgung des Ziels, ein kooperatives Frühwarnsystem für Grid-Strukturen zu entwerfen, eine Kooperation unterschiedlicher Organisationen miteinander von Nöten, die Vertrauensbeziehungen unterhalb der beteiligten Organisationen impliziert. Diesen beiden Tatbeständen wird in den Abschnitten 3.3.2 und 3.3.3 Rechnung getragen.

3.3.1 Generische Anforderungen

Bereits in Abschnitt 2.2.1 sind bei der allgemeinen Beschreibung der Konzepte und Architektur im Grid nach [Foster, 2002] einige Anforderungen an Systeme im Grid bzw. ihre Eigenschaften aufgekommen.

Dezentrale Organisation. Die im Grid verbundenen Ressourcen können sich über viele juristisch, organisatorisch und administrativ autonome und geografisch verteilte Unternehmen erstrecken. Grids unterliegen generell keiner zentralen Kontrolle.

Anf.-Nr.	Anwendungsfall *(siehe Abschnitt ...)*								
	3.2.2.1	3.2.2.2	3.2.2.3	3.2.2.4	3.2.2.5	3.2.2.6	3.2.3.1	3.2.3.2	3.2.3.3
1						✔		✔	✔
2			✔		✔				
3		✔	✔	✔	✔				
4	✔	✔	✔	✔		✔			
5							✔		
6	✔						✔		✔
7							✔		✔
8									✔
9	✔	✔	✔	✔		✔			✔
10								✔	
11					✔				
12	✔								
13	✔						✔		✔
14	✔	✔	✔	✔		✔			
15						✔			
16		✔	✔		✔				
17		✔			✔				
18	✔						✔		✔
19							✔		
20		✔	✔						
21									✔
22		✔	✔		✔				
23								✔	
24	✔								
25							✔		✔
26		✔							
27		✔	✔	✔					
28		✔	✔	✔					
29							✔		✔
30			✔		✔				

Tabelle 3.19**:** Übersicht der abgeleiteten Anforderungen je Anwendungsfall

Diese Anforderung widerspricht jedoch in keiner Weise der Anforderung mit der Ordnungsnummer 8 aus Abschnitt 3.2.4 („Etablierung einer vertrauenswürdigen Koordinationseinheit im GIDS“), die, wie in den weiteren Ausführungen beschrieben, auch als zentrale Anlaufstelle für die Koordination (in Anlehnung an z.B. ein GOC) eines GIDS dient. Auch der Aktor „Betreiber des GIDS“ (siehe auch Tabelle 3.6 auf Seite 48) steht hiermit nicht im Konflikt. Dieser ist insbesondere gefordert, da potentiell auch z.B. Virtuelle Organisationen, die keine eigene Instanz des GIDS betreiben wollen, durchaus zum Kundenkreis eines solchen Systems zählen können. Dies impliziert nicht, dass eine zentrale Organisation vorliegt, da durchaus weitere Instanzen, wie in den nachfolgenden Kapiteln genauer erläutert, untereinander kooperieren.

Heterogenität. Als Folge der dezentralen Organisation sind Grids oftmals sehr heterogen aufgebaut. Die eingebrachten Ressourcen können sich stark in Hardware, Software und Netzanbindung unterscheiden.

Diese Tatsache wird auch nochmals an dem gewählten Szenario des D-Grids (siehe Abschnitt 3.2.1) und den daraus abgeleiteten Anwendungsfällen deutlich. Der Umgang mit der Heterogenität auf mehreren Ebenen (verschiedene Grid-Middlewares, unterschiedliche Sicherheitsmechanismen und -vorkehrungen der Teilnehmer etc.) scheint folglich essentiell.

Verwendung standardisierter und offener Protokolle. Nur die Nutzung standardisierter, offener und breit unterstützter Protokolle sowie Schnittstellen als Basis von Grid-Technologien stellt die Interoperabilität innerhalb von komplexen, verteilten Strukturen nachhaltig sicher. Dies dient letztlich auch der Vermeidung von Abhängigkeiten.

Auch diese Tatsache ist bereits in der vorstehenden Anwendungsfall-getriebenen Anforderungsanalyse mehrfach zu Tage getreten. Da für diese Arbeit u.a. die Idee der kooperativen Nutzung von Sicherheitssystemen zur Formierung eines Grid-weiten Frühwarnsystems steht, bedarf es insbesondere der Verwendung standardisierter und offener Protokolle, um letztlich eine möglichst unkomplizierte Integration unterschiedlicher Systeme zu gewährleisten.

Hohe Leistungsfähigkeit. Der koordinierte Zugriff auf integrierte Ressourcen innerhalb von Grid-Verbünden soll zu einem im Vergleich zur Summe der Einzelsysteme signifikant größeren Nutzen und erhöhter Qualität des Gesamtsystems führen.

Weiter sind durch Grid-Strukturen noch einige generische Anforderungen bedingt, denen sich ein jedes neues System im Grid, also insbesondere auch ein Grid-basiertes Frühwarnsystem, stellen muss.

Skalierbarkeit. Nicht zuletzt durch die hohe Leistungsfähigkeit des Grids bedingt muss auch ein GIDS möglichst ressourcenschonend agieren und somit eine gute Skalierbarkeit gewährleisten. Dabei gilt es ein vermeintlich hohes Informationsaufkommen effizient zu verarbeiten und die Ressourcen im Grid nicht dafür zu verschwenden sich

selbst zu schützen, sondern eine minimale Auswirkung auf die Performanz des Grids zu erzielen (im Englischen auch mit *Low Intrusiveness* bezeichnet).

Nutzung von und Einbettung in bestehende Grid-Dienste. Die möglichst nahtlose Integration eines Frühwarnsystems in Grid-Umgebungen fördert u.a. auch die (Nutzer-)Akzeptanz des neuen Systems. Dazu ist die Nutzung bereits existierender Grid-Dienste und die Einbettung in die bestehende Grid-Landschaft notwendig. Diese Anforderungen in spezielleren und somit weniger generischen Ausprägungen spiegeln sich auch schon in der Anwendungsfall-getriebenen Anforderungsanalyse in Abschnitt 3.2 wider, z.B. durch die Anforderung „Einbindung bestehender AA-Mechanismen“ mit der Ordnungsnummer 4 aus Abschnitt 3.2.4.

Dynamik der Ressourcen. In Grid-Umgebungen ist es durchaus üblich, dass die Verfügbarkeit von Ressourcen sich dynamisch ändert. Dabei können entweder neue Ressourcen und Dienste in das Grid eingebracht oder aus dem Grid entfernt werden. In zweiterem Fall ist insbesondere in Bezug auf Frühwarnsysteme zu unterscheiden, ob ein Entfernen einer Grid-Ressource durch eine Störung oder einen erfolgreich durchgeführten Angriff bedingt ist oder ob dies bewusst und gewollt geschehen ist.

Durch diese Dynamik der Ressourcen bedingt folgt direkt eine immense Herausforderung, der ein Grid-weites Frühwarnsystem begegnen muss. Mit dem Hinzukommen und Wegfallen von Ressourcen (ob gewollt oder ungewollt spielt hier keine Rolle) ändert sich die Qualität und Quantität an Information, die dem GIDS zur Analyse bereitsteht. Es gilt auf der einen Seite damit überhaupt umgehen zu können und auf der anderen Seite die Information möglichst korrekt zu interpretieren, so dass keine unnötigen falsch positiven und falsch negativen (*False Positives* und *False Negatives*) Meldungen erzeugt werden bzw. deren Rate minimiert wird.

Dynamik der Nutzer und VOs. Unter anderem in [Schiffers, 2007] beschrieben, unterliegen VOs und deren Teilnehmer einer hohen Dynamik. Da diese durchaus zu dem Nutzerkreis eines GIDS gehören können, gilt es auch dieser Tatsache Rechnung zu tragen. Insbesondere bedeutet dieses, dass eine Integration eines Grid-basierten Frühwarnsystems in die VO-Management Werkzeuge und Tools vorgenommen werden muss. Unter Beachtung der zuvor genannten Anforderung der Nutzung von und Einbettung in bestehende Grid-Dienste kann diesem Tatbestand also bereits begegnet werden, da in hiesigem Fall die Anforderungen, die durch die Dynamik der Nutzer und VOs resultieren, eine Teilmenge der bereits angeführten Anforderungen sind.

Erweiterbarkeit und Flexibilität. Aspekte der Dynamik auf den verschiedensten Ebenen erheben vor allem auch die Anforderung, ein jedes Grid-System möglichst gut und einfach erweitern zu können und maximal flexibel gegenüber neuen Komponenten und Teilnehmern zu sein. Spezialfälle dieser pauschalen Anforderung finden sich auch bereits in der Anwendungsfall-getriebenen Anforderungsanalyse in Abschnitt 3.2 wieder. Ein Beispiel dafür ist die Anforderung des „Einbringen[s] zusätzlicher Infor-

mationsquellen in das GIDS während des Betriebs“ mit der Ordnungsnummer 5 in Abschnitt 3.2.4.

3.3.2 Mögliche Kooperationsmuster bei GIDS

Bereits einleitend ist die Idee aufgekommen, ein Grid-basiertes Frühwarnsystem als kooperatives System, an dem die einzelnen Teilnehmer des Grids partizipieren, zu gestalten. Dazu sollen insbesondere bestehende Informationsquellen der am GIDS beteiligten Parteien föderiert werden.

Es stellt sich sodann jedoch die Frage nach einem möglichen Kooperationsmuster für einen solchen Zusammenschluss. Grundlegend sind drei Arten der Kooperation denkbar, die in diversen Abwandlungen ausgeprägt sein können:

- keine Kooperation
- hierarchisch organisierte Kooperation
- Peer-to-Peer (P2P)

Eine extreme Form ist überhaupt keine Kooperation einzugehen. Dies widerspricht jedoch dem Ansatz dieser Arbeit, und somit wird dieser Fall nicht weiter betrachtet.

Eine weitere Möglichkeit besteht darin eine streng hierarchische Organisationsform zu wählen. Dies widerspricht jedoch u.a. der in Abschnitt 3.3.1 geforderten Eigenschaft der dezentralen Organisation eines Grid-Systems, wodurch zumindest eine strenge Hierarchie ausfällt.

Ein Peer-to-Peer Ansatz scheint als Alternative in Frage zu kommen, um insbesondere voranstehenden Anforderungen nachkommen zu können. Insbesondere kann hierdurch eine dezentrale Organisation bei erhöhter Ausfallsicherheit erreicht werden.

Ob eine strenge Einhaltung eines der vorgenannten Paradigmen notwendig ist, wird in den nachfolgenden Kapiteln 5 und 6 deutlich. Es ist durchaus denkbar die positiven Eigenschaften mehrerer Ansätze miteinander zu kombinieren um eine geeignete Organisationsform für den hiesigen Fall zu erzielen. Solche Mischformen finden sich häufig auch in Forschung und produktiven Einsätzen wieder wie auch in Kapitel 4 zu themenverwandten Ansätzen und Arbeiten nochmals deutlich wird.

3.3.3 Diskussion der Vertrauensbeziehungen unter Informationsanbietern

Um eine Kooperation unterschiedlicher Organisationen in einem beliebigen System auf Grundlage eines Informations- und Datenaustauschs gewährleisten zu können, muss eine Vertrauensbeziehung unter den Informationsanbietern bestehen. Dabei sind prinzipiell drei unterschiedliche Grade des Vertrauens festzustellen:

- kein Vertrauen

- eingeschränktes Vertrauen
- uneingeschränktes Vertrauen

Eine extreme Form einer Vertrauensbeziehung stellt die Tatsache dar, gar kein Vertrauen zu einem anderen Informationsanbieter zu haben. In Extremfällen kann dies implizieren, dass keine Informationen ausgetauscht werden und somit folglich auch keine Kooperation wie in Abschnitt 3.3.2 beschrieben stattfindet.

Das andere Extrem ist der Tatbestand des uneingeschränkten Vertrauens. Dies bedeutet, dass **alle** Informationen, wie sie bei den jeweiligen Informationsanbietern vorliegen, ungefiltert und nicht verfälscht weitergereicht werden. Diese Situation ist jedoch sehr unwahrscheinlich, da in realiter z.B. juristische Randbedingungen wie der Datenschutz einen solchen Informationsfluss unterbinden. Auch bei Kooperationen unter konkurrierenden Organisationen ist ein uneingeschränkter Datenaustausch eher unwahrscheinlich.

Aus voranstehend genannten Gründen folgt also die Anforderung an ein GIDS, mit eingeschränkten Vertrauensbeziehungen unterhalb der Kooperationspartner umgehen zu können. Bei korrekter Modellierung können dadurch ebenfalls die beiden Extremfälle, kein bzw. uneingeschränktes Vertrauen zu anderen Kooperationspartnern zu haben, abgedeckt werden. Ersteres entspricht nichts anderem als einer Informationstransformation in Form einer Nullfunktion[5] (alle Informationen werden verworfen), der zweite Fall der einer Identitätsabbildung (alle Informationen werden unverändert weitergeleitet).

Diese Arbeit soll sich nicht weiter mit dem Management von Vertrauensbeziehungen (*Trust Management*) befassen. Vielmehr ist diese Disziplin eine Notwendigkeit, um ein kooperatives Grid-basiertes Frühwarnsystem erfolgreich zu etablieren. Bei der weiteren Konzeption eines solchen müssen die Möglichkeiten, Trust Management Aspekte um- und durchzusetzen, vorgesehen werden. Dabei ist insbesondere zu beachten, dass Vertrauensbeziehungen in den wenigsten Fällen als ein einziger globaler Wert ausgedrückt werden können, sondern zumeist paarweise unterschiedliche Vertrauensbeziehungen zwischen den Teilnehmern (hier an einem GIDS) bestehen.

3.4 Kriterienkatalog für die Bewertung und Auswahl von IDS im Grid-Umfeld

Zusammenfassend ergibt sich aus voranstehenden Kapiteln nun ein Katalog an Anforderungen an ein IDS im Umfeld von Grids, der zur Bewertung und Auswahl eines IDS für Grids dienen kann. Zur Strukturierung der Anforderungen bietet sich eine Unterteilung in funktionale und nichtfunktionale Anforderungen, Sicherheitsanforderungen, organisatorische und Datenschutzanforderungen sowie Anforderungen an die Erkennungsleistung des IDS an.

[5] Sei $f : A \to B$. Wenn gilt, dass $f(x) = 0$ für alle $x \in A$, so ist f eine *Nullfunktion*.

Funktionale Anforderungen. Die funktionalen Anforderungen, die spezielle Funktionalitäten eines Systems spezifizieren, zerfallen in drei weitere Klassen. Zuerst finden sich eine Reihe Anforderungen, die generisch für ein jedes Intrusion Detection System scheinen, egal in welcher Umgebung es eingesetzt werden soll. Hinzu ergeben sich jedoch zwei weitere Klassen. Zum einen lassen sich ein Teil der funktionalen Anforderungen auf den Einsatzzweck des IDS in einer verteilten, föderierten Umgebung zurückführen. Diese sind zwar nicht primär durch Grids bedingt, sind aber dennoch insbesondere in Grids wie in anderen verteilten Systemen anzutreffen. Zudem lässt sich eine dritte Klasse der eindeutig durch Grids bedingten Anforderungen aufstellen. Die hierunter fallenden Anforderungen beziehen sich vorwiegend auf die Integration eines GIDS in die bestehenden Grid-Dienste sowie das eigene Bereitstellen eines neuen Grid-Dienstes.

Nichtfunktionale Anforderungen. Ähnlich wie bei den funktionalen Anforderungen lassen sich die nichtfunktionalen Anforderungen, die eine Eigenschaft eines Systems spezifizieren, in drei Unterklassen einteilen. Auch hier lässt sich die Unterteilung in die generischen, die unter anderem auch durch Grid bedingte sowie die eindeutig durch Grids bedingten Anforderungen vornehmen.

Sicherheitsanforderungen. In [Robertson u. Robertson, 2007] werden Sicherheitsanforderungen eigentlich als eine Unterklasse der nichtfunktionalen Anforderungen gesehen. Da im Rahmen dieser Arbeit jedoch ein System zur Durchsetzung von Sicherheitsanforderungen konzipiert wird, sollen die Sicherheitsanforderungen an dieses System gesondert hervorgehoben werden.

Aus den voranstehenden Teilabschnitten dieses Kapitels sind in Bezug auf die Sicherheit maßgeblich zwei Klassen von Anforderungen hervorgegangen. Zum einen sind dies Anforderungen an die Sicherheit der Kommunikation innerhalb und mit dem System, zum anderen werden Eigenschaften zur Nutzerverwaltung, also insbesondere der Authentifizierung und Autorisierung von Nutzern, gefordert.

Organisatorische und Datenschutzanforderungen. Durch eine Reihe von Randbedingungen unterliegt auch ein Grid-basiertes Intrusion Detection System organisatorischen und Datenschutzanforderungen. Auch diese Klasse von Anforderungen ist nach [Robertson u. Robertson, 2007] im eigentlichen Sinne eine Unterklasse der nichtfunktionalen Anforderungen. Es lässt sich dabei feststellen, dass eine Unterteilung zum einen in die eher prozessorientierten, organisatorischen Anforderungen eignet, eine zweite Unterklasse an Anforderungen bilden zumeist durch juristische Randbedingungen sowie Datenschutzanforderungen gegebene Notwendigkeiten ab.

Anforderungen an die Erkennungsleistung. Anforderungen, die sich an die Erkennungsleistung eines Grid-basierten Frühwarnsystems stellen, finden sich naturgemäß analog in herkömmlichen IDS wieder. Eine gewisse Ordnung dieser Klasse an Anforderungen lässt sich auch hier vornehmen, so existieren zum einen Anforderungen, die

sich an örtlichen Aspekten orientieren, zudem gilt es auch unterschiedliche Angriffstypen und -muster erkennen zu können. Also ergeben sich zum anderen Anforderungen bezüglich der Art und Dauer eines Angriffs.

Es ist zu beachten, dass die in diesem Kapitel erhobenen Anforderungen aus den Lebenszyklusphasen des Aufbaus und Betriebs eines Grid-basierten Frühwarnsystems entstammen. Es wird antizipiert, dass ein einmal etabliertes GIDS über die gesamte Zeitspanne des Betriebs des zu schützenden Grids existent bleibt und erst mit Beendigung der Grid-Infrastruktur ebenfalls terminiert wird. Ein entsprechender Prozess bleibt zu spezifizieren, dieser ist jedoch nicht Bestandteil dieser Arbeit. Dies bedeutet jedoch nicht, dass nachfolgende Anforderungen Änderungen am Frühwarnsystem (z.B. Hinzufügen und Entfernen von Grid-Sites oder Sensorik) zur Laufzeit ignorieren. Lediglich die Außerbetriebnahme des Gesamtsystems findet hier keine Beachtung.

Tabelle 3.20 stellt nochmals übersichtlich alle in diesem Kapitel erhobenen Anforderungen an ein Grid-basiertes Intrusion Detection System dar. Die Anordnung der Anforderungen folgt zuvorstehender Kategorisierung.

funktionale Anforderungen	Unterstützung verschiedener Granularitätsstufen bei der Berichterstattung	
	Berichterstattung zu qualitativ differierenden Angriffen	
	Aussagekräftige Informationsaufbereitung	
	Zugriffsmöglichkeit auf Sensordaten	
	Variationsmöglichkeit der Informationsquellen/Datenbasis zur Laufzeit	
	Proaktive Benachrichtigung der Kunden	
	u.a. auch Grid-bedingt	Nutzung verschiedener Kommunikationsmodelle (Push, Pull, Stream)
		Aggregatbildung
	Grid-bedingt	Informationspräsentation im Grid-Portal
		Nutzung bestehender Grid-Dienste
		Anbindung an bzw. Nutzung von bestehenden VO-Managementsystemen
nichtfunktionale Anforderungen	Integrierbarkeit in bestehende Management-Werkzeuge	
	Interoperabilität	
	Mandantenfähigkeit	
	Nachvollziehbarkeit durchgeführter Anfragen	
	Portabilität	
	Wiederverwendbarkeit	
	u.a. auch Grid-bedingt	Dezentrale Organisation
		Einheitliche Schnittstellen
		Erweiterbarkeit und Flexibilität
		Hohe Leistungsfähigkeit
		Skalierbarkeit
	Grid-bedingt	Dynamik der Nutzer und VOs
		Dynamik der Ressourcen
		Unterstützung etablierter (Grid-) Standards
		Unterstützung Virtueller Organisationen
Sicherheitsanforderungen	Kryptographische Anforderungen	Vertraulichkeit von Daten und Nachrichten
		Authentizität von Daten und Nachrichten
		Integrität von Daten und Nachrichten
		Einsatz symmetrischer und/oder asymmetrischer Kryptografie
		Kanal- oder nachrichtenbasierte Kommunikationssicherung

	Nutzerverwaltung	Integration in PKI
		Delegation von Identitäts- und Berechtigungsnachweisen
		Single Sign-On mit Proxy-Zertifikaten
		Einbindung bestehender AA-Mechanismen
		Zugriffsbeschränkung auf Informationen
Organisatorische und Datenschutzanforderungen	Organisatorische Anforderungen	Etablierung einer vertrauenswürdigen Koordinationseinheit
		Prozessspezifikation für (Signatur-) Updates
		Gewährleistung der Autonomie beteiligter Informationsanbieter
	Datenschutz	Anonymisierungs- und/oder Pseudonymisierungsmöglichkeiten
		Durchsetzung des Datenschutzes
		Nachhalten historischer Berichte
		Archivierung von Sensordaten
Erkennungsleistung	Örtliche Aspekte	Schutz der potentiellen Angriffsziele
		Geeignete Sensorplatzierung
	Angriffstypen und -muster	Erkennung verschiedener Angriffstypen (aktiv, passiv/autonom, DoS)
		Entdecken kurzzeitig angelegter bis hin zu zeitlich lang andauernden Angriffe

Tabelle 3.20: Zusammenfassung der erhobenen Anforderungen

Kapitel 4

State of the Art & Related Work

Inhalt des Kapitels

Der Bereich der themenverwandten Arbeiten lässt sich für die vorliegende Arbeit weitestgehend in zwei Blöcke unterteilen. Auf der einen Seite werden vielmals verteilte IDS auch in Grid-Umgebungen eingesetzt oder es werden zumindest recht große Anleihen bei ihnen gemacht. Abschnitt 4.1 greift exemplarisch zwei Systeme auf, die kurz vorgestellt werden

sollen. Der nachfolgende Abschnitt 4.2 stellt anschließend existierende Ansätze für Intrusion Detection Systeme vor, die eigens für Gris entwickelt wurden, bevor der abschließende Abschnitt 4.3 nochmals zusammenfassend die Defizite existierender Ansätze und Arbeiten herausstellt und dadurch nicht zuletzt implizit die wissenschaftliche Erfordernis dieser Arbeit nochmals hervorhebt.

4.1 Verteilte IDS

Verteilte Intrusion Detection Systeme stellen eine vergleichsweise weit gereifte Technologie bereit, die unter anderem in kooperativen und organisationsübergreifenden Szenarien zum Einsatz kommen kann. Für den Einsatz im Grid sind sie mit Einschränkungen auch geeignet, jedoch mangelt es ihnen in der Regel an für Grids spezifischen Eigenschaften oder notwendigen Anpassungen an Grid-Umgebungen. Dennoch sollen im Folgenden exemplarisch zwei verteilte IDS kurz vorgestellt werden, da eine Reihe Erkenntnisse und Erfahrungen, die im Laufe der schon länger währenden Entwicklung solcher Systeme haben gesammelt werden können, auch für die Konzeption eines IDS für Grids von Bedeutung sind.

4.1.1 Distributed Overlay for Monitoring InterNet Outbreaks (DOMINO)

Das *Distributed Overlay for Monitoring InterNet Outbreaks* (DOMINO) [Yegneswaran u. a., 2004] ist ein Ansatz um ein über viele administrative Domänen verteiltes Intrusion Detection System zu konzipieren. Die einzelnen Bestandteile des DOMINO sind als kooperierende Partner eines Overlay-Netzes[1] organisiert und versuchen dadurch bedingt Skalierbarkeit, die Integration heterogener Systeme sowie ein hohes Maß an Fehlertoleranz und Robustheit zu erreichen. Um das Overlay-Netz zu realisieren verwendet das DOMINO einen Peer-to-Peer (P2P) Ansatz, der auf verteilten Hash-Tabellen (engl. *Distributed Hash Table* (DHT)) basiert. Dadurch wird ein vollständig dezentralisiertes, skalierbares und sogar selbstorganisierendes System, das prinzipiell in der Lage ist sich Änderungen an der Infrastruktur dynamisch anzupassen, aufgebaut.

Abbildung 4.1 stellt grob den Aufbau des DOMINO dar. Wie viele andere Entwicklungen im Bereich verteilter IDS basiert auch das DOMINO auf *Chord* [Stoica u. a., 2001] als eine ausgereifte Instanziierung des P2P-Modells. Prinzipiell existieren drei verschiedene Typen an Knoten, die zusammen das Gesamtsystem bilden. Zentraler Bestandteil sind dabei die sogenannten *Achsenknoten* (engl. *Axis Nodes*, in Abbildung 4.1 einfach mit „A“ bezeichnet). Jeder dieser Achsenknoten hält sowohl eine lokale als auch eine globale Sicht aller sicherheitsrelevanten Ereignisse vor. Die lokale Sicht bezieht sich dabei auf die Ereignisse,

[1] Der Begriff *Overlay-Netz* ist definiert als ein Netz, das ein logisches Netz auf einer existierenden Netzinfrastruktur (dem sog. *Underlay-Netz*) aufbauend realisiert.

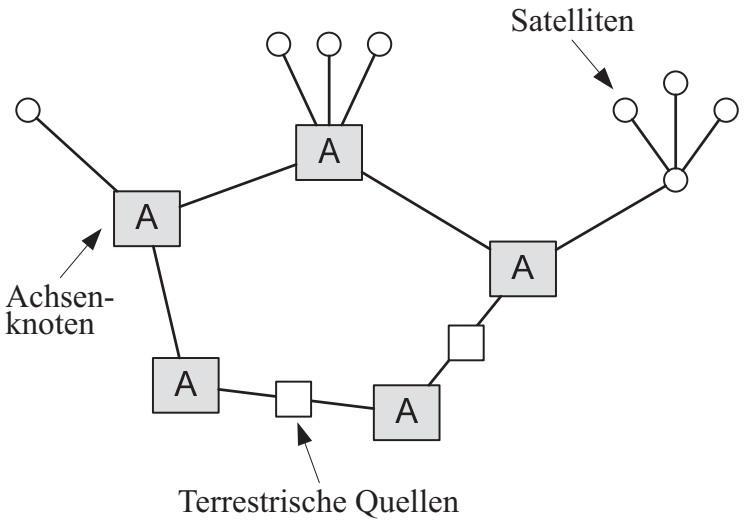


Abbildung 4.1**:** Distributed Overlay for Monitoring InterNet Outbreaks (DOMINO) nach [Yegneswaran u. a., 2004]

die im lokalen Netz durch die einem Achsenknoten zugeordneten *Satelliten* erhoben werden. Die Satelliten können wiederum untereinander hierarchisch organisiert werden. Die globale Sicht hingegen basiert auf den in regelmäßigen Abständen ausgetauschten Daten der Achsenknoten untereinander. Weiter liefern weniger vertrauenswürdige *terrestrische Quellen* (engl. *Terrestrial Contributers*) für eine Einbruchserkennung relevante Informationen. DOMINO stellt keine bestimmten Anforderungen an seine terrestrischen Quellen, weswegen hierfür eine große Menge an Informationslieferanten in Betracht gezogen werden können. Durch die hierdurch gewährleistete breite Datenbasis besteht die begründete Hoffnung die Erkennungsleistung des DOMINO erheblich zu verbessern. Zum Zwecke des Datenaustausches kommt eine Variante des *Intrusion Detection Message Exchange Format* [RFC4765, 2007]zum Einsatz, was die Interoperabilität unterschiedlichster Knoten sicherstellen und eine maximale Erweiterbarkeit mit Hinblick auf zukünftige Entwicklungen gewährleisten soll.

Zusammenfassung:

- Verteiltes IDS
- P2P Kooperation
- Overlay-Netz, nutzt verteilte Hash-Tabellen (DHT)
- Angriffserkennung durch sog. Achsenknoten
  - Sicht auf lokales Netz durch Satelliten
  - Globale Sicht durch Austausch von Informationen mit anderen Achsenknoten
- Nutzt offene Datenaustauschformate und zielt auf Erweiterbarkeit und Flexibilität ab

4.1.2 Large Scale Intrusion Detection Framework (LarSID)

Auch das *Large Scale Intrusion Detection Framework* (LarSID) [Zhou u. a., 2005, 2007] stellt ein verteiltes Intrusion Detection System unter der Nutzung einer leistungsfähigen und robusten Peer-to-Peer Architektur auf Basis verteilter Hash-Tabellen (hier kommt *OpenDHT* [OpenDHT] zum Einsatz) dar. Auf Basis eines dynamischen Overlay-Netzes wird eine Architektur entwickelt, mit der es möglich ist einzelne sicherheitsrelevante Ereignisse (z.T. bereits rudimentär vorverarbeitet) effizient zwischen einer Vielzahl von organisationsübergreifenden Systemen auszutauschen. Dadurch ist es durch das Konzept bedingt möglich, weit verteilt angelegte Angriffe mit Hilfe des LarSID möglichst frühzeitig zu erkennen.

Das LarSID ist als dienstorientiertes Rahmenwerk konzipiert, wobei die spezifische Dienstschicht vollkommen von der Schicht des Overlay-Netzes und dem dadurch bedingten Hashing getrennt wurde. LarSID nutzt ein flach organisiertes Publish-Subscribe Modell, in dem alle im Overlay-Netz organisierten Komponenten in regelmäßigen Abständen die in ihrem Bereich lokal erfassten und potenziell sicherheitsrelevanten Ereignisse mit den übrigen Teilnehmern austauschen können. Eine Besonderheit des LarSID ist, dass der Informationsaustausch unterhalb der Komponenten anonym geschieht. Alle am LarSID beteiligten Systeme haben im Wesentlichen zwei Aufgaben. Zum einen müssen sie sicherheitsrelevante Beobachtungen aufzeichnen, sie aggregieren und zur gemeinschaftlichen Auswertung dem Gesamtsystem verfügbar machen. Zum anderen erhalten sie von den anderen teilnehmenden Partnern Berichte, die es zu korrelieren gilt, um daraus potentielle Angriffe abzuleiten und darüber zu benachrichtigen.

Zum LarSID existiert eine prototypische Implementierung, anhand welcher insbesondere versucht wird eine Steigerung der Erkennungsleistung des Gesamtsystems zu erzielen. Hierzu wird ein empirisches Verfahren eingesetzt, das auf dem *DShield* Musterdatensatz (`http://www.dshield.org`) aufbaut. Hierdurch konnten bereits einige wichtige Erkenntnisse gesammelt werden, die maßgeblich zur Optimierung grundlegender Parameter des LarSID beigetragen haben.

Zusammenfassung:

- Verteiltes IDS
- P2P Kooperation
- Overlay-Netz, nutzt verteilte Hash-Tabellen (OpenDHT)
- Dienstorientiertes Framework
- Flach organisiertes Publish-Subscribe Modell
- Teilnehmer tauschen Informationen zu „suspicious activities“ aus

4.2 Grid-basierte IDS

Seit ca. 2002 finden auch Intrusion Detection Systeme in und für Grids in der Forschung ihren Platz. Bei den meisten der Arbeiten lassen sich eine Reihe an Analogien zu konventionellen verteilten IDS feststellen, Grid-Spezifika hingegen werden meist stiefmütterlich behandelt. Nachfolgend werden eine Reihe an Ansätzen für Grid-basierte IDS kurz vorgestellt. Bei einigen Ansätzen bleibt es dabei nur bei einem Konzept, zu dem leider keine (prototypische) Implementierung gefunden werden kann, einige Entwicklungen hingegen können auch mit einem Prototypen aufwarten. Die Reihenfolge der Vorstellungen orientiert sich aufsteigend nach dem Jahr der ersten Veröffentlichung der entsprechenden Arbeit.

4.2.1 Grid-Based Intrusion Detection System (GIDS)

Bereits 2003 ist in [Choon u. Samsudin, 2003] das *Grid-based Intrusion Detection System* (GIDS) vorgestellt worden. Dieser Vorschlag stellt eines der ersten Intrusion Detection Systeme für Grid-Umgebungen in der Forschung dar. Erstmals wird der VO-Aspekt in einem Intrusion Detection System aufgebracht. Vielmehr noch wird das GIDS selbst als eine VO modelliert, welche den Dienst des Grid-basierten IDS für andere VOs im Grid anbietet.

Als Anforderungen an das GIDS spezifizieren die beiden Autoren ...

1. den Umgang des GIDS mit der Grid-Umgebung (insbesondere dem Teilen von Ressourcen und der Kollaboration von Nutzern und Diensten),
2. die Autonomie des GIDS, so dass der Anwender und Administrator möglichst wenig interagieren müssen,
3. die Flexibilität des GIDS, insbesondere im Hinblick auf an die nutzerspezifische Systemanpassungen und den Einsatz von Policy-Rahmenwerken,
4. Skalierbarkeit,
5. Wiederverwendbarkeit
6. Erweiterbarkeit,
7. einen geringen Overhead, so dass das GIDS die Performanz des Grids nicht maßgeblich beeinflusst,
8. eine zeitnahe Auswertung von Angriffsberichten.

Abbildung 4.2 stellt schematisch nach der Arbeit in [Choon u. Samsudin, 2003] die Idee zum Aufbau des GIDS dar. Es folgt eine sehr wenig detaillierte Beschreibung des GIDS, vielmehr wird postuliert, dass eine Vielzahl verschiedener Auswertungsmechanismen und ein Policy-Based Management Ansatz zur Zugriffskontrolle zum Einsatz kommen kann, wobei das System für auf dem Globus Toolkit basierende Grids gedacht ist. Jedoch bewegt sich jegliche Beschreibung auf einem groben Niveau, vielmehr werden alleinstehend

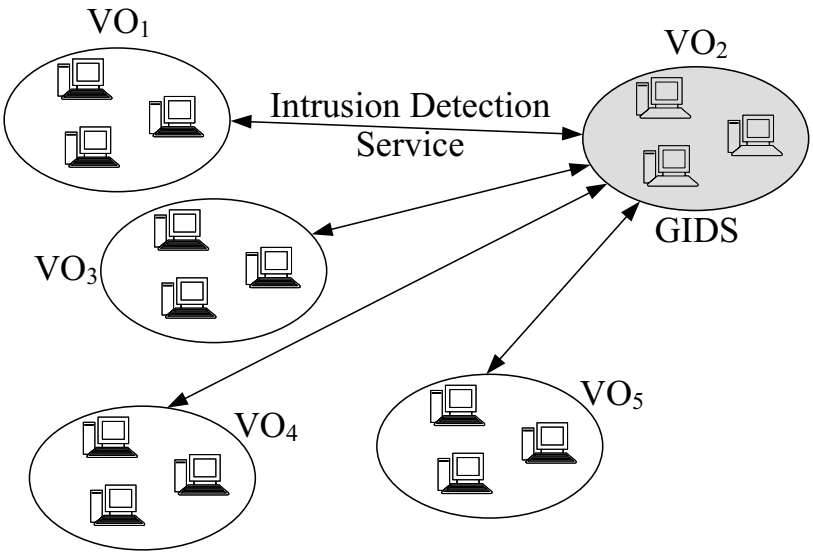


Abbildung 4.2**:** Grid-Based Intrusion Detection System (GIDS) nach [Choon u. Samsudin, 2003]

Schlagworte zur Datenanalyse in konventionellen Intrusion Detection Systemen benannt, diese aber nicht weiter verfolgt. Zur Sammlung einer Informationsbasis, auf der das GIDS arbeiten kann, wird nichts weiter verlautet. Weiter nachteilig erscheint, dass eine Angriffanalyse zentralisiert vorgenommen wird und seit 2003 eine Implementierung hierzu aussteht. Zusammenfassend wird der Vorschlag dieses GIDS als „Backup“ der *Grid Security Infrastructure* (GSI) bezeichnet, wobei aber die GSI keine jeglichen Frühwarnmechanismen vorsieht.

Zusammenfassung:

- Bringt VO-Aspekt auf
- Ist selbst als VO modelliert
- Bietet IDS-Dienst im Grid an
- Erkennbare Angriffstypen:
  - Der Ansatz fokussiert auf das Auditing des Globus Toolkit, Log-File Überwachung, Anomalieerkennung und signatur-basierte Missbrauchserkennung.
  - Das Konzept sieht jedoch eine Angriffserkennung mit beliebiger Analysefunktion vor, deswegen sind prinzipiell alle Angriffstypen erkennbar.
- Nachteile:
  - Einzig für das Globus Toolkit gedacht
  - Informationssammlung zur Angriffserkennung nicht berücksichtigt
  - Zentralisierte Angriffsanalyse
  - Keine Implementierung seit 2003

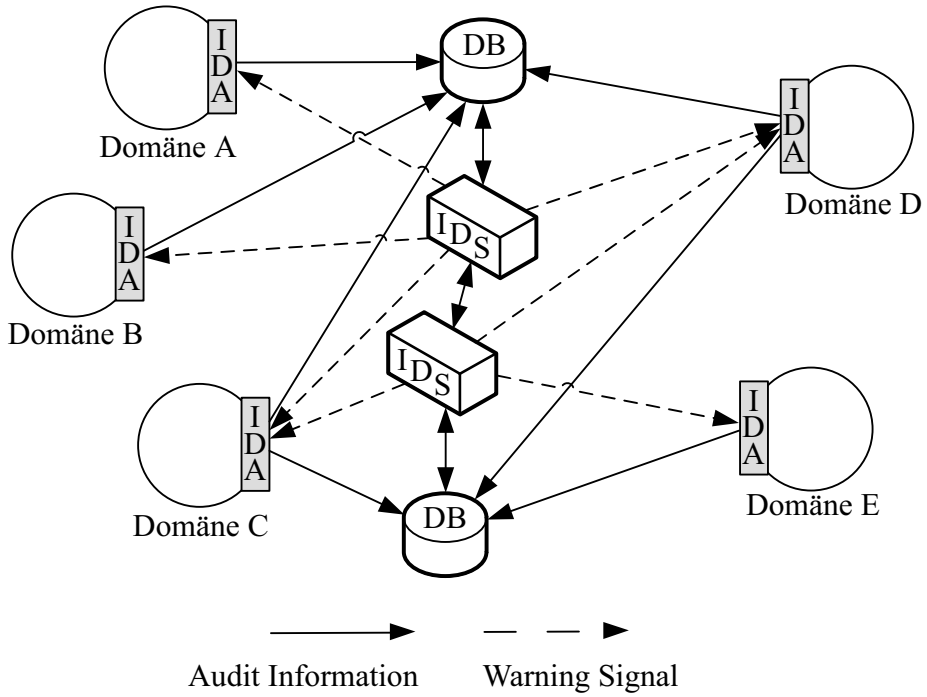


Abbildung 4.3**:** Grid Intrusion Detection Architecture (GIDA) nach [Tolba u. a., 2005a, b]

4.2.2 Grid Intrusion Detection Architecture (GIDA)

Die beiden Arbeiten [Tolba u. a., 2005a] und [Tolba u. a., 2005b] präsentieren den Ansatz der *Grid Intrusion Detection Architecture* (GIDA). Abbildung 4.3 stellt den schematischen Aufbau nach [Tolba u. a., 2005a, b] der GIDA dar. Die Autoren beschreiben für ihr System dabei die zwei zentralen Bestandteile des *Intrusion Detection Agent* (IDA) und des *Intrusion Detection Server* (hier mit IDS abgekürzt!). Ein IDA ist in dieser Architektur dafür verantwortlich Informationen für die GIDA zu sammeln, während der Intrusion Detection Server für die Auswertung der gesammelten Informationen verantwortlich zeichnet. Dabei ist eine nicht weiter beschriebene Kooperation der Server untereinander vorgesehen.

Bei der Konzeption der GIDA standen insgesamt sechs Anforderungen im Vordergrund:

1. Der Umgang der GIDA mit der Heterogenität im Grid
2. Skalierbarkeit
3. Umgang mit der Dynamik im Grid und Fehlertoleranz der GIDA
4. Keine zentralisierte Kontrolle
5. Die Nutzung von Standards, z.B. im Hinblick auf Protokolle
6. Nicht-triviale Auswertungsanalyse und die Unterstützung verschiedener Vertrauensbeziehungen

Bei der GIDA handelt es sich um ein verteiltes Gesamtsystem unter Nutzung eines Peer-to-Peer Ansatzes. Für die Angriffserkennung wird eine vollständige Informationsreplikation gewährleistet, die zusätzlich zu einer erhöhten Ausfallsicherheit beitragen soll. Als Analysefunktion soll ein Anomalieerkennungsverfahren zum Einsatz kommen, andere Angriffserkennungen sind derweil nicht vorgesehen. Es existiert eine prototypische Implementierung zur GIDA, die auf simulierten Grid-Infrastrukturen basiert und ein Anomalieerkennungsverfahren unter der Nutzung von *Learning Vector Quantization* (LVQ) als Spezialfall eines künstlichen Neuronalen Netzes implementiert. Allerdings setzt dieses System eine homogene Infrastruktur voraus, eine Erweiterung auf heterogene Umfelder ist noch ausstehend. Außerdem führt eine Variation der Teilnehmerzahl am IDS zu einer enormen Rate an Fehlalarmen und sämtliche VO-Aspekte im Grid bleiben unberücksichtigt.

Zusammenfassung:

- Verteiltes System, P2P Ansatz
- Vollständige Informationsreplikation
- Erkennbare Angriffstypen:
  - Als Auswertungslogik wird ein Anomalieerkennungsverfahren vorgeschlagen.
  - Die Analysefunktion ist prinzipiell beliebig, auch wenn dies nicht explizit durch die Autoren erwähnt wird.
  - Mit leichten Anpassungen sind vom Konzept her beliebige Angriffe erkennbar.
- Nachteile:
  - Erweiterung auf heterogene Umfelder ausstehend
  - Variation der Teilnehmerzahl führt zu enormer Fehlalarmrate
  - VO-Aspekt unberücksichtigt
  - Es kommen ausschließlich Anomalieerkennungsverfahren zur Angriffserkennung zum Einsatz

4.2.3 Performance-based Grid Intrusion Detection System (PGIDS)

In den Arbeiten von Leu et. al. [Leu u. a., 2005a, b] wird das *Performance-based Grid Intrusion Detection System* (PGIDS) vorgestellt. Dabei werden die Grid-Knoten als Analyseeinheiten eingesetzt, was eine nennenswert abweichende Vorgehensweise im Vergleich zu anderen Ansätzen darstellt. Insbesondere kann durch diesen Ansatz eine Lastverteilung, wenn auch zum Preis der zusätzlichen Belastung der verfügbaren Ressourcen im Grid, realisiert werden.

Das PGIDS besteht im wesentlichen aus den Komponenten *Dispatcher*, *Scheduler*, *Detection Nodes* (DN) und *Block List Database* (BLD). Als Annahme gilt, dass an einem PGIDS mehrere Subnetze beteiligt sind, die alle derselben *Network Management Unit* (NMU) und

somit der gleichen administrativen Domäne angehören. Ein jedes Subnetz erhält einen Dispatcher, der unter Nutzung des Spiegel-Ports seines zentralen Switches Flow-Daten aufzeichnet, die in sogenannten *Flow Files* (FF) abgelegt werden. Diese Dateien werden via GridFTP als Grid-Job zur Analyse verarbeitet. Der Scheduler ist dafür verantwortlich, dass ein geeigneter DN für die Analyse eines jeden FF ausgewählt wird. Die Ergebnisse der Analyse, also potentiell erkannte Angriffe, werden mit Informationen zu Zeit, Quell- und Zieladresse, Protokoll und Angriffstyp in der BLD hinterlegt. Diese Daten dienen dann dazu Firewalls zu rekonfigurieren um Angreifer effektiv vom Grid fernzuhalten.

Die Nachteile dieses Systems sind vor allem, dass es durch den zentralen Scheduler einen Single-Point-of-Failure bietet und ein vollständiges Vertrauen unterhalb der Teilnehmer voraussetzt. Zudem ist das System konstruktionsbedingt nur in der Lage unter Nutzung der Grid-Ressourcen netzbasierte Angriffe zu erkennen. Zudem werden nur Angriffe, die außerhalb des Grids ihren Ursprung finden, betrachtet, interne Angriffe hingegen bleiben unerkannt.

Zusammenfassung:

- Grid-Knoten als Analyseeinheiten
- Autonomes PGIDS je Partei
- Zentraler Scheduler je PGIDS
- Erkennbare Angriffstypen:
  - Laut Autoren sind Denial-of-Service Angriffe, verteilte DoS und Angriffe durch die Ausnutzung bestehender Schwachstellen in Software-Komponenten erkennbar.
  - Die Angriffserkennung erfolgt unter Nutzung Neuronaler Netze. Wenn auch nicht explizit erwähnt, scheint das Erkennungsverfahren jedoch austauschbar.
  - PGIDS erkennt durch seinen Aufbau bedingt (d.h. durch die geforderte Sensorplatzierung) nur netzbasierte Angriffe.
- Nachteile:
  - Erkennt nur netzbasierte Angriffe
  - Single-Points-of-Failure
  - Setzt vollständiges Vertrauen unter Teilnehmern voraus
  - Basiert ausschließlich auf dem Globus Toolkit als Middleware
  - Kann nur externe Angriffe erkennen

4.2.4 GridSec

In einer aus dem Projekt *GridSec* resultierenden Arbeit „Trusted Grid Computing with Security Binding and Self-Defense against Network Worms and DDoS Attacks“ [Hwang

u. a., 2005] wird eine Sicherheitsinfrastruktur vorgestellt, die Selbstverteidigungsmechanismen in Grid-Umgebungen zur Verfügung stellt. Dabei fokussiert das System auf die Abwendung von netzbasierten Angriffe, die durch Würmer initiiert sind, und das Verhindern verteilter Denial-of-Service Angriffe.

Architekturell orientiert sich das System an einem verteilten IDS, das durch ein Overlay-Netz die teilnehmenden Partner untereinander verbindet. Mit Hilfe dieses Netzes können Informationen zu Angriffen, die ein je Partner autonomes IDS generiert, unter Gewährleistung der Vertraulichkeit und Integrität ausgetauscht werden. Um die Vertrauenswürdigkeit eines angeschlossenen IDS zu bewerten, wird eine Methode vorgeschlagen, die aus verschiedenen Parametern zur Effizienz des IDS (z.B. Erfolgsrate, Auslastung etc.) aus historischen Daten einen sogenannten *Trust Index* (TI) eines jeden IDS bildet, wozu Ansätze aus der Fuzzy-Logic herangezogen werden.

Zusätzlich zur Erkennung von Angriffen durch die lokalen IDS-Instanzen wird eine globale Aggregation und Korrelation der Daten vorgenommen. Hierdurch wird das Ziel verfolgt, weit verteilte Angriffe effizienter erkennen zu können.

Neben der reinen Erkennung von aktiven Würmern und verteilten Denial-of-Service Angriffen steht auch ein Maßnahmenkatalog zur aktiven Abwehr erkannter Angriffe zur Verfügung. Das Einleiten von Gegenmaßnahmen wird dabei in Form von Verbindungsunterbrechungen vorgeschlagen.

Zusammenfassung:

- Verteiltes IDS
- Autonome, lokale IDS-Instanzen werden Grid-global zusammengeführt
- Lokal erkannte Angriffe werden Grid-global korreliert
- Gegenmaßnahmen in Form von Verbindungsunterbrechungen dienen als Verteidigungsmechanismen
- Erkennbare Angriffstypen:
  - Netzbasierte Würmer
  - Verteile Denial-of-Service Angriffe
- Nachteile:
  - Erkennt nur aktive Würmer und verteilte Denial-of-Service Angriffe
  - Keine Mechanismen zur Berichterstattung (Reporting)
  - Berücksichtigt keine VO-Aspekte

4.2.5 Grid-specific Host-based Intrusion Detection System (GHIDS)

Feng et. al. entwickeln in ihrer Arbeit das *Grid-specific Host-based Intrusion Detection System* (GHIDS) [Feng u. a., 2006]. Der Ausgangsgedanke dieser Arbeit ist, dass

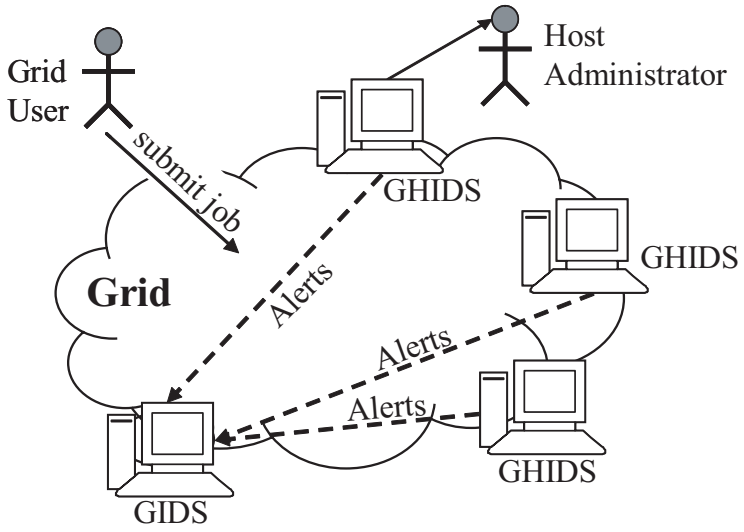


Abbildung 4.4**:** Architekturüberblick des GHIDS nach [Feng u. a., 2006]

herkömmliche hostbasierte IDS (HIDS) nicht in der Lage sind Grid-spezifische Angriffe zu erkennen und keine Grid-Spezifika (z.B. den Grid-Nutzer) kennen. Daraus resultiert die Idee, dass lokale Instanzen eines IDS die Aktionen der Grid-Anwender überwachen sollen und die daraus gewonnenen Berichte Grid-global korreliert werden. Dazu kommt eine angepasste Variante eines HIDS zum Tragen, die in der Lage ist, lokale Nutzerkennungen auf Kennungen im Grid abzubilden, so dass eine Korrelation im Grid ermöglicht wird. Diese Instanzen werden im Rahmen der Arbeit als *Grid-based HIDS* oder kurz *GHIDS* bezeichnet. Die von jedem GHIDS generierten Berichte und Alarme werden an ein *Grid-based IDS* (GIDS) weitergereicht, welches deren Korrelation Grid-global vornimmt. Zusätzlich können lokal generierte Alarme natürlich auch an einen lokal verantwortlichen Administrator übermittelt werden. Abbildung 4.4 illustriert das Konzept nochmals.

Zusammenfassung:

- Lokale HIDS werden um das Wissen von Grid-Nutzern erweitert
- Lokal erzeugte Alarme werden Grid-global korreliert
- Erkennbare Angriffstypen:
  - Dieser Ansatz erweitert hostbasierte IDS um das Wissen von Grid-Nutzeridentitäten.
  - Die Sensorplatzierung ist auf mit dem GHIDS ausgestatteten Grid-Rechner beschränkt.
  - Dadurch ist das GHIDS erstmal beschränkt auf die Erkennung hostbasierter Angriffe, durch eine Korrelation können noch gewisse Rückschlüsse auf verteile Angriffe gezogen werden.

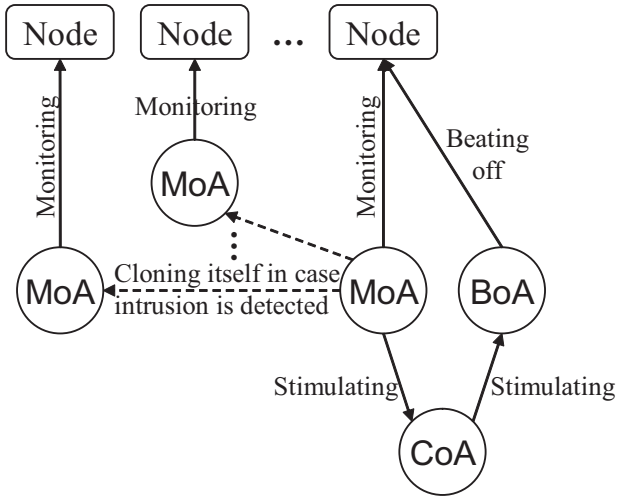


Abbildung 4.5: Das Intrusion Detection Modell der GIDIA nach [Gong u. a., 2006]

- Nachteile:
  - Sehr eingeschränkter Erkennungsfokus durch die Basis von HIDS
  - Insbesondere können netzbasierte Angriffe nur sehr eingeschränkt erkannt werden
  - Volles Vertrauen unter den Teilnehmern ist implizit und notwendig
  - VO-Aspekte bleiben vollkommen unberücksichtigt
  - Der Datenschutz wird nicht weiter betrachtet, Informationen werden ungefiltert weitergereicht

4.2.6 Grid Intrusion Detection Based on Immune Agent (GIDIA)

Eine zu allen anderen hier präsentierten Ansätzen vollkommen konträre Idee wird in [Gong u. a., 2006] vorgestellt. Gong et. al. stellen ein Intrusion Detection System vor, das sich das Konzept eines künstlichen Immunsystems zu Nutze macht. Abbildung 4.5 illustriert das Intrusion Detection Modell, welches der GIDIA zu Grunde liegt.

Sogenannte *Antigene* werden als Eigenschaften von Grid-Diensten spezifiziert und von *Monitoring Agents* (MoA) überwacht. Die Antigene teilen sich dabei in Eigenschaften auf, die der Anwender-, System-, Prozess- oder Paketebene zuordnen lassen. Innerhalb der MoAs wird auf Basis der erhobenen Monitoring-Informationen eine Einbruchserkennung durchgeführt. Sollte ein Angriff erfolgreich erkannt werden, so werden die sogenannten *Beating Off Agents* (BoA) stimuliert und mit möglichst detaillierten Informationen zu einem erkannten Angriff versorgt. Die BoAs reagieren nachfolgend auf den Angriff, wodurch das GIDIA nicht nur über einen Erkennungsmechanismus, sondern auch über eine Einheit zur Ausführung geeigneter Gegenmaßnahmen verfügt. Durch *Communicator Agents* (CoA) wird die Kommunikation unterhalb der Agenten sichergestellt und gewährleistet.

In [Ni u. a., 2007] wird im Jahre 2007 das *Self-adaptive Intrusion Detection System for Computational Grids* als auf das GIDIA aufbauendes System präsentiert. Es basiert primär auf den von der Grid Security Infrastructure bereitgestellten Diensten und bildet eine hierarchische Struktur von Agenten. Das Grid-basierte IDS wird anhand sogenannter *Trust Communities* (TC) unterteilt, die jeweils dynamisch als VO kreiert werden. Eine jede Trust Community kann selbstständig eine Angriffserkennung durchführen und nach Beschluss eines *Decision-Making Module* (DMM) unter Nutzung des *Response Module* (RM) auf einen erkannten Angriff reagieren.

Zusammenfassung:

- Idee eines künstlichen Immunsystems
- Monitoring Agenten überwachen einzelne Grid-Knoten
- Verteidigungsmechanismen durch Beating Off Agents
- Erkennbare Angriffstypen:
  - Die GIDIA nutzt einzig ein nicht änderbares oder erweiterbares Angriffserkennungsverfahren basierend auf einem künstlichen Immunsystem.
  - Die Platzierung der Sensoren ist auf die „Nodes“ beschränkt.
  - Daraus folgend werden diverse Angriffe aus allen Angriffsklassen nicht oder falsch erkannt werden.
- Nachteile:
  - Erkennungsfunktion nicht änderbar, z.B. keine signaturbasierte Analyse möglich
  - Durch eingeschränkte Sicht der MoAs und mangelnde Grid-globale Korrelation sind weit verbreitet angelegte Angriffe nur sehr schwer erkennbar
  - Der Datenschutz wird nicht weiter beachtet
  - Totale Kontrolle und absolutes Vertrauen ist Grundvoraussetzung für das GIDIA

4.2.7 Grid intrusion detection based on soft computing (SCGIDS)

In einem ähnlichen Ansatz zur zuvor erwähnten GIDIA versucht das *Grid intrusion detection based on soft computing* (SCGIDS) [Zhang u. Sun, 2006] das Normalverhalten eines Grid-Nutzer in einem neuronalen Netz zu modellieren und Abweichungen davon festzustellen, die berichtet werden können. Die Arbeit baut auf eine Vorarbeit von Kenny und Coghlan auf [Kenny u. Coghlan, 2004] und erweitert diese um den Ansatz des Soft Computing.

Das SCGIDS verfügt über eine Reihe Agenten (die sogenannten *SCGIDA*s, deren Abkürzung leider nicht expandiert wird), die das Nutzerverhalten der Grid-Nutzer, die zurzeit im Grid aktiv sind, beobachten und aufzeichnen. Zudem tauschen sie ihre Beobachtungen untereinander aus, um so jeder für sich eine Grid-weite Erkennung von Abweichungen in Bezug auf das zuvor festgestellte Normalverhalten durchzuführen. Diese Abweichungen werden durch das SCGIDS als Angriff interpretiert.

Ein SCGIDA besteht im Wesentlichen aus drei Datenbanken, der *Signature Identification Database* (SIDB), der *User Behavior Model Parameter Database* (UBMPDB) und der *Intrusion Evidence Database* (IEDB), sowie insgesamt acht weiteren Komponenten. Ein *Sniffing Agent* (SA) zeichnet Informationen des Grid-Nutzerverhaltens auf, die durch den *Signature Match Agent* (SMA) gegen die SIDB abgeglichen werden. Liegt kein Treffer vor, so wird ein Angriff antizipiert. Im Falle eines Treffers wird das erkannte Nutzerverhalten aus der UBMPDB in die UBMA transferiert. Nachfolgend kommt ein Neuronales Netz zum Einsatz, das unter Nutzung der aktualisierten UBMA die aktuellen Nutzungsprofile der Grid-Anwender auf ihre Normalität hin klassifiziert und die Parameter des Modells anpasst. Sollte hierbei ein abnormales Nutzerverhalten auffallen, kommt der *Trace-Back and Response Agent* (TBRA) zum Einsatz, der das abnormale Nutzerverhalten nachverfolgt und möglichst detailliertes Beweismaterial in der IEDB hinterlegt. Zur Interkommunikation der SCGIDAs kommt ein *Communication Agent* (CA) zum Einsatz, ein gewisser Selbstschutz des Systems wird durch den *Self Protection Agent* (SPA) durch regelmäßige Statusabfragen eines jeden Agenten realisiert.

Zusammenfassung:

- Basiert auf einer Vorarbeit, dem *Grid-wide Intrusion Detection* [Kenny u. Coghlan, 2004]
- Erweitert diesen Ansatz um Konzepte des Soft Computing unter Nutzung eines Neuronalen Netzes
- Kann in gewisser Art Beweise zu Angriffen für forensische Zwecke zur Verfügung stellen
- Erkennbare Angriffstypen:
  - Das SCGIDS einzig ein nicht änderbares oder erweiterbares Angriffserkennungsverfahren basierend auf Neuronalen Netzen und Ansätzen der Fuzzy-Logic.
  - Es wird versucht Abweichungen im Verhalten einzelner Nutzer zu erkennen.
  - Andere Angriffstypen sind nicht im Fokus des SCGIDS.
- Nachteile:
  - Erkennungsfunktion nicht änderbar, z.B. keine signaturbasierte Analyse möglich
  - Der Datenschutz wird nicht weiter beachtet
  - Absolutes Vertrauen unter den Agenten ist Grundvoraussetzung
  - Stellt eher ein verteiltes IDS dar, spricht keine Grid-Spezifika an

4.2.8 Integrated Grid-based Intrusion Detection System

In einer Arbeit der Universität von Santa Catarina (Brasilien) wird das *Integrated Grid-based Intrusion Detection System* vorgestellt [Schulter u. a., 2006a, b]. Dieses System

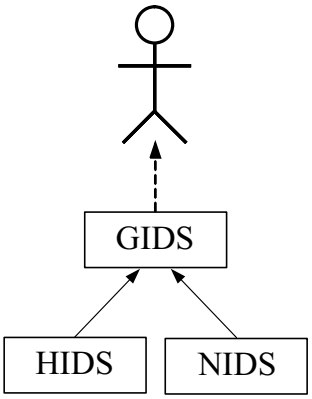


Abbildung 4.6: Integrated Grid-based Intrusion Detection System nach [Schulter u. a., 2006a, b]

fokussiert auf die Erkennung unerlaubter Zugriffe, missbräuchlicher Nutzung von Grid-Ressourcen, Exploits und host- oder netzspezifische Angriffe. Als generelle Unterschiede zwischen Angriffen auf Grids und solchen auf herkömmliche verteilte Systeme werden lediglich die zu erwartende höhere Geschwindigkeit und das aller Voraussicht nach größere Schadenspotenzial angesehen.

Die Grundidee ist ein übergeordnetes System zu entwickeln, das eine Zusammenführung von host- und netzbasierten IDS vorsieht, wie Abbildung 4.6 aus[Schulter u. a., 2006b] und [Schulter u. a., 2006a] darstellt. Auf diese Weise lassen sich übergreifend sicherheitskritische Vorfälle ableiten, um dann gegebenenfalls Alarmmeldungen generieren zu können. Der Datenaustausch geschieht unter Nutzung des XML-basierten Intrusion Detection Message Exchange Format (IDMEF).

Für die Umsetzung dieses Ansatzes wird eine abstrakte, dreigeteilte Architektur entwickelt, die sich auf Agenten, Analysekomponenten und eine Steuereinheit stützt. Agenten greifen die sicherheitsrelevanten Informationen der einzubindenden Intrusion Detection Systemen ab und speichern sie in Datenbanken im Grid. Jeder Zugriff eines Anwenders initiiert eine Überprüfung gegen das in der Datenbank gespeicherte Nutzerprofil. Sollten übermäßige Abweichungen zwischen einem aufgezeichneten Nutzerprofil und dessen tatsächlichem Verhalten auffallen, so wird eine weitergehende Analyse an Knoten des Grids mit freien Rechenkapazitäten vergeben, die ihrerseits mit den Datenbanken kommunizieren und ggf. Profile aktualisieren. Zusätzlich zur Analyse von Nutzerprofilen kann ebenfalls eine Aggregation und Korrelation gespeicherter sicherheitsrelevanter Ereignisse durchgeführt werden um so verteilte Angriffe identifizieren zu können.

Die Weiterentwicklung dieses Ansatzes ist noch immer im Gange, aktuelle Arbeiten wie [Silva u. a., 2007] befassen sich nach wie vor mit diesem System, dessen Entwicklung und Tests seiner Leistungsfähigkeit.

Zusammenfassung:

- Übergeordnetes System zu bestehenden HIDS und NIDS

- Datenaustausch IDMEF-basiert (XML)
- Speichert Informationen in DBs im Grid
- Erkennbare Angriffstypen:
  - Durch die Kombination von host- und netzbasierten IDS sind vom Konzept alle Angriffstypen erkennbar.
  - Die Arbeit beschränkt sich jedoch auf die Erkennung unerlaubter Zugriffe, missbräuchlicher Nutzung von Grid-Ressourcen, Exploits und host- oder netzspezifische Angriffe.
- Nachteile:
  - Zentralisierter Aufbau
  - Grid-Aspekte unberücksichtigt – es handelt sich eher um ein verteiltes IDS
  - Der Datenschutz wird nicht weiter beachtet
  - Volles Vertrauen unter den Teilnehmern ist implizit und notwendig
  - VO-Aspekte bleiben vollkommen unberücksichtigt

4.3 Defizite bestehender Ansätze

Dieser Abschnitt versucht einen Abgleich der zuvor in Abschnitt 4.2 vorgestellten Grid-spezifischen Intrusion Detection Systeme mit dem in Kapitel 3 erstellten und in Abschnitt 3.4 mit Tabelle 3.20 zusammengefassten Kriterienkatalog vorzunehmen. Die Hauptproblematik dabei ist, dass die vorliegenden Beschreibungen der themenverwandten Arbeiten zumeist durch die Form ihrer Veröffentlichung als einzelne Konferenzbeiträge und zudem in Mangel einer prototypischen Implementierung nur eingeschränkt mit dem hiesigen Kriterienkatalog abgeglichen werden können. Dennoch lassen sich im Folgenden die Hauptdefizite aller bisher bestehenden Arbeiten recht gut herausarbeiten.

Aus zuvorstehend genannten Gründen ist nachfolgend der Versuch einer Übersicht zu den jeweiligen Stärken und Schwächen der einzelnen Ansätze wie folgt tabellarisch dargestellt: Die Ordnung der Tabellen orientiert sich an den fünf groben Kategorien funktionale und nicht-funktionale Anforderungen, Sicherheitsanforderungen, organisatorische und Datenschutzanforderungen sowie Anforderungen an die Erkennungsleistung, wie sie zuvor bereits in Tabelle 3.20 aufgestellt worden sind. Die Semantik der Symbole innerhalb der Tabellen ist dabei wie folgt:

✔ Diese Anforderung ist erfüllt. Dies geht entweder eindeutig aus der Beschreibung des Ansatzes hervor (implizit) oder wird sogar explizit als Eigenschaft des Systems von den Autoren in ihrer Arbeit genannt.

(✔) Diese Anforderung ist mit Einschränkungen erfüllt. Entweder sind Teile dieser Anforderung erfüllt bzw. ist die Anforderung in Ansätzen erfüllt oder die Anforderung kann

durch eine geschickte Implementierung des Systems unter Umständen noch erfüllt werden. Jedoch machen die Autoren keine explizite Aussage zu einer solchen Eigenschaft ihres Systems, eine Erfüllung lässt sich jedoch mit etwas Geschick aus den vorliegenden Veröffentlichungen ableiten.

✘ Diese Anforderung ist nicht erfüllt, auch lässt sie sich nicht trivial in den Ansatz mit einbringen.

'leer' Da oftmals entweder gar keine oder nur sehr kurze Kriterienkataloge (ca. ein halbes Dutzend Anforderungen umfassend) bei der Konzeption der jeweiligen Systeme im Vordergrund standen, lässt sich zu einer Vielzahl an Anforderungen leider keine Aussage tätigen. Die entsprechenden Einträge in den nachfolgenden Tabellen sind leer gelassen, da in diesen Fällen auch eine Interpretation der Veröffentlichungen keine Aussage zu einer solchen Eigenschaft zulässt.

4.3.1 Funktionale und nicht-funktionale Anforderungen

Anforderung		System (*siehe Abschnitt ...*)							
		4.2.1	4.2.2	4.2.3	4.2.4	4.2.5	4.2.6	4.2.7	4.2.8
Granularität bei Berichten		(✔)							
Versch. Qual. bei Berichten									
Aussagekraft									
Zugriff auf Sensordaten			✘	(✔)		✘	✘	✘	✘
Variation der Sensoren			✘	✔	(✔)	(✔)			
Proaktive Benachrichtigung		✔	✘		✘	(✔)	✘	(✔)	✔
u.a. auch Grid-bedingt	versch. Kommunikationsmodelle		✘	✘	✘	✘	✘	✘	✘
	Aggregatbildung	✘	✘	(✔)	✔		✘		(✔)
Grid-bedingt	Grid-Portal	(✔)	✘	✘	✘	✘	✘	✘	✘
	Nutzung von Grid-Diensten	(✔)	(✔)	✔	✘	(✔)	✘	✘	✔
	Anbindung an VO-Mgmt.	(✔)	✘	✘	✘	✘	✘	✘	✘

Legende:

✔	Anforderung erfüllt
(✔)	Anforderung mit Einschränkungen erfüllt
✘	Anforderung nicht erfüllt
'leer'	Keine Angabe der Autoren

Tabelle 4.1**:** Defizite bei funktionalen Anforderungen

Anforderung		System (*siehe Abschnitt ...*)							
		4.2.1	4.2.2	4.2.3	4.2.4	4.2.5	4.2.6	4.2.7	4.2.8
Integrierbarkeit		✔	✔	(✔)		(✔)			(✔)
Interoperabilität			✔	(✔)		(✔)			✔
Mandantenfähigkeit		✘	✘	✘	✘	✘	✘	✘	✘
Nachvollziehbarkeit								✔	
Portabilität			(✔)	(✔)		(✔)			✔
Wiederverwendbarkeit			(✔)	(✔)		(✔)			✔
u.a. auch Grid-bedingt	Dezentrale Organisation	✘	✔	✘	(✔)	✘	(✔)	✔	✘
	Einh. Schnittstellen		✔	(✔)					✔
	Erweiterbarkeit, Flexibilität	(✔)	(✔)	(✔)		✘	✘	✘	(✔)
	Leistungsfähigkeit		✔	✘		✘	(✔)		
	Skalierbarkeit		(✔)	✘		(✔)	(✔)		
Grid-bedingt	Dynamik der Nutzer und VOs	(✔)	(✔)	✘	✘	✘		(✔)	✘
	Dynamik der Ressourcen	✘	✔	(✔)	(✔)	✔			
	Unterstützung v. Standards	(✔)	✔	✔		(✔)			✔
	Unterstützung v. VOs	(✔)	✘	✘	✘	✘	✘	✘	✘

Legende:

✔	Anforderung erfüllt
(✔)	Anforderung mit Einschränkungen erfüllt
✘	Anforderung nicht erfüllt
'leer'	Keine Angabe der Autoren

Tabelle 4.2**:** Defizite bei nicht-funktionalen Anforderungen

Die Tabellen 4.1 und 4.2 stellen übersichtlich die Hinlänglichkeiten und Defizite der zuvor vorgestellten Grid-spezifischen IDS in Bezug auf die zuvor erhobenen funktionalen und nicht-funktionalen Anforderungen dar. Es stellt sich im Bereich der durch das Grid bedingten Anforderungen schnell heraus, dass insbesondere der Begriff der Virtuellen Organisationen in die bisherigen Ansätze gar keinen bis kaum Einzug gehalten hat. Die einzige Ausnahme bildet dabei der älteste Ansatz, das *Grid-Based Intrusion Detection System (GIDS)* (siehe Abschnitt 4.2.1), das zumindest das IDS selbst als VO modelliert.

Auch aus dieser Tatsache folgend fällt auf, dass ebenfalls keine Kundenbegriffe und damit verbundene Geschäftsmodelle mit den Systemen aufkommen. Insbesondere Mandantenfähigkeiten und Berichterstattungsmechanismen bzw. die Repräsentation der Sicher-

heitsinformationen in Grid-Portalen kommt bei allen Ansätzen zu kurz oder wird schlicht ignoriert. Entsprechend sehen diese Ansätze zumeist eine unidirektionale Kommunikation zwischen ihren Sensoren und ihrer Analyseeinheit vor, da Ansätze für forensische Zwecke oder auch der Erfüllungsnachweis eines Service Level Agreements (SLA) zu einem späteren Zeitpunkt nicht mit betrachtet werden.

4.3.2 Sicherheitsanforderungen

Anforderung		System (*siehe Abschnitt ...*)							
		4.2.1	4.2.2	4.2.3	4.2.4	4.2.5	4.2.6	4.2.7	4.2.8
Kryptographische Anforderungen	Vertraulichkeit	✔	✔	✔	✔				
	Authentizität	(✔)	✔	✔	✔				
	Integrität	(✔)	✔	(✔)	✔				
	(a)symmetrische Kryptografie	✔	✔	✔	✔				
	Kommunikations-sicherung	✔	✔	✔	✔				
Nutzerverwaltung	Integration in PKI	(✔)	(✔)	✔	✘		✘	✘	✘
	Delegation von Identitäts- und Berechtigungs-nachweisen	(✔)	(✔)	(✔)	✔		✘	✘	✘
	Single Sign-On		✘	✘	✘	✘	✘	✘	✘
	Nutzung best. AA-Mechanismen	(✔)	(✔)	✘	✘		✘	✘	✘
	Zugriffs-beschränkung	✔	✘	✘	✘	✘	✘	✘	✘

Legende:

✔	Anforderung erfüllt
(✔)	Anforderung mit Einschränkungen erfüllt
✘	Anforderung nicht erfüllt
'leer'	Keine Angabe der Autoren

Tabelle 4.3**:** Defizite bei Sicherheitsanforderungen

Im Rahmen der Sicherheitsanforderungen bewegen sich alle dargelegten Ansätze auf einem recht hohen Niveau, wie auch Tabelle 4.3 verdeutlicht. Insbesondere im Bereich der Kryptographie ist durchweg ein hoher Standard angesetzt, da die meisten Systeme ohnehin auf im Grid etablierte Sicherheitsmechanismen zurückgreifen. Einige Ansätze sprechen zwar nicht explizit von ihrer kryptographischen Fähigkeit, dennoch ist eine solche für ein sicherheitsrelevantes System zu antizipieren.

Unzulänglichkeiten hingegen enthüllen sich im Bereich der Nutzerverwaltung. Wie bereits zuvor festgestellt, liegt ein Kundenbegriff und eine ordentliche Informationsaufbereitung für Grid-Anwender und VOs nicht im Fokus der beschriebenen Systeme. Folglich ist eine Integration in bestehende Public Key Infrastrukturen (PKI) und damit verbunden die Nutzung von Grid-eigenen AA-Mechanismen zumeist nur stiefmütterlich vorgenommen worden. Zugriffsbeschränkungen auf einzelne Berichte und Informationen gibt es so gut wie in keinem Ansatz, was zum einen auf den Mangel eines Kundenbegriffs und zum anderen auch auf das später noch festzustellende notwendige Vertrauensverhältnis unterhalb der am IDS teilnehmenden Informationsanbieter zurückzuführen ist.

4.3.3 Organisatorische und Datenschutzanforderungen

Anforderung		System (*siehe Abschnitt ...*)							
		4.2.1	4.2.2	4.2.3	4.2.4	4.2.5	4.2.6	4.2.7	4.2.8
Organisatorische Anforderungen	Koordinationseinheit	✘	✘	✘	✘	✘	✘	✘	(✔)
	Prozess für Updates	✘	✘	✘	✘	✘	✘	✘	✘
	Autonomie der Informationsanbieter	(✔)	(✔)	✘	(✔)	✘	✘	✘	✘
Datenschutz	Anonymisierung, Pseudonymisierung	✘	✘	✘	✘	✘	✘	✘	✘
	Durchsetzung des Datenschutzes	✘	✘	✘	✘	✘	✘	✘	✘
	Nachhalten alter Berichte							✔	
	Archivierung v. Sensordaten							(✔)	

Legende:

✔	Anforderung erfüllt
(✔)	Anforderung mit Einschränkungen erfüllt
✘	Anforderung nicht erfüllt
'leer'	Keine Angabe der Autoren

Tabelle 4.4**:** Defizite bei organisatorischen und Datenschutzanforderungen

Aus Tabelle 4.4 geht hervor, dass im Wesentlichen alle vorgestellten Ansätze und Systeme für Grid IDS vorwiegend auf technische, nicht aber organisatorische Aspekte fokussieren. Das gilt in vielen Fällen auch für eine notwendige Autonomie der am Gesamtsystem beteiligten Informationsanbieter, die in einigen Fällen zumindest organisatorisch gegeben ist.

Jedoch umfasst dies nicht die Autonomie bei der Informationsweitergabe und dem Datenschutz. In allen Ansätzen wird ein volles Vertrauen und eine (fast) uneingeschränkte Weitergabe an lokal gesammelten Informationen für das Funktionieren des Grid-globalen Systems vorausgesetzt. Dieser Fakt sollte sich in einem produktiven Umfeld einer Kooperation zwischen administrativ und organisatorisch voneinander unabhängigen Organisationen als absolut nicht durchsetzbar erweisen.

4.3.4 Anforderungen an die Erkennungsleistung

Anforderung		System (*siehe Abschnitt ...*)							
		4.2.1	4.2.2	4.2.3	4.2.4	4.2.5	4.2.6	4.2.7	4.2.8
Örtliche Aspekte	Schutz aller Angriffsziele		(✔)	✘		✘	✘	✘	(✔)
	Geeignete Sensorplatzierung		(✔)	✘		✘	✘	(✔)	(✔)
Angriffstypen und -muster	alle Angriffstypen	(✔)	(✔)	✘	✘	✘	✘	✘	(✔)
	versch. Angriffsdauer	(✔)	(✔)	(✔)	(✔)	(✔)	(✔)	(✔)	(✔)

Legende:

✔	Anforderung erfüllt
(✔)	Anforderung mit Einschränkungen erfüllt
✘	Anforderung nicht erfüllt
'leer'	Keine Angabe der Autoren

Tabelle 4.5**:** Defizite bei möglicher Erkennungsleistung

In Bezug auf die Erkennungsleistung der verschiedenen Ansätze zeigt sich ein differenziertes Bild. Ungefähr die Hälfte der Arbeiten ist prinzipiell in der Lage, alle möglichen Angriffsziele zu schützen und alle denkbaren Angriffstypen zu erkennen. Zumeist sind in diesen Fällen jedoch die eventuell vorhandenen oder auch nur vorgeschlagenen Prototypen der Systeme auf eine spezielle Ausprägung der Sensorplatzierung (z.B. auf den einzelnen Grid-Knoten oder an zentralen Firewalls oder Routern etc.) und/oder auf eine spezielle Auswertungsfunktion (z.B. signaturbasiert oder ein Anomalieerkennungsverfahren) festgelegt, jedoch ist eine Anpassung bzw. Variation und Ergänzung leicht denkbar und zum Teil sogar konzeptuell vorgesehen.

Die andere Hälfte der Ansätze hingegen ist bereits vom Konzept her an bestimmte Grundvoraussetzungen gebunden. So sind u.a. zum Beispiel eben Sensorplatzierungen und Analysefunktionen festgelegt und somit inhärent und nicht änderbar. Dadurch bedingt ist natürlich nicht zwingend jedes potentielle Angriffsziel geschützt bzw. jeder erdenkliche Angriffstyp erkennbar. Diesen Systemen fehlt es an einer hinreichenden Flexibilität.

Zuvor beschriebene Fakten lassen sich nochmals übersichtlich aus Tabelle 4.5 ableiten.

Kapitel 5

Ein Intrusion Detection System für Grids

Inhalt des Kapitels

Basierend auf den in Kapitel 3 abgeleiteten Anforderungen an ein IDS für Grids und den in Kapitel 4 gewonnenen Erkenntnissen zu bestehenden Ansätzen, wird in diesem Kapitel ein neuartiger Ansatz für ein Intrusion Detection System für Grids vorgestellt. Dabei wird nach wie vor die grundlegende Idee der kooperativen Nutzung von lokalen Sicherheitssystemen und der Austausch von Angriffsdaten verfolgt. Abschnitt 5.1 illustriert diese Idee nochmals und gibt einen ersten Überblick über die angestrebte Architektur. In Abschnitt 5.2 wird anschließend eine Detaillierung des Vorschlags vorgenommen und es werden die einzelnen notwendigen Komponenten und ihre Funktion anhand der erhobenen Anforderungen abgeleitet und beschrieben. Abschließend betrachtet Abschnitt 5.3 den Vorschlag kritisch und versucht potentielle durch die Architektur bedingte Schwächen und Herausforderungen zu diskutieren, bevor im nachfolgenden Kapitel 6 eine prototypische Implementierung des GIDS beschrieben und ein Abgleich mit dem in Abschnitt 3.4 zusammengefassten Kriterienkatalog durchgeführt wird.

5.1 Architekturüberblick zum Aufbau eines Grid-basierten IDS

Bereits einleitend in dieser Arbeit ist die Idee aufgekommen, GIDS als Föderation aus bestehenden, für die Ressourcenanbieter eines Grids spezifischen, sicherheitsrelevanten Komponenten zu einem Grid-weiten Frühwarnsystem zu konzipieren. Aus der Anforderungsanalyse in Kapitel 3 sind unter anderem Anforderungen abgeleitet worden, die die Autonomie der einzelnen Teilnehmer eines Grid-basierten IDS fordern, was nicht zuletzt für die Akzeptanz eines solchen Systems zwingend notwendig ist. Daraus abgeleitet bedingt sich eine verteilte Struktur und es entsteht eine lose Kopplung unterhalb der am GIDS beteiligten Partner, die daraus folgend jeder für sich organisatorisch und administrativ wie auch technisch unabhängig und autonom agieren können und sogar müssen.

Abbildung 2.1 auf Seite 7 hat in Kapitel 2.1 bereits die grundlegende Architektur und ihre Komponenten kurz umrissen. Für den hiesigen Anwendungsfall zur Konzeption eines föderierten Frühwarnsystems unter Wahrung der Autonomie aller teilnehmenden Partner bietet sich folglich eine Art der Schachtelung dieses allgemeingültigen Aufbaus an, wie es in Abbildung 5.1 illustriert ist. Ein Informationslieferant für das Grid-basierte IDS, also ein *Grid-Sensor* oder *-Agent*, soll durch eine administrative Domäne bzw. einen Ressourcenanbieter im Grid (hier auch synonym als *Site* bezeichnet) gegeben sein. Durch eine solche Konzeption lassen sich prinzipiell sowohl sämtliche Rohdaten (z.B. von bestehender, Site-spezifischer IDS-Sensorik, Firewall Logdateien, Netflow-Traces etc.) als auch veredelte Informationen durch z.B. bestehende, Site-lokale IDS-Installation in das Grid-basierte Frühwarnsystem einbringen.

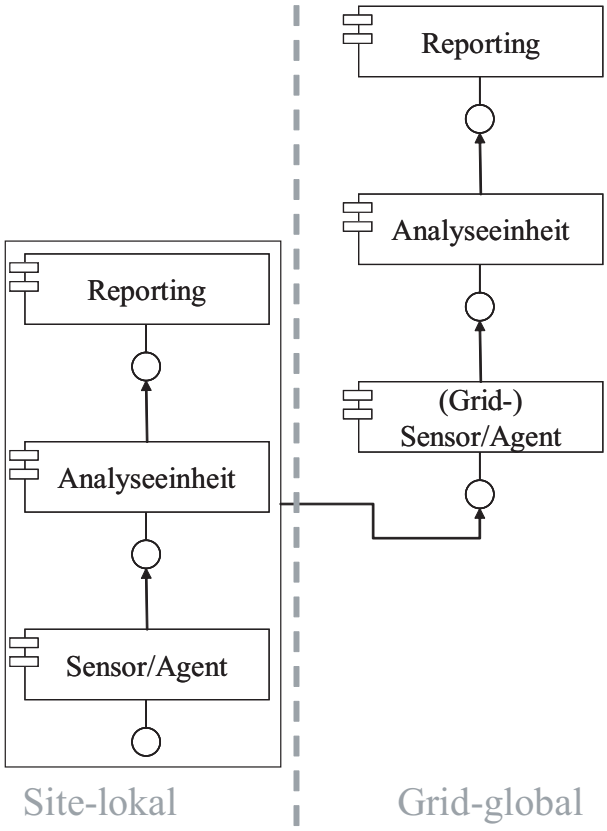


Abbildung 5.1: Grundlegende Idee zum Aufbau eines Grid-basierten Intrusion Detection Systems

Diese grundsätzliche Idee führt nun weiter zu einem Grid-globalen Aufbau eines IDS wie es in Abbildung 5.2 wenig detailliert aus einer Vogelperspektive dargestellt ist. Ein jeder Teilnehmer des GIDS erhält eine zentrale Datenbank, in die alle verfügbaren, für die Sicherheit relevanten Informationen abgelegt werden können. Wie bereits zuvor angesprochen können dies zum einen Rohdaten (z.B. von versuchten Zugriffen auf gesperrte Ports an einer Firewall) oder auch bereits veredelte oder aggregierte Informationen wie Berichte lokal installierter IDS sein. In jedem Fall gilt es zu beachten, dass an dieser Stelle eine Einigung auf ein einheitlichen Daten- und Informationsmodell notwendig ist. Eine Detaillierung findet sich in Abschnitt 5.2.

An einen solchen zentralen Datenspeicher angeschlossen kann ein Agent unter Beachtung einiger notwendiger Randbedingungen Informationen an ein Grid-weites IDS weiterreichen. Weiter kann dieser Agent Informationen aus dem GIDS entgegennehmen und im zentralen Datenspeicher hinterlegen. Zur Kommunikation bietet sich ein Multicast oder sogar Broadcast der Daten unterhalb der Agenten an, da dadurch bedingt ein verteilter Ansatz mit einer vollständigen Datenreplikation verwirklicht werden kann. Schlussendlich hat jeder teilnehmende Ressourcenanbieter die Möglichkeit, eine eigene Instanz des GIDS auf Basis seiner lokal vorgehaltenen Daten zu betreiben.

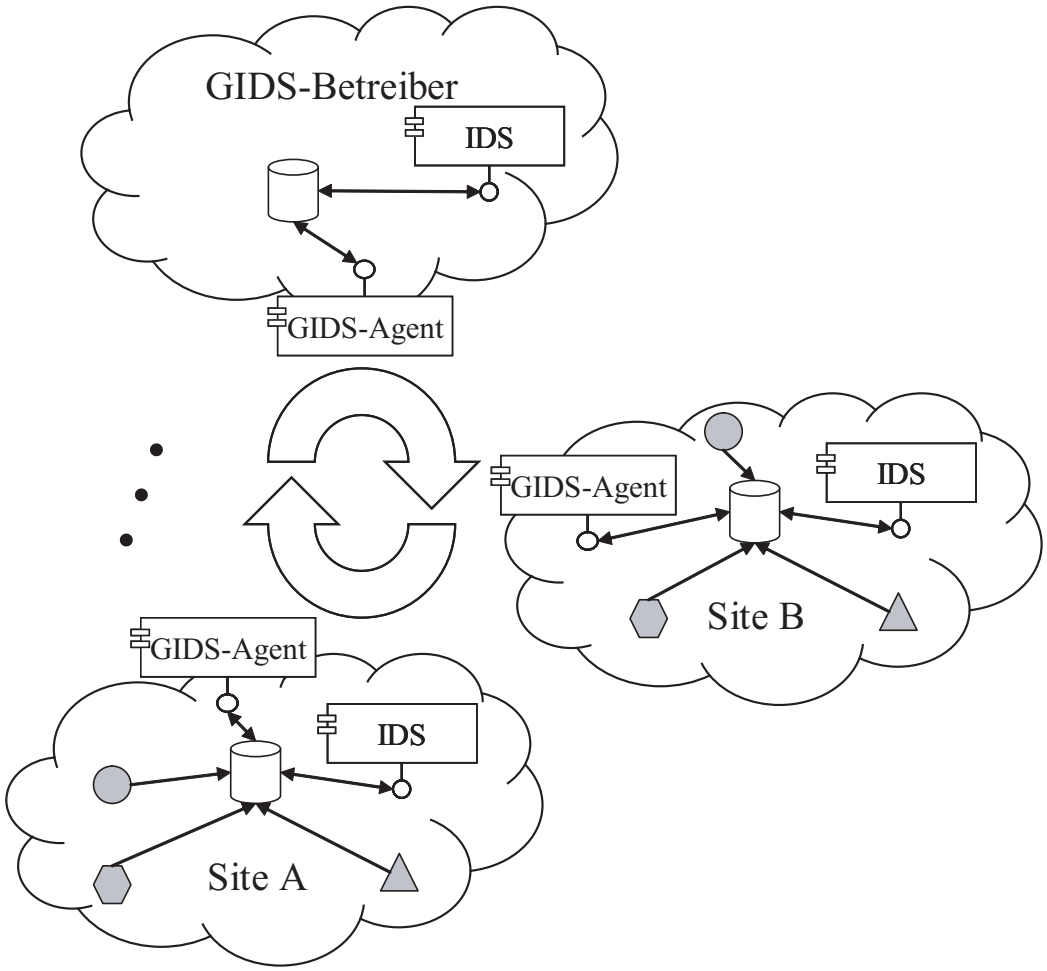


Abbildung 5.2**:** Grobgranulare Übersicht der Architektur des GIDS

Aus der Anforderungsanalyse ist hervorgegangen, dass ein Betreiber für eine Grid-globale Instanz des GIDS, zusätzlich zu den Site-lokal betriebenen Instanzen, notwendig ist (siehe Anforderungsanalyse in Abschnitt 3.2.2 und 3.2.3 mit den assoziierten Anwendungsfällen des Aktors *Aktor:GIDS:Provider*, Tabelle 3.6 auf Seite 48), um einen kunden- und VO-orientierten Dienst schaffen und im Grid bereitstellen zu können. Dieser Betreiber unterscheidet sich im wesentlichen von den Ressourcenanbietern dadurch, dass er keine eigene Sensorik bestehender Systeme mit einbringen kann, da er in seiner Rolle nicht als Ressourcenanbieter fungiert. Zudem impliziert er keinen zentralisierten Aufbau des GIDS, wie auch aus den folgenden Teilabschnitten hervorgeht. Eine Detaillierung des Aufbaus eines Grid-basierten Frühwarnsystems sowohl auf Seiten eines Ressourcenanbieters als auch auf Seiten des Betreibers des GIDS findet sich in Abschnitt 5.2 und seinen Unterabschnitten.

5.2 Detaillierung des Architekturvorschlags

Nachdem Abschnitt 5.1 die Idee zum Aufbau eines Grid-basierten Intrusion Detection Systems mit einer ersten Grobarchitektur vorgestellt hat, beschreibt dieser Abschnitt detailliert die einzelnen, organisatorisch unabhängigen Architekturteile, die in ihrer Summe das Grid-basierte IDS bilden. Die einzelnen dafür notwendigen Komponenten lassen sich aus den in Kapitel 3 erhobenen Anforderungen an ein GIDS ableiten, was nachfolgend in umgekehrter Reihenfolge anhand der fünf Anforderungskategorien kurz erörtert wird. Eine detaillierte Beschreibung der einzelnen Komponenten und deren Zusammenspiel ist dann in den Unterabschnitten dieses Kapitels zu finden.

Erkennungsleistung. Die Zweiteilung der Anforderungen zur Erkennungsleistung kann sich in ebenfalls zwei Komponenten des Systems widerspiegeln. Örtliche Aspekte können dabei durch die Einführung von *Agent*en, die verschiedene Informationsquellen (Firewalls und ihre Log-Dateien, lokale IDS-Instanzen etc.) anbinden, befriedigt werden. Die Erkennung verschiedener Angriffsmuster und -typen fordert die Notwendigkeit nach austauschbaren Analysefunktionen in einer *lokalen (G)IDS-Instanz* mit angeschlossener *GIDS-Datenbank*, um auch zeitlich lang andauernde Angriffe geeignet erkennen zu können.

Organisatorische und Datenschutzanforderungen. Zur (technischen) Durchsetzung von Informationsverbreitungsrichtlinien und der Gewährleistung von Datenschutzrichtlinien am GIDS partizipierender Ressourcenanbieter wird zum einen ein *Filter* und zum anderen ein *Anonymisierer/Pseudonymisierer* notwendig.

Sicherheitsanforderungen. Zur Realisierung kryptographischer Anforderungen bei der Interkommunikation der beteiligten Parteien bedarf es eines *GIDS-Agent*en, während die Anforderungen an eine Nutzerverwaltung und entsprechende AA-Mechanismen eine Anbindung an die bestehenden VO-Managementsysteme notwendig machen.

Nichtfunktionale Anforderungen. Aus dem umfangreichen Bereich der nichtfunktionalen Anforderungen lassen sich eine Menge Komponenten und Schnittstellen ableiten. Insbesondere folgt aus den Grid-bedingten Anforderungen ein weiteres Mal die Anbindung des GIDS an die bestehenden VO-Managementsysteme sowie auch an Monitoring-Komponenten bzw. deren Teilfunktionalität der bijektiven Abbildung von VOs zu den von ihnen im Grid genutzten Ressourcen. Durch Anforderungen wie Wiederverwendbarkeit, Erweiterbarkeit oder auch Flexibilität lässt sich einmal mehr die bereits zuvor erwähnte Komponente des *Agent*en ableiten. Zusätzlich bedingt die Forderung nach einer großen Leistungsfähigkeit und Skalierbarkeit auch die Möglichkeit der Informationsverdichtung. Daraus folgt direkt die Notwendigkeit einer *Aggregator/Verdichter*-Komponente.

Funktionale Anforderungen. Insbesondere die Einführung eines Kundenbegriffs im Rahmen Grid-basierter IDS motiviert die Existenz eines Betreibers des GIDS so-

wie die dazu gehörigen Komponenten. Natürlich muss auch mindestens eine *globale GIDS-Instanz* mit dazugehöriger *GIDS-Datenbank* und einem *GIDS-Agent*en zur Kommunikation gegeben sein. Zusätzlich bedarf es jedoch eines *Benutzerportal*s und einer Komponente für eine *proaktive Benachrichtigung* und deren Anbindung an VO-Managementsysteme sowie die Notwendigkeit der Abbildung von VOs zu den von ihnen genutzten Ressourcen.

Im weiteren Verlauf dieses Kapitels wird auf die genaue Funktionalität der geforderten Komponenten und deren Zusammenspiel sowie auf erste Hinweise auf eine mögliche technische Umsetzung eingegangen. Abschnitt 5.2.1 geht genauer auf die Architektur auf Seiten eines Ressourcenanbieters ein und Abschnitt 5.2.2 stellt den Aufbau auf Seiten des Betreibers des GIDS dar. Diese beiden organisatorischen Einheiten entsprechen den in Abbildung 5.2 dargestellten Wolken (*Site* und *GIDS-Betreiber*). Abschließend wird die Möglichkeit der Erweiterung des Gesamtsystems um zusätzliche Informationsanbieter in Abschnitt 5.2.4 kurz erläutert.

5.2.1 Architektur auf Seiten eines Ressourcenanbieters

Die grobe Übersicht über ein Grid-weites Frühwarnsystem in Abbildung 5.2 gibt bereits einen gewissen Eindruck über den Aufbau des Systems auf Seiten eines Ressourcenanbieters, in Abbildung 5.2 durch Site bezeichnet. Der genaue Aufbau des Systems innerhalb der administrativen Grenzen eines Ressourcenanbieters ist in Abbildung 5.3 dargestellt.

Die nachfolgend verwendeten Komponenten zum Aufbau des GIDS auf Seiten eines Ressourcenanbieters lassen sich vor allem aus dem in Tabelle 3.20 auf Seite 72 zusammengefassten Kriterienkatalog für IDS im Grid-Umfeld ableiten. So wird der Forderung nach der Möglichkeit der Aggregatbildung auch aus Gründen der Performanz, der Beachtung von Datenschutzaspekten (inkl. der Archivierung von Sensordaten und Berichten) und der Durchsetzung von Informationsverbreitungsrichtlinien jeweils durch die nachfolgend beschriebenen Komponenten der Datenbank, des Filters, des Aggregators bzw. Verdichters sowie des Anonymisierers und Pseudonymisierers Rechnung getragen. Die Art und Weise des Zusammenspiels dieser Komponenten resultiert insbesondere aus Grid-bedingten Anforderungen wie z.B. Dynamikaspekten und Gesichtpunkten der Leistungsfähigkeit und Performanz, während geeignete Implementierungstechniken weitere Anforderungen, wie zum Beispiel kryptographische oder die Erkennungsleistung des IDS betreffende Anforderungen, befriedigen können. Ein genauer Abgleich des erhobenen Anforderungskatalogs mit dem nachfolgend detaillierten Architekturvorschlag ist in Abschnitt 6.3 zu finden.

Um für ein Grid-weites IDS eine vollständige Datenreplikation gewährleisten zu können, kommt bei jedem Ressourcenanbieter eine zentrale Datenbank zum Einsatz. Diese Datenbank wird primär aus drei verschiedenen Informationsquellen gespeist:

Agent. Bei jedem Ressourcenanbieter können mehrere Agenten in verschiedenen Ausprägungen installiert sein und betrieben werden. Agenten dienen dazu, dass Informa-

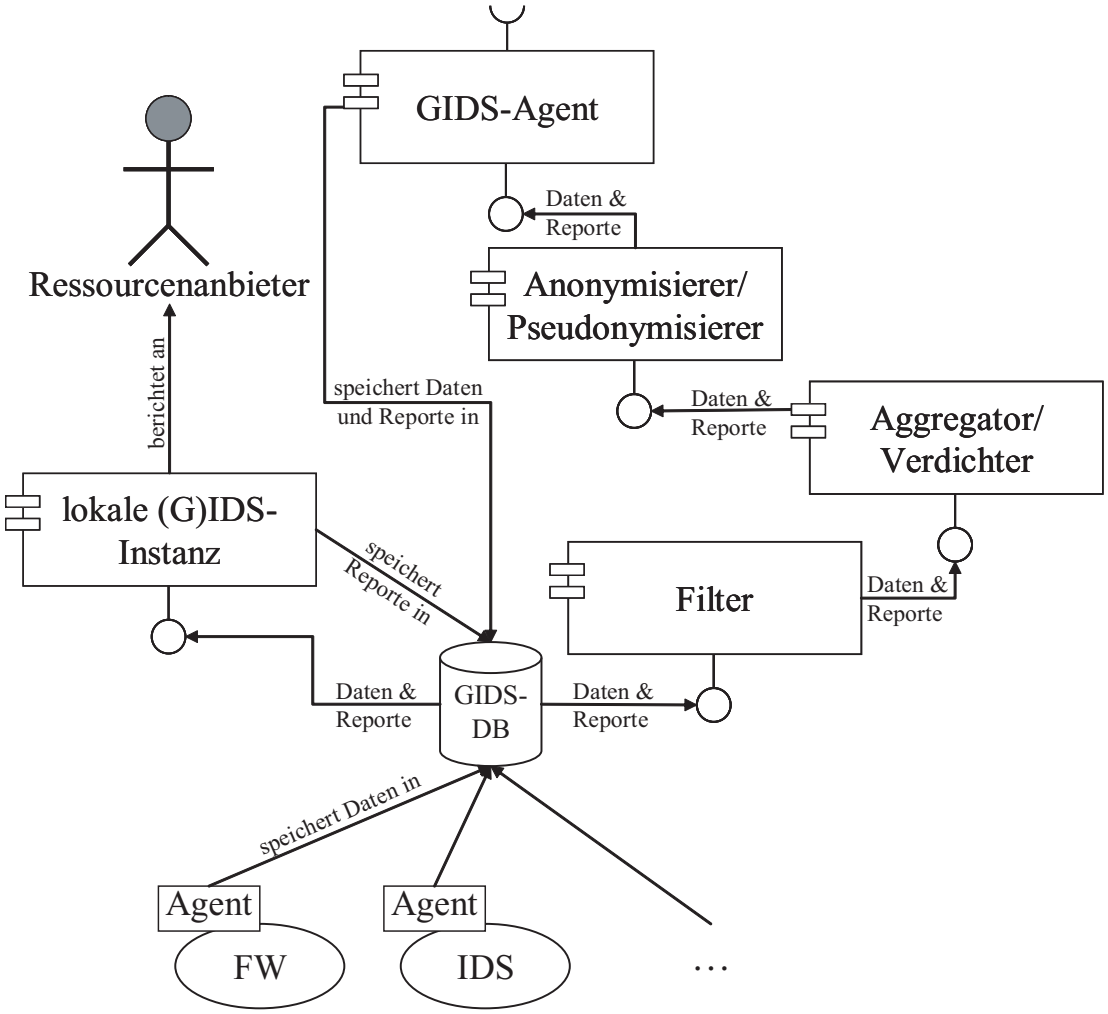


Abbildung 5.3: Architektur auf Seiten einer teilnehmenden Domäne

tionen bestehender Sicherheitsvorkehrungen (z.B. Netflow Traces oder Firewall Logs) in dem zentralen Datenspeicher abgelegt werden können.

GIDS-Agent. In Abgrenzung zu den Agenten ist der GIDS-Agent für die Kommunikation mit den anderen Teilnehmern des GIDS verantwortlich. Zum einen verschickt er ausgewählte Informationen (siehe hierzu die Vorverarbeitungsschritte weiter unten) an andere am GIDS teilnehmende GIDS-Agenten, zum anderen empfängt er eben solche Daten von anderen GIDS-Agenten und hinterlegt sie ebenfalls in der zentralen Datenbank.

Lokale (G)IDS-Instanz. Dadurch, dass ein (lokaler) Datenbestand sämtlicher im GIDS „öffentlich“ verfügbarer Informationen vorliegt, besteht die Möglichkeit bei jedem Ressourcenanbieter eine eigene Instanz des GIDS zu betreiben. Dadurch bedingt, dass der Site-spezifische Datenbestand unter anderem auch nicht im GIDS veröffentlichte In-

formationen enthalten kann, eignet sich diese Instanz des GIDS ebenfalls als mögliche Instanz eines lokalen, Site-spezifischen IDS, was die Schreibweise der lokalen (G)IDS-Instanz in Abbildung 5.3 begründet. Berichte dieses (G)IDS werden ebenfalls in der lokalen Datenbank abgelegt.

Bevor es zur Veröffentlichung jedweder Information im Grid durch den GIDS-Agenten kommt, durchlaufen sämtliche Informationen noch drei Vorverarbeitungsschritte.

Filter. Neue Datensätze, die in die zentrale Datenbank geschrieben werden, werden an einen Filter weitergereicht. Die primäre Aufgabe des Filters ist nun das Durchsetzen der Site-spezifischen Informationsverbreitungsrichtlinien. In Abgrenzung zu Datenschutzbestimmungen sind Informationsverbreitungsrichtlinien zumeist Bestandteil lokaler Sicherheitsrichtlinien, die zum Beispiel die Vermeidung der Verbreitung interner Topologiemerkmale, Sicherheitsverletzungen etc. fordern. Der Filter kann einen eingehenden Datensatz auf Grund bestimmter Auswahlkriterien verwerfen oder auch passieren lassen.

Eine weitere wichtige Aufgabe des Filters ist es Datensätze, die von einem GIDS-Agenten in die Datenbank geschrieben wurden, auszufiltern. Sollte dies nicht passieren, so kann es zu Duplikaten in den Datenbeständen und somit zwangsläufig zu Endlosschleifen der Nachrichten kommen, wodurch in kürzester Zeit eine Überlastsituation (Netzkapazitäten, Speicherkapazität der Datenbanken, Informationsflut für analysierenden IDS-Instanzen etc.) im GIDS zustande kommen würde.

Aggregator/Verdichter. Diejenigen Datensätze, die nicht zuvor durch den Filter aus dem Informationsstrom entfernt worden sind, können in dieser Komponente aggregiert oder verdichtet werden. Eine Aggregation (auch Konsolidierung oder Verdichtung) bezeichnet das Zusammenfassen vieler Daten mit wenig Informationen zu wenigen Daten mit entsprechend hohem Informationsgehalt. Für eine Aggregation wird eine Aggregationsfunktionen benötigt, die zum Beispiel im Falle einer Menge von Zahlen der Mittelwert, das Minimum, das Maximum oder die Summe sein können. Durch den Schritt der Aggregation kann also eine Datenverdichtung erfolgen und somit das Aufkommen an Informationen nochmals deutlich gesenkt werden.

Anonymisierer/Pseudonymisierer. Bevor die (aggregierten) Datensätze die administrativen Grenzen eines GIDS-Teilnehmers verlassen, müssen neben den Informationsverbreitungsrichtlinien, die durch die Filterkomponente gewahrt worden sind, auch Datenschutzbestimmungen eingehalten werden. Insbesondere rechtliche Randbedingungen zwingen einen Ressourcenanbieter unter anderem dazu keine personenbezogenen Daten nach außen zu tragen, was durch den Vorgang der Anonymisierung oder einer Pseudonymisierung gewährleistet wird.

Die Reihenfolge, in der alle Informationen aus der Datenbank die drei zuvorstehenden Komponenten durchlaufen, ist prinzipiell für die Funktionalität nicht entscheidend. Bei

der Anordnung dieser Komponenten wird aus Effizienzgründen an erster Stelle eine Filterung (also Löschung „unerwünschter“ Datensätze), dann eine Informationsverdichtung (also eine nochmalige Datenreduktion) und erst abschließend eine Anonymisierung bzw. Pseudonymisierung vorgenommen.

Alle notwendigen Komponenten zum Aufbau des Grid-basierten Intrusion Detection Systems werden in den nachfolgenden Unterabschnitten genauer beleuchtet sowie mögliche Techniken zu deren Umsetzung kurz angesprochen.

5.2.1.1 Agenten und zentraler Datenspeicher

Agenten dienen in erster Linie dazu Informationen, die für ein GIDS von Interesse sind, semantisch und syntaktisch anzupassen und in einem zentralen Datenspeicher (z.B. eine Datenbank) abzuspeichern. Diese Aufgabe wird typischer Weise unter Nutzung des *Adapter Pattern* (auch bekannt als *Wrapper Pattern*) [Gamma u. a., 1994] realisiert.

Bereits an dieser Stelle wird klar, dass zwingend ein einheitliches Daten- und Informationsmodell für das gesamte GIDS notwendig ist. Da prinzipiell Agenten für eine beliebige Datenquelle implementiert werden können, muss insbesondere eine Möglichkeit der Erweiterbarkeit des Modells gegeben sein. Ein mögliches Datenmodell wird im RFC 4765 mit dem *Intrusion Detection Message Exchange Format (IDMEF)* [RFC4765, 2007] spezifiziert. IDMEF hat sich bereits während einer ungewöhnlich langen Phase des Daseins als Internet Draft recht weit verbreitet. Es handelt sich bei diesem Nachrichtenformat um ein XML-Schema, das alle zuvor erwähnten Eigenschaften realisiert.

Für einen Domänen-zentralen Datenspeicher bietet sich ein Datenbanksystem an. Dieses arbeitet hoch optimiert und effizient und ist dadurch in der Lage das vergleichsweise hohe Datenvolumen, das an einer Domäne zu erwarten ist, zu verarbeiten. Ein weiterer Vorteil einer Datenbank ist, dass alle angefallenen Daten hierin sinnvoll archiviert werden können und zum Zwecke des Reporting und der Forensik auch zu einem späteren Zeitpunkt noch Abfragen getätigt werden können. Hierzu ist natürlich insbesondere ein Zeitstempel innerhalb eines jeden Datensatzes notwendig, um Abfragen sinnvoll gestalten zu können. Auch für eine Archivierung der Daten ist eine Zeitangabe notwendig.

Es bleibt für jede Site spezifisch einen Prozess zu spezifizieren, der entscheidet, welche Datensätze über welchen Zeitraum aufbewahrt werden dürfen und sollen. Außerdem ist es denkbar ältere Daten in einer höher aggregierten, durchaus mit Informationsverlust behafteten, Version zu speichern, bevor es zur endgültigen Löschung kommt. Eine solche Prozessspezifikation ist jedoch nicht weiter Bestandteil dieser Arbeit.

5.2.1.2 Filter

Jeder neu in die Datenbank eingefügte Datensatz muss zeitnah im GIDS veröffentlicht werden. Dazu müssen alle neu in die Datenbank eingefügten Datensätze an den Filter weitergegeben werden, der die erste Komponente in den Verarbeitungsschritten bis hin zur endgültigen Veröffentlichung der Information ist. Optimal ist hierzu ein Push-Verfahren

zur Kommunikation, bei dem die Datenbank nach erfolgreichem Einfügen eines Datensatzes diesen direkt an den Filter weitergibt. Sollte dies im Zuge der Implementierung des Systems nicht möglich sein, so kann alternativ ein Pull-Verfahren seitens des Filters realisiert werden. In diesem Fall muss der Filter in regelmäßigen Abständen eine Anfrage an die Datenbank nach neu eingetroffenen Datensätzen richten. Die Intervalle, in denen eine erneute Abfrage stattfindet, müssen individuell an das Szenario angepasst werden, sollten aber in jedem Fall im Bereich von Sekunden liegen. Für eine solche Pull-Kommunikation ist natürlich ein Zeitstempel der Einfügung eines neuen Datensatzes in die Datenbank zwingend notwendig, um eine intervallbasierte Abfrage realisieren zu können.

Wie bereits einleitend des Abschnitts 5.2.1 erwähnt, ist es eine der Hauptaufgaben des Filters keinen Datensatz, der von einem GIDS-Agenten in die Datenbank geschrieben wurde, wieder zur Neuveröffentlichung freizugeben. Dies ist notwendig um Endlosschleifen von Datensätzen im GIDS zu unterbinden. Egal ob eine Push- oder Pull-Kommunikation zwischen der Datenbank und dem Filter etabliert wird, um Datensätze, deren Ursprung ein GIDS-Agent ist, filtern zu können, bedarf es eines entsprechenden Datenfeldes. Eine nicht notwendige, aber dennoch sinnvolle Einrichtung ist hierzu ein Datenfeld, das die Herkunft einer Nachricht vermerkt. Die Herkunftsangabe kann natürlich in einer anonymisierten oder pseudonymisierten Form vorliegen, es reicht prinzipiell jedoch eine binäre Markierung, ob ein Datensatz von einem GIDS-Agenten geschrieben wurde oder nicht.

Eine weitere Aufgabe des Filters ist die (technische) Durchsetzung der Informationsverbreitungsrichtlinien, die spezifisch je Teilnehmer geregelt sind. Im Allgemeinen bietet sich hier die Nutzung regulärer Ausdrücke an. Im Falle der Nutzung eines XML-basierten Datenaustauschformates kann alternativ auch die *Extensible Stylesheet Language* (XSL) [XSL, 2006] und eine *XSL Transformation* (XSLT) zum Einsatz kommen. Hierfür stehen eine Reihe an Implementierungen bereits zur Verfügung

5.2.1.3 Aggregator/Verdichter

Nachdem eine Vorauswahl potentiell im GIDS zu veröffentlichender Informationen durch die vorgeschaltete Komponente des Filters getroffen worden ist, übernimmt der Aggregator oder auch Verdichter eine weitere Datenreduktion. Eine solche Reduktion ist vom Konzept her zwischen einer verlustfreien und einer verlustbehaftete Datenverdichtung zu unterscheiden. Ein Beispiel für eine verlustfreie Verdichtung ist, wenn es das Ziel ist über ein zuvor bekanntes Zeitintervall die Anzahl an übertragenen Paketen zu ermitteln. In diesem Fall könnte der Aggregator als Aggregationsfunktion eine Summierung aller übertragenen Pakete über eben dieses bekannte Zeitintervall vornehmen, was keinen Informationsverlust für dieses spezielle Auswertungsverfahren bedeuten würde. Einen Informationsverlust im Kontext dieses Beispiels müsste man hingegen bei der Aggregatsbildung über z.B. das doppelte Zeitintervall hinnehmen.

Bereits dieses triviale Beispiel verdeutlicht, dass der Verdichter in jedem Fall einen Puffermechanismus für eintreffende Daten haben muss. Er erhält seine Eingabedaten asynchron per Push-Kommunikation von der Komponente des Filters, aggregiert die einge-

henden Daten entsprechend seiner Spezifikationen und leitet die verdichteten Datensätze per Push-Kommunikation weiter an den Anonymisierer und Pseudonymisierer. Zweitere Kommunikationsbeziehung kann sowohl zeitgesteuert synchron als auch asynchron, durch bestimmte Ereignisse gesteuert, erfolgen. Z.B. ist es denkbar, dass alle x Minuten in jedem Fall ein Informationsaggregat ausgegeben wird, im Falle einer eintreffenden Alarmmeldung der lokalen (G)IDS-Instanz diese jedoch sofort weitergeleitet wird.

5.2.1.4 Anonymisierer und Pseudonymisierer

Insbesondere zur Wahrung von Datenschutzrichtlinien ist eine Anonymisierung und/oder Pseudonymisierung von Informationen vor ihrer Veröffentlichung in einem Grid-weit föderierten System zwingend notwendig. Sowohl die Anonymisierung als auch die Pseudonymisierung sind Maßnahmen zur Wahrung des Datenschutzes. Mit der *Anonymisierung* bezeichnet man das Verändern personenbezogener Daten, so dass diese Daten im Nachhinein nicht mehr eindeutig einer Person zugeordnet werden können. In Abgrenzung zur Anonymisierung bezeichnet man mit dem Vorgang der *Pseudonymisierung* eines Datensatzes das Ersetzen der Identifikationsmerkmale, die eine Zuordnung der Daten zu einer Person ermöglichen, durch ein Pseudonym. Die Bildung der Pseudonyme geschieht hierbei durch eine Abbildung (in der Regel bijektiv), so dass unter Kenntnis der Abbildungsfunktion, die auch als *Code* bezeichnet wird, eine Zuordnung eines Datensatzes zu einer Person eindeutig möglich ist. Im Gegensatz zur Anonymisierung bleiben bei der Pseudonymisierung Zusammenhänge unterhalb der pseudonymisierten Datensätze erhalten, unter der Voraussetzung, dass alle Datensätze durch dieselbe Abbildung pseudonymisiert worden sind.

Die Komponente des Anonymisierers und Pseudonymisierers bezieht als Eingabe seine Informationen asynchron, per Push-Kommunikation vom Aggregator. Als Ausgabe liefert er anonymisierte und/oder pseudonymisierte Datensätze im selben Datenformat wie die zuvor erhaltenen Eingabedaten, jedoch nur mit um den zu Datenschutzrichtlinien konform bereinigten Informationen. Es ist zweckmäßig sich Grid-weit auf eine einheitliche Anonymisierungs- oder Pseudonymisierungsfunktion festzulegen, um eine höhere Qualität der Angriffserkennung gewährleisten zu können, insbesondere im Falle der Pseudonymisierung (s.o.).

Für eine Implementierung kann eventuell eine Anlehnung an das *Secure Audit Logging for Linux (SAL)* Projekt [Secure Audit Logging, 2003], *SNARE (System iNtrusion Analysis & Reporting Environment)* [SNARE] oder auch *The Linux Basic Security Module Project (Linux BSM)* [Banford, 2000] in Betracht gezogen werden. Alle diese Projekte behaupten von sich konform zu den *Common Criteria for Information Technology Security Evaluation* (auch abgekürzt als *Common Criteria* oder *CC*) zu sein, die in dem internationalen Standard ISO/IEC 15408 [ISO/IEC 15408-1, 2005; ISO/IEC 15408-2, 2005; ISO/IEC 15408-3, 2005] niedergeschrieben sind und ein Äquivalent zu dem *U.S. Government's C2 standards for security* [DoD 5200.28-STD, 1985] (auch bekannt als „*The Orange Book*“) bilden. Im Falle der Nutzung eines XML-basierten Datenaustauschformates kann auch hier wie be-

reits beim Filter erwähnt die *Extensible Stylesheet Language* (XSL) [XSL, 2006] und eine *XSL Transformation* (XSLT) zum Einsatz kommen.

5.2.1.5 GIDS-Agent

Die GIDS-Agenten sind die Komponenten, die die Kommunikation, also den Informationsaustausch, unterhalb aller teilnehmenden Sites realisieren. Zu den Hauptaufgaben eines GIDS-Agenten gehört unter anderem die Authentifizierung und Autorisierung der anderen GIDS-Agenten sowie die Gewährleistung der Nachrichtenintegrität, Nachrichtenvertraulichkeit und Authentizität.

Für die Umsetzung der GIDS-Agenten bietet sich die Verwendung des *Observer Pattern* (auch bekannt als *Publish-Subscribe Pattern* oder *Dependents Pattern*) [Gamma u. a., 1994] an. Das Observer Pattern definiert dabei eine $1 : n$-Beziehung zwischen den GIDS-Agenten. Es ist dazu intendiert, dass, wenn sich der Zustand eines der Objekte ändert, alle anderen Objekte automatisch über diese Änderung informiert werden. Im Fall der GIDS-Agenten bedeutet dies, dass, wenn einer der GIDS-Agenten einen Datensatz zur Informationsverbreitung erhält (dies entspricht also der Zustandsänderung), so informiert er alle weiteren GIDS-Agenten über den Eingang dieses Datensatzes. Einen möglichen Ansatz zur Einführung des Publisher-Subscriber Design Pattern in Infrastrukturen verteilter IDS beschreiben Basicevic et. al. in [Basicevic u. a., 2007].

Zur Kommunikation der GIDS-Agenten untereinander bietet sich unter anderem das *The Intrusion Detection Exchange Protocol* (IDXP) [RFC4767, 2007] an. Unter Nutzung des *Blocks Extensible Exchange Protocol* (BEEP) [RFC3080, 2001], das auf TCP als unterliegendes Protokoll abgebildet werden kann [RFC3081, 2001] und durch *Transport Layer Security (TLS)* [RFC4346, 2006] gesichert wird, kann IDXP die Vertraulichkeit, Integrität und Authentizität von Nachrichten gewährleisten. Weiter entstehen in der Regel keine Komplikationen mit Firewalls durch das Tunneln über TCP bzw. beschränken sich ggf. Änderungen an Firewall-Regeln auf die Freigabe eines TCP-Ports für den GIDS-Agenten. Unter Umständen ist BEEP sogar auch in Kombination mit Network Address Translation (NAT) einsetzbar. Zusätzlich kann BEEP die Duplikation von Nachrichten bei der Kommunikation von GIDS-Agenten untereinander verhindern.

Um die Authentifizierung und Autorisierung weiterer GIDS-Agenten durchführen zu können, kann auf eine im Grid bereits bestehende Public Key Infrastruktur zurückgegriffen werden (siehe hierzu auch Abschnitt 2.2.1). Durch eine Anbindung an eine bestehende PKI stehen in den meisten Fällen vertrauenswürdig signierte X.509 Zertifikate [RFC3280, 2002] bereits zur Verfügung

5.2.1.6 Lokale (G)IDS-Instanz

Die eigentlich Logik des IDS liegt in den angeschlossenen (G)IDS-Instanzen bzw. ihrem Pendant den globalen GIDS-Instanzen (siehe hierzu Abschnitt 5.2.2.1). Diese implementieren eine Analysefunktionalität und sind somit neben der Platzierung der IDS-Sensoren

zentraler Bestandteil bei der Erkennung von Angriffen. Kritische Eigenschaften des lokalen (G)IDS-Instanz sind zum einen Performanz (wie viele eingehende Meldungen kann ich pro Zeiteinheit verarbeiten?), zum anderen die Erkennungsleistung (wie hoch sind die Raten der False Positives bzw. False Negatives?). Diese Eigenschaften werden essentiell sowohl vom Konzept der Analyseeinheit als auch ihrer Implementierung beeinflusst.

Für eine lokale (G)IDS-Instanz kommen prinzipiell alle Arten der Datenanalyse in Betracht (für eine grundlegende Beschreibung der verschiedenen Erkennungsverfahren siehe auch Abschnitt 2.1): Es können sowohl eine Anomalieerkennung wie auch eine signaturbasierte Angriffserkennung, ein hybrides Verfahren oder sogar mehrere Verfahren parallel zum Einsatz kommen.

Für eine konkrete Implementierung einer lokalen (G)IDS-Instanz bietet es sich an, auf bereits existierende Mechanismen zurückzugreifen und diese an die Grid-Umgebung anzupassen. So kann zum Beispiel für ein signaturbasiertes Erkennungsverfahren wie *Snort* [Snort] ein entsprechender Regelsatz neben dem klassischen Regelwerk entworfen und implementiert werden. Für eine Anomalieerkennung gilt es die für das Verfahren notwendigen Parameter an die neue Umgebung anzupassen.

5.2.2 Architektur auf Seiten des Betreibers des GIDS

Die Architektur auf Seiten des Betreibers des GIDS steht in Anlehnung an den Aufbau des Systems auf Seiten eines Ressourcenanbieters. In Abgrenzung zu einem Ressourcenanbieter stellt der Betreiber des GIDS in der Regel jedoch keine Rohdaten für eine gemeinschaftliche Angriffserkennung zur Verfügung, da er nicht über entsprechende Datenquellen verfügt. Vielmehr stellt er einen Grid-Dienst zur Verfügung, der eine Berichterstattung und Darstellung der im GIDS verfügbaren Informationen und Berichte für die unterschiedlichen Nutzergruppen (Ressourcenanbieter, VOs etc.) anbietet. Entsprechend ist die Architektur auf Seiten des Betreibers des GIDS um die Komponenten zur Datenakquise (die Agenten) und die Komponenten zur datenschutzkonformen Informationsaufarbeitung bereinigt. Zur Dienstbereitstellung jedoch werden Anbindungen an Grid-typische Dienste (z.B. VO-Managementsysteme) und eine Erweiterung um ein dem Grid konformes Dienstportal vorgenommen. Abbildung 5.4 stellt den genauen Aufbau des GIDS auf Seiten seines Betreibers graphisch dar.

GIDS-Agent und Datenspeicher. Sowohl aus Architektur- als auch aus Implementierungssicht ist der GIDS-Agent und der Datenspeicher identisch mit den vergleichbaren Komponenten auf Seiten eines Ressourcenanbieters. Im Falle des Betreibers des GIDS ist es die Hauptaufgabe des GIDS-Agenten den Datenspeicher mit Informationen, die von den anderen Partnern zur Analyse publiziert werden, zu füllen. Zusätzlich versendet er Grid-global erkannte Angriffe, die durch die entsprechende GIDS-Instanz erkannt wurden. In der Regel stellt der GIDS-Agent die einzige Datenquelle für den Betreiber des GIDS dar, eine mögliche Ausnahme ist in Abschnitt 5.2.4 aufgezeigt.

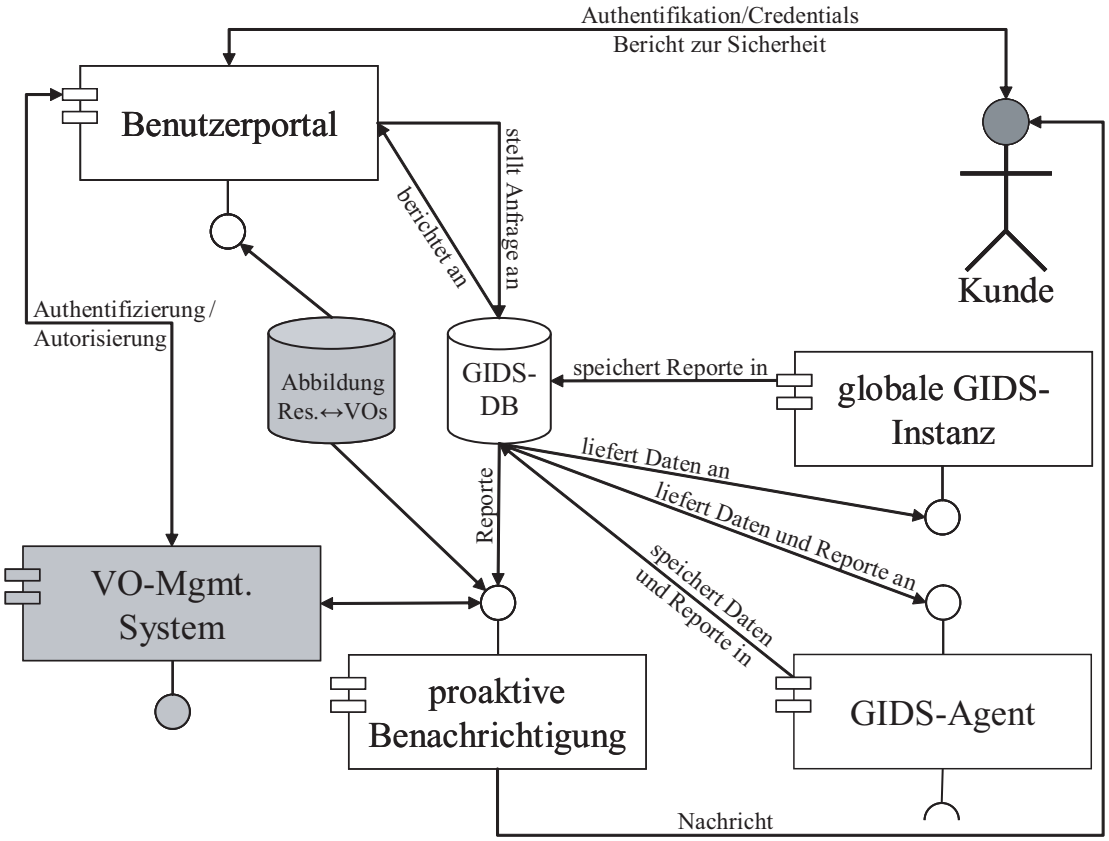


Abbildung 5.4**:** Architektur auf Seiten des Betreibers des GIDS

Durch die Gleichheit mit dem GIDS-Agenten und Datenspeicher auf Seiten eines Ressourcenanbieters bedingt wird nachfolgend nicht näher auf diese Komponenten eingegangen. Für eine genauere Beschreibung sei auf den Abschnitt 5.2.1 und darin insbesondere das Unterkapitel 5.2.1.5 verwiesen.

Grid-globale GIDS-Instanz. Der Grid-globalen Instanz zur Auswertung der im Grid verfügbaren Informationen stehen in der Regel die wenigsten Originalinformationen zur Analyse zur Verfügung. Dies liegt daran, dass in der zugehörigen Informationsbasis ausschließlich die im Grid „öffentlich“ verfügbaren Daten, die von den beteiligten GIDS-Agenten publiziert worden sind, verfügbar sind. Diese Informationen sind bereits von den jeweiligen Ressourcenanbietern gefiltert, aggregiert und verdichtet sowie anonymisiert und/oder pseudonymisiert worden (siehe auch Abschnitt 5.2.1). Daraus folgt, dass dieser Instanz des IDS nur eine Teilmenge der Informationen vorliegt, die paarweise jedem einzelnen Ressourcenanbieter zur Verfügung steht. Es ist also zu erwarten, dass die Grid-globale GIDS-Instanz in Bezug auf ihre Erkennungsleistung gegenüber den jeweils lokalen (G)IDS-Instanzen schlechter abschneidet. Aus diesem Grund erscheint es sinnvoll, dass auch durch (G)IDS-Instanzen erkannte Angriffe durch die Ressourcenanbieter Grid-global publiziert werden, wie auch konzep-

tuell in Abschnitt 5.2.1 vorgesehen.

Benutzerportal. Auf Basis der in der Datenbank hinterlegten Berichte stellt das Benutzerportal eine mandantenfähige Nutzeroberfläche zur Berichterstattung bereit. Nutzer (Mitglieder einer VO oder auch Ressourcenanbieter) können hier den aktuellen Sicherheitsstatus des Grids unter Nutzung ihrer im Grid gültigen Credentials einsehen sowie historische Berichte anfragen. Es werden verschiedene Sichten auf die Berichte je nach Rolle des Nutzers angeboten.

Da das Benutzerportal auf der einen Seite nur Abhängigkeiten von bestehenden Grid-Diensten und auf der anderen Seite von einer GIDS-Datenbank hat, kann es auch zum Betrieb bei einem beliebigen anderen Teilnehmer des GIDS verwendet werden. Dadurch kann eine redundante Auslegung bzw. eine Wiederverwendung dieser Komponente als Managementoberfläche bei den Ressourcenanbietern ermöglicht werden.

Proaktive Benachrichtigung. Diese Komponente dient dazu die Kunden des GIDS, also sowohl Mitglieder einer VO als auch Ressourcenanbieter, über die aktuelle Sicherheitslage stets proaktiv, d.h. sofort nach erkanntem Angriff, in Kenntnis zu setzen. Dies kann auf verschiedenen Kommunikationswegen wie z.B. Email oder SMS geschehen. Eine Auswahl des Kommunikationskanals kann in Abhängigkeit der Wichtigkeit einer Nachricht erfolgen.

Genau wie für das Nutzerportal auch, bestehen für die Komponente der proaktiven Benachrichtigung nur Abhängigkeiten zu bestehenden Grid-Diensten und einer GIDS-Datenbank. In hiesigem Fall ist die Anbindung an eine Datenbank zur Abbildung von Ressourcen auf die sie nutzenden VOs sowie an das VO-Managementsystem zum Bezug der Kontaktdaten der jeweils verantwortlichen Personen notwendig. Somit kann auch diese Komponente bei einem beliebigen anderen Teilnehmer des GIDS betrieben werden.

5.2.2.1 Grid-globale IDS-Instanz

Die beim Betreiber des GIDS zum Einsatz kommende Grid-globale IDS-Instanz ist in erster Linie identisch mit einer jeden lokalen (G)IDS-Instanz (vgl. Abbildung 5.3 und Abschnitt 5.2.1.6). Ein entscheidender Unterschied zwischen den beiden Instanziierungen dieser Komponente ist die Datenbasis, auf der sie arbeiten. Während einer lokalen (G)IDS-Instanz alle im Grid anonymisierten und/oder pseudonymisierten Daten zusätzlich zu den unveränderten Rohdaten der eigenen administrativen Domäne zur Auswertung zur Verfügung stehen, kann die Grid-globale IDS-Instanz ausschließlich auf den anonymisierten und/oder pseudonymisierten Datenbestand der an GIDS beteiligten Partner zurückgreifen. Diese Tatsache lässt eine geringere Erkennungsleistung auf Grund von Informationseinbußen an dieser Stelle erwarten. Zudem kann eine individuelle Anpassung der zum Betrieb der Grid-globalen IDS-Instanz notwendigen Regelwerke bzw. Parameter in Abhängigkeit von der Qualität des Informationsbestandes erforderlich sein.

5.2.2.2 Benutzerportal

Das Benutzerportal stellt eine zentrale Anlaufstelle für die Kunden des GIDS bereit, in dem kundenspezifische Sichten auf die verfügbaren Reporte realisiert werden. Zur Nutzerauthentifizierung ist eine Anbindung an bestehende AA-Infrastrukturen bzw. VO-Managementsysteme notwendig, die in Abbildung 5.4 grau als bereits existierende Systeme im Grid dargestellt sind. Eine Einschränkung des Nutzerkreises auf Grid-Nutzer ist insofern sinnvoll, als dass einem externen Angreifer kein Vorteil durch die Einsicht der verfügbaren GIDS-Berichte entstehen soll und er so evtl. Rückschlüsse auf bestehende Sicherheitslücken, Schwachstellen o.ä. schließen könnte.

Alle in diesem Portal verfügbaren Informationen sind durch ihre zuvor datenschutzkonforme Aufarbeitung prinzipiell für alle Anwender im Grid einsehbar. Es wird jedoch zusätzlich eine Sicht auf die Berichte angeboten, die nur die eigenen Ressourcen bzw. die von einer VO verwendeten Ressourcen umfasst. Dazu ist eine Abbildung von Ressourcen zu VOs notwendig. Eine solche Datenbank kann z.B. Bestandteil eines Grid-Monitoring-Systems sein wie u.a. im Falle des D-Grid. In jedem Fall kann hier wie bei der Nutzerauthentifikation auf einen bestehenden Dienst im Grid zurückgegriffen werden, weswegen die entsprechende Datenbank in Abbildung 5.4 ebenfalls grau schattiert dargestellt ist.

Zur Implementierung des Benutzerportals eignet sich beispielsweise eine Orientierung an dem sogenannten *Bridge Pattern* nach [Gamma u. a., 1994]. Hierdurch kann erreicht werden, dass die Abstraktion des Benutzerportals von seiner Implementierung vollkommen losgelöst ist. Im Gegensatz zu regulären Vererbungen wird durch den Einsatz des Bridge Pattern eine harte Bindung von Abstraktion und Implementierung vermieden, so dass sowohl die Abstraktion, als auch die Implementierung durch Unterklassen erweiterbar bleiben, während dies keinen Einfluss auf nachgelagerte Anwendungen hat.

5.2.2.3 Proaktive Benachrichtigung

Für eine proaktive Benachrichtigung der Kunden des Grid-basierten IDS ist eine mandantenfähige Komponente notwendig, mit deren Hilfe Benachrichtigungen über einen erkannten Sicherheitsvorfall abhängig von seinem Schweregrad (engl. *severity*) über verschiedene Kommunikationswege, wie beispielsweise Email oder SMS, versendet werden. Ein jeder Nutzer hat die Möglichkeit sich die Art der Benachrichtigung an seine eigenen Bedürfnisse anzupassen, wobei ein Ausbleiben einer Benachrichtigung auch eine zulässige Aktion ist. Zur Nutzerauthentifizierung und -autorisierung ist hierzu eine Anbindung an die AA-Infrastruktur als Teil des VO-Management (siehe grau schattierte Komponente in Abbildung 5.4) notwendig. Eine weitere Anbindung an eine Datenbank, die die im Grid verfügbaren Ressourcen auf die sie nutzenden VOs und umgekehrt abbildet, ist notwendig, um Benachrichtigungen selektiv an die sie jeweils betreffenden VO-Verantwortlichen zu übermitteln.

Für eine Implementierung dieser Komponente bietet sich das *Observer Pattern* (auch als *Dependents* oder *Publish-Subscribe Pattern* bekannt) nach [Gamma u. a., 1994] an. Die-

ses definiert eine „1-zu-viele“ (engl. *one-to-many*) Beziehung zwischen Objekten. Im Falle einer Statusänderung eines Objektes werden die anderen am System beteiligten Objekte über eben diese Änderung in Kenntnis gesetzt. Dieses Prinzip lässt sich somit einsetzen um die tatsächlich benachrichtigenden Teilkomponenten (z.B. ein Email- oder SMS-Benachrichtigungsmodul) über neu eingehende Sicherheitsvorfälle in Kenntnis zu setzen.

Um eine proaktive Benachrichtigungskomponente zu realisieren, bietet sich unter anderem der Einsatz bzw. die Modifikation bereits existierender Monitoring-Programme an. Es ist zum Beispiel denkbar, das Überwachungswerkzeug *Nagios* [Nagios] für solche Zwecke zu modifizieren. Nagios arbeitet mit Plug-Ins, die bei dem Eintritt eines Ereignisses (z.B. die Überschreitung eines Schwellwertes) eine Meldung generieren. Der Konfiguration von Nagios entsprechend wird auf Grund dieses Ereignisses ggf. eine Benachrichtigung eines zuvor konfigurierten Nutzerkreises über einen zuvor ebenfalls festgelegten Verbreitungsweg vorgenommen. Zusätzlich bietet Nagios eine Benutzeroberfläche, in der der aktuelle Status der überwachten Ressourcen grafisch dargestellt wird. Natürlich sind für solche Zwecke auch andere Monitoring-Lösungen wie beispielsweise die quelloffenen Produkte/Projekte *Cacti* (`http://www.cacti.net/`) oder auch *Zabbix* (`http://www.zabbix.com/`) prinzipiell denkbar. Bei kommerziellen Produkten wie z.B. *IBM Tivoli* oder *HP OpenView* bleibt die Einsatzmöglichkeit auf Grund ihrer nur bedingt anpassbaren Funktionsweise zu prüfen.

5.2.3 Kundenbegriff und Unterstützung Virtueller Organisationen

Neben der Tatsache, dass zurzeit kein Grid-spezifisches IDS für Produktivumgebungen verfügbar ist, hier jedoch konzipiert wird, ist ein weiteres Alleinstellungsmerkmal dieser Arbeit die eingehende Behandlung Virtueller Organisationen als Kunden eines GIDS. Damit verbunden etabliert sich ein neuartiger Kundenbegriff und die Möglichkeit einen neuen Dienst im Grid anzubieten.

Diese Arbeit legt den Grundstein für die Implementierung eines Benutzerportals als Grid-Dienst, mit dem Kunden bzw. VOs, die nicht zwingend aktiv am GIDS partizipieren müssen, die Möglichkeit der Berichterstattung zur Sicherheit als Dienstleistung im Grid nutzen. Dazu wird durch die Anbindung des Benutzerportals an die entsprechenden Dienste des VO-Managementsystems auf der einen Seite die Verwaltung der Nutzer und Rechte delegiert, zum anderen kann eine kundenspezifische Sichtenbildung (mehr hierzu siehe auch Abschnitt 6.1.2) und damit eine kundenspezifische Berichterstattung realisiert werden.

Diese Herangehensweise birgt den zusätzlichen Vorteil, dass ein neues Geschäftsfeld eröffnet werden kann. Auch wenn diese Arbeit nicht im Rahmen eines betriebswirtschaftlichen Kontextes verfasst ist, so wird nachfolgend kurz eine grundlegende Skizze für eine Idee eines Geschäftsmodells angeführt.

Marktanalyse. Zurzeit besteht noch kein Markt, der auf IDS für Grids abzielt. Insofern eröffnet das Vorhaben, ein IDS für Grids zu etablieren, ein neues Marktsegment. Als potentieller Kundenkreis sind in erster Linie Ressourcenanbieter sowie die Communities und VOs des Grids in Betracht zu ziehen. Anbieterseitig ist dieses Segment für die

Hersteller von „klassischen“ IDS Systemen und Hersteller entsprechende Appliances als neues Geschäftsfeld von großem Interesse.

Wertschöpfungsprozess und Ertragsmodell. Um Erträge durch die Erbringung des Spezialdienstes „IDS für das Grid“ zu erwirtschaften, ist es z.B. möglich Gelder von den Kunden für die Verwendung des Dienstes zu verlangen, die Kunden können somit die Erkennung und das Reporting von Angriffen im Grid auf die Betreiber des GIDS auslagern. Bisher ist ein solcher Dienst nicht existent und bietet einen erheblichen Mehrwert für die Kunden.

Risikoanalyse. Die Hauptrisiken beim wirtschaftlichen Einsatz eines Intrusion Detection Systems im Grid sind insbesondere Konkurrenten und eine daraus resultierende Sättigung des Marktes, was im Falle der heutigen Grid-Infrastrukturen noch auf Grund mangelnder Konkurrenz ausgeschlossen werden kann. Weiter besteht die Gefahr der mangelnden Nachfrage sowie von juristischen Folgen bei mangelhafter Diensterbringung.

Ausblick. Das hier entwickelte GIDS kann insbesondere als Basis für Reporting-Mechanismen in Bezug auf die Sicherheit angebotener Dienste und Ressourcen herangezogen werden. Damit kann u.a. auch der Erfüllungsnachweis zuvor geschlossener SLAs bzw. deren Anforderungen an die Sicherheit überprüft werden.

Wie die Nutzung des GIDS-Dienstes durch eine VO und die Realisierung der daraus resultierenden Anforderungen durchgesetzt werden können, zeigt die nachfolgende exemplarische Umsetzung des Anwendungsfalls *Zugriff einer VO als Nutzer eines GIDS*.

Exemplarische Umsetzung zum Anwendungsfall „*Zugriff einer VO als Nutzer eines GIDS*“

Der Anwendungsfall *Zugriff einer VO als Nutzer eines GIDS* (siehe Abschnitt 3.2.2.2 ab Seite 49 und Tabelle 3.9 auf Seite 50) beschreibt das Szenario, dass ein Community-Projekt im Rahmen einer Grid-Initiativen sensible Daten im Grid verarbeiten möchte und dadurch bedingt eine Berichterstattung zur Sicherheit und proaktive Benachrichtigung im Falle einer Sicherheitsverletzung benötigt. Die aus diesem Anwendungsfall anfallenden Anforderungen an ein GIDS und seine Berichterstattungsmöglichkeiten sind in Tabelle 5.1 nochmals zusammenfassen gardestellt. Ein ✔ steht dabei für eine Erfüllung der Anforderung, ein **(✔)** besagt, dass diese Anforderung mit Einschränkungen und ggf. abhängig von der Implementierung erfüllt werden kann, ein ✘ bedeutet, dass diese Anforderung nicht erfüllt ist.

Die Erfüllung der einzelnen Anforderungen ergibt sich durch die im Folgenden beschriebenen Eigenschaften des Konzepts:

1. Durch das Etablieren eines GIDS-Betreibers ist ein zentraler Ansprechpartner für die Kunden des GIDS gegeben. Dieser stellt mit der Komponente des Benutzerportals eine einheitliche Schnittstelle für den Endanwender bereit. Wie nutzerfreundlich

	Anforderung	
1.	Verwendung einer einheitlichen und übersichtlichen Benutzerschnittstelle (z.B. in Form eines Web-basierten Portals)	✔
2.	Unterstützung Virtueller Organisationen und daraus folgend Einbindung in VO-Managementsysteme	✔
3.	Mandantenfähigkeit, nutzergruppenabhängige Berichterstattungen und Sichten	✔
4.	Einbindung bestehender AA-Mechanismen	✔
5.	Nachvollziehbarkeit durchgeführter Anfragen	(✔)
6.	Aussagekräftige Informationsaufbereitung	(✔)
7.	Proaktive Benachrichtigung zuvor festgelegter Ansprechpartner über erkannte Unregelmäßigkeiten	✔
8.	Nachhalten historischer Berichte	(✔)
9.	Verschiedene Granularitätsstufen bei der Berichterstattung	(✔)
10.	Gesicherter Informationsaustausch (Nachrichtenintegrität, -authentizität und -vertraulichkeit)	✔
11.	Push- und Pull-Mechanismen für den Datenzugriff	✔

Tabelle 5.1**:** Erfüllungsnachweis der erhobenen Anforderungen

und übersichtlich diese gestaltet ist, hängt gewiss von der Implementierung ab, die grundlegenden Möglichkeiten der geordneten Informationsrepräsentation sind durch ein geeignetes Datenbankformat gewährleistet.

2. Die Anbindung der Benutzerschnittstelle an die im Grid gegebenen VO-Managementsysteme ermöglicht eine Unterstützung des VO-Aspekts und ...

3. ... ermöglicht es durch die Möglichkeit der Abbildung von Teilnehmern auf ihre VOs und weiter die zur Verwendung verfügbaren Ressourcen eine Mandantenfähigkeit und nutzergruppenabhängige Sichten auf die zur Verfügung gestellten Berichte zu implementieren.

4. Auch die Authentifizierung und Autorisierung wird durch die VO-Managementsysteme übernommen und ist somit delegiert. Es besteht also nicht die Notwendigkeit eine weitere AA-Infrastruktur für das GIDS zu entwickeln.

5. Die Nachvollziehbarkeit durchgeführter Anfragen kann prinzipiell gewährleistet werden. Dafür stehen die entsprechenden Datenbanken bereits zur Verfügung, allein Datenschutzaspekte müssen hierbei jedoch berücksichtigt werden, die es im Einzelfall zu überprüfen gilt. Aus diesem Grund ist diese Anforderung lediglich mit Einschränkungen erfüllt, jedoch durch das vorgestellte Konzept erfüllbar.

6. Da im Rahmen des hiesigen GIDS weder eine spezielle Implementierung noch die Verwendung einer vorgegebenen Logik gefordert ist, hängt implizit auch die aussagekräftige Informationsaufbereitung von den ein- und umgesetzten Mitteln und

Möglichkeiten ab. Deswegen wird auch diese Anforderung als nur mit Einschränkungen erfüllt bezeichnet.

7. Insbesondere durch eine Anbindung der Komponente zur proaktiven Benachrichtigung an die GIDS-Datenbank kann diese Anforderung erfüllt werden. Proaktive Meldungen von Sicherheitsvorfällen können sowohl an die Betreiber als auch die Nutzer und Nutzergruppen einer betroffenen Ressource versendet werden. Die entsprechenden Ansprechpartner sind in den Datenbanken des Grid-Monitorings und dem VO-Management hinterlegt.

8. Genau wie der fünfte Punkt ist auch das Nachhalten historischer Berichte durch die ohnehin vorhandenen Datenbanken vom Konzept her möglich. Einzig Datenschutzaspekte müssen im Einzelfall geprüft werden, was ggf. Einschränkungen mit sich bringen kann. Aus diesem Grund ist auch diese Anforderung als nur mit Einschränkungen erfüllt verzeichnet.

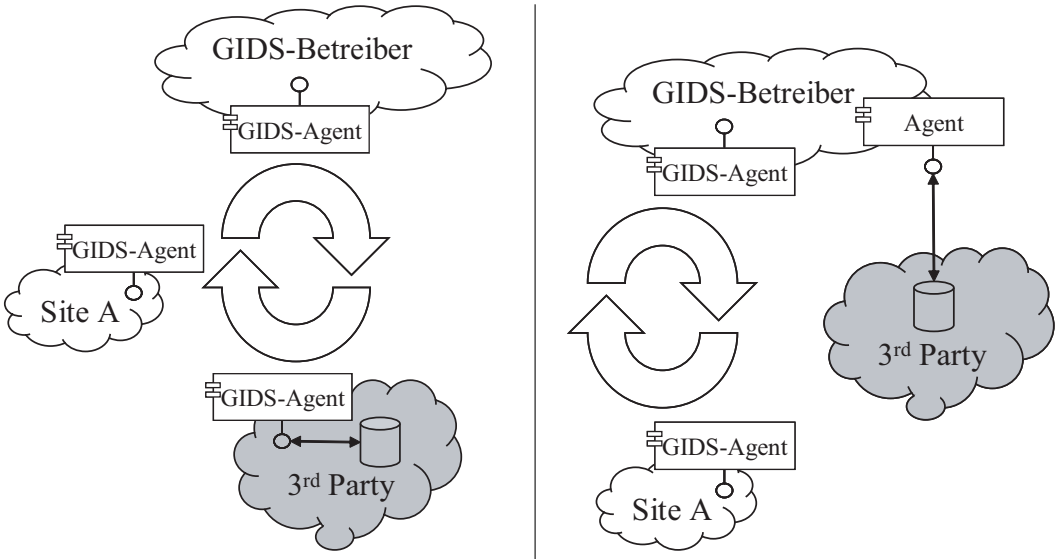
9. Die Granularität bei der Berichterstattung hängt maßgeblich auch vom verwendeten Datenmodell im Rahmen des GIDS ab. Da alle gängigen Datenmodelle jedoch Attribute wie zum Beispiel den Schweregrad eines Vorfalls aufweisen, ist diese Anforderung fast immer erfüllbar, dennoch abhängig von der Wahl der Instanziierung des Gesamtsystems.

10. Ein gesicherter Informationsaustausch ist ein Muss für ein jedes IDS und auch im Grid bereits seit seinen ersten Anfängen gegeben. Sämtliche Mechanismen zur Gewährleistung eines gesicherten Informationskanals sind inhärent vorhanden, und diese Anforderung kann somit als erfüllt erachtet werden.

11. Aus Nutzer- und Kundensicht ist auch diese Anforderung erfüllt. Zum einen kann eine proaktive Benachrichtigung (*Push*) konfiguriert werden, zum anderen kann aktiv auf ein Portal zur Berichterstattung zugegriffen werden (*Pull*).

5.2.4 Erweiterungsmöglichkeiten

In den bisherigen Abschnitten stand insbesondere die Konzeption und Inbetriebnahme eines Intrusion Detection Systems für Grids im Vordergrund. Betrachtet man jedoch einen Dienstlebenszyklus, so ist auch die Adaption des Dienstes während seiner Betriebsphase integraler Bestandteil. Hierzu soll in diesem Abschnitt die Möglichkeit der Erweiterung des zuvor konzipierten GIDS kurz beleuchtet werden. Die Erweiterungsmöglichkeiten zeigen sich dabei zweidimensional, zum einen kann das GIDS um weitere Informationsanbieter aus dem Grid (z.B. zusätzliche/neue Ressourcenanbieter) ergänzt werden, zum anderen können auch Informationen von dem Grid fremden Drittanbietern dienlich beigesteuert werden. Diese beiden Fälle werden nachfolgend kurz detailliert.



(a) direkt (b) via Mittelsmann

Abbildung 5.5**:** Erweiterung des GIDS um Informationen von Drittanbietern

5.2.4.1 Erweiterung um weitere Informationsanbieter aus dem Grid

Die einzig notwendige Vorraussetzung, um einen weiteren Informationsanbieter in das GIDS zu integrieren, ist, dass dieser einen GIDS-Agenten zur Kommunikation innerhalb des GIDS in Betrieb nimmt. Dieser neue GIDS-Agent muss dazu autorisiert werden am Datenaustausch teilzunehmen, was durch die koordinierende Instanz, also den GIDS-Betreiber, Grid-global vorgenommen werden kann. Prinzipiell kann diese Funktion jedoch ein beliebiger vertrauenswürdiger Teilnehmer des GIDS übernehmen.

Damit der neue Informationsanbieter die anfallenden Daten, die im GIDS verteilt werden, sinnvoll nutzen kann, muss er mindestens eine lokale Instanz eines (G)IDS mit einer angeschlossenen Datenbank vorhalten. Um auch aktiv Sensordaten und Sicherheitsberichte im Grid verbreiten zu können, sind zusätzlich Agenten an den lokal datensammelnden Komponenten zu platzieren sowie die Kaskade an Komponenten, die eine geregelte Datenweitergabe gewährleistet (*Filter*, *Aggregator* und *Anonymisierer/Pseudonymisierer*), zu etablieren.

5.2.4.2 Erweiterung um Informationen von Drittanbietern

Die in Abschnitt 5.1 vorgestellte Idee zum Aufbau eines Grid-weiten Frühwarnsystems in der Art und Weise, wie es vorgeschlagen ist, bietet insbesondere die Möglichkeit der Erweiterung der Informationsbasis durch das Hinzufügen weiterer Informationsanbieter. Wie auch in der Anforderungsanalyse deutlich geworden ist, können durchaus auch Informationen von Drittanbietern wie beispielsweise CERTs von großem Interesse sein. Da ein durch die Vernetzung der GIDS-Agenten bedingtes Informationsverbreitungssystem besteht, ist es denkbar auch externe Informationsquellen, die als vertrauenswürdig und hilfreich ein-

gestuft werden, mit einzubinden.

Dies geschieht zum Beispiel durch die Installation eines entsprechenden Agenten direkt bei dem Informationsanbieter, wie es in Abbildung 5.5a graphisch dargestellt wird. Dies hat jedoch unter anderem den Nachteil, dass eine im Grid nicht beteiligte Partei (in diesem Fall also der Drittanbieter) konzeptuell alle Informationen, die die am GIDS beteiligten Partner untereinander austauschen, mithören kann. Dies kann aus verschiedenen Gesichtspunkten (mangelndes Vertrauen, juristische Einschränkungen etc.) nicht erwünscht oder erlaubt sein.

Alternativ kann eine Anbindung über einen Mittelsmann im Grid, der die Informationen für das GIDS aufbereitet und bereitstellt, geschehen (siehe Abbildung 5.5b). Ein geeigneter Mittelsmann dafür ist z.B. der geforderte Betreiber des GIDS. Dieser kann unter Nutzung eines eigenen Agenten die Informationen des Drittanbieters entgegennehmen, aufbereiten und in seiner Informationsbasis hinterlegen. In Abweichung zu dem zuvor beschriebenem Fakt, dass der Betreiber des GIDS sich unter anderem von einem Ressourcenanbieter darin unterscheidet, dass er nicht als Informationsanbieter fungiert, stellt in diesem Fall der Betreiber des GIDS doch Informationen im GIDS bereit.

5.3 Kritische Diskussion des Architekturvorschlags

Dieser Abschnitt betrachtet die zuvor vorgeschlagene Architektur kritisch. Dabei stehen die Kategorien „Sicherheitsanforderungen“, „Organisatorische und Datenschutzanforderungen“ sowie die „Erkennungsleistung“, wie sie in Tabelle 3.20 ab Seite 72 identifiziert wurden, im Vordergrund. Es gilt nicht einen Abgleich gegenüber den in Abschnitt 3 erhobenen Anforderungen durchzuführen, welcher für eine konkrete Implementierung in Abschnitt 6.3 zu finden ist, sondern vielmehr, potentielle durch die Architektur bedingte Schwächen und Herausforderungen zu diskutieren. Eine Reihe an Herausforderungen wird nachfolgend nur aufgezeigt, jedoch keine konkrete Lösung dafür vorgegeben. Dies ist insbesondere dann der Fall, wenn es sich um eine organisatorische und keine konzeptuelle oder von einer Implementierung abhängige Herausforderung handelt. In diesen Fällen muss zumeist eine vom konkreten Einsatzszenario abhängige Lösung individuell erarbeitet werden.

5.3.1 Erwägungen zur Sicherheit und Erkennungsleistung

Natürlich kann auch ein jedes System zur Erhöhung der Sicherheit selber Ziel eines Angriffs sein oder ausgenutzt werden um Angriffe vorzubereiten oder durchzuführen. Ein Blick auf den Architekturvorschlag (insbesondere die Abbildungen 5.3 auf Seite 101 und Abbildung 5.4 auf Seite 108) und die darin enthaltenen Komponenten legt potentielle Angriffspunkte offen.

Vertraulichkeit und Authentizität. Beide Disziplinen spielen eine entscheidende Rolle bei der Kommunikation der einzelnen Komponenten des GIDS untereinander, insbesondere dann, wenn es sich dabei um die Kommunikation unter Nutzung öffentlicher Netze handelt. Durch den bereits geforderten Einsatz entsprechender kryptographischer Verfahren und Protokolle (siehe auch Tabelle 3.20 ab Seite 72), ist dies jedoch zufriedenstellend zu gewährleisten

Integrität. Genau wie die Vertraulichkeit und Authentizität der Kommunikation der einzelnen Komponenten des GIDS untereinander, ist auch die Integrität zu gewährleisten. Auch hierfür sind entsprechende kryptographische Verfahren und Protokolle gefordert, die zufriedenstellende Ergebnisse erzielen.

Im Falle der Datenintegrität ist jedoch noch eine entscheidende zweite Dimension in Betracht zu ziehen: Es gilt ebenfalls die Integrität gespeicherter Daten (z.B. Sensorrohdaten oder auch Sicherheitsberichte) zu garantieren. Auf Grund eines dynamischen Datenbestandes wird dies zumeist sträflich vernachlässigt, da sich Mechanismen zur Überprüfung der Datenintegrität (ein Beispiel hierzu ist das Host-basierte IDS *Samhain* [Samhain]) in der Regel nur auf statischen, sich nicht permanent ändernden Datenbeständen praktikabel anwendbar zeigen. Vielmehr wird die Datenintegrität oftmals über sichere Systeme mit strengen Zugriffsbeschränkungen und -kontrollen realisiert, was natürlich nicht optimal ist und Angriffspotentiale bietet. Als eine eventuell gangbare Lösung könnte sich eine Verschlüsselung schützenswerter Daten erweisen, mit der implizit auch die Datenintegrität überprüft werden kann.

Verfügbarkeit. Die Verfügbarkeit des GIDS spaltet sich ebenfalls in zwei Gruppen. Zum einen kann allein die Verfügbarkeit einzelner (Teil-)Komponenten beeinträchtigt sein. Dies lässt sich leicht zum Beispiel durch die Nutzung von regelmäßigen *Heartbeat-Nachrichten* erkennen und man kann entsprechend reagieren.

Eine weitaus gefährlichere Situation ist der Ausfall des gesamten GIDS. Auf Grund der verteilten Konzeption des Systems ist ein durch einen außenstehenden Angreifer provozierter Ausfall jedoch nur schwer durchzusetzen. Allerdings bietet die Architektur an sich eine Möglichkeit die Verfügbarkeit des Gesamtsystems anzugreifen. Wie bereits in Abschnitt 5.2.1 kurz angesprochen wurde, ist prinzipiell eine Überflutung des GIDS durch provozierte Endlosschleifen von Nachrichten möglich, die in einer Denial-of-Service Situation münden können. Diesem zu begegnen ist eine Aufgabe der Komponente *Filter* (siehe Abbildung 5.3 auf Seite 101). An dieser Stelle muss ganz besonders auf eine ordentliche Implementierung inklusive geeigneter Anti-Spoofing Mechanismen[1] geachtet werden.

Zugriffsbeschränkungen & Nutzerverwaltung. Auch wenn sich die Zugriffskontrollen und Nutzerverwaltungssystem in den meisten Fällen in den Bereich der Sicherheit

[1] Mit *Spoofing* (engl. für die Manipulation oder Verschleierung) bezeichnet man das Vorgeben einer falschen Eigenschaft oder Identität. *Anti-Spoofing* bezeichnet geeignete Maßnahmen um dem zu begegnen.

einordnen, so erwächst aus der Möglichkeit, verschiedene Nutzergruppen Zugriff auf ein gemeinsames System oder dessen Datenbestand zu gewähren, eine organisatorische Herausforderung. Es muss geklärt werden, wer auf welchen Datenbestand Zugriff hat und für welchen Gruppen von Kunden geeignete Sichten auf beispielsweise Analyseergebnisse des GIDS realisiert werden. Diese Fragen gilt es spezifisch für das jeweilige Szenario zu klären, die Möglichkeit der technischen Realisierung ist dazu gegeben.

Sensorplatzierung. Die Platzierung der Sensorik eines jeden Intrusion Detection Systems ist ein entscheidender Faktor für die Güte der Erkennungsleistung des Systems. Mit einem Blick zurück auf Abbildung 5.1 auf Seite 97 und der daraus folgenden Abbildung 5.2 auf Seite 98 steht sofort fest, dass die „Sensoren“ des GIDS als die Daten der Informationsanbieter inhärent festgelegt sind. Kritischer hingegen ist die Platzierung der Sensoren auf Seiten eines jeden Informationsanbieters. Jedoch ist dies eine weitere Herausforderung, die individuell für jeden Informationsanbieter gelöst und zur Gewährleistung dessen Autonomie von ihm selber realisiert werden muss.

Erkennungsleistung. Entscheidend für die Erkennungsleistung eines IDS ist zum einen die zuvor angesprochene Platzierung der Sensoren, zum anderen die Auswertungslogik, die zum Einsatz kommt, um aus der vorhandenen Datenbasis Rückschlüsse auf Angriffe zu ziehen. Da das in Abschnitt 5.2 vorgestellte Konzept zum Aufbau eines Grid-basierten IDS eine Art rekursive Schachtelung von Sicherheitssystemen vorsieht, mit der bei den Informationsanbietern bereits vorliegende Informationen zusammengeführt werden können, stellt sich zusätzlich jedoch die Frage, ob und wie Domänenlokal erkannte Angriffe (nicht die Rohdaten, die ohnehin übergreifend verarbeitet werden!) und ihre Meldung Grid-global korreliert werden können. Der Architekturvorschlag bietet eine entsprechende Basis dafür, in den Ausführungen dazu ist bereits kurz die Verbreitung von Meldungen zu lokal erkannten Angriffen erwähnt worden. Die Möglichkeit für eine weiterführende (Grid-globale) Korrelation ist ebenfalls durch den Entwurf des hiesigen GIDS gegeben, da entsprechende Verbreitungswege bereits existieren. Allein die Auswertungslogik muss an eine solche Korrelation notwendigerweise angepasst werden. Für eine Korrelation von (sicherheitsrelevanten) Daten existieren bereits Projekte wie unter anderem der *Simple Event Correlator* (SEC) [Simple Event Correlator] oder die entsprechenden Teile des *OSSEC* quelloffenen HIDS [OSSEC], die eine solche Funktion übernehmen können.

5.3.2 Aus dem Datenschutz erwachsende Herausforderungen

Viele juristische, aber auch einige technische Herausforderungen resultieren aus Anforderungen, die dem Datenschutz entstammen. Nachfolgend werden diese kurz diskutiert und vor allem aufgezeigt in welchen Spannungsfeldern man sich bewegt. Zumeist sind Lösungen jedoch abhängig vom Szenario zu suchen, welche Optionen allerdings zur Verfügung stehen, wird kurz umrissen.

Autonomie der Informationsanbieter. Zum einen ist die Autonomie der Informationsanbieter eine notwendige Voraussetzung, damit es zu einem gemeinschaftlichen GIDS kommen kann, zum anderen bringt sie eine Reihe Herausforderungen mit sich, die zum Teil bereits im vorangehenden Abschnitt angeschnitten wurden. In Bezug auf Datenschutzanforderungen ergeben sich zusätzliche Herausforderungen oder sogar Probleme, denen begegnet werden muss.

Eine entscheidende Anforderung bei der Konzeption des GIDS war, dass die am GIDS als Informationsanbieter beteiligten Partner selber die Kontrolle über die von ihnen preisgegebenen Informationen haben (Stichwort *Informationsverbreitungsrichtlinien* oder engl. *Information Sharing Policies*), dem im Systementwurf durch eine Reihe Komponenten Rechnung getragen wird. Da die Informationsverbreitungsrichtlinien zwischen den beteiligten Parteien im schlimmsten Fall paarweise verschieden sein können, stellt sich die Frage, wie mit den verschiedenen Information Sharing Policies der beteiligten Partner untereinander umgegangen werden kann. Ein sehr einfaches Beispiel dazu ist, dass Domäne A Domäne B vertraut, aber nicht Domäne C. A ist also bereit Informationen mit B zu teilen, nicht aber mit C. Daraus resultiert die Frage, wie (Broadcast, Multicast oder Unicast) und welche Informationen (z.B. speziell für die einzelnen beteiligten Parteien bereinigten Daten) dem GIDS zur Verfügung gestellt werden.

Für die Beantwortung dieser Frage gibt es zuerst einmal die beiden Extremfälle, dass entweder für jeden der beteiligten Partner individuell aufbereitete Informationen per Unicast verbreitet werden. Das andere Extremum ist die Verteilung des Minimums an Information, das man bereit ist jedem beteiligten Partner verfügbar zu machen, per Broadcast zu verbreiten. Damit findet man sich in der Situation wieder, dass zwischen Skalierungsschwierigkeiten im Falle massiver Unicast-Kommunikation und unnötigem Informationsverlust im Falle der Broadcast-Variante abgewogen werden muss. Ein vom Einsatzszenario abhängig gangbarer Mittelweg stellt die Bildung von Vertrauensgruppen mit nahe gleicher Vertrauensbasis dar, die durch die Bildung von Multicast-Gruppen alle mit dem gleichen Informationsgehalt bedient werden. Somit lässt sich unter Nutzung mehrerer Multicast-Gruppen ein Kompromiss zwischen Skalierbarkeit und künstlichem Informationsverlust eingehen.

Anonymisierung und Pseudonymisierung. Da Anonymisierungs- und Pseudonymisierungskomponenten im GIDS von jedem der beteiligten Partner selbst betrieben und somit auch konfiguriert werden sollen, obliegt es auch jedem Partner selber die Art und Weise der Anonymisierung und/oder Pseudonymisierung seiner Daten vor ihrer Veröffentlichung im GIDS festzulegen. Dabei gilt es zu beachten, dass geeignete Verfahren zum Einsatz kommen, die zum einen die rechtlichen Randbedingungen erfüllen, zum anderen kann es aber auch gewünscht sein, dass aus den verbreiteten Informationen keine weiteren Rückschlüsse auf die zu Grunde liegende Infrastruktur und deren Merkmale gezogen werden können. Ein sehr einfaches Beispiel hierzu ist eine Anonymisierung von IP-Adressen durch das Löschen ihrer Netzadresse. Durch

das Beibehalten der Hostadressen ist zwar die Herkunft der Daten nicht trivial, aber dennoch mit geringem Aufwand herauszufinden, die Topologiemerkmale des Informationsanbieters hingegen liegen direkt ohne weiteren Aufwand offen. Natürlich ist dies ein sehr simples Beispiel, jedoch ist (falls notwendig) bei jeder Art der Anonymisierung oder Pseudonymisierung darauf zu achten, welche Informationen sich implizit ableiten lassen und ob dies gewünscht oder zu tolerieren ist.

Archivierung und Datenschutz. Eine Archivierung von Informationen ist zumeist im Umfeld der Systemsicherheit ein wünschenswertes Vorhaben, um forensische Arbeiten durch einen möglichst detaillierten Datenbestand unterstützen zu können. Dem entgegen stehen zumeist auf der einen Seite juristische Vorgaben und der Datenschutz, auf der anderen Seite die aufzuwendenden Speicherkapazitäten und damit verbundenen Kosten, um feingranulare Daten großer Systeme über einen lange Zeitraum aufzubewahren. Der Architekturvorschlag für ein GIDS in dieser Arbeit sieht Datenbanken zur Informationshaltung innerhalb einer jeden am GIDS teilnehmenden Domäne vor. Diese eigenen sich insbesondere auch zur Archivierung von für die Sicherheit relevanten Daten. Vom Einsatzfall abhängig muss jedoch geklärt werden, wer welche Daten in welchem Detailgrad wie lange vorhalten muss, soll oder darf. Eine pauschale Aussage hierzu scheint an dieser Stelle nicht möglich und betrifft vorwiegend die Juristerei und den Datenschutz, weswegen dieses Problemfeld im Zuge dieser Arbeit nicht weiter behandelt wird.

5.3.3 Weitere Herausforderungen an eine Implementierung

Auch durch die funktionalen und nicht-funktionalen Anforderungen bedingt resultieren einige Herausforderungen an eine Implementierung, die im Vorhinein geklärt werden müssen und nicht unbedingt direkt durch den vorgestellten Architekturvorschlag erfüllt werden.

Zum einen muss durch eine konkrete Implementierung die Integrierbarkeit des GIDS in bestehende Management-Werkzeuge gewährleistet werden. Dazu zählen insbesondere auch Möglichkeiten zur Überwachung bzw. das Monitoring, sowie auch komfortable Möglichkeiten zur Steuerung (engl. *steering*) des GIDS als neuer Grid-Dienst. Dazu bietet es sich an, der Forderung nach der Wiederverwendbarkeit der entwickelten Komponenten durch die Nutzung einheitlicher Schnittstellen und die Unterstützung etablierter (Grid-) Standards nachzukommen. Durch den erreichten Grad an Interoperabilität mit anderen, zum Teil bereits bestehenden und erfolgreich eingesetzten Werkzeugen, kann implizit auch eine gewisse Portabilität (z.B. durch zusätzliche Verwendung einer Programmiersprache, die auf den unterschiedlichsten Plattformen einheitlich verfügbar ist) erzielt werden.

Ein zweiter Komplex von zu beachtenden notwendigen Eigenschaften tangiert die Erweiterbarkeit und Flexibilität des zu instanziierenden GIDS. Zum einen muss eine Erweiterbarkeit des Funktionsumfangs zum Beispiel durch das Einbringen neuer Komponenten möglich sein, zum anderen muss sich auch die Implementierung des Systems flexibel und tolerant gegenüber Variationen innerhalb der verfügbaren Informationsquellen bzw.

der Datenbasis sowie der hohen Dynamik der Ressourcen im Grid zeigen. Auch hierzu bieten etablierte (Grid-) Standards mögliche Lösungen an, wie zum Teil bereits in Abschnitt 2.2 beschrieben.

5.4 Zusammenfassung

Basierend auf der Idee der kooperativen Nutzung von lokalen Sicherheitssystemen und dem Austausch von Angriffsdaten stellt dieses Kapitel einen Architekturvorschlag für ein föderiertes Intrusion Detection System für Grids vor. Dazu sind die notwendigen Komponenten aus den in Kapitel 3 erhobenen Anforderungen an ein solches System abgeleitet sowie die Erfahrungen und das Wissen, das aus den in Kapitel 4 dargestellten themenverwandten Arbeiten gewonnen worden ist, zu deren Komposition herangezogen worden. Dieses Vorgehen mündet in einem verteilten GIDS, das die Föderation der am Grid beteiligten Ressourcenanbieter vorsieht. Erstmals ist dabei in diesem Bereich ein Kundenbegriff und die Unterstützung Virtueller Organisationen aufgekommen, die durch die Integration des GIDS in und die Anbindung an bestehende Grid-Dienste und -Ressourcen unterstützt und vielmehr erst ermöglicht wird. Abbildung 5.2 auf Seite 98 gibt zum grundlegenden Aufbau einen Überblick.

Organisatorisch und administrativ umfasst das konzipierte GIDS zwei unterschiedliche Rollen, die sich auch konzeptuell leicht voneinander unterscheiden. Zum einen wird ein Betreiber des GIDS eingeführt, der einen GIDS-Dienst im Grid bereitstellt und einen Kundenbegriff einbringt. Die dazu notwendigen Komponenten und Anbindungen an bereits existierende Grid-Dienste sowie den genauen Aufbau des GIDS auf Betreiberseite ist in Abschnitt 5.2.2 und insbesondere Abbildung 5.4 dargestellt.

Auf der anderen Seite stehen die Ressourcenanbieter als Informationslieferanten für das GIDS. Einen Überblick hierzu verschafft Abschnitt 5.2.1, Abbildung 5.3 stellt den genauen Aufbau auf Seiten eines Ressourcenanbieters graphisch dar. Verschiedene Komponenten dienen dabei der Gewährleistung der Autonomie und Selbstbestimmung der Teilnehmenden Partner sowie auch der Durchsetzung rechtlicher Randbedingungen wie zum Beispiel Datenschutzbestimmungen. Durch die zusätzliche Möglichkeit der Informationsreplikation aller im GIDS anliegenden Daten (Sensordaten wie auch Angriffs- und Sicherheitsberichte) und die Möglichkeit einer zusätzlichen lokalen Angriffserkennung ist insbesondere eine große Ausfallsicherheit und die Möglichkeit zur vollständig dezentralen (redundanten) Angriffserkennung gegeben.

Im weiteren Verlauf der Arbeit stellt das folgende Kapitel 6 eine prototypische Implementierung des hier präsentierten GIDS vor. In Abschnitt 6.3 ist dann auch ein Abgleich mit dem in Tabelle 3.20 zusammengefassten Kriterienkatalog zu finden, so dass auf der einen Seite ein Erfüllungsnachweis der erhobenen Anforderungen geführt und auf der anderen Seite nach wie vor bestehende Schwächen aufgezeigt werden können.

Kapitel 6

Erfüllungsnachweis des Konzepts & Prototypische Implementierung

Inhalt des Kapitels

Um einen Tragfähigkeitsnachweis des zuvor in Kapitel 5 vorgestellten Konzeptes zu führen, wird sich dieses Kapitel auf der einen Seite mit einem Abgleich des vorgeschlagenen GIDS mit dem in Abschnitt 3.4 aufgestellten Katalog an Anforderungen an ein IDS im Grid-Umfeld auseinandersetzen. Zum anderen werden sowohl unter Laborbedingungen erzielte Messergebnisse zum GIDS wie auch Erfahrungswerte aus dem Betrieb im Münchener Wissenschaftsnetz (MWN) angeführt. Die Basis für alle vorgenannten Erfüllungsnachweise des Konzepts bietet eine in Abschnitt 6.1 aufgezeigte prototypische Implementierung.

6.1 Prototypische Implementierung

Um eine prototypische Implementierung des in Kapitel 5 vorgeschlagenen GIDS zu beginnen, muss in erster Instanz ein gemeinschaftliches Datenmodell festgelegt werden.

Die *Intrusion Detection Working Group* (IDWG) der *Internet Engineering Task Force* (IETF) beschäftigt sich seit geraumer Zeit mit Datenformaten und Kommunikationsprotokollen zum Datenaustausch zwischen Komponenten von Intrusion Detection Systemen. Für diesen Prototypen fällt die Wahl auf das *Intrusion Detection Message Exchange Format* (IDMEF) nach [RFC4765, 2007].

Das IDMEF, das in Abbildung 6.1 illustriert wird, stellt ein einheitliches Nachrichtenformat für den Informationsaustausch zwischen Intrusion Detection Systemen dar. Ziel der Entwicklung eines solchen Formats ist die Flexibilität gegenüber verschiedenen Anwendungen. Dieser Anforderung trägt IDMEF durch eine Repräsentation der Nachrichten in der *Extensible Markup Language* (XML) Rechnung.

Ein großer Vorteil eines einheitlichen Nachrichtenformats ist die Möglichkeit zur Interoperabilität verschiedener Intrusion Detection Systeme. Anstatt herstellerspezifischer Kommunikationsmechanismen, die in der Regel nur eine Anbindung von Sicherheitskomponenten des gleichen Herstellers erlauben, bietet IDMEF eine Möglichkeit heterogene Sicherheitskonzepte zu realisieren.

Wie aus Abbildung 6.1 hervorgeht, existieren zwei Typen von IDMEF-Nachrichten. Es wird zwischen den sogenannten *Alert-* und den *Heartbeat-Nachrichten* differenziert.

Alert-Nachricht. Dieser Typ von Nachricht wird zur Meldung eines sicherheitsrelevanten Ereignisses versendet. Neben Informationen zu dem analysierenden Element, welches

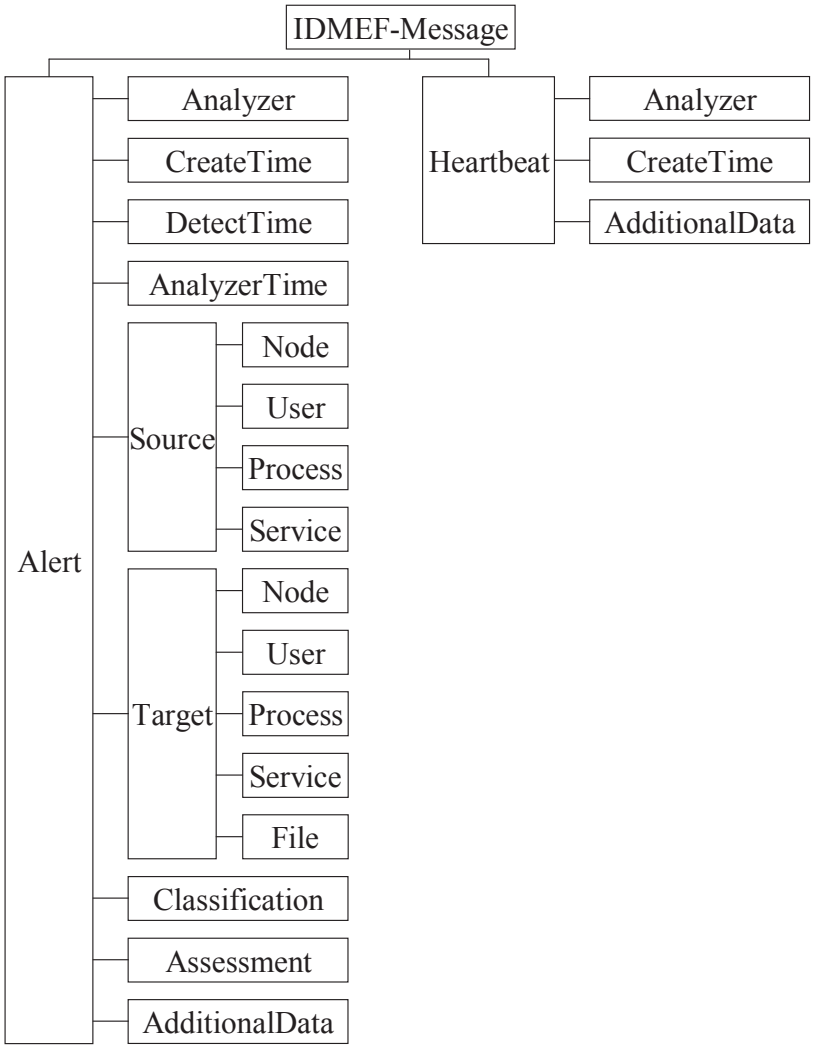


Abbildung 6.1: Das IDMEF-Datenmodell nach [RFC4765, 2007]

die Meldung generiert hat, sind in einer Alert-Nachricht detaillierte Informationen zu Quelle (*Source*) und Ziel (*Target*) des gemeldeten Ereignisses enthalten. Hierbei ist es insbesondere möglich Angaben über IP-Adressen und Ports zu machen. Zusätzlich können die Nachrichten dieses Typs neben weiteren Details mit einer Reihe verschiedener Zeitstempel versehen werden, wie aus Abbildung 6.1 im linken Zweig der Illustration hervorgeht.

Heartbeat-Nachricht. Heartbeat-Nachrichten sind dazu gedacht, in regelmäßigen Zeitabständen von den einzelnen Komponenten eines gemeinsamen Systems verschickt zu werden. Diese Nachrichten signalisieren die Funktionsfähigkeit der jeweiligen Komponente. Ein Fehlen einer oder mehrerer Heartbeat-Nachrichten weist oftmals auf eine Funktionsstörung oder den Ausfall eines Teils des IDS hin, was wiederum ein Indiz für einen Angriff sein kann.

Zwar steht IDMEF oftmals in der Kritik sehr Overhead-belastet durch seine XML-Basis zu sein, jedoch handelt es sich um ein standardisiertes Datenmodell, das unter anderem bereits von einigen Intrusion Detection Systemen erfolgreich eingesetzt wird. Vor allem liegen die Vorteile dieser Wahl in einer offenen und herstellerunabhängigen Spezifikation, zu der bereits einige Referenzimplementierungen existieren und somit eine Menge existierender Systeme problemlos in das GIDS integriert werden können.

6.1.1 Implementierungsvorschlag auf Seiten eines Ressourcenanbieters

Die grundlegende Idee bei der Implementierung der einzelnen Komponenten ist die Implementierung jeder Komponente als eigenständige Anwendung mit einem vorab definierten Ein- und Ausgabeformat (hier in den meisten Fällen IDMEF). Dadurch ist eine jede Komponente als eigener Prozess innerhalb eines (Betriebs-)Systems ausführbar, wodurch ein hoher Grad an Parallelität und eine effektive Nutzung von Mehrkern-Rechnerarchitekturen gewährleistet werden kann. Zur Inter-Prozesskommunikation verwenden wir in hiesigem Fall *named pipes*, die durch Linux FIFOs (*First In, First Out*) realisiert werden.

Abbildung 6.2 zeigt schematisch den Aufbau des GIDS auf Seiten eines Ressourcenanbieters. Zu beachten ist dabei die Einführung eines Nachrichten-*Dispatcher*s. Dieser ist nicht zwingend notwendig, allerdings aus Gründen der Performanz sehr ratsam, wie sich auch in der weiteren Arbeit zeigen wird.

6.1.1.1 Exemplarische Agenten

Direkt an den informationsliefernden Systemen angreifend, beziehen die Agenten des GIDS ihre Informationen, passen sie an das einheitliche Datenmodell an und speisen die neu generierten Meldungen in das GIDS ein. Nachfolgend werden exemplarisch zwei Agenten vorgestellt, die zum einen ausgewählte Pakete in einem Netz verarbeiten (*tcpdump*-Agent) oder aus Log-Daten eines oder mehrerer Systeme (*syslog-* und *syslog-ng*-Agent) operieren können.

6.1.1.1.1 Ein Agent für *tcpdump* Listing B.1 in Anhang B zeigt die Implementierung eines Agenten, der auf dem Paket-Sniffer *tcpdump* basiert. *tcpdump* bringt bereits von Hause aus die Möglichkeit der Auswahl der auszugebenden Paketdaten mit sich. Zum Beispiel gibt die Kommandozeile

```
# To print the start and end packets (the SYN and FIN packets)
# of each TCP conversation that involves a non-local host.
tcpdump "tcp[tcpflags] & (tcp-syn|tcp-fin) != 0 \
         and not src and dst net 192.168.0.0/16"
```

alle Pakete einer jeden TCP-Verbindung mit gesetztem SYN- oder FIN-Flag aus, wobei weder Quell- noch Ziel-IP-Adresse aus dem lokalen Netz `192.168.0.0` mit der Netzmaske

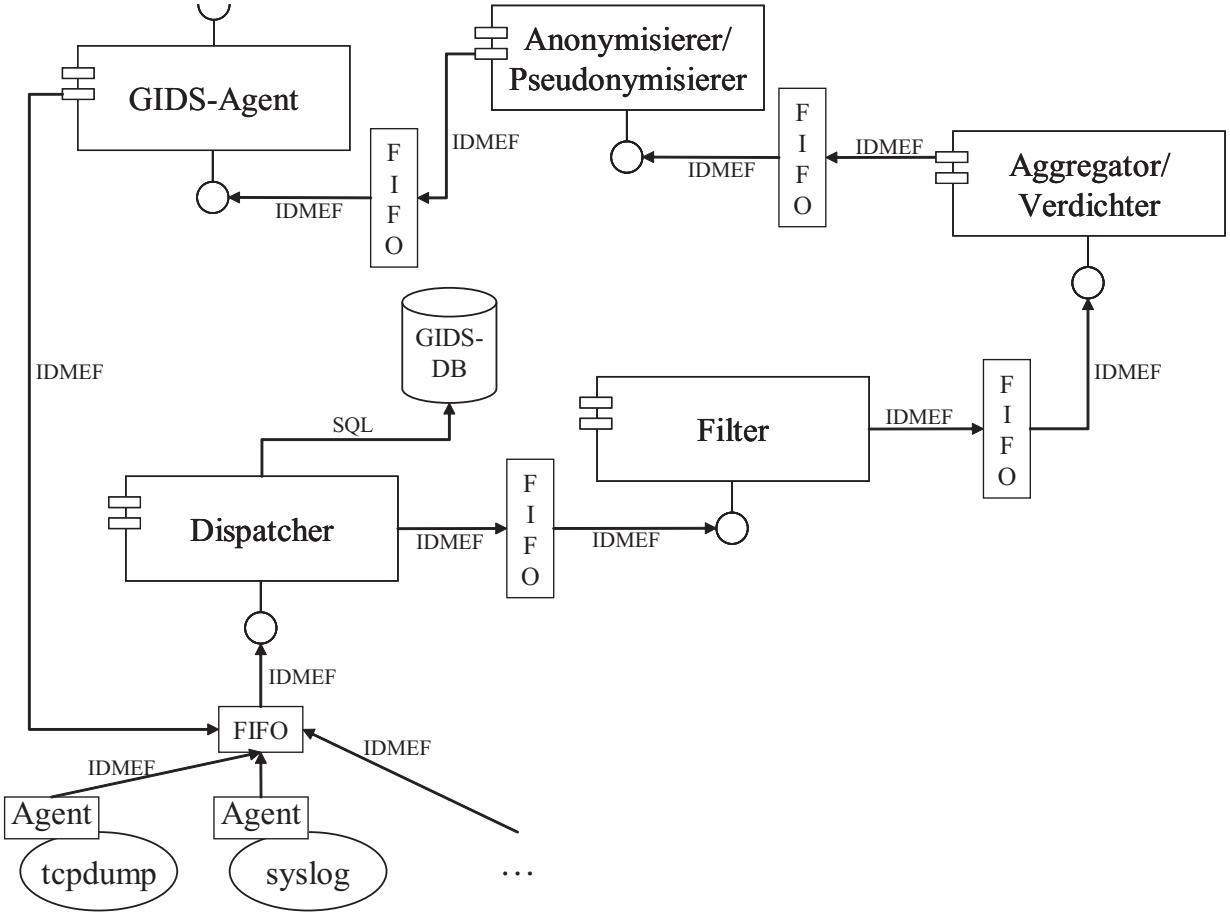


Abbildung 6.2: Implementierungsvorschlag zur Steigerung der Performanz

`255.255.0.0` stammen. Der Sinn dieser Einstellung sei dahingestellt, die Möglichkeiten und Mächtigkeit, über die *tcpdump* verfügt, werden jedoch damit gut demonstriert.

Die Platzierung eines solchen Sensors zum Beispiel an einem Spiegel-Port einer zentralen Koppelkomponente ermöglicht einen übergreifenden Überblick über große Netzsegmente. Ein Nachteil jedoch ist, dass die hiesige Implementierung eines solchen Agenten mit administrativen Rechten (*Root-Rechte*) ausgeführt werden muss. Daraus resultiert die Notwendigkeit das ausführende System entsprechend gut abzusichern und vor Angriffen insbesondere auf den Agenten zu schützen.

6.1.1.1.2 Ein Agent für *syslog* bzw. *syslog-ng* Auf Linux-basierten Systemen wird der *syslog*-Mechanismus dazu verwendet, Log-Daten beliebiger Prozesse und Anwendungen zentral zu verwalten. Beispiele für solche Informationen sind unter anderem Daten zu fehlerhaften Anmeldeversuchen am System (sowohl lokal als auch entfernte Zugriffe) oder auch Log-Daten von Firewalls wie beispielsweise *iptables*. Zurzeit kommen hierzu zumeist eine der beiden Varianten *syslog* oder das aktuellere *syslog-ng* zum Einsatz. Beide

Implementierungen bieten in ihrer Konfiguration das Filtern spezifizierter Log-Einträge und das gesonderte Abspeichern dieser Daten an einer zu spezifizierenden Stelle an. Dieser Mechanismus erlaubt es u.a. auch in Linux-FIFOs (oder auch *named pipe*s) zu schreiben. Listing 6.1 stellt dabei einen Auszug aus der Konfigurationsdatei für *syslog-ng* dar, um Nachrichten, die Authentifizierungsversuche (erfolgreich wie auch nicht erfolgreich) sowie als kritisch eingestufte Meldungen des Systemkerns in Kopie in eine Linux-FIFO schreiben. Listing 6.2 hingegen demonstriert, wie via *syslog* die äquivalenten Meldungen an ein entferntes System übermittelt und dann dort in eine Linux-FIFO geschrieben werden können.

Listing 6.1**:** Auszug aus der *syslog-ng*-Konfiguration

```
# set pipe as possible destination
destination d_SEC { pipe("./pipe" log_fifo_size(1000)); };
...
# filter for relevant events
filter f_SEC {
     facility(auth,authpriv)
        or level(err..emerg);
};
...
# Sending relevent events pipe
log {
        source(s_all);
        filter(f_SEC);
        destination(d_SEC);
};
```

Listing 6.2**:** Auszug aus der *syslog*-Konfiguration

```
...
# forwarding to foreign loghost
# (filter for relevant events)
auth,authpriv.* @<IP des loghost>
kern.err        @<IP des loghost>
...
```

Ein Agent für *syslog* bzw. *syslog-ng* ist dadurch im Wesentlichen ein Agent, der aus einer Linux-FIFO liest, die Informationen (in Form eines von der Art der Nachricht abhängig geformter Text-String) verarbeitet und in eine IDMEF-Nachricht konvertiert. Da die unterschiedlichen Log-Informationen alle syntaktisch und semantisch Unterschiede aufweisen, muss spezifisch für jeden zu verarbeitenden Meldungstyp eine Anpassung des Agenten vorgenommen werden. Vorteilhaft hingegen ist, dass dieser Agententyp in beliebigem Nutzerkontext ausgeführt werden kann, wenn allein die Berechtigungen der FIFO entsprechend gesetzt sind.

Listing B.2 im Anhang zeigt eine beispielhafte Implementierung eines solchen Agenten. In diesem Beispiel ist der Agent darauf ausgerichtet Informationen des SSH-Dämons eines Linux-basierten Systems zu verarbeiten. Die Zeilen

```
...
Oct 10 08:49:06 pcheger0 sshd[21203]: Did not receive identification string
```

```
                                        from 202.109.114.173
Oct 10 08:59:33 pcheger0 sshd[25987]: Invalid user staff from
                                        202.109.114.173
Oct 10 08:59:33 pcheger0 sshd[25987]: error: Could not get shadow
                                        information for NOUSER
Oct 10 09:01:53 pcheger0 sshd[573]: Failed password for root from
                                      202.109.114.173 port 38771 ssh2
...
```

sind Beispiele für Log-Daten eines OpenSSH-Servers in Form von syslog-Nachrichten, die in IDMEF-Nachrichten kodiert werden können. Eine genaue Darstellung des Vorgangs ist in Listing B.2 zu finden. Diese Implementierung des Agenten eignet sich zusätzlich, um Ausgaben beliebiger Anwendungen, die in einer textbasierten Datei gespeichert werden können, im GIDS veröffentlicht werden können. Dazu ist jedoch jeweils eine Anpassung an den Einzelfall notwendig, was allerdings auch bereits für die Nutzung von Log-Daten verschiedener Anwendungen, die syslog verwenden, gilt.

6.1.1.2 Ein Nachrichten-Dispatcher

Mit Hilfe dieser Komponente kann auf der einen Seite eine direkte Nachrichtenweitergabe an die nachfolgende Filterkette realisiert werden. Auf der anderen Seite kann der Dispatcher eine geeignete Caching-Strategie umsetzen, um das Schreiben in die GIDS-Datenbank zu bündeln und hierdurch das DB-Managementsystem zu entlasten. Zum Beispiel können jeweils fünf Meldungen gesammelt in der GIDS-Datenbank hinterlegt werden, anstatt sie einzeln einzufügen. Dabei besteht die berechtigte Hoffnung, dass ein Durchsatzgewinn an verarbeiteten Nachrichten pro Sekunde erzielt werden kann. Die Vermutung ist dadurch begründet, dass mit diesem Verfahren weniger Sitzungen auf- und wieder abgebaut werden müssen und somit der Overhead, der durch das Sitzungsmanagement bedingt ist, minimiert wird.

Listing C.1 in Anhang C stellt eine mögliche Implementierung eines Nachrichten-Dispatchers vor. Die grundlegende Idee dabei ist, dass eingehende IDMEF-Nachrichten zum einen direkt und unverändert durchgereicht werden. Zum anderen wird auf jede Nachricht eine *Extensible Stylesheet Language Transformation* (XSLT) [XSLT, 1999] mit Hilfe des *xsltproc* angewendet. Als Ausgabe der Transformation werden SQL-Anweisungen zum Einfügen der IDMEF-Nachricht in die GIDS-Datenbank generiert. Es besteht an dieser Stelle die Möglichkeit, mehrere SQL-Anweisungen zwischenzuspeichern und dann gemeinsam an die GIDS-Datenbank weiterzugeben (sogenanntes *Caching*). Die Vorzüge dieses Verfahren sind in Abschnitt 6.2.1.2 genauer ausgeführt.

6.1.1.3 Implementierung einer GIDS-Datenbank

Abbildung 6.3 und 6.4 zeigen einen Ausschnitt aus der Modellierung eines Datenbankschemas, das es ermöglicht im Intrusion Detection Message Exchange Format (IDMEF)

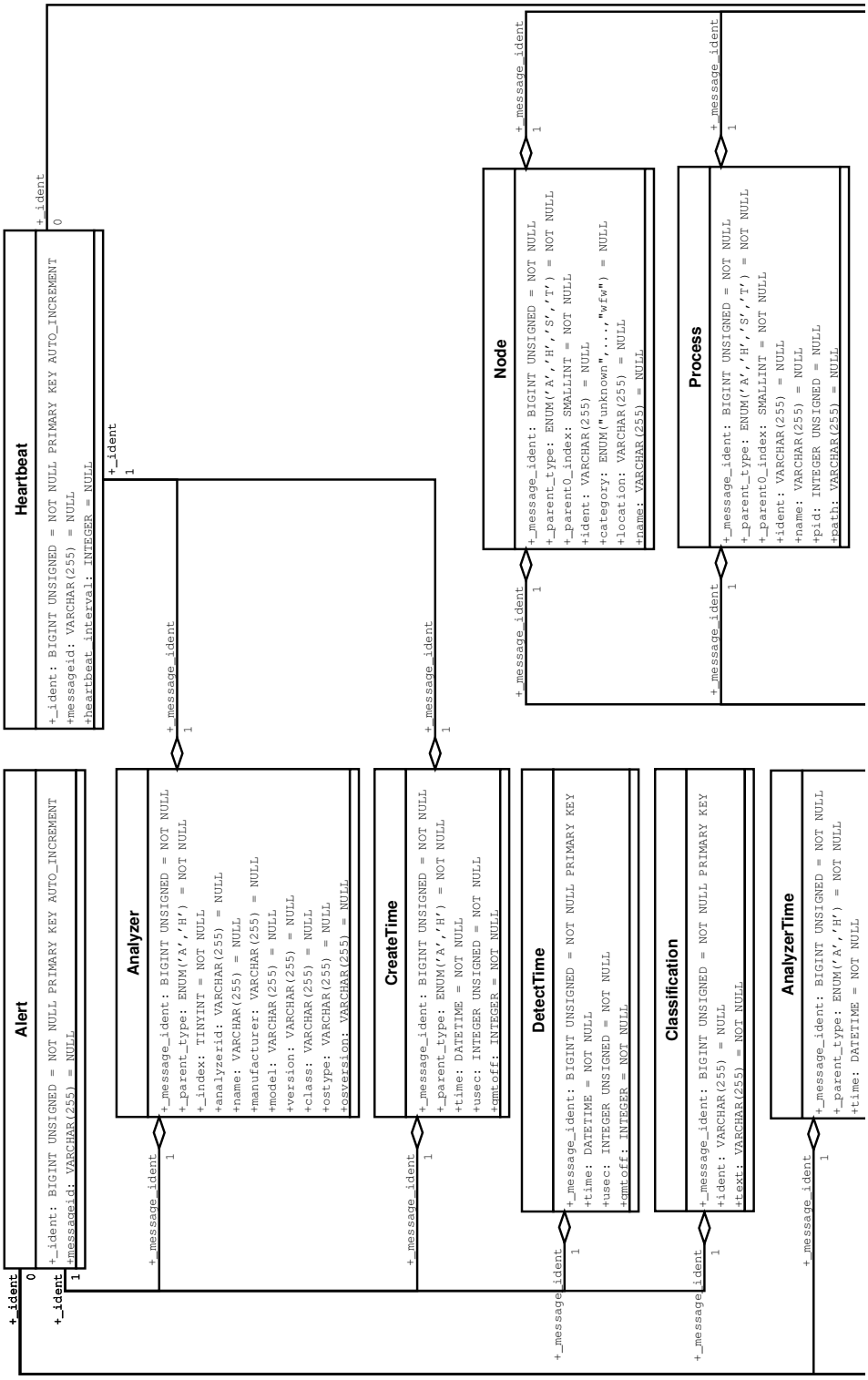


Abbildung 6.3**:** Datenbankschema für IDMEF-Nachrichten (Ausschnitt) – Teil 1/2

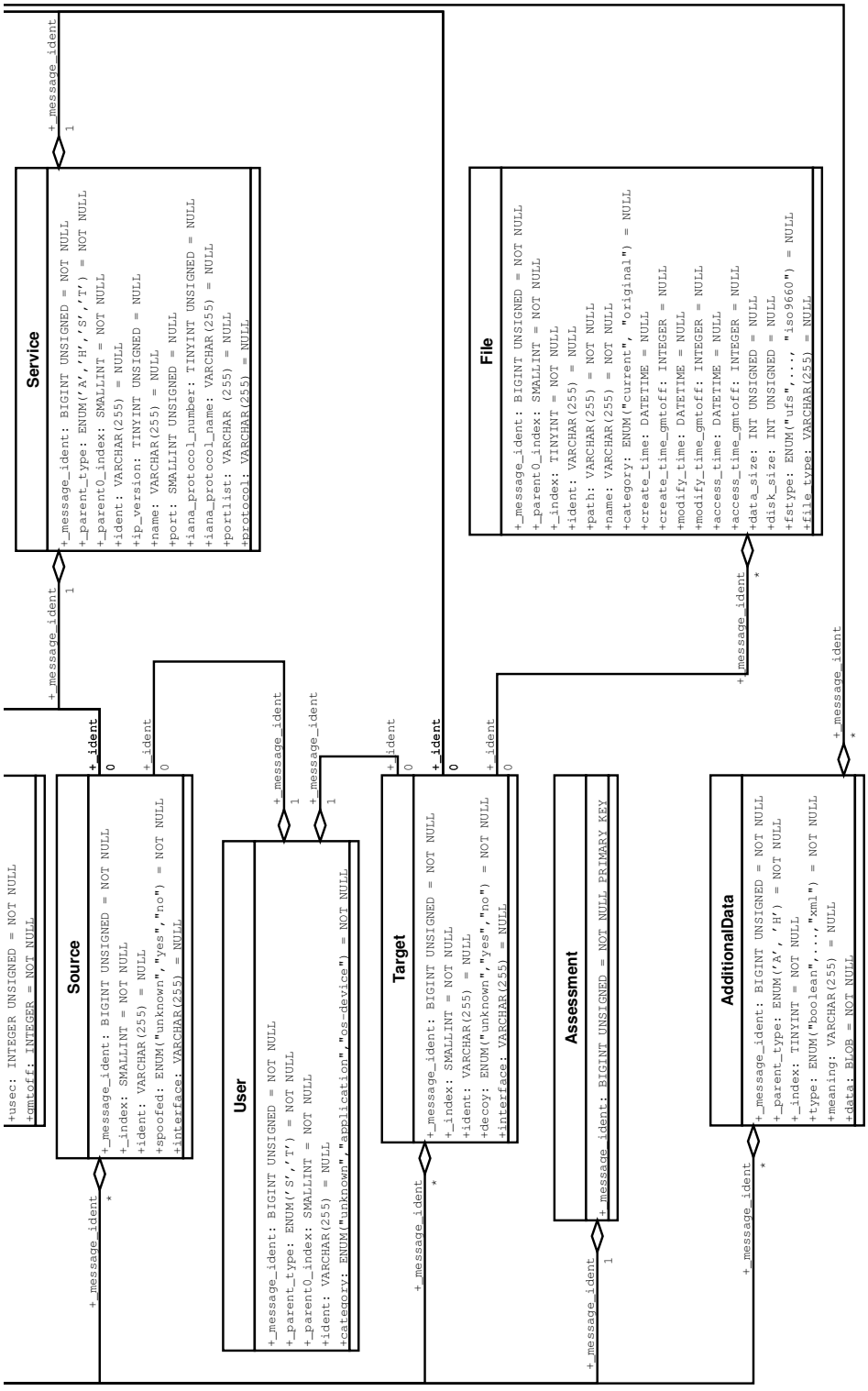


Abbildung 6.4**:** Datenbankschema für IDMEF-Nachrichten (Ausschnitt) – Teil 2/2

vorliegende Nachrichten zu speichern. Das Schema ist in Anlehnung an das im Projekt *Prelude-IDS* [Prelude-IDS] entwickelte *libpreludedb* entstanden. Ein SQL-Skript zur Erzeugung des vollständigen Datenbankschemas innerhalb einer *MySQL*-Datenbank (`http://www.mysql.de/`) ist in Listing A.1 ab Seite 169 in Anhang A zu finden.

Die Idee, die der Abbildung eines XML-basierten Nachrichtenformats auf ein Datenbankschema zu Grunde liegt, ist, dass ein jeder Knoten im XML-Dokument, der sich weiter verzweigen kann, durch eine Tabelle in der Datenbank dargestellt wird. Der Zusammenhang unterhalb der einzelnen Einträge in den jeweiligen Tabellen wird durch die Primärschlüssel der Datensätze (hier mit *_message_ident* bezeichnet) realisiert. Somit können auch Kardinalitäten > 1 abgebildet werden, und es entsteht durch die Speicherung der XML-inhärenten Baumstruktur in einer relationalen Datenbank kein Informationsverlust. Die Abbildungen 6.3 und 6.4 präsentieren die Idee anschaulich.

6.1.1.4 Nachrichtenfilter

An erster Stelle der Kaskade an Komponenten, die eine IDMEF-Nachricht vor ihrer Veröffentlichung durchläuft, steht ein Nachrichtenfilter, der eingehende Nachrichten nach zuvor festgelegten Regeln verwirft. Zur Spezifikation der Kriterien, die das Verwerfen einer Nachricht festlegen, kommen reguläre Ausdrücke zum Einsatz.

Listing D.1, zu finden in Anhang D auf Seite 185, zeigt eine Implementierung eines solchen Filters unter Einsatz von Standard-Werkzeugen, hier *grep*. Der Filter liest seine Eingabe entweder aus einer Datei oder besser (weil performanter) aus einer *named pipe*. Die Ausgabe hingegen wird an die Standardausgabe `stdout` gesendet und kann somit trivialer Weise wiederum in eine Datei oder *named pipe* umgeleitet werden. Die Variable `FILTER` spezifiziert den regulären Ausdruck in der Syntax einer *regexp*, der, wenn er zutrifft, das Verwerfen einer eingehenden Nachricht bewirkt. Zum Beispiel verwirft der Filter

```
# filter alle Nachrichten, die die IP-Adresse 192.168.*.* enthalten
FILTER="<idmef:address>192\.168\.[0-9]*\.[0-9]*</idmef:address>"
```

alle Nachrichten, die eine IP-Adresse aus dem Netz `192.168.0.0/16` enthalten. Der Filter

```
# filter alle Nachrichten, die die IP-Adresse 10.*.*.* enthalten
FILTER="<idmef:address>10\.[0-9]*\.[0-9]*\.[0-9]*</idmef:address>"
```

hingegen verwirft alle Nachrichten, die eine IP-Adresse aus dem Netz `10.0.0.0/8` enthalten.

6.1.1.5 Event-Korrelation

Der Schritt der Event-Korrelation dient zum einen der zusätzlichen Datenverdichtung, zum anderen kann durch sinnvoll gewählte Korrelationsvorschriften ein Informationsgewinn erzielt werden. Ein Beispiel für eine sinnvolle und zugleich datenverdichtenden Korrelation ist das Zusammenfassen mehrerer fehlgeschlagener Anmeldeversuche an einem überwachten

System. Dabei sind Kriterien für ein Zusammenfassen beispielsweise das mehrfache falsche Anmelden unter einem Nutzernamen, ggf. jedoch auch von verschiedenen Maschinen aus. Zum anderen könnten aber auch fehlerhafte Anmeldeversuche unter verschiedenen Nutzernamen, jedoch jeweils vom gleichen Quellrechner aus zusammengefasst werden. Beide Korrelationsregeln sind informationsverlustfrei, einzig die zeitliche Dimension wird durch ein zusammengefasstes und somit zeitlich verzögertes Melden von diesem Vorgang berührt.

Zur Korrelation der Daten im GIDS kommt der *Simple Event Correlator* (SEC) [Simple Event Correlator] zum Einsatz. Der Simple Event Correlator ist ein quelloffenes Projekt und in der Programmiersprache Perl geschrieben. Das Ziel des SEC ist es, Disziplinen wie das Netzmanagement, das Sicherheitsmanagement oder auch die Systemüberwachung auf Basis von Log-Daten zu unterstützen. Listing E.1 im Anhang stellt dar, wie der SEC in das GIDS eingebunden werden kann. Dazu wird grundlegend basierend auf einer *Extensible Stylesheet Language Transformation* (XSLT) unter Nutzung des *xsltproc* eine Vorverarbeitung der XML-Nachrichten durchgeführt. Dies erleichtert zum einen die Arbeit und zum anderen die Konfiguration des SEC.

Prinzipiell arbeitet der SEC zweistufig: Zuerst werden für die Korrelation relevante Nachrichten gefiltert, anschließend wird die Korrelation durchgeführt. Die Filterung der relevanten Nachrichten jedoch kann in hiesigem Fall bereits durch die Vorverarbeitung mit Hilfe der XSL Transformation durchgeführt werden, so dass der SEC zusätzlich entlastet und der Durchsatz gesteigert werden kann.

Listing 6.3**:** Korrelation fehlerhafter SSH-Anmeldeversuche [Schlamp, 2008]

```
# first context creation
type=Single
ptype=RegExp
pattern=^ACCESS, \S+, \S+, sshd, (\S+), \S+, \S+, \S+, (.*)
context=!REMLOGIN_FROM_$1 && !COR_REMLOGIN_FROM_$1
continue=TakeNext
desc=see comment above
action=eval %o ( $remlogin{"$1"} = {} ); \
       eval %o ( $remlogin{"$1"}->{"time"} = %u ); \
       create REMLOGIN_FROM_$1 900 eval %o ( delete $remlogin{"$1"} )

# saving events if correlating alert doesnt exist yet
type=Single
ptype=RegExp
pattern=^ACCESS, \S+, \S+, sshd, (\S+), \S+, \S+, \S+, (\S+), (.*)
context=REMLOGIN_FROM_$1 && !COR_REMLOGIN_FROM_$1
continue=TakeNext
desc=see comment above
action=eval %o ( $remlogin{"$1"}->{"$2"} = "$3" ); \
       set REMLOGIN_FROM_$1 900 eval %o ( delete $remlogin{"$1"} )

# putting initial alert and new context creation
type=Single
ptype=RegExp
pattern=^ACCESS, \S+, \S+, sshd, (\S+), \S+, (\S+), \S+, \S+, (.*)
context=REMLOGIN_FROM_$1 && !COR_REMLOGIN_FROM_$1 && \
        =( scalar(keys(%{$remlogin{"$1"}})) > $thresholds{"remlogin"}->{"prio2"} )
continue=DontCont
desc=see comment above
action=eval %x ( putAlert($remlogin{"$1"},"remote access fails",0,2, \
```

```
                "$1","$2","22",$thresholds{"remlogin"}->{"prio2"}) ); \
32      delete REMLOGIN_FROM_$1; \
        create COR_REMLOGIN_FROM_$1 300 eval %o ( delete $remlogin{"$1"} ); \
34      event COR_REMLOGIN, 2, $2, SEC, $1, NONE, $2, 22, %x

36 # saving events and updating existing alert
type=Single
38 ptype=RegExp
pattern=^ACCESS, \S+, \S+, sshd, (\S+), \S+, \S+, \S+, (\S+), (.*)
40 context=COR_REMLOGIN_FROM_$1
continue=DontCont
42 desc=see comment above
action=eval %o ( updateAlert($remlogin{"$1"}->{"ID"}, \
44              "$2","$3",$thresholds{"remlogin"}) ); \
        set COR_REMLOGIN_FROM_$1 300 eval %o ( delete $remlogin{"$1"} )
```

Listing 6.3 zeigt ein Konfigurationsbeispiel um fehlerhafte SSH-Anmeldeversuche mit SEC zu korrelieren. Dazu wird eine beim SEC eingehende Nachricht zuerst darauf hin überprüft, ob es sich um die Meldung eines fehlerhaften Anmeldeversuches handelt. Dies wird auf Basis eines zu spezifizierenden regulären Ausdrucks durchgeführt. Trifft die Prüfung zu, so wird überprüft, ob bereits ein fehlerhafter Anmeldeversuch des gleichen Nutzers oder vom gleichen Quellrechner aus existiert. Wenn ja, so wird der Zähler der fehlerhaften Anmeldeversuche um eins erhöht. Wenn nicht, so wird ein Zustand, der diese Meldung repräsentiert, erzeugt. Übersteigt ein Zähler einen spezifizierten Grenzwert, so wird eine korrelierte Alarmmeldung erzeugt.

Ein Vorteil von SEC ist, dass durch ein geschicktes Wiedereinspeisen erzeugter Korrelationsalarme eine weitere Korrelation dieser Alarme durchgeführt werden kann. Zum Beispiel kann somit der Schweregrad eines Alarm erhöht werden, wenn eine gewisse Anzahl bereits korrelierter fehlerhafter Anmeldeversuche wieder im Simple Event Correlator eingegangen sind.

6.1.1.6 Ein Anonymisierer/Pseudonymisierer

Für die Komponente eines Anonymisierers und/oder Pseudonymisierers werden im Folgenden zwei alternative Implementierungen vorgestellt. Zum einen werden reguläre Ausdrücke und referenzierende Ersetzungen verwendet, zum anderen kommt eine *Extensible Stylesheet Language Transformation* (XSLT) zum Einsatz.

6.1.1.6.1 Nutzung regulärer Ausdrücke Da in hiesigem GIDS ein Nachrichtenaustausch auf Basis des XML-basierten IDMEF erfolgt, liegt es nahe eine Transformation der Nachrichten mit dem Ziel der Anonymisierung und/oder Pseudonymisierung bestimmter in den Nachrichten enthaltener Daten mit Hilfe einer *Extensible Stylesheet Language Transformation* (XSLT) zu realisieren. Jedoch werden XML-basierten Nachrichten zumeist in einer menschenlesbaren Form als Text-String übermittelt, was die Möglichkeit eröffnet, dass eine Transformation des Textstroms auch unter Nutzung regulärer Ausdrücke und ggf. referenzierende Ersetzungen vorgenommen werden kann. Dadurch entsteht eine in der

Komplexität linear von der Länge der Nachricht abhängige Bearbeitungszeit, da ein einmaliges Abarbeiten ohne weitere Interpretation der XML-Nachricht notwendig ist.

Listing F.1 in Anhang F stellt einen Anonymisierer/Pseudonymisierer unter Nutzung regulärer Ausdrücke auf Basis eines Strom-Editors vor. Für eine Transformationsvorschrift sind bei der Konfiguration dieser Komponente zwei reguläre Ausdrücke in Form einer *regexp* notwendig. In der Variable `MATCH` muss der zu suchende Teil einer eingehenden Nachricht (z.B. eine IP-Adresse oder ein Port) spezifiziert werden. Durch die Variable `SUB` wird die Ersetzung des gefundenen Textteils angegeben. Dieser kann ebenfalls Teile des zuvor gesuchten und gefundenen Textteils referenzieren.

Ein Beispiel für eine Konfiguration sind folgende Einstellungen:

```
# ersetze in allen Nachrichten, die die IP-Adresse *.*.*.* enthalten,
#                          die IP durch die Adresse "*.0.0.0"
#
# regexp, gegen die geprüft werden soll
MATCH="\([0-9]*\)\.[0-9]*\.[0-9]*\.[0-9]*"
# Ersetzung durch...
SUB="\1\.0\.0\.0"
```

In diesem Fall wird nur das führende Byte einer jeden in einer Nachricht enthaltenen IP-Adresse beibehalten, die verbleibenden drei Byte werden auf Null gesetzt. Hierbei handelt es sich schon um eine extreme Art der Anonymisierung und Pseudonymisierung, die in den meisten Fällen nicht einmal Rückschlüsse auf das (Sub-)Netz zulässt, aus dem die abgeänderte IP-Adresse stammt.

6.1.1.6.2 Nutzung von XSLT Die typische Art und Weise XML-basierte Datenformate zu bearbeiten (oder auch zu „transformieren“) ist mit Hilfe der *Extensible Stylesheet Language Transformation* (XSLT) [XSLT, 1999]. Dazu muss eine sogenanntes *Stylesheet* in der *Extensible Stylesheet Language* (XSL) formuliert und auf ein XML-Dokument angewendet werden, was als *Transformation* bezeichnet wird. Zur Ausführung solcher Transformationen steht eine Vielzahl sogenannter *XSLT Prozessoren* zur Verfügung, unter anderem die *Microsoft XML Core Services*, das *.NET Framework*, die PHP XSL Funktionen (ab Version 5) oder auch der im Weiteren eingesetzte, frei verfügbare *xsltproc*, der in der Programmiersprache C geschrieben und hoch performant ausgelegt ist.

Um den xsltproc im Rahmen des hier vorgestellten GIDS sinnvoll einsetzen zu können, wird eine Einbettung als Komponente des GIDS benötigt. Listing F.2in Anhang F zeigt diese exemplarisch.

Listing 6.4: Ein beispielhaftes XSLT-Stylesheet

```
1 <xsl:stylesheet version="1.0"
                  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3                 xmlns:idmef="http://iana.org/idmef">

5  <xsl:template match="/">
```

```
    <xsl:apply-templates select="node()"/>
  </xsl:template>

  <xsl:template match="*">
    <xsl:choose>
      <xsl:when test="name() = 'idmef:address'">
        <xsl:choose>
          <xsl:when test="starts-with(text(),'192.168')">
            <idmef:address>192.168.*.*</idmef:address>
          </xsl:when>
          <xsl:otherwise>
            <idmef:address>
              <xsl:value-of select="."/>
            </idmef:address>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:when>
      <xsl:otherwise>
        <xsl:element name="{name()}">
          <xsl:apply-templates select="@*"/>
          <xsl:apply-templates select="node()"/>
        </xsl:element>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>

  <xsl:template match="@*">
    <xsl:attribute name="{name()}">
      <xsl:value-of select="."/>
    </xsl:attribute>
  </xsl:template>

  <xsl:template match="text()">
    <xsl:value-of select="." />
  </xsl:template>

</xsl:stylesheet>
```

Die eigentliche Konfiguration des XSLT-basierten Anonymisierers/Pseudonymisierers geschieht durch die Spezifikation eines geeigneten Stylesheets. Listing 6.4 zeigt beispielhaft ein solches XSLT-Stylesheet. In diesem Beispiel wird eine Anonymisierung aller internen IP-Adressen, die mit dem Präfix '`192.168.`' beginnen, realisiert. Wenn ein solcher Eintrag in einem Knoten des Typs '`<idmef:address>`' gefunden wird, so werden die beiden letzten Bytes der IP-Adresse durch Sternchen ersetzt. Dabei ist jedoch zu beachten, dass z.B. in einem beschreibenden Text innerhalb der IDMEF-Nachricht auftauchende IP-Adressen durch dieses Stylesheet, im Gegensatz zum Ansatz reguläre Ausdrücke zu verwenden, nicht erfasst werden. Solche Erweiterungen müssen je nach Bedarf noch hinzugefügt werden.

6.1.1.7 Kommunikation durch GIDS-Agenten

Die GIDS-Agenten bilden zentrale Stellen im Gesamtsystem, da sie die Kommunikation unterhalb der kooperierenden Partner sicherstellen. Dabei ist insbesondere die Nachrichtenintegrität, -authentizität und -vertraulichkeit zu gewährleisten.

Im Grid-Umfeld gibt es etablierte Kommunikationswege, die alle den Anforderungen eines GIDS-Agenten gerecht werden würden. Dennoch fällt bei dieser Implementierung die Wahl auf eine Lösung, die auf ein Virtuelles Privates Netz (VPN) baut. Durch eine solche logische Struktur, die auf die vorhandene physische Netzstruktur aufsetzt, lässt sich eine exklusiv von den GIDS-Agenten genutzte Broad- und Multicast-Umgebung schaffen, was durch den Anwendungsfall der GIDS-globalen Informationsverbreitung sehr sinnvoll ist.

In hiesigem Fall kommt für die Realisierung eines gesicherten VPNs die freie Software *OpenVPN* [OpenVPN] zum Einsatz. Dabei handelt es sich um eine VPN-Lösung, die auf einen SSL/TLS-Tunnel (*Secure Socket Layer/ Transport Layer Security*) setzt. OpenVPN ist eine leichtgewichtige und dadurch performante Software, die für eine Vielzahl an Betriebssystemen zur Verfügung steht, darunter auch Linux, Solaris, OpenBSD, FreeBSD, NetBSD, Mac OS X und Windows 2000/XP.

Durch die Basis von SSL/TLS kann eine auf OpenVPN basierende Infrastruktur nahtlos in die vorhandene Grid-Welt eingefügt werden. Es werden dieselben kryptographischen Verfahren verwendet und auch eine Integration in eine auf X.509-Zertifikaten basierende Public Key Infrastruktur ist problemlos möglich. Dadurch können ohne Einschränkungen allen an einen GIDS-Agenten gestellten Sicherheitsanforderungen genüge getan werden.

OpenVPN bietet zwei Arten des Tunnels an – es kann sowohl einen OSI-Schicht 2 wie auch OSI-Schicht 3 Tunnel bereitstellen. Aus Sicht des Betriebssystems wird lediglich ein weiterer (virtueller) Netzadapter hinzugefügt. Aus Gründen der Performanz und um den Overhead zu minimieren, wird hier ein OSI-Schicht 3 Tunnel eingesetzt.

6.1.2 Implementierungsvorschlag auf Seiten des Betreibers des GIDS

Einen Großteil der für den Betrieb eines GIDS als Dienst notwendigen Komponenten ist äquivalent zu den bei den Ressourcenanbietern eingesetzten Komponenten und somit sind auch deren Implementierungen wieder verwendbar (siehe auch Abschnitt 5.2 mit den Abbildungen 5.3 und 5.4). Für die Diensterbringung jedoch sind zusätzlich die beiden Komponenten des Benutzerportals und zur proaktiven Benachrichtigung zu Sicherheitsvorfällen notwendig, die im Weiteren näher ausgeführt werden.

6.1.2.1 Benutzerportal

Das Hauptziel, das mit der Konzeption eines Benutzerportals verfolgt wird, ist das Anbieten eines neuartigen Dienstes im Grid. Dazu ist es notwendig, dass nutzer- und kundenspezifische Sichten auf die Sicherheitsberichte, die durch das GIDS generiert worden sind, realisiert werden.

Die nachfolgenden Abschnitte demonstrieren prototypisch, wie eine solche Sichtenbildung am Beispiel des D-Grid umgesetzt werden kann. Dazu werden zuerst die notwendigen und

bereits im Grid verfügbaren Informationen und deren Quelle erörtert, bevor anschließend die tatsächliche Sichtenbildung beispielhaft demonstriert wird.

6.1.2.1.1 Verfügbare Informationen am Beispiel des D-Grid Die für die Realisierung des Nutzerportals notwendigen Informationen, die zur Sichtenbildung auf die GIDS-Berichte dienen, finden sich im Rahmen des D-Grid in Datenbankabbildungen der im *Virtual Organization Membership Service* (VOMS) [VO Membership Service]. Dazu sind insbesondere drei Tabellen, die in einer MySQL-Datenbank hinterlegt sind, von Relevanz.

Zum einen wird die Information aus der Tabelle *dgrid_dn_vo* benötigt, die wie folgt aufgebaut ist:

```
mysql> describe dgrid_dn_vo;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| member_id     | int(38)     | YES  |     | NULL    |       |
| vorname       | varchar(50) | YES  |     | NULL    |       |
| nachname      | varchar(100)| YES  |     | NULL    |       |
| member_dn     | varchar(200)| YES  |     | NULL    |       |
| member_status | varchar(15) | YES  |     | NULL    |       |
| vo_long       | varchar(15) | YES  |     | NULL    |       |
| vo_short      | varchar(2)  | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
7 rows in set (0.02 sec)
```

Hier finden sich die Informationen zu den Nutzern des Grid mit spezifischen Angaben zur Person wie Vor- und Nachname sowie die ihnen zugeordnete Identität im D-Grid. Zudem wird eine Zuordnung von Nutzern zu VOs geleistet, wobei zum einen eine menschenlesbare Angabe zur VO (*vo_long*) sowie zum anderen ein die VO eindeutig identifizierendes Kürzel (*vo_short*) gespeichert sind. Ist ein Nutzer Mitglied mehrerer VOs, so wird die Mitgliedschaft in jeder VO ein eigener Datensatz in dieser Tabelle angelegt. Dadurch sind Abfragen wie „*Welche Nutzer gehören einer bestimmten VO an?*“ oder „*Welchen VOs ist ein bestimmter Nutzer zugehörig?*“ trivial lösbar.

Die Tabelle *dgrid_ressourcen* hält Informationen zu den einzelnen im D-Grid verfügbaren Ressourcen und ihrer Verfügbarkeit im Rahmen der verschiedenen Middleware-Konzepte bereit.

```
mysql> describe dgrid_ressourcen;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| ress_key        | int(38)     | YES  |     | NULL    |       |
| dr_kind         | varchar(20) | YES  |     | NULL    |       |
```

```
| dr_scunicore     | varchar(150) | YES  |     | NULL    |       |
| dr_scglobus      | varchar(150) | YES  |     | NULL    |       |
| dr_scglite       | varchar(150) | YES  |     | NULL    |       |
| dr_scogsa        | varchar(150) | YES  |     | NULL    |       |
| dr_scdcache      | varchar(150) | YES  |     | NULL    |       |
| dr_scinteract    | varchar(150) | YES  |     | NULL    |       |
| dr_scadmin       | varchar(150) | YES  |     | NULL    |       |
| dr_scglobus2     | varchar(150) | YES  |     | NULL    |       |
| dr_shorty        | varchar(40)  | YES  |     | NULL    |       |
| dr_gram_node     | varchar(100) | YES  |     | NULL    |       |
| dr_glite_node    | varchar(100) | YES  |     | NULL    |       |
| dr_interact_node | varchar(100) | YES  |     | NULL    |       |
| dr_gram_node2    | varchar(100) | YES  |     | NULL    |       |
| dr_soinvest      | varchar(1)   | YES  |     | NULL    |       |
| dr_unicore_vers  | varchar(1)   | YES  |     | NULL    |       |
| dr_admin_mail    | varchar(60)  | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
18 rows in set (0.01 sec)
```

Das Feld *ress_key* ist ein eine Ressource eindeutig im D-Grid identifizierender Schlüssel, der im Folgenden im Falle des GIDS noch von Bedeutung sein wird. Weitere notwendige Informationen sind die zu den Ressourcen gespeicherten Daten, insbesondere zu deren Zertifikaten und einem administrativen Kontakt (*dr_admin_mail*). Über das Zertifikat, oder genauer den *Distinguished Name* (DN), lässt sich auf den DNS-Namen einer Ressource schließen und darüber wiederum auf deren IP-Adresse. Dies ist notwendig, da vielmals im Zuge eines IDS-Berichts IP-Adressen verwendet werden, wodurch sich somit eine Abbildung auf die in dieser Datenbanktabelle hinterlegten Informationen vornehmen lässt.

Um einen Zusammenhang zwischen den Nutzern und ihren VOs und den zur Verwendung freigegebenen Ressourcen vornehmen zu können, existiert eine weitere Tabelle namens *dgrid_ress_vo*.

```
mysql> describe dgrid_ress_vo;
+--------------+------------+------+-----+---------+-------+
| Field        | Type       | Null | Key | Default | Extra |
+--------------+------------+------+-----+---------+-------+
| drv_ress_key | int(38)    | YES  |     | NULL    |       |
| drv_vo       | varchar(2) | YES  |     | NULL    |       |
+--------------+------------+------+-----+---------+-------+
2 rows in set (0.02 sec)
```

Das Feld *drv_ress_key* referenziert dabei das Feld *ress_key* aus der Tabelle *dgrid_ressourcen* und das Feld *drv_vo* referenziert das Feld *vo_short* aus der Tabelle *dgrid_dn_vo*. Somit fungiert diese Tabelle einzig als Bindeglied zwischen den Zuordnungen von Nutzern bzw.

VOs zu den für sie zur Verwendung freigegebenen Ressourcen und lässt eine fast beliebige Anfrage in diesem Kontext zu.

6.1.2.1.2 Nutzerspezifische Sichtenbildung Die zuvorstehend beschriebenen Informationen lassen sich nun in Zusammenhang mit den GIDS-Berichten bringen. Listing 6.5 zeigt dazu ein beispielhaftes Programm, das als Eingabe das in der VOMS-Datenbank gespeicherte Kürzel für eine VO erwartet. Als Ausgabe werden alle verfügbaren GIDS-Berichte, die die VO tangieren, geliefert.

Listing 6.5**:** Sichtenbildung auf die Sicherheitsberichte für eine ausgewählte VO

```
#!/bin/bash

### Begin ###

# "externe" Config einlesen
. ./config

# fragt VOMS-DB an ($1 := Anfrage)
function queryVOMSDB () {
  DBQUERY="$1"
  echo "$DBQUERY" | mysql --column-names=0 --host=$VOMSHOST \
                          --user=$VOMSUSER --password=$VOMSPASS \
                          --database=$VOMSNAME
}

# fragt GIDS-DB an ($1 := Anfrage)
function queryGIDSDB () {
  DBQUERY="$1"
  echo "$DBQUERY" | mysql --column-names=1 --host=$DBHOST \
                          --user=$DBUSER --password=$DBPASS \
                          --database=$DBNAME
}

# gibt die Ressourcen einer VO zurück
# $1 := Kürzel der VO
# z.B. in ~ InGrid
#      c3 ~ C3-Grid
#      md ~ MediGRID
# Rückgabe: Liste an IP-Adressen
function getResources () {
  DN=$(queryVOMSDB "SELECT dr_scinteract \
          FROM dgrid_ress_vo \
                JOIN dgrid_ressourcen \
                ON (dgrid_ress_vo.drv_ress_key=dgrid_ressourcen.ress_key) \
          WHERE dgrid_ress_vo.drv_vo=\"$1\";");

  CN=""
  for i in $DN ; do
    CN="$CN $(echo $i | sed -e 's/.*\(CN=.*\)$/\1/' | grep ^CN=)"
  done

  IP=""
  for i in $CN ; do
    IP="$IP $(host $(echo $i | sed -e 's/^CN=//' | sed -e 's/^host\///') | \
              grep "has address [0-9]*.[0-9]*.[0-9]*.[0-9]*" | \
              sed -e 's/.*address\ \([0-9]*.[0-9]*.[0-9]*.[0-9]*\)/\1/')"
  done

```

```
  echo $IP
}

# gibt alle Alarme aus, die eine von einer gegebenen VO Ressource
# - entweder als Quelle
# - oder als Ziel beinhalten
# $1 := Kürzel der VO
# ACHTUNG: Unterabfragen erst ab MySQL v4.1 verfügbar!
function getAlertsForVO () {
  LIST="$(getResources "$1" | sort --field-separator=" " --unique)"
  LIST="$LIST 192.168.1.10"
  for i in $LIST ; do

    QUERYSRC="SELECT DISTINCT Analyzer._message_ident as ID, Analyzer.name as Analyzer, \
                     Service.protocol as Protocol, Address.address as SrcIP, \
                     Address.netmask as SrcMask, Service.port as SrcPort, \
                     Classification.text as text \
              FROM Analyzer, Service, Address, Classification \
              WHERE Analyzer._message_ident=Address._message_ident \
                    AND Analyzer._message_ident=Service._message_ident \
                    AND Analyzer._message_ident=Classification._message_ident \
                    AND Address.address=\"$i\" \
                    AND Address._parent_type=\"S\""

    QUERYTGT="SELECT DISTINCT Analyzer._message_ident as ID, Analyzer.name as Analyzer, \
                     Service.protocol as Protocol, Address.address as TrgtIP, \
                     Address.netmask as TrgtMask, Service.port as TrgtPort, \
                     Classification.text as text \
              FROM Analyzer, Service, Address, Classification \
              WHERE Analyzer._message_ident=Address._message_ident \
                    AND Analyzer._message_ident=Service._message_ident \
                    AND Analyzer._message_ident=Classification._message_ident \
                    AND Address.address=\"$i\" \
                    AND Address._parent_type=\"T\""

    QUERY="SELECT trgt.*, src.SrcIP, src.SrcMask \
           FROM ($QUERYTGT) AS trgt \
           JOIN (SELECT DISTINCT Address._message_ident as ID, \
                                 Address.address as SrcIP, Address.netmask as SrcMask, \
                                 Service.port as SrcPort \
                 FROM Address, Service \
                 WHERE Address._parent_type=\"S\" \
                       AND Address._message_ident=Service._message_ident) AS src \
           ON src.ID=trgt.ID; \
           SELECT src.*, trgt.TrgtIP, trgt.TrgtMask \
           FROM ($QUERYSRC) AS src \
           JOIN (SELECT DISTINCT Address._message_ident as ID, \
                                 Address.address as TrgtIP, Address.netmask as TrgtMask, \
                                 Service.port as TrgtPort \
                 FROM Address, Service \
                 WHERE Address._parent_type=\"T\" \
                       AND Address._message_ident=Service._message_ident) AS trgt \
           ON src.ID=trgt.ID;"

    echo "Alarmmeldungen für $i:"
    queryGIDSDB "$QUERY"
  done
}

### Main ###
echo -n "Bitte geben Sie das Kürzel der abzufragenden VO ein: "
read VO
getAlertsForVO "$VO"
```

Im Wesentlichen ist die tatsächliche Funktionalität in den beiden Funktionen *getResources* und *getAlertsForVO* ausgelagert.

function getResources (). Diese Funktion fragt die VO Management Services nach allen einer VO zur Verfügung stehenden Ressourcen an. Dazu erwartet sie als Parameter das Kürzel der VO von Interesse, welches als Auswahlkriterium innerhalb des Joins der beiden Tabellen *dgrid_ress_vo* und *dgrid_ressourcen* dient. Die Anfrage selektiert als einziges Merkmal der Ressourcen ihren Distinguished Name (DN), der u.a. einen DNS-Namen der bezeichneten Ressource enthält. Die auf die Anfrage folgenden Zeilen extrahieren zuerst den DNS-Namen, der anschließend durch eine Anfrage des *Domain Name Service*s auf die korrespondierende IP-Adresse abgebildet wird. Schlussendlich gibt die Funktion eine Liste durch Leerzeichen separierter IP-Adressen der Ressourcen zurück.

function getAlertsForVO (). Diese Funktion erwartet als Übergabewert ebenfalls das Kürzel der VO von Interesse, mit Hilfe dessen die Funktion *getResources* aufgerufen wird. Somit steht eine Liste an IP-Adressen zur Verfügung, nach denen in den in der GIDS-Datenbank gespeicherten Berichten gesucht werden soll. Die Herausforderung dabei besteht darin, dass ein jeder Bericht zwei Einträge in der Tabelle *Address* und einigen anderen (jeweils eine Quell- und eine Zieladresse) referenziert. Aus diesem Grund ist es nicht trivial möglich einen geeigneten Join zu formulieren. Seit der MySQL Version 4.1 sind jedoch verschachtelte Unterabfragen möglich, die in diesem Fall hilfreich angewendet werden können. So lassen sich zwei Anfragen formulieren, die zum einen nach den gegebenen IP-Adressen in der Liste der Quell- und einmal in der Liste der Zieladressen suchen und die entsprechenden Berichte bzw. hier beispielhaft einige relevante Informationen extrahieren. Anschließend kann ein Join mit der zugehörigen Ziel- bzw. Quelladresse vorgenommen werden, was durch die Verwendung der geschachtelten Unterabfragen erleichtert wird. Alle so gefundenen Berichte werden nach Ressource geordnet und unterhalb nach dem Kriterium „*Ressource als Angriffsziel*“ und „*Ressource als Ursprung des Angriffs*“ sortiert ausgegeben.

Natürlich lässt sich dieses sehr einfache Beispiel durch eine beliebig komplexe Anfrage ersetzen, wovon an dieser Stelle aus Gründen der Verständlichkeit und des Umfangs abgesehen wird. So lassen sich zum Beispiel auch Abfragen zu Berichten aus bestimmten Zeiträumen, Berichten zu einzelnen Ressourcen oder -anbietern und alle weiteren denkbaren Kombinationen formulieren.

6.1.2.2 Proaktive Benachrichtigung

Im Wesentlichen kann eine proaktive Benachrichtigung zu einem Angriff, der entweder eine bestimmte Ressource als Ausgangsort oder als Ziel beinhaltet, analog zur nutzerspezifischen Sichtenbildung vorgenommen werden. Der Ablauf der Anfragen ist jedoch in der Reihenfolge umgekehrt, da aus einer Alarmmeldung auf den verantwortlichen Betreiber geschlossen werden muss.

Listing 6.6: Proaktive Benachrichtigung an die Administratoren

```
#!/bin/bash

### Begin ###

# "externe" Config einlesen
. ./config

# fragt VOMS-DB an ($1 := Anfrage)
function queryVOMSDB () {
  DBQUERY="$1"
  echo "$DBQUERY" | mysql --column-names=0 --host=$VOMSHOST \
                          --user=$VOMSUSER --password=$VOMSPASS \
                          --database=$VOMSNAME
}

# fragt GIDS-DB an ($1 := Anfrage)
function queryGIDSDB () {
  DBQUERY="$1"
  echo "$DBQUERY" | mysql --column-names=0 --host=$DBHOST \
                          --user=$DBUSER --password=$DBPASS \
                          --database=$DBNAME
}

# reverse DNS lookup
# $1 := IP-Adresse der Ressource
function getResourceFromIP () {
  DNS=$(host $1 | grep -v not\ found | \
        sed -e 's/.*domain\ name\ pointer\ \(.*\)\.$/\1/')
  queryVOMSDB "SELECT dr_admin_mail \
               FROM dgrid_ressourcen \
               WHERE dr_scinteract LIKE \"%$DNS%\";"
}

# versendet die letzten Berichte aus der GIDS-DB
# an die verantwortlichen Administratoren
# $1 := Startzeit
# $2 := Endzeit
# ACHTUNG: Unterabfragen erst ab MySQL v4.1 verfügbar!
function sendPastReports () {
  QUERYSRC="SELECT DISTINCT CreateTime.time as Zeitstempel, \
                    Analyzer._message_ident as ID, Analyzer.name as Analyzer, \
                    Service.protocol as Protocol, Address.address as SrcIP, \
                    Address.netmask as SrcMask, Service.port as SrcPort, \
                    Classification.text as text \
            FROM Analyzer, Service, Address, Classification, CreateTime \
            WHERE     CreateTime.time >  \"$1\" \
                  AND CreateTime.time <= \"$2\" \
                  AND CreateTime._message_ident=Analyzer._message_ident \
                  AND Analyzer._message_ident=Address._message_ident \
                  AND Analyzer._message_ident=Service._message_ident \
                  AND Analyzer._message_ident=Classification._message_ident \
                  AND Address._parent_type=\"S\" \
                  AND Service._parent_type=\"S\" ";

  QUERYTGT="SELECT DISTINCT CreateTime.time as Zeitstempel, \
                    Analyzer._message_ident as ID, Analyzer.name as Analyzer, \
                    Service.protocol as Protocol, Address.address as TrgtIP, \
                    Address.netmask as TrgtMask, Service.port as TrgtPort, \
                    Classification.text as text \
            FROM Analyzer, Service, Address, Classification, CreateTime \
            WHERE     CreateTime.time >  \"$1\" \
                  AND CreateTime.time <= \"$2\" \
```

```
63 ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣AND␣CreateTime._message_ident=Analyzer._message_ident␣\
   ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣AND␣Analyzer._message_ident=Address._message_ident␣\
65 ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣AND␣Analyzer._message_ident=Service._message_ident␣\
   ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣AND␣Analyzer._message_ident=Classification._message_ident␣\
67 ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣AND␣Address._parent_type=\"T\"␣\
   ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣AND␣Service._parent_type=\"T\"␣";
69
   for i in $(queryGIDSDB "SELECT␣src.SrcIP␣from␣($QUERYSRC)␣as␣src") \
71             $(queryGIDSDB "SELECT␣tgt.TrgtIP␣from␣($QUERYTGT)␣as␣tgt"); do
     MAIL=$(getResourceFromIP "$i")
73   echo "nail␣-s␣\"GIDS ALERT for $i\"␣$MAIL"
   done
75 }

77 ### Main ###

79 START=$(date +%Y-%m-%d\ %T)
   END="$START"
81
   while true ; do
83   sendPastReports "$START" "$END"
     sleep 60s;
85   END="$START"
     START=$(date +%Y-%m-%d\ %T)
87 done
```

Listing 6.6 zeigt eine rudimentäre Möglichkeit eine proaktive Benachrichtigung zu realisieren. Im Gegensatz zu den anderen Listings werden in diesem Beispiel auf Grund des Umfangs keine vollständigen Berichte generiert und verschickt, lediglich ein Aufruf für das Versenden einer Email wird auf der Standardausgabe ausgegeben. Die Abfragen, um die Email mit Inhalt zu füllen, sind ebenfalls bereits in den Variablen `QUERYSRC` und `QUERYTGT` vorbereitet.

function getResourceFromIP (). Diese Funktion erwartet als Übergabewert eine IP-Adresse einer betroffenen Ressource und liefert auf diese Eingabe hin die Emailadresse des verantwortlichen Administrators zurück. Zu diesem Zweck bedarf des der Möglichkeit eine IP-Adresse auf den damit assoziierten DNS-Eintrag abzubilden, ein *DNS reverse lookup* muss also möglich sein. Dies ist notwendig, da in den VO-Managementsystemen Ressourcen nicht unter Nennung ihrer IP-Adressen abgelegt werden, sondern lediglich über den *Distinguished Name* (DN) ihres Zertifikats auf den DNS-Namen zurückgeschlossen werden kann.

Somit liegt das Vorgehen für diese Funktion auf der Hand: Zuerst muss die übergebene IP-Adresse per *DNS reverse lookup* auf ihren DNS-Namen abgebildet werden, bevor eine Ähnlichkeitsabfrage gegen die im VO-Managementsystem gespeicherten DNs vorgenommen werden kann. Alle dazu notwendigen Informationen sind in hiesigem Beispielszenario in der Tabelle *dgrid_ressourcen* der VOMS abgelegt. Das Ergebnis der Abfrage bzw. die daraus resultierende Emailadresse des verantwortlichen Administrators wird abschließend zurückgegeben.

function sendPastReports (). Mit Hilfe dieser Funktion können alle notwendigen Informationen zu in der GIDS-Datenbank gespeicherten Alarmmeldungen für einen be-

stimmten Zeitraum abgefragt werden. In diesem Beispiel werden die spezifizierten Abfragen jedoch aus Gründen des Umfangs lediglich dazu verwendet, um die Quell- und Ziel-IP-Adressen der betroffenen Systeme abzufragen. Mit Hilfe aller dieser IP-Adressen (egal ob Grid-Ressource oder nicht) wird die Funktion *getResourceFromIP* aufgerufen, die die Emailadresse des korrespondierenden Administrators zurückgibt. Sollte kein Eintrag zu einer Ressource vorliegen (z.B. weil die Ressource nicht am Grid beteiligt ist), so ist die Rückgabe leer. Für jeden erhaltenen Rückgabewert kann anschließend eine Benachrichtigung z.B. per Email versendet werden.

Durch die Basis auf Datenbanken entsteht eine zusätzliche Herausforderung im Falle der proaktiven Benachrichtigung. Im Gegensatz zu einem Benutzerportal, was einer *Pull*-Kommunikation entspricht, handelt es sich hier um eine *Push*-Kommunikation. Datenbanksysteme sehen hierzu zwar die sogenannten *Trigger* vor, diese sind jedoch für hiesige Belange nur sehr umständlich einzusetzen. Aus diesem Grund wird bei der prototypischen Implementierung eine *Push*-Kommunikation durch ein regelmäßiges Abfragen (*Pull*, hier alle 60 Sekunden) der Datenbank simuliert.

6.2 Simulationsergebnisse & Messungen

Um die Leistungsfähigkeit des gesamten GIDS und dessen Einsatzfähigkeit in großen Umgebungen zu zeigen, beschreiben nachfolgende Abschnitte die gesammelten Erfahrungen mit der zuvorstehend beschriebenen prototypischen Implementierung. In erster Instanz werden dazu Durchsatzmessungen der einzelnen Komponenten durchgeführt um ein Gefühl für die Skalierungsmöglichkeiten des GIDS und die benötigten Ressourcen einer Hardware, die später zum Einsatz kommen soll, abzuschätzen. Abschnitt 6.2.2 stellt dann nachfolgend die Erfahrungen im Testbetrieb im Münchener Wissenschaftsnetz und dem Lehrstuhlnetz vor.

6.2.1 Durchsatzmessung

In diesem Abschnitt sollen die theoretischen Grenzen der einzelnen Komponenten getestet werden. Dazu dient die in Abschnitt 6.1 prototypische Implementierung der einzelnen Teilkomponenten.

6.2.1.1 Testszenario

Für die nachfolgenden Leistungstests steht ein Testsystem zur Verfügung, welches über einen Intel® Core™2 Duo Prozessor (Modell E6750, Taktfrequenz 2.66GHz) mit 64KB 1st-Level und 4MB 2nd-Level Cache sowie 2GB Hauptspeicher (DDR2-800) verfügt.

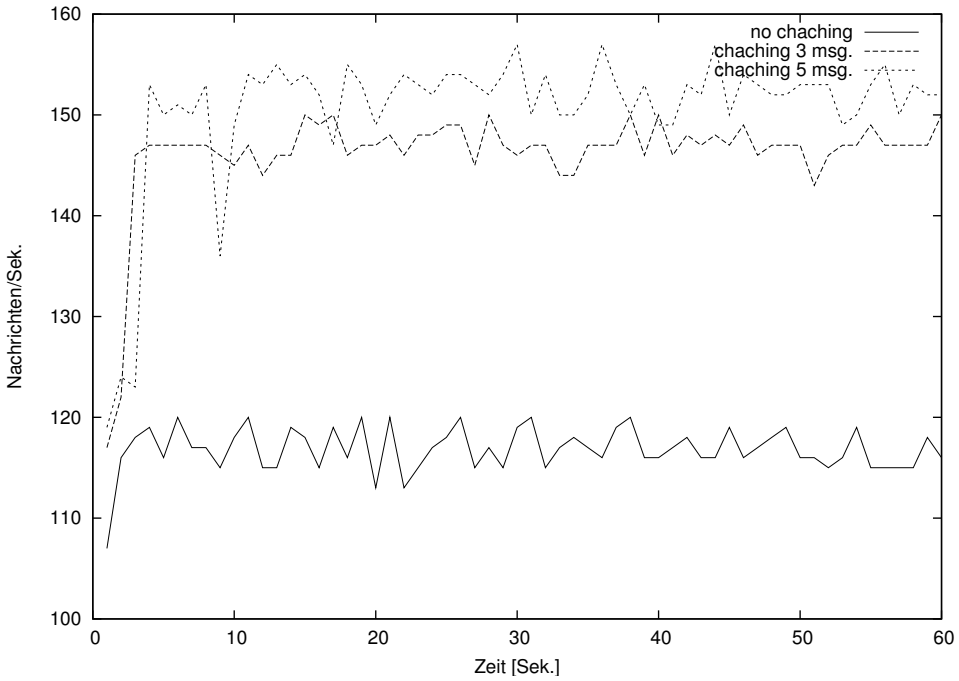


Abbildung 6.5**:** Durchsatzmessung des Dispatchers

Für die Messungen werden jeweils drei verschiedene IDMEF-Nachrichten in den Arbeitsspeicher des Systems geschrieben. Diese Nachrichten werden dann schnellstmöglich nacheinander in sich wiederholender Reihenfolge über eine Pipe an die jeweilige Komponente weitergegeben, die diese verarbeitet und die Ausgabedaten in eine Datensenke schreibt. Der Durchsatz der Anzahl Nachrichten pro Sekunde soll hier als Merkmal der Leistungsfähigkeit dienen. Eine Messreihe erstreckt sich jeweils über einen Zeitraum von 60 Sekunden, wobei jeweils die Anzahl der verarbeiteten Nachrichten pro Sekunde gemessen und graphisch über die Zeit dargestellt wird.

6.2.1.2 Nachrichten-Dispatcher

Um den Durchsatz des Nachrichten-Dispatchers zu messen, kommt eine MySQL-Datenbank der Version 5.0.51a zum Einsatz. Die Installation ist lokal auf dem Testsystem verfügbar, so dass keine durch ein Kommunikationsnetz bedingten Latenzen die Messungen beeinflussen.

Abbildung 6.5 stellt die markantesten Messreihen graphisch dar. Es werden die Messungen für den Durchsatz des Dispatchers ohne Caching sowie mit einem Caching von 3 und 5 Nachrichten dargestellt. Es fällt direkt ins Auge, dass ein gebündeltes Einfügen mehrerer IDMEF-Nachrichten in die Datenbank einen deutlichen Vorteil in Bezug auf die Performanz zur Folge hat. Die charakteristischen Messwerte ausgewählter Messreihen sind in nachfolgender Tabelle nochmal zusammengefasst.

	Nachrichten/Sek.			
Caching von... Nachrichten	*0*	*3*	*5*	*10*
Minimum:	107	117	119	112
Maximum:	120	150	157	148
arithmetisches Mittel:	116.8	146.2	150.5	118.1

Der höchste Durchsatz an Nachrichten pro Sekunde wird mit dem eingesetzten Testsystem bei einem Caching von fünf Nachrichten erzielt. Es fällt jedoch auf, dass bereits ein Zwischenspeichern sehr weniger Nachrichten (z.B. drei) bereits einen deutlichen Gewinn der Performanz bewirkt. Zu hohe Werte wiederum bewirken einen Einbruch der Leistungsfähigkeit, was einen Hinweis darauf liefert, dass im produktiven Einsatz dieser Komponente eine sinnvolle Anpassung der Parameter von Nöten sein kann.

Die Interpretation und Analyse der Ergebnisse lässt den Schluss zu, dass das reine Einfügen einer IDMEF-Nachricht, die bereits als SQL-Anweisungen vorliegt, schneller ist als die XSL Transformation einer IDMEF-Nachricht in eine eben solche SQL-Anweisungen. Das Maximum des Durchsatzes ergibt sich, auch durch die zwei Prozessorkerne bedingt, genau dann, wenn die XSL Transformationen der zwischenzuspeichernden Nachrichten genauso lange dauert wie der Verbindungsaufbau zur Datenbank, das Ausführen der SQL-Anweisungen zum Einfügen der Nachrichten und der abschließende Verbindungsabbau. Bei dem verwendeten Testsystem ist dies bei genau 5 zwischenzuspeichernden Nachrichten der Fall, wenn das System nicht durch andere Applikationen belastet wird.

6.2.1.3 Filter

Abbildung 6.6 stellt die Messergebnisse für die Komponente des Filters graphisch dar. Im Durchschnitt verarbeitet der Filter auf dem Testsystem ca. 370 Nachrichten pro Sekunde, wobei wiederholt bei den Messungen deutliche Ausreißer nach oben festgestellt werden konnten (siehe z.B. Sekunde 27 oder 55). Bei keiner Messung fiel diese Komponente auf einen Durchsatz von unter 356 Nachrichten pro Sekunde, was sie zu einer der performantesten der Komponenten macht. Nachfolgend sind nochmal die bereits graphisch aufbereiteten Kennzahlen der erzielten Messwerte aufgeführt.

	Nachrichten/Sek.
Minimum:	356
Maximum:	457
arithmetisches Mittel:	369.8

Die große Performanz des Filters ist insbesondere auf die Verwendung regulärer Ausdrücke zurückzuführen. Eingehende Nachrichten können als Datenstrom effizient verarbeitet werden und die durch XML gegebene Baumstruktur der IDMEF-Nachricht muss nicht aufwendig interpretiert werden.

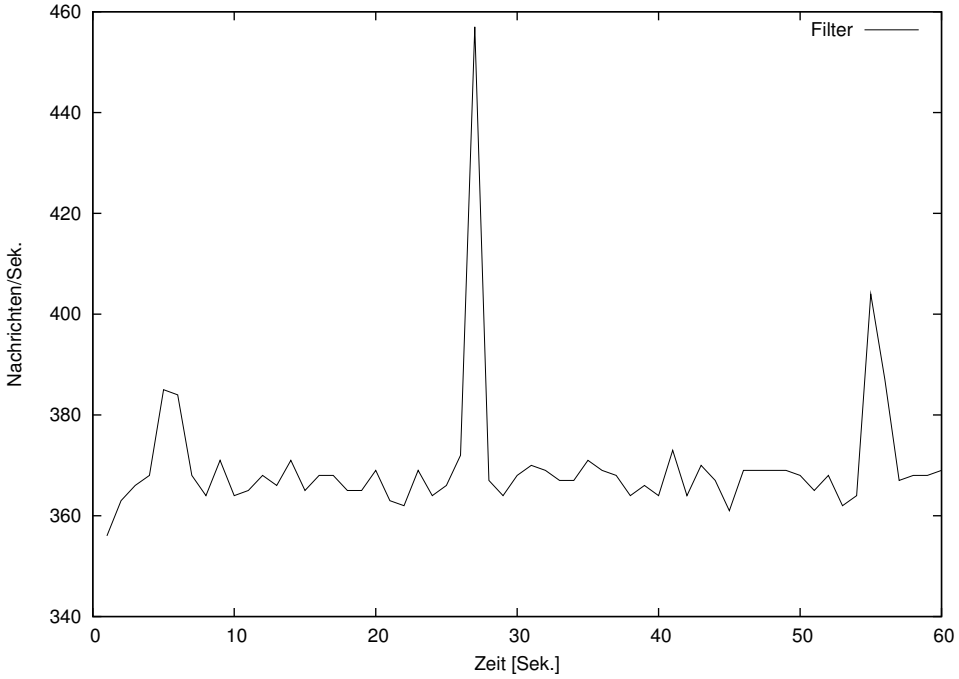


Abbildung 6.6**:** Durchsatzmessung des Filters

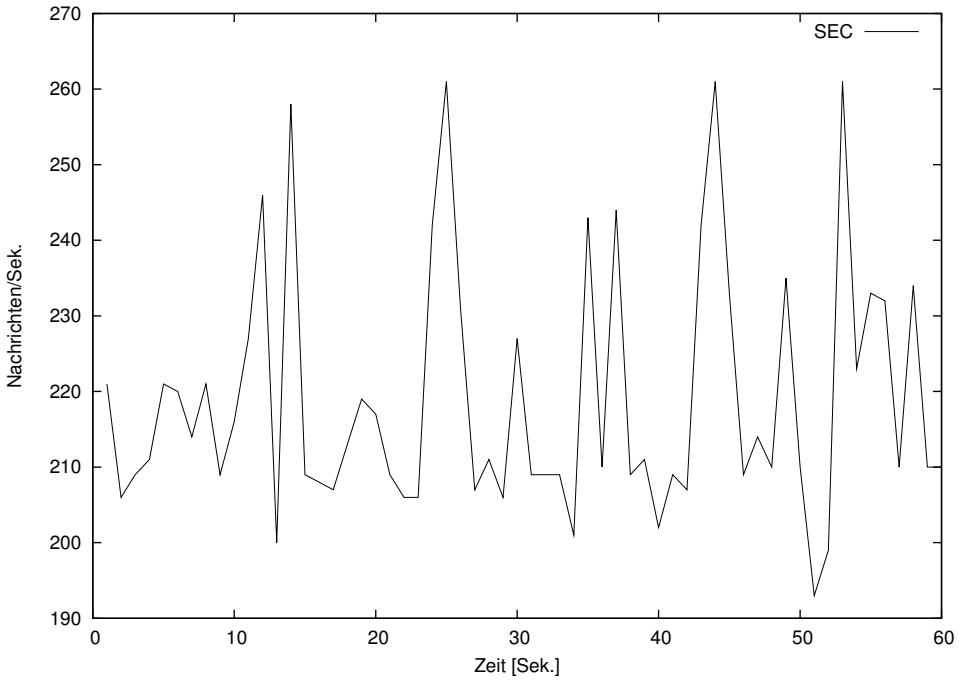


Abbildung 6.7**:** Durchsatzmessung des Korrelators

6.2.1.4 Simple Event Correlator

Durch die Implementierung des Korrelators bedingt, kann auf dem Testsystem die vorhandene Zweikern-Technologie des Prozessors sinnvoll eingesetzt werden, so dass selbst eine vergleichsweise zeitintensive Ereigniskorrelation relativ performant durchgeführt werden kann. Wie bereits in Abschnitt 6.1.1.5 angeführt, kann ein Prozessorkern die Aufgabe der XSL Transformation, die sowohl eine Nachrichtenfilterung der für den Korrelator relevanten Informationen, als auch eine syntaktische Vorverarbeitung umfasst, übernehmen. Der zweite Prozessorkern hat dadurch bedingt genügend Kapazitäten die tatsächliche Ereigniskorrelation durchzuführen.

Abbildung 6.7 stellt die Messergebnisse graphisch dar. Die charakteristischen Kenngrößen der Messung sind in nachfolgender Tabelle kurz zusammengefasst.

	Nachrichten/Sek.
Minimum:	193
Maximum:	261
arithmetisches Mittel:	219.0

Durch das Messverfahren bedingt (es werden wiederholt die gleichen Nachrichten in die zu bewertende Komponente eingegeben), kann für den SEC an dieser Stelle noch keine endgültige Aussage zu seinem Verhalten in einer produktiven Umgebung gemacht werden. Da diese Komponente inhärent zustandbehaftet ist, bleibt es abzuwarten, wie sich der Speicherbedarf und damit auch verbunden der Durchsatz an Nachrichten im produktiven Einsatz verhält. Ein Überblick dazu findet sich nachfolgend in Abschnitt 6.2.2.

6.2.1.5 Anonymisierer/Pseudonymisierer

In Abschnitt 6.1.1.6 sind zwei verschiedene Implementierungen zur Realisierung eines Anonymisierers und Pseudonymisierers vorgestellt worden – zum einen basierend auf regulären Ausdrücken, zum anderen unter Nutzung von XSL Transformationen. Abbildung 6.8 stellt die Leistungsfähigkeit beider Komponenten unabhängig von einander graphisch dar. In der nachfolgenden Tabelle sind die charakteristischen Kenngrößen der beiden Messreihen nochmal übersichtlich aufgeführt, wobei die XSL Transformation einzig eine Identitätsabbildung, also ein unverändertes durchreichen, aber vollständige Interpretation der IDMEF-Nachricht, bewirkt.

	Nachrichten/Sek.	
	sed	*xsltproc*
Minimum:	363	266
Maximum:	479	286
arithmetisches Mittel:	415.3	271.1

Es fällt auf, dass der Einsatz eines Stromeditors gegenüber einem XSLT-Prozessor einen erheblichen Geschwindigkeitsvorteil hat. In hiesigem Fall ist ein ca. um den Faktor 1.5

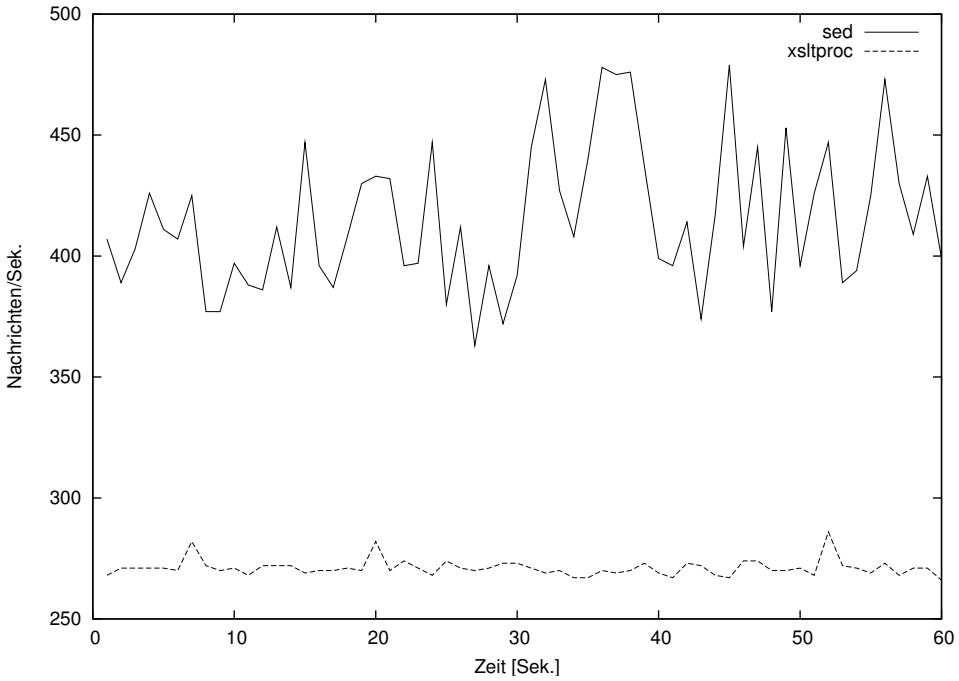


Abbildung 6.8: Durchsatzmessung des Anonymisierers/Pseudonymisierers

erhöhter Durchsatz zu beobachten. Auf der Kehrseite sind jedoch deutliche Schwankungen des Durchsatzes in Abhängigkeit des aktuellen Systemzustands zu beobachten. Solche Einflüsse wirken sich hingegen vergleichsweise wenig auf die XSL Transformation aus.

Der erhebliche Geschwindigkeitsvorteil durch die Verwendung des Stromeditors liegt auf der Hand, da hier wie bereits im Fall der Filterkomponente keine Interpretation der Baumstruktur des XML-Dokuments notwendig ist. Diese Art der Transformation kann in vielen Fällen ausreichend sein. Insbesondere jedoch, wenn eine komplexe Pseudonymisierung personenbezogener Daten in Nachrichten notwendig ist, stoßen reguläre Ausdrücke schnell an ihre Grenzen, wohingegen XSL Transformationen mehr Möglichkeiten bieten.

6.2.1.6 Zusammenfassung

Die ersten Tests lassen optimistisch auf einen möglichen Betrieb der einzelnen Komponenten auch in einer größeren Umgebung blicken. Bei einigen Komponenten kann man je nach Anwendungsfall noch einige Optimierungen durchführen, entweder durch den Einsatz performanterer Hilfsmittel oder über eine geeignete Konfiguration der Komponente selber.

Betreibt man die gesamte Kette der Informationsveröffentlichung im Grid (Filter, SEC, Anonymisierer/Pseudonymisierer), so werden im Durchschnitt gut 120 Nachrichten pro Sekunde auf dem Testsystem durchgesetzt. Bei einer angenommenen Datengröße von durchschnittlich ca. 1kB ergibt sich eine maximal notwendige Übertragungsrate von 120kB/Sek. oder knapp 1MBit/Sek. für jeden GIDS-Agenten. Bei einer durchgängigen

1-10GBit/Sek. Vernetzung im Rahmen des D-Grid ist somit die zusätzliche Last voraussichtlich überschaubar und tolerierbar. Wird zusätzlich auf demselben System auch noch eine lokale GIDS-Datenbank und der Dispatcher (Konfiguration zum Caching von fünf Nachrichten, bevor die Daten in die GIDS-Datenbank geschrieben werden) betrieben, so setzt das gesamte System noch ca. 85 Nachrichten im Mittel pro Sekunde durch.

Weitere Möglichkeiten der Optimierung des Durchsatzes und des Bedarfs an Hintergrundspeicher (z.B. durch Einschränken der Größe der GIDS-Datenbank) lassen sich u.a. auch durch zwei weitere Tricks verwirklichen, die keine Einschränkungen des GIDS mit sich bringen.

Zum einen kann der Filter noch vor dem Dispatcher zum Einsatz kommen, was durch die konsequente Verwendung eines einheitlichen Datenmodells problemlos möglich ist. Dadurch werden bereits sehr früh durch eine hoch performante Komponente (dem Filter) nicht erwünschte Informationen verworfen. Ein Vor- und gleichzeitig Nachteil ist jedoch, dass diese Informationen schlussendlich auch nicht in die GIDS-Datenbank geschrieben werden. Eine Anwendbarkeit muss entsprechend spezifisch für den Anwendungsfall entschieden werden.

Zum anderen kann durch geschicktes Anpassen des XSLT-Stylesheets die Funktionalität des Filters in den Dispatcher ausgelagert werden. Der Effekt ist derselbe wie in der vorangehend beschriebenen Konstellation, wie sich jedoch die Leistungsfähigkeit durch das Erhöhen der Komplexität der XSL Transformation auswirkt, bleibt zu überprüfen.

6.2.2 Testbetrieb im MWN

Als Teststellung für einen Einsatz des GIDS in einer einem Grid angemessenen Umgebung dient das Münchener Wissenschaftsnetz (MWN) und insbesondere die damit enthaltenen D-Grid-Installationen z.B. am Leibniz-Rechenzentrum (LRZ). Zur Auswahl eines geeigneten Standorts für die Sensorik fällt die Wahl auf eine zentrale Komponente – den Übergang zum Wissenschaftsnetz *X-WiN* (weitere Informationen zum X-WiN unter `http://www.dfn.de/xwin/`).

6.2.2.1 Teststellung

Für die Datensammlung steht ein Spiegelport mit einer Bandbreite von 1GBit/Sek. am Übergang des MWN zum X-WiN zur Verfügung, an dem ein in Abschnitt 6.1.1.1 beschriebener tcpdump-Agent betrieben wird. Um eine geeignete Vorverarbeitung der enormen Datenflut zu gewährleisten, fällt die Wahl Verbindungsaufbauversuche zu den für die Grid-Middleware *Globus Toolkit 2*, *Globus Toolkit 4* und *UNICORE* relevanten Ports. Als Referenz hierfür gilt im gesamten D-Grid die von Herr Grimm und Herr Valpota am Regionalen Rechenzentrum für Niedersachsen (RRZN) in Hannover ausgearbeitete „Empfehlung zur statischen Konfiguration von Firewalls im D-Grid“ [Grimm u. Volpato, 2006].

Daraus ergibt sich die Notwendigkeit die in Tabelle 6.1 aufgeführten TCP-Ports für die jeweilige Middleware zu überwachen. Da im Grid ohnehin alle Verbindungen verschlüsselt

Dienst	**Quelle**	**Ziel**
Globus Toolkit 2:		
GRAM Gatekeeper	external : *	localhost : 2119
	localhost : 20000-25000	external : *
GRAM Job-Manager	external : *	localhost : 20000-25000
	localhost : 20000-25000	external : *
MDS GRIS	external : *	localhost : 2135
MDS GIIS	external : *	localhost : 2135
GridFTP control	external : *	localhost : 2811
GridFTP data (single channel)	external : *	localhost : 20000-25000
GridFTP data (multiple channel)	external : *	localhost : 20000-25000
GridFTP data (multiple channel)	localhost : 20000-25000	external : *
GSI-SSH	external : *	localhost : 2222
MyProxy	external : *	localhost : 7512
Globus Toolkit 4:		
GRAM (job startup and control)	external : *	localhost : 8443
	localhost : 20000-25000	external : *
MDS	external : *	localhost : 8443
GridFTP control	external : *	localhost : 2811
GridFTP data (single channel)	external : *	localhost : 20000-25000
GridFTP data (multiple channel)	external : *	localhost : 20000-25000
GridFTP data (multiple channel)	localhost : 20000-25000	external : *
GSI-SSH	external : *	localhost : 2222
MyProxy	external : *	localhost : 7512
Unicore:		
Gatekeeper	external : *	localhost : 4433
	localhost : *	NJS : 8181
	NJS : 8181	localhost : *
NJS	localhost : 8181	gateway : *
(*Network Job Supervisor*)	gateway : *	localhost : 8181

Tabelle 6.1**:** Statische Konfiguration von Firewalls im D-Grid [Grimm u. Volpato, 2006]

sind, ist eine Analyse des Inhalts der Pakete nicht sinnvoll, es reichen also Verbindungsaufbauversuche um daraufhin z.B. eine Anomalieerkennung wie auch in [gentschen Felde, 2005; gentschen Felde u. a., 2006] beschrieben durchzuführen.

Daraus ergibt sich folgende Konfiguration für den tcpdump-Agenten bzw. folgender Filter für den tcpdump-Aufruf im Rahmen des Agenten:

```
tcpdump -i eth1 -n "tcp[tcpflags] & (tcp-syn) != 0 \
                    and not src and dst net 192.168.0.0/16 \
                    and not src and dst net 10.0.0.0/8 \
                    and (   port 2119 \
```

```
or port 2135 \
or port 2811 \
or port 2222 \
or port 7512 \
or port 8443 \
or port 4433 \
or port 8181 \
or portrange 20000-25000)"
```

Da die meisten im D-Grid betriebenen Ressourcen mit einer Variante von Linux oder UNIX betrieben werden, wird zusätzlich ein Agent eingeführt, der alle von außerhalb des MWN stammenden Verbindungsaufbauversuche zum SSH-Port (TCP-Port 22) registriert.

Als Testumgebung dient das in Abschnitt 6.2.1.1 Testsystem. Um insbesondere zu zeigen, dass das vorgeschlagene GIDS skaliert und auch in großen Umgebungen zum Einsatz kommen kann, stehen an erster Stelle die Messungen des Durchsatzes an Nachrichten, die durch den zuvor beschriebenen Aufbau erzeugt werden. Zu beachten ist, dass die in Abschnitt 6.1 vorgestellte Implementierung, die hier als Referenz herangezogen wird, eine Kommunikation zwischen den Komponenten via *named pipes* realisiert. Dadurch kann zwischen den einzelnen Komponenten ein hoch performanter Cache realisiert werden, so dass kurze Überlastsituationen gepuffert werden können.

6.2.2.2 Messergebnisse & Erfahrungen

Abbildung 6.9 stellt graphisch die notwendige Systemleistung für den Betrieb des GIDS am X-WiN-Übergang dar. Auf der ersten Achse ist dazu die Zeit (hier in Abgrenzung zu den anderen Grafiken in Minuten) aufgetragen, die zweite Achse trägt wie gewohnt die Nachrichten pro Sekunde auf. Die dargestellten Werte sind stets gemittelt über eine Minute und zeigen die Messungen aus der Zeit vom 21.10.2008 zwischen 16:00 und 17:00 Uhr. Die Mittelwertbildung über jeweils eine Minute dient dazu, dass die graphische Darstellung einen optisch guten Eindruck über das tatsächliche Nachrichtenaufkommen vermittelt. Bei sekundengenauer Auftragung ergeben sich erhebliche Schwankungen, so dass eine entsprechende Grafik eine Einschätzung unnötig erschwert. Die kurzen Lastspitzen können durch die zuvor beschriebenen Chaching-Mechanismen des prototypisch implementierten GIDS abgefangen werden.

In der Messreihe aus Abbildung 6.9 sind zwei verschiedene Messungen zu betrachten. Zum einen werden die Anzahl der Verbindungsaufbauversuche zu TCP-Port 22 (SSH), zum anderen zu den oben beschriebenen für das D-Grid typischen Ports dargestellt. Eine dritte Messkurve summiert über diese beiden Messreihen und vermittelt somit einen Eindruck über die tatsächlich durch diese Sensorik anfallende Last. Nachfolgende Übersicht gibt nochmal einen Überblick über die charakteristischen Daten der Messreihe.

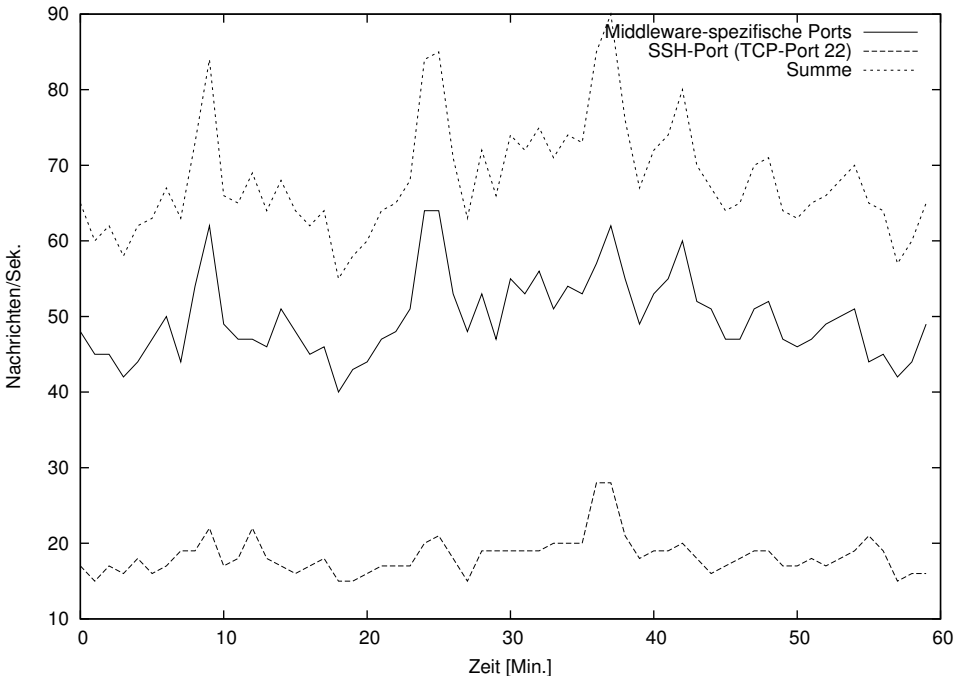


Abbildung 6.9**:** Verbindungsaufbauversuche am X-WiN-Übergang

	Nachrichten/Sek.		
	SSH	*Middleware*	*Summe*
Minimum:	15	40	55
Maximum:	28	62	90
arithmetisches Mittel:	18.3	49.8	68.1

Bei zuvor beschriebener Teststellung fallen im Durchschnitt knapp 70 Nachrichten jede Sekunde an, die durch das lokale GIDS verarbeitet werden müssen. Dabei setzten sich die Meldungen zu ca. 30% aus Daten des SSH-Sensors und zu ca. 70% aus Meldungen des Sensors für Grid-Middleware-spezifische Ports zusammen. Die zu verarbeitende Spitzenlast liegt in diesem Fall bei 90 Meldungen pro Sekunde, auch in anderen Messreihen sind stets Maxima von bis zu 100 Meldungen pro Sekunde beobachtet worden.

6.2.2.3 Zusammenfassung

Die durchgeführten Messungen haben am Beispiel der im Rahmen dieser Arbeiten entstandenen prototypischen Implementierung des vorgeschlagenen GIDS gezeigt, dass eine hinreichende Leistungsfähigkeit des Systems mit leistungsfähiger Hardware, die jedoch aus dem Endanwenderbereich stammt, erzielt werden kann. Das Testsystem arbeitet jedoch an seiner Kapazitätsgrenze, so dass für einen produktiven Einsatz eine leistungsfähigere Hardware aus dem Server-Bereich eingesetzt werden sollte. Die Implementierung lässt eine mas-

sive Parallelisierung der Aufgaben zu, so dass Mehrkern- und Mehrprozessor-Architekturen sinnvoll ausgelastet werden können.

Wie auf der Kehrseite jedoch auch zu sehen ist, stellt die Datenverarbeitung an nur einer Site des D-Grid bereits eine sehr ressourcenintensive Aufgabe dar. Kommt es nun dazu, dass die Daten mehrerer Sites miteinander kombiniert werden sollen, so müssen in jedem Fall hoch verdichtete und aggregierte Daten verteilt werden. Es besteht also der Bedarf eine gute Vorverarbeitung an den Sites, an denen die Daten anfallen, durchzuführen. Die entsprechenden Komponenten wie zum Beispiel eine Komponente zur Aggregation von Informationen und die Möglichkeit bereits lokal erzeugt Analyseergebnisse, z.B. von Intrusion Detection Systemen einer Site, zu verbreiten, legen dafür den Grundstein. Ein sinnvoller Einsatz und eine geeignete Konfiguration der Systeme sind für den Betrieb innerhalb einer großen Grid-Infrastruktur jedoch zwingend notwendig.

6.3 Bewertung des Systems anhand des erhobenen Anforderungskatalogs

Zur Bewertung des vorgeschlagenen GIDS und insbesondere der prototypischen Implementierung, die in Abschnitt 6.1 detailliert beschrieben ist, dient der in Kapitel 3 erarbeitete Anforderungskatalog. Nachfolgend wird der Erfüllungsgrad der einzelnen Kriterien kurz diskutiert.

Funktionale Anforderungen.

Verschiedene Granularitätsstufen bei der Berichterstattung. Die Möglichkeit verschiedene Granularitätsstufen bei der Berichterstattung anbieten zu können ist in weiten Teilen von den verwendeten Analysekomponenten, also den eingesetzten IDS-Instanzen abhängig. Alle gängigen Intrusion Detection Systeme unterstützen jedoch die Angabe unterschiedlicher Prioritäten einer Meldung (engl. *Severity*), so dass hierüber eine Auswahl getroffen werden kann.

Berichterstattung zu qualitativ differierenden Angriffen. Inwiefern eine Berichterstattung zu qualitativ unterschiedlichen Angriffen erbracht werden kann, ist maßgeblich von der Platzierung der vorhandenen Sensorik, der eingesetzten Analysefunktion sowie den von den Ressourcenanbietern verbreiteten Informationen abhängig. Damit ist diese Anforderung fallabhängig erfüllbar, der Grad der Erfüllung hängt jedoch stark vom jeweiligen Ressourcenanbieter ab.

Aussagekräftige Informationsaufbereitung. Abschnitt 6.1.2.1 hat lediglich eine Möglichkeit aufgezeigt die Informationen für ein Benutzerportal abzufragen, implementiert jedoch keine Nutzeroberfläche z.B. in Form eines Web-Portals. Wie aussagekräftig eine Darstellung realisiert wird ist somit ebenfalls von der

tatsächlichen Implementierung abhängig, die Informationsbasis hingegen ist gegeben.

Zugriffsmöglichkeit auf Sensordaten. Durch die Architektur bedingt ist sowohl bei jedem Ressourcenanbieter als auch beim Betreiber des GIDS eine GIDS-Datenbank installiert, die bei Bedarf sämtliche Sensordaten vorhalten kann. Wie viele Daten jedoch in welchem Detailgrad und wie lange gespeichert werden, hängt von anderen Randbedingungen ab (s.u., z.B. organisatorischer oder juristischer Art), so dass diese Anforderung voll erfüllt werden kann.

Variationsmöglichkeit der Informationsquellen/Datenbasis zur Laufzeit. Da eine jede Informationsquelle durch einen eigenen Agenten an das GIDS angebunden wird, ist eine Variation der Informationsquellen jederzeit möglich. Durch die Implementierung einer Inter-Prozesskommunikation mit Hilfe von *named pipes* ist sogar ein Hinzufügen und Entfernen einer Informationsquelle ohne den Neustart oder jedwede Änderung an den anderen Komponenten möglich. Zu beachten hingegen ist, dass eventuell angeschlossene analysierende Einheiten wie lokal betriebene Intrusion Detection Systeme je nach Auswertungslogik durch die Änderung der Datenbasis in Mitleidenschaft gezogen werden könnten.

Proaktive Benachrichtigung. Eine proaktive Benachrichtigung ist durch eine eigene Komponente innerhalb des GIDS realisiert.

Nutzung verschiedener Kommunikationsmodelle. Der Einsatz unterschiedlicher Kommunikationsmodelle ist systeminhärent gegeben. Allein die Betrachtung der Benutzerschnittstelle offenbart sowohl eine grafische Oberfläche (*Pull*) wie auch einen Mechanismus zur proaktiven Benachrichtigung (*Push*).

Aggregatbildung. Auch eine Bildung von Aggregaten gesammelter Daten ist durch eine eigene Komponente im GIDS realisiert. Der Prototyp setzt eine Informationskorrelation durch den *Simple Event Correlator* um.

Informationspräsentation im Grid-Portal. Wie bereits zuvor angesprochen, steht im Rahmen der prototypischen Implementierung kein Web-Portal zur Verfügung. Die Abfragelogik, um ein geeignetes Grid-Portal bereitzustellen, ist jedoch in Abschnitt 6.1.2.1 angeführt worden, so dass dieser Anforderung mit geringem Zusatzaufwand genüge getan werden kann.

Nutzung bestehender Grid-Dienste. Das Einbeziehen verschiedener Grid-Dienste wie z.B. Monitoring-Datenbanken oder verfügbare Kommunikationsmechanismen ist vorgesehen. Es wird ebenfalls Gebrauch davon innerhalb der prototypischen Implementierung gemacht.

Anbindung an VO-Managementsystemen. Wiederum bereits durch die Architektur bedingt und somit von einer Implementierung unabhängig, sieht das Architekturkonzept des GIDS eine Anbindung an vorhandene Grid-Dienste vor, wie z.B. die Nutzung des bestehenden VO-Managementsystems. Eine Anbindung ist zudem für eine volle Funktionsfähigkeit zwingend notwendig.

Nichtfunktionale Anforderungen.

Integrierbarkeit in bestehende Management-Werkzeuge. Die Integrierbarkeit des prototypischen GIDS in bestehende Management-Werkzeuge ist nicht trivial gegeben. Einzig der Einsatz instrumentierter Komponenten, die eine Verwaltung über ihre Management-Werkzeuge zulassen, erfüllt diese Anforderung, was hier nicht der Fall ist. Diese Anforderung bleibt im Rahmen des Prototypen unerfüllt.

Interoperabilität. Durch die Basis verschiedener Agenten je Informationsquelle lässt sich ein hoher Grad an Interoperabilität mit verschiedenen Komponenten realisieren. Exemplarisch sind einige Ansätze im Rahmen des Prototypen aufgezeigt und implementiert worden.

Mandantenfähigkeit. Als neuer Dienst im Grid konzipiert, bietet das GIDS die Möglichkeit nutzerspezifische Sichten auf erhobene Sicherheitsberichte zur Verfügung zu stellen. Die Mandantenfähigkeit ist durch die Ausführungen in Abschnitt 6.1.2.1 und deren Implementierung nochmal verdeutlicht worden, liegt aber bereits im Konzept des GIDS bedingt.

Nachvollziehbarkeit durchgeführter Anfragen. Die Nachvollziehbarkeit durchgeführter Anfragen ist fallabhängig erfüllt. Zwar sind die technischen Voraussetzungen durch die Verfügbarkeit geeigneter Datenbanken gegeben, jedoch können vor allem Datenschutzrichtlinien und weitere (z.B. organisatorische und juristische) Randbedingungen die Möglichkeiten der Speicherung relevanter Informationen unterbinden, so dass diese Anforderung nur mit Einschränkungen erfüllt sein kann.

Portabilität. Durch eine auf Standard-Applikationen und freie Software basierende prototypische Implementierung können die in Abschnitt 6.1 vorgestellten Komponenten auf einer Vielzahl an Systemen zur Ausführung gebracht werden, so dass der Prototyp die Anforderung an seine Portabilität voll erfüllt.

Wiederverwendbarkeit. Durch eine offene Implementierung und die Verwendung etablierter Standards ist eine Wiederverwendbarkeit der einzelnen Komponenten ebenfalls gewährleistet.

Dezentrale Organisation. Bereits durch das Konzept gegeben und damit von seiner Implementierung weitestgehend unabhängig ist eine dezentrale Organisation des GIDS erfüllt. Zwar erscheint auf den ersten Blick der Betreiber des GIDS fälschlicherweise als eine zentrale Organisationseinheit, jedoch ist nach genauerem Hinblick jeder einzelne Ressourcenanbieter in der Lage die Funktionalität des GIDS-Betreibers nahtlos zu übernehmen. Durch die gleiche Datenbasis der einzelnen GIDS-Datenbanken und denselben Aufbau der lokalen GIDS-Installationen wie beim GIDS-Betreiber, ist durch die damit verbundene Informationsreplikation eine vollständig dezentrale Organisation des Gesamtsystems gegeben.

Einheitliche Schnittstellen. Die prototypische Implementierung setzt einzig auf offene Standards und korrespondierende freie Implementierungen der Standards. Dadurch und durch die konsequente Verwendung ergeben sich zwangsweise auch einheitliche Schnittstellen um weitere Komponenten zu integrieren. ...

Erweiterbarkeit und Flexibilität. ... Diese Tatsache führt folglich zu einem hohen Grad an Erweiterungsmöglichkeiten und einer großen Flexibilität neuen Komponenten, die in das GIDS eingebracht werden sollen, gegenüber.

Hohe Leistungsfähigkeit. Die Leistungsfähigkeit der Implementierung ist in Abschnitt 6.2 demonstriert worden. Die jeweiligen Komponenten arbeiten jede für sich hoch performant und auch ein Zusammenspiel erweist sich als hinreichend effizient.

Skalierbarkeit. Die Skalierbarkeit des Systems ist weniger vom GIDS selber abhängig, sondern vielmehr von der zur Angriffserkennung verwendeten Auswertungseinheit und dem anliegenden Datenvolumen. Sinnvolle Informationsvorverarbeitungen und der Einsatz sehr leistungsfähiger IDS-Instanzen sind in großen Umgebungen wie Grids vonnöten, so dass diese Anforderung erfüllt werden kann.

Dynamik der Nutzer und VOs. Die Problematik der Dynamik von Nutzern und VOs wird an die VO-Managementsysteme ausgelagert. Durch eine Integration selbiger kann diese Anforderung ohne zusätzliche Aufwendungen als erfüllt bezeichnet werden, da aktuelle VO-Managementsysteme, wie zum Beispiel das verwendete VOMS, gut mit den Aspekten der Dynamik im Grid-Kontext umgehen können.

Dynamik der Ressourcen. Die Dynamik der Ressourcen hingegen ist unter zwei Gesichtspunkten zu betrachten. Zum einen hat diese Art der Dynamik eine Relevanz für die Berichterstattung und die nutzerspezifische Sichtenbildung. Diesem Aspekt wird wiederum durch die Integration der bestehenden VO-Managementsysteme Rechnung getragen. Zum anderen kann die Dynamik Auswirkungen auf die durchgeführte Angriffserkennung haben, was bisher unberücksichtigt geblieben ist und durch eine entsprechende Wahl eingesetzter IDS-Installationen berücksichtigt werden muss. Zusammenfassend wird deswegen diese Anforderung als mit Einschränkungen erfüllt, aber durchaus erfüllbar erachtet.

Unterstützung etablierter (Grid-)Standards. Die prototypische Implementierung setzt ausschließlich auf die Verwendung bestehender und etablierter Standards, wie zum Beispiel das XML-basierte IDMEF als Datenmodell.

Unterstützung Virtueller Organisationen. Bereits mehrfach voranstehend erwähnt ist die Unterstützung Virtueller Organisationen bereits durch das Konzept des GIDS bedingt gewährleistet. Eine Anbindung an die entsprechenden Komponenten zum Management und dem Umgang VO-spezifischer Aspekte wie z.B. einer sehr großen Dynamik der Nutzer und Ressourcen stellt die Erfüllung dieser Anforderung sicher.

Sicherheitsanforderungen. Den Sicherheitsanforderungen, die an ein Grid-basiertes IDS gestellt werden, kann in fast allen Belangen nachgekommen werden. Sowohl die kryptographischen wie auch die Anforderungen an eine Nutzerverwaltung werden durch die beschriebene prototypische Implementierung in fast allen Fällen voll erfüllt.

Kryptographische Anforderungen. Die kryptographischen Anforderungen, denen Genüge getan werden soll, sind wie folgt festgestellt worden:

- Vertraulichkeit von Daten und Nachrichten
- Authentizität von Daten und Nachrichten
- Integrität von Daten und Nachrichten
- Einsatz (a)symmetrischer Kryptografie
- Kanal- oder nachrichtenbasierte Kommunikationssicherung

Allen Anforderungen wird durch die Basis SSL/TLS-basierter Kommunikation unter Nutzung von X.509 Zertifikaten im Rahmen des Prototypen Rechnung getragen. Einzig der Forderung nach der Sicherstellung der Integrität des Datenbestandes (z.B. Log-Daten, Datenbanken etc.) kann durch die prototypische Implementierung nicht nachgekommen werden, weswegen die Anforderung der Gewährleistung der Integrität von Daten nicht, die Nachrichtenintegrität hingegen schon erfüllt ist.

Nutzerverwaltung. Die Nutzerverwaltung im Rahmen des GIDS wird durch die Einbindung des VO-Managementsystems realisiert. Dabei gilt es folgende Anforderungen zu erfüllen:

- Integration in PKI
- Delegation von Identitäts- und Berechtigungsnachweisen
- Single Sign-On mit Proxy-Zertifikaten
- Einbindung bestehender AA-Mechanismen
- Zugriffsbeschränkung auf Informationen

Die Integration in die bestehende Public Key Infrastruktur, Einbindung in bzw. Nutzung von bestehenden Authentifizierungs- und Autorisierungsmechanismen sowie eine Zugriffsbeschränkung auf Informationen und korrespondierende Berichte ist durch den Prototypen gewährleistet. Nutzerseitig ist jedoch bisweilen keine Delegation von Identitäts- und Berechtigungsnachweisen und Möglichkeiten für ein Single Sign-On mit Proxy-Zertifikaten umgesetzt. Dies ist insbesondere durch den Mangel an einem Grid-Portal zur Erbringung des GIDS als Dienst bedingt, kann aber problemlos bei dessen Implementierung realisiert werden.

Organisatorische und Datenschutzanforderungen.

Etablierung einer vertrauenswürdigen Koordinationseinheit. Durch die Einführung eines GIDS-Betreibers, dessen Rolle z.B. durch ein *Grid Operations Center* (GOC) übernommen werden kann, ist auch erstmalig die Etablierung einer ver-

trauenswürdigen Koordinationseinheit im Kontext eines Grid-weiten IDS durchgesetzt worden, wodurch diese Anforderung voll erfüllt wird.

Prozessspezifikation für (Signatur-) Updates. Im Rahmen dieser Arbeit, wie auch im Zuge einer prototypischen Implementierung, ist kein solcher Prozess spezifiziert worden. Diese Anforderung ist entsprechend nicht erfüllt.

Autonomie beteiligter Informationsanbieter. Durch den Betrieb einer Site-lokalen Instanz des GIDS und die Verwaltung aller zur Veröffentlichung von für das GIDS relevanten Informationen, ist eine Autonomie der beteiligten Informations- und Ressourcenanbietern sichergestellt.

Anonymisierungs-/Pseudonymisierungsmöglichkeiten. Die Möglichkeit zur Anonymisierung und/oder Pseudonymisierung von Informationen vor ihrer Veröffentlichung im Grid ist durch eine eigene Komponente gegeben, die individuell an die Bedürfnisse eines jeweiligen Informationsanbieters angepasst und von diesem entsprechend konfiguriert werden kann.

Durchsetzung des Datenschutzes. Insbesondere durch die Komponenten *Filter* und *Anonymisierer/Pseudonymisierer* ist die technische Grundlage für die Durchsetzung von Datenschutzaspekten berücksichtigt worden. Einzig die korrekte und individuelle Konfiguration obliegt dem jeweiligen Betreiber einer Site-lokalen Instanz des GIDS.

Nachhalten historischer Berichte. Wiederum durch das Vorhandensein von GIDS-Datenbanken besteht die technische Möglichkeit historische Berichte nachhaltig zu sichern. Einzig nicht-technische Randbedingungen wie zum Beispiel der Datenschutz können diese Anforderung unter Umständen unerfüllbar machen.

Archivierung von Sensordaten. Eine sinngemäße Argumentation gilt auch für die Archivierung von Sensordaten. Lediglich nicht-technische Randbedingungen können zur Unerfüllbarkeit dieser Anforderung führen, die technischen Voraussetzungen für eine Archivierung sind bereits durch entsprechende Datenbanken gegeben. Es bleibt jedoch zu prüfen wie sinnvoll bzw. notwendig eine Archivierung im Produktiveinsatz ist, insbesondere ob des enormen Speicherplatzbedarfs einer vollständigen Sensordatenhaltung.

Erkennungsleistung. Sämtliche die Erkennungsleistung betreffende Anforderungen können erfüllt werden, hängen jedoch vom Einzelfall ab. Die örtlichen Aspekte, die die Erkennungsleistung betreffen, sind als

- Schutz der potentiellen Angriffsziele
- Geeignete Sensorplatzierung

festgelegt worden. Beide Anforderungen gehen Hand in Hand und bedingen sich gegenseitig. Ihre Erfüllung ist vom jeweiligen Teilnehmer am GIDS abhängig und muss durch diesen gewährleistet werden. Das Konzept stellt bewusst die Autonomie der

Ressourcenanbieter über diese Anforderungen, um die Akzeptanz und Realisierbarkeit des Gesamtsystems durchzusetzen.

Ähnlich verhält es sich mit den Anforderungen verschiedene Angriffstypen und -muster erkennen zu können. Die dazu festgestellten Anforderungen

- Erkennung verschiedener Angriffstypen (aktiv, passiv/autonom, DoS)
- Entdecken zeitlich verschieden angelegter Angriffe

sind aus der Anforderungsanalyse hervorgegangen. Sie können zwar durch eine geeignete Wahl von Mechanismen zur Angriffserkennung in Kombination mit der Auswertung historischer Daten realisiert werden, müssen jedoch sowohl technisch als auch organisatorisch durch entsprechende Rahmenbedingungen unterstützt werden, die ebenfalls durch die Autonomie der Ressourcenanbieter bestimmt sind.

Tabelle 6.2 stellt die gerade diskutierten Ergebnisse nochmals übersichtlich dar. Ein ✔ steht dabei für eine Erfüllung der Anforderung, ein (✔) besagt, dass diese Anforderung mit Einschränkungen und ggf. abhängig von der Implementierung erfüllt werden kann, ein ✘ bedeutet, dass diese Anforderung nicht erfüllt ist.

Zusammenfassend lässt sich festhalten, dass durch das im Rahmen dieser Arbeit entworfene System ein brauchbarer Grundstein für ein föderiertes IDS für Grids gelegt worden ist. Durch die Betrachtung Grid-spezifischer Aspekte, wie zum Beispiel dem Umgang mit Virtuellen Organisationen oder das Etablieren eines neuen Dienstes im Grid und den dadurch implizierten Kundenbegriff, werden Lücken in der aktuellen Forschungslandschaft geschlossen. Dennoch hat insbesondere Abschnitt 6.3 aufgezeigt, dass ein paar Anforderungen auch durch dieses System nicht oder nur mit Einschränkungen erfüllt werden und es im Umfeld von Grid-basierten Intrusion Detection Systemen noch eine Reihe weiterführender Arbeiten gibt, denen in Zukunft Aufmerksamkeit geschenkt werden muss. Das abschließende Kapitel 7 gibt einen Ausblick dazu.

funktionale Anforderungen	Unterstützung verschiedener Granularitätsstufen bei der Berichterstattung		✔
	Berichterstattung zu qualitativ differierenden Angriffen		(✔)
	Aussagekräftige Informationsaufbereitung		(✔)
	Zugriffsmöglichkeit auf Sensordaten		✔
	Variationsmöglichkeit der Informationsquellen/Datenbasis zur Laufzeit		✔
	Proaktive Benachrichtigung der Kunden		✔
	u.a. auch Grid-bedingt	Nutzung verschiedener Kommunikationsmodelle (Push, Pull, Stream)	✔
		Aggregatbildung	✔
	Grid-bedingt	Informationspräsentation im Grid-Portal	(✔)
		Nutzung bestehender Grid-Dienste	✔
		Anbindung an bzw. Nutzung von bestehenden VO-Managementsystemen	✔
nichtfunktionale Anforderungen	Integrierbarkeit in bestehende Management-Werkzeuge		✘
	Interoperabilität		✔
	Mandantenfähigkeit		✔
	Nachvollziehbarkeit durchgeführter Anfragen		(✔)
	Portabilität		✔
	Wiederverwendbarkeit		✔
	u.a. auch Grid-bedingt	Dezentrale Organisation	✔
		Einheitliche Schnittstellen	✔
		Erweiterbarkeit und Flexibilität	✔
		Hohe Leistungsfähigkeit	✔
		Skalierbarkeit	(✔)
	Grid-bedingt	Dynamik der Nutzer und VOs	✔
		Dynamik der Ressourcen	(✔)
		Unterstützung etablierter (Grid-) Standards	(✔)
		Unterstützung Virtueller Organisationen	✔

Sicherheits-anforderungen	Kryptographische Anforderungen	Vertraulichkeit von Daten und Nachrichten	✔
		Authentizität von Daten und Nachrichten	✔
		Integrität von Daten und Nachrichten	(✔)
		Einsatz symmetrischer und/oder asymmetrischer Kryptografie	✔
		Kanal- oder nachrichtenbasierte Kommunikationssicherung	✔
	Nutzerverwaltung	Integration in PKI	✔
		Delegation von Identitäts- und Berechtigungsnachweisen	(✔)
		Single Sign-On mit Proxy-Zertifikaten	(✔)
		Einbindung bestehender AA-Mechanismen	✔
		Zugriffsbeschränkung auf Informationen	✔
Organisatorische und Datenschutz-anforderungen	Organisatorische Anforderungen	Etablierung einer vertrauenswürdigen Koordinationseinheit	✔
		Prozessspezifikation für (Signatur-) Updates	✘
		Gewährleistung der Autonomie beteiligter Informationsanbieter	✔
	Datenschutz	Anonymisierungs- und/oder Pseudonymisierungsmöglichkeiten	✔
		Durchsetzung des Datenschutzes	(✔)
		Nachhalten historischer Berichte	(✔)
		Archivierung von Sensordaten	(✔)
Erkennungs-leistung	Örtliche Aspekte	Schutz der potentiellen Angriffsziele	(✔)
		Geeignete Sensorplatzierung	(✔)
	Angriffstypen und -muster	Erkennung verschiedener Angriffstypen (aktiv, passiv/autonom, DoS)	(✔)
		Entdecken kurzzeitig angelegter bis hin zu zeitlich lang andauernden Angriffe	(✔)

Tabelle 6.2: Bewertung anhand der erhobenen Anforderungen

Kapitel 7

Zusammenfassung und Ausblick

Nicht zuletzt durch die bisher vorwiegend wissenschaftliche – aber freizügige – Nutzung von Grid-Infrastrukturen sind Fragestellungen des IT-Managements vernachlässigt worden. Dies betrifft insbesondere den Bereich des Sicherheitsmanagements. Ein Effekt daraus ist, dass eine weite Verbreitung auch durch eine wirtschaftliche Nutzung von Grid-Infrastrukturen bisher ausblieb. Um diese Lücke zu schließen, wurde im Rahmen dieser Arbeit erstmals ein föderiertes Intrusion Detection System für Grids (GIDS) entwickelt. Die Motivation für diese Arbeit leitet sich aus dem bisher bestandenen Fehlen eines geeigneten Frühwarnsystems für Grid-Umgebungen ab. Als Lösungsansatz wurde die Idee verfolgt, bestehende Sicherheitsmechanismen zu koppeln und aus den anfallenden Daten Informationen zur Sicherheit zu extrapolieren, da dies bisher im Grid-Kontext nicht möglich war. Dieses System realisiert sowohl die Aufgabe der Überwachung des Grid auf Sicherheitsangriffe sowie die Berichterstattung erkannter sicherheitsrelevanter Ereignisse. Damit wird es nun möglich, Grid-Technologien einem breiteren Anwenderfeld verfügbar zu machen.

Eine große Herausforderung im Grid-Umfeld ist die Kooperation zwischen autonomen Partnern, die gemeinsam ein Sicherheitswerkzeug, hier in Form eines Intrusion Detection Systems, betreiben möchten. Die Kooperationspartner sind dabei sowohl organisatorisch autonom in Form von verschiedenen realen Organisationen, als auch geographisch, zum Teil sogar länderübergreifend, voneinander getrennt.

Dadurch und durch neue durch das Grid bedingte Objekte, die bei konventionellen vernetzten Systemen in der Regel nicht in vergleichbarer Weise existierten (als Beispiel seien an dieser Stelle die Virtuellen Organisationen genannt), erwachsen eine Menge bisher ungelöster Probleme. Teilfragestellungen, denen sich diese Arbeit gewidmet hat, sind unter anderem

- welche Angriffsszenarien auf Grids gibt es,
- welche Kriterien bestimmen die Eignung eines IDS für den Einsatz in Grids,
- was ist eine geeignete (generische) Architektur eines Grid-Frühwarnsystems,
- skaliert ein GIDS für den produktiven Einsatz?

Zur Klärung dieser und weiterer Fragestellungen sind im Rahmen dieser Arbeit eine Menge wissenschaftlicher Ergebnisse erzielt worden.

Defizite bestehender Arbeiten. Bestehende technische und wissenschaftliche Arbeiten wurden anhand des in Abschnitt 3.4 beschriebenen Kriterienkatalogs auf ihre Schwächen für den Einsatz im Grid hin untersucht und ihre diesbezüglichen Defizite herausgearbeitet. Dabei fiel insbesondere auf, dass Grid-Spezifika, wie z.B. die Beachtung Virtueller Organisationen (VOs), zumeist außen vor bleiben oder ihnen gar keine angemessene Aufmerksamkeit geschenkt wird. Zusätzlich ließ sich feststellen, dass organisatorische und Datenschutzaspekte bei allen betrachteten Systemen unbeachtet blieben. Es wird in den meisten Fällen die realitätsferne Annahme, dass sich alle am GIDS teilnehmenden Partner untereinander uneingeschränkt vertrauen und sämtliche (personenbezogenen) Daten miteinander austauschen, getroffen. Die bereits einleitend erwähnte Problematik der Autonomie der beteiligten Ressourcenanbieter verwehrt somit zumeist den bestehenden Systemen die realistische Chance auf einen produktiven Einsatz.

Klassifikation von Angriffsszenarien in Grids. Mit Hilfe von Methoden des Security Engineerings wurde im Rahmen einer Bedrohungsanalyse in Kapitel 3.1 die generische Klassifikation von Angriffsszenarien in Grids durchgeführt. Dabei sind allgemeingültig die Angriffsziele und Risiken im Rahmen des Grid-Computings erörtert worden. Eine Klassifikation der Angreifer sowie Angriffe, die im Grid-Kontext zu erwarten sind, führte schlussendlich zu einer Schutzzieldefinition. Mit Hilfe dieser Ergebnisse ist es im Weiteren erst möglich eine vom Szenario abhängige Risikobewertung vorzunehmen. Die Ergebnisse dieses Abschnitts der Arbeit legen somit den Grundstein für die Bewertungen eines für die Sicherheit verantwortlichen Managers in Grids, falls in Zukunft entsprechende Prozessrahmenwerke mit korrespondierenden Rollen auch für den organisationsübergreifenden Betrieb von IT-Infrastrukturen und -Diensten aufkommen.

Erstellung eines Kriterienkatalogs für die Bewertung und Auswahl von IDS für Grids. Durch das Mittel der Anwendungsfall-getriebenen Anforderungsanalyse und aufgrund der Betrachtung durch Grids bedingter Anforderungen wurde ein Kriterienkatalog für die Bewertung und Auswahl von IDS für Grids erstellt. Die angefallenen Anforderungen ließen sich grob in fünf Kategorien einordnen – funktionale Anforderungen, nichtfunktionale Anforderungen, Sicherheitsanforderungen, organisatorische und Datenschutzanforderungen sowie Anforderungen an die Erkennungsleistung eines GIDS. Dieser Kriterienkatalog bietet die Möglichkeit anhand generischer, objektiver Kriterien ein IDS und seine Eignung für den Einsatz im Grid zu untersuchen. Weiter dient er dazu Stärken und Schwächen lokalisieren und Klassifizieren zu können.

Entwicklung einer generischen Architektur für ein GIDS. Eine von dem Kriterienkatalog abgeleitete und durch Ideen themenverwandter Arbeiten beeinflusste generische Architektur für ein föderiertes Intrusion Detection System für Grids schloss in Kapitel 5 eine Lücke bisher unerfüllter Anforderungen an einen solchen Sicherheitsmechanismus. Auf der einen Seite wurde erstmals die vollkommene Autonomie der beteiligten Ressourcenan-

bieter gewährleistet. Diese erstreckt sich vom selbstständigen Betrieb und Konfigurieren der notwendigen Systeme und Komponenten bis hin zur Durchsetzung der Site-lokalen Datenschutz- und Informationsverbreitungsrichtlinien. Auf der anderen Seite wurden durch diesen Architekturvorschlag interorganisationale Virtuelle Organisationen und der daraus resultierende Kundenbegriff gewürdigt. Durch die Anbindung an Komponenten des Grid-Monitorings sowie VO-Managementsysteme wurde erstmals eine VO-spezifische Sichtenbildung ermöglicht. Die Arbeit beschrieb dazu die notwendigen Kommunikationsbeziehungen sowie Informations- und Kontrollflüsse der Komponenten untereinander. Auf Basis dieser kundenspezifischen Sichtenbildung besteht nun zukünftig die Möglichkeit der Erbringung eines neuen Dienstes im Grid. Dazu wurde im Rahmen der Arbeit bereits die Rolle des Betreibers des GIDS formuliert.

Prototypische Implementierung und Messungen. Die Arbeit schloss mit einer prototypischen Implementierung der zuvor entwickelten generischen Architektur am Beispiel des D-Grids in Kapitel 6. Dieser Prototyp sowie die in Kapitel 5 dargelegte Architektur wurden kritisch gegen den erhobenen Kriterienkatalog abgeglichen. Zudem diente die prototypische Implementierung als Basis für das Führen eines Tragfähigkeitsnachweises im Rahmen dieser Arbeit. Ein solcher wurde sowohl unter Laborbedingungen mit Hilfe von Lasttests sowie durch den Einsatz im Münchener Wissenschaftsnetz (MWN) geführt.

Ausblick

Der Fokus der Arbeit lag auf der Bewertung von und der Erstellung eines IDS für Grids. Daher konnten einige weitere Teilfragestellungen nur kurz angerissen werden. Zudem sind auch zusätzliche neue Fragestellungen im Rahmen dieser Arbeiten zu Tage getreten. Sie alle müssen einer weiteren gründlichen, wissenschaftlichen Diskussion zugeführt werden.

Im Bereich der Angriffserkennung in und für Grids ist die Frage zu klären, welche Analyseverfahren besonders für Grids geeignet sind bzw. welche sich gar nicht eignen. Dabei sind besonders die Einflüsse, die sich durch die hohe Dynamik im Grid bedingen, auf die eingesetzten Analyseverfahren zu bewerten. Es bietet sich an, einen generischen Kriterienkatalog für die Auswahl geeigneter Angriffserkennungsverfahren für ein konkretes Grid-Szenario zu erarbeiten.

Im Bereich der Angriffserkennungsverfahren bedarf es zusätzlich Methoden, wie und in welcher Form zusätzlich verfügbare Informationen, die z.B. aus (Job-)Monitoring oder VO-Managementsystemen gewonnen werden können, sinnvoll eingesetzt und darauf basierend Analysen betrieben werden können. Zurzeit ist vollkommen ungeklärt, wie sich ein „normales“ Nutzerverhalten in solchen Daten widerspiegelt und wie Anomalien in diesen Datenbeständen erkannt werden können.

Zur endgültigen (wirtschaftlichen) Nutzung des vorgeschlagenen GIDS bedarf es zudem der Ausarbeitung eines geeigneten Betriebsmodells. Zwar sind im Rahmen dieser Arbeit bereits die grundlegenden Rollen, die eine Beteiligung am GIDS finden, postuliert worden, dennoch mangelt es insbesondere an einer Einbettung in die entsprechenden Managementprozesse. Bestehende Best-Practice Rahmenwerke wie die *IT Infrastructure Library* (ITIL) [Office of Government and Commerce (OGC), 2007] oder auch die ISO/IEC 20000 [ISO/IEC 20000-1, 2005; ISO/IEC 20000-2, 2005] fokussieren zurzeit nur auf den intraorganisationalen Betrieb von IT-Infrastrukturen, so dass die Niederschrift vergleichbarer Betriebsprozesse für interorganisationale Anwendungsfälle notwendig ist. Im Kontext dieser Arbeiten sind naturgemäß Fragestellungen mit einem besonderen Fokus auf die für die Sicherheit relevanten Prozesse zu klären, offen ist ferner, wie diese zu instanziieren sowie durch technische Hilfsmittel zu unterstützen sind.

Obwohl die Autonomie der im GIDS kooperierenden Ressourcenanbieter durch die hier entwickelte Architektur gewährleistet ist, bestehen dennoch erhebliche Defizite in Bezug auf den Datenschutz sowie notwendige Informationsverbreitungsrichtlinien. Im Rahmen dieser Arbeit sind dazu die Problemfelder umrissen und die technischen Möglichkeiten zu deren Umsetzung aufgezeigt worden, jedoch müsste man noch korrespondierende Konzepte entwickeln. Auf der einen Seite bedarf es der Erarbeitung eines Datenschutzkonzepts, auf der anderen Seite ist die Frage zu klären, wie wissenschaftliche Erkenntnisse z.B. im Bereich des Trust-Level Managements mit einfließen können. Dies war jedoch nicht Gegenstand der vorgelegten Arbeit. Bisher wurde stets bei der Weitergabe von personenbezogenen Daten und Informationen der „kleinste gemeinsame Nenner“ erlaubter Daten herausgegeben, wie auch im Rahmen dieser Arbeit. Durch Methoden des Trust-Level Managements jedoch kann das Etablieren komplexer Vertrauensbeziehungen unterhalb der Ressourcenanbieter und deren technische Unterstützung zu einer Qualitätssteigerung des GIDS führen.

Anhang A

Ein MySQL-Datenbankschema für das IDMEF

Listing A.1 stellt ein MySQL-Skript für die Abbildung von IDMEF-Nachrichten in einer MySQL-Datenbank nach [Prelude-IDS] bereit. Dieses Skript implementiert das in Kapitel 6.1.1.3 in den Abbildungen 6.3 und 6.4 (siehe Seiten 130 und 131) modellierte Datenbank-Schema zur verlustlosen Speicherung der IDMEF-Datenstruktur in einer relationalen Datenbank am Beispiel von MySQL.

Listing A.1: SQL-Schema für IDMEF-Nachrichten nach [Prelude-IDS]

```
1  CREATE TABLE _format (
     name VARCHAR(255) NOT NULL,
3    version VARCHAR(255) NOT NULL
   );
5  INSERT INTO _format (name, version) VALUES('classic', '14.7');

7  CREATE TABLE Alert (
     _ident BIGINT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT,
9    messageid VARCHAR(255) NULL
   ) TYPE=InnoDB;
11
   CREATE TABLE Alertident (
13   _message_ident BIGINT UNSIGNED NOT NULL,
     _index INTEGER NOT NULL,
15   _parent_type ENUM('T','C') NOT NULL, # T=ToolAlert C=CorrelationAlert
     alertident VARCHAR(255) NOT NULL,
17   analyzerid VARCHAR(255) NULL,
     PRIMARY KEY (_parent_type, _message_ident, _index)
19 ) TYPE=InnoDB;

21 CREATE TABLE ToolAlert (
     _message_ident BIGINT UNSIGNED NOT NULL PRIMARY KEY,
23   name VARCHAR(255) NOT NULL,
     command VARCHAR(255) NULL
25 ) TYPE=InnoDB;

27 CREATE TABLE CorrelationAlert (
     _message_ident BIGINT UNSIGNED NOT NULL PRIMARY KEY,
29   name VARCHAR(255) NOT NULL
   ) TYPE=InnoDB;
31
   CREATE TABLE OverflowAlert (
33   _message_ident BIGINT UNSIGNED NOT NULL PRIMARY KEY,
     program VARCHAR(255) NOT NULL,
```

```
35  size INTEGER UNSIGNED NULL,
    buffer BLOB NULL
37 ) TYPE=InnoDB;

39 CREATE TABLE Heartbeat (
    _ident BIGINT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT,
41  messageid VARCHAR(255) NULL,
    heartbeat_interval INTEGER NULL
43 ) TYPE=InnoDB;

45 CREATE TABLE Analyzer (
    _message_ident BIGINT UNSIGNED NOT NULL,
47  _parent_type ENUM('A','H') NOT NULL, # A=Alert H=Hearbeat
    _index TINYINT NOT NULL,
49  analyzerid VARCHAR(255) NULL,
    name VARCHAR(255) NULL,
51  manufacturer VARCHAR(255) NULL,
    model VARCHAR(255) NULL,
53  version VARCHAR(255) NULL,
    class VARCHAR(255) NULL,
55  ostype VARCHAR(255) NULL,
    osversion VARCHAR(255) NULL,
57  PRIMARY KEY (_parent_type,_message_ident,_index)
   ) TYPE=InnoDB;
59
   CREATE TABLE Classification (
61  _message_ident BIGINT UNSIGNED NOT NULL PRIMARY KEY,
    ident VARCHAR(255) NULL,
63  text VARCHAR(255) NOT NULL
   ) TYPE=InnoDB;
65
   CREATE TABLE Reference (
67  _message_ident BIGINT UNSIGNED NOT NULL,
    _index TINYINT NOT NULL,
69  origin ENUM("unknown","vendor-specific","user-specific",
              "bugtraqid","cve","osvdb") NOT NULL,
71  name VARCHAR(255) NOT NULL,
    url VARCHAR(255) NOT NULL,
73  meaning VARCHAR(255) NULL,
    PRIMARY KEY (_message_ident, _index)
75 ) TYPE=InnoDB;

77 CREATE TABLE Source (
    _message_ident BIGINT UNSIGNED NOT NULL,
79  _index SMALLINT NOT NULL,
    ident VARCHAR(255) NULL,
81  spoofed ENUM("unknown","yes","no") NOT NULL,
    interface VARCHAR(255) NULL,
83  PRIMARY KEY (_message_ident, _index)
   ) TYPE=InnoDB;
85
   CREATE TABLE Target (
87  _message_ident BIGINT UNSIGNED NOT NULL,
    _index SMALLINT NOT NULL,
89  ident VARCHAR(255) NULL,
    decoy ENUM("unknown","yes","no") NOT NULL,
91  interface VARCHAR(255) NULL,
    PRIMARY KEY (_message_ident, _index)
93 ) TYPE=InnoDB;

95 CREATE TABLE File (
    _message_ident BIGINT UNSIGNED NOT NULL,
97  _parent0_index SMALLINT NOT NULL,
```

```
 _index TINYINT NOT NULL,
 ident VARCHAR(255) NULL,
 path VARCHAR(255) NOT NULL,
 name VARCHAR(255) NOT NULL,
 category ENUM("current", "original") NULL,
 create_time DATETIME NULL,
 create_time_gmtoff INTEGER NULL,
 modify_time DATETIME NULL,
 modify_time_gmtoff INTEGER NULL,
 access_time DATETIME NULL,
 access_time_gmtoff INTEGER NULL,
 data_size INT UNSIGNED NULL,
 disk_size INT UNSIGNED NULL,
 fstype ENUM("ufs", "efs", "nfs", "afs", "ntfs", "fat16", "fat32",
             "pcfs", "joliet", "iso9660") NULL,
 file_type VARCHAR(255) NULL,
 PRIMARY KEY (_message_ident, _parent0_index, _index)
) TYPE=InnoDB;

CREATE TABLE FileAccess (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent0_index SMALLINT NOT NULL,
 _parent1_index TINYINT NOT NULL,
 _index TINYINT NOT NULL,
 PRIMARY KEY (_message_ident, _parent0_index, _parent1_index, _index)
) TYPE=InnoDB;

CREATE TABLE FileAccess_Permission (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent0_index SMALLINT NOT NULL,
 _parent1_index TINYINT NOT NULL,
 _parent2_index TINYINT NOT NULL,
 _index TINYINT NOT NULL,
 permission VARCHAR(255) NOT NULL,
 PRIMARY KEY (_message_ident, _parent0_index, _parent1_index, _parent2_index, _index)
) TYPE=InnoDB;

CREATE TABLE Linkage (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent0_index SMALLINT NOT NULL,
 _parent1_index TINYINT NOT NULL,
 _index TINYINT NOT NULL,
 category ENUM("hard-link","mount-point","reparse-point","shortcut",
              "stream","symbolic-link") NOT NULL,
 name VARCHAR(255) NOT NULL,
 path VARCHAR(255) NOT NULL,
 PRIMARY KEY (_message_ident, _parent0_index, _parent1_index, _index)
) TYPE=InnoDB;

CREATE TABLE Inode (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent0_index SMALLINT NOT NULL,
 _parent1_index TINYINT NOT NULL,
 change_time DATETIME NULL,
 change_time_gmtoff INTEGER NULL,
 number INT UNSIGNED NULL,
 major_device INT UNSIGNED NULL,
 minor_device INT UNSIGNED NULL,
 c_major_device INT UNSIGNED NULL,
 c_minor_device INT UNSIGNED NULL,
 PRIMARY KEY (_message_ident, _parent0_index, _parent1_index)
) TYPE=InnoDB;
```

```
CREATE TABLE Checksum (
  _message_ident BIGINT UNSIGNED NOT NULL,
  _parent0_index SMALLINT NOT NULL,
  _parent1_index TINYINT NOT NULL,
  _index TINYINT NOT NULL,
  algorithm ENUM("MD4", "MD5", "SHA1", "SHA2-256", "SHA2-384", "SHA2-512",
                 "CRC-32", "Haval", "Tiger", "Gost") NOT NULL,
  value VARCHAR(255) NOT NULL,
  checksum_key VARCHAR(255) NULL, # key is a reserved word
  PRIMARY KEY (_message_ident, _parent0_index, _parent1_index, _index)
) TYPE=InnoDB;

CREATE TABLE Impact (
  _message_ident BIGINT UNSIGNED NOT NULL PRIMARY KEY,
  description TEXT NULL,
  severity ENUM("info", "low","medium","high") NULL,
  completion ENUM("failed", "succeeded") NULL,
  type ENUM("admin", "dos", "file", "recon", "user", "other") NOT NULL
) TYPE=InnoDB;

CREATE TABLE Action (
  _message_ident BIGINT UNSIGNED NOT NULL,
  _index TINYINT NOT NULL,
  description VARCHAR(255) NULL,
  category ENUM("block-installed", "notification-sent", "taken-offline", "other") NOT NULL,
  PRIMARY KEY (_message_ident, _index)
) TYPE=InnoDB;

CREATE TABLE Confidence (
  _message_ident BIGINT UNSIGNED NOT NULL PRIMARY KEY,
  confidence FLOAT NULL,
  rating ENUM("low", "medium", "high", "numeric") NOT NULL
) TYPE=InnoDB;

CREATE TABLE Assessment (
  _message_ident BIGINT UNSIGNED NOT NULL PRIMARY KEY
) TYPE=InnoDB;

CREATE TABLE AdditionalData (
  _message_ident BIGINT UNSIGNED NOT NULL,
  _parent_type ENUM('A', 'H') NOT NULL,
  _index TINYINT NOT NULL,
  type ENUM("boolean","byte","character","date-time","integer","ntpstamp","portlist",
            "real","string","byte-string","xml") NOT NULL,
  meaning VARCHAR(255) NULL,
  data BLOB NOT NULL,
  PRIMARY KEY (_parent_type, _message_ident, _index)
) TYPE=InnoDB;

CREATE TABLE CreateTime (
  _message_ident BIGINT UNSIGNED NOT NULL,
  _parent_type ENUM('A','H') NOT NULL, # A=Alert H=Hearbeat
  time DATETIME NOT NULL,
  usec INTEGER UNSIGNED NOT NULL,
  gmtoff INTEGER NOT NULL,
  PRIMARY KEY (_parent_type,_message_ident)
) TYPE=InnoDB;

CREATE TABLE DetectTime (
  _message_ident BIGINT UNSIGNED NOT NULL PRIMARY KEY,
  time DATETIME NOT NULL,
  usec INTEGER UNSIGNED NOT NULL,
  gmtoff INTEGER NOT NULL
```

```
) TYPE=InnoDB;

CREATE TABLE AnalyzerTime (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent_type ENUM('A','H') NOT NULL, # A=Alert H=Hearbeat
 time DATETIME NOT NULL,
 usec INTEGER UNSIGNED NOT NULL,
 gmtoff INTEGER NOT NULL,
 PRIMARY KEY (_parent_type, _message_ident)
) TYPE=InnoDB;

CREATE TABLE Node (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent_type ENUM('A','H','S','T') NOT NULL, # A=Analyzer T=Target S=Source H=Heartbeat
 _parent0_index SMALLINT NOT NULL,
 ident VARCHAR(255) NULL,
 category ENUM("unknown","ads","afs","coda","dfs","dns","hosts","kerberos",
              "nds","nis","nisplus","nt","wfw") NULL,
 location VARCHAR(255) NULL,
 name VARCHAR(255) NULL,
 PRIMARY KEY(_parent_type, _message_ident, _parent0_index)
) TYPE=InnoDB;

CREATE TABLE Address (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent_type ENUM('A','H','S','T') NOT NULL, # A=Analyser T=Target S=Source H=Heartbeat
 _parent0_index SMALLINT NOT NULL,
 _index TINYINT NOT NULL,
 ident VARCHAR(255) NULL,
 category ENUM("unknown","atm","e-mail","lotus-notes","mac","sna","vm","ipv4-addr",
              "ipv4-addr-hex","ipv4-net","ipv4-net-mask","ipv6-addr","ipv6-addr-hex",
              "ipv6-net","ipv6-net-mask") NOT NULL,
 vlan_name VARCHAR(255) NULL,
 vlan_num INTEGER UNSIGNED NULL,
 address VARCHAR(255) NOT NULL,
 netmask VARCHAR(255) NULL,
 PRIMARY KEY (_parent_type, _message_ident, _parent0_index, _index)
) TYPE=InnoDB;

CREATE TABLE User (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent_type ENUM('S','T') NOT NULL, # T=Target S=Source
 _parent0_index SMALLINT NOT NULL,
 ident VARCHAR(255) NULL,
 category ENUM("unknown","application","os-device") NOT NULL,
 PRIMARY KEY (_parent_type, _message_ident, _parent0_index)
) TYPE=InnoDB;

CREATE TABLE UserId (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent_type ENUM('S','T', 'F') NOT NULL, # T=Target User S=Source User F=File Access
 _parent0_index SMALLINT NOT NULL,
 _parent1_index TINYINT NOT NULL,
 _parent2_index TINYINT NOT NULL,
 _index TINYINT NOT NULL,
 ident VARCHAR(255) NULL,
 type ENUM("current-user","original-user","target-user","user-privs","current-group",
           "group-privs","other-privs") NOT NULL,
 name VARCHAR(255) NULL,
 tty VARCHAR(255) NULL,
 number INTEGER UNSIGNED NULL,
 PRIMARY KEY (_parent_type,_message_ident,_parent0_index,
              _parent1_index,_parent2_index,_index)
```

```
287         # _parent_index1 and _parent2_index will always be zero if parent_type = 'F'
) TYPE=InnoDB;
289
CREATE TABLE Process (
291  _message_ident BIGINT UNSIGNED NOT NULL,
  _parent_type ENUM('A','H','S','T') NOT NULL, # A=Analyzer T=Target S=Source H=Heartbeat
293  _parent0_index SMALLINT NOT NULL,
  ident VARCHAR(255) NULL,
295  name VARCHAR(255) NOT NULL,
  pid INTEGER UNSIGNED NULL,
297  path VARCHAR(255) NULL,
 PRIMARY KEY (_parent_type, _message_ident, _parent0_index)
299 ) TYPE=InnoDB;

301 CREATE TABLE ProcessArg (
  _message_ident BIGINT UNSIGNED NOT NULL,
303  _parent_type ENUM('A','H','S','T') NOT NULL DEFAULT 'A', # A=Analyser T=Target S=Source
  _parent0_index SMALLINT NOT NULL,
305  _index TINYINT NOT NULL,
  arg VARCHAR(255) NOT NULL,
307 PRIMARY KEY (_parent_type, _message_ident, _parent0_index, _index)
) TYPE=InnoDB;
309
CREATE TABLE ProcessEnv (
311  _message_ident BIGINT UNSIGNED NOT NULL,
  _parent_type ENUM('A','H','S','T') NOT NULL, # A=Analyser T=Target S=Source
313  _parent0_index SMALLINT NOT NULL,
  _index TINYINT NOT NULL,
315  env VARCHAR(255) NOT NULL,
 PRIMARY KEY (_parent_type, _message_ident, _parent0_index, _index)
317 ) TYPE=InnoDB;

319 CREATE TABLE Service (
  _message_ident BIGINT UNSIGNED NOT NULL,
321  _parent_type ENUM('S','T') NOT NULL, # T=Target S=Source
  _parent0_index SMALLINT NOT NULL,
323  ident VARCHAR(255) NULL,
  ip_version TINYINT UNSIGNED NULL,
325  name VARCHAR(255) NULL,
  port SMALLINT UNSIGNED NULL,
327  iana_protocol_number TINYINT UNSIGNED NULL,
  iana_protocol_name VARCHAR(255) NULL,
329  portlist VARCHAR (255) NULL,
  protocol VARCHAR(255) NULL,
331 PRIMARY KEY (_parent_type, _message_ident, _parent0_index)
) TYPE=InnoDB;
333
CREATE TABLE WebService (
335  _message_ident BIGINT UNSIGNED NOT NULL,
  _parent_type ENUM('S','T') NOT NULL, # T=Target S=Source
337  _parent0_index SMALLINT NOT NULL,
  url VARCHAR(255) NOT NULL,
339  cgi VARCHAR(255) NULL,
  http_method VARCHAR(255) NULL,
341 PRIMARY KEY (_parent_type, _message_ident, _parent0_index)
) TYPE=InnoDB;
343
CREATE TABLE WebServiceArg (
345  _message_ident BIGINT UNSIGNED NOT NULL,
  _parent_type ENUM('S','T') NOT NULL, # T=Target S=Source
347  _parent0_index SMALLINT NOT NULL,
  _index TINYINT NOT NULL,
349  arg VARCHAR(255) NOT NULL,
```

```
 PRIMARY KEY (_parent_type, _message_ident, _parent0_index, _index)
) TYPE=InnoDB;

CREATE TABLE SnmpService (
 _message_ident BIGINT UNSIGNED NOT NULL,
 _parent_type ENUM('S','T') NOT NULL, # T=Target S=Source
 _parent0_index SMALLINT NOT NULL,
 snmp_oid VARCHAR(255) NULL, # oid is a reserved word in PostgreSQL
 message_processing_model INTEGER UNSIGNED NULL,
 security_model INTEGER UNSIGNED NULL,
 security_name VARCHAR(255) NULL,
 security_level INTEGER UNSIGNED NULL,
 context_name VARCHAR(255) NULL,
 context_engine_id VARCHAR(255) NULL,
 command VARCHAR(255) NULL,
 PRIMARY KEY (_parent_type, _message_ident, _parent0_index)
) TYPE=InnoDB;
```

Anhang B

Implementierung ausgewählter Agenten

B.1 Ein Agent für tcpdump

Der in Listing B.1 dargestellte Agent basiert auf Paket-Sniffer *tcpdump*. Es ist seine Aufgabe die von *tcpsump* ausgegebenen Informationen prototypisch in IDMEF-Nachrichten zu konvertieren. Die Ausgabe erfolgt auf die Standardausgabe und kann somit leicht umgeleitet oder weiterverarbeitet werden. Nähere Informationen finden sich in Abschnitt 6.1.1.1.1 auf Seite 126.

Listing B.1: tcpdump-Agent – Konvertierung von tcpdump-Ausgaben in das IDMEF

```
#!/bin/bash

# Interface, auf das gehört werden soll
INTERFACE="-i eth0"


### Begin ###

# "externe" Config einlesen
. ./config

# tcpdump-Parameter
TCPDUMPPARAM="-tttt $INTERFACE -vv -e -n ip"

# FIFO anlegen
FIFO=./$0.fifo
mkfifo $FIFO

# ein wenig Statisktik
n=0

# Debug- + Statistik-Meldungen auf stderr
START=$(date +%s)
echo $0: $(date): Starting... >> $LOGFILE

# tcpdump in FIFO schreiben lassen, backgrounding...
echo $0: $(date): Spawning tcpdump... >> $LOGFILE && \
/usr/sbin/tcpdump $TCPDUMPPARAM | grep ^[012] > $FIFO &
echo $0: $(date): tcpdump started. >> $LOGFILE

```

```
# Signal-Handling
function term {
  echo $0: $(date): Caught SIGTERM. Finishing... >> $LOGFILE
  END=1;
}
END=0; trap term SIGTERM

while read line && [ $END -eq 0 ]
do

  # statistische Ausgaben zwischendurch
  let n=$n+1; let t=$n%1000
  if [ $t == 0 ] ; then
    echo $0: $(date): Processing message no. $n... >> $LOGFILE
  fi

  # die Informationen aus dem tcpdump extrahieren
  TIME=$(echo $line | cut -d " " -f 1,2 - | sed -e 's/\(.*\)\ \(.*\)\..*$/\1T\2Z/')
  layer4proto=$(echo $line | cut -d " " -f 22 -)
  srcIp=$(echo $line | cut -d " " -f 26 - | sed -e 's/\(.*\..*\..*\..*\)\.\(.*\)/\1/')
  srcPort=$(echo $line | cut -d " " -f 26 - | sed -e 's/\(.*\..*\..*\..*\)\.\(.*\)/\2/')
  dstIp=$(echo $line | cut -d " " -f 28 - | sed -e 's/\(.*\..*\..*\..*\)\.\(.*\)/\1/')
  dstPort=$(echo $line | cut -d " " -f 28 - | sed -e 's/\(.*\..*\..*\..*\)\.\(.*\)/\2/' \
                                            | sed -e 's/:$//')

  # Beginn der IDMEF-Nachricht
  IDMEF="<?xml version=\"1.0\"?>"
  IDMEF="$IDMEF<idmef:IDMEF-Message version=\"1.0\""
  IDMEF="$IDMEF xmlns:idmef=\"http://iana.org/idmef\">"

    # einen Alert generieren
    IDMEF="$IDMEF<idmef:Alert>"

      # TCPdump als Analyzer
      IDMEF="$IDMEF<idmef:Analyzer name=\"tcpdump\"/>"

      # Zeitstempel setzen
      IDMEF="$IDMEF<idmef:CreateTime>$TIME</idmef:CreateTime>"
      #IDMEF="$IDMEF<idmef:DetectTime>$TIME</idmef:DetectTime>"

      # Classification
      IDMEF="$IDMEF<idmef:Classification text=\"DARPA tcpdump training data\"/>"

      # Source
      IDMEF="$IDMEF<idmef:Source>"
        IDMEF="$IDMEF<idmef:Node>"
          IDMEF="$IDMEF<idmef:Address category=\"ipv4-addr\">"
            IDMEF="$IDMEF<idmef:address>$srcIp</idmef:address>"
          IDMEF="$IDMEF</idmef:Address>"
        IDMEF="$IDMEF</idmef:Node>"
        IDMEF="$IDMEF<idmef:Service ip_version=\"4\">"
          IDMEF="$IDMEF<idmef:port>$srcPort</idmef:port>"
          IDMEF="$IDMEF<idmef:protocol>$layer4proto</idmef:protocol>"
        IDMEF="$IDMEF</idmef:Service>"
      IDMEF="$IDMEF</idmef:Source>"

      # Target
      IDMEF="$IDMEF<idmef:Target>"
        IDMEF="$IDMEF<idmef:Node>"
          IDMEF="$IDMEF<idmef:Address category=\"ipv4-addr\">"
            IDMEF="$IDMEF<idmef:address>$dstIp</idmef:address>"
          IDMEF="$IDMEF</idmef:Address>"
        IDMEF="$IDMEF</idmef:Node>"
```

```
94              IDMEF="$IDMEF<idmef:Service_ip_version=\"4\">"
                  IDMEF="$IDMEF<idmef:port>$dstPort</idmef:port>"
96                IDMEF="$IDMEF<idmef:protocol>$layer4proto</idmef:protocol>"
              IDMEF="$IDMEF</idmef:Service>"
98          IDMEF="$IDMEF</idmef:Target>"

100       IDMEF="$IDMEF</idmef:Alert>"

102     # Ende der IDMEF-Nachricht
        IDMEF="$IDMEF</idmef:IDMEF-Message>"
104     echo $IDMEF

106 done < $FIFO

108 # Debug- + Statistik-Meldungen auf stderr
    echo $0: $(date): terminated. >> $LOGFILE
110 ENDE=$(date +%s)
    let ZEIT=$ENDE-$START
112 echo $0: processed $n message\(s\) in $ZEIT second\(s\). >> $LOGFILE
    if [ $ZEIT != 0 ] ; then
114   let AVG=$n/$ZEIT
      echo $0: avg. of $AVG msg/sec >> $LOGFILE
116 fi

118 # aufräumen
    sleep 1s; rm -f $FIFO
```

B.2 Ein Agent für syslog

Um den in Listing B.2 beschriebenen Agenten für *syslog(-ng)* verwenden zu können, muss *syslog* bzw. *syslog-ng* wie in Abschnitt 6.1.1.1.2 beschrieben konfiguriert sein um ausgewählte Log-Einträge in Kopie in eine Linux-FIFO (*named pipe*) zu schreiben. Unter dieser Voraussetzung kann der Agent speziell formatierte syslog-Einträge in eine IDMEF-Nachricht verpacken. In erster Linie ist der Agent für die Übersetzung von syslog-Einträgen, die durch den OpenSSH-Dämon erzeugt werden, gedacht. Es lassen sich sämtliche nur erdenkliche Erweiterungen zur Unterstützung weiterer syslog-Einträge jedoch einfach auf Basis der aufgezeigten Implementierung hinzufügen. Für weitere Informationen sei auf Abschnitt 6.1.1.1.2 auf Seite 127 verwiesen.

Listing B.2: syslog-Agent – Konvertierung von syslog-Ausgaben in das IDMEF

```
#!/bin/bash

### Begin ###

# "externe" Config einlesen
. ./config

# ein wenig Statisktik
n=0

# Debug- + Statistik-Meldungen auf stderr
START=$(date +%s)
echo $0: $(date): Starting... >> $LOGFILE

# Signal-Handling
function term {
  echo $0: $(date): Caught SIGTERM. Finishing... >> $LOGFILE
  END=1;
}
END=0; trap term SIGTERM

while read line && [ $END -eq 0 ]
do

  # statistische Ausgaben zwischendurch
  let n=$n+1; let t=$n%1000
  if [ $t == 0 ] ; then
    echo $0: $(date): Processing message no. $n... >> $LOGFILE
  fi

  # Informationen extrahieren
  YEAR=$(date +%Y)
  MONTH=$(echo $line | cut -d "␣" -f 2 -)
  case $MONTH in
    Jan )
      $MONTH=01 ;;
    Feb )
      $MONTH=02 ;;
    Mar )
      $MONTH=03 ;;
    Apr )
      $MONTH=04 ;;
    May )
      $MONTH=05 ;;
```

```
45      Jun )
          $MONTH=06 ;;
47      Jul )
          $MONTH=07 ;;
49      Aug )
          $MONTH=08 ;;
51      Sep )
          $MONTH=09 ;;
53      Oct )
          $MONTH=10 ;;
55      Nov )
          $MONTH=11 ;;
57      Dec )
          $MONTH=12 ;;
59    esac
    DAYTIME=$(echo $line | cut -d " " -f 2,3 - | sed -e 's/\(.*\)\ \(.*\).*$/\1T\2Z/')
61  TIME=$YEAR\-$MONTH\-$DAYTIME

63  HOST=$(echo $line | cut -d " " -f 4 -)
    PROCESS=$(echo $line | cut -d " " -f 5 - | sed -e 's/:$//')
65
    TEXT=$(echo $line | cut -d " " -f 6- -)
67  # kann leere Strings ergeben!!!
    srcIp=$(echo $TEXT | \
69    grep "from [0123456789]*.[0123456789]*.[0123456789]*.[0123456789]*" | \
      sed -e 's/.*from\ \([0123456789]*.[0123456789]*.[0123456789]*.[0123456789]\).*/\1/')
71  srcPort=$(echo $TEXT | grep "port [0123456789]*" | \
                           sed -e 's/.*port\ \([0123456789]*\).*/\1/')
73
    # Beginn der IDMEF-Nachricht
75  IDMEF="<?xml version=\"1.0\"?>"
    IDMEF="$IDMEF<idmef:IDMEF-Message version=\"1.0\""
77  IDMEF="$IDMEF xmlns:idmef=\"http://iana.org/idmef\">"

79    # einen Alert generieren
      IDMEF="$IDMEF<idmef:Alert>"
81
        # TCPdump als Analyzer
83      IDMEF="$IDMEF<idmef:Analyzer name=\"syslog\"/>"

85      # Zeitstempel setzen
        IDMEF="$IDMEF<idmef:CreateTime>$(date +%Y-%m-%dT%TZ)</idmef:CreateTime>"
87      IDMEF="$IDMEF<idmef:DetectTime>$TIME</idmef:DetectTime>"

89      # Classification
        IDMEF="$IDMEF<idmef:Classification text=\"syslog data: $HOST $PROCESS\"/>"
91
        # Source
93      if [[ $srcIp != "" || $srcPort != "" ]]; then
          IDMEF="$IDMEF<idmef:Source>"
95          if [[ $srcIp != "" ]]; then
              IDMEF="$IDMEF<idmef:Node>"
97              IDMEF="$IDMEF<idmef:Address category=\"ipv4-addr\">"
                  IDMEF="$IDMEF<idmef:address>$srcIp</idmef:address>"
99              IDMEF="$IDMEF</idmef:Address>"
              IDMEF="$IDMEF</idmef:Node>"
101         fi
            if [[ $srcPort != "" ]]; then
103           IDMEF="$IDMEF<idmef:Service ip_version=\"4\">"
                IDMEF="$IDMEF<idmef:port>$srcPort</idmef:port>"
105             #IDMEF="$IDMEF<idmef:protocol>$layer4proto</idmef:protocol>"
              IDMEF="$IDMEF</idmef:Service>"
107         fi
```

```
          IDMEF="$IDMEF</idmef:Source>"
109       fi

111     IDMEF="$IDMEF</idmef:Alert>"

113   # Ende der IDMEF-Nachricht
      IDMEF="$IDMEF</idmef:IDMEF-Message>"
115   echo $IDMEF

117 done < $SYSLOG

119 # Debug- + Statistik-Meldungen auf stderr
    echo $0: $(date): terminated. >> $LOGFILE
121 ENDE=$(date +%s)
    let ZEIT=$ENDE-$START
123 echo $0: processed $n message\(s\) in $ZEIT second\(s\). >> $LOGFILE
    if [ $ZEIT != 0 ] ; then
125   let AVG=$n/$ZEIT
      echo $0: avg. of $AVG msg/sec >> $LOGFILE
127 fi
```

Anhang C

Implementierung eines Nachrichten-Dispatchers

Eine für den in Abschnitt 6.1.1.2 auf Seite 127 beschriebene Implementierung eines Nachrichten-Dispatchers ist in Listing C.1 dargestellt. Die Hauptaufgabe des Dispatchers ist, dass eingehende Nachrichten auf der einen Seite direkt wieder zur Weiterverarbeitung auf der Standardausgabe ausgegeben werden. Auf der anderen Seite wird eine Caching-Strategie implementiert, so dass mehrere Nachrichten auf einmal in eine angeschlossene MySQL-Datenbank geschrieben werden können, um den Verwaltungs-Overhead, der durch das Datenbankmanagementsystem und die Verbindungsauf- und -abbaukosten bedingt ist, zu minimieren.

Listing C.1: Ein Nachrichten-Dispatcher

```
#!/bin/bash

# Anzahl zwischenzuspeichernder Anfragen
CACHING=5

### Begin ###

# "externe" Config einlesen
. ./config

# ein wenig Statisktik
n=0

# Debug- + Statistik-Meldungen auf stderr
START=$(date +%s)
echo $0: $(date): Starting... >> $LOGFILE

# temporäre Variablen
DBQUERY=""
anz=0

# Signal-Handling
function term {
  echo $0: $(date): Caught SIGTERM. Finishing... >> $LOGFILE
  END=1;
}
END=0; trap term SIGTERM

# transformiert IMDEF (XML) in eine SQL-Anfrage
```

```
30 # schreibt SQL query auf die Standardausgabe
   function idmef2sql () {
32   echo $(echo $1 | xsltproc ./idmef2sql.xslt -)
   }
34
   # schreibt $1 (IDMEF-Nachricht!) in die Datenbank, wenn genügend
36 # Anfragen zwischengespeichert wurden
   function writeToDB () {
38   DBQUERY="$DBQUERY $(idmef2sql "$1")"
     let anz=$anz+1
40   if [ $anz == $CACHING ] ; then
       echo "$DBQUERY" | mysql --host=$DBHOST --user=$DBUSER \
42                               --password=$DBPASS --database=$DBNAME &
       DBQUERY=""
44     anz=0
     fi
46 }

48 while read line && [ $END -eq 0 ]
   do
50
     # statistische Ausgaben zwischendurch
52   let n=$n+1
     let t=$n%1000
54   if [ $t == 0 ] ; then
       echo $0: $(date): Processing message no. $n... >> $LOGFILE
56   fi

58   # Ausgabe auf stdout
     echo "$line"
60
     # schreiben in DB
62   writeToDB "$line"

64 done < $AGENTS


66
   # Debug- + Statistik-Meldungen auf stderr
68 echo $0: $(date): terminated. >> $LOGFILE
   ENDE=$(date +%s)
70 let ZEIT=$ENDE-$START
   echo $0: processed $n message\(s\) in $ZEIT second\(s\) >> $LOGFILE
72 if [ $ZEIT != 0 ] ; then
     let AVG=$n/$ZEIT
74   echo $0: avg. of $AVG msg/sec >> $LOGFILE
   else
76   echo "" >> $LOGFILE
   fi
```

Anhang D

Implementierung eines Filters

Listing D.1 zeigt die prototypische Implementierung einer in Abschnitt 6.1.1.4 auf Seite 132 näher beschriebenen Filter-Komponente. Zur Spezifikation eines Filterkriteriums wird dabei auf den Einsatz regulärer Ausdrücke gesetzt.

Listing D.1: Die Filter-Komponente unter Nutzung regulärer Ausdrücke

```
#!/bin/bash

# regexp, der Nachrichten, die gefiltert werden sollen
# z.B.
# - filter alle Nachrichten, die die IP-Adresse 192.168.x.x enthalten
# - filter alle Nachrichten, die die IP-Adresse 10.x.x.x enthalten
#
#FILTER="<idmef:address>192\.168\.[0123456789]*\.[0123456789]*</idmef:address>"
FILTER="<idmef:address>10\.[0123456789]*\.[0123456789]*\.[0123456789]*</idmef:address>"


### Begin ###

# "externe" Config einlesen
. ./config

# ein wenig Statisktik
n=0; filtered=0; passed=0

# Debug- + Statistik-Meldungen auf stderr
START=$(date +%s)
echo $0: $(date): Starting... >> $LOGFILE

# Signal-Handling
function term {
  echo $0: $(date): Caught SIGTERM. Finishing... >> $LOGFILE
  END=1;
}
END=0; trap term SIGTERM

while read line && [ $END -eq 0 ]
do

  # statistische Ausgaben zwischendurch
  let n=$n+1; let t=$n%1000;
  if [ $t == 0 ] ; then
    echo $0: $(date): Processing message no. $n... >> $LOGFILE
  fi

  echo $line | grep -v $FILTER
```

```
41
  GREPEXIT=$?
43  if [ $GREPEXIT == 1 ] ; then
      let filtered=$filtered+1
45  elif [ $GREPEXIT == 0 ] ; then
      let passed=$passed+1
47  fi

49 done < $DISPATCH

51 # Debug- + Statistik-Meldungen auf stderr
   echo $0: $(date): terminated. >> $LOGFILE
53 ENDE=$(date +%s)
   let ZEIT=$ENDE-$START
55 STATS="$passed message(s) passed, $filtered message(s) dropped."
   echo $0: processed $n message\(s\) in $ZEIT second\(s\)\; $STATS >> $LOGFILE
57 if [ $ZEIT != 0 ] ; then
     let AVG=$n/$ZEIT
59   echo $0: avg. of $AVG msg/sec >> $LOGFILE
   fi
```

Anhang E

Implementierung eines Korrelators

Zur Ereigniskorrelation wird im Kontext der prototypischen Implementierung der *Simple Event Correlator* (SEC) eingesetzt. Wie dieser im Rahmen des vorgeschlagenen GIDS eingebunden werden kann, ist in Listing E.1 aufgezeigt. Im Wesentlichen wird eine Vorverarbeitung eingehender IDMEF-Nachrichten durch eine XSL Transformation durchgeführt, bevor die so vorverarbeiteten Informationen zur Ereigniskorrelation an den SEC weitergereichtet werden. Dazu werden wiederum Linux-FIFOs verwendet. Nähere Informationen zur Implementierung eines Korrelators sind in Abschnitt 6.1.1.5 auf Seite 132 zu finden.

Listing E.1: Einbindung des Simple Event Correlators (SEC)

```
#!/bin/bash

### Begin ###

# "externe" Config einlesen
. ./config

# ein wenig Statisktik
n=0; altered=0; passed=0;

# Debug- + Statistik-Meldungen auf stderr
START=$(date +%s)
echo $0: $(date): Starting... >> $LOGFILE

# Signal-Handling
function term {
  echo $0: $(date): Caught SIGTERM. Finishing... >> $LOGFILE
  kill $PIDS
  sleep 1s
  rm -f $FIFOS
  END=1;
}
END=0; trap term SIGTERM

# filter und correlator starten
PIDS=""; FIFOS="$SEC_FILTER_INPUT ./sec_filter.pipe"
mkfifo ./sec_filter.pipe
mkfifo $SEC_FILTER_INPUT
$SEC -input=$SEC_FILTER_INPUT -conf=$SEC_FILTER_CONF -log=$SEC_FILTER_LOG \
     -debug=4 -intevents &
PIDS="$PIDS $!"
$SEC -input=./sec_filter.pipe -conf=$SEC_CORRELATOR_CONF -log=$SEC_CORRELATOR_LOG \
     -debug=4 -intevents &
PIDS="$PIDS $!"
```

```
35
while read line && [ $END -eq 0 ]
37 do

39   # statistische Ausgaben zwischendurch
     let n=$n+1
41   let t=$n%1000
     if [ $t == 0 ] ; then
43     echo $0: $(date): Processing message no. $n... >> $LOGFILE
     fi
45
     # XML-string auf stdout ausgeben
47   echo $line
     # Nachricht konvertieren und zur Korrelation weiterreichen
49   echo $line | xsltproc correlator.xslt - > $SEC_FILTER_INPUT

51 done < $FILTER


53
# Debug- + Statistik-Meldungen auf stderr
55 echo $0: $(date): terminated. >> $LOGFILE
ENDE=$(date +%s)
57 let ZEIT=$ENDE-$START
echo $0: processed $n message\(s\) in $ZEIT second\(s\) >> $LOGFILE
59 if [ $ZEIT != 0 ] ; then
     let AVG=$n/$ZEIT
61   echo $0: avg. of $AVG msg/sec >> $LOGFILE
else
63   echo "" >> $LOGFILE
fi
```

Anhang F

Implementierung eines Anonymisierers/Pseudonymisierers

Aus Gründen der Performanz sind im Rahmen der prototypischen Implementierung des GIDS zwei verschiedene Implementierungen eines Anonymisierers/Pseudonymisierers entstanden. Zum einen ist die sehr performante Variante unter Nutzung regulärer Ausdrücke entstanden, zum anderen kommt eine XSL Transformation zum Einsatz.

F.1 Anonymisierer/Pseudonymisierer unter Nutzung regulärer Ausdrücke

Listing F.1 stellt eine prototypische Implementierung eines in Abschnitt 6.1.1.6.1 auf Seite 134 beschriebenen Anonymisierers/Pseudonymisierers unter Nutzung regulärer Ausdrücke vor. Die Basis der Implementierung ist der Stromeditor *sed*.

Listing F.1: Anonymisierer/Pseudonymisierer unter Nutzung regulärer Ausdrücke

```
#!/bin/bash

# regexp, gegen die geprüft werden soll
MATCH="\(10\)\.[0123456789]*\.[0123456789]*\.[0123456789]*"
# Ersetzung durch...
SUB="\1\.0\.0\.0"

# statistische Ausgaben?
# 0: keine Ausgaben; sonst: stat. Auswertung
STAT=0


### Begin ###

# "externe" Config einlesen
. ./config

# ein wenig Statisktik
n=0; altered=0; passed=0;

```

```
# Debug- + Statistik-Meldungen auf stderr
START=$(date +%s)
echo $0: $(date): Starting... >> $LOGFILE

# sed-Kommando zusammensetzen
SEDCMD="sed -e s/$MATCH/$SUB/g"

# Signal-Handling
function term {
  echo $0: $(date): Caught SIGTERM. Finishing... >> $LOGFILE
  END=1;
}
END=0; trap term SIGTERM

while read line && [ $END -eq 0 ]
do

  # statistische Ausgaben zwischendurch
  let n=$n+1
  let t=$n%1000
  if [ $t == 0 ] ; then
    echo $0: $(date): Processing message no. $n... >> $LOGFILE
  fi

  echo $line | $SEDCMD

  # und noch einen für die Statisktik
  if [ $STAT != 0 ] ; then
    echo $line | grep $MATCH > /dev/null
    GREPEXIT=$?
    if [ $GREPEXIT == 0 ] ; then
      let altered=$altered+1
    elif [ $GREPEXIT == 1 ] ; then
      let passed=$passed+1
    fi
  fi

done < $CORRELATED

# Debug- + Statistik-Meldungen auf stderr
echo $0: $(date): terminated. >> $LOGFILE
ENDE=$(date +%s)
let ZEIT=$ENDE-$START
echo $0: processed $n message\(s\) in $ZEIT second\(s\) >> $LOGFILE
if [ $STAT != 0 ] ; then
  echo "$0: $passed message(s) untouched, $altered message(s) altered." >> $LOGFILE
fi
if [ $ZEIT != 0 ] ; then
  let AVG=$n/$ZEIT
  echo $0: avg. of $AVG msg/sec >> $LOGFILE
fi
```

F.2 Anonymisierer/Pseudonymisierer unter Nutzung einer XSL Transformation

Als Alternative zu der zuvor erwähnten Implementierung eines Anonymisierers/Pseudonymisierers stellt Listing F.2 eine Variante unter Nutzung einer XSL Transformation vor. Die Transformationsregeln müssen in einem XSLT-Stylesheet separat angegeben werden. Listing 6.4 auf Seite 135 zeit ein mögliches Stylesheet auf. Weitere Informationen zu diesem Typ des Anonymisierers/Pseudonymisierers sind in Abschnitt 6.1.1.6.2 auf Seite 135 zu finden.

Listing F.2: Anonymisierer/Pseudonymisierer unter Nutzung einer XSL Transformation

```
#!/bin/bash

### Begin ###

# "externe" Config einlesen
. ./config

# ein wenig Statisktik
n=0; altered=0; passed=0;

# Debug- + Statistik-Meldungen auf stderr
START=$(date +%s)
echo $0: $(date): Starting... >> $LOGFILE

# Signal-Handling
function term {
  echo $0: $(date): Caught SIGTERM. Finishing... >> $LOGFILE
  END=1;
}
END=0; trap term SIGTERM

while read line && [ $END -eq 0 ]
do

  # statistische Ausgaben zwischendurch
  let n=$n+1
  let t=$n%1000
  if [ $t == 0 ] ; then
    echo $0: $(date): Processing message no. $n... >> $LOGFILE
  fi

  echo $line | xsltproc $XSLT -

done < $CORRELATED

# Debug- + Statistik-Meldungen auf stderr
echo $0: $(date): terminated. >> $LOGFILE
ENDE=$(date +%s)
let ZEIT=$ENDE-$START
echo $0: processed $n message\(s\) in $ZEIT second\(s\)\;\ $STATS >> $LOGFILE
if [ $ZEIT != 0 ] ; then
  let AVG=$n/$ZEIT
  echo $0: avg. of $AVG msg/sec >> $LOGFILE
fi
```

Abbildungsverzeichnis

Tabellenverzeichnis

LITERATUR

AIDE
Lehti, Rami (Hrsg.) ; Virolainen, Pablo (Hrsg.) ; van den Berg, Richard (Hrsg.): *AIDE - Advanced Intrusion Detection Environment*. `http://sourceforge.net/projects/aide`

Asadzadeh u. a. 2006
Kapitel 22. In: Asadzadeh, Parvin ; Buyya, Rajkumar ; Kei, Chun Ling ; Nayar, Deepa ; Venugopal, Srikumar: *Global Grids and Software Toolkits: A Study of Four Grid Middleware Technologies*. John Wiley & Sons, Inc., 2006. – ISBN 0–471–65471–X, S. 431–458

Banford 2000
Banford, Jeremy: *The Linux Basic Security Module Project (Linux BSM)*. `http://linuxbsm.sourceforge.net/`. Version: November 2000

Basicevic u. a. 2007
Basicevic, Ilija ; Popovic, Miroslav ; Kovacevic, Vladimir: Use of Publisher-Subscriber Design Pattern in Infrastructure of Distributed IDS Systems. In: *ICNS '07: Proceedings of the Third International Conference on Networking and Services*. Washington, DC, USA : IEEE Computer Society, 2007. – ISBN 0–7695–2858–9

Binder 2007
Binder, T.: *Föderation von IDS zur Erkennung von Angriffen auf Grid Infrastrukturen*, Diplomarbeit, Dezember 2007

Bleichenbacher 1998
Bleichenbacher, Daniel: Chosen Ciphertext Attacks against Protocols Based on the RSA Encryption Standard PKCS #1. In: *Proceedings of the 18th Annual International Cryptology Conference (CRYPTO '98) – Advances in Cryptology*, Springer Berlin / Heidelberg, August 1998. – ISBN 978–3–540–64892–5, S. 629–660

Brenner 2007
Brenner, Michael: *Werkzeugunterstützung für ITIL-orientiertes Dienstmanagement – Ein modellbasierter Ansatz*, Ludwig-Maximilians-Universität München, Diss., Juli 2007

Bruegge u. Dutoit 2003
Bruegge, Bernd ; Dutoit, Allen H.: *Object-Oriented Software Engineering: Using UML, Patterns and Java, Second Edition*. Prentice Hall, 2003. – ISBN 0130471100

BSI 2002
Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): BSI-Leitfaden zur Einführung von Intrusion-Detection-Systemen. 2002. – Forschungsbericht. Studie im Auftrag des BSI

Chetty u. Buyya 2002
Chetty, Madhu ; Buyya, Rajkumar: Weaving computational grids: How analogous are they with electrical grids? In: *Computing in Science and Engineering* 4 (2002), August, Nr. 4, 61-71. http://www.gridbus.org/papers/WeavingGrid.pdf. – ISSN 1521–9615

Choon u. Samsudin 2003
Choon, Ong Tian ; Samsudin, Azman: Grid-based Intrusion Detection System. In: *Proceedings of the 9th Asia-Pacific Conference on Communications (APCC)* Bd. 3, 2003, S. 1028–1032

D-Grid Referenzinstallation 2007
Referenzinstallation für die BMBF Sonderinvestitionen. http://www.d-grid.de/fileadmin/dgrid_document/Dokumente/Workshop_Referenzinstallation_15012007.ppt. Version: Januar 2007

Denning 1987
Denning, Dorothy E.: An Intrusion-Detection Model. In: *IEEE Transactions on Software Engineering* SE-13 No. 2 (1987), Februar

DoD 5200.28-STD 1985
Department of Defense (DoD) (Hrsg.): *Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD*. http://www.fas.org/irp/nsa/rainbow/std001.htm. Version: Dezember 1985

Dussa u. a. 2006
Dussa, Tobias ; Epting, Ursula ; Filipovic, Bartol ; Foest, Gerti ; Glowka, Jürgen ; Götze, Joachim ; Grimm, Christian ; Hillenbrand, Markus ; Kohlschütter, Christian ; Lohner, Rudolf ; Makedanz, Siegfried ; Müller, Paul ; Pattloch, Marcus ; Piger, Stefan ; Straub, Tobias ; Wiebelitz, Jan ; Grimm, Christian (Hrsg.) ; Pattloch, Marcus (Hrsg.): Analyse von AA-Infrastrukturen in Grid-Middleware. Version: März 2006. http://www.d-grid.de/fileadmin/user_upload/documents/DGI-FG3-4/Analyse-AAI_v1_1.pdf. D-Grid Integrationsprojekt (DGI). – Forschungsbericht. – Elektronische Ressource

Eckert 2007
Eckert, Claudia: *IT-Sicherheit – Konzepte - Verfahren - Protokolle*. Oldenbourg, 2007. – ISBN 3486582704

Feng u. a. 2006
Feng, Guofu ; Dong, Xiaoshe ; Liu, Weizhe ; Chu, Ying ; Li, Junyang: GHIDS: Defending Computational Grids against Misusing of Shared Resources. In: *Proceedings of the IEEE Asia-Pacific Conference on Services Computing (APSCC)*, 2006, S. 526–533

Filipovic u. Straub 2006
Filipovic, Bartol ; Straub, Tobias: Grid Security Infrastructure – ein Überblick. In: *DFN Tagungsband 2006*, 2006, S. 115–126

Foster 2002
Foster, Ian: What is the grid? A three point checklist. In: *GRIDtoday* 1 (2002), Juli, Nr. 6. `http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf`

Foster 2007
Foster, Ian: *Grid Tutorial – 4th IFIP Summer School.* `http://www-fp.mcs.anl.gov/~foster/Talks/GridFosterSouthAfricaMarch2007v2pdf.pdf`. Version: März 2007

Foster u. Kesselman 1998
Foster, Ian (Hrsg.) ; Kesselman, Carl (Hrsg.): *The Grid: Blueprint for a New Computing Infrastructure*. 2. San Francisco, CA, USA : Morgan Kaufmann Publishers, 1998. – ISBN 1–55860–933–4

Foster u. Kesselman 2003
Foster, Ian ; Kesselman, Carl: *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2003. – ISBN 1558609334

Foster u. a. 2001
Foster, Ian ; Kesselman, Carl ; Tuecke, Steven: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. In: *International Journal of Supercomputer Applications* 15 (2001), Nr. 3

Gamma u. a. 1994
Gamma, Erich ; Helm, Richard ; Johnson, Ralph ; Vlissides, John M.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series, 1994. – ISBN 978–0201633610

gentschen Felde 2005
gentschen Felde, N.: Einsatz der graphbasierten Meldungsstrukturanalyse in domänenübergreifenden Meta–IDS. In: *Lecture Notes in Informatics — Informatik 2005, Informatik LIVE!* Bonn, Germany : Gesellschaft für Informatik, September 2005 (Band 2 P-68), S. 653–657

gentschen Felde u. a. 2006
gentschen Felde, N. ; Jahnke, M. ; Martini, P. ; Tölle, J.: Impact of Sanitized Message Flows in a Cooperative Intrusion Warning System. In: *Proceedings of the 25th Military Communications Conference (MILCOM 2006)*. Washington, DC, USA : IEEE, Oktober 2006, S. 1–7

gLITE
EGEE Project (Hrsg.): *gLite – Lightweight Middleware for Grid Computing*. http://glite.web.cern.ch/glite/

Globus Toolkit 4
The Globus Alliance (Hrsg.): *The Globus Toolkit 4*. http://www.globus.org/

Gong u. a. 2006
Gong, Xun ; Li, Tao ; Wang, Tiefang ; Yang, Jin ; Liang, Gang ; Hu, Xiaoqin: Grid Intrusion Detection Based on Immune Agent. In: *ICNC (2)*, 2006, S. 73–82

Grimm u. Volpato 2006
Grimm, Christian ; Volpato, Gian Luca: Empfehlungen zur statischen Konfiguration von Firewalls im D-Grid / RRZN, Universität Hannover. Version: Juni 2006. http://www.gac-grid.net/project-products/grid-support/grid-installation/FG3-5_Empfehlungen_Statischer_Firewalls_v12.pdf. – Forschungsbericht. – Elektronische Ressource

Gurley Bace 2000
Gurley Bace, Rebecca: *Intrusion Detection*. Macmillan Technical Publishing, 2000. – ISBN 1–57870–185–6

Gürich 2007
Gürich, Wolfgang: Betriebskonzept für die D-Grid Infrastruktur. Version: Oktober 2007. http://www.d-grid.de/fileadmin/user_upload/documents/Kern-D-Grid/Betriebskonzept/D-Grid-Betriebskonzept.pdf. D-Grid Integrationsprojekt (DGI), Fachgebiet 2.2. – Forschungsbericht. – Elektronische Ressource

Hegering u. a. 2003
Hegering ; Hiller ; Maschuw ; Reinefeld ; Resch: *D-Grid: Auf dem Weg zur eScience in Deutschland. (Strategiepapier)*. www.d-grid.de. Version: 2003

Hegering u. a. 1999
Hegering, H.-G. ; Abeck, S. ; Neumair, B.: *Integriertes Management vernetzter Systeme: Konzepte, Architekturen und deren betrieblicher Einsatz*. dpunkt-Verlag, 1999. – ISBN 3–932588–16–9

Hennicker 2006
Hennicker, Rolf: *Objektorientierte Software-Entwicklung*. Vorlesungsskriptum Wintersemester 2005/2006, Institut für Informatik an der Ludwig-Maximilians-Universität München. http://www.pst.informatik.uni-muenchen.de/lehre/WS0506/oose/. Version: 2006

Hollingsworth 1995
Hollingsworth, David: *The Workflow Reference Model*. http://www.wfmc.org/standards/docs/tc003v11.pdf. Version: Januar 1995

Hwang u. a. 2005
Hwang, K. ; Kwok, Y. ; Song, S. ; Cai, M. ; Zhou, R. ; Chen, Y. ; Chen, Y. ; Lou, X.: GridSec: Trusted Grid Computing with Security Binding and Self-Defense against Network Worms and DDoS Attacks. In: *International Workshop on Grid Computing Security and Resource Management (GSRM'05),in conjunction with ICCS 2005*, 2005

ISO/IEC 15408-1 2005
International Organization for Standardization (ISO) (Hrsg.): *Evaluation criteria for IT security – Part 1: Introduction and general model.* `http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html`. Version: Oktober 2005

ISO/IEC 15408-2 2005
International Organization for Standardization (ISO) (Hrsg.): *Evaluation criteria for IT security – Part 2: Security functional requirements.* `http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html`. Version: Oktober 2005

ISO/IEC 15408-3 2005
International Organization for Standardization (ISO) (Hrsg.): *Evaluation criteria for IT security – Part 3: Security assurance requirements.* `http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html`. Version: Oktober 2005

ISO/IEC 20000-1 2005
International Organization for Standardization (ISO) (Hrsg.): *ISO/IEC 20000-1:2005 – Service management – Part 1: Specification.* `http://www.iso.org/`. Version: Dezember 2005

ISO/IEC 20000-2 2005
International Organization for Standardization (ISO) (Hrsg.): *ISO/IEC 20000-2:2005 – Service management – Part 2: Code of practice.* `http://www.iso.org/`. Version: Dezember 2005

Jonsson u. Kaliski 2002
Jonsson, Jakob ; Kaliski, Burton S.: On the Security of RSA Encryption in TLS. In: *Proceedings of the 22nd Annual International Cryptology Conference (CRYPTO 2002) – Advances in Cryptology*, Springer Berlin / Heidelberg, August 2002. – ISBN 978–3–540–44050–5, S. 183–199

Kenny u. Coghlan 2004
Kenny, Stuart ; Coghlan, Brian: *Grid-wide Intrusion Detection.* Dezember 2004

Kenny u. Coghlan 2005
Kenny, Stuart ; Coghlan, Brian: Towards a grid-wide intrusion detection system. In: *Proceedings of the EGC : European grid conference*, 2005. – ISBN 3–540–26918–5, S. 275–284

Klíma u. a. 2003
Klíma, Vlastimil ; Pokorný, Ondrej ; Rosa, Tomáscaron: Attacking RSA-Based Sessions in SSL/TLS. In: *Cryptographic Hardware and Embedded Systems (CHES 2003)*, Springer Berlin / Heidelberg, Oktober 2003. – ISBN 978–3–540–40833–8, S. 426–440

Leu u. a. 2005a
Leu, Fang-Yie ; Li, Ming-Chang ; Lin, Jia-Chun ; Yang, Fu-Yi: Integrating Grid with Intrusion Detection. In: *Proceedings of the 19th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2005, S. 304–309

Leu u. a. 2005b
Leu, Fang-Yie ; Lin, Jia-Chun ; Li, Ming-Chang ; Yang, Chao-Tung: A Performance-Based Grid Intrusion Detection System. In: *Proceedings of the 29th International Computer Software and Applications Conference (COMPSAC)* Bd. 1, 2005. – ISBN 0–7695–2413–3, S. 525–530

Li u. Baker 2005
Li, Maozhen ; Baker, Mark: *The Grid: Core Technologies*. Wiley, 2005. – ISBN 0470094176

Maier 2008
Maier, Christian: *Anforderungen an Grid-basierte Intrusion Detection Systeme*, Diplomarbeit, März 2008

Meder u. a. 2001
Meder, S. ; Welch, V. ; Chicago, U. ; Tuecke, S. ; Engert, D.: GSS-API Extensions / Global Grid Forum – Grid Security Infrastructure (GSI) WG. Version: Februar 2001. `http://ogf.org/documents/GFD.24.pdf`. – Forschungsbericht. – Elektronische Ressource

Nagios
nagios.org (Hrsg.): *Nagios – Network Monitoring*. `http://www.nagios.org/`

Ni u. a. 2007
Ni, Jiancheng ; Li, Zhishu ; Sun, Jirong ; Xing, Jianchuan: Self-adaptive Intrusion Detection System for Computational Grid. In: *Proceedings of the 1st IEEE & IFIP International Symposium on Theoretical Aspects of Software Engineering (TASE)*, 2007, S. 97–106

Oberhaitzinger u. a. 2004
Oberhaitzinger, Barbara ; Gerloni, Helmar ; Reiser, Helmut ; Plate, Jürgen: *Praxisbuch Sicherheit für Linux-Server und -Netze*. Hanser Fachbuchverlag, 2004. – ISBN 3446226265

Office of Government and Commerce (OGC) 2007
Office of Government and Commerce (OGC): *Introduction to the ITIL Service Lifecycle (ITIL Version 3)*. Stationery Office Books, 2007. – ISBN 0113310617

OGSA
Open Grid Services Architecture Working Group (OGSA-WG) (Hrsg.): *Open Grid Services Architecture*. http://forge.gridforum.org/projects/ogsa-wg

OpenDHT
OpenDHT – a Publicly Accessible Distributed Hash Table (DHT) Service. http://www.opendht.org/

OpenVPN
Telethra, Inc. (Hrsg.): *OpenVPN – The Open Source VPN*. http://openvpn.net/

OSSEC
OSSEC (Hrsg.): *OSSEC – an Open Source Host-based Intrusion Detection System*. http://www.ossec.net/

Prelude-IDS
prelude-ids.org (Hrsg.): *Prelude-IDS – The Hybrid IDS framework*. http://www.prelude-ids.org/

RFC1321 1992
Rivest, R. (Hrsg.): *The MD5 Message-Digest Algorithm*. http://www.ietf.org/rfc/rfc1321.txt. Version: April 1992

RFC2743 2000
Linn, J. (Hrsg.): *Generic Security Service Application Program Interface*. http://www.ietf.org/rfc/rfc2743.txt. Version: Januar 2000

RFC2744 2000
Wray, J. (Hrsg.): *Generic Security Service API Version 2 : C-bindings*. http://www.ietf.org/rfc/rfc2744.txt. Version: Januar 2000

RFC3080 2001
Rose, M. (Hrsg.): *The Blocks Extensible Exchange Protocol Core*. http://www.ietf.org/rfc/rfc3080.txt. Version: März 2001

RFC3081 2001
Rose, M. (Hrsg.): *Mapping the BEEP Core onto TCP*. http://www.ietf.org/rfc/rfc3081.txt. Version: März 2001

RFC3174 2001
Eastlake, D. (Hrsg.) ; Jones, P. (Hrsg.): *US Secure Hash Algorithm 1 (SHA1)*. http://www.ietf.org/rfc/rfc3174.txt. Version: September 2001

RFC3280 2002
Housley, R. (Hrsg.) ; Polk, W. (Hrsg.) ; Ford, W. (Hrsg.) ; Solo, D. (Hrsg.): *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. http://www.ietf.org/rfc/rfc3280.txt. Version: April 2002

RFC4346 2006
Dierks, T. (Hrsg.) ; Rescorla, E. (Hrsg.): *The Transport Layer Security (TLS) Protocol*. `http://www.ietf.org/rfc/rfc4346.txt`. Version: April 2006

RFC4765 2007
Debar, H. (Hrsg.) ; Curry, D. (Hrsg.) ; Feinstein, B. (Hrsg.): *The Intrusion Detection Message Exchange Format (IDMEF)*. `http://www.ietf.org/rfc/rfc4765.txt`. Version: März 2007

RFC4767 2007
Feinstein, B. (Hrsg.) ; Matthews, G. (Hrsg.): *The Intrusion Detection Exchange Protocol (IDXP)*. `http://www.ietf.org/rfc/rfc4767.txt`. Version: März 2007

Riedel u. Mallmann 2006
Riedel, Morris ; Mallmann, Daniel: Standardization Processes of the UNICORE Grid System. In: Volkert, J. (Hrsg.) ; Fahringer, T. (Hrsg.) ; Kranzlmüller, D. (Hrsg.) ; Schreiner, W. (Hrsg.): *Proceedings of 1st Austrian Grid Symposium 2005*, Austrian Computer Society, 2006. – ISBN 3–85404–210–2, S. 191–203

Robertson u. Robertson 2007
Robertson, James ; Robertson, Suzanne: *Volere Requirements Specification Template*. `http://www.volere.co.uk/`. Version: August 2007

Samhain
Samhain Labs (Hrsg.): *Samhain*. `http://www.la-samhna.de/samhain/`

Schiffers 2007
Schiffers, Michael: *Management dynamischer Virtueller Organisationen in Grids*, Ludwig-Maximilians-Universität München, Diss., Juli 2007

Schlamp 2008
Schlamp, Johann: Entwurf und Implementierung eines metrikbasierten Reporting-Systems für IT-Sicherheit / Technische Universität München (TUM). 2008. – Forschungsbericht

Schulter u. a. 2006a
Schulter, Alexandre ; Albuquerque Reis, Julio ; Koch, Fernando ; Becker Westphall, Carlos: A Grid-based Intrusion Detection System. In: *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL)*, 2006

Schulter u. a. 2006b
Schulter, Alexandre ; Navarro, Fabio ; Koch, Fernando ; Becker Westphall, Carlos: Towards Grid-based Intrusion Detection. In: *Proceedings of the 10th Network Operations and Management Symposium (NOMS)*, 2006. – ISBN 1–4244–0142–9, S. 1–4

Secure Audit Logging 2003
Defense Advanced Research Projects Agency (DARPA) (Hrsg.): *Secure Audit Logging for Linux (SAL) – Software Design Document*. http://secureaudit.sourceforge.net/docs/SAL_SDD_1.pdf. Version: Februar 2003

Silva u. a. 2007
Silva, P.F. ; Westphall, C.B. ; Westphall, C.M. ; de Assuncao, M.D.: Design and Evaluation of a Grid Computing Based Architecture for Integrating Heterogeneous IDSs. In: *Proceedings of the Global Telecommunications Conference (GLOBECOM)*, 2007. – ISBN 978–1–4244–1043–9, S. 338–342

Simple Event Correlator
Vaarandi, Risto (Hrsg.): *SEC – simple event correlator*. http://www.estpak.ee/~risto/sec/

SNARE
interSect Alliance (Hrsg.): *SNARE (System iNtrusion Analysis & Reporting Environment)*. http://www.intersectalliance.com/projects/Snare/

Snort
snort.org (Hrsg.): *Snort – The Open Source Network Intrusion Detection System*. http://www.snort.org

SOAP
World Wide Web Consortium (W3C) (Hrsg.): *Simple Object Access Protocol (SOAP)*. http://www.w3.org/TR/soap/

Stoica u. a. 2001
Stoica, Ion ; Morris, Robert ; Karger, David ; Kaashoek, Frans M. ; Balakrishnan, Hari: Chord: A scalable peer-to-peer lookup service for internet applications. In: *Proceedings of the ACM SIGCOMM Conference*, 2001, S. 149–160

Streit u. a. 2005
Streit, A. ; Erwin, D. ; Lippert, Th. ; Mallmann, D. ; Menday, R. ; Rambadt, M. ; Riedel, M. ; Romberg, M. ; Schuller, B. ; Wieder, Ph. ; Grandinetti, L. (Hrsg.): UNICORE – From Project Results to Production Grids. Version: 2005. http://www.unicore.eu/documentation/files/streit-2005-UFP.pdf. Elsevier. – Forschungsbericht. – Elektronische Ressource. – 357–376 S

Tierney u. a. 2000
Tierney, B. ; Aydt, R. ; Gunter, D. ; Smith, W. ; Swany, M. ; Taylor, V. ; Wolski, R.: A Grid Monitoring Architecture / Global Grid Forum (GGF). Version: März 2000. http://www.ogf.org/documents/GFD.7.pdf. – Forschungsbericht. – Elektronische Ressource

Tolba u. a. 2005a
Tolba, Mohamed F. ; Abdel-Wahab, Mohammad S. ; Taha, Ismail A. ; Al-Shishtawy, A. M.: Distributed Intrusion Detection System for Computational

Grids. In: *Proceedings of the 2nd International Conference on Intelligent Computing and Information Systems*, 2005

Tolba u. a. 2005b
Tolba, Mohamed F. ; Abdel-Wahab, Mohammad S. ; Taha, Ismail A. ; Al-Shishtawy, A. M.: GIDA: Toward Enabling Grid Intrusion Detection Systems. In: *Proceedings of the 5th IEEE International Symposium on Cluster Computing and the Grid*, 2005

Tripwire
Tripwire, Inc. (Hrsg.): *Tripwire*. `http://www.tripwire.org`

Tuecke 2001
Tuecke, S.: *Grid Security Infrastructure (GSI) Roadmap*. `http://www.ggf1.nl/abstracts/SEC/draft-ggf-gsi-roadmap-02.pdf`. Version: Februar 2001

UNICORE
UNICORE Forum e.V. (Hrsg.): *UNICORE (Uniform Interface to Computing Resources)*. `http://www.unicore.eu/`

VO Membership Service
European DataGrid Project (Hrsg.): *Virtual Organization Membership Service*. `http://edg-wp2.web.cern.ch/edg-wp2/security/voms/voms.html`

von Laszewski u. Amin 2004
Kapitel 5. In: von Laszewski, Gregor ; Amin, Kaizar: *Grid Middleware*. John Wiley and Sons, 2004. – ISBN 0–470–86206–8

WebServices 2004
Booth, David (Hrsg.) ; Haas, Hugo (Hrsg.) ; McCabe, Francis (Hrsg.) ; Newcomer, Eric (Hrsg.) ; Champion, Michael (Hrsg.) ; Ferris, Chris (Hrsg.) ; Orchard, David (Hrsg.): *Web Services Architecture*. `http://www.w3.org/TR/ws-arch/`. Version: Februar 2004

X.509 2000
International Telecommunication Union (ITU-T) (Hrsg.): *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*. `http://www.itu.int/rec/T-REC-X.509/en`. Version: März 2000

XSL 2006
W3C – World Wide Web Consortium (Hrsg.): *Extensible Stylesheet Language (XSL)*. `http://www.w3.org/TR/xsl/`. Version: Dezember 2006

XSLT 1999
W3C – World Wide Web Consortium (Hrsg.): *Extensible Stylesheet Language Transformation (XSLT)*. `http://www.w3.org/TR/xslt/`. Version: November 1999

Yegneswaran u. a. 2004
Yegneswaran, Vinod ; Barford, Paul ; Jha, Somesh: Global Intrusion Detection in the DOMINO Overlay System. In: *Proceedings of the 11th Network and Distributed System Security Symposium (NDSS)*, 2004

Zhang u. Sun 2006
Zhang, Guiling ; Sun, Jizhou: Grid intrusion detection based on soft computing by modeling real-user's normal behaviors. In: *Proceedings of the Granular Computing (GrC) – IEEE International Conference*, 2006. – ISBN 1–4244–0134–8, S. 558–561

Zhou u. a. 2005
Zhou, Chenfeng V. ; Karunasekera, Shanika ; Leckie, Christopher: A Peer-to-Peer Collaborative Intrusion Detection System. In: *Proceedings of the IEEE International Conference on Networks (ICON)*, 2005, S. 118–123

Zhou u. a. 2007
Zhou, Chenfeng V. ; Karunasekera, Shanika ; Leckie, Christopher: Evaluation of a Decentralized Architecture for Large Scale Collaborative Intrusion Detection. In: *Proceedings of the 10th IEEE International Symposium on Integrated Network Management (IM)*, 2007, S. 80–89

Index

Abkürzungsverzeichnis

AAI	Authentifizierungs- und AutorisierungsInfrastruktur
ACM	Association for Computing Machinery
BEEP	Blocks Extensible Exchange Protocol
BLD	Block List Database
BMBF	BundesMinisterium für Bildung und Forschung
BoA	Beating off Agents
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web Services
BPML	Business Process Modelling Language
CA	Certification Authority Communication Agent
CC	Common Criteria
CERN	European Organization for Nuclear Research
CERT	Computer Emergency Response Team
CoA	Communicator Agents
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CVS	Concurrent Versions System
DAI	Data Access & Integration
DDoS	Distributed Denial-of-Service
DGI	D-Grid Initiative

DHT	Distributed Hash Table
DMM	Decision-Making Module
DN	Distinguished Name Detection Node
DOMINO	Distributed Overlay for Monitoring InterNet Outbreaks
DoS	Denial-of-Service
EDG	European DataGrid
EGEE	Enabling Grids for E-sciencE
FF	Flow Files
FIFO	First-In-First-Out
FTP	File Transfer Protocol
GACG	German Astronomy Community Grid
GGF	Global Grid Forum
GHIDS	Grid-specific Host-based Intrusion Detection System
GIDA	Grid Intrusion Detection Architecture
GIDIA	Grid Intrusion Detection Based on Immune Agent
GIDS	Grid-based Intrusion Detection System
GMA	Grid Monitoring Architecture
GOC	Grid Operations Center
GPA	Grid Protocol Architecture
GRAM	Grid Resource Allocation & Management
GridPMA	Grid Policy Management Authority
GSI	Grid Security Infrastructure
GSS-API	Generic Security Service Application Programming Interface
GT	Globus Toolkit
HIDS	Hostbasiertes Intrusion Detection System
HPC	High Performance Computing
HTTP	Hypertext Transfer Protocol

IDA	Intrusion Detection Agent
IDMEF	Intrusion Detection Message Exchange Format
IdP	Identity Provider
IDS	Intrusion Detection System
IDWG	Intrusion Detection Working Group
IDXP	Intrusion Detection Exchange Protocol
IEDB	Intrusion Evidence Database
IETF	Internet Engineering Task Force
IGTF	International Grid Trust Federation
IP	Internet Protocol
IPS	Intrusion Prevention Systemen
IRS	Intrusion Response Systemen
ITIL	IT Infrastructure Library
LarSID	Large Scale Intrusion Detection Framework
LCG	LHC Computing Grid
LHC	Large Hadron Collider
LinuxBSM	Linux Basic Security Module Project
LRZ	Leibniz-Rechenzentrum
LVQ	Learning Vector Quantization
MD5	Message-Digest Algorithm 5
MDS	Monitoring and Discovery Services
MoA	Monitoring Agents
MWN	Münchener Wissenschaftsnetz
NAT	Network Address Translation
NJS	Network Job Supervisor
NMU	Network Management Unit
OGSA	Open Grid Services Architecture

P2P	Peer-to-Peer
PBS	Portable Batch System
PGIDS	Performance-based Grid Intrusion Detection System
PKI	Public Key Infrastructure
RDL	Resource Description Language
RFT	Reliable File Transfer
RM	Response Module
RRZN	Regionales Rechenzentrum für Niedersachsen (in Hannover)
SA	Sniffing Agent
SAL	Secure Audit Logging for Linux
SAML	Security Assertion Markup Language
SCGIDA	Grid intrusion detection agents based on soft computing
SCGIDS	Grid intrusion detection based on soft computing
SCT	Security Context Token
SEC	Simple Event Correlator
SGE	Sun Grid Engine
SHA-1	Secure Hash Algorithm
SIDB	Signature Identification Database
SLA	Service Level Agreement
SMA	Signature Match Agent
SMS	Short Message Service
SOAP	Simple Object Access Protocol
SPA	Self Protection Agent
SSH	Secure SHell
SSL	Secure Sockets Layer
SSO	Single Sign-On
TBRA	Trace-Back and Response Agent

TC	Trust Communities
TCP	Transmission Control Protocol
TI	Trust Index
TLS	Transport Layer Security
UBMA	User Behavior Model Archive
UBMPDB	User Behavior Model Parameter DataBase
UDP	User Datagram Protocol
UML	Unified Modeling Language
UNICORE	UNiform Interface to COmputing REsources
VO	Virtuelle Organisation
VOMS	Virtual Organization Membership Service
VPN	Virtuelles Privates Netz
WAYF	Where-are-you-from
WfMC	Workflow Management Coalition
WFMS	Workflow Management System
WSDL	Web Services Description Language
WSFL	Web Services Flow Language
WSS	Web Services Security
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformation